Курс

по службе каталога

ALD Pro

для 2.4.0+
от 12.05.2025



Оглавление

Часть 1. ASTRA LINUX **30**

Сравнение Windows vs Linux **30**

Модуль 1. Введение в операционные системы **34**

1.1 Введение . . . 34

1.2 Операционная система . . . 35

1.3 История появления Linux . . . 39

1.4 Операционная система Astra Linux . . . 41

1.4.1 Версии Astra Linux . . . 41

1.4.2 Репозитории Astra Linux . . . 43

1.4.3 Уровни защищенности ALSE . . . 43

1.4.4 Программа Ready for Astra . . . 44

1.4.5 Экосистема Астры . . . 44

1.4.6 Лицензирование Astra Linux . . . 46

1.5 Практика и тестирование . . . 47

1.5.1 Тест: Модуль 1. Введение в операционные системы . . . 47

1.6 Заключение . . . 50

1.7 Дополнительные источники информации . . . 51

1.8 Обратная связь . . . 51

Модуль 2. Установка ОС Astra Linux **51**

2.1 Введение . . . 51

2.2 Требования для установки ОС Astra Linux SE 1.7.6 . . . 51

2.3 Порядок установки Astra Linux . . . 52

2.3.1 Шаг 1. Выбор графического или текстового режима установки . . . 52

2.3.2 Шаг 2. Ознакомление с лицензионным соглашением и его принятие . . . 53

2.3.3 Шаг 3. Настройка клавиатуры . . . 54

2.3.4 Шаг 4. Поиск и монтирование CD-ROM (автоматически) . . . 56

2.3.5 Шаг 5. Загрузка компонентов программы установки с компакт-диска (автоматически) . . . 57

2.3.6 Шаг 6. Определение сетевой карты (автоматически) . . . 58

2.3.7 Шаг 7. Настройка сети . . . 58
2.3.8 Шаг 8. Настройка учетных записей пользователей и паролей . . . 60
2.3.9 Шаг 9. Настройка времени. Выбор часового пояса. . . . 62
2.3.10 Шаг 10. Определение дисков и запуск разметки (автоматически) . . . 64
2.3.11 Шаг 11. Разметка дисков . . . 64
2.3.12 Шаг 12. Установка базовой системы. . . . 70
2.3.13 Шаг 13. Настройка менеджера пакетов (автоматически) . . . 73
2.3.14 Шаг 14. Выбор и установка дополнительного ПО . . . 73
2.3.15 Шаг 15. Дополнительные настройки ОС . . . 76
2.3.16 Шаг 16. Установка системного загрузчика GRUB . . . 78
2.3.17 Шаг 17. Завершение установки . . . 83
2.4 Первоначальная настройка Astra Linux . . . 86
2.5 Практика и тестирование . . . 95
2.5.1 Требования, правила и цели выполнения практической работы . . . 95
2.5.2 Практическая работа: Модуль 2. Установка ОС Astra Linux . . . 96
2.5.3 Тест: Модуль 2. Установка ОС Astra Linux . . . 103
2.6 Заключение . . . 105
2.7 Дополнительные источники информации . . . 105
2.8 Обратная связь . . . 105

Модуль 3. Работа с графическим интерфейсом Fly 105
3.1 Введение . . . 105
3.2 Вход в систему . . . 105
3.3 Рабочий стол . . . 109
3.4 Обзор основных графических утилит . . . 113
3.4.1 Менеджер файлов . . . 113
3.4.2 Рабочий стол и меню «Пуск» . . . 114
3.4.3 Системный монитор . . . 114
3.4.4 Панель управления . . . 116
3.4.5 Прочие утилиты . . . 118
3.5 Практика и тестирование . . . 123
3.5.1 Практическая работа: Модуль 3. Работа с графическим интерфейсом Fly . 123
3.5.2 Тест: Модуль 3. Работа с графическим интерфейсом Fly . . . 127
3.6 Заключение . . . 129
3.7 Дополнительные источники информации . . . 129
3.8 Обратная связь . . . 129

Модуль 4. Работа с терминалом и оболочкой Bash 129
4.1 Введение . . . 129

4.2 Графический интерфейс в современных системах . . . 129
4.3 Аппаратный терминал . . . 130
4.4 Системная и виртуальные консоли . . . 131
4.5 Эмулятор терминала . . . 137
4.6 Командная оболочка Bash . . . 141
4.6.1 Командные оболочки . . . 141
4.6.2 Приглашение к вводу команд . . . 142
4.6.3 Структура и синтаксис команд . . . 142
4.6.4 Несколько базовых команд Bash . . . 144
4.6.5 Окружение пользователя . . . 145
4.6.6 Переменные в Bash . . . 146
4.6.7 Действия над переменными . . . 148
4.6.8 Псевдонимы в Bash . . . 150
4.6.9 Автодополнение и редактирование команд . . . 151
4.6.10 Управление курсором в терминале . . . 151
4.6.11 Работа с историей команд Bash . . . 152
4.6.12 Запуск нескольких команд за раз . . . 156
4.6.13 Стандартные потоки и перенаправление ввода/вывода . . . 157
4.6.14 Конвейер . . . 158
4.6.15 Работа с буфером обмена в консоли . . . 159
4.6.16 Форматирование текста в консоли . . . 159
4.6.17 Управление отображением информации . . . 159
4.7 Практика и тестирование . . . 160
4.7.1 Практическая работа: Модуль 4. Работа с терминалом и оболочкой Bash . 160
4.7.2 Тест: Модуль 4. Работа с терминалом и оболочкой Bash . . . 167
4.8 Заключение . . . 170
4.9 Дополнительные источники информации . . . 170
4.10 Обратная связь . . . 170

Модуль 5. Работа со справочными системами 170
5.1 Введение . . . 170
5.2 Справочные системы Linux . . . 171
5.3 Справочник по встроенным командам оболочки Bash . . . 171
5.4 Справочные страницы системы manpages . . . 173
5.4.1 Расположение справочных страниц . . . 174
5.4.2 Просмотр справочных страниц . . . 174
5.4.3 Навигация по справочной странице . . . 174
5.4.4 Разделы справочных страниц . . . 176
5.4.5 Структура справочных страниц . . . 177

5.4.6 Использование утилиты man и связанных с ней утилит apropos, whatis, mandb . . . 178
5.4.7 Поиск информации на справочной странице . . . 181
5.4.8 Использование утилит поиска whereis и which . . . 185
5.5 Справочные страницы системы info . . . 186
5.6 Справка Astra Linux . . . 187
5.7 Практика и тестирование . . . 191
5.7.1 Практическая работа: Модуль 5. Работа со справочными системами . . . 191
5.7.2 Тест: Модуль 5. Работа со справочными системами . . . 194
5.8 Заключение . . . 198
5.9 Дополнительные источники информации . . . 198
5.10 Обратная связь . . . 198

Модуль 6. Работа с файлами и каталогами, FHS **199**
6.1 Введение . . . 199
6.2 Основные отличия в работе с файлами . . . 199
6.2.1 Всё есть файл . . . 199
6.2.2 Единое пространство имен . . . 199
6.2.3 Подход к хранению файлов программ . . . 200
6.2.4 Зависимость от регистра символов . . . 200
6.2.5 Подход к использованию ссылок . . . 200
6.2.6 Прямая косая черта в качестве разделителя . . . 201
6.3 Устройство файловой системы Linux . . . 201
6.3.1 Корневая файловая система и стандарт FHS . . . 201
6.3.2 Организации хранения данных на диске . . . 206
6.3.3 Типы индексных дескрипторов . . . 212
6.3.4 Жесткие и символические ссылки . . . 214
6.3.5 Правила именования файлов и каталогов . . . 218
6.3.6 Расширения файлов . . . 220
6.3.7 Абсолютный и относительный путь . . . 221
6.4 Работа с файлами и каталогами . . . 222
6.4.1 Список основных команд для работы с каталогами . . . 222
6.4.2 Список основных команд для работы с файлами . . . 225
6.5 Практика и тестирование . . . 240
6.5.1 Практическая работа: Модуль 6. Работа с файлами и каталогами, FHS . . . 240
6.5.2 Тест: Модуль 6. Работа с файлами и каталогами, FHS . . . 241
6.6 Заключение . . . 243
6.7 Дополнительные источники информации . . . 244
6.8 Обратная связь . . . 244

Модуль 7. Работа с текстовой информацией **244**

7.1 Введение . . . 244

7.2 Основные команды для просмотра текстовых файлов . . . 244

7.2.1 Утилита cat . . . 245

7.2.2 Утилита tac . . . 247

7.2.3 Утилита less . . . 248

7.2.4 Утилита head . . . 248

7.2.5 Утилита tail . . . 249

7.3 Основные команды для редактирования текстов из командной строки . . . 250

7.3.1 Утилита wc . . . 251

7.3.2 Утилита nl . . . 252

7.3.3 Утилита tee . . . 253

7.3.4 Утилита sort . . . 254

7.3.5 Утилита uniq . . . 257

7.3.6 Утилита cut . . . 261

7.3.7 Утилита tr . . . 262

7.3.8 Утилита xargs . . . 263

7.4 Утилиты grep, sed и awk и регулярные выражения . . . 266

7.4.1 Утилита grep . . . 266

7.4.2 Утилита sed . . . 273

7.4.3 Утилита awk . . . 275

7.5 Текстовые редакторы . . . 277

7.5.1 Текстовый редактор vim . . . 277

7.5.2 Текстовый редактор nano . . . 282

7.5.3 Текстовый редактор mcedit . . . 283

7.5.4 Графический текстовый редактор kate . . . 284

7.6 Практика и тестирование . . . 285

7.6.1 Практическая работа: Модуль 7. Работа с текстовой информацией . . . 285

7.6.2 Тест: Модуль 7. Работа с текстовой информацией . . . 294

7.7 Заключение . . . 297

7.8 Дополнительные источники информации . . . 297

7.9 Обратная связь . . . 298

Модуль 8. Управление пользователями и группами **298**

8.1 Введение . . . 298

8.2 Аутентификация в системе Linux . . . 298

8.3 Идентификаторы SID, UID и GID . . . 300

8.4 Формат хранения информации об учетных записях пользователей и групп . . . 302

8.4.1 Хранение учетных записей пользователей (/etc/passwd) . . . 303

8.4.2 Хранение паролей пользователей. Файл /etc/shadow. 304
8.4.3 Хранение групп (/etc/group) 307
8.4.4 Хранение паролей групп (/etc/gshadow) 308
8.4.5 Изменение файлов /etc/passwd, /etc/group и /etc/shadow 309
8.4.6 Резервные копии файлов /etc/passwd, /etc/group и /etc/shadow 310
8.5 Управление учетными записями пользователей из командной строки 310
8.5.1 Просмотр учетных записей пользователей 310
8.5.2 Создание учетных записей пользователей 311
8.5.3 Изменение учетных записей пользователей 317
8.5.4 Удаление учетных записей пользователей 323
8.6 Управление учетными записями групп из командной строки 325
8.6.1 Просмотр групп 325
8.6.2 Создание учетных записей групп 325
8.6.3 Изменение учетных записей групп 326
8.6.4 Создание пароля группы 326
8.6.5 Удаление учетных записей групп 327
8.7 Анализ активности пользователей 327
8.7.1 Утилиты who и w (what) 328
8.7.2 Утилиты last и lastb 328
8.7.3 Утилита lastlog 328
8.7.4 Утилита faillog 329
8.7.5 Утилита lslogins 329
8.8 Управление пользователями и группами в графическом интерфейсе 330
8.9 Практика и тестирование 331
8.9.1 Практическая работа: Модуль 8. Управление пользователями и группами . 331
8.9.2 Тест: Модуль 8. Управление пользователями и группами 336
8.10 Заключение 339
8.11 Дополнительные источники информации 339
8.12 Обратная связь 339

Модуль 9. Работа с правами доступа, ACL **339**
9.1 Введение 339
9.2 Управление доступом через идентификаторы, ресурсы и разрешения 340
9.3 Модель UGO «пользователь — группа — остальные» 342
9.4 Специальные флаги suid, sgid, sticky 344
9.4.1 Значения специальных флагов к исполняемым файлам 345
9.4.2 Значения специальных флагов к папкам 346
9.5 Понятие umask 348
9.6 Атрибуты файлов 349

9.7 Списки доступа ACL . . . 351
9.8 Сравнение возможностей POSIX и Windows ACL . . . 355
9.9 Практика и тестирование . . . 356
9.9.1 Практическая работа: Модуль 9. Работа с правами доступа, ACL . . . 356
9.9.2 Тест: Модуль 9. Работа с правами доступа, ACL . . . 362
9.10 Заключение . . . 365
9.11 Дополнительные источники информации . . . 366
9.12 Обратная связь . . . 366

Модуль 10. Система аутентификации PAM и управление правами SUDO 366
10.1 Введение . . . 366
10.2 Управление привилегиями в Linux . . . 366
10.2.1 Утилита su . . . 368
10.2.2 Утилита sudo . . . 369
10.3 Практика и тестирование . . . 393
10.3.1 Практическая работа: Модуль 10. Система аутентификации PAM и управление правами SUDO . . . 393
10.3.2 Тест: Модуль 10. Система аутентификации PAM и управление правами SUDO 396
10.4 Заключение . . . 399
10.5 Дополнительные источники информации . . . 399
10.6 Обратная связь . . . 399

Модуль 11. Работа со сценариями (скриптами) оболочки Bash 399
11.1 Введение . . . 399
11.2 Сравнение Bash и CMD/PowerShell . . . 400
11.3 Запуск скриптов . . . 402
11.4 Коды возврата . . . 404
11.5 Локальные переменные . . . 405
11.5.1 Присвоение значений переменной . . . 405
11.5.2 Строки и строковые переменные . . . 406
11.5.3 Числа в Bash . . . 407
11.5.4 Дробные числа и числа с плавающей точкой . . . 409
11.5.5 Массивы . . . 410
11.5.6 Ветвления . . . 411
11.5.7 Циклы . . . 413
11.5.8 Функции . . . 414
11.5.9 Экранирование символов . . . 416
11.5.10 Подстановка команд . . . 418
11.6 Специальные переменные . . . 419

11.6.1 Переменные окружения . . . 419
11.6.2 Позиционные параметры . . . 420
11.7 Отладка скриптов Bash . . . 420
11.8 Эквиваленты команд Bash и PowerShell . . . 425
11.9 Практика и тестирование . . . 425
11.9.1 Практическая работа: Модуль 11. Работа со сценариями (скриптами) оболочки Bash . . . 425
11.9.2 Тест: Модуль 11. Работа со сценариями (скриптами) оболочки Bash . . . 431
11.10 Заключение . . . 434
11.11 Дополнительные источники информации . . . 434
11.12 Обратная связь . . . 434

Модуль 12. Работа с профилями пользователей 435
12.1 Введение . . . 435
12.2 Профили пользователя в Linux . . . 435
12.2.1 Порядок инициализации профиля . . . 435
12.2.2 Профиль рабочего стола fly . . . 441
12.3 Переменные окружения . . . 443
12.3.1 Уровни переменных окружения . . . 443
12.3.2 Команды для работы с переменными окружения . . . 444
12.3.3 Примеры использования переменных окружения . . . 446
12.4 Практика и тестирование . . . 454
12.4.1 Практическая работа: Модуль 12. Работа с профилями пользователей . . . 454
12.4.2 Тест: Модуль 12. Работа с профилями пользователей . . . 458
12.5 Заключение . . . 461
12.6 Дополнительные источники информации . . . 461
12.7 Обратная связь . . . 461

Модуль 13. Процессы, потоки и управление их жизненным циклом 461
13.1 Введение . . . 461
13.2 Процессы и потоки в Linux . . . 462
13.2.1 Процессы в Linux и Windows . . . 462
13.2.2 Потоки в Linux . . . 471
13.3 Сигналы для процессов в Linux . . . 472
13.4 Планировщик задач в Linux и управление приоритетами процессов . . . 475
13.4.1 Планировщик задач . . . 475
13.4.2 Группы процессов . . . 475
13.4.3 Политики планирования . . . 477
13.4.4 Приоритет процессов . . . 477

13.4.5 Задания, интерактивный и фоновый режимы работы процесса 478
13.5 Извлечение информации о процессах 480
13.5.1 Содержимое /proc/PID/ 480
13.5.2 Содержимое /proc 484
13.6 Управление процессами 487
13.6.1 Управление процессами через консольные утилиты 488
13.6.2 Управление через графику в Системном мониторе 495
13.7 Практика и тестирование 498
13.7.1 Практическая работа: Модуль 13. Процессы, потоки и управление их жизненным циклом 498
13.7.2 Тест: Модуль 13. Процессы, потоки и управление их жизненным циклом . 509
13.8 Заключение 512
13.9 Дополнительные источники информации 512
13.10 Обратная связь 512

Модуль 14. Архивирование и сжатие информации **512**
14.1 Введение 512
14.2 Архивация Linux vs Windows 512
14.3 Работа с архивами из графического интерфейса Fly 514
14.4 Работа с архивами из командной строки 516
14.4.1 Архивирование и распаковка утилитой tar 516
14.4.2 Сжатие и восстановление данных утилитами gzip, bzip2, xz 521
14.4.3 Архивирование и сжатие в стиле Windows утилитами zip и 7z 527
14.5 Практика и тестирование 530
14.5.1 Практическая работа: Модуль 14. Архивирование и сжатие информации . 530
14.5.2 Тест: Модуль 14. Архивирование и сжатие информации 533
14.6 Заключение 535
14.7 Дополнительные источники информации 536
14.8 Обратная связь 536

Модуль 15. Управление ПО в Astra Linux **536**
15.1 Введение 536
15.2 Исполняемые файлы ELF 536
15.3 Пакеты приложений и менеджер dpkg 538
15.4 Управление зависимостями и приложения apt, aptitude, synaptic 542
15.5 Управление списком источников 545
15.5.1 Подключение репозиториев Astra Linux 545
15.5.2 Подключение сторонних репозиториев 547
15.6 Управление ПО с помощью APT 550

15.6.1 Утилиты apt и apt-* . . . 550
15.6.2 Менеджер пакетов aptitude . . . 559
15.6.3 Менеджер пакетов Synaptic . . . 563
15.7 Внутреннее устройство deb-пакета . . . 569
15.7.1 Обычные пакеты . . . 569
15.7.2 Специальные типы deb-пакетов . . . 570
15.7.3 Внутреннее устройство deb-пакета . . . 570
15.8 Репозитории Astra Linux . . . 574
15.9 Создание собственных репозиториев . . . 575
15.10 Практика и тестирование . . . 576
15.10.1 Практическая работа: Модуль 15. Управление ПО в Astra Linux . . . 576
15.10.2 Тест: Модуль 15. Управление ПО в Astra Linux . . . 579
15.11 Заключение . . . 582
15.12 Дополнительные источники информации . . . 582
15.13 Обратная связь . . . 583

Модуль 16. Работа с подсистемой печати 583
16.1 Введение . . . 583
16.2 Служба печати в Linux . . . 583
16.2.1 Архитектура CUPS . . . 584
16.2.2 Управление CUPS . . . 591
16.3 Практика и тестирование . . . 600
16.3.1 Практическая работа: Модуль 16. Работа с подсистемой печати . . . 600
16.3.2 Тест: Модуль 16. Работа с подсистемой печати . . . 609
16.4 Заключение . . . 610
16.5 Дополнительные источники информации . . . 611
16.6 Обратная связь . . . 611

Модуль 17. Загрузка системы и управление службами systemd 611
17.1 Введение . . . 611
17.2 Общее представление о процессе загрузки ОС Linux . . . 611
17.3 Базовые системы BIOS и UEFI . . . 613
17.4 Разметки диска MBR и GPT . . . 614
17.5 Загрузчик GRUB2 . . . 616
17.6 Запуск загрузчика . . . 617
17.6.1 Способ загрузки BIOS > MBR > PBR . . . 617
17.6.2 Способ загрузки BIOS > MBR . . . 619
17.6.3 Способ загрузки BIOS > GPT и UEFI > MBR . . . 620
17.6.4 Способ загрузки UEFI > GPT . . . 620

17.6.5 Способ загрузки PXE . . . 621
17.7 Работа загрузчика GRUB2 . . . 621
17.7.1 Работа с меню загрузки . . . 622
17.7.2 Просмотр и редактирование команд запуска . . . 622
17.7.3 Загрузка ядра и RAM-диска . . . 627
17.7.4 Настройка GRUB2 . . . 628
17.8 Запуск ядра ОС . . . 630
17.8.1 Содержимое RAM-диска . . . 631
17.8.2 Запуск первого процесса /sbin/init с PID=0 . . . 632
17.9 Подсистема инициализации и управления службами systemd . . . 632
17.9.1 Модули systemd . . . 634
17.9.2 Цели systemd . . . 636
17.10 Практика и тестирование . . . 662
17.10.1 Практическая работа: Модуль 17. Загрузка системы и управление службами systemd . . . 662
17.10.2 Тест: Модуль 17. Загрузка системы и управление службами systemd . . . 672
17.11 Заключение . . . 675
17.12 Дополнительные источники информации . . . 675
17.13 Обратная связь . . . 676

Модуль 18. Управление устройствами и модулями ядра ОС 676
18.1 Введение . . . 676
18.2 Управление устройствами . . . 676
18.2.1 Псевдофайловая система sysfs . . . 676
18.2.2 Правила udev . . . 677
18.2.3 Просмотр информации об устройствах . . . 688
18.3 Модули ядра операционной системы . . . 690
18.3.1 Размещение модулей ядра . . . 691
18.3.2 Просмотр модулей ядра . . . 692
18.3.3 Управление модулями ядра . . . 693
18.4 Практика и тестирование . . . 699
18.4.1 Практическая работа: Модуль 18. Управление устройствами и модулями ядра ОС . . . 699
18.4.2 Тест: Модуль 18. Управление устройствами и модулями ядра ОС . . . 701
18.5 Заключение . . . 704
18.6 Дополнительные источники информации . . . 704
18.7 Обратная связь . . . 704

Модуль 19. Файловая система в Linux, работа с блочными устройствами и LVM **704**

19.1 Введение . . . 704

19.2 Подсистема хранения данных в Linux . . . 704

19.3 Виртуальная файловая система . . . 708

19.4 Файловая система ext4 . . . 710

19.4.1 Состав группы блоков 0 . . . 714

19.4.2 Состав групп блоков 1-N . . . 715

19.5 Рекомендуемая схема разбиения . . . 716

19.6 Управление дисковыми разделами и файловыми системами . . . 718

19.6.1 Управление дисковыми разделами . . . 718

19.6.2 Управление файловыми системами . . . 730

19.6.3 Графическая утилита GParted . . . 731

19.7 Монтирование файловых систем . . . 736

19.7.1 Временное монтирование . . . 737

19.7.2 Постоянное монтирование . . . 740

19.8 Менеджер логических томов LVM . . . 742

19.8.1 Структура и состав LVM . . . 743

19.8.2 Управление LVM . . . 745

19.9 Типовой сценарий переноса данных на другой носитель . . . 752

19.9.1 Описание сценария . . . 752

19.9.2 Перенос данных на внешнюю ФС . . . 753

19.10 Практика и тестирование . . . 757

19.10.1 Практическая работа: Модуль 19. Файловая система в Linux, работа с блочными устройствами и LVM . . . 757

19.10.2 Тест: Модуль 19. Файловая система в Linux, работа с блочными устройствами и LVM . . . 759

19.11 Заключение . . . 762

19.12 Дополнительные источники информации . . . 762

19.13 Обратная связь . . . 762

Модуль 20. Сеть и стек TCP/IP, управление сетевыми интерфейсами **762**

20.1 Введение . . . 762

20.2 Основы компьютерных сетей . . . 763

20.2.1 История вопроса . . . 763

20.2.2 Объединяем два компьютера «точка-точка» . . . 763

20.2.3 Объединяем несколько компьютеров в топологии «звезда» . . . 767

20.2.4 Основы маршрутизации . . . 768

20.3 Стек протоколов TCP/IP . . . 771

20.3.1 Модель TCP/IP vs OSI . . . 771

20.3.2 Уровни модели TCP/IP . . . 772
20.3.3 Протокол TCP в деталях . . . 777
20.3.4 Протокол IP в деталях . . . 786
20.4 Работа сетевого стека в Linux . . . 792
20.4.1 Подготовка стека к работе . . . 794
20.4.2 Физический уровень . . . 809
20.4.3 Уровень очередей . . . 810
20.4.4 Уровень протоколов . . . 812
20.4.5 Механизм сокетов . . . 814
20.5 Практика и тестирование . . . 821
20.5.1 Практическая работа: Модуль 20. Сеть и стек TCP/IP, управление сетевыми интерфейсами . . . 821
20.5.2 Тест: Модуль 20. Сеть и стек TCP/IP, управление сетевыми интерфейсами 831
20.6 Заключение . . . 834
20.7 Дополнительные источники информации . . . 834
20.8 Обратная связь . . . 834

Модуль 21. Работа с журналами и отладка системы 834
21.1 Введение . . . 834
21.2 Работа с журналами и сбор логов . . . 835
21.2.1 Система журналирования syslog . . . 835
21.2.2 Система журналирования syslog-ng . . . 838
21.2.3 Система журналирования systemd-journald . . . 849
21.2.4 Настройка ротации журналов . . . 850
21.2.5 Графическая утилита «Системный журнал» . . . 852
21.2.6 Инструмент astra-create-debug-logs . . . 853
21.3 Сбор информации о работе ОС и ее производительности . . . 853
21.3.1 Информация о процессах . . . 854
21.3.2 Информация о подсистеме хранения . . . 854
21.3.3 Системные вызовы . . . 855
21.3.4 Сеть и еще раз сеть . . . 855
21.3.5 Универсальные программы и наборы утилит . . . 856
21.4 Поиск и устранение неисправностей . . . 859
21.4.1 Подходы к выполнению отладки . . . 859
21.4.2 Неисправности, возникающие на начальных стадиях загрузки . . . 861
21.4.3 Восстановление GRUB . . . 876
21.4.4 Пересоздание образа initrd . . . 878
21.4.5 Утилита badblocks . . . 880
21.4.6 Утилита fsck . . . 881

21.4.7 Неисправности, возникающие на финальных стадиях загрузки 881
21.5 Практика и тестирование 883
21.5.1 Практическая работа: Модуль 21. Работа с журналами и отладка системы 883
21.5.2 Тест: Модуль 21. Работа с журналами и отладка системы 888
21.6 Заключение 893
21.7 Дополнительные источники информации 893
21.8 Обратная связь 893

Часть 2. ALD PRO **894**

Модуль 1. Система ALD Pro и ее компоненты **894**
1.1 Введение 894
1.2 Система ALD Pro 894
1.2.1 Возможности службы каталога 896
1.2.2 Почему FreeIPA, а не Samba AD 900
1.2.3 Система ALD Pro и ее компоненты 910
1.3 Перед развертыванием ALD Pro 916
1.3.1 Минимальные требования к инфраструктуре для развертывания ALD Pro 916
1.3.2 Рекомендации по планированию дополнительных ресурсов службы каталога 917
1.3.3 Рекомендации по именованию домена 922
1.4 Практика и тестирование 924
1.4.1 Тест: Модуль 1. Система ALD Pro и ее компоненты 924
1.5 Заключение 927
1.6 Дополнительные источники информации 927
1.7 Обратная связь 927

Модуль 2. Развертывание ALD Pro **927**
2.1 Введение 927
2.2 Подготовка виртуальной среды для прохождения этого курса 927
2.2.1 Описания тестового стенда 927
2.2.2 Минимальные требования для тестовой среды 928
2.2.3 Настройка натированной сети 929
2.3 Развертывание первого контроллера домена 931
2.3.1 Подготовка виртуальной машины 931
2.3.2 Настройка сетевого интерфейса для постоянной работы 943
2.3.3 Настройка репозиториев продукта ALD Pro 946
2.3.4 Настройка имени контроллера домена 950
2.3.5 Установка пакетов контроллера домена 951
2.3.6 Повышение роли сервера до контроллера домена 952

2.3.7 Проверка базовых функций домена . . . 954
2.3.8 Дополнительная настройка DNS-службы . . . 960
2.3.9 Установка глобального каталога . . . 963
2.3.10 Установка модуля синхронизации . . . 964
2.4 Присоединение компьютера к домену . . . 965
2.4.1 Подготовка виртуальной машины . . . 965
2.4.2 Настройка сетевого интерфейса для постоянной работы . . . 967
2.4.3 Настройка репозиториев продукта ALD Pro . . . 968
2.4.4 Установка пакетов клиентской части . . . 968
2.4.5 Проверка имени хоста на уникальность . . . 969
2.4.6 Консольная утилита aldpro-client-installer . . . 969
2.4.7 Графическая утилита . . . 974
2.4.8 Повышение безопасности при вводе компьютеров в домен . . . 983
2.5 Практика и тестирование . . . 989
2.5.1 Требования, правила и цели выполнения практической работы . . . 989
2.5.2 Практическая работа: Модуль 2. Развертывание ALD Pro . . . 990
2.5.3 Тест: Модуль 2. Развертывание ALD Pro . . . 1027
2.6 Заключение . . . 1030
2.7 Дополнительные источники информации . . . 1030
2.8 Обратная связь . . . 1030

Модуль 3. Базовые функции и настройка ALD Pro **1031**
3.1 Введение . . . 1031
3.2 Портал управления ALD Pro . . . 1031
3.2.1 Архитектура портала управления . . . 1031
3.2.2 Вход в портал управления . . . 1033
3.2.3 Интерфейс портала управления . . . 1037
3.2.4 Управление организационной структурой . . . 1044
3.2.5 Управление пользователями . . . 1048
3.2.6 Управление компьютерами . . . 1051
3.2.7 Удаленный доступ для оказания поддержки . . . 1052
3.2.8 Управление группами . . . 1054
3.3 Служба синхронизации времени . . . 1056
3.3.1 Механизм синхронизации времени в ALD Pro . . . 1056
3.3.2 Протокол синхронизации времени . . . 1057
3.3.3 Пакет chrony и служба chronyd . . . 1060
3.4 Служба разрешения имен . . . 1061
3.4.1 Протокол DNS . . . 1062
3.4.2 Конфигурационные файлы службы разрешения имен . . . 1070

3.4.3 Зоны DNS и параметры зоны 1073
3.4.4 Записи в доменной зоне 1076
3.5 Практика и тестирование 1080
3.5.1 Практическая работа: Модуль 3. Базовые функции и настройка ALD Pro 1080
3.5.2 Тест: Модуль 3. Базовые функции и настройка ALD Pro 1082
3.6 Заключение 1085
3.7 Дополнительные источники информации 1086
3.8 Обратная связь 1086

Модуль 4. Работа службы каталога ALD Pro 1086
4.1 Введение 1086
4.2 Архитектура службы каталога ALD Pro 1086
4.3 Сервер FreeIPA 1088
4.3.1 Управление сервером FreeIPA 1089
4.4 Сервер LDAP 389 Directory Server 1117
4.4.1 Каталог LDAP 1117
4.4.2 Управление LDAP-каталогом 1126
4.5 Сервер MIT Kerberos krb5-kdc 1143
4.6 Сервер Samba 1147
4.7 Идентификаторы безопасности 1149
4.7.1 Диапазоны идентификаторов 1149
4.8 Практика и тестирование 1155
4.8.1 Практическая работа: Модуль 4. Работа службы каталога ALD Pro . . . 1155
4.8.2 Тест: Модуль 4. Работа службы каталога ALD Pro 1160
4.9 Заключение 1163
4.10 Дополнительные источники информации 1163
4.11 Обратная связь 1163

Модуль 5. Работа Linux-компьютера в домене 1164
5.1 Введение 1164
5.2 Служба sssd 1164
5.2.1 Краткий обзор 1164
5.2.2 Архитектура службы sssd 1166
5.2.3 Механизм кэширования службы sssd 1173
5.3 Способы аутентификации в домене 1179
5.3.1 Аутентификация по протоколу LDAP 1180
5.3.2 Аутентификация по протоколу Kerberos 1185
5.3.3 Аутентификация по протоколу NTLM 1200
5.4 Безопасный обмен данными с применением SSL/TLS 1201

5.4.1 Механизм защиты данных по протоколу SSL 1202
5.4.2 Доступ к каталогу по протоколам LDAP+StartTLS и LDAPS 1205
5.5 Учетная запись компьютера в домене 1207
5.6 Работа компьютера в автономном режиме 1210
5.6.1 Автономный вход по кэшу пароля 1210
5.6.2 Автоматическая Kerberos-аутентификация при переходе в онлайн режим 1212
5.7 Динамическое обновление DNS-записей в домене 1213
5.8 Отладка sssd службы и Kerberos 1214
5.8.1 Журналы отладки sssd 1214
5.8.2 Общие рекомендации отладки sssd 1216
5.8.3 Отладка Бэкенда sssd 1217
5.8.4 Устранение неисправностей аутентификации, изменения пароля и контроля доступа 1219
5.9 Расположение значимых файлов 1220
5.10 Практика и тестирование 1221
5.10.1 Практическая работа: Модуль 5. Работа Linux-компьютера в домене . . . 1221
5.10.2 Тест: Модуль 5. Работа Linux-компьютера в домене 1227
5.11 Заключение 1229
5.12 Дополнительные источники информации 1229
5.13 Обратная связь 1229

Модуль 6. Управление политиками ALD Pro 1229
6.1 Введение 1229
6.2 Концепция групповых политик 1230
6.3 Политики паролей 1230
6.3.1 Политики паролей Windows 1231
6.3.2 Политики паролей FreeIPA 1233
6.3.3 Настройка политики паролей 1237
6.3.4 Группы с автоматическим участием 1240
6.4 Политики доступа к узлу (HBAC-правила) 1242
6.4.1 Политика локального входа Windows 1243
6.4.2 Политики доступа к узлу FreeIPA 1244
6.4.3 Настройка политик доступа к узлу 1245
6.5 Политики повышения привилегий (SUDO-правила) 1251
6.5.1 Запуск от имени другого пользователя в Windows 1251
6.5.2 Политики повышения привилегий FreeIPA 1252
6.5.3 Настройка политики повышения привилегий 1254
6.6 Групповые политики 1260
6.6.1 Групповые политики Windows 1260

6.6.2 Групповые политики ALD Pro 1275
6.6.3 Разработка дополнительных параметров 1293
6.6.4 Рекомендации к salt-скриптам 1309
6.7 Управление и настройка ПО с помощью политик ALD Pro 1322
6.7.1 Управление через «Групповые политики» 1323
6.8 LAPS - Управление паролями локальных администраторов 1333
6.8.1 Механизм работы LAPS 1334
6.8.2 Установка решения 1336
6.8.3 Настройка политики LAPS в домене 1338
6.8.4 Просмотр информации LAPS в графическом интерфейсе 1340
6.9 Практика и тестирование 1343
6.9.1 Практическая работа: Модуль 6. Управление политиками ALD Pro 1343
6.9.2 Тест: Модуль 6. Управление политиками ALD Pro 1359
6.10 Заключение 1362
6.11 Дополнительные источники информации 1362
6.12 Обратная связь 1363

Модуль 7. Дополнительные подсистемы ALD Pro **1363**
7.1 Введение 1363
7.2 Процедура развертывания подсистем 1363
7.2.1 Установка ролей в Windows Server 1363
7.2.2 Установка подсистем в ALD Pro 1365
7.3 Подсистема «Общий доступ к файлам» 1369
7.3.1 Архитектура подсистемы 1370
7.3.2 Установка подсистемы 1372
7.3.3 Работа с сервером общего доступа к файлам 1375
7.3.4 Аутентификация на файловом сервере Samba 1383
7.3.5 Авторизация на файловом сервере Samba 1387
7.3.6 Особенности настройки прав доступа к файлам в Samba 1388
7.4 Практика и тестирование 1403
7.4.1 Практическая работа: Модуль 7. Дополнительные подсистемы ALD Pro . .1403
7.4.2 Тест: Модуль 7. Дополнительные подсистемы ALD Pro 1404
7.5 Заключение 1405
7.6 Дополнительные источники информации 1405
7.7 Обратная связь 1405

Модуль 8. Управление топологией в домене **1406**
8.1 Введение 1406
8.2 Повышение роли сервера до контроллера домена 1406

8.2.1 Дополнительные настройки 1407
8.3 Продвижение сервера до контроллера домена с портала управления 1408
8.4 Механизм репликации 1412
8.4.1 Репликация в Windows 1413
8.4.2 Репликация в ALD Pro 1415
8.5 FSMO роли в MS AD 1429
8.6 Работа механизмов автоматического обнаружения сервисов 1431
8.6.1 Автоматическое обнаружение при вводе хоста в домен утилитой ipa-client-install 1431
8.6.2 Автоматическое обнаружение сервисов в SSSD 1433
8.6.3 Автоматическое обнаружение сервисов в библиотеке libkrb5 1436
8.6.4 Автоматическое обнаружение сервисов в клиенте IPA 1439
8.6.5 Автоматическое обнаружение сервисов клиентом LDAP 1440
8.6.6 Автоматическое обнаружение сервисов ALD Pro 1441
8.7 Отладка репликации 1441
8.7.1 Проверка состояния репликации 1441
8.7.2 Просмотр количества изменений по соглашению (ldapsearch) 1447
8.7.3 Просмотр журнала изменений (dbscan) 1447
8.7.4 Просмотр таблицы RUV (dsconf) 1447
8.7.5 Просмотр конфликтов репликации (dsconf) 1448
8.7.6 Повторная инициализация контроллера 1448
8.7.7 Механика работы плагина топологии и односторонние соглашения о репликации 1449
8.8 Практика и тестирование 1450
8.8.1 Практическая работа: Модуль 8. Управление топологией в домене 1450
8.8.2 Тест: Модуль 8. Управление топологией в домене 1456
8.9 Заключение 1458
8.10 Дополнительные источники информации 1458
8.11 Обратная связь 1458

Модуль 9. Миграция и гибридное развертывание с MS AD **1458**
9.1 Введение 1458
9.2 Стратегии миграции 1459
9.2.1 Возможности гибридного развертывания 1459
9.2.2 Типовые сценарии развертывания 1462
9.3 Доверительные отношения 1466
9.3.1 Механизм работы доверительных отношений 1466
9.3.2 Настройка доверительных отношений 1474
9.3.3 Проверка работоспособности доверительных отношений 1480

9.4 Глобальный каталог . . . 1483
9.4.1 Глобальный каталог AD . . . 1483
9.4.2 Глобальный каталог ALD Pro . . . 1483
9.4.3 Установка глобального каталога ALD Pro . . . 1486
9.4.4 Проверка глобального каталога ALD Pro . . . 1486
9.5 Присоединение Windows-машины к домену ALD Pro . . . 1489
9.6 Практика и тестирование . . . 1491
9.6.1 Практическая работа: Модуль 9. Миграция и гибридное развертывание с MS AD . . . 1491
9.6.2 Тест: Модуль 9. Миграция и гибридное развертывание с MS AD . . . 1502
9.7 Заключение . . . 1504
9.8 Дополнительные источники информации . . . 1504
9.9 Обратная связь . . . 1505

Модуль 10. Резервное копирование ALD Pro **1505**
10.1 Введение . . . 1505
10.2 Резервное копирование дисков . . . 1505
10.3 Резервное копирование данных . . . 1508
10.3.1 Контроллеры домена . . . 1508
10.3.2 Подсистема общего доступа к файлам . . . 1512
10.3.3 Подсистема печати . . . 1514
10.3.4 Подсистема репозиториев ПО . . . 1515
10.3.5 Подсистема DHCP . . . 1517
10.3.6 Подсистема PXE . . . 1519
10.3.7 Подсистема мониторинга . . . 1520
10.3.8 Подсистема журналирования . . . 1522
10.3.9 Настройка регулярного резервного копирования через CRON . . . 1524
10.4 Практика и тестирование . . . 1525
10.4.1 Практическая работа: Модуль 10. Резервное копирование ALD Pro . . . 1525
10.4.2 Тест: Модуль 10. Резервное копирование ALD Pro . . . 1526
10.5 Заключение . . . 1528
10.6 Дополнительные источники информации . . . 1528
10.7 Обратная связь . . . 1528

Модуль 11. Дополнительные подсистемы и интеграции **1528**
11.1 Введение . . . 1528
11.2 Изменения в версии ALD Pro 2.4.0 . . . 1528
11.2.1 Команды форсирования . . . 1528
11.3 Подсистема «Репозиторий программного обеспечения» . . . 1529

11.3.1 Репозитории пакетного менеджера APT 1530
11.3.2 Архитектура подсистемы репозиториев 1533
11.3.3 Повышение роли сервера до подсистемы репозитория 1535
11.3.4 Создание репозитория из ISO-файла 1536
11.3.5 Создание репозитория вручную 1539
11.3.6 Изменение опубликованной версии репозитория 1542
11.3.7 Политики установки программного обеспечения 1543
11.3.8 Установка ПО через дополнительные параметры групповых политик . . . 1546
11.4 Подсистема «Служба печати» 1550
11.4.1 Повышение роли сервера до подсистемы службы печати 1550
11.4.2 Архитектура подсистемы службы печати 1551
11.4.3 Добавление нового принтера 1552
11.4.4 Установка принтера с помощью групповой политики 1556
11.5 Подсистема «Мониторинг» 1558
11.6 Подсистема «Аудит» 1561
11.7 Интеграция с приложениями сторонних разработчиков 1563
11.8 Практика и тестирование 1565
11.8.1 Практическая работа: Модуль 11. Дополнительные подсистемы и интеграции 1565
11.8.2 Тест: Модуль 11. Дополнительные подсистемы и интеграции 1567
11.9 Заключение 1569
11.10 Дополнительные источники информации 1569
11.11 Обратная связь 1570

Правила межсетевого экрана 1571

Известные ограничения 1574

Оптимизация для больших инфраструктур 1574

Политика AllowNBToInternet (KB5028166) 1575

Работа Samba RPC на серверах ALSE 1.7.6.UU2 1575

Проверка целостности глобального каталога 1576

Часто задаваемые вопросы (FAQ) 1577

Словарь терминов 1588

Изменения в курсе 1598

Все материалы в PDF-формате 1599

Информация о курсе

Курс посвящен администрированию рабочих мест под управлением ОС Astra Linux в домене ALD Pro. В первой части курса рассматриваются вопросы администрирования системы Linux, когда компьютер используется автономно, а во второй части раскрываются особенности работы системы в составе домена ALD Pro на базе службы каталога FreeIPA.

Материалы разработаны специально для Windows-администраторов, которым нужно сделать быстрый переход на стек Linux, поэтому многие технологии объясняются в сравнении с тем, как они реализованы в Windows. Курс поможет понять разницу между этими двумя системами, а также приобрести базовые навыки администрирования операционных систем Linux в целом и Astra Linux в частности в объеме, которого будет достаточно для самостоятельной поддержки инфраструктуры домена ALD Pro на базе службы каталога FreeIPA.

Представленная информация будет полезна даже в том случае, если в вашей инфраструктуре используется пока еще «ванильная» служба каталога FreeIPA или вообще только служба SSSD в домене Samba AD.

Информация является публичной, вы можете использовать ее для самообучения и подготовки сотрудников. При цитировании материалов ссылка на первоисточник обязательна.

Благодарности

В материалах курса отражены не только общедоступные сведения, но и результаты большого числа исследований, выполненных командой архитекторов, имеющей на полтора десятка человек больше ста лет сисадминского опыта и множество успешных развертываний ИТ-инфраструктуры, в том числе и на сотни тысяч рабочих мест.

Я хотел бы поблагодарить за эту огромную работу весь авторский коллектив в целом и назвать каждого по отдельности. В содержание курса вошли материалы Азата Абдрахманова, Александра Кузнецова, Антона Нечаева, Вадима Фещенко, Владимира Кудрявцева, Игоря Логинова, Ильи Бытева, Ильи Князева, Марата Умарова, Михаила Афанасьева, Руслана Федорова, Сергея Третьякова, Тимура Романова, Эдуарда Саблина, Юрия Баткова.

Особенно хотели бы выделить вклад Павла Ермолаева за лидирующую роль в работе над курсом, Эдуарда Саблина за внедрение инструментов Docs as Code и рефакторинг всех скриптов, Вячеслава Нигай за организацию фокус-групп для апробации материалов курса, Оксаны Беловой за вычитку, а также Александра Дурашкина за консультации по особенностям работы отдельных технологий.

И, конечно же, весь этот проект был бы невозможен без всеобъемлющей поддержки со стороны руководства. Благодарим Алексея Фоменко и Евгения Паутова за помощь в выявлении по-

требностей заказчиков в результатах качественных исследований компонентов используемого технологического стека и выделении ресурсов на формирование группы Research & Integration для выполнения этих работ.

– Анатолий Лысов,
Менеджер продукта ALD Pro

Требования к обучаемым

Наличие опыта администрирования ОС Windows и домена MS Active Directory значительно упростит освоение материалов, но не является обязательным требованием. Все технологии рассматриваются, начиная с основ, поэтому с помощью этого курса можно не только выполнить переквалификацию, но и освоить Linux с нуля, получив широкие представления о том, как отдельные технологии реализованы в Windows.

Объем и продолжительность курса

Курс состоит из двух частей и включает 32 модуля, которые рассчитаны на 64 часа работы с учебным материалом. Еще столько же времени нужно будет уделить выполнению практических заданий.

В каком порядке рекомендуется изучать материалы

Модули курса рекомендуется изучать в порядке их следования, т.к. в своих объяснениях мы довольно часто будем опираться на ту информацию, которая уже была предоставлена ранее.

А что, если...

- Если у вас будут вопросы по обучению, обращайтесь с ними в группу ALD Pro - Learn.
- Если вы захотите рассказать нам о своих впечатлениях, направляйте отзывы на адрес электронной почты info@astralinux.ru.
- Если вы найдете опечатки или захотите помочь нам улучшить качество материалов курса, направляйте свои пожелания в виде запросов на слияние в открытый репозиторий проекта на GitFlic.

Результаты апробации

С материалами курса уже ознакомились более 500 инженеров из компаний наших партнеров и заказчиков. Мы получили много положительных отзывов и смогли учесть замечания и пожелания, которыми с нами поделились коллеги.

Отзыв

Первая часть курса, посвященная базовому администрированию Linux, оказалась не лишняя даже для «не новичка». При этом материал подан максимально доступно и начинается не с абстракций, а с конкретики: загрузка, оборудование, компоненты, службы, работа с процессами и файлами. Все демонстрируется на наглядных примерах.

Вторая часть по работе с ALD Pro затрагивала уже работу с протоколами LDAP(S) и Kerberos, что очень важно при работе с любой службой каталога и IDM, а не только в разрезе ALD Pro.

– Седых Геннадий,
Ведущий инженер СевенПро

Отзыв

Я прошёл обучение по материалам ALD Pro и хочу поделиться своими впечатлениями. Программа сбалансированная, включает в себя ключевые аспекты контроллера домена ALD Pro. А в сообществе всегда можно найти высококвалифицированных специалистов, которые готовы помочь.

Особенно понравилась практическая часть, которая позволяет лучше усвоить материал и научиться применять полученные знания на практике. Рекомендую эти материалы всем, кто хочет получить новые знания и навыки по внедрению и эксплуатации доменов на Linux.

– Максим Ч.,
Руководитель проектного отдела ООО «Перемена»

Отзыв

Для тех, кто не знаком с Linux (и в частости с ALD Pro) данный курс – это отличный трамплин для изучения и понимания базовых технологий Linux-подобных систем. Отличная структура и подача материала. По каждому модулю предусмотрена проверка знаний.

Данный курс помог мне лучше понять Astra Linux, научиться самостоятельно развертывать и администрировать ALD Pro.

– Эдуард Ж.,
Пресейл-инженер Treolan

Отзыв

Мне курс оказался полезен по нескольким причинам. Во-первых, несмотря на то, что я продол-

жительное время работаю в Debian-based операционных системах, первая часть курса закрыла некоторые пробелы в моих знаниях по ним. Во-вторых, одновременно с освоением второй части курса (той, что непосредственно по ALD Pro) получалось применять эти знания сразу в работе.

Даже несмотря на то, что я работаю в разработке ПО, а не в системном администрировании, всё равно возникали задачи, требующие полноценного знания продукта. Могу сказать, что курс с задачей такого освоения справляется на отлично.

– Иван К.,
Ведущий инженер-программист Астра

Отзыв

Понравилось структурирование материала, элементы юмора и описание первопричин текущей ситуации (почему так работает, как так сложилось).

– Вячеслав Н.

Отзыв

Должен отметить, что технические специалисты очень высоко отмечают насыщенность и качество курса. Один даже сказал, что наконец, кажется, начал понимать Линукс.

– Александр Лунин,
Руководитель отдела развития инфраструктуры

Отзыв

Отличный курс. Материал подаётся последовательно, ясным и понятным языком. Описаны все нюансы внедрения и последующего администрирования. Подойдет даже тем, кто не был знаком с Linux в целом. При должном уровне внимания, усердия и практики на выходе можно получить специалиста, способного развернуть и настроить с нуля или импортозаместить имеющиеся решения на крупном предприятии. При возникновении сложностей авторы оперативно и подробно отвечают на любые вопросы.

– Владимир Р.,
Старший инженер Origin Security

Отзыв

Я с нашими инженерами полностью согласна) Очень грамотный, детализированный , глубоко

проработанный курс с отличной обратной связью.

Поразило то, что он оказался очень полезным и специалисту, который до этого времени работал только с Windows и не был знаком с Linux, и второму нашему сотруднику, который уже лет 20 как работает преимущественно с Linux и очень комфортно чувствует себя в этой среде. На удивление, он тоже нашел для себя много интересных и полезных моментов в этом курсе.

Огромное спасибо вашей команде за организацию этого обучения. Надеемся, что это не последний курс и не последнее обучение, а только начало нового интересного направления в экосистеме Астры.

– Софья К.,
Директор по развитию Origin Security

Отзыв

Курс оказался ценным и структурированным материалом для специалистов, знакомящихся с Astra Linux и ALD Pro. Очень понравилось, что материал ориентирован на администраторов с опытом работы в Windows, что упрощает переход к новой системе.

Информация подана с учётом практических аспектов, позволяя сформировать необходимую базу знаний и навыков для самостоятельной поддержки инфраструктуры ALD Pro. Высокая насыщенность материала и доступное объяснение отличий между Linux и Windows делают курс полезным и для начинающих, и для опытных администраторов.

– Вилен М.,
Технический специалист ООО «ВИ ИНТЕГРАЦИЯ»

Отзыв

Для меня, как относительно новичка в Linux (а тем более в Astra Linux) и доменах, курс оказался в большинстве аспектов фундаментальным и однозначно полезным. Хотелось бы, чтобы коллеги собрали воедино все модули и превратили их в небольшой учебник.

– Никита Б.

Отзыв

Курс отличный, охватывает большой объем информации не только по ALD Pro, но также и по самой Астре. Спасибо организаторам за предоставленную возможность поучаствовать. Всем рекомендую к изучению!

– Мирошников Иван,
Руководитель отдела ИТ-проектов ООО «Фриком»

Отзыв

Курс понравился.

Достоинства:

1. *Есть интересные моменты даже для тех, кто не первый год работает с Linux.*
2. *Авторы курса отзывчивые и компетентные люди, которые могут подсказать или уточнить у коллег.*
3. *Курс достаточно подробно разбирает работу самого продукта и его взаимосвязи.*

Недостатки:

4. *Нет занятий по развертыванию DHCP и сетевой установки.*
5. *Слишком много исторической справки, которая никак не поможет вам в работе, но зато сожрет академические часы, которые можно было бы потратить на DHCP и PXE.*

– Кирилл Гусъков,
Ведущий системный администратор ООО «Радиус»

Отзыв

Курс простой и понятный. Подходит для самого начального уровня. Хорошо сбалансированные лабораторные работы. Отлично сформирована программа обучения, понравилось наличие практических заданий и лабораторных работ. Многие изученные вопросы помогут мне в дальнейшей работе и помогли освежить в памяти старые знания.

– Алексей Филинов,
Системный администратор ООО «Радиус»

Отзыв

Курс предоставляет детальное понимание платформы Astra Linux и ALD Pro. В данном курсе решали реальные задачи и кейсы на уровне, необходимом для внедрения и поддержки в производственных условиях. Раскрыто много нюансов развертывания и эксплуатации доменной структуры ALD Pro.

– Никита Хозяинов,
Системный администратор ООО «Рик-ком»

Отзыв

Курс дает фундаментальные понятия платформы Linux, качественно структурирован, снабжен схемами и описаниями механик. Материалы сопровождаются сравнениями и отсылками к OS/AD Microsoft, что помогает администраторам Microsoft адаптироваться к новой для себя платформе.

Спасибо организаторам и преподавателям за глубокую проработку курса и увлекательную подачу материала.

– Намазов Руслан,
ООО «Программное Ателье»

Отзыв

Курс достаточен для получения базовых навыков администрирования ОС Astra Linux и службы каталогов ALD Pro. Процесс обучения удачно сочетает теоретическую и практическую часть, позволяя после успешного прохождения курса, сразу применить навыки в работе. Материалы хорошо написаны и могут служить справочным материалом при повседневной работе.

– Евгений Ли,
ООО «Программное Ателье»

Отзыв

Очень полезный технический курс с минимумом «воды», который позволит не только познакомиться с архитектурой программного комплекса ALD Pro и его компонентами, но и освоить полезные средства для диагностики проблем, возникающих при эксплуатации продукта. Уверен что даже опытные Linux-администраторы откроют для себя что-то новое, поймут свои «слабые» места и смогут укрепить их материалами из курса. Понравилось наличие практических заданий, а также среза знаний (тестов) в каждом модуле.

– Константин К.,
Ведущий инженер Астра

Часть 1. Astra Linux

Первая часть курса посвящена вопросам администрирования автономных рабочих мест вне домена. Однако темы занятий подобраны таким образом, чтобы охватить обязательный минимум, необходимый для обслуживания домена, поэтому успешное изучение первых двадцати модулей является обязательным требованием для обучения по второй части курса.

Сравнение Windows vs Linux

Операционные системы Windows и Linux очень похожи, так как решают одни и те же задачи, работают в общей компьютерной сети и на одном и том же «железе», но вместе с тем они имеют ряд отличий, знание которых поможет вам сделать быстрый старт. В данном разделе мы постарались кратко отразить наиболее важные из них.

Подход к работе с дисками

В системе Windows есть понятие дисков или логических томов, которые обозначаются буквами латинского алфавита (например, A:\, B:\, C:\ и т.д.), а полный путь к файлу указывается, начиная с имени диска (например, `C:\Windows\System32\drivers\etc\hosts`). В Windows есть возможность монтировать диск в пустую папку NTFS, однако эта операция не используется широко.

В системе Linux используется единая иерархия каталогов и файловых систем, а полный путь к файлу указывается от корня файловой системы, который обозначается наклонной чертой «/» (например, `/etc/hosts`). Корень файловой системы физически расположен на системном диске компьютера, а дополнительные диски можно смонтировать в любую точку, что позволяет, например, перенести папку `/home` с рабочими каталогами пользователей на отдельный физический диск.

Подход к хранению файлов программ

В системе Windows служебные файлы операционной системы находятся в каталоге `C:\Windows`, а файлы приложений в каталоге `C:\Program Files`, причем для каждого приложения создается

отдельная папка, в которой хранятся все его файлы, организованные любым способом на усмотрение разработчиков приложения. Для хранения настроек приложений обычно используются файлы `*.ini`, и только приложения продвинутых вендоров используют специальную базу данных, которая называется реестром Windows.

В Linux для организации файлов придерживаются так называемого стандарта иерархии файловых систем (Filesystem Hierarchy Standard, FHS), который унифицирует размещение файлов по их назначению. Например, все исполняемые файлы должны находиться в каталогах `/bin` и `/sbin` (от англ. binary, бинарные файлы), конфигурационные файлы всех установленных приложений находятся в папке `/etc` (от et cetera, прочие), а изменяемые файлы, такие как журналы, всегда можно найти в каталоге `/var` (от англ. variable, изменчивый).

Концепция «все есть файл» vs «все есть объект»

В системе Windows файлы – это просто файлы, а для реализации межпроцессного взаимодействия используются такие технологии, как API, COM, RPC, Pipes и др.

В Linux есть аналоги этих технологий, например, шина D-Bus, но для того, чтобы без разработки специальных интерфейсов предоставить простой унифицированный доступ ко всем возможностям операционной системы, Linux заимствовал у Unix концепцию «все есть файл». Поэтому в иерархии файлов Linux некоторые каталоги являются точками монтирования псевдо-систем или так называемых виртуальных файловых систем.

Например, с помощью команды `cat /proc/meminfo` можно увидеть параметры использования памяти, так как каталог `/proc` является не обычным каталогом, а виртуальной файловой системой, которая предоставляет информацию о системе и процессах.

Еще необычнее для Windows-администраторов могут показаться возможности взаимодействия между приложениями через так называемые сокеты домена Unix или IPC-сокеты (от inter-process communication). Эти сокеты представляют собой специальные файлы на диске, через которые приложения могут устанавливать быстрые соединения, исключающие накладные расходы, связанные с TCP/IP стеком. Например, если в Windows для отправки задания на печать приложения обращаются напрямую к API операционной системы, то в Linux они обращаются к общей службе печати CUPS (Common Unix Printing System) по сокету `/var/run/cups/cups.sock`.

Регистрозависимость в именах файлов

В Windows имена файлов могут содержать как прописные, так и строчные символы, но два файла с именами `readme.txt` и `README.txt` не могут находиться в одном каталоге, т.к. не пройдут проверку на уникальность.

В Linux оба файла вполне могут находиться в одной директории, поэтому довольно сложно заметить проблему, когда мы указываем полный путь к файлу.

```
localadmin@astra:~$ touch readme.txt
localadmin@astra:~$ touch README.txt
localadmin@astra:~$ ls
Desktop   readme.txt  Видео      Загрузки     Музыка
Desktops  README.txt  Документы  Изображения  Общедоступные
```

С этой проблемой очень часто сталкиваются начинающие веб-дизайнеры, которые создают html-страницы на Windows-компьютере и потом долго не могут понять, почему их сайт не работает на Linux-хостинге, как задумано.

Подход к использованию ссылок

Несмотря на то, что команда `mklink` для создания ссылок появилась в системе Windows еще со времен NT, большинство пользователей Windows ограничиваются использованием ярлыков, которые представляют собой обычные файлы с расширением `*.lnk`, содержащие настройки для быстрого доступа к файлам и папкам. Аналогом lnk-файлов в Linux являются файлы `*.desktop`.

В системе Linux жесткие (hard link) и мягкие ссылки (symbolic link) получили широкое распространение, поэтому изменения в одном файле могут отражаться сразу в нескольких местах, что сначала может сбивать с толку.

Например, с помощью команд `stat /run/cups/cups.sock` и `stat /var/run/cups/cups.sock` вы можете убедиться, что эти файлы указывают на inode с одним и тем же номером, а все потому, что папка /var/run является мягкой ссылкой на папку `/run`, что было сделано для обратной совместимости. Проверьте это командой `ls -l /var/run`.

Еще интереснее, когда на один и тот же файл указывают сразу несколько жестких ссылок. Например, с помощью команд `stat /usr/bin/gunzip` и `stat /usr/bin/uncompress` вы можете легко убедиться, что это действительно две разные жесткие ссылки, которые указывают на один и тот же inode, поэтому до тех пор, пока мы не удалим обе ссылки, этот файл будет физически оставаться на диске.

Подход к распространению программного обеспечения

Для установки Windows-приложений администраторам обычно достаточно одного файла `*.exe` или `*.msi`, и мы очень редко сталкивается с зависимостями, когда перед установкой приложения нужно поставить что-то еще, например, .NET Framework, Java, DirectX, медиа-кодеки. Хорошим тоном считается, когда Windows-разработчики добавляют в свой установочный пакет все необходимые библиотеки, и администратор устанавливает несколько приложений подряд, просто кликая «далее > далее > далее».

В Linux концепция повторного использования программного кода доведена до абсолюта, поэтому одни и те же компоненты могут использоваться повторно во множестве разных приложений. Это позволяет сократить время разработки и улучшить качество программного обеспечения, но значительно усложняет процесс установки и обновления приложений.

По указанной причине Linux-приложения распространяются в виде пакетов, которые размещаются в специальных репозиториях. В настройках пакетов указываются ссылки на другие пакеты, которые также необходимы для корректной работы приложения, а для установки приложений используются специальные пакетные менеджеры, которые умеют обрабатывать эти зависимости.

В разных дистрибутивах Linux используются разные пакетные менеджеры. Например, Astra Linux основана на Debian, поэтому для установки приложений используются менеджеры dpkg (*от англ. Debian Package*) и apt (*от англ. Advanced Packaging Tool*).

Официальные репозитории операционной системы доступны по адресу https://download.astralinux.ru, но так как индексация каталогов внутри репозитория запрещена, то просматривать вручную эти каталоги не получится, и с ними нужно работать через менеджер пакетов apt, см. *Модуль 15. Управление ПО в Astra Linux*.

Подход к назначению прав доступа на файлы и папки

В системе Windows с незапамятных времен используется файловая система NTFS, которая позволяет назначать права доступа с помощью так называемых списков контроля доступа (Access Control Lists, ACL), поэтому мы без проблем можем назначить права доступа любому количеству пользователей и групп. Данная модель является очень гибкой и позволяет решить любые задачи бизнеса в части распределения прав доступа.

В системе Linux изначально была реализована упрощенная модель «владелец-группа-остальные» (User-Group-Others, UGO), которая обеспечивала простой и безопасный способ управления правами, но не обладала гибкостью ACL. В дальнейшем эта модель была расширена возможностями POSIX ACL, и современные Linux дистрибутивы поставляются уже с предустановленными пакетами, реализующими эти функции. Однако, концепция UGO осталась, по сути, прежней, поэтому любому уважающему себя администратору нужно хорошо ориентироваться в том, что означают числа 755, 644 и почему за 777 может потребоваться писать объяснительную. Этой теме обязательно будет посвящено одно из наших следующих занятий.

Роль суперпользователя в системе

В системе Windows учетная запись суперпользователя называется Administrator и имеет идентификатор безопасности SID, заканчивающийся на 500, но жесткой привязки к идентификатору нет, и эти привилегии можно назначить любому другому пользователю. Службы Windows работают от различных системных учетных записей, например, System, Local Service, Network Service.

В Linux суперпользователь имеет имя root и POSIX-идентификатор 0, который до версии ядра 2.2 использовался в качестве исключения во всех проверках прав доступа. В современных системах Linux полномочия суперпользователя разделены на блоки, которые называются привилегиями (*от англ. capabilities*), и так же, как в Windows, могут быть назначены любому пользователю, см. справку на утилиту `capsh` (*от capability shell*).

Отношение к графическим интерфейсам

Разработчики Windows изначально делали ставку на графический интерфейс, чтобы сделать свою систему доступной для широкого круга пользователей, поэтому администраторы долгое время избегали работу в командной строке. С появлением PowerShell ситуация стала меняться, но количество установок Windows Server Core без графического интерфейса все еще крайне незначительно.

В системе Linux первый графический интерфейс появился примерно в те же годы, но целевой аудиторией операционной системы долгое время были продвинутые пользователи, для которых командная строка была более привычной, поэтому для многих зрелых Linux-продуктов графические инструменты либо отсутствуют, либо предоставляют доступ только к части возможностей продукта.

В общем, будьте готовы к тому, что для успешного внедрения Linux-решений на предприятии вам потребуется значительно повысить свою квалификацию в части работы с командной строкой.

Отличий Windows и Linux, конечно же, намного больше, но этого минимума будет уже достаточно для того, чтобы перейти к углубленному изучению операционной системы Linux.

Модуль 1. Введение в операционные системы

1.1 Введение

В этом модуле вы узнаете об истории возникновения GNU/Linux и базовых принципах работы ОС, с которыми может быть знаком не каждый специалист, обслуживающий Windows системы. Мы познакомимся с операционной системой Astra Linux и узнаем о ее версиях, уровнях защищенности, вариантах лицензирования, а также посмотрим, из каких компонентов состоит экосистема Астры.

Специальные обозначения в курсе

В тексте курса будут приводиться примеры выполнения команд с результатом вывода. Эта информация будет полезна для понимания материала, когда под рукой нет доступа к стенду:

```
admin@astra:~$ ls
Desktop  Desktops  SystemWallpapers  Видео  Документы ...
```

Обратите внимание, что текст до символа доллар `$` является «приглашением для ввода», и его не нужно копировать, когда вы захотите проверить работу команды у себя в терминале. В случае, если вы будете работать из-под суперпользователя, текст приглашения будет заканчиваться символом решетки `#`, например, в следующем примере команда `ls` выполняется из-под root на компьютере с именем astra:

```
root@astra:~# ls
Desktop
```

В тексте могут встречаться шаблоны команд. В этом случае мы используем скобки `<>`, внутрь которых требуется подставить необходимое значение. Например, для получения справочной информации на любую утилиту требуется выполнить команду `man <имя_утилиты>`.

Горячие сочетания клавиш обозначаются в виде кнопки, например, для поиска команд из истории нажмите в терминале Ctrl + R.

1.2 Операционная система

Операционная система (ОС) – это базовое системное программное обеспечение, которое управляет аппаратными ресурсами вычислительного узла, выступая в роли посредника для прикладного программного обеспечения.

От эффективности работы механизмов операционной системы напрямую зависит производительность, надежность и безопасность работы прикладного программного обеспечения. Ниже на рисунке представлена схема взаимодействия ОС и её окружения.

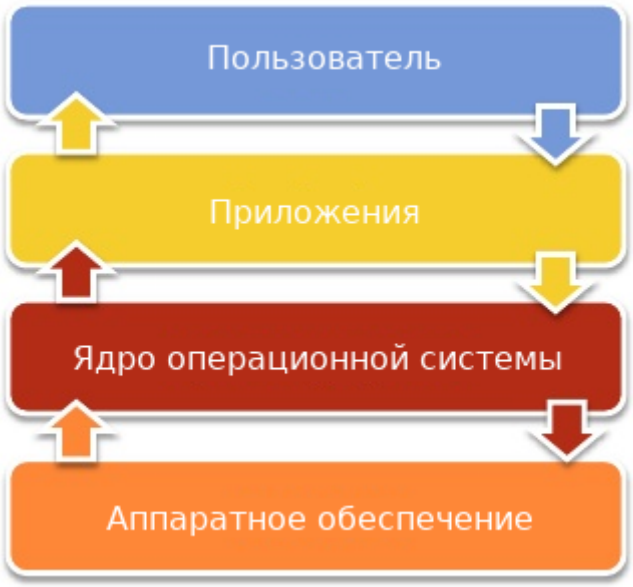


рис. 1.1 – Схема взаимодействия ОС и окружения

Важно: Особенность операционной системы заключается в том, что вы никогда не должны ее замечать, потому что в действительности мы используем не операционную систему, а программы на своем компьютере, и в жизни операционной системы есть только одна задача — это помочь данным программам работать.

Другими словами, операционная система никогда не делает что-то сама по своей инициативе, она лишь только ждёт, пока программы не попросят ее предоставить доступ к каким-либо ресурсам, например, файлу на диске, или выход в интернет, и тогда в игру вступает операционная система, которая упрощает разработчикам написание прикладных программ.

– **Линус Торвальдс**

Современная ОС должна обладать следующими возможностями:

- **Управление процессами** – создание, завершение и блокировка процессов, каждый из которых представляет собой исполняемый экземпляр какой-то компьютерной программы.

  В системе Windows для управления процессами есть приложение «Диспетчер задач», в Astra Linux доступно сопоставимое по функциональности приложение «Системный монитор» `ksysguard`, а в базовых дистрибутивах Linux из командной строки всегда можно воспользоваться утилитами `ps` и `top`.

- **Управление памятью** – обработка запросов на выделение и освобождение оперативной памяти, за что отвечает так называемый Менеджер памяти.

- **Управление вводом-выводом** – взаимодействие операционной системы с внешним миром, в роли которого может выступать как пользователь, так и любая другая система обработки информации.

- **Управление прерываниями** – обработка специальных сигналов от программного или аппаратного обеспечения. Когда процессор получает такой сигнал, он приостанавливает выполнение текущей программы и передает управление обработчику прерываний (ISR - Interrupt Service Routine). Обработчик отрабатывает событие, например, нажатие кнопки на клавиатуре. Как правило, после выполнения необходимых действий, обработчик прерываний возвращает управление прерванной программе.

- **Управление файлами** – создание, переименование, редактирование, копирование, перемещение, удаление и поиск файлов.

- **Управление периферийными устройствами** – обеспечение работы с клавиатурами, мышками, принтерами и другими устройствами через так называемые драйвера.

- **Поддержка компьютерных сетей TCP/IP** – обеспечение обмена данными между вычислительными устройствами (например, компьютерами, серверами, маршрутизаторами и т.д.) через сети, такие как локальные сети (LAN) или интернет.

- **Защита компьютера** – обеспечение безопасности за счет реализации функций аутентификации, авторизации, межсетевого экранирования и др.

Современные операционные системы, такие как Windows и Linux, состоят из следующих компонентов:

- **Ядро** – это основной компонент операционной системы, который предоставляет приложениям доступ к аппаратным ресурсам компьютера (таким как процессорное время, память, файлы, компьютерная сеть, внешние устройства ввода/вывода и другому аппаратному обеспечению) через обращение к системным вызовам и обеспечивает взаимодействие приложений между собой за счет механизмов межпроцессного взаимодействия.

  В операционных системах Windows код ядра находится в файле `C:\Windows\System32\ntoskrnl.exe`, а в операционной системе Linux ядро может находиться, например, в файле `/boot/vmlinuz-5.15.0-33-generic`. Разработчики дистрибутивов Linux поставляют их, как правило, со своими собственными сборками ядра, которые скомпилированы с определенным набором параметров и содержат необходимые драйвера устройств, но от этого ядро не перестает быть ядром Linux, поэтому мы можем, например, запускать контейнеры Red Hat на Debian и наоборот.

- **Системные библиотеки** – это специальные функции, с помощью которых прикладные программы и утилиты могут выполнять системные вызовы и получать доступ к функциям ядра

операционной системы. Существуют статические и динамические библиотеки.

- **Статическая библиотека** — это файл с машинным кодом, который при сборке приложения полностью включается в исполняемый файл.

- **Динамическая библиотека** — это файл, программный код которого не включается в исполняемый файл, а загружается в память только при запуске приложения, то есть динамически. Код динамических библиотек может использоваться несколькими приложениями, поэтому их называют так же разделяемыми объектами (shared object, so). Основной динамической библиотекой для языка Си, с помощью которой можно получить доступ к системным вызовам Linux, является библиотека `libc.so`.

- **Оболочка с системными утилитами** – набор служебных программ, с помощью которых можно выполнять специализированные задачи по управлению операционной системой и оборудованием. Например, с помощью утилиты `ls` (от англ. **list**, список) можно просматривать информацию об объектах файловой системы.

  Значительная часть системных утилит, используемых совместно с ядром Linux, изначально была заимствована из проекта GNU, поэтому первые сборки даже называли GNU/Linux. В дальнейшем разработчики дистрибутивов стали развивать эти инструменты и компоновать их разным образом, поэтому появились Red Hat Linux, Debian Linux, Ubuntu Linux и другие. То есть дистрибутивы Linux различаются между собой в первую очередь набором системных утилит, с которыми они поставляются из коробки.

Для обеспечения безопасности операционные системы не только помогают приложениям получить доступ к аппаратным ресурсам, но еще и ограничивают их в возможности обращаться к этим ресурсам напрямую. Приложения работают как бы в песочнице и без содействия ядра не могут выделить себе память, открыть файл, послать сетевой пакет или вывести на экран строчку текста. Поэтому утечки памяти не могут пережить процесс, который их породил, а при завершении приложения будут закрыты все файлы, которые были открыты во время его работы. Именно поэтому современные Windows/Linux системы могут иметь сотни часов аптайма, в отличие от Windows 95/98, в которых для обеспечения обратной совместимости некоторые ресурсы оставили доступными напрямую, что негативно сказалось на их стабильности.

Концепция информационной безопасности и функциональной отказоустойчивости операционных систем на основе уровней привилегий получила название колец защиты, а поддержка этих функций на аппаратном уровне называется режимами работы процессора. В системе Multics, которая являлась прародителем Unix, таких колец было восемь, но в микропроцессорах x86 была реализована поддержка только четырех колец, а в современных операционных системах Windows/Linux так вообще используется только два из них: ядро работает на нулевом уровне, а пользовательские приложения на третьем уровне, см. рис. 1.2.

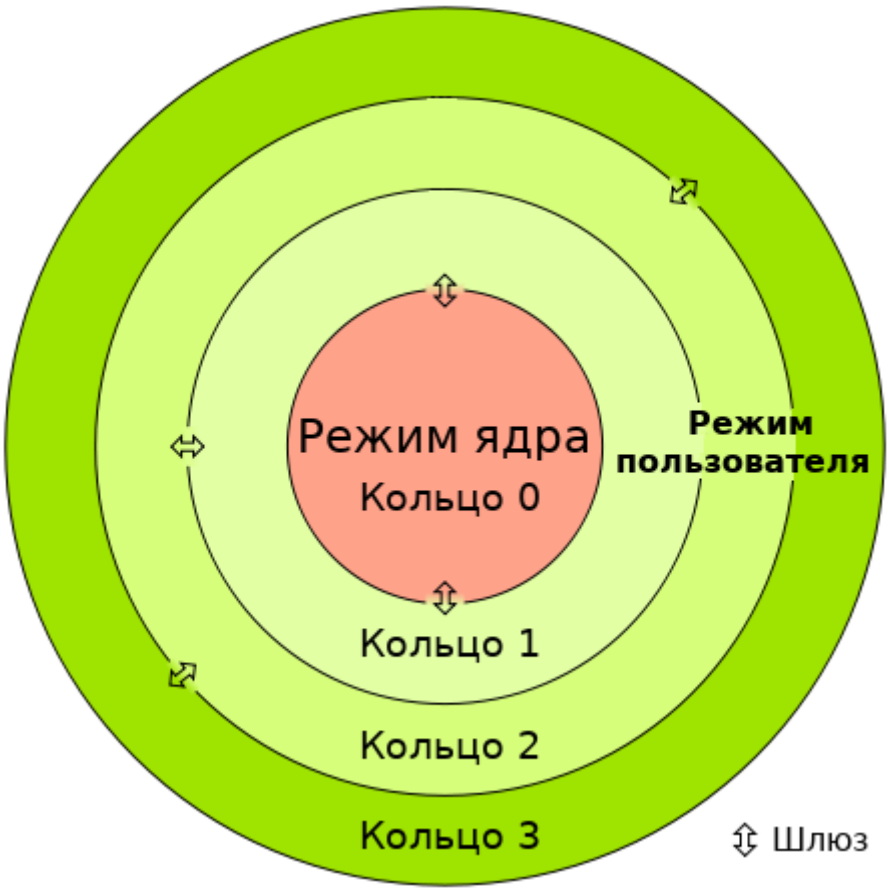


рис. 1.2 – Кольца защиты

На уровне системного администрирования мы никак не влияем на работу колец защиты, ядро операционной системы само ведет учет активных процессов, какие из них принадлежат ядру, а какие запущены пользователями. Для общего понимания достаточно знать, что каждый квант процессорного времени планировщик берет из списка следующий процесс, готовый к исполнению, и формирует команду на переключение, в которой указывается адрес продолжения и номер кольца, поэтому процесс возобновляет свою работу на соответствующем уровне защиты.

1.3 История появления Linux

Если вы только начинаете погружаться в мир Linux, то для экономии вашего времени лучше сразу ответить на вопросы «чем Linux отличается от Unix», «кто создал все это» и «как выйти из редактора vi». Мы не станем рассказывать всю историю, хотя по ней вполне можно снять увлекательный блокбастер, но затронем основные вехи, чтобы стали понятны ключевые моменты. Правда, чтобы узнать о редакторе `vi`, вам все же придется потерпеть до 7го модуля.

Одной из первых операционных систем была Multics от Bell Labs, и нас будет интересовать только ее название Multiplexed Information and Computing Service, которое можно перевести как Комплексная информационная и вычислительная служба. Концепции этой системы получили дальнейшее

развитие в следующей разработке Unics (Uniplexed Information and Computing System), первый релиз которой вышел в 1969 году. В названии этой системы «односложность» (Uniplexed) противопоставлялась «комплексности» (Multiplexed), чтобы подчеркнуть стремление создать более простое решение. В итоге название Unics было сокращено до Unix, и в таком виде оно дошло до наших дней. На ядре Unix базируются такие современные операционные системы как Solaris от Oracle, AIX от IBM, HP-UX от Hewlett-Packard, Darwin от Apple и другие.

В силу юридических ограничений на коммерциализацию компьютерных технологий компания-правообладатель AT&T передала исходные коды Unix в университет Беркли, что привело к выходу в 1978 году системы BSD (от англ. Berkeley Software Distribution). Сначала это было взаимовыгодное сотрудничество, но в дальнейшем, несмотря на то, что BSD разработала собственное ядро и перестала использовать исходные коды Unix, это сотрудничество привело к судебным спорам о нарушении авторских прав. В настоящее время все юридические проблемы урегулированы, но эта история крайне негативно повлияла на распространение систем из семейства BSD. На ядре BSD базируются такие современные операционные системы, как FreeBSD, OpenBSD, NetBSD и другие.

На фоне лицензионных ограничений и судебных претензий к BSD появился повышенный интерес к альтернативным проектам, таким как GNU и Linux. Название проекта GNU представляет собой рекурсивный акроним GNU is Not UNIX («GNU – это вам не UNIX»), и его автор Ричард Столлман (Richard Stallman) изначально хотел создать полную альтернативу Unix. Но, если в разработке системных утилит проект оказался вполне успешен, то с ядром у разработчиков совсем не заладилось, так как они сделали ставку на микро-ядерную архитектуру, которая оказалась сложна в разработке и отладке.

Проект Linux стартовал значительно позже, но фокусировался исключительно на создании ядра операционной системы, и его автор Линус Торвальдс сделал ставку на монолитную архитектуру, что позволило ему добиться успеха намного быстрее. Просто для сравнения: первый релиз ядра Linux вышел в 1991 году, а стабильной версии ядра GNU HURD нет до сих пор.

Вместе с тем, эти два проекта отлично дополнили друг друга, так как, объединяя свободное ядро Linux и свободный набор программного обеспечения GNU, разработчики смогли собрать полноценные операционные системы, с помощью которых можно решать конкретные прикладные задачи. Поэтому наравне с GNU\Linux появились Red Hat Linux, Debian Linux, Ubuntu Linux и еще более пяти сотен дистрибутивов, больше половины из которых уже ушли в историю.

Можно отметить, что за каждым успешным проектом стоят большие инвестиции и интересы большого числа пользователей. Успех продукта в конечном итоге определяется тем, сколько усилий в него вложено и сколько людей в него поверило. И, если говорить о российском рынке, то одним из признанных лидеров является операционная система Astra Linux Special Edition (ALSE), доля которой по результатам 2023 года составила более 75%.

1.4 Операционная система Astra Linux

ПАО Группа Астра является одним из лидеров российского рынка информационных технологий в области разработки программного обеспечения и средств защиты информации. В портфеле компании два десятка продуктов, основным из которых является сертифицированная операционная система специального назначения Astra Linux со встроенными средствами защиты информации (СЗИ) для безопасной работы инфраструктур любого масштаба и с данными любой степени конфиденциальности, что подтверждается сертификатами ФСТЭК, ФСБ и Минобороны России.

Операционная система Astra Linux позволяет безопасно использовать кодовую базу Debian, включая большое количество прикладных программ, и предоставляет своевременный доступ к необходимым исправлениям безопасности. Система оснащена собственными средствами защиты информации — модулем мандатного контроля доступа и целостности замкнутой программной среды. И это не считая оригинальной графической оболочки Fly, которая покажется привычной большинству пользователей и упростит процесс импортозамещения.

Операционная система Astra Linux поддерживает большой перечень современного оборудования, и сотрудники компании активно работают над расширением этого списка. Уже более 3 000 предприятий используют Astra Linux и экономят время на подборе и тестировании ПО и комплектующих. С актуальным перечнем совместимого оборудования вы можете ознакомиться на сайте компании https://astralinux.ru/ready-for-astra/

1.4.1 Версии Astra Linux

Версионирование Astra Linux отличается от того, как это принято в Windows. Номер версии состоит из нескольких чисел, разделенных символом точки, см. рис. 1.3.

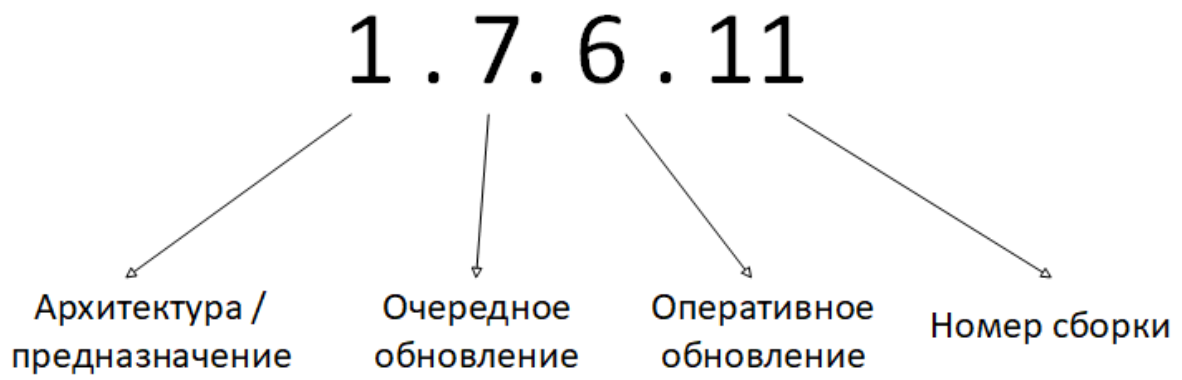


рис. 1.3 – Версионирование ОС Astra Linux

Первое число (1) в номере версии обозначает архитектуру или предназначение. Возможны следу-

ющие значения:

- 1 — Astra Linux Special Edition (ALSE) под архитектуру x86-64;
- 2 — Astra Linux Common Edition (ALCE) под архитектуру x86-64;
- 4 — ALSE под архитектуру ARM;
- 8 — ALSE под архитектуру Эльбрус.

Второе число (7) соответствует номеру очередного обновления (версии ОС). Выпуск очередных обновлений необходим для:

- реализации и совершенствования функциональных возможностей;
- поддержки современного оборудования;
- обеспечения соответствия актуальным требованиям безопасности информации;
- повышения удобства использования, управления компонентами и другое.

Таким образом, версии 1.7 и 4.7 являются одной и той же версией операционной системы с номером очередного обновления 7, но под разные архитектуры. Версия 1.7 предназначена для работы на x86-64 процессорах, а версия 4.7 будет работать на ARM.

Третье число (6) соответствует номерам оперативных обновлений, выпуск которых позволяет устранять критические уязвимости и уязвимости высокого уровня опасности. Такие обновления могут публиковаться не только в виде файлов программных компонентов, в которые внесены изменения, но и в виде инструкций и методических указаний по настройке и особенностям эксплуатации операционной системы, содержащих сведения о компенсирующих мерах или ограничениях по применению операционной системы при эксплуатации.

Текущую версию установленной системы можно посмотреть в файле `/etc/astra_version`, версия сборки ОС находится в файле `/etc/astra/build_version`

```
$ cat /etc/astra_version
1.7.6
$ cat /etc/astra/build_version
1.7.6.11
```

Кроме очередных и оперативных обновлений компания выпускает так называемые срочные оперативные обновления, которые обозначаются префиксом UU (от англ. urgent update). Например, для ALSE 1.7.6 было выпущено два срочных обновления UU1 и UU2.

Лицензиатам, использующим ОС «Astra Linux Special Edition», сертифицированную по требованиям безопасности информации ФСТЭК России (сертификат № 2557), необходимо применять опубликованные в бюллетенях безопасности методики и устанавливать обновления программного обес-

печения для нейтрализации угроз уязвимостей в информационных системах.

1.4.2 Репозитории Astra Linux

> Официальные репозитории операционной системы доступны по адресу https://download.astralinux.ru, но так как индексация каталогов внутри репозитория запрещена просматривать вручную эти каталоги не целесообразно, с ними нужно работать через менеджер пакетов apt, см. *Модуль 15. Управление ПО в Astra Linux*.

Для версии ALSE x.7 доступно 4 репозитория:

- **main** — основной репозиторий операционной системы. Программный код из этого репозитория реализует все функциональные возможности и функции безопасности продукта.
- **update** — репозиторий с оперативными обновлениями основного репозитория.
- **base** — базовый репозиторий, который объединяет main и update, а также содержит пакеты со средствами разработки.
- **extended** — содержит дополнительное программное обеспечение, которое может функционировать в среде Astra Linux. Указанное ПО не дорабатывается для реализации функций безопасности и может быть несовместимо с последними оперативными обновлениями основного/-базового репозитория, что фиксируется в бюллетене оперативного обновления и устраняется по мере обновления расширенного репозитория. Программное обеспечение расширенного репозитория не попадает под сертификацию.

1.4.3 Уровни защищенности ALSE

Начиная с очередного обновления ALSE 1.7, операционная система поставляется в виде единого дистрибутива, способного функционировать в одном из трех режимов безопасности (уровне защищенности), реализованном запатентованными средствами защиты информации (СЗИ):

- **Базовый уровень защищенности «ОРЕЛ»**. Рекомендован для обработки общедоступной информации, не требующей специальных средств защиты.
- **Усиленный уровень защищенности «ВОРОНЕЖ»**. Рекомендован для обработки конфиденциальной информации, не содержащей сведения, составляющие государственную тайну.
- **Максимальный уровень защищенности «СМОЛЕНСК»**. Рекомендован для обработки информации любой категории доступа в государственных информационных системах.

1.4.4 Программа Ready for Astra

Компания выполняет проверку программного и аппаратного обеспечения сторонних производителей на совместимость с Astra Linux, сертификаты совместимости выданы уже для более чем 2100 продуктов. Одновременно с этим компания разрабатывает собственные продукты по наиболее востребованным направлениям для расширения возможностей экосистемы и упрощения реализации программы импортозамещения.

Подробнее о программе Ready for Astra можно узнать по ссылке https://astralinux.ru/ready-for-astra/.

1.4.5 Экосистема Астры

Экосистема Астры представляет собой группу продуктов, которые обеспечивают комплексный подход к импортозамещению программного обеспечения на предприятиях. Экосистема реализует следующие базовые сервисы: операционная система, служба каталога, виртуализация, резервное копирование, почта, инфраструктура виртуальных рабочих мест, база данных и другие, см. рис. 1.4.



рис. 1.4 – Экосистема Астры

В экосистему Астры входят следующие продукты:

- **ALD Pro** – это система централизованного управления ИТ-инфраструктурой (аналог Microsoft Active Directory). Продукт предоставляет возможности централизованного управления учетными записями пользователей, единую точку сквозной аутентификации по про-

токолу Kerberos v5 и централизованное администрирование доменных компьютеров через механизм групповых политик. В системе дополнительно реализована возможность автоматического развертывания наиболее востребованных сервисов: репозиторий программного обеспечения, файловый сервер, принт-сервер, DHCP, установка ОС по сети, аудит, мониторинг.

- **RuPost** – система для управления корпоративной почтой, календарями, контактами и адресными книгами (аналог Microsoft Exchange). Благодаря встроенному в систему веб-клиенту рабочие места конечных пользователей могут работать под разными ОС: Windows, Linux, iOS и Android.

- **WorksPad** – клиент-серверное корпоративное решение класса Enterprise Mobility Management (EMM) для организации защищенной мобильной работы сотрудников на смартфонах и планшетах под управлением iOS и Android. Продукт позволяет обеспечить высочайший уровень защиты данных и предотвратить их утечку за счет установки на клиентское мобильное устройство контролируемого компанией приложения-контейнера с настраиваемыми политиками безопасности и шифрования. При этом приложение не контролирует приватные данные владельца устройства и никак на них не влияет. При необходимости решение можно применять в комплексе с другими средствами для управления (MDM) и предотвращения утечек информации (DLP).

- **Tantor** – удобная в использовании СУБД российского производства с повышенной производительностью и встроенной системой администрирования и мониторинга (аналог Microsoft SQL Server).

- **Брест** - российская платформа со встроенными средствами защиты информации ОС Astra Linux Special Edition для создания и управления облачными виртуальными ИТ-инфраструктурами любой сложности (аналог Microsoft Private Cloud). Позволяет развернуть частное или публичное облако, в том числе распределенное, и гибко управлять виртуальной инфраструктурой и ее компонентами, учетными записями пользователей и виртуальными машинами: создавать их из шаблонов, клонировать, удалять и т.д.

- **VMmanager** – платформа управления аппаратной и контейнерной виртуализации (аналог Microsoft Hyper-V).

- **DCImanager** – платформа для управления мультивендорной IT-инфраструктурой.

- **BILLmanager** – платформа для авторизации частного и гибридного облака.

- **Termidesk** – реализация инфраструктуры виртуальных рабочих мест для организации удаленной и безопасной работы сотрудников с любых устройств из любой точки мира (аналог Microsoft Enterprise Desktop Virtualization).

- **RuBackup** – клиент-серверное приложение для автоматизации резервного копирования и вос-

становления данных.

С дополнительной информацией вы можете ознакомиться на официальном сайте: https://astralinux.ru/

1.4.6 Лицензирование Astra Linux

Astra Linux Special Edition доступна в трех типах лицензий и трех режимах безопасности.

Типы лицензий:

- Для рабочей станции
- Серверная лицензия
- Для тонкого клиента

Режимы безопасности операционной системы Astra Linux Special Edition:

- «ОРЕЛ» - базовый уровень защищенности
- «ВОРОНЕЖ» - усиленный уровень защищенности
- «СМОЛЕНСК» - максимальный уровень защищенности

табл. 1.1 – Сравнительная таблица редакций Astra Linux Special Edition

Характеристика	Орел	Воронеж	Смоленск
Сертификат ФСТЭК	_	v	v
Сертификат ФСБ	_	_	v
Сертификат МО	_	_	v
Мандатный контроль целостности	_	v	v
Ролевое управление доступом	_	v	v
Дискреционное управление доступом	_	v	v
Мандатное управление доступом	_	_	v
Механизмы защиты СУБД и средств виртуализации	v	v	v
Режим «киоск» — «белый» список разрешенных к запуску приложений	v	v	v
Корректная работа с электронной подписью	v	v	v
Дополнительная изоляция компонентов виртуализации и контейнеров Docker(отдельный уровень целостности)	_	v	v
Динамический контроль целостности (замкнутая программная среда)	_	v	v
Регистрация событий безопасности	v	v	v
Контроль подключения съемных машинных носителей информации	v	v	v

1.5 Практика и тестирование

1.5.1 Тест: Модуль 1. Введение в операционные системы

Пожалуйста, ответьте на несколько вопросов, чтобы закрепить полученные знания. Правильные ответы для самопроверки вы сможете найти в конце списка. Обратите внимание, что у некоторых вопросов может быть несколько верных вариантов ответов.

1.5.1.1 Вопросы

Вопрос 1. Какие бывают режимы работы процессора по уровню привилегий?

a. Режим пользователя

b. Однопользовательский режим

c. Служебный режим

d. Режим ядра

Вопрос 2. Выберите четыре характеристики релиза, которые указываются в версии операционной системы **ALSE 1.7.6.11**:

a. Номер сборки

b. Архитектура

c. Срочное оперативное обновление

d. Очередное обновление

e. Оперативное обновление

f. Внеочередное обновление

Вопрос 3. Какие репозитории используются в ALSE x.7?

a. Основной репозиторий (main)

b. Глобальный репозиторий (global)

c. Базовый репозиторий (base)

d. Расширенный репозиторий (extended)

e. Дополнительный репозиторий (additional)

f. Оперативные обновления основного репозитория (update)

g. Общий репозиторий (common)

h. Служебный репозиторий (service)

i. Продуктовый репозиторий (product)

Вопрос 4. Какие уровни безопасности поддерживает операционная система ALSE?

a. Базовый

b. Общий

c. Специальный

d. Максимальный

e. Служебный

f. Наивысший

g. Усиленный

h. Пониженный

Вопрос 5. Для чего предназначен продукт Брест?

a. СУБД

b. Платформа виртуализации

c. Инфраструктура виртуальных рабочих мест

d. Организация защищенной мобильной работы на смартфонах и планшетах под управлением iOS и Android

Вопрос 6. Для чего предназначен продукт Tantor?

a. СУБД

b. Платформа виртуализации

c. Инфраструктура виртуальных рабочих мест

d. Организация защищенной мобильной работы на смартфонах и планшетах под управлением iOS и Android

Вопрос 7. Для чего предназначен продукт WorksPad?

a. СУБД

b. Платформа виртуализации

c. Инфраструктура виртуальных рабочих мест

d. Организация защищенной мобильной работы на смартфонах и планшетах под управлением iOS и Android

Вопрос 8. Для чего предназначен продукт Termidesk?

a. СУБД

b. Платформа виртуализации

c. Инфраструктура виртуальных рабочих мест

d. Организация защищенной мобильной работы на смартфонах и планшетах под управлением iOS и Android

Ответы на вопросы

1. a, d
2. a, b, d, e
3. a, c, d, f
4. a, d, g
5. b
6. a
7. d
8. c

1.6 Заключение

В этом модуле мы узнали:

- Основные отличия между системами Windows и Linux
- Что такое операционная система, её предназначение и компоненты
- История появления Linux
- Основные особенности ОС Astra Linux Special Edition
- Как определить версию ALSE и чем они отличаются
- Какие бывают уровни защищенности ALSE и их особенности
- Лицензирование ALSE
- Что такое программа Ready for Astra
- Что такое экосистема Astra и что в неё входит

На этом мы заканчиваем исторический экскурс и обзорный модуль про операционные системы и переходим к следующему модулю, посвященному установке системы.

1.7 Дополнительные источники информации

- Подробнее о лицензиях Unix и Linux вы можете почитать по ссылке https://www.gnu.org/licenses/licenses.html
- Список программного и аппаратного обеспечения, совместимого с Astra Linux https://astralinux.ru/ready-for-astra/
- С условиями лицензирования программных продуктов группы Астра вы можете ознакомиться на официальном сайте компании https://astragroup.ru/info/law/

1.8 Обратная связь

Если остались вопросы, то их всегда можно задать в специальной теме.

Модуль 2. Установка ОС Astra Linux

2.1 Введение

Вы узнаете о системных требованиях, порядке установки и первоначальной настройке операционной системы Astra Linux SE 1.7.6.

2.2 Требования для установки ОС Astra Linux SE 1.7.6

Минимальные системные требования — это условия, при которых ваш компьютер сможет стабильно работать без графического интерфейса рабочего стола Fly, без приложений, которым требуется Fly (веб-браузер, офисный пакет и так далее), а также без дополнительных сервисов, которые входят в состав дистрибутива, но не включены по умолчанию.

Минимальные системные требования для установки ОС Astra Linux SE 1.7.6 следующие:

- Аппаратная платформа – процессор с архитектурой x86-64 (AMD, Intel).
- Объем оперативной памяти – не менее 1,5 ГБ.
- Объем свободного дискового пространства – не менее 4 ГБ.
- Устройства чтения DVD-дисков.
- Стандартный монитор SVGA.

- Совместимость с оборудованием (штатное, предусмотренное документацией).

Рекомендуемые системные требования обеспечивают стабильную работу операционной системы с графическим интерфейсом рабочего стола Fly и всеми программами, входящими в состав дистрибутива. Это не касается случаев, когда в документации указаны особые требования.

Рекомендуемые системные требования для установки ОС Astra Linux SE 1.7.6 следующие:

- Аппаратная платформа – процессор с архитектурой x86-64 (AMD, Intel).
- Объем оперативной памяти – не менее 4 ГБ.
- Объем свободного дискового пространства – не менее 16 ГБ.
- Устройства чтения DVD-дисков.
- Стандартный монитор SVGA.
- Совместимость с оборудованием (штатное, предусмотренное документацией).

Примечание: Функционирование обеспечивается только на рекомендованном изготовителем ОС совместимом оборудовании, перечень которого вместе с регламентом сертификации на совместимость опубликованы на сайте http://astralinux.ru/.

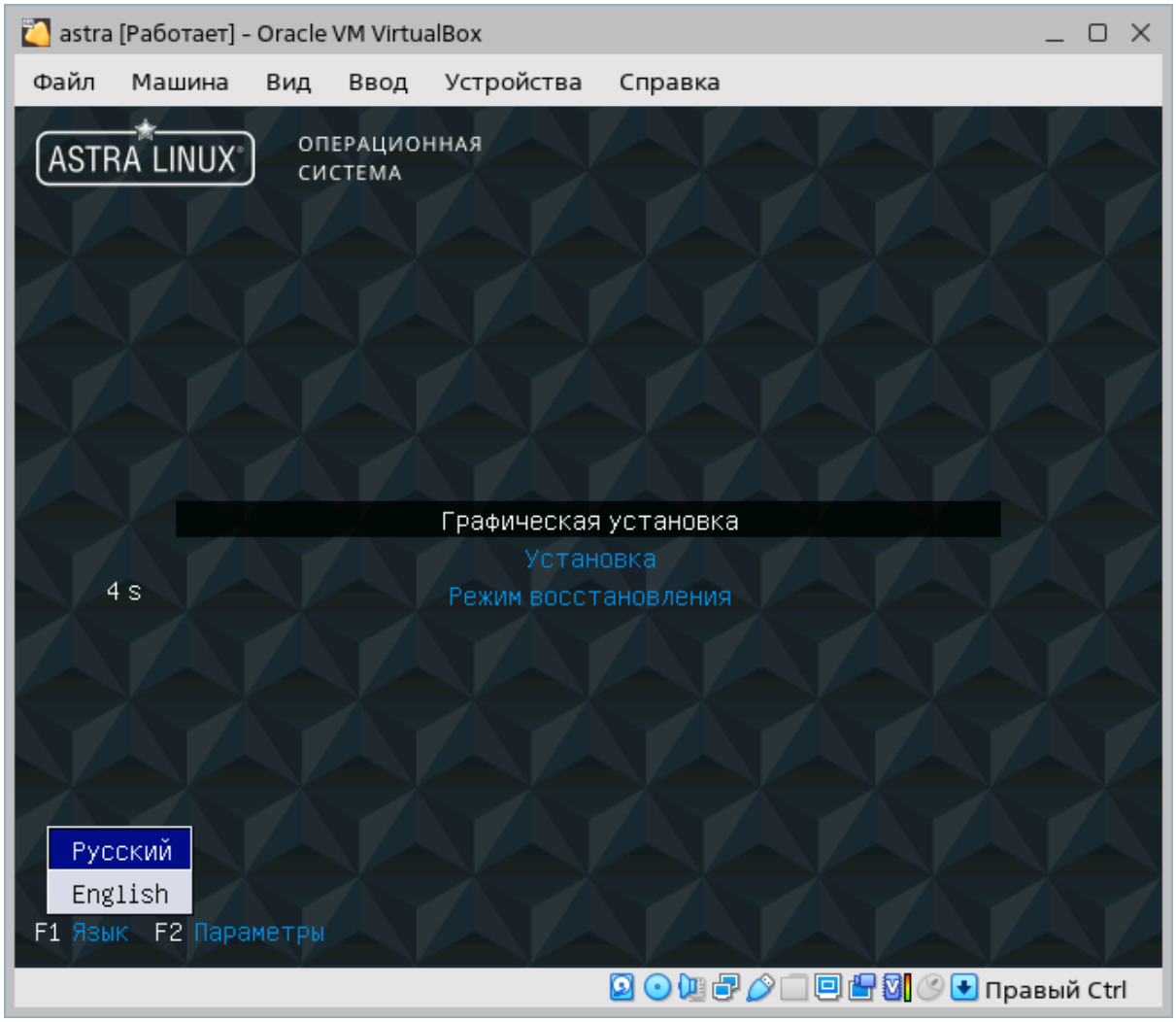
2.3 Порядок установки Astra Linux

Установка Astra Linux включает несколько шагов, некоторые из которых выполняются автоматически, то есть без вашего прямого участия. Рассмотрим каждый из этих шагов последовательно.

2.3.1 Шаг 1. Выбор графического или текстового режима установки

При загрузке системы с установочного диска Astra Linux сначала стартует загрузчик GRUB, который предлагает выбрать параметры установки:

- Графическая установка (Graphic installation).
- Установка с использованием текстового интерфейса (Installation).
- Режим восстановления (Rescue mode).

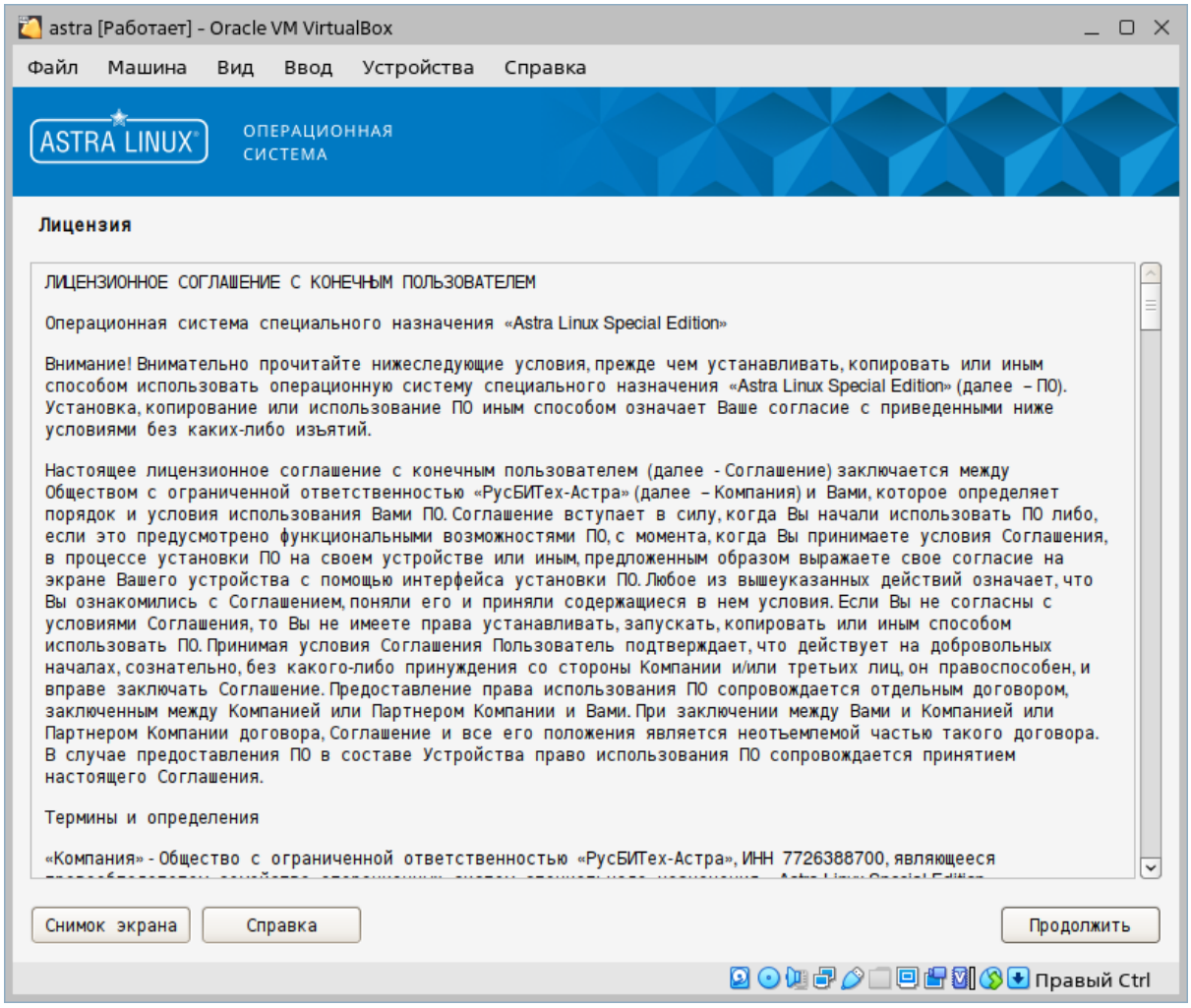


Если ничего не предпринимать, то через 5 секунд автоматически начнется установка в графическом режиме с русским интерфейсом. Подождем указанное время или продолжим принудительно нажатием клавиши Enter.

Примечание: Если виртуальная машина перехватила управление мышью, то вы можете вернуть контроль горячим сочетанием клавиш: Левый Ctrl + Правый Ctrl.

2.3.2 Шаг 2. Ознакомление с лицензионным соглашением и его принятие

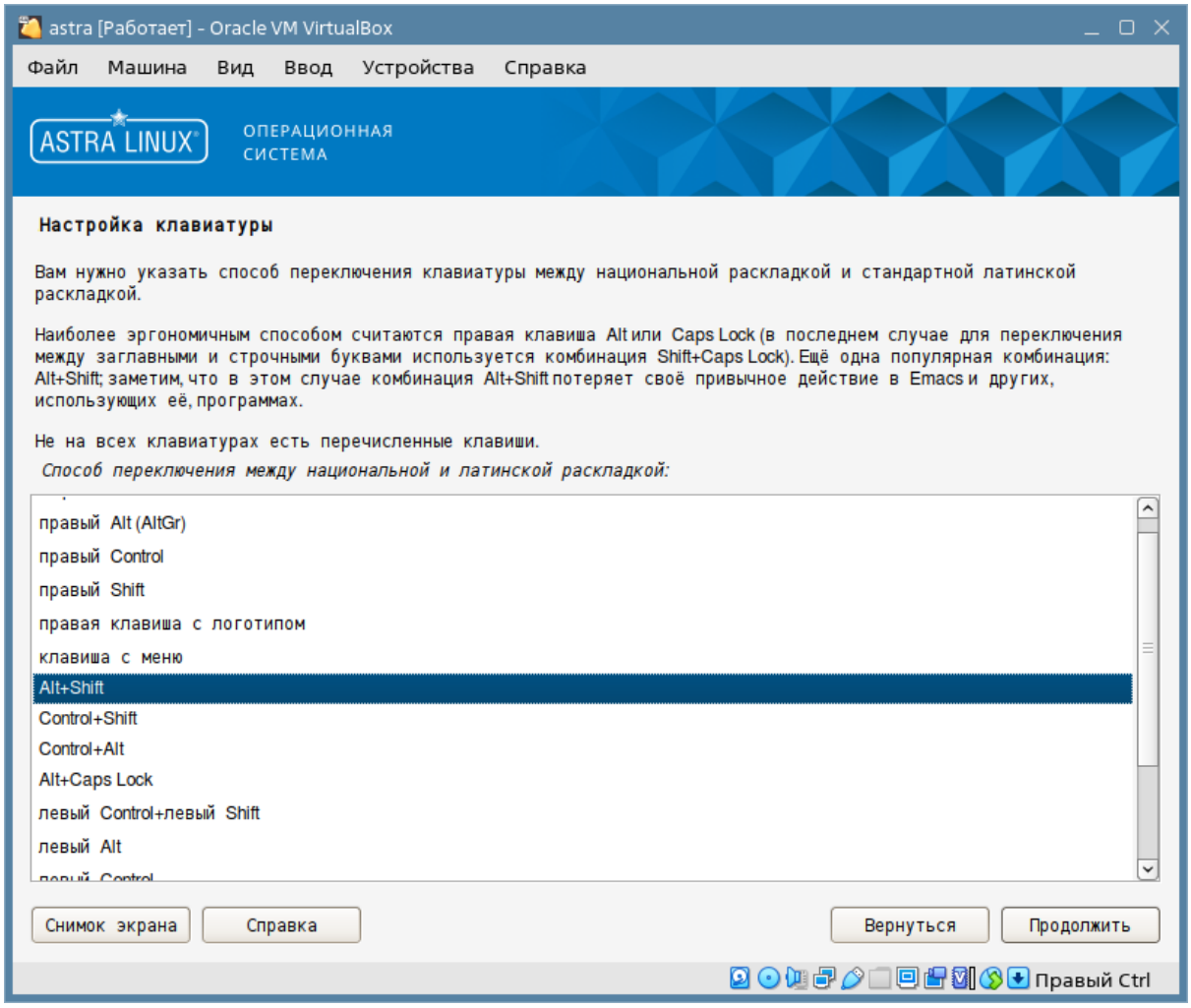
Далее перед вами появится лицензионное соглашение. Начиная с этого окна, мы работаем уже в Linux, и, если вам интересно это проверить, то включите виртуальную клавиатуру и перейдите на второй экран горячим сочетанием клавиш: Ctrl + Alt + F2. Чтобы вернуться назад, используйте: Ctrl + Alt + F5.



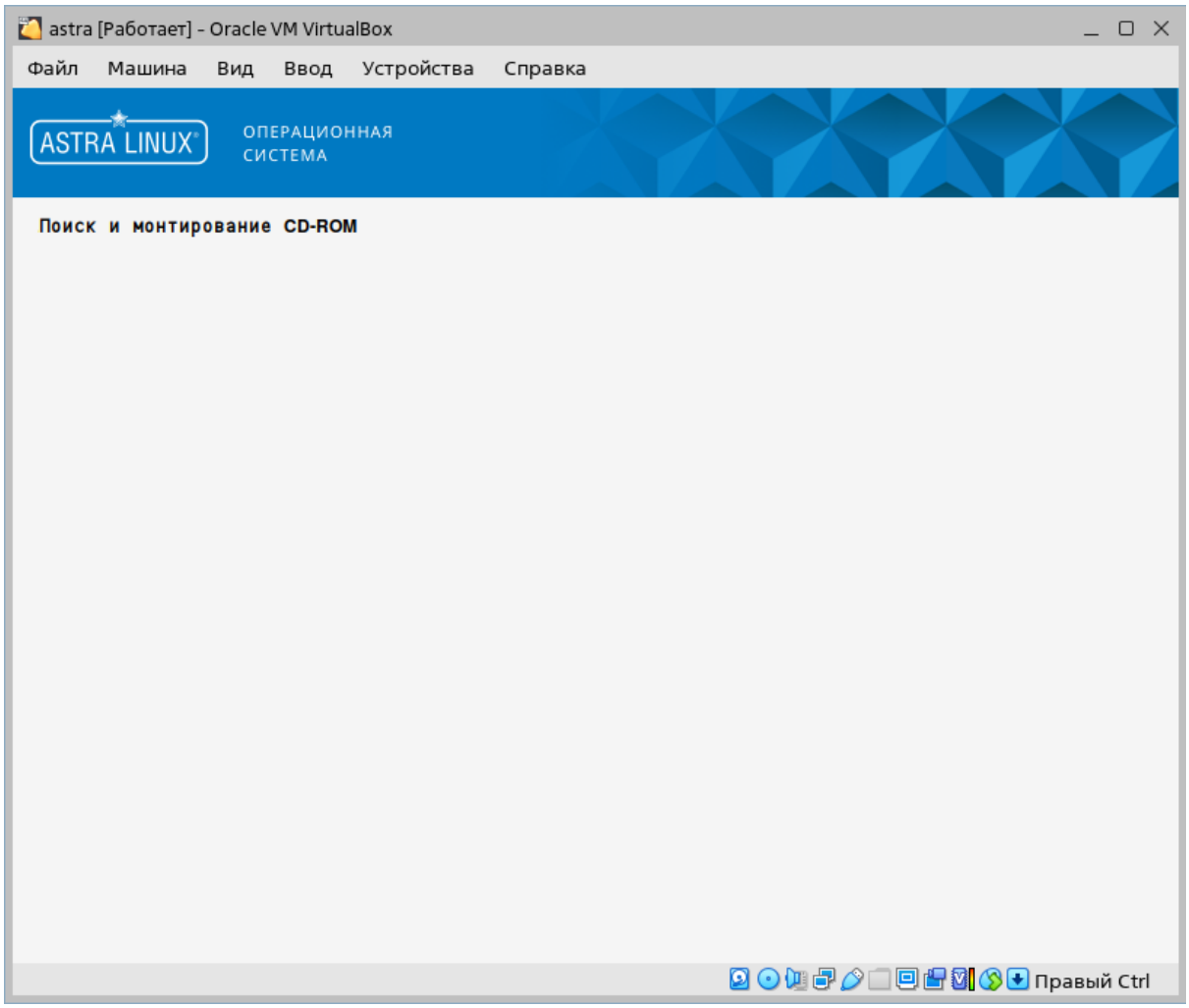
Если же вам некогда отвлекаться по пустякам, то ознакомьтесь с лицензионным соглашением и нажмите на кнопку *Продолжить*. Кстати, говорят, что у жителей постсоветского пространства черный пояс по скорости чтения лицензионных соглашений к устанавливаемому программному обеспечению.

2.3.3 Шаг 3. Настройка клавиатуры

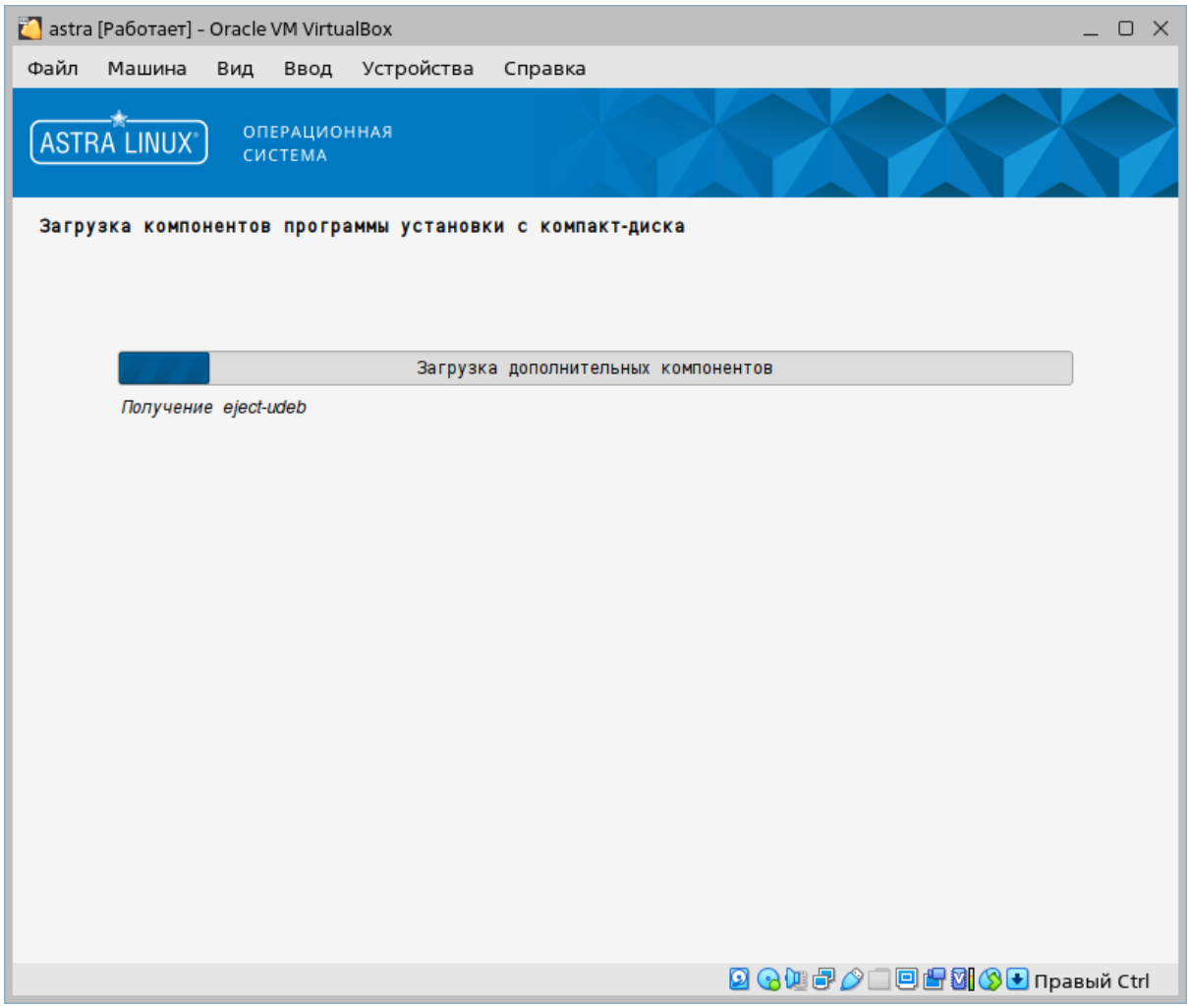
Выберите удобный для вас способ переключения раскладки клавиатуры (по умолчанию это Alt + Shift).


astra [Работает] - Oracle VM VirtualBox
Файл Машина Вид Ввод Устройства Справка
ASTRA LINUX
ОПЕРАЦИОННАЯ СИСТЕМА
Настройка клавиатуры
Вам нужно указать способ переключения клавиатуры между национальной раскладкой и стандартной латинской раскладкой.
Наиболее эргономичным способом считаются правая клавиша Alt или Caps Lock (в последнем случае для переключения между заглавными и строчными буквами используется комбинация Shift+Caps Lock). Ещё одна популярная комбинация: Alt+Shift; заметим, что в этом случае комбинация Alt+Shift потеряет своё привычное действие в Emacs и других, использующих её, программах.
Не на всех клавиатурах есть перечисленные клавиши.
Способ переключения между национальной и латинской раскладкой:
правый Alt (AltGr)
правый Control
правый Shift
правая клавиша с логотипом
клавиша с меню
Alt+Shift
Control+Shift
Control+Alt
Alt+Caps Lock
левый Control+левый Shift
левый Alt
Снимок экрана
Справка
Вернуться
Продолжить
Правый Ctrl

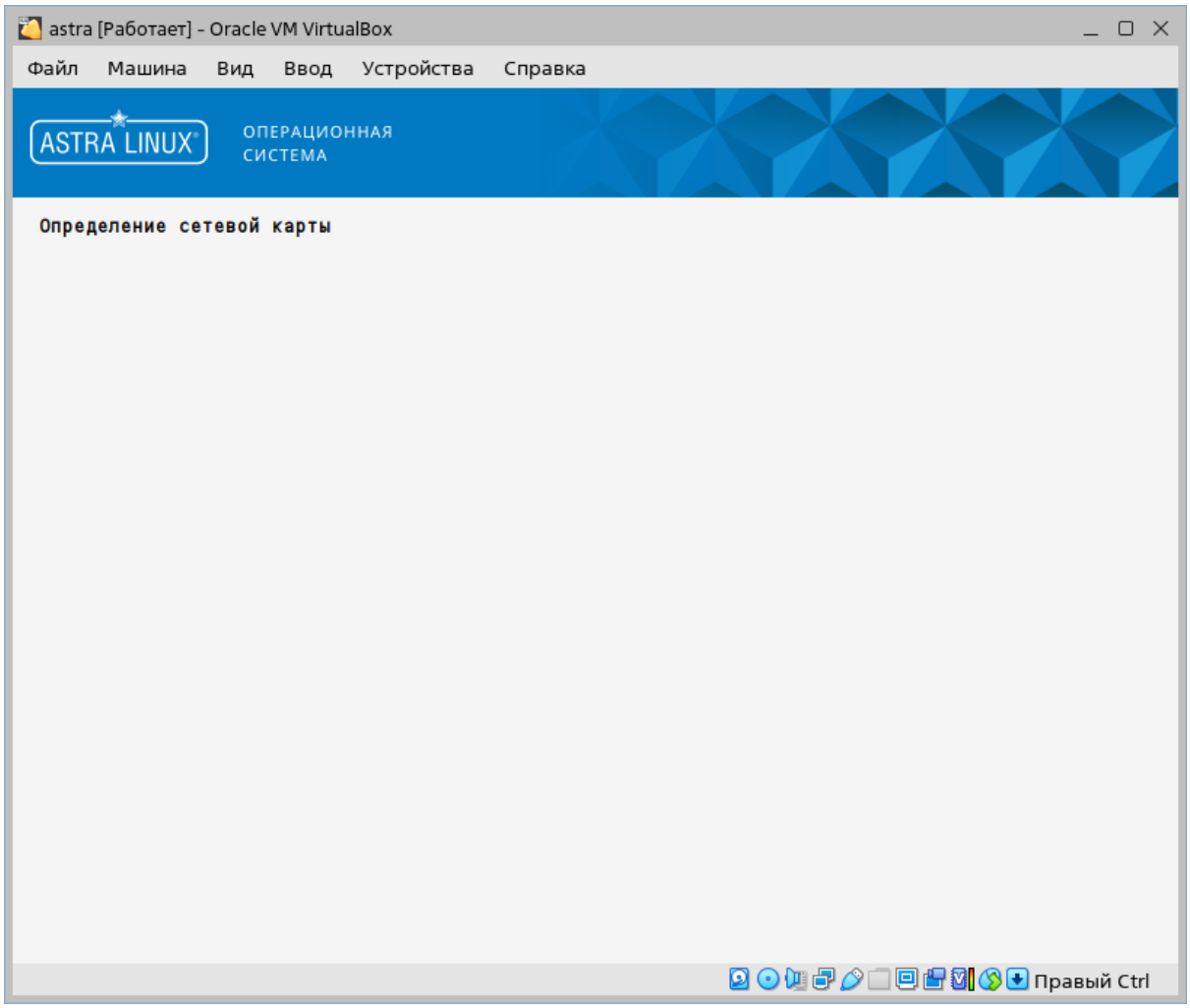
2.3.4 Шаг 4. Поиск и монтирование CD-ROM (автоматически)



2.3.5 Шаг 5. Загрузка компонентов программы установки с компакт-диска (автоматически)

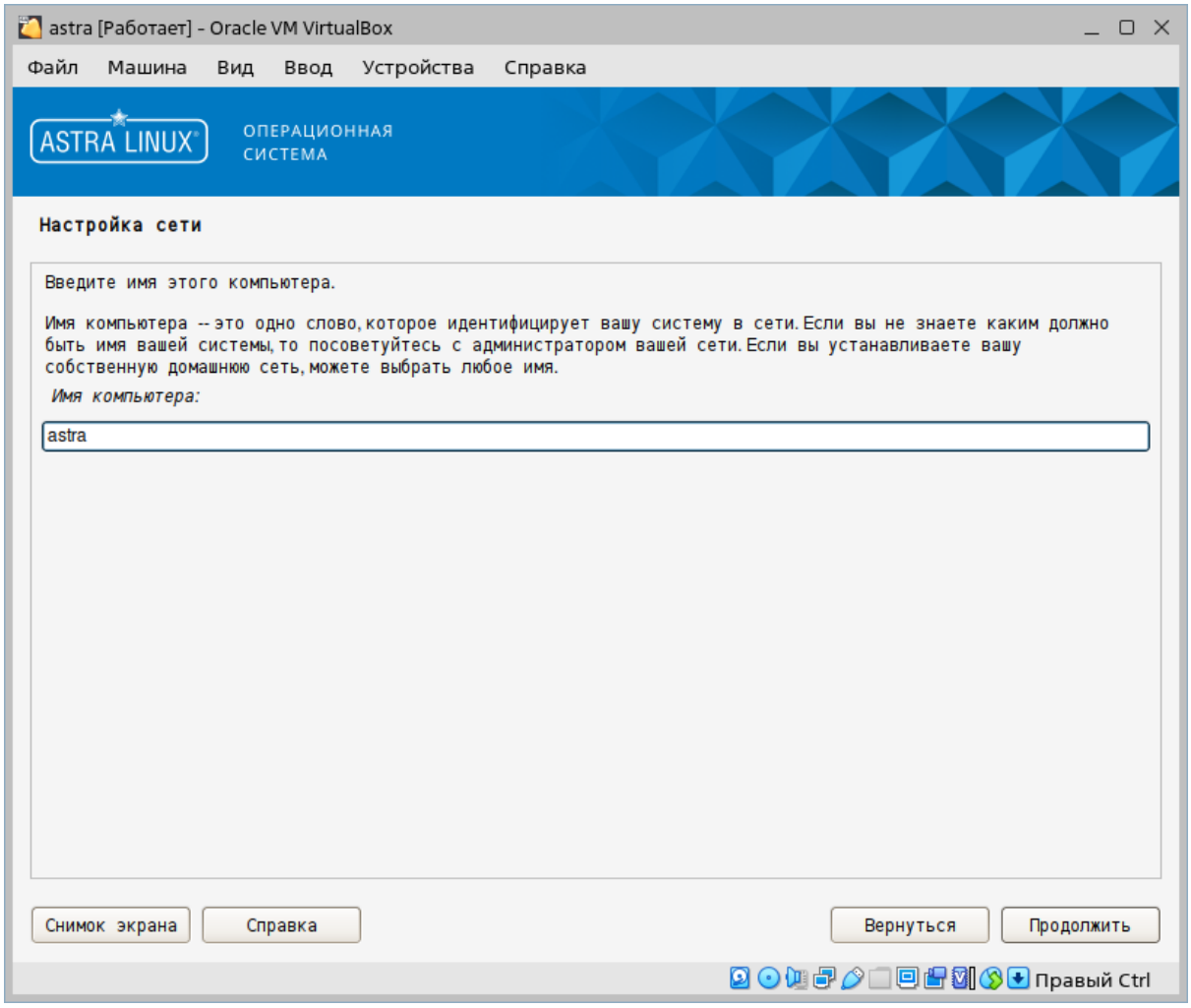


2.3.6 Шаг 6. Определение сетевой карты (автоматически)

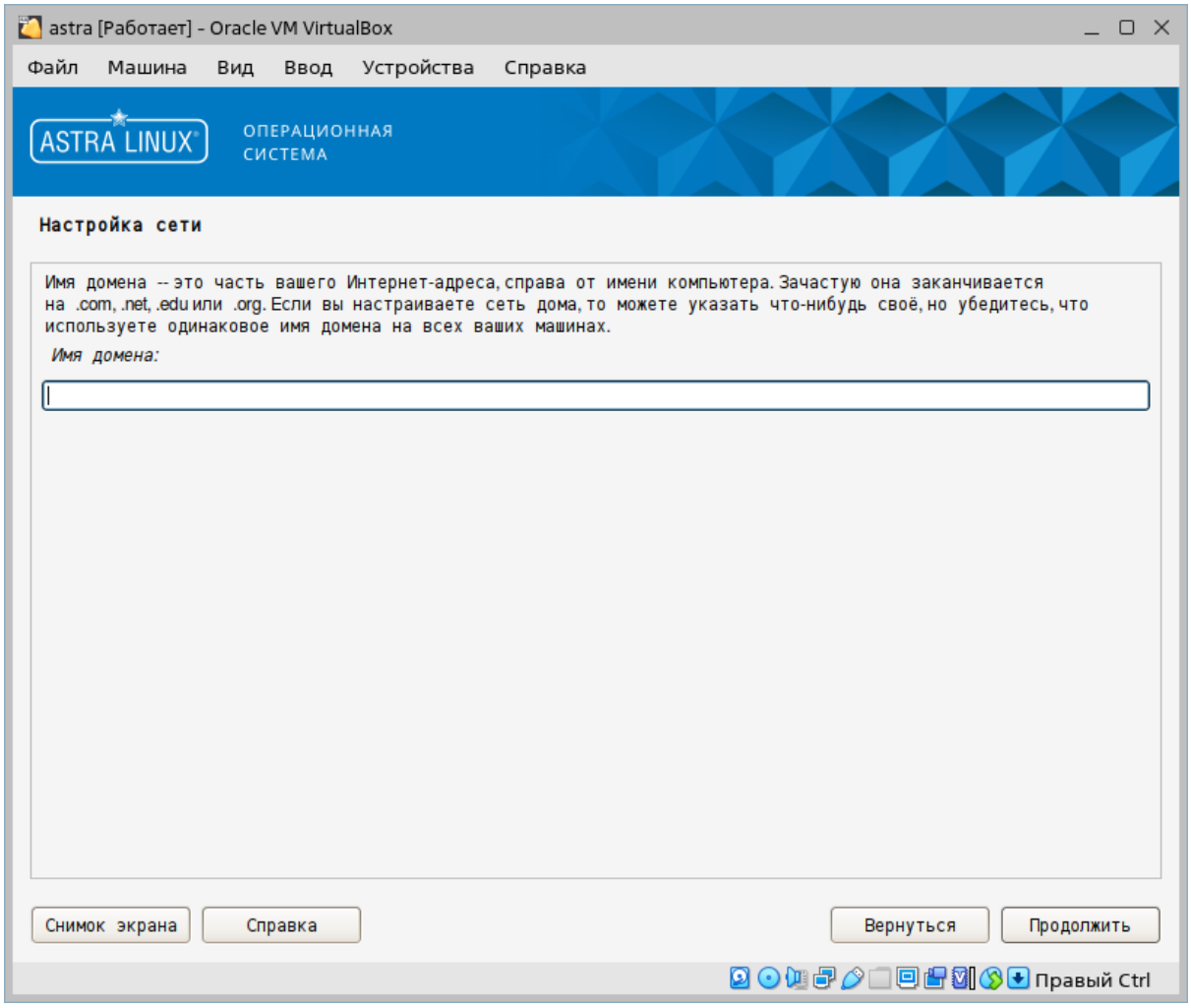


2.3.7 Шаг 7. Настройка сети

Введите желаемое имя компьютера или оставьте значение **astra** по умолчанию. Имя компьютера идентифицирует операционную систему в компьютерной сети.

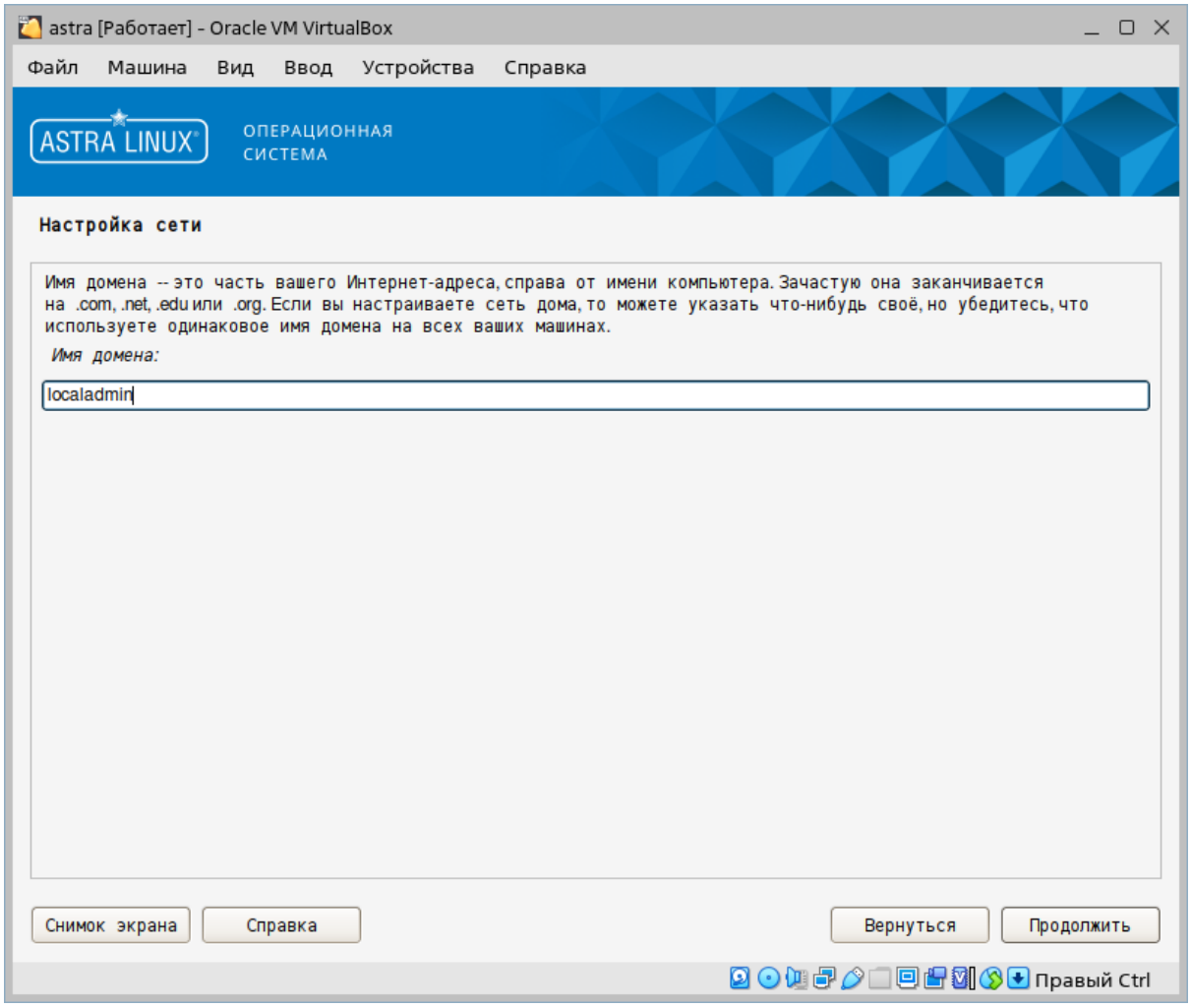


Введите имя домена, которое является суффиксом полного имени компьютера и может заканчиваться на .lan, .ru, .com и т.д. По умолчанию значение не задано и его можно не заполнять.

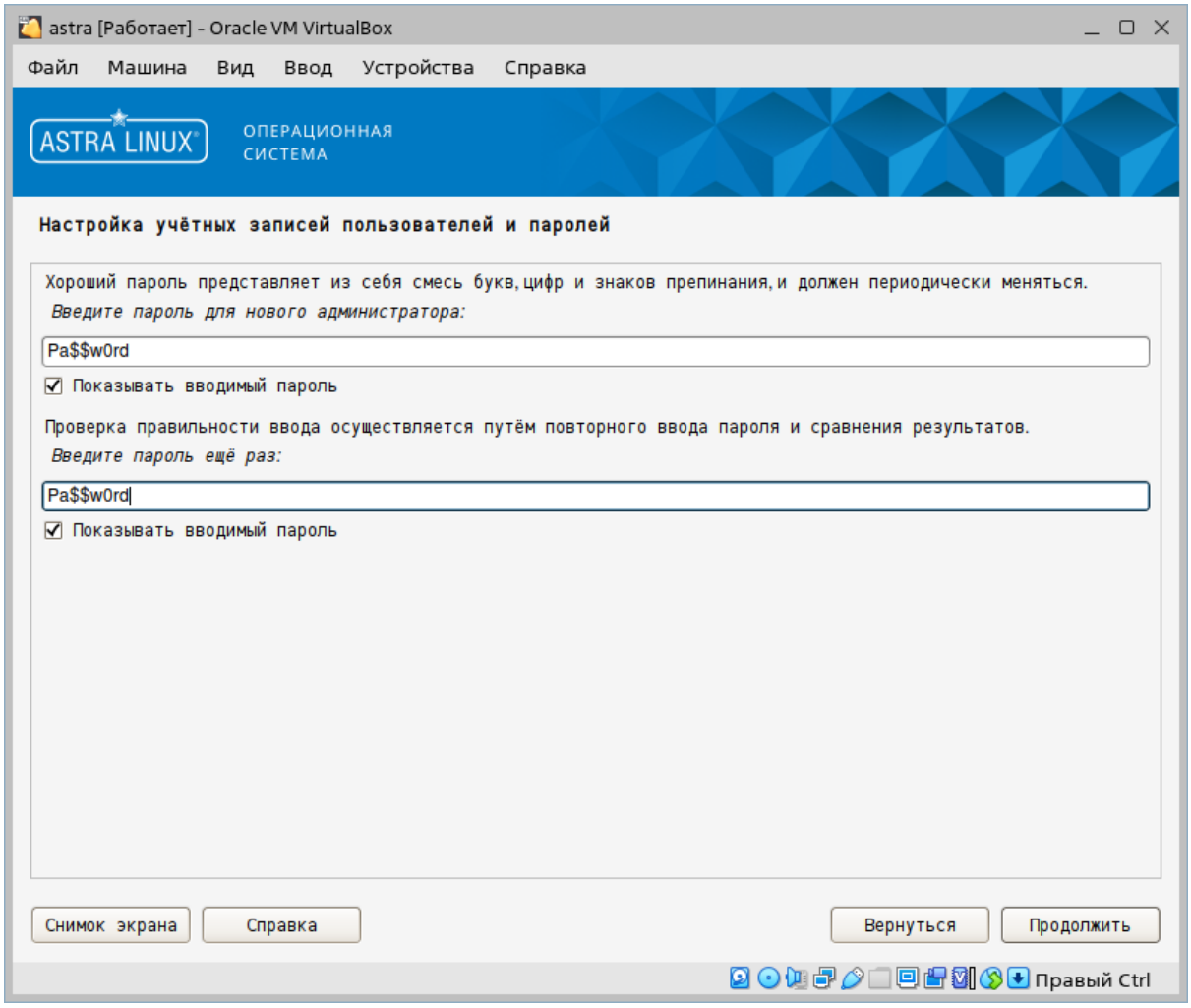


2.3.8 Шаг 8. Настройка учетных записей пользователей и паролей

Введите имя локального администратора для этого компьютера, например, **localadmin**.



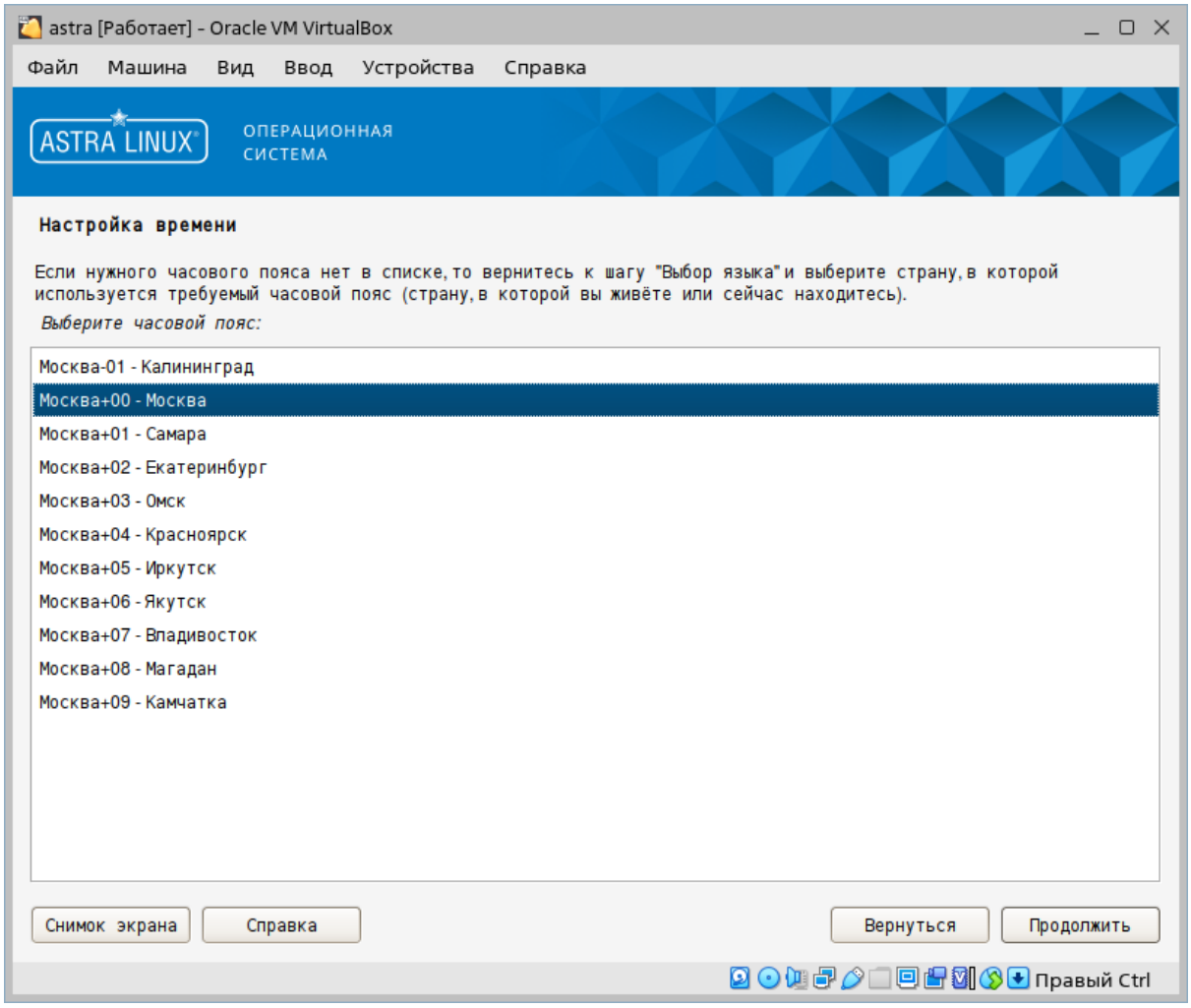
Введите пароль учетной записи и нажмите на кнопку *Продолжить*.



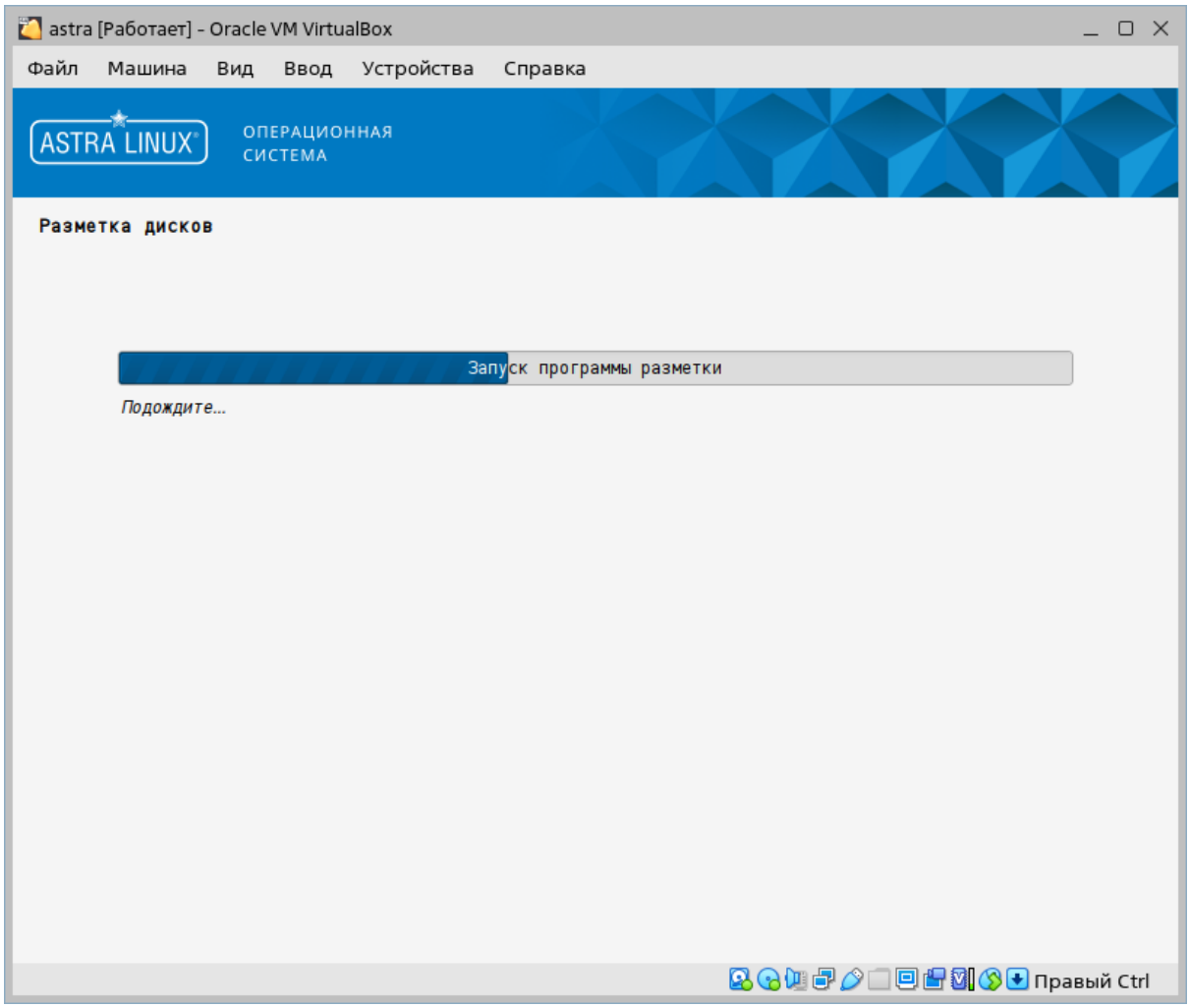
Примечание: Так же как при установке Windows создается учетная запись локального администратора с идентификатором безопасности RID=500, в Astra Linux на этом шаге создается локальная учетная запись пользователя с идентификатором POSIX UID=1000. Данная учетная запись будет автоматически включена в группу astra-admin, которой даны права на повышение привилегий до уровня суперпользователя root через утилиту sudo, см. файлы /etc/sudoers и /etc/group.

2.3.9 Шаг 9. Настройка времени. Выбор часового пояса.

Выберите часовой пояс (по умолчанию это Москва) и нажмите на кнопку *Продолжить*.


astra [Работает] - Oracle VM VirtualBox
Файл Машина Вид Ввод Устройства Справка
ASTRA LINUX
ОПЕРАЦИОННАЯ СИСТЕМА
Настройка времени
Если нужного часового пояса нет в списке, то вернитесь к шагу "Выбор языка" и выберите страну, в которой используется требуемый часовой пояс (страну, в которой вы живёте или сейчас находитесь).
Выберите часовой пояс:
Москва-01 - Калининград
Москва+00 - Москва
Москва+01 - Самара
Москва+02 - Екатеринбург
Москва+03 - Омск
Москва+04 - Красноярск
Москва+05 - Иркутск
Москва+06 - Якутск
Москва+07 - Владивосток
Москва+08 - Магадан
Москва+09 - Камчатка
Снимок экрана
Справка
Вернуться
Продолжить
Правый Ctrl

2.3.10 Шаг 10. Определение дисков и запуск разметки (автоматически)

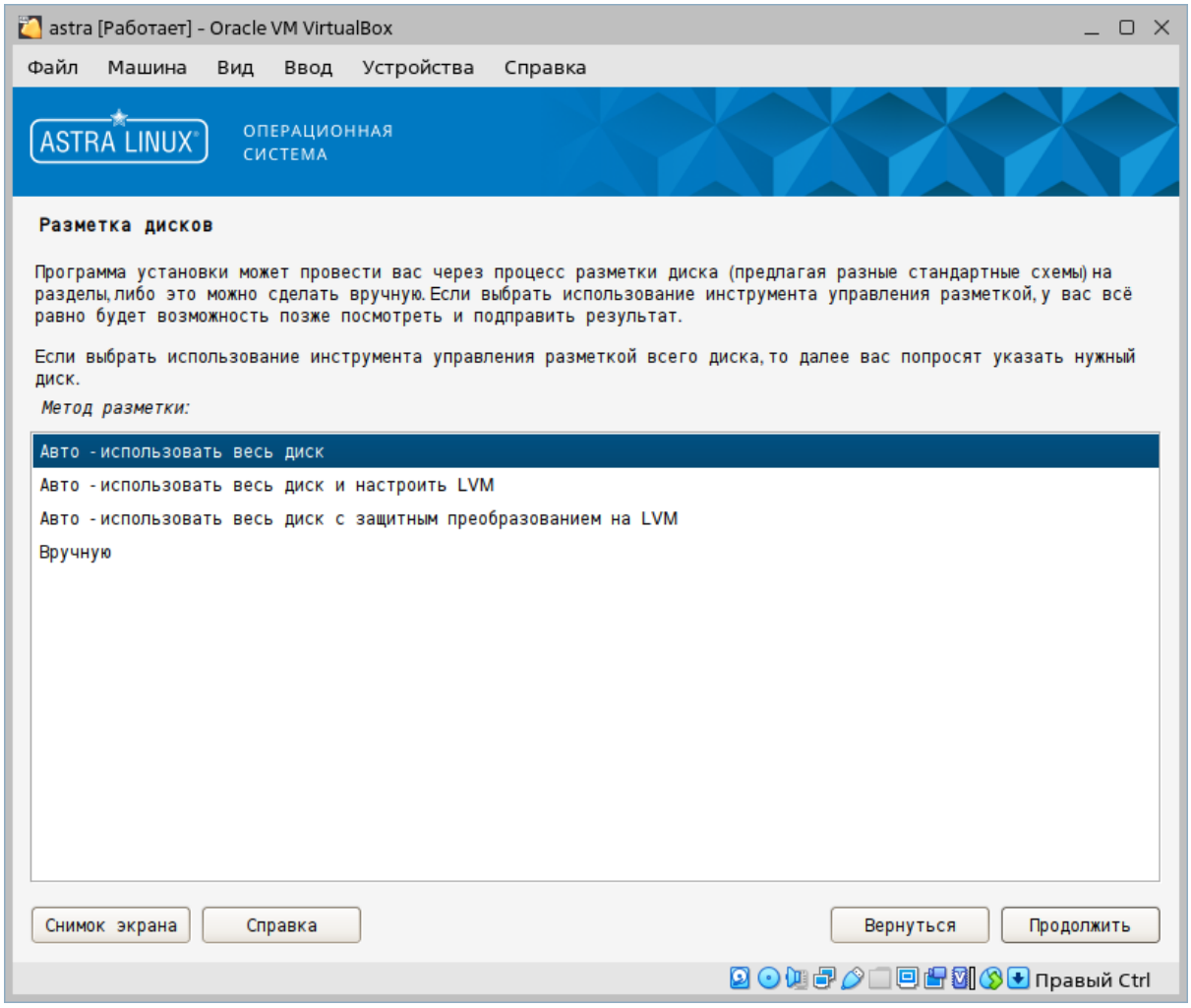


2.3.11 Шаг 11. Разметка дисков

Мастер установки предлагает несколько вариантов разметки:

- Авто – использовать весь диск (по умолчанию).
- Авто – использовать весь диск и настроить LVM.
- Авто – использовать весь диск с защитным преобразованием на LVM.
- Вручную.

В продуктивной среде рекомендуется использовать LVM, но для первой установки можно оставить значение по умолчанию *Авто – использовать весь диск*.

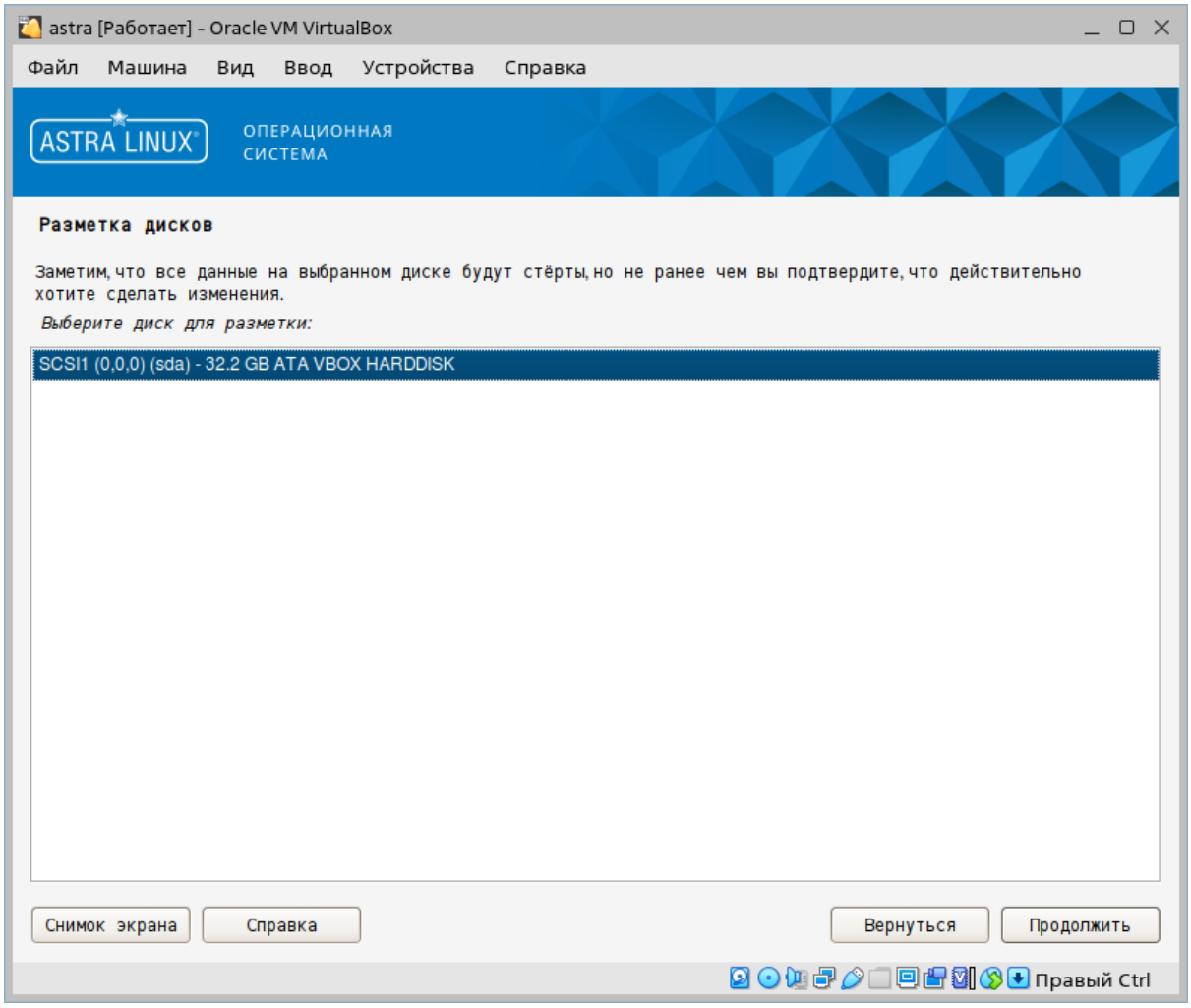


Важно: Разметка дисков — это важный этап установки системы Linux. Из-за особенностей файловых систем, использующихся в Linux, при непродуманной разметке диска на компьютере могут возникнуть проблемы с производительностью, или он не сможет создавать новые файлы на диске даже при наличии свободного места (исчерпание количества inodes).

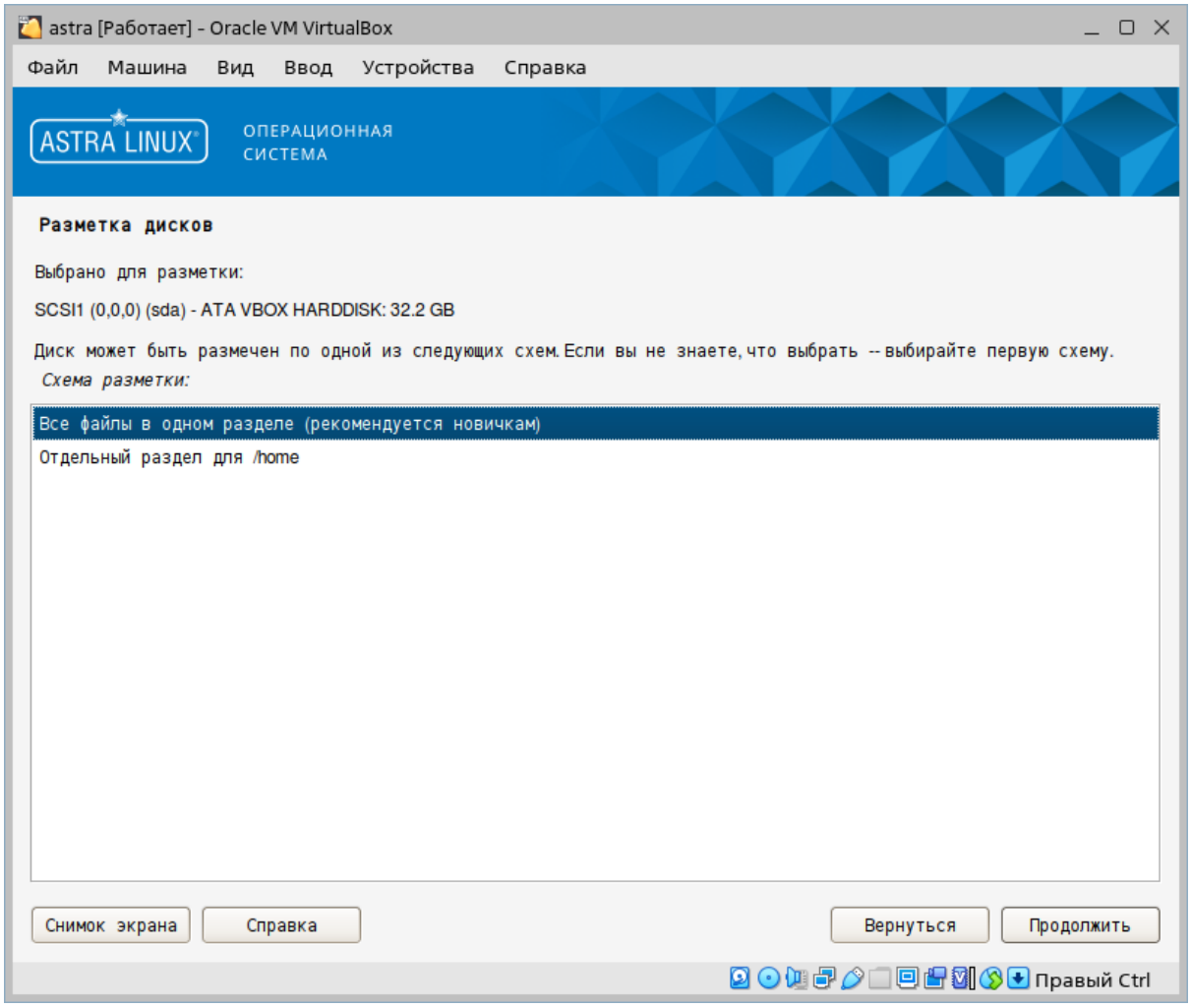
При разметке диска необходимо учитывать назначение компьютера, какие роли он будет исполнять, и оценивать планируемую нагрузку и иерархию файловой системы. Все эти вопросы будут подробно рассмотрены в модулях, посвященных файлам и каталогам, файловым системам и оптимизации производительности.

Процедура разметки состоит всего из 5 шагов:

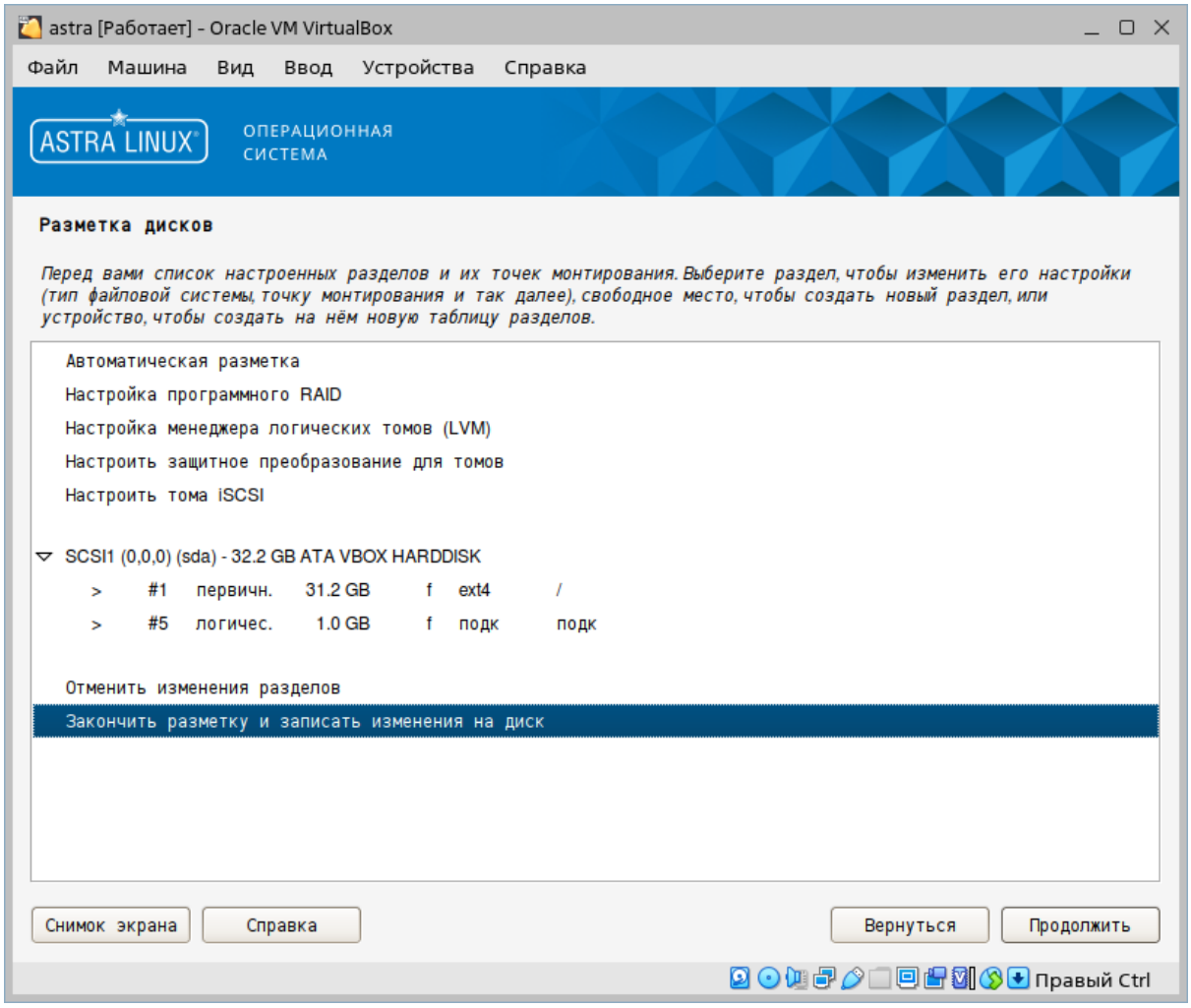
1. Выбор диска. **Если у вас несколько дисков, важно не ошибиться в выборе!**



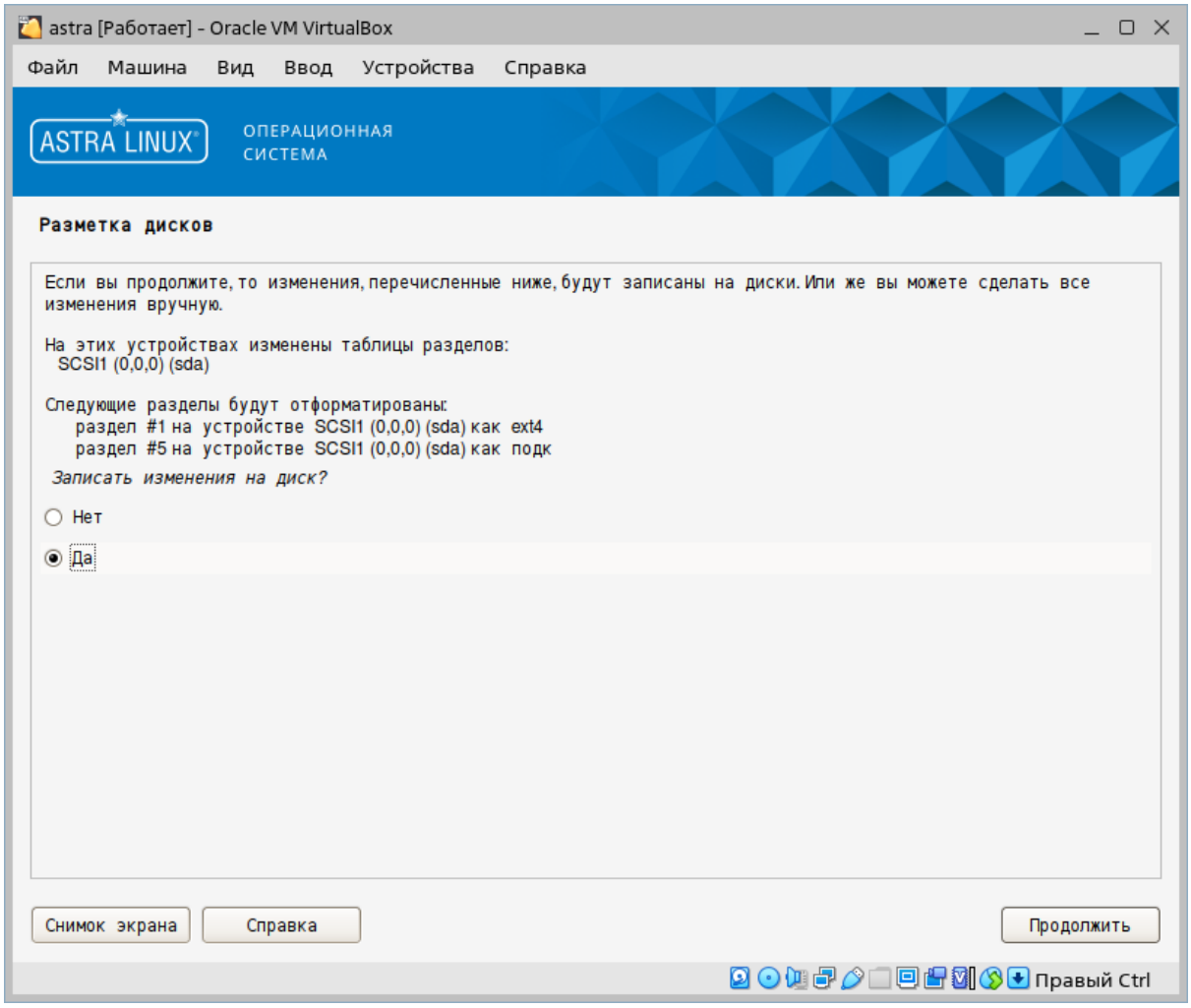
2. Выбор схемы разметки диска. Все файлы в одном разделе или выделенный раздел для `/home`. Выберем опцию *Все файлы в одном разделе (рекомендуется новичкам)*.



3. Ознакомление с настройками разделов.



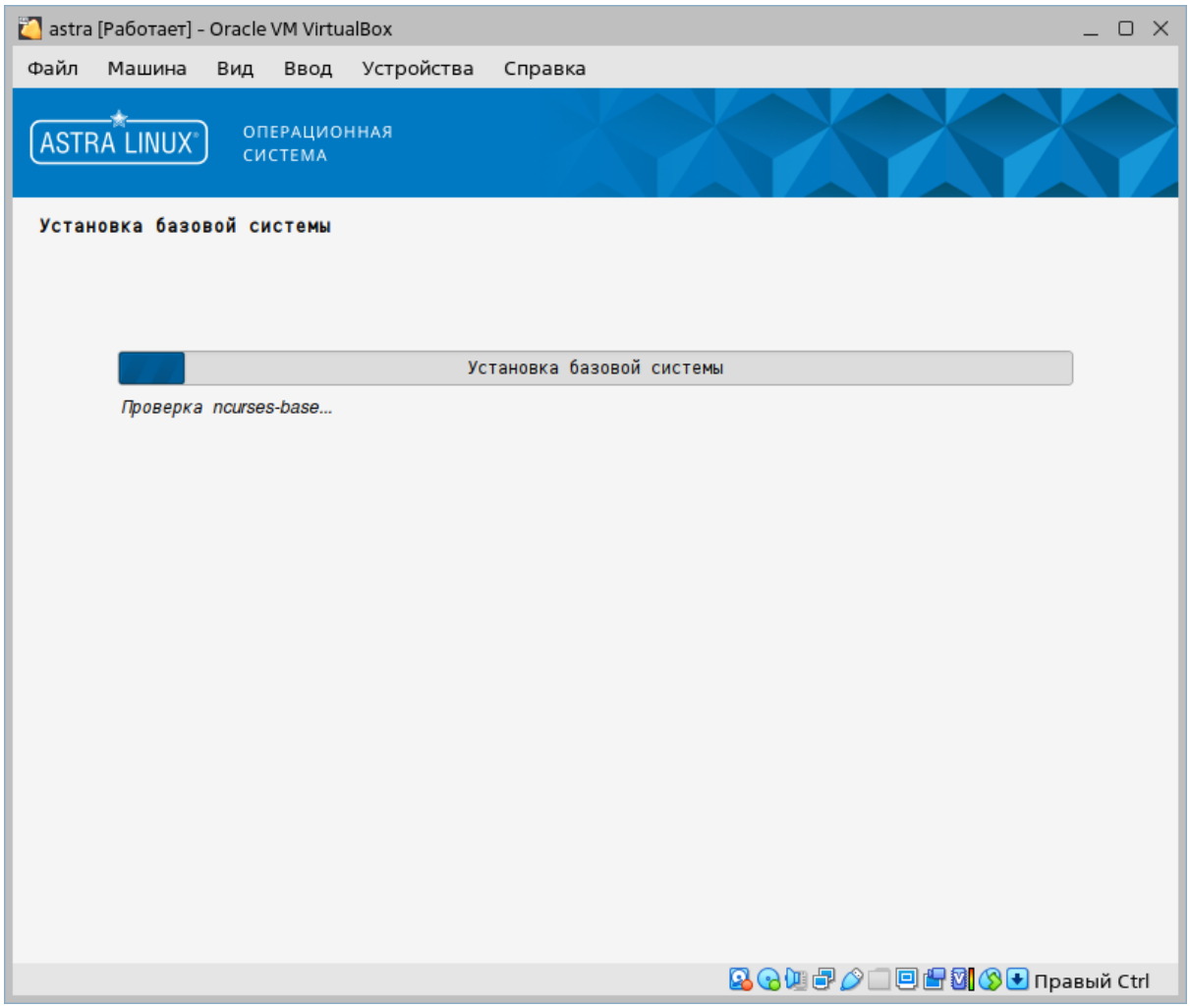
4. Подтверждение применения настроек. После подтверждения **будут выполнены необратимые операции по разметке диска,** и **вся имеющаяся информация на диске будет уничтожена**!



5. Применение изменений и их запись на диск.

2.3.12 Шаг 12. Установка базовой системы.

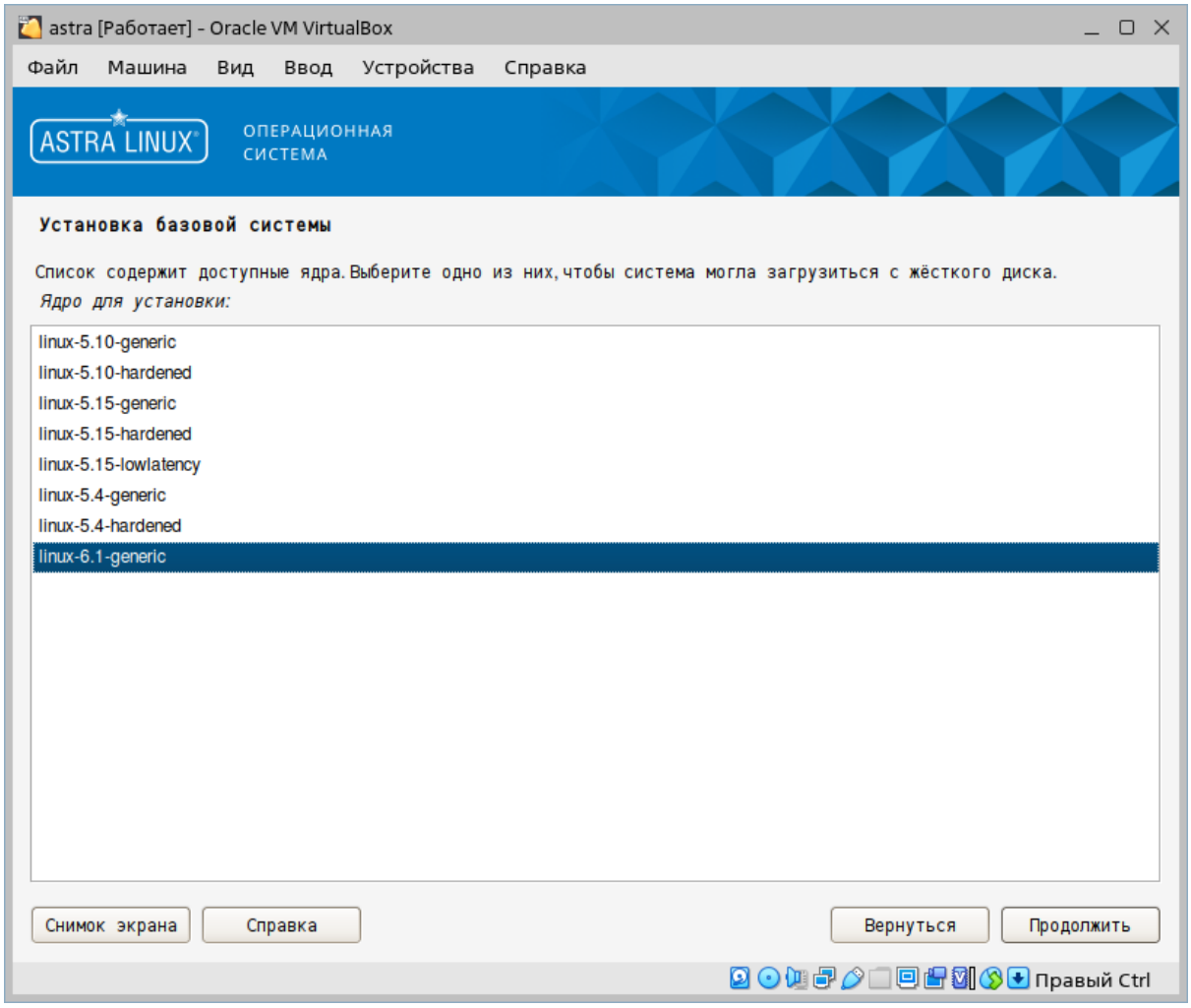
2.3.12.1 Установка базовой системы: начало установки (автоматически)



2.3.12.2 Установка базовой системы: выбор ядра ОС

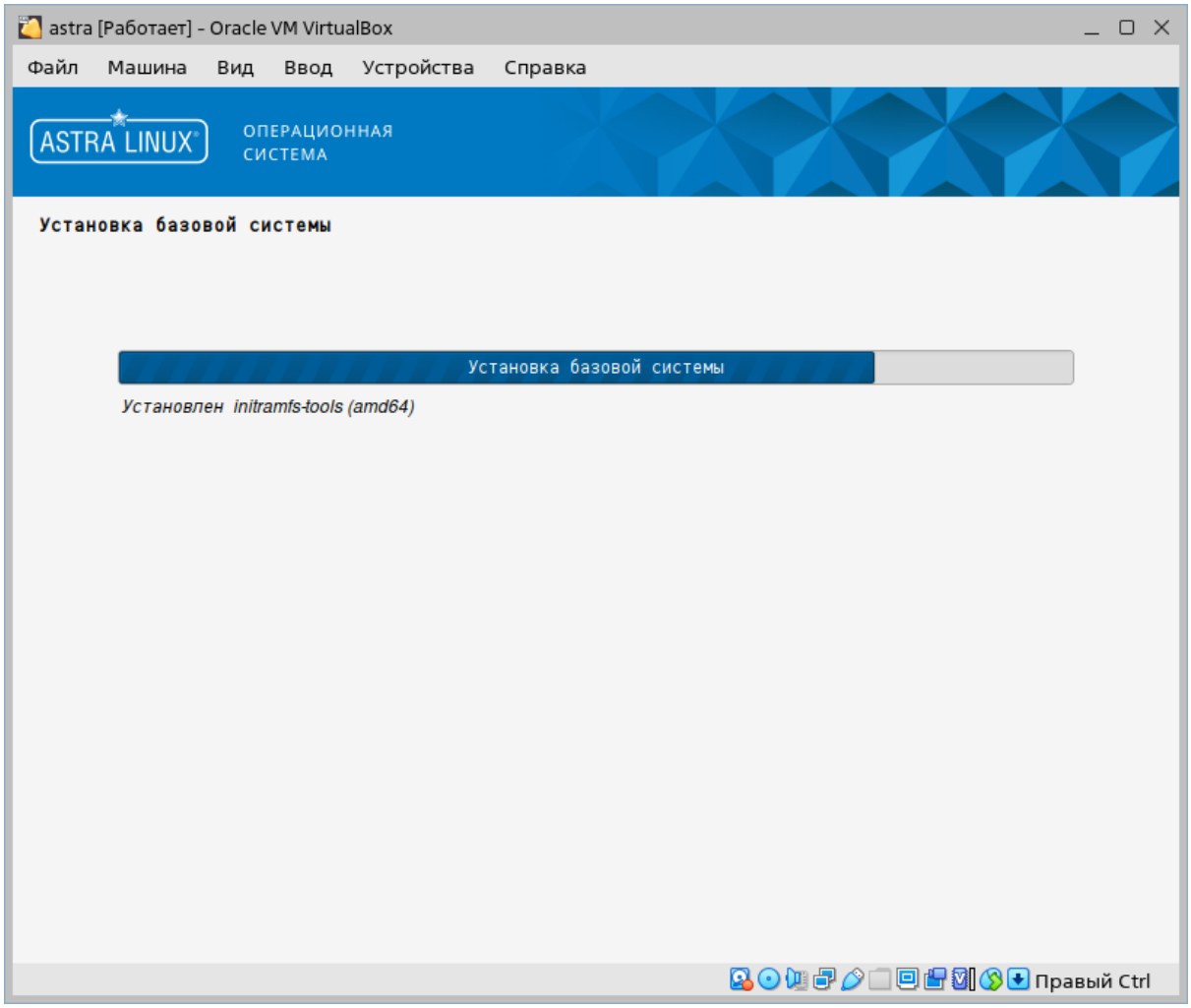
На этой странице есть возможность выбрать версию ядра Linux и вариант сборки:

- **generic** — сборка общего назначения;
- **hardened** — сборка предоставляет возможности усиленной самозащиты за счет дополнительных возможностей по очистке остаточной информации;
- **lowlatency** — сборка с увеличенной до 1000 Гц частотой переключения задач, что обеспечивает ускоренную реакцию системы, однако вызывает увеличение накладных расходов на переключение задач. Субъективное улучшение работы компьютера происходит за счет снижения полезной производительности и повышения энергопотребления.

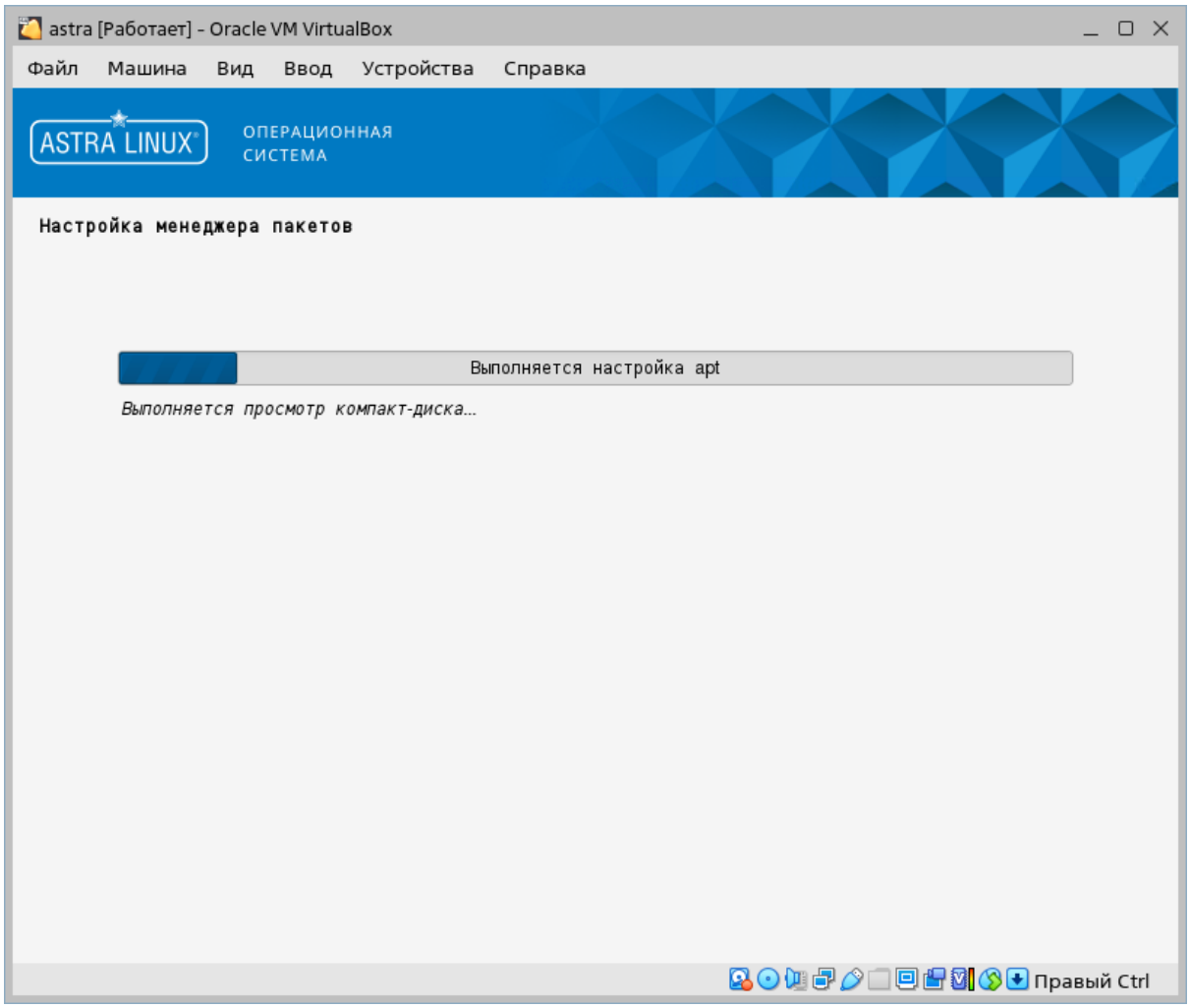


Примечание: Следует учесть, что некоторые продукты имеют ограничения по используемым версиям ядра Linux, например, ALD Pro нельзя устанавливать на версию hardened. Подробнее вопросы рассмотрены в *Модуль 18. Управление устройствами и модулями ядра ОС*.

2.3.12.3 Установка базовой системы: завершение установки (автоматически)



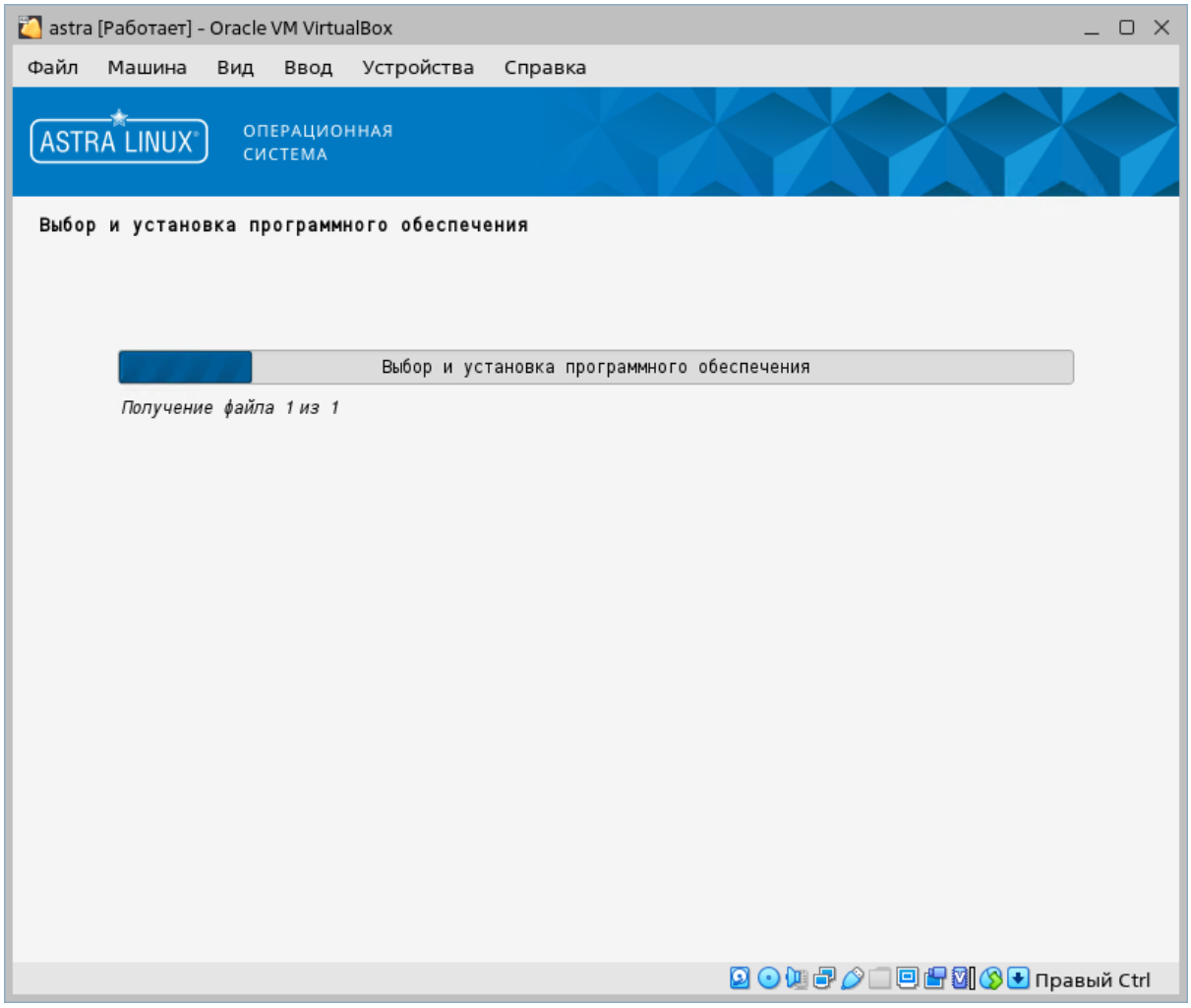
2.3.13 Шаг 13. Настройка менеджера пакетов (автоматически)



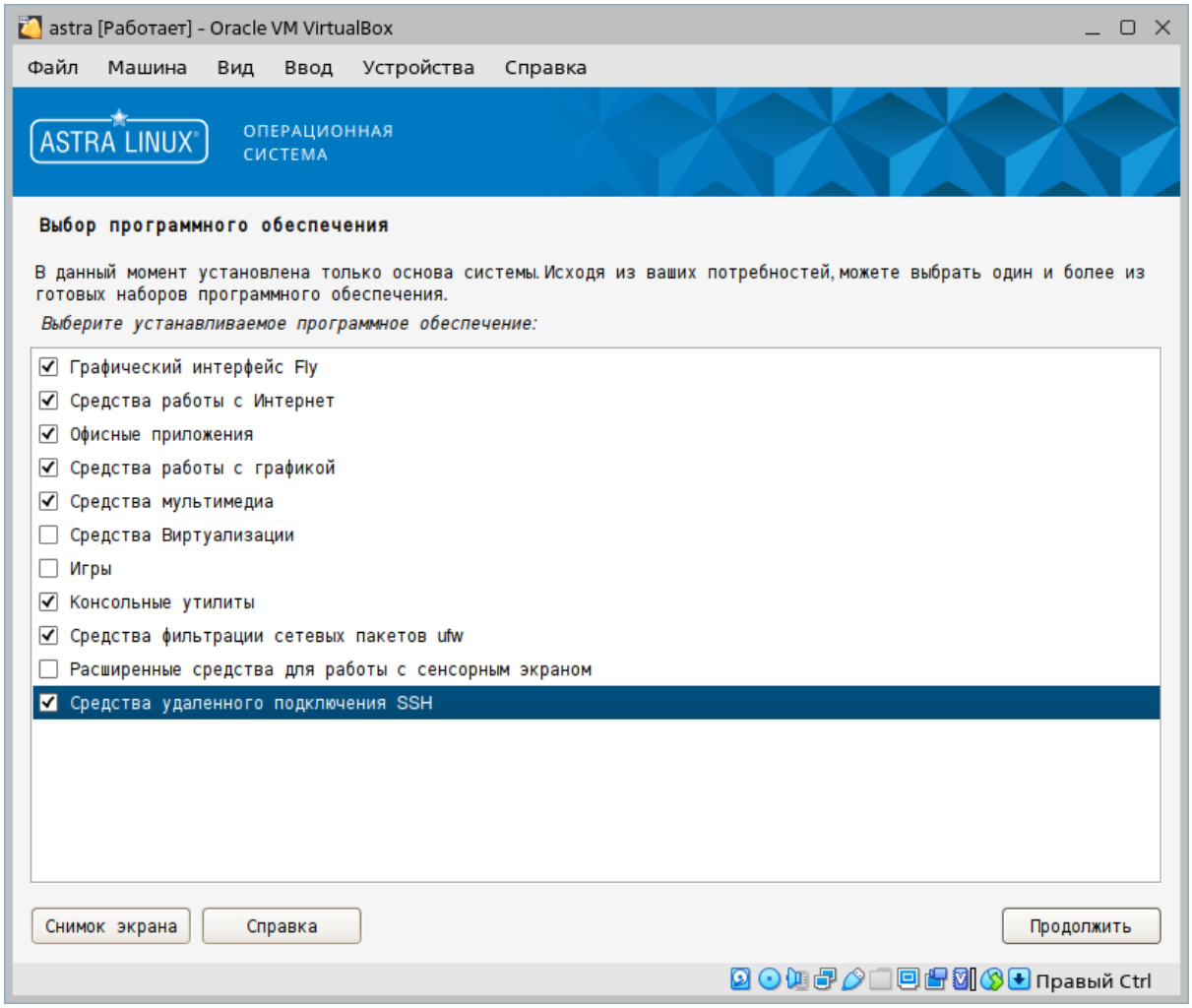
2.3.14 Шаг 14. Выбор и установка дополнительного ПО

2.3.14.1 Выбор и установка дополнительного ПО

Сначала выполняется автоматическая подготовка мастера.



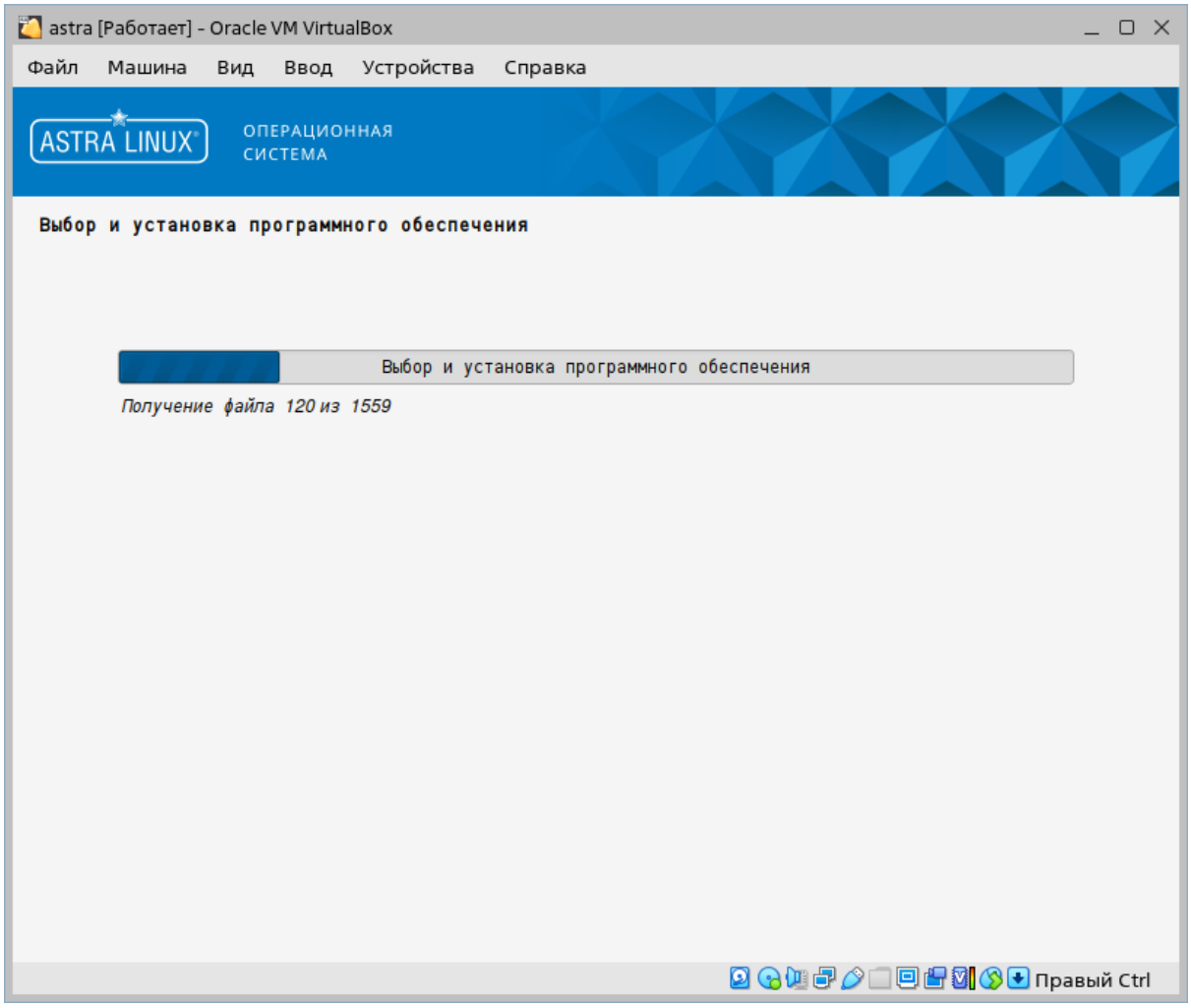
Далее предлагается краткий список программного обеспечения, доступного «из коробки» в составе основного дистрибутива ОС. Для удобства приложения сгруппированы по разделам, что позволяет выбрать сразу комплект тематически связанных программ. Подробный состав комплекта можно посмотреть в статье wiki Astra Linux. Рекомендуем оставить набор программного обеспечения по умолчанию и добавить только **Средства удаленного подключения SSH**.



Примечание: В мире Linux, в отличие от Windows, дистрибутивы программ принято называть пакетами. Управление пакетами подробно рассматривается в *Модуль 15. Управление ПО в Astra Linux*

2.3.14.2 Выбор и установка дополнительного ПО: установка ПО (автоматически)

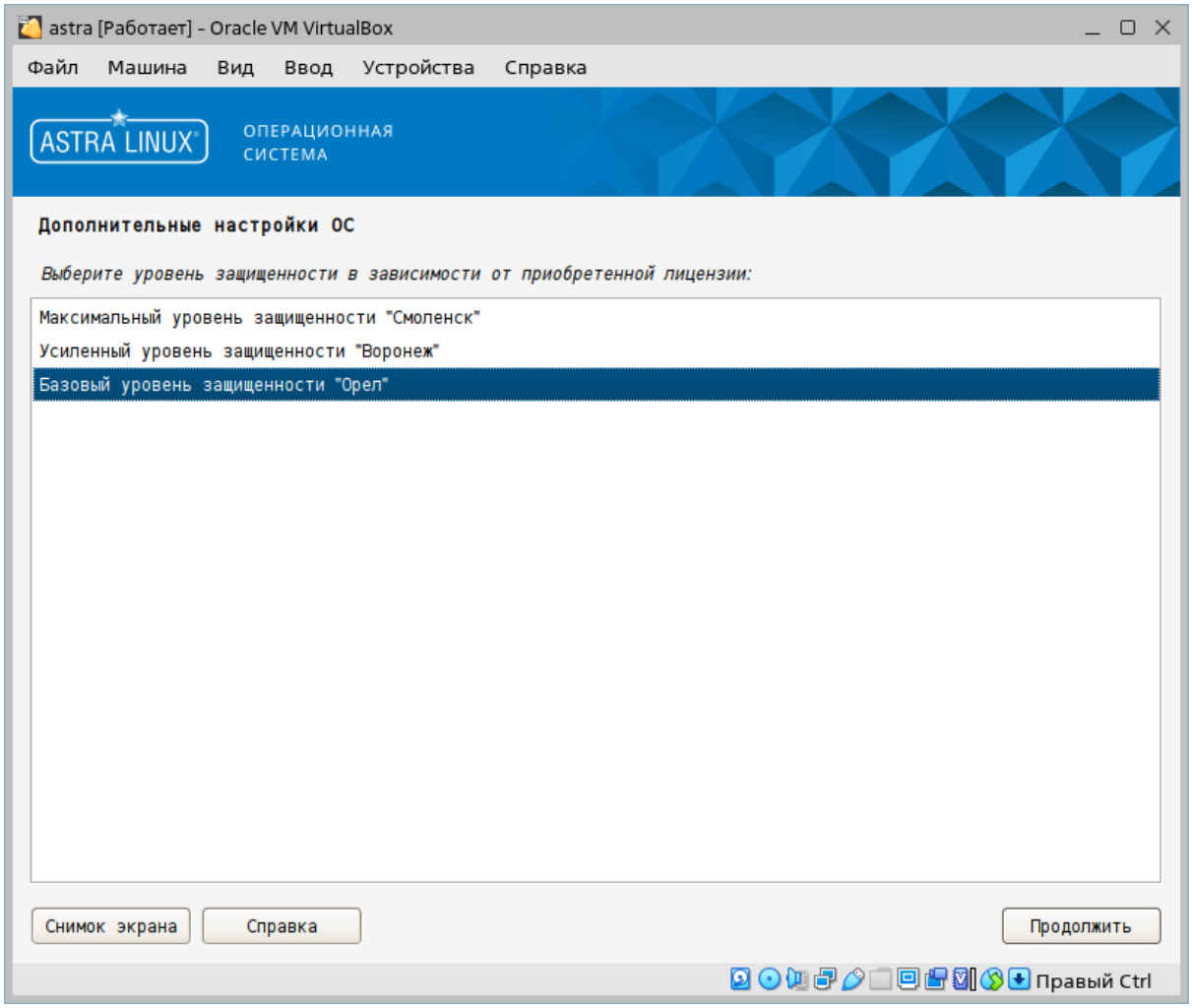
Далее выполняется установка выбранных приложений, что может занять продолжительное время, поэтому наберитесь терпения.



2.3.15 Шаг 15. Дополнительные настройки ОС

2.3.15.1 Дополнительные настройки ОС: выбор уровня защищенности

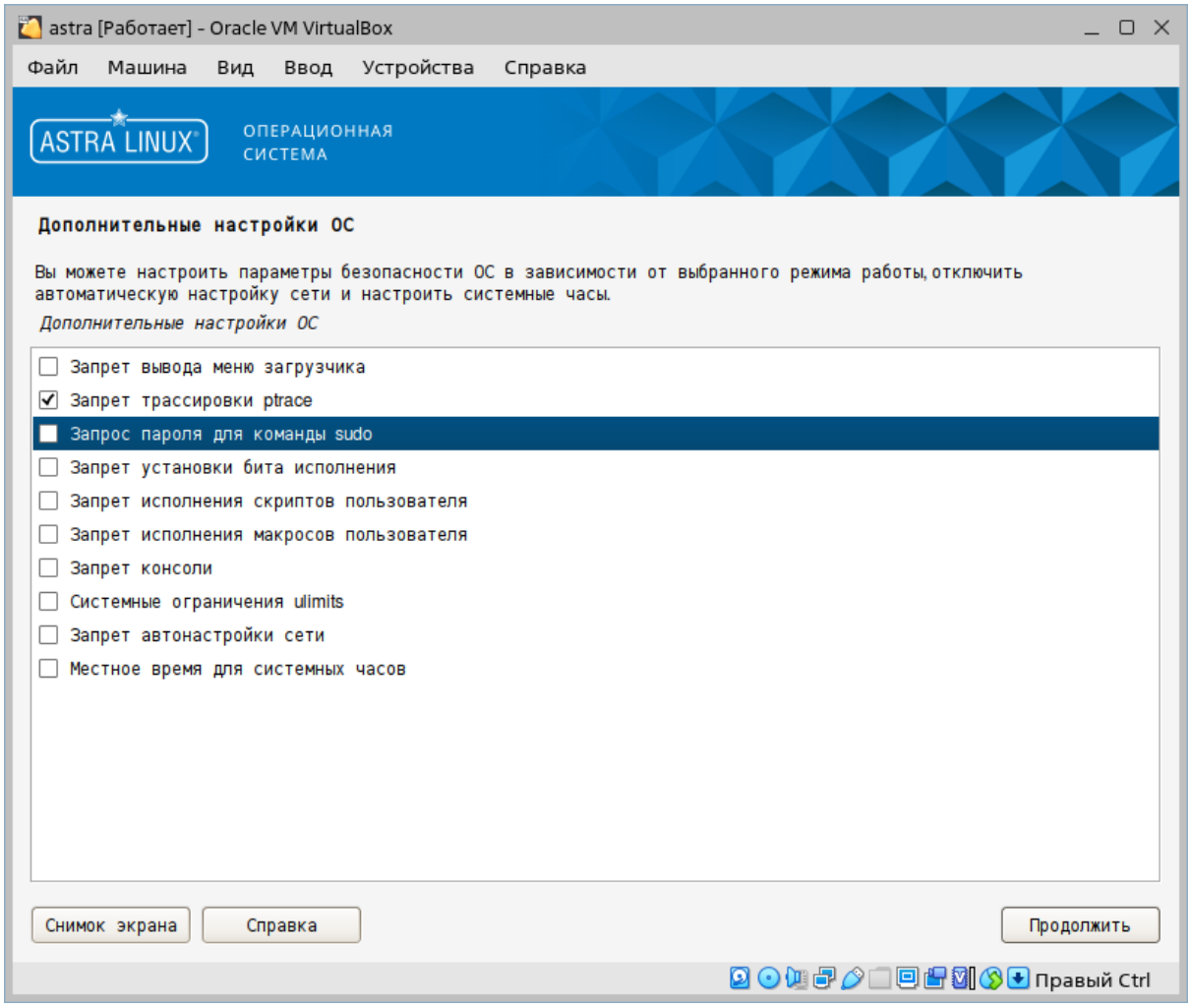
Уровень защищенности должен быть выбран в зависимости от потребностей в информационной защите и от типа приобретенной лицензии. Для первой установки выберем базовый уровень «Орел».



Примечание: Вопросы про информационную безопасность не входят в список тем этого курса, но вы можете обратиться к **Руководству по КСЗ** на странице https://wiki.astralinux.ru/pages/viewpage.action?pageId=137563555.

2.3.15.2 Дополнительные настройки ОС: выбор параметров безопасности

На этапе выбора параметров безопасности можно оставить все значения по умолчанию или отключить **Запрос пароля для команды sudo** для удобства.

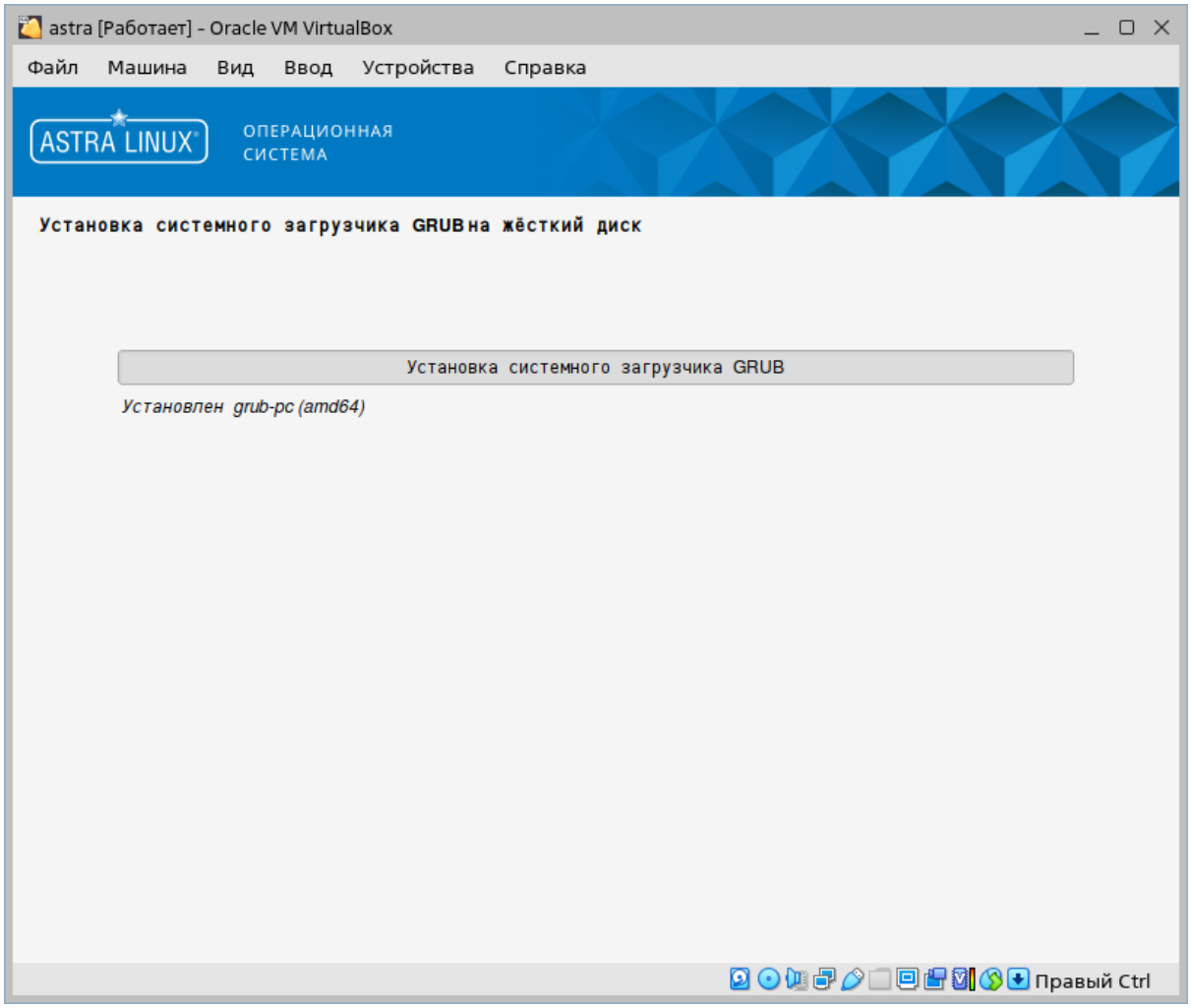


2.3.16 Шаг 16. Установка системного загрузчика GRUB

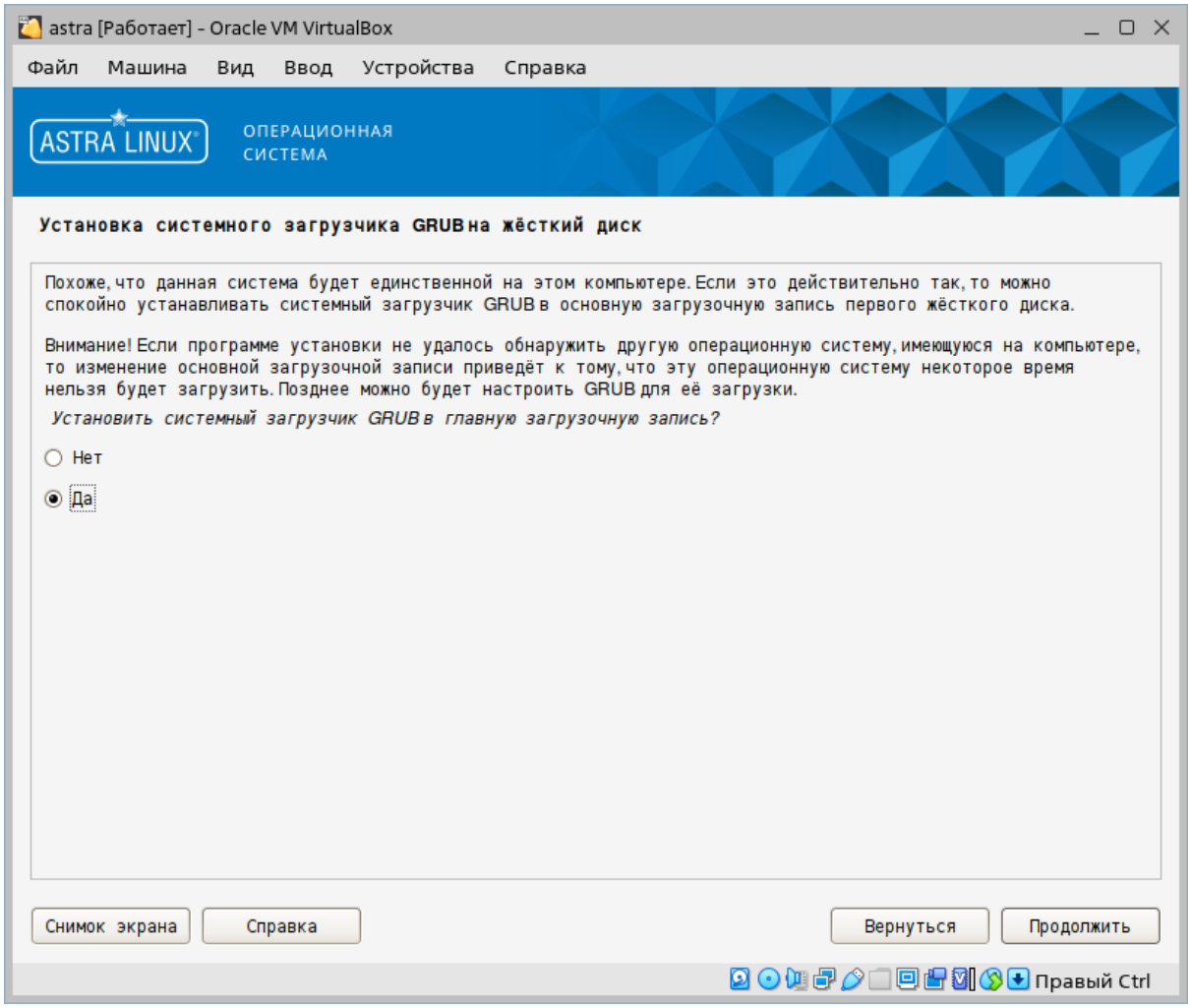
GRUB (*англ. GRand Unified Bootloader*) — загрузчик операционной системы от проекта GNU, который позволяет иметь на одном компьютере несколько операционных систем и при включении компьютера выбирать одну из них для загрузки.

Примечание: Вопросы про этапы загрузки ОС и загрузчик GRUB будут подробно рассмотрены в дальнейшем, см. *Модуль 17. Загрузка системы и управление службами systemd*.

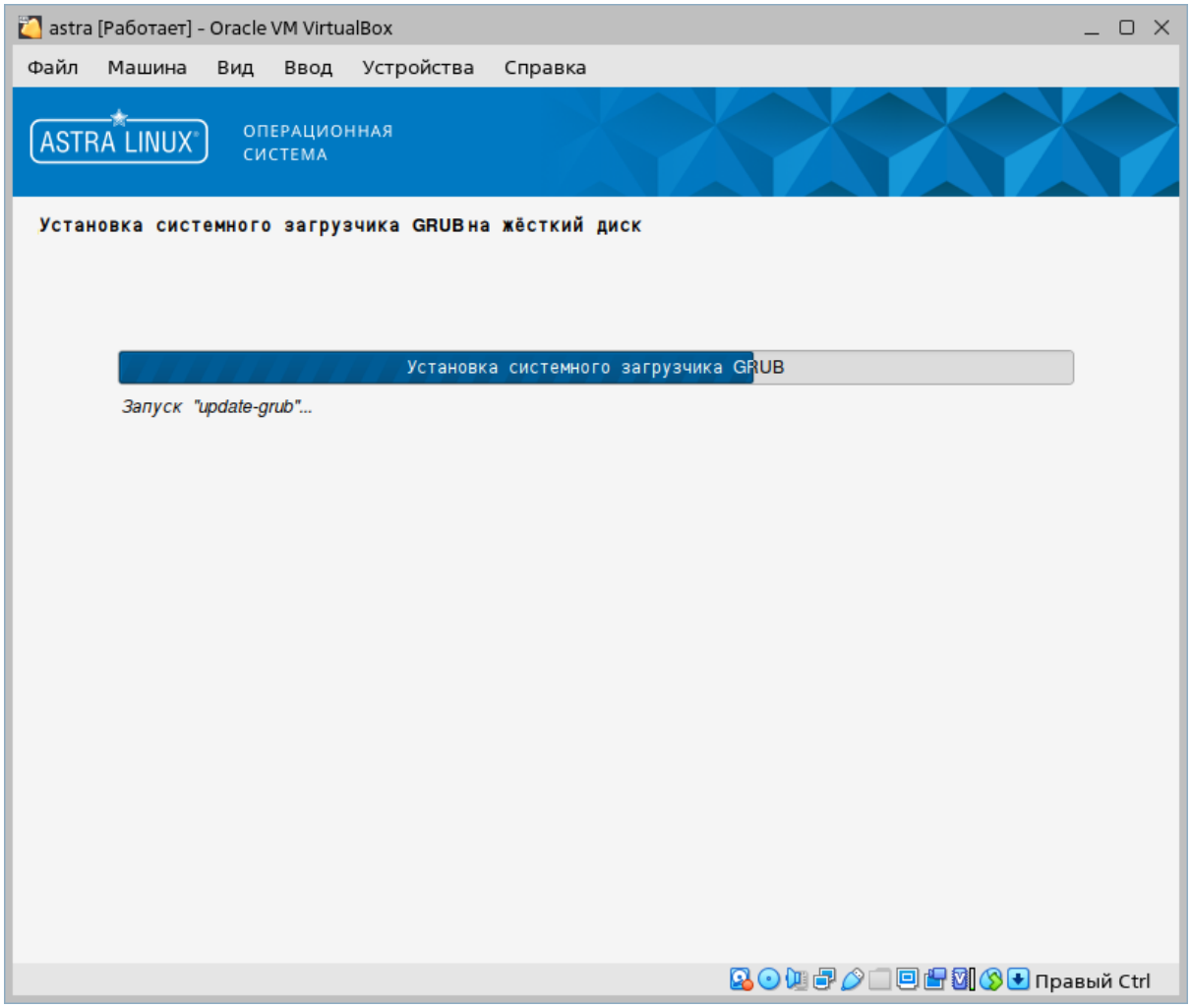
2.3.16.1 Установка системного загрузчика GRUB: подготовка к установке (автоматически)



2.3.16.2 Установка системного загрузчика GRUB: подтверждение установки

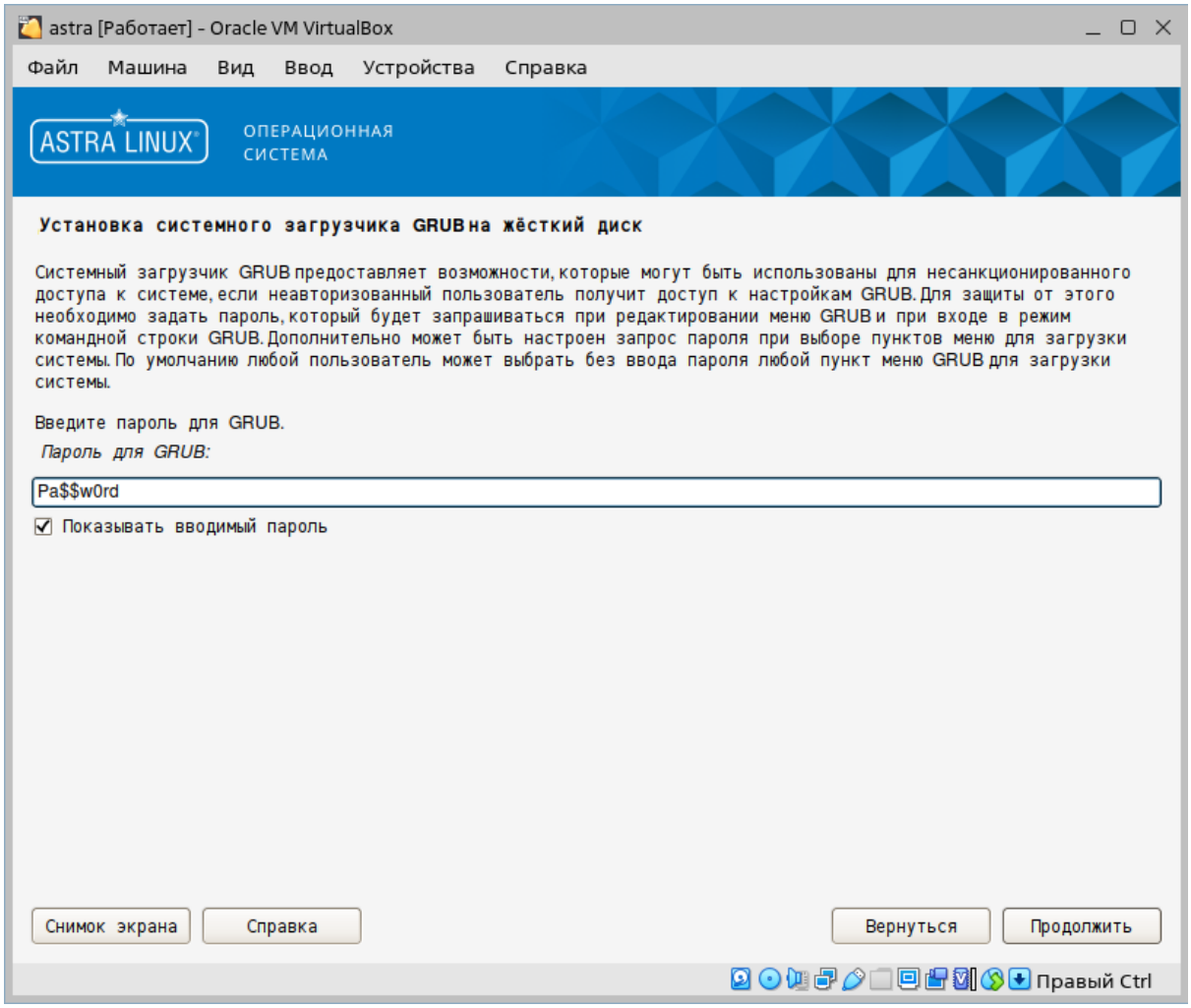


2.3.16.3 Установка системного загрузчика GRUB: установка (автоматически)

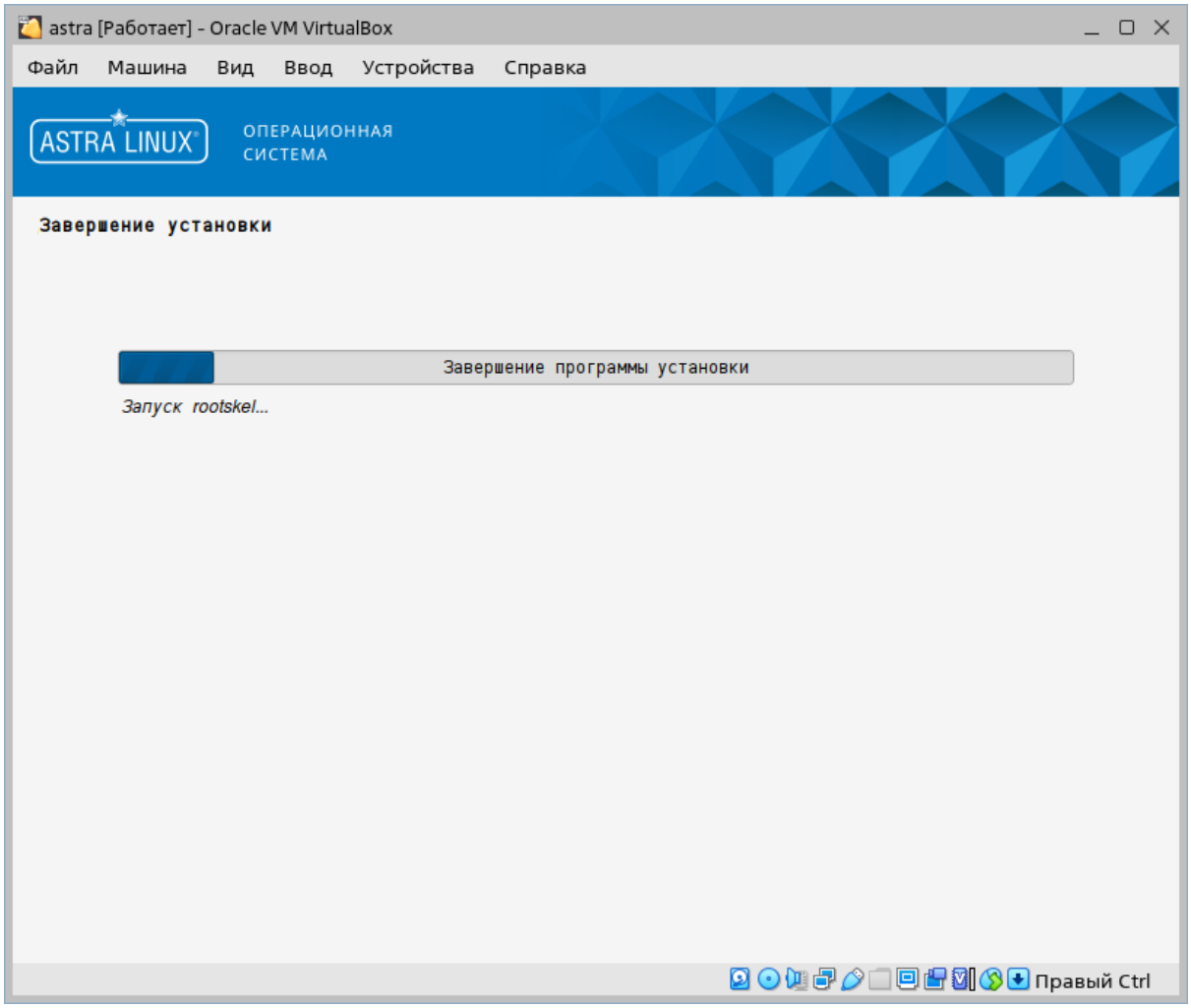


2.3.16.4 Установка системного загрузчика GRUB: установка пароля для GRUB

Для повышения безопасности системы от несанкционированного изменения настроек GRUB необходимо установить пароль, что является обязательным шагом при установке системы.

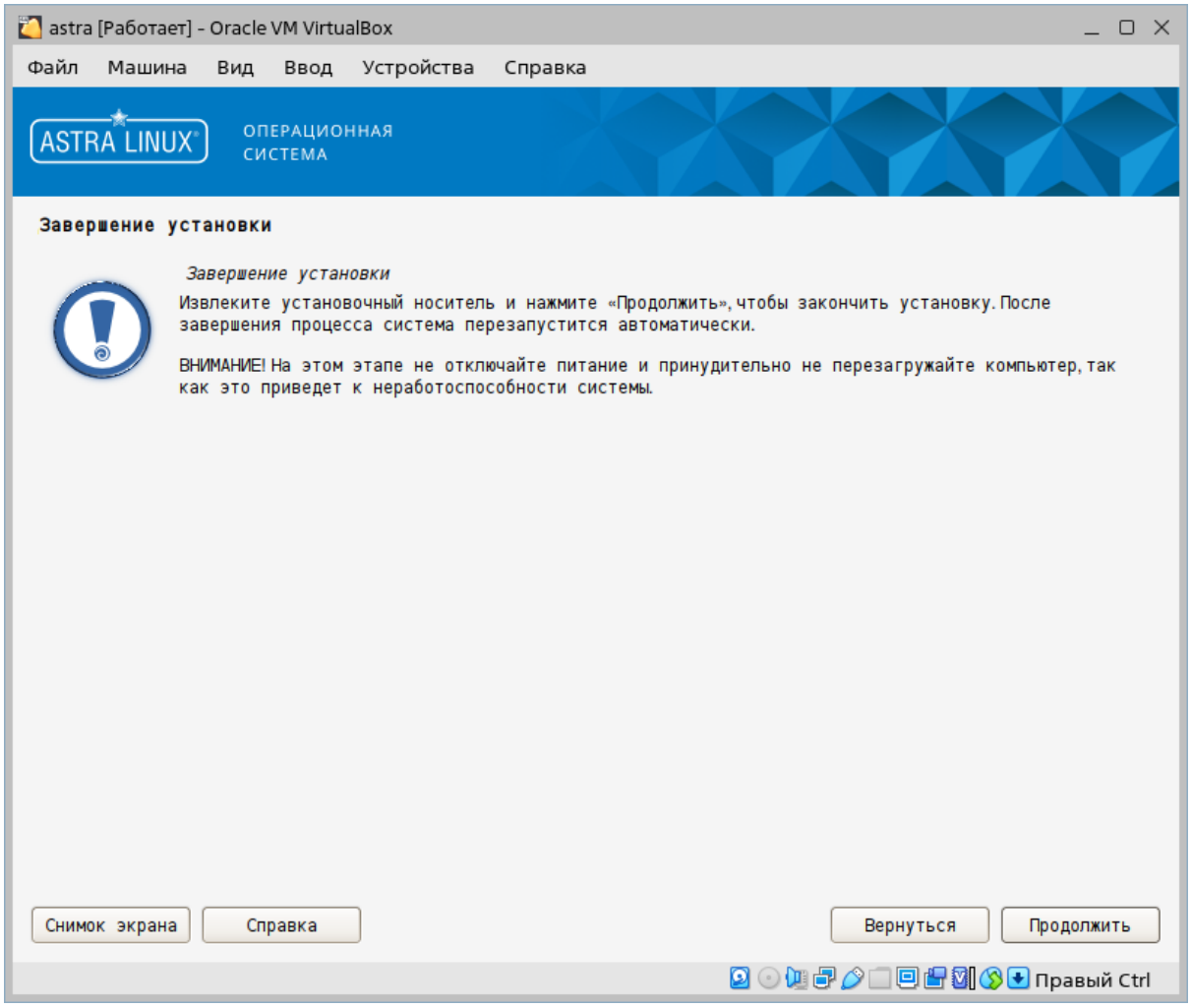

astra [Работает] - Oracle VM VirtualBox
Файл Машина Вид Ввод Устройства Справка
ASTRA LINUX
ОПЕРАЦИОННАЯ СИСТЕМА
Установка системного загрузчика GRUB на жёсткий диск
Системный загрузчик GRUB предоставляет возможности, которые могут быть использованы для несанкционированного доступа к системе, если неавторизованный пользователь получит доступ к настройкам GRUB. Для защиты от этого необходимо задать пароль, который будет запрашиваться при редактировании меню GRUB и при входе в режим командной строки GRUB. Дополнительно может быть настроен запрос пароля при выборе пунктов меню для загрузки системы. По умолчанию любой пользователь может выбрать без ввода пароля любой пункт меню GRUB для загрузки системы.
Введите пароль для GRUB.
Пароль для GRUB:
Pa$$w0rd
Показывать вводимый пароль
Снимок экрана
Справка
Вернуться
Продолжить
Правый Ctrl

2.3.16.5 Установка системного загрузчика GRUB: завершение установки (автоматически)

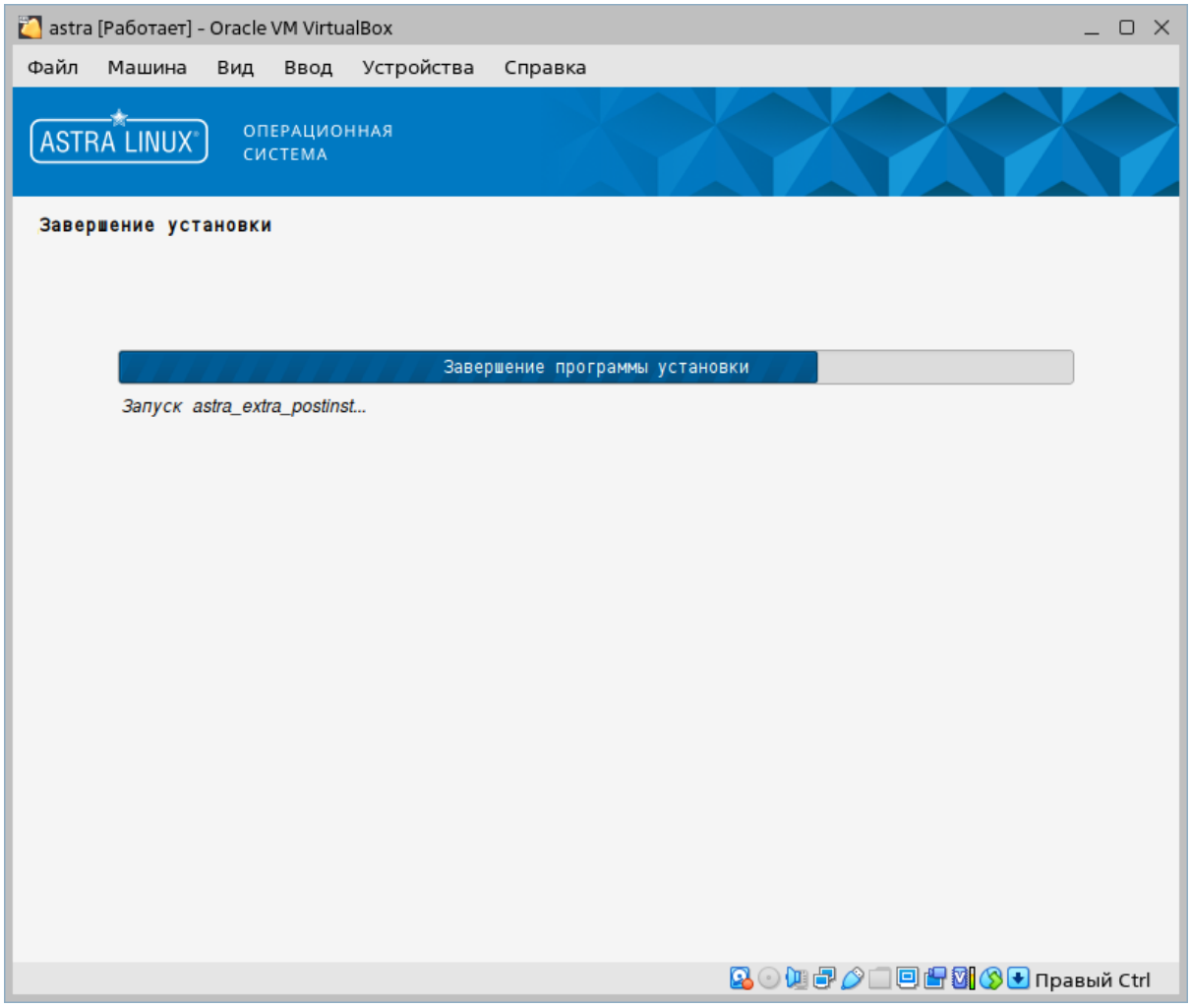


2.3.17 Шаг 17. Завершение установки

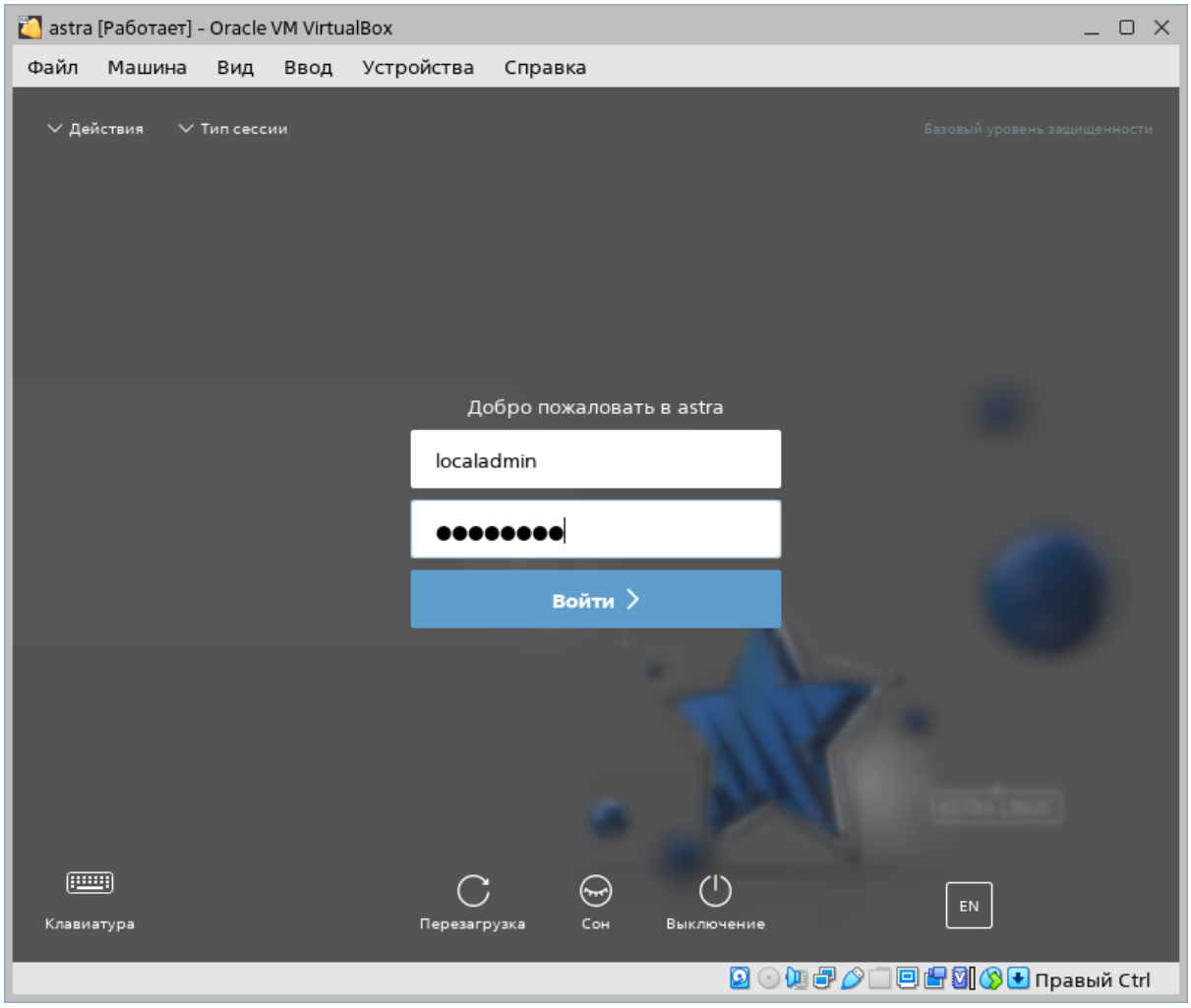
Система просит вас извлечь установочный носитель и предупредит о недопустимости преждевременной перезагрузки или отключения питания компьютера.



Пожалуйста, дождитесь полного завершения установки и автоматической перезагрузки системы.



Поздравляем, вы установили ОС Astra Linux на свой компьютер! После перезагрузки вы попадете в графический интерфейс и увидите приглашение для ввода учетных данных. Для входа в систему введите имя и пароль учетной записи в соответствующие поля и нажмите на кнопку *Войти*. Подробнее о входе в ОС и о работе в графическом интерфейсе вы узнаете в следующем модуле: *Модуль 3. Работа с графическим интерфейсом Fly*.



2.4 Первоначальная настройка Astra Linux

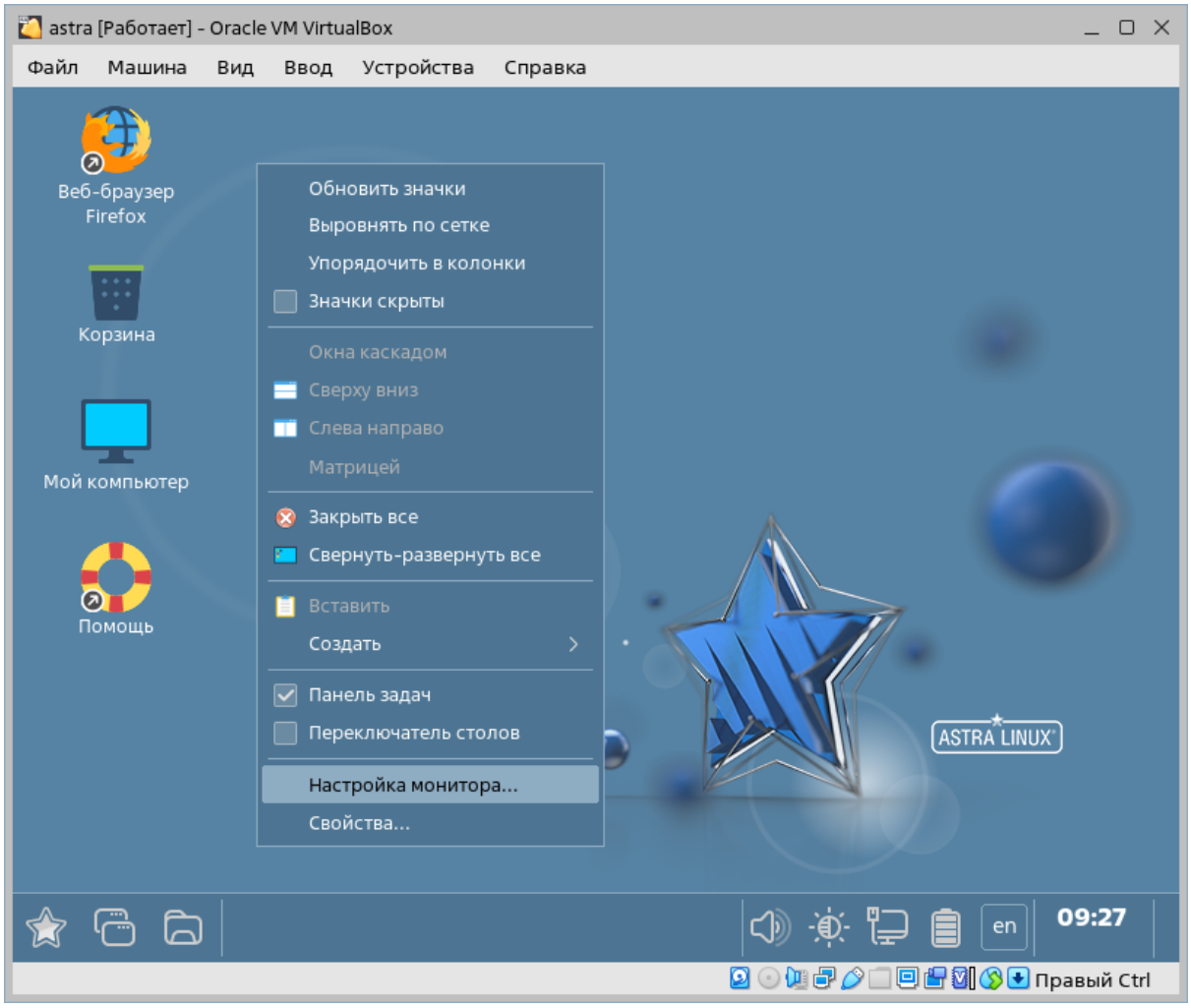
В этом разделе мы не будем заострять внимание на «зачем» и «почему», а сосредоточимся на том, «как» выполнить ряд настроек для обеспечения работы системы в сети и удобства её использования.

Выберите комфортное для себя разрешение экрана:

1. Кликните правой кнопкой мыши на пустом месте рабочего стола и выберите в контекстном меню пункт *Настройка монитора...*. Откроется окно «Настройка монитора» (утилита fly-admin-screen).

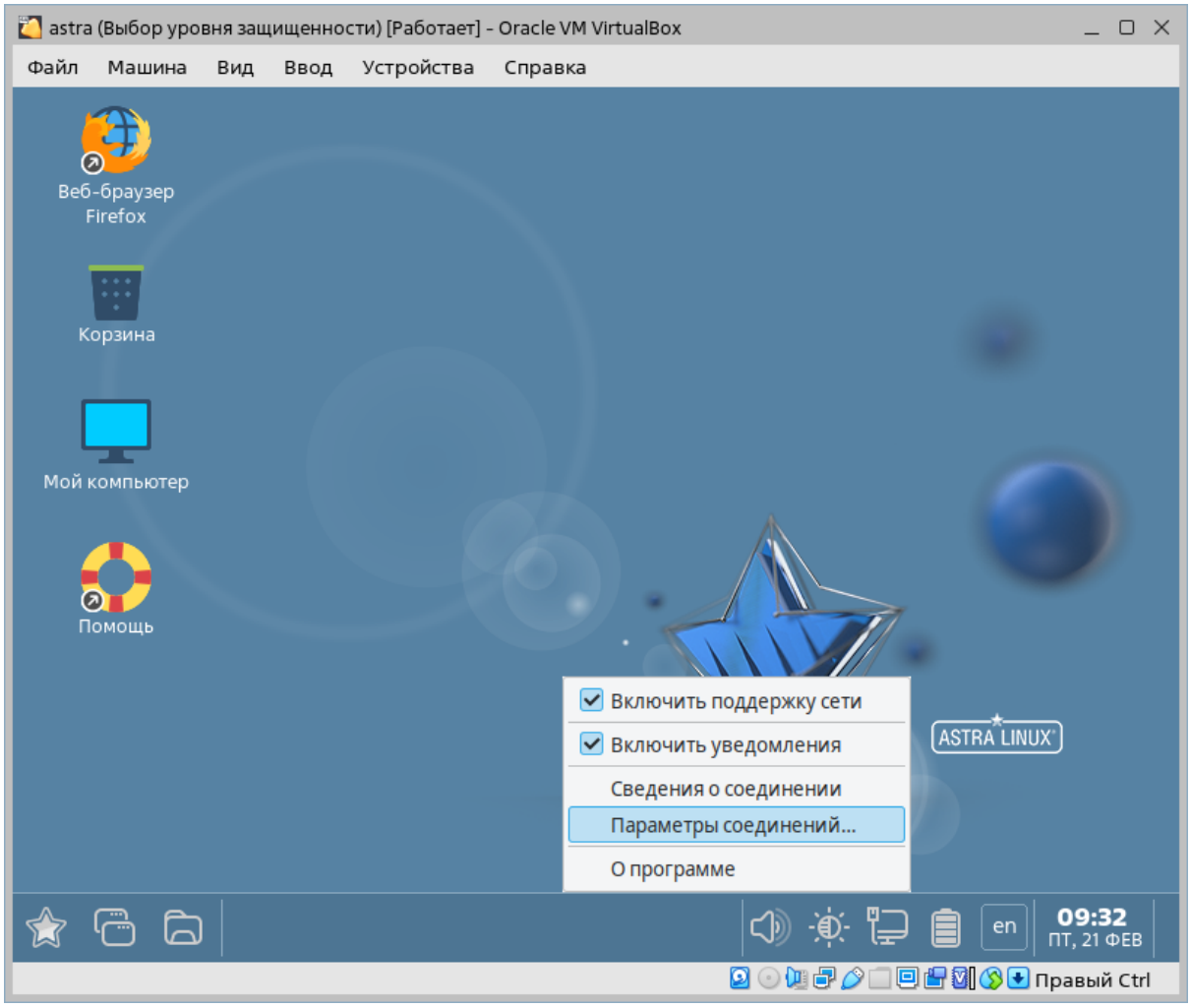
2. Выберите желаемое разрешение экрана.

3. Примите настройки, нажав на кнопку *Да*, которая отмечена стрелкой (текст кнопки может быть скрыт, если окно программы не поместилось на экран).

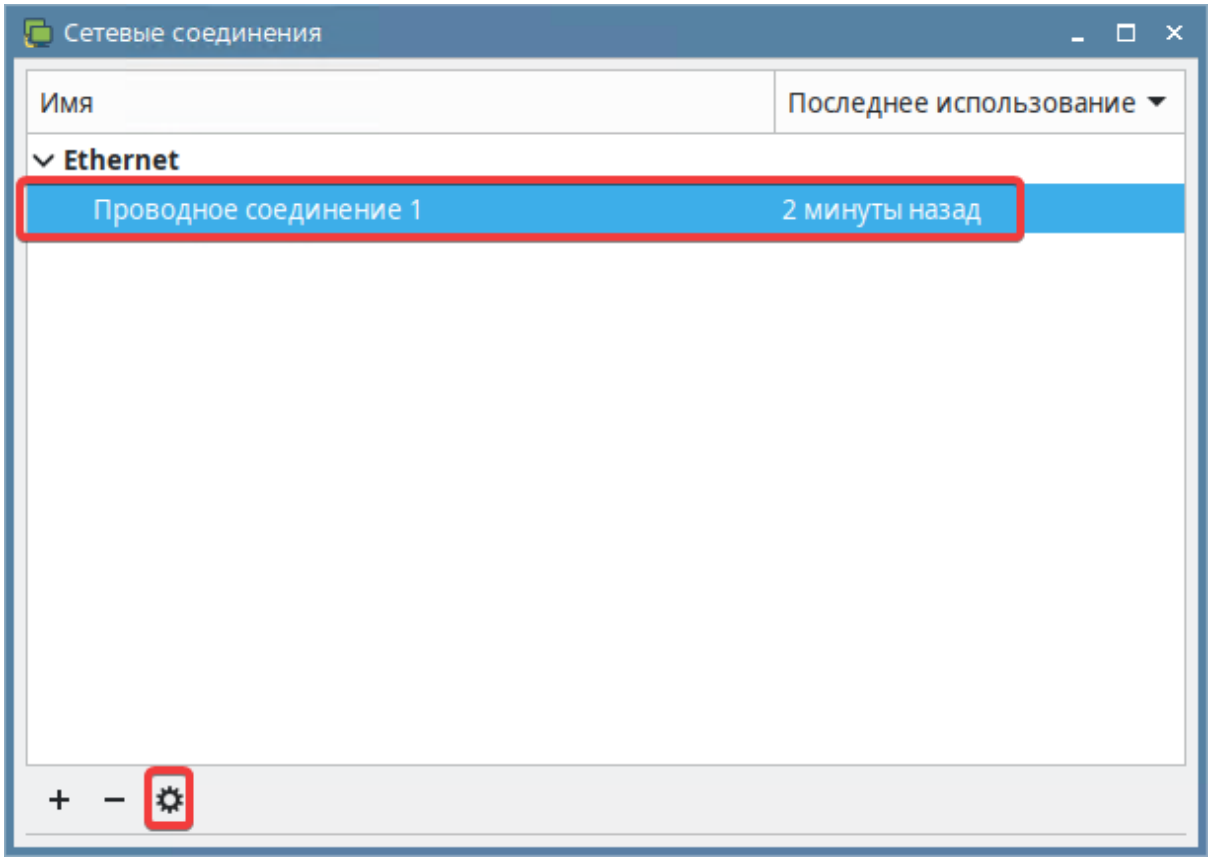


Выполните настройку сети:

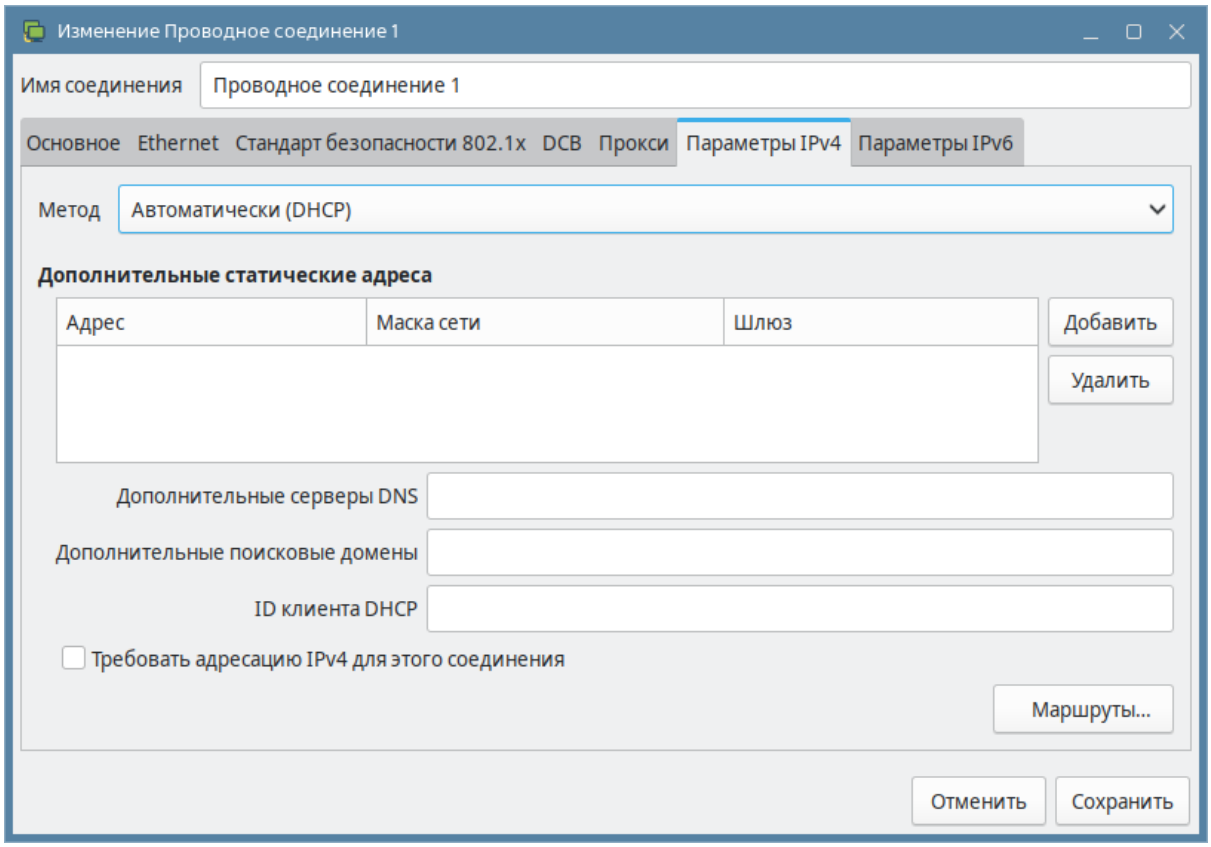
1. Кликните правой кнопкой мыши на значок сети (апплета Network Manager) на панели задач и выберите пункт *Параметры соединений...* .



2. В открывшемся окне сделайте двойной клик по вашему соединению или нажмите кнопку в виде шестеренки в левом нижнем углу.

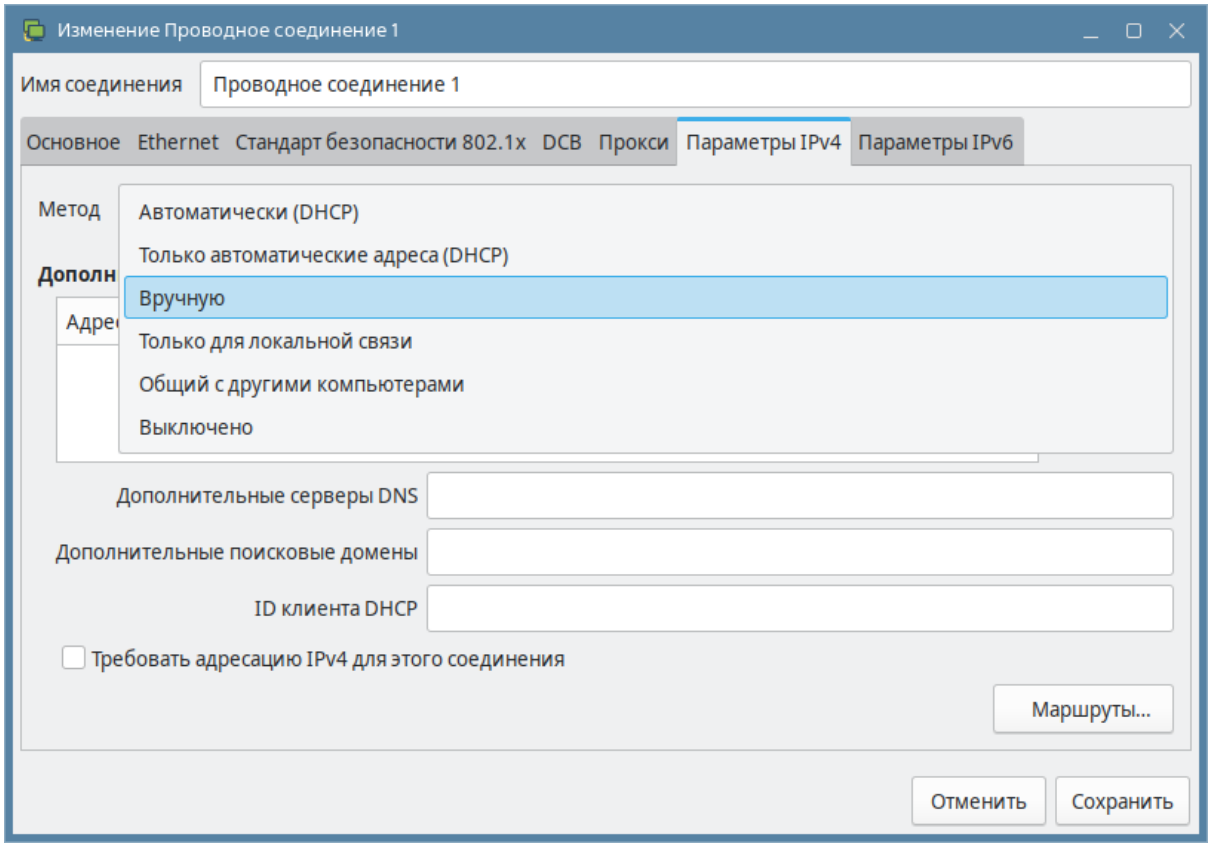


3. В открывшемся окне перейдите на вкладку *Параметры IPv4* :



4. Из основных настроек тут можно выделить:

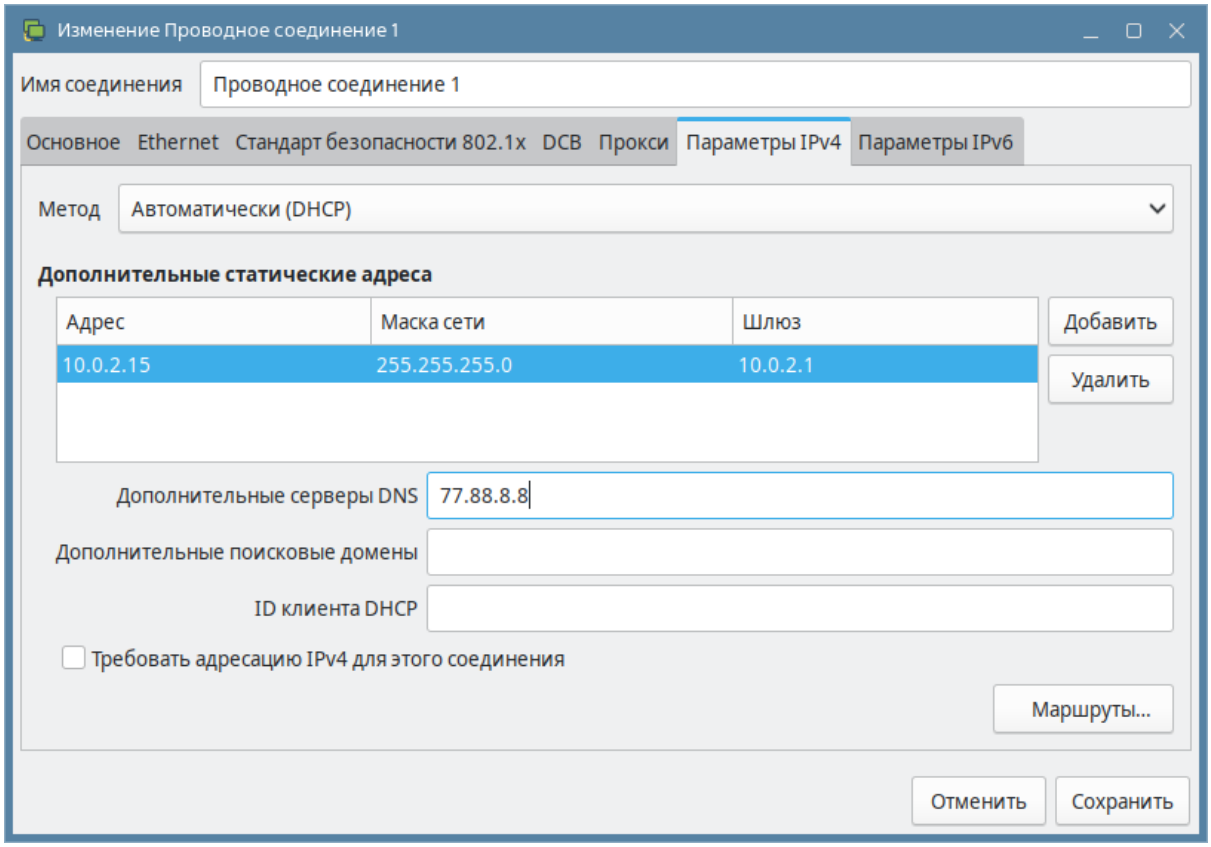
- Метод получения IP-адреса.
- Возможность назначения статического IP-адреса.
- Настройка дополнительных DNS.
- Возможность указания поискового домена (это значение добавляется к коротким именам хостов при разрешении имен).



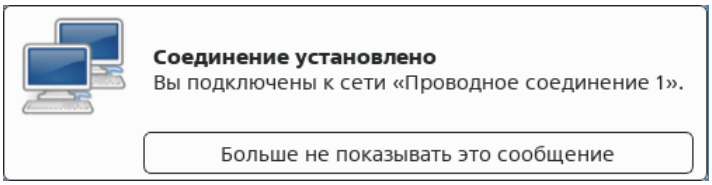
Настроем статический адрес для сетевого соединения:

- Выберите метод получения IP-адреса *Вручную*.
- Кликните на кнопку *Добавить*.
- В появившейся строке введите:
    1. Адрес **10.0.2.15**.
    2. Маска сети **255.255.255.0**.
    3. Шлюз **10.0.2.1**.

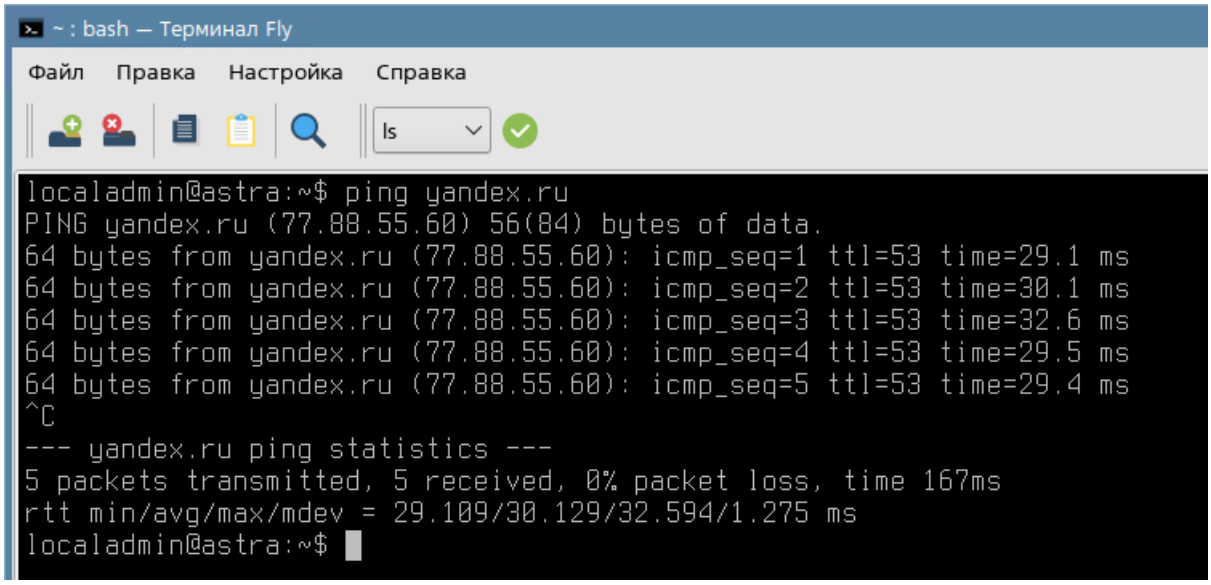
- Укажите DNS-сервер **77.88.8.8**.
- Нажмите на кнопку *Сохранить*.



Чтобы изменения вступили в силу требуется кликнуть правой кнопкой мыши по значку сети в трее (апплет Network Manager) и в контекстном меню сначала выключить поддержку сети, а затем включить ее обратно. Вы должны будете увидеть уведомление о том, что соединение успешно установлено.



Для проверки доступности ресурсов из сети Интернет откройте терминал сочетанием клавиш Alt + T и введите команду `ping yandex.ru`. Успешное выполнение пинга будет свидетельствовать о доступности сервера и работы механизма преобразования имен в IP-адреса. Чтобы прервать работу утилиты, нажмите сочетание клавиш Ctrl + C.



Подключите репозитории **base** и **extended** путем редактирования файла `/etc/apt/source.list` напрямую:

```
sudo nano /etc/apt/sources.list
```

Или откройте файл на редактирование с помощью специальной команды пакетного менеджера apt, чтобы воспользоваться дополнительными проверками при сохранении изменений:

```
sudo apt edit-sources
```

Содержимое файла зависит от версии операционной системы:

- Если вы установили Astra Linux SE 1.7.6 (1.7.6.11), как было рекомендовано ранее, и хотите зафиксировать эту версию, то содержимое файла должно быть следующим:

```
deb https://dl.astralinux.ru/astra/frozen/1.7_x86-64/1.7.6/repository-base/ 1.7_x86-
↪64 main contrib non-free
deb https://dl.astralinux.ru/astra/frozen/1.7_x86-64/1.7.6/repository-extended/ 1.7_
↪x86-64 main contrib non-free
```

- Если вы хотите использовать крайние версии пакетов ALSE 1.7, то содержимое файла должно быть следующим:

```
deb https://dl.astralinux.ru/astra/stable/1.7_x86-64/repository-base/ 1.7_x86-64 main␣
↪contrib non-free
deb https://dl.astralinux.ru/astra/stable/1.7_x86-64/repository-extended/ 1.7_x86-64␣
↪main contrib non-free
```

Для сохранения изменений в редакторе `nano` нажмите сочетание клавиш Ctrl + O и Enter. Для выхода из приложения нажмите Ctrl + X.

Для обновления базы с информацией о доступных пакетах (индекса), выполните в терминале

команду `sudo apt update`. Чтобы обновить уже сами компоненты системы, выполните команду `sudo astra-update -a -r -T`. Ключ `-a` позволяет выполнить обновление в интерактивном режиме, поэтому вы сможете ознакомиться со списком планируемых изменений, чтобы принять окончательное решение о необходимости выполнения обновления. В случае возникновения ошибок выполните команду `sudo apt -f install` и заново запустите команду обновления.

Примечание: Утилита `astra-update` задаёт пакетному менеджеру правильную стратегию обновления, которая предусмотрена разработчиками операционной системы.

Обратите внимание, что Linux-приложения могут использовать иную стратегию. Например, если `astra-upgrade` использует стратегию `--force-confold`, то для ALD Pro применяется стратегия `--force-confnew`, поэтому содержимое некоторых конфигурационных файлов может быть переопределено при выполнении обновления.

Все это является следствием высокой степени повторного использования программного кода, поэтому, работая с Linux, вам нужно обязательно следовать рекомендациям разработчиков по обновлению их приложений.

Для получения дополнительной информации по работе с пакетами и репозиториями программного обеспечения см. *Модуль 15. Управление ПО в Astra Linux*.

После установки обновлений рекомендуется выполнить перезагрузку, которую можно запустить командой `sudo reboot`.

Этот курс подразумевает, что для выполнения практических заданий будет использована виртуальная машина на гипервизоре VirtualBox. Этот гипервизор содержит все необходимые инструменты для настройки и запуска виртуальной машины с ОС Astra Linux и является универсальным, он может быть запущен на машине и с Windows, и с Linux.

Для поддержки всех возможностей по взаимодействию с виртуальной машиной, в том числе использования буфера обмена между размещающей (хостовой) и гостевой машинами, необходимо установить так называемые «гостевые дополнения VirtualBox» (guest additions). Для их установки сделайте следующее:

1. Откройте терминал сочетанием клавиш Alt + T и введите команды:

    a. `sudo apt update` – для обновления списка подключённых репозиториев и пакетов.

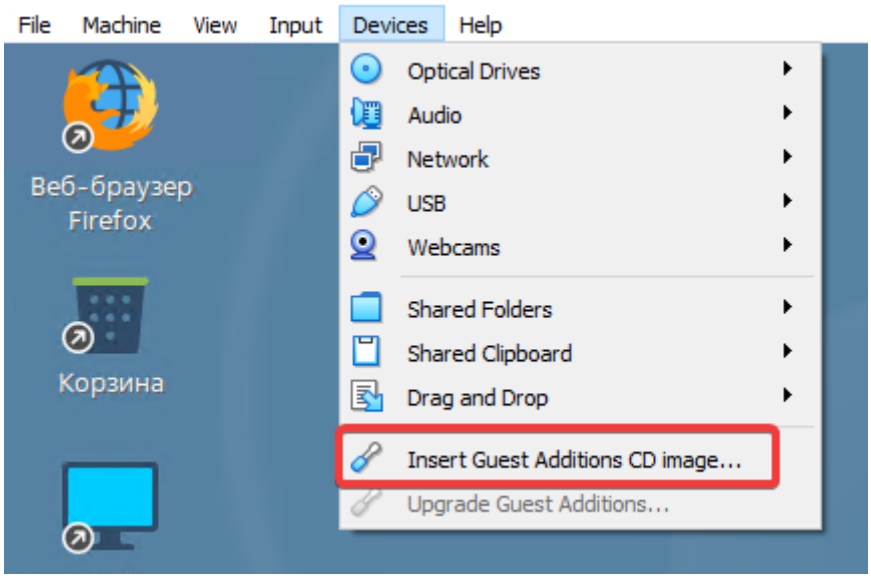
    b. `sudo apt install gcc make linux-headers-6.1` – для установки дополнительных пакетов.

    Обратите внимание, что версия пакета linux-headers должна соответствовать версии используемого ядра. Если вы установили систему с ядром 6.1, то вам нужно будет установить пакет

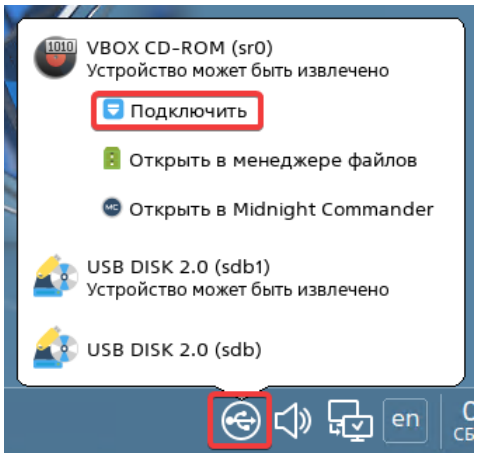
linux-headers-6.1. Для проверки версии ядра можно воспользоваться командой `uname -r`.

Ознакомьтесь с выводом команды и подтвердите установку пакетов. Дождитесь завершения процесса установки.

2. В окне виртуальной машины выберите меню *Devices* и пункт *Insert Guest Additions CD image...* .



3. Кликните левой кнопкой мыши на значок внешних накопителей на панели задач и для устройства «VBOX CD-ROM (sr0)» выберите пункт *Подключить*, а затем «Открыть в менеджере файлов».



4. Запустите в терминале установку гостевых дополнений следующей командой:

```
sudo sh /media/cdrom/VBoxLinuxAdditions.run
```

5. Перезагрузите систему командой `sudo reboot`, чтобы изменения вступили в силу. В меню *Devices* окна виртуальной машины в пункте *Shared Clipboard* выберите пункт *Bidirectional* .

2.5 Практика и тестирование

2.5.1 Требования, правила и цели выполнения практической работы

Практическая работа в рамках этого курса проводится на виртуальной машине (ВМ) в гипервизоре VirtualBox. Допускается использование других систем виртуализации, но примеры будут приведены именно для VirtualBox ввиду его относительной легковесности и универсальности.

Требования к размещающей (хостовой) машине:

- Процессор с архитектурой x86_64
- Объем оперативной памяти – не менее 4 ГБ
- Объем свободного дискового пространства - не менее 50 ГБ
- Хостовая операционная система: Astra Linux 1.7.6 или Windows 10
- Гипервизор VirtualBox 7.0 или новее
- Доступ в интернет

Необходимые материалы:

- Релиз ОС Astra Linux SE 1.7.6

Правила и методика выполнения практической работы:

Для достижения наилучшего результата практическую работу рекомендуется выполнять без помощи раздела с ответами. Алгоритм должен быть следующий:

1. Прочитать и понять суть задания
2. Выполнить задание
3. В случае затруднений выполнить задание
    a. используя лекционный материал и примеры
    b. используя раздел с ответами (пошаговые инструкции)
4. Сравнить ваш способ решения с информацией из раздела с ответами

Цели практической работы:

- Закрепление знаний, полученных в ходе изучения лекционного материала, на практике.

Оценочное время на выполнение: 40-60 минут

2.5.2 Практическая работа: Модуль 2. Установка ОС Astra Linux

См. также *Требования, правила и цели выполнения практической работы*

2.5.2.1 Практические задания

В ходе выполнения практических заданий мы рекомендуем периодически делать холодные снимки виртуальной машины, чтобы вы могли быстро вернуть систему в исходное состояние. Использовать горячие снимки не рекомендуется, т.к. это увеличивает размер снимков на объем используемой оперативной памяти, да и некоторые компоненты системы (особенно в случае контроллера домена) могут быть чувствительны к резкому изменению времени.

Задание 1.

Установите операционную систему Astra Linux SE 1.7.6 на виртуальную машину. Образ для установки доступен по ссылке https://dl.astralinux.ru/astra/frozen/1.7_x86-64/1.7.6/iso/installation-1.7.6.11-26.08.24_17.26.iso.

Параметры ВМ:

- CPU: 2 (рекомендуется не менее двух ядер, т.к. в предыдущих версиях VirtualBox наблюдалось зависание виртуальных машин Linux, если назначено одно ядро).
- RAM: 2 ГБ.
- HDD: 30 ГБ.
- Сеть: NAT.
- EFI: включить.

Параметры при установке ОС:

- Имя пользователя и пароль, пароль для GRUB на усмотрение обучаемого.
- Имя компьютера: pc-1.
- Разметка дисков: Авто – использовать весь диск.
- Установить наборы ПО:
  - Графический интерфейс Fly.
  - Средства работы с Интернет.

    - Консольные утилиты.
- Уровень защищенности: Базовый «Орел».
- Остальные настройки можно оставить по умолчанию.

После установки системы выключите виртуальную машину и сделайте снимок.

Задание 2.

Выполните первоначальную настройку ОС Astra Linux SE 1.7.6:

- Убедитесь, что тестовая машина имеет доступ в интернет через NAT.
- Обновите ОС до актуальной версии.
- Установите гостевые дополнения (опционально, только если в качестве гипервизора используется VirtualBox).

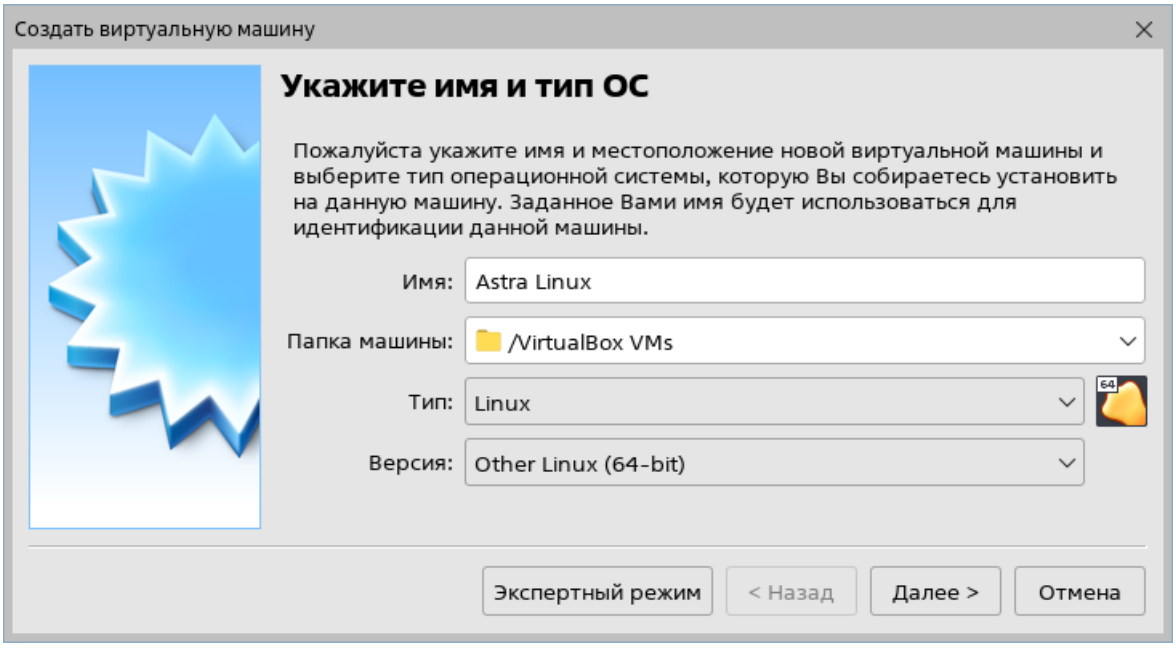
После завершения настроек выключите виртуальную машину и сделайте снимок.

2.5.2.2 Ответы на практические задания (пошаговые инструкции)

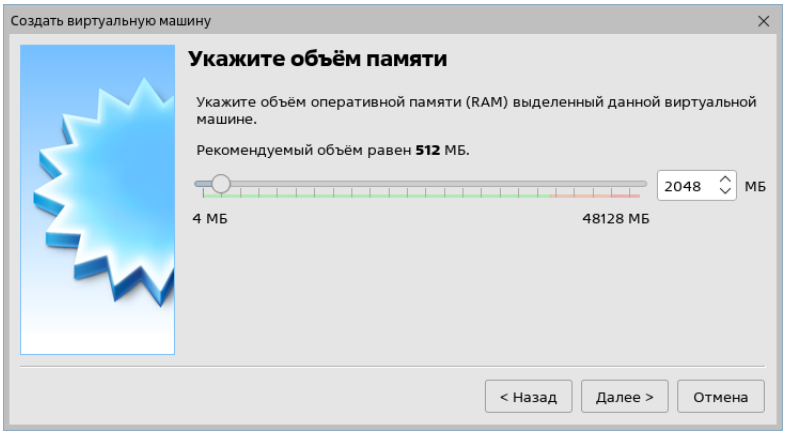
Задание 1.

Установите ОС Astra Linux SE 1.7.6 на виртуальную машину (ВМ) со следующими параметрами ВМ и ОС:

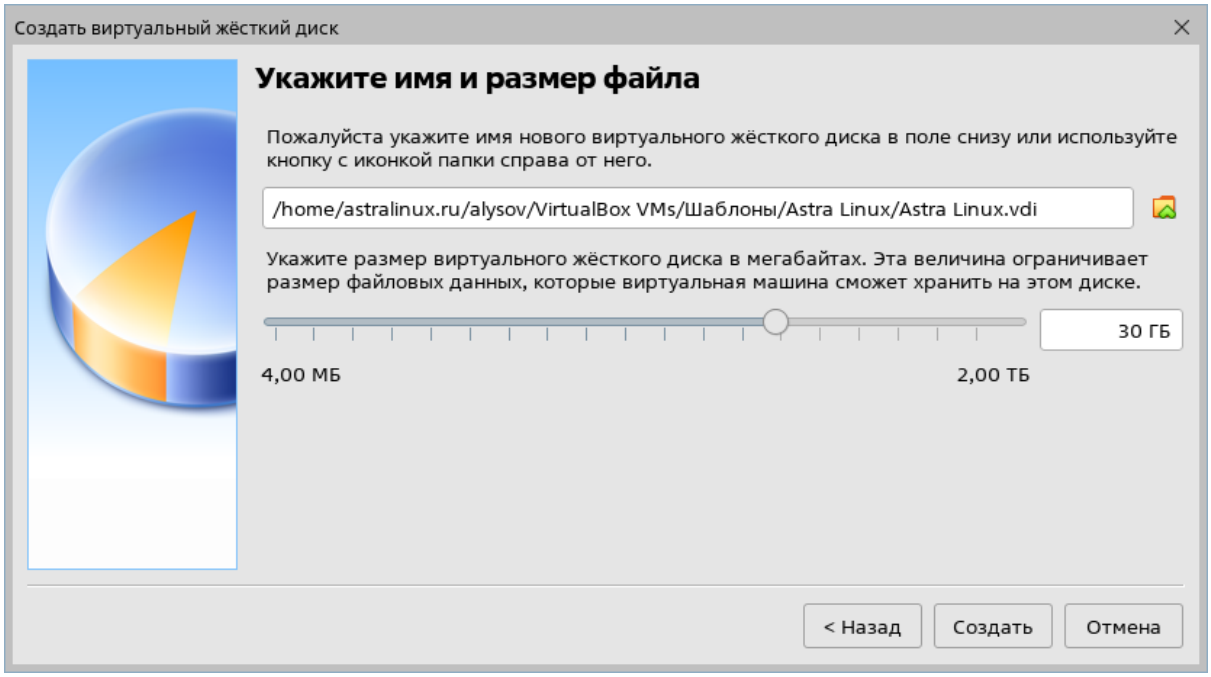
1. Скачайте образ ALSE по ссылке https://dl.astralinux.ru/astra/frozen/1.7_x86-64/1.7.6/iso/installation-1.7.6.11-26.08.24_17.26.iso.
2. Создайте новую ВМ в VirtualBox:
    a. Задайте имя ВМ, например, **Astra Linux**.
    b. Оставьте папку машины по умолчанию.
    c. Тип системы **Linux**
    d. Версия **Other Linux (64-bit)**



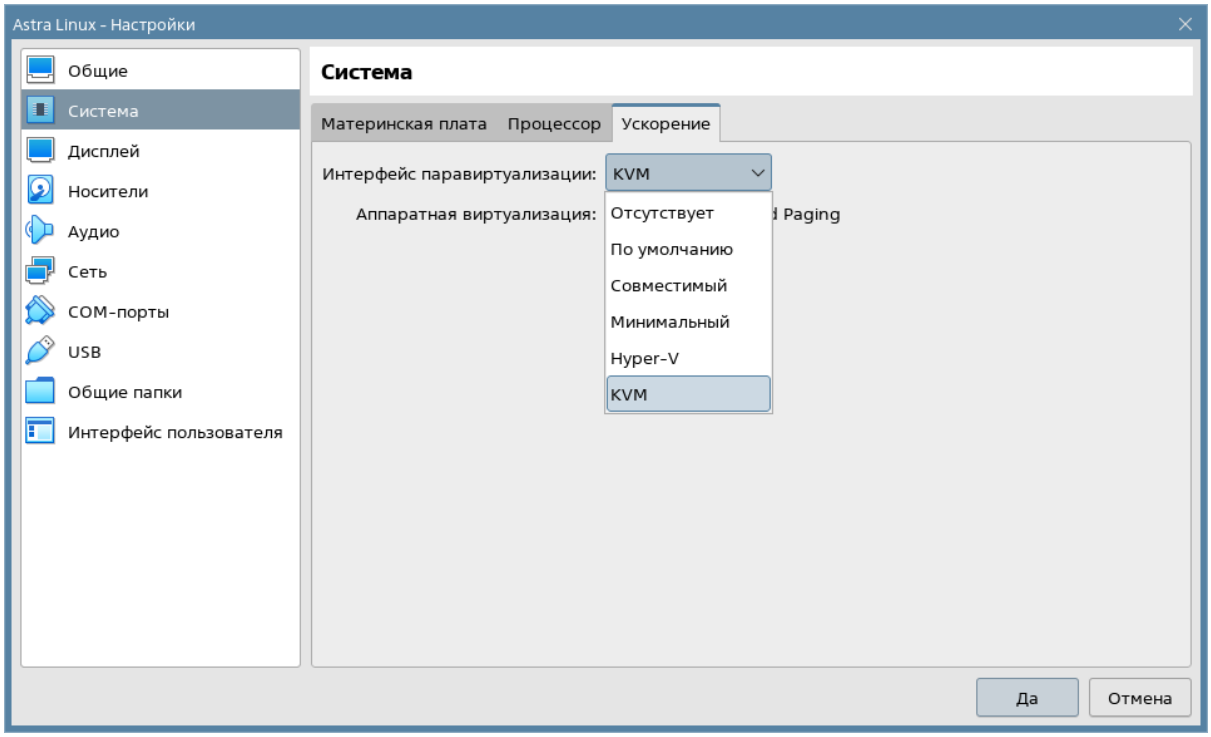
e. Память **2048 МБ**



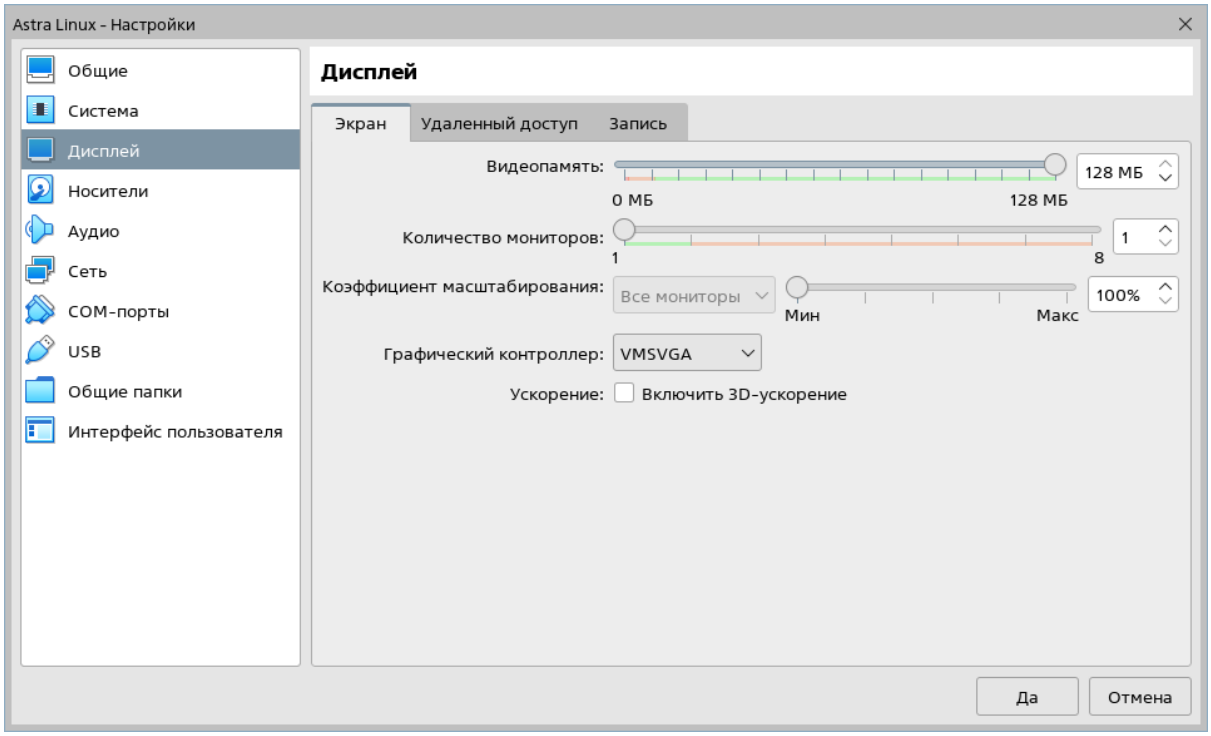
g. Жесткий диск **30 ГБ**



3. В разделе *Система* настроек виртуальной машины на вкладке *Ускорение* выберите **KVM**.



4. В разделе *Дисплей* для видеопамяти установите значение **128 МБ** (если оставить значение 16 Мб по умолчанию, то после входа в систему может появляться черный экран).



5. В разделе *Сеть* оставьте тип подключения **NAT** по умолчанию.

6. В разделе *Носители* для оптического привода выберите образ установочного диска операционной системы.

7. Включите виртуальную машину, чтобы начать установку системы:

   a. В качестве имени пользователя и пароля можете использовать значения **localadmin** и **Pa$$w0rd** соответственно.

   b. Имя компьютера: **pc-1**.

   c. Разметка дисков: **Авто – использовать весь диск**.

   d. Установить наборы ПО:

      - Графический интерфейс Fly.
      - Средства работы с Интернет.
      - Консольные утилиты.
      - Средства удаленного подключения SSH.

   e. Уровень защищенности: Базовый «Орел».

   f. Остальные настройки можно оставить по умолчанию.

8. Дождитесь завершения установки, перезагрузите систему и войдите в нее под созданной во время установки учетной записью администратора.

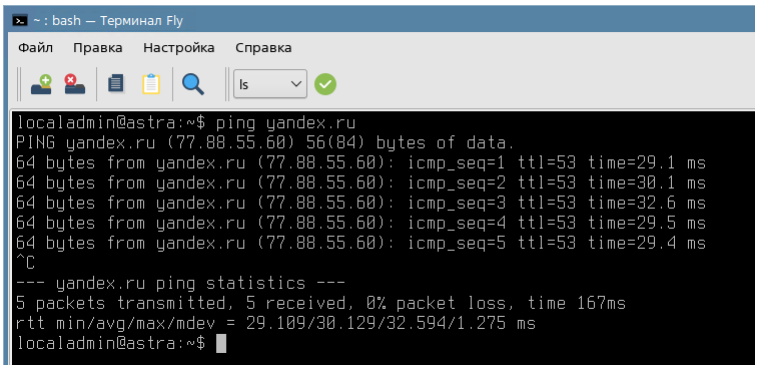
9. Выключите ВМ перед созданием снимка.

10. После установки системы выключите виртуальную машину и сделайте снимок.

Задание 2.

Выполните первоначальную настройку ОС Astra Linux SE 1.7.6.

1. Убедитесь, что тестовая машина имеет доступ в Интернет через NAT. Для проверки доступности ресурсов из сети Интернет откройте терминал сочетанием клавиш Alt + T и введите команду `ping yandex.ru`. Успешное выполнение пинга будет свидетельствовать о доступности сервера и работы механизма преобразования имен в IP-адреса. Чтобы прервать работу утилиты, нажмите сочетание клавиш Ctrl + C.



2. Обновите ОС до актуальной версии.

   a. Подключите репозитории Astra Linux:

   - В окне терминала выполните команду `sudo nano /etc/apt/sources.list` и подтвердите выполнение привилегированной команды от имени суперпользователя вводом пароля.

   - Установите следующее содержимое файла:

```
deb https://dl.astralinux.ru/astra/frozen/1.7_x86-64/1.7.6/repository-
↪base/ 1.7_x86-64 main contrib non-free
deb https://dl.astralinux.ru/astra/frozen/1.7_x86-64/1.7.6/repository-
↪extended/ 1.7_x86-64 main contrib non-free
```

   Для сохранения изменений в редакторе `nano` нажмите сочетание клавиш Ctrl + O и Enter. Для выхода из приложения нажмите Ctrl + X.

   b. Для обновления базы с информацией о доступных пакетах (индекса), выполните в тер-

минале команду `sudo apt update`. Чтобы обновить уже сами компоненты системы, выполните команду `sudo astra-update -a -r -T`. Ключ `-a` позволяет выполнить обновление в интерактивном режиме, поэтому вы сможете ознакомиться со списком планируемых изменений, чтобы принять окончательное решение о необходимости выполнения обновления. В случае возникновения ошибок выполните команду `sudo apt -f install` и заново запустите команду обновления.

c. После установки обновлений рекомендуется выполнить перезагрузку, которую можно запустить командой `sudo reboot`.

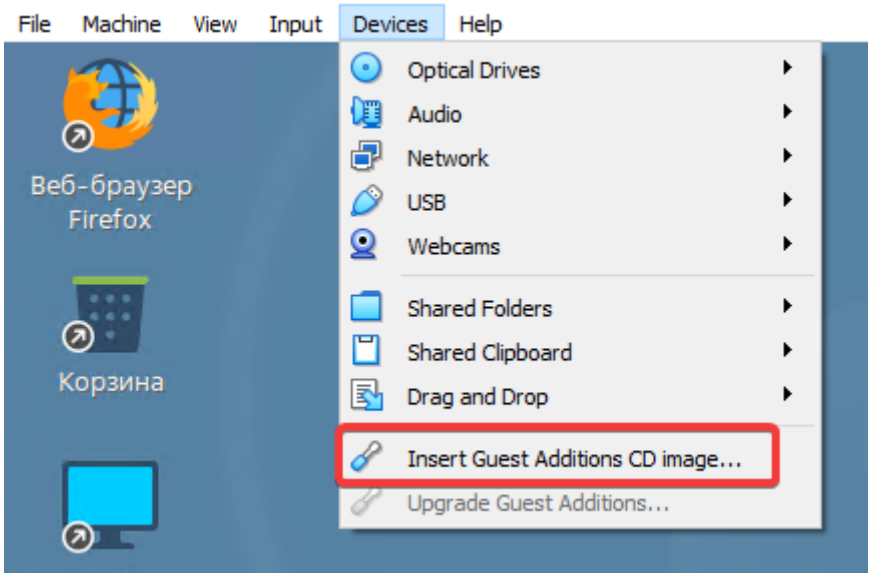
3. Установите гостевые дополнения:

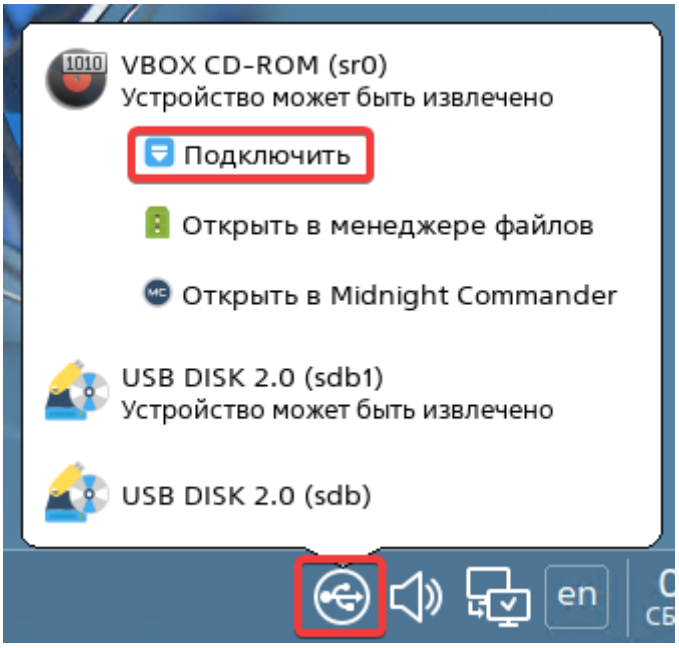
a. Выполните команду `sudo apt update` для обновления списка подключённых репозиториев и пакетов.

b. Выполните команду `sudo apt install gcc make linux-headers-6.` для установки пакетов программ, необходимых для работы гостевых дополнений, где 6.1 – версия установленного ядра ОС. Для проверки версии можно воспользоваться командой `uname -r`.

Ознакомьтесь с выводом команды и подтвердите установку пакетов. Дождитесь завершения процесса установки.

c. В окне виртуальной машины выберите меню *Devices* и пункт *Insert Guest Additions CD image...*. Далее кликните левой кнопкой мыши на значок внешних накопителей на панели задач и для устройства *VBOX CD-ROM (sr0)* выберите пункт *Подключить*.





d. Выполните команду `sudo sh /media/cdrom/VBoxLinuxAdditions.run` для установки гостевых дополнений. Ознакомьтесь с выводом команды и подтвердите установку пакетов. Дождитесь завершения процесса установки.

e. Выполните команду `sudo reboot` для перезагрузки компьютера.

4. В меню *Devices* окна виртуальной машины включите двунаправленный буфер *Shared Clipboard* → *Bidirectional*.

После завершения настроек выключите виртуальную машину и сделайте снимок.

2.5.3 Тест: Модуль 2. Установка ОС Astra Linux

Пожалуйста, ответьте на несколько вопросов, чтобы закрепить полученные знания. Правильные ответы для самопроверки вы сможете найти в конце списка. Обратите внимание, что у некоторых вопросов может быть несколько верных вариантов ответов.

2.5.3.1 Вопросы

Вопрос 1. Сколько места на диске рекомендуется иметь для установки ОС Astra Linux SE 1.7.6 с графической оболочкой Fly и программным обеспечением, входящим в состав дистрибутива?

a. 4 ГБ

b. 12 ГБ

c. 16 ГБ

d. 20 ГБ

Вопрос 2. В какую группу, дающую право выполнять команды от имени суперпользователя root, будет включена учетная запись администратора, созданная в процессе установки ОС?

a. Группа `sudo-astra`

b. Группа `astra-admin`

c. Группа `admin-astra`

d. Группа `astra-sudo`

Вопрос 3. Какой вариант разметки диска был рекомендован при демонстрации процесса установки операционной системы?

a. Авто – использовать весь диск

b. Авто – использовать весь диск и настроить LVM

c. Авто – использовать весь диск с защитным преобразованием на LVM

d. Вручную

Вопрос 4. Что такое GRUB?

a. Утилита для разметки дисков

b. Текстовый редактор

c. Утилита для установки пакетов

d. Загрузчик операционной системы

Ответы на вопросы

1. c
2. b
3. a
4. d

2.6 Заключение

Вы узнали о системных требованиях и порядке установки операционной системы Astra Linux SE 1.7.6. Установили ее и подготовили к работе первоначальной настройкой. Далее мы ознакомимся с графическим интерфейсом Fly

2.7 Дополнительные источники информации

- См. раздел *Дополнительная документация → Руководство по установке* на сайте wiki: Документации по ALSE 1.7

2.8 Обратная связь

Если остались вопросы, то их всегда можно задать в специальной теме.

Модуль 3. Работа с графическим интерфейсом Fly

3.1 Введение

Из этого модуля вы узнаете о том, как в Astra Linux работает графический режим, что представляет собой рабочий стол, как из командной строки запустить утилиты панели управления и зачем они все нужны.

3.2 Вход в систему

После включения компьютера и загрузки ОС вы увидите окно для входа в систему. Если в Windows за вход в систему отвечает приложение winlogon.exe, то в Astra Linux его аналогом является fly-dm_greet из состава пакета fly-qdm.

Окружение рабочего стола Fly реализовано на базе системы окон X Window System (X11, X или просто «иксы», см. рис. 1.5), терминология которой противоречит общепринятой концепции клиент-серверных приложений:

- **«X Сервер»** (или как его еще называют дисплейный сервер) – это первое приложение, которое запускается при старте «графики». Его основная задача заключается во взаимодействии с видеокартой, монитором и различными устройствами ввода. Сервер предоставляет приложениям холст определенных размеров и передает им информацию о движениях мыши, нажатии

клавиш на клавиатуре и других событиях, поступающих от устройств ввода. В Astra Linux функции сервера выполняет приложение fly-dm (Fly Display Manager).

- **«X Клиент»** – любая программа, которая использует дисплейный сервер для отображения своего интерфейса на экране. Программа может запускаться как локально (например, тот же fly-dm_greet), так и удаленно, например, когда требуется администрировать удаленный сервер из графики.

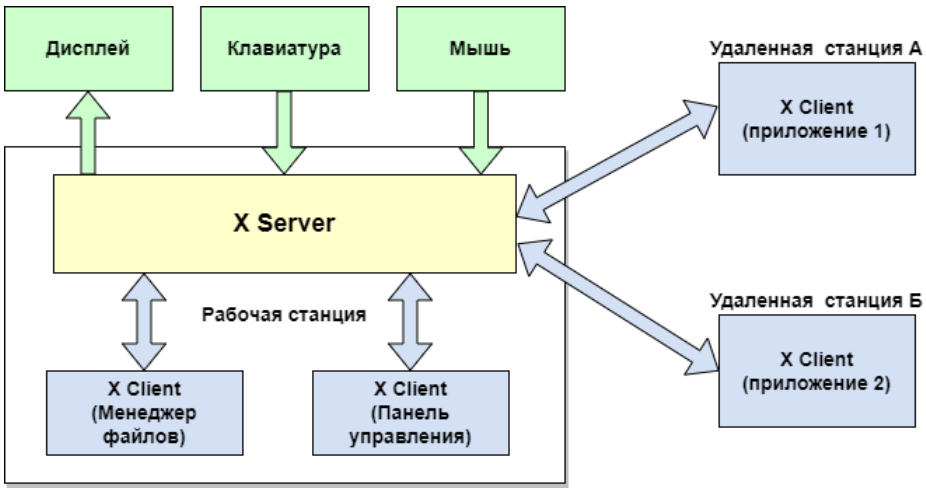


рис. 1.5 – Архитектура X Window System

Внешний вид графических элементов и эффекты определяются не X Сервером, а менеджером окон, функции которого в Astra Linux выполняет fly-wm (Window Manager for Fly Desktop). Это приложение реализует рабочий стол, классическое меню «Пуск», панель задач, переключатель рабочих столов, блокировщик экрана.

Операционная система Astra Linux позволяет изменить внешний вид окна входа, сделать это можно с помощью утилиты `fly-admin-dm` («Вход в систему»), запускать которую нужно от имени администратора. Описание утилиты приведено в электронной справке.

Примечание: Вызов электронной справки осуществляется с помощью ярлыка «Помощь», размещенного на рабочем столе, а также путем нажатия горячих клавиш Alt + F1 или F1 в активном окне графической программы.

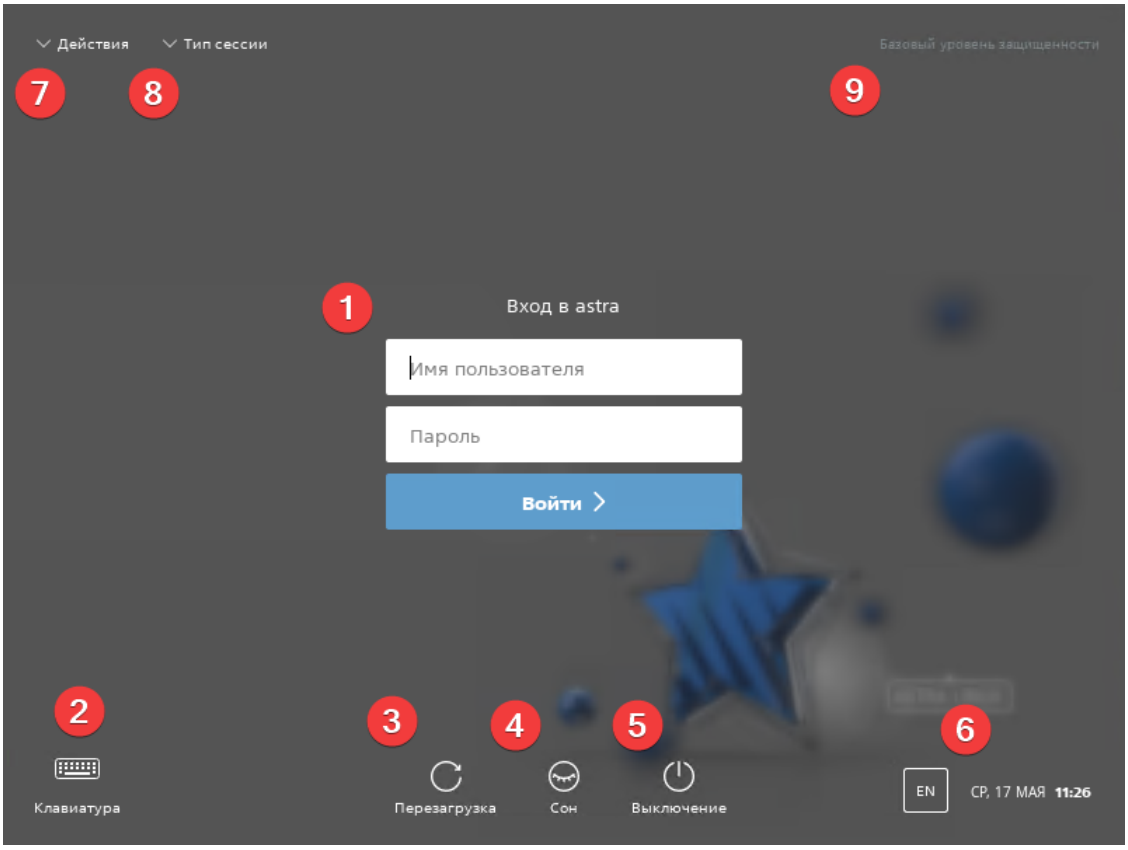
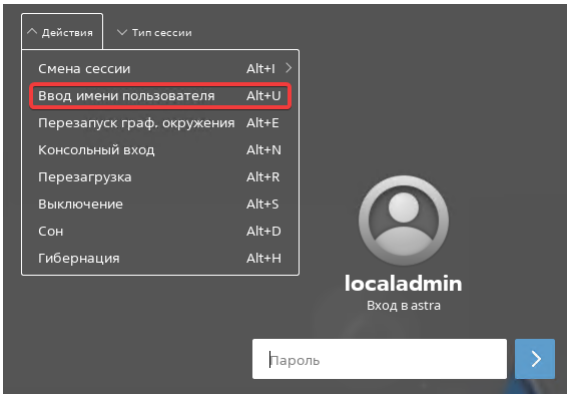


рис. 1.6 – Окно входа в Astra Linux

Перечислим элементы интерфейса окна входа в систему и их назначение, см. рис. 1.6:

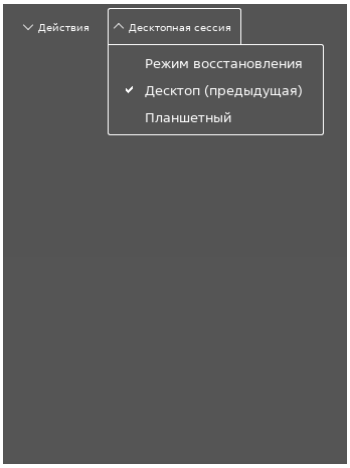
1. Поля для ввода имени пользователя и пароля с кнопкой *Войти*.
2. Кнопка вызова экранной клавиатуры.
3. Кнопка перезагрузки – перезагружает компьютер без запрашивания дополнительных подтверждений.
4. Кнопка перехода в режим сна – после получения подтверждения приостанавливает работу системы. Данные хранятся в оперативной памяти, т.е. компьютеру в этом режиме необходимо электропитание.
5. Кнопка выключения – выключает компьютер без запрашивания дополнительных подтверждений.
6. Кнопка для смены языковой раскладки и область отображения системного времени.
7. Меню действий содержит следующие команды:
   - **Смена сессии** – если запущено несколько сессий, то с помощью этого действия можно переключиться на одну из них.

- **Выбор пользователя** – изменяет интерфейс ввода учетных данных. Вместо поля ввода имени пользователя нам становится доступен список пользователей, под которыми может быть осуществлен вход в систему (этот список настраивается утилитой `fly-admin-dm` – «вход в систему»). После выполнения указанного действия имя этого пункта меню меняется на «Ввод имени пользователя». Выбрав это действие, вы переключите интерфейс обратно в режим ввода имени пользователя и пароля.



- **Перезапуск граф. окружения** – перезапуск графической среды Astra Linux без полной перезагрузки системы (аналог перезагрузки Windows с зажатой клавишей Shift).
- **Перезагрузка** – перезагружает компьютер без запроса дополнительных подтверждений.
- **Выключение** – выключает компьютер без запроса дополнительных подтверждений.
- **Сон** – приостановка работы системы после получения подтверждения. Данные хранятся в оперативной памяти, т.е. компьютеру в этом режиме необходимо электропитание.
- **Гибернация** – приостановка работы системы после получения подтверждения. Все данные из оперативной памяти сохраняются на диске, поэтому компьютеру в этом состоянии не требуется электропитание.

8. Меню «тип сессии». По умолчанию система запускает тот тип сессии, который был использован последним. На выбор доступны:
   - **Режим восстановления** – для восстановления системы через терминал.
   - **Планшетный** – адаптирован для устройств, оснащенных сенсорным экраном.
   - **Десктоп** – для компьютеров, оснащенных клавиатурой и мышью.



9. Область отображения текущего уровня защищенности ОС.

Выполним вход с систему под учетной записью, которую мы создали во время установки ОС. Введем имя пользователя и пароль и нажмем кнопку *Войти*. В случае успешного прохождения аутентификации будет выполнен вход в сессию пользователя.

3.3 Рабочий стол

Если в Windows рабочий стол – это приложение explorer.exe, то в Astra Linux функции рабочего стола Fly реализованы в приложении fly-wm (Windows Manager, менеджер окон), а графические утилиты управления представляют собой оболочки соответствующих утилит командной строки.



рис. 1.7 – Рабочий стол Astra Linux

Интерфейс рабочего стола ОС Astra Linux во многом похож на Windows (рис. 1.7): тут есть и меню пуск, и привычный формат панели задач, и привычные контекстные меню – все это значительно упрощает обучение пользователей при решении задач импортозамещения.

Для начала настроим комфортное разрешение экрана рабочего стола. Для этого:

1. Кликните правой кнопкой мыши на пустом месте рабочего стола и выберите в контекстном меню пункт *Настройка монитора...*. Откроется окно «Настройка монитора» (утилита `fly-admin-screen`), см. рис. 1.8.

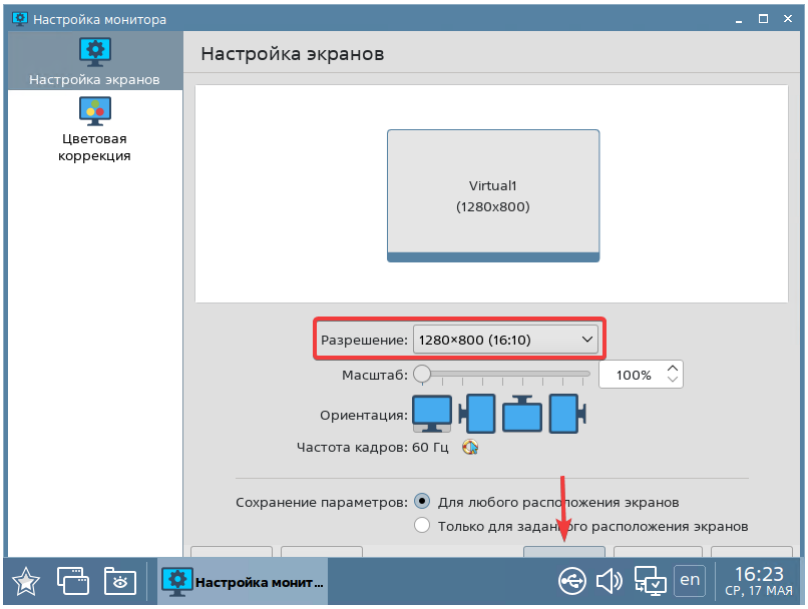


рис. 1.8 – Настройка монитора

2. Выберите желаемое разрешение экрана.

3. Примените настройки, нажав кнопку *Да*, отмеченную на рисунке стрелкой (текст кнопки может быть скрыт, если окно программы не поместилось на экран), см. рис. 1.8.

Теперь разберем назначение элементов панели задач, см. рис. 1.9:



рис. 1.9 – Панель задач

1. Кнопка меню *Пуск*.

2. Панель быстрого запуска. По умолчанию на ней доступны переключатель окон и менеджер файлов.

3. Кнопка-переключатель рабочих столов.

4. Панель переключения задач.

5. Область уведомлений. По умолчанию на ней доступны «Аудиомикшер KMix», «Апплет Network Manager», «Раскладка клавиатуры» и «Дата и время». При подключении внешнего носителя информации появится соответствующая иконка.

6. Кнопка, позволяющая свернуть все окна, находится в правой части панели задач (отмечена стрелкой).

Меню *Пуск* напоминает классическое меню операционной системы Windows, см. рис. 1.10.

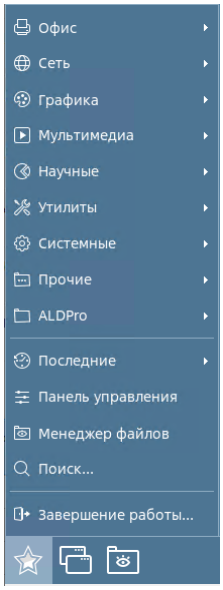


рис. 1.10 – Меню «Пуск»

Здесь нам доступны тематические разделы прикладных программ и утилит, раздел с последними запущенными программами, есть возможность вызова панели управления, запуска менеджера файлов, поиска и меню завершения работы.

Электронную справку по рабочему столу можно вызвать клавишами Alt + F1, далее выберите раздел *Документация → Графический интерфейс → Рабочий стол*.

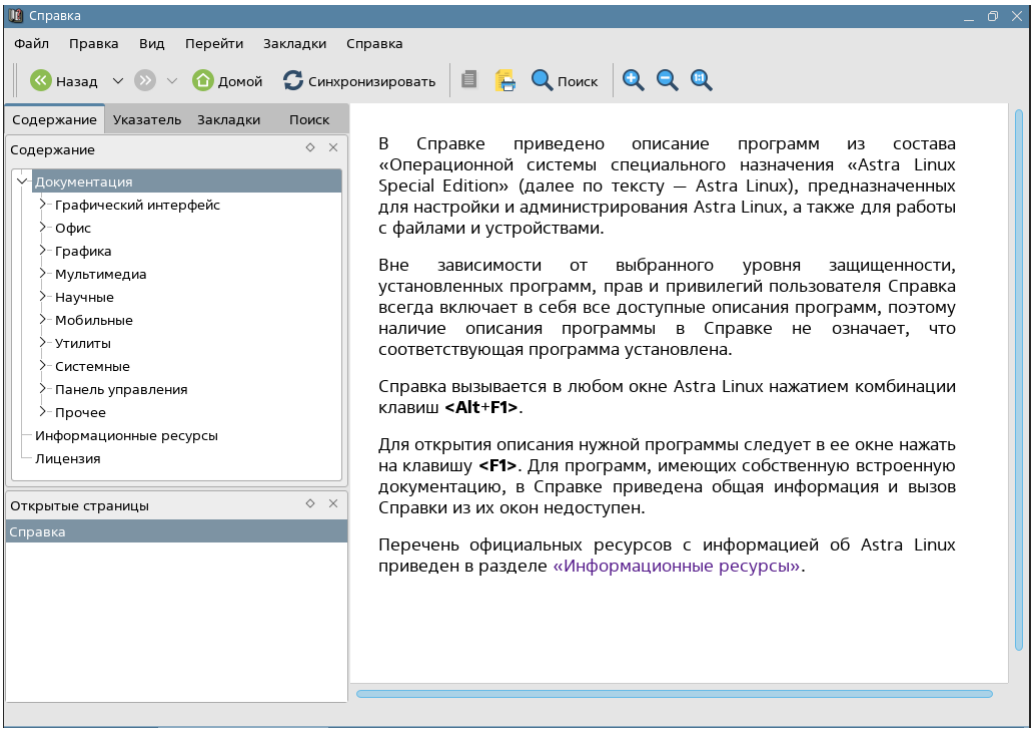


рис. 1.11 – Электронная справка Astra Linux

3.4 Обзор основных графических утилит

3.4.1 Менеджер файлов

Программа «Менеджер файлов» (fly-fm) предназначена для управления объектами файловой системы. Она позволяет подключать (монтировать) и отключать (размонтировать) файловые системы устройств хранения данных, таких как локальные жесткие диски и их разделы, CD/DVD-диски, USB-накопители. Она также позволяет обращаться к сетевым SMB-ресурсам, работать с архивами и управлять шифрованием дисков (защитным преобразованием).

Файловый менеджер Fly позволяет работать не только с базовыми правами доступа «Пользователь-Группа-Остальные», но и с расширенными списками доступа POSIX ACL, что является его существенным преимуществом. Окно менеджера файлов похоже на проводник Windows, но обратите внимание на то, что сохранение параметров внешнего вида выполняется в момент закрытия приложения, что сначала может сбить с толку. Для настройки желаемого внешнего вида вам нужно выполнить следующие действия:

1. откройте только одно окно приложения «Менеджер файлов»;
2. внесите желаемые изменения в настройки внешнего вида, например, включите отображение скрытых файлов (меню *Вид → Отображать скрытые*);
3. закройте окно приложения, чтобы сохранить внесенные изменения.

После выполнения указанных действий вы можете продолжить работу с менеджером файлов, и во всех последующих окнах отображение скрытых файлов будет включено по умолчанию.

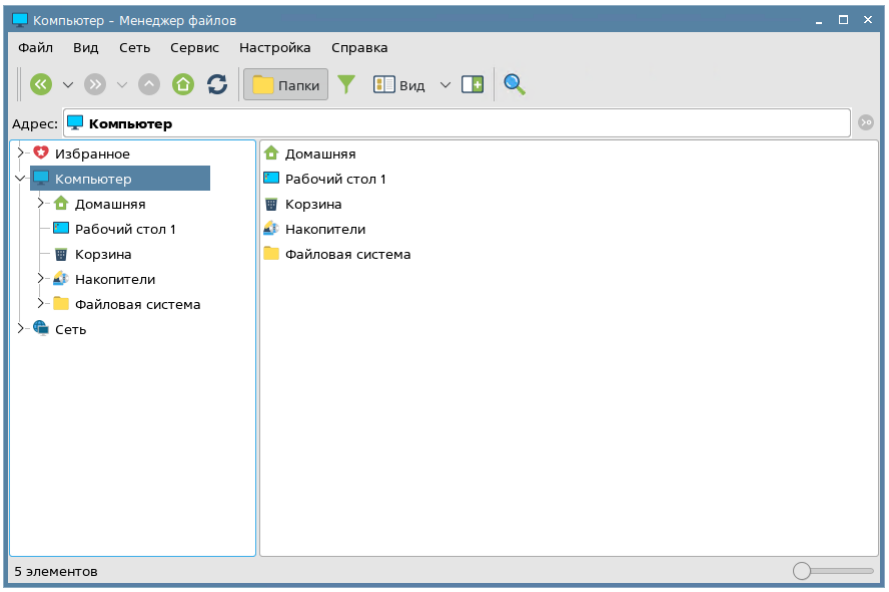


рис. 1.12 – Окно программы «Менеджер файлов»

Важно: Если название файла или каталога начинается с символа точки «.», то в Linux он будет считаться скрытым. Например, в рабочей папке пользователя есть такие файлы, как .bashrc, .profile и др. Чтобы увидеть скрытые файлы из командной строки, следует использовать команду ls с ключом «-a»: `ls -a`.

3.4.2 Рабочий стол и меню «Пуск»

Каждый пользователь в системе имеет возможность выполнить индивидуальные настройки отображения меню «Пуск», внешнего вида, расположения элементов, особенностей работы с клавиатурой и мышью.

Меню «Пуск» поддерживает два режима отображения: классическое меню и меню панель. По умолчанию после установки ОС настроено отображение классического меню. Полное описание режимов отображения меню «Пуск» и порядок переключения между ними приведены в электронной справке.

Часть настроек рабочего стола задается администратором и недоступна для изменений обычному пользователю. Некоторые из возможностей настройки могут быть реализованы при использовании утилит настройки из меню *Пуск → Панель управления*.

Категория «Рабочий стол» программы «Панель управления» объединяет графические утилиты, которые могут быть применены для индивидуальной настройки рабочего стола.

Полное описание рабочего стола Fly, его режимов работы и предоставляемых возможностей, а также описание графических программ приведено в электронной справке.

3.4.3 Системный монитор

С помощью горячего сочетания клавиш Ctrl + Alt + Del вы можете открыть приложение Системный монитор (`ksysguard`), которое предоставляет оперативную информацию о процессах и общей загрузке системы: ЦПУ, памяти, раздела подкачки и сети.

Интерфейс приложения во многом похож на Диспетчер задач Windows. По умолчанию открываются две вкладки с предустановленными названиями и интервалами обновления: «Таблица процессов» (рис. 1.13) и «Общая загрузка системы» (рис. 1.14), но вы можете создать и собственные вкладки с графиками, используя любые доступные данные о процессах и загрузке системы, см. команду *Файл → Создать вкладку*.

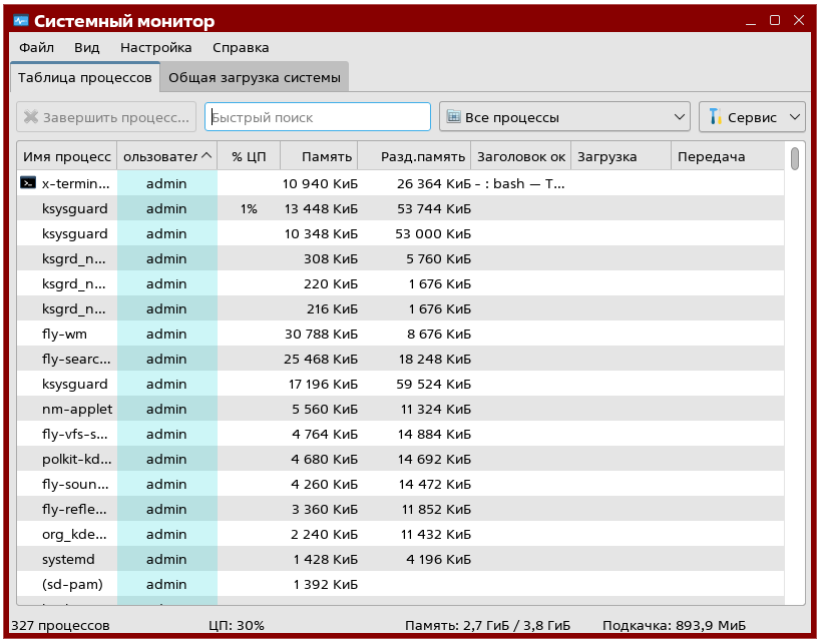


рис. 1.13 – Вкладка «Таблица процессов»

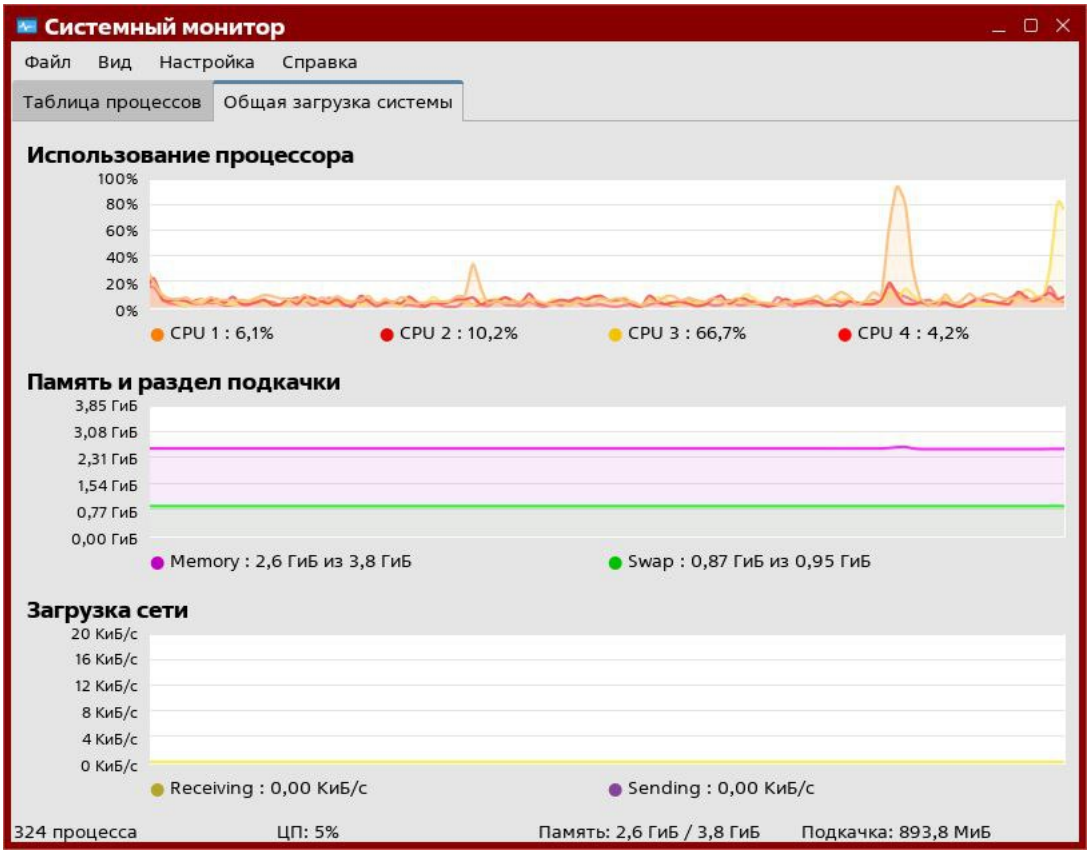


рис. 1.14 – Вкладка «Общая загрузка системы»

3.4.4 Панель управления

Панель управления (`fly-admin-center`) доступна через меню *Пуск → Панель управления* и представляет собой графический инструмент для навигации по утилитам управления пользователями, устройствами и другими параметрами системы, включая аспекты безопасности, см. рис. 1.15.

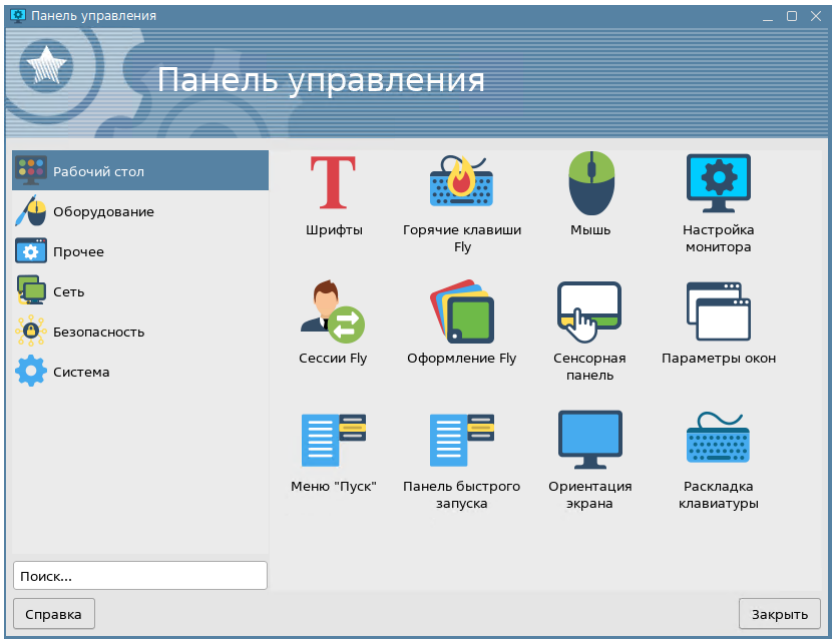


рис. 1.15 – Окно «Панели управления»

Ярлыки утилит панели управления представляют собой файлы `*.desktop`, которые располагаются в папке `/usr/share/applications/`

- Рабочий стол
  - Шрифты (`fly-admin-fonts`)
  - Горячие клавиши (`fly-admin-hotkeys`)
  - Мышь (`fly-admin-mouse`)
  - Настройка монитора (`fly-admin-screen`)
  - Сессии Fly (`fly-admin-session`)
  - Оформление Fly (`fly-admin-theme`)
  - Сенсорная панель (`fly-admin-touchpad`)
  - Параметры окон (`fly-admin-winprops`)
  - Меню «Пуск» (`fly-menuedit ~/.fly/startmenu`)

    - Панель быстрого запуска (`fly-menuedit ~/.fly/toolbar`)
    - Ориентация экрана (`fly-orientation`)
    - Раскладка клавиатуры (`fly-xkbmap`)
- Оборудование
    - Менеджер устройств (`fly-admin-device-manager`)
    - Форматирование внешнего носителя (`fly-admin-format`)
    - Электропитание (`fly-admin-power`)
    - Принтеры (`fly-admin-printer`)
    - Обработка «горячего» подключения (`fly-admin-reflex`)
    - Сервис удаленных USB-накопителей (`fly-admin-usbip`)
    - Настройка яркости Fly (`fly-brightness`)
- Прочее
    - Редактор репозиториев (`fly-admin-repo`)
    - TeXdoctk (`texdoctk`)
- Сеть
    - Параметры сети (`fly-admin-network`)
    - Настройка межсетевого экрана (`gufw`)
    - Сетевые соединения (`nm-connection-editor`)
- Безопасность
    - Настройка регистрации системных событий (`fly-admin-events`)
    - Проверка целостности системы (`fly-admin-int-check`)
    - Системный киоск (`fly-admin-kiosk`)
    - Санкции PolicyKit-1 (`fly-admin-policykit-1`)
    - Политика безопасности (`fly-admin-smc`)
    - Журнал системных событий (`fly-event-viewer`)

- Журнал аудита (ksystemlog)
- Конфигурация аудита (system-config-audit)
- Изменение пароля (fly-passwd) - очень полезная графическая утилита для смены пароля, которая работает как для локальных пользователей, так и для пользователей доменов FreeIPA / Samba AD. Если устанавливать систему с настройками по умолчанию, то может потребоваться установить дополнительно пакет fly-passwd.

- Система
  - Управление драйверами (fly-kcmshell5 --desktopfile driver driver)
  - Системные альтернативы (fly-admin-alternatives)
  - Автозапуск (fly-admin-autostart)
  - Планировщик задач (fly-admin-cron)
  - Дата и время (fly-admin-date)
  - Вход в систему (fly-admin-dm)
  - Переменные окружения (fly-admin-env)
  - Загрузчик GRUB2 (fly-admin-grub2)
  - Мультитерминальный режим (fly-admin-multiseat)
  - Синхронизация времени (fly-admin-time)
  - Установка обновлений (fly-astra-update)
  - Приложения для типов файлов (fly-mimeapps)
  - Центр системных отчетов (fly-sosreport)
  - Инициализация системы (systemdgenie)
  - Менеджер пакетов Synaptic (synaptic-pkexec)

3.4.5 Прочие утилиты

Ниже представлен полный список пакетов, начинающихся с fly-*. Стоит учесть, что не все из перечисленных утилит устанавливаются по умолчанию при установке ОС.

Описание утилит доступно в электронной справке. Вызов электронной справки осуществляется

путем нажатия клавиши F1 в активном окне графической утилиты.

fly-admin-autostart – **«Автозапуск»**

Управление автозапуском программ, автоматическим открытием файлов и каталогов при входе в сессию

fly-admin-dm – **«Вход в систему»**

Настройка графического входа в сессию

fly-admin-date – **«Дата и время»**

Просмотр времени и календаря, настройка формата отображения даты и времени

fly-admin-time – **«Синхронизация времени»**

Настройка синхронизации времени

fly-admin-grub2 – **«Загрузчик GRUB2»**

Графическая утилита настройки загрузчика ОС GRUB2

systemdgenie – **«Инициализация системы»**

Управление службой инициализации системы Systemd

synaptic – **«Менеджер пакетов Synaptic»**

Графическая утилита управления пакетами Synaptic

gufw – **«Настройка межсетевого экрана»**

Настройка межсетевого экрана UFW (Uncomplicated Firewall)

fly-admin-reflex – **«Обработка «горячего» подключения»**

Настройка действий, выполняемых при подключении устройств

fly-admin-env – **«Переменные окружения»**

Добавление, изменение и удаление переменных окружения

fly-admin-cron – **«Планировщик задач»**

Установка расписания задач для выполнения в фоновом режиме, настройка среды выполнения задачи (переменных окружения), разрешение или запрет на выполнение уже установленной задачи

fly-admin-smc – **«Политика безопасности»**

Управление локальными политиками безопасности. Для импорта и экспорта локальных неперсонализированных параметров политики безопасности (не связанных с параметрами и настройками пользователей или групп) используется инструмент командной строки astra-adminctl (подробнее см. в справке astra-adminctl -h)

fly-mimeapps – **«Приложения для типов файлов»**

Просмотр доступных приложений и установка приложения по умолчанию для типов файлов

`fly-admin-printer` – **«Принтеры»**

Управление принтерами, настройка печати и управление заданиями на печать

`fly-admin-policykit-1` – **«Санкции PolicyKit-1»**

Просмотр, предоставление и аннулирование санкций на выполнение привилегированных действий, управляемых с использованием PolicyKit-1

PolicyKit - это средство определения и приведения в исполнение правил обращения непривилегированных процессов к привилегированным, которое противопоставляется использованию таких утилит, как `sudo`, но не наделяет процесс пользователя правами администратора, а позволяет точно контролировать, что разрешено, а что запрещено.

`fly-admin-session` – **«Сессии Fly»**

Настройки параметров входа и выхода из сессий пользователя

`nm-connection-editor` – **«Сетевые соединения»**

Настройка сетевых соединений

`fly-admin-network` – **«Параметры сети»**

Управление автозапуском сетевых служб

`fly-admin-alternatives` – **«Системные альтернативы»**

Управление системой альтернатив, которая позволяет нескольким программам, выполняющим сходные функции, зарегистрироваться как альтернативные приложения, одно из которых будет использоваться по умолчанию. Например, при запуске утилиты `visudo` для безопасного редактирования файла sudoers используется программа, указанная в альтернативе editor. По умолчанию это nano, но вы можете установить vim.basic, если вам будет удобнее этот редактор.

`fly-admin-kiosk` – **«Системный киоск»**

Управление ограничением среды пользователя. С помощью киоска Fly можно настроить операционную систему таким образом, чтобы при загрузке запускалось какое-то одно приложение без рабочего стола (например, Firefox), что позволяет создавать терминалы самообслуживания.

`hp-setup` – **«Установка принтеров, факсов и сканеров HP»**

Установка новых устройств Hewlett-Packard

`hp-plugin` **«Установка дополнительного плагина HP»**

Установка драйверов Hewlett-Packard

`fly-admin-power` – **«Электропитание»**

Настройка и управление параметрами электропитания и энергосбережения

`fly-admin-service` – **«Сервисы»**

Статус и конфигурация служб, их запуск и остановка, автоматизированная настройка служб для функционирования с аутентификацией в режиме ЕПП и PAM

`ksystemlog` – **«Просмотр системных журналов Ksystemlog»**

Просмотр журнала расширенной системы протоколирования

`system-config-audit` – **«Конфигурация аудита»**

Настройки аудита системы

`fly-sosreport` – **«Центр системных отчетов»**

Сбор данных о конфигурации системы и работе подсистем для последующей диагностики

`fly-run` – **«Запуск приложения»**

Запуск программы или осуществление доступа к ресурсу из командной строки, в том числе от имени другого пользователя и/или с другими мандатными атрибутами

`sudo afick-tk` – **«Контроль целостности файлов»**

Графическая утилита программы afick мониторинга изменений целостности объектов файловой системы

`fly-admin-device-manager` – **«Менеджер устройств»**

Просмотр доступных устройств, настройка их драйверов и параметров

`fly-admin-usbip` – **«Сервис удаленных USB-накопителей»**

Предоставление удаленного доступа к USB-носителям и токенам с помощью USB-over-IP

`fly-admin-format` – **«Форматирование внешнего носителя»**

Удаление данных и форматирование внешнего носителя и его разделов

`fly-admin-iso` – **«Запись ISO образа на USB носитель»**

Программа записи iso-образа на USB-носитель

`fly-admin-int-check` – **«Проверка целостности системы»**

Проверка целостности системы для рабочего стола Fly

`fly-admin-marker` – **«Редактор маркеров»**

Просмотр и изменение настроек маркировки печати, редактирование шаблонов маркера

`fly-print-station` – **«Управление печатью документов»**

Маркировка и печать документов ограниченного доступа с возможностью перемещения заданий на другой принтер

`sudo gparted` **«Редактор разделов Gparted»**

Создание, перераспределение или удаление системных разделов ОС

`ksysguard` – **«Системный монитор»**

Просмотр информации о процессах и общей загрузке системы: ЦПУ, памяти, раздела подкачки и сети

`fly-term` – **«Терминал Fly»**

Эмулятор терминала, позволяющий взаимодействовать с консолью

`mc` – **«Менеджер файлов Midnight commander»**

Просмотр папок и элементов ФС, выполнение основных функций управления файлами, обращение к сетевым ресурсам, работа с архивами

`kgpg` **«Управление ключами KGpg»**

Управление ключами GPG (GNU Privacy Guard). GnuPG — свободная программа для шифрования информации и создания электронных цифровых подписей

`fly-admin-ald-server` **«Доменная политика безопасности»**

Настройка службы контроллера домена ALD (в настоящий момент компания развивает продукт ALD Pro на базе службы каталога FreeIPA, а приложение ALD не рекомендуется к развертыванию)

`fly-admin-ald-client` – **«Настройка ALD клиента Fly»**

Ввод клиентского компьютера в домен ALD (в настоящий момент компания развивает продукт ALD Pro на базе службы каталога FreeIPA, а приложение ALD не рекомендуется к развертыванию)

`fly-admin-ad-client` – **«Настройка клиента Active Directory Fly»**

Ввод клиентского компьютера в домен AD Windows

`fly-admin-ad-server` **«Настройка сервера Active Directory Fly»**

Запуск службы контроллера домена AD

`fly-admin-ad-sssd-client` – **«Настройка клиента SSSD Fly»**

Ввод клиентского компьютера в существующий домен AD Windows, при этом будет задействована служба управления аутентификацией и авторизацией SSSD (System Security Services Daemon)

`fly-admin-freeipa-server` – **«Настройка FreeIPA server Fly»**

Установка и настройка сервера FreeIPA

`fly-admin-freeipa-client` – **«Настройка FreeIPA client Fly»**

Ввод клиентского компьютера в существующий домен FreeIPA

`fly-admin-multiseat` – **«Мультитерминальный режим»**

Подготовка компьютера для одновременной работы нескольких пользователей

`fly-admin-dhcp` – **«Настройка DHCP-сервера»**

Настройка сервера DHCP

`fly-admin-openvpn-server` – **«Настройка OpenVPN сервера Fly»**

Настройка сервера VPN

`fly-admin-ftp` – **«FTP-сервер»**

Настройка сервера FTP

`fly-admin-ntp` – **«Синхронизация времени (NTP)»**

Настройка сервера времени NTP

`fly-admin-samba` – **«Общие папки (Samba)»**

Управление общими папками Samba

`fly-passwd` – **«Изменить пароль»**

Позволяет изменить пароль, причем это может быть как локальный пользователь, так и пользователь домена (ALD Pro, ALD, FreeIPA, Samba AD)

`fly-su` – **«Подмена пользователя»**

Выполнение команды от имени другого пользователя, представляет собой графическую обертку над утилитой `su`

`fly-astra-update` – **«Установка обновлений»**

Программа установки обновлений

`fly-admin-repo` – **«Редактор репозиториев»**

Создание и управление репозиториями

`fly-admin-driver` – **«Управление драйверами»**

Установка и выбор графических драйверов

3.5 Практика и тестирование

3.5.1 Практическая работа: Модуль 3. Работа с графическим интерфейсом Fly

См. также *Требования, правила и цели выполнения практической работы*

3.5.1.1 Практические задания

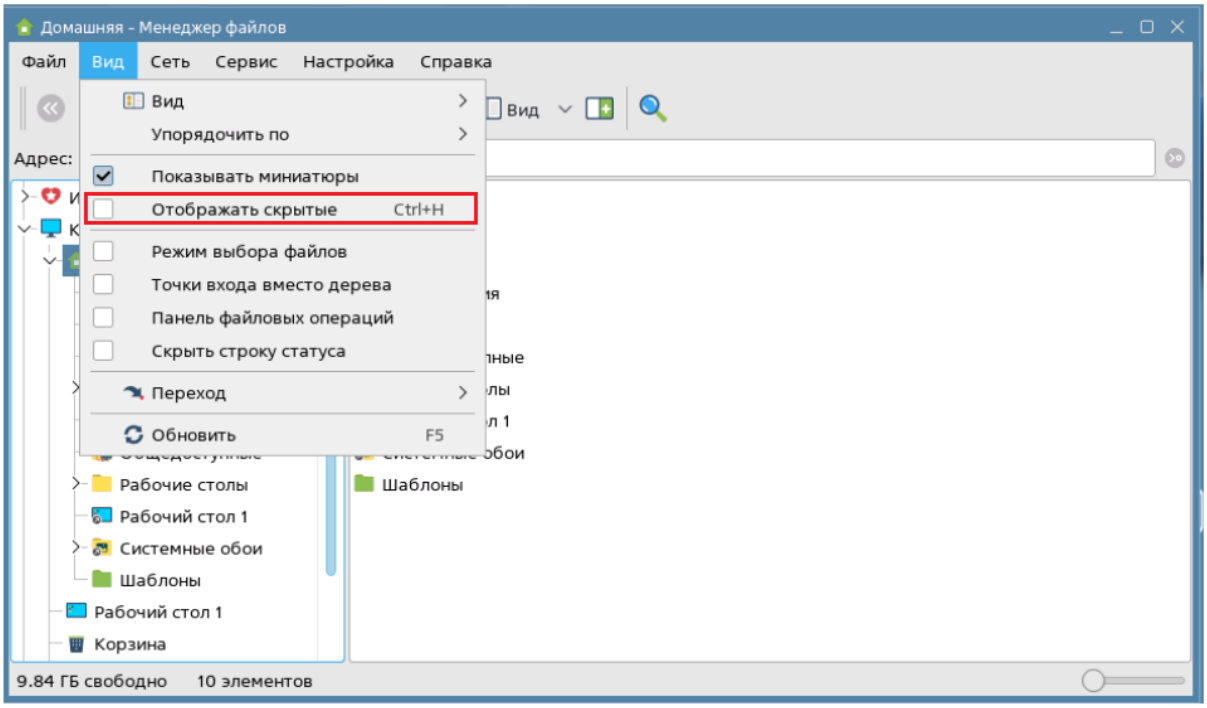
1. Включите отображение скрытых объектов в менеджере файлов.
2. Измените пароль администратора системы через панель управления.
3. Используя справку, найдите информацию о пакете ksysguard, запустите это приложение.
4. Самостоятельно изучите содержимое панели задач.

3.5.1.2 Ответы на практические задания (пошаговые инструкции)

Задание 1.

Чтобы включить отображение скрытых файлов:

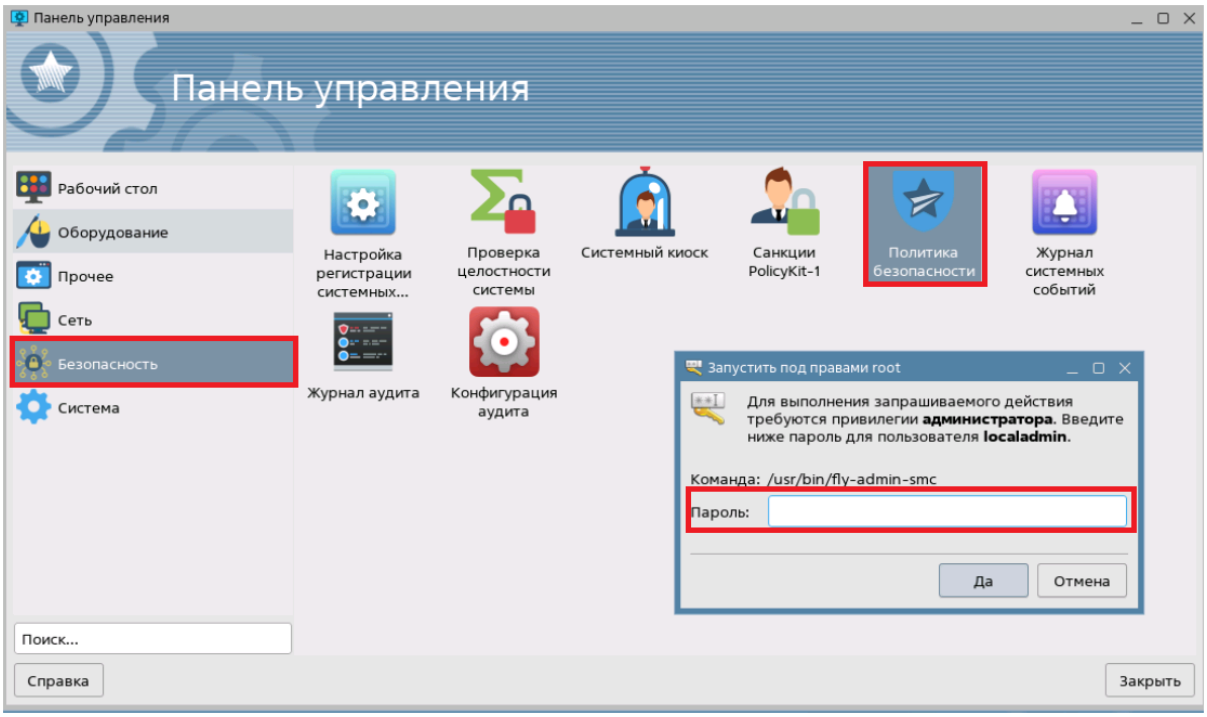
1. Нажмите кнопку *Пуск*;
2. Запустите приложение «Менеджер файлов»;
3. В окне «Менеджер файлов» в горизонтальном меню *Вид* включите флажок *Отображать скрытые*;
4. Для сохранения изменений закройте приложение «Менеджер файлов», иначе в новом окне не будут отображаться скрытые объекты.



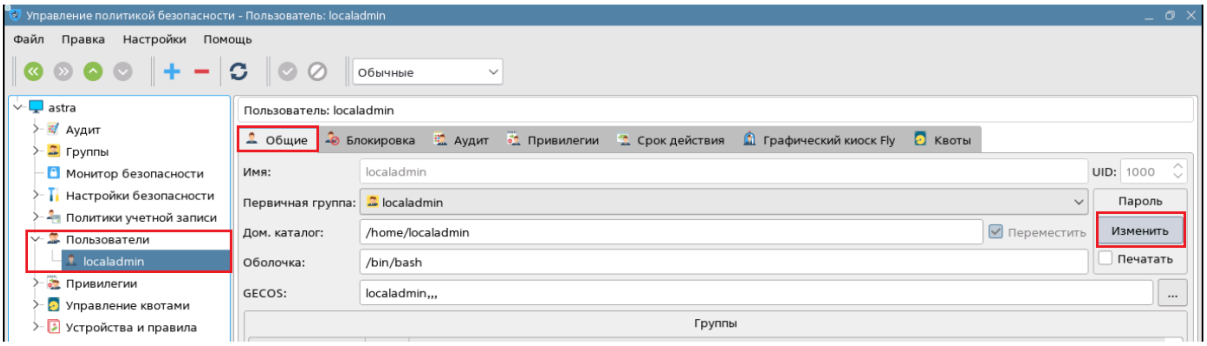
Задание 2.

Чтобы изменить пароль администратора через панель управления:

1. Нажмите кнопку *Пуск*;
2. Запустите приложение «Панель управления»;
3. В левом меню выберите раздел *Безопасность*;
4. Запустите утилиту «Политика безопасности» и введите пароль от администратора для повышения привилегий;



5. В разделе «Пользователи» окна управления политикой безопасности выберите учетную запись администратора;
6. В окне справа на вкладке «Общие» в поле «Пароль» нажмите кнопку *Изменить*.



В процессе проверки пароля система может вывести сообщение о том, что пароль слабый, но при этом система не блокирует его использование, т.е. вы сможете использовать его в дальнейшем на свой страх и риск. Мы рекомендуем использовать сложные пароли для лучшей безопасности!

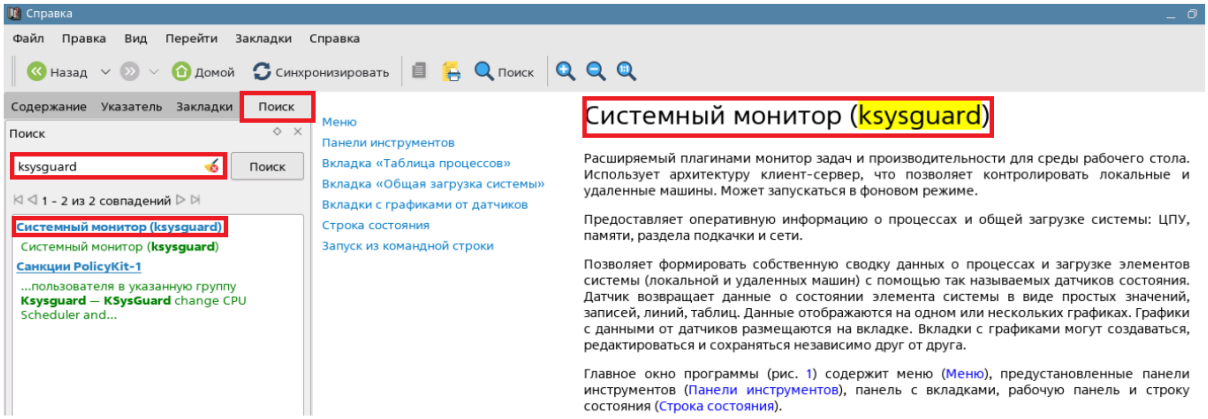
Задание 3.

Чтобы найти информацию о пакете ksysguard:

1. Откройте «Справку» с помощью ярлыка «Помощь» с рабочего стола;

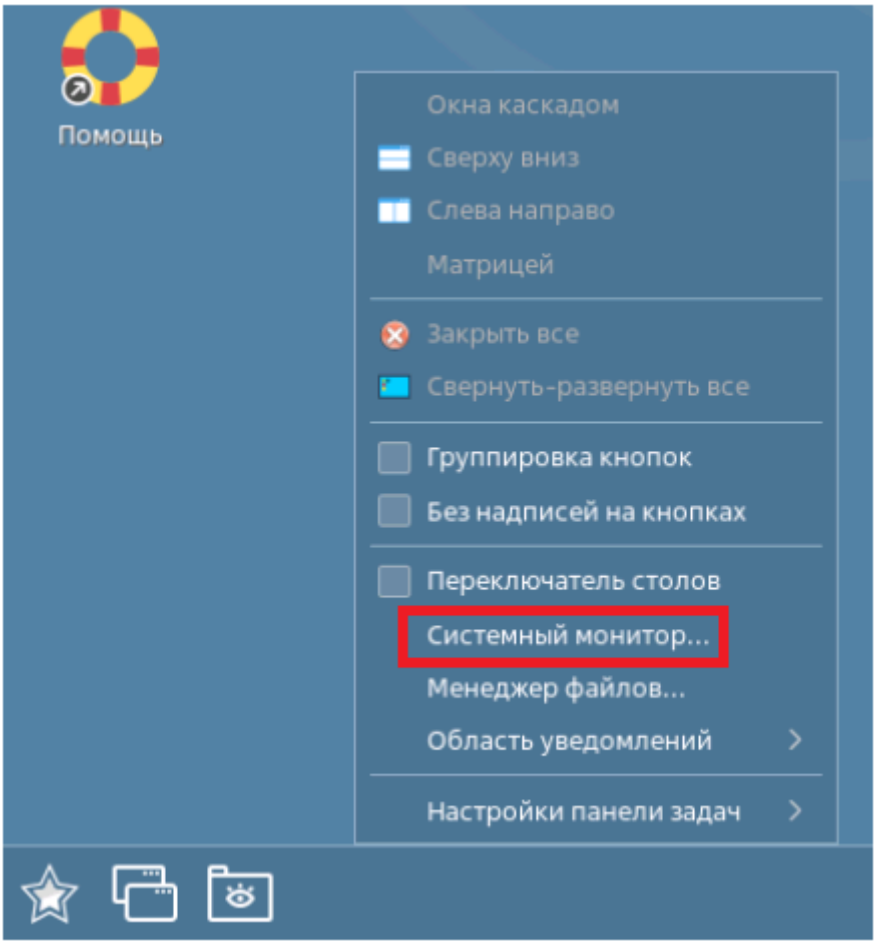
2. В колонке слева выберите вкладку «Поиск»;

3. В соответствующем поле наберите слово «ksysguard» и нажмите кнопку *Поиск*;



4. Перейдите по первой ссылке «Системный монитор» и справа вы увидите описание этого пакета;

5. Для запуска приложения «Системный монитор» достаточно вызвать контекстное меню на панели задач нажатием правой кнопки мыши (а также вы можете использовать горячее сочетание клавиш Ctrl + Alt + Del).



3.5.2 Тест: Модуль 3. Работа с графическим интерфейсом Fly

Пожалуйста, ответьте на несколько вопросов, чтобы закрепить полученные знания. Правильные ответы для самопроверки вы сможете найти в конце списка. Обратите внимание, что у некоторых вопросов может быть несколько верных вариантов ответов.

3.5.2.1 Вопросы

Вопрос 1. У вас произошел сбой графического окружения. Как можно исправить ситуацию наименее затратным и быстрым способом?

a. Перезагрузить компьютер.

b. Выйти из сессии, нажать «Действие» и выбрать «Перезапуск граф. окружения».

c. Поднять заявку в службу поддержки.

d. Выйти из сессии, нажать «Тип сессии» и выбрать «Режим восстановления».

Вопрос 2. Вы включили компьютер и после ввода имени пользователя и пароля вы оказываетесь

в режиме терминала, рабочего стола нет. Почему так происходит?

a. Нарушена файловая система, и вы попали в командную строку grub.

b. Вы ввели неверный логин или пароль.

c. Вы зашли под учетной записью администратора.

d. Предыдущий вход в систему был выполнен в «Режиме восстановления».

Вопрос 3. Какое приложение отвечает за работу рабочего стола?

a. Приложение `fly-wm`

b. Приложение `fly-admin-center`

c. Приложение `fly-fm`

d. Приложение `fly-dm`

Вопрос 4. Вы хотите переключиться на другой рабочий стол, но не видите соответствующего поля в панели задач. Что нужно сделать?

a. Перезагрузить компьютер, это временная проблема.

b. Необходимо вызвать контекстное меню на рабочем столе и выбрать «Переключатель столов».

c. Необходимо вызвать контекстное меню на рабочем столе и выбрать «Создать новый рабочий стол».

d. В Astra Linux только один рабочий стол.

Ответы на вопросы

1. b
2. d
3. a
4. b

3.6 Заключение

В этом модуле мы узнали о том, что такое X Server, и познакомились с основными элементами графического интерфейса Astra Linux: меню пуск, панелью задач и несколькими утилитами, такими как менеджер файлов и системный монитор.

Однако графический интерфейс — это всего лишь надстройка, необязательный компонент системы. В то время как Windows Server Core все еще остается экзотикой, большинство серверов Linux эксплуатируется вообще без графики, поэтому навыки работы в терминале являются обязательными для успешного администрирования ОС на базе ядра Linux. Работе в терминале и оболочке и будет посвящен наш следующий модуль:

3.7 Дополнительные источники информации

- Отсутствуют

3.8 Обратная связь

Если остались вопросы, то их всегда можно задать в специальной теме.

Модуль 4. Работа с терминалом и оболочкой Bash

4.1 Введение

Из этого модуля мы узнаем, чем терминал отличается от консоли, как работает оболочка командной строки, каким сочетанием клавиш можно удивить даже бывалых сисадминов.

4.2 Графический интерфейс в современных системах

В современных ОС графический интерфейс нашел широкое применение – это наглядно и удобно. Однако для выполнения задач администрирования одного только графического интерфейса, как правило, бывает недостаточно, особенно в задачах автоматизации. Даже в ОС Windows, для которой графический интерфейс управления создавался как основное средство взаимодействия с пользователем, все более активно используются CMD и PowerShell, а некоторые настройки, например, для Microsoft Exchange, можно сделать только из командной строки.

4.3 Аппаратный терминал

На заре зарождения вычислительной техники компьютеры позаимствовали у телеграфа так называемые телетайпы (teletype, TeleTYpewriter или сокращённо TTY) — обычные электромеханические печатные машинки, которые использовались для передачи текстовых сообщений между двумя абонентами. Однако использование таких инструментов ввода требовало предельной осторожности в наборе команд, поэтому очень скоро разработчики реализовали технологию редактирования вводимых команд, которую назвали «Line Discipline».

Термин Line Discipline обычно переводят как «Дисциплина линии», что крайне неудачно. Слово «line» в этом контексте означает «строку», а «discipline» можно перевести как «правила», то есть более уместно говорить о «правилах обработки ввода строки».



рис. 1.16 – Телетайп модель 33 ASR

Суть дисциплины линии очень проста: на стороне пользователя есть печатная машинка, а на стороне компьютера есть буфер, куда поступают напечатанные символы, и установлены правила, в соответствии с которыми содержание этого буфера может быть скорректировано при поступлении специальных символов.

В 70-х годах телетайпы оснастили сначала принтерами, а затем дисплеями и научили получать обратную связь от компьютера, что позволило существенно расширить возможности дисциплины линии, например, стало возможным перемещать курсор по строке текста. Устройства, поддерживающие такой режим работы, стали называть терминалами (с англ. terminal или terminal end — оконечное устройство).

Много биткоинов утекло с тех времен, но базовые механизмы ввода до сих пор работают на тех же проверенных принципах. Поэтому с помощью любого сниффера вы можете легко убедиться, что при подключении к серверу через SSH ваш терминал передает по сети нажатие каждой клавиши,

которое обрабатывается дисциплиной линии на стороне сервера, за счет чего работает, например, механизм автоматического дополнения команд по нажатию клавиши Tab.



рис. 1.17 – Терминал VT100

4.4 Системная и виртуальные консоли

Интерфейс для взаимодействия с пользователем в Linux называют консолью. Сразу после включения компьютера в консоли отображается ход загрузки, а после инициализации ядра консоль начинает предоставлять доступ к нескольким виртуальным консолям.

Для переключения между которыми используются горячие клавиши Ctrl + Alt + F<N>, где <N> – целое число от 1 до 8, например: Ctrl + Alt + F1. Виртуальные консоли позволяют выполнять несколько задач параллельно и мониторить работу системы, что было одной из киллер-фич Linux.

В Astra Linux по умолчанию инициализируется 7 виртуальных консолей: tty1 ... tty7. Консоли с первой по шестую являются текстовыми, а в седьмой консоли запускается графика. Каждой виртуальной консоли в системе соответствует одноименный символьный файл этого устройства в каталоге /dev. Например, консоли tty1 соответствует файл /dev/tty1. Устройство /dev/console автоматически привязывается к активной виртуальной консоли. Количеством виртуальных консолей в системе можно управлять (максимум 63 штуки) – для этого есть параметр NAutoVTs в конфигурационном файле /etc/systemd/logind.conf.

Примечание: Есть еще один способ, как можно быстро переключаться между виртуальными консолями. Сочетание клавиш Alt + <стрелка влево> или Alt + <стрелка вправо> позволяет переключаться между консолями по порядку. Однако после перехода в графику это сочетание перестает работать, и, чтобы перейти обратно в текстовую консоль, необходимо снова использовать Ctrl + Alt + <F1-F6>.

Для работы виртуальных консолей в системе Linux запускается несколько экземпляров приложения agetty, каждое из которых подключается к своему tty и реализует приглашение для ввода логина. После того как пользователь введет имя учетной записи, agetty запустит программу login

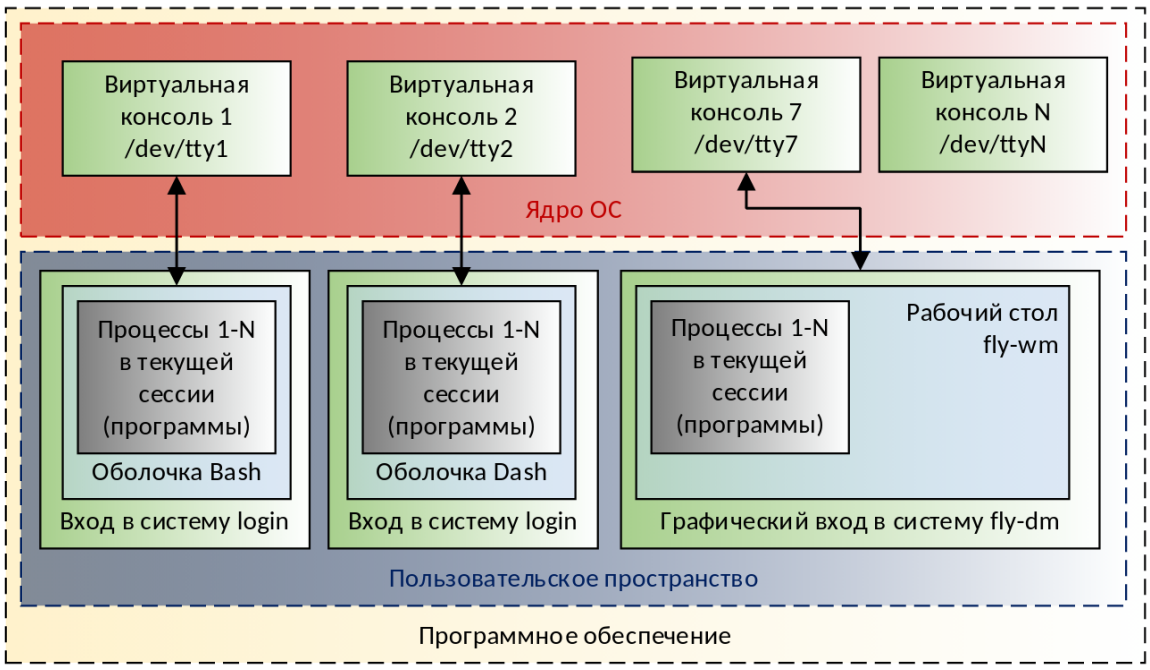
и передаст ей введенное значение. Приложение login запросит у пользователя пароль, выполнит проверку учетных данных через PAM-стек и в случае успешной аутентификации запустит оболочку командной строки Bash из-под этого пользователя.

```
Astra Linux 1.7.3 astra tty1
astra login: localadmin
Password:
Last login: Tue May 16 12:08:32 MSK 2023 on :0
localadmin@astra:~$ _
```

Оболочка (англ. shell) или интерпретатор командной строки — это системное приложение, главное предназначение которого состоит в том, чтобы предоставить пользователю возможность запускать другие приложения.

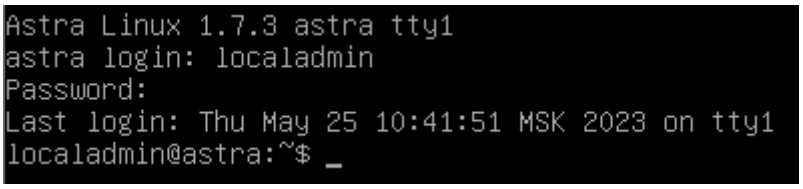
Существует большое количество различных оболочек, самой популярной из которых является Bash (в Astra Linux используется по умолчанию). Другие примеры: Dash (легковесный Shell, доступен, если запустить бинарник по адресу `/bin/sh`), Zsh. По синтаксису все оболочки довольно схожи, так что с первого взгляда не всегда понятно, какая именно оболочка запущена.

Важно: Не путайте понятия виртуальной консоли (или просто консоли) и оболочки. Консоль реализуется функциями ядра, а оболочка является обычной программой пользовательского пространства. На следующей схеме представлена логика взаимодействия между виртуальными консолями, утилитами входа в систему, оболочками и прикладным ПО (допущены некоторые упрощения).



Немного практики:

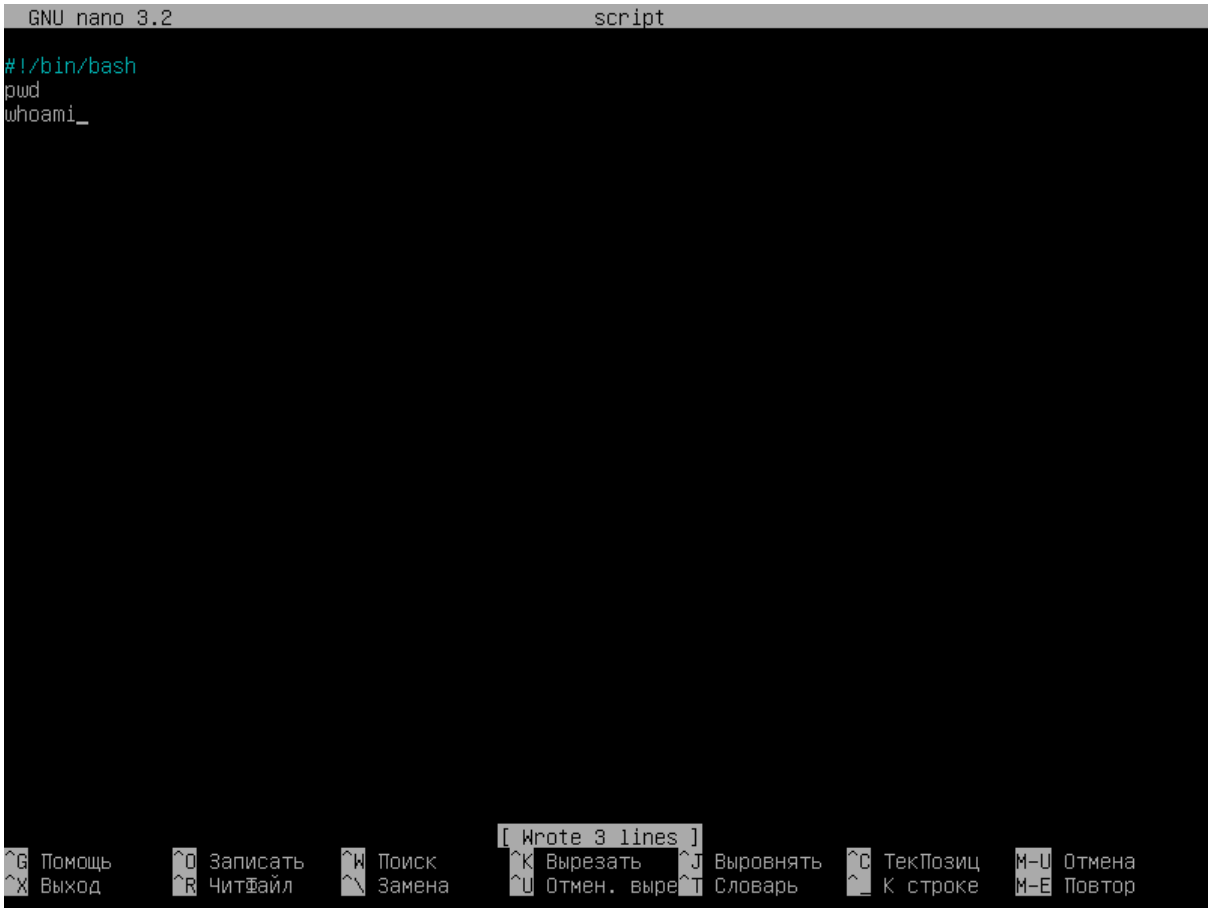
Перейдем в виртуальную консоль tty1 горячим сочетанием клавиш Ctrl + Alt + F1 и введем учетные данные пользователя, который был создан при установке системы (далее под «админом» мы будем иметь в виду именно эту учетную запись).



Выполним команду `nano script`, которая откроет нам текстовый редактор nano. Введем в редактор текст:

```
#!/bin/bash
pwd
whoami
```

Сохраним файл нажатием Ctrl + O и Enter.



Перейдем в виртуальную консоль tty2 нажатием клавиш Ctrl + Alt + F2 и выполним вход от имени адмика.

```
Astra Linux 1.7.3 astra tty2
astra login: localadmin
Password:
Last login: Thu May 25 10:42:13 MSK 2023 on tty1
localadmin@astra:~$ _
```

Посмотрим, какие файлы и каталоги у нас находятся в текущей директории с помощью команды `ls`.

```
localadmin@astra:~$ ls
Desktop   script           VBoxLinuxAdditions.run  Документы  Изображения  Общедоступные
Desktops  SystemWallpapers  Видео                   Загрузки   Музыка       Шаблоны
```

Как видим, у нас появился файл script. Попробуем запустить его, выполнив команду `./script`, но... у нас это не получится. Система скажет нам, что у нас нет прав. Чтобы всё-таки запустить скрипт, выполним команду `sudo chmod u+x script`, которая разрешит выполнять этот файл, и команду `./script`, которая непосредственно запустит этот скрипт.

```
localadmin@astra:~$ ./script
-bash: ./script: Отказано в доступе
localadmin@astra:~$ sudo chmod u+x script
[sudo] пароль для localadmin:
localadmin@astra:~$ ./script
/home/localadmin
localadmin
localadmin@astra:~$ _
```

Скрипт успешно отработал, и на экран будут выведены два сообщения:

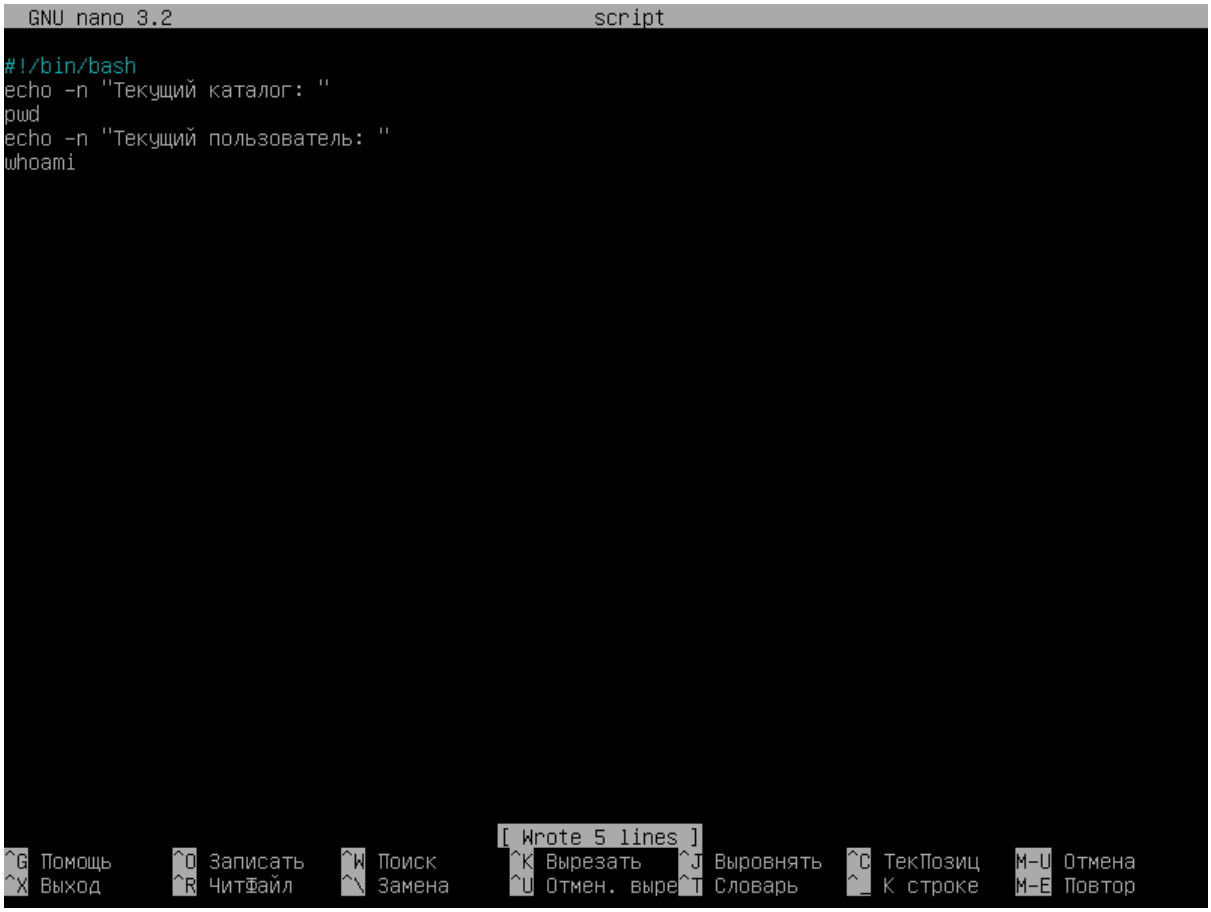
1) текущая директория (в которой он был запущен)

2) и имя учетной записи (от которой был запущен скрипт).

Переключимся на первую консоль tty1, использовав сочетание Alt + <стрелка влево>, и немного изменим наш скрипт:

```
#!/bin/bash
echo -n "Текущий каталог: "
pwd
echo -n "Текущий пользователь: "
whoami
```

Сохраним файл нажатием Ctrl + O и Enter.



Переключимся на вторую консоль tty2, использовав сочетание Alt + <стрелка вправо>, и запустим скрипт еще раз, выполнив команду `./script`:

```
localadmin@astra:~$ ./script
Текущий каталог: /home/localadmin
Текущий пользователь: localadmin
```

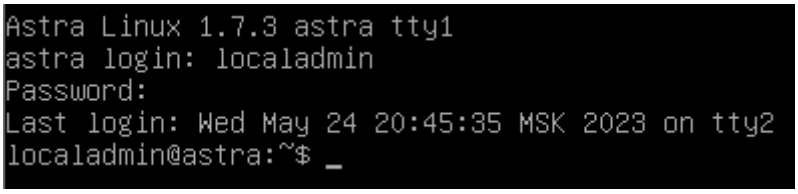
Как видите, вывод скрипта ожидаемо изменился. Таким образом, переключаясь между консолями, можно эффективно работать на компьютерах с Linux даже без графической оболочки.

Примечание: Пользовательский процесс может изменять работу любого TTY-устройства, управляя соответствующим файлом в папке `/dev`. По сути, символьные файлы tty[1-7] не являются файлами с данными, хранимыми на диске, — это специальные файлы устройств, которые являются указателями на их драйверы.

Поэтому, когда процесс обращается к файлу tty[1-7], он, по сути, работает с драйвером этого устройства. Естественно, для этого процесс должен обладать правами записи в этот файл. Когда пользователь входит в систему и подключается к определённому TTY, этот пользователь должен стать владельцем файла, соответствующего этому TTY.

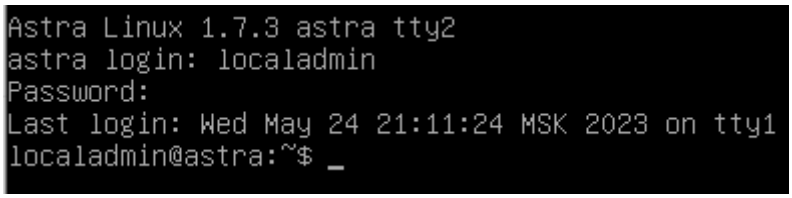
Меняет права доступа к файлу терминала программа login, которая в свою очередь обладает необходимыми привилегиями, т.к. запускается от имени суперпользователя.

Перейдем в виртуальную консоль tty1 от имени админа.



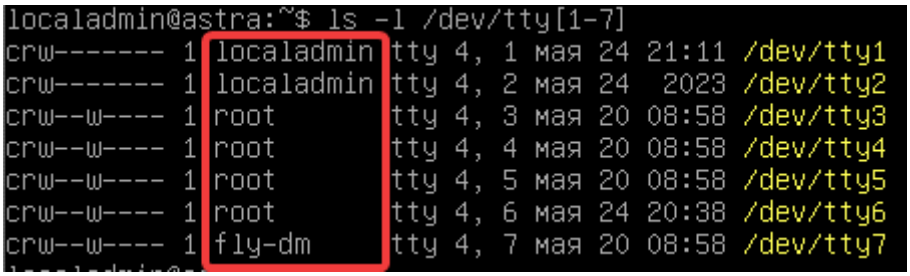
Откроем текстовый редактор nano (просто введите nano и нажмите «Enter»).

Перейдем в виртуальную консоль tty2 и выполним вход от имени админа.



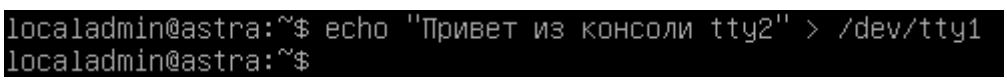
Выполним команду `ls -l /dev/tty[1-7]`, которая выведет на экран подробную информацию о файлах `/dev/tty1`, `/dev/tty2 ... /dev/tty7`. Мы подробно разберем, что же означает этот вывод, в отдельных модулях этого курса, но сейчас нас интересует выделенный столбец. В нем содержится информация о владельце этого файла.

Как видите, владельцами первых двух виртуальных консолей является localadmin, с третьей по шестую – суперпользователь root, а владельцем седьмой консоли, на которой запущен графический интерфейс, является служебный пользователь fly-dm.



А теперь сделаем магический трюк. Оставаясь в консоли tty2, введем команду:

```
$ echo “Привет из консоли tty2” > /dev/tty1
```



Перейдем в виртуальную консоль tty1 и...



мы увидим переданный нами текст.

Введенная нами команда перенаправила текст «Привет из консоли tty2» в консоль tty1 через файл консоли `/dev/tty1`. Однако, если мы опять перейдем на консоль tty2 и попытаемся отправить текст на консоль tty3, то получим сообщение о том, что у нас нет на это прав, потому что мы не являемся владельцем файла `/dev/tty3`.

```
localadmin@astra:~$ echo "Привет из консоли tty2" > /dev/tty3
-bash: /dev/tty3: Отказано в доступе
```

4.5 Эмулятор терминала

Шли годы и в конце 90-х стали появляться первые графические оболочки для Linux. Для обеспечения обратной совместимости и возможности запускать консольные приложения из графической оболочки были разработаны специальные приложения – **эмуляторы терминала** (fly-term, xterm, ssh, telnet).

Предназначение у них ровно такое же, как и у виртуальных консолей, - доставлять ввод пользователя в запущенную программу и отображать вывод программы на дисплей. Но механизм работы в корне отличается, потому что сам эмулятор терминала работает в пользовательском пространстве, а для обратной совместимости работы приложений требовалось обеспечить работу механизмов на уровне ядра ОС. Для этого изобрели такое программное устройство, как **псевдотерминал** (*PTY, акроним Pseudo-TTY*). Сейчас, когда аппаратных терминалов нет, под «терминалом» понимают именно псевдотерминал.

Linux не ограничивает количество псевдотерминалов в системе и создает их по запросам от эмуляторов терминала. Псевдотерминал может быть запущен внутри другого псевдотерминала (так поступают, например, утилита screen или клиент ssh). Каждому псевдотерминалу в системе динамически генерируется номер и создается одноименный файл в каталоге `/dev/pts`, например, файл `/dev/pts/2` – это файл псевдотерминала под номером 2.

Из графического интерфейса эмулятор терминала можно запустить сочетанием клавиш Alt + T или из меню *Пуск → Системные → Терминал Fly*. В одном окне терминала Fly можно создавать несколько вкладок, на каждой из которых будет свой отдельный псевдотерминал. Создать

вкладку можно как с помощью соответствующей кнопки в окне приложения, так и сочетанием клавиш Ctrl + T, а увеличить/уменьшить шрифт можно сочетанием клавиш Ctrl + <плюс/минус> или прокруткой колеса мыши с удержанием клавиши Ctrl.



Посмотреть и изменить настройки отображения терминала можно через меню *Настройка*, например, для просмотра и изменения горячих сочетаний клавиш выберите команду *Настройка* → *Комбинации клавиш...*

Далее в курсе термин « **терминал** » будет означать окно эмулятора терминала fly-term.

На схеме ниже отражена логика взаимодействия между программами и оболочками при работе в эмуляторе терминала (через псевдотерминал).

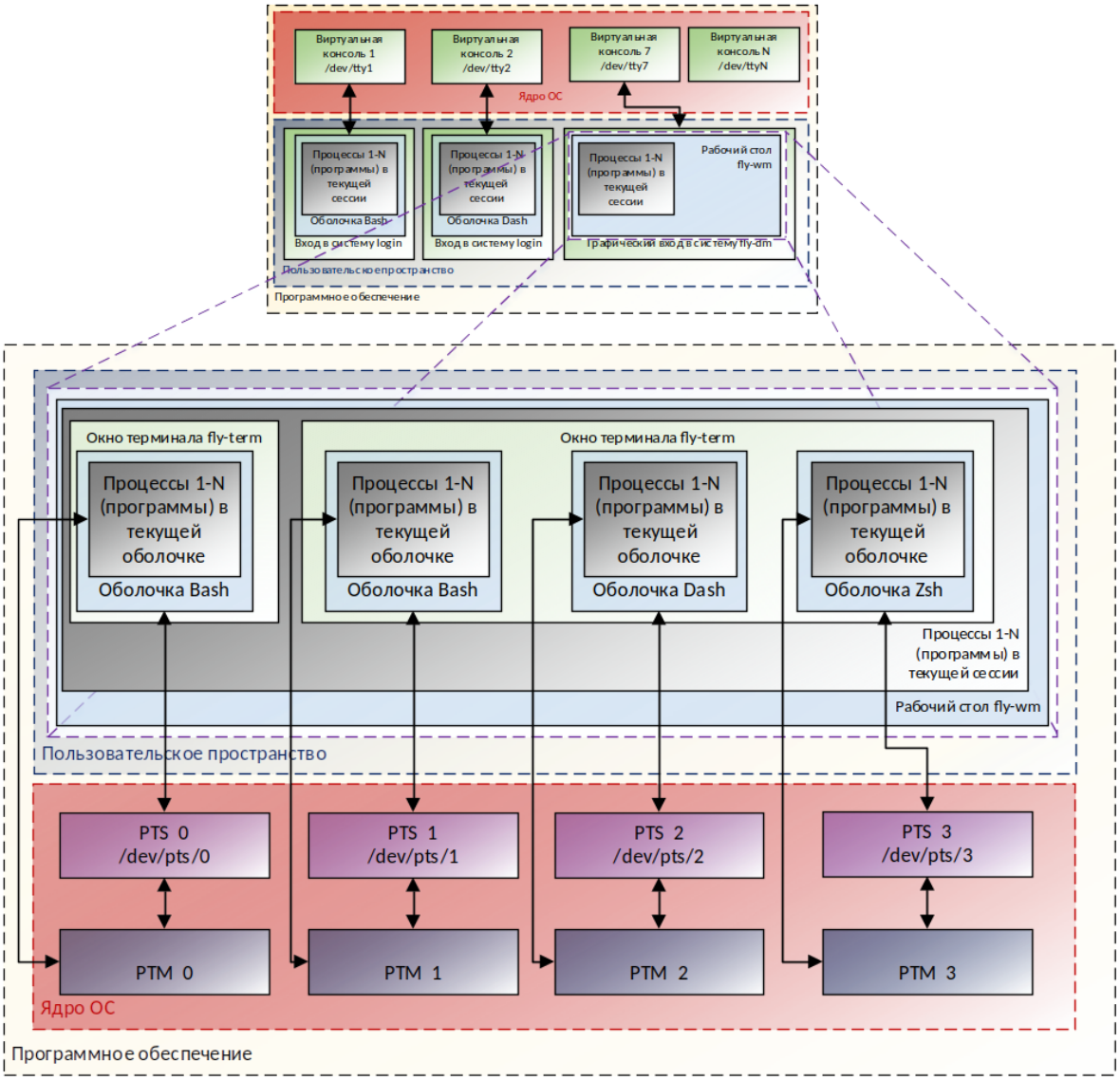


рис. 1.18 – Дополненная схема с работой эмуляторов терминала и псевдотерминалов.

Немного практики:

Для того чтобы узнать номер псевдотерминала, с которым работает ваш терминал, достаточно ввести команду `tty`.

```
localadmin@astra:~$ tty
/dev/pts/0
```

Есть несколько специальных команд, которые терминал умеет обрабатывать особым образом. Всем, кто работал в Windows, хорошо знакомо сочетание Ctrl + C, которое прерывает выполнение программы. В Linux оно работает точно таким же образом.

Примечание: Команды, вызываемые с удержанием клавиши Ctrl, принято обозначать символом

^, после которого следует имя клавиши, то есть горячее сочетание клавиш Ctrl + C — это то же самое, что ^C.

Пример. Введем команду `du /`, которая выводит размер всех каталогов в файловой системе. Это небыстрый процесс, и, если мы не хотим ждать его завершения, то достаточно нажать Ctrl + C, и сигнал SIGINT завершит выполнение этой команды.

```
localadmin@astra:~$ du /
...
0       /proc/391/task/391/fd
0       /proc/391/task/391/fdinfo
du: невозможно прочитать каталог '/proc/391/task/391/ns': Отказано в доступе
0       /proc/391/task/391/ns
0       /proc/391/task/391/net/stat
^C
localadmin@astra:~$
```

Еще полезно знать, что делает сочетание клавиш Ctrl + Z, т.к. его часто используют по привычке для отмены последнего действия в Windows. Если вы воспользуетесь этим сочетанием клавиш во время редактирования текста в редакторе nano, то приложение исчезнет с экрана, и вы увидите сообщение *«Используйте «fg» для возврата в nano»*.

На самом деле, сочетание Ctrl + Z отправляет процесс в фоновый режим, и, чтобы вернуть его обратно в интерактивный режим, вам достаточно будет ввести команду `fg`. Если в фоне находится несколько приложений, то `fg` вернет к жизни последнее из них, но вы можете вызвать команду `fg` с аргументом и явно указать номер требуемого приложения.

Пример. Введем команду `nano test.txt` и напишем в редакторе, например, «Привет мир!».



Теперь нажмем сочетание клавиш Ctrl + Z и получим сообщение, что процесс nano был остановлен.

```
localadmin@astra:~$ nano test.txt
Используйте «fg» для возврата в nano
```

(continues on next page)

(продолжение с предыдущей страницы)

```
[1]+   Остановлен        nano test.txt
localadmin@astra:~$
```

Используя команду fg, переведем процесс nano, отправленный ранее в фон, обратно в интерактивный режим. Как видим, введенный нами текст никуда не делся, то есть это тот же самый процесс.

4.6 Командная оболочка Bash

4.6.1 Командные оболочки

При открытии терминала запускается командная оболочка, указанная в файле /etc/passwd. В современных дистрибутивах основной оболочкой является /bin/bash. Список всех установленных в системе оболочек можно посмотреть в файле /etc/shells.

```
localadmin@astra:~$ cat /etc/shells
# /etc/shells: valid login shells
/bin/sh
/bin/bash
/usr/bin/bash
/bin/rbash
/usr/bin/rbash
/bin/dash
/usr/bin/dash
```

Приведенные значения соответствуют следующим командным оболочкам:

- **/bin/sh** – Bourne Shell, старейшая оболочка UNIX, созданная Стивом Борном в 1979 году, которая в настоящее время практически не используется. Файл /bin/sh в зависимости от дистрибутива является символической ссылкой на /bin/bash или на /bin/dash (в Astra Linux /bin/sh - это символическая ссылка на /bin/bash). Если вызвать /bin/sh, то запустится bash в POSIX-совместимом режиме (с опцией --posix). Более подробно о POSIX режиме написано на сайте Bash-POSIX-Mode.html.

- **/usr/bin/bash** – Bourne Again Shell (возрожденный Shell), усовершенствованный командный интерпретатор sh от проекта GNU, который в настоящее время является одной из самых распространенных оболочек в Linux. Этот интерпретатор используется в Astra Linux по умолчанию.

- **/bin/rbash** – это bash, но с некоторыми ограничениями. Запускается по имени rbash или bash с добавлением параметра -r. В такой оболочке не разрешен, например, переход в другой каталог, невозможно поменять переменные окружения SHELL, PATH, ENV или BASH_ENV. Более подробно этот режим описан в man-документации на утилиту bash (секция «Ограни-

ченный командный интерпретатор»).

- **/bin/dash** – реализация Bourne Shell, совместимая с POSIX стандартом. Занимает всего 112 Кбайт на диске, в то время как размер bash составляет 1,1 Мбайт. По сравнению с bash имеет меньше функций для интерактивной работы в командной строке, чаще используется в скриптах, так как имеет меньший размер и высокую скорость работы по сравнению с bash.

4.6.2 Приглашение к вводу команд

При открытии терминала мы видим командную строку с приглашением ввода.

```
localadmin@astra:~$
```

Вот, что нам может сказать строка приглашения:

Часть	Значение
localadmin	текущий пользователь, от имени которого запущена командная оболочка
@	разделитель, как в электропочте
astra	имя компьютера
:	разделитель имени и рабочей директории
~	текущая директория (символ `~` обозначает домашний каталог пользователя)
$	режим работы: • **$** - с правами простого пользователя • # - с правами суперпользователя root

---

Примечание: Текст приглашения командной строки можно менять. Он формируется на основе настроек в профиле пользователя, которые будут рассмотрены в модуле: *Модуль 12. Работа с профилями пользователей*.

4.6.3 Структура и синтаксис команд

В командных оболочках мы можем выполнять команды, большинство из которых вызываются как внешние утилиты, но некоторые команды все-таки встроены в саму оболочку.

Например, команда `echo` по умолчанию является встроенной командой, поэтому при выполнении

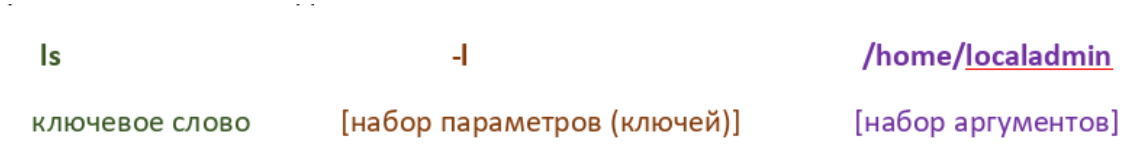
`type echo` мы увидим «echo — это встроенная команда bash».

Но если отключить эту встроенную команду `enable -n echo`, то команда `type` будет выдавать уже «echo является /usr/bin/echo», т.е. для выполнения `echo` будет вызываться уже внешняя утилита, см. пример:

```
localadmin@astra:~$ type echo
echo - это встроенная команда bash
localadmin@astra:~$ enable -n echo
localadmin@astra:~$ type echo
echo является /usr/bin/echo
localadmin@astra:~$
```

Список всех встроенных команд, доступных в оболочке, можно посмотреть с помощью команды `help`.

Рассмотрим синтаксис команд Bash:



- **Ключевое слово** может содержать путь до бинарного файла утилиты. Например, `ls` и `/usr/bin/ls` — это одно и то же. Если полный путь не задан, то оболочка выполнит поиск по системным каталогам, которые указаны в переменной PATH.
- **Параметры и аргументы** являются необязательными, их может вообще не быть. Для указания ключей в Linux одновременно используют три разных синтаксиса:
  - **синтаксис POSIX** – перед ключом ставится один символ дефиса, а сам ключ представляет собой строчную или прописную букву латинского алфавита. Ключи могут группироваться, когда после одного дефиса указывается сразу несколько ключей без пробелов. Например, ps -aux.
  - **синтаксис BSD** – перед ключом не ставится символ дефиса. Например, ps aux.
  - **синтаксис GNU** – используются длинные ключи, перед которыми ставится двойной дефис. Например, `ps --sort cmd`. Длинные ключи могут дублировать короткие, но часто бывает, что между ними нет аналогов.
  - **Но есть и исключения** из этих правил, например, команда `find` использует ключи: *-name*, *-type*, *-size* и др.
- **Ключи и аргументы** являются регистрозависимыми, то есть ключ «-a» и «–A» — это два совершенно разных ключа. В этом Linux сильно отличается от Windows, что многих сначала

сильно путает.

- **Лишние пробелы** в командах игнорируются.

Примечание: В переменной окружения $PATH содержится список каталогов, в которых система будет искать исполняемый файл. В Windows этот механизм работает схожим образом.

```
localadmin@astra:~$ echo "$PATH"
/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games
```

4.6.4 Несколько базовых команд Bash

Давайте познакомимся с несколькими наиболее часто употребимыми командами:

- `whoami` (кто я?) — команда выводит на экран текущего пользователя, от имени которого запущен терминал:

```
localadmin@astra:~$ whoami
localadmin
```

- `echo` (эхо) — команда для отображения строки текста:

```
localadmin@astra:~$ echo "Привет"
Привет
```

- `pwd` (Print Working Directory, напечатать рабочую директорию) — команда для вывода на экран текущего каталога:

```
localadmin@astra:~$ pwd
/home/localadmin
```

- `cd` (Change Directory – изменить директорию) — команда для перехода по дереву каталогов. Выполним команду `cd /` и перейдем в корень файловой системы, а с помощью команды `cd -` мы вернемся обратно в каталог, в котором были:

```
localadmin@astra:~$ cd /
localadmin@astra:/$ pwd
/
localadmin@astra:/$ cd -
/home/localadmin
```

4.6.5 Окружение пользователя

При запуске оболочка инициализирует *окружение пользователя* — область памяти, в которой хранится набор переменных, с помощью которых осуществляется управление программами.

Для инициализации окружения оболочка bash просматривает и запускает команды из определенных файлов, находящихся в домашней директории пользователя и в системной директории /etc.

Существует несколько вариантов запуска bash (подробнее можно почитать на сайте проекта GNU 6.2 Bash Startup Files):

- **Интерактивный режим оболочки входа** (interactive login shell). Если пользователь подключается к удаленной машине по ssh, заходит на машину через локальную консоль или запускает команду `su -`, то после ввода логина и пароля запускается командная оболочка bash, которая просматривает инструкции из файлов в следующем порядке: `/etc/profile`, `~/.bash_profile`, `~/.bash_login`, `~/.profile`.

Примечание: В Astra Linux сразу после установки отсутствуют файлы `~/.bash_profile` и `~/.bash_login`.

- **Интерактивный режим без входа в систему** (interactive non-login shell). Если пользователь уже находится в графической оболочке или в текстовой консоли, то при открытии нового окна терминала запускается оболочка bash, которая читает и выполняет инструкции из файла `~/.bashrc`.
- **Неинтерактивный режим оболочки входа** (non-interactive login shell). Если мы запускаем bash-скрипт, то оболочка смотрит на переменную окружения BASH_ENV и выполняет инструкции, которые находятся в указанном файле.
- **Неинтерактивный режим без входа в систему** (non-interactive non-login shell). Если выполнить команду `bash -c command`, то в этом режиме также выполняются инструкции из файла, заданного переменной окружения BASH_ENV.

Примечание: Подробнее о профиле пользователя и порядке применения инструкций из конфигурационных файлов будет рассказано в *Модуль 12. Работа с профилями пользователей*

4.6.6 Переменные в Bash

Переменная — это ссылка на участок памяти, в котором содержится информация для управления приложениями. Эта информация может быть считана, записана и перезаписана неограниченное количество раз.

Следует различать имя переменной (например, var1) и ссылку на её значение $var1. Если нам нужно определить переменную, то мы используем синтаксис `var1=5`, а если нам нужно вывести значение переменной на экран, то это делается так: `echo $var1`.

4.6.6.1 Типы переменных

Переменные в Bash можно разделить на три группы:

1. **переменные среды** (переменные окружения, глобальные переменные)
2. **пользовательские переменные**
3. **особые переменные** (позиционные)

4.6.6.2 Переменные среды (окружения)

Переменные среды заданы в файлах `/etc/.profile`, `~/.profile`, `~/.bashrc`, `~/.bash_profile` и инициализируются при загрузке оболочки.

Переменные окружения принято писать прописными буквами. Для просмотра всех переменных среды, доступных в текущей оболочке, можно воспользоваться командой `env` или `printenv`.

Вот некоторые из таких переменных:

- **$BASH** - полный путь до исполняемого файла командной оболочки Bash.
- **$BASH_VERSION** - версия Bash.
- **$CDPATH** - хранит пути поиска каталога (используется при вводе команды `cd имя_каталога` без слэша).
- **$CLASSPATH** - содержит список каталогов для поиска файлов классов Java и архивов Java.
- **$HOME** - домашний каталог текущего пользователя.
- **$HOSTNAME** - хранит имя компьютера.
- **$HISTSIZE** - количество событий, хранимых в истории за 1 сеанс.

- **$HISTFILE** -расположение файла истории событий.
- **$HISTFILESIZE** - количество событий, хранимых в истории между сеансами.
- **$IFS** - хранит символы, являющиеся разделителями команд и параметров (по умолчанию - пробел, табуляция и новая строка).
- **$LANG** - текущая установка локализации, которая позволяет настроить командную оболочку для использования в различных странах и на различных языках.
- **$OSTYPE** - содержится описание операционной системы.
- **$PATH** - список каталогов для поиска команд и приложений, когда полный путь к файлу не задан. Когда пользователь вводит команду, система будет проверять каталоги в указанном здесь порядке при поиске исполняемого файла.
- **$PS1** - PS1 используется как основная строка приглашения (в нашем случае *[localadmin@astra ~]$*).
- **$PS2** - PS2 используется как вторичная строка приглашения.
- **$PROMPT_COMMAND** - эта команда должна быть выполнена до отображения строки приглашения Bash.
- **$PWD** - полный путь к текущему рабочему каталогу.
- **$OLDPWD** - предыдущий рабочий каталог. Эта информация сохраняется оболочкой, чтобы выполнять переход к предыдущему каталогу с помощью команды `cd -`.
- **$SHELL** - полный путь к текущей командной оболочке.
- **$USER** - имя текущего пользователя.

4.6.6.3 Пользовательские переменные

При запуске оболочки никаких пользовательских переменных в ней нет. Пользовательская переменная появляется, только если пользователь сам объявит переменную, присвоив ей какое-либо значение.

Примечание: Пользовательская переменная будет доступна только в пределах процесса оболочки, в которой она была объявлена, однако её можно превратить в переменную среды командой `export`, тогда она станет доступна и для дочерних процессов, запускаемых оболочкой.

4.6.6.4 Позиционные переменные

При вызове команды или сценария с аргументами имя команды и ее аргументы являются позиционными переменными. Позиционными они называются, потому что внутри сценария обращение к ним происходит по позиции в командной строке. Позиционные переменные мы рассмотрим в модуле, посвященном сценариям Bash.

4.6.7 Действия над переменными

Рассмотрим несколько базовых действий над переменными.

4.6.7.1 Объявление переменной и присвоение ей значения

У объявления переменной или присвоения ей значения следующий синтаксис:

`var1="Переменная1"`

Если значение переменной не содержит пробелов, то кавычки можно опустить, в противном случае они обязательны:

```
localadmin@astra:~$ var1=Переменная1
localadmin@astra:~$ var2="Переменная     2"
localadmin@astra:~$ var3=Переменная 3
bash: 3: команда не найдена
```

Пробелы между именем переменной, знаком равно и её значением недопустимы:

```
localadmin@astra:~$ var3 ="Переменная 3"
bash: var3: команда не найдена
localadmin@astra:~$ var3= "Переменная 3"
bash: Переменная 3: команда не найдена
```

Значение одной переменной можно присвоить другой `var3=$var1`:

```
localadmin@astra:~$ var3="$var1"
```

4.6.7.2 Вывод значения переменной

При выводе переменной необходимо использовать не имя этой переменной, а ссылку на её значение `echo $var1` (символ **$** перед ней):

```
localadmin@astra:~$ echo "$var1"
Переменная1
localadmin@astra:~$ echo var1
var1
```

Если в значении переменной между словами более одного пробела, то её значение необходимо обернуть в кавычки, иначе пробелы в выводе «пропадут»:

```
localadmin@astra:~$ echo "$var2"
Переменная     2
localadmin@astra:~$ echo $var2
Переменная 2
```

Внутри одинарных кавычек, а также при экранировании символа $ не производится подстановка значений переменных:

```
localadmin@astra:~$ echo '$var1'
$var1
localadmin@astra:~$ echo \$var1
$var1
```

4.6.7.3 Обнуление и удаление переменной

Для обнуления значения переменной достаточно присвоить ей пустое значение:

```
localadmin@astra:~$ var1=
```

Однако это не удалит саму переменную. Для удаления переменной предназначена команда `unset <имя переменной>`:

```
localadmin@astra:~$ unset var1
```

4.6.7.4 Превращение пользовательской переменной в переменную среды

Для превращения пользовательской переменной, доступной только в пределах процесса текущей оболочки, в переменную среды, которая доступна и в дочерних процессах, необходимо воспользоваться командой `export`:

```
localadmin@astra:~$ env | grep var2
localadmin@astra:~$ export var2
localadmin@astra:~$ env | grep var2
var2=Переменная     2
```

Если мы теперь откроем дочерний процесс bash и попытаемся вывести значения переменных, то мы увидим, что переменная var2 нам доступна, а переменная var3 – нет:

```
localadmin@astra:~$ echo "$var2"
Переменная     2
localadmin@astra:~$ echo "$var3"
Переменная 3
localadmin@astra:~$ bash
localadmin@astra:~$ echo "$var2"
Переменная     2
```

(continues on next page)

(продолжение с предыдущей страницы)

```
localadmin@astra:~$ echo "$var3"
localadmin@astra:~$ exit
```

4.6.8 Псевдонимы в Bash

Псевдонимы в Bash - это не что иное, как «горячие клавиши», средство, позволяющее избежать набора длинных строк в командной строке. Если, к примеру, в файл `~/.bashrc` вставить строку `alias lm="ls -l | more"`, то потом вы сможете экономить свои силы и время, набирая просто команду `lm`. Установив `alias rm="rm -i"` (интерактивный режим удаления файлов), вы сможете избежать многих неприятностей, потому что сократится вероятность удаления важных файлов по неосторожности.

4.6.8.1 Создание псевдонимов

Для создания псевдонима существует команда: `alias имя_псевдонима="значение"`

```
localadmin@astra:~$ alias ll="ls -la"
localadmin@astra:~$ ll
итого 6632
drwx------ 34 localadmin localadmin    4096 мая 24 12:19  .
drwxr-xr-x  8 root       root          4096 июл 26 10:38  ..
-rw------- 1 localadmin localadmin   26916 мая 24 12:18  .bash_history
-rw-r--r-- 1 localadmin localadmin     220 мая 16 12:06  .bash_logout
-rw-r--r-- 1 localadmin localadmin    3526 мая 16 12:06  .bashrc
```

Созданный таким образом псевдоним будет работать только в текущей консоли. Чтобы сделать его постоянным, необходимо добавить эту команду в файл `~/.bash_profile` или `~/.bashrc`.

4.6.8.2 Просмотр псевдонимов

Чтобы посмотреть все псевдонимы, необходимо использовать ключ `-p`: `alias -p`

```
localadmin@astra:~$ alias -p
alias ll='ls -ll'
alias ls='ls --color=auto'
```

4.6.8.3 Удаление псевдонима

Для удаления псевдонима существует специальная команда `unalias <имя псевдонима>`:

```
localadmin@astra:~$ unalias ll
localadmin@astra:~$ alias -p
alias ls='ls --color=auto'
```

4.6.9 Автодополнение и редактирование команд

- Tab - это одна из наиболее часто используемых горячих клавиш, которая позволяет дополнить название команды или путь к файлу. Одинарное нажатие приводит к автоматическому дополнению команды, если возможен только один вариант. Двойное нажатие отображает на экране все возможные варианты, если их несколько.
- Ctrl + Shift + <минус> - это сочетание клавиш отменяет последний ввод в терминал. Можно использовать для отмены автодополнения, если выбрали не тот вариант.
- Ctrl + T - меняет местами два соседних символа — тот, на котором курсор, и стоящий перед ним (слева). Удобно для исправления опечаток.

Примечание: В терминале fly-term на это сочетание клавиш назначено создание новой вкладки.

- Alt + T - меняет местами два соседних слова — то, на котором находится курсор, и стоящее перед ним (слева).

Примечание: В графической оболочке Fly на это сочетание клавиш назначено открытие нового окна терминала.

- Alt + Backspace - удаляет слово, стоящее перед курсором (слева), или часть слова, если курсор находится в его середине.
- Alt + D - удаляет слово, стоящее после курсора (справа), или часть слова, если курсор находится в его середине.

4.6.10 Управление курсором в терминале

Для быстрого и комфортного перемещения по тексту в командной строке (а команды и их последовательности бывают довольно длинные), стрелок навигации может быть недостаточно. Для этого есть несколько полезных комбинаций:

- Ctrl + A - переход в начало строки. Аналог клавиши Home.
- Ctrl + E - переход в конец строки. Аналог клавиши End.
- Ctrl + B - перемещение курсора назад (влево) на один символ, но проще стрелками.
- Ctrl + F - перемещение курсора вперёд (вправо) на один символ, но проще стрелками.

- Alt + B - перемещение курсора назад (влево) на одно слово.
- Alt + F - перемещение курсора вперед (вправо) на одно слово.
- Ctrl + XX - пожалуй, это для самых преданных фанатов Linux. Выполняет перемещение курсора между двумя точками, причем цикличное. Как правило, это начало и конец строки. Однако, если после перемещения в начало строки, был набран какой-то текст длины m, то курсор будет перемещаться между символом в строке под номером m и n-m, где m – текущая длина строки.

4.6.11 Работа с историей команд Bash

Все команды, которые вы набираете в окне терминала, попадают в историю команд. Если ввести команду и нажать стрелку вверх, то последняя выполненная команда появится в строке ввода, и вы сможете её выполнить повторно сразу или после того, как отредактируете. В командных оболочках Windows это работает таким же образом.

История команд хранится как в оперативной памяти, так и в файле `.bash_history`, который расположен в домашней папке пользователя. В Astra Linux по умолчанию в историю команд попадают команды со всех сессий и терминалов пользователя, это удобно.

Примечание: Количество строк, хранимых в истории, определяется переменными HISTSIZE и HISTFILESIZE. Первая определяет количество строк, хранимых в памяти, вторая — в файле. Значения этих переменных по умолчанию задаются в профиле пользователя `~/.bashrc` и равны 1000 и 2000 соответственно.

В профиле задаются также опции оболочки, которыми можно управлять с помощью команды `shopt`. Например, чтобы включить сохранение истории со всех сессий и терминалов пользователя, в профиле прописана команда `shopt -s histappend`. Опциями оболочки можно управлять и с помощью команды `set`, список таких опций доступен в переменной SHELLOPTS.

Профили пользователей и работу с ними мы подробно разберем в отдельном модуле этого курса.

4.6.11.1 Просмотр команд в истории и передвижение по ней

- Ctrl + P - переход на одну команду вверх, альтернатива клавише стрелка вверх>.
- Ctrl + N - переход на одну команду вниз, альтернатива клавише стрелка вниз>.
- `history` - вывести на экран всю историю команд.

- `history 5` - вывести на экран 5 последних команд.

```
localadmin@astra:~$ history 5
38 echo "$var2"
39 echo "$var3"
40 bash
41 history
42 history 5
```

- `history | grep pwd` - отфильтровать историю с помощью утилиты `grep`.

```
localadmin@astra:~$ history | grep pwd
4 pwd
8 pwd
11 pwd
15 pwd
43 history | grep pwd
```

4.6.11.2 Поиск команд в истории

Команды в истории можно искать по совпадению с введенной строкой. Для этого есть два сочетания клавиш.

- `Ctrl + R` - включение режима поиска. После включения режима поиска наберите текст, который должен содержаться в искомой команде, и будет найдено первое совпадение от конца к началу истории. Для перехода к следующему совпадению нажимайте это же сочетание `Ctrl + R`. Чтобы выйти из режима поиска, нажмите `Ctrl + C`.

```
(reverse-i-search)`pwd': history | grep pwd
```

- `Ctrl + S` - поиск совпадения в обратную сторону. Поиск всегда начинается с конца истории, поэтому его следует начинать с `Ctrl + R`. Чтобы выйти из режима поиска, нажмите `Ctrl + C`.

```
localadmin@astra:~$
(i-search)`echo': echo "$var2"
```

Примечание: На это же сочетание `Ctrl + S` назначена «заморозка» терминала, когда вывод информации на него приостанавливается, но по умолчанию эта функция отключена. Однако, если терминал перестал выводить информацию, его можно «разморозить», нажав сочетание `Ctrl + Q`.

4.6.11.3 Выполнение команд из истории

С помощью символа « **!** » можно выполнить команду из истории по её номеру, например, `!39`:

```
localadmin@astra:~$ history 7
39  echo "$var3"
40  bash
41  history
42  history 5
43  history | grep pwd
44  history -d 44
45  history 7
localadmin@astra:~$ !39
echo "$var3"
Переменная 3
```

Это выражение может быть частью другой команды, например, `!41 | grep pwd`:

```
localadmin@astra:~$ !41 | grep pwd
history | grep pwd
   4  pwd
   8  pwd
  11  pwd
  15  pwd
  43  history | grep pwd
  50  history | grep pwd
```

Нумерацию команд можно вести с конца списка, используя относительные, отрицательные значения, например, `!-2`:

```
localadmin@astra:~$ history 3
53  history | grep pwd
54  pwd
55  history 3
localadmin@astra:~$ !-2
pwd
/home/localadmin
```

А специальным выражением **!!** можно быстро вызвать предыдущую команду еще раз:

```
localadmin@astra:~$ !!
pwd
/home/localadmin
```

Можно совместить поиск команды в истории и её выполнение. Выражением **!начало_команды** можно выполнить поиск первого совпадения по началу команды в истории и выполнить её:

```
localadmin@astra:~$ !echo
echo "$var3"
Переменная 3
```

А выражением **!?часть_команды?** можно выполнить поиск первого совпадения по любой части команды и выполнить её:

```
localadmin@astra:~$ !?$var2?
echo "$var2"
Переменная     2
```

4.6.11.4 Очистка истории команд

При выполнении команд, содержащих конфиденциальные данные, например, пароль учетной записи, эти данные попадут в историю команд. Чтобы такого не случилось, вы можете поставить пробел перед именем команды. Если вам потребуется исключить из занесения в историю несколько команд, то вы можете временно отключить журналирование. Если же команда уже попала в историю, то вы можете ее удалить оттуда.

Для временного отключения журналирования команд необходимо выполнить команду `set +o history`, а после `set -o history`, например:

```
$ set +o history
$ somethingScript -u username -p password -a parameter
$ set -o history
```

Если конфиденциальные данные уже попали в историю команд, то их можно удалить. Можно выборочно удалять команды или очистить всю историю сразу.

- `history -d <Номер_команды_в_истории>` - удаление команды из истории по её номеру, например, `history –d 58`:

```
localadmin@astra:~$ history 3
57  echo "$var3"
58  echo "$var2"
59  history 3
localadmin@astra:~$ history -d 58
localadmin@astra:~$ history 4
57  echo "$var3"
58  history 3
59  history -d 58
60  history 4
```

- `history -c` - полная очистка истории команд

```
localadmin@astra:~$ history -c
localadmin@astra:~$ history
   1  history
```

Удаление команд из истории можно произвести, отредактировав файл `.bash_history` напрямую. В текстовом редакторе можно выборочно менять его содержимое, а для полного удаления содержимого можно выполнить команду `cat /dev/null > ~/.bash_history`.

Примечание: обычный пользователь может просматривать и редактировать только свою историю команд, однако суперпользователь root может просматривать и редактировать историю команд всех пользователей в системе.

4.6.12 Запуск нескольких команд за раз

Оболочка Bash позволяет нам запускать несколько команд за раз, управляя при этом порядком их выполнения.

- Линейное выполнение – несколько команд будут выполняться поочередно друг за другом, независимо от результата выполнения друг друга. Для этого их необходимо разделять символом точки с запятой « **;** »:

```
localadmin@astra:~$ echo -n "Привет ";whoami
Привет localadmin
```

Ключ «n» в команде `echo` был использован, чтобы после вывода слова “Привет” не вводился невидимый символ перевода каретки и имя пользователя было выведено в той же самой строке.

- Линейное выполнение **зависимых** команд – следующая команда будет выполнена только в случае успешного завершения предыдущей. В этом случае разделителем команд будет оператор двойной амперсанд «**&&**» (логическое и):

```
localadmin@astra:~$ echo -n "Привет, " && whoami
Привет, localadmin

localadmin@astra:~$ WrongCommand && echo "Все работает!"
bash: WrongCommand: команда не найдена
localadmin@astra:~$
```

- Нелинейное выполнение зависимых команд – следующая команда будет выполнена только в случае, если предыдущая завершилась с ошибкой. В этом случае разделителем команд будет оператор «||» (логическое или):

```
localadmin@astra:~$ WrongCommand || echo "Что-то пошло не так!"
bash: WrongCommand: команда не найдена
Что-то пошло не так!
localadmin@astra:~$
```

- Параллельное выполнение независимых команд. В этом случае команды необходимо взять в скобки, а скобки разделить символом « **&** »:

```
localadmin@astra:~$ (echo "первое,")&(echo "второе,")&(echo "компот");echo " и можно␣
↪все сразу!"
[5] 13473
[6] 13474
первое,
компот
второе,
[5]   Завершён        ( echo "первое," )
и можно все сразу!
[6]   Завершён        ( echo "второе," )
```

4.6.13 Стандартные потоки и перенаправление ввода/вывода

Если вы когда-нибудь работали с cmd или .bat файлами (батниками) в Windows, то, вероятно, сталкивались с конструкцией «2>». Ну, а с конструкциями «>», «>>» уж точно знакомы все. В Linux все работает таким же образом.

В Linux существуют 3 стандартных потока, которые обозначаются положительным целым числом:

- `0` – стандартный ввод (stdin) используется для передачи данных от пользователя (обычно это ввод данных с клавиатуры или из файла) к программе.
- `1` – стандартный вывод (stdout) используется для передачи данных, сгенерированных во время работы программы. По умолчанию данные выводятся на экран, но могут быть перенаправлены.
- `2` – вывод ошибок (stderr), которые могут возникнуть при выполнении программы. Ошибки выводятся отдельным потоком по умолчанию на экран, но могут быть перенаправлены.

Из стандартного потока ввода программа может только читать данные, а два других потока используются только для записи. Данные выводятся на экран, а считываются с клавиатуры, так как стандартные потоки ассоциированы с терминалом пользователя. Потоки можно подключать к файлам, программам и даже устройствам. В Bash такая операция называется перенаправлением.

Существует несколько способов перенаправления:

- `< file` – использовать файл как источник данных для стандартного потока ввода.
- `> file` – направить стандартный поток вывода в файл. Если файл не существует, он будет создан, если существует — перезаписан сверху. **Внимание! Все содержимое файла будет безвозвратно удалено!**
- `>>file` – направить стандартный поток вывода в файл. Если файл не существует, он будет создан, а если существует, то данные будут добавлены в конец файла.
- `2>file` – направить стандартный поток ошибок в файл.
- `2>>file` – направить стандартный поток ошибок в файл. Если файл не существует, он будет создан, если существует — данные будут дописаны к нему в конец.
- `&>file` или `>&file` – направить стандартный поток вывода и стандартный поток ошибок в файл.

  Другая форма записи: `>file 2>&1`.

Разделение потоков позволяет, например, записывать вывод программы в один файл, а ошибки в

другой.

Примеры:

- перенаправление результата выполнения команды в файл.

```
localadmin@astra:~$ echo "Привет!" > hello.txt
localadmin@astra:~$ cat hello.txt
Привет!
```

В примере использована команда `cat`, которая копирует стандартный поток ввода в стандартный поток вывода. Её часто применяют для просмотра содержимого файлов, объединения нескольких файлов в один и записи короткого текста в файл.

- перенаправление результата выполнения команды в конец файла.

```
localadmin@astra:~$ echo "Как дела?" >> hello.txt
localadmin@astra:~$ cat hello.txt
Привет!
Как дела?
```

- передача содержимого в команду в качестве аргументов.

```
localadmin@astra:~$ cat < hello.txt
Привет!
Как дела?
```

4.6.14 Конвейер

Конвейер (*англ. pipe*) – очень мощный инструмент, позволяющий передавать результаты работы одних команд в виде аргументов другим командам. Для этого используется оператор | (вертикальная черта). Тем, кто работал в PowerShell, этот инструмент должен быть хорошо знаком.

При его использовании стандартный вывод команды слева перенаправляется на стандартный ввод команды справа. При этом количество команд и перенаправлений не ограничено, что позволяет строить длинные цепочки команд, например, `cmd1 | cmd2 | .... | cmdN`. При этом конвейер обеспечивает асинхронное выполнение команд с использованием буферизации ввода/вывода. То есть нам не нужно дожидаться полного завершения первой команды, чтобы началось выполнение второй — первая команда будет передавать свои результаты второй команде по мере готовности.

Пример:

`ls` – команда выводящая содержимое текущей директории. Выполним эту команду:

```
localadmin@astra:~$ ls
Desktop    script                   Видео      Изображения    Шаблоны
Desktops   SystemWallpapers         Документы  Музыка
hello.txt  VBoxLinuxAdditions.run   Загрузки   Общедоступные
```

Если мы хотим вывести на экран только файлы (или каталоги), содержащие в имени «hello», мы можем использовать команду `grep`, фильтрующую строки по выбранному шаблону, передав в неё результат выполнения команды `ls` через конвейер.

Выполним команду `ls | grep "hello"`:

```
localadmin@astra:~$ ls | grep "hello"
hello.txt
```

4.6.15 Работа с буфером обмена в консоли

В терминале и, что важнее, в консоли мы можем работать с буфером обмена. Вот комбинации для работы с буфером:

- Ctrl + W - вырезает слово, стоящее перед курсором (слева) и копирует его в буфер обмена.
- Ctrl + U - вырезает текст с начала строки до положения курсора и помещает его в буфер обмена.
- Ctrl + K - вырезает и копирует в буфер обмена все символы в строке после курсора (справа).
- Ctrl + Y - вставляет текст из буфера обмена перед курсором.

4.6.16 Форматирование текста в консоли

В Linux, в отличие от Windows, регистр имеет большое значение – имена файлов и папок, команды, ключи являются регистрозависимыми. Для упрощения работы с регистром можно использовать комбинации:

- Alt + U - меняет регистр на верхний для всех символов после курсора (справа) до конца текущего слова.
- Alt + L - меняет регистр на нижний для всех символов после курсора (справа) до конца текущего слова.
- Alt + C - меняет регистр символа под курсором на верхний и переводит курсор в конец слова.

4.6.17 Управление отображением информации

- Ctrl + L - очистка терминала от текста, альтернатива команде `clear`, аналог cls в Windows.
- Ctrl + S - приостановка вывода данных в окно терминала программами. Программа при этом продолжает работать.

Примечание: По умолчанию эта функция отключена. Для её включения можно выполнить команду `stty ixon`, однако тогда перестанет работать обратный поиск по истории команд, назначенный на то же сочетание.

- `Ctrl + Q` - отмена приостановки вывода данных в окно терминала (отмена `Ctrl + S`).

4.7 Практика и тестирование

4.7.1 Практическая работа: Модуль 4. Работа с терминалом и оболочкой Bash

См. также *Требования, правила и цели выполнения практической работы*

4.7.1.1 Практические задания

Задание 1.

1. Определите файл tty текущей сессии псевдотерминала.
2. Отправьте эхо на файл псевдотерминала. Для этого откройте новый терминал, отправьте приветствие.
3. Выведите на экран переменные $COLUMNS и $LINES.
4. Запустите `sleep 9000` и через некоторое время нажмите `Ctrl + C` для прерывания команды.
5. Откройте утилиту `mc`, посмотрите иерархию файлов в mc и попробуйте закрыть ее нажатием `Ctrl + C`.
6. Почему `mc` не закрывается через `Ctrl + C`?

Задание 2.

1. Прежде чем начать работать, узнайте, в какой рабочей директории вы находитесь.
2. Измените текущую директорию на etc.

   Что изменилось в строке ввода команд?
3. Проверьте еще раз рабочий каталог.
4. Выведите список всех объектов командой `ls`.

5. Посмотрите на имя хоста, выведите командой `cat` на экран hostname.

   Ответьте на вопрос: изменился ли файл после выполнения `cat`?

   Куда команда `cat` вывела содержимое файла?

   Безопасна ли команда `cat`?

6. Сделайте копию hostname перед изменением перенаправления STDOUT.

   `cat hostname > ~/hostname.old`

   Куда сохранился файл hostname.old?

7. Проверьте, как сохранился бекап:

   `cat < ~/hostname.old`

8. Поменяйте имя хоста с помощью редактора nano:

   `nano /etc/hostname`

   Ответьте на вопрос: почему подчеркивает красным цветом [File „/etc/hostname“ is unwritable]?

   Закройте редактор с помощью Ctrl + X. И если при выходе из nano редактор запросит сохранить изменения, то нужно нажать N и Enter.

9. Повторите команду с повышенными правами.

```
$ sudo !!
```

Комбинация символов **!!** подставляет предыдущую команду

Важно: Выполнять предыдущую команду `sudo !!` можно, только если вы знаете, какая команда была до этого. Иначе можно сломать что-нибудь важное в системе. **Используйте аккуратно, зная, что root может всё.**

---

1. Измените текст на dc-1 (будущее название хоста).

2. Мы написали что-то неправильно и на автомате нажали Ctrl + Z.Привычка Windows отменять введенный текст этим сочетанием.

   Нажмите Ctrl + Z.

   Этим действием мы отправили процесс на паузу управляющей последовательностью Ctrl + Z.

3. Верните процесс из фонового режима командой `fg`.
4. Сохраните нужный нам текст dc-1, нажимая Ctrl + O и Enter.
5. Закройте редактор nano с помощью Ctrl + X.
6. Чтобы сменилось название хоста, выполните перезагрузку.
7. После смены имени хоста будут возникать проблемы с отображением `sudo`, и для этого надо изменить хост в `/etc/hosts`.

Задание 3.

1. Выведите приветствие текущему пользователю. Где можно использовать приветствие и переменную $USER?
2. Посмотрите, какие есть общие глобальные переменные окружения.
3. Выведите все переменные текущей сессии.
4. Найдите с помощью `grep` фильтра в текстовых данных конвейером только LINES или COLUMNS.
5. Попробуйте перезапустить пк в текущем пользователе без root.

Ответьте, почему не получилось, посмотрев на переменную окружения $PATH:

1) Посмотрите, какие бинарные файлы может запускать обычный пользователь.
2) Посмотрите, какие файлы может запускать root пользователь. Для этого выполните вход в сессию root пользователем.

Ответьте на вопрос: Почему пользователь не может найти команду `reboot`, хотя она есть у root пользователя?

4.7.1.2 Ответы на практические задания (пошаговые инструкции)

Задание 1.

1. Определите файл tty текущей сессии псевдотерминала.

```
localadmin@astra:~$ tty
```

2. Отправьте эхо на файл псевдотерминала. Для этого откройте новый терминал, отправьте приветствие.

```
localadmin@astra:~$ echo “Hello from $(tty)!” > /dev/pts/0
```

Как мы видим, первое окно получило сообщение из второго терминала.

3. Выведите на экран переменные $COLUMNS и $LINES.

```
localadmin@astra:~$ echo "$COLUMNS $LINES"
```

4. Запустите `sleep 9000` и через некоторое время нажмите Ctrl + C для прерывания команды.

```
localadmin@astra:~$ sleep 9000
```

5. Откройте утилиту `mc`, посмотрите иерархию файлов в `mc` и попробуйте закрыть ее нажатием Ctrl + C.

```
localadmin@astra:~$ mc
```

6. Почему `mc` не закрывается через Ctrl + C?

Ответ прост: `mc` обрабатывает управляющую последовательность Ctrl + C и для выхода можно нажать F10 или мышкой на кнопку *10 Выход*.

Задание 2.

1. Прежде чем начать работать, узнайте, в какой рабочей директории вы находитесь.

```
localadmin@astra:~$ pwd
/home/localadmin
```

2. Измените текущую директорию на etc.

```
localadmin@astra:~$ cd /etc
```

Что изменилось в строке ввода команд?

```
localadmin@astra:/etc$
                  ^_ рабочий каталог теперь /etc
```

3. Проверьте еще раз рабочий каталог.

```
localadmin@astra:/etc$ pwd
/etc
```

4. Выведите список всех объектов командой `ls`.

```
localadmin@astra:/etc$ ls -l
итого 1240
drwxr-xr-x  3 root root       4096 июн 19 10:40 acpi
```

(continues on next page)

(продолжение с предыдущей страницы)

```
-rw-r--r--  1 root root       2998 июн 19 11:19 adduser.conf
-rw-------  1 root root       5172 июн 19 10:41 afick.conf
-rw-r--r--  1 root root        217 июн 19 10:40 aliases
drwxr-xr-x  3 root root       4096 июн 19 10:38 alsa
drwxr-xr-x  2 root root       4096 сен 17 17:46 alternatives
-rw-r--r--  1 root root        401 мая 19  2019 anacrontab
...
```

5. Посмотрите на имя хоста, выведите командой `cat` на экран hostname.

```
localadmin@astra:/etc$ cat hostname
```

- Ответьте на вопрос: изменился ли файл после выполнения `cat`?
  - Ответ: Файл остался неизменным
- Куда команда `cat` вывела содержимое файла?
  - Ответ: Команда вывела содержимое в поток STDOUT.
- Безопасна ли команда `cat`?
  - Ответ: Да, потому что она только читает файл.

6. Сделайте копию hostname перед изменением перенаправления STDOUT.

```
localadmin@astra:/etc$ cat hostname > ~/hostname.old
```

или

```
localadmin@astra:/etc$ cat hostname 1> ~/hostname.old
```

Куда сохранился файл hostname.old? Ответ: в домашнюю директорию пользователя, так как мы указали путь, начиная с тильды слеш `~/`

7. Проверьте, как сохранился бекап:

```
localadmin@astra:/etc$ cat < ~/hostname.old
```

8. Поменяйте имя хоста с помощью редактора nano:

```
localadmin@astra:/etc$ nano /etc/hostname
```

- Ответьте на вопрос: почему подчеркивает красным цветом [File „/etc/hostname“ is unwritable]? - Не хватает прав пользователю localadmin

Закройте редактор с помощью `Ctrl + X`. И если при выходе из nano редактор запросит сохранить изменения, то нужно нажать `N` и Enter.

9. Повторите команду с повышенными правами.

```
localadmin@astra:~$ sudo !!
```

!! подставляет предыдущую команду

Важно: Выполнять предыдущую команду `sudo !!` можно, только если вы знаете, какая команда была до этого. Иначе можно сломать что-нибудь важное в системе. **Используйте аккуратно, зная, что root может всё.**

---

10. Измените текст на dc-1 (будущее название хоста).
11. Мы написали что-то неправильно и на автомате нажали Ctrl + Z. Привычка Windows отменять введенный текст этим сочетанием.

    Нажмите Ctrl +Z.

    Этим действием мы отправили процесс на паузу управляющей последовательностью Ctrl + Z.
12. Верните процесс из фонового режима командой `fg`.
13. Сохраните нужный нам текст dc-1, нажимая Ctrl + O и Enter.
14. Закройте редактор nano с помощью Ctrl + X.
15. Чтобы сменилось название хоста, выполните перезагрузку.

```
localadmin@astra:/etc$ sudo reboot
```

16. После смены имени хоста будут возникать проблемы с отображением `sudo`, и для этого надо изменить хост в `/etc/hosts`.

```
localadmin@astra:~$ sudo nano /etc/hosts
```

Задание 3.

1. Выведите приветствие текущему пользователю.

```
localadmin@astra:~$ echo "Hello $USER"
Hello localadmin
```

Где можно использовать приветствие и переменную $USER?

Ответ: Можно использовать в скрипте, чтобы получить доступ к папке пользователя

my_dir=»/home/$USER»

2. Посмотрите, какие есть общие глобальные переменные окружения:

```
localadmin@astra:~$ env
```

3. Выведите все переменные текущей сессии.

```
localadmin@astra:~$ (set -o posix;set)
console=/bin/console
console_ALIASES=()
console_ARGC=([0]="0")
console_ARGV=()
console_CMDS=()
console_COMPLETION_VERSINFO=([0]="2" [1]="8")
console_ENV=/home/localadmin/.consolerc
console_LINENO=()
console_SOURCE=()
console_VERSINFO=([0]="5" [1]="0" [2]="3" [3]="1" [4]="release" [5]="x86_64-pc-linux-
↪gnu")
console_VERSION='5.0.3(1)-release'
...
```

4. Найдите с помощью `grep` фильтра в текстовых данных конвейером только LINES или COLUMNS.

```
localadmin@astra:~$ (set -o posix;set) | grep -E "LINES|COLUMNS"
COLUMNS=103
LINES=26
```

5. Попробуйте перезапустить пк в текущем пользователе без root.

```
localadmin@astra:~$ reboot
console: reboot: команда не найдена
```

Ответьте, почему не получилось, посмотрев на переменную окружения $PATH:

1) Посмотрите, какие бинарные файлы может запускать обычный пользователь.

```
localadmin@astra:~$ echo $PATH
/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games
```

2) Посмотрите, какие файлы может запускать root пользователь. Для этого выполните вход в сессию root пользователем:

```
localadmin@astra:/etc$ sudo -i
```

Теперь у нас root сессия, и мы выведем значение PATH.

```
root@astra:/etc# echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
```

- Ответьте на вопрос: Почему пользователь не может найти команду `reboot` , хотя она есть у root пользователя?
    - Ответ: Потому что пользователь не может выполнять команды из системных директорий `/sbin` и `/usr/sbin` .

4.7.2 Тест: Модуль 4. Работа с терминалом и оболочкой Bash

Пожалуйста, ответьте на несколько вопросов, чтобы закрепить полученные знания. Правильные ответы для самопроверки вы сможете найти в конце списка. Обратите внимание, что у некоторых вопросов может быть несколько верных вариантов ответов.

4.7.2.1 Вопросы

Вопрос 1. Как называется устройство, которое может отправлять команды ЭВМ и выводит на экран полученный результат?

a. Line printer

b. Тикерный аппарат

c. Виртуальный терминал

d. Телетайп

Вопрос 2. Какая папка отвечает за конфигурационные файлы?

a. Папка `/boot`

b. Папка `/root`

c. Папка `/etc`

d. Папка `/proc`

Вопрос 3. Какая управляющая последовательность завершает операцию?

a. Горячее сочетание `Ctrl + C`

b. Горячее сочетание `Ctrl + Z`

c. Горячее сочетание `Ctrl + D`

d. Клавиша `Esc`

Вопрос 4. Какая команда выводит список файлов и каталогов текущей директории?

a. Команда `ls`

b. Команда `cd`

c. Команда `cat test1`

d. Команда `df`

Вопрос 5. Какими текстовыми редакторами можно редактировать файл?

a. nano

b. find

c. vi

d. less

Вопрос 6. Какой командой можно получить справку на любую команду?

a. Команда `which имя_команды`

b. Команда `file имя_команды`

c. Команда `stat имя_команды`

d. Команда `man имя_команды`

Вопрос 7. Какой командой можно перенаправить стандартный вывод в файл hosts.bak?

a. Команда `sudo cat /etc/hosts > /etc/hosts.bak`

b. Команда `cat /etc/hosts | grep bak`

c. Команда `nano /etc/hosts.bak`

d. Команда `cp /etc/hosts hosts.bak`

Вопрос 8. В какой переменной хранится список каталогов для запуска исполняемых файлов?

a. Переменная `$HOME`

b. Переменная `$USER`

c. Переменная `$LANG`

d. Переменная `$PATH`

Вопрос 9. Какой поток данных передается по конвейеру?

a. STDOUT

b. STDIN

c. STDERR

Вопрос 10. Какая команда отображает историю команд?

a. Команда `history`

b. Команда `man`

c. Команда `type history`

d. Команда `which help`

Вопрос 11. Какой файл содержит профиль текущего пользователя?

a. Файл `/etc/profile`

b. Файл `/home/$USER/.bashrc`

c. Файл `/home/$USER/.bash_rc`

d. Файл `/home/$USER/.bash_history`

Ответы на вопросы

1. c
2. c
3. a
4. a
5. a, c
6. d
7. a
8. d
9. a
10. a
11. b

4.8 Заключение

В этом модуле вы узнали:

- в чем различие между терминалом, консолью и виртуальной консолью
- как устроен и работает терминал в Linux
- что такое оболочка Bash

В этом модуле вы научились:

- работать в терминале Linux
- работать в оболочке Bash

В следующем модуле будут рассмотрены вопросы работы со справочными системами в Astra Linux.

4.9 Дополнительные источники информации

- О режиме POSIX написано на сайте Bash-POSIX-Mode.html
- Варианты запуска bash 6.2 Bash Startup Files

4.10 Обратная связь

Если остались вопросы, то их всегда можно задать в специальной теме.

Модуль 5. Работа со справочными системами

5.1 Введение

Из этого модуля мы узнаем о том, какие справочные системы есть в Linux и как ими эффективно пользоваться, чтобы не приходилось мучительно вспоминать то, что не знал, а потом еще и забыл.

5.2 Справочные системы Linux

В Linux тот, кто просит помощи, всегда ее получает, и для этого даже не обязательно иметь выход в Интернет — достаточно воспользоваться встроенной справкой, чтобы получить всю необходимую информацию о любой команде системы: ее описание, синтаксис, входные и выходные данные, настройки и примеры использования.

Основных справочных систем в Linux две:

- **manpages** — появилась еще на заре создания Unix и до сих пор остается стандартом де-факто для всех *nix-подобных систем;
- **info** — справочная система на гиперссылках от проекта GNU.

В дополнение к ним есть еще `help` — справочник по встроенным командам оболочки Bash и собственная справка Astra Linux по работе с графическим интерфейсом, но об этом позже.

Следует отметить, что в Astra Linux многие справочные страницы переведены на русский язык и существенно доработаны, в них уточнены важные моменты и добавлены полезные примеры. Далее мы рассмотрим каждую из упомянутых справочных систем и несколько команд, которые очень помогают в работе с ними.

5.3 Справочник по встроенным командам оболочки Bash

Знакомство начнем со справки по встроенным командам оболочки Bash. Если в терминале мы введем команду `help`, то увидим список внутренних команд Bash (см. рис. 1.19). В Windows cmd есть похожая справка, но в Linux она намного полнее.

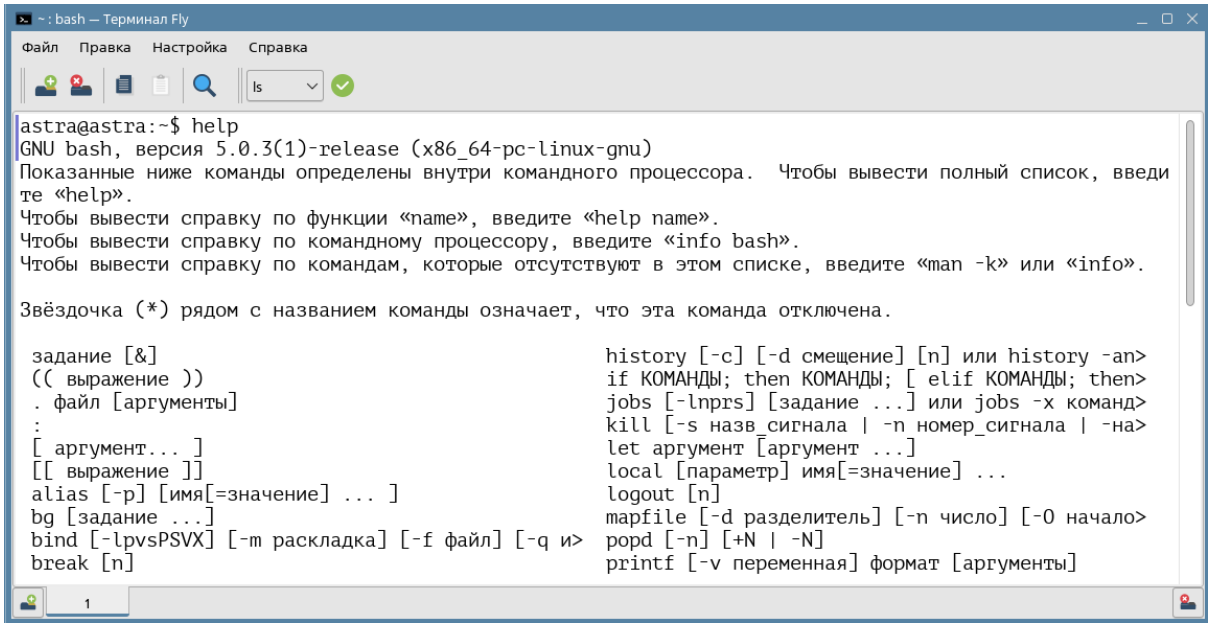


рис. 1.19 – Список внутренних команд Bash

Можно обратить внимание, что в первом же параграфе есть указание на другие справочные системы, доступ к которым можно получить с помощью утилит `man` и `info`.

Введем команду `help help`, чтобы получить через утилиту `help` справку о самой себе, см. рис. 1.20:

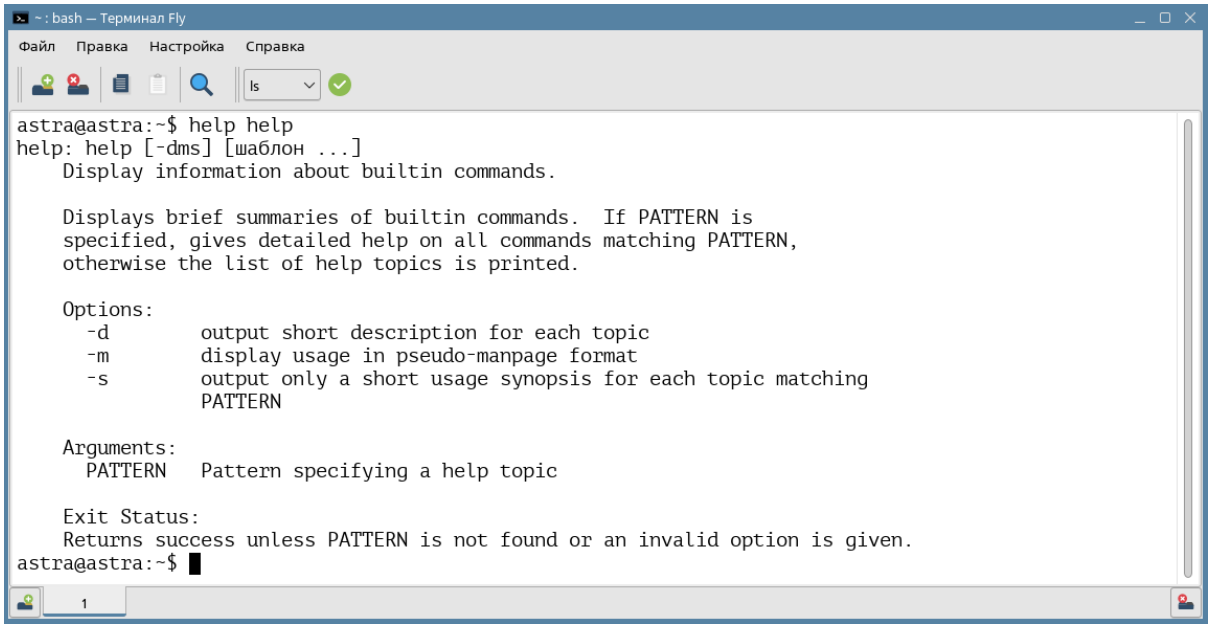


рис. 1.20 – Справка по команде help

Как видите, ключей у команды `help` всего три, см. рис. 1.21:

- `-d` — выводит короткое описание команды;

- `-m` — выводит справку в псевдо-manpage формате;
- `-s` — выводит список ключей в кратком формате без их описания.

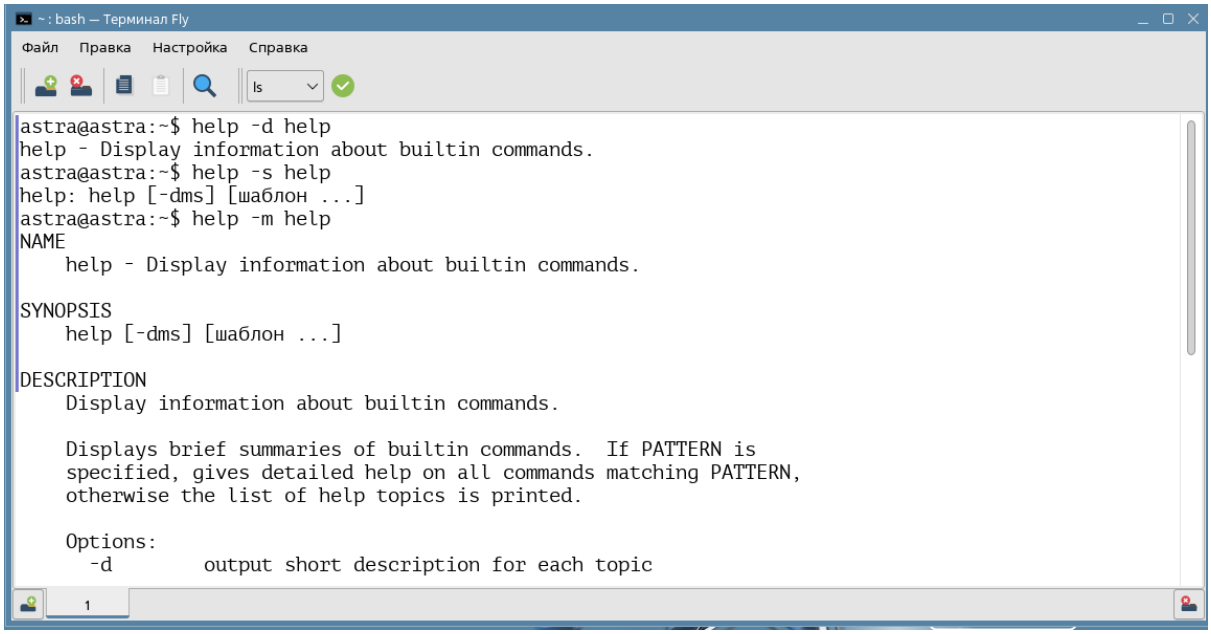


рис. 1.21 – Вывод информации по ключам команды help

Можем посмотреть справку на команду `help enable`, с помощью которой мы отключали встроенную команду `enable -n echo`. Как вы помните, после этого оболочка переключается на использование одноименной утилиты, которая работает схожим образом. Чтобы включить `echo` обратно, команду `enable` нужно вызвать без ключа `enable echo`, а вызов команды вообще без параметров отобразит полный список всех активных команд оболочки.

5.4 Справочные страницы системы manpages

Уже через пару лет после выхода Unix в ней появились первые справочные страницы, хотя формат их в корне отличался от современной справки. Срочно нужна была справочная система, которая позволила бы динамически обновлять свое содержимое и выполнять по нему централизованный поиск. Такой системой стала manpages, название которой образовано от двух слов manual pages, то есть «страницы документации».

В системе manpages содержится справка по отдельным командам, специальным файлам устройств, конфигурационным файлам, системным и библиотечным вызовам и другим аспектам работы системы, которые представлены обычно в восьми типовых разделах.

5.4.1 Расположение справочных страниц

В современной версии manpages справочные страницы написаны на языке разметки nroff и представляют собой файлы, сжатые gzip. Справочные страницы на английском языке хранятся в каталогах `/usr/share/man/man<N>`, где <N> – номер раздела справки. Локализованные страницы находятся в каталогах `/usr/share/man/<locale>/man<N>`, где <locale> – код языка. Например, полный путь до сжатого файла со страницей справки о самой утилите `man` можно узнать с помощью команды `man -w man`:

```
localadmin@astra:~$ man -w man
/usr/share/man/ru/man1/man.1.gz
```

5.4.2 Просмотр справочных страниц

Страницы manpages мы можем просматривать с помощью приложения `man` (входит в пакет man-db), которое использует несколько вспомогательных утилит. Сначала утилита `man` выполняет поиск запрашиваемой страницы, затем она распаковывает соответствующий файл при помощи `gzip`, форматирует текст в разметке roff или groff с помощью процессора `troff` и, наконец, выводит результат в окно терминала с помощью утилиты `less`.

Процессор `troff` позволяет адаптировать текст под любую ширину экрана и выполнять подсветку синтаксиса, используя так называемые управляющие последовательности терминала. Утилита `less` позволяет листать длинные страницы справки и выполнять поиск по тексту.

5.4.3 Навигация по справочной странице

Рассмотрим возможности утилиты `less` в части навигации по странице. Введем в терминале команду `man man`, чтобы вызвать справку о самой команде `man`, см. рис. 1.22:

```
localadmin@astra:~$ man man
```

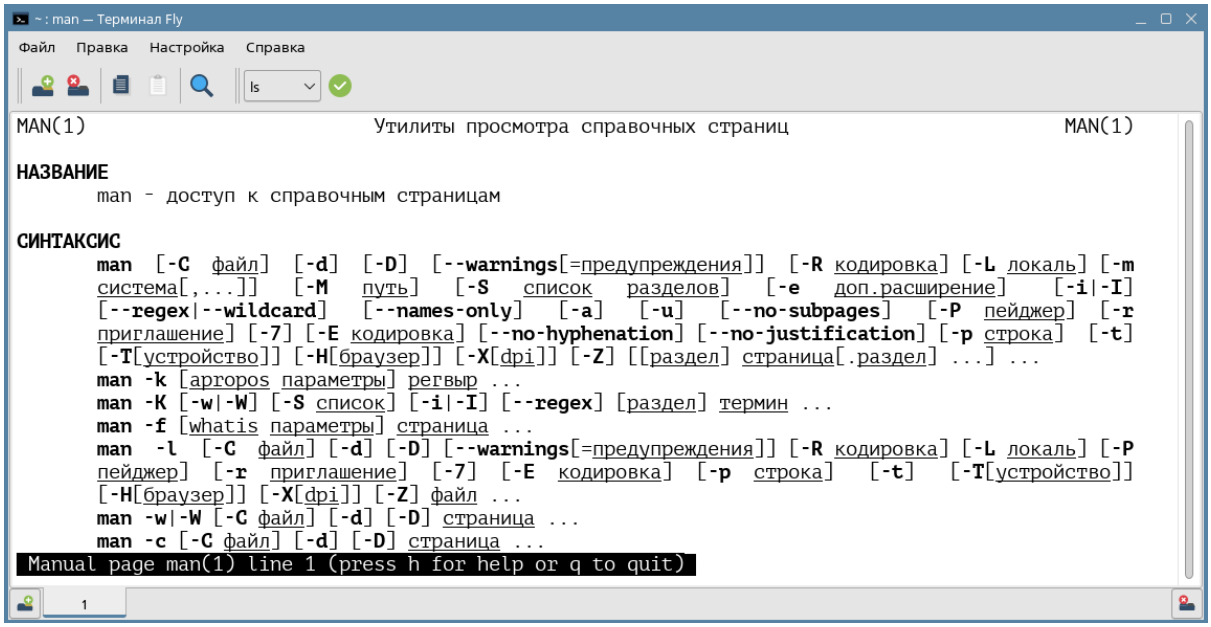


рис. 1.22 – Открытый просмотр less справочной страницы о команде man

Перед нами страница справки об утилите man из первого раздела документации, на что указывает метка MAN(1) в левом верхнем углу страницы. Просмотр страницы, как упоминалось ранее, осуществляется с помощью утилиты less (с англ. меньше), которая позволяет «прокручивать» текст вверх/вниз и выполнять поиск.

Рассмотрим несколько приемов работы с программой less :

- Построчное перемещение:
  - Вниз — колесо мыши вниз, клавиши <стрелка вниз>, Enter.
  - Вверх — колесо мыши вверх, клавиша <стрелка вверх>.
- Постраничное перемещение:
  - Вниз — клавиши <пробел>, PgDown.
  - Вверх — клавиша PgUp.
- Переход в начало страницы — клавиша Home.
- Переход в конец страницы — клавиша End.
- Подробная справка по командам — клавиша h.
- Выход — клавиша q.

Для детального изучения справочной страницы man(1) потребуется слишком много времени, по-

этому остановимся только на самых важных моментах и основных приемах работы.

Примечание: Во многих инструкциях вы будете сталкиваться с отсылками на справочные страницы в формате login(1) или login(5), где цифры в скобках указывают на раздел документации, в котором следует искать необходимую информацию. Утилита `login`, конечно же, остается прежней, но вот информация о ней в этих разделах может существенно различаться как по глубине, так и по тематической направленности.

5.4.4 Разделы справочных страниц

В manpages используют восемь основных и один нестандартный раздел справочных страниц:

1. Основные команды (General commands).
2. Системные вызовы (System calls).
3. Функции библиотек (Library functions).
4. Специальные файлы (Special files), например, устройства из `/dev`.
5. Форматы файлов и соглашения (File formats and conventions), например, *passwd, cron table.*
6. Игры (Games).
7. Разное (Miscellaneous).
8. Команды администрирования системы (System administration).
9. Процедуры ядра (Kernel Routines) *[нестандартный раздел].*

Для каждого раздела справки есть заголовочная страница intro, описывающая его назначение.

Для просмотра этой страницы необходимо ввести команду `man <N> intro`, где <N> – указывает желаемый раздел справки (см. выше `/usr/share/man/man<N>`).

Например, чтобы вывести заголовочную страницу для первого раздела, нужно ввести команду, см. рис. 1.23:

```
localadmin@astra:~$ man intro
```

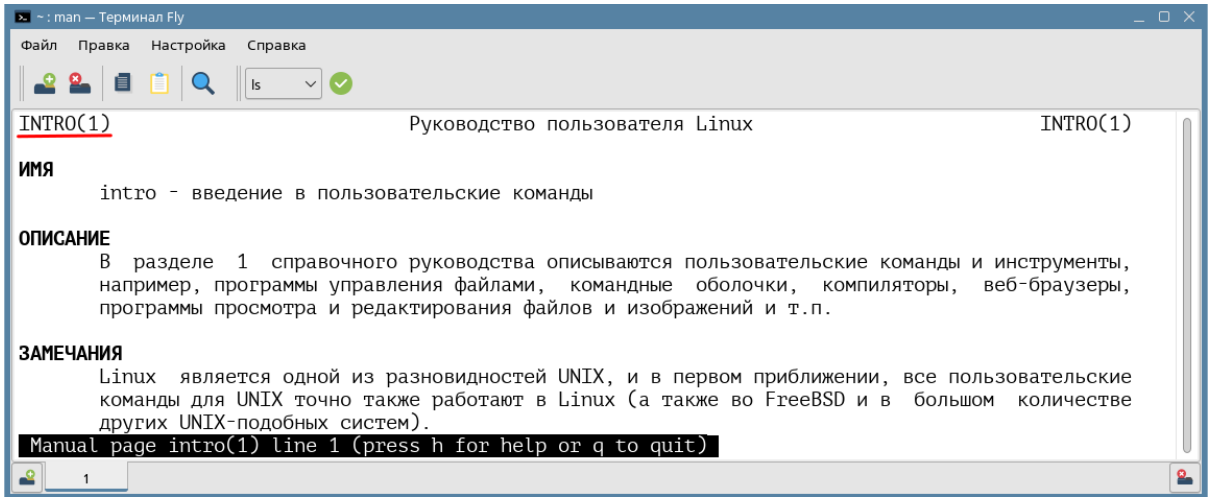


рис. 1.23 – Intro страница раздела справки по команде man

Для вывода справки по четвертому разделу необходимо использовать команду:

```
localadmin@astra:~$ man 4 intro
```

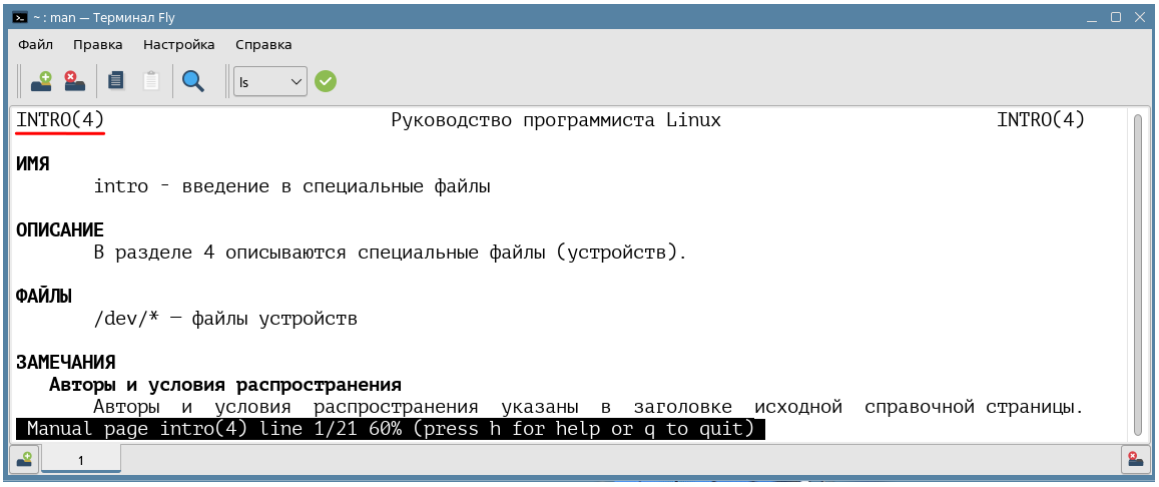


рис. 1.24 – Введение в специальные файлы раздел 4

В справке на утилиту `man` указан порядок поиска информации по разделам (1 n l 8 3 2 3posix 3pm 3perl 3am 5 4 9 6 7), что следует учитывать, вызывая утилиту без указания конкретного раздела.

5.4.5 Структура справочных страниц

При составлении справочных страниц стараются придерживаться общих правил, поэтому большинство страниц содержит следующие секции:

- **Название (Name)** — наименование команды, о которой рассказывается на этой справочной странице.

- **Синопсис (Synopsis)** — краткое описание команды и ее синтаксиса.
- **Конфигурация (Configuration)** — эта секция присутствует на страницах о специальных файлах (4), так как описывает конфигурацию устройств.
- **Описание (Description)** — описание основного назначения программы.
- **Опции (Ключи)** — опции, которые принимает команда.
- **Примеры (Examples)** — один и более примеров использования команды.
- **Авторы (Authors)** — информация о разработчиках, кто участвовал в создании утилиты и занимается ее дальнейшим развитием.

5.4.6 Использование утилиты man и связанных с ней утилит apropos, whatis, mandb

5.4.6.1 Просмотр справочных страниц по командам, конфигурационным файлам, службам

Для получения справочной информации по конкретной команде необходимо вызвать `man <команда>`, например, `man whereis`:

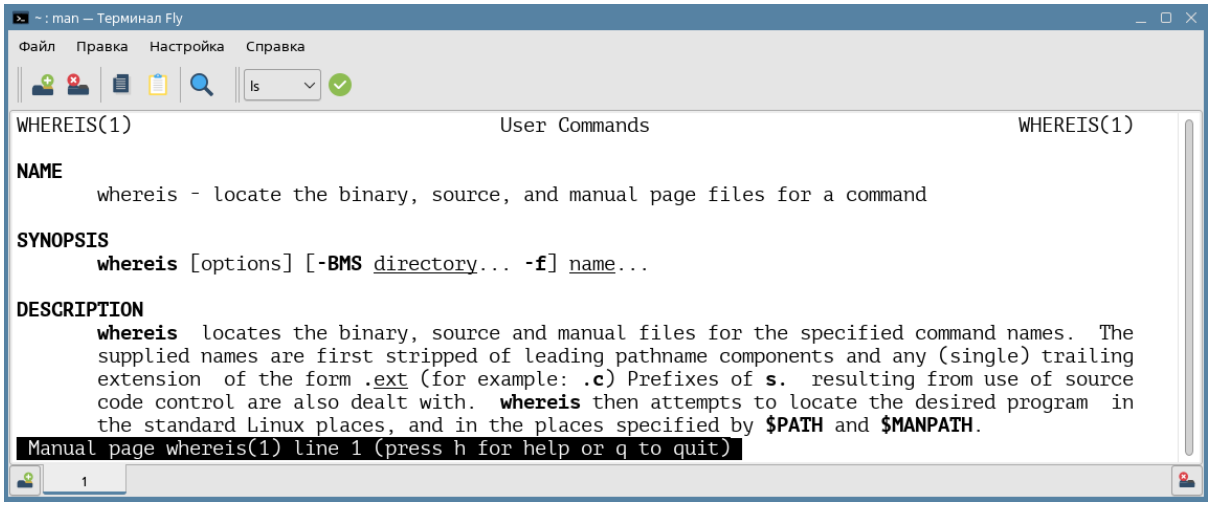


рис. 1.25 – Справочные страницы по команде whereis

Справочные страницы есть в том числе для большинства утилит и конфигурационных файлов системы, поэтому вы можете подставить вместо наименования команды относительный или полный путь к конфигурационному файлу, например, `man hosts`:

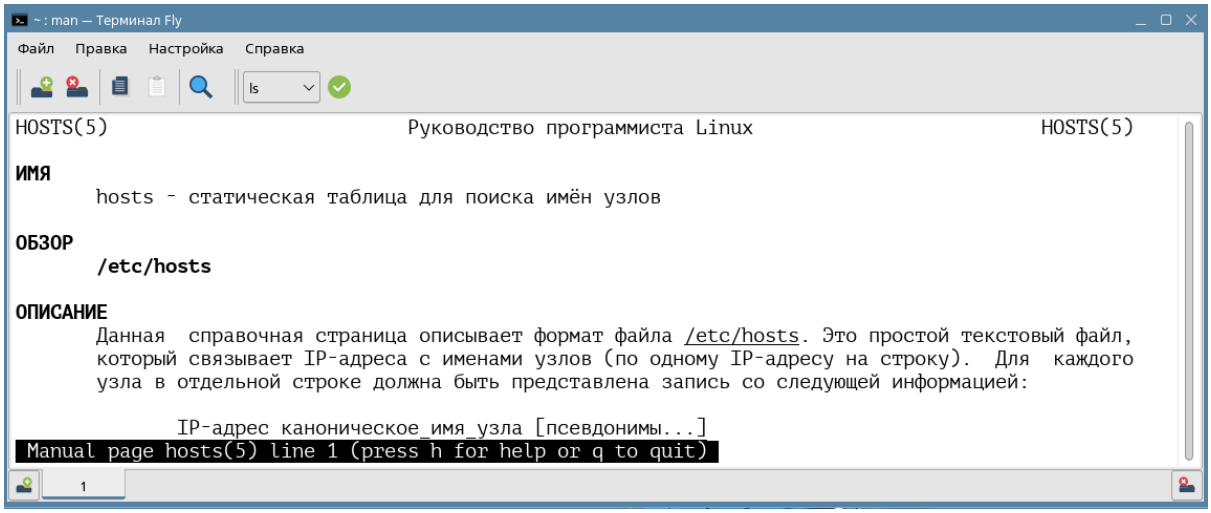


рис. 1.26 – Справочные страницы конфигурационного файла hosts

Пролистав до конца клавишей End, можно увидеть секцию «СМОТРИТЕ ТАКЖЕ», в которой описаны дополнительные ссылки на похожие темы и разделы справки, например конфигурационный файл resolv.conf описан в разделе 5

```
...
СМОТРИТЕ ТАКЖЕ
    hostname(1), resolver(3), host.conf(5), resolv.conf(5),
    resolver(5), hostname(7), named(8)

    Internet RFC 952
```

Откроем раздел 5 resolv.conf командой:

```
localadmin@astra:~$ man 5 resolv.conf
```

Точно так же вы можете получать справку на службы (которые в Linux традиционно называют демонами), например, man sysklogd:

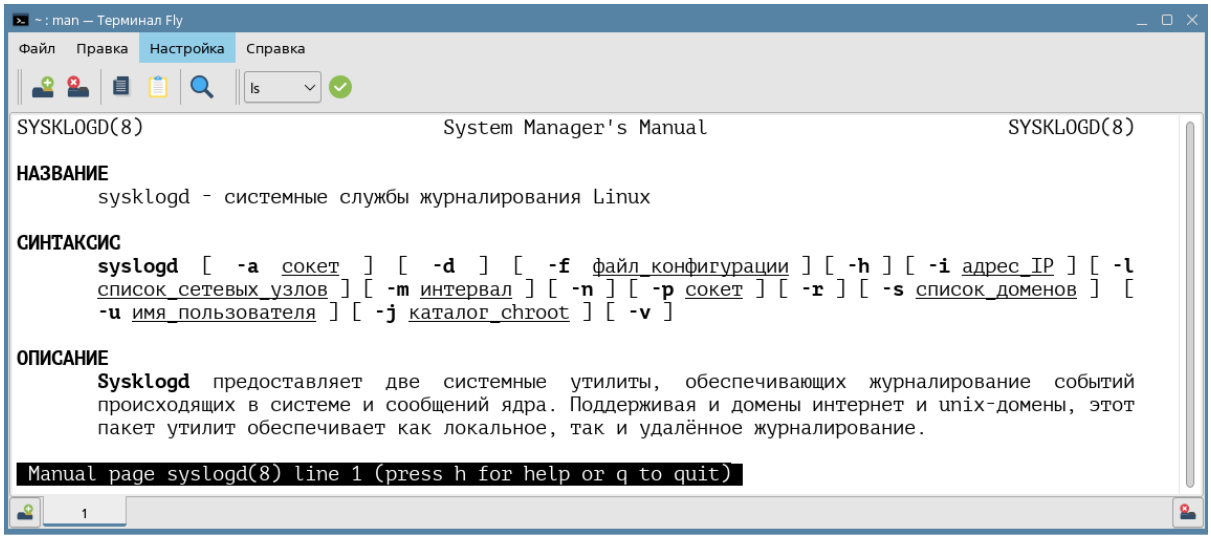


рис. 1.27 – Справочные страницы на службу sysklogd

5.4.6.2 Поиск справочных страниц

Без указания раздела утилита `man` выполняет поиск в том порядке, который определен по умолчанию. Чтобы получить информацию по конкретному разделу, вы можете указать его явно. Но как узнать, в каких разделах может быть необходимая вам информация, чтобы не перебирать их по очереди? Для этого есть команды `man -f <строка поиска>` и `whatis <строка поиска>` — они выведут на экран список справочных страниц из всех разделов, имена которых совпадают с введенной строкой, например, `man -f passwd`:

```
localadmin@astra:~$ man -f passwd
passwd (1)           - изменяет пароль пользователя
passwd (5)           - файл паролей
passwd (1ssl)        - compute password hashes
```

Но вы можете искать не только по заголовкам, но и по описанию страниц. Воспользуйтесь для этого командой `man -k <регулярное выражение>` или `apropos <регулярное выражение>`, например, `man -k passwd`:

```
localadmin@astra:~$ man -k passwd
chpasswd (8)         - обновляет пароли в пакетном режиме
gpasswd (1)          - управление /etc/group и /etc/gshadow
passwd (1)           - изменяет пароль пользователя
passwd (5)           - файл паролей
passwd2des (3)       - шифрование пароля RFS
update-passwd (8)    - безопасное обновление файлов /etc/passwd, /etc/shadow ...
chgpasswd (8)        - update group passwords in batch mode
exim4_passwd (5)     - Files in use by the Debian exim4 packages
exim4_passwd_client (5) - Files in use by the Debian exim4 packages
expect_mkpasswd (1)  - generate new password, optionally apply it to a user
fgetpwent_r (3)      - get passwd file entry reentrantly
getpwent_r (3)       - get passwd file entry reentrantly
grub-mkpasswd-pbkdf2 (1) - generate hashed password for GRUB
```

(continues on next page)

(продолжение с предыдущей страницы)

```
openssl-passwd (1ssl) - compute password hashes
pam_localuser (8)     - require users to be listed in /etc/passwd
passwd (1ssl)         - compute password hashes
smbpasswd (5)         - The Samba encrypted password file
smbpasswd (8)         - change a user's SMB password
```

5.4.6.3 Индексация базы справочных страниц

Для того чтобы поиск по справке работал максимально быстро, утилиты обращаются не к файлам /usr/share/man напрямую, а к встроенной базе данных index, файлы которой размещаются в папке /var/cache/man. В большинстве случаев база данных справочной системы man в Linux обновляется автоматически при установке новых пакетов или при обновлении существующих. Однако есть ситуации, когда может потребоваться ручное обновление:

- **Добавление новых страниц man**: Если вы добавили свои собственные страницы man в специальный каталог, вам может потребоваться вручную обновить базу данных.
- **Изменение существующих страниц**: Если вы изменили содержимое страниц man в системе, желательно обновить их, чтобы изменения отразились.

Обновить вручную базу данных страниц man можно с помощью команды `mandb`:

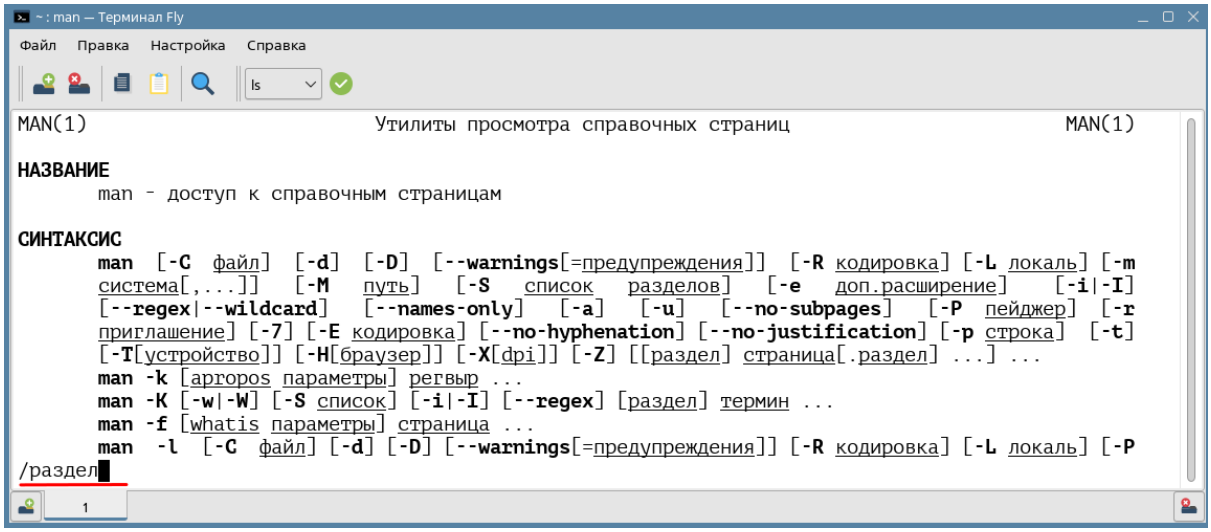
```
localadmin@astra:~$ mandb
Обработка справочных страниц в /home/astra/.local/share/man...
Обновление индексного кэша для пути `/home/astra/.local/share/man/man1'. Ждите...
↪завершено.
Проверка побочных cat в /home/astra/.local/share/man...
Обработка справочных страниц в /home/astra/.local/share/man/cat1...
В 1 man-подкаталоге содержатся более новые справочные страницы.
Добавлена 1 справочная страница.
Добавлено 0 побочных cat-страниц.
Вычищено 0 старых записей базы данных.
```

5.4.7 Поиск информации на справочной странице

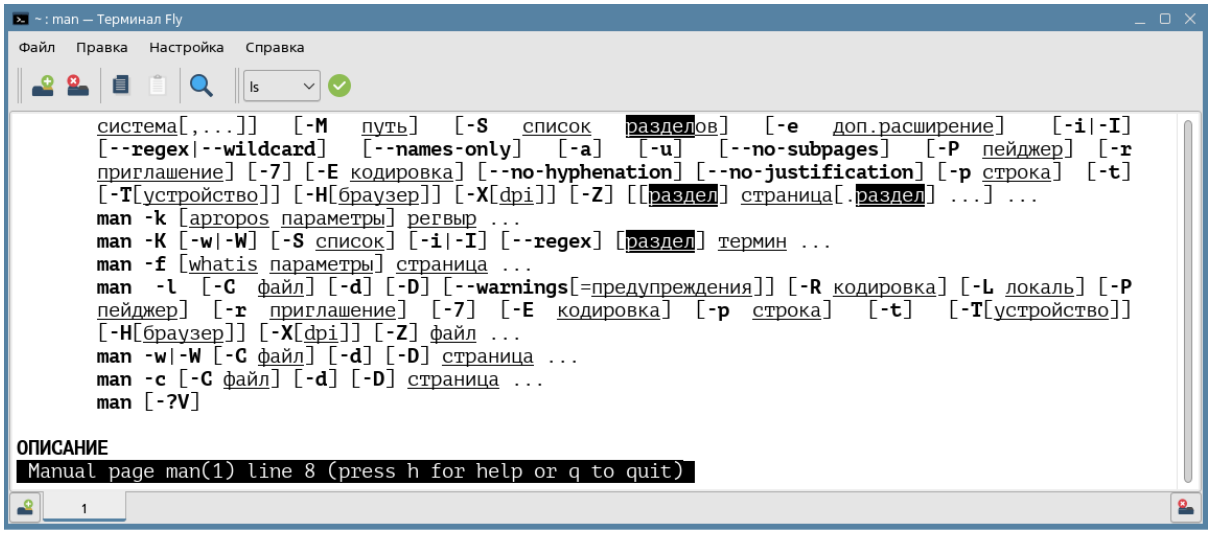
Интерфейс утилиты `less` позволяет выполнять поиск по тексту справочной страницы, предлагая два способа: обычный поиск и поиск с отображением совпадающих строк.

5.4.7.1 Обычный поиск

Откроем еще раз справку `man man`, затем введем символ косой черты `</>` и наберем слово, которое мы хотим найти, например, `/раздел`:



Далее нажмем клавишу Enter, чтобы выполнить переход на строку, содержащую первое вхождение этого слова:



Клавишами n и N можно переходить вперед/назад по вхождениям соответственно.

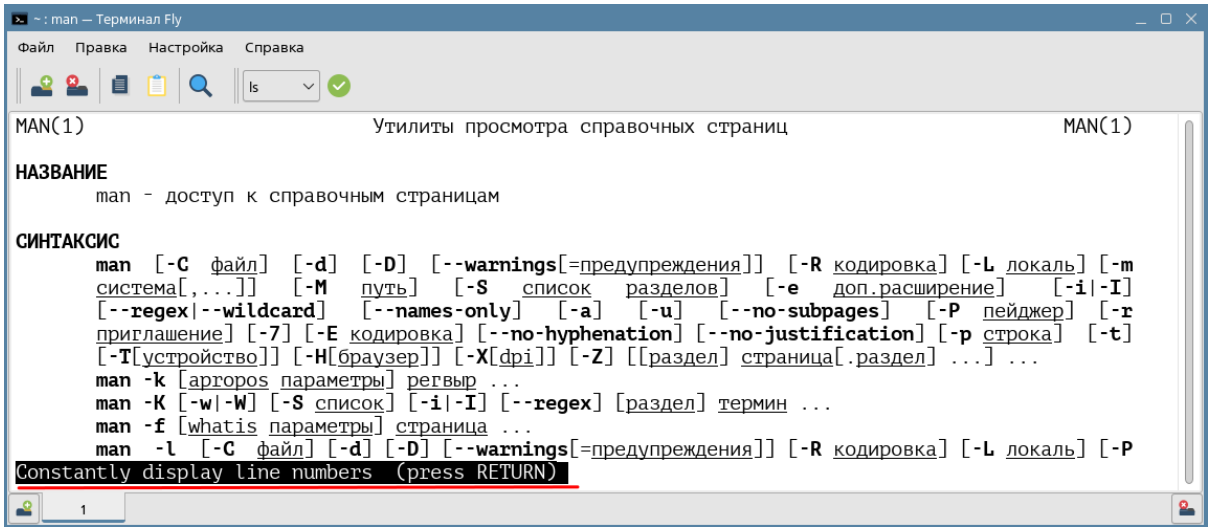
Все вхождения этого слова будут выделены цветом, и, чтобы включить/отключить подсветку найденных вхождений, можно воспользоваться сочетанием клавиш Esc + U.

Для выполнения поиска в обратном направлении (если вы находитесь в конце страницы) вместо символа косой черты используйте символ вопросительного знака <?>, например, ?страница.

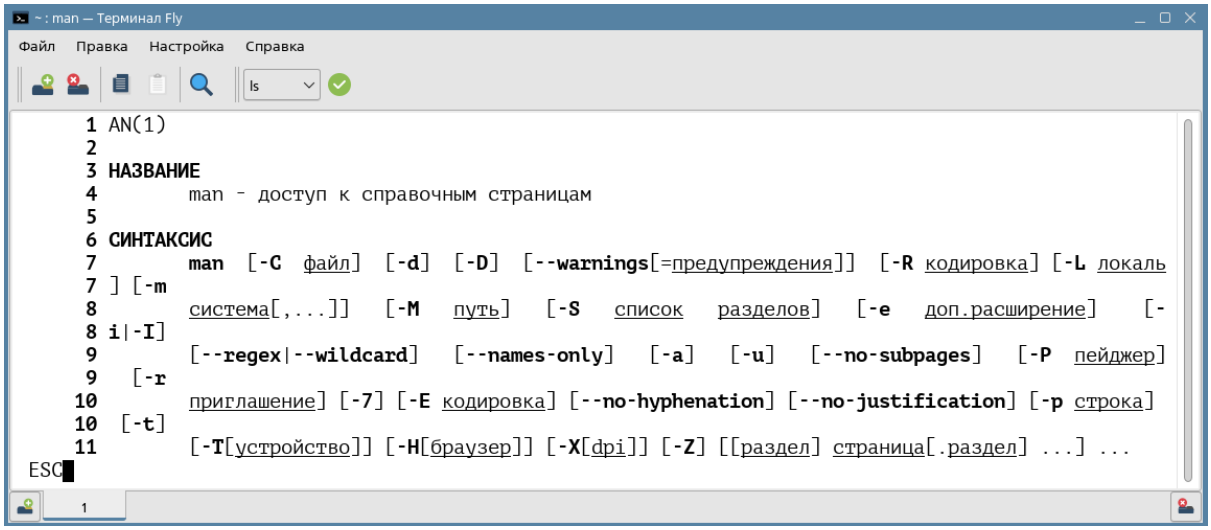
5.4.7.2 Поиск с отображением совпадающих строк

Для быстрого ознакомления с результатами поиска вы можете вывести все вхождения на одной странице. При использовании этого способа рекомендуется включить нумерацию строк.

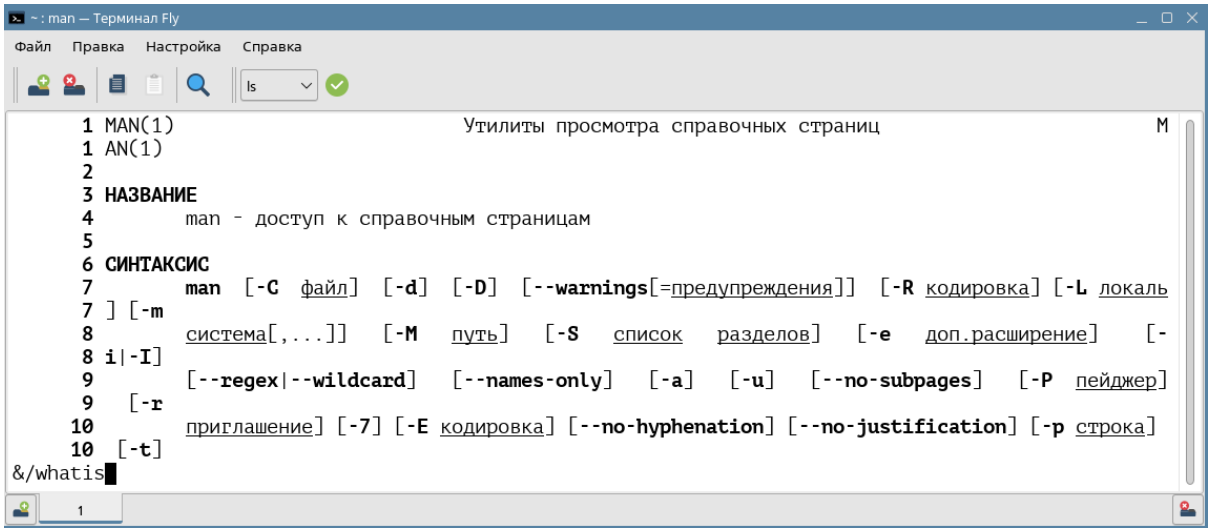
Для этого введите `-N` команда сменится на надпись « `Constantly display line numbers (press RETURN)` » и затем нажмите Enter:



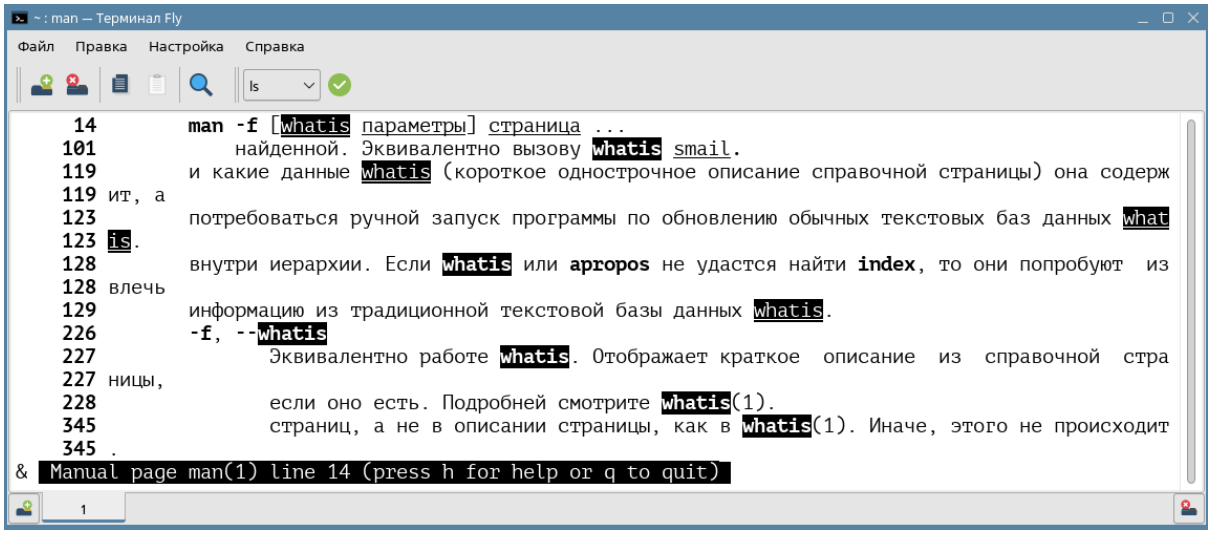
В интерфейсе less появится нумерация строк.



Теперь введем `<&>` (символ «/» появится автоматически), наберем искомое слово whatis и нажмем клавишу Enter:

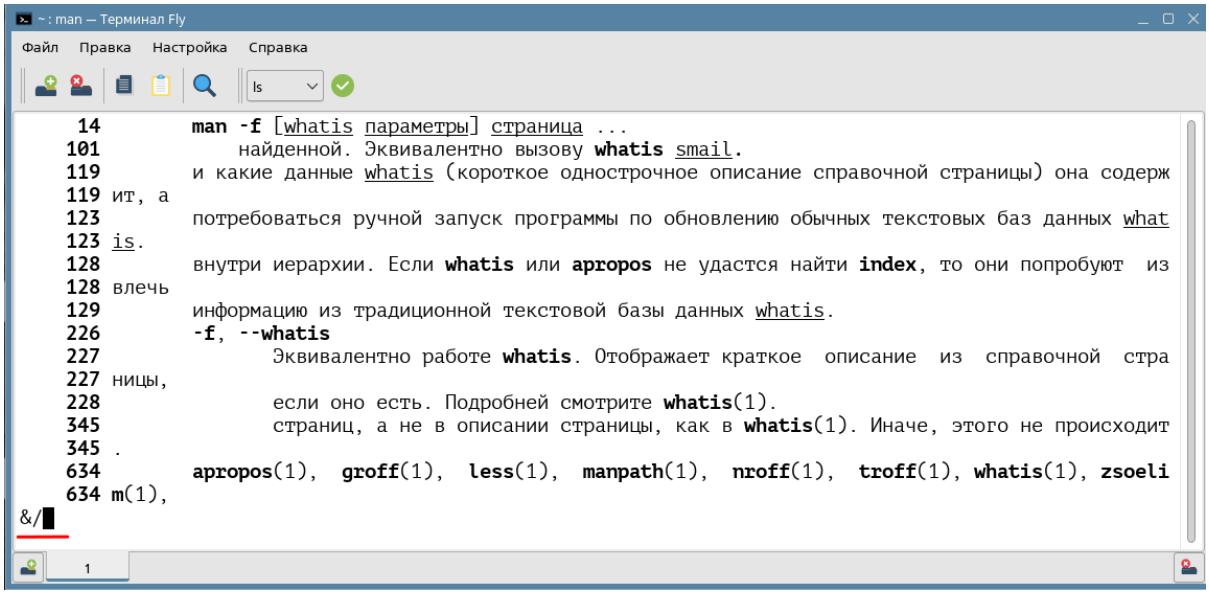


На экране отобразятся только те строки, в которых найдено это слово.

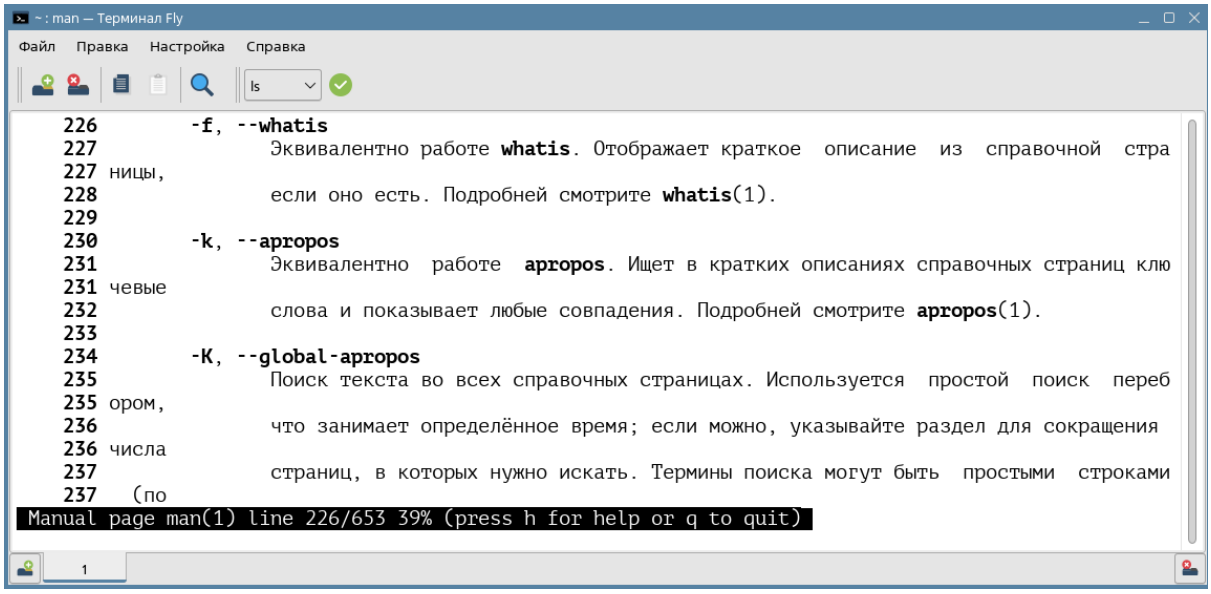


Если вас заинтересует конкретная строка, вы сможете перейти к ней по ее номеру, например 226.

Для этого сначала верните отображение всех строк командой (&) затем (Enter):



Затем введите номер строки 226 и нажмите клавишу g, чтобы выполнить переход.



Для отключения режима нумерации строк введите команду -N и нажмите клавишу Enter.

5.4.8 Использование утилит поиска whereis и which

Утилита whereis позволяет найти расположение файлов на диске. Приложение выполняет поиск точно так же, как это делает оболочка командной строки, т.е. просматриваются все каталоги, перечисленные в переменной PATH. Чтобы увидеть полный список этих каталогов, вы можете вызвать команду whereis -l (строчная L):

```
localadmin@astra:~$ whereis -l
bin: /usr/bin
bin: /usr/sbin
bin: /usr/lib/x86_64-linux-gnu
bin: /usr/lib
```

Команда `whereis <имя команды>` ищет все файлы, относящиеся к запрашиваемой команде:

```
localadmin@astra:~$ whereis ls
ls: /usr/bin/ls /usr/share/man/man1/ls.1.gz
```

Команда с ключом `whereis -b <имя команды>` ищет только двоичные файлы:

```
localadmin@astra:~$ whereis -b ls
ls: /usr/bin/ls
```

Команда с ключом `whereis -m <имя команды>` ищет только по документации:

```
localadmin@astra:~$ whereis -m ls
ls: /usr/share/man/man1/ls.1.gz
```

Утилита `which` отображает полный путь к указанным командам:

```
localadmin@astra:~$ which man
/usr/bin/man
```

По умолчанию команда выводит только первое из найденных значений, но при использовании ключа `which -a man` можно получить весь список:

```
localadmin@astra:~$ which -a man
/usr/bin/man
/bin/man
```

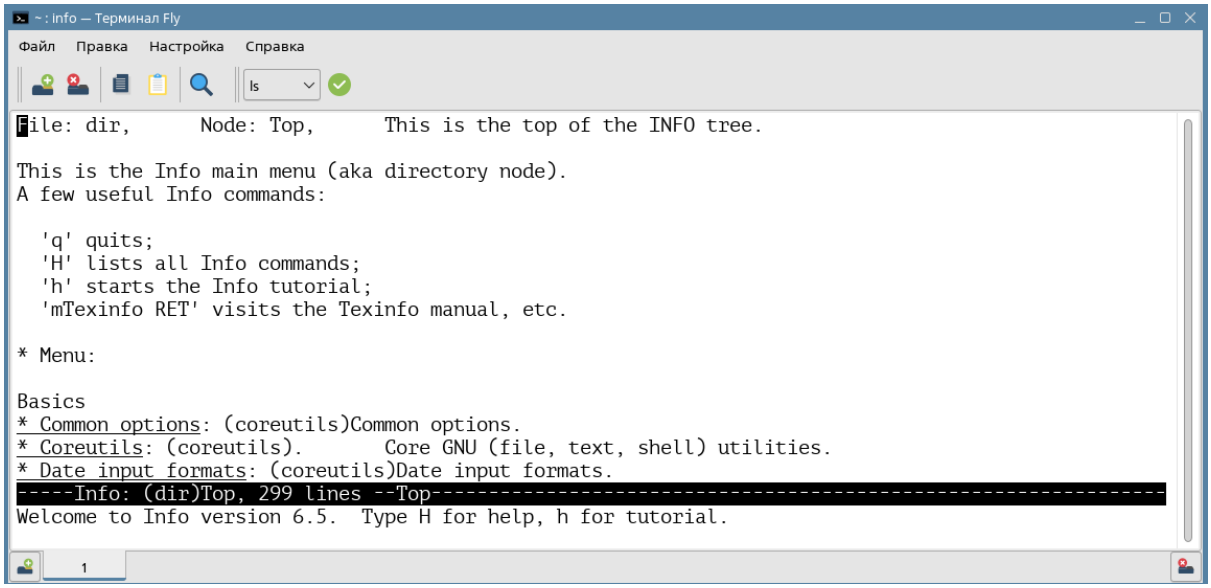
5.5 Справочные страницы системы info

Еще одной системой документации является Texinfo, которая, в отличие от `man`, позволяет создавать большие развернутые руководства.

Язык разметки texinfo позволяет форматировать текст, создавать ссылки, таблицы и т.д. На основе этих исходных файлов можно генерировать справочные документы в различных форматах, поэтому команды разметки имеют инструкции условного включения текста для каждого из выходных форматов.

Утилита `info` работает с набором предварительно сгенерированных гипертекстовых страниц из каталога `/usr/share/info`. Каждая страница имеет иерархическую структуру, состоящую из так называемых «нод», которые можно сравнить с разделами, подразделами, главами или секциями в книгах.

Команда `info` выполнит запуск справочной системы, а команда `info <команда>`, например, `info cat`, выведет справочную страницу по этой команде. Внутрь texinfo встроена справка по командам, вывести на экран ее можно нажатием клавиши <**H**>. Подробнее о системе вы можете узнать в справке `man info`.



5.6 Справка Astra Linux

В графическом интерфейсе Astra Linux доступна еще одна справочная система, открыть которую можно комбинацией клавиш Alt + F1. В этот момент запускается приложение assistant, которому на вход передается путь к файлу `flyhelp.qhc`:

```
assistant -collectionFile /usr/share/doc/fly/qthelp/ru/flyhelp.qhc
```

В этой справке достаточно подробно изложено описание графического интерфейса, утилит, настроек и программ, входящих в комплект поставки дистрибутива. Если у вас возникли вопросы по использованию рабочего стола, панели управления, менеджера устройств и других функций, вы можете обратиться к этой справочной системе.

Справка располагает интуитивно понятным интерфейсом:

- в левой колонке есть четыре вкладки «содержание», «указатель», «закладки» и «поиск»;

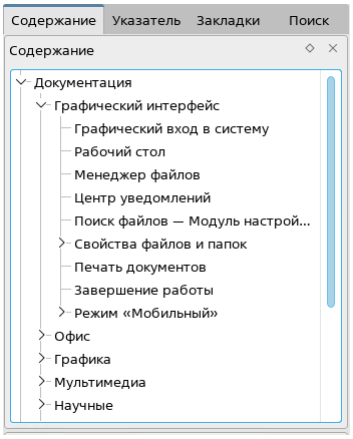


рис. 1.28 – Содержание справки

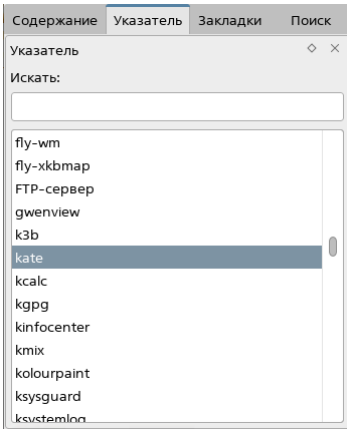


рис. 1.29 – Вкладка указатель на разделы справки

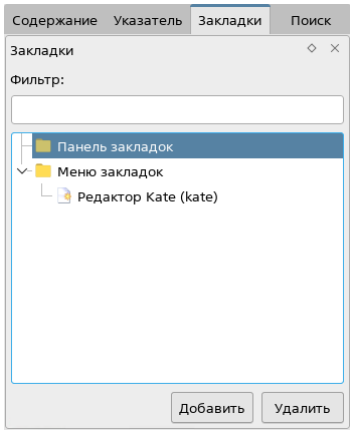


рис. 1.30 – Панель закладок, для сохранения страниц в закладки

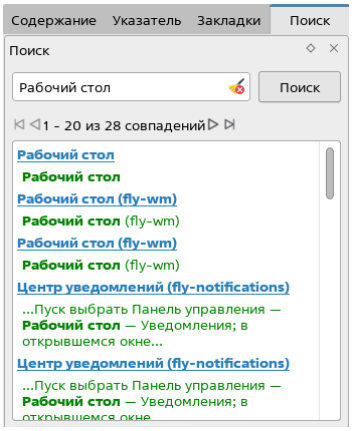


рис. 1.31 – Вкладка поиска по разделам справки

- чуть ниже расположен список открытых страниц;

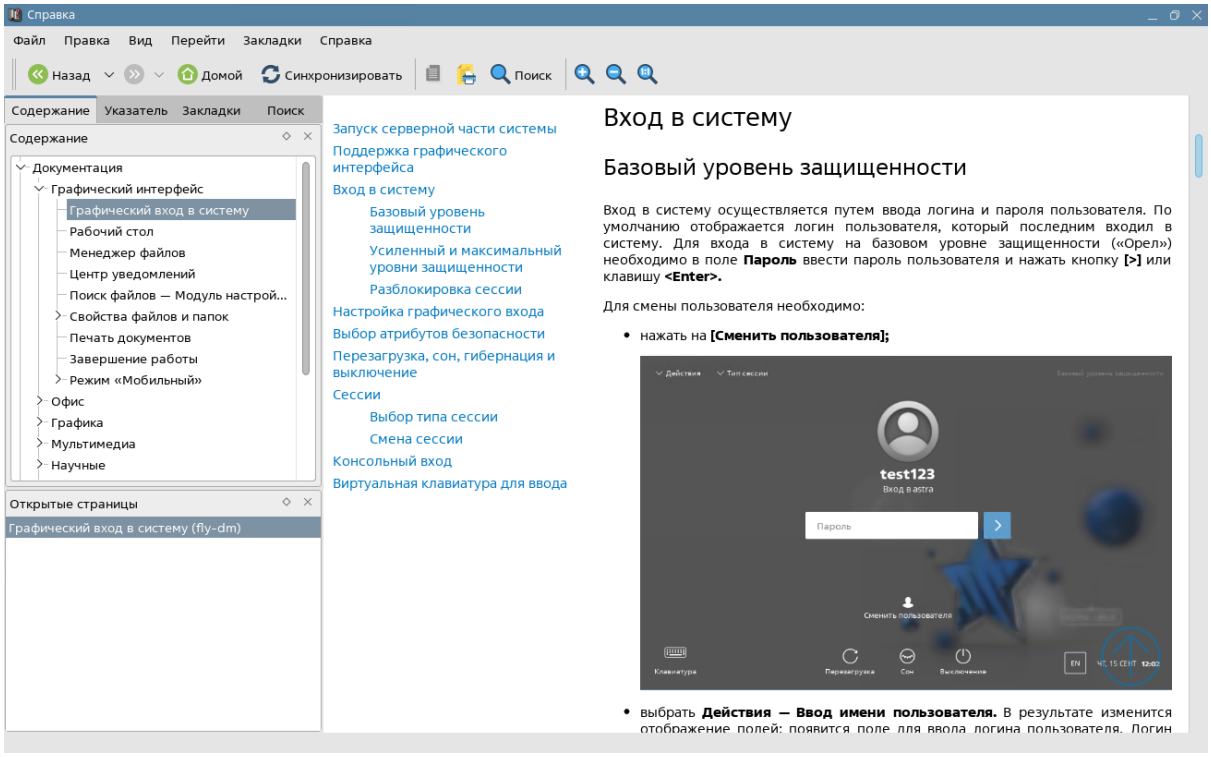


рис. 1.32 – Справка по входу в систему

- в правой части окна находится текст просматриваемой в данный момент страницы.

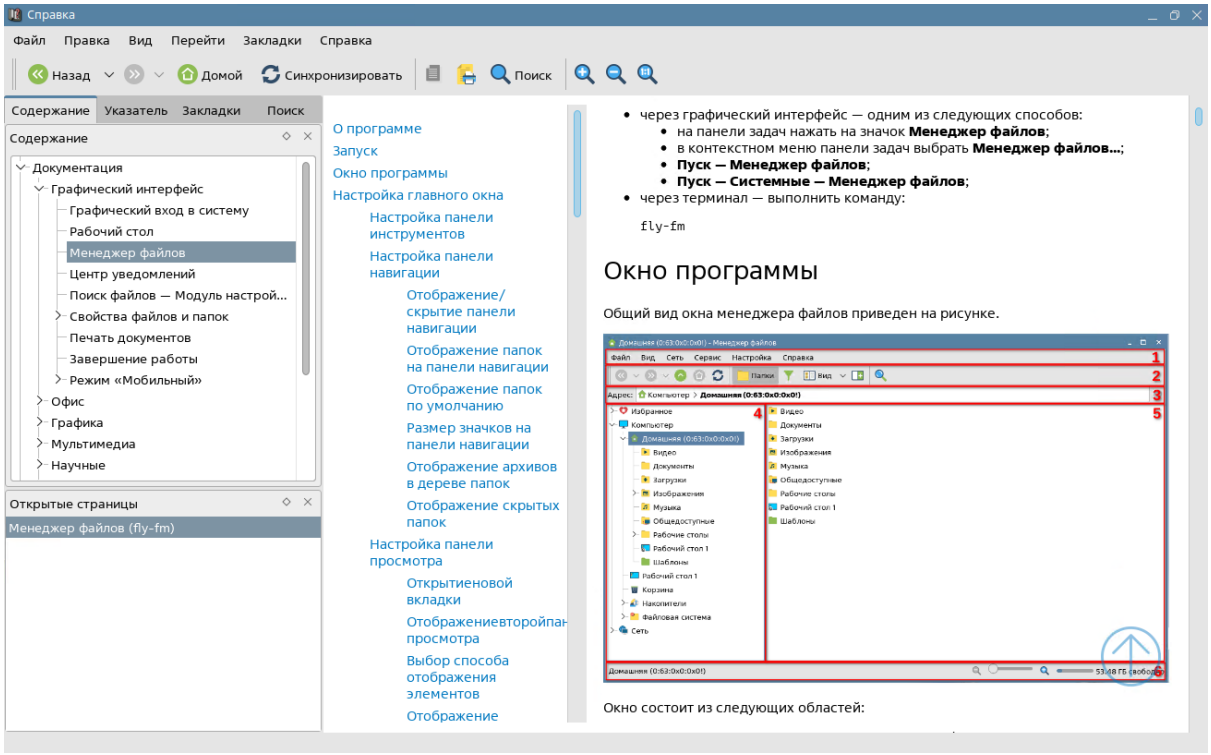


рис. 1.33 – Справка по файловому менеджеру fly-fm

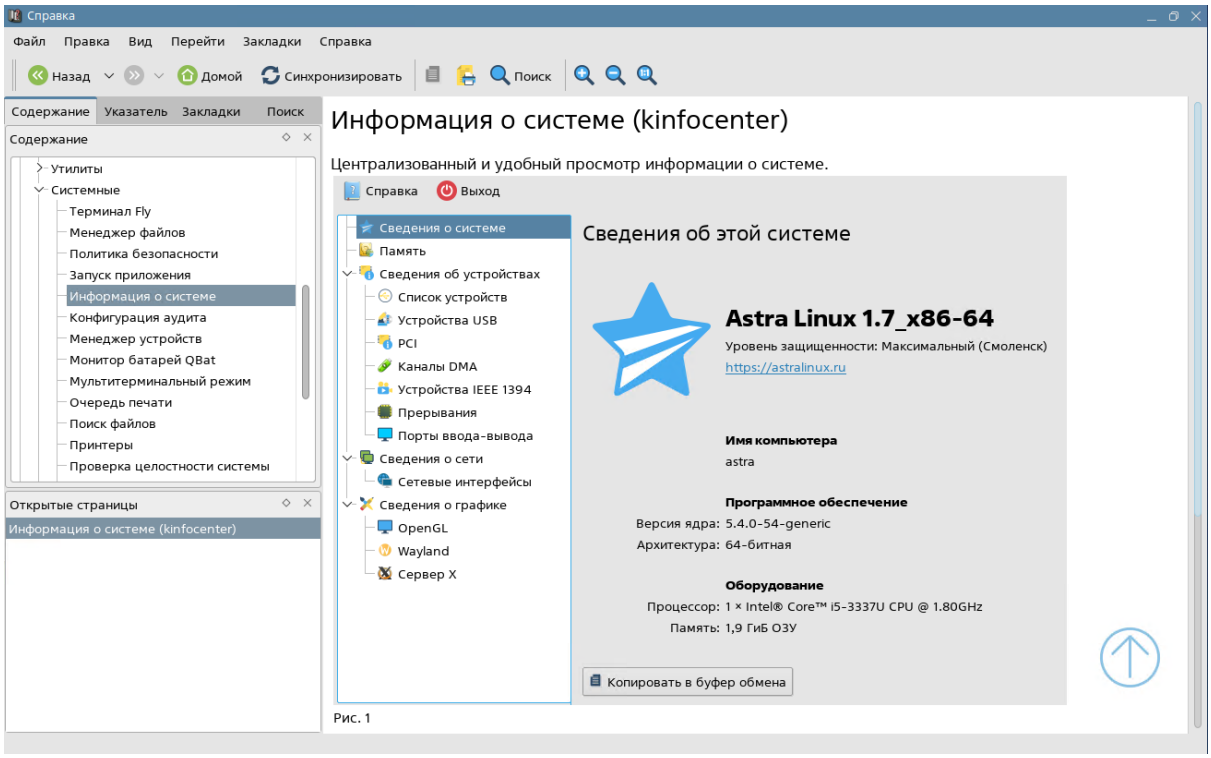


рис. 1.34 – Справка по утилите Информация о системе

5.7 Практика и тестирование

5.7.1 Практическая работа: Модуль 5. Работа со справочными системами

См. также *Требования, правила и цели выполнения практической работы*

5.7.1.1 Практические задания

Задание 1.

1. Введите команду `pwd` и затем команду `!!`. Убедитесь, что вызов `!!` привел к повторному выполнению команды `pwd`.
2. По умолчанию команды `!147` и `!!` повторяют выполнение 147-й и предыдущей команды соответственно, но можно сделать так, чтобы текст команды просто появлялся в строке ввода для дальнейшего редактирования без запуска команды.

Подсказки:

- Вам нужно будет изменить значение одного из параметров оболочки bash, что делается встроенной командой.
- Поиск информации можно начать с фразы «подстановка из списка истории».

1. Измените настройки и проверьте, что вызов команды `!!` не повторяет выполнение предыдущей команды, а помещает текст этой команды в строку ввода.

Задание 2.

1. Откройте справочную систему texinfo и изучите справку по командам.
2. Из основного меню справки перейдите в раздел «Coreutils», а из него в «Special file types», используя поиск и переход по ссылкам.
3. Изучите список доступных команд и справку по команде `link`.
4. Закройте приложение и вызовите справку texinfo для команды `ln`.
5. Объясните, чем команды `link` и `ln` похожи, а чем отличаются.

Задание 3.

1. Изучите графическую справку и её интерфейс управления.

2. Найдите страницу «Поиск файлов» с помощью «Указателя» и ознакомьтесь с ней.

3. Найдите справочную страницу о программе fly-admin-time и ознакомьтесь с ней.

5.7.1.2 Ответы на практические задания (пошаговые инструкции)

Задание 1.

1. Введите команду `pwd` и затем команду `!!`. Убедитесь, что вызов `!!` привел к повторному выполнению команды `pwd`.

   a. Выполните `pwd`.

   b. Выполните `!!` и убедитесь, что была выполнена команда `pwd`.

2. По умолчанию команды `!147` и `!!` повторяют выполнение 147-й и предыдущей команды соответственно, но можно сделать так, чтобы текст команды просто появлялся в строке ввода для дальнейшего редактирования без запуска команды.

   a. Выполните команду `man bash`.

   b. Перейдите к параграфу, который содержит подстроку «подстановка из списка истории», используя обычный поиск / или поиск с отображением совпадающих строк.

   c. Если вы выбрали поиск с отображением совпадающих строк:

      i. Включите отображение нумерации строк командой `-N` затем `Enter`.

      ii. Выполните поиск всех вхождений командой `&подстановка из списка истории` затем `Enter`.

      iii. Запомните желаемый номер строки и отмените результаты поиска командой `&` затем `Enter`.

      iv. Перейдите к нужной части страницы командой `N` затем `<g>`, где N – это номер строки.

   d. Изучите текст на странице и найдите имя встроенной команды, которая переключает опции оболочки и название интересующей опции.

   e. Повторите шаги c – e для изучения справки по команде `shopt`.

3. Измените настройки и проверьте, что вызов команды `!!` не повторяет выполнение предыдущей команды, а помещает текст этой команды в строку ввода.

   a. Измените режим работы оболочки bash командой `shopt -s histverify`.

b. Выполните команду `pwd`, а затем `!!`. Убедитесь, что вызов `!!` не выполняет предыдущую команду, а подставляет её текст в строку ввода.

Задание 2.

1. Откройте справочную систему texinfo и изучите справку по командам.

   Откройте справочную систему командой `info` и с помощью горячей клавиши `H` отобразите список доступных команд. Используйте горячее сочетание `H` еще раз, чтобы скрыть список команд.

2. Из основного меню справки перейдите в раздел «Coreutils», а из него в «Special file types», используя поиск и переход по ссылкам.

   a. Нажмите клавишу `Tab` несколько раз, чтобы перейти к ссылке «Coreutils», и нажмите клавишу `Enter`, чтобы открыть эту страницу.

   b. Нажмите `/`, введите фразу для поиска «Special file types» и нажмите клавишу `Enter`.

3. Изучите список доступных команд и справку по команде `link`.

   С помощью клавиши `Tab` перейдите к первой ссылке «link invocation» и откройте эту страницу клавишей <Enter>. Изучите содержимое страницы и вернитесь назад горячей клавишей <u>. Перейдите по следующей ссылке «link invocation» и ознакомьтесь с содержанием страницы. Закройте справку texinfo горячей клавишей <q>.

4. Закройте приложение и вызовите справку texinfo для команды `ln`.

   Введите команду `info ln` и ознакомьтесь со справочной страницей.

5. Объясните, чем команды `link` и `ln` похожи, а чем отличаются.

   Как видите, обе утилиты позволяют создавать жесткие ссылки на файлы, но утилита `link` не имеет дополнительных опций и ожидает только два параметра: путь к исходному файлу и путь для новой жесткой ссылки. Возможности утилиты `ln` гораздо богаче. Она позволяет создавать как жесткие, так и мягкие ссылки, а также предлагает ряд дополнительных опций.

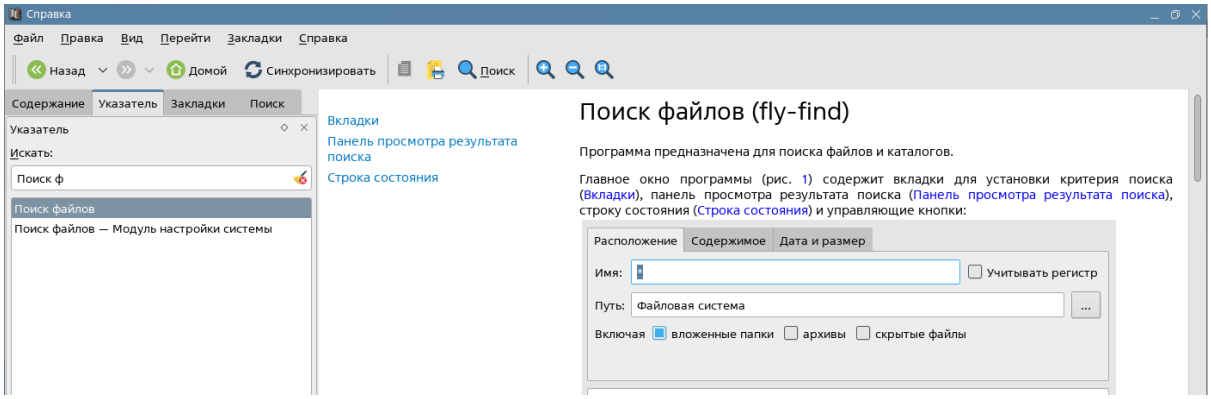
Задание 3.

1. Изучите графическую справку и её интерфейс управления.

   Откройте графическую справку сочетанием `Alt +F1`, изучите её интерфейс и доступные элементы управления.

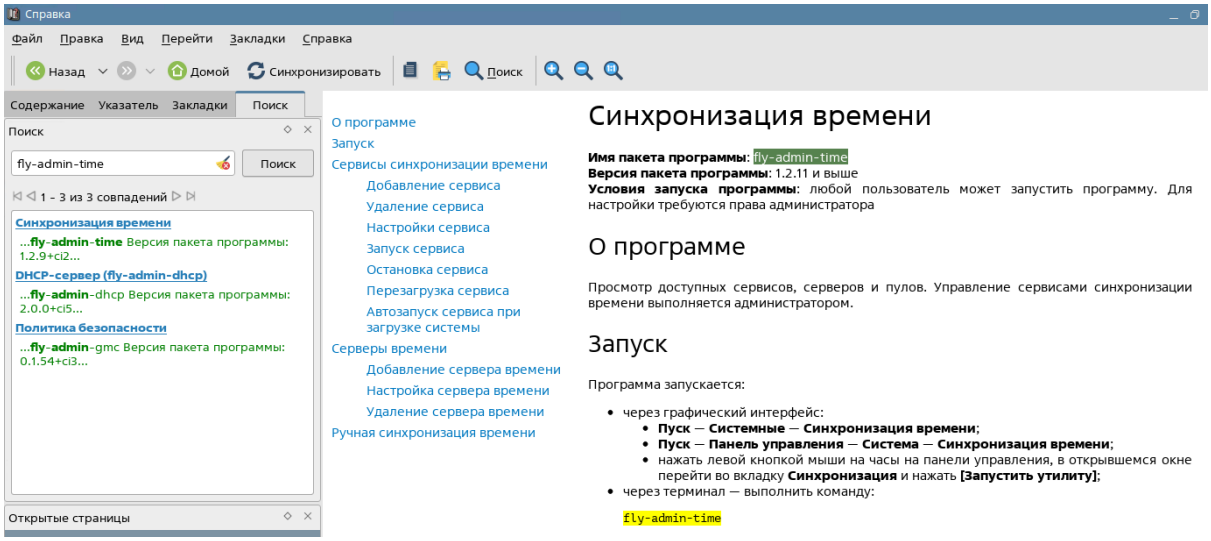
2. Найдите страницу «Поиск файлов» с помощью «Указателя» и ознакомьтесь с ней.

   Перейдите на вкладку «Указатель» в левой колонке и введите строку «Поиск ф». В результатах поиска выберите страницу с заголовком «Поиск файлов». Изучите содержимое этой страницы в основной части окна приложения.



3. Найдите справочную страницу о программе fly-admin-time и ознакомьтесь с ней.

   Перейдите на вкладку «Поиск» в левой колонке и введите строку «fly-admin-time». В результатах поиска перейдите по ссылке с заголовком «Синхронизация времени». Изучите содержимое этой страницы в основной части окна приложения.



5.7.2 Тест: Модуль 5. Работа со справочными системами

Пожалуйста, ответьте на несколько вопросов, чтобы закрепить полученные знания. Правильные ответы для самопроверки вы сможете найти в конце списка. Обратите внимание, что у некоторых вопросов может быть несколько верных вариантов ответов.

5.7.2.1 Вопросы

Вопрос 1. Как можно получить справку по встроенным командам оболочки Bash в Astra Linux:

a. Горячим сочетанием `Alt + F1`

b. Командой `help`

c. Командой `man`

d. Командой `info`

Вопрос 2. Как можно получить информацию из справочных страниц в Astra Linux:

a. Горячим сочетанием `Alt + F1`

b. Командой `help`

c. Командой `man`

d. Командой `info`

Вопрос 3. Как можно открыть справку проекта GNU в Astra Linux:

a. Горячим сочетанием `Alt + F1`

b. Командой `help`

c. Командой `man`

d. Командой `info`

Вопрос 4. Как можно открыть справку в графическом режиме Astra Linux:

a. Горячим сочетанием `Alt + F1`

b. Командой `help`

c. Командой `man`

d. Командой `info`

Вопрос 5. В каком каталоге расположены справочные страницы manpages?

a. В каталоге `/var/cache/man`

b. В каталоге `/usr/share/man`

c. В каталоге `/usr/lib/man`

d. В каталоге `/opt/share/man`

e. В каталоге `/usr/share/info`

Вопрос 6. В каком формате хранятся справочные страницы man?

a. Страницы html

b. Сжатые текстовые файлы xml

c. Сжатые текстовые файлы с разметкой roff/groff

d. Текстовые файлы с разметкой texinfo

e. Сжатые обычные текстовые файлы

Вопрос 7. Какая команда ищет расположение двоичных файлов, файлов исходного кода и файлов справочной страницы?

a. Команда `mandb`

b. Команда `apropos`

c. Команда `whatis`

d. Команда `whereis`

e. Команда `which`

Вопрос 8. Какая команда выводит список всех справочных страниц, в названиях или кратких описаниях которых содержится указанная строка, рассматриваемая как регулярное выражение?

a. Команда `mandb`

b. Команда `apropos`

c. Команда `whatis`

d. Команда `whereis`

e. Команда `which`

Вопрос 9. Какая команда выполняет индексирование справочных страниц?

a. Команда `mandb`

b. Команда `apropos`

c. Команда `whatis`

d. Команда `whereis`

e. Команда `which`

Вопрос 10. Какая команда отображает полный путь к указанным командам или скриптам?

a. Команда `mandb`

b. Команда `apropos`

c. Команда `whatis`

d. Команда `whereis`

e. Команда `which`

Вопрос 11. Какая команда выводит список справочных страниц из всех разделов, имена которых совпадают с введенной строкой?

a. Команда `mandb`

b. Команда `apropos`

c. Команда `whatis`

d. Команда `whereis`

e. Команда `which`

Ответы на вопросы

1. b
2. c
3. d
4. a
5. b
6. c
7. d
8. b
9. a

10. e

11. c

5.8 Заключение

В этом модуле мы познакомились с несколькими типами справочных систем в Astra Linux:

- Справка по встроенным командам оболочки bash
- Справочные страницы
- Справка от проекта GNU
- Справка по работе в графическом режиме Astra Linux

Мы изучили основные принципы работы с этими системами и попробовали найти нужную информацию только с их помощью. Затем, воспользовавшись этой информацией, мы настроили одну из опций оболочки.

В следующем модуле мы познакомимся с концепцией «Все есть файл» и стандартом иерархии файловых систем (Filesystem Hierarchy Standard, FHS). Мы узнаем об особенностях устройства классических файловых систем для Linux, познакомимся с различными типами файлов и научимся работать с ними.

5.9 Дополнительные источники информации

- Проект cht.sh, чит-коды по разным командам
- Проект tldr.sh, когда нужно посмотреть разные примеры команд

5.10 Обратная связь

Если остались вопросы, то их всегда можно задать в специальной теме.

Модуль 6. Работа с файлами и каталогами, FHS

6.1 Введение

Из этого модуля вы узнаете, что такое стандарт FHS, почему «всё есть файл», и познакомитесь с необычными видами файлов, которые обитают в мире Linux. Вы узнаете, как устроена файловая система Linux изнутри и с помощью каких команд наиболее эффективно управлять объектами файловой системы.

6.2 Основные отличия в работе с файлами

Давайте вспомним, какие различия между Windows и Linux мы обсуждали в первом модуле.

6.2.1 Всё есть файл

Первым существенным отличием является то, что в Linux файлы — это не то, чем они кажутся. Файлы используются не только для хранения информации на диске. С их помощью реализованы, например, каталоги и ссылки. А еще интерфейс доступа к файлам используется для взаимодействия с устройствами, каналами и сокетами.

Такой подход по обеспечению доступа ко всем возможностям операционной системы через унифицированный интерфейс файловой системы получил название «всё есть файл». Этот интерфейс, конечно, может быть не всегда оптимальным для решения поставленной задачи, зато он всегда одинаковый. Для примера вспомним каталог `/proc`. Мы уже говорили, что в эту точку монтируется виртуальная файловая система, которая предоставляет доступ к информации о системе и её процессах, поэтому с помощью команды `cat /proc/meminfo` можно увидеть параметры использования памяти.

Если в Linux придерживаются концепции «всё есть файл», то в Windows следуют концепции «всё есть объект», то есть у всех системных ресурсов — файлов, каталогов, аппаратных устройств — есть свойства и методы, как в объектно-ориентированном программировании.

6.2.2 Единое пространство имен

Вторым отличием от Windows является единое пространство имен файловой системы. Если в Windows каждому блочному устройству назначается собственная буква и абсолютный путь к локальному файлу имеет вид `C:\pagefile.sys`, то в *nix-подобных системах все диски встраиваются в единое пространство имен, которое начинается с косой черты `/` (корня).

Например, в операционной системе Astra Linux при подключении внешнего usb-накопителя его файловая система монтируется в точку `/run/user/<UID>/media/<ID устройства>` с помощью модуля FUSE (*от англ. filesystem in userspace — «файловая система в пользовательском пространстве»*). Более того, любые каталоги, например, `/home` или `/var`, могут физически располагаться на других устройствах, в том числе сетевых, и монтироваться средствами операционной системы.

Примечание: Вопросы, связанные с файловыми системами, будут подробно рассмотрены в *Модуль 19. Файловая система в Linux, работа с блочными устройствами и LVM*.

6.2.3 Подход к хранению файлов программ

Третьим отличием является подход к хранению файлов. В системе Windows служебные файлы операционной системы находятся в каталоге `C:\Windows`, а файлы приложений в каталоге `C:\Program Files`, причем для каждого приложения создается отдельная папка, в которой хранятся все его файлы, организованные любым способом на усмотрение разработчиков приложения.

В Linux для организации файлов придерживаются так называемого стандарта иерархии файловых систем (Filesystem Hierarchy Standard, FHS), который унифицирует размещение файлов по их назначению. Например, исполняемые файлы приложений должны находиться в каталогах `/bin` и `/sbin`` (*от англ. binary - бинарные*), конфигурационные файлы находятся в папке `/etc` (*от англ. «et cetera» - прочие*), а изменяемые файлы, такие как журналы, должны находиться в каталоге `/var` (*от англ. variable - изменчивый*).

6.2.4 Зависимость от регистра символов

Четвертое отличие состоит в том, что файловые системы Linux традиционно учитывают регистр символов, то есть в одном каталоге одновременно могут находиться файлы `test.txt`, `Test.txt` и `TEST.txt`, которые будут считаться совершенно разными файлами.

Если же примонтировать в Linux диск, отформатированный в NTFS или exFAT, то мы сможем создать только один из этих файлов и обращаться к нему по любому из предложенных вариантов именования, как в Windows. То есть эта особенность является особенностью файловой системы, а не операционки.

6.2.5 Подход к использованию ссылок

Рассмотрим пятое отличие, которое заключается в использовании ссылок.

Несмотря на то, что команда `mklink` для создания ссылок появилась в системе Windows еще со времен NT, большинство пользователей Windows ограничиваются использованием ярлыков, которые представляют собой обычные файлы с расширением `*.lnk` и содержат настройки для быстрого доступа к файлам и папкам. Полным аналогом lnk-файлов в Linux являются файлы `*.desktop`.

В системе Linux жесткие (hard link) и мягкие ссылки (symbolic link) получили широкое распространение, поэтому изменения в одном файле могут отражаться сразу в нескольких местах, что сначала может сильно сбивать с толку.

Например, с помощью команд `stat /run/cups/cups.sock` и `stat /var/run/cups/cups.sock` вы можете убедиться, что эти файлы указывают на айноду с одним и тем же номером, а все потому, что папка `/var/run` является мягкой ссылкой на папку `/run`, что можно проверить командой `ls -l /var/run`. Чуть позднее мы узнаем, что сделано так было для обеспечения обратной совместимости.

Еще интереснее, когда на один и тот же файл указывают сразу несколько жестких ссылок. Например, с помощью команд `stat /usr/bin/gunzip` и `stat /usr/bin/uncompress` вы можете легко убедиться, что это действительно две разные жесткие ссылки, которые указывают на одну и ту же айноду, поэтому до тех пор, пока мы не удалим обе ссылки, этот файл будет физически оставаться на диске.

6.2.6 Прямая косая черта в качестве разделителя

В завершение напомним, что в Linux в качестве разделителя между именами объектов файловой системы используют символ прямой косой черты «/» (слэш), например, `/etc/hosts`. В Windows можно использовать слэш, но более традиционным считается все же использование обратной косой черты «\» (бэкслэш), например, `C:\Windows\System32\drivers\etc\hosts`.

6.3 Устройство файловой системы Linux

6.3.1 Корневая файловая система и стандарт FHS

Системный диск подключается в Linux непосредственно к операционной системе, и точкой монтирования выступает та самая косая черта, которую называют еще корнем, а файловую систему системного диска, соответственно, называют корневой файловой системой. Именно к папкам корневой файловой системы в дальнейшем монтируются файловые системы других устройств.

Для того чтобы мы всегда знали, в каких каталогах искать нужные нам файлы в бессчетном множестве «аналоговнет», созданных на базе Linux, разработчикам потребовалось упорядочить структуру корневой системы, и они создали единый стандарт файловой системы (Filesystem Hierarchy Standard, FHS). Это очень странный стандарт, которого придерживаются практически все, но ко-

торому в полной мере не следует никто.

Причина такого положения вещей заключается в том, что некоторые идеи FHS уже устарели, но сообщество опасается делать резкие движения и продолжает двигаться по инерции. К слову, разработчики GoboLinux попробовали предпринять смелую попытку изменить этот статус-кво, но их похвальные усилия подверглись только насмешкам и были преданы анафеме.

Поэтому, изучая FHS, некоторые моменты лучше принять «как есть», а что-то станет понятным только в сугубо историческом контексте. А если вы думаете, что в структуре каталогов Windows нет подобных атавизмов, то вы просто еще не смотрели на нее глазами разработчиков!

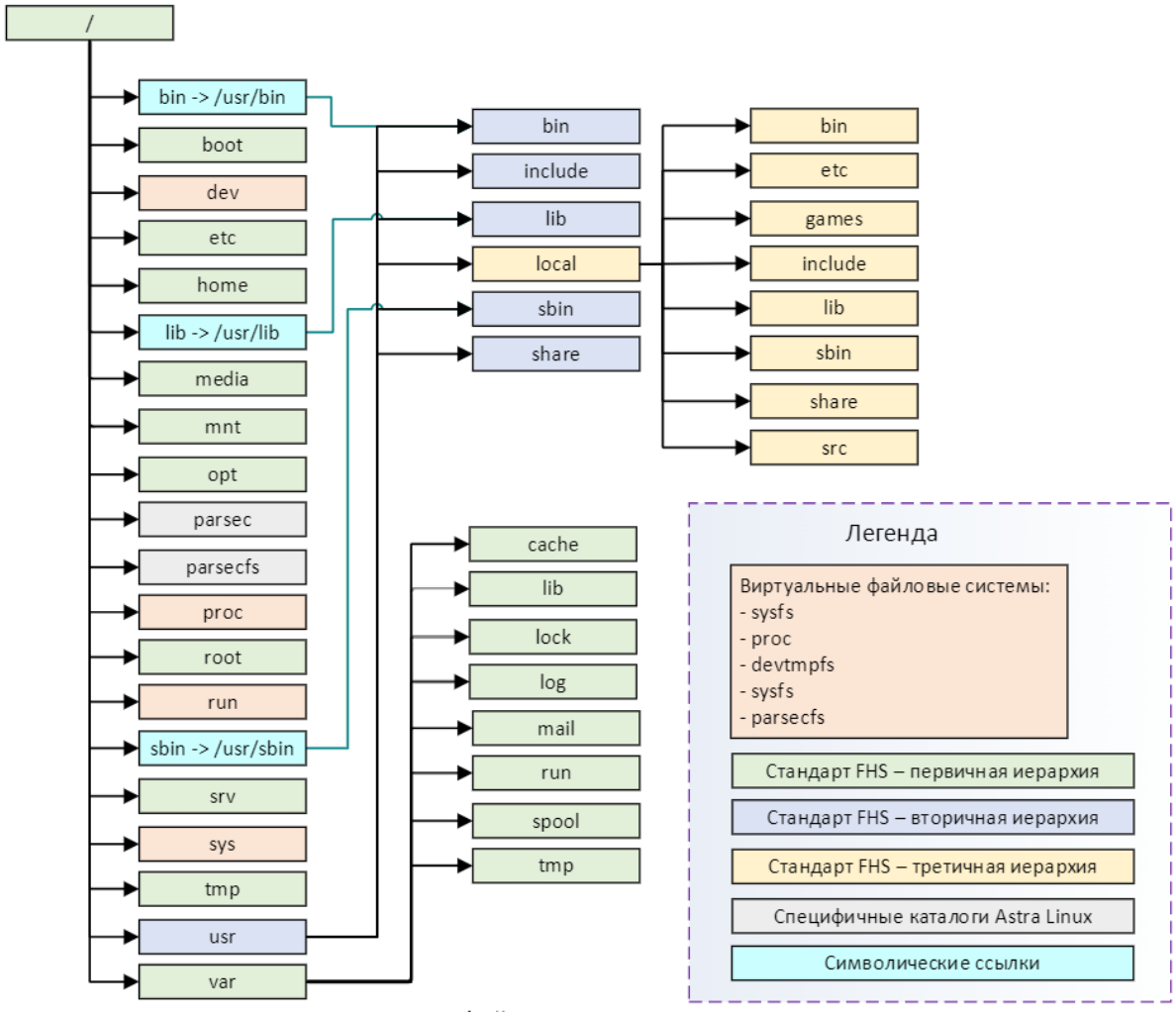


рис. 1.35 – Корневая файловая система в Astra Linux

Рассмотрим основные каталоги системы не в алфавитном порядке, чтобы наша история получилась более целостной:

- `/` – это корневой каталог системы, в эту точку монтируется системный диск.
- `/boot` – содержит конфигурационные файлы загрузчика GRUB, образы ядра системы, фай-

лы временной файловой системы initrd, используемые ядром Linux при начальной загрузке.

- `/dev` – это каталог псевдофайловой системы devtmpfs, который содержит файлы устройств (*от англ. device - устройство*). Этот каталог является примером концепции «всё есть файл», поэтому в Windows подобного каталога не существует.

  Все устройства можно разделить на две группы:

  - **Символьные устройства** — обеспечивают последовательный поток ввода/вывода. Например, мышь является символьным устройством, и каждое нажатие кнопки приводит к отправке символа в файл `/dev/input/mouse0`, в чем можно убедиться, если сделать cat этого файла.

  - **Блочные устройства** — предоставляют произвольный доступ к данным. Например, жесткий диск является блочным устройством, так как обеспечивает чтение/запись данных в произвольном порядке.

- `/media` – точка для автоматического монтирования сменных носителей, таких как FDD, CD-ROM, DVD-ROM. В эту категорию не попадают флешки – напомним, что внешние USB-носители монтируются в точку `/run/user/<UID>/media/` средствами модуля FUSE (*от англ. filesystem in userspace — «файловая система в пользовательском пространстве»*).

- `/mnt` – точка для монтирования внешних носителей вручную (*от англ. mount - монтировать*).

- `/proc` – в эту точку монтируется виртуальная файловая система procfs, с помощью которой можно получать информацию о состоянии ядра операционной системы и запущенных процессах (*от англ. process - процесс*). Например, команда `ps` берет свои данные именно через файлы procfs.

  Отметим, что некоторыми файлами procfs можно манипулировать, чтобы изменять конфигурацию ядра. Например, если записать новое значение в файл `/proc/sys/kernel/hostname`, то можно изменить имя хоста «на лету».

- `/sys` - в эту точку монтируется виртуальная файловая система sysfs, которая является новой реализацией procfs и постепенно заменяет ее функции (*от англ. system - система*).

- `/run` – каталог содержит pid-файлы процессов, файлы временных сокетов и другие данные, описывающие систему с момента ее загрузки. Каталог размещается в TMPFS, поэтому очищается автоматически при перезагрузке системы.

Примечание: Ранее для тех же целей использовался каталог `/var/run`, который все еще остался в системе для обеспечения обратной совместимости, но является теперь символической ссылкой на

каталог `/run`.

- `/bin` – содержит бинарные файлы стандартных утилит Linux, которые должны быть доступны всем пользователям системы (*от англ. binaries - двоичные файлы*). В системе Windows такие файлы имеют расширение `*.exe` и находятся в каталоге `C:\Windows\System32`. Каталог является мягкой ссылкой на каталог `/usr/bin`, о чем будет далее по тексту.

- `/sbin` – каталог содержит набор системных утилит, которые необходимы для загрузки системы и ее восстановления в случае сбоя, например, init, iptables, ifconfig (*от англ. system binaries - системные двоичные файлы*). Запускать эти утилиты имеет право только root. Каталог является мягкой ссылкой на каталог `/usr/sbin`, о чем будет далее по тексту.

- `/lib` – в этом каталоге находятся динамические библиотеки, которые нужны для работы приложений из каталогов `/bin/` и `/sbin` (*от англ. library - библиотека*). Каталог является мягкой ссылкой на каталог `/usr/lib`, о чем будет далее по тексту.

- `/usr/bin`, `/usr/sbin` и `/usr/lib` – каталоги, на которые ссылаются `/bin`, `/sbin` и `/lib`. Причину, по которой эти каталоги существуют в двух экземплярах, можно понять только в контексте развития вычислительной техники.

  Давным-давно, когда жесткие диски были размером всего в пару мегабайт, разработчики операционной системы Unix довольно быстро подошли к этому пределу, и им пришлось перенести часть файлов из каталогов bin, sbin и lib на пользовательский диск (поэтому точка монтирования и называется `/usr`, от слова user). Критерий разнесения файлов по каталогам был прост — те файлы, которые не требовались системе до монтирования пользовательского диска, переносились в каталог `/usr`.

  В дальнейшем размер дисков кратно увеличился, и это решение потеряло актуальность, но для обеспечения обратной совместимости каталоги оставили и сделали на них мягкие ссылки. Вот и получается, что этому атавизму уже более 35 лет. И в ближайшее время вряд ли что-то изменится.

- `/etc` – содержит конфигурационные файлы операционной системы и всех ее служб (от лат. «et cetera» – «и другие», «и тому подобное»). Данный каталог можно сравнить с **реестром Windows**, который был разработан для преодоления ограничений FAT16, связанных с отсутствием возможности разграничения прав доступа и медленным доступом к большому количеству маленьких INI-файлов с настройками приложений.

- `/var` – содержит файлы, которые подвергаются наиболее частому изменению (*от англ. variable - переменный*):

  - `/var/cache` - каталог предназначен для хранения файлов кэша, генерация которых тре-

бует значительного количества вычислительных ресурсов и/или операций ввода-вывода. Удаление этих файлов не должно приводить к потере данных.

- `/var/lib` – содержит файлы, которые используются для передачи информации о состоянии приложения между его вызовами или для обмена информацией между несколькими экземплярами приложения, которые работают одновременно. В этом каталоге, например, находятся файлы службы SSSD после ее установки.

- `/var/lock` – файлы блокировки, указывающие на занятость устройств и других ресурсов. Эти файлы необходимы для возможности использования одного и того же ресурса несколькими приложениями.

- `/var/log` – каталог содержит файлы журналов.

- `/var/mail` – каталог содержит почтовые ящики локальных пользователей в устаревшем формате mbox, который представляет собой обычный текстовый файл. Для отправки сообщений локальным пользователям достаточно, чтобы в системе была установлена утилита `sendmail`, а, если в системе настроен модуль pam_mail, то при входе в терминал пользователи будут получать уведомления «you have new mail». Локальная почта в какой-то степени является аналогом Windows-утилиты msg.exe.

- `/var/run` – устаревший каталог, который оставлен для обратной совместимости и является теперь символической ссылкой на каталог `/run`.

- `/var/spool` – задачи, ожидающие обработки (например, очередь печати или очередь входящих/исходящих писем).

- `/var/tmp` – временные файлы, которые должны сохраняться между перезагрузками системы.

- `/opt` – каталог предназначен для размещения дополнительных приложений, которые отклоняются от стандарта FHS и хранят все свои файлы в одном месте (*от англ. optional - дополнительный*).

  Каждому приложению выделяется отдельный каталог `/opt/<package>`, внутри которого могут быть каталоги bin, lib, man и др. для хранения исполняемых файлов, библиотек и справочных страниц соответственно. Установка таких приложений сводится к копированию одной папки и может выполняться даже без использования пакетных менеджеров. Таким образом, каталог `/opt` является ближайшим аналогом для каталога `C:\Program files` системы Windows.

- `/srv` – каталог предназначен для хранения файлов сетевых служб (*от англ. service - служба*). Например, если хост является веб-сервером, то в этом каталоге может размещаться папка

www с файлами обслуживаемых веб-сайтов.

- `/tmp` – каталог, в котором хранятся временные файлы. Linux, в отличие от Windows, следит за временными файлами и регулярно очищает этот каталог.

- `/home` – содержит домашние каталоги обычных пользователей, в которых хранятся личные файлы, настройки программ и прочее. В системе Windows для тех же целей предназначен каталог `C:\Users`.

- `/root` – домашний каталог суперпользователя. И тут пытливый камрад должен бы поинтересоваться, а почему каталог `/root` лежит в корне, а не в папке `/home`? И мы спешим авторитетно ответить, что это очередной артефакт из прошлого.

  > Давным-давно, когда жесткие диски были размером... ну, вы поняли. В общем, одного жесткого диска хватало впритык для хранения файлов операционной системы, поэтому каталог `/home` почти всегда монтировали с отдельного жесткого диска. Но чтобы суперпользователь не терял доступ к своему домашнему каталогу даже в тех случаях, когда пользовательский диск не подключен, его домашний каталог был вынесен отдельно в корень системы. Как говорится, своя флешка ближе к делу.

- `/usr` – каталог предназначен для хранения прикладных программ и может быть смонтирован по сети. В нем размещаются общие файлы, которые могут совместно использоваться множеством компьютеров сразу, поэтому в нем не должно быть файлов, специфичных для конкретного хоста, а у пользователей не должно быть прав на запись в этот каталог. В Windows для получения того же эффекта мы можем подключить сетевую папку с архивом полезных приложений, не требующих установки.

Примечание: В корне Astra Linux есть еще два каталога, специфичных для этой операционной системы: parsec и parsecfs. Они связаны с подсистемой безопасности PARSEC, входящей в состав ядра ОС. Однако эти вопросы выходят за рамки данного курса.

6.3.2 Организации хранения данных на диске

Файловые системы Windows и Linux используют немного разные подходы к организации хранения данных на диске, понимание которых позволит избежать проблем в будущем.

В Windows основной файловой системой является NTFS. Диск разбивается на кластеры, а для хранения метаинформации о файлах используется специальная база данных, которая называется главной файловой таблицей (Master File Table, MFT). Строкам этой таблицы или так называемым записям (MFT record) соответствуют файлы, а в столбцах таблицы хранятся значения конкретных

атрибутов файлов.

По мере создания новых файлов размер MFT увеличивается, и файл этой базы может стать фрагментирован. Для того чтобы избежать деградации производительности системы в связи с фрагментированностью MFT, в момент форматирования диска система резервирует под MFT 12,5% дискового пространства и удерживает его до тех пор, пока не будет исчерпано все свободное пространство диска.

Наиболее популярной файловой системой Linux является EXT4. Диск разбивается на блоки, которые являются аналогом кластеров NTFS. А для хранения метаинформации о файлах используется структура данных, которая называется айнодой (*от англ. information node, inode — информационный узел*) и является аналогом записей MFT.

Вывод команды `ls -l /home/localadmin/hello.txt` основан на данных из айноды:



1. Права на файл. Права будут подробно рассмотрены в отдельном модуле этого курса.
2. Количество жестких ссылок на файл – о них далее.
3. Пользователь, являющийся владельцем файла.
4. Группа, являющаяся владельцем.
5. Размер файла в байтах.
6. Дата последнего изменения файла.
7. Полный путь до файла.

А с помощью ключа `-i` утилита `ls` может показать номер индексного дескриптора:

```
localadmin@astra:~$ ls -i /home/localadmin/hello.txt
424373 /home/localadmin/hello.txt
```

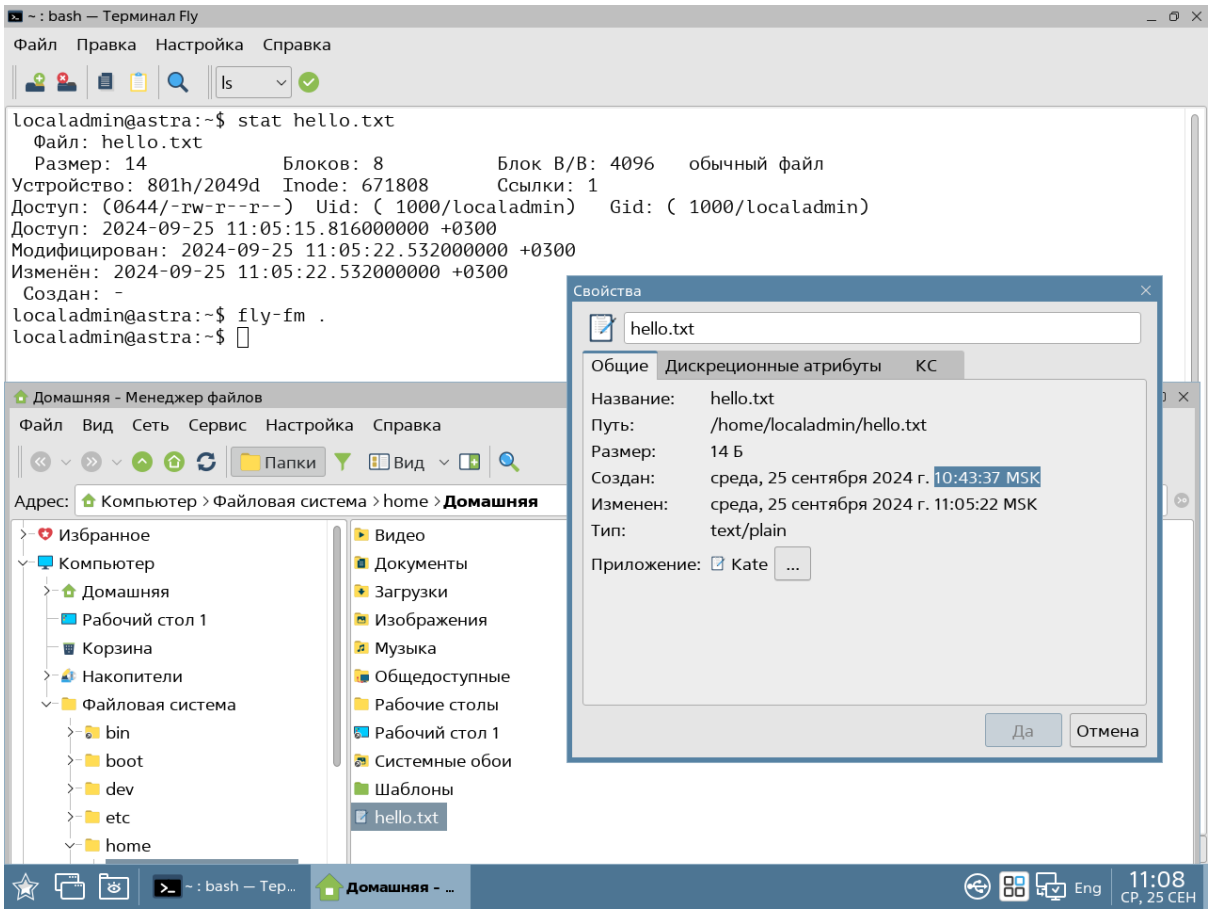
Используя команду `stat`, можно получить ряд дополнительных атрибутов файла, хранящихся в его айноде. Одним из преимуществ команды `stat` в сравнении с `ls` является то, что она показывает права доступа к файлу в числовом формате, например, в приведенном ниже примере на файл

hello.txt установлены права 0644:

```
localadmin@astra:~$ stat /home/localadmin/hello.txt
  Файл: hello.txt
  Размер: 14            Блоков: 8          Блок В/В: 4096   обычный файл
Устройство: 801h/2049d  Inode: 671808      Ссылки: 1
Доступ: (0644/-rw-r--r--)  Uid: ( 1000/localadmin)   Gid: ( 1000/localadmin)
Доступ: 2024-09-25 11:05:15.816000000 +0300
Модифицирован: 2024-09-25 11:05:22.532000000 +0300
Изменён: 2024-09-25 11:05:22.532000000 +0300
 Создан: -
```

Примечание: Утилита `stat` в текущих редакциях ALSE не показывает дату создания файла, но вы без труда найдете ее в графической утилите «Менеджер файлов».

Кстати открыть fly-fm в текущей директории можно командой: `fly-fm .`, где символ точки – это текущий каталог, который будет отправлен как аргумент по умолчанию. Также можно открыть текущий каталог в VS Code `code -n .`.



Однако, в отличие от таблицы MFT, которая может изменять свой размер динамически, количество индексных дескрипторов статично и определяется только в момент форматирования диска. С этой

особенностью связана одна из распространенных проблем Linux, которая возникает, когда какое-нибудь приложение выжигает все доступные дескрипторы, создавая большое количество маленьких файлов, например, в каталоге для журналов. В этом случае свободного места остается много, но создать новый файл не получится. Давайте посмотрим, что можно с этим сделать.

Если подключить к виртуальной машине дополнительный диск объемом 32 Гб, затем создать на диске раздел с помощью утилиты `fdisk` и отформатировать этот раздел в ext4 утилитой `mkfs` с параметрами по умолчанию, то мы получим 8 млн. блоков по 4Кб и 2 млн. индексных дескрипторов.

```
$ lsblk
...
$ fdisk /dev/sdb
...
$ mkfs -t ext4 /dev/sdb1
...
Creating filesystem with 8388352 4k blocks and 2097152 inodes
...
```

По умолчанию размер блока составляет 4Кб, а количество дескрипторов определяется из расчета 1 дескриптор на каждые 16 384 байт дискового пространства, см. параметры blocksize и inode_ratio в конфигурационном файле `/etc/mke2fs.conf`. Вы можете изменить этот конфигурационный файл или явно задать размер блока и желаемое количество дескрипторов в момент форматирования диска с помощью ключей `-b` и `-N`:

```
$ mkfs -t ext4 -b 1024 -N 5000000 /dev/sdb1
...
Creating filesystem with 33553408 1k blocks and 5000000 inodes
...
```

Примечание: Файловая система XFS (X File System) позволяет динамически выделять блоки индексных дескрипторов, что снимает проблему их исчерпания. Если же вы используете более традиционную систему EXT4, то на этапе планирования вам следует определить оптимальные параметры диска и периодически следить за статистикой использования айнод.

Узнать количество занятых и свободных айнод можно командой `df -i`.

```
localadmin@astra:~$ df -i
Файловая система Iнодов IИспользовано ICвободно IИспользовано% Смонтировано в
udev              240770           484    240286             1% /dev
tmpfs             253151           678    252473             1% /run
/dev/sda1         987360        226603    760757            23% /
tmpfs             253151             3    253148             1% /dev/shm
tmpfs             253151             4    253147             1% /run/lock
tmpfs             253151            19    253132             1% /sys/fs/cgroup
tmpfs             253151            10    253141             1% /run/user/107
tmpfs             253151            38    253113             1% /run/user/1000
```

Еще одним отличием является способ хранения имен файлов. Если в Windows имя файла является частью записи MFT, то в Linux хранение имен реализовано с помощью жестких ссылок.

Дело в том, что индексный дескриптор представляет собой структуру данных фиксированного размера (для ext4 это 256 байт), внутри которой может находиться только ограниченный набор атрибутов: тип файла, размер, дата создания/изменения, информация о владельце и группе, базовые права доступа rwx, ну и, конечно же, ссылки на блоки данных. Поэтому дескриптор плохо подходит для хранения имен произвольной длины. Но для этого отлично подходят специальные файлы каталогов.

В файлы каталогов записываются так называемые жесткие ссылки, которые связывают имя файла с номером индексного дескриптора. У дескриптора таких ссылок может быть сразу несколько, но все они являются равнозначными, поэтому файл будет удален с диска только в том случае, если будут удалены все его жесткие ссылки.

Для того чтобы можно было легко принимать решение об удалении файла, внутри айноды есть атрибут, с помощью которого файловая система ведет подсчет количества жестких ссылок. При создании новой ссылки значение счетчика увеличивается на единицу, при удалении — уменьшается, а если счетчик станет равен нулю, то файл будет удален.

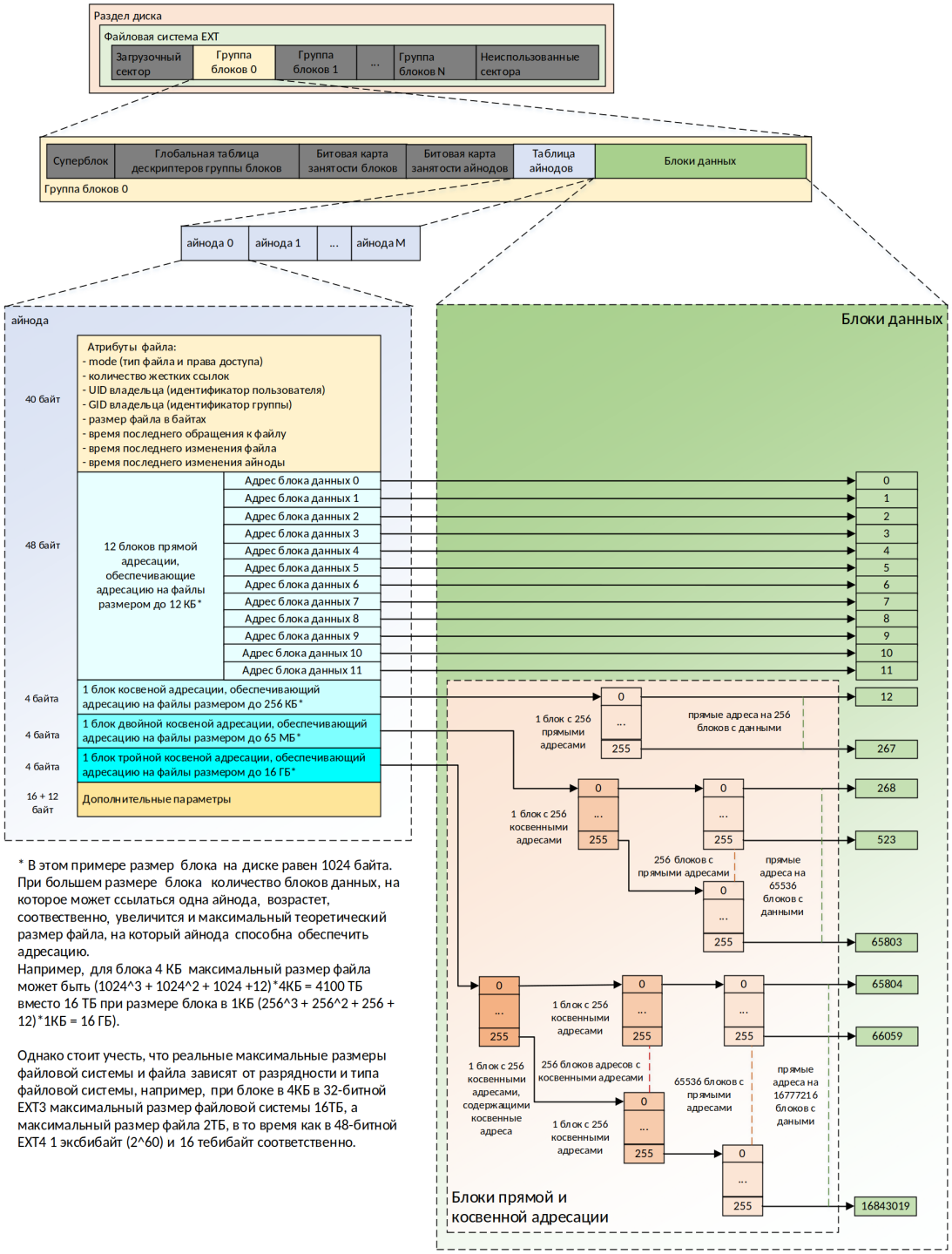


рис. 1.36 – Структура индексного дескриптора и схема адресации

Давайте рассмотрим процесс, как файловая система работает с айнодами, см. рис. 1.36. Если мы захотим прочитать файл, то система возьмет номер дескриптора из жесткой ссылки, считает метаинформацию о файле, выполнит проверку прав доступа и вернет содержимое файла, прочитав его с диска по ссылкам на блоки данных. В тех случаях, когда размер файла укладывается в

12 блоков, системе будет достаточно так называемых прямых ссылок. Для более крупных файлов есть несколько ссылок косвенной адресации, которые позволяют использовать блоки данных для хранения ссылок. Такой подход позволяет адресовать файлы размером до 16 терабайт при стандартном размере блока в 4 Кб. Структура индексного дескриптора и схема адресации блоков данных представлена на рисунке выше.

6.3.3 Типы индексных дескрипторов

В Linux существует семь типов айнод, пять из которых относят к специальным файлам, см. таблицу ниже.

Тип файла	Обозначение в ls	Назначение
Обычные файлы	**-**rwxrwxrwx	Содержат различные данные, например, тексты, картинки, архивы, скрипты, библиотеки, исполняемые файлы и т. д.
Каталоги	**d**rwxrwxrwx	Содержат жесткие ссылки, связывающие имена файлов с индексными дескрипторами.
Специальные файлы:		
Символические ссылки	**l**rwxrwxrwx	Предоставляют альтернативное имя для доступа к файлам, которые расположены в любом месте файловой системы, в том числе на других носителях (аналог mklink в Windows).
Блочные устройства	**b**rwxrwxrwx	Предоставляют интерфейс для взаимодействия с аппаратным обеспечением компьютера.
Символьные устройства	**c**rwxrwxrwx	
Каналы	**p**rwxrwxrwx	Обеспечивают в Linux работу механизмов межпроцессного взаимодействия.
Сокеты	**s**rwxrwxrwx	

6.3.3.1 Обычные файлы

К обычным файлам можно отнести все файлы с данными, например, тексты, картинки, архивы, скрипты, исполняемые файлы, библиотеки и т. д. Кратким обозначением обычных файлов является символ дефиса «-».

Откроем терминал и введем команду `ls -l`, знакомую нам по практике из модуля по работе с терминалом. В том же модуле мы создали 2 файла – script и hello.txt:

```
localadmin@astra:~$ ls -l
итого 5872
lrwxrwxrwx 1 localadmin localadmin      17 мая 23 16:35 Desktop -> Desktops/Desktop1
drwx------ 6 localadmin localadmin    4096 мая 16 12:08 Desktops
-rw-r--r-- 1 localadmin localadmin      31 мая 26 12:44 hello.txt
-rwxr--r-- 1 localadmin localadmin     117 мая 25 11:09 script
lrwxrwxrwx 1 localadmin localadmin      21 мая 16 12:08 SystemWallpapers -> /usr/
↪share/wallpapers
-r-xr-xr-x 1 localadmin localadmin 5968312 мая 20 08:48 VBoxLinuxAdditions.run
drwxr-xr-x 2 localadmin localadmin    4096 мая 16 12:08 Видео
drwxr-xr-x 2 localadmin localadmin    4096 мая 16 12:08 Документы
drwxr-xr-x 2 localadmin localadmin    4096 мая 16 12:08 Загрузки
drwxr-xr-x 3 localadmin localadmin    4096 мая 16 12:08 Изображения
drwxr-xr-x 2 localadmin localadmin    4096 мая 16 12:08 Музыка
drwxr-xr-x 2 localadmin localadmin    4096 мая 16 12:08 Общедоступные
drwxr-xr-x 2 localadmin localadmin    4096 мая 16 12:08 Шаблоны
```

В выводе команды мы можем увидеть несколько типов файлов, что можно определить по цвету строки и первому символу. Как вы можете заметить, символьные ссылки выделяются бирюзовым цветом, каталоги - синим, исполнимые файлы - зеленым, а обычные файлы отображаются цветом по умолчанию.

6.3.3.2 Каталоги

В Linux каталог - это особый файл, который хранит список жестких ссылок, связывающих имя дочернего файла с номером индексного дескриптора. С помощью обычных текстовых редакторов мы можем только посмотреть содержимое этих файлов, т.е. прямое редактирование файлов каталогов запрещается. Редактированием этих файлов занимается ядро операционной системы, получая команды создания, удаления, переименования файлов. Для просмотра содержимого каталога можно воспользоваться утилитами `ls`, `dir` или `tree`.

Примечание: Утилита `tree` выводит структуру каталогов в псевдографическом виде, который похож на вывод tree в cmd. Для её установки введите команду `sudo apt install tree`.

6.3.3.3 Символьные ссылки

Символьная или символическая ссылка – это специальный файл, внутри которого содержится ссылка на адрес другого файла по его имени, а не индексному дескриптору. Ссылки будут подробно рассмотрены чуть дальше по тексту.

6.3.3.4 Символьные и блочные устройства

Файлы устройств предназначены для обращения к аппаратному обеспечению компьютера (дискам, принтерам, терминалам и др.). Когда происходит обращение к файлу устройства, ядро операционной системы передает запрос драйверу этого устройства.

К символьным устройствам обращение происходит последовательно. Примером символьного устройства является мышь. Блочные устройства позволяют работать с данными в произвольном порядке. Примером блочного устройства является жесткий диск.

6.3.3.5 Сокеты и каналы

И каналы, и сокеты предназначены для организации взаимодействия между процессами. Ключевое отличие канала от сокета заключается в том, что канал является однонаправленным, а сокет позволяет передавать данные сразу в двух направлениях. Примером сокетов является файл диспетчера печати `/run/cups/cups.sock`.

6.3.4 Жесткие и символические ссылки

Как мы уже знаем, в Linux есть два типа ссылок: жесткие (*англ. hard link*) и символические (*англ. symbol link, символьная ссылка, «симлинк» или мягкая ссылка*). Кроме того, в графической оболочке Fly используются ярлыки `*.desktop`, которые абсолютно аналогичны ярлыкам `*.lnk` Windows.

Примечание: Стоит сказать, что файловая система NTFS поддерживает жесткие и символические ссылки, а также точки соединения (*англ. directory hard link, junction point*). Для их создания необходимо использовать `fsutil`, `mklink`, PowerShell. Однако в Windows такие ссылки используются крайне редко.

6.3.4.1 Жесткая ссылка

В Linux жесткая ссылка представляет собой запись в файле каталоге, которая связывает имя файла с номером его айноды.

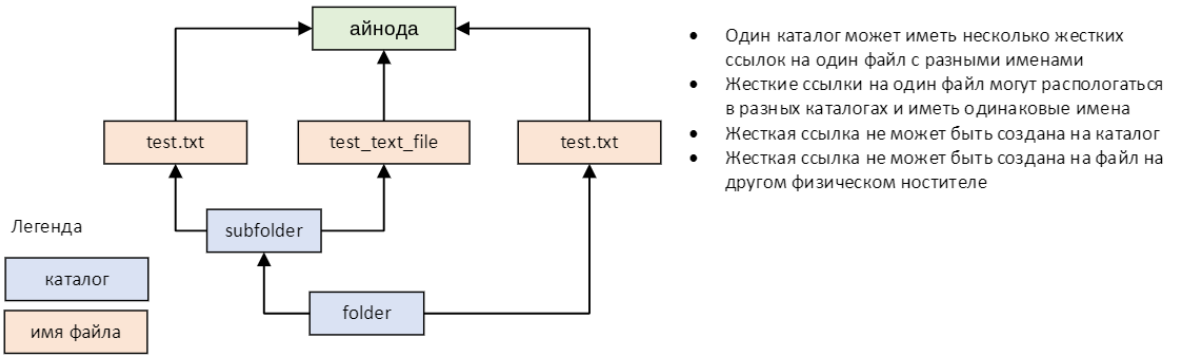


рис. 1.37 – Схема, отображающая связь между именами файлов и индексным дескриптором

При создании любого файла или каталога для него создается жесткая ссылка, и счетчик ссылок становится равен единице ($0 + 1 = 1$). Выполним команду `mkdir test` для создания папки test, перейдем в эту папку командой `cd test` и выполним команды:

- `touch test1` для создания файла test1
- `ls -il` для просмотра количества ссылок на объекты в текущем каталоге и номеров их индексных дескрипторов.

```
localadmin@astra:~$ mkdir test
localadmin@astra:~$ cd test
localadmin@astra:~/test$ touch test1
localadmin@astra:~/test$ ls -il
итого 0
453079 -rw-r--r-- 1 localadmin localadmin 0 июн  8 16:02 tes
```

Теперь создадим для этого файла жесткую ссылку командой `ln test1 hardlink_test1`, и просмотрим ссылки еще раз `ls -il`:

```
localadmin@astra:~/test$ ln test1 hardlink_test1
localadmin@astra:~/test$ ls -il
итого 0
453079 -rw-r--r-- 2 localadmin localadmin 0 июн  8 16:02 hardlink_test1
453079 -rw-r--r-- 2 localadmin localadmin 0 июн  8 16:02 test1
```

Как видим, счетчик увеличился на единицу. Создадим подкаталог командой `mkdir subtest` и создадим в нем еще одну жёсткую ссылку на этот файл командой:

`ln hardlink_test1 subtest/hardlink2_test1`

При обращении к файлу по жесткой ссылке мы можем использовать любое имя, так как все ссылаются на одну и ту же айноду и нельзя сказать, какая из жестких ссылок была создана раньше, а какая позже. Убедиться в этом можно, выполнив команду `ls -il subtest`.

Теперь обновим время последнего изменения файла командой `touch hardlink_test1` и еще раз

посмотрим свойства объектов в test и subtest:

```
localadmin@astra:~/test$ mkdir subtest
localadmin@astra:~/test$ ln hardlink_test1 subtest/hardlink2_test1
localadmin@astra:~/test$ ls -il subtest
итого 0
453079 -rw-r--r-- 3 localadmin localadmin 0 июн  8 16:02 hardlink2_test1
localadmin@astra:~/test$ touch hardlink_test1
localadmin@astra:~/test$ ls -il
итого 4
453079 -rw-r--r-- 3 localadmin localadmin    0 июн  8 16:11 hardlink_test1
453080 drwxr-xr-x 2 localadmin localadmin 4096 июн  8 16:08 subtest
453079 -rw-r--r-- 3 localadmin localadmin    0 июн  8 16:11 test1
localadmin@astra:~/test$ ls -il subtest
итого 0
453079 -rw-r--r-- 3 localadmin localadmin 0 июн  8 16:11 hardlink2_test1
```

При удалении файла командой `rm` происходит удаление жесткой ссылки на этот файл и уменьшение счётчика жестких ссылок на единицу, но с блоками данных ничего не происходит. Если счётчик ссылок станет равен нулю, то система удалит такой файл и через какое-то время задействует освобожденные блоки данных для хранения других файлов.

Так как жесткая ссылка представляет собой просто ссылку на айноду файла, в Linux запрещено создавать жесткие ссылки на айноды за пределами устройства, на котором находится эта айнода, потому что идентификатор айноды уникален только в пределах текущей файловой системы.

Те самые «.» и «..» в относительных путях являются жесткими ссылками на текущий и родительский каталог, созданные самой операционной системой. Запрещено создавать дополнительные жесткие ссылки на каталоги, так как это может привести к зацикливанию, когда один каталог ссылается на второй, а тот в свою очередь на первый.

```
localadmin@astra:~/test$ ln subtest hardlink_subtest
ln: subtest: не допускается создавать жёсткие ссылки на каталоги
```

Примечание: У команды `ln` есть специальный ключ `-d`, который позволяет на ряде систем с правами суперпользователя создать жесткую ссылку на каталог, однако в Astra Linux эта операция не разрешена:

```
localadmin@astra:~/test$ sudo ln -d subtest hardlink_subtest
ln: не удалось создать жёсткую ссылку 'hardlink_subtest' => 'subtest': Операция не␣
↪позволена
```

6.3.4.2 Символическая ссылка

Символическая ссылка, в отличие от жесткой, позволяет ссылаться на файлы и каталоги на любом устройстве, доступном в корневой файловой системе, потому что представляет собой файл, в котором хранится полный путь к файлу, на который ведет эта ссылка. Айноды символических ссылок не дублируют атрибуты тех файлов, на которые они указывают, поэтому на символические ссылки можно выставить свои собственные права доступа. На дополненной схеме связи между именами файлов и индексным дескриптором представлена связь между символическими ссылкам и именами файлов.

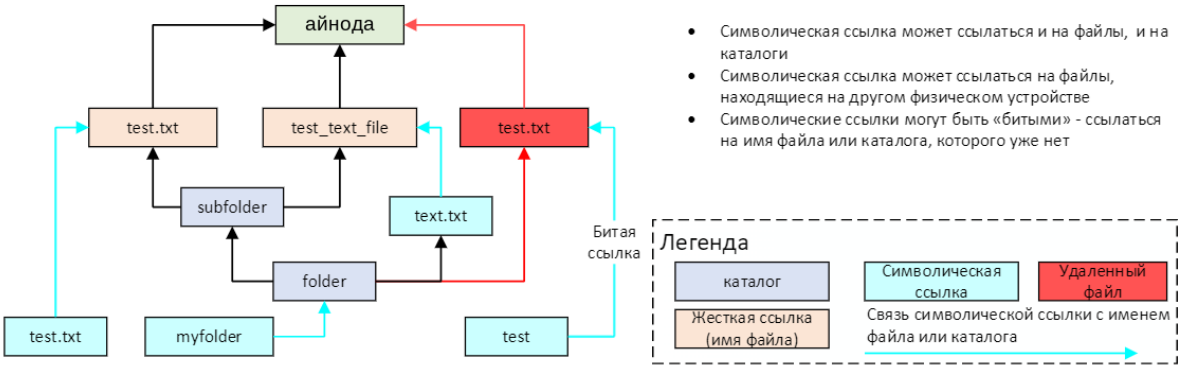


рис. 1.38 – Расширенная схема, отображающая связь между символьными ссылками, именами файлов и индексным дескриптором

Символическая ссылка ссылается на путь к файлу или папке. Если объект, на который она ссылается, будет удален (перемещен), то ссылка будет вести на несуществующий объект, то есть станет разорванной (*англ. broken link*) или «битой».

Создадим несколько символических ссылок на файлы и папку с помощью команды `ln` с ключом `-s` (`--symbolic`):

- `ln -s ~/test/subtest ~/subtest` – создаем ссылку subtest в домашнем каталоге на каталог subtest в каталоге test.

```
localadmin@astra:~$ cd ~/test
localadmin@astra:~/test$ ln -s ~/test/subtest ~/subtest
localadmin@astra:~/test$ ls -l ~/subtest
lrwxrwxrwx 1 localadmin localadmin 29 июн  9 10:38 /home/localadmin/subtest ->
/home/localadmin/test/subtest
localadmin@astra:~/test$ ln -s test1 test1_link
localadmin@astra:~/test$ ln -s hardlink_test1 hardlink_test1_link
localadmin@astra:~/test$ ls -l
итого 4
-rw-r--r-- 3 localadmin localadmin    0 июн  8 16:11 hardlink_test1
lrwxrwxrwx 1 localadmin localadmin   14 июн  9 10:41 hardlink_test1_link -> hardlink_
↪test1
drwxr-xr-x 2 localadmin localadmin 4096 июн  8 16:08 subtest
-rw-r--r-- 3 localadmin localadmin    0 июн  8 16:11 test1
```

(continues on next page)

(продолжение с предыдущей страницы)

```
lrwxrwxrwx 1 localadmin localadmin    5 июн  9 10:40 test1_link -> test1
```

Попробуем получить доступ к файлам и каталогам через символьные ссылки. Выполним команды `ls ~/subtest`, `echo "test text" > hardlink_test1_link` и `cat test1_link`:

```
localadmin@astra:~/test$ ls ~/subtest
hardlink2_test1
localadmin@astra:~/test$ echo "test text" > hardlink_test1_link
localadmin@astra:~/test$ cat test1_link
test text
localadmin@astra:~/test$ cat test1
test text
```

Как видим, с ссылками можно работать точно так же, как и с обычными файлами.

Удалим жесткую ссылку hardlink_test1 командой `rm hardlink_test1` и проверим доступ до файла через жесткую и символическую ссылки командами `cat hardlink_test1_link`, `cat test1_link`, `cat test1`:

```
localadmin@astra:~/test$ rm hardlink_test1
localadmin@astra:~/test$ ls -l
итого 8
lrwxrwxrwx 1 localadmin localadmin   14 июн  9 10:41 hardlink_test1_link -> hardlink_
↪test1
drwxr-xr-x 2 localadmin localadmin 4096 июн  8 16:08 subtest
-rw-r--r-- 2 localadmin localadmin   10 июн  9 10:50 test1
lrwxrwxrwx 1 localadmin localadmin    5 июн  9 10:40 test1_link -> test1
localadmin@astra:~/test$ cat hardlink_test1_link
cat: hardlink_test1_link: Нет такого файла или каталога
localadmin@astra:~/test$ cat test1_link
test text
localadmin@astra:~/test$ cat test1
test text
```

Как видим, после удаления жесткой ссылки вывод команды `ls` подсвечивает разорванную ссылку красным цветом, и при попытке прочитать файл мы получим сообщение, что такого файла не существует. Однако по символической ссылке на другое имя файла этот файл по-прежнему доступен, как и по другой жесткой ссылке.

В завершение отметим, что при удалении символической ссылки жесткая ссылка на файл и сами данные не удаляются.

6.3.5 Правила именования файлов и каталогов

Правила именования файлов и каталогов (помним, что каталог — это тоже файл) идентичны. Имя не может превышать 255 байт, а общая длина пути (включая все компоненты) не может превышать 4096 байт.

Минимальный безопасный набор символов для использования в именах файлов:

- буквы (как латиницы, так и кириллицы в любом регистре);
- цифры;
- символ «.» (точка);
- символ «_» (подчеркивание);
- символ «-» (тире).

Важно: Если название файла или каталога начинается с символа точки «.», то в Linux он будет считаться скрытым. Например, в рабочей папке пользователя есть такие файлы, как `.bashrc`, `.profile` и др., которые обычно не отображаются. Чтобы увидеть все файлы, включая скрытые, следует использовать команду `ls` с ключом `-a`.

Использование спецсимволов, например, пробела, символов «*», «?», «$», «’», «”» не рекомендуется, так как их наличие в именах файлов может вызвать некорректную работу некоторых программ, хотя допустимо использовать любые символы, кроме «/», например, команда `touch test\ file.txt` создаст в текущем каталоге файл, в имени которого будет содержаться пробел:

```
localadmin@astra:~$ touch test\ file.txt
localadmin@astra:~$ ls
...'test file.txt'...
```

Для решения проблем со спецсимволами в именах файлов при работе в командной строке есть несколько способов:

- **Экранирование спецсимволов.** Для экранирования специальных символов перед ними нужно поставить символ обратной косой черты «\» или взять все имя файла в одинарные кавычки, например, `cat test\ file.txt` или `cat 'test file.txt'`:

```
localadmin@astra:~$ echo text > test\ file.txt
localadmin@astra:~$ cat 'test file.txt'
text
```

- **Автодополнение** по нажатию клавиши Tab дополнит путь и автоматически сделает экранирование символов:

```
localadmin@astra:~$ cat test\ file.txt
```

- **Работа с файлами с символом тире в начале имени.** Если в начале имени файла стоит символ тире `-`, то командный интерпретатор будет воспринимать его как параметр команды. Тут не поможет стандартное экранирование символа. Вместо него можно использовать или относи-

тельный путь к файлу, или специальный параметр `--`, который сообщает команде, что это не ключ, а параметр.

```
localadmin@astra:~$ echo text > '-test.txt'

localadmin@astra:~$ cat -test.txt
cat: неверный ключ - «.»

localadmin@astra:~$ cat \-test.txt
cat: неверный ключ - «.»

localadmin@astra:~$ cat '-test.txt'
cat: неверный ключ - «.»

localadmin@astra:~$ cat -- -test.txt
text

localadmin@astra:~$ cat ./-test.txt
text
```

Примечание: Правила именования файловых объектов в Astra Linux детально рассмотрены в этой статье на wiki.astralinux.ru.

6.3.6 Расширения файлов

В именах файлов Linux, также как в Windows, могут использоваться расширения или так называемые суффиксы, например, `*.txt` или `*.doc`. Но если в Windows расширения играют ключевую роль в определении типа файла, то в Linux они игнорируются, поэтому иногда говорят, что в Linux нет расширений.

Тип файла в Linux определяется по его сигнатуре, то есть нескольким байтам в начале файла, которые однозначно указывают на тип файла и являются, таким образом, его подписью. По этой причине переименование изображения `logo.png` в текстовый файл `logo.txt` никак не сможет обмануть утилиту `file`, которая продолжит считать его цветным PNG изображением:

```
localadmin@astra:~$ wget https://astralinux.ru/logo.png
--2024-09-25 11:21:05--  https://astralinux.ru/logo.png
Распознаётся astralinux.ru (astralinux.ru)... 84.201.172.233
Подключение к astralinux.ru (astralinux.ru)|84.201.172.233|:443... соединение␣
↪установлено.
HTTP-запрос отправлен. Ожидание ответа... 200 OK
Длина: 35129 (34K) [image/png]
Сохранение в: «logo.png»

logo.png       100%[===================================>]  34,31K  --.-KB/s    за 0,
↪001s
```

(continues on next page)

(продолжение с предыдущей страницы)

```
2024-09-25 11:21:05 (26,3 MB/s) - «logo.png» сохранён [35129/35129]

localadmin@astra:~$ file logo.png
logo.png: PNG image data, 2969 x 789, 8-bit/color RGBA, non-interlaced
localadmin@astra:~$ mv logo.png logo.txt
localadmin@astra:~$ file logo.txt
logo.txt: PNG image data, 2969 x 789, 8-bit/color RGBA, non-interlaced
```

Однако стоит заметить, что файловый менеджер Astra Linux работает с расширениями аналогично тому, как это делает проводник Windows, поэтому рекомендуется все же придерживаться общепринятых соглашений, чтобы файлы можно было открывать из графики двойным кликом.

6.3.7 Абсолютный и относительный путь

Путь к объектам файловой системы представляет собой цепочку символов, которые показывают место расположения файла или каталога в файловой системе. Путь может быть абсолютным и относительным. Абсолютный или полный путь указывает на одно и то же место в файловой системе вне зависимости от текущего рабочего каталога или других обстоятельств, поэтому полный путь всегда начинается с корневого каталога. Относительный путь представляет собой путь по отношению к текущему рабочему каталогу пользователя или активных приложений.

Относительные пути могут использовать символы подстановки:

- Символ точки «.» в начале пути означает текущий каталог, например, введем команду `ls ./Desktop`, и она выведет содержимое каталога `/home/localadmin/Desktop`. В этом можно убедиться, введя команду с абсолютным путем `ls /home/localadmin/Desktop`:

```
localadmin@astra:~$ ls ./Desktop
firefox.desktop  fly-help.desktop  mycomp.desktop  mytrash.desktop
localadmin@astra:~$ ls /home/localadmin/Desktop
firefox.desktop  fly-help.desktop  mycomp.desktop  mytrash.desktop
```

- Два символа точки «..» указывают, что системе для поиска файла нужно подняться на один уровень выше. Например, если эти символы будут в самом начале пути, то они укажут на родительский каталог. Введем команду `ls ..`, и она выведет содержимое родительского каталога `/home/`:

```
localadmin@astra:~$ ls ..
localadmin
```

- Символ «~» подставляет полный путь к домашнему каталогу пользователя. Введем команду `cd /` , чтобы перейти в корень, а затем введем команду `cd ~`, чтобы перейти обратно в домашний каталог. Командой `pwd` можно проверить, в каком каталоге мы находимся, однако это не обязательно – в строке приглашения мы можем видеть текущий каталог (в данном

случае символ «~» был изменен на «/», а затем обратно на «~»):

```
localadmin@astra:~$ cd /
localadmin@astra:/$ pwd
/
localadmin@astra:/$ cd ~
localadmin@astra:~$ pwd
/home/localadmin
```

6.4 Работа с файлами и каталогами

6.4.1 Список основных команд для работы с каталогами

К основным командам можно отнести следующие:

- `pwd` — вывести полный путь к текущему каталогу (*от англ. print working directory - распечатать рабочий каталог*).
- `cd [<directory>]` — перейти в указанный каталог (*от англ. change directory - сменить каталог*).
- `mkdir <directory>` — создать каталог.
- `rmdir <directory>` — удалить пустой каталог (*от англ. remove - удалить*).
- `rm -rf <directory>` — удалить каталог со всем его содержимым.
- `du [<directory>]` — вывести размер каталога (*от англ. disk usage - использование диска*).

Рассмотрим их более подробно с примерами.

6.4.1.1 Команда pwd

Выводит полный путь к текущему каталогу, например:

```
localadmin@astra:~$ pwd
/home/localadmin
```

6.4.1.2 Команда ls

Выводит список содержимого каталога. У команды довольно много ключей, полный их список можно найти в справке `man ls`.

Рассмотрим основные:

- `ls -a` (`ls --all`) при выводе не скрывает файлы, начинающиеся с точки «.».

- `ls -A` (`ls --almost-all`) выводит список всех файлов (в том числе и скрытых), кроме «.» и «..».
- `ls -l` – в дополнение к имени каждого файла выводится дополнительная информация: тип файла, права доступа к файлу, количество ссылок на файл, имя владельца, имя группы, размер файла в байтах и временной штамп (время последней модификации файла, если не задано другое).
- `ls -R` – выводит информацию рекурсивно из всех подкаталогов.

Примеры использования:

```
localadmin@astra:~/test$ ls -a
.  ..  hardlink_test1_link  subtest  test1  test1_link
localadmin@astra:~/test$ ls -A
hardlink_test1_link  subtest  test1  test1_link
localadmin@astra:~/test$ ls -l
итого 8
lrwxrwxrwx 1 localadmin localadmin   14 июн  9 10:41 hardlink_test1_link hardlink_
↪test1
drwxr-xr-x 2 localadmin localadmin 4096 июн  8 16:08 subtest
-rw-r--r-- 2 localadmin localadmin   10 июн  9 10:50 test1
lrwxrwxrwx 1 localadmin localadmin    5 июн  9 10:40 test1_link -> test1
localadmin@astra:~/test$ ls -la
итого 16
drwxr-xr-x  3 localadmin localadmin 4096 июн  9 10:55 .
drwx------ 18 localadmin localadmin 4096 июн  9 10:38 ..
lrwxrwxrwx  1 localadmin localadmin   14 июн  9 10:41 hardlink_test1_link hardlink_
↪test1
drwxr-xr-x  2 localadmin localadmin 4096 июн  8 16:08 subtest
-rw-r--r--  2 localadmin localadmin   10 июн  9 10:50 test1
lrwxrwxrwx  1 localadmin localadmin    5 июн  9 10:40 test1_link -> test1
localadmin@astra:~/test$ ls -R
.:
hardlink_test1_link  subtest  test1  test1_link
./subtest:
hardlink2_test1
```

6.4.1.3 Команда cd

Переходит в указанный каталог. Удобно использовать относительные пути и символ «-», означающий предыдущий каталог, из которого ранее был совершён переход.

- `cd` – переход в домашний каталог пользователя
- `cd ..` – переход в родительский каталог

Примеры использования:

```
localadmin@astra:~/test$ cd
localadmin@astra:~$ cd test
```

(continues on next page)

(продолжение с предыдущей страницы)

```
localadmin@astra:~/test$ cd -
/home/localadmin
localadmin@astra:~$ cd /
localadmin@astra:/$ cd ~/test/subtest
localadmin@astra:~/test/subtest$ cd ..
localadmin@astra:~/test$
```

6.4.1.4 Команда mkdir

Команда mkdir создает каталоги

- `mkdir test2` – создаст каталог test2 в текущем каталоге.
- `mkdir –p test3/subtest/subsubtest` – создаст каталог subsubtest и все недостающие родительские каталоги в пути до него.

Примеры использования:

```
localadmin@astra:~$ ls -l | grep ^d | grep test
drwxr-xr-x 3 localadmin localadmin    4096 июн  9 10:55 test
localadmin@astra:~$ mkdir test2
localadmin@astra:~$ ls -l | grep ^d | grep test
drwxr-xr-x 3 localadmin localadmin    4096 июн  9 10:55 test
drwxr-xr-x 2 localadmin localadmin    4096 июн  9 12:46 test2
localadmin@astra:~$ mkdir ~/test3/subtest/subsubtest
mkdir: невозможно создать каталог «/home/localadmin/test3/subtest/subsubtest»: Нет␣
↪такого файла или каталога
localadmin@astra:~$ mkdir -p ~/test3/subtest/subsubtest
localadmin@astra:~$ tree test3
test3
└── subtest
    └── subsubtest
2 directories, 0 files
```

6.4.1.5 Команда rmdir

Удаляет каталоги, если они пустые.

- `rmdir test2` – удалит папку test2, если она пустая. В противном случае выдаст ошибку.

Примеры использования:

```
localadmin@astra:~$ rmdir test2
localadmin@astra:~$ rmdir test3
rmdir: не удалось удалить 'test3': Каталог не пуст
```

6.4.1.6 Команда rm -rf

Команда `rm -rf <путь>` удаляет папку вместе с её содержимым. Ключ `-r` указывает, что требуется рекурсивно удалить вложенные каталоги, ключ `-f` указывает, что не требуется запрашивать подтверждение от пользователя. Например, команда `rm -rf test3` удалит каталог *test3* и все его содержимое без дополнительных подтверждений.

Примеры использования:

```
localadmin@astra:~$ rm -rf test3
```

Внимание: Это одна из самых опасных команд в Linux. При неосторожном использовании особенно с правами суперпользователя можно безвозвратно удалить часть данных или всю корневую файловую систему, включая информацию на примонтированных носителях (`rm –rf /`).

6.4.2 Список основных команд для работы с файлами

К основным командам можно отнести следующие:

- `touch <file>` — создать файл или изменить временные метки у существующего файла.
- `realpath <file>` — узнать абсолютный путь к файлу.
- `file <file>` — определить тип файла.
- `stat <file>` — получить информацию о файле (размер файла, дата создания файла и т. д.) и проверить существование файла.
- `cp <file1> <file2>` — копировать файл «file1» в файл с именем «file2» (*от англ. copy - копировать*). Если файл с таким же названием уже существует, то он будет заменён.
- `mv <file1> <file2>` — переименовать файл «file1» в «file2» (*от англ. move - переместить*).
- `mv <file> <directory>` — переместить файл в каталог.
- `rm <file>` — удалить файл (*от англ. remove - удалить*).
- `find <file>` — найти файл.
- `sh <file>` — запустить файл со сценарием Bash.
- `./file` — запустить исполняемый файл.

Рассмотрим их более подробно с примерами.

6.4.2.1 Команда touch

Основная функция команды – обновить временные метки у файла. Однако, если требуется создать новый пустой файл, это можно сделать в том числе и этой командой.

- `touch` – обновляет временные метки существующего файла или создает новый файл.
- `touch -a` – изменяет только время последнего доступа к файлу (время, когда файл был прочитан в последний раз).
- `touch -m` – изменяет только время последнего изменения файла (время последнего изменения содержимого файла).
- `touch -d "STRING"` – может задать время, в том числе в свободной форме, например, «next Wednesday», подробнее `man touch`.
- `touch -r <referencefile> <newfile>` – копирует значения даты и времени у указанного файла.

Примеры использования:

Подготовка для демонстрации тестового каталога

```
localadmin@astra:~$ mkdir test4
localadmin@astra:~$ cd test4
localadmin@astra:~/test4$ ls
```

Создание файла

```
localadmin@astra:~/test4$ touch test.txt
localadmin@astra:~/test4$ ls -l
итого 0
-rw-r--r-- 1 localadmin localadmin 0 июн 13 18:37 test.txt
localadmin@astra:~/test4$ stat test.txt
 Файл: test.txt
 Размер: 0             Блоков: 0          Блок В/В: 4096   пустой обычный файл
Устройство: 801h/2049d  Inode: 442812      Ссылки: 1
Доступ: (0644/-rw-r--r--)  Uid: ( 1000/localadmin)   Gid: ( 1000/localadmin)
Доступ: 2023-06-13 18:37:01.960076695 +0300
Модифицирован: 2023-06-13 18:37:01.960076695 +0300
Изменён: 2023-06-13 18:37:01.960076695 +0300
Создан: -
```

Обновление временных меток файла:

```
localadmin@astra:~/test4$ touch test.txt
localadmin@astra:~/test4$ stat test.txt
  Файл: test.txt
  Размер: 0             Блоков: 0          Блок В/В: 4096   пустой обычный файл
Устройство: 801h/2049d  Inode: 442812      Ссылки: 1
Доступ: (0644/-rw-r--r--)  Uid: ( 1000/localadmin)   Gid: ( 1000/localadmin)
Доступ: 2023-06-13 18:38:35.536075019 +0300
```

(continues on next page)

(продолжение с предыдущей страницы)

```
Модифицирован: 2023-06-13 18:38:35.536075019 +0300
Изменён: 2023-06-13 18:38:35.536075019 +0300
Создан: -
```

Обновление временной метки последнего доступа:

```
localadmin@astra:~/test4$ touch -a test.txt
localadmin@astra:~/test4$ stat test.txt
 Файл: test.txt
 Размер: 0             Блоков: 0          Блок В/В: 4096   пустой обычный файл
Устройство: 801h/2049d  Inode: 442812      Ссылки: 1
Доступ: (0644/-rw-r--r--)  Uid: ( 1000/localadmin)   Gid: ( 1000/localadmin)
Доступ: 2023-06-13 18:39:09.220074416 +0300
Модифицирован: 2023-06-13 18:38:35.536075019 +0300
Изменён: 2023-06-13 18:39:09.220074416 +0300
Создан: -
```

Обновление временной метки последней модификации:

```
localadmin@astra:~/test4$ touch -m test.txt
localadmin@astra:~/test4$ stat test.txt
  Файл: test.txt
 Размер: 0             Блоков: 0          Блок В/В: 4096   пустой обычный файл
Устройство: 801h/2049d  Inode: 442812      Ссылки: 1
Доступ: (0644/-rw-r--r--)  Uid: ( 1000/localadmin)   Gid: ( 1000/localadmin)
Доступ: 2023-06-13 18:39:09.220074416 +0300
Модифицирован: 2023-06-13 18:39:28.928074063 +0300
Изменён: 2023-06-13 18:39:28.928074063 +0300
Создан: -
```

Обновление временной метки последней модификации с выбором даты в свободной форме

```
localadmin@astra:~/test4$ touch -d "next Wednesday" test.txt
localadmin@astra:~/test4$ stat test.txt
 Файл: test.txt
 Размер: 0             Блоков: 0          Блок В/В: 4096   пустой обычный файл
Устройство: 801h/2049d  Inode: 442812      Ссылки: 1
Доступ: (0644/-rw-r--r--)  Uid: ( 1000/localadmin)   Gid: ( 1000/localadmin)
Доступ: 2023-06-14 00:00:00.000000000 +0300
Модифицирован: 2023-06-14 00:00:00.000000000 +0300
Изменён: 2023-06-13 18:43:22.472069880 +0300
Создан: -
```

Примечание: Можно было бы предположить, что по аналогии с командой `mkdir` должна существовать специальная команда `mkfile`. Действительно, например, в ОС Solaris есть такая команда, однако в Linux такой команды нет. Вместо неё принято использовать команду `touch`. Пустой файл можно создать также, например, с помощью оператора > (> file) или через текстовый редактор.

```
localadmin@astra:~/test$ ls
```

(continues on next page)

(продолжение с предыдущей страницы)

```
hardlink_test1_link  subtest  test1  test1_link

localadmin@astra:~/test$  > emptyfile
localadmin@astra:~/test$ ls -l
итого 8
-rw-r--r-- 1 localadmin localadmin    0 июн 13 18:30 emptyfile
lrwxrwxrwx 1 localadmin localadmin   14 июн  9 10:41 hardlink_test1_link -> hardlink_
↪test1
drwxr-xr-x 2 localadmin localadmin 4096 июн  8 16:08 subtest
-rw-r--r-- 2 localadmin localadmin   10 июн  9 10:50 test1
lrwxrwxrwx 1 localadmin localadmin    5 июн  9 10:40 test1_link -> test1
```

6.4.2.2 Команда realpath

Команда realpath <file> – выводит полный путь до файла.

Примеры использования:

Вывод полного пути до файла:

```
localadmin@astra:~/test4$ ls
test.txt
localadmin@astra:~/test4$ realpath test.txt
/home/localadmin/test4/test.txt
```

6.4.2.3 Команда file

Команда file <file> – определяет тип файла.

Примеры использования:

```
localadmin@astra:~$ file hello.txt
hello.txt: UTF-8 Unicode text

localadmin@astra:~$ file script
script: Bourne-Again shell script, UTF-8 Unicode text executable

localadmin@astra:~$ file VBoxLinuxAdditions.run
VBoxLinuxAdditions.run: POSIX shell script executable (binary data)

localadmin@astra:~$ file /usr/bin/ls
/usr/bin/ls: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynamically ...

localadmin@astra:~$ file ~/Изображения/Wallpapers/800x600/fly-default#5682a3.png
/home/localadmin/Изображения/Wallpapers/800x600/fly-default#5682a3.png: PNG image ...
```

6.4.2.4 Команда stat

Команда stat выводит атрибуты индексного дескриптора или статус файловой системы.

- `stat <file>` – выводит содержимое атрибутов индексного дескриптора.
- `stat -f <file>` – выводит статус файловой системы, на которой расположен файл.

Примеры использования:

Вывод информации из дескриптора файла:

```
localadmin@astra:~/test4$ stat test.txt
 Файл: test.txt
 Размер: 0             Блоков: 0          Блок В/В: 4096   пустой обычный файл
Устройство: 801h/2049d  Inode: 442812      Ссылки: 1
Доступ: (0644/-rw-r--r--)  Uid: ( 1000/localadmin)   Gid: ( 1000/localadmin)
Доступ: 2023-06-14 00:00:00.000000000 +0300
Модифицирован: 2023-06-14 00:00:00.000000000 +0300
Изменён: 2023-06-13 18:50:34.144062150 +0300
Создан: -
```

Вывод информации о файловой системе, на которой расположен файл:

```
localadmin@astra:~/test4$ stat -f test.txt
 Файл: «test.txt»
   ID: 383b368255943a03 Длина имени: 255     Тип: ext2/ext3
Размер блока: 4096       Базисный размер блока: 4096
Блоки: Всего: 3865570    Свободно: 2036428    Доступно: 1835136
Inodes: Всего: 987360     Свободно: 760820
```

6.4.2.5 Команда cp

Команда `cp <источник> <назначение>` копирует файлы и каталоги, где *источник* или несколько *<источников>* копируются в указанное *<назначение>*:

- `cp file1 file2` – копирует файл *file1* в файл *file2*.
- `cp file1 destination/` – копирует файл *file1* в каталог *destination*.
- `cp -r source/ destination` (`-r`, `--recursive`) - копирует каталог source и все подкаталоги в каталог destination, если каталог destination существует. Если каталог не существует, то создаст его и скопирует **содержимое** каталога source.
- `cp -rT source destination/` (`-r`, `--recursive`; `-T`, `--no-target-directory`) - копирует содержимое каталога source и все подкаталоги в каталог destination. Если каталога destination не существует, создаст его.
- `cp -r source/* destination/` – копирует содержимое каталога source и все подкаталоги в каталог destination, если каталог destination существует. Если каталога не существует, выдаст ошибку.
- `cp -i file1 destination/` (`-i`, `--interactive`) - копирует файл *file1* в каталог с запросом на под-

твержение перезаписи. Каталог должен существовать.

- `cp -n file1 destination` – ключ `-n` запрещает заменять существующие файлы.
- `cp -b file1 destination` – ключ `-b` создает резервную копию файла.
- `cp slink destination` – копирует не символическую ссылку slink, а оригинальный файл.
- `cp hlink destination` – копирует не жесткую ссылку hlink, а оригинальный файл.
- `cp -P slink destination` – копирует символическую ссылку именно как ссылку.
- `cp --preserve=links file1_hlink1 file1_hlink2_file destination` – если file1_hlink1 и file1_hlink2 являются жесткими ссылками на один и тот же индексный дескриптор, то утилита скопирует исходный файл и создаст для него две жесткие ссылки в каталоге destination.
- `cp -p source/\* destination` – копирует содержимое каталога source в каталог destination с сохранением информации о владельцах и прав доступа, хранящейся в индексном дескрипторе.
- `cp --preserve=timestamps file destination` – копирует файл в каталог destination с сохранением временных меток оригинального файла.
- `cp -a source/* destination` – копирует структуру файловой системы 1 в 1, включая временные метки, символические и жесткие ссылки.

Примеры использования:

Подготовка для демонстрации тестового каталога с набором файлов, ссылок и подкаталогом с файлами.

```
localadmin@astra:~$ mkdir -p ./test_cp/destination
localadmin@astra:~$ cd test_cp
localadmin@astra:~$ mkdir ./source
localadmin@astra:~$ touch ./file{1..3}
localadmin@astra:~$ touch ./source/file{4..6}
localadmin@astra:~$ ln ./file1 ./hlink_file1
localadmin@astra:~$ ln -s $(realpath file2) ./slink_file2
localadmin@astra:~/test_cp$ tree
.
├── destination
├── file1
├── file2
├── file3
├── hlink_file1
├── slink_file2 -> /home/localadmin/file2
└── source
    ├── file4
    ├── file5
    └── file6
2 directories, 8 files
```

Копирование файла в файл

```
localadmin@astra:~/test_cp$ cp file1 file1_1
localadmin@astra:~/test_cp$ ls
destination  file1  file1_1  file2  file3  hlink_file1  slink_file2 source
```

Копирование файла в каталог назначения. Каталог должен существовать.

```
localadmin@astra:~/test_cp$ cp file1 destination
localadmin@astra:~/test_cp$ ls ./destination/
file1
```

Копирование каталога в каталог. Каталог должен существовать

```
localadmin@astra:~/test_cp$ cp -r source destination/
localadmin@astra:~/test_cp$ ls destination/
file1  source

localadmin@astra:~/test_cp$ cp -r source destination2/
localadmin@astra:~/test_cp$ ls destination2
file4  file5  file6

localadmin@astra:~/test_cp$ cp -r source destination2/
localadmin@astra:~/test_cp$ ls destination2
file4  file5  file6   source
```

Копирование содержимого каталога в каталог. Каталог должен существовать.

Способ 1.

```
localadmin@astra:~/test_cp$ cp -rT source destination/
localadmin@astra:~/test_cp$ ls destination/
file1  file4  file5  file6  source
```

Удаление скопированных файлов

```
localadmin@astra:~/test_cp$ rm ./destination/file{4..6}
localadmin@astra:~/test_cp$ ls destination/
file1  source
```

Способ 2.

```
localadmin@astra:~/test_cp$ cp source/* destination/
localadmin@astra:~/test_cp$ ls destination/
file1  file4  file5  file6  source
```

Копирование файла в каталог с запросом на подтверждение перезаписи. Каталог должен существовать.

```
localadmin@astra:~/test_cp$ cp -i file1 destination/
cp: переписать 'destination/file1'? y
localadmin@astra:~/test_cp$ stat destination/file
stat: не удалось выполнить stat для 'destination/file': Нет такого файла или каталога
```

(continues on next page)

(продолжение с предыдущей страницы)

```
localadmin@astra:~/test_cp$ stat destination/file1
 Файл: destination/file1
 Размер: 0             Блоков: 0          Блок В/В: 4096   пустой обычный файл
Устройство: 801h/2049d  Inode: 453120      Ссылки: 1
Доступ: (0644/-rw-r--r--)  Uid: ( 1000/localadmin)   Gid: ( 1000/localadmin)
Доступ: 2023-06-13 22:29:16.231827158 +0300
Модифицирован: 2023-06-13 22:58:56.091795284 +0300
Изменён: 2023-06-13 22:58:56.091795284 +0300
Создан: -
```

Создание резервной копии файла

```
localadmin@astra:~/test_cp$ cp -b file1 destination
localadmin@astra:~/test_cp$ ls destination
file1  file1~  file4  file5  file6  source
```

Копирование символических ссылок

```
localadmin@astra:~/test_cp$ cp slink_file2 destination/
localadmin@astra:~/test_cp$ ls -li slink_file2
442359 lrwxrwxrwx 1 localadmin localadmin 30 июн 15 09:38
slink_file2->/home/localadmin/test_cp/file2
localadmin@astra:~/test_cp$ ls -li destination/slink_file2
442645 -rw-r--r-- 1 localadmin localadmin 0 июн 15 09:40 destination/slink_file2
localadmin@astra:~/test_cp$ cp -P slink_file2 destination/
localadmin@astra:~/test_cp$ ls -li destination/slink_file2
442645 lrwxrwxrwx 1 localadmin localadmin 30 июн 15 09:41
destination/slink_file2->/home/localadmin/test_cp/file2
```

Копирование жестких ссылок

```
localadmin@astra:~/test_cp$ rm -rf destination/file1
localadmin@astra:~/test_cp$ ls -li *file1
453106 -rw-r--r-- 2 localadmin localadmin 0 июн 13 22:05 file1
453106 -rw-r--r-- 2 localadmin localadmin 0 июн 13 22:05 hlink_file1
localadmin@astra:~/test_cp$ cp file1 hlink_file1 destination/
localadmin@astra:~/test_cp$ ls -li destination/*file1
442910 -rw-r--r-- 1 localadmin localadmin 0 июн 15 10:22 destination/file1
453100 -rw-r--r-- 1 localadmin localadmin 0 июн 15 10:22 destination/hlink_file1
localadmin@astra:~/test_cp$ rm -rf destination/*file1
localadmin@astra:~/test_cp$ cp --preserve=links file1 hlink_file1 destination/
localadmin@astra:~/test_cp$ ls -li destination/*file1
442910 -rw-r--r-- 2 localadmin localadmin 0 июн 15 10:23 destination/file1
442910 -rw-r--r-- 2 localadmin localadmin 0 июн 15 10:23 destination/hlink_file1
```

Сохранение атрибутов (в этом примере временных меток) при копировании

```
localadmin@astra:~/test_cp$ cp file2 destination/file2_1
localadmin@astra:~/test_cp$ cp --preserve=timestamps file2 destination/file2_2
localadmin@astra:~/test_cp$ ls -li file2
453107 -rw-r--r-- 1 localadmin localadmin 0 июн 13 22:05 file2
localadmin@astra:~/test_cp$ ls -li destination/file2*
453100 -rw-r--r-- 1 localadmin localadmin 0 июн 15 10:25 destination/file2_1
453108 -rw-r--r-- 1 localadmin localadmin 0 июн 13 22:05 destination/file2_2
```

Полное копирование с сохранением всех атрибутов

```
localadmin@astra:~/test_cp$ cp -a source destination/source_backup
localadmin@astra:~/test_cp$ ls -l source
итого 0
-rw-r--r-- 1 localadmin localadmin 0 июн 13 22:05 file4
-rw-r--r-- 1 localadmin localadmin 0 июн 13 22:05 file5
-rw-r--r-- 1 localadmin localadmin 0 июн 13 22:05 file6
localadmin@astra:~/test_cp$ ls -l destination/source_backup
итого 0
-rw-r--r-- 1 localadmin localadmin 0 июн 13 22:05 file4
-rw-r--r-- 1 localadmin localadmin 0 июн 13 22:05 file5
-rw-r--r-- 1 localadmin localadmin 0 июн 13 22:05 file6
```

6.4.2.6 Команда mv

Команда mv – перемещает (переименовывает) файлы и каталоги.

- `mv file destination` – перемещает файл file в каталог destination.
- `mv directory destination` – перемещает каталог directory в каталог destination.
- `mv file1 file2` – переименовывает файл file1 в file2.
- `mv directory1 directory2` – переименовывает каталог directory1 в directory2.

Примеры использования:

Подготовка для демонстрации тестового каталога с набором файлов и подкаталогами с файлами.

```
localadmin@astra:~$ cd
localadmin@astra:~$ mkdir test_mv
localadmin@astra:~$ cd test_mv
localadmin@astra:~/test_mv$ touch ./file{1..3}
localadmin@astra:~/test_mv$ mkdir destination
localadmin@astra:~/test_mv$ mkdir source
localadmin@astra:~/test_mv$ touch ./source/file{4..5}
localadmin@astra:~/test_mv$ tree
.
├── destination
├── file1
├── file2
├── file3
└── source
    ├── file4
    └── file5
2 directories, 5 files
```

Перемещение файла.

```
localadmin@astra:~/test_mv$ ls -il file1
453139 -rw-r--r-- 1 localadmin localadmin 0 июн 15 10:47 file1
localadmin@astra:~/test_mv$ mv file1 destination/
```

(continues on next page)

(продолжение с предыдущей страницы)

```
localadmin@astra:~/test_mv$ ls -il destination/file1
453139 -rw-r--r-- 1 localadmin localadmin 0 июн 15 10:47 destination/file1
```

Перемещение каталога

```
localadmin@astra:~/test_mv$ mv source destination/
localadmin@astra:~/test_mv$ tree
.
├── destination
│   ├── file1
│   └── source
│       ├── file4
│       └── file5
├── file2
└── file3
2 directories, 5 files
```

Переименование файла

```
localadmin@astra:~/test_mv$ ls -li file2
453140 -rw-r--r-- 1 localadmin localadmin 0 июн 15 10:47 file2
localadmin@astra:~/test_mv$ mv file2 file4
localadmin@astra:~/test_mv$ ls -li file4
453140 -rw-r--r-- 1 localadmin localadmin 0 июн 15 10:47 file4
```

6.4.2.7 Команда rm

Команда `rm` – удаляет файлы и каталоги.

- `rm file` – удаляет файл file.
- `rm -d directory` – удаляет каталог directory, если он пуст.
- `rm -i file` – перед удалением файла file запрашивает подтверждение у пользователя.
- `rm -I files` – запрашивает подтверждение при удалении более 3 файлов.
- `rm -f file` – удаляет файл file, игнорирует ключ `-i` и отсутствующие файлы и аргументы. Никогда не запрашивает подтверждения.
- `rm -Ir directory` – рекурсивно удаляет каталоги и их содержимое. Запрашивает подтверждение.
- `rm -rf directory` – **рекурсивно удаляет каталоги и их содержимое. Никогда не запрашивает подтверждения!!!**

Примеры использования:

Подготовка для демонстрации тестового каталога с набором файлов, ссылок и подкаталогом с

файлами.

```
localadmin@astra:~$ cd
localadmin@astra:~$ mkdir -p test_rm/directory1/subdirectory
localadmin@astra:~$ cd test_rm/
localadmin@astra:~/test_rm$ mkdir directory2
localadmin@astra:~/test_rm$ touch file{1..3}
localadmin@astra:~/test_rm$ touch directory1/file{4..5}
localadmin@astra:~/test_rm$ touch directory1/subdirectory/file{6..7}
localadmin@astra:~/test_rm$ tree
.
├── directory1
│   ├── file4
│   ├── file5
│   └── subdirectory
│       ├── file6
│       └── file7
├── directory2
├── file1
├── file2
└── file3
3 directories, 7 files
```

Удаление простого файла

```
localadmin@astra:~/test_rm$ rm file1
```

Удаление простого файла с запросом на подтверждение

```
localadmin@astra:~/test_rm$ rm -i file2
rm: удалить пустой обычный файл 'file2'? y
localadmin@astra:~/test_rm$ rm -I file3
```

Удаление пустого каталога

```
localadmin@astra:~/test_rm$ rm directory2
rm: невозможно удалить 'directory2': Это каталог
localadmin@astra:~/test_rm$ rm -d directory2
```

Удаление непустого каталога

```
localadmin@astra:~/test_rm$ rm -d directory1/subdirectory
rm: невозможно удалить 'subdirectory': Каталог не пуст
localadmin@astra:~/test_rm$ rm -dr directory1/subdirectory
localadmin@astra:~/test_rm$ tree
.
├── directory1
│   ├── file4
│   └── file5
1 directory, 2 files
```

Удаление непустого каталога с запросом на подтверждение

```
localadmin@astra:~/test_rm$ rm -drI directory1
rm: удалить 1 аргумент рекурсивно? n
```

(continues on next page)

(продолжение с предыдущей страницы)

```
localadmin@astra:~/test_rm$ rm -rI directory1
rm: удалить 1 аргумент рекурсивно? n
```

Удаление непустого каталога без запроса на подтверждение

```
localadmin@astra:~/test_rm$ rm -rf directory1
```

6.4.2.8 Команда find

Команда `find` - ищет файлы в иерархии каталогов. Это мощное средство с большим количеством критериев поиска и возможностью совершать ряд действий с найденными файлами. Для получения полной справки используйте команду `man find`.

Синтаксис команды:

`find [каталог] [критерий поиска] [действие]`

Где:

- **каталог** – это отправная точка поиска,
- **критерий поиска** можно указывать несколько критериев, применяя операторы условий:

  Основные критерии поиска:

  - `-name` – поиск по имени с учетом регистра
  - `-iname` – поиск по имени без учета регистра,
  - `-type` – поиск по типу файла (f, d, l, b, c, p, s),
  - `-size` – поиск по размеру файла («+» - больше, «-» - меньше, без знака – точное совпадение, «c» - байт, «k» – Кбайт, «M» – Мбайт, «G» - Гбайт), например, «+1G», «-1M», «24K».
  - `-empty` – поиск пустых фалов,
  - `-cmin` – поиск по времени изменения в минутах, например, для поиска файлов, измененных за последний час «-60», для поиска всех файлов, кроме тех, что были изменены за последний час «+60»,
  - `-atime` – поиск файлов по времени доступа к ним в днях, например, «+30» – поиск файлов, к которым не обращались за последние 30 дней,
  - `-user` – поиск файлов по владельцу,

- `-perm` – поиск файлов по набору прав доступа, например, «755» или «777».

Основные операторы условий поиска:

- `-and` – для объединения условий логическим «и»,
- `-or` – для объединения условий логическим «или»,
- `-not` – для инвертирования условия.

- **действие** – что необходимо делать с каждым найденным файлом. Основные действия:
  - `-delete` – удаляет объекты, соответствующие результатам поиска,
  - `-ls` – выводит более подробные результаты поиска,
  - `-print` – применяется по умолчанию, если не указать другое действие, показывает полный путь к найденным файлам,
  - `-exec` – выполняет указанную команду в каждой строке результатов поиска:

    `-exec command {} \;`

    Где: `command` – команда, например, `cp`, `{}` – файл из результата поиска, а `\;` – завершение команды.

Использование команды:

- `find . -name file` – поиск файлов в текущем каталоге по имени или маске file.
- `find ./directory -name file` – поиск файлов в каталоге directory по имени или маске file.
- `find . -size +1M` – поиск файлов в текущем каталоге и подкаталогах с размером более 1МБ.
- `find . -size +1M -exec du {} \;` – поиск файлов в текущем каталоге и подкаталогах с размером более 1МБ и вывод размеров этих файлов.
- `find ./directory –empty` – поиск пустых файлов и каталогов в каталоге directory и подкаталогах.
- `find ./directory -empty -delete` – поиск пустых файлов и каталогов в каталоге directory и подкаталогах и их удаление.
- `find . -cmin -60 -size +10k` – поиск файлов и каталогов в текущем каталоге и подкаталогах, которые были изменены в течение прошлых 60 минут и у которых размер более 10КБ.
- `find . -name file\* -or -name \*txt` – поиск файлов и каталогов в текущем каталоге и подкаталогах по маске «file*» ИЛИ «*.txt».

- `find . -name file\* -type d` – поиск каталогов в текущем каталоге и подкаталогах по маске «file *».

- `find /etc -type f -exec grep -iH 'nameserver' {} \;` – поиск файлов в каталоге /etc, которые содержат подстроку nameserver.

Примеры использования:

Поиск файлов в текущем каталоге и подкаталогах по маске «test*.txt»

```
localadmin@astra:~$ find . -name test\*.txt
./test\.txt
./test4/test3.txt
./test4/test2.txt
./test4/test.txt
./test.txt
```

Поиск файлов в каталоге test4 и подкаталогах по маске «test*.txt»

```
localadmin@astra:~$ find ./test4 -name test\*.txt
./test4/test3.txt
./test4/test2.txt
./test4/test.txt
```

Поиск файлов в текущем каталоге и подкаталогах с размером более 1МБ.

```
localadmin@astra:~$ find . -size +1M
./VBoxLinuxAdditions.run
./.local/share/flyhelp/ru/.flyhelp/fts
./.cache/icon-cache.kcache
./.cache/mesa_shader_cache/index
```

Поиск файлов в текущем каталоге и подкаталогах с размером более 1МБ и вывод размеров этих файлов

```
localadmin@astra:~$ find . -size +1M -exec du {} \;
5832    ./VBoxLinuxAdditions.run
3492    ./.local/share/flyhelp/ru/.flyhelp/fts
10304   ./.cache/icon-cache.kcache
68      ./.cache/mesa_shader_cache/index
```

Поиск пустых файлов и каталогов в указанном каталоге и его подкаталогах

```
localadmin@astra:~$ find ./test -empty
./test/emptyfile
```

Поиск пустых файлов и каталогов в указанном каталоге и его подкаталогах и их удаление

```
localadmin@astra:~$ find ./test -empty -delete
localadmin@astra:~$ find ./test -empty
```

Поиск файлов и каталогов в текущем каталоге и подкаталогах, которые были изменены в течение

последних 60 минут

```
localadmin@astra:~$ find . -cmin -60
./.xsession-errors
./.config/rusbitech
./.config/rusbitech/fly-term.conf
./.local/share/baloo/index
./.local/share/baloo/index-lock
./test4
./test4/test3.txt
./test4/test2.txt
./test4/test.txt
```

Поиск файлов и каталогов в текущем каталоге и подкаталогах, которые были изменены в течение последних 60 минут и у которых размер более 10КБ

```
localadmin@astra:~$ find . -cmin -60 -size +10k
./.xsession-errors
./.local/share/baloo/index
```

Поиск файлов и каталогов в текущем каталоге и подкаталогах по маске «test*» или «*txt»

```
localadmin@astra:~$ find . -name test\* -or -name \*txt
./test
./test/test1
./test/test1_link
./Desktops/Desktop1/Новый файл.txt
./-test.txt
./.local/share/Trash/files/test
./.local/share/Trash/files/.txt
./.local/share/Trash/info/test.trashinfo
./hello.txt
./test1
./test\\.txt
./test4
./test4/test3.txt
./test4/test2.txt
./test4/test.txt
./test.txt
./test3
```

Поиск каталогов в текущем каталоге и подкаталогах по маске «test*»

```
localadmin@astra:~$ find . -name test\* -type d
./test_cp
./test
./test_cat
./test5
./test5/test_cp
./.local/share/Trash/files/test
./test4
./test_tac
./test3
./test_mv
./test_mv/test
```

6.5 Практика и тестирование

6.5.1 Практическая работа: Модуль 6. Работа с файлами и каталогами, FHS

См. также *Требования, правила и цели выполнения практической работы*

6.5.1.1 Практические задания

Задание 1.

- Найдите в директории /usr/share все файлы размером более 500 Кб.
- Полученный в предыдущем пункте список сохраните в файл /tmp/search.list.
- Выясните размер файла /tmp/search.list в килобайтах и сохраните это значение в файл /tmp/search-size.txt.
- Выясните количество строк в файле /tmp/search.list и запишите результат в файл /tmp/search-size.txt, сохранив предыдущие данные.

Задание 2.

- Создайте одной командой директорию /tmp/block6/task/1/2/3 и переместите в нее так же одной командой файлы /tmp/search.list и /tmp/search-size.txt.
- Создайте жесткую и символическую ссылки на файл /tmp/block6/task/1/2/3/search-size.txt.
- Выведите содержимое каталога /tmp/block6/task/1/2/3/ с получением айноды всех файлов.
- Удалите созданный каталог /tmp/block6/task/1/2/3/ вместе с содержимым.

6.5.1.2 Ответы на практические задания (пошаговые инструкции)

Задание 1.

- Найдите в директории /usr/share все файлы размером более 500 Кб:

```
$ find /usr/share/ -size +500k
```

- Полученный в предыдущем пункте список сохраните в файл /tmp/search.list:

```
$ find /usr/share/ -size +500k > /tmp/search.list
```

- Выясните размер файла /tmp/search.list в килобайтах и сохраните это значение в файл

/tmp/search-size.txt:

```
$ ls -sh /tmp/search.list > /tmp/search-size.txt
```

- Выясните количество строк в файле /tmp/search.list и запишите результат в файл /tmp/search-size.txt, сохранив предыдущие данные:

```
$ wc /tmp/search.list -l >> /tmp/search-size.txt
```

- Создайте одной командой директорию /tmp/block6/task/1/2/3 и переместите в нее так же одной командой файлы /tmp/search.list и /tmp/search-size.txt:

Задание 2.

```
$ mkdir -p /tmp/block6/task/1/2/3

$ mv /tmp/search.list /tmp/search-size.txt /tmp/block6/task/1/2/3/
```

- Создайте жесткую и символическую ссылки на файл /tmp/block6/task/1/2/3/search-size.txt:

```
$ ln /tmp/block6/task/1/2/3/search-size.txt /tmp/block6/task/1/2/3/search-size-hard.
↪txt

$ ln -s /tmp/block6/task/1/2/3/search-size.txt /tmp/block6/task/1/2/3/search-size-
↪soft.txt
```

- Выведите содержимое каталога /tmp/block6/task/1/2/3/ с получением айноды всех файлов:

```
$ ls -li /tmp/block6/task/1/2/3/
```

- Удалите созданный каталог /tmp/block6/task/1/2/3/ вместе с содержимым:

```
$ rm -rf /tmp/block6/task/1/2/3/
```

6.5.2 Тест: Модуль 6. Работа с файлами и каталогами, FHS

Пожалуйста, ответьте на несколько вопросов, чтобы закрепить полученные знания. Правильные ответы для самопроверки вы сможете найти в конце списка. Обратите внимание, что у некоторых вопросов может быть несколько верных вариантов ответов.

6.5.2.1 Вопросы

Вопрос 1. Какой каталог содержит конфигурационные файлы?

a. Каталог `/home`

b. Каталог `/usr`

c. Каталог `/etc`

d. Каталог `/var`

Вопрос 2. Можно ли в ОС Linux создать два файла /tmp/test.txt и /tmp/TEST.txt?

a. Нет

b. Да

Вопрос 3. Какая максимальная длина имени файла в ОС Linux?

a. 64

b. 127

c. 256

d. 255

Вопрос 4. Каким символом в выводе команды `ls -l` помечаются обычные файлы?

a. Символом `-`

b. Символом `d`

c. Символом `s`

d. Символом `+`

Вопрос 5. Каким ключом команды `ls` можно отобразить номер айноды?

a. Ключом `-l`

b. Ключом `-d`

c. Ключом `-i`

d. Ключом `-S`

Вопрос 6. Можно ли создать жесткую ссылку на файл, расположенный на другой файловой системе?

a. Да

b. Нет

Вопрос 7. Путь «tmp/1/2.txt» является абсолютным или относительным?

a. Относительный

b. Абсолютный

Вопрос 8. Что будет при вызове команды `touch` с существующим файлом?

a. Файл будет перезаписан

b. Возникнет ошибка

c. Обновятся метки времени файла

d. Ничего не произойдет

Ответы на вопросы

1. c
2. b
3. d
4. a
5. c
6. b
7. a
8. c

6.6 Заключение

В этом модуле мы узнали:

- Как работать с файлами и каталогами
- Структуру каталогов Linux и назначение директорий
- Правила именования файлов

- Какие бывают файлы в Linux
- Как работать с файлами и директориями

Теперь, научившись работать с файлами и каталогами и узнав, что в Linux все есть файл, мы понимаем важность умения эффективно работать с текстовой информацией в Linux. Следующий модуль как раз посвящен этой теме. Мы познакомимся и научимся работать с текстовой информацией, используя консольные утилиты и текстовые редакторы.

6.7 Дополнительные источники информации

- Правила именования файловых объектов в Astra Linux детально рассмотрены в статье на wiki.astralinux.ru.

6.8 Обратная связь

Если остались вопросы, то их всегда можно задать в специальной теме.

Модуль 7. Работа с текстовой информацией

7.1 Введение

В Windows большинство настроек хранится в реестре, поэтому администраторам так важно знать о том, что такое regedit. А в Linux все находится в папке `/etc` в виде обычных текстовых файлов, поэтому системным администраторам Linux не обойтись без продвинутых навыков работы с текстовыми файлами.

Из этого модуля вы узнаете о том, как редактировать файлы из командной строки и какие утилиты можно использовать для поиска, замены, слияния и сортировки текста. Мы обсудим, зачем нужны регулярные выражения, и, конечно же, разберемся в конце концов, как выйти из редактора vi/vim без полной перезагрузки компьютера)))

7.2 Основные команды для просмотра текстовых файлов

Для просмотра содержания текстовых файлов чаще всего используют следующие утилиты:

- `cat` — объединяет файлы и отправляет данные на стандартный вывод;
- `tac` — работает как команда `cat`, но выводит строки в обратном порядке;

- `less` — открывает файл в режиме постраничного просмотра;
- `head` — выводит несколько строк с начала файла;
- `tail` — выводит несколько строк с конца файла.

Примечание: Напомним, что утилитами называются внешние исполняемые файлы, а командами – встроенные директивы оболочки bash. Для повышения производительности работы некоторые утилиты встраивают внутрь оболочки в виде команд, например, `echo`. Понять, является ли используемая команда собственно командой или утилитой, вы можете с помощью команды `type`.

Рассмотрим синтаксис и примеры использования этих команд более подробно.

7.2.1 Утилита cat

Утилита `cat` объединяет файлы в один поток (от англ. con**cat**enate - объединять, поэтому произносится как «кат», а не «кэт»). Чаще всего утилиту `cat` используют следующим образом:

- `cat file1` – выводит содержимое файла file1 в стандартный поток вывода, т.е. на экран;
- `cat file1 file2` – объединяет содержимое двух файлов file1 и file2 в один поток последовательно друг за другом и выводит на экран;
- `cat -n file1` – выводит содержимое файла file1 с автоматической нумерацией строк;
- `cat -s file1` – выводит содержимое файла file1, заменяя несколько подряд идущих пустых строк на одну;
- `cat file1 file2 > file3` – перенаправляет содержимое файлов file1 и file2 в файл file3.

Однако некоторые продвинутые пользователи используют утилиту `cat` как простенький редактор для создания небольших текстовых файлов. Если выполнить команду `cat > file1`, то весь набранный далее текст будет сохранен в указанный файл по нажатию горячих клавиш `Ctrl + D`. Комбинация `Ctrl + C` позволяет выйти без сохранения, но предыдущее содержание файла будет потеряно.

Рассмотрим несколько примеров использования:

Проверим настройки DNS, для этого выведем на экран содержимое файла `resolv.conf`:

```
localadmin@astra:~$ cat /etc/resolv.conf
# Generated by NetworkManager
nameserver 10.0.2.3
```

По результату выполнения команды видим, что файл управляется службой NetworkManager.

Выполним объединение двух журналов в один поток, что может быть полезно в дальнейшем, например, для поиска сразу по двум файлам:

```
localadmin@astra:~$ sudo cat /var/log/boot.log /var/log/kern.log
...
May 10 16:21:27 astra kernel: IPv6: ADDRCONF(NETDEV_CHANGE): eth0: link becomes ready
May 10 18:14:26 astra kernel: e1000: eth0 NIC Link is Down
May 10 18:14:28 astra kernel: e1000: eth0 NIC Link is Up 1000 Mbps Full Duplex, Flow␣
↪Control: RX
May 10 20:53:13 astra kernel: 17:53:03.546136 timesync vgsvcTimeSyncWorker: Radical␣
↪host time change: 8 956 796 000 000ns (HostNow=1 716 227 583 546 000 000
May 11 00:00:12 astra kernel: 17:53:13.547092 timesync vgsvcTimeSyncWorker: Radical␣
↪guest time change: 9 518 866 935 000ns (GuestNow=1 716 227 593 547 082 00
May 11 00:00:12 astra kernel: traps: syslog-ng[467] general protection fault␣
↪ip:7635a8f7f8a0 sp:7ffed1a22630 error:0 in libpython2.7.so.1.0[7635a8e5c000+17f000]
```

Перенаправим результат объединения в новый файл:

```
localadmin@astra:~$ sudo cat /var/log/boot.log /var/log/kern.log > ~/logs
localadmin@astra:~$ cat logs
...
May 10 16:21:27 astra kernel: IPv6: ADDRCONF(NETDEV_CHANGE): eth0: link becomes ready
May 10 18:14:26 astra kernel: e1000: eth0 NIC Link is Down
May 10 18:14:28 astra kernel: e1000: eth0 NIC Link is Up 1000 Mbps Full Duplex, Flow␣
↪Control: RX
May 10 20:53:13 astra kernel: 17:53:03.546136 timesync vgsvcTimeSyncWorker: Radical␣
↪host time change: 8 956 796 000 000ns (HostNow=1 716 227 583 546 000 000
May 11 00:00:12 astra kernel: 17:53:13.547092 timesync vgsvcTimeSyncWorker: Radical␣
↪guest time change: 9 518 866 935 000ns (GuestNow=1 716 227 593 547 082 00
May 11 00:00:12 astra kernel: traps: syslog-ng[467] general protection fault␣
↪ip:7635a8f7f8a0 sp:7ffed1a22630 error:0 in libpython2.7.so.1.0[7635a8e5c000+17f000]
```

Результат выполнения будет аналогичен, но строки будут выводиться теперь из файла `logs`.

Выведем на экран содержимое файла настроек диспетчера службы имен с нумерацией строк:

```
localadmin@astra:~$ cat -n /etc/nsswitch.conf
...
 7 passwd:         files
 8 group:          files
 9 shadow:         files
10 gshadow:        files
...
```

По результату выполнения команды в строках 7-10 видим, что источниками пользователей, групп и паролей для хоста являются только локальные базы files.

Воспользуемся утилитой `cat` в качестве текстового редактора, чтобы добавить репозиторий продукта ALD Pro. Для выполнения команды нужно зайти в сессию суперпользователя, т.к. выполнения команды через `sudo` будет недостаточно. Для выхода из режима редактирования выполните команду `Ctrl+C`:

```
root@astra:~# cat > /etc/apt/sources.list.d/aldpro.list
deb https://dl.astralinux.ru/aldpro/frozen/01/2.4.0 1.7_x86-64 main base
^C
```

После создания файла можете проверить его содержимое так же командой cat, только уже без перенаправления вывода:

```
root@astra:~# cat /etc/apt/sources.list.d/aldpro.list
deb https://dl.astralinux.ru/aldpro/frozen/01/2.4.0 1.7_x86-64 main base
```

Примечание: «Редактор» `cat`, конечно же, суров, но бородатые администраторы используют его еще и в сочетании со специальной конструкцией языка bash, которая задается двумя знаками меньше «<<» и называется «встроенные документы» (*от англ. here-document*).

Если ввести команду `cat > file1 << EOF`, то запустится редактор текста, созданный средствами bash, в котором вы сможете вводить текст построчно, а для сохранения результатов нужно будет ввести строку с текстом ограничителя «EOF». В качестве ограничителя может выступать любое сочетание символов, но обычно используют именно эту аббревиатуру, которая является сокращением от фразы «End of File» (*с англ. конец файла*).

7.2.2 Утилита tac

Команда `tac` является зеркальным отражением команды `cat` — она тоже объединяет файлы в один поток, но располагает строки файлов в обратном порядке. Ввиду необходимости изменения порядка строк техническая реализация утилиты немного отличается: сначала `tac` вычитывает все строки файла, затем отправляет их в поток вывода в обратном порядке. Чаще всего утилиту `tac` используют следующим образом:

- `tac file1 file2` – выводит строки файла file1 в обратном порядке, затем строки файла file2 также в обратном порядке.

Рассмотрим пример использования:

В работе с журналами может быть полезным инвертировать порядок строк, что можно сделать как раз с помощью утилиты `tac`:

```
localadmin@astra:~$ sudo tac /var/log/boot.log /var/log/kern.log
May 6 15:46:46 astra kernel: AMD AuthenticAMD
May 6 15:46:46 astra kernel: Intel GenuineIntel
May 6 15:46:46 astra kernel: KERNEL supported cpus:
May 6 15:46:46 astra kernel: Command line: BOOT_IMAGE=/boot/vmlinuz-5.15.0-70-generic␣
↪root=UUID=33f4245d-33e8-48a1-9be7-0841215e1f46 ro parsec.max_ilev=63 quiet net.
↪ifnames=0
May 16 15:46:46 astra kernel: Linux version 5.15.0-70-generic (builder@build5) (gcc␣
↪(AstraLinuxSE 8.3.0-6) 8.3.0, GNU ld (GNU Binutils for AstraLinux) 2.31.1) #astra2␣
↪SMP Mon Apr 10 13:35:08 UTC 2023 (Astra 5.15.0-70.astra2-generic 5.15.92)
```

В результате выполнения команды в конце вывода мы увидим строки из начала файла

7.2.3 Утилита less

Утилита `less` (от англ. меньше) реализует функции программы для просмотра текстов. Она дает возможность листать многостраничные файлы и выполнять по ним поиск. Приемы работы с интерфейсом утилиты `less` были подробно рассмотрены в модуле 5 «Работа со справочными системами», поэтому не будем на этом останавливаться.

Чаще всего утилита вызывается таким образом:

- `less file1` – просмотр файла file1 утилитой `less` ;
- `cat file | less` – вывод содержимого файла file1 в стандартный поток передачи его по конвейеру утилите `less` .

7.2.4 Утилита head

Утилита `head` выводит n первых строк файла (англ. head - заголовок). По умолчанию 10 строк.

- `head file` – выводит первые 10 строк файла file на экран;
- `head -n 3 file` – выводит первые 3 строки файла file на экран.

Примеры использования:

Вывод первых 10 строк файла `/etc/passwd`

```
localadmin@astra:~$ head /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
```

(continues on next page)

(продолжение с предыдущей страницы)

```
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
```

Вывод первых 3 строк файла `/etc/passwd`

```
localadmin@astra:~$ head -n 3 /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
```

7.2.5 Утилита tail

Утилита `tail` выводит n последних строк файла (*англ. tail - хвост*). По умолчанию 10 строк. Утилита может продолжать показывать новые строки по мере их добавления к файлу.

- `tail file` – выводит последние 10 строк файла file в стандартный поток вывода, т.е. на экран;
- `tail -n 3 file` – выводит последние 3 строки файла file на экран;
- `tail -f -n 3 file` – выводит последние 3 строки файла file на экран и продолжает свою работу, отображая новые строки, добавляемые к этому файлу. Для завершения работы утилиты нужно использовать горячее сочетание клавиш Ctrl + C.

Примеры использования:

Вывод последних 10 строк файла `/etc/passwd`:

```
localadmin@astra:~$ tail /etc/passwd
fly-dm:x:107:118::/var/lib/fly-dm:/usr/sbin/nologin
ntp:x:108:121::/nonexistent:/usr/sbin/nologin
Debian-exim:x:109:122::/var/spool/exim4:/usr/sbin/nologin
nm-openvpn:x:110:124:NetworkManager OpenVPN,,,:/var/lib/openvpn/chroot:/usr/sbin/
↪nologin
hplip:x:111:7:HPLIP system user,,,:/run/hplip:/bin/false
logcheck:x:112:125:logcheck system account,,,:/var/lib/logcheck:/usr/sbin/nologin
avahi:x:113:126:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/usr/sbin/nologin
localadmin:x:1000:1000:localadmin,,,:/home/localadmin:/bin/bash
systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
vboxadd:x:998:1::/var/run/vboxadd:/bin/false
```

Вывод последних 3 строк файла `/etc/passwd`:

```
localadmin@astra:~$ tail -n 3 /etc/passwd
localadmin:x:1000:1000:localadmin,,,:/home/localadmin:/bin/bash
systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
vboxadd:x:998:1::/var/run/vboxadd:/bin/false
```

Вывод последних 3 строк файла `/var/log/auth.log` и новых строк в интерактивном режиме:

```
localadmin@astra:~$ sudo tail -f -n 3 /var/log/auth.log
[sudo] пароль для localadmin:
Jun 14 10:30:01 astra CRON[2096]: pam_unix(cron:session): session opened for user␣
↪root by (uid=0)
Jun 14 10:30:01 astra CRON[2096]: pam_unix(cron:session): session closed for user root
Jun 14 10:53:42 astra sudo[2173]: pam_unix(sudo:auth): authentication failure;␣
↪logname=localadmin uid=
1000 euid=0 tty=/dev/pts/0 ruser=localadmin rhost= user=localadmin
Jun 14 10:53:49 astra sudo[2173]: localadmin : TTY=pts/0 ; PWD=/home/localadmin ;␣
↪USER=root ; COMMAND=
/usr/bin/tail -f -n 3 /var/log/auth.log
Jun 14 10:53:49 astra sudo[2173]: pam_kiosk2(sudo:session): need_continue: UID 0␣
↪detected, skipping. U
ser: root
Jun 14 10:53:49 astra sudo[2173]: pam_unix(sudo:session): session opened for user␣
↪root by localadmin(u
id=0)
^C
```

Примечание: Для выхода из интерактивного режима используйте Ctrl + C, о чем сообщает «^C» на последней строке.

7.3 Основные команды для редактирования текстов из командной строки

Рассмотрим еще несколько часто используемых команд:

- `wc [file]` — выводит на экран статистику по количеству строк, слов и байт в указанном файле;
- `nl [file]` — добавляет нумерацию строк и выводит на экран текст из указанного файла. Пустые строки не нумеруются;
- `tee file` — перенаправляет входной поток одновременно и в файл file, и в стандартный поток вывода. Утилита может перенаправлять поток сразу в несколько файлов;
- `sort` — выполняет сортировку строк;
- `uniq` — выполняет удаление повторяющихся строк;
- `cut` — вырезает фрагмент из каждой строки файла и выводит результаты в стандартный поток вывода, с помощью чего можно вывести данные в табличном виде;
- `tr` — заменяет подстроку и передает поток дальше;

- `xargs` — выполняет указанную команду, дополняя ее аргументы информацией из строк, которые она получает из стандартного ввода. Команда может быть выполнена несколько раз по количеству строк, которые будут направлены ей.

Рассмотрим синтаксис и примеры использования этих команд более подробно.

7.3.1 Утилита wc

Анализирует содержимое файла и отправляет в стандартный поток вывода статистику по количеству строк, слов и байт (от англ. word count – количество слов).

- `wc file` – выведет на экран количество строк, слов и байт в файле file;
- `wc -l file` – выведет на экран количество строк в файле file;
- `wc -w file` – выведет на экран количество слов в файле file;
- `wc -m file` – выведет на экран количество символов в файле file;
- `wc -c file` – выведет на экран количество байт в файле file;
- `wc -l file1 file2 file3` – выведет на экран количество строк в каждом из файлов file1, file2 и file3;
- `ls | wc -l` – выведет на экран количество файлов и каталогов в текущей директории, за исключением скрытых объектов. Файлы и директории в Linux считаются скрытыми (hidden), если их название начинается с точки «.»;
- `ls -A | wc -l` – выведет на экран количество всех файлов и каталогов в текущей директории, за исключением скрытых объектов;
- `ps -e | wc -l` – выведет на экран количество процессов в системе. Реальное число процессов меньше на один (команда `wc` считает все строки, в том числе и заголовок таблицы).

Примеры использования:

Выведет на экран количество строк, слов и байт в файле `.bashrc`:

```
localadmin@astra:~$ cd
localadmin@astra:~$ wc .bashrc
113 483 3526 .bashrc
```

Выведет на экран количество строк в файле `.bash_history`:

```
localadmin@astra:~$ wc -l .bash_history
113 .bash_history
```

Выведет на экран количество файлов и каталогов в текущей директории, за исключением скрытых:

```
localadmin@astra:~$ ls | wc -l
27
```

Выведет на экран количество всех файлов и каталогов в текущей директории, за исключением скрытых:

```
localadmin@astra:~$ ls -A | wc -l
48
```

Выведет на экран количество процессов в системе. Реальное число процессов меньше на один (команда wc считает все строки, в том числе и заголовок таблицы):

```
localadmin@astra:~$ ps -e | wc -l
151
```

7.3.2 Утилита nl

Добавляет в выходной поток нумерацию строк (от англ. number lines – нумерация строк).

- `nl file` – выведет на экран файл file, добавив нумерацию строк с выравниванием нумерации по правому краю.
- `nl -n ln file` – выведет на экран файл file, добавив нумерацию строк с выравниванием нумерации по левому краю.

Примеры использования:

Выведет на экран файл file, добавив нумерацию строк с выравниванием нумерации по правому краю.

```
localadmin@astra:~$ nl /etc/passwd
    1  root:x:0:0:root:/root:/bin/bash
    2  daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
    3  bin:x:2:2:bin:/bin:/usr/sbin/nologin
    4  sys:x:3:3:sys:/dev:/usr/sbin/nologin
    5  sync:x:4:65534:sync:/bin:/bin/sync
...
   30  hplip:x:111:7:HPLIP system user,,,:/run/hplip:/bin/false
   31  logcheck:x:112:125:logcheck system account,,,:/var/lib/logcheck:/usr/sbin/
↪nologin
   32  avahi:x:113:126:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/usr/sbin/nologin
   33  localadmin:x:1000:1000:localadmin,,,:/home/localadmin:/bin/bash
   34  systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
   35  vboxadd:x:998:1::/var/run/vboxadd:/bin/false
```

Выведет на экран файл file, добавив нумерацию строк с выравниванием нумерации по левому краю

```
localadmin@astra:~$ nl -n ln /etc/passwd
1       root:x:0:0:root:/root:/bin/bash
2       daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
3       bin:x:2:2:bin:/bin:/usr/sbin/nologin
4       sys:x:3:3:sys:/dev:/usr/sbin/nologin
5       sync:x:4:65534:sync:/bin:/bin/sync
...
30      hplip:x:111:7:HPLIP system user,,,:/run/hplip:/bin/false
31      logcheck:x:112:125:logcheck system account,,,:/var/lib/logcheck:/usr/sbin/
↪nologin
32      avahi:x:113:126:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/usr/sbin/nologin
33      localadmin:x:1000:1000:localadmin,,,:/home/localadmin:/bin/bash
34      systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
35      vboxadd:x:998:1::/var/run/vboxadd:/bin/false
```

7.3.3 Утилита tee

Утилита перенаправляет входной поток одновременно и в указанные файлы, и в стандартный поток вывода. Используется следующим образом:

- `<command> | tee file1 file2` – результат команды `command` будет выведен на экран и записан в файлы file1 и file2;
- `<command> <directory> -h -d1 | tee file | nl` – результат выполнения команды `command` будет записан в файл «как есть», а на экран будет выведен с нумерацией строк.

Примеры использования:

Результат команды command будет выведен на экран и записан в файлы file1 и file2:

```
localadmin@astra:~$ date | tee file1 file2
Пт июн 16 08:47:28 MSK 2023
localadmin@astra:~$ cat file1
Пт июн 16 08:47:28 MSK 2023
localadmin@astra:~$ cat file2
Пт июн 16 08:47:28 MSK 2023
```

Результат выполнения команды du будет записан в файл «как есть», а на экран будет выведен с нумерацией строк:

```
localadmin@astra:~$ du /usr -h -d1 | tee du.out | nl
    1  4,0K    /usr/libx32
    2  4,0K    /usr/games
    3  4,0K    /usr/lib64
    4  4,1G    /usr/lib
    5  4,0K    /usr/lib32
```

(continues on next page)

(продолжение с предыдущей страницы)

```
    6  76K     /usr/local
    7  342M    /usr/bin
    8  1,2M    /usr/libexec
    9  33M     /usr/sbin
   10  12M     /usr/include
   11  189M    /usr/src
   12  2,0G    /usr/share
   13  8,0K    /usr/etc
   14  6,6G    /usr

localadmin@astra:~$ cat du.out
4,0K    /usr/libx32
4,0K    /usr/games
4,0K    /usr/lib64
4,1G    /usr/lib
4,0K    /usr/lib32
76K     /usr/local
342M    /usr/bin
1,2M    /usr/libexec
33M     /usr/sbin
12M     /usr/include
189M    /usr/src
2,0G    /usr/share
8,0K    /usr/etc
6,6G    /usr
```

7.3.4 Утилита sort

Сортирует строки текстовых файлов. Сортировка идет в следующем порядке:

- строки с цифрами размещаются выше других строк;
- строки, начинающиеся с букв нижнего регистра, размещаются выше;
- сортировка выполняется в соответствии с алфавитом;
- строки сортируются сначала по алфавиту, а уже вторично по другим правилам.

Утилита используется следующим образом:

- `sort file` – выведет на экран отсортированный файл file.
- `command | sort -r` – сортировка вывода команды `command` в обратном порядке (-r).
- `sort -u file` – выведет на экран отсортированный файл file и удалит из вывода дубликаты строк.
- `sort –k 3 file` – выведет на экран содержимое файла file с разделением на столбцы (используя пробел в качестве символа-разделителя) и сортировкой по третьему столбцу (-k 3).

- `sort –k1,1 -k3,3 file` – выведет на экран содержимое файла file с разделением на столбцы и сортировкой по 1 и 3 столбцам (-k 1,1 -k3,3). Ключ (-k1,1) означает сортировать только по первому столбцу, т.е. начать с первого столбца и закончить на первом же столбце. При указании ключа в виде (-k1), сортировка будет сделана по всем столбцам, начиная с первого, соответственно, сортировка по третьему столбцу может не сработать.

- `cat /etc/passwd | sort -t: -k 3 -nr | head` – выведет первые 10 строк (head) результата сортировки файла /etc/passwd по 3 столбцу (-k 1) с разделителем столбцов «:» (-t:) в обратном порядке (-r) при числовой сортировке (-n).

Примеры использования:

Создадим 2 тестовых файла для демонстрации команды `sort`:

```
localadmin@astra:~$ echo -e "computer\nmouse\nLAPTOP\ndata\nAstraLinux\nlaptop\nLinux\
↪nlaptop" > test_sort1.txt
localadmin@astra:~$ echo -e "a c b d\nd b a c\nc a d b\nb da fg r\nc e s we\nc ds a ck
↪" > test_sort2.txt
```

Команда `echo` с ключом **–e** интерпретирует специальные символы, в данном случае **n** – символ новой строки:

```
localadmin@astra:~$ cat test_sort1.txt
computer
mouse
LAPTOP
data
AstraLinux
laptop
Linux
laptop
```

Отсортируем файл `test_sort1.txt`:

```
localadmin@astra:~$ sort test_sort1.txt
AstraLinux
computer
data
laptop
laptop
LAPTOP
Linux
mouse
```

Отсортируем файл `test_sort1.txt` в обратном порядке:

```
localadmin@astra:~$ sort -r test_sort1.txt
mouse
Linux
LAPTOP
laptop
laptop
data
computer
AstraLinux
```

Отсортируем файл `test_sort1.txt` и уберем из вывода дубликаты:

```
localadmin@astra:~$ sort -u test_sort1.txt
AstraLinux
computer
data
laptop
LAPTOP
Linux
mouse
```

Отсортируем файл по 2 столбцу (разделитель по умолчанию пробел):

```
localadmin@astra:~$ cat test_sort2.txt
a c b d
d b a c
c a d b
b da fg r
c e s we
c ds a ck
localadmin@astra:~$ cat test_sort2.txt | sort -k 2
c a d b
d b a c
a c b d
b da fg r
c ds a ck
c e s we
```

Отсортируем файл по 1 и 3 столбцам (по значению с в колонке 1 видно, что работает и сортировка по колонке 3):

```
localadmin@astra:~$ cat test_sort2.txt | sort -k1,1 -k3,3
a c b d
b da fg r
c ds a ck
c a d b
c e s we
d b a c
```

Выведем первые 10 строк результата сортировки файла `/etc/passwd` по 3 столбцу (-k 3) с разделителем столбцов: (-t:) в обратном порядке (-r) при числовой сортировке (-n):

```
localadmin@astra:~$ cat /etc/passwd | sort -t: -k 3 -nr | head
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
localadmin:x:1000:1000:localadmin,,,:/home/localadmin:/bin/bash
systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
vboxadd:x:998:1::/var/run/vboxadd:/bin/false
avahi:x:113:126:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/usr/sbin/nologin
logcheck:x:112:125:logcheck system account,,,:/var/lib/logcheck:/usr/sbin/nologin
hplip:x:111:7:HPLIP system user,,,:/run/hplip:/bin/false
nm-openvpn:x:110:124:NetworkManager OpenVPN,,,:/var/lib/openvpn/chroot:/usr/sbin/
↪nologin
Debian-exim:x:109:122::/var/spool/exim4:/usr/sbin/nologin
ntp:x:108:121::/nonexistent:/usr/sbin/nologin
```

7.3.5 Утилита uniq

Утилита позволяет найти или удалить повторяющиеся строки. Для правильной фильтрации список должен быть предварительно отсортирован. Следует иметь в виду, что результат команды может отличаться от ожидаемого, например, ключ `-i`, который позволяет игнорировать регистр при сравнении, не работает с кириллицей.

- `command | sort | uniq` – результат команды `command` будет отсортирован, и все строки будут выведены на экран без повторов.
- `uniq -d` – выводит только строки, у которых есть повторы. Из каждого набора повторяющихся строк будет выведена только одна.
- `uniq -u` – выводит только строки, у которых нет повторов.
- `uniq -c` – выводит уникальные и неуникальные строки с указанием того, сколько раз эти строки присутствовали в исходном тексте (см. число в начале строки).
- `uniq -i` – не учитывает регистр при сравнении строк. Не работает с кириллицей.

Примеры использования:

Создание для демонстрации двух файлов с набором символов, в том числе повторяющихся:

```
localadmin@astra:~$ echo -e "а\nб\nв\nА\nг\nв\nд\nв\nД" > test_ru_uniq.txt
localadmin@astra:~$ echo -e "a\nb\nc\nA\nb\nd\nc\ne\nC" > test_en_uniq.txt
```

Первый файл с символами кириллицы. Фильтрация командой `uniq` не работает с неотсортированным списком:

```
localadmin@astra:~$ cat test_ru_uniq.txt | uniq
а
б
```

(continues on next page)

(продолжение с предыдущей страницы)

```
в
А
г
в
д
в
Д
```

Вывод всех уникальных строк:

```
localadmin@astra:~$ cat test_ru_uniq.txt | sort | uniq
а
А
б
в
г
д
Д
```

Вывод только тех строк, у которых нет повторов.

```
localadmin@astra:~$ cat test_ru_uniq.txt | sort | uniq -u
а
А
б
г
д
Д
```

Вывод только тех строк, у которых есть повторы. Из каждого набора повторяющихся строк будет выведена только одна:

```
localadmin@astra:~$ cat test_ru_uniq.txt | sort | uniq -d
в
```

Вывод только уникальных строк, не учитывая регистр (не работает с кириллицей):

```
localadmin@astra:~$ cat test_ru_uniq.txt | sort | uniq -i
а
А
б
в
г
д
Д
```

Второй файл с символами латиницы. Фильтрация командой uniq не работает с неотсортированным списком:

```
localadmin@astra:~$ cat test_en_uniq.txt | sort | uniq
a
A
b
c
C
d
e
```

Вывод только уникальных строк, не учитывая регистр:

```
localadmin@astra:~$ cat test_en_uniq.txt | sort | uniq -i
a
b
c
d
e
```

Вывод всех уникальных строк и количества повторов (цифра в начале строки):

```
localadmin@astra:~$ cat test_en_uniq.txt | sort | uniq -c
      1 a
      1 A
      2 b
      2 c
      1 C
      1 d
      1 e
```

Вывод всех уникальных строк и количества повторов (цифра в начале строки), без учета регистра:

```
localadmin@astra:~$ cat test_en_uniq.txt | sort | uniq -ic
      2 a
      2 b
      3 c
      1 d
      1 e
```

Вывод количества строк в файле `/var/log/user.log`:

```
localadmin@astra:~$ sudo cat /var/log/user.log | sort | wc -l
17
```

Вывод количества уникальных строк в файле `/var/log/user.log`:

```
localadmin@astra:~$ sudo cat /var/log/user.log | sort | uniq | wc -l
10
```

Вывод строк в файле `/var/log/user.log`, у которых нет повторов:

```
localadmin@astra:~$ sudo cat /var/log/user.log | sort | uniq -u
Jun 15 09:24:44 astra vboxadd-service.sh[1008]: Starting VirtualBox Guest Addition␣
↪service.
Jun 15 09:24:44 astra vboxadd-service.sh[1021]: VirtualBox Guest Addition service␣
↪started.
Jun 15 09:33:55 astra pulseaudio[1638]: [alsa-source-Intel ICH] alsa-source.c: ALSA␣
↪woke us up to read new data from the device, but there was actu
ally nothing to read.
Jun 15 09:33:55 astra pulseaudio[1638]: [alsa-source-Intel ICH] alsa-source.c: Most␣
↪likely this is a bug in the ALSA driver 'snd_intel8x0'. Please
report this issue to the ALSA developers.
Jun 15 09:33:55 astra pulseaudio[1638]: [alsa-source-Intel ICH] alsa-source.c: We␣
↪were woken up with POLLIN set -- however a subsequent snd_pcm_ava
il() returned 0 or another value < min_avail.
Jun 15 09:33:55 astra pulseaudio[1680]: [pulseaudio] pid.c: Daemon already running.
```

Вывод только повторяющихся строк в файле `/var/log/user.log`:

```
localadmin@astra:~$ sudo cat /var/log/user.log | sort | uniq -D
Jun 15 09:33:54 astra pulseaudio[1638]: [pulseaudio] alsa-util.c: Disabling timer-
↪based scheduling because running inside a VM.
Jun 15 09:33:54 astra pulseaudio[1638]: [pulseaudio] alsa-util.c: Disabling timer-
↪based scheduling because running inside a VM.
Jun 15 09:33:55 astra org.kde.powerdevil.backlighthelper[1706]: QDBusArgument: read␣
↪from a write-only object
Jun 15 09:33:55 astra org.kde.powerdevil.backlighthelper[1706]: QDBusArgument: read␣
↪from a write-only object
Jun 15 09:33:55 astra org.kde.powerdevil.backlighthelper[1706]: QDBusArgument: read␣
↪from a write-only object
Jun 15 09:33:55 astra org.kde.powerdevil.chargethresholdhelper[1693]: QDBusArgument:␣
↪read from a write-only object
Jun 15 09:33:55 astra org.kde.powerdevil.chargethresholdhelper[1693]: QDBusArgument:␣
↪read from a write-only object
Jun 15 09:33:55 astra org.kde.powerdevil.chargethresholdhelper[1693]: QDBusArgument:␣
↪read from a write-only object
Jun 15 09:33:55 astra org.kde.powerdevil.discretegpuhelper[1673]: QDBusArgument: read␣
↪from a write-only object
Jun 15 09:33:55 astra org.kde.powerdevil.discretegpuhelper[1673]: QDBusArgument: read␣
↪from a write-only object
Jun 15 09:33:55 astra org.kde.powerdevil.discretegpuhelper[1673]: QDBusArgument: read␣
↪from a write-only object
```

Вывод только повторяющихся строк в файле `/var/log/user.log`, где из каждого набора повторяющихся строк будет выведена только одна:

```
localadmin@astra:~$ sudo cat /var/log/user.log | sort | uniq -d
Jun 15 09:33:54 astra pulseaudio[1638]: [pulseaudio] alsa-util.c: Disabling timer-
↪based scheduling because running inside a VM.
```

(continues on next page)

(продолжение с предыдущей страницы)

```
Jun 15 09:33:55 astra org.kde.powerdevil.backlighthelper[1706]: QDBusArgument: read␣
↪from a write-only object
Jun 15 09:33:55 astra org.kde.powerdevil.chargethresholdhelper[1693]: QDBusArgument:␣
↪read from a write-only object
Jun 15 09:33:55 astra org.kde.powerdevil.discretegpuhelper[1673]: QDBusArgument: read␣
↪from a write-only object
```

7.3.6 Утилита cut

Утилита вырезает фрагмент из каждой строки файла и выводит результаты в стандартный поток вывода, с помощью чего можно вывести данные в табличном виде. Пример использования:

- `cut -c 2,3,5 file` – выведет на экран текст из файла file построчно и только 2, 3 и 5 символы в каждой из строк;
- `cut -c 3-6 file` – выведет на экран текст из файла file построчно и только символы с 3 по 6 в каждой из строк;
- `cut -c -5 file` – выведет на экран текст из файла file построчно и только первые 5 символов в каждой из строк;
- `cut -c 3- file` – выведет на экран текст из файла file построчно и только символы с 3 до последнего в каждой из строк;
- `cut -d : -f 1,3 file` – выведет на экран текст из файла file построчно и только первый и 3 столбцы. Строки разделены на столбцы разделителем «:».

Примеры использования:

Выводит на экран список всех пользователей в текущей системе:

```
localadmin@astra:~$ cut -d : -f 1 /etc/passwd
root
daemon
bin
sys
sync
games
man
lp
mail
...
```

Выводит список пользователей, авторизованных в текущей системе:

```
localadmin@astra:~$ who | cut -d' ' -f1 | sort
localadmin
localadmin
```

Выводит имя службы, использующей порт 80:

```
localadmin@astra:~$ grep -w 80 /etc/services | cut -f 1 | uniq
http
```

Выводит тип CPU:

```
localadmin@astra:~$ cat /proc/cpuinfo | grep "name" | cut -d : -f2 | uniq
Intel(R) Xeon(R) CPU E5-2680 v4 @ 2.40GHz
```

7.3.7 Утилита tr

Выполняет преобразование, подстановку (замену), сокращение и/или удаление символов, поступающих из стандартного входного потока, записывая результат на стандартное устройство вывода.

- `tr строка1 строка2 < file` – выведет на экран содержимое файла file, заменив в нем символы из строка1 на символы из строка2;
- `tr -c строка1 строка2 < file1 > file2` – сохранит в файле file2 содержимое файла file1, заменив в нем символы, отсутствующие в строка1, на символы из строка2;
- `<command> | tr -d строка1` – выведет на экран результат команды `<command>`, удалив из него символы, встречающиеся в строка1;
- `<command> | tr -s строка1 строка2` – выведет на экран результат команды `<command>`, заменив в нем последовательность символов из строка1;
- `<command> | tr -cs строка1 строка2` – выведет на экран результат команды `<command>`, заменив в нем символы из строка1 на символы из строка2 и сжав последовательности символов из строка2 (удалив дубликаты символов).

Примеры использования:

Выведет на экран текст в верхнем регистре:

```
localadmin@astra:~$ echo "text" | tr '[:lower:]' '[:upper:]'
TEXT

localadmin@astra:~$ echo "text" | tr 'a-z' 'A-Z'
TEXT
```

Выведет на экран только цифры из текста:

```
localadmin@astra:~$ echo "номер телефона 8-960-999-99-99" | tr -cd [:digit:]
89609999999
```

Выведет на экран текст, заменив каждую из последовательностей буквы r на одну букву:

```
localadmin@astra:~$ echo "Astrrrrra" | tr -s 'r'
Astra
```

Выведет на экран только буквы и цифры в тексте, а все другие символы заменит на символ новой строки:

```
localadmin@astra:~$ echo 'first second third' | tr -cs '[:alnum:]' '\n'
first
second
third
```

Выведет на экран текст, удалив из него все цифры:

```
localadmin@astra:~$ echo "1 пожалуйста 2 удалите 3 все 4 цифры" | tr -d [:digit:]
пожалуйста  удалите  все  цифры
```

Выведет на экран содержимое переменной PATH построчно, заменив разделитель „:“ на символ новой строки:

```
localadmin@astra:~$ echo $PATH | tr ':' '\n'
/usr/local/bin
/usr/bin
/bin
/usr/local/games
/usr/games
```

7.3.8 Утилита xargs

Выполнение команд с использованием данных, поступающих из стандартного потока ввода. Обычно используется в сочетании с другими командами через конвейер.

- `echo "file1 file2 file3" | xargs touch` – для каждого аргумента со стандартного ввода (file1, file2 и file3) будет выполнена команда `touch` и будут созданы (если их нет) 3 файла. Команда `touch file1 file2 file3` выполнит то же самое;
- `echo "file1 file2 file3" | xargs -t touch` – сделает то же самое, но выведет на экран команду перед её выполнением;
- `echo "file1 file2 file3" | xargs -p touch` – перед выполнением команды будет запрошено подтверждение у пользователя. Для подтверждения необходимо ввести «y»;
- `echo "file1 file2 file3" | xargs -n 1 -t touch` – ключ `-n` ограничивает количество аргументов, передающихся команде. В данном случае будут выполнены 3 команды `touch` с одним аргументом в каждой команде вместо одной команды `touch` с тремя аргументами;

- `echo "file1;file2;file3" | xargs -d \; -t touch` – ключ `-d` позволяет указать разделитель (отдельный символ или escape-последовательность) и разбить текст на отдельные аргументы;
- `xargs -t -L 1 -a ips.txt ping -c 1` – ключ `-a` позволяет указать файл, который будет использоваться вместо стандартного ввода. А ключ `-L` позволяет указать количество строк, считываемых из файла за один раз, перед передачей их команде на выполнение в качестве аргумента.
- `find ~/test* -type f -print0 | xargs -0 rm -f` – ключ print0 в команде `find` указывает ей, что необходимо вывести полный путь до найденного файла и вместо символа новой строки использовать символ null для правильной интерпретации файлов, у которых есть в именах символы новой строки или другие пробельные символы. Ключ «0» в команде `xargs` позволяет правильно интерпретировать такой вывод команды `find`.
- `echo "file1 file2 file3" | xargs -t -I % sh -c '{ touch %; ls -l %; }'` – ключ `-I` позволяет указать символ или строку, каждое вхождение которой в команде будет заменено на аргументы. Это используется для запуска нескольких команд (в данном случае `touch` и `ls` – сначала файлы будут созданы, а затем информация о них будет считана и выведена на экран), которые выполняются последовательно одна за другой.
- `echo 192.168.1.{1..254} | tr " " : | xargs -d : -L 1 -P 100 ping -c 1` – ключ `-P` позволяет распараллелить выполнение команды. Цифра указывает максимальное количество процессов, которые будут выполняться одновременно. В этом примере можно «пропинговать» все адреса хостов в сети за 3 итерации. Другой пример: параллельная загрузка файлов из интернета с помощью конвейера из команд `wget` и `xargs`.

Примеры использования:

Для каждого аргумента со стандартного ввода (file1, file2 и file3) будет выполнена команда touch и будут созданы (если их нет) 3 файла:

```
localadmin@astra:~$ echo "file1 file2 file3" | xargs touch
localadmin@astra:~$ ls file{1..3}
file1  file2  file3
```

Сделает то же самое, но выведет команду перед её выполнением:

```
localadmin@astra:~$ echo "file1 file2 file3" | xargs -t touch
touch file1 file2 file3
```

Запросит подтверждение пользователя перед выполнением:

```
localadmin@astra:~$ echo "file1 file2 file3" | xargs -p touch
touch file1 file2 file3 ?...y
```

Ключ `-n` ограничивает количество аргументов, передающихся команде:

```
localadmin@astra:~$ echo "file1 file2 file3" | xargs -n 1 -t touch
touch file1
touch file2
touch file3
```

В данном случае будут выполнены 3 команды touch с одним аргументом в каждой, вместо одной команды touch с тремя аргументами

Ключ `-d` позволяет выбрать разделитель для деления текста, поступающего на вход команде на отдельные аргументы:

```
localadmin@astra:~$ echo "file1;file2;file3" | xargs -d \; touch
```

Входной текст можно передавать не только из стандартного ввода, но и читать из файла. Ключ `-L` позволяет указать, сколько строк текста будут обрабатываться как один аргумент:

```
localadmin@astra:~$ echo -e "77.88.8.8\n77.88.8.1" > ips.txt
localadmin@astra:~$ cat ips.txt
77.88.8.8
77.88.8.1
localadmin@astra:~$ xargs -t -L 1 -a ips.txt ping -c 1
ping -c 1 77.88.8.8
PING 77.88.8.8 (77.88.8.8) 56(84) bytes of data.
64 bytes from 77.88.8.8: icmp_seq=1 ttl=246 time=36.2 ms
--- 77.88.8.8 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 36.214/36.214/36.214/0.000 ms
ping -c 1 77.88.8.1
PING 77.88.8.1 (77.88.8.1) 56(84) bytes of data.
64 bytes from 77.88.8.1: icmp_seq=1 ttl=52 time=50.7 ms
--- 77.88.8.1 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 50.739/50.739/50.739/0.000 ms
```

Ключ `-print0` в команде `find` указывает ей, что необходимо вывести полный путь до найденного файла и вместо символа новой строки использовать символ null для правильной интерпретации файлов, у которых есть в именах символы новой строки или другие пробельные символы. Ключ `-0` в команде `xargs` позволяет правильно интерпретировать такой вывод команды `find`:

```
localadmin@astra:~$ find ~/test* -type f -print0 | xargs -0 rm -f
```

Ключ `-I` позволяет указать символ или строку, каждое вхождение которой в команде будет заменено на аргументы. Это используется для запуска нескольких команд (в данном случае touch и ls – сначала файлы будут созданы, а затем информация о них будет считана и выведена на экран), которые выполняются последовательно одна за другой:

```
localadmin@astra:~$ echo "file1 file2 file3" | xargs -t -I % sh -c '{ touch %; ls -l
↪%; }'
sh -c { touch file1 file2 file3; ls -l file1 file2 file3; }
-rw-r--r-- 1 localadmin localadmin 0 июн 21 10:21 file1
-rw-r--r-- 1 localadmin localadmin 0 июн 21 10:21 file2
-rw-r--r-- 1 localadmin localadmin 0 июн 21 10:21 file3
```

Ключ `-P` позволяет распараллелить выполнение команды, цифра указывает максимальное количество процессов, которые будут выполняться одновременно. В этом примере можно пропинговать все адреса хостов в сети за 3 итерации:

```
echo 192.168.1.{1..254} | tr " " : | xargs -d : -L 1 -P 100 ping -c 1
```

7.4 Утилиты grep, sed и awk и регулярные выражения

В этом разделе представлены три дополнительных инструмента для работы с текстом:

- `grep` – утилита для поиска строк по шаблону;
- `sed` – кроме поиска, утилита позволяет преобразовывать текст, например, выполнять вставку или замену фрагментов текста;
- `awk` – самый мощный инструмент в представленном списке. Это целый язык сценариев для обработки и фильтрации текста, предлагающий множество функций, которых нет в первых двух утилитах.

Примечание: Есть еще кое-что общее между этими тремя утилитами. Все они умеют работать с регулярными выражениями. Более того, использование регулярных выражений очень существенно расширяет их возможности. Подробное изучение регулярных выражений не является темой этого курса, однако будут рассмотрены базовые приемы работы с ними.

Далее рассмотрим каждую из них подробнее.

7.4.1 Утилита grep

Название команды `grep` расшифровывается как «search **g**lobally for lines matching the **r**egular **e**xpression, and **p**rint them» (глобальный поиск строк по регулярному выражению и их вывод).

Команда `grep` предоставляет много возможностей для фильтрации текста: выбрать нужные строки из текстовых файлов, отфильтровать вывод команд и искать файлы в файловой системе, кото-

рые содержат определенные строки. Эта утилита очень популярна, поэтому предустановлена почти во всех дистрибутивах.

Два основных синтаксиса использования команды `grep`:

- `grep [<опции>] <шаблон> [/путь/к/файлу/или/папке...]`
- `<команда> | grep [<опции>] <шаблон>`

Где:

- **Опции** — это дополнительные параметры, с помощью которых указываются различные настройки поиска и вывода, например, количество строк или режим инверсии.
- **Шаблон** — это любая строка или регулярное выражение, по которому будет выполняться поиск.
- **Имя файла или папки** — это то место, где будет выполняться поиск. `Grep` позволяет искать в нескольких файлах и даже в каталоге, используя рекурсивный режим.

Основные опции утилиты `grep`:

- `-E` (`--extended-regexp`) - включить расширенный режим регулярных выражений (ERE);
- `-F` (`--fixed-strings`) - рассматривать шаблон поиска как обычную строку, а не регулярное выражение;
- `-G` (`--basic-regexp`) - интерпретировать шаблон поиска как базовое регулярное выражение (BRE);
- `-P` (`--perl-regexp`) - рассматривать шаблон поиска как регулярное выражение Perl;
- `-e` (`--regexp`) - альтернативный способ указать шаблон поиска. Опцию можно использовать несколько раз, что позволяет указать несколько шаблонов для поиска файлов, содержащих один из них;
- `-f` (`--file`) - читать шаблон поиска из файла;
- `-i` (`--ignore-case`) - не учитывать регистр символов;
- `-v` (`--invert-match`) - вывести только те строки, в которых шаблон поиска не найден;
- `-w` (`--word-regexp`) - искать шаблон как слово, отделенное пробелами или другими знаками препинания;
- `-x` (`--line-regexp`) - искать шаблон как целую строку от начала и до символа перевода строки;

- `-c` - вывести количество найденных строк;
- `--color` - включить цветной режим, доступные значения: **never, always и auto**;
- `-L` (`--files-without-match`) - выводить только имена файлов. Будут выведены все файлы, в которых выполняется поиск;
- `-l` (`--files-with-match`) - аналогично предыдущему, но будут выведены только файлы, в которых есть хотя бы одно вхождение;
- `-m` (`--max-count`) - остановить поиск, после того как будет найдено указанное количество строк;
- `-o` (`--only-matching`) - отображать только совпавшую часть вместо отображения всей строки;
- `-h` (`--no-filename`) - не выводить имя файла;
- `-q` (`--quiet`) - не выводить ничего;
- `-s` (`--no-messages`) - не выводить ошибки чтения файлов;
- `-A` (`--after-content`) - показать вхождение и n строк после него;
- `-B` (`--before-content`) - показать вхождение и n строк до него;
- `-C` - показать n строк до и после вхождения;
- `-a` (`--text`) - обрабатывать двоичные файлы как текст;
- `--exclude` - пропустить файлы, имена которых соответствуют регулярному выражению;
- `--exclude-dir` - пропустить все файлы в указанной директории;
- `-I` - пропускать двоичные файлы;
- `--include` - искать только в файлах, имена которых соответствуют регулярному выражению;
- `-r` - рекурсивный поиск по всем подпапкам;
- `-R` - рекурсивный поиск, включая ссылки.

Примеры использования:

- `grep string file` – выведет на экран строки из файла file, содержащие вхождения строки string.
- `grep -i string file` – опция -i позволяет игнорировать регистр при поиске вхождений. String, STRING, string будут эквиваленты при поиске.

- `grep -e string1 -e string2 file` – опция -e позволяет задать несколько условий для поиска.
- `grep -n string file` – опция -n позволяет выводить номера строк, в которых были найдены вхождения.
- `grep -F “special symbol” file` – опция -F позволяет явно указать утилите, что строка является строкой, а не регулярным выражением. В противном случае при использовании специальных символов (если регулярное выражение записано неверно) команда выдаст соответствующую ошибку.
- `command | grep string` – фильтрация вывода команды – выведет на экран все строки из вывода команды `command`, содержащие строку string.
- `grep -A4 string file` задать количество строк (в данном случае 4), которое будет выведено на экран, после строки с найденным вхождением. Полезно для понимания контекста, например, при поиске ошибок в логах. Опции -B и -C похожи по функционалу: -B – выводит заданное количество строк до строки с вхождением (например, -B3), а -C – строки как до, так и после строки с вхождением (например, -C2).
- `grep -r string directory` – опция -r позволяет вести поиск файлов не только в указанном каталоге, но и рекурсивно - во всех его подкаталогах.
- `grep -rl string directory` – опция -I (заглавная i) позволяет исключить из результатов поиска двоичные файлы.
- `grep -rs string directory` – опция -s позволяет скрыть ошибки доступа к файлам, если у вас недостаточно прав на их чтение.
- `grep -r --include=“string_file” string directory` – опция `--include` позволяет вести поиск только по тем файлам, которые удовлетворяют маске поиска string_file. Опция –exclude применяется аналогично, но для исключения из поиска определенных файлов.
- `grep -w string file` – опция -w позволяет искать не вхождения строки в слова, а полностью совпадающие слова, например, без опции -w строка поиска abc будет соответствовать словам aabc, abc123 и т.д., а с опцией -w только словам abc.
- `grep -c string file` – опция -c позволяет вывести количество строк с найденными вхождениями строки string.
- `grep -v string file` – опция -v позволяет вывести строки, которые не содержат вхождения string.
- `grep -lr string directory` – опция l (строчная L) позволяет вывести только имена файлов, в которых есть хотя бы одно вхождение строки string.

- `grep --color=always` – опция `--color` позволяет задать подсветку найденных вхождений в строках.

7.4.1.1 Регулярные выражения на примере утилиты grep

Регулярные выражения — это специальным образом записанные строки, используемые для поиска символьных шаблонов в тексте. Чем-то они похожи на групповые символы в оболочке, но их возможности куда шире. Многие утилиты для работы с текстом в Linux и языки программирования включают в себя механизм регулярных выражений. Существует много диалектов регулярных выражений, однако большинство утилит в Linux соответствуют стандарту регулярных выражений POSIX. Их мы и рассмотрим.

Существуют два вида реализации регулярных выражений в стандарте POSIX – базовый Basic Regular Expression (BRE) и расширенный Extended Regular Expression (ERE). BRE поддерживают практически все POSIX-совместимые программы, а ERE позволяет создавать более сложные регулярные выражения, но поддерживается не всеми утилитами, однако grep поддерживает оба этих вида.

Простые регулярные выражения

В простых регулярных выражениях применяются следующие метасимволы: ^ $. [ ] * \ -. Метасимволы – это категория символов для составления критериев поиска в регулярных выражениях. Простые символы принято называть литералами, например, выражение «dir» обозначает строку, соответствующую следующим критериям: в строке не меньше трех символов; в строке присутствуют символы «d», «i», «r», причем именно в таком порядке; между ними нет других символов. Символы, соответствующие сами себе (как «d», «i», «r»), и есть литералы. Для использования метасимвола в качестве литерала его необходимо экранировать обратной косой чертой (\). Некоторые метасимволы в регулярных выражениях имеют специальное значение в оболочке, поэтому при использовании их в регулярных выражениях строку с ними необходимо заключать в кавычки.

Метасимвол «.» соответствует любому символу в этой позиции (но не null или «пустота»). Пример использования: `ls /bin | grep '.dir'`.

Метасимвол звездочка «*» после другого символа в шаблоне означает, что этот символ может повторяться любое количество раз, в том числе и ни одного раза. Пример использования: `ls /bin | grep 'mk*dir'`.

Метасимволы «^» и «$» выполняют роль якорей. Якоря позволяют искать вхождение шаблона не просто в строке, а явно указать, что вхождение должно быть в начале строки (символ «карет» (^)) или в конце строки (символ «доллар» ($)). Регулярное выражение «^$» соответствует пустым строкам. Примеры использования: `ls /bin | grep '^dir'`; `ls /bin | grep 'dir$'`.

Метасимволы квадратные скобки «[]» позволяют указать множество, любой член которого будет удовлетворять символу в этой позиции. Например, `ls /bin | grep '[km]dir'`.

Все метасимволы, кроме двух, в квадратных скобках теряют свои специальные значения.

- Метасимвол «^» сразу после «[» добавляет логическое «не», т.е. символу на этой позиции будет соответствовать любой символ, не входящий в множество, указанное в квадратных скобках. Например, `ls /bin | grep '[^km]dir'`. Если символ «^» стоит в скобках не в начале, то он интерпретируется как обычный символ.
- Метасимвол дефис «-» определяет диапазоны символов. Можно задать диапазоны букв или цифр. Например, `ls /bin | grep '[o-z]dir'`; `ls /dev | grep 'tty1[4-7]'`.

В POSIX есть несколько классов, описывающих разные множества символов.

- **[:alnum:]** — Алфавитно-цифровые символы; эквивалент диапазона [A-Za-z0-9] в ASCII.
- **[:alpha:]** — Алфавитные символы; эквивалент диапазона [A-Za-z] в ASCII.
- **[:digit:]** — Цифры от 0 до 9.
- **[:lower:]** и **[:upper:]** — Символы нижнего и верхнего регистра соответственно.
- **[:space:]** — Пробельные символы, включая пробел, табуляцию, возврат каретки, перевод строки, вертикальную табуляцию и перевод формата.

Расширенные регулярные выражения

В расширенных регулярных выражениях применяются следующие метасимволы:

- Круглые скобки: ( );
- Фигурные скобки: { };
- Вопросительный знак: ?;
- Символ плюса: +;
- Вертикальная черта: |.

Для использования расширенных регулярных выражений утилитой `grep` необходимо использовать ключ `-E`.

Метасимвол вертикальная черта «|» означает логическое ИЛИ. Он позволяет задать два набора символов в шаблоне, вхождения каждого из которых будут найдены. Например:

- Команда `echo "123" | grep -E '123|345'` выведет на экран строку 123, так как подстрока 123 указана в параметре `-E`;
- Команда `echo "345" | grep -E '123|345'` выведет на экран строку 345, так как подстрока 345 указана в параметре `-E`;
- Команда `echo "234" | grep -E '123|345'` ничего не выведет на экран, так как подстрок 123 и 345 нет в строке.

Метасимволы круглые скобки «()» применяются для группировки условий.

Например, `ls /dev | grep -E 'tty1(1|2|3)'` выведет все строки, содержащие tty11, tty12 или tty13, а то же выражение без скобок `ls /dev | grep -E 'tty11|2|3'` выведет строки, содержащие или tty11, или 2, или 3.

Метасимвол вопросительный знак «?» после другого символа в шаблоне, означает совпадение с этим символом ноль или один раз. Другими словами, или этот символ есть в тексте, или его может не быть, например:

- Команда `echo "13" | grep -E '12?3'` выведет на экран строку 13, так как в параметре -E символ 2 не является обязательным;
- Команда `echo "123" | grep -E '12?3'` выведет на экран строку 123, так как в параметре -E символ 2 допустим;
- Команда `echo "1223" | grep -E '12?3'` ничего не выведет, так как символ 2 допустим только один раз.

Метасимвол знак плюс «+» после другого символа в шаблоне означает совпадение с этим символом один или более раз. Похож на символ «*», однако символ «*» допускает ни одного совпадения, а символ «+» минимум одно. Пример использования: `ls /bin | grep -E 'mk+dir'`.

Метасимволы фигурные скобки «{}» позволяют задать количество совпадений символа, после которого они идут:

- **{n}** — Совпадение, если предыдущий элемент встречается точно n раз.
- **{n,m}** — Совпадение, если предыдущий элемент встречается не менее n и не более m раз.
- **{n,}** — Совпадение, если предыдущий элемент встречается n или более раз.
- **{,m}** — Совпадение, если предыдущий элемент встречается не более m раз.

Примеры использования:

```
$ echo "13" | grep -E '12{1,3}3'
```

(continues on next page)

(продолжение с предыдущей страницы)

```
$ echo "123" | grep -E '12{1,3}3'
123

$ echo "1223" | grep -E '12{1,3}3'
1223

$ echo "12223" | grep -E '12{1,3}3'
12223

$ echo "122223" | grep -E '12{1,3}3'
```

7.4.2 Утилита sed

Sed – это потоковый редактор текста (*от англ. stream editor*), с помощью которого можно выполнять множество операций с файлами, например, поиск и замену, вставку или удаление. При этом чаще всего он используется именно для поиска и замены.

Sed позволяет редактировать файлы не открывая, что существенно ускоряет работу, чем при использовании того же vi. Помимо этого, sed поддерживает регулярные выражения, с помощью которых можно выполнять сложное сопоставление шаблонов.

Формат утилиты: sed [<параметры>] '<правила_отбора> <инструкции>' [файлы]

Параметры:

- -E – использовать расширенный вариант регулярных выражений. При этом регулярное выражение в правилах отбора обрамляется символами слэша /регулярное выражение/.
- -e – используется, если в команде необходимо указать несколько инструкций. Он размещается перед каждой инструкцией.
- -i (--in-place) - производить изменения в указанном файле.
- -n – запрет вывода всех строк (по умолчанию выводятся все строки).

Правила отбора:

- В соответствии с номером строк – 1,5 – с первой по 5 строки, 5,$ - с пятой по последнюю строки.
- В соответствии с регулярным выражением.

Инструкции:

- n – запрет вывода всех строк (по умолчанию выводятся все строки)

- `p` – вывод строк, попадающих под критерии отбора. Часто применяется с -n
- `d` – удаление строк
- `i'текст'` – вставка текста перед строкой, попадающей под критерии отбора
- `a'текст'` – вставка текста после строки, попадающей под критерии отбора
- `c'текст'` – замена строки, попадающей под критерии отбора на текст
- `s/строка_поиска/строка_замены/` – поиск и замена текста

Полную информацию по утилите `sed` можно найти в справке `man sed`.

Примеры использования:

Вывод первых 5 строк из файла `/etc/passwd`:

```
localadmin@astra:~$ cat /etc/passwd | sed -n '1,5 p'
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
```

Вывод 1 и 2 строк (удаление всех других) из файла `/etc/passwd`.

```
localadmin@astra:~$ cat /etc/passwd | sed '3,$ d'
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
```

Вывод строк с 990 по 999:

```
localadmin@astra:~$ cat /etc/passwd | sed -n -E '/99[0-9]/ p'
systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
vboxadd:x:998:1::/var/run/vboxadd:/bin/false
```

Вставка текста перед найденной строкой:

```
localadmin@astra:~$ cat /etc/passwd | sed -e "/root/i Следующая строка содержит␣
↪данные пользователя root:" | head -n5
Следующая строка содержит данные пользователя root:
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
```

Вставка текста после найденной строки:

```
localadmin@astra:~$ cat /etc/passwd | sed -e "/root/aПредыдущая строка содержит␣
↪данные пользователя root" | head -n5
root:x:0:0:root:/root:/bin/bash
Предыдущая строка содержит данные пользователя root
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
```

Вставка текста до и после найденной строки:

```
localadmin@astra:~$ cat /etc/passwd | sed -e "/root/i----------------------" -e "/
↪root/a | ----------------------" | head -n5
----------------------
root:x:0:0:root:/root:/bin/bash
----------------------
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
```

Замена найденного текста в строке:

```
localadmin@astra:~$ cat /etc/passwd | sed -n '/localadmin/ p'
localadmin:x:1000:1000:localadmin,,,:/home/localadmin:/bin/bash
localadmin@astra:~$ cat /etc/passwd | sed -e 's#/bash#/sh#' | grep localadmin
localadmin:x:1000:1000:localadmin,,,:/home/localadmin:/bin/sh
```

7.4.3 Утилита awk

AWK – это скриптовый язык, который полезен при работе в командной строке и широко применяется для обработки текста. Обработка текста идет построчно.

При использовании awk вы можете выбирать данные – один или более отдельных фрагментов текста – на основе заданного критерия отбора. Например, с помощью awk можно выполнять поиск конкретного символа, слова или шаблона во фрагменте текста, а также выбирать определённую строку или столбец. Возможности awk позволяют проводить сложную обработку текстов, составлять отчеты, проводить анализ информации и составлять тексты команд для последующего их выполнения интерпретатором.

Формат утилиты: `awk [<параметры>] '<правила_отбора> {<действия>}' [<файлы>]`

Параметры:

С помощью `awk` можно работать со строками как с полями строк в таблице. Для этого с помощью параметра `-F` можно задать разделитель, например, `-F ': '`, и обращаться к разделенным столбцам через встроенные переменные: $0 – вся строка, $1 – первое поле, $2 – второе поле и так далее. В переменной NF хранится значение количества полей в строке, а NR – хранит номер строки.

Действия:

В качестве действия могут использоваться команды вывода текста, такие как `printf` или `print`.

Можно использовать следующие правила отбора:

Регулярные выражения – обрамляются слэшами.

Операции сравнения – используется следующая конструкция: {переменная_номера_поля}{операнд}{значение}, например, $1<10. В качестве операндов используются:

- Арифметические (сравнения): - >, >=, <, <=, ==, !=
- Строковые (соответствия): ==, !=
- Регулярные выражения (соответствия): ~, !~

Примеры использования:

Вывод имен пользователей с uid от 990 до 999:

```
localadmin@astra:~$ cat /etc/passwd | awk -F ':' '/99[0-9]/{print$1}'
systemd-coredump
vboxadd
```

Вывод имен, uid и оболочки пользователей с uid от 10 до 20:

```
localadmin@astra:~$ cat /etc/passwd | awk -F ':' '$3>10 && $3 < 20 {print $1,$3,$NF}'
proxy 13 /usr/sbin/nologin
```

Вывод id и команды запуска всех процессов, принадлежащих пользователю:

```
localadmin@astra:~$ ps aux | awk '$1~/localad/{print $2,$11}'
1084 /lib/systemd/systemd
1085 (sd-pam)
1107 fly-wm
1172 /usr/bin/dbus-launch
1173 /usr/bin/dbus-daemon
1196 /usr/bin/kglobalaccel5
1202 /usr/bin/VBoxClient
...
```

Полную информацию по утилите `awk` можно найти в справке `man awk`.

7.5 Текстовые редакторы

7.5.1 Текстовый редактор vim

Vim — это текстовый редактор, который Windows-пользователям покажется крайне необычным. Больше всего начинающих администраторов сбивает с толку то, что в этом редакторе используются несколько режимов ввода, основными из которых являются «командный» и «текстовый».

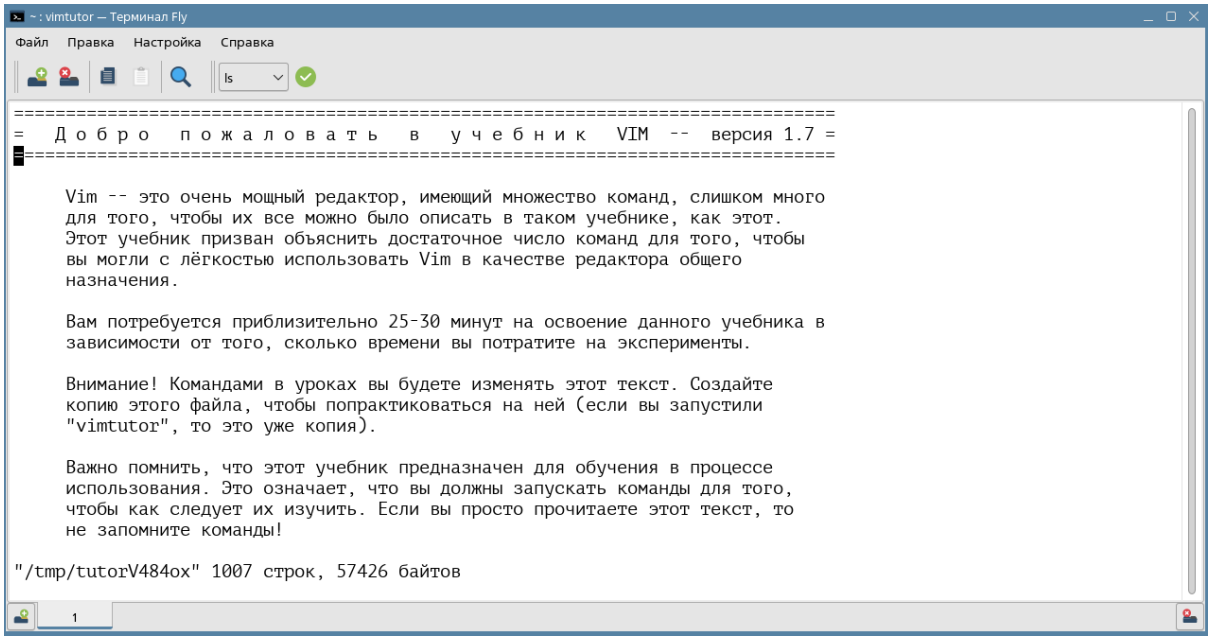
Приведем краткую инструкцию по применению:

- Выполните команду `vi test.txt`, чтобы открыть файл `test.txt` для редактирования.
- Сразу после запуска приложения вы находитесь в режиме ввода команд и можете, например, ввести команду выхода``:q`` затем нажмите Enter, чтобы закрыть программу.
- Для редактирования документа вам потребуется переключиться в текстовый режим с помощью клавиши Insert. Нажимая клавишу Insert несколько раз, вы можете выбрать подрежим вставки или замены.
- Для того чтобы скопировать текст, просто выделите его с помощью указателя мыши. Для того чтобы вставить текст из этого буфера, наведите указатель мыши в желаемое место и нажмите на колесико мыши. Вы можете так же использовать более привычный буфер обмена, для вставки которого используются горячие клавиши Ctrl + V — для этого при выделении текста удерживайте клавишу Shift, а для копирования вставки команды из контекстного меню, которое доступно по нажатию правой кнопки мыши.
- Чтобы вернуться обратно в режим команд и сохранить изменения, вам нужно нажать клавишу Esc и ввести команду `:wq`.
- Для выхода без сохранения используйте команду `:q!`.

За счёт командных функций Vim существенно повышает скорость работы с текстами, поэтому его выбирают профессионалы, которым требуется высокая производительность в работе из консоли. Однако, довести необходимые умения до автоматизма будет непросто.

7.5.1.1 Учебник vimtutor

После установки редактора Vim у вас появляется доступ к встроенному учебнику по работе с приложением, который можно открыть с помощью команды `vimtutor`. При этом открывается копия файла, поэтому вы можете практиковаться, не опасаясь повредить содержание учебника. Освоение Vim крайне желательно начать с изучения материалов этого пособия.



Далее будут рассмотрены основные приемы работы с Vim.

Запуск, сохранение и завершение работы с Vim

Запустить Vim не сложно, сложно из него выйти. Для запуска достаточно вызвать команду `vim [опции] [имя_файла]`.

Ниже перечислены некоторые опции запуска:

- `-R` – открыть в режиме «только чтение»;
- `-g` – включить графический режим;
- `-w` - сохранить все действия в файл;
- `-b` – воспользоваться двоичным режимом, чтобы редактировать исполняемые файлы;
- `-d` – найти различия в файлах (потребуется указать несколько файлов для открытия);
- `-n` – отключить функцию автоматического сохранения для восстановления файла при ошибке;
- `«+команда»**` — эта опция отвечает за выполнение команды после запуска редактора;
- `+номер` — перемещение курсора к определённой пользователем строке после запуска;
- `+/шаблон` — найти нужный участок в шаблоне и переместить курсор к его началу.

Запустим `vim vim_test.txt`. Сейчас мы находимся в командном режиме. Чтобы выйти из Vim,

необходимо ввести одну из команд:

- `:q` – выйти из редактора, если не было произведено изменений. В противном случае он не даст выйти этим способом.
- `:q!` – выход без сохранения. Более быстрый и удобный способ ввести **ZQ**.
- `:qw` – сохранить и выйти. Более быстрый и удобный способ ввести **ZZ**.

Команда `:w` просто сохранит текущие изменения, и вы сможете продолжить редактирование файла.

Работа в командном режиме

Введенные в этом режиме символы воспринимаются как команды. Можно повторять команды несколько раз, вводя команду и цифру, или уточнять область действия команд. Например, ввод `dl` удалит символ под курсором, `dw` удалит слово целиком, а `d3w` удалит 3 слова справа от курсора.

Но следует понимать, что у вас при этом должна быть включена английская раскладка.

Команды навигации по тексту:

- `h`, `l` , `j` или `k` – перемещают курсор влево/вправо и вверх/вниз, однако проще будет использовать стрелочки мыши;
- `w` или `b` – переместить курсор на 1 слово влево или вправо;
- `H` – уйти в верхнюю часть экрана;
- `L` – уйти в нижнюю часть экрана;
- `G` – пролистать до конца файла;
- `zz`, `zt`, `zb` – перемещать по экрану строку с курсором в середину/вверх/вниз;
- `Ctrl + u` или `Ctrl+d` – прокрутить область просмотра в верхнем или нижнем направлении на полстраницы.

Команды для вставки и копирования:

- `p` или `P` – вставить текст после курсора или перед ним;
- `y` – скопировать символ;
- `yy` или `Y` – скопировать целую строку.

Команды для отмены и удаления:

- u – отменить последнее действие;
- U – отменить последнее действие в указанной строке, где стоит курсор;
- d – удалить символ;
- dd – удалить строку целиком;
- D – удалить всё, что находится на участке от курсора до конца строки;
- dt – удалить часть текста, которая находится между местом курсора и следующим вхождением конкретного символа;
- ciw – удалить конкретное слово и сразу переключиться в режим ввода;
- C – удалить текст, начиная от места курсора и до конца строки, затем снова автоматически происходит переход в режим ввода.

Ещё несколько полезных команд:

- v – выделить текст;
- ggvG= – выровнять текст во всём файле (переместиться в начало, переключиться на режим VISUAL, выделить весь текст документа, настроить отступ выделенных строк);
- . – повторить ещё раз последнюю команду;
- n или N – продолжить поиск вперёд или назад;
- ~ – переключить регистр у символа, который выделен или на котором находится курсор;
- /шаблон – искать вхождение;
- %s/шаблон/заменить – заменить первое слово вторым.

Работа в текстовом режиме

Для переключения в текстовый режим, в котором введенные символы будут восприниматься как набираемый текст, нужно нажать клавишу Insert. Повторное нажатие Insert позволяет переключаться между текстовыми подрежимами вставки и замены.

Для возврата в командный режим достаточно нажать клавишу Esc.

Команды для перехода в текстовый режим:

- I (строчная буква i) – переключиться в текстовый режим и перейти в начало строки;
- A – переключиться в текстовый режим и перейти в конец строки;
- a – переключиться в текстовый режим и перейти к следующему символу;
- r – заменить один символ;
- R – заменить несколько символов подряд;
- o – создать строку снизу;
- O – создать строку выше.

Режим командной строки

В режиме командной строки ввод команды начинается с ввода символа «:». Мы уже познакомились с ним, когда рассматривали завершение работы и сохранение файла. Ниже представлены некоторые другие команды этого режима:

- `:e файл` – создать новый файл;
- `:r файл` – вставить содержимое этого файла в текущий файл;
- `:r!` – задать выполнение команды оболочки и вставку полученного ответа в Vim;
- `:buffers` – просмотреть открытые файлы;
- `:set переменная=значение` – задать значение переменной для управления процессами редактора.

Визуальный режим

В этом режиме с помощью команд навигации можно выделить часть текста (или весь текст) для его дальнейшего редактирования. Это удобно, так как пользователь видит выделенный участок.

Использование визуального режима сводится к выполнению трёх операций:

1. Находясь в командном режиме, нажмите горячую клавишу v, чтобы начать выделение текста.
2. Переместите курсор в конец визуального фрагмента для выделения текста.
3. Введите команду-оператор, которая выполнит действия над выделенным текстом. Например, если вы напечатаете символ ~, то будет изменен регистр символов и весь текст станет про-

писными буквами.

Приведем полезные команды-операторы:

- y — скопировать текст в буфер обмена
- p — вставить текст из буфера обмена
- ~ — изменить регистр символов
- d — удалить выделенный текст

Кроме посимвольного режима v есть еще построчный V и режим блоков Ctrl + V. С дополнительной и информацией о Vim вы можете ознакомится в справке `man vim` и по ссылке https://ru.wikibooks.org/wiki/Vim

7.5.2 Текстовый редактор nano

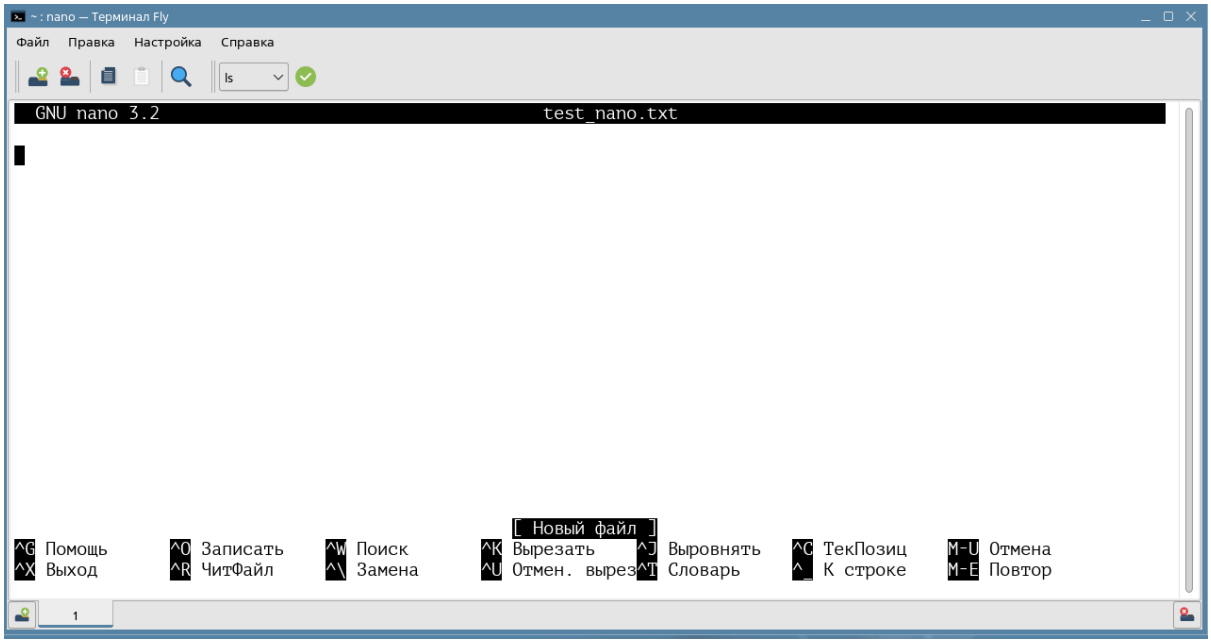
Текстовый редактор nano входит в набор программ GNU и является стандартным консольным редактором для многих дистрибутивов Linux. Главное преимущество nano заключается в простоте использования — работать с ним так же просто, как в notepad.exe, поэтому у вас получится довольно быстро освоить его на высоком уровне для решения любых задач.

Для запуска достаточно вызвать команду `nano [опции] [имя_файла]`.

Некоторые из опций:

- `+номер` - указать строку в файле, с которой нужно начинать редактирование;
- `-B` - создавать резервную копию для файла при сохранении;
- `-C` - папка для резервной копии;
- `-D` - выводить текст жирным;
- `-E` - конвертировать табуляции в пробелы;
- `-I` - не читать конфигурацию из файлов nanorc;
- `-P` - запоминать и восстанавливать позицию курсора во время последнего редактирования;
- `-m` - включить поддержку мыши;
- `-v` - режим только для чтения, вы не можете сохранить файл.

Запустим текстовый редактор `nano test_nano.txt`.



В отличие от Vim, тут нет особого режима, можно перемещаться по тексту и сразу редактировать его. В нижней части окна показана подсказка по командам, которая может меняться в зависимости от текущего состояния. В комбинациях используются связки клавиш Ctrl или Alt с другими клавишами. Напоминаем, что символ ^ является кратким обозначением клавиши Ctrl.

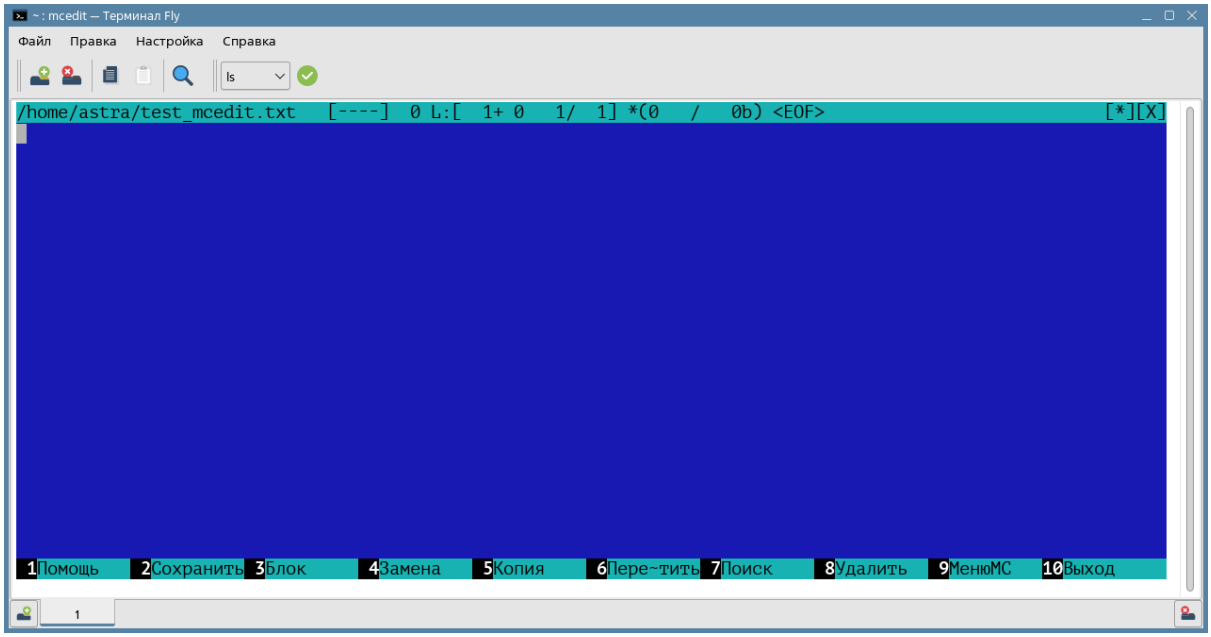
Полный список команд доступен по нажатию клавиши F1 или сочетания Ctrl +G.

Редактор nano может поддерживать поиск по регулярным выражениям. Для этого в конфигурационном файле в профиле пользователя ~/.nanorc должен быть установлен параметр set regexp.

В редакторе можно включить подсветку синтаксиса для популярных языков программирования. Для этого в конфигурационный файл ~/.nanorc с помощью директивы include нужно включить файлы с настройками подсветки, находящиеся в /usr/share/nano. Например, для подсветки sh нужно добавить строку «include /usr/share/nano/sh.nanorc».

7.5.3 Текстовый редактор mcedit

Редактор mcedit устанавливается вместе с файловым менеджером Midnight Commander и имеет схожий интерфейс.

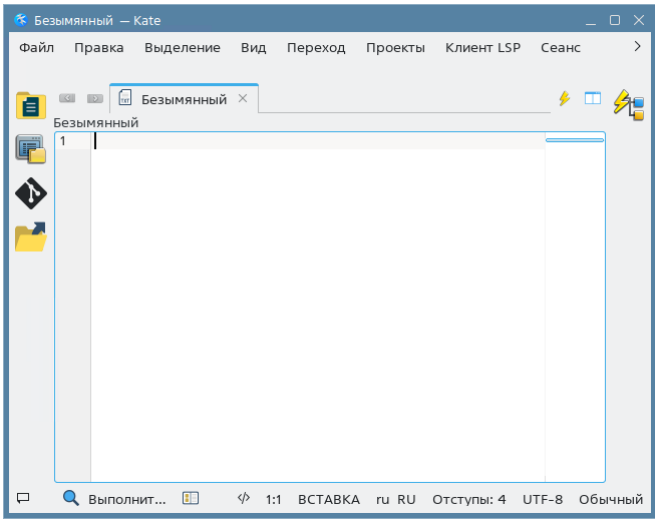


В нижней строке перечислены базовые команды, которые выполняются с помощью функциональных клавиш F1-10. Например, справка вызывается по нажатию F1, а основное меню клавишей F9.

Как и nano, редактор mcedit поддерживает работу с регулярными выражениями и подсветку синтаксиса популярных языков программирования и конфигурационных файлов. Для ручного управления подсветкой можно воспользоваться настройкой *Меню (F9) → Настройки → Цветовыделение синтаксиса*.

7.5.4 Графический текстовый редактор kate

Графический текстовый редактор kate имеет интуитивно понятный интерфейс и предоставляет большие возможности по настройке. По функциональности он очень похож на notepad++, т.е. сильно больше блокнота, но меньше Visual Studio Code.



Как и упомянутые выше редакторы, kate поддерживает работу с регулярными выражениями и подсветку синтаксиса языков программирования и конфигурационных файлов. Рекомендован к использованию новичкам (и не только) в графическом режиме.

7.6 Практика и тестирование

7.6.1 Практическая работа: Модуль 7. Работа с текстовой информацией

См. также *Требования, правила и цели выполнения практической работы*

7.6.1.1 Практические задания

В рамках этого модуля мы предлагаем выполнить расширенную лабораторную работу.

Задание 1.

Создайте текстовый файл sample со следующим содержанием:

```
Добрый день, Петр Васильев!

Рамки и место обучения  кадров представляют собой интересный эксперимент
проверки дальнейших направлений развития см. Рис 10. Идейные соображения
высшего порядка, а также  консультация с широким активом способствуют
подготовке и реализации дальнейших направлений развития по стандарту
ISO-9001. Не следует, однако, забывать, что рамки и место обучения кадров
способствуют подготовке и реализации модели развития.
Повседневная практика показывает, что укрепление и развитие структуры
требуют определения и уточнения системы обучения кадров, соответствует
насущным потребностям.
```

Задание 2.

Создайте еще один файл «sign» с подписью:

```
Пишите нам на email: info@company.lan
Мобильный: +79881112230
Офисный многоканальный: 8 800 2201100
Сайт: www.company.lan
```

Задание 3.

Выведите текст письма в стандартный поток вывода.

Задание 4.

Выведите содержимое двух файлов на экран подряд одной командой.

Задание 5.

Найдите количество строк в файле.

Задание 6.

Выведите первую строку из файла sample.

Задание 7.

Найдите имя в первом предложении и удалите у имени пробел.

Задание 8.

Выведите содержимое файлов «sample» и «sign» на экран, пронумеруйте строки при выводе.

Задание 9.

Объедините содержимое файлов «sample» и «sign» и запишите его в новый файл «message».

Задание 10.

Выведите на стандартный поток вывода только последнюю строку файла «message».

Задание 11.

Выведите на стандартный поток вывода номер рисунка из файла «message». Используйте для этого команду `grep` и регулярное выражение.

Задание 12.

Выведите на стандартный поток вывода адрес электронной почты из файла «message».

Подсказка: Будем считать, что адрес email состоит только из букв, цифр, дефисов, точек, подчёркиваний и знака «@». Поиск всех возможных вариантов адреса электронной почты может оказаться весьма непростой задачей: https://stackoverflow.com/questions/46155/how-can-i-validate-an-email-address-in-javascript/1373724#1373724

Задание 13.

Выведите на стандартный поток вывода номер *8 800* из файла «message».

Задание 14.

Выведите на стандартный поток вывода интернациональные телефонные номера из файла «message».

Задание 15.

Удалите два и более пробелов из файла «message».

7.6.1.2 Ответы на практические задания (пошаговые инструкции)

Задание 1.

Откроем файл sample в редакторе nano командой `nano letter` и наполним его следующим содержанием:

```
Добрый день, Петр Васильев!

Рамки и место обучения кадров представляют собой интересный эксперимент
проверки дальнейших направлений развития см. Рис 10. Идейные соображения
высшего порядка, а также консультация с широким активом способствуют
```

(continues on next page)

(продолжение с предыдущей страницы)

```
подготовке и реализации дальнейших направлений развития по стандарту
ISO-9001. Не следует, однако, забывать, что рамки и место обучения кадров
способствуют подготовке и реализации модели развития.
Повседневная практика показывает, что укрепление и развитие структуры
требуют определения и уточнения системы обучения кадров, соответствует
насущным потребностям.
```

Задание 2.

Создадим еще один файл sign с подписью `nano sign`.

```
Пишите нам на email: info@company.lan
Мобильный: +79881112230
Офисный многоканальный: 8 800 2201100
Сайт: www.company.lan
```

Задание 3.

Выведем текст письма в стандартный поток вывода, который затем печатается в консоли.

```
cat letter
```

Задание 4.

Выведем оба файла на экран подряд с помощью команды `cat`, которая склеивает файлы:

```
cat letter sign
```

Задание 5.

Найдем количество строк в файле.

```
cat letter sign | wc -l
```

Задание 6.

Найдем первую строку в файле sample.

```
cat letter | head -1
```

Задание 7.

Найдем имя в первом предложении и удалим у имени пробел:

```
cat letter | head -1 | cut -d',' -f 2 | xargs
```

Задание 8.

Сгруппируем стандартные выводы.

```
(cat letter; echo -e "\n__"; cat sign)
```

Пронумеруем вывод и посмотрим, как прошло слияние группировкой.

```
(cat letter; echo -e "\n__"; cat sign) | nl
```

Такую технику можно использовать для генерации заголовков колонок, а затем данных.

```
(echo "col1;col2;col3"; echo "val2;val2;val2"; echo "val3;val3;val3") | column -t -s';
↪'
```

Задание 9.

Данная группировка позволяет объединить несколько текстовых блоков для дальнейшей обработки.

```
(cat letter; echo -e "\n__"; cat sign) | tee message
```

Задание 10.

Найдем последнюю строку в файле message.

```
cat message | tail -1
```

Команда `tail` может следить за логами `tail -f`.

```
sudo tail -f /var/log/auth.log
```

a. Нажмем пару раз <Enter> для разделения информации (для разделения старой информации от новой).

b. Откроем новую вкладку терминала ALT+T.

c. Перейдем в сессию root `sudo -i`.

d. Закроем новую сессию командой `exit`.

e. Мы увидим изменение в первом окне команды `tail -f`.

f. Завершим `tail -f` сочетанием клавиш Ctrl+C.

Команда `tee` создает и выводит файл, удобна для создания конфигурационных файлов с применением sudo.

```
(echo "[main]"; echo 'key1=value'; echo "key2=value") | sudo tee /etc/test.ini
```

Задание 11.

Вернемся к сообщению и найдем номер рисунка:

```
cat message | grep -Eo 'Рис\s[0-9]{1,10}' | cut -d' ' -f 2
```

Задание 12.

Вернемся к сообщению и найдем email:

```
cat message | grep -Eo '([a-zA-Z0-9._-]+@[a-zA-Z0-9._-]+\.[a-zA-Z0-9_-]+)'
```

В качестве вспомогательного инструмента воспользуйтесь https://regex101.com/r/hrIiGz/1.

Задание 13.

Вернемся к сообщению, найдем номер 8 800.

```
cat message | grep -Eo '8\s800\s[0-9]{1,10}'
```

В качестве вспомогательного инструмента воспользуйтесь https://regex101.com/r/2dURa1/1.

Задание 14.

Найдем интернациональные телефонные номера:

```
cat message | grep -Eo '\+?[1-9][0-9]{7,14}'
```

В качестве вспомогательного инструмента воспользуйтесь https://regex101.com/r/cCfkrI/1.

Задание 15.

Удалим два и более пробелов.

```
cat message | sed 's/ \{1,\}/ /g'
```

В качестве вспомогательного инструмента воспользуйтесь https://regex101.com/r/p2YkJB/1.

7.6.1.3 Дополнительные упражнения

Конфигурационные файлы

- Выведем конфигурацию локальной петли 127.

```
cat /etc/hosts | grep ^127
```

- Видим, что разделитель пробел и можно вывести полное имя.

```
cat /etc/hosts | grep ^127 | cut -d' ' -f 2
```

Команда `grep` нашла ip 127, а `cut` вырезала второе поле по разделителю пробел.

- Посмотрим локализацию по умолчанию.

```
cat /etc/default/locale | cut -d"=" -f 2 | tr -d "\""
```

- Удалим комментарий с символа #.

```
cat /etc/default/locale | grep -v '#' | cut -d"=" -f 2 | tr -d "\""
```

Командой `tr` мы удалили кавычки для дальнейшей работы.

Текстовые файлы с разделителем CSV

Создадим в Libreoffce Calc таблицу.

табл. 1.2 – Содержимое users.csv

Пользователь	Имя	Фамилия
ivani	Иван	Иванов
ivani	Иван	Иванченко
alexa	Алексей	Антонов

Сохраним как ~/users.csv с разделителем запятые.

- Выведем файл cat users.csv. Должен получиться такой файл:

```
Пользователь,Имя,Фамилия
ivani,Иван,Иванов
ivani,Иван,Иванченко
alexa,Алексей,Антонов
```

- Выведем файл в табличном виде.

```
cat users.csv | column -t -s,
```

- Выведем заголовок с названиями колонок.

```
grep $(head -1 users.csv) users.csv
```

- Выведем уникальных и отсортированных пользователей, которых необходимо создать.

```
grep -v $(head -1 users.csv) users.csv | cut -d"," -f 1 | sort | uniq
```

- Запустим команду `xargs` для создания локальных пользователей по списку.

```
grep -v $(head -1 users.csv) users.csv | cut -d"," -f 1 | sort | uniq | xargs -I {}␣
↪sudo useradd {}
```

- Проверим наличие локальных пользователей.

```
grep -v $(head -1 users.csv) users.csv | cut -d"," -f 1 | uniq |
xargs -I {} grep {} /etc/passwd
```

Данные в Markup Language

Формат XML

Для работы xml нужно установить xpath `sudo apt install curl libxml-xpath-perl`.

Документация по xpath https://www.w3.org/TR/xpath-30/.

- Поработаем с **https://www.cbr-xml-daily.ru/daily_utf8.xml**.
    - Скачаем текущий курс на сайте.

```
curl -s https://www.cbr-xml-daily.ru/daily_utf8.xml > daily.xml
```

Утилита `curl` получает данные из интернета по URL адресу. Их можно сохранить для последующей обработки.

- Выведем узел с ID = R01235.

```
xpath -q -e "//Valute[@ID='R01235']" ./daily.xml
```

- Выведем курс доллара в рублях.

```
xpath -q -e "//Valute[@ID='R01235']/Value/text()" ./daily.xml
```

где -e «//Valute[@ID=“R01235“]/Value/text()» это запрос XPath, который применяется для поиска данных в форматах xml/html

Формат JSON

Для работы с JSON потребуется установить утилиту `jq` с помощью команды `sudo apt install jq`.

Документация по утилите `jq` https://jqlang.github.io/jq/manual/.

- Выведем курс Доллара.

```
curl -s https://www.cbr-xml-daily.ru/latest.js | jq -r ".rates.USD"
```

- Список всех валют.

```
curl -s https://www.cbr-xml-daily.ru/latest.js | jq -r ".rates | to_entries[] | .key"
```

- Список всех валют и значений.

```
curl -s https://www.cbr-xml-daily.ru/latest.js | jq -r ".rates | to_entries[] | .key,␣
↪.value"
```

Формат LDIF

- Создадим пример данных `nano users.ldif`.

```
dn: uid=admin,cn=users,cn=accounts,dc=ald,dc=company,dc=lan
uid: admin
sn: Administrator

dn: uid=oolegov,cn=users,cn=accounts,dc=ald,dc=company,dc=lan
uid: oolegov
sn:: 0J7Qu9C10LPQvtCy

dn: uid=yanaz,cn=users,cn=accounts,dc=ald,dc=company,dc=lan
uid: yanaz
sn:: 0JfRg9Cx0LjQvdCw
```

Скопируем с пустой строкой в конце файла, т. к. это разделитель записи в LDIF формате.

- Найдем запись uid=oolegov из users.ldif файла.

```
cat users.ldif | sed -ne '/^dn: uid=oolegov/,/^$/p'
```

- Найдем атрибут uid.

```
cat users.ldif | sed -ne '/^dn: uid=oolegov/,/^$/p' | grep ^uid
```

- Выведем значение uid oolegov.

```
cat users.ldif | sed -ne '/^dn: uid=oolegov/,/^$/p' | grep '^uid' | cut -d' ' -f 2
```

- Выведем фамилию из поля sn, применив base64 -d.

```
cat users.ldif | sed -ne '/^dn: uid=oolegov/,/^$/p' | grep '^sn' | cut -d' ' -f 2 |␣
↪base64 -d
```

Формат YAML

- Установим бинарный файл yq для архитектуры linux amd64.

```
sudo wget https://github.com/mikefarah/yq/releases/latest/download/yq_linux_amd64 -O /
↪usr/bin/yq && sudo chmod +x /usr/bin/yq
```

Документация по утилите https://mikefarah.gitbook.io/yq/.

- Создадим файл task.yaml.

```
- name: Пример переменной Playbook
  hosts: all
  vars:
   login: ivani

- tasks:
  - name: Добавить пользователя {{ login }}
    ansible.builtin.user:
      - name: "{{ login }}"
        state: present
```

- Выведем название playbook.

```
yq '.[0].name' task.yaml
```

- Найдем элементы, содержащие .hosts со значением all.

```
cat task.yaml | yq 'filter(.hosts == "all")'
```

7.6.2 Тест: Модуль 7. Работа с текстовой информацией

Пожалуйста, ответьте на несколько вопросов, чтобы закрепить полученные знания. Правильные ответы для самопроверки вы сможете найти в конце списка. Обратите внимание, что у некоторых вопросов может быть несколько верных вариантов ответов.

7.6.2.1 Вопросы

Вопрос 1. Какая команда позволяет вывести в стандартный поток STDOUT один или более файлов по порядку?

a. Команда `tr`

b. Команда `sed`

c. Команда `grep`

d. Команда `cat`

Вопрос 2. Какая команда позволяет просмотреть несколько строк из конца файла?

a. Команда `tab`

b. Команда `head`

c. Команда `tail -n кол-во`

d. Команда `wc`

Вопрос 3. Какая команда позволяет построчно отфильтровать логи?

a. Команда `sed /var/log/*`

b. Команда `sudo grep session /var/log/*.log`

c. Команда `tr /var/logs`

d. Команда `cat logs | nl`

Вопрос 4. Какая команда разбивает текст на столбцы и управляет отображением из столбцов?

a. Команда `tail`

b. Команда `touch`

c. Команда `sed`

d. Команда `cut`

Вопрос 5. Как можно в файле найти определенную последовательность символов, например, номер телефона, имя или фамилию?

a. Использовать символ звездочки `*`

b. Использовать специальный язык регулярных выражений

c. Использовать ЗАГЛАВНЫЕ символы

d. Написать свой сценарий

Вопрос 6. Какая утилита предоставляет скриптовый язык для построчной обработки текста?

a. Утилита `sed`

b. Утилита `cut`

c. Утилита `head`

d. Утилита `awk`

Вопрос 7. Какая утилита называется потоковым текстовым редактором?

a. Утилита `grep`

b. Утилита `sed`

c. Утилита `wc`

d. Утилита `mc`

Вопрос 8. Какая утилита позволяет посчитать количество строк или слов?

a. Утилита `less`

b. Утилита `tail`

c. Утилита `wc`

d. Утилита `head`

Вопрос 9. Как называется текстовый редактор, который имеет несколько режимов?

a. vim

b. nano

c. tr

d. less

Вопрос 10. Какая утилита позволяет выполнять построчные команды, которые поступают из потока?

a. Утилита `ls`

b. Утилита `nl`

c. Утилита `xargs`

d. Утилита `cut`

Ответы на вопросы

1. d
2. c
3. b
4. d
5. b
6. d
7. b
8. c
9. a
10. c

7.7 Заключение

В этом модуле мы рассмотрели инструменты, с помощью которых можно просматривать и редактировать текстовые документы, выполнять по ним поиск, сортировку и фильтрацию, в том числе с использованием регулярных выражений. Мы рассмотрели возможности нескольких текстовых редакторов, доступных в Astra Linux «из коробки», и закрепили полученные знания на практике. То, какими инструментами пользоваться в повседневной работе, каждый выбирает сам, но проблем с их недостатком у нас теперь точно не будет.

Темой следующего модуля будет управление локальными пользователями и группами в Astra Linux, информация о которых, кстати, хранится в обычных текстовых файлах.

7.8 Дополнительные источники информации

- Регулярные выражения можно создавать и тестировать на сайте regex101
- Библиотека регулярных выражений RegExLib
- С дополнительной и информацией о Vim вы можете ознакомится в справке `man vim` и по ссылке https://ru.wikibooks.org/wiki/Vim

7.9 Обратная связь

Если остались вопросы, то их всегда можно задать в специальной теме.

Модуль 8. Управление пользователями и группами

8.1 Введение

Из этого модуля вы узнаете от том, что такое POSIX идентификаторы и чем они отличаются от SID'ов из мира Windows. Мы посмотрим, в каких файлах хранится информация о локальных пользователях и группах и как редактировать эти файлы с помощью стандартных утилит.

В общем, этот модуль еще не предложение красной таблетки от Морфеуса, но дверь в страну чудес мы точно приоткроем.

8.2 Аутентификация в системе Linux

В операционной системе Linux функция аутентификации вынесена отдельно от приложений и реализована в виде подключаемых модулей аутентификации (*от англ. Pluggable Authentication Modules, PAM*). Поэтому приложениям, таким как login, достаточно получить у пользователя учетные данные и передать их библиотеке libpam, чтобы та выполнила за них всю грязную работу.

Этот подход хорошо зарекомендовал себя, так как позволяет встраивать в систему новые способы аутентификации, не изменяя исходного кода прикладных приложений. И, более того, для разных приложений можно использовать разный состав модулей и запускать их с разными параметрами — за эти настройки отвечают файлы из каталога `/etc/pam.d`

Возможности PAM-стека отлично дополняются функциями Диспетчера службы имен (*от англ. Name Service Switch, NSS*). Если PAM-стек обеспечивает разные способы аутентификации, то система NSS позволяет подключать различные источники для извлечения информации о пользователях, паролях и группах.

Сравнивая содержимое файлов `/etc/nsswitch.conf` на разных компьютерах, вы сможете заметить, что на обычном компьютере используются только локальные базы, которые обозначаются как *files* и *db*, а на доменных компьютерах появляется дополнительный источник *sss*, который позволяет извлекать информацию из домена через службу SSSD:

```
localadmin@astra:~$ cat /etc/nsswitch.conf

passwd:         files sss
group:          files sss
```

(continues on next page)

(продолжение с предыдущей страницы)

```
shadow:         files sss

hosts:          files dns
networks:       files

protocols:      db files
services:       db files sss
ethers:         db files
rpc:            db files

netgroup:       nis sss
sudoers:        files sss

automount:      sss
```

Источники NSS строго формализованы и должны предоставлять типовой набор данных, например, база пользователей passwd должна содержать: имя пользователя, хеш пароля, идентификатор пользователя UID, идентификатор первичной группы GID, атрибут GECOS, пути к домашнему каталогу и оболочке командной строки.

Это не означает, что все атрибуты должны быть «заполнены». Например, PAM-модулю службы SSSD не требуется прямой доступ к хешам паролей всех пользователей, т.к. аутентификация может быть выполнена через контроллер домена по сетевому протоколу Kerberos V5. Однако информация об идентификаторах UID/GID и участии пользователей в группах нужна системе обязательно, т.к. без нее не будет работать авторизация доступа к локальным ресурсам.

Примечание: В системе Windows аналогом технологий PAM и NSS является система аутентификации, которая включает в себя такие компоненты, как Локальный центр безопасности (Local Security Authority, LSA), Интерфейс поставщика поддержки безопасности (Security Support Provider Interface, SSPI) и другие.

Подсистема LSA управляет аутентификацией пользователей, авторизацией и контролем доступа, а программный интерфейс SSPI обеспечивает различные методы аутентификации, включая Kerberos, NTLM и др.

Компоненты Windows тоже имеют модульную архитектуру, что позволяет расширять их функциональность. Например, довольно часто разработчики средств защиты информации встраивают в Windows LSA свои фильтры паролей для их проверки на соответствие дополнительным требованиям.

8.3 Идентификаторы SID, UID и GID

Давайте вспомним, что в Windows у всех субъектов безопасности (пользователей, групп, служебных учетных записей и т.д.) есть уникальный идентификатор безопасности, который представляет собой строку вида:

```
S-R-IA-SA-SA-RID
```

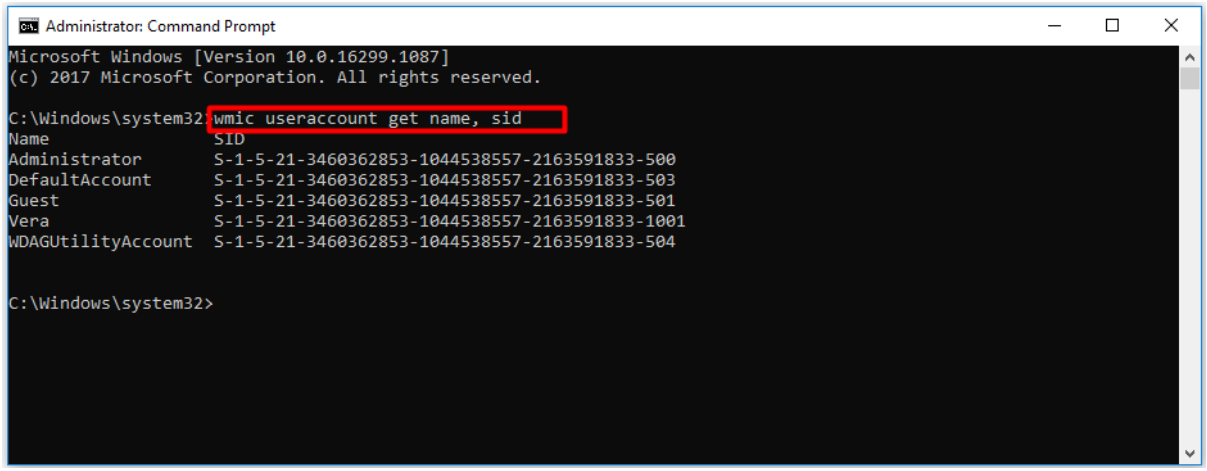
Например, идентификатор доменного пользователя *iivanov@ALD.COMPANY.LAN*:

```
S-1-5-21-2174213541-4214376712-4023651427-1006
```

Именно по этим идентификаторам, а не по именам, обеспечивается доступ пользователей к ресурсам. Рассмотрим подробно, что означает каждая из частей идентификатора:

- **S** — указывает на то, что строка является идентификатором безопасности Windows.
- **R** — первое число является номером редакции (revision). До второй версии дело так и не дошло, поэтому у всех SID'ов в этой части будет указана единица.
- **IA** — источник выдачи (issuing authority). Практически все идентификаторы безопасности в операционной системе Windows указывают NT Authority номер 5. Исключением являются универсальные широко известные (universal well-known) учётные записи групп и пользователей, например, идентификатор создателя объекта S-1-3-X (англ. SECURITY_CREATOR_SID_AUTHORITY).
- **SA** — идентификатор уполномоченного центра (sub-authority), который выдал идентификатор безопасности. Первое число определяет специальные группы и функции, например, 21 указывает, что идентификатор безопасности был выдан контроллером домена или изолированным компьютером. Далее идут три числа, которые идентифицируют компьютер/домен. Эти числа генерируются автоматически в момент установки ОС, поэтому в домене недостаточно просто клонировать машину и обязательно требуется выполнить процедуру sysprep, которая повторно генерирует это значение.
- **RID** — относительный идентификатор (Relative Identifier). Это целое число в диапазоне от 0 до 2^{31} (до 2 млрд.), которое уникально идентифицирует субъект безопасности в пределах компьютера или домена. Относительные идентификаторы обычных пользователей начинаются с 1 000. В нашем примере это значение равно 1 006.

Чтобы посмотреть значения SID на Windows, нужно запустить командную строку с правами администратора и выполнить команду `wmic useraccount get name, sid`:



Теперь мы сможем разобрать несколько особенностей идентификаторов Linux:

- Если в Windows идентификатор SID используется для всех субъектов, то в Linux для пользователей и групп используются разные идентификаторы. Идентификаторы пользователей называются UID (от англ. User Identifier), идентификаторы групп называются GID (*от англ. Group Identifier*), а все вместе они называются POSIX-идентификаторами по названию стандартов, которым следуют *nix-подобные системы (*англ. Portable Operating System Interface — переносимый интерфейс операционных систем*).

- Оба вида идентификаторов представляют собой целые числа в диапазоне от 0 до 2^{32} (4 294 967 296). Идентификаторы в диапазоне 0-99 обычно резервируются под систему, в диапазоне 100-999 используются для разного рода служб (например, Apache). А все, что начинается с 1000, отдается пользователям. Astra Linux следует этой практике, но в других дистрибутивах подходы могут отличаться, поэтому смотрите файл /etc/login.defs, какие значения там определены для переменных UID_MIN, UID_MAX, GID_MIN и GID_MAX.

- Как вы можете заметить, максимальный размер идентификаторов в 4 млрд сопоставим с емкостью относительных идентификаторов Windows. Однако у POSIX-идентификаторов нет уникальной доменной части, поэтому получается, что все пользователи и группы Linux находятся как бы в одном большом домене. Это может стать проблемой при необходимости объединения двух инфраструктур, но есть разные способы решения этой проблемы. Например, служба каталога ALD Pro (FreeIPA) выделяет для каждого нового домена 200 тыс. идентификаторов в случайном месте этого диапазона, что значительно снижает вероятность конфликта.

- Обе системы имеют так называемые широко известные идентификаторы (*от англ. well-known identifiers*). Например, у администратора Windows SID заканчивается на `...-500`, а у группы администраторов на `...-512`. В системе Linux у суперпользователя root и его первичной группы идентификаторы всегда равны 0. И эти значения желательно помнить наизусть.

- Еще одним существенным отличием Linux от Windows являются первичные группы пользо-

вателей. Дело в том, что в Linux при создании файлов и папок нужно обязательно указать не только пользователя-владельца, но и группу, которая будет считаться владельцем этого объекта. По этой причине пользователям был добавлен атрибут «первичная группа», значение которого проставляется у новых объектов файловой системы.

> Однако использовать какую-нибудь общую группу Users в качестве первичной группы для всех новых пользователей будет небезопасно. В этом случае пользователи смогут получить доступ к личным файлам друг друга, что крайне нежелательно в многопользовательских системах. Поэтому в Linux при создании нового пользователя система автоматически создает одноименную группу, идентификатор которой записывает в качестве первичной группы пользователя.

Примечание: Идентификаторы пользователей иногда совпадают с идентификаторами их первичных групп. Например, у пользователя root идентификатор UID равен 0, и у группы root идентификатор GID тоже равен 0. Но следует понимать, что это не считается какой-то лучшей практикой и для большинства обычных пользователей эти значения будут расходиться.

Например, в системе Astra Linux автоматически создается группа astra-admin с GID=1001, поэтому при создании нового пользователя ему будет присвоен первый свободный UID=1001, а идентификатор его группы будет GID=1002.

8.4 Формат хранения информации об учетных записях пользователей и групп

В Windows данные об учетных записях хранятся в реестре Windows NT, а именно в файле SAM (от англ. Security Account Manager). Этот файл находится на жестком диске в каталоге `%windows%/system32/config`. Система имеет единоличный доступ к этому файлу, и пока она загружена, доступа к нему нет даже у администраторов.

В Linux для хранения данных об учетных записях локальных пользователей, их паролей и групп используется несколько баз данных, которые представляют собой обычные текстовые файлы:

- `/etc/passwd` — база пользователей
- `/etc/shadow` — база паролей пользователей
- `/etc/group` — база групп
- `/etc/gshadow` — база паролей групп

Рассмотрим каждую их этих баз данных отдельно.

8.4.1 Хранение учетных записей пользователей (/etc/passwd)

В файле `/etc/passwd` хранятся данные об учетных записях пользователей. Это текстовый файл, информация в котором хранится в табличном виде, а разделителем столбцов является символ двоеточия «:». Первая строка, как правило, содержит учетную запись суперпользователя root, а новые учетные записи добавляются в конец файла.

```
localadmin@astra:~$ cat /etc/passwd | head
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
...
localadmin@astra:~$ grep 1000 /etc/passwd
localadmin:x:1000:1000:localadmin,,,:/home/localadmin:/bin/bash
```

Формат полей файла хранения учетных данных пользователей:

Поле в таблице	Значение
Имя пользователя	localadmin
Хеш пароля	x
UID	1000
GID	1000
GECOS	localadmin,,,
Домашний каталог	/home/localadmin
Оболочка	/bin/bash

1. **Имя пользователя** — это строка, которую пользователь вводит при входе в систему в качестве своего имени. Имя не должно начинаться с цифр, содержать большие буквы или буквы кириллицы. Максимальная длина не более 32 символов.

2. **Хеш пароля** — уникальная строка символов, полученная после применения к исходному тексту пароля специальной хеш-функции (от англ. hash - фарш). Как фарш невозможно провернуть назад, так и хеш невозможно превратить обратно в исходные данные. Проще всего хеш-функцию представить в виде синусоиды, где бесконечному значению X соответствуют значения Y строго в интервале от -1 до 1, т.е. хеширование всегда идет с потерей данных. Например, MD5-хеш любого файла всегда будет длиной 32 символа, и не важно, была ли это PNG-картинка на пару килобайт или ISO-образ размером 5 ГБ.

   В первых системах Unix хеш пароля находился непосредственно в `/etc/passwd`, но к этому файлу должен быть доступ на чтение у всех пользователей, поэтому такой подход не был защищен от взлома методом перебора. Первым, кто реализовал этот вектор атаки, стал

Ричард Столман, основатель проекта GNU, после чего хеши перенесли в файл `/etc/shadow`, к которому есть доступ только у суперпользователя.

В этой колонке сейчас указывают значение «x», если пароль задан и находится в файле shadow, и значение «!», если пароль не задан и интерактивный вход запрещен.

3. **UID** — идентификатор пользователя.
4. **GID** — идентификатор первичной группы пользователя. Эта группа проставляется владельцем для файлов и каталогов, которые будут созданы пользователем.
5. **GECOS** — содержит несколько атрибутов, разделенных запятой:
   - Полное имя пользователя или название приложения.
   - Номер комнаты.
   - Рабочий номер телефона.
   - Домашний телефон.
   - Другая контактная информация.
6. **Домашний каталог** — абсолютный путь к домашнему каталогу пользователя, в котором находятся его личные файлы. По умолчанию домашние каталоги называются по имени пользователя и располагаются в каталоге `/home`.
7. **Оболочка** — абсолютный путь к оболочке, которая запускается для пользователя при входе в терминал. В большинстве дистрибутивов Linux оболочкой входа по умолчанию является Bash, в том числе и в Astra Linux. Список установленных в системе оболочек можно посмотреть в файле `/etc/shells`.

Примечание: Изначально атрибут GECOS был добавлен для совместимости с операционной системой General Comprehensive Operating System, которую часто использовали в качестве принт-сервера. Уже и системы той нет, а атрибут остался, поэтому в него обычно записывают полное имя пользователя.

8.4.2 Хранение паролей пользователей. Файл /etc/shadow.

Файл `/etc/shadow` содержит хеши паролей пользователей, поэтому доступ к нему должен быть только у суперпользователя. Для просмотра файла необходимо воспользоваться командой повышения привилегий `sudo`.

```
localadmin@astra:~$ sudo -i
[sudo] пароль для localadmin:
root@astra:~# cat /etc/shadow | head
root:!:19493:0:99999:7:::
daemon:*:19493:0:99999:7:::
bin:*:19493:0:99999:7:::
sys:*:19493:0:99999:7:::
sync:*:19493:0:99999:7:::
games:*:19493:0:99999:7:::
man:*:19493:0:99999:7:::
lp:*:19493:0:99999:7:::
mail:*:19493:0:99999:7:::
news:*:19493:0:99999:7:::
root@astra:~# cat /etc/shadow | grep localadmin
```

Строка найденная в базе по имени localadmin:

```
localadmin:$gost12512hash$Ag32kwCZw/YI1iaX
↪$7d6oorvJr4EnfprtlXzeA6IUJzAaZ0yab1ghPbQbVGGixLxhWgI3K1qOG9WKkw4sh0ov/Fe4OUWZ0.p.
↪j1oC91:19493:0:99999:7:::
```

Синтаксис файла такой же, как у passwd — информация хранится в табличном виде, разделителем столбцов является символ двоеточие «:». Как правило, первая строка в файле содержит учетную запись суперпользователя root, а для новых учетных записей информация добавляется в виде новых строк в конец файла.

Формат полей файла `/etc/shadow`:

Поле в таблице	Значение
Имя пользователя	localadmin
Хеш пароля	$gost12512hash$Ag32...
Последнее изменение пароля	19493
Минимальный возраст пароля	0
Максимальный возраст пароля	99999
Период предупреждения	7
Период бездействия	пусто
Срок действия уч. записи	пусто
Резерв	пусто

1. **Имя пользователя** — строка, которую вы вводите при входе в систему. Учетная запись пользователя должна существовать в системе.

2. **Хеш пароля** — уникальная строка символов, полученная после применения специальной хеш-функции исходному тексту пароля.

   Хеш задается в формате $type$salt$hashed:

- **Подстрока $type** — это метод криптографического алгоритма хеширования, который может принимать следующие значения:
  - **1** — алгоритм хеширования MD5
  - **$2a$** — алгоритм хеширования Blowfish
  - **$2y$** — алгоритм хеширования Eksblowfish
  - **5** — алгоритм хеширования SHA-256
  - **6** — алгоритм хеширования SHA-512
  - **$gost12512hash$** — алгоритм хеширования ГОСТ Р 34.11-2012

    В Astra Linux при генерации пароля применяется алгоритм защищенного преобразования ГОСТ Р 34.11.-2012.
- **Подстрока $salt$** — задает набор случайных символов, который объединяется с паролем при хешировании для усложнения взлома паролей с использованием радужных таблиц (от англ. rainbow table).
- **hashed** — само значение хеша в кодировке base64

Однако, вместо хеша ($type$salt$hashed) поле может содержать следующие специальные символы:

- Символ «*» – Если поле пароля начинается с символа звёздочка (*), то пользователь не сможет войти в систему с использованием аутентификации по паролю. Другие методы входа, такие как аутентификация на основе ключей или переключение на пользователя, по-прежнему разрешены.
- Символ «!» – Если поле пароля начинается с символа восклицательный знак (!), то пользователь вообще не сможет войти в систему даже по ключу. Пароль в этом случае считается заблокированным.

3. **Последнее изменение пароля** — это дата последнего изменения пароля в количестве дней, прошедших с 1 января 1970 года, как начала эпохи Unix. Преобразовать это значение в реальную дату можно командой: `date -d @$((19493*24*3600))`
4. **Минимальный срок действия пароля** — количество дней, которое должно пройти, прежде чем пароль пользователя может быть изменен. Как правило, минимальный срок действия пароля не задан, поэтому в этой колонке будет 0.
5. **Максимальный срок действия пароля** — количество дней, по истечении которых пароль будет считаться истекшим. Как правило, максимальный срок не задан, поэтому в этой колонке

будет установлено значение 99999.

6. **Период предупреждения** — количество дней до истечения срока действия пароля, в течение которых пользователь будет получать предупреждение о необходимости изменить пароль.

7. **Период бездействия** — количество дней после истечения срока действия пароля, во время которых пароль всё ещё будет приниматься. Это поле обычно остается пустым.

8. **Срок действия учетной записи** — дата истечения срока действия учётной записи в формате timestamp, после которой учетная запись будет заблокирована. Обратите внимание, если истек срок действия учетной записи, то пользователь не сможет войти в систему. Если же закончился срок пароля, то пользователь сможет сменить его и продолжить работу.

9. **Резерв** — это поле было зарезервировано для использования в будущем. В настоящий момент не используется.

Важно: Крайне не рекомендуется редактировать файл `/etc/passwd` вручную, т.к. это может привести к повреждению данных.

Используйте специальные команды, например:

- `passwd` — для смены пароля;
- `fly-passwd` — для смены пароля в графическом интерфейсе;
- `chage` — для изменения срока действия пароля.

8.4.3 Хранение групп (/etc/group)

Файл `/etc/group` содержит информацию о группах, зарегистрированных в системе, включая информацию об участниках этих групп.

```
localadmin@astra:~$ cat /etc/group | head -n 25 | tail
fax:x:21:
voice:x:22:
cdrom:x:24:localadmin
floppy:x:25:localadmin
tape:x:26:
sudo:x:27:
audio:x:29:pulse,localadmin
dip:x:30:localadmin
www-data:x:33:
backup:x:34:
```

(continues on next page)

(продолжение с предыдущей страницы)

```
localadmin@astra:~$ grep astra-admin /etc/group
localadmin:x:1001:localadmin
```

Это текстовый файл, информация в котором хранится в табличном виде, а разделителем столбцов является символ двоеточие «:».

Формат полей файла информации о группах `/etc/group`:

Поле в таблице	Значение
Имя группы	localadmin
Хеш пароля	x
GID	1001
Список участников группы	localadmin

1. **Имя группы** — набор символов, по которым можно уникально идентифицировать группу.
2. **Хеш пароля** — аналогично с паролем пользователя хеши перенесены в файл `/etc/group`. Пароль группы позволяет пользователю самостоятельно присоединиться к группе для расширения своих прав доступа.
3. **GID** – идентификатор группы.
4. **Список участников группы** — список пользователей, разделенный запятыми без пробелов.

8.4.4 Хранение паролей групп (/etc/gshadow)

Файл `/etc/gshadow` хранит информацию о паролях локальных групп. Доступ к этому файлу есть только у суперпользователя, поэтому для его просмотра необходимо воспользоваться командой повышения привилегий sudo.

```
localadmin@astra:~$ sudo cat /etc/gshadow | head
root:*::
daemon:*::
bin:*::
sys:*::
adm:*::loaches
tty:*::
disk:*::
lp:*::
mail:*::
news:*::

localadmin@astra:~$ sudo cat /etc/gshadow | grep astra-admin
astra-admin:!::localadmin
```

Это текстовый файл, информация в котором хранится в табличном виде, а разделителем столбцов является символ двоеточие «:». Как правило первая строка в файле содержит учетную запись группы суперпользователя root, а для новых учетных записей информация добавляется в виде новых строк в конец файла.

Формат полей файла информации о паролях локальных групп `/etc/gshadow`:

Поле в таблице	Значение
Имя группы	astra-admin
Хеш пароля	!
Администраторы	
Участники группы	localadmin

1. **Имя группы** — должно содержать имя группы, которая зарегистрирована в системе.
2. **Хеш пароля** - позволяет пользователю самостоятельно присоединиться к группе для расширения своих прав доступа. Если поле содержит символ «!» или «*», то пользователь не сможет использовать пароль для доступа в группу.
3. **Администраторы** — список имён пользователей, перечисленных через запятую. Администраторы имеют те же права, как у обычных участников, и дополнительно могут изменять список участников группы и назначать новый пароль.
4. **Участники группы** — список имён пользователей, перечисленных через запятую. Участники группы могут иметь доступ к группе без ввода пароля.

8.4.5 Изменение файлов /etc/passwd, /etc/group и /etc/shadow

Если вы хотите изменить указанные файлы с помощью обычного текстового редактора, рекомендуется использовать специальные утилиты, которые проверяют корректность файла в момент сохранения данных:

- `vipw` – редактирование базы пользователей `/etc/passwd`
- `vipw -g` – редактирование базы групп пользователей `/etc/group`
- `vipw -s` – редактирование базы паролей пользователей `/etc/shadow`

При этом не важно, какой именно редактор вы выберете для работы – это может быть `vim`, `nano`, `mcedit`.

8.4.6 Резервные копии файлов /etc/passwd, /etc/group и /etc/shadow

В системе существуют резервные копии рассмотренных выше файлов (со знаком тире в конце имени файла):

- `/etc/shadow-` – резервный файл хешей паролей пользователей
- `/etc/passwd-` – резервный файл базы пользователей
- `/etc/group-` – резервный файл базы групп пользователей
- `/etc/gshadow-` – резервный файл паролей групп пользователей

Резервные копии хранят предыдущую версию файлов. Когда утилиты управления меняют пользователя или группу, они создают резервные копии основных файлов. В этом можно убедиться командой `diff`. Например, после создания пользователя выполните в терминале команду `diff /etc/passwd /etc/passwd-`.

8.5 Управление учетными записями пользователей из командной строки

8.5.1 Просмотр учетных записей пользователей

Список учетных записей можно посмотреть в файле `/etc/passwd`:

```
localadmin@astra:~$ less /etc/passwd
```

Проверить, существует ли пользователь, и вывести по нему информацию можно командой `id`:

```
localadmin@astra:~$ id localadmin
uid=1000(localadmin) gid=1000(localadmin)
группы=1000(localadmin),24(cdrom),25(floppy),29(audio),30(dip),44(video),46
(plugdev),109(netdev),113(lpadmin),114(scanner),333(astra-console),1001(astra-admin)
```

Команда `getent` отображает записи из любой базы данных, которая поддерживается библиотеками NSS. С помощью неё можно получить информацию как о пользователе, так и о группе или пароле:

```
localadmin@astra:~$ getent passwd localadmin
localadmin:x:1000:1000:localadmin,,,:/home/localadmin:/bin/bash
```

Утилита `lslogins` позволяет вывести информацию об учетных записях пользователей, атрибутах паролей и информацию о сеансах:

```
localadmin@astra:~$ lslogins | tail -n 5
112 logcheck             0                logcheck system account,,,
113 avahi                2                Avahi mDNS daemon,,,
998 vboxadd              0
999 systemd-coredump     0                systemd Core Dumper
```

(continues on next page)

(продолжение с предыдущей страницы)

```
1000 localadmin          43      08:32    localadmin,,,
...
```

8.5.2 Создание учетных записей пользователей

Для создания учетных записей пользователей из консоли используются команды useradd/adduser.

Команда useradd — это встроенная команда Linux, которую можно найти в любой системе Linux. А вот adduser не является стандартной командой Linux, это дружественный интерфейс к программам useradd и usermod.

8.5.2.1 Утилита useradd

Команда useradd регистрирует нового пользователя или изменяет информацию по умолчанию о новых пользователях. Команда для работы требует привилегий до суперпользователя.

Синтаксис команды: useradd <опции> <имя_пользователя>

Все доступные опции и описание команды можно найти в справке man useradd. Далее рассмотрим наиболее часто используемые сценарии применения этой команды.

Утилита useradd имеет свой собственный конфигурационный файл /etc/default/useradd, в котором определены следующие параметры:

- **SHELL** — задает командную оболочку /bin/sh, которая будет назначаться пользователям по умолчанию, если значение не было задано явно в момент вызова команды.
- **SKEL** — задает директорию с шаблонами по умолчанию /etc/skel. Шаблоны — это директории и файлы, которые копируются в домашний каталог пользователя при его создании.
- **HOME** — указывает на директорию, в которой будут создаваться домашние папки пользователей.
- **CREATE_MAIL_SPOOL** — определяет, будет ли создаваться файл для хранения локальной почты пользователя. По умолчанию он не создается. Это можно увидеть, запустив команду useradd -D. В выводе команды будут отражены настройки по умолчанию, применяемые при создании пользователя.

```
localadmin@astra:~$ sudo useradd -D
GROUP=100
HOME=/home
INACTIVE=-1
EXPIRE=
```

(continues on next page)

(продолжение с предыдущей страницы)

```
SHELL=/bin/sh
SKEL=/etc/skel
CREATE_MAIL_SPOOL=no
```

```
localadmin@astra:~$ cat /etc/default/useradd
```

Результат вывода:

```
# Default values for useradd(8)
#
# The SHELL variable specifies the default login shell on your
# system.
# Similar to DHSELL in adduser. However, we use "sh" here because
# useradd is a low level utility and should be as general
# as possible
SHELL=/bin/sh
#
# The default group for users
# 100=users on Debian systems
# Same as USERS_GID in adduser
# This argument is used when the -n flag is specified.
# The default behavior (when -n and -g are not specified) is to create a
# primary user group with the same name as the user being added to the
# system.
# GROUP=100
#
# The default home directory. Same as DHOME for adduser
# HOME=/home
#
# The number of days after a password expires until the account
# is permanently disabled
# INACTIVE=-1
#
# The default expire date
# EXPIRE=
#
# The SKEL variable specifies the directory containing "skeletal" user
# files; in other words, files such as a sample .profile that will be
# copied to the new user's home directory when it is created.
# SKEL=/etc/skel
#
# Defines whether the mail spool should be created while
# creating the account
# CREATE_MAIL_SPOOL=yes
```

Для простого создания нового пользователя достаточно выполнить команду:

```
sudo useradd username
```

Пользователь будет создан с параметрами по умолчанию. Для явного выбора оболочки необходимо использовать параметр `-s` и полный путь до исполнимого файла, например, `sudo useradd username -s /bin/bash`

Для того чтобы пользователь смог войти в систему, необходимо также задать пароль:

`sudo passwd username`

Для создания пользователя с домашним каталогом в стандартной директории `/home` необходимо воспользоваться ключом `-m`. При этом будет создан новый каталог `/home/username`:

`sudo useradd -m username`

При необходимости домашний каталог может быть создан не в стандартной директории /home, а в любой другой. Для этого необходимо:

1. Опционально: создать родительский каталог для домашнего каталога будущего пользователя (пустой каталог):

   `sudo mkdir -p /users`

2. Создать пользователя с домашним каталогом в требуемой директории:

   `sudo useradd -d /users/username -m username`

```
localadmin@astra:~$ sudo mkdir -p /users
localadmin@astra:~$ sudo useradd -d /users/user1 -m user1
```

Для создания пользователя с определенным UID и с определенной группой по умолчанию необходимо воспользоваться ключами -u и -g соответственно (группа с таким GID должна быть уже предварительно создана):

```
localadmin@astra:~$ sudo useradd -u 1111 -g 1222 username
```

Чтобы создать пользователя и добавить его в дополнительные группы, используется ключ `-G`, например:

```
localadmin@astra:~$ sudo useradd -G sudo,ssh username
```

Таким образом, команда для создания нового пользователя с домашним каталогом в стандартной директории /home, включенного в группы sudo и ssh, и оболочкой bash будет выглядеть так:

```
localadmin@astra:~$ sudo useradd -m -G sudo,ssh -s /bin/bash username
```

Не стоит забывать, что требуется еще задать пароль пользователю, чтобы он смог войти в систему:

```
localadmin@astra:~$ sudo passwd username
```

Создадим несколько новых пользователей с заданными параметрами. Пароль для тестовых пользователей зададим *Pa$$w0rd*.

Создадим простого пользователя user1:

```
localadmin@astra:~$ sudo useradd -m -u 1002 -s /bin/bash user1
localadmin@astra:~$ sudo passwd user1
Новый пароль :
Повторите ввод нового пароля :
passwd: пароль успешно обновлён
```

Создадим простого пользователя user2:

```
localadmin@astra:~$ sudo useradd -m -s /bin/bash user2
localadmin@astra:~$ sudo passwd user2
Новый пароль :
Повторите ввод нового пароля :
passwd: пароль успешно обновлён
```

Создадим администратора admin1

```
localadmin@astra:~$ sudo useradd -m -G astra-admin -s /bin/bash admin1
localadmin@astra:~$ sudo passwd admin1
Новый пароль :
Повторите ввод нового пароля :
passwd: пароль успешно обновлён
```

Создадим группу developers (подробнее о создании групп далее в этом модуле) с GID 1200

```
localadmin@astra:~$ sudo groupadd developers -g 1200
```

Создадим пользователя user3 с заданными UID и GID

```
localadmin@astra:~$ sudo useradd -m -u 1150 -g 1200 -s /bin/bash user3
localadmin@astra:~$ sudo passwd user3
Новый пароль :
Повторите ввод нового пароля :
passwd: пароль успешно обновлён
```

Посмотрите содержимое файлов `/etc/passwd` и `/etc/group`, чтобы ответить на вопросы:

- Была ли группа user3 создана для пользователя user3 и почему?
- Какая оболочка была выбрана для пользователя user3 и почему?

8.5.2.2 Утилита adduser

Команда `adduser` – это Perl-скрипт для команд `useradd` и `usermod`, убедиться в этом поможет команда `file /usr/sbin/adduser`. У этого скрипта есть свои конфигурационные файлы:

- `/etc/adduser.conf` – основной конфигурационный файл;
- `/usr/local/sbin/adduser.local` – по своей сути является скриптом sh для настройки нового пользователя. Если этот файл существует, то он будет запущен сразу после создания учётной записи пользователя, чтобы выполнить необходимые локальные настройки.

Команда `adduser` в отличие от `useradd` по умолчанию назначает пользователю командную оболочку `/bin/bash` (параметр DSHELL). В параметре EXTRA_GROUPS указываются дополнительные группы, в которые должен быть включен новый пользователь. При использовании `useradd` группы надо указывать явно с помощью параметра `-G`.

```
localadmin@astra:~$ cat /etc/adduser.conf
```

```
# /etc/adduser.conf: 'adduser' configuration.
# See adduser(8) and adduser.conf(5) for full documentation.

# The DSHELL variable specifies the default login shell on your
# system.
DSHELL=/bin/bash

# The DHOME variable specifies the directory containing users' home
# directories.
DHOME=/home

# If GROUPHOMES is "yes", then the home directories will be created as
# /home/groupname/user.
GROUPHOMES=no

# If LETTERHOMES is "yes", then the created home directories will have
# an extra directory - the first letter of the user name. For example:
# /home/u/user.
LETTERHOMES=no

# The SKEL variable specifies the directory containing "skeletal" user
# files; in other words, files such as a sample .profile that will be
# copied to the new user's home directory when it is created.
SKEL=/etc/skel

# FIRST_SYSTEM_[GU]ID to LAST_SYSTEM_[GU]ID inclusive is the range for UIDs
# for dynamically allocated administrative and system accounts/groups.
# Please note that system software, such as the users allocated by the base-passwd
# package, may assume that UIDs less than 100 are unallocated.
FIRST_SYSTEM_UID=100
LAST_SYSTEM_UID=999

FIRST_SYSTEM_GID=100
LAST_SYSTEM_GID=999

# FIRST_[GU]ID to LAST_[GU]ID inclusive is the range of UIDs of dynamically
# allocated user accounts/groups.
FIRST_UID=1000
LAST_UID=59999

FIRST_GID=1000
LAST_GID=59999

# The USERGROUPS variable can be either "yes" or "no". If "yes" each
# created user will be given their own group to use as a default. If
# "no", each created user will be placed in the group whose gid is
# USERS_GID (see below).
USERGROUPS=yes
```

(continues on next page)

(продолжение с предыдущей страницы)

```
# If USERGROUPS is "no", then USERS_GID should be the GID of the group
# `users' (or the equivalent group) on your system.
USERS_GID=100

# If DIR_MODE is set, directories will be created with the specified
# mode. Otherwise the default mode 0755 will be used.
DIR_MODE=0700

# If SETGID_HOME is "yes" home directories for users with their own
# group the setgid bit will be set. This was the default for
# versions << 3.13 of adduser. Because it has some bad side effects we
# no longer do this per default. If you want it nevertheless you can
# still set it here.
SETGID_HOME=no

# If QUOTAUSER is set, a default quota will be set from that user with
# `edquota -p QUOTAUSER newuser'
QUOTAUSER=""

# If SKEL_IGNORE_REGEX is set, adduser will ignore files matching this
# regular expression when creating a new home directory
SKEL_IGNORE_REGEX="dpkg-(old|new|dist|save)"

# Set this if you want the --add_extra_groups option to adduser to add
# new users to other groups.
# This is the list of groups that new non-system users will be added to
# Default:
EXTRA_GROUPS="fuse weston-launch dialout cdrom floppy audio video plugdev users"

# If ADD_EXTRA_GROUPS is set to something non-zero, the EXTRA_GROUPS
# option above will be default behavior for adding new, non-system users
ADD_EXTRA_GROUPS=1

# check user and group names also against this regular expression.
#NAME_REGEX="^[a-z][-a-z0-9_]*\$"
```

При запуске `adduser` запрашивает дополнительную информацию о пользователе:

```
localadmin@astra:~$ sudo adduser testuser
Добавляется пользователь «testuser» ...
Добавляется новая группа «testuser» (1002) ...
Добавляется новый пользователь «testuser» (1001) в группу «testuser» ...
Создаётся домашний каталог «/home/testuser» ...
Копирование файлов из «/etc/skel» ...
Новый пароль :
Повторите ввод нового пароля :
passwd: пароль успешно обновлён
Изменение информации о пользователе testuser
Введите новое значение или нажмите ENTER для выбора значения по умолчанию
      Полное имя []:
      Номер комнаты []:
      Рабочий телефон []:
      Домашний телефон []:
      Другое []:
Данная информация корректна? [Y/n]
```

(continues on next page)

(продолжение с предыдущей страницы)

```
Добавляется новый пользователь «testuser» в дополнительные группы ...
adduser: Группа «fuse» не существует.
adduser: Группа «weston-launch» не существует.
Добавляется пользователь «testuser» в группу «dialout» ...
Добавляется пользователь «testuser» в группу «cdrom» ...
Добавляется пользователь «testuser» в группу «floppy» ...
Добавляется пользователь «testuser» в группу «audio» ...
Добавляется пользователь «testuser» в группу «video» ...
Добавляется пользователь «testuser» в группу «plugdev» ...
Добавляется пользователь «testuser» в группу «users» ...
```

8.5.3 Изменение учетных записей пользователей

С помощью команды `usermod` мы можем изменять параметры учетных записей пользователей, задаваемые утилитой `useradd`, и управлять дополнительными параметрами. Как и команда `useradd`, команда `usermod` требует прав суперпользователя.

Синтаксис команды: `useradd <опции> <имя_пользователя>`

Все доступные опции и описание команды можно найти в справке `man usermod`. Далее рассмотрим наиболее часто используемые сценарии применения этой команды.

8.5.3.1 Изменение основной группы

Для изменения основной группы необходим ключ `-g`. Создадим группу user3 с GID=1150 и назначим её в качестве основной для пользователя user3.

```
localadmin@astra:~$ sudo groupadd -g 1150 user3
localadmin@astra:~$ sudo usermod -g user3 user3
localadmin@astra:~$ id user3
uid=1150(user3) gid=1150(user3) группы=1150(user3)
```

Вместо имени группы можно указать её GID: `sudo usermod -g 1150`.

8.5.3.2 Добавление и исключение из группы

Командой `usermod` можно добавить пользователя в дополнительную группу.

Добавим пользователя user3 в группы developers и adm. Когда мы используем ключ `-G`, нам необходимо перечислить все текущие группы пользователя, кроме группы по умолчанию, и добавить к этому списку все группы, в которые его необходимо включить. В противном случае пользователь будет исключен из всех текущих групп, т.к. они не были перечислены. Но если в дополнение к ключу `-G` использовать ключ `-a`, то можно будет не указывать список текущих групп, и пользователь не будет из них исключен.

Для наглядности сделаем это в 2 этапа. У пользователя user3 сейчас нет ни одной дополнительной группы, и мы применим ключ `-G` для добавления его в группу developers, а затем добавим его еще и в группу adm, воспользовавшись ключами `-a` и `-G`:

```
localadmin@astra:~$ sudo usermod -G developers user3
localadmin@astra:~$ id user3
uid=1150(user3) gid=1150(user3) группы=1150(user3),1200(developers)
localadmin@astra:~$ sudo usermod -a -G adm user3
localadmin@astra:~$ id user3
uid=1150(user3) gid=1150(user3) группы=1150(user3),4(adm),1200(developers)
```

Командой `usermod` можно исключить пользователя из группы в явном виде. Для этого используем ключи `-r`, `-G` и имя дополнительной группы, из которой надо исключить пользователя.

Чтобы исключить пользователя из некоторых дополнительных групп, необходимо выполнить команду `usermod -G` и перечислить только те группы, в которых пользователь должен остаться.

Для исключения пользователя из всех дополнительных групп необходимо задать пустые кавычки после ключа `-G`:

```
localadmin@astra:~$ sudo usermod -G developers,adm user3
localadmin@astra:~$ id user3
uid=1150(user3) gid=1150(user3) группы=1150(user3),4(adm),1200(developers)
localadmin@astra:~$ sudo usermod -r -G adm user3
localadmin@astra:~$ id user3
uid=1150(user3) gid=1150(user3) группы=1150(user3),1200(developers)
localadmin@astra:~$ sudo usermod -G "" user3
localadmin@astra:~$ id user3
uid=1150(user3) gid=1150(user3) группы=1150(user3)
```

Исключить пользователя из группы в явном виде также можно командой `deluser` и синтаксис команды в таком случае будет: `deluser <пользователь> <группа>`

```
localadmin@astra:~$ sudo usermod -a -G developers user3
localadmin@astra:~$ groups user3
user3: user3 developers
localadmin@astra:~$ sudo deluser user3 developers
Удаляется пользователь «user3» из группы «developers» ...
Готово.
localadmin@astra:~$ groups user3
user3: user3
```

8.5.3.3 Изменение домашнего каталога пользователя

Перед изменением домашнего каталога пользователя во избежание потери данных в случае сбоя лучше сделать его резервное копирование с сохранением прав доступа командой `cp -rp <источник>/ <назначение>/`. Сделаем резервную копию домашнего каталога пользователя user3:

```
localadmin@astra:~$ sudo cp -rp /home/user3 /home/user3_bkp
localadmin@astra:~$ sudo ls -l /home/user3_bkp/
итого 4
```

Для изменения домашнего каталога пользователя необходим ключ `-d`. Для создания нового домашнего каталога необходимо просто указать полный путь до него. Зададим новый каталог `/home/new_dir` для пользователя user3:

```
localadmin@astra:~$ grep user3 /etc/passwd
user3:x:1150:1150::/home/user3:/bin/bash
localadmin@astra:~$ sudo usermod -d /home/new-dir user3
localadmin@astra:~$ grep user3 /etc/passwd
user3:x:1150:1150::/home/new-dir:/bin/bash
localadmin@astra:~$ ls -l /home
итого 24
drwx------  3 admin1     admin1     4096 июл 25 11:37 admin1
drwx------ 27 localadmin localadmin 4096 июл 18 12:21 localadmin
drwx------  3 user1      user1      4096 июл 25 11:37 user1
drwx------  3 user2      user2      4096 июл 25 11:37 user2
drwx------  3 user3      user3      4096 июл 25 12:03 new-dir
```

Как видим, в конфигурационный файл `/etc/passwd` были внесены изменения, был создан новый каталог `/home/new-dir`, файлы из старого каталога были скопированы в новый, а старый каталог удалился.

8.5.3.4 Изменение оболочки пользователя

Изменить оболочку пользователя можно ключом `-s`. Список доступных оболочек хранится в файле `/etc/shells`. Назначим пользователю user3 оболочку sh.

```
localadmin@astra:~$ cat /etc/shells
# /etc/shells: valid login shells
/bin/sh
/bin/bash
/usr/bin/bash
/bin/rbash
/usr/bin/rbash
/bin/dash
/usr/bin/dash
localadmin@astra:~$ sudo usermod -s /bin/sh user3
localadmin@astra:~$ getent passwd user3
user3:x:1150:1150::/home/new-dir:/bin/sh
```

Обычный пользователь может изменить свою оболочку командой `chsh -s имя_интерпретатора` (но только на то, что перечислено в `/etc/shells`).

8.5.3.5 Изменение UID пользователя

Для изменения UID пользователя необходимо воспользоваться ключом `-u`. Изменим UID пользователя user3 на 1005.

```
localadmin@astra:~$ id user3
uid=1150(user3) gid=1150(user3) группы=1150(user3)
localadmin@astra:~$ sudo usermod -u 1005 user3
localadmin@astra:~$ id user3
uid=1005(user3) gid=1150(user3) группы=1150(user3)
```

Если необходимо задать не уникальный UID (UID, принадлежащий уже какому-то пользователю, например, 1003), необходимо добавить ключ `-o`: `sudo usermod -o -u 1003 user3`. Как вы понимаете, назначать один и тот же идентификатор двум и более субъектам возможно, но крайне не рекомендуется из соображений безопасности.

8.5.3.6 Изменение имени пользователя

Изменить имя пользователя можно ключом `-l` (строчная L). Изменим имя пользователя admin1 на admin:

```
localadmin@astra:~$ sudo usermod -l admin admin1
localadmin@astra:~$ ls /home
admin1  localadmin  user1  user2  new-dir  user3_bkp
```

Однако при этом не происходит изменения имени домашнего каталога и группы по умолчанию.

Имя каталога может быть изменено сразу при выполнении команды или после, как рассматривалось выше с помощью ключей `-m` и `-d`:

```
localadmin@astra:~$ sudo usermod -l admin1 admin
localadmin@astra:~$ sudo usermod -l admin admin1 -m -d /home/admin1
localadmin@astra:~$ ls /home
admin1  localadmin  user1  user2  new-dir  user3_bkp
localadmin@astra:~$ sudo groupmod -n admin admin1
localadmin@astra:~$ id admin
uid=1004(admin) gid=1004(admin) группы=1004(admin),1001(astra-admin)
```

8.5.3.7 Блокировка пользователя

Чтобы блокировать пользователю вход по паролю, необходимо воспользоваться ключом -L. При этом перед зашифрованным паролем пользователя в файле /etc/shadow добавляется восклицательный знак. Заблокируем пользователя user3:

```
localadmin@astra:~$ sudo grep user3 /etc/shadow
user3:$gost12512hash$t...:19563:0:99999:7:::
localadmin@astra:~$ sudo usermod -L user3
```

(continues on next page)

(продолжение с предыдущей страницы)

```
localadmin@astra:~$ sudo grep user3 /etc/shadow
user3:!$gost12512hash$t...:19563:0:99999:7:::
```

Еще пользователя можно заблокировать командой `sudo passwd -l username`, а вывести информацию о состоянии учетной записи командой `sudo passwd -S username`:

```
localadmin@astra:~$ sudo passwd -l user3
passwd: пароль изменён.
localadmin@astra:~$ sudo passwd -S user3
user3 L 07/26/2023 0 99999 7 -1
```

Однако другие способы входа, например, по сертификату все еще доступны пользователю. Чтобы полностью заблокировать пользователя, необходимо добавить параметр `--expiredate` со значением 1. Полностью заблокируем пользователя user3:

```
localadmin@astra:~$ sudo usermod -L --expiredate 1 user3
localadmin@astra:~$ sudo grep user3 /etc/shadow
user3:!$gost12512hash$t...:19563:0:99999:7::1:
```

Для разблокировки входа по паролю необходимо воспользоваться ключом -U, а для полной разблокировки ключом `--expiredate` с пустыми кавычками “”. Разблокируем пользователя user3:

```
sudo usermod -U --expiredate "" user3
sudo grep user3 /etc/shadow
```

```
localadmin@astra:~$ sudo usermod -U --expiredate "" user3
localadmin@astra:~$ sudo grep user3 /etc/shadow
user3:$gost12512hash$t...:19563:0:99999:7:::
```

Можно задать определенную дату блокировки учетной записи с помощью ключа `--expiredate` в формате ГГГГ-ММ-ДД, а с помощью команды `chage` с опцией `-l` (строчная L) посмотреть срок действия учетной записи и другие параметры:

```
localadmin@astra:~$ sudo usermod --expiredate 2024-12-31 user3
localadmin@astra:~$ sudo chage -l user3
Последний раз пароль был изменён                                    : июл 25, 2023
Срок действия пароля истекает                                       : никогда
Пароль будет деактивирован через                                    : никогда
Срок действия учётной записи истекает                               : дек 31, 2024
Минимальное количество дней между сменой пароля                     : 0
Максимальное количество дней между сменой пароля                    : 99999
Количество дней с предупреждением перед деактивацией пароля        : 7
```

8.5.3.8 Изменение информационного поля GECOS

Для изменения информационных полей GECOS мы можем воспользоваться ключом `-c` команды `usermod` или специальной командой `chfn` (change finger information). Рассмотрим оба варианта.

Запустим команду `sudo chfn admin` и в интерактивной форме заполним поля GECOS для учетной записи admin. Пользователь может самостоятельно менять эти поля у своей собственной учетной записи. Прав суперпользователя при этом не требуется:

```
localadmin@astra:~$ sudo chfn admin
Изменение информации о пользователе admin
Введите новое значение или нажмите ENTER для выбора значения по умолчанию
      Полное имя []: Иван Сидоров
      Номер комнаты []: 201
      Рабочий телефон []: 555-1234
      Домашний телефон []: 666-4321
      Другое []: Администратор Linux
chfn: имя «Иван Сидоров» содержит не ASCII-символы
chfn: в «Администратор Linux» содержатся не ASCII-символы
localadmin@astra:~$ getent passwd admin
admin:x:1004:1004:Иван Сидоров,201,555-1234,666-4321,Администратор Linux:/home/admin:/
↪bin/bash
```

Как видим, в файл `/etc/passwd` были внесены соответствующие изменения. Командой `chfn` можно менять и отдельные поля в GECOS. Например, ключом `-f` можно изменить полное имя, а ключом -o изменить поле «Другое». Полная справка по этой команде доступна через `man chfn`.

```
localadmin@astra:~$ sudo chfn -o "Linux администратор" admin
chfn: имя «Иван Сидоров» содержит не ASCII-символы
chfn: в «Linux администратор» содержатся не ASCII-символы
localadmin@astra:~$ getent passwd admin
admin:x:1004:1004:Иван Сидоров,201,555-1234,666-4321,Linux администратор:/home/admin:/
↪bin/bash
```

Теперь изменим значение GECOS с помощью ключа `-c` команды `usermod`. При этом нам необходимо передавать в кавычках 5 полей, разделенных запятой без пробелов, со следующими значениями соответственно: ФИО, номер комнаты (или офиса), номер рабочего телефона, номер домашнего телефона, дополнительная информация. Добавим GECOS для учетной записи user3:

```
localadmin@astra:~$ sudo usermod -c "Петр Иванов,202,555-1233,555-1122,Разработчик"␣
↪user3
localadmin@astra:~$ getent passwd user3
user3:x:1150:1200:Петр Иванов,202,555-1233,555-1122,Разработчик:/home/user3:/bin/bash
```

8.5.3.9 Изменение параметров пароля учетной записи

Изменить параметры срока действия пароля учетной записи можно командой `chage`. Мы уже применяли её, когда просматривали этот параметр. Выведем текущие параметры пароля у пользователя admin:

```
localadmin@astra:~$ sudo chage -l admin
Последний раз пароль был изменён                                    : июл 25, 2023
Срок действия пароля истекает                                       : никогда
Пароль будет деактивирован через                                    : никогда
```

(continues on next page)

(продолжение с предыдущей страницы)

```
Срок действия учётной записи истекает                                  : никогда
Минимальное количество дней между сменой пароля                        : 0
Максимальное количество дней между сменой пароля                       : 99999
Количество дней с предупреждением перед деактивацией пароля           : 7
```

Посмотрим те же параметры в файле `/etc/shadow`:

```
localadmin@astra:~$ sudo grep -E ^admin /etc/shadow
admin:$gost12512hash$hG5F9o[...].:19533:0:99999:7:::
```

Чтобы поменять дату последнего изменения пароля, применяется ключ `-d` со значением в количестве дней с начала эпохи Unix (01.01.1970) или в формате ГГГГ-ММ-ДД.

```
localadmin@astra:~$ sudo chage -d 2023-07-24 admin
localadmin@astra:~$ sudo chage -l admin
Последний раз пароль был изменён                                       : июл 24, 2023
Срок действия пароля истекает                                          : никогда
Пароль будет деактивирован через                                       : никогда
Срок действия учётной записи истекает                                  : никогда
Минимальное количество дней между сменой пароля                        : 0
Максимальное количество дней между сменой пароля                       : 99999
Количество дней с предупреждением перед деактивацией пароля           : 7
```

Дату устаревания учетной записи можно задать с помощью ключа `-E` со значением в количестве дней с начала эпохи (01.01.1970) или в формате ГГГГ-ММ-ДД. Для отключения устаревания необходимо задать значение -1.

```
localadmin@astra:~$ sudo chage -E 2025-12-31 admin
localadmin@astra:~$ sudo chage -l admin
Последний раз пароль был изменён                                       : июл 24, 2023
Срок действия пароля истекает                                          : никогда
Пароль будет деактивирован через                                       : никогда
Срок действия учётной записи истекает                                  : дек 31, 2025
Минимальное количество дней между сменой пароля                        : 0
Максимальное количество дней между сменой пароля                       : 99999
Количество дней с предупреждением перед деактивацией пароля           : 7
```

Посмотрим, как изменилась строка для пользователя admin в файле `/etc/shadow`:

```
localadmin@astra:~$ sudo grep -E ^admin /etc/shadow
admin:$gost12512hash$hG5F9o[...].:19562:0:99999:7::20453:
```

Полный список ключей доступен в справке `man chage`.

8.5.4 Удаление учетных записей пользователей

С помощью команды `userdel` мы можем удалить пользователя из системы. Как и команда `useradd`, команда `usermod` требует прав суперпользователя.

Синтаксис команды: `userdel <опции> <имя_пользователя>`

Все доступные опции и описание команды можно найти в справке: `man userdel`. Далее рассмотрим частые сценарии применения этой команды.

Если дать команду без ключей, учетная запись выбранного пользователя будет удалена, однако домашний каталог пользователя и его почтовый ящик в каталоге /var/mail (если он существует) останутся. Если пользователь уже вошел в систему, эта команда не будет выполнена.

Войдем в систему в консоли tty1 под пользователем user3 и попробуем его удалить. Повторим то же действие после выхода пользователя user3 из сессии.

Войдите в tty1 под user3, нажав Ctrl + Alt + F1, и переключитесь обратно на tty7 Ctrl + Alt + F7, затем выполните команду в терминале:

```
localadmin@astra:~$ sudo userdel user3
userdel: user user3 is currently used by process 748
```

Выйдите из сессии в tty1 Ctrl + Alt + F1 и `exit`. После закрытия сессии в tty1 переключитесь обратно на tty7 Ctrl + Alt + F7 и выполните удаление повторно:

```
localadmin@astra:~$ sudo userdel user3
localadmin@astra:~$ id user3
id: «user3»: такого пользователя нет
localadmin@astra:~$ ls -l /home
итого 24
drwx------  3 admin      admin      4096 июл 25 11:37 admin
drwx------ 27 localadmin localadmin 4096 июл 18 12:21 localadmin
drwx------  3 user1      user1      4096 июл 25 11:37 user1
drwx------  3 user2      user2      4096 июл 25 11:37 user2
drwx------  3       1150       1150 4096 июл 25 11:38 user3
drwx------  4       1005       1150 4096 июл 26 10:00 user3_bkp
```

Как видим, в первом случае учетная запись не была удалена из-за активной сессии, а во втором - удалена успешно, однако домашний каталог остался нетронутым. Вместо имени пользователя и группы в правах на домашний каталог мы видим их UID и GID. Для удаления домашнего каталога и почтового ящика необходимо применить параметр `-r`. Удалим пользователя user2 и его домашний каталог:

```
localadmin@astra:~$ sudo userdel -r user2
userdel: почтовый ящик user2 (/var/mail/user2) не найден
localadmin@astra:~$ ls -l /home
итого 20
drwx------  3 admin      admin      4096 июл 25 11:37 admin
drwx------ 27 localadmin localadmin 4096 июл 18 12:21 localadmin
drwx------  3 user1      user1      4096 июл 25 11:37 user1
drwx------  3       1150       1150 4096 июл 25 11:38 user3
drwx------  4       1005       1150 4096 июл 26 10:00 user3_bkp
```

Для удаления пользователя, даже если у него открыта активная сессия в системе или если другой

пользователь использует его домашний каталог или его первичную группу, необходимо воспользоваться ключом `-f`. Войдем в систему под пользователем user1 и после этого удалим его.

Войдите в tty1 под user3 и переключитесь обратно на tty7:

```
localadmin@astra:~$ sudo userdel -f -r user1
userdel: user user1 is currently used by process 943
userdel: почтовый ящик user1 (/var/mail/user1) не найден
localadmin@astra:~$ id user1
id: «user1»: такого пользователя нет
localadmin@astra:~$ ls /home
admin  localadmin  user3  user3_bkp
```

Удалим домашние каталоги user3 и user3_bkp.

```
localadmin@astra:~$ rm -rf /home/user3*
```

8.6 Управление учетными записями групп из командной строки

8.6.1 Просмотр групп

Список всех учетных записей групп можно посмотреть в файле `/etc/group`:

```
localadmin@astra:~$ less /etc/group
```

8.6.2 Создание учетных записей групп

Для создания учетных записей групп в консоли используются две команды: `groupadd` и `addgroup`. `Groupadd` — это встроенная команда Linux, которую можно найти в любой системе Linux. Команда `addgroup` не является стандартной командой Linux. Это дружественный интерфейс к программам groupadd и groupmod.

Команда `groupadd` создает новую группу в системе. Команда для успешного выполнения требует привилегий суперпользователя.

Синтаксис команды: `groupadd <имя_группы> <опции>`

Все доступные опции и описание команды можно найти в справке: `man groupadd`. Далее рассмотрим наиболее используемые сценарии применения этой команды.

Чтобы просто создать новую группу, необходимо выполнить команду `sudo groupadd groupname`, где groupname - имя новой группы.

```
localadmin@astra:~$ sudo groupadd programmers
localadmin@astra:~$ grep programmers /etc/group
programmers:x:1201:
```

Создадим группу с заданным GID, для этого воспользуемся ключом `-g`:

```
localadmin@astra:~$ sudo groupadd managers -g 1100
localadmin@astra:~$ grep managers /etc/group
managers:x:1100:
```

8.6.3 Изменение учетных записей групп

С помощью утилиты `groupmod` можно изменить параметры группы, например, GID, имя или пароль.

Синтаксис команды: `groupmod <опции> <имя_группы>`

Изменим имя созданной нами группы programmers:

```
localadmin@astra:~$ sudo groupmod -n programmers_grp1 programmers
localadmin@astra:~$ cat /etc/group | grep programmers
programmers_grp1:x:1201:
```

8.6.4 Создание пароля группы

Чтобы создать пароль группы, используйте утилиту `gpasswd`.

Синтаксис команды: `gpasswd [<опции>] <имя_группы>`

Если команда дана без опций, появится запрос на изменение пароля группы:

```
localadmin@astra:~$ sudo gpasswd programmers_grp1
Изменение пароля для группы programmers_grp1
Новый пароль:
Повторите пароль:
```

Применяя опции, можно создать администратора группы (пользователь с правом добавления членов группы), добавить пользователей группы и т.д.

Пользователь может присоединиться к группе, применив команду `newgrp`. Если группе назначен пароль, пользователь получит приглашение его ввести:

```
localadmin@astra:~$ sudo login user3
Пароль:
user3@astra:~$ newgrp programmers_grp1
Пароль:
user3@astra:~$ groups
programmers_grp1 user3
user3@astra:~$ exit
exit
user3@astra:~$ groups
user3
```

С помощью команды `newgrp` пользователь присоединяется к группе лишь на время. Набрав команду `exit` либо выйдя из системы, пользователь выйдет и из группы, к которой он присоединился командой `newgrp`.

8.6.5 Удаление учетных записей групп

С помощью команды `groupdel` можно удалить учетные записи групп.

Удалим созданную нами группу.

```
sudo groupdel programmers_grp1
```

```
localadmin@astra:~$ sudo groupdel programmers_grp1
localadmin@astra:~$ cat /etc/group | grep programmers
```

8.7 Анализ активности пользователей

Активность пользователей регистрируется в пяти файлах: utmp, btmp, wtmp, faillog и lastlog. Файл utmp расположен в каталоге `/run`, остальные в каталоге `/var/log`. Все эти файлы двоичные, т.е. их содержимое можно посмотреть только специальными утилитами.

Поговорим об этих файлах подробнее:

- `/run/utmp` — содержит информацию о текущих пользователях, включая имя пользователя, терминал и время последнего входа. Используется утилитами вроде `who` и `w`.
- `/var/log/wtmp` — исторический журнал всех входов и выходов пользователей, содержащий информацию о дате и времени каждого события. Используется утилитой `last`.
- `/var/log/btmp` — похож на wtmp, но содержит информацию о неудачных попытках входа пользователей, включая имя пользователя, терминал и причину ошибки. Используется утилитой `lastb`.
- `/var/log/faillog` – хранит число неудачных попыток входа в систему и их предельное количество для каждой учётной записи. Используется утилитой `faillog`.
- `/var/log/lastlog` в Linux хранит информацию о предыдущих входах пользователей в систему. Он содержит данные об имени пользователя, порте и дате последнего входа. Используется утилитой `lastlog`.

8.7.1 Утилиты who и w (what)

С помощью утилиты `who` можно получить информацию обо всех зарегистрировавшихся в системе пользователях. Утилита `w` (what) дополнительно сообщает о том, какую программу они запустили. Информация о зарегистрированных пользователях извлекается из файла `/run/utmp`.

```
root@pc-1:~# who
admin    pts/0        2024-01-30 15:51 (10.0.64.10)
user     pts/2        2024-01-30 18:24 (10.0.64.10)
root@pc-1:~#
root@pc-1:~#
root@pc-1:~# w
 18:25:10 up 1 day, 10:35,  2 users,  load average: 0,07, 0,07, 0,07
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
admin    pts/0    10.0.64.10       15:51    4.00s  0.03s  0.01s sshd: admin [priv]
user     pts/2    10.0.64.10       18:24   46.00s  0.01s  0.00s pager
root@pc-1:~#
root@pc-1:~#
```

8.7.2 Утилиты last и lastb

Утилита `last` - отобразит список всех пользователей, осуществивших вход и выход с момента создания файла `wtmp`. Информация читается из файла `/var/log/wtmp`:

```
root@pc-1:~# last
user     pts/1        10.0.64.10       Tue Jan 30 18:32   still logged in
user     pts/2        10.0.64.10       Tue Jan 30 18:24   still logged in
user     pts/1        10.0.64.10       Tue Jan 30 18:23 - 18:24  (00:01)
user     pts/1        10.0.64.10       Tue Jan 30 18:23 - 18:23  (00:00)
admin    pts/0        10.0.64.10       Tue Jan 30 15:51   still logged in
admin    pts/0        10.0.64.10       Tue Jan 30 07:52 - 12:53  (05:01)
admin    pts/0        10.0.64.10       Mon Jan 29 07:53 - 10:37  (02:43)
```

Утилита `lastb` - отобразит список всех пользователей, вход которых был неудачным. Информация читается из файла `/var/log/btmp`:

```
root@pc-1:~# lastb
user     ssh:notty    10.0.64.10       Tue Jan 30 18:47 - 18:47  (00:00)
test     pts/0                         Tue Jan 30 12:19 - 12:19  (00:00)

btmp begins Tue Jan 30 12:19:15 2024
root@pc-1:~# _
```

8.7.3 Утилита lastlog

Утилита `lastlog` - выводит отчёт о последней регистрации в системе всех или указанного пользователя. Информация читается из файла `/var/log/lastlog`:

```
root@pc-1:~# lastlog
Пользователь     Порт     С              Последний раз
root                                     **Никогда не входил в систему**
daemon                                   **Никогда не входил в систему**
bin                                      **Никогда не входил в систему**
sys                                      **Никогда не входил в систему**
sync                                     **Никогда не входил в систему**
games                                    **Никогда не входил в систему**
man                                      **Никогда не входил в систему**
lp                                       **Никогда не входил в систему**
mail                                     **Никогда не входил в систему**
news                                     **Никогда не входил в систему**
uucp                                     **Никогда не входил в систему**
proxy                                    **Никогда не входил в систему**
www-data                                 **Никогда не входил в систему**
backup                                   **Никогда не входил в систему**
```

8.7.4 Утилита faillog

Утилита `faillog` - показывает записи из файла `/var/log/faillog` или задаёт предел неудачных попыток входа в систему.

Установим максимальное число неудачных попыток входа, после чего учетная запись будет заблокирована:

```
root@pc-1:~# faillog -m 10
root@pc-1:~# _
```

Сбросим счетчик неудачных попыток входа:

```
root@pc-1:~# faillog -a |grep user
user            3       10   01/31/24 13:42:27 +0300  10.0.64.10
root@pc-1:~# faillog -r -u user
root@pc-1:~# faillog -a |grep user
user            0       10   01/31/24 13:42:27 +0300  10.0.64.10
root@pc-1:~# _
```

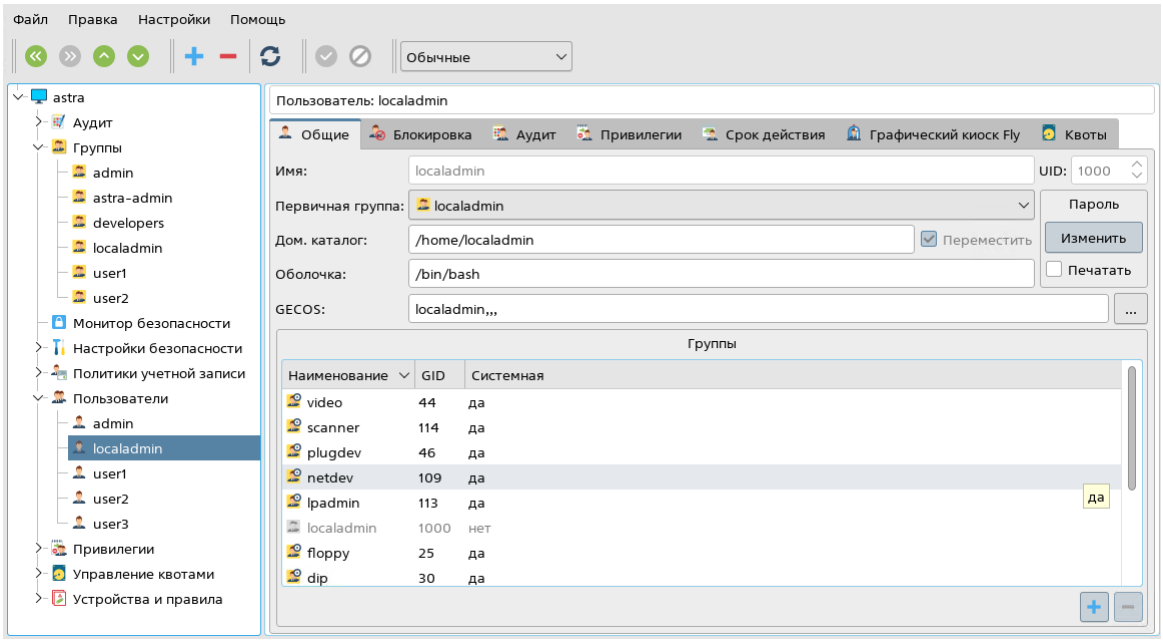
8.7.5 Утилита lslogins

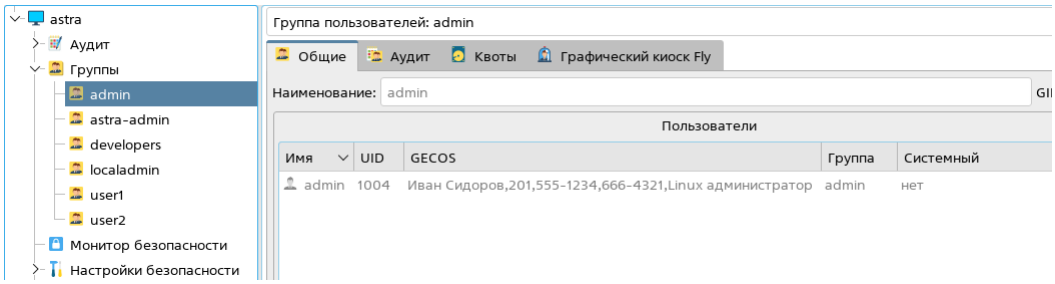
Утилита `lslogins` – выводит информацию обо всех локальных пользователях. Информация считывается из файлов `/etc/passwd`, `/etc/shadow` и `/var/log/wtmp`:

```
root@pc-1:~# lslogins
      UID USER              PROC PWD-LOCK PWD-DENY  LAST-LOGIN GECOS
        0 root               118        0        1             root
        1 daemon               0        0        1             daemon
        2 bin                  0        0        1             bin
        3 sys                  0        0        1             sys
        4 sync                 0        0        1             sync
        5 games                0        0        1             games
        6 man                  0        0        1             man
        7 lp                   0        0        1             lp
        8 mail                 0        0        1             mail
        9 news                 0        0        1             news
       10 uucp                 0        0        1             uucp
       13 proxy                0        0        1             proxy
```

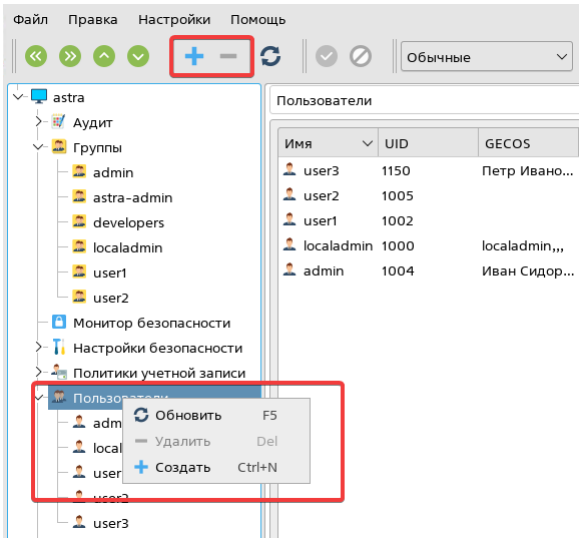
8.8 Управление пользователями и группами в графическом интерфейсе

В графическом интерфейсе доступен удобный и интуитивно понятный интерфейс для управления пользователями и группами «Политика безопасности». Вызвать его можно через меню *Пуск* → *Панель управления* → *Безопасность* → *Политика безопасности* или из командной строки по названию утилиты fly-admin-smc.





С помощью контекстного меню на пунктах *Группы* и *Пользователи* или с помощью элементов управления + и - можно добавлять и удалять пользователей и группы.



Атрибуты и другие параметры учетных записей пользователей и групп распределим по вкладкам. Для пользователей предназначение вкладок следующее:

- **Общие** – основные атрибуты учетной записи и пароль.
- **Блокировка** – блокировка пароля и учетной записи.
- **Аудит** – настройка аудита событий, вызванных действиями пользователя.
- **Привилегии** – установка Linux и Parsec привилегий.
- **МРД** – установка мандатных атрибутов пользователя.
- **Срок действия** – сроки действия пароля и учетной записи.
- **Графический киоск** – настройка работы компьютера, когда основное приложение открывается в полный экран (режим графического киоска).
- **Квоты** – настройка дисковых квот для пользователя.

После внесения изменений их можно принять или отклонить.



8.9 Практика и тестирование

8.9.1 Практическая работа: Модуль 8. Управление пользователями и группами

См. также *Требования, правила и цели выполнения практической работы*

8.9.1.1 Практические задания

Задание 1.

1. Создайте пользователя user1 и user2. У них должны быть созданы домашние каталоги, и они должны иметь оболочку входа /bin/bash.
2. Назначьте пароли для пользователей user1, user2.
3. Назначьте для каждого пользователя минимальный срок действия пароля 1 день, максимальный срок действия пароля 30 дней, время предупреждения об истечении пароля 3 дня, период неактивности 3 дня.

Задание 2.

1. Создайте группы group1, group2, group3. Назначьте пароль группе group3 и ее администратором сделайте пользователя user1.
2. Включите пользователей user1, user2 в группы group1, group2.
3. Войдите в систему под пользователем user2 и получите права группы group3. Проверьте, что после ввода пароля user2 состоит в группе group3. Выйдите из системы.
4. Войдите в систему под пользователем user1 и включите user2 в группу group3. Выйдите из системы.

Задание 3.

1. Заблокируйте пользователя user2.
2. Принудительно заставьте пользователя user1 сменить свой пароль. Войдите под user1 в систему и смените пароль. Выйдите из системы.
3. Смените имя пользователя user2 на user3. Разблокируйте пользователя user3 и войдите в систему под его именем.
4. Перенесите домашние каталоги пользователей user1, user3 в /mnt/home. Проверьте, что домашние каталоги пользователей были успешно перенесены.

Задание 4.

1. Просмотрите журналы удачных и неудачных входов пользователей user1, user2.

2. Удалите группы group1, group2, group3.

3. Удалите пользователей user1, user3 вместе с их домашними каталогами.

8.9.1.2 Ответы на практические задания (пошаговые инструкции)

Задание 1.

1. Создайте пользователя user1 и user2. У них должны быть созданы домашние каталоги, и они должны иметь оболочку входа /bin/bash.

   Выполните:

```
useradd -m -s /bin/bash user1
useradd -m -s /bin/bash user2
```

2. Назначьте пароли для пользователей user1, user2.

   Выполните:

```
passwd user1
passwd user2
```

3. Назначьте для каждого пользователя минимальный срок действия пароля 1 день, максимальный срок действия пароля 30 дней, время предупреждения об истечении пароля 3 дня, период неактивности 3 дня.

   Выполните:

```
chage -m 1 -M 30 -W 3 -I 3 user1
chage -m 1 -M 30 -W 3 -I 3 user2
```

Задание 2.

1. Создайте группы group1, group2, group3. Назначьте пароль группе group3 и ее администратором сделайте пользователя user1.

   Выполните команды:

```
groupadd group1
groupadd group2
groupadd group3
gpasswd group3
gpasswd -A user1 group3
```

   Команда `gpasswd` устанавливает пароль на группу group3 и добавляет администратора группы group3.

2. Включите пользователей user1, user2 в группы group1, group2.

   Добавить пользователя в группу можно также с помощью команды `adduser`, например, `adduser user1 group1`.

```
usermod -a -G group1 user1
usermod -a -G group1 user2
```

3. Войдите в систему под пользователем user2 и получите права группы group3. Проверьте, что после ввода пароля user2 состоит в группе group3. Выйдите из системы.

   Вход можно выполнить по ssh или с помощью команды `login`.

```
login user2
newgr - group3
id user2
logout
logout
```

   Команда `id user2` должна показать, что user2 входит в состав группы group3. Команду `logout` необходимо выполнить дважды, так как newgr запускает еще одну оболочку.

4. Войдите в систему под пользователем user1 и включите user2 в группу group3. Выйдите из системы. Войдите в систему под пользователем user1 и запустите `gpasswd`.

```
login user1
gpasswd -a user2 group3
id user2
```

   Команда `id user2` должна показать, что user2 входит в состав группы group3.

Задание 3.

1. Заблокируйте пользователя user2.

   Выполните команды:

```
passwd -l user2
```

2. Принудительно заставьте пользователя user1 сменить свой пароль. Войдите под user1 в систему и смените пароль. Выйдите из системы.

   Смените пароль пользователя user1 принудительно.

```
chage -d 0 user1
login user1
logout
```

   После команды `login` будет запрошена смена текущего пароля для user1.

3. Смените имя пользователя user2 на user3. Разблокируйте пользователя user3 и войдите в систему под его именем.

   Смените имя пользователя user2 на user3.

```
usermod -l user3 user2
id user2
id user3
login user3
logout
```

4. Перенесите домашние каталоги пользователей user1, user3 в /mnt/home. Проверьте, что домашние каталоги пользователей были успешно перенесены.

   Создайте директорию /mnt/home и перенесите домашние каталоги user1 и user3 в новое место. Введите команды:

```
mkdir /mnt/home/
usermod -m -d /mnt/home/user1 user1
usermod -m -d /mnt/home/user3 user3
ls -al /mnt/home
```

Задание 4.

1. Просмотрите журналы удачных и неудачных входов пользователей user1, user2.

   Выполните команды:

```
last user1
last user3
lastb user1
lastb user3
```

2. Удалите группы group1, group2, group3.

   Удалите группы:

```
groupdel group1
groupdel group2
groupdel group3
```

3. Удалите пользователей user1, user3 вместе с их домашними каталогами.

   Удалите пользователей:

```
userdel -r user1
userdel -r user3
```

8.9.2 Тест: Модуль 8. Управление пользователями и группами

Пожалуйста, ответьте на несколько вопросов, чтобы закрепить полученные знания. Правильные ответы для самопроверки вы сможете найти в конце списка. Обратите внимание, что у некоторых вопросов может быть несколько верных вариантов ответов.

8.9.2.1 Вопросы

Вопрос 1. Назовите четыре файла, в которых содержится информация о пользователях и группах?

a. Файл `/etc/passwd`

b. Файл `/etc/user`

c. Файл `/etc/passwords`

d. Файл `/etc/shadow`

e. Файл `/etc/shadows`

f. Файл `/etc/group`

g. Файл `/etc/users`

h. Файл `/etc/groups`

i. Файл `/etc/gshadow`

j. Файл `/etc/gshadows`

Вопрос 2. Под каким именем сохраняется резервная копия файла `/etc/shadow`?

a. Файл `/etc/shadow.backup`

b. Файл `/etc/shadow-`

c. Файл `/etc/shadow.old`

d. Такого файла нет

Вопрос 3. Что означает символ `x` в поле пароля в файле `/etc/passwd`?

a. Пароль не установлен

b. Пользователь заблокирован

c. Пароль находится в файле `/etc/shadow`

d. Пароль состоит из одного символа `x`

Вопрос 4. Какой файл хранит настройки диапазонов UID и GID?

a. Файл `/etc/login.defs`

b. Файл `/etc/default/useradd`

c. Файл `/etc/adduser.conf`

d. Файл `/etc/sudoers`

Вопрос 5. Что делает команда `useradd -D` ?

a. Удаляет всех пользователей

b. Показывает настройки по умолчанию для команды `useradd`

c. Удаляет настройки по умолчанию для команды `useradd`

d. Показывает опции команды `useradd`

Вопрос 6. Какой UID будет присвоен первому пользователю операционной системы?

a. 1000

b. 1001

c. 100

d. 101

Вопрос 7. Какие две команды могут быть использованы для блокировки и разблокировки пользователя?

a. Команды `passwd` и `chmod`

b. Команды `usermod` и `passwd`

c. Команды `usermod` и `chage`

d. Команды `users` и `lslogins`

Вопрос 8. Какая команда заставит пользователя `iivanov` принудительно сменить свой пароль при следующем входе?

a. Команда `chage -d 0 iivanov`

b. Команда `usermod -u iivanov`

c. Команда `passwd -l iivanov`

d. Команда `id iivanov`

Вопрос 9. Что делает команда `faillog -u iivanov -r`?

a. Выводит информацию о пользователе `iivanov`

b. Показывает записи о неудачных попытках входа для `iivanov`

c. Очищает записи о неудачных попытках входа для `iivanov`

d. Сбрасывает счетчик неудачных попыток входа для `iivanov`

Вопрос 10. Какой файл хранит даты последнего входа пользователей?

a. Файл `/var/log/utmp`

b. Файл `/var/log/wtmp`

c. Файл `/var/log/lastlog`

d. Файл `/var/log/auth.log`

Ответы на вопросы

1. a, d, f, i
2. b
3. c
4. a
5. b
6. a
7. c
8. a
9. d
10. c

8.10 Заключение

В этом модуле мы узнали, что такое UID/GID и чем они отличаются от SID. Мы познакомились с конфигурационными файлами, отвечающими за хранение данных о локальных пользователях и группах. Мы научились управлять субъектами безопасности как из командной строки, так и с помощью графической оснастки.

В следующем модуле мы познакомимся с классической моделью управления правами доступа в Linux, возможностями расширенных списков доступа POSIX ACL и разберемся, почему 555 намного безопаснее 777.

8.11 Дополнительные источники информации

- Алгоритм защищенного преобразования ГОСТ Р 34.11.-2012,

8.12 Обратная связь

Если остались вопросы, то их всегда можно задать в специальной теме.

Модуль 9. Работа с правами доступа, ACL

9.1 Введение

Из этого модуля мы узнаем, чем отличается управление доступом в Windows и Linux, что такое модель UGO и как на самом деле работает алгоритм проверки прав доступа к файлам в Linux.

Правильная настройка прав доступа крайне важна для безопасной эксплуатации ИТ-инфраструктуры. Например, если всем пользователям Linux будет разрешено чтение файла `/etc/shadow`, в котором содержатся хеши паролей, то злоумышленник сможет взломать любой аккаунт, а если пользователям будет доступна запись на исполняемый файл какой-либо службы, то злоумышленник сможет выполнять в системе любые действия с повышенными привилегиями, ни в чем себе не отказывая.

Как говорят ИБэшники: «Самый безопасный компьютер - это тот, который отключен от сети... от сети в 220 Вольт». Если же вы не можете позволить себе столь кардинальные меры защиты, то материалы этого модуля вам будут жизненно необходимы.

9.2 Управление доступом через идентификаторы, ресурсы и разрешения

Система Linux разрабатывалась как многопользовательская операционная система, поэтому в ней изначально был реализован механизм дискреционного разграничения прав доступа. **Дискреционное** или избирательное управление доступом (*от англ. discretion – по усмотрению*) — это управление доступом **субъектов** безопасности к **ресурсам** на основе **разрешений**. Рассмотрим каждый из компонентов этой триады подробнее.

В операционной системе Linux мы работаем с такими **субъектами**, как пользователи и группы, причем так же, как в Windows, в состав локальных групп можно включать только учетные записи пользователей, а вложенные группы доступны только при работе компьютера в составе домена.

Как мы уже знаем, для того чтобы мы могли отличить один субъект от другого, пользователям назначаются идентификаторы пользователей (User Identifier, UID), а группам — идентификаторы групп (Group Identifier, GID), и оба вида POSIX-идентификаторов представляют собой целые числа в диапазоне от 0 до 2^{32} (4 294 967 296).

В части работы с идентификаторами субъектов в Linux много отличий от Windows. Например, в Windows идентификаторы пользователей и групп находятся в одном пространстве, а в Linux это два разных множества, поэтому пользователь root с идентификатором 0 и одноименная группа root с идентификатором 0 — это два совершенно разных субъекта безопасности. Еще одной интересной особенностью является то, что один и тот же uid можно назначить нескольким пользователям, в то время как Windows строго следит за этим и не позволяет использовать один и тот же SID дважды. Но все эти моменты мы уже разбирали в предыдущем модуле, поэтому не будем на них останавливаться.

Вторым элементом избирательного управления доступом являются **ресурсы**. В случае обычной файловой системы ресурсами являются файлы и папки, а в случае Linux, как мы знаем, файлы являются также интерфейсами для доступа к устройствам, ядру операционной системы, другим процессам и т. д. Метаинформация о файлах (такая как дата и время создания, количество жестких ссылок, владелец, группа и права доступа), как вы помните, хранится в структуре данных, которая называется айнодой. И эти айноды похожи на записи MFT с некоторыми отличиями.

И, наконец, **разрешения** определяют, какие действия могут быть выполнены субъектами по отношению к ресурсам. И в соответствии с базовой моделью безопасности в Linux доступны три разрешения — чтение (**R**ead), запись (**W**rite) и выполнение (e**X**ecute).

Значение разрешений rwx зависит от контекста:

1. Разрешения применительно к файлам, см. табл. 1.3:

- **Чтение (r)** — дает право на чтение содержимого файла.

- **Запись (w)** — позволяет полностью переопределить содержимое файла или добавить новые данные в конец. При этом не обязательно иметь права на чтение файла.

- **Выполнение (x)** — актуально для бинарных исполняемых файлов. Если файл является скриптом sh или python, то для его выполнения интерпретатором достаточно, чтобы файл был доступен для чтения. Если же запускать скрипт напрямую по имени `./test.sh`, то вам потребуются права на выполнение.

табл. 1.3 – Пример выполнения команд с разными правами доступа на файл

Пользователь	root	обыч.	обыч.	обыч.	обыч.
Права на папку	- - -	- - x	- - x	- - x	- - x
Права на файл	- - -	- - -	r - -	- w -	- - x
`cat /folder/file.txt`	+	-	+	-	-
`echo "hello" > /folder/file.txt`	+	-	-	+	-
`echo "hello" >> /folder/file.txt`	+	-	-	+	-
`sh /folder/my.sh`	+	-	+	-	-
`/folder/cmd.bin (запуск)`	-	-	-	-	-

2. Значение разрешений применительно к папкам, см. табл. 1.4:

- **Чтение (r)** — дает право на чтение имен дочерних объектов, но вы не сможете обратиться к айнодам и прочитать их атрибуты.

- **Запись (w)** — имеет силу, только если есть доступ на выполнение. Расширяет этот доступ правами на создание новых и переименование/удаление существующих файлов и папок.

- **Выполнение (x)** — дает право на вход в папку (cd), чтение метаинформации по дочерним объектам и доступ к ним в соответствии с установленными правами доступа.

  Другими словами, если у пользователя есть право на чтение файла, то для того чтобы он смог воспользоваться этим правом, ему нужны права на выполнение для всех вышестоящих каталогов.

табл. 1.4 – Пример выполнения команд с разными правами доступа на папку

Пользователь	root	обыч.	обыч.	обыч.	обыч.
Права на папку	- - -	r - -	- w -	- - x	- w x
Права на файл	- - -	- - -	- - -	- - -	- - -
`ls /folder/file.txt`	+	-	-	+	+
`ls /folder`	+	+	-	-	-
`cd /folder`	+	-	-	+	+
`stat /folder/file.txt`	+	-	-	+	+
`mkdir /folder/subfolder`	+	-	-	-	+
`mv /folder/file /folder/file2`	+	-	-	-	+
`touch /folder/newfile.txt`	+	-	-	-	+
`rm /folder/newfile.txt`	+	-	-	-	+

Разрешения rwx группируются по три бита в восьмеричное число от 0 до 7, причем биты следуют в обратном порядке:

- $2^0 = 1$ — Исполнение
- $2^1 = 2$ — Запись
- $2^2 = 4$ — Чтение

Полному доступу, например, соответствует число 7 (4 чтение + 2 запись + 1 исполнение), а на общие каталоги (обычно) полных прав не дают и урезают права до 5 (4 чтение + 1 исполнение).

Примечание: Лучшей практикой считается выдавать пользователям минимально возможные права, которых будет достаточно для выполнения ими должностных обязанностей. То же самое относится и к служебным приложениям: если на сервере работает Apache, то крайне желательно, чтобы эта служба работала не от суперпользователя, а из-под своей собственной учетной записи, например, www-data. Тогда в случае взлома веб-сервера злоумышленники будут ограничены правами учетной записи www-data, что значительно сократит количество дальнейших векторов атак.

9.3 Модель UGO «пользователь — группа — остальные»

Операционная система Linux унаследовала от UNIX простую модель безопасности «пользователь — группа — остальные» (от англ. User — Group — Others), в соответствии с которой у каждого объекта файловой системы есть три категории пользователей:

- **Пользователь** — это пользователь, который считается владельцем объекта. Владелец не толь-

ко получает права, определенные для этой категории, но и может изменять права доступа на объект с помощью команды chmod (от англ. change mode - изменить режим).

- **Группа** — это группа пользователей, участникам которой можно назначить уникальные права доступа. Эта группа считается владельцем объекта, но, несмотря на это, участники группы не могут изменять права доступа к объекту. Изменять права может только пользователь-владелец и суперпользователь.

- **Остальные** — это все остальные пользователи, не вошедшие в две предыдущие категории.

Примечание: Иногда эту модель называют «владелец — группа-владелец — остальные» (owner — group — others), но во избежание путаницы лучше придерживаться стандартной терминологии.

Сменить владельца можно с помощью утилиты `chown`, но привилегия CAP_CHOWN, позволяющая произвольно менять значения UID/GID для любых файлов, по умолчанию доступна только суперпользователю. Владелец файла с помощью `chown` может изменить только группу-владельца и только на ту группу, участником которой является сам. Если можно было бы устанавливать любую группу, то в сочетании с флагами sgid и execute это создавало бы большую угрозу безопасности.

В следующем примере мы установим пользователя ivanov и группу drivers владельцами файла `test.txt`, используя привилегии суперпользователя:

```
localadmin@astra:~$ touch test.txt
localadmin@astra:~$ sudo chown ivanov:drivers test.txt
```

Информация о том, какой пользователь является владельцем и какой группе принадлежит файл, записана в айноде. Там же хранится информация о том, какие права доступа определены этим категориям пользователей.

Алгоритм определения прав доступа работает строго «слева направо», о чем (к величайшему сожалению) не догадывается подавляющее большинство администраторов Linux, см. рис. 1.39:

- Если пользователь является владельцем, он получает права пользователя и процедура проверки прав доступа завершается.

- Если пользователь не является владельцем, но входит в состав участников группы, то он получает права группы и процедура завершается.

- Если пользователь не является ни владельцем, ни участником группы, то он получает права, предназначенные для всех остальных пользователей.

Это означает, что пользователь не сможет прочитать файл, на который установлены права 007, если этот пользователь является владельцем этого файла. Владелец просто не попадает в категорию

остальных пользователей, он сразу получает «ноль», и на этом проверка прав доступа завершается. Однако, являясь владельцем файла, пользователь всегда сможет поменять права и после этого получить доступ к содержимому файла.

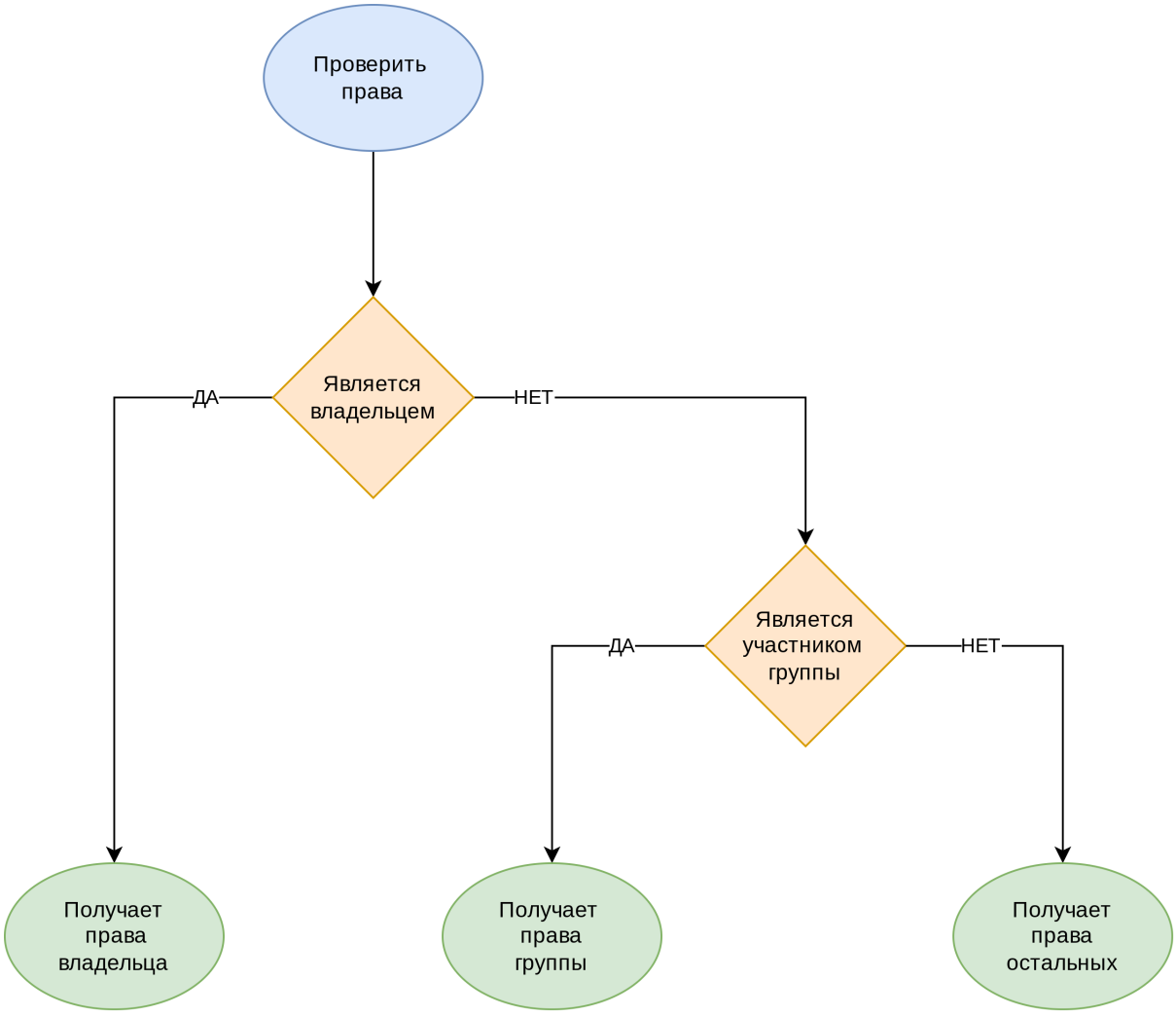


рис. 1.39 – Базовый алгоритм определения прав доступа к объекту файловой системы Linux

9.4 Специальные флаги suid, sgid, sticky

Когда мы запускаем какую-нибудь утилиту (например, `cat`), операционная система создает процесс и передает ему соответствующие параметры. Посмотреть список процессов можно с помощью команды `ps -aux`.

Каждый процесс Linux запускается от имени конкретного пользователя и получает права доступа, соответствующие этому пользователю и группам, участником которых он является. Если процесс создает объекты файловой системы, то в айноде этих объектов в качестве владельца вписывается uid пользователя, запустившего процесс, а в качестве группы — gid первичной группы этого пользователя.

Кроме стандартных прав доступа в айноде хранятся три специальных флага SetUID, SetGID и Sticky bit, которые позволяют изменить описанную выше логику. Значение этих флагов зависит от контекста.

9.4.1 Значения специальных флагов к исполняемым файлам

SetUID (suid) на исполняемый файл:

Если установлен флаг suid на исполняемый файл, то при запуске приложения операционная система выполнит его от имени того пользователя, который является владельцем этого файла (т.е. подменит эффективный идентификатор пользователя на uid владельца исполняемого файла).

Такой фокус нам показывает, например, утилита `passwd` – напрямую у нас нет прав редактировать файл /etc/shadow, но через suid все невозможное становится возможным.

Продемонстрируем работу suid с помощью утилиты `id`:

Вначале определяем место расположения утилиты:

```
localadmin@astra:~$ which id
/usr/bin/id
```

Копируем утилиту в домашний каталог, меняем владельца и выставляем suid:

```
localadmin@astra:~$ cp /usr/bin/id ~/
localadmin@astra:~$ sudo chown root:root ~/id
localadmin@astra:~$ sudo chmod u=s,g=,o=x ~/id
```

Вызываем штатную утилиту

```
localadmin@astra:~$ id

uid=491200000(admin) gid=491200000(admins)
группы=491200000(admins),113(lpadmin),1001(astra-admin),491200006(fs_users)
```

Вызываем утилиту, для которой установлен suid, см. euid=0(root)

```
localadmin@astra:~$ ~/id

uid=491200000(admin) gid=491200000(admins) euid=0(root)
группы=491200000(admins),113(lpadmin),1001(astra-admin),491200006(fs_users)
```

SetGID (sgid) на исполняемый файл:

Если установлены флаги sgid и execute на исполняемый файл, то при запуске приложения операционная система заменит у процесса его эффективный gid на тот gid, который задан для файла, и добавит этот gid в список групп, участником которых является владелец процесса, поэтому процесс получит в том числе те права, которые будут назначены группе, являющейся владельцем

исполняемого файла. Пожалуй, без полбита и не разберешься.

Рассмотрим на практическом примере, но обратите внимание на то, что флаг sgid не будет работать без прав на execute для группы, т.е. в выводе утилиты `ls` вы должны видеть строчную «s»:

Устанавливаем флаг sgid без execute, работать не будет, а прописная S указывает, что флаг выполнения не установлен:

```
localadmin@astra:~$ sudo chmod u=,g=s,o=x ~/id
localadmin@astra:~$ ls -l ~/id
------S--x 1 root root 48336 фев 22 00:00 /home/admin/id
```

Устанавливаем флаг sgid вместе с execute, теперь будет работать, и строчная буква s указывает, что флаг выполнения установлен:

```
localadmin@astra:~$ sudo chmod u=,g=sx,o=x ~/id
localadmin@astra:~$ ls -l ~/id
------s--x 1 root root 48336 фев 22 00:00 /home/admin/id
```

Вызываем штатную утилиту id для просмотра прав:

```
localadmin@astra:~$ id
uid=491200000(admin) gid=491200000(admins)
группы=491200000(admins),113(lpadmin),1001(astra-admin),491200006(fs_users)
```

Вызываем утилиту, для которой установлены флаги:

```
localadmin@astra:~$ ~/id
uid=491200000(admin) gid=491200000(admins) egid=0(root)
группы=0(root),113(lpadmin),1001(astra-admin),491200000(admins),491200006(fs_users)
```

Sticky bit на исполняемый файл:

Когда-то давно этот флаг позволял оставлять программу в памяти после завершения процесса для ускорения повторного запуска приложения, отсюда и его название «липкий бит». Но в современных Linux системах этот флаг более не имеет значения для исполняемых файлов и используется только для папок.

9.4.2 Значения специальных флагов к папкам

SetUID (suid) для папок:

Данный флаг на папку не имеет значения.

SetGID (sgid) для папок:

Если на папку установлен флаг sgid, то при создании объектов внутри этой папки в качестве группы-владельца им будет назначаться та же группа, которая является владельцем родительской

папки.

```
localadmin@astra:~$ mkdir folder
localadmin@astra:~$ sudo chown root:root folder
localadmin@astra:~$ sudo chmod u=,g=s,o=rwx folder
localadmin@astra:~$ ls -l
...
d-----Srwx 2 root root 4096 фев 22 00:25 folder
...
localadmin@astra:~$ touch folder/test.txt
localadmin@astra:~$ ls -l folder/test.txt
-rw-r--r-- 1 admin root 0 фев 22 00:26 test.txt
```

Sticky bit для папок: — если на общую папку установлен липкий бит, то удалить или переименовать объекты внутри этой папки сможет только владелец этого объекта, (реальное применение папка /tmp).

```
localadmin@astra:~$ sudo mkdir folder
localadmin@astra:~$ sudo touch folder/test.txt

localadmin@astra:~$ sudo chmod u=,g=,o=rwxt folder/
localadmin@astra:~$ sudo chmod u=,g=,o=rwx folder/test.txt
localadmin@astra:~$ mv folder/test.txt folder/test2.txt
mv: невозможно переместить 'folder/test.txt' в 'folder/test2.txt': Операция не␣
↪позволена

localadmin@astra:~$ sudo chmod u=,g=,o=rwx folder/
localadmin@astra:~$ mv folder/test.txt folder/test2.txt
localadmin@astra:~$ ls -l folder/
итого 0
-------rwx 1 root root 0 фев 22 00:47 test2.txt

localadmin@astra:~$ sudo chmod u=,g=,o=rwxt folder/
localadmin@astra:~$ sudo chown admin:admins folder/test2.txt
localadmin@astra:~$ mv folder/test2.txt folder/test3.txt
localadmin@astra:~$ ls -l folder/
итого 0
-------rwx 1 admin admins 0 фев 22 00:47 test3.txt

root@astra:~# ls -la /tmp | head -2
итого 48
drwxrwxrwt 11 root       root       4096 сен 14 15:02 .
```

Несмотря на то, что специальные флаги назначают с помощью операторов +/-/= применительно к разным категориям пользователей, например, suid через u+s, sgid через g+s, а sticky через o+t, эти флаги так же, как разрешения rwx, группируются в отдельное восьмеричное число в следующем порядке:

- $2^0 = 1$ — Sticky
- $2^1 = 2$ — SetGID
- $2^2 = 4$ — SetUID

При назначении прав доступа через число специальные флаги задаются в первом разряде, поэтому для назначения липкого бита на папку с правами rw-r–r– нужно использовать число 1644:

```
localadmin@astra:~$ sudo chmod 1644 folder/
localadmin@astra:~$ ls -l
drw-r--r-T 2 root root 4096 фев 20 15:51 folder
```

9.5 Понятие umask

Новые объекты файловой системы создаются с **правами по умолчанию**, значение которых зависит от типа создаваемого объекта: каталоги создаются с правами **777**, а обычные текстовые файлы с правами **666**. Однако, если вы проверите это на практике, то обнаружите, что у новых каталогов права 755, а у новых файлов 644, и происходит так по той простой причине, что на значение прав по умолчанию дополнительно **накладывается** так называемая **маска**.

Маска позволяет ограничить права доступа «сверху», указывая, какие биты следует сбрасывать. Если маска будет равна 000, то она не будет никак ограничивать права доступа к новым объектам. Если же выставить маску 777, то при создании новых объектов ни у кого не будет доступа к ним.

Чтобы получить текущее значение маски, вызовите утилиту `umask` с ключом -p или -S:

```
localadmin@astra:~$ umask -p
umask 0022
localadmin@astra:~$ umask -S
u=rwx,g=rx,o=rx
```

Обратите внимание, что утилита возвращает четырехразрядное число, но первый разряд не является маской для специальных флагов, маска так не работает. Ведущий ноль в этой строке просто придает числу формат восьмеричной константы в стиле языка программирования Си. Утилиты оболочки bash всегда интерпретируют разрешения как числа в восьмеричной системе, поэтому ведущий ноль не является обязательным.

Чтобы установить значение маски 0022, мы можем использовать любую из следующих команд:

```
localadmin@astra:~$ umask 022
localadmin@astra:~$ umask 0022
localadmin@astra:~$ umask u=rwx,g=rwx,o=
```

Задавая маску в текстовом представлении, категории пользователей можно объединять или даже задавать маску сразу для всех категорий с помощью параметра a (all).

```
localadmin@astra:~$ umask ug=rwx,o=rx
localadmin@astra:~$ umask a=rwx
```

Также имеется возможность работы с отдельными битами. С помощью операторов +/- вы можете включить/отключить определенный бит маски, оставляя другие биты без изменений.

```
localadmin@astra:~$ umask ug-w
localadmin@astra:~$ umask a+w
```

Помимо прочего, вы можете комбинировать два предыдущих способа. Например, разрешить пользователю все операции, а группе и остальным пользователям убрать право на чтение.

```
localadmin@astra:~$ umask u=rwx,go-r
```

Утилита `umask` влияет на работу пользователя только в пределах одной сессии терминала. Для установления значения маски по умолчанию вы можете задать значение параметра UMASK в файлах `/etc/default/login` или `/etc/login.defs`.

Примечание: Вам следует обратить внимание на различия между утилитами `umask` и `chmod`. Утилита `chmod` устанавливает права доступа на уже существующие объекты, а утилита `umask` влияет на то, с какими правами доступа будут создаваться новые объекты.

9.6 Атрибуты файлов

Помимо прав доступа, каждый из файлов стандартной файловой системы Linux имеет набор атрибутов, регламентирующих особенности работы с ним. Атрибуты поддерживаются такими файловыми системами Linux, как Ext4, Btrfs и XFS. Текущие установленные атрибуты можно посмотреть с помощью утилиты `lsattr`:

```
localadmin@astra:~$ touch test.txt
localadmin@astra:~$ lsattr test.txt
--------------e---- test.txt
```

Наиболее важными параметрами утилиты `lsattr` являются:

- `-R` — позволяет рекурсивно выводить атрибуты файлов в дереве папок;
- `-a` — позволяет выводить информацию и об атрибутах скрытых файлов;
- `-d` — позволяет выводить информацию об атрибутах папок вместо обработки их содержимого.

В том случае, если утилите не передаются имена файлов, она выводит информацию об атрибутах файлов из текущей папки.

Основные файловые атрибуты приведены в таблице:

табл. 1.5 – Файловые атрибуты

Атрибут	Значение
A	Запрещает обновлять метку времени доступа к файлу.
a	Автоматически устанавливает режим дополнения при открытии файла для записи.
C	Запрещает использовать механизм копирования при записи (Copy-on-Write) при модификации содержимого файла.
D	При применении к папке активирует режим синхронной записи изменений содержащихся в ней файлов.
d	Запрещает утилите `dump` создавать резервную копию файла.
E	Указывает на ошибку сжатия содержимого файла ядром ОС (не может быть установлен пользователем).
e	Указывает на использование экстентов для ссылок на соответствующие файлу дисковые блоки (не может быть установлен пользователем).
h	Указывает на то, что размер файла исчисляется в количестве блоков ФС, а не ее секторов, то есть размер файла превышал или превышает в данный момент 2 ТБ (не может быть установлен пользователем).
I	Указывает на то, что содержимое директории было проиндексировано утилитой `htree`.
i	Запрещает всем пользователям, в том числе супрепользователю, модифицировать файл, а именно: записывать в него данные, удалять, переименовывать или создавать ссылки на него. Например, если вы хотите запретить скриптам установки службы каталога вносить правки в файл `/etc/resolv.conf`, вы можете установить этот атрибут для указанного файла.
j	Принудительно активирует режим журналирования ФС при записи данных в файл.
s	Активирует механизм надежного удаления, автоматически записывающий нулевые блоки на диск после удаления файла пользователем.
S	Активирует режим синхронной записи изменений содержимого файла на диск.
T	При применении к папке указывает на то, что ее подпапки не связаны и могут размещаться в отдельных группах блоков.
t	Запрещает оптимизации использования блоков файла.
u	Запрещает удаление содержимого файла при его удалении из ФС с целью получения возможности его последующего восстановления.
X	Указывает на возможность прямого доступа к содержимому сжатого ядром ОС файла (не может быть установлен пользователем).
Z	Указывает на неактуальность сжатого ядром ОС файла (не может быть установлен пользователем).

Для изменения атрибутов используется команда `chattr`.

Примеры использования:

```
root@astra:~# id
uid=0(root) gid=0(root) группы=0(root)

root@astra:~# touch test.txt
root@astra:~# chattr +i test.txt
root@astra:~# lsattr test.txt
----i---------e---- test.txt

root@astra:~# echo "qwe" > test.txt
bash: test.txt: Операция не позволена
root@astra:~# rm -f test.txt
rm: невозможно удалить 'test.txt': Операция не позволена
root@astra:~# touch test2.txt
root@astra:~# echo "Text 1" > test2.txt
root@astra:~# cat test2.txt
Text 1
root@astra:~# chattr +a test2.txt
root@astra:~# rm -f test2.txt
rm: невозможно удалить 'test2.txt': Операция не позволена
root@astra:~# echo "Text NEW" > test2.txt
bash: test2.txt: Операция не позволена
root@astra:~# echo "Text NEW" >> test2.txt
root@astra:~# cat test2.txt
Text 1
Text NEW
```

9.7 Списки доступа ACL

Базовая модель безопасности Linux «пользователь — группа — остальные» проста и эффективна, но не обладает достаточной гибкостью, чтобы с ее помощью эффективно решать задачи по организации совместного доступа сотрудников к файлам.

Например, если у вас есть группа IT сотрудников и группа бухгалтеров, то для совместной работы над счетами от поставщиков техники вам потребуется создать третью группу, куда следует поместить всех пользователей первых двух групп, т.к. локальные группы не могут быть участниками других групп. Внедрение домена частично решает эту задачу, т.к. появляется возможность использовать вложенные группы, но более простым и эффективным решением является использование списков доступа (Access Control Lists, ACL).

Впервые списки доступа появились в 1994 году в файловой системе NTFS от Microsoft, что стало одним из факторов, который помог этой операционной системе занять лидирующие позиции в корпоративном сегменте рынка. В Linux этот механизм появился в 2002 году, поэтому отставания уже давно нет. В Astra Linux ядро скомпилировано уже с поддержкой POSIX ACL, поэтому данные функции доступны «из коробки» и ничего дополнительно делать не потребуется.

Если администраторов, разбирающихся в работе базового алгоритма проверки прав доступа Linux, просто мало, то специалистов, которые хорошо ориентируются в версии, усложненной POSIX ACL, не сыщешь днем с огнем. Отчасти так происходит еще и потому, что этот алгоритм практически нигде нормально не описан, поэтому давайте разбираться в деталях, см. рис. 1.40:

- На первом шаге все то же самое. Если пользователь является владельцем, он получает права пользователя и процедура завершается.

- А со второго шага начинаются отличия. Если пользователь не является владельцем, но включен в ACL как пользователь, то он получает права, указанные в списке доступа с учетом маски, и процедура завершается. Маска — это специальное правило POSIX ACL, которое определяет максимально возможные права доступа.

- Если пользователь не является владельцем и не включен в список ACL как пользователь, то проверка переходит на группы. Если пользователь является участником группы-владельца или одной из групп ACL, то он получает сумму этих прав с учетом маски и процедура завершается.

- Если пользователь не попал ни в одну из предыдущих категорий, то он получает права, предназначенные для всех остальных пользователей.

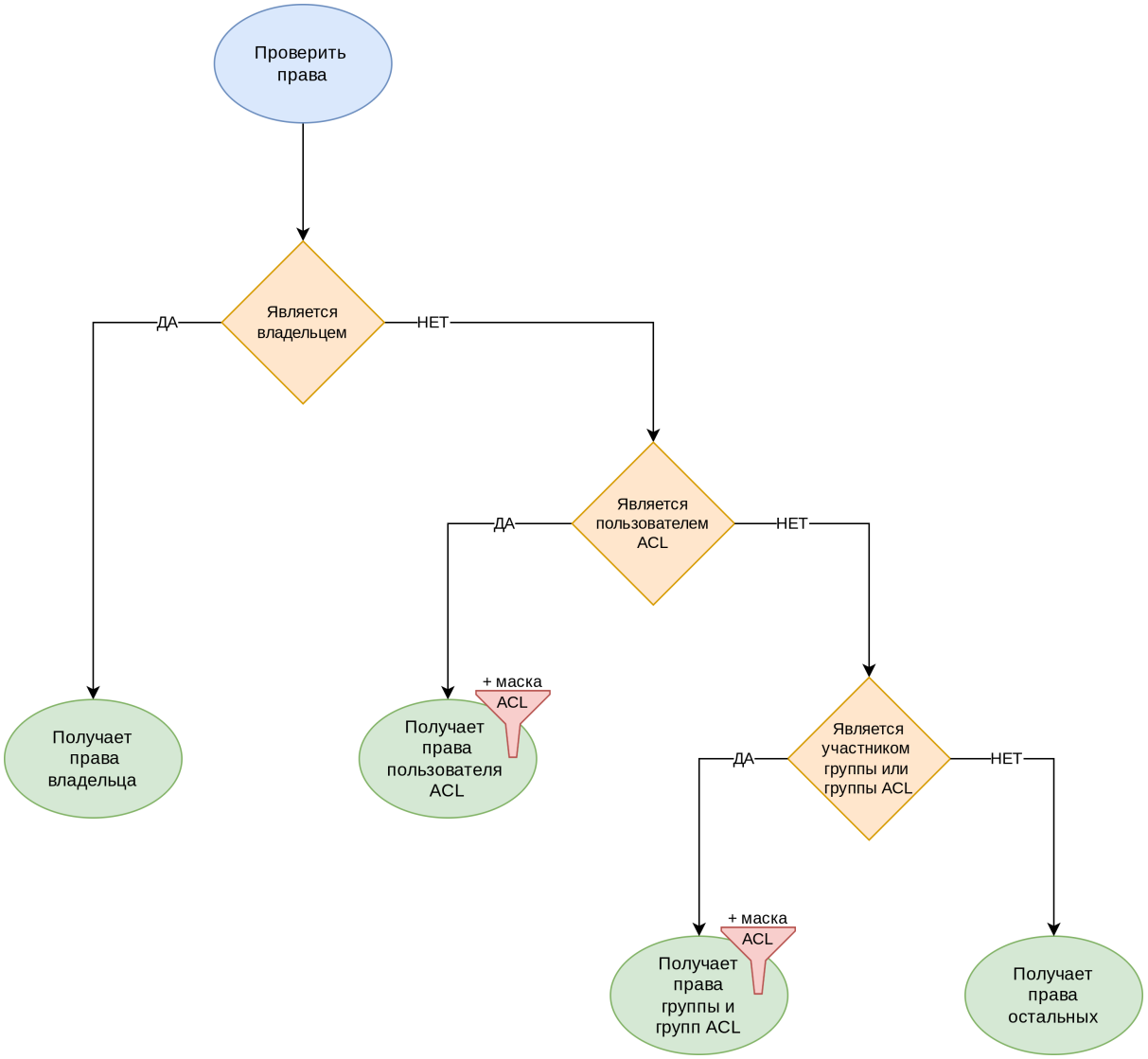


рис. 1.40 – Алгоритм определения прав доступа к объекту файловой системы Linux с учетом ACL

Для просмотра списков доступа предназначена утилита `getfacl` (Get File ACL), для редактирования этих списков предназначена утилита `setfacl` (Set File ACL). Продемонстрируем, как установка маски wx препятствует возможности прочитать файл.

```
localadmin@astra:~$ echo "hello" > test.txt
localadmin@astra:~$ sudo chown root:admins test.txt
localadmin@astra:~$ sudo chmod u=,g=r,o= test.txt
localadmin@astra:~$ sudo setfacl -m m::wx test.txt
localadmin@astra:~$ ls -l test.txt
-----wx---+ 1 root admins 6 фев 22 02:41 test.txt
          ^ символ «+» - означает расширенные права
```

Посмотрим расширенные права командой getfacl:

```
getfacl test.txt
```

Результат вывода:

```
# file: test.txt
# owner: root
# group: admins
user::---
group::r--                      #effective:---
mask::-wx
other::---
```

```
localadmin@astra:~$ cat test.txt
cat: test.txt: Отказано в доступе
localadmin@astra:~$ sudo setfacl -m m::rwx test.txt
localadmin@astra:~$ cat test.txt
hello
```

При вызове утилиты `setfacl -m m::wx test.txt` мы использовали следующие параметры:

- `-m` (modify) — указывает, что мы хотим изменить записи ACL. Может использоваться ключ `-x` для удаления записи из ACL.
- `m:` (mask) — указывает, что мы хотим изменить маску. Возможны еще значения u/g для пользователя/группы, но в этом случае после двоеточия нужно будет задать имя субъекта.
- `wx` — задает значение маски.
- `test.txt` — имя файла.

Списки доступа ACL хранятся в расширенных атрибутах файла, поэтому при обычном копировании эта информация теряется:

```
echo "hello" > test.txt
setfacl -m m::rwx test.txt
cp test.txt test2.txt
getfacl test2.txt
```

Результат выполнения:

```
# file: test2.txt
# owner: admin
# group: admins
user::rw-
group::r-x
other::r--
```

Для того чтобы скопировать файл с сохранением значений расширенных атрибутов, требуется установить ключ `--preserve`:

```
cp --preserve=all test.txt test3.txt
getfacl test3.txt
```

Результат вывода:

```
# file: test3.txt
# owner: admin
# group: admins
user::rw-
group::r--
mask::rwx
other::r--
```

Следует отметить, что, в отличие от подавляющего большинства операционных систем, Astra Linux предоставляет файловый менеджер, с помощью которого можно из графики редактировать права доступа, включая расширенные POSIX ACL, см. рис. 1.41.

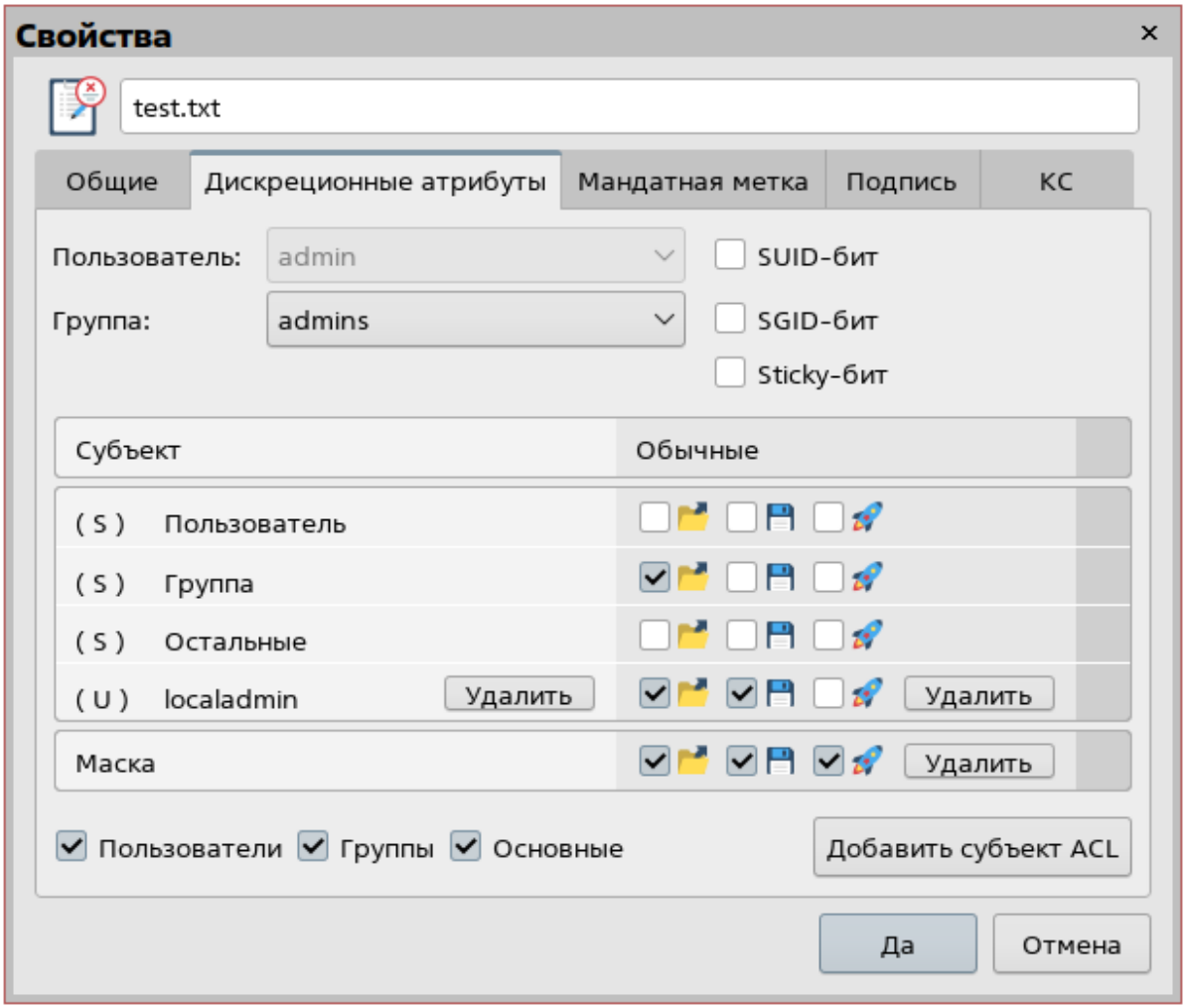


рис. 1.41 – Редактирование прав доступа к файлам из графического интерфейса

9.8 Сравнение возможностей POSIX и Windows ACL

Списки доступа POSIX ACL решают ту же задачу, что и NTFS ACL, но отличаются технической реализацией, поэтому для получения похожего результата вам потребуется вручную анализировать доступные параметры. Однако, если вам нужно максимально близко подойти к возможностям фай-

лового сервера Windows, то POSIX ACL точно не хватит, и можно рекомендовать воспользоваться набором приложений Samba, в котором реализована поддержка подключаемых модулей виртуальной файловой системы (Virtual File System, VFS).

Служба Samba действует на сервере от имени суперпользователя, но позволяет эмулировать несколько моделей безопасности. По умолчанию Samba выполняет проверки в стиле POSIX ACL, но при подключении модулей VFS может приближать свое поведение к NTFS. Следует обратить внимание на следующие VFS-модули:

- **Модуль acl_xattr** — использует одновременно как базовые права доступа к файлам (включая списки доступа ACL), так и расширенные атрибуты для поддержки списков доступа Windows. Но ввиду некоторой несовместимости моделей безопасности Windows/Linux при использовании этого модуля вы будете сталкиваться с рядом ограничений. Например, какие-то изменения настроек прав доступа, выполненные из Windows, будут настойчиво отказываться сохраняться.
- **Модуль nfs4acl_xattr** — использует только расширенные атрибуты, игнорируя базовые права доступа Linux и списки доступа POSIX ACL. Преимуществом модуля является практически полная совместимость со списками доступа Windows, но при подключении к файловому серверу напрямую через ssh/xrdp/vnc пользователи не смогут получить доступ к файлам, т.к. Linux игнорирует значения расширенных атрибутов, и владелец у всех объектов будет root.

9.9 Практика и тестирование

9.9.1 Практическая работа: Модуль 9. Работа с правами доступа, ACL

См. также *Требования, правила и цели выполнения практической работы*

9.9.1.1 Практические задания

Задание 1. Права rw на текстовый файл.

1. Создайте пользователя user1, назначьте для него пароль и выполните вход (команды `useradd`, `passwd`, `login`).
2. Создайте файл в домашней директории пользователя `~/file.txt` с контентом «hello». Владельцем файла должен быть user1:user1, права доступа по умолчанию 644 (u=rw,g=r,o=r).
3. Назначьте на файл права доступа 000 (u=,g=,o=) и проверьте, что вы

    a. не можете прочитать файл

b. не можете записать в файл новую строку «world»

4. Назначьте на файл права доступа 400 (u=r,g=,o=) и проверьте, что запись все еще недоступна, но чтение появилось.

5. Назначьте на файл права доступа 600 (u=rw,g=,o=) и проверьте, что вы можете теперь и читать, и писать в файл.

6. Назначьте на файл права доступа 006 (u=,g=,o=rw) и проверьте, что вы снова не можете ни читать, ни писать в файл.

Задание 2. Права x на исполняемом файле.

1. Скопируйте исполняемый файл cat в домашнюю директорию пользователя.

2. Выполните чтение из файла `file.txt` с помощью утилиты `cat` из домашней директории.

3. Назначьте на исполняемый файл права доступа 600 (u=rw,g=,o=) и проверьте, что вы не можете запускать эту утилиту для чтения файлов.

4. Назначьте на исполняемый файл права 100 (u=x,g=,o=) и проверьте, что права доступа на запуск утилиты снова появились.

Задание 3. Права rwx на каталог.

1. Создайте каталог `~/folder` и файл `~/folder/file.txt` с контентом «hello».

2. Назначьте на каталог права 000 (u=,g=,o=) и проверьте, что вы не можете более просматривать содержимое папки.

3. Назначьте на каталог права 400 (u=r,g=,o=) и проверьте, что теперь вы можете просмотреть список файлов, но без прав доступа к ним. Создание новых файлов все так же недоступно.

4. Назначьте на каталог права 600 (u=rw,g=,o=) и проверьте, что ничего не изменилось. Запись все так же недоступна.

5. Назначьте на каталог права 700 (u=rwx,g=,o=) и проверьте, что x дает возможность читать права на дочерние файлы, а в сочетании wx появилась возможность создания файлов.

Задание 4. ACL на пользователя.

1. Откройте еще одно окно терминала с правами root и создайте файл `~/file2.txt` с контентом «hello». Владельцем файла должен быть root:root, права доступа по умолчанию 644

(u=rw,g=r,o=r).

2. Назначьте на файл права доступа 007 (u=,g=,o=rwx).

3. Добавьте пользователя user1 в ACL файла с правами 4 (u:user1:r).

   Убедитесь, что пользователь потерял права на запись, которые ему ранее предоставляла категория others, и теперь у него права только на чтение через ACL.

4. Установите на файл через ACL маску с правами 0 (m::).

Убедитесь, что права на чтение, которые ему ранее предоставляла запись в ACL, перекрываются маской, поэтому он «выпадает» из ACL, и ему снова выдаются права в соответствии с правами категории others.

9.9.1.2 Ответы на практические задания (пошаговые инструкции)

Задание 1. Права rw на текстовый файл.

1. Создайте пользователя user1, назначьте для него пароль и выполните вход (команды `useradd`, `passwd`, `login`).

```
root@astra:~# useradd user1
root@astra:~# passwd user1
root@astra:~# login user1
```

2. Создайте файл в домашней директории пользователя `~/file.txt` с контентом «hello». Владельцем файла должен быть user1:user1, права доступа по умолчанию 644 (u=rw,g=r,o=r).

```
user1@astra:~$ echo "hello" > ~/file.txt
user1@astra:~$ ls -l ~/file.txt
-rw-r--r-- 1 user1 user1 0 фев 22 07:20 /home/user1/file.txt
```

3. Назначьте на файл права доступа 000 (u=,g=,o=) и проверьте, что вы

   a. не можете прочитать файл

   b. не можете записать в файл новую строку «world»

```
user1@astra:~$ chmod u=,g=,o= ~/file.txt
user1@astra:~$ ls -l ~/file.txt
---------- 1 user1 user1 6 фев 22 07:13 /home/user1/file.txt
user1@astra:~$ cat ~/file.txt
cat: /home/user1/file.txt: Отказано в доступе
user1@astra:~$ echo "world" >> ~/file.txt
-sh: /home/user1/file.txt: Отказано в доступе
```

4. Назначьте на файл права доступа 400 (u=r,g=,o=) и проверьте, что запись все еще недоступна,

но чтение появилось.

```
user1@astra:~$ chmod u=r,g=,o= ~/file.txt
user1@astra:~$ cat ~/file.txt
hello
user1@astra:~$ echo "world" >> ~/file.txt
-sh: /home/user1/file.txt: Отказано в доступе
```

5. Назначьте на файл права доступа 600 (u=rw,g=,o=) и проверьте, что вы можете теперь и читать, и писать в файл.

```
user1@astra:~$ chmod u=rw,g=,o= ~/file.txt
user1@astra:~$ echo "world" >> ~/file.txt
user1@astra:~$ cat ~/file.txt
hello
world
```

6. Назначьте на файл права доступа 006 (u=,g=,o=rw) и проверьте, что вы снова не можете ни читать, ни писать в файл.

```
user1@astra:~$ chmod u=,g=,o=rw ~/file.txt
user1@astra:~$ cat ~/file.txt
cat: /home/user1/file.txt: Отказано в доступе
user1@astra:~$ echo "world" >> ~/file.txt
-sh: /home/user1/file.txt: Отказано в доступе
```

Задание 2. Права x на исполняемом файле.

1. Скопируйте исполняемый файл cat в домашнюю директорию пользователя.

```
user1@astra:~$ which cat
/usr/bin/cat
user1@astra:~$ cp /usr/bin/cat ~/
user1@astra:~$ ls -l ~/cat
-rwxr-xr-x 1 user1 user1 40080 фев 22 07:50 /home/user1/cat
```

2. Выполните чтение из файла `file.txt` с помощью утилиты `cat` из домашней директории.

```
user1@astra:~$ ~/cat ~/file.txt
hello
world
```

3. Назначьте на исполняемый файл права доступа 600 (u=rw,g=,o=) и проверьте, что вы не можете запускать эту утилиту для чтения файлов.

```
user1@astra:~$ chmod u=rw,g=,o= ~/cat
user1@astra:~$ ~/cat ~/file.txt
-sh: /home/user1/cat: Отказано в доступе
```

4. Назначьте на исполняемый файл права 100 (u=x,g=,o=) и проверьте, что права доступа на запуск утилиты снова появились.

```
user1@astra:~$ chmod u=x,g=,o= ~/cat
user1@astra:~$ ~/cat ~/file.txt
hello
world
```

Задание 3. Права rwx на каталог.

1. Создайте каталог `~/folder` и файл `~/folder/file.txt` с контентом «hello».

```
user1@astra:~$ mkdir ~/folder
user1@astra:~$ echo "hello" > ~/folder/file.txt
user1@astra:~$ ls -l ~/folder/
итого 4
-rw-r--r-- 1 user1 user1 6 фев 22 07:58 file.txt
```

2. Назначьте на каталог права 000 (u=,g=,o=) и проверьте, что вы не можете более просматривать содержимое папки.

```
user1@astra:~$ chmod u=,g=,o= ~/folder/
user1@astra:~$ ls -l ~/folder/
ls: невозможно открыть каталог '/home/user1/folder/': Отказано в доступе
user1@astra:~$ echo "hello2" > ~/folder/file2.txt
-sh: /home/user1/folder/file2.txt: Отказано в доступе
user1@astra:~$ cd ~/folder/
-sh: cd: /home/user1/folder/: Отказано в доступе
```

3. Назначьте на каталог права 400 (u=r,g=,o=) и проверьте, что теперь вы можете просмотреть список файлов, но без прав доступа к ним. Создание новых файлов все так же недоступно.

```
user1@astra:~$ chmod u=r,g=,o= ~/folder/
user1@astra:~$ ls -l ~/folder/
ls: невозможно получить доступ к '/home/user1/folder/file.txt': Отказано в доступе
итого 0
-????????? ? ? ? ? ? file.txt
user1@astra:~$ echo "hello2" > ~/folder/file2.txt
-sh: /home/user1/folder/file2.txt: Отказано в доступе
```

4. Назначьте на каталог права 600 (u=rw,g=,o=) и проверьте, что ничего не изменилось. Запись все так же недоступна.

```
user1@astra:~$ chmod u=rw,g=,o= ~/folder/
user1@astra:~$ echo "hello2" > ~/folder/file2.txt
-sh: /home/user1/folder/file2.txt: Отказано в доступе
```

5. Назначьте на каталог права 700 (u=rwx,g=,o=) и проверьте, что x дает возможность читать права на дочерние файлы, а в сочетании wx появилась возможность создания файлов.

```
user1@astra:~$ chmod u=rwx,g=,o= ~/folder/
user1@astra:~$ ls -l ~/folder/
итого 4
```

(continues on next page)

(продолжение с предыдущей страницы)

```
-rw-r--r-- 1 user1 user1 6 фев 22 07:58 file.txt
```

Задание 4. ACL на пользователя.

1. Откройте еще одно окно терминала с правами root и создайте файл `~/file2.txt` с контентом «hello». Владельцем файла должен быть root:root, права доступа по умолчанию 644 (u=rw,g=r,o=r).

```
root@astra:~# cd /home/user1/
root@astra:/home/user1# echo "hello2" > file2.txt
# ls -l file2.txt
-rw-r--r-- 1 root root 0 фев 22 09:15 file2.txt
```

2. Назначьте на файл права доступа 007 (u=,g=,o=rwx).

```
root@astra:/home/user1# chmod u=,g=,o=rwx file2.txt
```

В этом случае пользователь user1 получит права rwx через категорию пользователей other.

```
user1@astra:~$ echo "world2" >> ~/file2.txt
user1@astra:~$ cat ~/file2.txt
hello2
world2
```

3. Добавьте пользователя user1 в ACL файла `file2.txt` с правами 4 (u:user1:r).

```
root@astra:/home/user1# setfacl -m u:user1:r file2.txt
root@astra:/home/user1# getfacl file2.txt
# file: file2.txt
# owner: root
# group: root
user::---
user:user1:r--
group::---
mask::r--
other::rwx
```

Убедитесь, что пользователь потерял права на запись, которые ему ранее предоставляла категория others, и теперь у него права только на чтение через ACL.

```
user1@astra:~$ echo "wide2" >> ~/file2.txt
-sh: /home/user1/file2.txt: Отказано в доступе
user1@astra:~$ cat ~/file2.txt
hello
world2
```

4. Установите на файл через ACL маску с правами 0 (m::).

```
setfacl -m m::- file2.txt
getfacl file2.txt
```

```
# file: file2.txt
# owner: root
# group: root
user::---
user:user1:r-- #effective:---
group::---
mask::---
other::rwx
```

Убедитесь, что права на чтение, которые ему ранее предоставляла запись в ACL, перекрываются маской, поэтому он «выпадает» из ACL, и ему снова выдаются права в соответствии с правами категории others.

```
user1@astra:~$ echo "wide2" >> ~/file2.txt
user1@astra:~$ cat ~/file2.txt
hello2
world2
wide2
```

9.9.2 Тест: Модуль 9. Работа с правами доступа, ACL

Пожалуйста, ответьте на несколько вопросов, чтобы закрепить полученные знания. Правильные ответы для самопроверки вы сможете найти в конце списка. Обратите внимание, что у некоторых вопросов может быть несколько верных вариантов ответов.

9.9.2.1 Вопросы

Вопрос 1. Какой командой можно изменить владельца файла/папки?

a. Командой `chgrp`

b. Командой `chmod`

c. Командой `chown`

d. Командой `chage`

e. Командой `chfn`

f. Командой `getfacl`

g. Командой `setfacl`

h. Командой `gettext`

i. Командой `setterm`

Вопрос 2. Какой командой можно изменить права доступа на файл/папку?

a. Командой `chgrp`

b. Командой `chmod`

c. Командой `chown`

d. Командой `chage`

e. Командой `chfn`

f. Командой `getfacl`

g. Командой `setfacl`

h. Командой `gettext`

i. Командой `setterm`

Вопрос 3. Что дает право read на файл?

a. Прочитать права доступа к файлу

b. Изменить права доступа к файлу

c. Прочитать содержимое файла

d. Записать содержимое файла

e. Дописать в конец файла новое содержимое

f. Выполнить файл

Вопрос 4. Что дает право write на файл?

a. Прочитать права доступа к файлу

b. Изменить права доступа к файлу

c. Прочитать содержимое файла

d. Записать содержимое файла

e. Дописать в конец файла новое содержимое

f. Выполнить файл

Вопрос 5. Что дает право execute на файл?

a. Прочитать права доступа к файлу

b. Изменить права доступа к файлу

c. Прочитать содержимое файла

d. Записать содержимое файла

e. Дописать в конец файла новое содержимое

f. Выполнить файл

Вопрос 6. Что дает право read на папку?

a. Прочитать список дочерних объектов папки

b. Прочитать права доступа на дочерние объекты папки

c. Создать дочерний объект в папке

d. Получить доступ к дочернему объекту папки

e. Ничего

Вопрос 7. Что дает право write на папку?

a. Прочитать список дочерних объектов папки

b. Прочитать права доступа на дочерние объекты папки

c. Создать дочерний объект в папке

d. Получить доступ к дочернему объекту папки

e. Ничего

Вопрос 8. Что дает право execute на папку?

a. Прочитать список дочерних объектов папки

b. Прочитать права доступа на дочерние объекты папки

c. Создать дочерний объект в папке

d. Получить доступ к дочернему объекту папки

e. Ничего

Вопрос 9. Что дает право write вместе с execute на папку?

a. Прочитать список дочерних объектов папки

b. Прочитать права доступа на дочерние объекты папки

c. Создать дочерний объект в папке

d. Получить доступ к дочернему объекту папки

e. Ничего

Вопрос 10. Если обычный пользователь является владельцем файла и на него установлены права 007, сможет ли пользователь прочитать файл командой `cat`?

a. Да

b. Нет

Ответы на вопросы

1. c
2. b, g
3. c
4. d, e
5. f
6. a
7. e
8. b, d
9. c или более полно b, c, d
10. b, если не суперпользователь

9.10 Заключение

В этом модуле мы познакомились с базовой моделью дискреционного разграничения прав доступа в Linux и ее расширением POSIX ACL. Эта тема, на самом деле, является крайне сложной, и чтобы

стать сильным администратором Linux, вам нужно будет заучить приведенные алгоритмы, внимательно выполнить предложенные практические задания и первое время максимально тщательно проверять сделанные вами настройки, чтобы выявлять ошибки и обязательно разбираться, почему система работает не так, как вы от нее ожидаете.

А далее нас ждет еще более интересная тема – будем еще глубже зарываться в механизмы работы PAM и SUDO!

9.11 Дополнительные источники информации

- Глава «Управление доступом» из книги «UNIX AND Linux. Руководство системного администратора» Эви Немет, Гарт Снайдер, Трент Хейн, Бэн Уэйли

9.12 Обратная связь

Если остались вопросы, то их всегда можно задать в специальной теме.

Модуль 10. Система аутентификации PAM и управление правами SUDO

10.1 Введение

Из этого модуля вы узнаете, почему нельзя «сидеть под рутом» и как именно работают утилиты `sudo` и `su`, позволяющие временно перейти в режим бога. Мы вскроем внутреннее устройство PAM-стека и даже пришьем к своему франкенштейну парочку новых PAM-модулей. В ходе экспериментов ни один сисадмин не пострадает.

10.2 Управление привилегиями в Linux

Установщик операционной системы Linux всегда создает учетную запись суперпользователя с именем root (от англ. корень), идентификатор которой равен нулю. Но это не просто привилегированный пользователь, как встроенный администратор Windows с RID 500, которого при большом желании можно и удалить. Пользователь root по своей сути является аналогом учетной записи SYSTEM. В Linux от имени root стартует подсистема инициализации `/sbin/init` и запускаются все остальные системные процессы, поэтому, если удалить эту учетную запись из локальной базы пользователей `/etc/passwd`, то система просто не взлетит.

Суперпользователь root получает свою суперсилу исключительно потому, что его идентификатор UID равен 0. Однако с версии ядра 2.2 стало возможным делегировать эти полномочия через так

называемые привилегии (capabilities). Поэтому теперь root может быть не единственным пользователем с суперсилой, но в пантеоне богов Linux имя суперпользователя навечно закрепилось именно за ним.

Например, для того чтобы пользователь localadmin мог произвольно менять UID/GID любым файлам и папкам, ему нужно делегировать привилегию CAP_CHOWN. Сделать это можно с помощью оснастки «Управление политикой безопасности». Просто выберите учетную запись localadmin в колонке слева и на вкладке «Привилегии» в основной части окна установите флажок напротив «CAP_CHOWN», см. рис. 1.42.

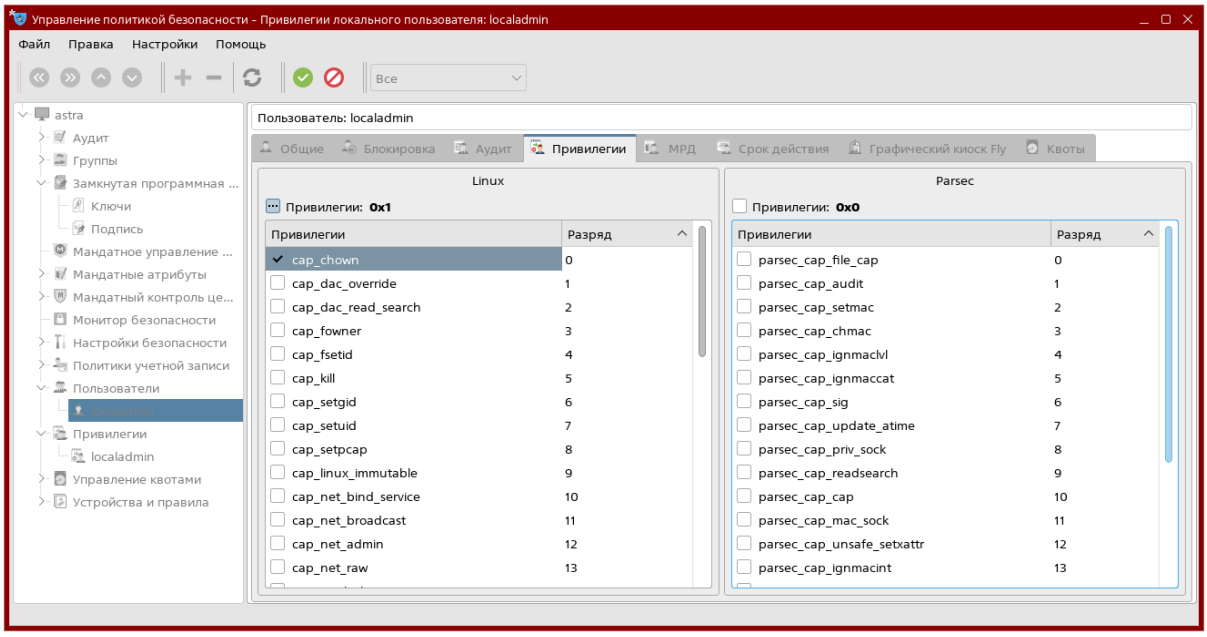


рис. 1.42 – Делегирование привилегий пользователю

В обычных линуксах привилегии пользователей задаются в файле `/etc/security/capability.conf`, а за их применение отвечает модуль pam_cap.so. В Astra Linux реализована собственная подсистема безопасности PARSEC, поэтому:

- кроме обычных Linux-привилегий есть еще привилегии Parsec,
- все настройки для пользователя localadmin с UID=1000 записываются в файл `/etc/parsec/capdb/1000`,
- а установку наследуемых привилегий на процесс, в котором запускается оболочка пользователя, выполняет модуль pam_parsec_cap.

Чтобы изменения вступили в силу, нужно перезайти в систему, но после этого для пользователя будет включен режим бога, по крайней мере, в части использования `chown`.

Привилегии стали одним из существенных улучшений системы безопасности Linux. А до их появ-

ления системным администраторам иногда приходилось создавать для root так называемые псевдонимы — это когда несколько пользователей используют один и тот же идентификатор 0 (см. ключ `--non-unique` утилиты `useradd`). Но если вы спросите любого ИБэшника, то он вам сразу и от души втолкует, что лучше так не делать никогда, никогда, никогда. Данная практика считается настолько порочной, что об этом ящике пандоры теперь предпочитают даже не вспоминать.

И, кстати, столь же порочными практиками считают стикеры с паролями на мониторе и вход в систему из-под рута. Но если со стикерами все очевидно, то во втором случае угроза состоит в том, что все пользовательские приложения будут иметь привилегии суперпользователя, и через малейшую уязвимость в Firefox, LibreOffice или CS:GO злоумышленники смогут захватить всю вашу систему целиком.

Из всего вышесказанного следует, что, когда вам самим потребуется выполнить какое-то действие с повышенными привилегиями или нужно будет предоставить такое право другим пользователям, следует использовать только правильные инструменты, такие как утилиты `su` и `sudo`.

10.2.1 Утилита su

В системе Linux запустить что-то из-под другого пользователя можно, например, с помощью утилиты `su` (*от англ. Substitute User - подменить пользователя*), которая похожа на «Run as administrator» или «Run as different user» в Windows.

По умолчанию утилита запускает оболочку из-под указанного пользователя, но с помощью ключей `-c` или `--command` можно определить строго ограниченный набор команд. Для указания имени целевого пользователя следует использовать длинный ключ `--login` (`su -login username`), короткий ключ `-l` (`su –l username`) или не указывать имя ключа вовсе, обозначив его символом дефиса (`su - username`).

Если имя пользователя не задать явно, то будет подразумеваться, что мы хотим запустить оболочку от имени root, поэтому утилита будет ожидать ввод пароля от учетной записи суперпользователя (команда `su -` эквивалентна команде `su - root`).

В использовании утилиты su есть еще один существенный нюанс. Если вы указываете ключ login или обозначаете его хотя бы дефисом, то оболочка запускается в окружении целевого пользователя так же, как будто он выполнил полноценный интерактивный вход в систему. Если же вы запустите утилиту `su` без ключа (например, `su` или `su localadmin`), то все переменные, за исключением PATH, USER и некоторых других, будут унаследованы от пользователя, который запустил утилиту `su`.

Продемонстрируем это на примере переменной myuser_var. Определяем в окружении пользователя myuser переменную myuser_var:

```
myuser@astra:~$ export myuser_var=test_value
myuser@astra:~$ echo $myuser_var
test_value
```

Проверяем, что при запуске su без ключа переменная доступна:

```
myuser@astra:~$ su localadmin -c 'echo myuser_var=$myuser_var'
Пароль:
myuser_var=test_value
```

Проверяем, что при запуске su с обозначением ключа переменная не доступна:

```
myuser@astra:~$ su - localadmin -c 'echo myuser_var=$myuser_var'
Пароль:
myuser_var=
```

Примечание: Переменные окружения играют очень важную роль, т.к. с их помощью может выполняться настройка системных утилит и клиентских приложений.

Например, если выполнить в терминале « `export KRB5_TRACE=/dev/stdout` », то это заставит утилиту `kinit` на доменных компьютерах отображать полную трассировку всех вызовов.

Утилита `su` всем хороша, за исключением следующих моментов:

- Для возможности использования утилиты `su` пользователю нужно предоставить пароль от привилегированной учетной записи, а в соответствии с лучшими практиками интерактивный вход из-под рута заблокирован, и пароля у этой учетной записи нет. Внезапно, правда! :)
- Учитывая, что для использования утилиты su пользователю предоставляется пароль от привилегированной учетной записи, его полномочия невозможно ограничить каким-либо перечнем команд. Он сможет выполнить в системе из-под этого пользователя все, что угодно.

Ввиду указанных причин более продвинутым инструментом следует считать утилиту `sudo`.

10.2.2 Утилита sudo

В результате развития идей гранулярного делегирования полномочий на выполнение команд появилась утилита `sudo` (от англ. Substitute User and DO - подменить пользователя и выполнить).

Утилита `sudo` позволяет системному администратору делегировать полномочия конкретным пользователям на запуск конкретных команд от имени других конкретных пользователей. И все это с журналированием выполняемых действий. Настройки для утилиты `sudo` задаются в конфигурационном файле `/etc/sudoers` и называются правилами SUDO. Дополнительные настройки могут

находиться так же в папке `/etc/sudoers.d/`.

10.2.2.1 Синтаксис команды SUDO

Команда `sudo <команда [опции]>` позволяет выполнить команду от имени суперпользователя, например, `sudo touch new_file`.

```
localadmin@astra:~$ sudo touch new_file
[sudo] пароль для localadmin:
localadmin@astra:~$ ls -l new_file
-rw-r--r-- 1 root root 0 сен 18 10:14 new_file
```

Если необходимо выполнить не одну команду, а несколько, то можно запустить интерактивный режим работы sudo, использовав опцию `-i` (длинный ключ `--login`) `sudo -i`. Для выхода из интерактивного режима необходимо ввести команду `exit`.

```
localadmin@astra:~$ sudo -i
root@astra:~# chmod +x /home/localadmin/new_file
root@astra:~# chown localadmin /home/localadmin/new_file
root@astra:~# exit
выход
localadmin@astra:~$ ls -l new_file
-rwxr-xr-x 1 localadmin root 0 сен 18 10:14 new_file
```

По умолчанию утилита `sudo` выполняет команду из-под суперпользователя root, но с помощью ключа `-u` вы можете указать и любого другого пользователя системы:

```
localadmin@astra:/etc$ sudo -u myuser id
uid=1001(myuser) gid=1002(myuser)
группы=1002(myuser),20(dialout),24(cdrom),25(floppy),29(audio),44(video),46(plugdev),
↪100(users)
```

Полное описание ключей команды вы найдете в справке sudo (8), которую можно вызвать командой `man 8 sudo`.

Примечание: Возможности утилиты sudo настолько прекрасны, что Microsoft реализовали нечто очень похожее https://github.com/microsoft/sudo. Будем посмотреть, как этот проект станет развиваться дальше.

10.2.2.2 Правила SUDO

Правила SUDO создают дополнительный слой авторизации, позволяя делегировать конкретным пользователям право выполнения отдельных команд с повышенными привилегиями. Локальные настройки утилиты `sudo` находятся в файле `/etc/sudoers`, который назван так потому, что пользо-

вателей, кому разрешено повышать привилегии с помощью утилиты `sudo`, называют **sudo e**nabled use**rs** или кратко sudoers.

В файле могут быть строки трех типов: параметры по умолчанию, псевдонимы (алиасы, именованные списки или проще переменные) и сами правила. Синтаксис правил представлен на рисунке ниже:

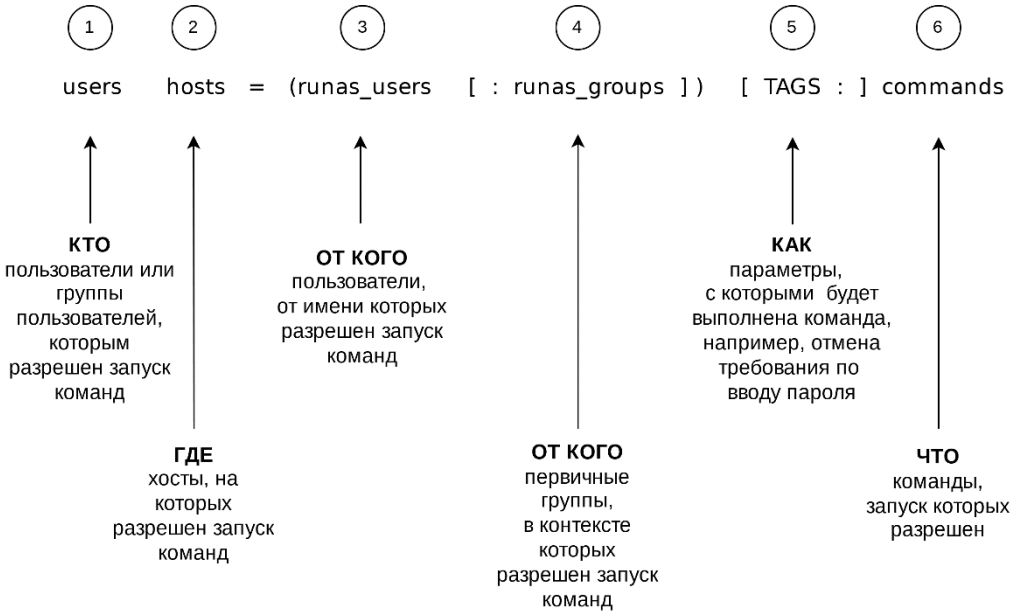


рис. 1.43 – Синтаксис правил SUDO

Правила могут быть как разрешающими, так и запрещающими, но по умолчанию считается, что прав на выполнение команд через sudo ни у кого нет. Для первых четырех компонентов правил следует определить область действия одним из двух способов:

- **Любой субъект (ALL)** — правило будет распространяться на все субъекты данного вида.
- **Указанные субъекты** — правило будет распространяться только на указанный перечень субъектов данного вида. Если требуется задать несколько значений, элементы списка должны быть разделены запятой.

  Для упрощения работы с большими списками синтаксис файла позволяет задавать именованные списки или так называемые алиасы. Для удобства настройки синтаксис именованных списков позволяет также исключать из них отдельные значения.

Рассмотрим внимательнее каждый компонент правила:

1. **Пользователи** и **группы пользователей**, которым разрешен запуск команд в рамках данного правила. Перед именем группы следует указывать символ процента.
2. **Хосты**, на которых разрешен запуск команд. Это может быть имя компьютера или его IP-

адрес. Параметр полезен, если один и тот же файл копируется на несколько хостов сразу.

3. **Пользователи, от имени которых разрешен запуск команд**. При выполнении команды через sudo по умолчанию предполагается, что команда запускается от имени root, но можно указать имя пользователя в явном виде с помощью ключа `-u`, и данный компонент правила позволяет ограничить перечень допустимых значений.

4. **Первичные группы, в контексте которых разрешен запуск команд**. По умолчанию используется первичная группа пользователя, от имени которого выполняется команда, но группу можно указать явно с помощью ключа `-g`. Данное значение проявляет себя, когда выполняются команды, которые создают новые файлы и папки, например, `touch` или `mkdir`. Этот параметр не является обязательным.

5. **Параметры, с которыми будет выполнена утилита sudo**, позволяют изменить поведение утилиты sudo. Например, с помощью NOPASSWD можно отключить запрос пароля. Этот параметр не является обязательным. Полный перечень доступных значений: EXEC, NOEXEC, FOLLOW, NOFOLLOW, LOG_INPUT, NOLOG_INPUT, LOG_OUTPUT, NOLOG_OUTPUT, MAIL, NOMAIL, PASSWD, NOPASSWD, SETENV и NOSETENV. О значении параметров можно посмотреть в справке man sudoers.

6. **Команды**, которые разрешено запускать в рамках этого правила. Это должен быть полный путь к исполняемому файлу. В конце строки можно указать допустимые параметры вызова.

   Напомним, что полный путь к исполняемому файлу системной утилиты можно узнать с помощью команды which, которую лучше запускать на целевых хостах, где предполагается использование этих правил SUDO.

В правилах SUDO можно использовать не только конкретные значения, но и шаблоны. В Astra Linux до версии 1.7.4 включительно используется проверенная версия sudo 1.8.x, в которой доступны только шаблоны в стиле shell, обработка которых выполняется через функции glob и fnmatch. С версии sudo 1.9.10 появится возможность использовать полноценные регулярные выражения.

В шаблонах можно использовать следующие символы подстановки (wildcards) или так называемые метасимволы:

- «?» – соответствует одному любому символу.
- «*» – соответствует любому количеству любых символов, в том числе и пустой строке. С использованием символа * следует быть крайне осторожным.
- «\» – позволяет экранировать спецсимволы, т.е. отключить их управляющую функцию. Используется, когда нужен знак вопроса, звездочки, двоеточия и др.

- «»»» (две подряд идущие двойные кавычки) – соответствует пустой строке. Если пустая строка указана в качестве единственного параметра команды, то эта команда может быть выполнена только без параметров.

- [. . .] – соответствует одному символу из указанного диапазона, например:

  - [abc] соответствует символу a, b или c;

  - [a-z] соответствует строчному символу латинского алфавита;

  - [[\:lower\:]] соответствует строчному символу латинского алфавита, но диапазон задан с помощью именованного класса символов (named character classes), полный перечень которых включает alnum, alpha, blank, cntrl, digit, graph, lower, print, punct, space, upper, xdigit.

- [!. . .] – восклицательный знак в начале диапазона позволяет инвертировать набор символов, т.е. шаблон соответствует любому символу, который не входит в указанный диапазон.

10.2.2.3 Механизм работы правил SUDO

Как уже было сказано, настройки утилиты `sudo` определяются содержимым файла `/etc/sudoers`. В этом файле находится также инструкция `#includedir /etc/sudoers.d`, которая включает содержимое дополнительных файлов из указанной директории. Инструкция начинается с символа решетки # как обычный комментарий, но комментарием не является, что может ввести в заблуждение. Сделано это так из соображений обратной совместимости, т.к. инструкции *include* и *includedir* были добавлены значительно позже, в 2004 и 2017 годах соответственно. С версии 1.9.1 появится возможность использовать символ @ собачки вместо решетки, что уменьшит путаницу.

В начале файла вы найдете предупреждение о том, что редактировать правила sudo напрямую не рекомендуется и лучше использовать утилиту `visudo`. Указанная утилита откроет файл в nano и обеспечит проверку синтаксиса перед сохранением изменений. Если вам ближе `vi` или `mcedit`. Вы можете заменить редактор по умолчанию:

```
localadmin@astra:~$ sudo update-alternatives --config editor
[sudo] пароль для localadmin:
Есть 4 варианта для альтернативы editor (предоставляет /usr/bin/editor).

  Выбор   Путь            Приор Состояние
------------------------------------------------------------
* 0            /bin/nano            40        автоматический режим
  1            /bin/nano            40        ручной режим
  2            /usr/bin/mcedit      25        ручной режим
  3            /usr/bin/vim.basic   30        ручной режим
  4            /usr/bin/vim.tiny    15        ручной режим

Press <enter> to keep the current choice[*], or type selection number: 0
```

Ниже представлено содержимое файла `/etc/sudoers` по умолчанию (без внесенных после установки системы изменений)

```
localadmin@astra:~$ sudo cat /etc/sudoers
```

```
#
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults        env_reset
Defaults        mail_badpass
Defaults        secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/
↪bin"
Defaults        !lecture

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "#include" directives:

#includedir /etc/sudoers.d
%astra-admin    ALL=(ALL:ALL) ALL
```

В файле по умолчанию присутствуют несколько параметров:

- **Параметр env_reset** – позволяет ограничить набор переменных из среды окружения пользователя, которые будут доступны запускаемой утилите. Это важно из соображений безопасности, поскольку эти переменные могут влиять на поведение утилит, запускаемых с привилегиями суперпользователя.

- **Параметр mail_badpass** – предписывает системе отправлять уведомления о неудачных попытках ввода пароля при выполнении команды sudo. Предполагается доставка в локальный почтовый ящик `/var/mail/root`.

- **Параметр secure_path** – позволяет задать список каталогов, в которых будет выполняться поиск запускаемых через sudo утилит, когда не указан полный путь к файлу. Это исключает запуск вредоносных приложений с повышенными привилегиями.

- **Параметр lecture** – позволяет определить, как часто будет выводиться предупреждение поль-

зователю о том, что следует с осторожностью использовать команду sudo, и возможность указать специальный файл, текст которого будет выводиться (`lecture_file`).

Рассмотрим правила, представленные в файле `/etc/sudoers` сразу после установки операционной системы:

- Правило «root ALL=(ALL:ALL) ALL» означает, что пользователь root может на любом хосте от имени любого пользователя и в контексте любой первичной группы выполнить любую команду.
- Правило «%sudo ALL=(ALL:ALL) ALL» означает то же самое для группы sudo.
- Правило «%astra-admin ALL=(ALL:ALL) ALL» означает то же самое для группы astra-admin.

Внимание: На компьютерах в домене пользователь admin автоматически вносится в список участников локальной группы astra-admin, за счет чего получает право на выполнение команд от имени суперпользователя через утилиту sudo.

10.2.2.4 Управление правилами SUDO

Для примера возьмем такую задачу: у нас есть два пользователя user1 и user2, которым необходимо делегировать право на выполнение команды ls от имени суперпользователя, например, для просмотра домашних каталогов других пользователей.

1. Запустим утилиту `sudo visudo` для редактирования файла `/etc/sudoers`.
2. Добавим в раздел «User aliases» псевдоним для пользователей user1 и user2 GROUPAUDITORS:

```
User_Alias GROUPAUDITORS = user1 user2
```

3. Добавим в раздел «Cmnd_aliases» псевдоним AUDIT для команды `ls -la /home/*`:

```
Cmnd_Alias AUDIT = /usr/bin/ls -la /home/*
```

4. И добавим в конец файла строку, разрешающую группе пользователей GROUPAUDITORS запускать команду AUDIT на локальной машине:

```
GROUPAUDITORS astra = AUDIT
```

Вот как должен выглядеть файл `/etc/sudoers`:

```
# User alias specification
User_Alias    GROUPAUDITORS=user1,user2

# Cmnd alias specification
Cmnd_Alias    AUDIT=/usr/bin/ls -la /home/*

# User privilege specification
root    ALL=(ALL:ALL) ALL

# Allow members of group sudo to execute any command
%sudo    ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "#include" directives:

#includedir /etc/sudoers.d
%astra-admin    ALL=(ALL:ALL) ALL
GROUPAUDITORS    astra=AUDIT
```

Воспользуемся командой su - user1 и проверим, работает ли наше правило для пользователя user1 sudo ls -la /home/user2.

Откроем оболочку из-под пользователя user1 и убедимся, что прав на чтение каталогов других пользователей у нас нет:

```
localadmin@astra:~$ su - user1
Пароль:
user1@astra:~$ ls /home/user2/
ls: невозможно открыть каталог '/home/user2/': Отказано в доступе
```

Выполним то же самое из-под sudo, но без ключей, доступа быть не должно:

```
user1@astra:~$ sudo ls /home/user2/
[sudo] пароль для user1:
Sorry, user user1 is not allowed to execute '/usr/bin/ls /home/user2/' as root on␣
↪astra.
```

Выполним команду с правильным набором ключей, доступ должен быть:

```
user1@astra:~$ sudo ls -la /home/user2/
[sudo] пароль для user1:
итого 28
drwx------ 3 user2 user2 4096 сен 18 11:22 .
drwxr-xr-x 8 root  root  4096 июл 26 10:38 ..
-rw------- 1 user2 user2    5 сен 18 11:22 .bash_history
-rw-r--r-- 1 user2 user2  220 сен 16  2020 .bash_logout
...
user1@astra:~$ exit
выход
```

Как видим, при обращении без sudo мы получили отказ, так как у пользователя user1 нет прав на домашний каталог пользователя user2.

При попытке выполнить команду без параметров мы получили отказ, т.к. такая команда не попа-

дает под заданные правила.

И только в третий раз, когда мы запустили `sudo ls -la /home/user2/`, система разрешила выполнить эту команду из-под суперпользователя и вывела нам результат её выполнения.

10.2.2.5 Отладка правил SUDO

Список правил пользователя

Результирующий набор правил SUDO для текущего пользователя можно вывести на экран с помощью ключа `-l` (`--list`):

```
localadmin@astra:~$ sudo -l
[sudo] пароль для localadmin:
Matching Defaults entries for localadmin on astra:
   env_reset, mail_badpass,
   secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin,
   !lecture,
   secure_path=/usr/lib/parsec/bin\:/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/
↪bin\:/sbin\:/bin
User localadmin may run the following commands on astra:
   (ALL : ALL) ALL
```

Результирующий набор правил SUDO для конкретного пользователя можно узнать вызовом на целевой машине команды sudo с ключами `-l` и `-U`:

```
localadmin@astra:~$ sudo -l -U user1
Matching Defaults entries for user1 on astra:
   env_reset, mail_badpass,
   secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin,
   !lecture,
   secure_path=/usr/lib/parsec/bin\:/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/
↪bin\:/sbin\:/bin
User user1 may run the following commands on astra:
   (root) /usr/bin/ls -la /home/*
```

Журналирование sudo

Sudo имеет три варианта журналирования:

- Кто что запускал – `/var/log/auth.log`
- Полные журналы сессий – sudoreplay
- Журнал отладки sudo – debug

Файл журнала /var/log/auth.log

В журнале `/var/log/auth.log` заносится информация вида:

```
date hostname progname: username : TTY=ttyname ; PWD=cwd ;
USER=runasuser ; GROUP=runasgroup ; TSID=logid ; ENV=env_vars
COMMAND=command
```

Например:

```
Sep 26 10:27:50 astra sudo[17629]: localadmin : TTY=pts/3 ;
PWD=/home/localadmin ; USER=root ; COMMAND=/usr/local/bin/tail /var/log/auth.log
```

Полные журналы сессий

Полные журналы сессий sudoreplay записывает и в последствии воспроизводит или перечисляет выходные журналы, созданные sudo. При воспроизведении sudoreplay может воспроизвести сеанс работы с sudo в режиме реального времени или с выбранной скоростью воспроизведения (быстрее или медленнее) в зависимости от параметров командной строки.

Для включения записи сессий sudo необходимо создать директорию */var/log/sudo-io* с правами 750 `sudo mkdir -m 750 /var/log/sudo-io` и отредактировать файл `/etc/sudoers`, добавив 3 строки:

```
Defaults log_output
Defaults!/usr/bin/sudoreplay !log_output
Defaults!/sbin/reboot !log_output
```

После этого начнется запись сеансов работы и командой `sudo sudoreplay -l` можно просматривать эти сеансы.

```
localadmin@astra:~$ sudo sudoreplay -l
Сен 26 10:50:17 2024 : localadmin : TTY=/dev/pts/3 ;
CWD=/home/localadmin ; USER=root ; TSID=000001 ;
COMMAND=/usr/local/bin/ls -l
Сен 26 10:50:51 2024 : localadmin : TTY=/dev/pts/3 ;
CWD=/home/localadmin ; USER=root ; TSID=000002 ;
COMMAND=/usr/local/bin/cat /etc/sudoers
```

Полный список опций доступен в `man sudoreplay`.

Журнал отладки sudo

Журнал отладки позволяет отследить как sudo обрабатывает ваши политики. Для включения журналирования в режиме debug требуется создать файл `/etc/sudo.conf`.

Синтаксис файла состоит из ключевого слова **Debug**, за которым следует имя **программы** или **плагина**, **подсистема для дебага** и **список флагов отладки**, разделенных запятыми.

```
Debug sudo /var/log/sudo_debug all@warn
```

Поддерживаются следующие флаги (приоритеты) отладки в порядке уменьшения важности: crit, err, warn, notice, diag, info, trace и debug. Каждый приоритет также включает все приоритеты выше него. Например, приоритет notice будет включать отладочные сообщения, зарегистрированные на уровне notice и выше.

Подсистемы для дебага:

- all - matches every subsystem
- auth - user authentication
- defaults - sudoers file Defaults settings
- env - environment handling
- logging - logging support
- match - matching of users, groups, hosts, and netgroups in the sudoers file
- netif - network interface handling
- parser - sudoers file parsing
- perms - permission setting
- etc. . .

Например, создадим файл `/etc/sudo.conf` со следующим содержимым:

```
Debug sudo /var/log/sudo_debug.log all@debug
Debug sudoers.so /var/log/sudo_debug.log all@debug
```

```
localadmin@astra:~$ cat /etc/sudo.conf
Debug sudo /var/log/sudo_debug.log all@debug
Debug sudoers.so /var/log/sudo_debug.log all@debug
```

В файле `/var/log/sudo_debug.log` будет представлена информация о пользователе и среде окружения в момент запуска команды sudo. На один вызов sudo будет создано более 5 тыс. строк журнала, включая информацию о настройках и информацию о пользователе:

```
sudo[24829] settings: progname=sudo
sudo[24829] settings: network_addrs=10.0.2.15/255.255.255.0
fe80::a00:27ff:fe19:4a3a/ffff:ffff:ffff:ffff::
sudo[24829] settings: plugin_dir=/usr/lib/sudo/
sudo[24829] settings: plugin_path=/usr/lib/sudo/sudoers.so
. . .
sudo[24829] user_info: user=localadmin
```

(continues on next page)

(продолжение с предыдущей страницы)

```
sudo[24829] user_info: pid=24829
sudo[24829] user_info: ppid=23717
sudo[24829] user_info: pgid=24829
sudo[24829] user_info: tcpgid=24829
sudo[24829] user_info: sid=23717
sudo[24829] user_info: uid=1000
sudo[24829] user_info: euid=0
sudo[24829] user_info: gid=1000
sudo[24829] user_info: egid=1000
sudo[24829] user_info: groups=24,25,29,30,44,46,109,113,114,333,1000,1001
sudo[24829] user_info: umask=022
sudo[24829] user_info: cwd=/home/localadmin
sudo[24829] user_info: tty=/dev/pts/4
sudo[24829] user_info: host=astra
sudo[24829] user_info: lines=35
sudo[24829] user_info: cols=88
```

После сбора журнала, лучше отключить режим отладки, закомментировав строки в файле /etc/sudo.conf.

```
localadmin@astra:~$ cat /etc/sudo.conf
#Debug sudo /var/log/sudo_debug.log all@debug
#Debug sudoers.so /var/log/sudo_debug.log all@debug
```

10.2.2.6 Лучшие практики

Ниже представлено несколько лучших практик при управлении правилами SUDO.

Будьте осторожны с символами подстановки

Символ «звездочки» в правилах SUDO следует использовать крайне осторожно, так как ошибки в его использовании могут привести к предоставлению несанкционированного доступа.

Допустим, администратору нужно было предоставить сотруднику доступ на чтение журналов messages, и он создал следующее правило:

```
localadmin@astra:~$ sudo grep user3 /etc/sudoers
user3      ALL=(ALL:ALL) NOPASSWD :        /usr/bin/cat /var/log/messages
```

На первый взгляд все правильно, и пользователь сможет получить доступ к журналам:

```
localadmin@astra:~$ su - user3
Пароль:
user3@astra:~$ cat /var/log/messages | head -n 2
cat: /var/log/messages: Отказано в доступе
user3@astra:~$ sudo cat /var/log/messages | head -n 2
Sep 17 00:00:36 astra syslog-ng[25203]: The current log file has a
mismatching size/inode information, restarting from the beginning;
```

(continues on next page)

(продолжение с предыдущей страницы)

```
state='affile_sd_curpos(/var/log/cups/access_log)',
stored_inode='407048', cur_file_inode='406780', stored_size='0',
cur_file_size='0', raw_stream_pos='134091'
Sep 17 00:00:36 astra syslog-ng[25203]: Configuration reload request
received, reloading configuration;
user3@astra:~$ sudo cat /var/log/messages.1 | head -n 2
Sep 10 00:00:53 astra syslog-ng[27895]: Error initiating reload,
reload is already ongoing;
Sep 10 00:00:53 astra syslog-ng[27895]: The current log file has a
mismatching size/inode information, restarting from the beginning;
state='affile_sd_curpos(/var/log/cups/access_log)',
stored_inode='406760', cur_file_inode='406666', stored_size='0',
cur_file_size='0', raw_stream_pos='67158'
```

Но утилита cat называется так от слова concatenate (сцеплять), и на самом деле она позволяет объединять в один поток содержимое сразу нескольких файлов, поэтому никто не помешает пользователю добавить к журналу `messages` содержимое файла `shadow`, чтобы увидеть пароли:

```
user3@astra:~$ sudo cat /var/log/messages /etc/shadow | tail -n 2
user2:$gost12512hash$vs18/gLTU.BNZfhz$Gby5XxXNbMz4Zhb6Rg7Uzf8tCykTWl0ynLCYsdJ.
↪9CY0nurmkY
zl0SzizoOr.Ex07pdSO402vCQHnt.wGKFnF.:19564:0:99999:7:::
user3:$gost12512hash$3zJkoCuxXSfG.CaT
↪$ZyRoq2N4Roagxeb9u83doUVEml0tkFwJUJk6qePyqGSW0ZpJLi
wdeDuD358j6lkTkfYChDdeF.9rKJEQevUcU1:19564:0:99999:7:::
```

Конкретно в этом случае для предотвращения нежелательного поведения утилиты sudo в шаблон следует добавить еще одну команду, которая будет запрещать вызов команды cat с пробелами в параметре:

```
user3      ALL=(ALL:ALL) NOPASSWD :        /usr/bin/cat /var/log/messages*, !/usr/bin/
↪cat /var/log/messages* *
```

Не разрешайте использование редактора vi

Работая в приложении vi, пользователь может не только редактировать текст, но и запускать команды оболочки, что дает значительные преимущества. Например, если в процессе редактирования файла конфигурации потребуется ввести точный путь к какому-то сертификату, пользователь сможет выполнить команду :shell, чтобы провалиться в оболочку и стандартными командами cd и ls уточнить необходимую информацию, а затем командой exit вернуться к редактированию файла.

Вместе с тем, такая реализация утилиты делает крайне опасным использование этого редактора вместе с правилами SUDO. Допустим, администратору нужно было предоставить сотруднику право на редактирование файла ldap.conf, и он создал следующее правило:

```
localadmin@astra:~$ sudo cat /etc/sudoers | grep user3
user3      ALL=(ALL:ALL) NOPASSWD :        /usr/bin/vi /etc/ldap/ldap.conf
```

На первый взгляд все правильно, и пользователь сможет получить право редактировать файл от имени суперпользователя. Но при этом ему ничто не помешает запустить оболочку из редактора и прочитать содержимое файла shadow.

```
localadmin@astra:~$ su - user3
Пароль:
user3@astra:~$ sudo vi /etc/ldap/ldap.conf
#
# LDAP Defaults
#
# See ldap.conf(5) for details
# This file should be world readable but not world writable.
. . .
:shell
```

После выполнения команды :shell совершается уязвимость, и мы получаем доступ суперпользователя:

```
root@astra:/home/user3# cat /etc/shadow | tail -n 2
user2:$gost12512hash$vs18/gLTU.BNZfhz$Gby5XxXNbMz4Zhb6Rg7Uzf8tCykTWl0ynLCYsdJ.
↪9CY0nurmkY
zl0SzizoOr.Ex07pdSO402vCQHnt.wGKFnF.:19564:0:99999:7:::
user3:$gost12512hash$3zJkoCuxXSfG.CaT
↪$ZyRoq2N4Roagxeb9u83doUVEml0tkFwJUJk6qePyqGSW0ZpJLi
wdeDuD358j6lkTkfYChDdeF.9rKJEQevUcU1:19564:0:99999:7:::
```

Обязательно используйте sudoedit

Команда `sudoedit` – символическая ссылка на `sudo -e`. При вызове создает временную копию файла, которую может редактировать пользователь, вызывающий sudo. При этом сам редактор, в котором будет происходить работа с копией файла, запускается под обычным (непривилегированным) пользователем. При сохранении файла только в случае успешной проверки синтаксиса sudoedit заменит оригинальный файл его измененной версией.

Примечание: По умолчанию не позволяет редактировать файл, если родительская директория доступна для записи пользователем, вызывающим команду. Это предотвращает race condition, которая могла бы позволить пользователю перезаписать произвольный файл. Bug 707: разрешение пользователям редактировать файлы в директориях, к которым они имеют право на запись с помощью sudoedit, может позволить им обойти ограничения, установленные в конфигурации sudoedit (и фактически редактировать любой файл в системе).

Ограничивайте путь для исполняемых файлов

С помощью директивы secure_path определите пути, в которых будет выполняться поиск запускаемых через sudo утилит, когда не указан полный путь к файлу.

```
Defaults secure_path="/usr/sbin:/usr/bin:/sbin:/bin:/usr/local/bin:/usr/local/sbin"
```

Используйте группы пользователей, оставляйте комментарии

Довольно простой рекомендацией является отказ от назначения прав на конкретных пользователей – используйте вместо этого группы. В этом случае и список правил будет короче, и за списками участников групп обычно удается лучше следить.

Давайте только минимально необходимые полномочия

При настройке правил SUDO следует предоставлять доступ только к тем командам, которые необходимы сотрудникам для выполнения должностных обязанностей. Чтобы избежать наличие излишних привилегий у пользователей крайне важно регулярно проводить аудит и отзывать те разрешения, которые более не требуются.

10.2.2.7 Система аутентификации PAM

Программы, предоставляющие пользователям доступ к ресурсам, должны уметь выполнять их аутентификацию. Когда вы входите в систему, вы предоставляете ваши учетные данные (имя пользователя и пароль). Процесс регистрации использует эти данные для проверки вашей аутентичности, действительно ли вы тот, за кого себя выдаете. В системе Linux эти проверки реализованы с использованием стека PAM-модулей.

Система аутентификации PAM (англ. Pluggable Authentication Modules, «подключаемые модули аутентификации») – это набор разделяемых библиотек, которые позволяют интегрировать различные низкоуровневые методы аутентификации в виде единого высокоуровневого программного интерфейса. PAM реализует модульный подход к системе аутентификации, что позволяет системному администратору задавать правила аутентификации без необходимости перекомпиляции прикладных приложений.

PAM позволяет абстрагировать аутентификацию пользователей через различные приложения. Системному администратору не надо знать, как работает система аутентификации в конкретном приложении, достаточно знать, как работает PAM, и иметь соответствующую библиотеку, т.е. PAM-модуль.

В свою очередь приложениям и их разработчикам нет необходимости учитывать низкоуровневую

логику, чтобы работать с бэкендами, например, базами данных, службой LDAP, файлами паролей, веб-службами с поддержкой WS-Security или другими бэкендами, некоторые из которых еще даже не существуют и будут созданы в будущем. Любой желающий (при наличии знаний и ресурсов) может создать свой собственный PAM-модуль, например, для работы приложений с новым сервисом аутентификации.

При установке программ, требующих аутентификации, они автоматически вносят все изменения, необходимые для их нормальной работы. Однако для понимания того, как происходит аутентификация пользователя в Linux, необходимо разбираться в архитектуре и работе PAM. А если понадобится изменить конфигурацию, необходимо понимать структуру конфигурационного файла PAM.

Состав и логическая схема PAM

На рисунке ниже представлена логическая схема взаимодействия компонентов PAM.

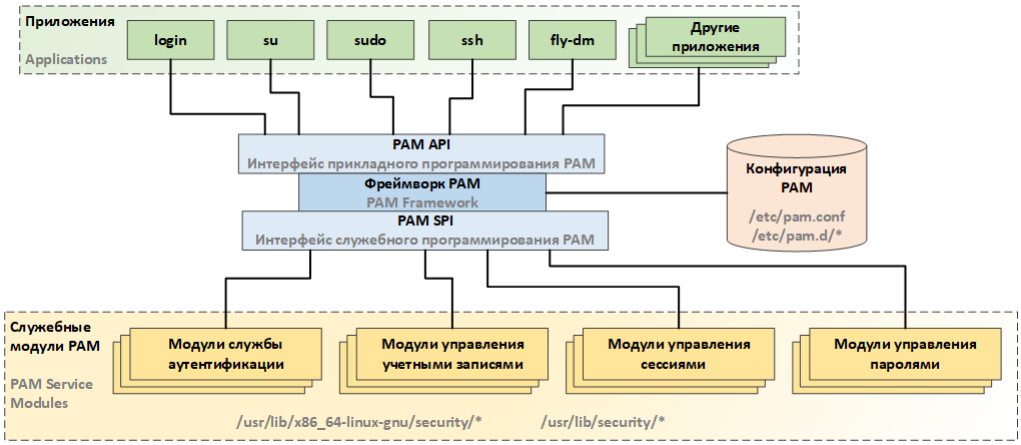


рис. 1.44 – Логическая схема взаимодействия компонентов PAM

В качестве клиентов PAM могут выступать различные приложения, которым требуется выполнять аутентификацию. Это и программа login, которая запускается в самом конце процесса загрузки системы в текстовом режиме, это и рабочий стол fly-dm, который тоже проводит проверку аутентичности пользователей, это и такие приложения, как ssh, sudo, su и другие.

В состав PAM входят:

- Фреймворк PAM, который включает API (интерфейс прикладного программирования, служащий для взаимодействия с ним прикладных программ, которые используют эту систему) и SPI (интерфейс служебного программирования, служащий для взаимодействия фреймворка PAM с набором модулей).

- Конфигурация PAM – набор конфигурационных файлов, содержащих набор правил для про-

верки аутентификации пользователя (/etc/pam.conf и файлы из каталога /etc/pam.d/).

- Модули PAM – набор библиотек, определяющих логику проверки.

Модули PAM

Модули PAM – это подключаемые библиотеки. Они по умолчанию хранятся в каталогах /usr/lib/x86_64-linux-gnu/security/ и /lib/security/.

```
localadmin@astra:~$ ls /usr/lib/x86_64-linux-gnu/security/
pam_access.so     pam_gnome_keyring.so  pam_mail.so        pam_selinux.so     pam_tty_
↪audit.so
pam_cracklib.so   pam_group.so          pam_mkhomedir.so   pam_sepermit.so    pam_
↪umask.so
pam_debug.so      pam_issue.so          pam_motd.so        pam_shells.so      pam_
↪unix.so
pam_deny.so       pam_keyinit.so        pam_namespace.so   pam_stress.so      pam_
↪userdb.so
pam_echo.so       pam_kiosk2.so         pam_nologin.so     pam_succeed_if.so  pam_
↪warn.so
pam_env.so        pam_lastlog.so        pam_permit.so      pam_systemd.so     pam_
↪wheel.so
pam_exec.so       pam_limits.so         pam_pwhistory.so   pam_tally2.so      pam_
↪xauth.so
pam_faildelay.so  pam_listfile.so       pam_rhosts.so      pam_tally.so
pam_filter.so     pam_localuser.so      pam_rootok.so      pam_time.so
pam_ftp.so        pam_loginuid.so       pam_securetty.so   pam_timestamp.so

localadmin@astra:~$ ls /lib/security/
pam_parsec_aud.so  pam_parsec_cap.so  pam_parsec_mac.so   pam_parsec_xauth.so
```

Модули бывают 4 типов:

- **Auth (аутентификация)** — модули этого блока проверяют подлинность пользователя. Они могут включать такие методы, как проверка пароля или использование двухфакторной аутентификации.

- **Account (управление правами доступа)** — этот блок выполняет проверки после успешной аутентификации, например, проверку срока действия учетной записи или групповых принадлежностей.

- **Password (управление паролем)** — модули этого блока обрабатывают такие операции с паролями, как их изменение или проверка сложности.

- **Session (управление сеансом)** — здесь определяются действия, выполняемые при открытии или закрытии сеанса. Например, установка переменных окружения или монтирование файловых систем.

Модули могут использоваться как индивидуально, так и в виде целых цепочек модулей. Порядок

расположения модулей играет важную роль в процессе аутентификации, поэтому это дает возможность администратору легко задавать условия, которые должны быть пройдены пользователем в процессе аутентификации, прежде чем он получит необходимые полномочия.

Модули бывают стандартные и пользовательские:

- Стандартные модули, такие как pam_unix для локальной аутентификации. Эти официально поддерживаемые и широко используемые модули – проверенное и стабильно работающее решение.
- Сторонние разработчики могут создавать свои собственные модули, расширяя возможности аутентификации, что позволяет адаптировать систему PAM под конкретные потребности организаций.

Ниже перечислены несколько стандартных модулей с кратким описанием:

- **pam_unix** – обеспечивает аутентификацию через локальные учетные записи. Это основной модуль для проверки паролей в файле /etc/passwd.
- **pam_tally2** – обеспечивает ограничение доступа после определенного числа неудачных попыток входа. Повышает безопасность, предотвращая подбор паролей методом перебора.
- **pam_limits** – предназначен для управления системными ресурсами для пользователей. Позволяет установить ограничения на использование ресурсов, таких как количество открытых файлов или объем памяти.
- **pam_exec** – обеспечивает выполнение внешних команд или сценариев в процессе аутентификации. Предоставляет возможность настраивать дополнительные действия при входе/выходе пользователя.
- **pam_access** – предназначен для управления доступом на основе IP-адресов. Позволяет определить, с каких IP-адресов разрешен или запрещен доступ.

Для работы компьютера в домене используется PAM-модуль от службы SSSD.

Конфигурационные файлы PAM

В PAM используется целый набор конфигурационных файлов в каталоге /etc/pam.d/. Файл /etc/pam.conf может содержать инструкции на тот случай, если PAM не обнаружит каталог /etc/pam.d. Однако в Astra Linux после установки никаких инструкций в этом файле нет, поэтому вы можете использовать только файлы из каталога /etc/pam.d/.

```
localadmin@astra:~$ ls /etc/pam.d/
chfn             common-session                fly-dm-np  polkit-1    su-l
```

(continues on next page)

(продолжение с предыдущей страницы)

```
chpasswd          common-session-noninteractive  fly-wm     runuser     sumac.xauth
chsh              cron                           login      runuser-l   systemd-user
common-account    cups                           newusers   samba
common-auth       fly-dm                         other      su
common-password   fly-dm-greeter                  passwd     sudo
```

Имена этих файлов соответствуют именам идентификаторов, с которыми приложения запускают функцию pam_start. Обычно эти идентификаторы совпадают с именем приложения, например, у приложений login, fly-dm и др. Но отдельные приложения могут использовать несколько идентификаторов, например, утилита sudo использует sudo и sudo-i, а pam-модуль для сервера Apache может использовать уникальный идентификатор для каждого раздела веб-сайта.

Синтаксис конфигурационных файлов PAM

Посмотрим, например, содержимое файла `/etc/pam.d/sudo`:

```
localadmin@astra:~$ cat /etc/pam.d/sudo
#%PAM-1.0
@include common-auth
@include common-account
@include common-session-noninteractive
session required pam_limits.so
```

Как видите, с помощью директивы @include в него включены три других файла конфигурации: common-auth, common-account и common-session-noninteractive. И уже после них идет инструкция по использованию модуля pam_limits.so.

Посмотрим содержимое файла `cat /etc/pam.d/common-auth`:

```
#
# /etc/pam.d/common-auth - authentication settings common to all services
#
# This file is included from other service-specific PAM config files,
# and should contain a list of the authentication modules that define
# the central authentication scheme for use on the system
# (e.g., /etc/shadow, LDAP, Kerberos, etc.).  The default is to use the
# traditional Unix authentication mechanisms.
#
# As of pam 1.0.1-6, this file is managed by pam-auth-update by default.
# To take advantage of this, it is recommended that you configure any
# local modules either before or after the default block, and use
# pam-auth-update to manage selection of other modules.  See
# pam-auth-update(8) for details.
# here are the per-package modules (the "Primary" block)
auth    [success=ignore default=2]      pam_localuser.so
auth    [success=1 default=ignore]      pam_succeed_if.so quiet user ingroup astra-
↪admin
auth    [success=ignore default=die]    pam_tally.so per_user deny=8
auth    [success=1 default=ignore]      pam_unix.so nullok_secure try_first_pass
```

(continues on next page)

(продолжение с предыдущей страницы)

```
# here's the fallback if no module succeeds
auth    requisite                       pam_deny.so
# prime the stack with a positive return value if there isn't one already;
# this avoids us returning an error just because nothing sets a success code
# since the modules above will each just jump around
auth    required                        pam_permit.so
# and here are more per-package modules (the "Additional" block)
# end of pam-auth-update config
```

Каждая значащая строка должна содержать инструкцию для вызова модуля, которая может включать 3 обязательных элемента и один дополнительный:

- **Тип модуля** (обязательный) – существуют 4 типа модулей. О них было рассказано в предыдущем разделе.
- **Флаги управления** (обязательный) – определяет, как результат работы конкретного модуля (результат проверки, осуществляемой в нем) повлияет на процесс аутентификации в целом.
- **Путь к модулю** (обязательный) – это абсолютное или относительное имя пути к файлу PAM.
- **Параметры модуля** (дополнительный) – это список маркеров, которые могут быть использованы для влияния на функциональность модуля.

Флаги управления

Как мы видели на примерах в файле `/etc/pam.d/common-auth`, значение флагов может представлять собой одно слово, например, «required», или конструкцию из выражений в квадратных скобках, например, «[success=ignore default=2]».

В первом случае используется одно из 4 слов:

- **required** (обязательный) – указанный модуль PAM должен обязательно вернуть код успешной проверки, чтобы проверка этого сервиса считалось успешной. Но последующие модули будут вызываться дальше по цепочке, даже если результат проверки этого модуля был отрицательный и уже понятно, что проверка сервиса считается проваленной.
- **requisite** (необходимый) – указанный модуль PAM должен обязательно вернуть код успешной проверки, чтобы проверка этого сервиса считалась пройденной. В отличие от required, если результат проверки этого модуля был отрицательный и уже понятно, что проверка сервиса считается проваленной, последующие модули не будут вызываться, а сам сервис будет недоступен.
- **sufficient** (достаточный) – если указанный модуль PAM вернёт код успешной проверки, весь сервис будет разрешен, а остальные модули из файла вызываться не будут. Однако, если мо-

дуль PAM вернёт код проваленной проверки, оставшиеся модули пройдут проверку, а неудача данного модуля не будет приниматься во внимание.

- **optional** (дополнительный) – результат проверки указанного модуля PAM будет иметь значение, только если это единственный модуль для этого сервиса.

- **include** – флаг «include» позволяет включить другой файл конфигурации PAM в текущий файл. Это способ организации конфигурации для повторного использования.

Во втором случае используется конструкция из одного или нескольких выражений, заключенных в квадратные скобки и разделенные пробелами:

```
[значение1=действие1 значение2=действие2 ...]
```

Модули PAM могут возвращать около 30-ти значений, которые зависят от типа модуля (Auth, Account, Session или Password). Вот список возможных значений: success, open_err, symbol_err, service_err, system_err, buf_err, perm_denied, auth_err, cred_insufficient, authinfo_unavail, user_unknown, maxtries, new_authtok_reqd, acct_expired, session_err, cred_unavail, cred_expired, cred_err, no_module_data, conv_err, authtok_err, authtok_recover_err, authtok_lock_busy, authtok_disable_aging, try_again, ignore, abort, authtok_expired, module_unknown, bad_item, conv_again, incomplete и default. Значение default означает все значения, которые явно не заданы.

В качестве действия может быть указан один из следующих вариантов:

- **ignore** – игнорировать значение, результат не влияет на общий код возврата от этого модуля.

- **bad** – определяет значение как сбой модуля, но продолжит проверки других модулей.

- **die** – определяет значение как сбой модуля и провал проверки всего сервиса в целом.

- **ok** – определяет значение как успешную проверку модуля, но не отменяет сбой модуля.

- **done** – определяет значение как успешную проверку модуля и всего сервиса в целом.

- **Цифра** – определяет, сколько строк в конфигурации необходимо пропустить после текущей.

При этом надо учесть, что модуль возвращает только одно значение (из 30-ти возможных) и действие, назначенное именно этому значению, и будет выполнено.

Таким образом, вышеупомянутые 4 варианта значения флагов можно представить и в виде расширенной формы записи:

- required: [success=ok new_authtok_reqd=ok ignore=ignore default=bad]

- requisite: [success=ok new_authtok_reqd=ok ignore=ignore default=die]

- sufficient: [success=done new_authtok_reqd=done default=ignore]
- optional: [success=ok new_authtok_reqd=ok default=ignore]

Разберем подробнее расширенную запись, например, для required:

- success=ok – модуль вернул значение успешно. Проверка этого модуля будет считаться успешно пройденной. А проверка сервиса будет считаться успешно пройденной (на этом этапе), если ни один из предыдущих модулей не вызвал сбой.
- new_authtok_reqd=ok – модуль сигнализировал, что пароль пользователя желательно обновить. Проверка этого модуля будет считаться успешно пройденной. А проверка сервиса будет считаться успешно пройденной (на этом этапе), если ни один из предыдущих модулей не вызвал сбой.
- Ignore=ignore – модуль сообщает, что его проверку необходимо игнорировать. Результат проверки не будет учитываться при оценке сервиса в целом.
- default=bad – модуль предписывает считать все остальные значения как сбой. При их появлении проверка этого модуля (как и сервиса в целом) не будет пройдена, однако проверки других модулей продолжатся.

Параметры модуля

Рассмотрим параметры вызова модуля на примере конфигурационного файла `cat /etc/pam.d/common-password`:

```
#
# /etc/pam.d/common-password - password-related modules common to all services
#
# This file is included from other service-specific PAM config files,
# and should contain a list of modules that define the services to be
# used to change user passwords.  The default is pam_unix.
# Explanation of pam_unix options:
#
# The "sha512" option enables salted SHA512 passwords.  Without this option,
# the default is Unix crypt.  Prior releases used the option "md5".
#
# The "obscure" option replaces the old \`OBSCURE_CHECKS_ENAB' option in
# login.defs.
#
# See the pam_unix manpage for other options.
# As of pam 1.0.1-6, this file is managed by pam-auth-update by default.
# To take advantage of this, it is recommended that you configure any
# local modules either before or after the default block, and use
# pam-auth-update to manage selection of other modules.  See
# pam-auth-update(8) for details.
# here are the per-package modules (the "Primary" block)
password        requisite                       pam_cracklib.so retry=3 minlen=8␣
```

(continues on next page)

(продолжение с предыдущей страницы)

```
↪difok=3
password        [success=1 default=ignore]      pam_unix.so obscure use_authtok try_
↪first_pass gost12_512
# here's the fallback if no module succeeds
password        requisite                       pam_deny.so
# prime the stack with a positive return value if there isn't one already;
# this avoids us returning an error just because nothing sets a success code
# since the modules above will each just jump around
password        required                        pam_permit.so
# and here are more per-package modules (the "Additional" block)
password        optional        pam_gnome_keyring.so
# end of pam-auth-update config
```

Как вы можете видеть, при вызове модулей pam_cracklib.so и pam_unix.so используются параметры вызова модулей:

- pam_cracklib.so retry=3 minlen=8 difok=3
- pam_unix.so obscure use_authtok try_first_pass

Откроем приложение «Политика безопасности» (fly-admin-smc), *Пуск → Панель управления → Безопасность → Политика безопасности* и перейдем в раздел *Политики учетной записи → Политика паролей*.

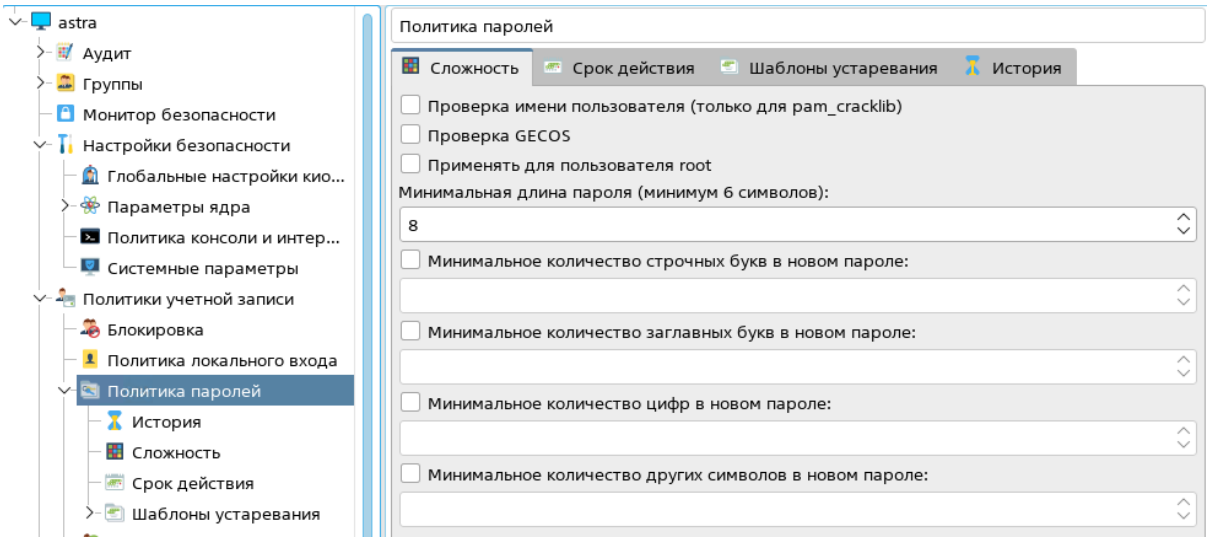
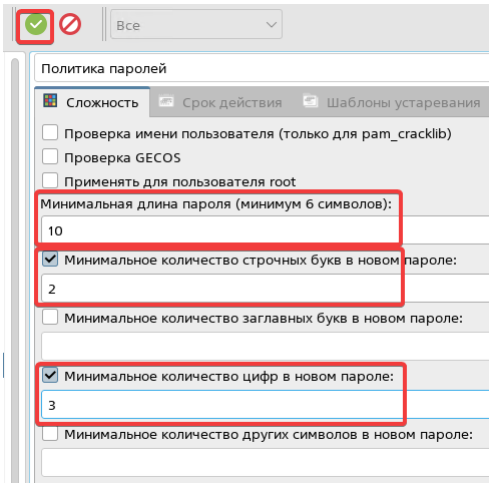


рис. 1.45 – Политика безопасности. Политика паролей.

Изменим значения:

- Минимальная длина пароля: 10 символов.
- Минимальное количество строчных букв: 2.
- Минимальное количество цифр: 3.



Вернемся в терминал и проверим содержимое конфигурационного файла `/etc/pam.d/common-password` еще раз:

```
cat /etc/pam.d/common-password | grep pam_
```

```
# used to change user passwords.  The default is pam_unix.
# Explanation of pam_unix options:
# See the pam_unix manpage for other options.
password        requisite                       pam_cracklib.so   retry=3 difok=3␣
↪minlen=10 lcredit=-2 ucredit=0 dcredit=-3 ocredit=0
password        [success=1 default=ignore]      pam_unix.so obscure use_authtok try_
↪first_pass gost12_512
password        requisite                       pam_deny.so
password        required                        pam_permit.so
password        optional        pam_gnome_keyring.so
```

Как видим, значения параметров вызова модуля pam_cracklib изменились в соответствии со сделанными нами настройками:

- minlen=10 – минимальная длина пароля 10 символов.
- it=-2 – минимальное количество строчных букв 2.
- dcredit=-3 – минимальное количество цифр 3.

Таким образом, меняя конфигурацию PAM непосредственно через редактирование конфигурационных файлов или через программы графического интерфейса, мы можем управлять параметрами и процессом аутентификации пользователей в различных приложениях.

А теперь попробуйте установить службу sssd командой sudo apt install sssd, и вы убедитесь, что во всех общих файлах common-account, common-auth и др. появились строчки, подключающие модуль pam_sss.so от службы SSSD. Именно благодаря этому модулю и обеспечивается работа машины в домене.

10.3 Практика и тестирование

10.3.1 Практическая работа: Модуль 10. Система аутентификации PAM и управление правами SUDO

См. также *Требования, правила и цели выполнения практической работы*

10.3.1.1 Практические задания

Задание 1. Изменение пароля суперпользователя

1. Используя команду sudo, измените пароль суперпользователя (root). Напоминаем, что это плохая практика и задание носит учебный характер.
2. Убедитесь, что новый пароль работает при входе в систему.

Задание 2. Настройка sudoers.

1. Создайте пользователей john, user1, user2.
2. Задайте пароли для пользователей user1, user2.
3. Создайте файл правил 01_usermanagement (с использованием `visudo`).
4. Добавьте пользователю john права использования `sudo` без ввода пароля для команд `groupadd` и `usermod`.

Задание 3. Ограничение доступа к команде `su`

1. Переключитесь в пользователя john.
2. Создайте группу g_admin и добавьте в нее пользователей user1 и user2.
3. Отредактируйте файл `/etc/pam.d/su` и добавьте ограничение, чтобы только пользователи из группы «g_admin» могли использовать команду `su`.
4. Проверьте, что user1 и user2 могут использовать `su`, а пользователь john – не может.

Задание 4. Использование модуля pam_tally для блокировки учетной записи

1. Настройте модуль pam_tally для отслеживания неудачных попыток входа в систему.
2. Установите правило блокировки учетной записи на 5 минут после 3 неудачных попыток входа.

3. Проверьте функциональность. Для этого введите неправильные учетные данные и убедитесь, что учетная запись блокируется.

Замечания:

- Перед внесением изменений в конфигурационные файлы, создайте резервные копии для предотвращения потенциальных проблем.
- Во время выполнения задач следите за журналами системы (`/var/log/auth.log` или `/var/log/secure`), чтобы отслеживать любые ошибки аутентификации или доступа.

10.3.1.2 Ответы на практические задания (пошаговые инструкции)

Задание 1. Изменение пароля суперпользователя

1. Используя команду `sudo`, измените пароль суперпользователя (root). Напоминаем, что это плохая практика и задание носит учебный характер.

```
localadmin@astra:~$ sudo passwd root
Введите новый пароль для суперпользователя (root)
```

2. Убедитесь, что новый пароль работает при входе в систему.

Задание 2. Настройка sudoers

1. Создайте пользователей john, user1, user2.

```
localadmin@astra:~$ sudo useradd -s /bin/bash -m john
localadmin@astra:~$ sudo useradd -s /bin/bash -m user1
localadmin@astra:~$ sudo useradd -s /bin/bash -m user2
```

2. Задайте пароли для пользователей user1, user2.

```
localadmin@astra:~$ sudo passwd user1
localadmin@astra:~$ sudo passwd user2
```

3. Создайте файл правил `01_usermanagement` (с использованием `visudo`).

```
localadmin@astra:~$ sudo visudo -f /etc/sudoers.d/01_usermanagement
```

4. Добавьте пользователю john права использования sudo без ввода пароля для команд `groupadd` и `usermod`. Добавьте следующую строку в файл `sudoers`:

```
john ALL = NOPASSWD: /usr/sbin/groupadd,/usr/sbin/usermod
```

Задание 3. Ограничение доступа к команде su

1. Переключитесь в пользователя john.

```
localadmin@astra:~$ sudo su - john
```

2. Создайте группу g_admin и добавьте в нее пользователей user1 и user2.

Создайте группу g_admin:

```
localadmin@astra:~$ sudo groupadd g_admin
```

Добавьте пользователей в группу g_admin:

```
localadmin@astra:~$ sudo usermod -aG g_admin user1
localadmin@astra:~$ sudo usermod -aG g_admin user2
```

3. Отредактируйте файл `/etc/pam.d/su` и добавьте ограничение, чтобы только пользователи из группы «g_admin» могли использовать команду `su`.

Выйдите из пользователя john командой `logout`.

Отредактируйте файл `/etc/pam.d/su`, добавив в него следующую строку:

```
auth required pam_wheel.so group*g_admin
```

4. Проверьте, что user1 и user2 могут использовать `su`, а пользователь john – не может.

Переключитесь на пользователя user1.

```
localadmin@astra:~$ sudo su - user1
```

Убедитесь, что переключение в пользователя user2 работает.

```
localadmin@astra:~$ su user2
localadmin@astra:~$ exit
```

Переключитесь на пользователя john.

```
localadmin@astra:~$ sudo su - john
```

Убедитесь, что переключение в пользователя user2 НЕ работает.

```
localadmin@astra:~$ su user2
localadmin@astra:~$ su: Доступ запрещен
```

Задание 4. Использование модуля pam_tally для блокировки учетной записи

1. Настройте модуль pam_tally для отслеживания неудачных попыток входа в систему.

2. Установите правило блокировки учетной записи на 5 минут после 3 неудачных попыток входа.

Отредактируйте файл `/etc/pam.d/common-auth`, изменив параметры для модуля pam_tally:

```
auth [success*ignore default*die] pam_tally.so per_user deny*3
unlock_time*300
```

3. Проверьте функциональность. Для этого введите неправильные учетные данные и убедитесь, что учетная запись блокируется.

Переключитесь на пользователя user1 (ему разрешено вызывать `su`).

```
localadmin@astra:~$ sudo su - user1
```

Убедитесь, что блокировка работает – введите пароль неправильно 3 раза.

```
localadmin@astra:~$ su user2
Пароль:
su: Сбой при проверке подлинности
localadmin@astra:~$ su user2
Пароль:
su: Сбой при проверке подлинности
localadmin@astra:~$ su user2
Учетная запись заблокирована как следствие неудачных попыток входа (всего -- 4).
su: Сбой при проверке подлинности
```

10.3.2 Тест: Модуль 10. Система аутентификации PAM и управление правами SUDO

Пожалуйста, ответьте на несколько вопросов, чтобы закрепить полученные знания. Правильные ответы для самопроверки вы сможете найти в конце списка. Обратите внимание, что у некоторых вопросов может быть несколько верных вариантов ответов.

10.3.2.1 Вопросы

Вопрос 1. Какая из следующих команд позволяет вам временно получить права суперпользователя без переключения в сессию root?

a. Команда `su`

b. Команда `sudo`

c. Команда `elevate`

d. Команда `runas`

Вопрос 2. Какой файл конфигурации из нижеперечисленных используется для настройки правил утилиты `sudo`?

a. Файл `/etc/sudorules`

b. Файл `/etc/sudo.conf`

c. Файл `/etc/sudoers`

d. Файл `/etc/sudo/rules.conf`

Вопрос 3. Какой пользователь обычно указывается в команде `su` для переключения на суперпользователя (выберите все варианты)?

a. Пользователь `-`

b. Пользователь `superuser`

c. Пользователь `admin`

d. Пользователь `root`

Вопрос 4. Какой из следующих ключей команды `sudo` позволяет выполнить команду с правами другого пользователя?

a. Ключ `-s`

b. Ключ `-u`

c. Ключ `-g`

d. Ключ `-p`

Вопрос 5. Что делает команда `sudo -l`?

a. Переключает пользователя на root

b. Отображает список доступных команд для текущего пользователя

c. Устанавливает новый пароль суперпользователя

d. Завершает сеанс суперпользователя

Вопрос 6. Какая строка должна быть в файле `/etc/sudoers`, чтобы пользователь `alice` могла выполнять команды через утилиту `sudo` без ввода пароля?

a. Строка `alice ALL*(ALL) NOPASSWD: ALL`

b. Строка `alice ALL*ALL sudo`

c. Строка `alice NOPASSWD: ALL*(ALL)`

d. Строка `NOPASSWD: ALL alice*(ALL)`

Вопрос 7. Какая команда позволяет изменить текстовый редактор по умолчанию, используемый при редактировании файла sudoers через `visudo`?

a. Команда `sudo seteditor`

b. Команда `sudoedit -e`

c. Команда `sudo update-alternatives --config editor`

d. Команда `visudo --set-editor`

Вопрос 8. Какой модуль PAM обеспечивает базовую аутентификацию по паролю?

a. Модуль `pam_secure`

b. Модуль `pam_unix`

c. Модуль `pam_authenticate`

d. Модуль `pam_basic_auth`

Вопрос 9. Для чего используется модуль `pam_limits`?

a. Для установки ограничений на ресурсы, доступные пользователю

b. Для управления правами доступа к сети

c. Для обработки входящих сетевых запросов

d. Для создания резервных копий конфигурационных файлов

Вопрос 10. Какая основная функция PAM в Linux?

a. Обеспечение совместимости аутентификации с Windows

b. Создание резервных копий файлов аутентификации

c. Обеспечение модульности и переносимости аутентификации

d. Защита от вирусов и вредоносных программ

Ответы на вопросы

1. b
2. c

3. a, d
4. b
5. b
6. a
7. c
8. b
9. a
10. c

10.4 Заключение

В этом модуле мы познакомились с механизмами повышения привилегий su/sudo, разобрались, как работают правила SUDO, и углубились в настройки PAM-стека.

Далее у нас будут основы автоматизации в Linux с применением сценариев Bash, чтобы почувствовать себя немного DevOps'ом.

10.5 Дополнительные источники информации

- Официальный сайт sudo

10.6 Обратная связь

Если остались вопросы, то их всегда можно задать в специальной теме.

Модуль 11. Работа со сценариями (скриптами) оболочки Bash

11.1 Введение

С помощью материалов этого модуля вы сможете сделать быстрый переход с PowerShell и CMD на инструменты автоматизации для Linux. Мы подобрали для вас наглядные примеры и удачные срав-

нения, сделали акценты на том, что показалось необычным нам самим и что подсказали коллеги. В общем, спешите воспользоваться опытом тех, кто уже в баше!

Материалы модуля составлены из предположения, что вы уже немного умеете программировать. Но если мечта стать кодером до сих пор только маячит у вас на горизонте, который, как известно, всегда удаляется по мере приближения, то рекомендуем обратиться к списку дополнительной литературы и начать действовать незамедлительно.

11.2 Сравнение Bash и CMD/PowerShell

Для понимания принципиальной разницы между Bash, CMD и PowerShell следует немного погрузиться в исторический контекст.

Оболочка Bash появилась в 1987 году в рамках проекта GNU на замену Bourne shell (отсюда и акроним Bourne again shell — перерожденная оболочка). В ней были реализованы все те лучшие идеи, которые были придуманы в мире *nix, начиная с 70-х годов, и она продолжает улучшаться до сих пор.

Командная оболочка CMD берет свое начало с первых релизов MS-DOS в начале 80-х годов, но по функциональности так и осталась жалким подобием того, что было в Unix. А после выхода Windows 2000 труп CMD разве что обвели цветными мелками, чтобы создать ощущение праздника.

Но в Microsoft уже с конца 90-х годов начали понимать, что их графический интерфейс не подходит для системных администраторов, которым требуется автоматизировать управление ИТ-инфраструктурой, поэтому они снова и снова пытались предложить пользователям что-то новое.

Одной из таких попыток стал Сервер сценариев Windows (Windows Script Host, WHS), который позволил управлять системой из скриптов на языках JScript и VBScript через объектную модель компонентов (Component Object Model, COM). Однако WHS не решил всех проблем, поэтому позднее был дан зеленый свет проекту Monad под руководством Джеффри Сновера, который в дальнейшем стал известен как PowerShell.

Примечание: Термин «монада» был взят из книги «Монадология» немецкого философа Лейбница и означает фундаментальную единицу для выполнения задачи. Но согласитесь, сколь бы фундаментальной эта монада ни была, а слово командлет звучит намного приятнее.

Разработчики PowerShell хотели создать инструмент управления операционной системой Windows, который бы по функциональности не уступал тому, что есть в Unix, но для этого нужно было преодолеть фундаментальное противоречие между концепциями «все есть файл» и «все есть объект». Нужны были конвейеры для объединения команд в цепочки, но требовалось обеспечить передачу

объектов между командлетами, чтобы не приходилось каждый раз парсить текст для получения доступа к структурированным данным. Данная задача и определила объектно-ориентированный характер языка PowerShell.

Из всего вышесказанного можно сделать несколько очень простых выводов:

- Обе оболочки Microsoft: и CMD, и PowerShell — были вдохновлены оболочкой Bash и ее предшественницами.
- Оболочка Bash намного функциональнее CMD, но она принципиально не поддерживает возможности PowerShell по работе с объектами. Если вам нужны объекты, то добро пожаловать в Python, у которого среди прочего есть и продвинутый интерфейс командной строки.

Ну, а возможности bash в части программирования вы можете оценить по играм из коллекции shellgames, скачать которую можно командой:

```
$ git clone https://github.com/bolknote/shellgames
```

Арканоид в псевдографике просто шикарен, см. рис. 1.46.

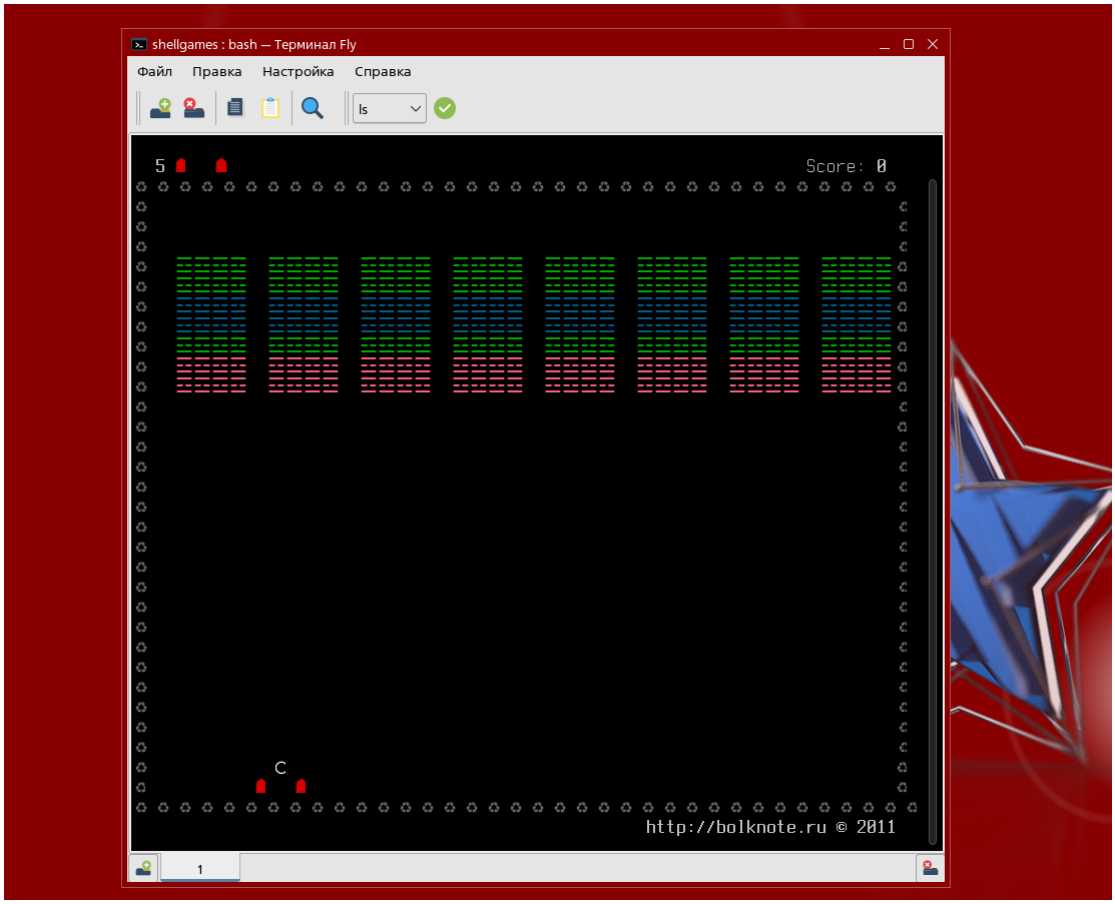


рис. 1.46 – Пример игры «Арканоид» на Bash

11.3 Запуск скриптов

Мы уже знаем, что при открытии терминала у пользователя запускается оболочка, которая указана у него оболочкой по умолчанию в базе `/etc/passwd`. Мы уже умеем запускать команды в оболочке и знаем, как выжать максимум из командной строки с помощью конвейеров, но для решения комплексных задач автоматизации этого все равно недостаточно. Чтобы не копировать в окно терминала сотни строк из заранее подготовленных файлов, давайте разберемся как правильно запускать такие файлы скриптов в пакетном режиме.

Скрипты представляют собой текстовые файлы с набором команд, которые отличаются от обычных программ принципиально тем, что для их использования нужен интерпретатор. В роли интерпретатора bash-скриптов выступает исполняемый файл оболочки `/bin/bash`, которому в качестве параметра можно передать имя файла, и тогда новый экземпляр приложения выполнит команды из указанного файла, после чего завершит свою работу.

Примечание: При необходимости bash-скрипты, конечно, можно «скомпилировать» и распространять в виде исполняемых файлов, например, с помощью того же shc (shell script compiler), но в этом случае скрипты все равно останутся скриптами и после распаковки будут просто передаваться интерпретатору через опцию `-c` (*от англ. command*).

В том случае, когда мы передаем файл скрипта интерпретатору в качестве параметра, нам нужны только права на чтение этого файла, см. *Модуль 9. Работа с правами доступа, ACL*. Давайте вместо традиционного приложения «Hello World» создадим скрипт `select_string.sh`, который будет имитировать работу одноименного командлета PowerShell, и проверим его работу:

```
echo ""
echo -n "Укажите путь для поиска: "
read find_path

echo -n "Укажите шаблон для отбора файлов: "
read find_name

echo -n "Укажите строку поиска:"
read find_pattern

echo ""
echo "Результаты поиска:"
find $find_path -type f -name "$find_name" -exec grep -iH "$find_pattern" {} \\; 2>/
↪dev/null
```

Попробуйте запустить файл через обращение к исполняемому файлу bash:

```
localadmin@astra:~$ bash select_string.sh
```

(continues on next page)

(продолжение с предыдущей страницы)

```
Укажите путь для поиска: /etc
Укажите шаблон для отбора файлов: *
Укажите строку поиска:pam_sss

Результаты поиска:

/etc/pam.d/common-password:password   [success=1 default=ignore]      pam_sss.so  ␣
↪use_authtok
/etc/pam.d/common-auth:auth    [success=1 default=ignore]     pam_sss.so use_first_
↪pass
/etc/pam.d/common-account:account      [default=bad success=ok user_unknown=ignore] ␣
↪pam_sss.so
/etc/pam.d/common-session:session      optional                        pam_sss.so
```

Если же мы хотим запускать скрипт напрямую по его имени, например, `./select_string.sh`, то нам потребуется установить права на выполнение, и нужно будет подсказать операционной системе, какой именно интерпретатор следует использовать в этом случае.

Как вы помните, Windows определяет тип файла по расширению, а Linux опирается на его сигнатуру. Для правильной идентификации скриптов используют сигнатуру, состоящую из символов «#!» в начале файла, после которых в строке указывают полный путь до требуемого интерпретатора:

```
#! /bin/bash
echo ""
echo -n "Укажите путь для поиска: "
read find_path
...
```

Вы можете убедиться, что после внесения указанных изменений в файл `select_string.sh` утилита `file` без проблем распознает этот файл как Bourne-Again shell script:

```
localadmin@astra:~$ file select_string.sh
select_string.sh: Bourne-Again shell script, UTF-8 Unicode text executable
```

Учитывая, что символом решетки в большинстве языков обозначают комментарий, с помощью сигнатуры «#! ...» можно специфицировать скрипты Python, Perl, PHP, Ruby и других языков — интерпретаторы просто проигнорируют эту строку. Однако на этом возможности сигнатуры не ограничиваются, и вы можете с ее помощью не только указать путь к файлу, но и задать параметры вызова, например, включить режим подробного вывода:

```
#! /bin/bash --verbose
echo ""
echo -n "Укажите путь для поиска: "
read find_path
...
```

Чтобы упростить коммуникации, символы «#!» получили даже собственное имя «Шебанг» (И тут выскакивает Рики Мартин: she bangs, she bangs, oh baby, when she moves, she moves...). На самом

деле этот термин является просто сокращением от полного словосочетания «hash bang», где слово bang означает восклицание, а не взрыв, как в ТБВ.

И осталось только изменить права доступа. Давайте проверим, что будет с флагом выполнения и без него:

```
localadmin@astra:~$ ./select_string.sh
bash: ./select_string.sh: Отказано в доступе

localadmin@astra:~$ chmod u+x select_string.sh
localadmin@astra:~$ ./select_string.sh

Укажите путь для поиска: /etc
...
```

11.4 Коды возврата

В качестве результатов скрипты и утилиты могут возвращать совершенно любую текстовую информацию, содержание которой может зависеть в том числе от того, какая локаль установлена у пользователя по умолчанию (см. переменную $LANG). Чтобы в заданиях автоматизации можно было однозначно идентифицировать успешное/неуспешное выполнение команды, приложения используют так называемый код возврата последней операции, который можно получить с помощью служебной переменной «$?». Приведем пример с удалением файла:

```
localadmin@astra:~$ touch testfile.txt
localadmin@astra:~$ rm testfile.txt
localadmin@astra:~$ echo "$?"
0
```

Код возврата будет равен 0, что соответствует успешному выполнению операции.

```
localadmin@astra:~$ rm testfile.txt
rm: невозможно удалить 'testfile.txt': Нет такого файла или каталога
localadmin@astra:~$ echo "$?"
1
```

При попытке удаления несуществующего файла код возврата будет равен 1.

Из скриптов вы можете задать этот код с помощью команды `exit <N>`, где <N> – это код возврата, который представляет собой целое число в диапазоне 0 - 255. Код 0 обычно назначают успешному выполнению команды. Приведем пример скрипта `high_five.sh`, который всегда даст пятюню:

```
#!/bin/bash
exit 5      # Дай пять!
```

Скрипт выдает код возврата 5, который можно потом использовать в обработке ошибок:

```
localadmin@astra:~$ ./high_five.sh
localadmin@astra:~$ echo $?
5
```

Скрипты, которые разрабатываются под Windows, редко используют в заданиях автоматизации, но в CMD и PowerShell, конечно же, есть аналогичные инструменты. В PowerShell вызов команды делается точно также, только в качестве служебной переменной используется `$LastExitCode`, а в CMD команду нужно вызывать с параметром «`exit /b 5`», а переменная называется %errorlevel%.

11.5 Локальные переменные

Переменные являются основой любого языка программирования. С их помощью выполняются математические операции, обработка строк и любые другие манипуляции с данными. Физически переменные представляют собой именованные участки памяти, в которых хранится определенная информация, и обычно приводят такие аналогии из жизни:

- **склад** - оперативная память компьютера;
- **стеллаж** – ячейка памяти, выделяемая под хранение значения переменной;
- **запись в ведомости** - имя переменной, по которому можно быстро определить адрес ячейки памяти, т.е. код стеллажа, где на складе находится нужный нам товар.

11.5.1 Присвоение значений переменной

Для объявления переменных на языке Bash им следует присвоить значение с помощью оператора «=». Слева от символа «равно» указывается имя переменной, справа — ее значение, а рядом не должно быть никаких лишних пробелов. Значения, состоящие из нескольких слов, должны обрамляться кавычками. Чтобы подставить значение переменной в строку, в начале имени переменной следует указать символ доллара. Пример присвоения значения переменной на Bash:

```
localadmin@astra:~$ a="Здравствуйте"
localadmin@astra:~$ b="уважаемый пользователь"
localadmin@astra:~$ echo "$a, $b"
Здравствуйте, уважаемый пользователь
```

Используя переменные bash, нужно учитывать следующие особенности:

- Использовать можно только латинские буквы, цифры и символ подчеркивания. При этом имя переменной не должно начинаться с цифр. Цифровые имена зарезервированы для позиционных параметров.
- Не рекомендуется использовать имена, совпадающие с названием команд и утилит. Например,

не стоит использовать имя file, т.к. есть одноименная утилита.

- Имена переменных Bash чувствительны к регистру, поэтому переменные **sum_invoice** и **Sum_Invoice** – это две совершенно разные переменные.

В языке CMD, как вы знаете, для присвоения переменным значений используется команда `set`. Так же, как в Bash, рядом с символом «=» не должно быть лишних пробелов, но вот значения, состоящие из нескольких слов, не обрамляются никакими кавычками. Все, что справа от знака «=», будет интерпретировано как присваиваемое значение. Для подстановки значения переменной ее имя следует обрамлять с двух сторон символом процента:

Пример на CMD:

```
C:\> set a=Здравствуйте
C:\> set b=уважаемый пользователь
C:\> echo %a%, %b%
Здравствуйте, уважаемый пользователь
```

Еще одним существенным отличием CMD является нечувствительность к регистру, поэтому, в отличие от Bash, к переменной **sum_invoice** мы можем обратиться по имени **Sum_Invoice**, **SUM_invoce** и как угодно еще.

В языке PowerShell имена переменных всегда начинаются с символа доллара вне зависимости от контекста использования, а наличие пробелов до и после оператора присвоения «=» не имеют никакого значения. И, опять же, полная нечувствительность к регистру в именах переменных.

Пример на PowerShell:

```
PS> $a = "Здравствуйте"
PS> $b = "уважаемый пользователь"
PS> echo "$a, $b"
Здравствуйте, уважаемый пользователь
```

Казалось бы, отличия совсем незначительные, но сначала они будут сбивать вас с толку.

11.5.2 Строки и строковые переменные

Язык Bash поддерживает несколько разных инструментов для работы со строками, которые обладают схожей функциональностью, но имеют разный синтаксис, что сначала может довольно сильно путать.

Во-первых, есть операции подстановки параметров, с помощью которых можно получить длину строки и вырезать часть подстроки, для начала операции над строкой определим ее:

```
localadmin@astra:~$ msg="Привет, уважаемый пользователь"
localadmin@astra:~$
```

Вычислим длину строки в переменной msg с помощью конструкции `${#имя_переменной}`:

```
localadmin@astra:~$ echo "${#msg}"
30
```

Выведем часть подстроки, начиная с 8го символа с помощью конструкции `${имя_переменной:стартовая_позиция}`:

```
localadmin@astra:~$ echo "${msg:8}"
уважаемый пользователь
```

Выведем первые 6 символов с помощью конструкции `${имя_переменной:стартовая_позиция:количество_символо`

```
localadmin@astra:~$ echo "${msg:0:6}"
Привет
```

Кстати, стартовую позицию можно не указывать, если мы хотим получить фрагмент строки, начиная с первого символа: `echo ${msg::6}`.

Во-вторых, есть универсальный обработчик выражений `expr`, который содержит гораздо больше полезных инструментов.

```
localadmin@astra:~$ echo `expr length "$msg"`
30
```

Обратите внимание на вызов подстановки через одинарные кавычки апостроф `expr length "$msg"` или конструкцию «$(command)», по сути они равны, и лучше всегда использовать в двойных кавычках «$(...)».

Определим, с каких символов начинается обращение уже с правильным синтаксисом «$(expr ...)»:

```
user_position="$(expr index "$msg" ', ')"
user_position="$(expr $user_position + 2)"
```

Выведем часть подстроки, начиная с первого символа обращения

```
localadmin@astra:~$ echo "$(expr substr "$msg" $user_position 22)"
уважаемый пользователь
```

Выведем первые 6 символов

```
localadmin@astra:~$ echo `expr substr "$msg" 1 6`
Привет
```

11.5.3 Числа в Bash

В языке Bash нет числовых переменных, есть только строки. Однако, это не означает, что вам недоступны математические операции и работать можно только целыми числами. О дробях мы

поговорим в следующем разделе, а пока используйте команду `let`, чтобы присваиваемое значение было интерпретировано как математическое выражение, например скрипт `millenium.sh`:

```
millenium=2000; year_now=$(date +%Y)
years_after_millenium=$year_now-$millenium
echo "Прошло $years_after_milleniumr. после миллениума"
```

```
localadmin@astra:~$ ./millenium.sh
Прошло 2024-2000г. после миллениума
```

Как видим переменные просто подставились в строку без каких либо вычислений, тогда используйте инструкцию let для вычисления:

```
millenium=2000; year_now=$(date +%Y)
let years_after_millenium=$year_now - $millenium
echo "Прошло ${years_after_millenium}г. после миллениума"
```

```
localadmin@astra:~$ ./millenium.sh
Прошло 24г. после миллениума
```

При использовании команды `let` вы можете оперировать не только целыми числами в десятеричной системе счисления. Поставьте префикс `0`, и это будет уже восьмеричное число, а если поставите префикс `0x`, то получите шестнадцатеричное число и только не удивляйтесь, что после «let c=011+022» переменная $c равна 27, а не 33, т.к. результат будет выводиться как обычно в десятичном формате.

Выражения могут быть заданы не только командой `let`, но также с помощью универсального обработчика выражений `expr` и с помощью двойных круглых скобок. Посчитаем выражение с помощью `expr`:

```
millenium=2000; year_now=$(date +%Y)
years_after_millenium="$(expr $year_now - $millenium)"
echo "Прошло ${years_after_millenium}г. после миллениума"
```

```
localadmin@astra:~$ ./millenium.sh
Прошло 24г. после миллениума
```

Посчитаем следующий год *next_ year.sh* через двойные скобки `$(( выражение ))`:

```
year_now="$(date +%Y)"
next_year="$((year_now + 1))"
echo "Следующий год $next_year"
```

```
localadmin@astra:~$ ./next_year.sh
Следующий год 2025
```

Двойные круглые скобки используются также в циклах и могут быть задействованы для присвоения значений переменным в стиле языка Си, когда с обеих сторон от символа «=» стоят пробелы и могут использоваться дополнительные математические операции:

```
localadmin@astra:~$ (( year = 2000 + 24 ))
localadmin@astra:~$ echo "Год $year"
Год 2024
```

Язык CMD придерживается тех же принципов, только вместо команды `let` используется команда `set` с ключом /a. Давайте посмотрим, как будет выглядеть тот же пример на CMD.

Пример вычислений на CMD:

```
C:\> set a=2000
C:\> set b=24
C:\> set /a c=a+b
2024
C:\> echo Год %c%
Год 2024
```

Если же говорить о PowerShell, то в части типизации данных он намного ближе к классическим языкам программирования и поддерживает не только числа, строки, массивы, но и специфические объекты для работы с датами, хеш-таблицами, XML-структурами и пр. Поэтому на PowerShell все максимально привычно для программистов.

Пример вычислений на PowerShell:

```
PS> [int]$a = 2000
PS> [int]$b = 24
PS> [int]$c = $a + $b
PS> [string]$msg = "Year $c"
PS> echo $msg
Год 2024
```

11.5.4 Дробные числа и числа с плавающей точкой

Для обработки чисел с дробной частью в самом языке Bash ничего нет, но можно использовать сторонние утилиты, например, утилиты `awk`, `bc` или `python`:

```
localadmin@astra:~$ year_now=`date +%Y`
localadmin@astra:~$ echo "$year_now^2 = `awk "BEGIN{print $year_now ^ 2}"`"
2024^2 = 4096576

localadmin@astra:~$ echo "$(bc <<< 'scale=2; 100/3')"
33.33

localadmin@astra:~$ echo 'print(10/3+65.4)' | python3
68.73333333333333
```

11.5.5 Массивы

Если переменные позволяют хранить только одно значение, то массивы предназначены для хранения списков, в чем и состоит их ключевое преимущество. Для обращения к конкретному элементу списка нужно указать имя массива и порядковый номер этого элемента (так называемый индекс).

Массивы очень помогают в разработке скриптов на Bash. Приведем пример, демонстрирующий возможность работы со списком пользователей.

Пример на Bash:

```
# Определим массив из трех элементов
users=("Ivanov" "Petrov" "Sidorov")

# Добавим четвертый элемент в конец
users[3]="Kuznetsov"

# Выведем все элементы массива на экран
echo ${users[\*]}
Ivanov Petrov Sidorov Kuznetsov

# Выведем количество элементов в массиве
echo ${#users[\*]}
4

# Выведем четвертый элемент массива, учитывая, что нумерация начинается с 0
echo ${users[3]}
Kuznetsov

# Пройдем циклом по всем элементам массива
for user in ${users[\*]}; do echo $user; done
Ivanov
Petrov
Sidorov
Kuznetsov
```

Язык CMD разрабатывали с оглядкой на Unix, но массивов в нем так и не появилось. Есть, конечно, несколько обходных приемов, но использовать их крайне неудобно.

В языке PowerShell обычные массивы представляют наборы данных, в которые возможно добавить новые элементы `$myArray += 1, 2, 3, 4`, но также есть много других классов объектов, с помощью которых можно реализовать все, что душе угодно. Наиболее популярным из них является ArrayList.

Пример массивов на PowerShell:

```
PS C:\> $users = [System.Collections.ArrayList]@("Ivanov", "Petrov", "Sidorov")
PS C:\> $users.Add("Kuznetsov")
PS C:\> echo $users
Ivanov
Petrov
Sidorov
Kuznetsov
```

```
PS C:\> echo $users[3]
Kuznetsov

PS C:\> foreach ($user in $users) { echo $user }
Ivanov
Petrov
Sidorov
Kuznetsov
```

11.5.6 Ветвления

Ветвление является конструкцией языка программирования, которая позволяет направить алгоритм на выполнение одного из возможных блоков команд в зависимости от результатов проверки заданного условия.

Для создания ветвления в Bash используется конструкция if/then/elif/else/fi. Интерпретатор выполняет условное выражение и, если код возврата равен 0, то переходит далее к выполнению команд основной ветки, так как 0 означает «успех».

Существует два способа определения условных выражений:

- Старая команда `test`, синонимом которой являются одинарные квадратные скобки: `[ ... ]`. Эта команда воспринимает выражение как операцию сравнения или как файловую проверку и возвращает код завершения, где: **0** - истина, **1** - ложь. Команда не поддерживает регулярные выражения, давно устарела и оставлена исключительно для обеспечения обратной совместимости, например, скрипты загрузчика GRUB.

- Современный вариант команды: `test [[ ... ]]`, который обладает расширенными функциями, в т. ч. поддерживает составные условия сравнения с использованием логических операторов «&&» и «||».

В качестве примера проработки условий создадим проверку, является ли пользователь суперпользователем (root), и если нет, то выполняем выход с ошибкой.

Пример инсталлятора на Bash `installer.sh`:

```
#!/bin/bash
echo "Установка необходимых пакетов"
if [[ "$UID" -eq 0 ]]; then
  echo "- Начинаем установку пакетов"
else
  echo "- Для установки приложений запустите скрипт от пользователя root"
  exit 99
fi
# выполняем установку необходимых пакетов для работы
apt install -y git curl wget jq
```

Запуск скрипта от root пользователя:

```
localadmin@astra:~$ bash installer.sh
Установка необходимых пакетов
- Для установки приложений запустите скрипт от пользователя root
localadmin@astra:~$ sudo bash installer.sh
Установка необходимых пакетов
- Начинаем установку пакетов
Чтение списков пакетов... Готово
Построение дерева зависимостей
Чтение информации о состоянии... Готово
...
```

Ветвление может быть записано в одну строку. Обратите внимание на синтаксис выражения – пробелы и символы разделения «;». Обратите внимание на пробелы между квадратными скобками и условным выражением:

```
root@astra:~# if [[ "$UID" -eq 0 ]]; then echo "root"; else echo "user"; fi
root
```

Их отсутствие вызовет синтаксическую ошибку:

```
localadmin@astra:~$ if [["$UID" -eq 0]]; then echo "root"; else echo "user"; fi
bash: [[1000: команда не найдена
user
```

Пропуск символа «;» в необходимом месте тоже приведет к ошибке:

```
localadmin@astra:~$ if [[ "$UID" -eq 0 ]] then echo "root"; else echo "user"; fi
bash: синтаксическая ошибка рядом с неожиданным маркером «then»
```

В языке CMD используется конструкция «if условие (...) else (...)» с группировкой команд в блоки с помощью круглых скобок, а в языке PowerShell использует синтаксис, который максимально приближен к языку Си «if (условие) {...} else {...}». Но, как известно, на вкус и цвет все фломастеры разные, поэтому кому-то окажется ближе Си-шный синтаксис PowerShell, а кто-то придет в восторг от того, что закрывающая инструкция блока fi – это тот же if только в зеркальном отражении (а ыв ьсилавинембо в евтстед имакворфиш?).

При сравнении целых чисел в скриптах Bash используют следующие операторы:

- `-eq` — равно (от англ. equal — равный)
- `-ne` — не равно (от англ. not equal — неравный)
- `-gt` — больше (от англ. greater than — больше чем)
- `-ge` — больше или равно (от англ. greater or equal — больше или равно)
- `-lt` — меньше (от англ. lower than — меньше чем)

- `-le` — меньше или равно (от англ. lower or equal — меньше или равно)

При сравнении целых чисел внутри двойных круглых скобок (выражений) можно использовать привычные операторы сравнения <, <=, >, >=, например, ((«$a» < «$b»)).

При сравнении строк используют следующие операторы:

- `=` или `==` — строки равны
- `!=` — строки не равны
- `<` — первая строка меньше по порядку сортировки ASCII
- `>` — первая строка больше по порядку сортировки ASCII
- `-z` — строка является «пустой», т.е. имеет нулевую длину
- `-n` — строка не является «пустой», т.е. содержит символы

11.5.7 Циклы

Цикл — является конструкцией языка программирования, которая позволяет выполнять определенный блок команд многократно до тех пор, пока не будет достигнуто условие выхода из цикла.

Циклы for традиционно используются в тех случаях, когда количество итераций известно заранее. На языке Bash это можно реализовать, используя условие в двойных круглых скобках:

```
for((инициализация;условие;счетчик)) do
  # блок команд
done
```

Давайте создадим в простом цикле 10 новых пользователей. Пример цикла на Bash:

```
localadmin@astra:~$ for((i=1;i<=10;i++)) do echo "Создаем user$i"; sudo useradd "user
↪$i"; done
Создаем user1
Создаем user2
Создаем user3
Создаем user4
Создаем user5
Создаем user6
Создаем user7
Создаем user8
Создаем user9
Создаем user10
```

Можно использовать цикл for и для обхода по всем элементам массива.

Пример на Bash:

```
localadmin@astra:~$ users=("Ivanov" "Petrov" "Sidorov")
localadmin@astra:~$ for user in ${users[\*]}; do echo "Создаем $user"; sudo useradd "
↪$user"; done
Создаем Ivanov
Создаем Petrov
Создаем Sidorov
```

Циклы while и until используются обычно в тех случаях, когда количество итераций заранее не определено. Прервать бесконечный цикл можно также командой `break`.

Давайте напишем скрипт, который в цикле будет проверять доступность интернет связи на компьютере и выведет время разрыва. Проверять связь будем утилитой ping с параметром -c, который позволяет задать количество запросов. Для эмуляции падения сети будем использовать в соседнем окне команду `sudo systemctl stop networking.service`.

Скрипт проверки Интернета на Bash `check_internet.sh`:

```
#! /bin/bash
host=$1
echo "Проверяем работу сети Интернет по доступности узла $host"
while ping -c 1 $host 1>/dev/null; do
  echo -n "."
  sleep 1
done
echo "Связь пропала $(date)"
```

Выполним скрипт проверки Интернета:

```
localadmin@astra:~$ ./check_internet.sh 77.88.8.8
Проверяем работу сети Интернет по доступности узла 77.88.8.8
..............................connect: Сеть недоступна
Связь пропала Пн мая 20 19:15:48 MSK 2024
```

11.5.8 Функции

В языке Bash, как в самом настоящем языке программирования, есть функции.

Функция — это блок программного кода, который решает узконаправленную задачу и может быть вызван как подпрограмма со своим набором параметров. Функции должны использоваться везде, где есть фрагменты повторяющегося кода.

Передача параметров в функцию выполняется с помощью обычных аргументов (позиционных параметров), а возврат значения осуществляется через код завершения. Однако вместо команды `exit`, которая предназначена для выхода из сценария Bash, используется команда `return`, имеющая тот же синтаксис.

Пример на Bash:

```
cat load_files.sh
#! /bin/bash
load_file () {
  start=$(date +%s)
  wget $1 2>/dev/null
  stop=$(date +%s)
  let total_time=stop-start
  echo ""
  echo "Загружаем файл $1 сек"
  echo "Время загрузки $total_time сек"
  return 0
}

load_file "https://download.yandex.ru/downloadable_soft/browser/linux/yandex-browser-
↪downloader.deb"
load_file "https://www.opennet.ru/man.shtml?topic=curl-config&category=1&russian=3"
```

```
localadmin@astra:~$ ./load_files.sh
Загружаем файл https://download.yandex.ru/downloadable_soft/browser/linux/yandex-
↪browser-
downloader.deb сек
Время загрузки 2 сек
Загружаем файл https://www.opennet.ru/man.shtml?topic=curl-config&category=1&
↪russian=3 сек
Время загрузки 1 сек
```

Коды возврата – это хорошо, но хотелось бы получить назад какую-нибудь строку. Существуют разные техники, одна из которых предусматривает работу с потоками вывода.

Пример `welcome.sh` на Bash:

```
#! /bin/bash
welcome () {
  res="Привет, $1!"
  echo "$res"
}
msg=$(welcome "Александр")
echo $msg
```

Выполним пример `welcome.sh`:

```
localadmin@astra:~$ ./welcome.sh
Привет, Александр!
```

В языке CMD функций нет совсем, а в PowerShell, напротив, есть все необходимое и даже больше – вы без проблем можете передавать в функцию параметры и получать результат, причем возвращаемое значение может быть не только числом, строкой, но и коллекцией объектов.

11.5.9 Экранирование символов

Кавычки в программном коде нужны не только для обозначения границ строк, они также экранируют специальные символы. На языке Bash вы можете использовать как двойные, так и одинарные кавычки (апострофы), но работают они по-разному:

- Строки, заключенные в двойные кавычки, экранируют пробелы, но позволяют подставлять значения переменных.
- Строки, заключенные в одинарные кавычки, экранируют все символы и передают значение «как есть», т.е. подстановка переменных в таких строках не выполняется.

Пример на Bash:

```
localadmin@astra:~$ a=2000
localadmin@astra:~$ echo "Год $a"
Год 2000

localadmin@astra:~$ echo 'Год $a'
Год $a
```

Одинарные кавычки экранируют все символы, включая экранирующий символ обратной косой черты «\», поэтому в тех случаях, когда в тексте нужен апостроф, приходится прибегать к следующему синтаксису.

Пример экранирования символов на Bash:

```
localadmin@astra:~$ msg="Апостроф ставят перед s (например, Ann\'s), когда что-то␣
↪принадлежит кому-либо"
localadmin@astra:~$ echo "$msg"
Апостроф ставят перед s (например, Ann's), когда что-то принадлежит кому-либо
```

Внимание: Для экранирования строк с секретами, в которых может содержаться символ доллара, рекомендуется использовать одинарные кавычки. Если же использовать двойные кавычки, то в секрете «Pas$w0rd» часть подстроки «$w0rd» будет интерпретирована как подстановка значения переменной.

Еще пример где часть текста пропала, т.к. была предпринята попытка подстановки несуществующей переменной:

```
localadmin@astra:~$ echo "Pas$w0rd"
Pas
```

При использовании одинарных кавычек текст секрета записан в переменную полностью:

```
localadmin@astra:~$ echo 'Pas$w0rd'
Pas$w0rd
```

Приведем еще один пример в качестве дополнительного пояснения проблемы:

```
localadmin@astra:~$ w0rd=ВНИМАНИЕ
localadmin@astra:~$ echo "Pas$w0rd"
PasВНИМАНИЕ
```

К сожалению, на этой ошибке попадаются даже самые толковые системные администраторы. Причем по несколько раз.

В языке CMD, как мы уже говорили, для присвоения значений кавычки не требуются — все, что будет справа от символа «=», считается значением переменной. Пример на CMD:

```
C:\> set a=a=2000
C:\> echo %a%
a=2000
```

Тем не менее, если при вызове команд параметры состоят из нескольких слов, то без двойных кавычек не обойтись.

Пример кавычек в переменной на CMD:

```
C:\>dir "C:\Program Files"
 Volume in drive C has no label.
 Volume Serial Number is 16B1-A54B

 Directory of C:\Program Files

05/23/2024  03:05 AM     <DIR>        .
05/23/2024  03:05 AM     <DIR>        ..
12/07/2019  02:31 AM     <DIR>        Common Files
05/23/2024  04:14 AM     <DIR>        Internet Explorer
...
```

Язык PowerShell перенял у Bash подход в работе с кавычками, поэтому в строках, заключенных в одинарные кавычки, подстановка значений переменных не выполняется.

Пример на PowerShell:

```
PS > $a = 2000
PS > echo "Год $a"
Год 2000
PS > echo 'Год $a'
Год $a
```

Вместе с тем синтаксис PowerShell упрощает экранирование кавычек — достаточно использовать символ кавычки дважды, чтобы первый из них экранировал второй.

Пример на PowerShell:

```
PS > $msg = 'Апостроф ставят перед s (например, Ann''s), когда что-то принадлежит␣
↪кому-либо'
PS > echo $msg
Апостроф ставят перед s (например, Ann's), когда что-то принадлежит кому-либо
```

11.5.10 Подстановка команд

Подстановка команд позволяет использовать вывод команды вместо самого имени команды. Она позволяет запускать команду прямо в тексте, например, в команде `echo`. Для этого команду необходимо заключить в конструкцию вида $(*command*). Этот синтаксис соответствует стандарту POSIX.

Выполним команду whoami прямо в аргументе команды echo:

```
localadmin@astra:~$ echo "Я $(whoami)"
Я localadmin
```

При таком запуске команды ее выполнение происходит во вложенной оболочке.

Существует и устаревшая форма синтаксиса, когда команда обрамляется обратными кавычками (теми, что на клавише с тильдой или буквой Ё) `` `command` ``. Вы можете встретить и устаревшую форму синтаксиса на исполнение команды:

```
localadmin@astra:~$ echo "Я `whoami`"
Я localadmin
```

Устаревшую форму применять не рекомендуется. При использовании аргументов с кавычками в запускаемой команде она может выдавать противоречивый результат. Формат $(*command*) лишен этого недостатка. Кроме того, при каскадном вызове команд устаревшая оболочка менее читабельна:

С устаревшей формой читабельность кода оставляет желать лучшего:

```
localadmin@astra:~$ echo `echo `echo `echo 1` 2` 3` 4
1 2 3 4
```

Тут читабельность существенно лучше:

```
localadmin@astra:~$ echo "$(echo $(echo $(echo 1) 2) 3) 4"
1 2 3 4
```

В языке CMD невозможно выполнить подстановку команды напрямую, только используя файл и переменную.

Пример подстановки команды на CMD:

```
C:\> whoami > name.txt
C:\> set /p name=<name.txt
C:\> echo Я %name%
Я pc-1\\admin
```

Язык PowerShell позволяет выполнять подстановку команд и использует точно такой же синтаксис, как и в POSIX варианте Bash.

Пример подстановки команды на PowerShell:

```
PS> Write-Host "Я $(whoami)"
Я admin
```

11.6 Специальные переменные

В предыдущем разделе мы рассмотрели локальные переменные, область видимости которых ограничена блоком кода или телом функции, но в операционных системах переменные используются значительно шире.

11.6.1 Переменные окружения

Переменные окружения (среды) – это глобальные системные переменные, с помощью которых выполняется дополнительная настройка работы системных и прикладных приложений. Мы уже рассматривали эти переменные в модуле 4 «Работа с терминалом и оболочкой Bash» и знаем следующее:

- Имена переменных окружения традиционно пишут большими буквами, например, PATH, USER и т.д.
- Переменные окружения задаются в файлах: `/etc/.profile`, `~/.profile`, `~/.bashrc`, `~/.bash_profile` и инициализируются при загрузке оболочки.
- Полный список всех переменных окружения можно посмотреть с помощью утилиты `env`. Установить переменную окружения можно с помощью команды `export`, а удалить с помощью команды `unset`.

При переходе на Linux вам следует обратить внимание на тот факт, что в операционных системах Microsoft переменные окружения в основном используются только для нужд системы, а прикладных приложений, которые можно было бы настраивать через переменные окружения, практически нет. Так сложилось исторически потому, что в операционных системах MS-DOS и Windows все файлы приложения хранятся в одном месте, и для его конфигурирования проще было использовать INI-файлы.

В операционной системе Linux переменные окружения используются для конфигурирования прикладных приложений повсеместно, поэтому, например, вас не должно удивлять, что перед вызовом утилиты `kinit` может быть команда `env`, устанавливающая значение переменной окружения «**env** KRB5_TRACE=/dev/stdout **kinit**».

11.6.2 Позиционные параметры

При написании скриптов автоматизации крайне важно уметь использовать позиционные параметры, с помощью которых можно вычитывать значения параметров, которые передаются скрипту из командной строки, как при вызове исполняемых файлов. Это совсем несложно, но значительно упрощает эксплуатацию скриптов в будущем.

Параметры вызова скрипта подставляются в цифровые переменные $1, $2 ... ${N}, где число определяет порядковый номер параметра в списке полученных значений. При использовании десяти и более параметров порядковый номер следует заключать в фигурные скобки, например, ${10}, ${11} и т.д. Специальные переменные $* и $@ содержат все позиционные параметры вместе, разделенные через пробел. Приведем пример скрипта `remove_empty_files.sh`, который получает в качестве входного параметра папку и удаляет из нее все пустые файлы:

```
#!/bin/bash
echo "Удаляем пустые файлы из каталога: $1"
find $1 -empty -type f -delete
echo "Готово"
```

```
localadmin@astra:~$ ./remove_empty_files.sh .
Удаляем пустые файлы из каталога: .
Готово
```

Язык CMD не стал выдумывать что-то новое и позаимствовал тот же синтаксис, заменив символ доллара на знак процента: %1, %2 и %*.

Разработчики PowerShell, конечно же, реализовали возможность оперировать как простым массивом параметров args в стиле Bash, обращаясь к элементам по индексу $args[0], $args[1] ... $args[N]. Но не менее удобно использовать конструкцию Param, которая позволяет в несколько строк описывать именованные ключи для их последующего использования в формате: `.\remove-empty-files.ps1 -Path $PWD.Path`

11.7 Отладка скриптов Bash

Командная оболочка Bash не имеет своего отладчика, каких-либо специальных команд или конструкций обработки ошибок, типа try...catch. Синтаксические ошибки или опечатки могут вызывать сообщения об ошибках, которые мало помогают в отладке. Инструменты, которые могут

помочь вам при отладке неработающих сценариев:

- команда `echo` в критических точках сценария поможет отследить состояние переменных и отобразить ход исполнения.
- команда `tee`, которая поможет перенаправить ввод/вывод, чтобы вы смогли проверить потоки данных в критических местах.
- ключ `-n` проверит наличие синтаксических ошибок, не запуская сам сценарий. Если ошибок нет, то команда отработает без сообщений.
- ключ `-v` выводит каждую команду прежде, чем она будет выполнена.

```
localadmin@astra:~$ touch f1
localadmin@astra:~$ touch f2
localadmin@astra:~$ bash -v remove_empty_files.sh
```

Результат выполнения отладки:

```
# ~/.bashrc: executed by bash(1) for non-login shells.
# see /usr/share/doc/bash/examples/startup-files (in the package bash-doc)
# for examples

# If not running interactively, don't do anything
case $- in
    *i*) ;;
    *) return;;
esac
#!/bin/bash
echo "Удаляем файлы из каталога: $1"
Удаляем файлы из каталога:
num=$(find $1 -empty -type f -delete | wc -l)
echo "Готово $num"
Готово 0
```

- ключ `-x` выводит результат исполнения каждой команды в краткой форме.

Пример запуск с ключом `-x` на Bash:

```
localadmin@astra:~$ touch f1
localadmin@astra:~$ touch f2
localadmin@astra:~$ bash -x remove_empty_files.sh
+ case $- in
+ return
+ echo 'Удаляем файлы из каталога: .'
Удаляем файлы из каталога: .
++ find . -empty -type f -delete
++ wc -l
+ num=0
+ echo 'Готово 0'
```

- устанавливайте ловушку с помощью команды `trap` на сигналы **SIGINT**, **SIGTERM**, **SIGHUP**, **SIGQUIT**, **EXIT**, **ERR**, например:

```
#!/bin/bash
trap 'echo Список переменных --- a = $1 b = $2 code=$?' EXIT
if [[ "$1" -lt "$2" ]]; then
  exit 0
else
  exit 1
fi
```

Вывод будет следующим:

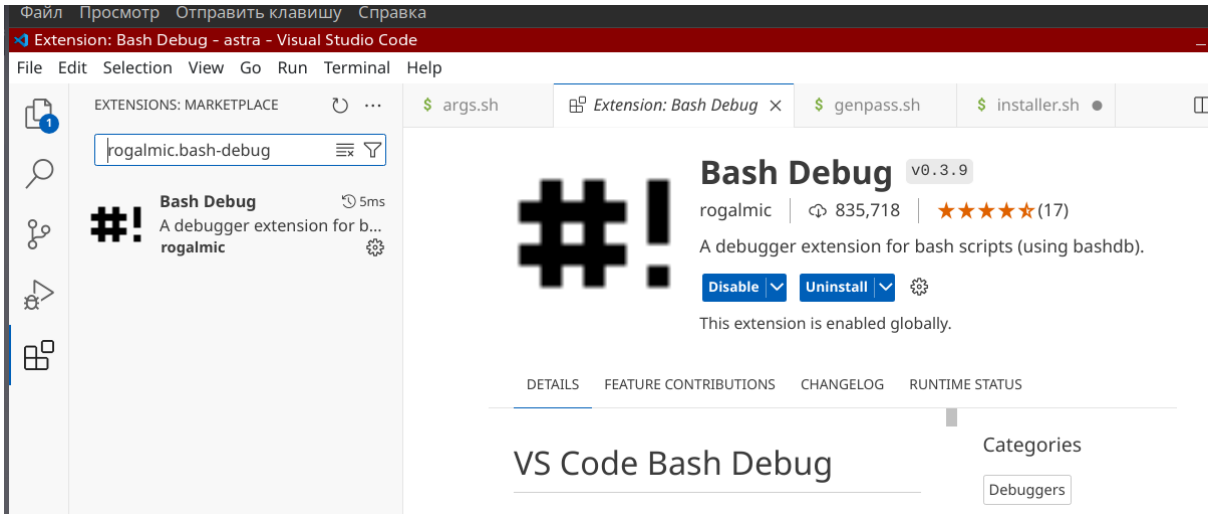
```
localadmin@astra:~$ ./trap.sh 100 200
Список переменных --- a = 100 b = 200 code=0
localadmin@astra:~$ ./trap.sh 200 100
Список переменных --- a = 200 b = 100 code=1
```

- используйте команду `trap` с аргументом DEBUG, чтобы заставить сценарий выполнять указанное действие после выполнения каждой команды. Это можно использовать для трассировки переменных.

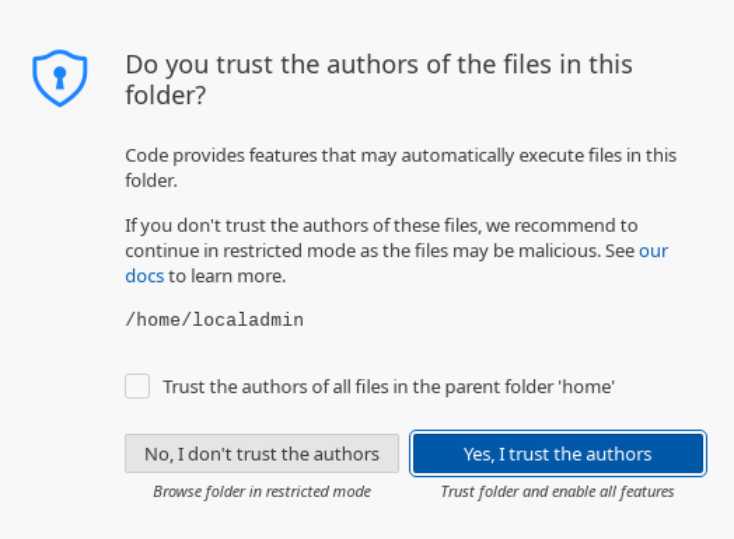
Удобным инструментом отладки является плагин `rogalmic.bash-debug` для Visual Studio Code, который выполняет скрипт по шагам, позволяет устанавливать точки остановки и отображает значения переменных.

Вот пошаговая инструкция по его установке и использованию:

1. Установим vs code и расширение rogalmic.bash-debug.



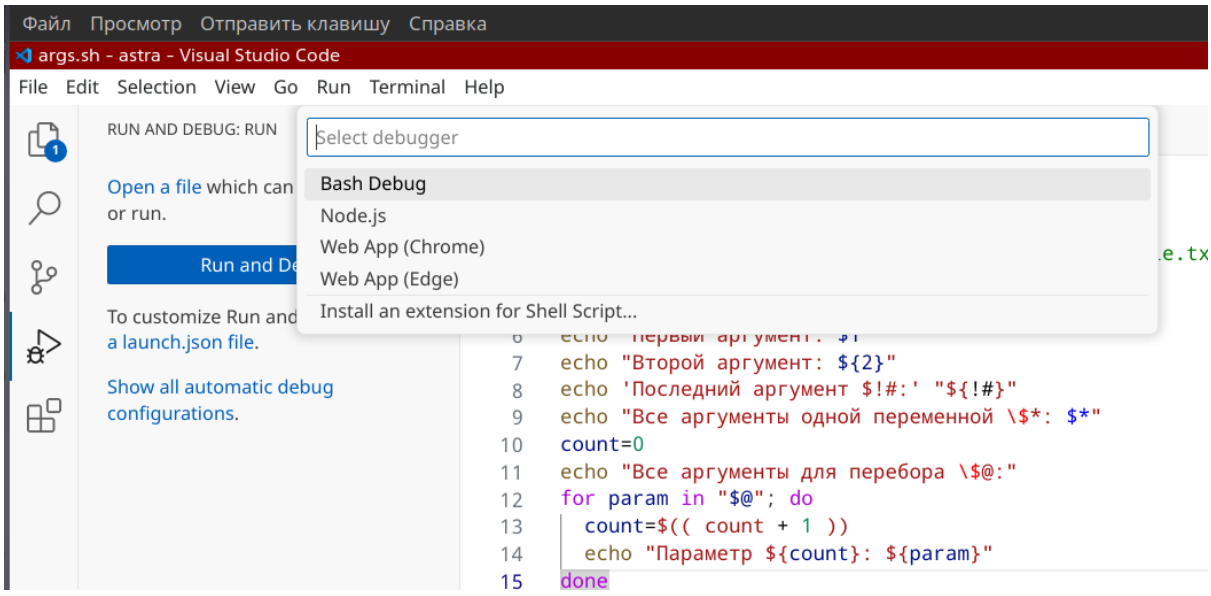
2. Откроем папку со скриптом. Если возникло предупреждение о том, что источник является недоверенным, переубедите редактор нажатием кнопки «Yes, I trust the authors».



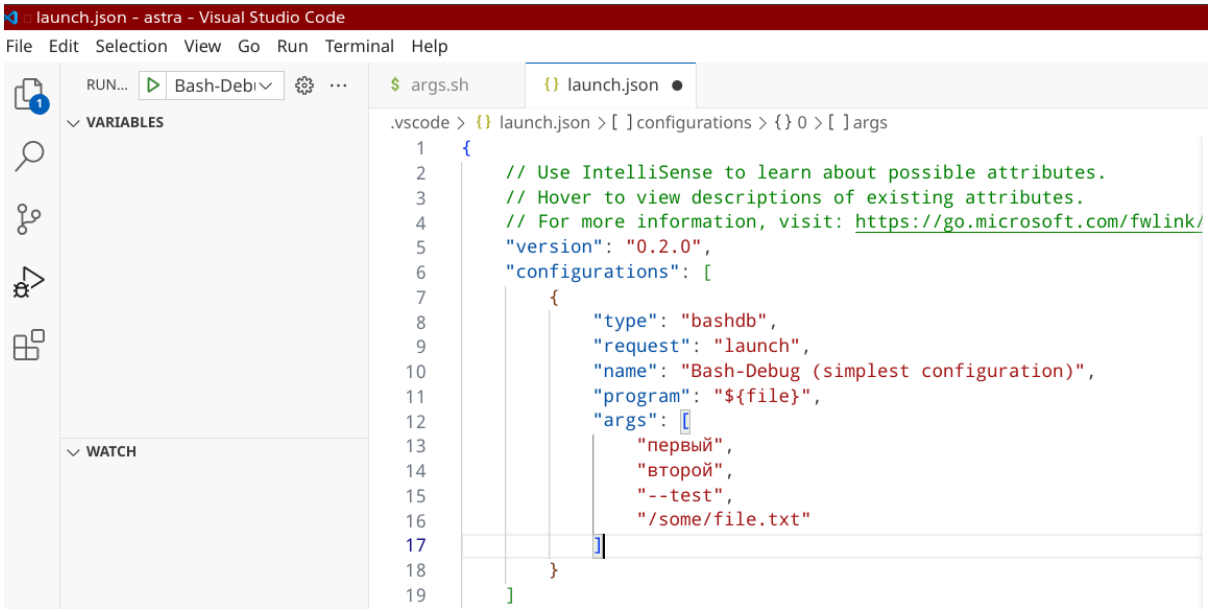
3. Выберите нужный файл, чтобы открыть его на редактирование.

4. Нажмите горячее сочетание клавиш Ctrl + Shift + G для замены левой вкладки на «Run and Debug».

5. Перейдите по ссылке *create a launch.json file* и в открывшемся меню выберите «Bash Debug».

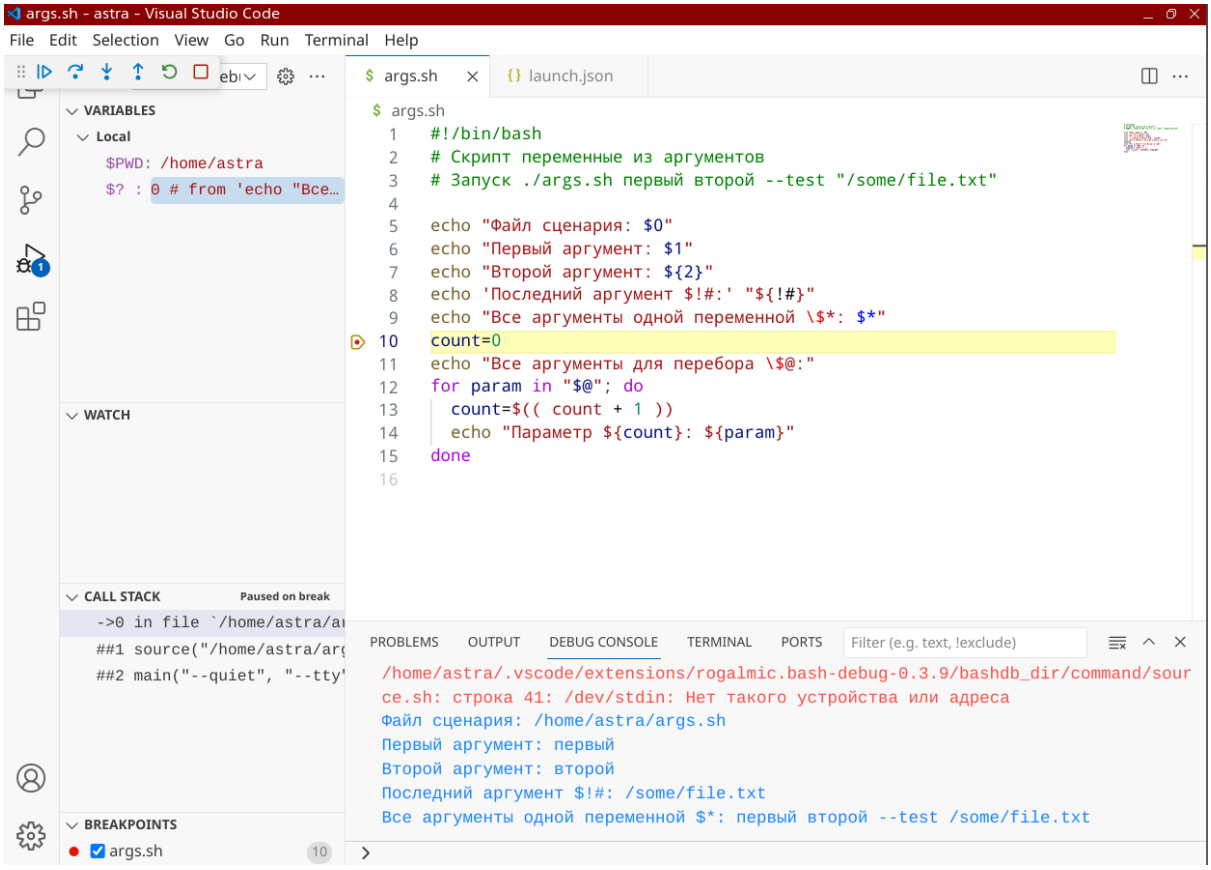


6. В редакторе *launch.json* добавьте необходимые аргументы для запуска.



7. Перейдите в файл скрипта, установите курсор на нужную строку и нажмите F9, чтобы создать точку остановки.

8. Запустите сценарий клавишей F5, выполнение скрипта будет остановлено в выбранной точке. С помощью клавиши F11 переходите к следующим командам. Информация о состоянии выполнения будет отображаться на вкладке «Debug console».



11.8 Эквиваленты команд Bash и PowerShell

В качестве подсказки приведем сопоставление команд, которые решают похожие задачи:

Команда Linux	Команда PowerShell	Псевдонимы PowerShell
cat	Get-Content	
cd	Set-Location	cd, sl, chdir
cp	Copy-Item	cp, copy cpi
find	Get-ChildItem	
grep	Select-String	
kill	Stop-Process	
ls	Get-ChildItem	ls, dir
mkdir	New-Item -ItemType Directory	
ping	TestConnection	
ps	Get-Process	
pwd	Get-Location	pwd, gl
rm	Remove-Item	rm, ri, rmdir, rd, del
tail	Get-Content	
touch	New-Item -ItemType File	
wc	Measure-Object	
whoami	whoami	

11.9 Практика и тестирование

11.9.1 Практическая работа: Модуль 11. Работа со сценариями (скриптами) оболочки Bash

См. также *Требования, правила и цели выполнения практической работы*

11.9.1.1 Практические задания

Задание 1.

1) Создайте скрипт приветствие hello.sh.

2) Назначьте скрипт запускаемым и выполните.

3) Выполните приветствие через bash.

Задание 2.

Переменные: Генератор пароля часть 1.

В операционной системе Linux пароль можно сгенерировать с помощью утилиты pwgen, но мы в учебных целях напишем скрипт, который способен решить эту задачу.

Создайте переменную алфавита для генерации простого и сложного пароля:

1) Задайте переменную для цифр, которые будем использовать в простом пароле.
2) Задайте переменную для спец. символов, которые будут в сложном пароле.
3) Задайте переменную алфавита из символов верхнего регистра A-Z + нижнего a-z + цифр + спец. символы.
4) Выведите весь алфавит: ABCDEFGJKLMNOPQRSTVUWXYZabcdefghjklmnopqrstvuwxyz0123456789+-=_.~|!@#%^

Задание 3.

Циклы: Генератор пароля часть 2.

1) Вычислите в переменную количество символов в алфавите легкого пароля алфавит A-Z + a-z + 0-9.
2) Вычислите положение случайного символа для легкого пароля.
3) Вырежьте символ из алфавита по случайному положению.
4) Приклейте вырезанный символ к простому паролю.
5) Повторите действие в цикле, добавив 14 случайных символов с переменной пароля.
6) Выведите легкий пароль из 14 случайных символов.

Самостоятельно *

1) По аналогии сделайте вывод простого и сложного пароля.
2) Выведите список из 9 пар паролей (Простой-Сложный)

Задание 4.

Условия: инсталлятор с проверкой root прав.

1) Проверьте, является ли текущий пользователь суперпользователем root.

2) Если пользователь не является root, то выведите сообщение об ошибке и завершите работу скрипта.

3) Если пользователь root, то установите apt install -y git curl wget jq.

Задание 5.

Переменные из аргументов.

Создайте скрипт, который выводит аргументы из командной строки:

1) Выведите «Текущий файл сценария **$0**».

2) Выведите «Первый аргумент: **$1**».

3) Выведите «Второй аргумент: **${2}**».

4) Выведите «Последний аргумент **${!#}**».

5) Выведите «Все аргументы одной переменной **$***».

6) Выведите в цикле «Все аргументы из переменной **$@**».

Задание 6.

Отладка скрипта. *

1) Запустите отладку скрипта args.sh через bash -x и посмотрите трассировку.

2) Установите vs code.

3) Установите расширение отладки vs code rogalmic.bash-debug.

4) Откройте в VS Code домашнюю папку со скриптом args.sh.

5) Настройте параметры запуска launch.json.

6) Откройте файл скрипта и поставьте точку остановки на 10 строке (Клавиша <F9>).

7) Запустите отладку <F5> скрипта args.sh через расширение bash debug.

11.9.1.2 Ответы на практические задания (пошаговые инструкции)

Задание 1.

1) Создайте скрипт приветствие hello.sh:

```
nano hello.sh
```

```
#!/bin/bash
echo "Привет, $USER!"
```

1) Назначьте скрипт запускаемым и выполните:

```
chmod +x hello.sh
./hello.sh
```

1) Выполните приветствие через bash:

```
bash hello.sh
```

Задание 2.

Переменные: Генератор пароля часть 1.

Создайте переменную алфавита для генерации простого и сложного пароля:

1) Задайте переменную для цифр, которые будем использовать в простом пароле.

2) Задайте переменную для спец. символов, которые будут в сложном пароле.

3) Задайте переменную алфавита из символов верхнего регистра A-Z + нижнего a-z + цифр + спец. символы.

4) Выведите весь алфавит: ABCDEFGJKLMNOPQRSTVUWXYZabcdefghjklmnopqrstvuwxyz0123456789+-=_.~|!@#%^

```
#!/bin/bash
# genpass.sh - генератор списка простых и сложных паролей

# объявим переменную для цифр, которые будем использовать в простом пароле
numbers="0123456789"

# объявим переменную для спец. символов, которые будут в сложном пароле
special='+-=\_.~|!@#%^'

# комбинируем алфавит для пароля из строк, используя $numbers и $special
alphabet="ABCDEFGJKLMNOPQRSTVUWXYZabcdefghjklmnopqrstvuwxyz${numbers}${special}"

# обратимся к переменной по имени и выведем результат
echo "${alphabet}"
```

Задание 3.

Циклы: Генератор пароля часть 2.

1) Вычислите в переменную количество символов в алфавите легкого пароля алфавит A-Z + a-z + 0-9.

2) Вычислите положение случайного символа для легкого пароля.

3) Вырежьте символ из алфавита по случайному положению.

4) Приклейте вырезанный символ к простому паролю.

5) Повторите действие в цикле, добавив 14 случайных символов с переменной пароля.

6) Выведите легкий пароль из 14 случайных символов.

Решение с простым паролем.

```
#!/bin/bash
# genpass.sh - генератор списка простых и сложных паролей

# зададим переменную для цифр, которые будем использовать в простом пароле
numbers="0123456789"

# зададим переменную для спец. символов, которые будут в сложном пароле
special='+-=\_.~|!@#%^'

# комбинируем алфавит для пароля из строк, используя $numbers и $special
alphabet="ABCDEFGJKLMNOPQRSTVUWXYZabcdefghjklmnopqrstvuwxyz${numbers}${special}"

# обратимся к переменной по имени и выведем результат# echo
"${alphabet}"

# посчитаем количество символов для простого пароля
simple_count="$((${#alphabet}-${#special}))"

# очистим переменную простого пароля
unset pass_simple
# создадим простой пароль в цикле из 14 итераций
for s in $(seq 1 14); do
  # возьмем случайную позицию символа для простого пароля
  random_position="$((RANDOM % simple_count))"
  # добавим к паролю новый случайный символ
  pass_simple="${pass_simple}${alphabet:random_position:1}"
done
# выведем простой пароль
echo "${pass_simple}"
```

Самостоятельно *

1) По аналогии сделайте вывод простого и сложного пароля. cZCn7z79ClrkSW ^PjO9!pX4m%#reEGf|Qx2KR3

2) Выведите список из 9 пар паролей:

```
1 cZCn7z79ClrkSW ^PjO9!pX4m%#reEGf|Qx2KR3
2 N39apYC1C4OOGa ufFeKrouP43xUefJ.zX.Q%jV
3 e3DnEG8C9PkjkX ZbskOB@SkM|7@O#RNSOKQM^B
4 Zw6FKpEKNOZs7O Wc1+R~XGo8DkP#6Rkw0AQ2%6
5 LcBPWZO8sAYOg3 ^BoE~egaFf-SrPOCn@m_1=9G
6 UfFwYG5vazQOBR q#Po!J|yMoWQBS=3b%^ur@p\
7 TfRcADRfaNgR7J a+WwPZ#kY.Ez_kUL@LZXr=gU
8 ApmZp4ZGOmkEsO Ov6|+JdN6hh4_dK86ZLRP5_o
9 RhZruKEpRe3OOc oc39o=ck|mVS63_asGZqtgbX
```

Задание 4.

Условия: инсталлятор с проверкой root прав.

1) Проверьте, является ли текущий пользователь суперпользователем root.

2) Если пользователь не является root, то выведите сообщение об ошибке и завершите работу скрипта.

3) Если пользователь root, то установите apt install -y git curl wget jq.

Решение

```
#!/bin/bash
echo "Установка необходимых пакетов"
if [[ "$UID" -eq 0 ]]; then
 echo "- Текущий пользователь root"
else
 echo "- Для запуска установки требуются root права"
 exit 99
fi
# выполняем установку необходимых пакетов для работы
apt install -y git curl wget jq
```

Задание 5.

Переменные из аргументов.

Создайте скрипт, который выводит аргументы из командной строки:

1) Выведите «Текущий файл сценария **$0**».

2) Выведите «Первый аргумент: **$1**».

3) Выведите «Второй аргумент: **${2}**».

4) Выведите «Последний аргумент **${!#}**».

5) Выведите «Все аргументы одной переменной **$***».

6) Выведите в цикле «Все аргументы из переменной **$@**».

Запустите скрипт с разными аргументами:

```
./args.sh первый второй --test "/some/file.txt"
```

Решение

```
#!/bin/bash
# Скрипт переменные из аргументов
# Запуск ./args.sh первый второй --test "/some/file.txt"

echo "Файл сценария: $0"
echo "Первый аргумент: $1"
echo "Второй аргумент: ${2}"
echo 'Последний аргумент $!#:' "${!#}"
echo "Все аргументы одной переменной \\$\*: $\*"
count=0
echo "Все аргументы для перебора \\$@:"
for param in "$@"; do
 count=$(( count + 1 ))
 echo "Параметр ${count}: ${param}"
done
```

Задание 6.

Отладка скрипта.

1) Запустите отладку скрипта args.sh через bash -x и посмотрите трассировку.

2) Установите vs code.

3) Установите расширение отладки vs code rogalmic.bash-debug.

4) Откройте в VS Code домашнюю папку со скриптом args.sh.

5) Настройте параметры запуска launch.json.

6) Откройте файл скрипта и поставьте точку остановки на 10 строке (Клавиша <F9>).

7) Запустите отладку <F5> скрипта args.sh через расширение bash debug.

11.9.2 Тест: Модуль 11. Работа со сценариями (скриптами) оболочки Bash

Пожалуйста, ответьте на несколько вопросов, чтобы закрепить полученные знания. Правильные ответы для самопроверки вы сможете найти в конце списка. Обратите внимание, что у некоторых вопросов может быть несколько верных вариантов ответов.

11.9.2.1 Вопросы

Вопрос 1. Какая команда позволяет посмотреть, является ли команда встроенной в оболочку Bash?

a. Команда `set`

b. Команда `done`

c. Команда `type`

d. Команда `man bash`

Вопрос 2. Какая нотация используется для глобальных переменных?

a. PascalCase

b. camelCase

c. snake_case

d. UPPER_SNAKE

Вопрос 3. Какие кавычки позволяют подставить значение переменной?

a. Двойные кавычки `echo "Идентификатор пользователя $UID"`

b. Одинарные кавычки `echo 'Идентификатор пользователя $UID'`

c. Обратные кавычки ``echo ``Идентификатор пользователя $UID``` ``

Вопрос 4. Какие кавычки экранируют специальный символ доллара `$`, не позволяя подставлять значения переменных?

a. Двойные кавычки `echo "Pa$$w0rd"`

b. Одинарные кавычки `echo 'Pa$$w0rd'`

c. Обратные кавычки ``echo ``Pa$$w0rd``` ``

Вопрос 5. Какие конструкции языка позволяют повторить блок кода несколько раз в зависимости от условия?

a. for

b. while

c. if

d. until

Вопрос 6. Какой конструкцией языка можно выполнить ветвление кода?

a. set

b. exit

c. if

d. do

Вопрос 7. Как в языке Bash можно получить значение позиционных переменных?

a. С помощью `%1` ... `%N` и `%*`

b. С помощью `$1` ... `${N}`, `$*` и `$@`

c. С помощью `$args[0]` ... `$args[N]`

Вопрос 8. Какая команда позволяет очистить переменную?

a. Команда `set`

b. Команда `unset`

c. Команда `clear`

d. Команда `do`

Вопрос 9. Какая переменная позволяет получить значения всех аргументов?

a. Переменная `$0`

b. Переменная `$@`

c. Переменная `$1`

d. Переменная `$#!`

Вопрос 10. Какое расширение vscode позволяет упростить отладку bash-скриптов?

a. YAML

b. todo tree

c. launch.json

d. Bash-debug

Ответы на вопросы

1. c
2. d
3. a, c
4. b
5. a, b, d
6. c
7. b
8. b
9. b
10. d

11.10 Заключение

Сегодня мы увидели, в чем bash похож и в чем отличается от CMD/PowerShell. Объем материала огромный, а чтобы его освоить, нужно только одно – много практиковаться в написании больших, полезных и надежных скриптов.

Далее мы узнаем подробнее о профилях пользователей, порядке загрузки файлов и особенностях инициализации профиля для рабочего стола fly-wm.

11.11 Дополнительные источники информации

- Книга «Расширенное руководство по написанию сценариев Bash» (Advanced Bash Scripting Guide by Mendel Cooper)
- Рекомендации по написанию bash от Google: Shell Guide

11.12 Обратная связь

Если остались вопросы, то их всегда можно задать в специальной теме.

Модуль 12. Работа с профилями пользователей

12.1 Введение

В домене ALD Pro у нас есть групповые политики, с помощью которых мы можем централизованно управлять переменными окружения. Но для того чтобы эффективно применять эти инструменты на практике, крайне желательно иметь хорошие представления о том, как профили пользователей работают на «низком» уровне.

Из этого модуля вы узнаете, как именно Linux создает окружение пользователя и почему набор инициализационных файлов далеко не ограничивается `.bashrc`. Мы также поговорим о порядке инициализации рабочего стола fly-wm и его конфигурационных файлах, чтобы вы могли применить эти знания и для настройки графического окружения. Материал будет не очень сложный, но крайне полезный!

12.2 Профили пользователя в Linux

Под профилем пользователя в широком смысле понимают весь его домашний каталог целиком, а в более узком — набор скриптов, которые выполняются каждый раз при запуске оболочки и позволяют определить индивидуальные настройки окружения пользователя.

При создании локального пользователя и первом входе доменного пользователя в систему создается его домашний каталог, в который копируются файлы из папки `/etc/skel`, задающие основу профиля (*сокращение от англ. skeleton — каркас*). Большинству администраторов Linux хорошо известен такой файл как `.bashrc`, однако он является далеко не единственным скриптом, определяющим окружение пользователя. Есть еще `.bash_aliases`, `.bash_login`, `.bash_profile`, profile и многие другие.

Все эти файлы довольно неплохо описаны в `man bash`, однако в справке нет никаких объяснений, почему этих файлов так много и зачем все они были созданы. А без таких подробностей вы будете довольно часто сталкиваться с тем, что ваши изменения либо не оказывают желаемого эффекта, либо проявляются непредсказуемым образом. Ни первое, ни второе нас не устраивает, а еще компот.

12.2.1 Порядок инициализации профиля

Как мы знаем, за годы эволюции никсов было создано великое множество разных командных оболочек, см. рис. 1.47. Много оболочек для Linux — это очень хорошо... А учитывая, что разработчики этих приложений еще и конкурируют между собой — это очень-очень хорошо. Жаль только, что все эти «смотрикакяумею» по-разному инициализируют профиль пользователя, а в

остальном конкуренция — это, конечно, очень хорошо.

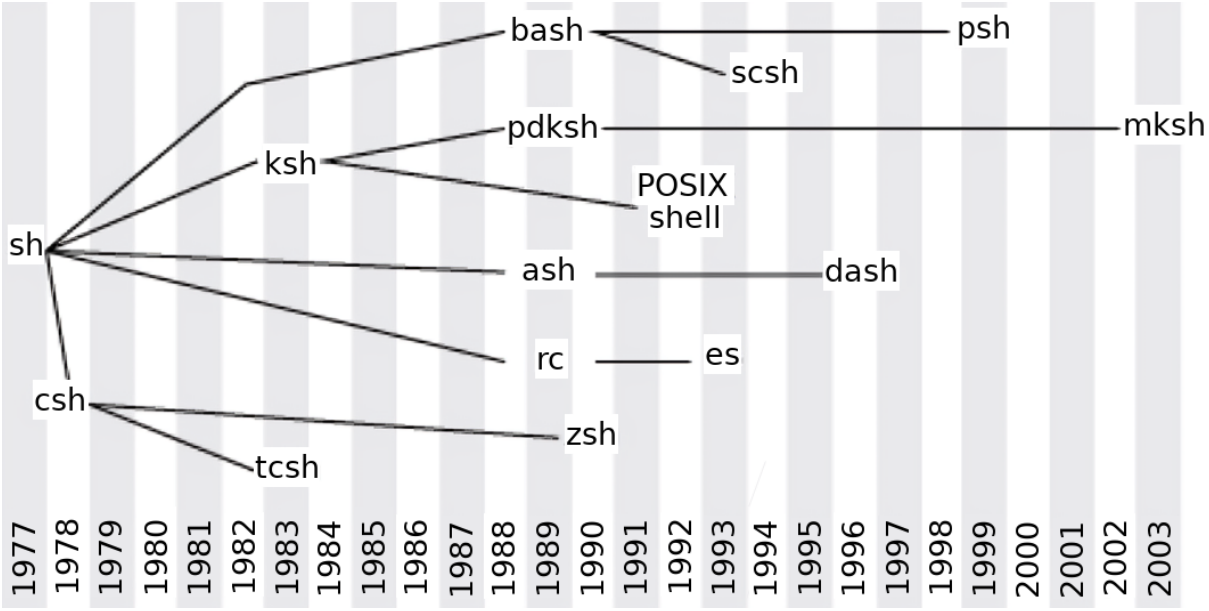


рис. 1.47 – История развития командных оболочек

Процесс инициализации профиля мы будем рассматривать на примере Bash, т.к. она завоевала наибольшую популярность и используется в Astra Linux по умолчанию. Однако приготовьтесь, что даже в этом случае мы столкнемся с кучей артефактов, унаследованных Bash от прародителей или добавленных в нее для обеспечения прямой/обратной совместимости с другими оболочками. Начать нам нужно с нескольких существенных нюансов.

Во-первых, скрипты инициализации бывают разные:

- Часть скриптов инициализации создали для того, чтобы можно было определить «параметры компьютера» в целом, т.е. их действие распространяется на всех пользователей системы сразу. Такие скрипты называют **общесистемными** (System-wide Startup Files), и обычно они находятся в каталоге `/etc`.

- Другая часть предназначена для кастомизации окружения конкретного пользователя. Они называются **пользовательскими** файлами (User-specific startup files) и находятся как раз в домашних каталогах.

- И, конечно же, не забываем про сценарии выхода **из системы** (logout). А то у нас, как обычно, такая традиция, как «присесть на дорожку», есть, а такой традиции, как «убрать за собой», нет.

Во-вторых, для возможности тонкой настройки системы порядок инициализации профиля зависит от контекста использования оболочки. Выделяют следующие режимы ее работы:

- По способу аутентификации пользователя:

- **Вход в систему с вводом учетных данных** (login) — режим, для входа в который пользователь должен пройти аутентификацию, т.е. система запросит у него учетные данные. В этом режиме окружение пользователя начинает формироваться с «чистого листа». Пример: вход с помощью утилит `login`, `su`, подключение по `ssh`.

- **Вход в сеанс без ввода учетных данных** (non-login) — режим, для входа в который от пользователя не требуется прохождение аутентификации. В этом режиме на вход скриптов инициализации подается набор переменных, которые были определены в рамках входа в систему с вводом учетных данных. Примеры: запуск `fly-term` из графики, создание новой вкладки в этом приложении, вход в новый экземпляр оболочки из текущей сессии командой bash без параметров.

- По способу взаимодействия пользователя с оболочкой:

  - **Интерактивный** или командный (interactive) — режим, в котором пользователь может вводить команды вручную и наблюдать в окне результаты их выполнения.

  - **Неинтерактивный** или пакетный (non-interactive) — режим, в котором оболочка не взаимодействует с пользователем и завершает свою работу сразу же после выполнения указанных команд. Например, локальное и удаленное выполнение строки команд или целого скрипта:

    - `bash -c 'echo $PWD'`,
    - `bash myscript.sh`,
    - `./bash myscript.sh`,
    - `ssh user@host 'echo $PWD'`.

Ну, и в-третьих, есть такая традиция отделять мух от котлет, поэтому для хранения определенных настроек выделяют отдельные файлы. Например, команды, создающие псевдонимы, выносят в отдельный файл `bash_aliases`.

Теперь мы готовы перейти к рассмотрению алгоритма, см. рис. 1.48:

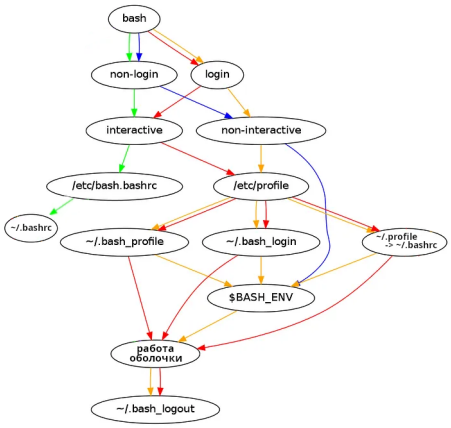


рис. 1.48 – Алгоритм инициализации профиля пользователя в оболочке Bash

12.2.1.1 Интерактивный вход без аутентификации

Когда вы открываете новое окно терминала `fly-term` или вкладку в окне этого приложения, порядок инициализации скриптов должен быть таким, как показан на схеме рис. 1.48 зелеными стрелочками. Чтобы убедиться в этом, для трассировки вызовов этих файлов добавьте:

- В начало : `echo "$(realpath $0) (начало)" >> "~/bash_initialize_log"`
- В конец: `echo "$(realpath $0) (конец)" >> "~/bash_initialize_log"`.

Результат `bash_initialize_log` будет следующим (добавили отступы для лучшего восприятия):

```
localadmin@admin:~$ cat ~/bash_initialize_log
/etc/bash.bashrc` (начало)
/etc/bash.bashrc (конец)
/home/localadmin/.bashrc (начало)
  ~/.bash_aliases (начало)
  ~/.bash_aliases (конец)
/home/localadmin/.bashrc (конец)
```

Скрипт /etc/bash.bashrc является общесистемным скриптом. В нем могут устанавливаться размер окна, внешний вид приглашения командной строки, настройки автодополнения, обработка выполнения несуществующих команд.

Следует понимать, что скрипт `/etc/bash.bashrc` начинает формировать окружение не «с чистого листа», а получает на вход ряд переменных, которые были установлены ранее в рамках «интерактивного входа в систему с аутентификацией».

Скрипт ~/.bashrc является пользовательским. В нем устанавливаются настройки сохранения истории команд HISTCONTROL, HISTSIZE, HISTFILESIZE и др, подключается файл `~/.bash_aliases` со списком псевдонимов команд.

Параметры, устанавливаемые с помощью этого скрипта, предназначены только для интерактив-

ного режима работы оболочки. При установке пакета bash-completion из этого файла дополнительно подключаются скрипты этого пакета. Кстати, буквы rc в конце имени файла означают «run command» и унаследованы от оболочки «RUNCOM», поэтому вы будете иногда встречать в системе файлы с таким суффиксом.

Скрипт ~/.bash_aliases является пользовательским и определяет перечень псевдонимов команд. Алиасы могут быть определены в скрипте `~/.bashrc`, но правильнее это делать в специальном файле :file:` ~/.bash_aliases`, который подключается из `~/.bashrc`. Приведем пару псевдонимов команд в качестве примера:

```
alias ll='ls -l'
alias la='ls -la'
```

Примечание: Графическая оболочка Fly при запуске приложений совершает дополнительные обращения к системе, которые приводят к вызову скрипта `~/.bashrc` в неинтерактивном режиме, поэтому в начале скрипта `~/.bashrc` на эту тему есть дополнительная проверка.

12.2.1.2 Неинтерактивный вход без аутентификации

Если вы запускаете скрипты по имени, передаете файл скрипта на вход приложению bash или запускаете отдельные команды через ключ `-c`, то порядок инициализации будет таким, как показано на схеме рис. 1.48 синими стрелочками, т.е. будет запускаться скрипт, определенный в переменной BASH_ENV.

Так же, как и при интерактивном входе без аутентификации, скрипт из BASH_ENV начинает формировать окружение не «с чистого листа», а получает ряд переменных, которые были установлены ранее в рамках «интерактивного входа в систему с аутентификацией».

12.2.1.3 Интерактивный вход с аутентификацией

Если вы выполняете вход в систему из консоли `Ctrl + Alt + F1`, подключаетесь через SSH или выполняете интерактивный вход утилитами `login`, `su`, `sudo -i` и др., то порядок инициализации будет таким, как показано на схеме рис. 1.48 красными стрелочками:

```
localadmin@astra:~$ cat ~/bash_initialize_log
/etc/profile (начало)
  /etc/bash.bashrc (начало)
  /etc/bash.bashrc (конец)
  /etc/profile.d/* (начало)
  /etc/profile.d/* (конец)
/etc/profile (конец)
~/.profile (начало)
```

(continues on next page)

(продолжение с предыдущей страницы)

```
  ~/.bashrc (начало)
     ~/.bash_aliases (начало)
     ~/.bash_aliases (конец)
  ~/.bashrc (конец)
~/.profile (конец)
```

Скрипт /etc/profile является общесистемным скриптом. В нем устанавливаются инструкции для настройки переменных окружения PATH, PS1, и вызывается скрипт `/etc/bash.bashrc`. Дополнительно вызываются скрипты из каталога `/etc/profile.d/*`, чтобы пользовательские приложения могли расширить набор инициализируемых переменных.

Скрипт ~/.profile является пользовательским скриптом и позволяет переопределить те переменные окружения, которые были заданы на глобальном уровне. Например, разработчики программного обеспечения могут расширить значение переменной PATH и включить в нее каталоги своих проектов.

Обратите внимание, что на втором шаге загрузки Bash выбирает один из трех файлов: `~/.bash_profile`, `~/.bash_login` или `~/.profile`. Изначально в оболочке Bourne был только `~/.profile`, но разработчики Bash добавили обработку собственных файлов, чтобы можно было определять дополнительные инструкции, специфичные для этой оболочки. В настоящий момент эта возможность используется не во всех дистрибутивах, и в Astra Linux, к примеру, этих файлов нет, поэтому запускается скрипт `~/.profile`.

12.2.1.4 Неинтерактивный вход с аутентификацией

Остался последний случай, когда мы выполняем команды в пакетном режиме, но с прохождением аутентификации, например, через ssh:

```
localadmin@astra:~$ ssh user@host 'echo $PWD'
```

Порядок инициализации будет таким, как показано на схеме рис. 1.48 желтыми стрелочками, т.е. дополнительно будет запускаться скрипт, определенный переменной BASH_ENV.

12.2.1.5 Сценарии выхода из системы

И не забываем прибраться за собой, т.е. про скрипт `~/.bash_logout`. Из этого скрипта можно, например, удалить какие-нибудь временные файлы приложения. Но не забывайте, что он срабатывает только при полном выходе из системы. Выход из сессии, например, закрытие вкладки окна терминала, не приводит к запуску этого скрипта.

Примечание: Упомянем также, что в профиле пользователя есть файл `~/.bash_history`, который

представляет собой обычный текстовый файл со списком ранее выполненных команд. Этот файл загружается в момент входа в систему, и в него выгружается история команд при заполнении буфера и корректном завершении сессии.

12.2.2 Профиль рабочего стола fly

Выбор режима рабочего стола Fly выполняется в меню «Тип сессии» в окне графического входа в систему (утилита `fly-dm`). По умолчанию предусмотрено несколько режимов, но администратор системы может добавить новые режимы, например, для систем с низкими характеристиками производительности или удаленных терминалов можно создавать режим fly-light и т.д.

Для создания нового режима необходимо добавить файл (файлы) сессии с расширением `*.desktop` в директории `/usr/share/fly-dm/sessions` и создать соответствующие конфигурационные файлы для `fly-wm`.

При входе через fly-dm выставляется переменная `DESKTOP_SESSION=имя_режима`, например: *fly*, *fly-desktop*, *fly-tablet*. Данная переменная является именем ярлыка сессии из каталога */usr/share/fly-dm/sessions* (но без расширения `.desktop`), которая указывает на тип сессии. Например: **DESKTOP_SESSION=fly** — десктопный; **DESKTOP_SESSION=fly-tablet** — планшетный.

Данное имя сессии добавляется как суффикс «.$DESKTOP_SESSION» к базовому имени конфигурационного файла и используется для выбора конфигурационных файлов менеджера окон fly-wm в соответствии с типом сессии. Если тип сессии десктопный, т.е. DESKTOP_SESSION=fly, то конфигурационные файлы остаются без суффикса для обратной совместимости.

Существуют следующие конфигурационные файлы в `/usr/share/fly-wm/` и соответствующие им файлы в `~./fly/`.

```
localadmin@astra:~$ cat ~/.fly/en.fly-wmrc
;update
[Variables]
;Dont edit this main file, edit files included below

ImagePath       =/usr/share/fly-wm/images:/usr/share/fly-wm/images:/usr/share/icons
SoundPath       =/usr/share/fly-wm/sounds:/usr/share/fly-wm/sounds
XmmPath         =/usr/share/fly-wm/keymaps:/usr/share/fly-wm/keymaps

include ~/.fly/paletterc
include ~/.fly/sessrc
include /usr/share/fly-wm/keyshortcutrc
include ~/.fly/keyshortcutrc
include ~/.fly/apprc
include ~/.fly/theme/current.themerc
include ~/.fly/en.miscrc
```

Файл `en.fly-wmrc` – главный файл, из которого вызываются остальные:

- `~/.fly/paletterc` – цветовая схема
- `~/.fly/sessr` – задание переменных, определяющих некоторые параметры сессий (использование менеджера сессий, диалоги при выключении компьютера, значения по умолчанию, новый вход и так далее)
- `/usr/share/fly-wm/keyshortcutrc` – глобальные настройки комбинаций клавиш (для всех пользователей)
- `~/.fly/keyshortcutrc` – индивидуальные настройки комбинаций клавиш
- `~/.fly/apprc` – настройка атрибутов для приложений
- `~/.fly/theme/current.themerc` – настройки текущей темы рабочего стола
- `~/.fly/en.miscrc` – конфигурация различных меню: Пуск, контекстные меню рабочего стола и его элементов (панель задач, область уведомлений и так далее)

Такие же файлы существуют и для других режимов работы, это:

- `~/.fly/en.fly-wmrc.fly-kiosk`
- `~/.fly/en.fly-mini`
- `~/.fly/en.fly-wmrc.fly-mobile`
- `~/.fly/en.fly-tablet`
- `~/.fly/en.fly-wmrc.fly-tablet-kiosk`

12.2.2.1 Вызов функций рабочего стола

Вызовом команды `fly-wmfunc <имя_функции> [<параметры>]` можно управлять элементами рабочего стола, например, чтобы свернуть все окна, необходимо выполнить команду `fly-wmfunc FLYWM_MINIMIZE_ALL`, а чтобы восстановить их – `fly-wmfunc FLYWM_RESTORE_ALL`.

```
localadmin@astra:~$ fly-wmfunc FLYWM_MINIMIZE_ALL
localadmin@astra:~$ fly-wmfunc FLYWM_RESTORE_ALL
```

По ссылке доступен список и описание функций fly-wm: Описание функций Fly-wm.

12.2.2.2 Принудительное обновление пользовательских конфигурационных файлов при входе пользователя

Для того чтобы при каждом входе пользователя пользовательская конфигурация автоматически перезаписывалась конфигурацией, заданной администратором системы, нужно указать параметр «;update» в первой строке файлов в каталоге /usr/share/fly-wm/:

```
;update
```

Файлы, в которых указан этот параметр, будут заменять соответствующие пользовательские файлы при входе пользователя в графическую сессию.

Подробнее о принудительном обновлении конфигурационных файлов при входе пользователя и их отдельных параметров вы можете почитать по ссылке: Управление пользовательскими конфигурациями оконного менеджера fly-wm.

12.3 Переменные окружения

Мы уже знаем, что такое переменные окружения (среды). Впервые они появились еще в Unix и с тех пор были включены во все современные операционные системы, включая Linux, MacOS, Windows. В отличие от Windows, в операционной системе Linux не только системные, но и многие пользовательские приложения используют переменные окружения в своей работе.

12.3.1 Уровни переменных окружения

Как мы теперь знаем, переменные могут быть определены на нескольких уровнях:

- **На уровне сеанса**

  Переменная определяется с помощью команды `export` и действует в рамках текущего сеанса. Например, вы можете «на лету» переопределить переменную PATH с помощью команды `export PATH="$PATH:~/.local/bin"`, но это значение не станет доступно новым экземплярам оболочки.

- **На уровне пользователя**

  В тех случаях, когда необходимо установить переменные для конкретного пользователя, их определяют в скриптах из его домашнего каталога, например, `~/.bashrc`. При запуске оболочки этим пользователем переменные будут устанавливаться автоматически в соответствии с порядком инициализации профиля.

- **На уровне системы**

Если же необходимо, чтобы переменные окружения устанавливались для всех пользователей компьютера, их нужно определить в файле `/etc/profile`. При запуске оболочки эти переменные будут устанавливаться автоматически в соответствии с порядком инициализации профиля.

12.3.2 Команды для работы с переменными окружения

12.3.2.1 Команда printenv

Команда `printenv` показывает все переменные окружения командной оболочки: и те, что были установлены в скриптах инициализации, и те, которые пользователь установил сам с помощью команды `export`:

```
localadmin@astra:~$ printenv
SHELL=/bin/bash
PWD=/home/localadmin
LOGNAME=localadmin
HOME=/home/localadmin
LANG=ru_RU.UTF-8
LS_COLORS=rs=0:di=01;34:ln=01;36....
pf=00;36:
TERM=xterm
USER=localadmin
SHLVL=3
PATH=/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games:/tmp/bin
MAIL=/var/mail/localadmin
_=/usr/bin/printenv
```

12.3.2.2 Команда set

Команда `set` выводит более подробный список переменных окружения, локальных переменных оболочки, а также различных функций, используемых командным интерпретатором. Вывод команды большой, поэтому выведем только первые 20 строк:

```
localadmin@astra:~$ set | head -n 20
BASH=/usr/bin/bash
BASHOPTS=checkwinsize:cmdhist:complete_fullquote:expand_
↪aliases:extglob:extquote:force_f
ignore:globasciiranges:histappend:interactive_comments:progcomp:promptvars:sourcepath
BASH_ALIASES=()
BASH_ARGC=([0]="0")
BASH_ARGV=()
BASH_CMDS=()
BASH_COMPLETION_VERSINFO=([0]="2" [1]="8")
BASH_LINENO=()
BASH_SOURCE=()
BASH_VERSINFO=([0]="5" [1]="0" [2]="3" [3]="1" [4]="release" [5]="x86_64-pc-linux-gnu
↪")
BASH_VERSION='5.0.3(1)-release'
```

(continues on next page)

(продолжение с предыдущей страницы)

```
COLUMNS=88
DIRSTACK=()
EUID=1000
GROUPS=()
HISTCONTROL=ignoreboth
HISTFILE=/home/localadmin/.bash_history
HISTFILESIZE=2000
HISTSIZE=1000
HOME=/home/localadmin
```

12.3.2.3 Команда export

Команду `export` мы уже использовали. Она устанавливает значение переменной окружения в рамках текущей сессии.

12.3.2.4 Команда env

Без параметров команда выводит информацию, аналогичную printenv, но основной смысл ее не в этом. Команда позволяет запускать приложения с измененным окружением без переопределения переменных в рамках текущей сессии. Например, использование Kerberos утилиты `kinit` на доменном компьютере:

Если выполнить установку переменной через env, то ее не видно в окружении

```
admin@dc-1:~$ env KRB5_TRACE=/dev/stdout kinit
[4435] 1727525373.569792: Getting initial credentials for admin@ALD.COMPANY.LAN
[4435] 1727525373.569794: Sending unauthenticated request
[4435] 1727525373.569795: Sending request (191 bytes) to ALD.COMPANY.LAN
[4435] 1727525373.569796: Initiating TCP connection to stream 10.0.1.11:88
[4435] 1727525373.569797: Sending TCP request to stream 10.0.1.11:88
[4435] 1727525373.569798: Received answer (514 bytes) from stream 10.0.1.11:88
[4435] 1727525373.569799: Terminating TCP connection to stream 10.0.1.11:88
[4435] 1727525373.569800: Response was from master KDC
...
Password for admin@ALD.COMPANY.LAN:
admin@dc-1:~$
```

Следующая команда выведет пустую строку:

```
admin@dc-1:~$ echo "$KRB5_TRACE"

admin@dc-1:~$
```

Если сделать то же самое через `export`, то переменная сохранится в текущей сессии, и трассировка станет выполняться по умолчанию для всех запросов `kinit`:

```
admin@dc-1:~$ export KRB5_TRACE=/dev/stdout
admin@dc-1:~$ echo "$KRB5_TRACE"
```

(continues on next page)

(продолжение с предыдущей страницы)

```
/dev/stdout
admin@dc-1:~$ kinit
[5093] 1727525617.029713: Getting initial credentials for admin@ALD.COMPANY.LAN
[5093] 1727525617.029715: Sending unauthenticated request
[5093] 1727525617.029716: Sending request (191 bytes) to ALD.COMPANY.LAN
[5093] 1727525617.029717: Initiating TCP connection to stream 10.0.1.11:88
[5093] 1727525617.029718: Sending TCP request to stream 10.0.1.11:88
...
Password for admin@ALD.COMPANY.LAN:
```

Еще можно переопределить язык запускаемой команды, например, прочитать английскую версию bash в manpages (пустая переменная означает локаль по умолчанию *en_ US.UTF-8*):

```
localadmin@astra:~$ env LANG= man bash
```

Но также можно задать локаль принудительно зная ее iso-код в файле `/etc/locale.gen` :

```
localadmin@astra:~$ locale
LANG=ru_RU.UTF-8
LANGUAGE=
LC_CTYPE="ru_RU.UTF-8"
LC_NUMERIC="ru_RU.UTF-8"
LC_TIME="ru_RU.UTF-8"
...
localadmin@astra:~$ env LANG=en_US.UTF-8 man bash
localadmin@astra:~$ env LANG=ru_RU.UTF-8 man bash
```

12.3.3 Примеры использования переменных окружения

12.3.3.1 Переменная PATH

Если необходимо добавить новую директорию, где буду храниться пользовательские скрипты и программы, потребуется изменить переменную PATH. Создадим директорию `scripts` в домашней папке пользователя и файл `myscript.sh` внутри нее. Для простоты `myscript.sh` выводит сообщение «Hello!». Для текущего сеанса достаточно выполнить команду `export` . После изменения PATH скрипт будет запускаться без указания полного пути к скрипту:

```
localadmin@astra:~$ mkdir ~/scripts
localadmin@astra:~$ cat <<EOF> ~/scripts/myscript.sh
> #!/usr/bin/env bash
> echo "Hello!"
> EOF
localadmin@astra:~$
localadmin@astra:~$ chmod +x ~/scripts/myscript.sh
localadmin@astra:~$ echo "$PATH"
/home/localadmin/.local/bin:/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games:/
↪home/localadmin/.local/bin
localadmin@astra:~$ myscript.sh
bash: ./myscript.sh: Нет такого файла или каталога
```

(continues on next page)

(продолжение с предыдущей страницы)

```
localadmin@astra:~$ export PATH="$PATH:~/scripts"
localadmin@astra:~$ echo "$PATH"
/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games:/home/localadmin/scripts
localadmin@astra:~$ myscript.sh
Hello!
```

После завершения текущей командной оболочки переменная PATH не сохранится. Чтобы сделать изменения постоянными, необходимо добавить новое значение PATH для пользователя в файлы инициализации `~/.bash_profile`, `~/.profile` или `~/.bashrc`. Добавим в файл `~/.bashrc`, так как при запуске оболочки из графического режима инструкции из `~/.bash_profile`, `~/.profile` не будут просмотрены командной оболочкой.

Откроем новую вкладку в терминале и выполним команды:

```
localadmin@astra:~$ echo "$PATH"
/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games
localadmin@astra:~$ echo 'export PATH="$PATH:~/scripts"' >> ~/.bashrc
localadmin@astra:~$ bash
localadmin@astra:~$ echo "$PATH"
/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games:/home/localadmin/scripts
localadmin@astra:~$ myscript.sh
Hello!
```

Если требуется изменить PATH для всех пользователей системы, тогда необходимо добавить изменения в глобальный файл инициализации `/etc/profile`. Чтобы скрипт `myscript.sh` стал общим, переименуем его в *script.sh* и положим в директорию */opt/scripts*. Для демонстрации примера создадим отдельную директорию и отредактируем `/etc/profile`. Так как директория `/usr/local/bin` включена в PATH всех пользователей, то достаточно было бы положить свои программы и скрипты в эту директорию без изменения `/etc/profile`. Изменения `/etc/profile` требуют права root. После внесения изменений перезайдем на машину под пользователем localadmin.

```
localadmin@astra:~$ sudo -i
[sudo] пароль для localadmin:
root@astra:~# mkdir /opt/scripts
root@astra:~# cp /home/localadmin/scripts/myscript.sh /opt/scripts/script.sh
root@astra:~# echo 'export PATH="$PATH:/opt/scripts"' >> /etc/profile
root@astra:~# su - localadmin
localadmin@astra:~$ script.sh
Hello!
localadmin@astra:~$ echo "$PATH"
/home/localadmin/.local/bin:/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games:/
↪opt/scripts:/home/localadmin/.local/bin:~/scripts
localadmin@astra:~$ whereis script.sh
script: /usr/bin/script /opt/scripts/script.sh /usr/share/man/man1/script.1.gz
```

Как мы видим *file:/opt/scripts* добавилось для всех пользователей в переменную PATH

12.3.3.2 Переменные истории команд

Переменные истории команд HISTCONTROL, HISTFILE, HISTSIZE, HISTFILESIZE, HISTIGNORE, HISTTIMEFORMAT.

HISTCONTROL – определяет, как команды сохраняются в истории команд. Принимает следующие значения (несколько значений могут быть указаны через двоеточие):

- **ignorespace** – если команда начинается с пробела, она не сохраняется в истории.
- **ignoredups** – если введенная команда содержится в истории, она не сохраняется.
- **ignoreboth** – включает в себя обе опции **ignorespace** и **ignoredups**.
- **erasedups** – если введенная команда содержится в истории, вся ее предыдущая история удаляется и команда сохраняется заново.

В Astra Linux значение по умолчанию `HISTCONTROL=ignoreboth`. Если значение не задано или указано неправильное значение, все введенные команды сохраняются как есть. Чтобы команды, начинающиеся с пробела, сохранялись в истории, необходимо поменять значение HISTCONTROL. После изменения команды с пробелом будут сохраняться. В примере команда `ping 88.77.8.8` будет сохранена.

```
localadmin@astra:~$ HISTCONTROL=ignoredups
localadmin@astra:~$ ping 88.77.8.8
PING 88.77.8.8 (88.77.8.8) 56(84) bytes of data.
64 bytes from 88.77.8.8: icmp_seq=1 ttl=54 time=85.7 ms
^C
--- 88.77.8.8 ping statistics ---
2 packets transmitted, 1 received, 50% packet loss, time 2ms
rtt min/avg/max/mdev = 85.723/85.723/85.723/0.000 ms
localadmin@astra:~$ history -a
localadmin@astra:~$ tail -2 .bash_history
ping 88.77.8.8
history -a
```

HISTFILE – определяет файл, в который записывается история. По умолчанию это `~/.bash_history`. Если задать другой файл, то надо иметь в виду, что только у пользователя должно быть право на чтение и запись этого файла.

```
localadmin@astra:~$ ls -l .bash_history
-rw------- 1 localadmin localadmin 43857 сен 15 21:37 .bash_history
```

HISTSIZE – определяет максимальное число строк, которое будет запомнено командной оболочкой. Если HISTSIZE равен 0, история команд не будет сохраняться в памяти оболочки. Если HISTSIZE меньше нуля, будут запоминаться все команды. Если число строк будет превышено, из файла будут удалены старые строки и добавлены новые так, что размер файла не превысит HISTSIZE.

HISTFILESIZE – определяет максимальный размер файла истории в строках. Если размер равен 0, из файла удаляются все строки. Если размер задан в виде нечисловых символов или равен числу меньше 0, размер файла не ограничен.

Чтобы сделать хранение истории неограниченным, можно присвоить пустые значения:

- HISTSIZE=
- HISTFILESIZE=

HISTIGNORE – определяет шаблоны команд, которые не попадут в память оболочки для сохранения в файле истории. Несколько шаблонов могут быть перечислены через двоеточие.

```
localadmin@astra:~/scripts$ HISTIGNORE="ls -l"
localadmin@astra:~/scripts$ ls -l
итого 4
. . .
localadmin@astra:~/scripts$ history -a
localadmin@astra:~/scripts$ tail -2 ~/.bash_history
HISTIGNORE="ls -l"
history -a
```

Но при этом команда `ls -la` попадет в историю.

```
localadmin@astra:~/scripts$ ls -la
итого 12
. . .
localadmin@astra:~/scripts$ history -a
localadmin@astra:~/scripts$ tail -2 ~/.bash_history
ls -la
history -a
```

Чтобы исключить все команды, начинающиеся с ls -l, можно добавить символ «звездочка» (*).

```
localadmin@astra:~/scripts$ HISTIGNORE="ls -l*"
localadmin@astra:~/scripts$ ls -la
итого 12
. . .
localadmin@astra:~/scripts$ history -a
localadmin@astra:~/scripts$ tail -2 ~/.bash_history
HISTIGNORE="ls -l*"
history -a
```

HISTIGNORE удобен для того, чтобы исключить из истории стандартные команды (ls, exit и другие) и команды, содержащие чувствительную информацию.

- **HISTTIMEFORMAT** – формат сохранения истории. По умолчанию в истории команд не показывается дата, так как не задана переменная HISTTIMEFORMAT. Для отображения даты необходимо определить её формат, например, HISTTIMEFORMAT=%H:%M:%S%d-%m-%Y

```
localadmin@astra:~$ cd scripts
```

(continues on next page)

(продолжение с предыдущей страницы)

```
localadmin@astra:~/scripts$ history | tail -n 2
1632  ls
1633  history | tail -n 2
localadmin@astra:~/scripts$ echo "$HISTTIMEFORMAT"

localadmin@astra:~/scripts$
```

Как мы видим формат имеет пустое значение, а теперь установим переменную для формата времени истории:

```
HISTTIMEFORMAT='%H:%M:%S %d-%m-%Y '
```

Проверим вывод команды истории `history`:

```
localadmin@astra:~/scripts$ history -a
localadmin@astra:~/scripts$ history | tail -n 3
1636  21:49:40 15-09-2023 ls
1637  21:49:51 15-09-2023 history -a
1638  21:49:54 15-09-2023 history | tail -n 3
```

Примечание: Все перечисленные переменные окружения могут быть определены на всех трех уровнях.

12.3.3.3 Переменная PROMPT_COMMAND

По умолчанию история команд сохраняется при нормальном завершении сеанса, например, командой `exit` или сочетанием клавиши Ctrl + D. Однако, при аварийном завершении команды не сохранятся. С помощью PROMPT_COMMAND история может быть сохранена немедленно.

```
localadmin@astra:~$ PROMPT_COMMAND="history -a;$PROMPT_COMMAND"
localadmin@astra:~$ echo "Hello"
Hello
localadmin@astra:~$ tail -2 .bash_history
PROMPT_COMMAND="history -a;$PROMPT_COMMAND"
echo "Hello"
localadmin@astra:~$ echo "Who am I"
Who am I
localadmin@astra:~$ tail -2 .bash_history
tail -2 .bash_history
echo "Who am I"
localadmin@astra:~$
```

Еще с помощью этой переменной можно, например, добавить дату в приглашении командной строки.

```
localadmin@astra:~$ PROMPT_COMMAND="echo -n [$(date +%H:%M:%Y)]"
[15:09:2024]localadmin@astra:~$
```

12.3.3.4 Переменные PS0, PS1, PS2, PS3, PS4

Переменные PS0, PS1, PS2, PS3, PS4 для настройки вида приглашения командной строки.

PS0 – задает значение, которое выводится перед выводом команды. По умолчанию не определено.

```
localadmin@astra:~$ PS0="Команда запущена пользователем \u на машине \h:\n\n"
localadmin@astra:~$ echo Hello
Команда запущена пользователем user на машине astra:

Hello
```

PS1 – определяет вид приглашения командной строки. По умолчанию для пользователей Astra Linux командная строка выделена разными цветами в виде строки `localadmin@astra:~$`.

```
localadmin@astra:~$ echo "$PS1"
\[\e]0;\u@\h: \w\a\]${debian_chroot:+($debian_chroot)}\u@\h:\w\$
```

- **`\[\e]0;\u@\h: \w\a\]` - задает название окна терминала (отображается в верхней части окна) в виде `localadmin@astra:~`, где:**
  - **\u** – имя пользователя,
  - **\h** – имя машины,
  - **\w** – текущий каталог. Символы **[** и **]** обозначают начало и конец последовательности непечатных символов.
  - **\e]0** – это управляющая ESCAPE последовательность, которая определяет действие для операционной системы, в данном случае поменять заголовок окна.
  - **\a** – обозначает ASCII символ звукового сигнала (bell).
- `${debian_chroot:+($debian_chroot)}` – Если задана переменная debian_chroot, добавляет ее к приглашению.
- **$** - выводит символ $.

Командная строка может быть выделена разными цветами, например, значение переменной PS1:

```
PS1="\[\e]0;\u@\h:\w\a\]${debian_chroot:+($debian\_chroot)}\[\033[01;91m\][\D{%H:%M:
↪%S %d.%m.%Y}] \[\033[01;32m\]\u@\h\[\033[00m\]:\[\033[01;34m\]\w\[\033[00m\]\$ "
```

- `\[\033[01;32m\]\u@\h` – выводит в определенном цвете логин пользователя и имя машины, разделенные символом @. Управляющая Esc последовательность [\033[01;32m\] указывает

операционной системе вывести следующий за ней текст в зеленом цвете.

- `[\033[00m\]:` – выводит символ : в белом цвете.
- `[\033[01;34m\]\w` – выводит текущую директорию в синем цвете.
- `\[\033[00m\]\$` – выводит символ $ в белом цвете.

Можно добавить отображение даты к текущему виду приглашения в виде `\[\033[01;91m\]\D{%H:%M:%S %d.%m.%Y}` , где `\D` – дата в формате, заданном в фигурной скобке. Дата будет окрашена в красный цвет.

```
localadmin@astra:~$ PS1="\[\e]0;\u@\h:\w\a\]${debian_chroot:+($debian\_chroot)}\[\
↪033[01;91m\][\D{%H:%M:%S %d.%m.%Y}] \[\033[01;32m\]\u@\h\[\033[00m\]:\[\033[01;34m\
↪]\w\[\033[00m\]\$ "
[22:04:29 15.09.2023] localadmin@astra:~\$
```

Примечание: Оттенков цветов очень много. Для примера здесь приведена таблица с цветами, а здесь – информация о специальных символах.

PS2 – устанавливает символ продолжения строки. Если строка не помещается на экране целиком, по умолчанию это символ `>`.

```
localadmin@astra:~$ echo "$PS2"
>
localadmin@astra:~$ PS2=">>>"
localadmin@astra:~$ ping 88.77.8.8 \
>>>-c 10 -l 1 \
>>>-W 10
```

PS3 – выводит строку приглашения при использовании в скриптах команды `select`. Если запустить скрипт сначала без установки PS3, приглашение выбора выглядит в виде #?.

```
localadmin@astra:~$ echo "$PS3"
```

Создадим такой скрипт `~/scripts/user_ps3.sh`:

```
#!/usr/bin/env bash

select i in пон вт ср exit
do
  case $i in
    пон) echo "Понедельник";;
    вт) echo "Вторник";;
    ср) echo "Среда";;
    exit) exit;;
  esac
done
```

Запустим его через bash:

```
localadmin@astra:~$ bash ~/scripts/user_ps3.sh
1) пон
2) вт
3) ср
4) exit
#? 1
Понедельник
```

Если определить PS3, вид приглашения изменится.

```
#!/usr/bin/env bash

PS3="Выберите день недели: "

select i in пон вт ср exit
do
  case $i in
    пон) echo "Понедельник";;
    вт) echo "Вторник";;
    ср) echo "Среда";;
    exit) exit;;
  esac
done
```

Запустим его повторно:

```
localadmin@astra:~$ bash ~/scripts/user_ps3.sh
1) пон
2) вт
3) ср
4) exit
Выберите день недели: 1
Понедельник
```

PS4 – определяет вид приглашения при включении режима отладки в скрипте. Переименуем и изменим немного предыдущий скрипт `user_ps3.sh`. Добавим ключ `-x`, который включит режим отладки (вывод дополнительной служебной информации) для поиска ошибок в скрипте. Также установим переменную PS4 равным `PS4="$0.$LINENO "`, где `$0` – имя скрипта, а `$LINENO` – номер строки.

```
localadmin@astra:~$ echo "$PS4"
+
```

Создадим еще один скрипт для проверки PS4 `~/scripts/user_ps4.sh`

```
#!/usr/bin/env bash

PS3="Выберите день недели: "
PS4="$0.$LINENO "
select i in пон вт ср exit
 do
```

(continues on next page)

(продолжение с предыдущей страницы)

```
  case $i in
    пон) echo "Понедельник";;
    вт) echo "Вторник";;
    ср) echo "Среда";;
    exit) exit;;
  esac
done
```

Проверим его работу через bash:

```
localadmin@astra:~$ bash -x ~/scripts/user_ps4.sh
+ PS3='Выберите день недели: '
+ PS4='scripts/user_ps4.sh.6 '
scripts/user_ps4.sh.6 select i in пон вт ср exit
1) пон
2) вт
3) ср
4) exit
Выберите день недели: 1
scripts/user_ps4.sh.6 case $i in
scripts/user_ps4.sh.6 echo Понедельник
Понедельник
Выберите день недели: 2
scripts/user_ps4.sh.6 case $i in
scripts/user_ps4.sh.6 echo Вторник
Вторник
```

К выводу скрипта добавилась дополнительная информация.

Мы рассмотрели некоторые примеры использования переменных окружения в командной оболочке bash, которая используется по умолчанию в Astra Linux. Документация по всем переменным доступна в `man bash` или `info bash`.

12.4 Практика и тестирование

12.4.1 Практическая работа: Модуль 12. Работа с профилями пользователей

См. также *Требования, правила и цели выполнения практической работы*

12.4.1.1 Практические задания

Задание 1.

1) Создайте директорию `tmp_files` в `/etc/skel`.

2) Убедитесь, что в домашнем каталоге нового созданного пользователя появилась папка `tmp_files`. Войдите в систему под новым пользователем, создайте файл в директории

tmp_files.

3) Добавьте команду очистки файлов и папок в директории tmp_files в .bash_logout. Сделайте вход в систему командой login и после выхода проверьте, что файлы в директории tmp_files исчезли.

Задание 2.

1) Создайте директорию my_programm в домашней папке пользователя.

2) Добавьте путь к директории my_programm в переменную окружения PATH. Убедитесь, что после выхода и входа в систему переменная PATH содержит путь к my_programm.

Задание 3.

1) Установите переменную окружения HISTTIMEFORMAT в виде «%H:%M:%S %d-%m-%Y» глобально.

2) Войдите под пользователем, введите несколько команд, убедитесь, что в выводе команды history появилась дата ввода команд.

3) Создайте файл .bash_aliases, добавьте псевдоним команды удаления файлов rm, но только с подтверждением перед удалением. Проверьте, что файлы не удаляются сразу, а сначала запрашивается разрешение пользователя.

12.4.1.2 Ответы на практические задания (пошаговые инструкции)

Задание 1.

1) Создайте директорию tmp_files в /etc/skel.Выполните:

Выполните команды:

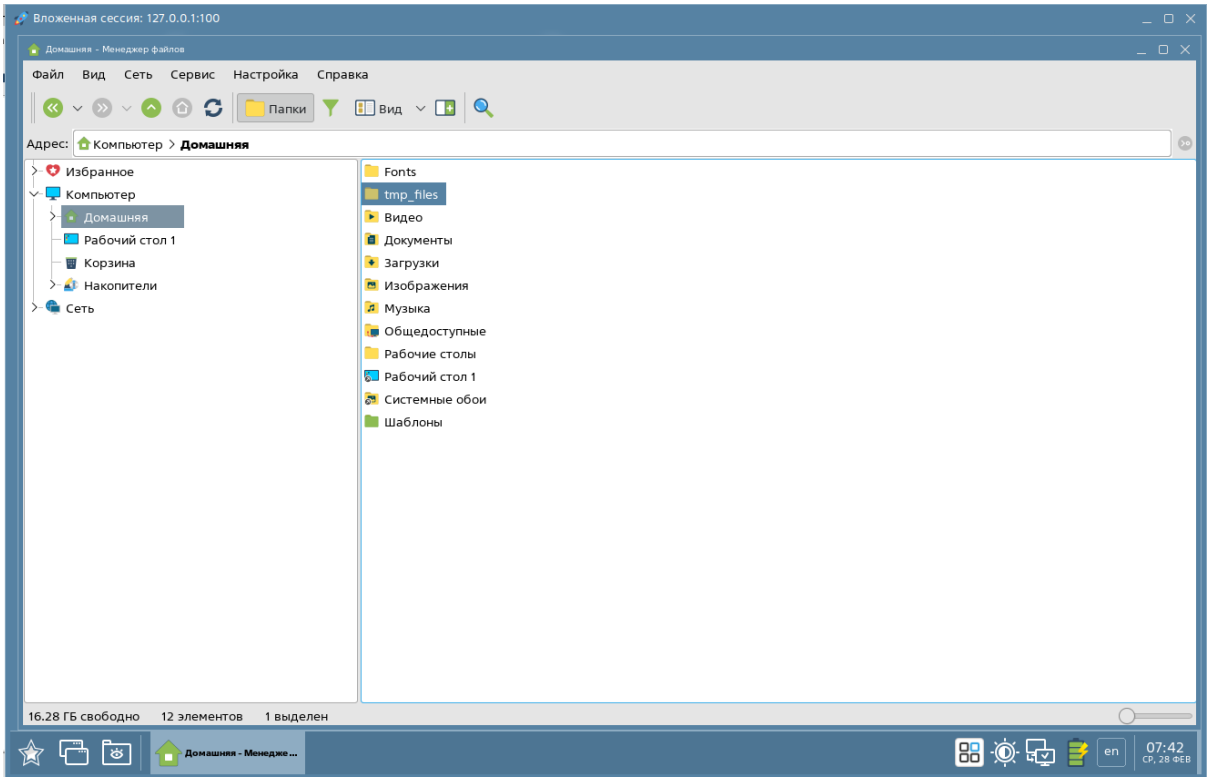
```
root@astra:~# mkdir /etc/skel/tmp_files
root@astra:~# useradd -m -s /bin/bash user0
root@astra:~# passwd user0
```

2) Убедитесь, что в домашнем каталоге нового созданного пользователя появилась папка tmp_files. Войдите в систему под новым пользователем, создайте файл в директории tmp_files.

```
root@astra:~# login user0
```

или ssh user0@ip, где ip — адрес тестовой машины. Вход также можно выполнить через GUI, см.

рисунок ниже.



Проверка того, что директория `tmp_files` существует:

```
user0@astra:~$ ls -al
user0@astra:~$ touch tmp_files/file
```

3) Добавьте команду очистки файлов и папок в директории `tmp_files` в `.bash_logout`. Сделайте вход в систему командой login и после выхода проверьте, что файлы в директории `tmp_files` исчезли.

   Добавьте строку в файл .bash_logout и выйдите из системы:

```
user0@astra:~$ echo "rm -rf ~/tmp_files/*" >> .bash_logout
user0@astra:~$ logout
```

4) Снова войдите в систему под user0 и запустите:

```
user0@astra:~$ ls -al tmp_files
```

Задание 2.

1) Создайте директорию `my_programm` в домашней папке пользователя.

Войдите в систему под пользователем user0 и выполните:

```
user0@astra:~$ mkdir -p ~/my_programm
```

2) Добавьте путь к директории `my_programm` в переменную окружения PATH. Убедитесь, что после выхода и входа в систему переменная PATH содержит путь к `my_programm`.

Выполните команды:

```
user0@astra:~$ echo "export PATH=$PATH:~/my_programm" >> .bashrc
user0@astra:~$ logout
```

Снова зайдите под пользователем user0 и убедитесь, что переменная окружения PATH содержит путь к директории `/home/user0/my_programm`

```
user0@astra:~$ echo $PATH
/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games:/home/user0/my_programm
```

Задание 3.

1) Установите переменную окружения HISTTIMEFORMAT в виде «%H:%M:%S %d-%m-%Y» глобально.

Выполните команды из-под администратора системы:

```
root@astra:~# echo "export HISTTIMEFORMAT='%H:%M:%S %d-%m-%Y'" >> /etc/profile
```

2) Войдите под пользователем, введите несколько команд, убедитесь, что в выводе команды history появилась дата ввода команд.

Войдите в систему под пользователем user0 и выполните несколько команд:

```
user0@astra:~$ ls
user0@astra:~$ ping 8.8.8.8
user0@astra:~$ cat /etc/hosts
```

`history -a` — команда для того, чтобы немедленно добавить текущую историю в файл `.bash_history`.

Введите history, чтобы проверить, что в истории команд отображаются даты:

```
user0@astra:~$ history | tail -5
```

3) Создайте файл `~/.bash_aliases`, добавьте псевдоним команды удаления файлов rm, но только с подтверждением перед удалением. Проверьте, что файлы не удаляются сразу, а сначала запрашивается разрешение пользователя.

Добавьте псевдоним команды rm созданный файл .bash_aliases

```
user0@astra:~$ echo "alias rm='rm -i'" > ~/.bash_aliases
```

Выйдите из системы и войдите снова. Проверьте, что псевдоним команды создался. Создайте файл file и удалите его. Вы должны увидеть, что команда rm запрашивает разрешение на удаление файла:

```
user0@astra:~$ alias
alias ls='ls --color=auto'
alias rm='rm -i'
user0@astra:~$ rm file
rm: удалить пустой обычный файл 'file'? y
```

12.4.2 Тест: Модуль 12. Работа с профилями пользователей

Пожалуйста, ответьте на несколько вопросов, чтобы закрепить полученные знания. Правильные ответы для самопроверки вы сможете найти в конце списка. Обратите внимание, что у некоторых вопросов может быть несколько верных вариантов ответов.

12.4.2.1 Вопросы

Вопрос 1. Какая командная оболочка используется в Astra Linux по умолчанию?

a. Оболочка `/etc/bash`

b. Оболочка `/etc/dash`

c. Оболочка `/etc/ksh`

d. Оболочка `/etc/zsh`

Вопрос 2. Как называется файл в Astra Linux, в котором определяются настройки окружения для конкретного пользователя?

a. Файл `~/.bash_profile`

b. Файл `~/.bash_login`

c. Файл `~/.profile`

d. Файл `/etc/profile`

Вопрос 3. В каком файле определяется переменная окружения `PATH` для всех пользователей?

a. Файл `/etc/environment`

b. Файл `/etc/profile`

c. Файл `/etc/bash.bashrc`

d. Файл `/etc/bashrc`

Вопрос 4. Какая директория хранит шаблоны файлов и папок для создаваемых пользователей?

a. Директория `/etc/astra`

b. Директория `/home`

c. Директория `/etc/systemd`

d. Директория `/etc/skel`

Вопрос 5. Если пользователь создает свою переменную окружения, в каком файле он может ее определить?

a. Файл `/etc/profile`

b. Файл `/etc/bash.bashrc`

c. Файл `.bash_logout`

d. Файл `.bashrc`

Вопрос 6. Что делает команда `printenv`?

a. Выводит значения всех переменных окружения или одного из них

b. Создает новую переменную окружения

c. Сохраняет в файле значения всех переменных окружения или одного из них

d. Выводит все переменные и функции bash

Вопрос 7. Какой файл определит настройки окружения пользователя, если в домашней директории пользователя лежат сразу четыре файла - .bash_profile, .bash_login, .profile, .bashrc?

a. Файл `.bash_login`

b. Файл `.bashrc`

c. Файл `.bash_profile`

d. Файл `.profile`

Вопрос 8. Какая команда позволяет создать переменную окружения в текущем сеансе пользователя?

a. Команда `MY_ENV=value`

b. Команда `echo "export MY_ENV=value" >> .bashrc`

c. Команда `export MY_ENV=value`

d. Команда `env MY_ENV=value bash`

Вопрос 9. Какая команда позволяет запустить программу, передав ей измененные значения переменных окружения?

a. Команда `env`

b. Команда `set`

c. Это невозможно, команда наследует все переменные окружения

d. Команда `run`

Вопрос 10. В каком отдельном файле можно определить псевдонимы команд для конкретного пользователя?

a. Файл `.bash_aliases`

b. Файл `.bash_profile`

c. Файл `.aliases`

d. Файл `/etc/bash_aliases`

Ответы на вопросы

1. a
2. c (так как по умолчанию в Astra Linux не существуют файлов .bash_profile и .bash_login)
3. b
4. d
5. d
6. a
7. c
8. c

9. a
10. a

12.5 Заключение

В этом модуле мы узнали, что такое профили пользователей, рассмотрели порядок их инициализации и узнали, что в этом процессе используются обычные bash-скрипты.

В следующем модуле мы узнаем, как работают процессы в Linux, познакомимся с их жизненным циклом и способами управления. Также мы подробнее посмотрим на псевдофайловую систему `/proc` и увидим, какую информацию из нее можно выудить. Отправляемся на рыбалку!

12.6 Дополнительные источники информации

- Описание функций Fly-wm;
- Управление пользовательскими конфигурациями оконного менеджера fly-wm.

12.7 Обратная связь

Если остались вопросы, то их всегда можно задать в специальной теме.

Модуль 13. Процессы, потоки и управление их жизненным циклом

13.1 Введение

Этот модуль расскажет об особенностях жизненного цикла процессов в Linux, об их рождении, жизни и смерти. Вы узнаете, что такое сигналы, для чего они нужны и как их использовать. Будет рассмотрена работа планировщика задач CFS и способы приоритизации процессов в Linux. Мы также научимся продвинутым приемам извлечения информации о процессах и управления ими.

13.2 Процессы и потоки в Linux

Основным назначением операционной системы, как мы знаем, является создание условий для работы прикладных приложений. Однако установленные в системе приложения — это всего лишь файлы, которые просто лежат на диске. Для того чтобы приложение «заработало», создается так называемый процесс.

При запуске программы операционная система выделяет экземпляру приложения независимое адресное пространство памяти, назначает уникальный идентификатор (Process Identifier, PID) и вносит запись в таблицу процессов. С этого момента планировщик задач (scheduler) начинает выделять процессу кванты процессорного времени, и программа оживает.

Процессы Linux и Windows во многом очень похожи, но есть и ряд отличий, понимание которых упростит работу с системой. Давайте разбираться.

13.2.1 Процессы в Linux и Windows

13.2.1.1 Дерево процессов в Linux

Приложение начинает работать, как только информация об экземпляре приложения заносится в таблицу процессов и планировщик начинает выделять ему кванты процессорного времени.

Для просмотра таблицы процессов в CMD мы использовали команду `tasklist`, в PowerShell командлет Get-Process, а в Linux для этого предназначена утилита *ps*. И начнем мы с наиболее часто используемых ключей **aux**:

```
localadmin@astra:~$ ps aux
USER       PID %CPU %MEM    VSZ   RSS TTY   STAT START  TIME COMMAND
root         1  0.0  0.2 103172 11500 ?     Ss   05:56  0:00 /sbin/init
root         2  0.0  0.0      0     0 ?     S    05:56  0:00 [kthreadd]
root         3  0.0  0.0      0     0 ?     I<   05:56  0:00 [rcu_gp]
...
```

Назначение использованных нами ключей следующее:

- **a** — отобразить процессы всех пользователей;
- **u** — показать, кто является владельцем процесса;
- **x** — показать процессы, у которых нет управляющего терминала, например, службы.

Внимание: Для обеспечения обратной совместимости Bash-скриптов утилита ps поддерживает ключи, используемые разными операционными системами: BSD (без дефисов), POSIX (с дефисами) и GNU (с двумя дефисами и длинными ключами). Поэтому короткие ключи, совпадая по написанию,

могут иметь совершенно разный смысл, например, команды `ps -a` и `ps a` покажут совсем не одно и то же.

Еще интереснее пример с командой `ps -aux`, когда перед группой ключей ставится символ дефиса. По правилам POSIX эта команда должна вывести все процессы для пользователя с именем «x», но т.к. такого пользователя не существует в системе, утилита понимает, что мы имели в виду «aux» без дефиса, и показывает нам этот результат.

Давайте с помощью ключей `-ef` выведем ту же информацию, но с идентификатором родителя и поговорим об иерархии процессов в Linux:

```
localadmin@astra:~$ ps -ef
UID          PID  PPID  C STIME TTY          TIME CMD
root           1     0  0 09:41 ?        00:00:01 /sbin/init
root           2     0  0 09:41 ?        00:00:00 [kthreadd]
root           3     2  0 09:41 ?        00:00:00 [rcu_gp]
root           4     2  0 09:41 ?        00:00:00 [rcu_par_gp]
root           5     2  0 09:41 ?        00:00:00 [slub_flushwq]
root           6     2  0 09:41 ?        00:00:00 [netns]
root           8     2  0 09:41 ?        00:00:00 [kworker/0:0H-events_highpri]
...
```

Назначение использованных нами ключей следующее:

- Ключ `-e` — отобразить процессы всех пользователей (то же самое, что ключ `-a`);
- Ключ `-f` — вывести больше доступной информации, например, идентификатор родителя.

```
localadmin@astra:~$ ps aux
UID          PID  PPID  C STIME TTY        TIME  CMD
root         1       0  0 20:03 ?      00:00:00  /sbin/init
root         2       0  0 20:03 ?      00:00:00  [kthreadd]
root         3       2  0 20:03 ?      00:00:00  [rcu_gp]
...
```

Существенным отличием Windows и Linux является строгая иерархия процессов. В системе Windows дочерние процессы, конечно же, сохраняют идентификатор родителя, но система не следит за целостностью этой информации, и через некоторое время это значение может соответствовать несуществующему или неправильному процессу. Так происходит, если родительский процесс был закрыт, и этот идентификатор был повторно выдан другому процессу.

В системе Linux, напротив, установлена жесткая иерархия, поэтому описанная выше ситуация невозможна. Иерархический подход позволяет при закрытии терминала отправить всем дочерним процессам сигнал SIGHUP, чтобы закрыть их автоматически. Если же процесс был запущен через команду `nohup` и нужно, чтобы он продолжил свою работу, то он станет «осиротевшим» и будет автоматически усыновлен процессом init (systemd) с PID=1.

Проведем эксперимент, который наглядно покажет разницу в поведении систем:

- Если запустить приложение calc.exe из терминала cmd или PowerShell, то после закрытия терминала калькулятор продолжит работу как ни в чем ни бывало, что является вполне ожидаемым поведением.
- Если же мы из терминала Gnome Terminator запустим калькулятор в фоновом режиме командой `kcalc &` и закроем окно терминала с помощью ALT + F4, а не через команду `exit`, то терминал закроется вместе с калькулятором. А это уже ни разу не ожидаемое поведение.
- А теперь повторим то же самое в fly-term от Astra Linux и увидим, что терминал можно закрывать хоть крестиком, хоть через `exit`, а калькулятор остается на месте, как будто его запустили через nohup, что намного привычнее для Windows-администраторов. Ой, ну, что вы... что вы... даже не благодарите.

Используя значение PPID, мы можем легко найти все процессы, запущенные из текущей оболочки. При этом мы, конечно, можем поупражняться в составлении трехэтажных регулярных выражений, но проще будет воспользоваться специальной утилитой `pgrep` и системной переменной **$$**, в которой содержится идентификатор текущего процесса. Команда в этом случае будет выглядеть так:

```
localadmin@astra:~$ ps -f -p$(pgrep --parent=$$)
UID        PID  PPID  C STIME TTY      STAT   TIME  CMD
localad+  2322  1395  0 00:06 pts/0    Sl     0:00  kcalc
localad+  2328  1395  0 00:06 pts/0    Sl     0:00  kcalc
localad+  2334  1395  0 00:06 pts/0    Sl     0:00  kcalc
```

Возвращаясь к дереву процессов, обратим ваше внимание на то, что дерево, на самом деле, не одно, см. рис. 1.49. Чтобы убедиться в этом на практике, воспользуемся утилитой `pstree` и выведем список всех потомков процесса с PID=0, которые были порождены ядром системы:

```
localadmin@astra:~$ pstree -p 0
?()---kthreadd(2)---acpi_thermal_pm(96)
     |              |-ata_sff(84)
     |              |-audit_prune_tre(367)
     |              |-blkcg_punt_bio(82)
...
     |              |-writeback(30)
     |              `-zswap-shrink(125)
     `-systemd(1)---NetworkManager(474)---{NetworkManager}(502)
                    |                    `-{NetworkManager}(503)
                    |-VBoxClient(1232)---VBoxClient(1233)---{VBoxClient}(1235)
                    |                                      `-{VBoxClient}(1236)
                    |-VBoxClient(1244)---VBoxClient(1245)---{VBoxClient}(1247)
                    |                                      `-{VBoxClient}(1248)
                    |-VBoxClient(1251)---VBoxClient(1253)---{VBoxClient}(1255)
                    |                                      |-{VBoxClient}(1256)
                    |                                      `-{VBoxClient}(1257)
...
```

Процесс systemd или `/sbin/init` является процессом подсистемы инициализации и управления службами systemd, которая пришла на замену устаревшей подсистеме init. Новая реализация обеспечила распараллеливание запуска служб, что позволило существенно ускорить загрузку ОС. В настоящий момент systemd полностью вытеснила init SysV, но файл `/sbin/init` используется до сих пор и является символьной ссылкой на `/lib/systemd/systemd`.

Процесс [kthreadd] является диспетчером потоков, а квадратные скобки указывают нам на то, что это процесс самого ядра ОС. Второе дерево потребовалось, потому что для управления потоками нельзя было использовать процесс systemd, который работает в пользовательском пространстве.

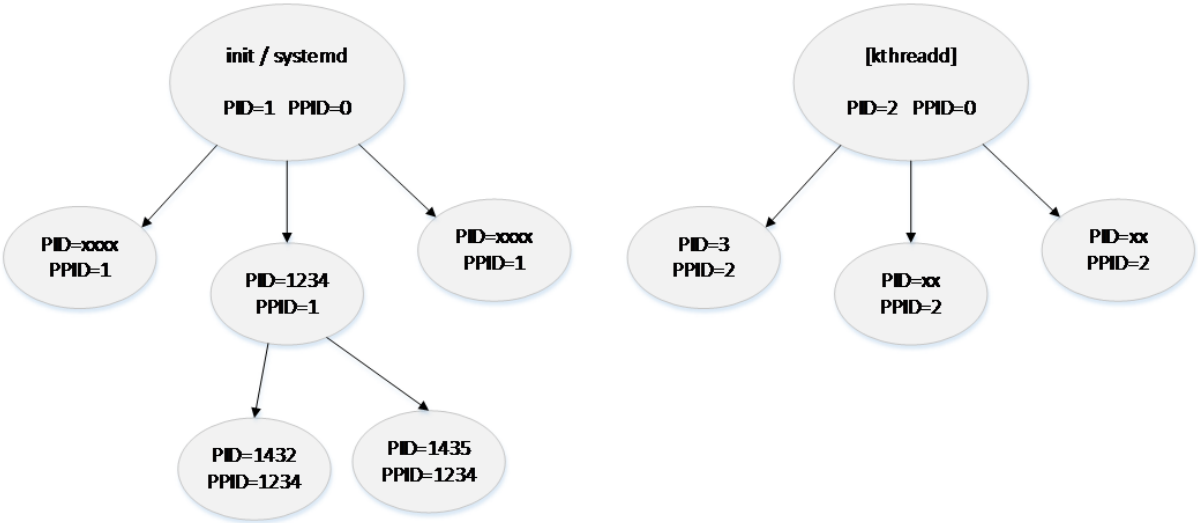


рис. 1.49 – Деревья процессов в Linux

13.2.1.2 Адресное пространство

Как уже было сказано ранее, каждому процессу выделяется независимое адресное пространство памяти. Максимальный размер пространства в 64-битных системах в теории может доходить до 2^{64}B, а на практике для x86 процессоров не превышает 2^{48}B, т.е. до 256 ТБ (см. вывод команды `cat /proc/cpuinfo` «address sizes ... 48 bits virtual»). Но это все равно чуть больше, чем дохарда. Хотя, кто его знает... Биллу Гейтсу до сих пор припоминают его «640 КБ должно хватить всем».

Важно понимать, что размер адресного пространства и объем памяти, потребляемый процессом, — это две большие разницы. Все адресное пространство разбивается на страницы объемом в 4Кб (см. `getconf PAGE_SIZE`), которых у процесса может быть, как у дурака фантиков. Однако большая часть этих страниц остается пустой, то есть не сопоставлена с реальной памятью компьютера, см. рис. 1.50:, поэтому процесс занимает ровно столько памяти, сколько он запросил, или как еще говорят «аллоцировал» (*от англ. allocation – резервировать*).

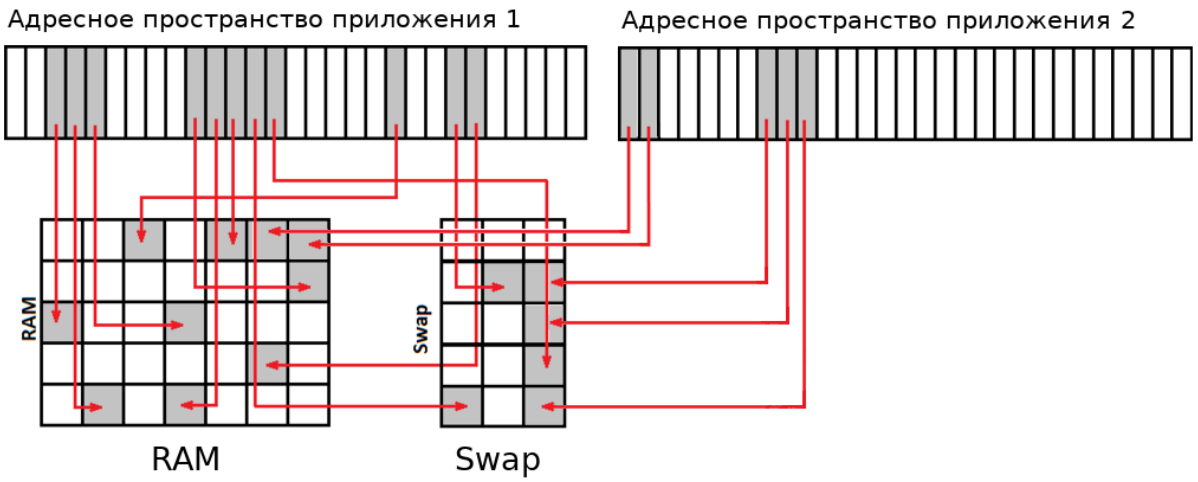


рис. 1.50 – Отображение страниц адресного пространства приложений на ОЗУ и файл подкачки

Обычно процессы могут аллоцировать весь доступный объем памяти, и ничего настраивать дополнительно не требуется. Выполните команду `ulimit -a`, и вы увидите, что для «virtual memory» установлено значение «unlimited»:

```
localadmin@astra:~$ ulimit -a
core file size          (blocks, -c) 0
data seg size           (kbytes, -d) unlimited
scheduling priority             (-e) 0
file size               (blocks, -f) unlimited
pending signals                 (-i) 15317
max locked memory       (kbytes, -l) 65536
max memory size         (kbytes, -m) unlimited
open files                      (-n) 1024
pipe size            (512 bytes, -p) 8
POSIX message queues     (bytes, -q) 819200
real-time priority              (-r) 0
stack size              (kbytes, -s) 8192
cpu time               (seconds, -t) unlimited
max user processes              (-u) 15317
virtual memory          (kbytes, -v) unlimited
file locks                      (-x) unlimited
```

Однако бывают ситуации, когда мы сталкиваемся, например, с утечкой памяти, и, чтобы дожить до патча, устраняющего проблему, приходится подкручивать настройки. Например, с помощью команды `ulimit -v 512000` мы можем ограничить потребление памяти процессами «на лету», а с помощью строки «* - as 512000» в конфигурационном файле `/etc/security/limits.conf` выставить эти ограничения на постоянной основе.

Добавим еще, что из соображений безопасности адресные пространства процессов независимы друг от друга, поэтому один процесс не может напрямую получить доступ к переменным и структурам данных другого процесса. В тех случаях, когда это действительно требуется, процессы должны обмениваться информацией через механизмы межпроцессного взаимодействия (*от англ. Inter-Process*

Communication, IPC).

Как вы помните, в Linux основными механизмами IPC являются файлы, каналы (pipe, fifo), сокеты (Unix и TCP/IP), сигналы, разделяемая память и др. В Windows есть аналогичные технологии каналов, TCP/IP сокетов, разделяемой памяти, но при организации межпроцессного взаимодействия в большей степени используются программные интерфейсы, такие как объектная модель компонентов (*от англ. Component Object Model, COM*) или удаленный вызов процедур (*от англ. Remote Procedure Call, RPC*). В том числе по этой причине Linux работает пошустрее.

13.2.1.3 Порождение процесса

В системе Linux новый процесс создается вызовом ядра fork(), во время выполнения которого происходит полное копирование адресного пространства родительского процесса, назначается собственный PID/PPID и обнуляется статистика выполнения в таблицах ОС. В ряде случаев этого бывает достаточно, но если требуется запустить другой программный код (например, выполнить ps из командной строки bash), то сразу после fork() следует еще один системный вызов execve(), см. рис. 1.51.

Существует целое семейство вызовов exec* (см. справку `man 3 exec`), которые отличаются набором параметров и логикой работы. Например, вызов execl() имеет дело со списком, а execv() с вектором. Но все они решают одну и ту же задачу, просто немного по-разному.

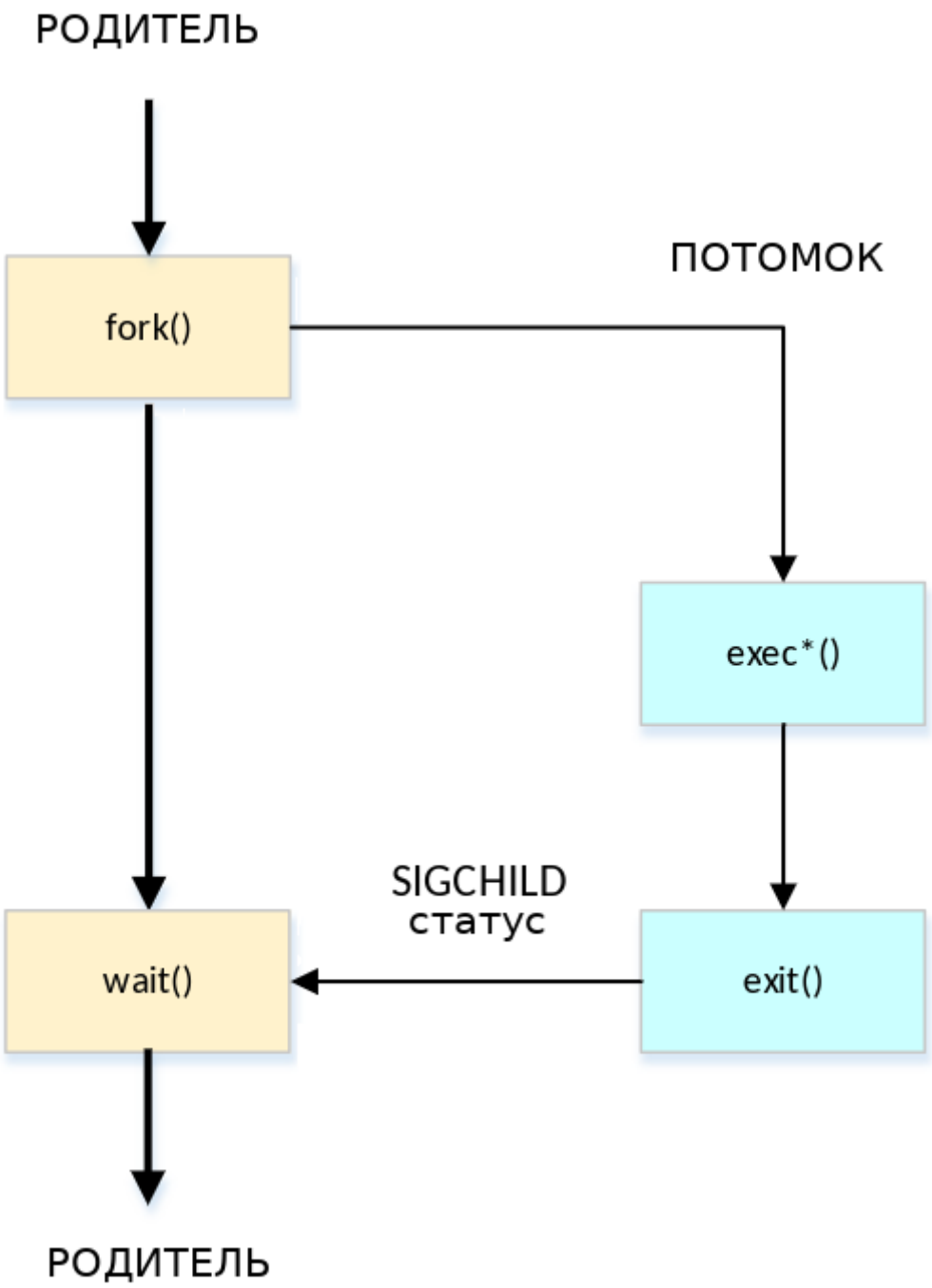


рис. 1.51 – Схема ветвления процессов

Важно понимать, что копирование процесса происходит чрезвычайно быстро, так как копируются не сами данные, а только список страниц памяти, где эти данные хранятся. На все эти страницы устанавливается отметка Copy-on-Write (COW), поэтому, как только одному из процессов требуется внести изменения, для него создается копия соответствующей страницы. В итоге fork() оказывается значительно быстрее, чем функция CreateProcess() из Win32 API, хотя на современных процессорах это уже не столь важно.

Понятное дело, что системные администраторы не всегда обладают навыками программирования, и подробности о вызовах fork() и execve() могут показаться вам излишними. Но мы очень часто сталкиваемся с упоминаниями этих функций в журналах отладки и трассировки, поэтому общие представления о работе этого механизма крайне полезны.

13.2.1.4 Завершение процесса

Процесс может завершить свою работу штатно, когда программный код полностью выполнил поставленную перед ним задачу, из-за непредвиденной ошибки или аварийно по требованию администратора, например, командой `kill -9 $(pidof kcalc)`.

Вне зависимости от того, как завершился процесс, он освобождает все свои ресурсы и становится пустой записью в таблице процессов, в которой остается только статус завершения, необходимый родительскому процессу для корректной обработки результатов. Для того чтобы родитель поскорее вычитал статус, ему направляется сигнал SIGCHLD, но до тех пор, пока это не произойдет, запись будет оставаться в таблице, поэтому такие процессы называются «зомби» — процесс уже «умер», но еще не «погребён» и продолжает занимать идентификатор.

Проблема с неупокоенными процессами заключается в том, что количество идентификаторов PID в дистрибутивах Linux всегда ограничено. К примеру, в Astra Linux их по умолчанию может быть не больше 4 194 304 (см. вывод команды `cat /proc/sys/kernel/pid_max`). Поэтому, когда такие зомби восстают регулярно и в большом количестве, наступает зомби-апокалипсис, и система перестает запускать новые процессы.

Зависшие зомби-процессы могут появляться как по причине зависания родительского процесса, так и в связи с ошибками в программном коде, но отправлять SIGKILL зомби-процессам совершенно бесполезно, ибо то, что мертво, умереть не может. Чтобы очистить систему от зомби, можно еще раз вручную направить сигнал SIGCHLD родительскому процессу с помощью команды `kill -s SIGCHLD <PPID>`. Если же родительский процесс завис и по-прежнему отказывается хоронить зомби, следующим шагом будет удаление родительского процесса. И не забываем про «семь бед – один reset» — после перезагрузки ни один зомби точно не выживет.

13.2.1.5 Состояния процесса во время его существования

Жизненный цикл процесса предусматривает 5 основных состояний, см. рис. 1.52. Посмотреть текущее состояние можно с помощью команды `ps aux` в колонке STAT:

- **R** - готов (ready) или выполняется (running).
- **D** - непрерываемое ожидание (uninterruptible sleep) - ожидание определенного события, чаще всего ожидания операции ввода-вывода.

- **S** - прерываемое ожидание (interruptible sleep) - ожидание неопределенного события или сигнала.
- **T** - остановка - процесс приостановлен, например, отладчиком.
- **Z** - зомби (zombie) - завершен, но сигнал SIGCHLD не получен родителем.

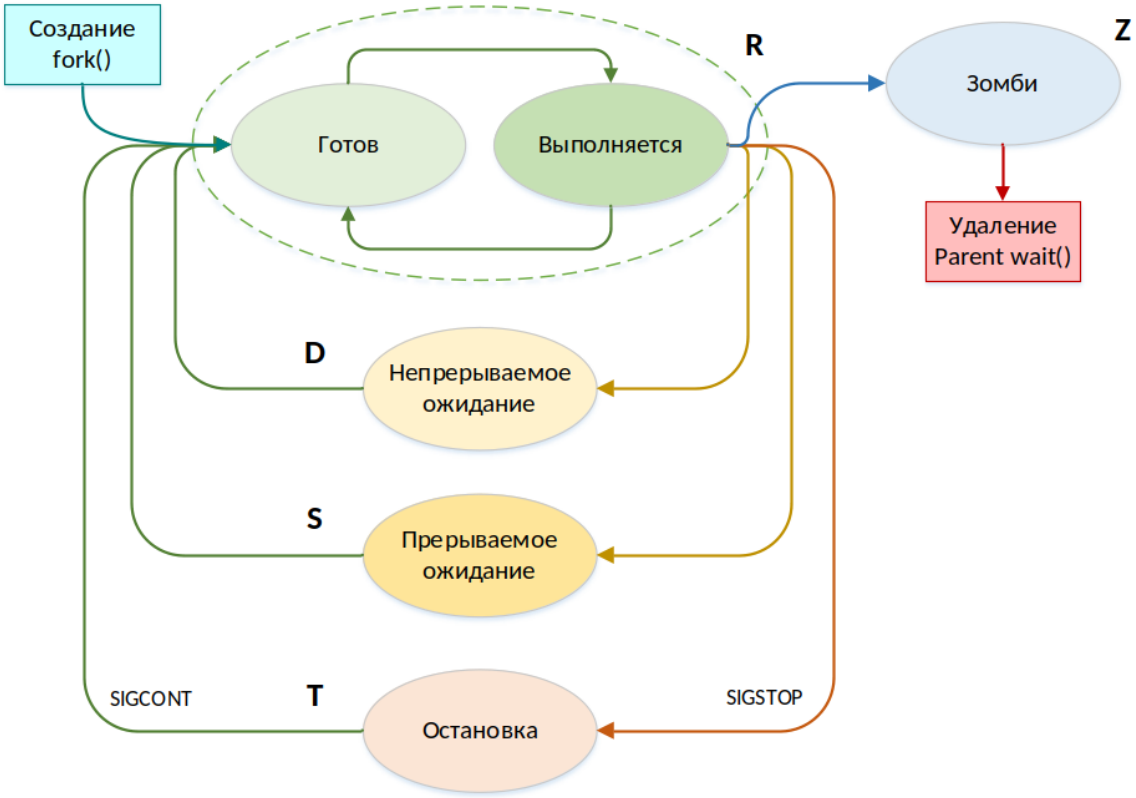


рис. 1.52 – Жизненный цикл процесса

Статусы **готов** и **выполняется** обозначаются одной буквой R, т.к. процесс получает кванты процессорного времени по сто раз в секунду, и эта информация не несет для пользователя никакой полезной информации.

Если процессу требуется доступ к устройству ввода-вывода (например, прочитать данные из файла), то без получения этих данных дальнейшее выполнение процесса будет невозможным, и процесс перейдет в состояние **непрерываемого ожидания (D)**. Это ожидание называется непрерываемым, так как даже команда `kill -9 pid` в таком состоянии не более, чем предложение «свернуть удочки», которое процесс может полностью проигнорировать.

Существует также **прерываемое ожидание (S)**. Это состояние наступает, когда процесс завершил выполнение задач и ожидает следующих запросов. Например, вы открыли калькулятор, и окно приложения ожидает ваших указаний. Прерываемым это ожидание называется потому, что про-

цесс в таком состоянии может быть завершен (убит) администратором. Это не всегда безопасно, особенно применительно к базам данных, хотя и возможно.

Отправкой сигнала SIGSTOP процесс может быть переведен в состояние **остановлен (T)**, в котором он замирает до получения сигнала разморозки SIGCONT. Вы можете использовать это состояние для приостановки процесса, который начал потреблять слишком много ресурсов, чтобы спокойно проанализировать причины, не завершая работу приложения сигналом SIGKILL.

Пятое состояние **зомби (Z)** мы уже рассмотрели подробно выше. Оно возникает, когда процесс завершил работу, но родитель еще не получил его статус. Если такой процесс зависнет в системе вы можете повторно отправить SIGCHLD родителю, завершить работу родителя или перезагрузить систему

13.2.2 Потоки в Linux

Кроме вызова fork() существует также системный вызов clone(), который позволяет создавать процессы с разделяемым адресным пространством или так называемые потоки (англ. thread).

Потоки позволяют организовать параллельную обработку данных на нескольких ядрах вычислительного узла, что сильно повышает итоговую скорость выполнения задачи. Например, если у вас 4 ядра плюс гипертрейдинг, то вы можете архивировать данные в 8 потоков, что будет в 8 раз быстрее.

В зависимости от используемых флагов потоки могут обладать разными возможностями:

- **CLONE_VM** — общее адресное пространство (любой из потоков может поменять страницы в памяти, и эти изменения будут видны всем потокам);
- **CLONE_FS** — общие сведения о файловой системе (корневой и текущий каталоги, umask, права на создаваемые файлы и каталоги по умолчанию);
- **CLONE_FILES** — общая таблица открытых файлов;
- **CLONE_SIGHAND** — общая таблица обработчиков сигналов;
- **CLONE_PARENT** — позволяет создавать сестринские процессы с тем же PPID, как у себя.

С точки зрения Linux управление процессами и потоками – это очень похожие задачи, поэтому и для работы с ними используются одни и те же инструменты. Например, команда `ps -T -p <pid>` покажет вам все потоки указанного процесса.

13.3 Сигналы для процессов в Linux

Мы уже знаем о том, что сигналы предназначены для межпроцессного взаимодействия. Этот механизм напоминает системные прерывания, поэтому сигналы иногда называют еще мягкими или программными прерываниями. Однако, если системные прерывания обрабатываются ядром, то обработчик сигналов находится внутри приложения.

Для отправки сигналов из командной строки предназначена утилита `kill` (*от англ. убить*). Название утилиты объясняется тем, что без дополнительных уточнений она отправляет сигнал SIGTERM, который как раз и означает завершение работы. Для того чтобы отправить какой-то другой сигнал, нужно указать цифровой код сигнала -<signal> или его имя с помощью ключа `-s` (`--signal`):

Запускаем калькулятор в фоновом режиме:

```
localadmin@astra:~$ kcalc &
[1] 5403
```

Завершаем процесс с выбранным PID 5403 из прошлой команды:

```
localadmin@astra:~$ kill -SIGTERM 5403
localadmin@astra:~$
```

Альтернативные варианты написания команды:

- `kill –TERM 5403`
- `kill -15 5403`
- `kill 5403`

Процессы Linux поддерживают 64 сигнала, список которых можно посмотреть с помощью ключа `-L` (`-l`, `--list`) команды `kill`:

```
localadmin@astra:~$ kill -l
 1) SIGHUP       2) SIGINT       3) SIGQUIT      4) SIGILL       5) SIGTRAP
 2) SIGABRT      7) SIGBUS       8) SIGFPE       9) SIGKILL     10) SIGUSR1
1)  SIGSEGV     12) SIGUSR2     13) SIGPIPE     14) SIGALRM     15) SIGTERM
2)  SIGSTKFLT   17) SIGCHLD     18) SIGCONT     19) SIGSTOP     20) SIGTSTP
3)  SIGTTIN     22) SIGTTOU     23) SIGURG      24) SIGXCPU     25) SIGXFSZ
4)  SIGVTALRM   27) SIGPROF     28) SIGWINCH    29) SIGIO       30) SIGPWR
5)  SIGSYS      34) SIGRTMIN    35) SIGRTMIN+1  36) SIGRTMIN+2  37) SIGRTMIN+3
6)  SIGRTMIN+4  39) SIGRTMIN+5  40) SIGRTMIN+6  41) SIGRTMIN+7  42) SIGRTMIN+8
7)  SIGRTMIN+9  44) SIGRTMIN+10 45) SIGRTMIN+11 46) SIGRTMIN+12 47) SIGRTMIN+13
8)  SIGRTMIN+14 49) SIGRTMIN+15 50) SIGRTMAX-14 51) SIGRTMAX-13 52) SIGRTMAX-12
9)  SIGRTMAX-11 54) SIGRTMAX-10 55) SIGRTMAX-9  56) SIGRTMAX-8  57) SIGRTMAX-7
10) SIGRTMAX-6  59) SIGRTMAX-5  60) SIGRTMAX-4  61) SIGRTMAX-3  62) SIGRTMAX-2
11) SIGRTMAX-1  64) SIGRTMAX
```

Чаще всего администраторы используют в работе сигналы:

- **Сигнал SIGINT(2)** предназначен для прерывания работы приложений по требованию пользователя (от англ. *interrupt* — *прервать*) и отправляется автоматически активному процессу терминала по нажатию клавиш Ctrl + C.

  Чаще всего администраторы отправляют этот сигнал приложениям через горячие клавиши, но можно сделать это и через утилиту kill. Продвинутые приложения могут реализовывать более сложную логику обработки этого сигнала, чем просто прерывание работы.

  Например, Postgres запускает быстрое выключение: в этом случае сервер запрещает новые подключения и посылает всем процессам сигнал SIGTERM, в результате чего их транзакции прерываются, и сами процессы завершаются. Управляющий процесс ждёт, пока будут завершены все эти процессы и затем завершается сам.

- **Сигнал SIGKILL(9)** требует процесс немедленно завершить свою работу. Процесс не имеет права проигнорировать этот сигнал и не может переопределить его обработчик, поэтому, если процесс не завершился по сигналу -9, то это указывает на какие-то проблемы в работе операционной системы. По своей сути сигнал SIGKILL похож на команду End task диспетчера задач Windows.

- **Сигнал SIGTERM(15)** отправляется процессу по умолчанию, если команде kill передать только идентификатор процесса. Сигнал просит процесс приступить к штатному завершению работы (от англ. *software termination* — завершение работы программного обеспечения).

  Если обработчик сигнала реализован в приложении правильно, то при получении SIGTERM процесс выполнит «умное выключение», т.е. обработает все несохраненные изменения, закроет файлы, сохранит состояние и вернет код 0, соответствующий успешному завершению.

  При выключении операционной системы процессы сначала получают сигнал SIGTERM и только по таймауту завершаются принудительно.

Продвинутые администраторы расширяют набор своих инструментов сигналами:

- **Сигнал SIGHUP(1)** посылается всем дочерним процессам терминала при его закрытии, чтобы они тоже завершили свою работу (от англ. *hang up* — повесить трубку). Название сигнала берет свое начало со времен аппаратных терминалов и телефонных линий связи.

  Для того чтобы приложение игнорировало сигнал -1, его можно запустить с помощью команды nohup. Если закрыть терминал, такие процессы «осиротеют» и будут удочерены процессом init (systemd).

```
localadmin@astra:~$ kcalc &
[3] 20613
```

Приложение завершает свою работу:

```
localadmin@astra:~$ kill -SIGHUP 20613
localadmin@astra:~$ nohup kcalc &
```

Запускаем команду через `nohup`:

```
localadmin@astra:~$ nohup kcalc &
[4] 20991
[3] Обрыв терминальной линии kcalc
```

Завершить процесс с помощью сигнала HUP больше не удается, он игнорируется приложением:

```
localadmin@astra:~$ kill -SIGHUP 20991
localadmin@astra:~$
```

- **Сигнал SIGCHLD(17)** направляется родителю при завершении дочернего процесса. При появлении зомби-процессов сигнал может быть направлен вручную для обработки очереди.
- **Сигнал SIGSTOP(19)** переводит процесс в состояние «остановлен», что удобно для отладки проблем с высоким потреблением ресурсов, когда требуется вернуть систему к жизни, не завершая проблемный процесс полностью. Процесс не имеет права проигнорировать этот сигнал и не может переопределить его обработчик. Для возобновления работы процесса предназначен сигнал SIGCONT.
- **Сигнал SIGTSTP(20)** предназначен для остановки работы приложений из терминала (от англ. *TTY stop*) и отправляется автоматически активному процессу терминала по нажатию клавиш Ctrl + Z.
- **Сигнал SIGCONT(18)** возобновляет работу процесса, который был остановлен с помощью сигналов SIGSTOP и SIGTSTP.

Каждый сигнал является указателем на область памяти, где находится обработчик, и многие из этих обработчиков могут быть переопределены внутри приложения через функцию trap(). Например, служба SSSD, обеспечивающая работу компьютера в домене, принимает сигналы SIGUSR2 и SIGUSR1 для того, чтобы переключиться в режим онлайн/офлайн соответственно:

```
admin@dc-1:~$ sudo kill -SIGUSR1 "$(pidof sssd)"
admin@dc-1:~$ sudo sssctl domain-status ald.company.lan
Online status: Offline

Active servers:
IPA: dc-1.ald.company.lan

Discovered IPA servers:
- dc-1.ald.company.lan

admin@dc-1:~$ sudo kill -SIGUSR2 "$(pidof sssd)"
admin@dc-1:~$ sudo sssctl domain-status ald.company.lan
```

(continues on next page)

(продолжение с предыдущей страницы)

```
Online status: Online

Active servers:
IPA: dc-1.ald.company.lan

Discovered IPA servers:
- dc-1.ald.company.lan
```

13.4 Планировщик задач в Linux и управление приоритетами процессов

13.4.1 Планировщик задач

Планировщик задач – это часть ядра ОС, необходимая для распределения ресурсов процессора между процессами. Планировщик преследует несколько целей:

- Максимизировать пропускную способность, то есть количество задач, выполняемых за единицу времени.
- Минимизировать время ожидания, то есть время, прошедшее с момента готовности процесса до начала его выполнения.
- Минимизировать время ответа, то есть время, прошедшее с момента готовности процесса до завершения его выполнения.
- Максимизировать равнодоступность, то есть справедливое распределение ресурсов между задачами.

Для достижения этих целей и обеспечения стабильности работы системы и приоритетного выполнения важных системных процессов используются системы приоритизации и распределения ресурсов по политикам планирования (группам процессов).

13.4.2 Группы процессов

В Linux существуют три основные группы процессов: First In, First Out (FIFO), Round Robin (RR) и Other. Первые две представляют собой процессы реального времени, которые должны выполняться немедленно, без задержек, а последняя, Other - условные процессы, при выполнении которых задержки не являются критичными для работы системы.

Таким образом, в системе существуют три очереди процессов, и логика выделения процессорного времени (логика выбора приоритета процесса) в каждой из них своя.

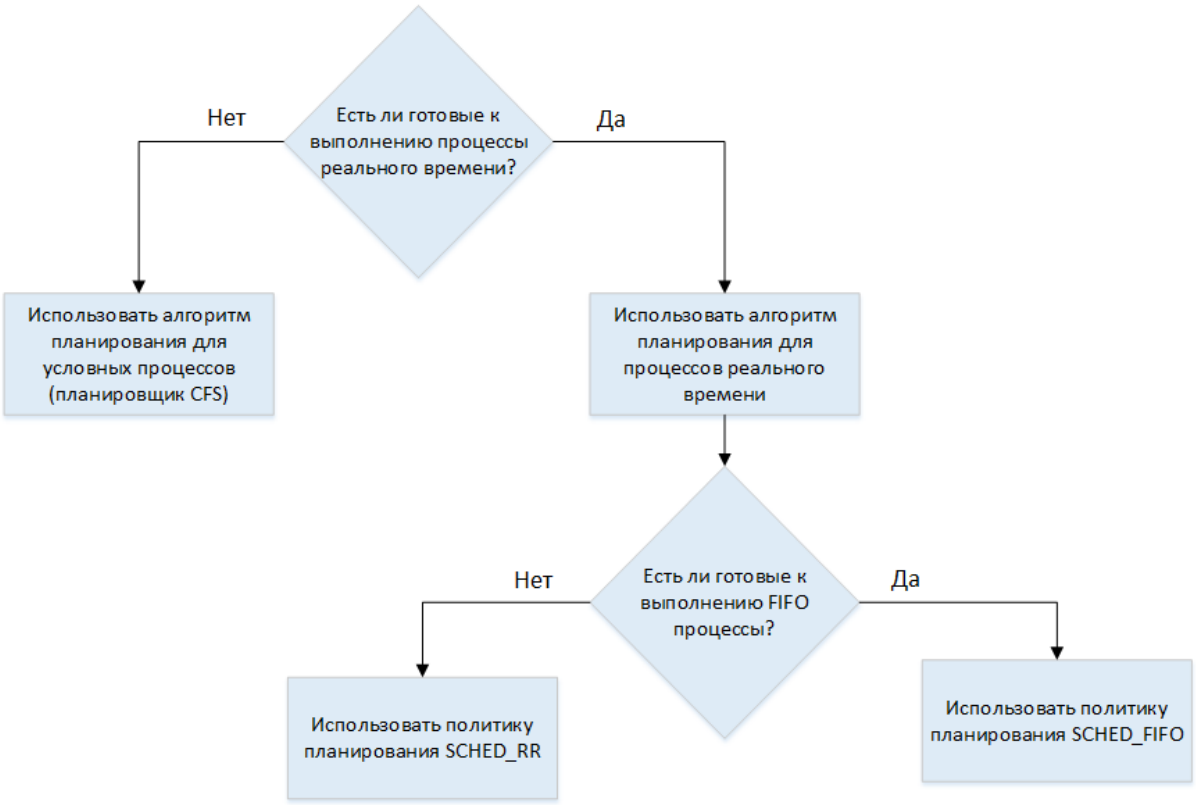


рис. 1.53 – Логика выбора алгоритма и политики планирования процессов

FIFO процессы (*first in, first out, первый вошел, первый вышел*) имеют самый высокий приоритет из всех. Такой процесс фактически работает в однопрограммном режиме и не реагирует ни на какие сигналы. Использование таких процессов несет серьезные риски, так как ошибка в его выполнении может критически повлиять на всю систему в целом.

В Linux используется всего несколько FIFO потоков ядра, для выполнения которых критически важно, чтобы их операции не прерывались и завершались без сбоев, например, изменение статуса процесса диспетчером процессов. В противном случае мы могли бы получить некорректную таблицу состояний процессов.

Процессы **Round Robin** (*по круговой схеме*) имеют более низкий приоритет по сравнению с FIFO процессами, поэтому при активизации FIFO процесса будут им вытеснены. Процессы RR равны между собой по приоритету в пределах одной очереди. Пока существует несколько процессов Round Robin. Каждому из них диспетчер выделяет определенный квант времени, и они будут выполняться по кругу, пока все такие процессы не будут завершены или не перейдут в состояния ожидания или остановки.

Процессы **Other** (все остальные) имеют самый низкий приоритет, то есть могут выполняться, только если нет активных FIFO или Round Robin процессов. В современных дистрибутивах Linux этими процессами занимается планировщик CFS (*Completely Fair Scheduler, абсолютно справедливый планировщик*), который старается соблюдать принцип справедливости, чтобы каждый процесс по-

лучил одинаковое количество процессорного времени. Кванты времени вычисляются так, чтобы исключить слишком частые переключения между процессами.

Отличительной особенностью планировщика CFS является высокая скорость сортировки процессов по потреблению ими процессорного времени за время log(N), так как для сортировки используется красно-черное дерево. Благодаря этому удалось добиться действительно справедливого распределения процессорного времени между процессами при высокой скорости самого планирования.

13.4.3 Политики планирования

Группы процессов FIFO, RR и Other соответствуют политикам планирования SCHED_FIFO, SCHED_RR и SCHED_OTHER (всего таких политик 6). Посмотреть список политик планирования можно командой `chrt -m`.

```
localadmin@astra:~$ chrt -m
SCHED_OTHER min/max priority    : 0/0
SCHED_FIFO min/max priority     : 1/99
SCHED_RR min/max priority       : 1/99
SCHED_BATCH min/max priority    : 0/0
SCHED_IDLE min/max priority     : 0/0
SCHED_DEADLINE min/max priority : 0/0
```

Как видим, выбор приоритета (а именно приоритета реального времени) доступен только в политиках SCHED_FIFO и SCHED_RR - от 1 до 99, для остальных политик он всегда 0.

13.4.4 Приоритет процессов

Каждому процессу присваивается приоритет. Это целое число от 39 до -100, где 39 - наименьший приоритет, а -100 - наивысший. Процессы с приоритетами от 39 до 0 - это обычные процессы из группы Other. Их приоритет не может быть меньше 0. По умолчанию все процессы из политики Other имеют приоритет 20, однако это значение может быть скорректировано как при запуске процесса (команда `nice`), так и во время его работы (команда `renice`). Эти команды меняют параметр NICE процессов - его «вежливость» (от англ. niceness), которая по умолчанию равна 0.



рис. 1.54 – Соотношение параметров приоритета процесса, его «вежливости» и приоритета реального времени

Параметр NICE может принимать значение от +19 до -20. При этом значение приоритета по умолчанию (20) суммируется со значением NICE, и, как упоминалось ранее, итоговый результат может принимать значение от 39 до 0.

Процессы с приоритетом от -1 до -100 - это процессы реального времени. У таких процессов кроме обычного приоритета существует приоритет реального времени, который используется в политиках планирования FIFO и RR. Для преобразования приоритета реального времени (rt_prior) в обычный приоритет процесса необходимо из -1 вычесть rt_prior. Например, при rt_prior равному 50, обычный приоритет будет равен -51 (-1 - 50 = -51).

Узнать текущее значение приоритета и параметра NICE можно в выводе команды `htop` - колонки PRI и NI.

13.4.5 Задания, интерактивный и фоновый режимы работы процесса

Оболочка Bash позволяет останавливать работу процессов, переводить их в интерактивный и фоновый режим по ходу работы, для чего предназначено несколько команд.

Остановка процессов. Когда мы запускаем утилиты как обычно, например, `kcalc`, они работают в интерактивном режиме, поэтому блокируют работу с терминалом. Если нам нужно вернуться к работе в терминале, мы можем закрыть приложение полностью клавишами Ctrl + C или временно приостановить его работу сочетанием Ctrl + Z. При этом калькулятор останется на экране, но перестанет реагировать на наши действия.

Перевод процесса в интерактивный режим. Когда мы захотим продолжить работу с калькулятором, нам нужно будет вернуть его обратно в интерактивный режим командой `fg`. В этом случае калькулятор снова начнет реагировать на наши действия, но терминал опять окажется заблокирован.

Команда `fg <задание>` переводит указанное задание в интерактивный режим (от англ. foreground — передний план). Задание может быть указано следующим образом:

- **<N>** — порядковый номер задания, см. команду jobs далее.
- **%шаблон** — поиск задания по его имени, например, %ping или %nmap.
- **%+** и **%-** — следующая и предыдущая задача.
- **%%** — последняя задача. То же самое, что не указывать номер задания.

Перевод процесса в фоновый режим. Если же мы хотим продолжить работу с калькулятором, но не хотим терять доступ к терминалу, мы можем перевести приложение не в интерактивный, а в фоновый режим командой `bg`. В этом случае нам будут доступны и калькулятор, и терминал одновременно, как будто калькулятор изначально был запущен командой `kcalc &` с амперсандом на конце.

Команда `bg <задание>` переводит указанное задание в фоновый режим (от англ. background — задний план). Задание задается таким же образом, как в команде `fg`, поэтому не будем повторяться.

Просмотр фоновых заданий выполняется командой `jobs`. У этой команды есть следующие ключи:

- Ключ `-l` – выводит список процессов заданий с PID процесса и информацией по умолчанию.
- Ключ `-n` – выводит только те процессы, статус которых был изменен с последнего оповещения.
- Ключ `-p` – выводит список только PID фоновых процессов.
- Ключ `-r` – выводит только запущенные фоновые процессы.
- Ключ `-s` – выводит только остановленные фоновые процессы.

Давайте запустим три фоновых процесса и посмотрим список заданий:

```
localadmin@astra:~$ kcalc &
[1] 2514
localadmin@astra:~$ kcalc &
[2] 2523
localadmin@astra:~$ kcalc &
[3] 2531
```

(continues on next page)

(продолжение с предыдущей страницы)

```
localadmin@astra:~$ jobs
[1]   Запущен          kcalc &
[2]-  Запущен          kcalc &
[3]+  Запущен          kcalc &
```

13.5 Извлечение информации о процессах

Информация о процессах доступна через псевдофайловую систему `/proc`. Это универсальный интерфейс для доступа к данным о процессах и работе ядра ОС. Кроме того, используя определенные файлы, можно не только получать информацию о процессах, но и управлять некоторыми настройками ядра ОС. Утилиты для работы с процессами, такие как `ps`, `top` или `htop`, в качестве источника информации используют как раз эту псевдофайловую систему.

Для пользователя взаимодействие с файлами `/proc` выглядит как работа с обычными текстовыми файлами, но это совсем не так. Например, при чтении какого-либо файла командой `cat` происходит системный вызов, и система формирует текстовый ответ, который передается команде.

Часть файлов предоставляет человекочитаемый формат данных, содержащих, например, пары ключ-значение, а часть – машиночитаемые, используемые при работе утилит, которые удобно использовать, например, в скриптах автоматизации.

Вначале мы разберем, что хранится в каталогах `/proc/PID/`, где PID - числовой идентификатор процесса, например, `/proc/1/`, а затем посмотрим содержимое самого каталога `/proc/`.

13.5.1 Содержимое /proc/PID/

```
localadmin@astra:$ sudo ls /proc/1
arch_status          fd          numa_maps      smaps_rollup
attr                 fdinfo      oom_adj        stack
autogroup            gid_map     oom_score      stat
auxv                 io          oom_score_adj  statm
cgroup               limits      pagemap        status
clear_refs           loginuid    patch_state    syscall
cmdline              map_files   personality    task
comm                 maps        projid_map     timens_offsets
coredump_filter      mem         root           timers
cpu_resctrl_groups   mountinfo   sched          timerslack_ns
cpuset               mounts      schedstat      uid_map
cwd                  mountstats  sessionid      wchan
environ              net         setgroups
exe                  ns          smaps
```

Файл `/proc/cmdline` – в нем содержится строка запуска процесса.

```
localadmin@astra:~$ cat /proc/1/cmdline && echo
/sbin/init
```

Файл `/proc/exe` – символическая ссылка, ведущая к полному пути до исполняемого файла. Полный путь необходим для однозначного понимания, какой именно файл был запущен.

```
localadmin@astra:~$ sudo ls -l --color=always /proc/1/exe
lrwxrwxrwx 1 root root 0 июл 18 11:32 exe -> /usr/lib/systemd/systemd
```

Файл `/proc/cwd` – текущий рабочий каталог процесса PID1.

```
localadmin@astra:$ sudo ls -l --color=always /proc/1/cwd
lrwxrwxrwx 1 root root 0 июл 27 13:06 cwd -> /
```

Файл `/proc/environ` – окружение процесса, создающее контекст его выполнения.

```
localadmin@astra:~$ sudo cat /proc/1/environ && echo
SHLVL=1HOME=/init=/sbin/initTERM=linuxBOOT_IMAGE=/boot/vmlinuz-5.15.0-33-genericdrop_
↪caps=PATH=/sbin:/usr/sbin:/bin:/usr/binPWD=/rootmnt=/root
```

Каталог `/proc/fd/` – дескрипторы открытых файлов (*от англ. file descriptors*). Каждый файл в `fd/` представляет собой символическую ссылку и имеет числовое имя. Это не номер айноды в файловой системе, это уникальный номер в пределах процесса, который позволяет процессу обращаться к конкретным файлам, с которыми он работает. Дескрипторы 0, 1 и 2 - это стандартные ввод, вывод и вывод ошибок.

```
localadmin@astra:~$ sudo ls -l /proc/1/fd --color=always
итого 0
lrwx------ 1 root root 64 июл 18 11:32 0 -> /dev/null
lrwx------ 1 root root 64 июл 18 11:32 1 -> /dev/null
...
lrwx------ 1 root root 64 авг  4 09:03 109 -> 'socket:[20856]'
lr-x------ 1 root root 64 авг  4 09:03 11 -> anon_inode:inotify
...
lr-x------ 1 root root 64 авг  4 09:03 119 -> 'pipe:[16902]'
lr-x------ 1 root root 64 июл 18 11:32 12 -> /dev/autofs
lrwx------ 1 root root 64 авг  4 09:03 120 -> 'socket:[16920]'
lrwx------ 1 root root 64 авг  4 09:03 121 -> /run/initctl
lrwx------ 1 root root 64 авг  4 09:03 122 -> 'socket:[16904]'
lrwx------ 1 root root 64 авг  4 09:03 123 -> 'socket:[16906]'
lr-x------ 1 root root 64 авг  4 09:03 13 -> /proc/1/mountinfo
lr-x------ 1 root root 64 авг  4 09:03 14 -> anon_inode:inotify
lr-x------ 1 root root 64 авг  4 09:03 15 -> /proc/swaps
...
lrwx------ 1 root root 64 июл 18 11:32 2 -> /dev/null
lrwx------ 1 root root 64 июл 18 11:32 20 -> 'socket:[27113]'
...
```

Файл `/proc/io` – содержит сведения об объемах данных, прочитанных и записанных процессом в хранилище информации.

```
localadmin@astra:~$ sudo cat /proc/1/io
rchar: 11296824165
wchar: 159986595
syscr: 6481108
syscw: 467703
read_bytes: 801298432
write_bytes: 258985984
cancelled_write_bytes: 43184128
```

Файл /proc/limits – отображает различные ограничения процесса, установленные конфигурационным файлом /etc/security/limits.conf и командой ulimit.

```
localadmin@astra:~$ sudo cat /proc/1/limits
Limit                     Soft Limit           Hard Limit           Units
Max cpu time              unlimited            unlimited            seconds
Max file size             unlimited            unlimited            bytes
Max data size             unlimited            unlimited            bytes
Max stack size            8388608              unlimited            bytes
Max core file size        0                    unlimited            bytes
Max resident set          unlimited            unlimited            bytes
Max processes             7523                 7523                 processes
Max open files            1048576              1048576              files
Max locked memory         67108864             67108864             bytes
Max address space         unlimited            unlimited            bytes
Max file locks            unlimited            unlimited            locks
Max pending signals       7523                 7523                 signals
Max msgqueue size         819200               819200               bytes
Max nice priority         0                    0
Max realtime priority     0                    0
Max realtime timeout      unlimited            unlimited            us
```

Файл /proc/maps – физические адреса страниц памяти, используемые в данный момент.

```
localadmin@astra:~$ sudo cat /proc/1/maps |  head
5e9ee4066000-5e9ee4094000 r--p 00000000 08:01 657629                     /usr/lib/
↪systemd/systemd
5e9ee4094000-5e9ee41af000 r-xp 0002e000 08:01 657629                     /usr/lib/
↪systemd/systemd
5e9ee41af000-5e9ee4204000 r--p 00149000 08:01 657629                     /usr/lib/
↪systemd/system
...
```

Файл /proc/pagemap для каждой виртуальной страницы определяет её фактическое положение: в физической памяти или в файле подкачки. Информация хранится в двоичном коде.

Файл /proc/sched –отображение текущих значений переменных планировщика процессов, необходимых для корректной работы планировщика SFC.

```
localadmin@astra:~$ sudo cat /proc/1/sched
systemd (1, #threads: 1)
se.exec_start                                :    1290085360.092074
se.vruntime                                  :         11829.729715
se.sum_exec_runtime                          :         21150.129581
```

(continues on next page)

(продолжение с предыдущей страницы)

```
se.nr_migrations                         :                 2199
nr_switches                              :                39002
nr_voluntary_switches                    :                29552
nr_involuntary_switches                  :                 9450
se.load.weight                           :              1048576
se.avg.load_sum                          :                  285
se.avg.runnable_sum                      :               291840
se.avg.util_sum                          :               271360
se.avg.load_avg                          :                    0
se.avg.runnable_avg                      :                    0
se.avg.util_avg                          :                    0
se.avg.last_update_time                  :      1290085360091136
se.avg.util_est.ewma                     :                    7
se.avg.util_est.enqueued                 :                    0
uclamp.min                               :                    0
uclamp.max                               :                 1024
effective uclamp.min                     :                    0
effective uclamp.max                     :                 1024
policy                                   :                    0
prio                                     :                  120
clock-delta                              :                   95
mm->numa_scan_seq                        :                    0
numa_pages_migrated                      :                    0
numa_preferred_nid                       :                   -1
total_numa_faults                        :                    0
current_node=0, numa_group_id=0
numa_faults node=0 task_private=0 task_shared=0 group_private=0 group_shared=0
```

Файлы `/proc/stat` и `/proc/status` – близкие по своей сути, отражают основные сведения о процессе. Файл **stat** - машиночитаемый формат, а **status** - человекочитаемый. Большинство параметров, выводимых утилитой `ps`, она получает из структур данных, аналогичных тем, которые отображаются в этих файлах.

```
localadmin@astra:~$ sudo cat /proc/1/stat
1 (systemd) S 0 1 1 0 -1 4194560 111979 3580285 170 48914 1384 733 32339 27124 20 0 1␣
↪0 11
1056563202873 18446744073709551615 104036523655168 104036524814201 140723869122320 0␣
↪0 0
671173123 4096 12601 0 0 17 0 0 0 0 0 0 104036525166288 104036525396288␣
↪104036533329920
140723869130571 140723869130582 140723869130582 140723869130733 0
localadmin@astra:/proc/1$ sudo cat status
Name:   systemd
Umask:  0000
State:  S (sleeping)
Tgid:   1
Ngid:   0
Pid:    1
PPid:   0
TracerPid:      0
Uid:    0       0       0       0
Gid:    0       0       0       0
FDSize: 128
Groups:
```

(continues on next page)

(продолжение с предыдущей страницы)

```
NStgid: 1
NSpid:  1
NSpgid: 1
NSsid:  1
VmPeak:   168716 kB
VmSize:   103180 kB
...
Mems_allowed_list:      0
voluntary_ctxt_switches:        29574
nonvoluntary_ctxt_switches:     9459
```

Файл **statm** - статистика по использованию памяти. Это 7 чисел, из которых только 5 имеют значения в современных версиях ядра. Все значения - количество страниц памяти:

- **1** – общий размер памяти, занимаемой программой (то же, что и VmSize в файле status)
- **2** – размер резидентной памяти (то же, что и VmRSS в файле status)
- **3** – разделяемые страницы памяти (то же, что и RssFile+RssShmem в файле status)
- **4** – размер сегмента, отведенного под код
- **5** – не используется (всегда 0, начиная с ядра 2.6)
- **6** – размер сегмента данных и стека
- **7** – не используется (всегда 0, начиная с ядра 2.6)

```
localadmin@astra:~$ cat /proc/1/statm
25795 2873 2179 283 0 4927 0
```

13.5.2 Содержимое /proc

Ниже представлено содержимое каталога `/proc/`. В выводе команды все каталоги с PID заменены на {PID 1,2,...,N} для лучшего восприятия. Файлы, хранящиеся в корне каталога `/proc/`, характеризуют состояние и характеристики всего вычислительного узла и ядра ОС.

```
localadmin@astra:~$ ls /proc
{PID 1,2, ... N}
asound          irq           schedstat      bootconfig     kallsyms      scsi
buddyinfo       kcore         self           bus            keys          slabinfo
cgroups         key-users     softirqs       cmdline        kmsg          stat
consoles        kpagecgroup   swaps          cpuinfo        kpagecount    sys
crypto          kpageflags    sysrq-trigger  devices        loadavg       sysvipc
diskstats       locks         thread-self    dma            mdstat        timer_list
driver          meminfo       tty            dynamic_debug  misc          uptime
execdomains     modules       version        fb             mounts        version_
↪signature
filesystems     mtrr          vmallocinfo    fs             net           vmstat
```

(continues on next page)

(продолжение с предыдущей страницы)

```
interrupts      pagetypeinfo zoneinfo       iomem          partitions
ioports         pressure
```

Файл /proc/cmdline – список параметров, которые были переданы ядру при загрузке.

```
localadmin@astra:~$ cat /proc/cmdline
BOOT_IMAGE=/boot/vmlinuz-5.15.0-33-generic root=UUID=1864fbac-4e75-42ec-9a52-
↪c0944d4fd6b9 ro parsec.mac=0 quiet net.ifnames=0
```

Файл /proc/cpuinfo – сведения о всех установленных процессорах.

```
localadmin@astra:~$ cat /proc/cpuinfo
processor       : 0
vendor_id       : GenuineIntel
cpu family      : 6
model           : 79
model name      : Intel(R) Xeon(R) CPU E5-2680 v4 @ 2.40GHz
stepping        : 1
cpu MHz         : 2394.456
cache size      : 35840 KB
physical id     : 0
siblings        : 2
core id         : 0
cpu cores       : 2
...
```

Файл /proc/diskstats – статистика операций со всеми дисками. Для каждого диска своя строка. Описание полей доступно по ссылке.

```
localadmin@astra:~$ cat /proc/diskstats
 7   0 loop0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 . . .
 8   1 sda1 44186 11778 2356130 13854 240755 305673 4556040 132537 0 336236 146391 0␣
↪0 0 0 0 0
 8   2 sda2 2 0 4 0 0 0 0 0 8 0 0 0 0 0 0 0
 8   5 sda5 69 0 5000 22 0 0 0 0 0 68 22 0 0 0 0 0 0
11   0 sr0 33 0 160 11 0 0 0 0 0 44 11 0 0 0 0 0 0
```

Файл /proc/meminfo – отображение информации о состоянии памяти. Предоставляет больше параметров, чем утилита free.

```
localadmin@astra:~$ free
               total        used        free      shared  buff/cache   available
Mem:         4007624      448304     1761592       27816     1797728     3284820
Swap:         998396           0      998396

localadmin@astra:~$ cat /proc/meminfo
MemTotal:        2025208 kB
MemFree:           88064 kB
MemAvailable:    1295928 kB
Buffers:          181228 kB
Cached:          1101408 kB
SwapCached:            0 kB
```

(continues on next page)

(продолжение с предыдущей страницы)

```
Active:          1024936 kB
Inactive:         596484 kB
...
```

Файл `/proc/devices` – перечень устройств в системе.

```
localadmin@astra:~$ cat /proc/devices
Character devices:
  1 mem
  4 /dev/vc/0
  4 tty
  4 ttyS
  5 /dev/tty
...
Block devices:
  2 fd
  7 loop
  8 sd
  9 md
 11 sr
 65 sd
 66 sd
 67 sd
...
```

Файл `/proc/filesystem` – перечень файловых систем, поддерживаемых ядром ОС.

```
localadmin@astra:~$ cat /proc/filesystems
nodev   sysfs
nodev   tmpfs
nodev   bdev
nodev   proc
...
        ext3
        ext2
        ext4
...
```

Файл `/proc/mounts` – перечень смонтированных файловых систем (формат данных, аналогичен файлу /etc/fstab).

```
localadmin@astra:~$ cat /proc/mounts
sysfs /sys sysfs rw,nosuid,nodev,noexec,relatime 0 0
proc /proc proc rw,nosuid,nodev,noexec,relatime 0 0
udev /dev devtmpfs rw,nosuid,relatime,size=963080k,nr_inodes=240770,mode=755,inode64␣
↪0 0
devpts /dev/pts devpts rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000 0 0
tmpfs /run tmpfs rw,nosuid,noexec,relatime,size=202524k,mode=755,inode64 0 0
/dev/sda1 / ext4 rw,relatime,errors=remount-ro 0 0
parsecfs /parsecfs parsecfs rw,sync,relatime 0 0
securityfs /sys/kernel/security securityfs
rw,nosuid,nodev,noexec,relatime 0 0
tmpfs /dev/shm tmpfs rw,nosuid,nodev,inode64 0 0
```

Файл `/proc/modules` – список подгруженных модулей ядра. Ядро Linux монолитное, но модульное, и мы можем управлять тем, какие модули подгружать, а какие нет, тем самым определяя функционал запускаемой ОС.

```
localadmin@astra:~$ cat /proc/modules
binfmt_misc 24576 1 - Live 0x0000000000000000 (E)
vboxvideo 36864 0 - Live 0x0000000000000000 (OE)
drm_ttm_helper 16384 1 vboxvideo, Live 0x0000000000000000 (E)
intel_rapl_msr 20480 0 - Live 0x0000000000000000 (E)
intel_rapl_common 32768 1 intel_rapl_msr, Live 0x0000000000000000 (E)
```

Файл `/proc/swaps` – список разделов подкачки.

```
localadmin@astra:~$ cat /proc/swaps
Filename                Type            Size        Used        Priority
/dev/sda5               partition       998396      0           -2
```

Файл `/proc/version` – версия ядра ОС.

```
localadmin@astra:~$ cat /proc/version
Linux version 5.15.0-33-generic (builder@build5) (gcc (AstraLinuxSE 8.3.0-6) 8.3.0,␣
↪GNU ld (GNU Bin utils for AstraLinux) 2.31.1) #astra2+ci96 SMP Fri Oct 28 18:23:37␣
↪UTC 2
```

Каталог `/sys/kernel/` содержит набор файлов, которые позволяют нам оперативно без перезагрузки изменять параметры ядра ОС (но таких параметров не очень много). Поэтому часть файлов в этом каталоге доступна для записи, и через эти интерфейсы мы можем управлять ядром. Некоторые из них будут рассмотрены в модуле, посвящённом ядру ОС.

```
localadmin@astra:~$ ls /proc/sys/kernel/
acct                    ngroups_max             sched_energy_aware
acpi_video_flags        nmi_watchdog            sched_rr_timeslice_ms
auto_msgmni             ns_last_pid             sched_rt_period_us
bootloader_type         numa_balancing          sched_rt_runtime_us
bootloader_version      oops_all_cpu_backtrace  sched_schedstats
bpf_stats_enabled       osrelease               sched_util_clamp_max
cad_pid                 ostype                  sched_util_clamp_min
cap_last_cap            overflowgid             sched_util_clamp_min_rt_default
...
```

13.6 Управление процессами

Для управления процессами в Linux существует набор утилит. Рассмотрим работу с основными из них: консольными утилитами (`ps`, `top` и `htop`, `kill`), а для графической (системный монитор).

13.6.1 Управление процессами через консольные утилиты

13.6.1.1 Просмотр процессов

Для просмотра процессов есть три основные консольные утилиты (ps, top и htop) и одна графическая (системный монитор). Утилита ps позволяет получить снимок текущего состояния всех или заданных процессов и их параметров.

```
localadmin@astra:~$ ps
PID TTY          TIME CMD
23526 pts/4    00:00:19 bash
32038 pts/4    00:00:00 ps
```

Просмотр процессов через утилиты top и htop

Утилиты top и htop позволяют просматривать загрузку системы и процессы в реальном времени, изменять сортировку, набор и вид отображаемых параметров, менять значение nice процессов или завершать процессы.

Основное отличие между ними в интерфейсе: htop более удобен для новичков и тех, кто работал с Windows Task Manager. Он поддерживает управление мышью и имеет более «дружелюбный» интерфейс управления.

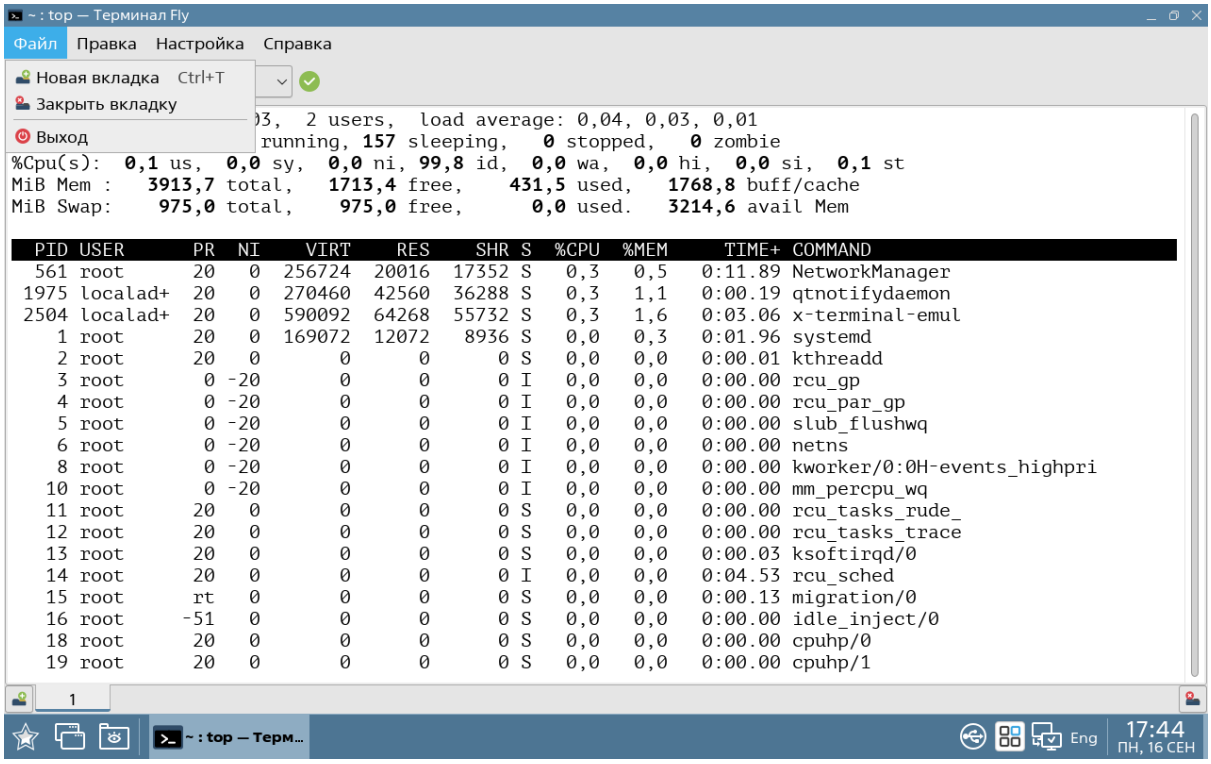


рис. 1.55 – Интерфейс утилиты top

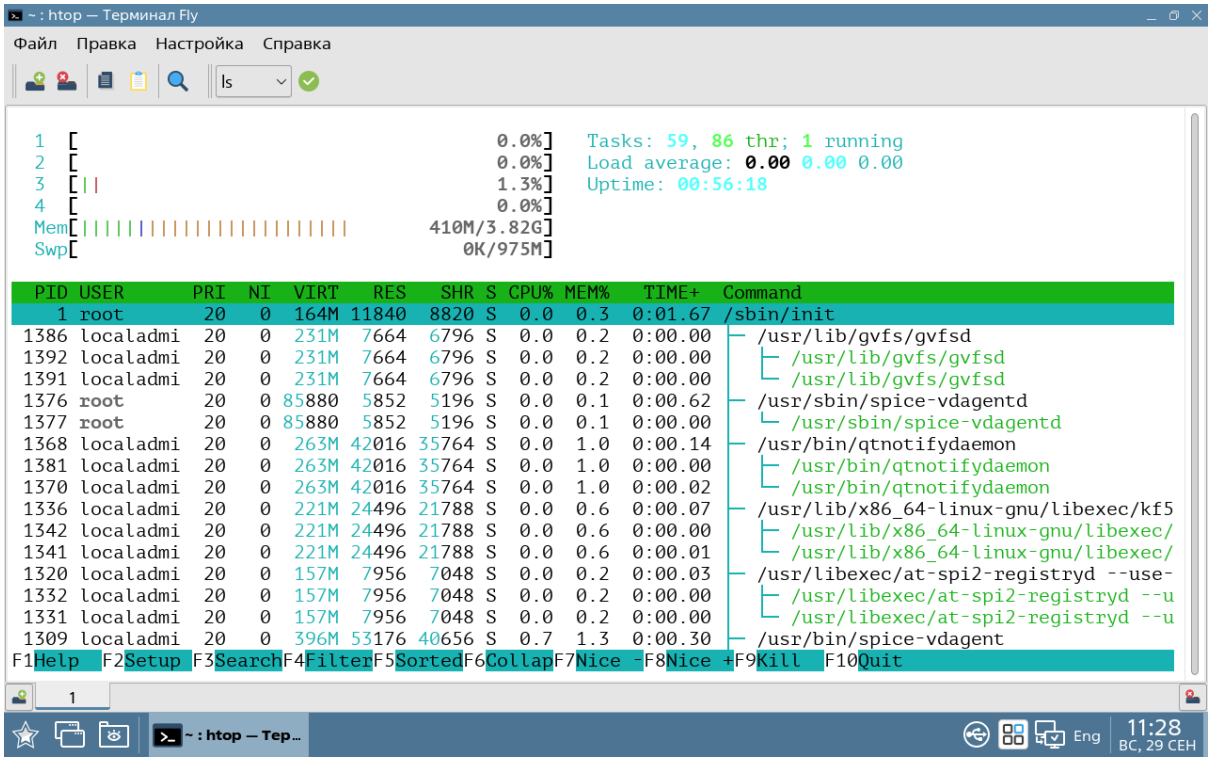
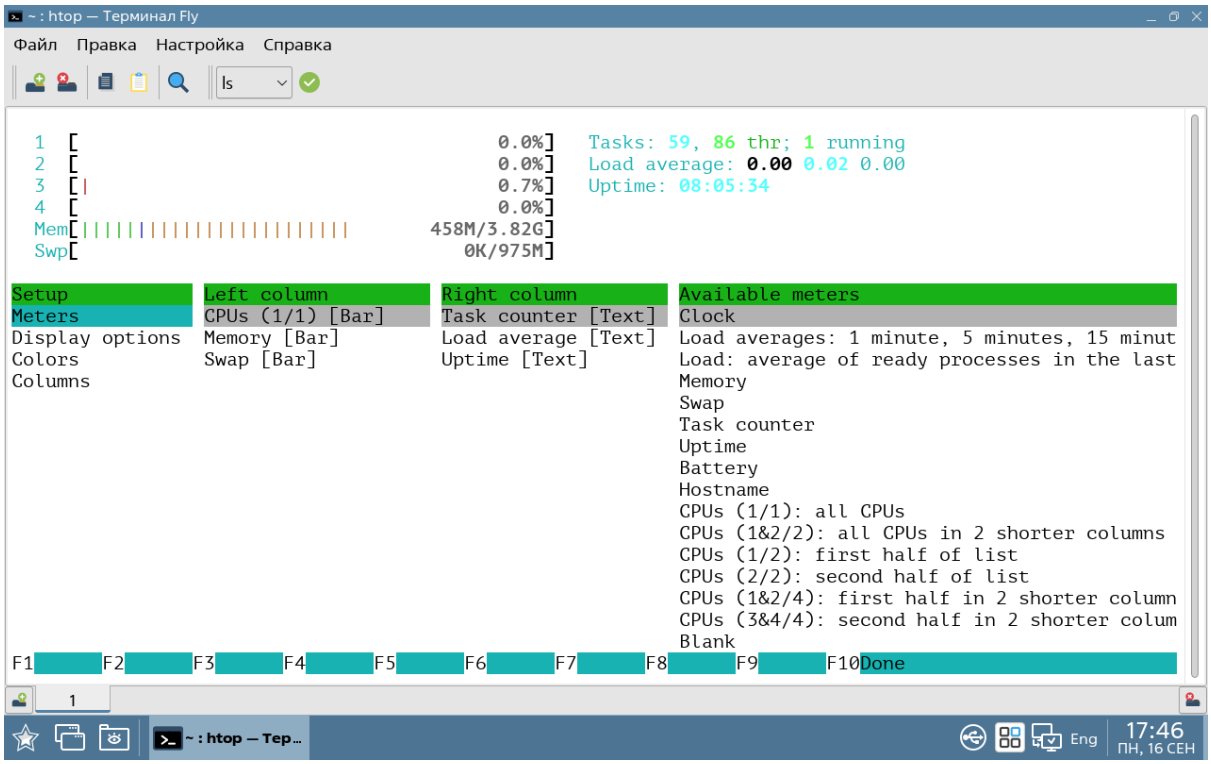
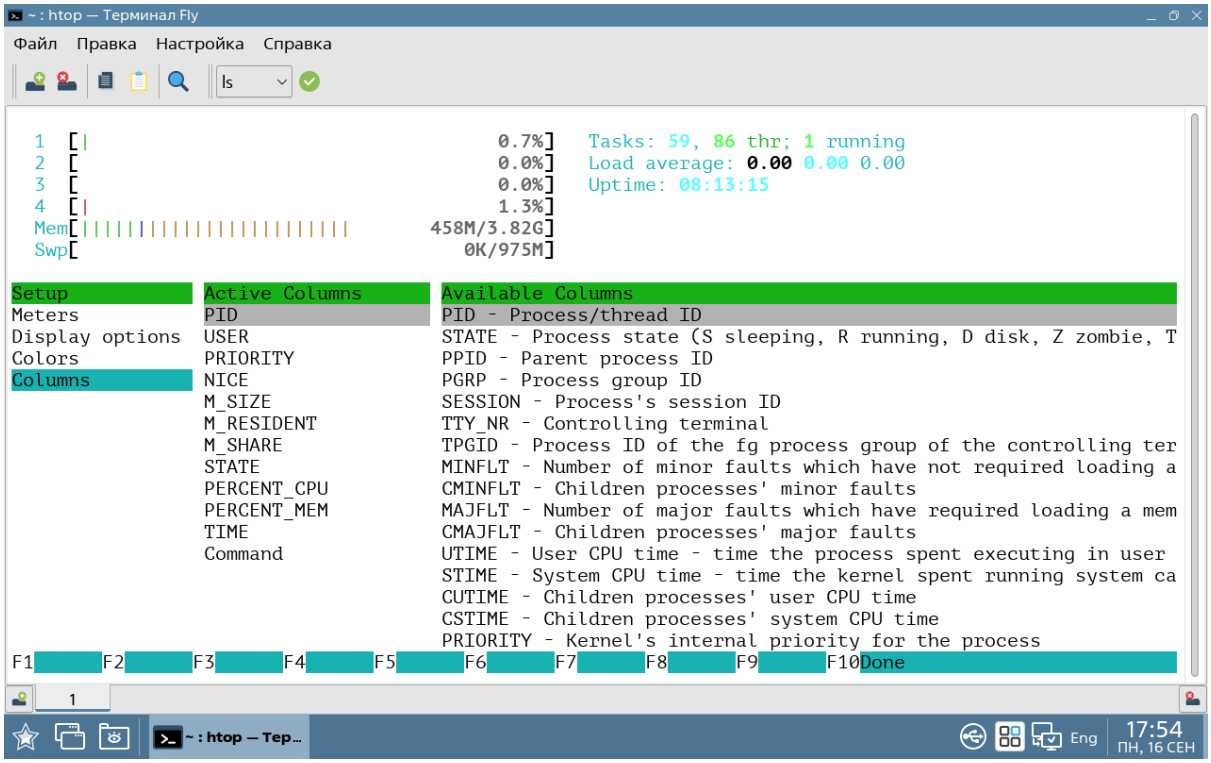


рис. 1.56 – Интерфейс утилиты htop

В нижней части экрана расположена подсказка про доступные действия с помощью функциональных клавиш. Например, при нажатии F2 мы попадем в меню настроек.



В меню Columns нам доступен выбор отображаемых столбцов и их описание.



Через клавишу F1 нам доступна кратка справка. Полная справка доступна через man htop.

Утилита ps

Почти все те же действия можно выполнять и с помощью утилиты ps, но без удобного интерактивного режима. Вот некоторые её опции:

- `-e` – вывести на дисплей процессы всех пользователей. Не отображаются только процессы, не связанные с терминалом.
- `-a` – вывести информацию обо всех процессах, за исключением лидеров сессий и процессов, не ассоциированных с терминалом.
- `-t` – показывать только процессы из этого терминала.
- `-p` – показывать информацию только об указанном процессе.
- `-u` – вывод подробной информации в пользовательско-ориентированном формате.
- `-x` – показать процессы без управляющего терминала.
- `-o` – задать формат вывода информации.

Таким образом, команда `ps aux` выводит все активные процессы.

Для сортировки процессов можно воспользоваться параметром `--sort`, например,

`ps aux --sort=%mem` или `ps aux --sort=%cpu`.

```
localadmin@astra:~$ ps aux --sort=%mem | tail -n 3
localad+  1866  0.0  1.7 565016 68348 ?        Ssl  12:30   0:00 nm-applet
localad+  1868  0.0  2.7 1524876 111060 ?      Ssl  12:30   0:00 fly-sound-applet
root       778  0.0  3.1 912424 125044 tty7    Ssl+ 09:41   0:08 /usr/lib/xorg/Xorg
...
```

Команда `ps -eo euser,ruser,suser,fuser,f,comm,label` выведет информацию об атрибутах EUID, RUID, SUID.

```
localadmin@astra:~$ ps -eo euser,ruser,suser,fuser,f,comm,label
EUSER    RUSER    SUSER    FUSER    F COMMAND         LABEL
root     root     root     root     4 systemd         0:0:0:0
root     root     root     root     1 kthreadd        0:0:0:0
...
localad+ localad+ localad+ localad+ 4 systemd         0:0:0:0
localad+ localad+ localad+ localad+ 5 (sd-pam)        0:0:0:0
localad+ localad+ localad+ localad+ 4 fly-wm          0:0:0:0
...
```

С помощью утилиты ps можно посмотреть и потоки. Воспользуйтесь командами `ps -eLf` и `ps axms`.

```
localadmin@astra:~$ ps -eLf
UID        PID  PPID   LWP  C NLWP STIME TTY          TIME CMD
root         1     0     1  0    1 фев22 ?     00:00:08 /sbin/init
. . .
localad+  1493  1411  1493  0    1 фев22 pts/2 00:00:00 /bin/bash
root     15418     1 15418  0   11 фев26 ?     00:00:29 /usr/sbin/syslog-ng -F --no-
↪caps
root     15418     1 15430  0   11 фев26 ?     00:00:00 /usr/sbin/syslog-ng -F --no-
↪caps
root     15418     1 15431  0   11 фев26 ?     00:00:00 /usr/sbin/syslog-ng -F --no-
↪caps
root     15418     1 19949  0   11 00:01 ?        00:00:00 /usr/sbin/syslog-ng -F --
↪no-caps
root     15418     1 19950  0   11 00:01 ?        00:00:00 /usr/sbin/syslog-ng -F --
↪no-caps
root     15418     1 19951  0   11 00:01 ?        00:00:00 /usr/sbin/syslog-ng -F --
↪no-caps
root     15418     1 19952  0   11 00:01 ?        00:00:00 /usr/sbin/syslog-ng -F --
↪no-caps
root     15418     1 19953  0   11 00:01 ?        00:00:00 /usr/sbin/syslog-ng -F --
↪no-caps
root     15418     1 19954  0   11 00:01 ?        00:00:00 /usr/sbin/syslog-ng -F --
↪no-caps
root     15418     1 19955  0   11 00:01 ?        00:00:02 /usr/sbin/syslog-ng -F --
↪no-caps
root     15418     1 19957  0   11 00:01 ?        00:00:06 /usr/sbin/syslog-ng -F --
↪no-caps
localad+ 17557  1468 17557  0    1 фев26 pts/0 00:00:00 man sudoedit
localad+ 17568 17557 17568  0    1 фев26 pts/0 00:00:00 pager
...
```

```
localadmin@astra:~$ ps axmu
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.5 168924 11548 ?        -    фев22   0:08 /sbin/init
root         -  0.0    -      -     - -        Ss   фев22   0:08 -
. . .
localad+  1493  0.0  0.2   7856  4592 pts/2    -    фев22   0:00 /bin/bash
localad+     -  0.0    -      -     - -        Ss+  фев22   0:00 -
root         -  0.0    -      -     - -        Ssl  фев26   0:29 -
root         -  0.0    -      -     - -        Ssl  фев26   0:00 -
root         -  0.0    -      -     - -        Ssl  фев26   0:00 -
root         -  0.0    -      -     - -        Ssl  00:01   0:00 -
root         -  0.0    -      -     - -        Ssl  00:01   0:00 -
root         -  0.0    -      -     - -        Ssl  00:01   0:00 -
root         -  0.0    -      -     - -        Ssl  00:01   0:00 -
root         -  0.0    -      -     - -        Ssl  00:01   0:00 -
root         -  0.0    -      -     - -        Ssl  00:01   0:00 -
root         -  0.0    -      -     - -        Ssl  00:01   0:02 -
root         -  0.0    -      -     - -        Ssl  00:01   0:06 -
localad+ 17557  0.0  0.2   7288  4248 pts/0    -    фев26   0:00 man sudoedit
localad+     -  0.0    -      -     - -        S+   фев26   0:00 -
localad+ 17568  0.0  0.0   5592   904 pts/0    -    фев26   0:00 pager
localad+     -  0.0    -      -     - -        S+   фев26   0:00 -
...
```

Утилита pstree

С помощью утилиты `pstree` можно вывести дерево процессов. Для её использования необходим пакет psmisc `sudo apt install psmisc`.

```
localadmin@astra:~$ pstree
systemd───3*[VBoxClient───VBoxClient───3*[{VBoxClient}]]
         ├─VBoxClient───VBoxClient───4*[{VBoxClient}]
         ├─VBoxDRMClient───4*[{VBoxDRMClient}]
         ├─VBoxService───8*[{VBoxService}]
         ├─agetty
         ├─alsactl
         ├─astra-orientati───2*[{astra-orientati}]
         ├─at-spi2-registr───2*[{at-spi2-registr}]
         ├─auditd───{auditd}
         ├─avahi-daemon───avahi-daemon
         ├─cron
         ├─cupsd
         ├─2*[dbus-daemon]
         ├─dbus-launch
         ├─dhclient───{dhclient}
         ├─fly-dm───Xorg───{Xorg}
                 └─fly-dm───fly-wm───astra-event-wat───{astra-event-wat}
                                    ├─at-spi-bus-laun───dbus-daemon
                                    │                   └─3*[{at-spi-bus-laun}]
                                    ├─baloo_file───2*[{baloo_file}]
                                    ├─fly-cups-watch───{fly-cups-watch}
                                    ├─fly-notify-osd-───3*[{fly-notify-osd-}]
                                    ├─fly-reflex-serv───3*[{fly-reflex-serv}]
                                    ├─fly-search-pane───3*[{fly-search-pane}]
```

(continues on next page)

(продолжение с предыдущей страницы)

```
          │                                  ├─fly-sound-apple───9*[{fly-sound-apple}]
          │                                  ├─kscreend───2*[{kscreend}]
          │                                  ├─nm-applet───3*[{nm-applet}]
          │                                  ├─org_kde_powerde───6*[{org_kde_powerde}]
          │                                  ├─polkit-kde-auth───4*[{polkit-kde-auth}]
          │                                  ├─ssh-agent
          │                                  └─x-terminal-emul───bash───man───pager
          │                                                      ├─bash───htop
          │                                                      ├─2*[bash]
          │                                                      ├─bash───pstree
          │                                                      └─2*[{x-terminal-emul}]
          ├─fly-getexe
          ...
```

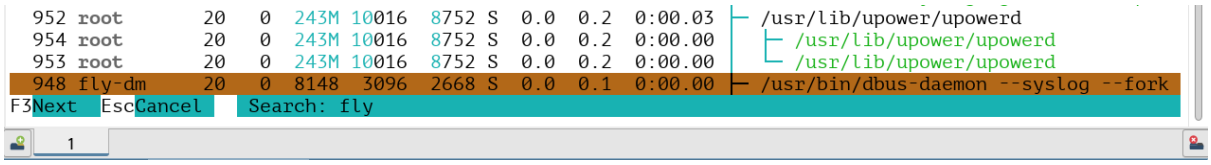
Ключ `-p` позволяет вывести PID процессов `pstree -p`.

```
localadmin@astra:~$ pstree -p
systemd(1)─┬─VBoxClient(1265)─┬─VBoxClient(1266)─┬─{VBoxClient}(1267)
           │                  │                  ├─{VBoxClient}(1268)
           │                  │                  └─{VBoxClient}(1269)
           ├─VBoxClient(1280)─┬─VBoxClient(1281)─┬─{VBoxClient}(1283)
           │                  │                  ├─{VBoxClient}(1284)
...
```

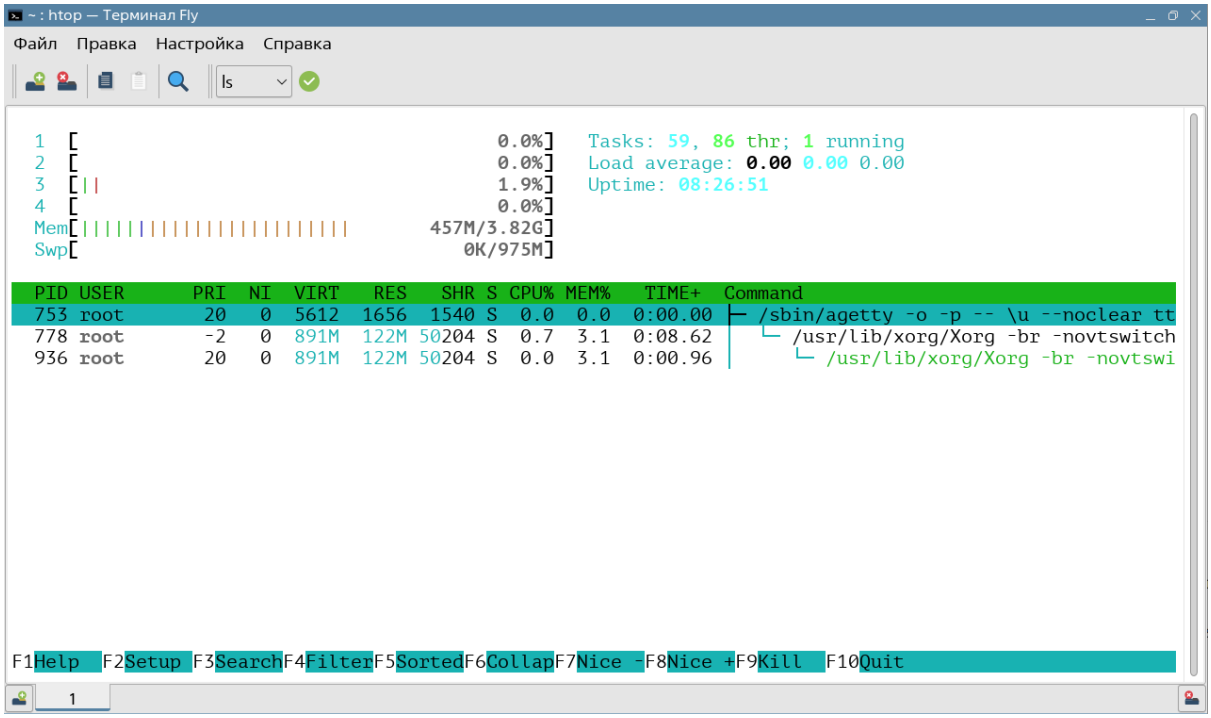
13.6.1.2 Поиск процессов

Утилита htop

Для поиска процессов в htop необходимо воспользоваться пунктом Search. Для этого нажмите клавишу F3. Вам станет доступна строка поиска. После выполнения поиска вы можете переходить от одного процесса к другому. Для выхода из режима поиска нажмите клавишу Esc.



В целях поиска может использоваться и фильтрация. Она доступна по нажатию клавиши F4.



Для очистки фильтра нажмите клавишу Esc.

Утилита ps

Для фильтрации вывода команды `ps` используйте встроенные опции: `-123` или `123`, `-p "1 2"` – вывод информации по PID процессу(ам).

```
localadmin@astra:~$ ps -1
   PID TTY      STAT   TIME COMMAND
   1 ?        Ss     0:08 /sbin/init
localadmin@astra:~$ ps 1
   PID TTY      STAT   TIME COMMAND
   1 ?        Ss     0:08 /sbin/init
localadmin@astra:~$ ps -p "1 2"
   PID TTY          TIME CMD
   1 ?        00:00:08 systemd
   2 ?        00:00:00 kthreadd
```

`-C <имя_команды>` – вывод информации по процессам, имя команды которых соответствует указанному, например, `ps -C bash`.

```
localadmin@astra:~$ ps -C bash
   PID TTY          TIME CMD
1468 pts/0    00:00:00 bash
1486 pts/1    00:00:00 bash
1493 pts/2    00:00:00 bash
```

Используя опцию -o pid, можно вывести только колонку с PID найденных процессов, а используя -o pid=, только PID без названия колонки.

```
localadmin@astra:~$ ps -C bash -o pid
   PID
1468
1486
1493
```

```
localadmin@astra:~$ ps -C bash -o pid=
1468
1486
1493
```

Другие опции фильтрации ps в справке `man 1 ps`.

Для дополнительной фильтрации можно использовать утилиту **grep**, например, `ps aux | grep tty`.

```
localadmin@astra:~$ ps aux | grep tty
root       753  0.0  0.0   5612  1656 tty1     Ss+  09:41   0:00 /sbin/agetty -o -p --
↪ \u --noclear tty1 linux
root       778  0.0  3.1 912424 125044 tty7    Ssl+ 09:41   0:08 /usr/lib/xorg/Xorg -
↪br -novtswitch -quiet -keeptty :0 vt7 -seat seat0 -auth /var/run/xauth/A:0-ZTNgnb
localad+  4933  0.0  0.0   6096   872 pts/0    S+   18:11   0:00 grep tty
```

13.6.1.3 Завершение процессов

При работе с утилитой htop для отправки сигнала процессу необходимо нажать F9. По умолчанию будет выбран сигнал SIGTERM. Для завершения процесса с этим сигналом необходимо нажать F9 и Enter (или перед нажатием Enter вы можете выбрать по необходимости другой сигнал).

13.6.2 Управление через графику в Системном мониторе

Системный монитор напоминает «Менеджер задач» в Windows, что было сделано специально для удобства пользователей и администраторов. Он имеет интуитивно понятный интерфейс и не должен вызвать сложностей при своем использовании.

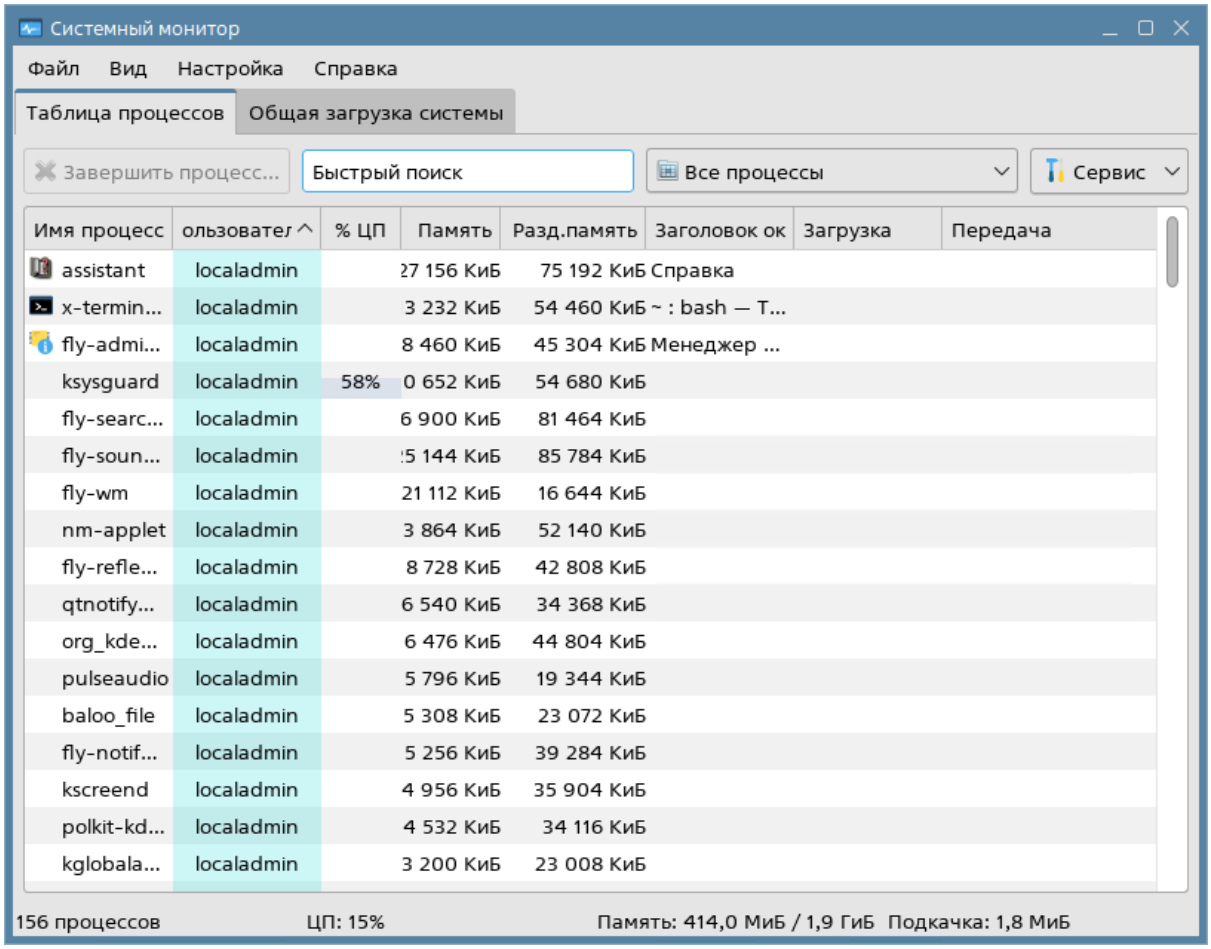


рис. 1.57 – Системный монитор: таблица процессов

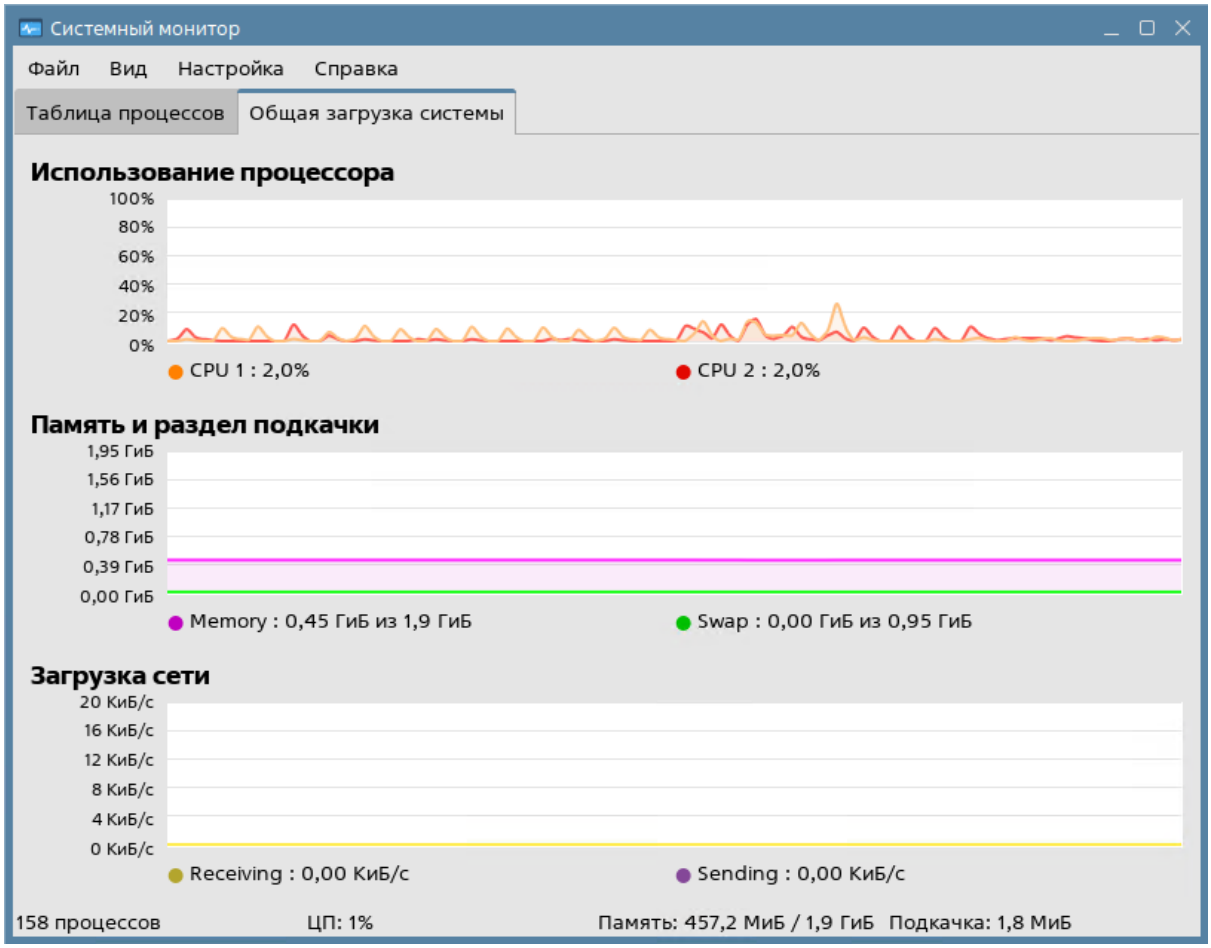
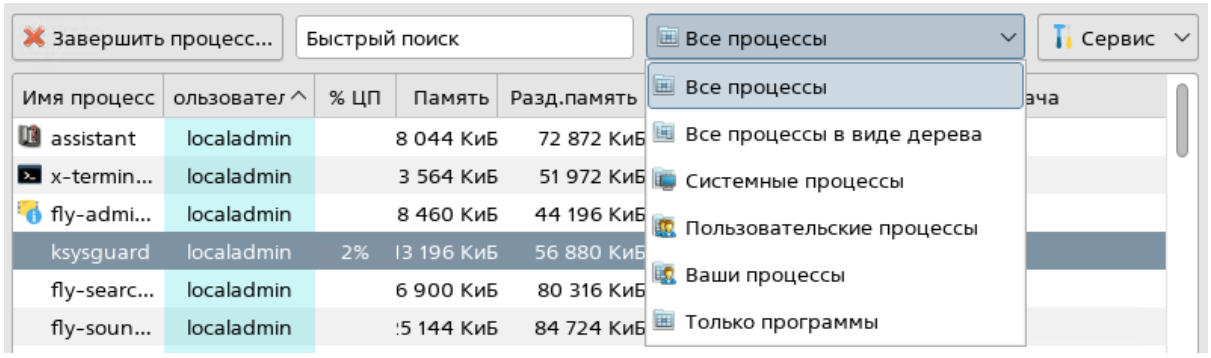
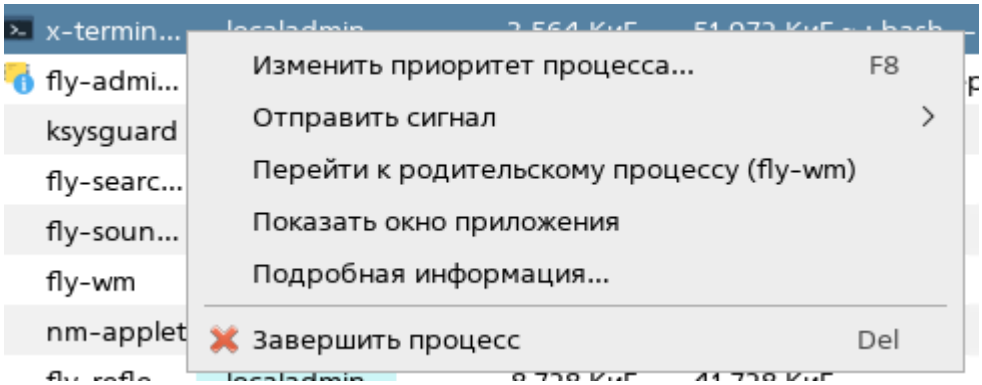


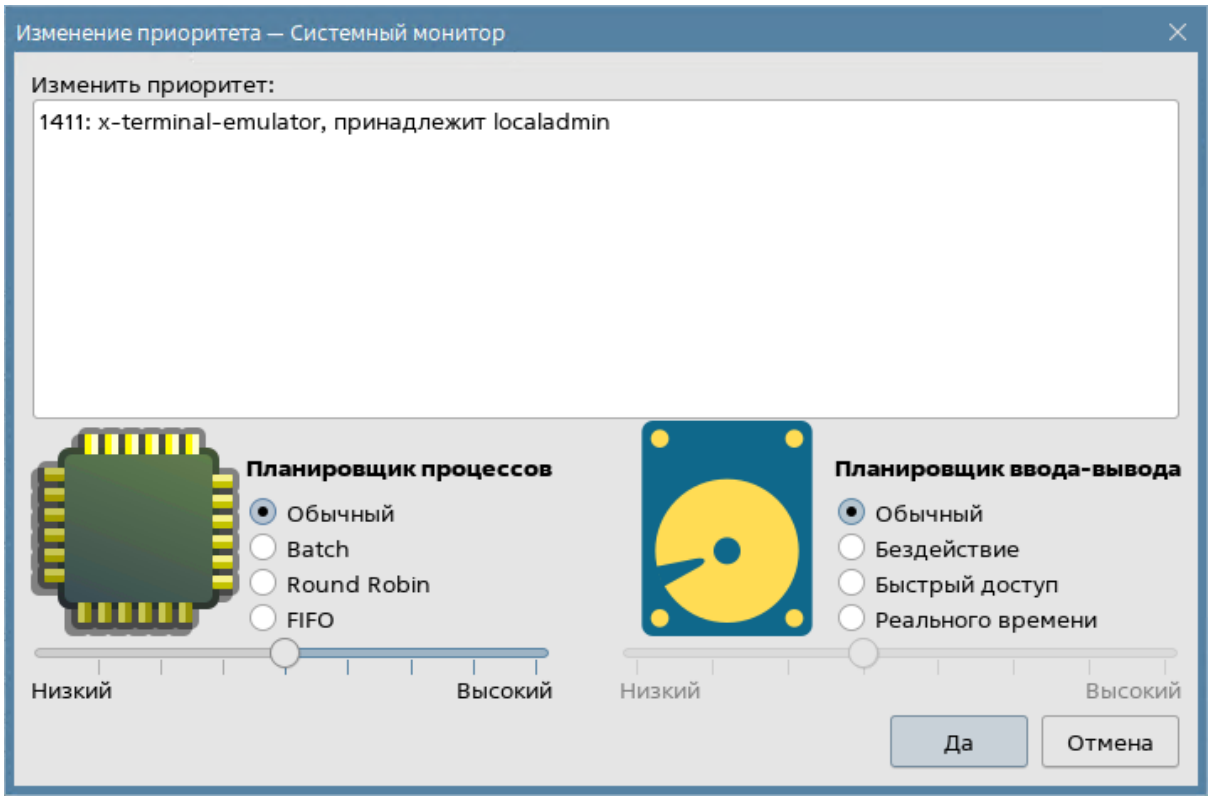
рис. 1.58 – Системный монитор: общая загрузка системы

Управление доступно через контекстное меню и органы управления. Для поиска процессов доступна строка быстрого поиска, для завершения процесса – соответствующая кнопка над списком задач. С помощью выпадающих списков вы можете отфильтровать процессы по категориям.



В контекстном меню процесса можно перейти к окну этого приложения, вывести подробную информацию о процессе, перейти к родительскому процессу, отправить сигнал процессу или изменить его приоритет.





С подробной справкой вы можете ознакомиться в справочной системе Alt + F1.

13.7 Практика и тестирование

13.7.1 Практическая работа: Модуль 13. Процессы, потоки и управление их жизненным циклом

См. также *Требования, правила и цели выполнения практической работы*

13.7.1.1 Практические задания

Задание 1

Внимание: Для корректного выполнения первого задания необходимо выделить виртуальной машине не менее 2 vCPU.

- Выведите на экран все потоки процесса syslog-ng, используя команду `ps`.
- Попробуйте выполните то же задание, используя утилиту `htop`.

Задание 2

Выведите на экран информацию, в том числе PID, %MEM, EUID, RUID, SUID и Command о топ-15 процессов по использованию памяти, используя команду `ps`. Попробуйте выполнить то же задание, используя утилиту `htop`.

Задание 3

Создайте файл в домашнем каталоге `test_nano` и откройте его текстовым редактором nano. В этом же терминале, используя командную строку, найдите этот процесс и повысьте его приоритет до 15. Верните из фона процесс nano и закройте его. Откройте файл `test_nano` текстовым редактором `nano`, задав ему приоритет 25 при создании. Завершите процесс `nano`, отправив ему сигнал SIGKILL.

Задание 4

Тут вам необходимо будет управлять приоритетами процессов. Для наглядности работы по управлению приоритетом процессов используйте скрипт, который создаст нагрузку на CPU.

Создайте файлы `~/scripts/resource-demanding1.sh` и `~/scripts/resource-demanding2.sh` со следующим содержимым (два файла нам нужны, чтобы впоследствии различать процессы не только по PID, но и по команде запуска):

```
#!/bin/bash
while [ 1 == 1 ]; do
   for x in {1..60}; do
   let "res=2**x"
   done
done
EOT
```

- Запустите утилиту `htop` в новом терминале и отсортируйте процессы по использованию CPU.
- Сделайте файлы исполнимыми для владельца и запустите их с помощью команды `taskset` в **фоновом** режиме **на первом** ядре процессора. Сколько процентов первого ядра процессора используют созданные процессы?
- Измените приоритет первого процесса, понизив его до 25, 30, 35 и 39 последовательно. Посмотрите, как менялось при этом использование процессора.
- Повысьте приоритет первого процесса до 15, 10, 5, 0 последовательно. Посмотрите, как менялось при этом использование процессора?
- Задайте приоритет второму процессу так, чтобы он потреблял не менее 75% ресурсов первого ядра CPU, не изменяя при этом приоритет первого процесса (0).
- Завершите процессы, отправив первому сигнал SIGINT, а второму SIGKILL.

Задание 5

Задайте ограничение для текущей оболочки на количество запущенных потоков. Задайте ограничение, превышающее текущее количество запущенных потоков пользователем на 3, запустите несколько раз новый процесс bash, убедитесь, что лимит действует, и закройте терминал с ошибкой.

Задание 6

Получите информацию из псевдофайловой системы `/proc`:

О процессе syslog-ng:

- строка запуска процесса
- окружение процесса
- сведения об объемах данных, прочитанных и записанных процессом в хранилище информации
- лимиты процесса
- информация о статусе процесса

Об ОС:

- список параметров, которые были переданы ядру при загрузке

- сведения о всех установленных процессорах
- информация о состоянии памяти
- перечень устройств в системе
- перечень файловых систем, поддерживаемых ядром ОС
- перечень смонтированных файловых систем
- список разделов подкачки

13.7.1.2 Ответы на практические задания (пошаговые инструкции)

Задание 1 (Ответ)

Выведите на экран все потоки процесса syslog-ng, используя команду `ps`.

Тут может быть несколько вариантов решения. Так как в системе один процесс с именем команды `syslog-ng`, то можно выполнить команду:

```
localadmin@astra:~$ ps -p $(ps -C syslog-ng -o pid=) -L
  PID   LWP TTY          TIME CMD
  532   532 ?        00:00:05 syslog-ng
  532   875 ?        00:00:00 gmain
  532   876 ?        00:00:00 gdbus
  532  1321 ?        00:00:00 syslog-ng
  532  1322 ?        00:00:00 syslog-ng
...
```

Где:

- `ps -C syslog-ng -o pid=` - вернет список PID найденных процессов,
- `ps -p <PID> -L` – выведет список потоков этого процесса.

Для исключения возможной ошибки, когда нам вернется несколько PID, можно добавить команду `head -n 1` для получения только первого из них:

```
localadmin@astra:~$ ps -p $(ps -C syslog-ng -o pid= | head -n 1) -L
  PID   LWP TTY          TIME CMD
  532   532 ?        00:00:05 syslog-ng
  532   875 ?        00:00:00 gmain
  532   876 ?        00:00:00 gdbus
  532  1321 ?        00:00:00 syslog-ng
  532  1322 ?        00:00:00 syslog-ng
...
```

Универсальное решение задачи можно, например, реализовать или через цикл, или через преобразование нескольких строк с PID в одну строку, удовлетворяющую формату опции `-p` команды

`ps`. При этом при использовании цикла необходимо будет избавиться от заголовка в выводе второй и последующих итераций цикла, а второй вариант не требует этого действия:

Вариант 1:

```
localadmin@astra:~$ for pid in $(ps -C bash -o pid=); do ps -p "$pid" -L; done
  PID   LWP TTY          TIME CMD
 1434  1434 pts/0    00:00:00 bash
```

Вариант 2:

```
localadmin@astra:~$ c=1;for pid in $(ps -C bash -o pid=); do if [[(($c -le 1))]];␣
↪then ps -p "$pid" -L;c=$((c+1));else ps -p "$pid" -L --no-headers; fi; done
 PID   LWP TTY          TIME CMD
1468  1468 pts/0    00:00:00 bash
1486  1486 pts/1    00:00:00 bash
1493  1493 pts/2    00:00:00 bash
...
```

Вариант 3:

```
localadmin@astra:~$ ps -p $((ps -C bash -o pid=) | tr '\n' ' ' | tr -s ' ')
  PID TTY      STAT   TIME COMMAND
1468 pts/0    Ss     0:00 /bin/bash
1486 pts/1    Ss     0:00 /bin/bash
1493 pts/2    Ss+    0:00 /bin/bash
17582 pts/3    Ss+    0:02 /bin/bash
```

Где:

- `tr '\n' ' '` – преобразует список строк с PID в последовательность PID в одной строке, разделенных пробельными символами,
- `tr -s ' '` – заменяет каждую последовательность пробельных символов на один пробел.

При использовании утилиты htop можно перейти в меню *Setup* или F2 и в разделе *Display options* убедиться, что активирована опция «Display threads in a different color» для удобства и наглядности. Затем вернуться в основное окно, нажав F10, и выполнить поиск F3 или выполнить фильтрацию F4 по фразе «syslog-ng».

Задание 2 (Ответ)

Выведите на экран информацию, в том числе PID, %MEM, EUID, RUID, SUID и Command о топ-15 процессов по использованию памяти, используя команду `ps`:

```
localadmin@astra:~$ ps -axo pid,%mem,euser,ruser,suser,command --sort=-%mem | head -n␣
↪15
```

Где:

- `-ax` – вывод всех процессов,
- `-o pid,%mem,euser,ruser,suser,command` – выбор столбцов в выводе команды,
- `--sort=-%mem` – сортировка от большего к меньшему,
- `head -n 15` – вывод первых 15 строк, поступивших на стандартный вход команды.

При использовании утилиты `htop` необходимо кликнуть на заголовке столбца MEM% или выполнить сортировку через меню *SortBy* или F6. В меню *Setup* или F2 перейти в раздел Columns, поискать столбцы RUID, EUID, SUID и не найти их. Открыть справку `man htop`, выполнить поиск по UID и убедиться, что htop не может выводить эту информацию.

Задание 3 (Ответ)

Создайте файл в домашнем каталоге `test_nano` и откройте его текстовым редактором nano

```
localadmin@astra:~$ touch ~/test_nano
localadmin@astra:~$ nano ~/test_nano
```

Для того чтобы продолжить работу в этом терминале, необходимо отправить nano в фон сочетанием клавиш Ctrl+Z. Ищем PID процесса:

```
localadmin@astra:~$ ps aux | grep test_nano
localad+  5094  0.0  0.0   6848  3828 pts/0    T    18:35   0:00 nano /home/
↪localadmin/test_nano
localad+  5096  0.0  0.0   6228   880 pts/0    S+   18:35   0:00 grep test_nan
```

Меняем приоритет командой `renice`, PID берем из прошлой команды

```
localadmin@astra:~$ sudo renice -n -5 -p 5094
[sudo] пароль для localadmin:
5094 (process ID) old priority 0, new priority -5
```

Убеждаемся, что он изменился корректно:

```
localadmin@astra:~$ ps -p 5094 -o pid,priority,nice
  PID PRI  NI
 5094  15  -5
```

Еще можно выполнить поиск PID командой

```
localadmin@astra:~$ sudo renice -n -5 -p $(ps -axo pid,command | grep test_nano | tr -
↪s ' ' | cut -d" " -f2 | head -n 1)
5094 (process ID) old priority -5, new priority -5
localadmin@astra:~$ ps -p $(ps -axo pid,command | grep test_nano | tr -s ' ' | cut -d
↪" " -f2 | head -n 1) -o pid,priority,nice
  PID PRI  NI
 5094  15  -5
```

Вызываем из фона процесс `nano`:

```
localadmin@astra:~$ fg
```

Закрываем nano Ctrl + X:

Открываем файл nano, задав приоритет 25. Тут нам sudo не нужно, так как мы создаем процесс с более низким приоритетом, чем стандартный

```
localadmin@astra:~$ nice -n 5 nano ~/test_nano
```

Для того чтобы продолжить работу в этом терминале, необходимо отправить nano в фон сочетанием клавиш Ctrl + Z и убеждаемся, что приоритет был выбран корректно:

```
localadmin@astra:~$ ps -p $(ps -axo pid,command | grep test_nano | tr -s ' ' | cut -d
↪" " -f2 | head -n 1) -o pid,priority,nice
  PID PRI  NI
 5230  25   5
```

Ищем PID процесса, и у вас будут свои идентификаторы:

```
localadmin@astra:~$ ps aux | grep test_nano
localad+  5230  0.0  0.0   6848  3828 pts/0    TN   18:47   0:00 nano /home/
↪localadmin/test_nano
localad+  5247  0.0  0.0   6228   880 pts/0    S+   18:47   0:00 grep test_nano
```

Посылаем сигнал 9 процессу 5230

```
localadmin@astra:~$ kill -9 5230
```

Или ищем PID и выполняем действие автоматически:

```
localadmin@astra:~$ kill -TERM `ps -axo pid,command | grep nano | tr -s ' ' | cut -d"
↪" -f2 | head -n 1`
```

Задание 4 (Ответ)

Тут вам необходимо будет управлять приоритетами процессов. Для наглядности работы по управлению приоритетом процессов используйте скрипт, который создаст нагрузку на CPU.

Создайте файлы `~/scripts/resource-demanding1.sh` и `~/scripts/resource-demanding2.sh` со следующим содержимым (два файла нам нужны, чтобы впоследствии различать процессы не только по PID, но и по команде запуска):

```
#!/bin/bash
while [ 1 == 1 ]; do
   for x in {1..60}; do
      let "res=2**x"
```

(continues on next page)

(продолжение с предыдущей страницы)

```
   done
done
```

И создайте resource-demanding2.sh, скопировав resource-demanding1.sh

```
localadmin@astra:~$ cp ~/scripts/resource-demanding1.sh ~/scripts/resource-demanding2.
↪sh
```

Сделайте файлы исполнимыми:

```
localadmin@astra:~$ sudo chmod u+x ~/scripts/resource-demanding{1,2}.sh
```

Запустите утилиту htop в новом терминале и отсортируйте процессы по использованию CPU, чтобы удобно было смотреть.

Запустите первый скрипт командой, которая запустит процесс bash на первом ядре процессора:

```
localadmin@astra:~$ taskset 0x00000001 bash ~/scripts/resource-demanding1.sh &
```

Запустите второй скрипт на первом ядре процессора в фоновом режиме командой:

```
localadmin@astra:~$ taskset 0x00000001 bash ~/scripts/resource-demanding2.sh &
```

Проверьте в открытом htop, что оба запущенных процесса потребляют примерно по 50% процессорного времени.

Измените приоритет первого процесса, понизив его до 25, 30, 35 и 39 последовательно. Посмотрите, как менялось при этом использование процессора:

Установим приоритет для первого процесса 25:

```
localadmin@astra:~$ renice -n 5 $(ps -axo pid,command | grep resource-demanding1 | tr␣
↪-s ' ' | cut -d" " -f2 | head -n 1)
```

Распределение использования процессора должно быть примерно 25%/75%

Установим приоритет для первого процесса 30

```
localadmin@astra:~$ renice -n 10 $(ps -axo pid,command | grep resource-demanding1 |␣
↪tr -s ' ' | cut -d" " -f2 | head -n 1)
```

Распределение использования процессора должно быть примерно 10%/90%

Установим приоритет для первого процесса 35

```
localadmin@astra:~$ renice -n 15 $(ps -axo pid,command | grep resource-demanding1 |␣
↪tr -s ' ' | cut -d" " -f2 | head -n 1)
```

Распределение использования процессора должно быть примерно 3%/97%

Установим приоритет для первого процесса 39:

```
localadmin@astra:~$ renice -n 19 $(ps -axo pid,command | grep resource-demanding1 |␣
↪tr -s ' ' | cut -d" " -f2 | head -n 1)
```

распределение использования процессора должно быть примерно 1%/99%

Измените приоритет первого процесса, повысив его до 15, 10, 5 и 0 последовательно. Посмотрите, как менялось при этом использование процессора:

Установим приоритет для первого процесса 15:

```
localadmin@astra:~$ sudo renice -n -5 $(ps -axo pid,command | grep resource-
↪demanding1 | tr -s ' ' | cut -d" " -f2|head -n 1)
```

Распределение использования процессора должно быть примерно 75%/25%

Установим приоритет для первого процесса 10

```
localadmin@astra:~$ sudo renice -n -10 $(ps -axo pid,command | grep resource-
↪demanding1 | tr -s ' ' | cut -d" " -f2|head -n 1)
```

Распределение использования процессора должно быть примерно 90%/10%

Установим приоритет для первого процесса 5

```
localadmin@astra:~$ sudo renice -n -15 $(ps -axo pid,command | grep resource-
↪demanding1 | tr -s ' ' | cut -d" " -f2|head -n 1)
```

Распределение использования процессора должно быть примерно 97%/3%

Установим приоритет для первого процесса 0

```
localadmin@astra:~$ sudo renice -n -20 $(ps -axo pid,command | grep resource-
↪demanding1 | tr -s ' ' | cut -d" " -f2|head -n 1)
```

распределение использования процессора должно быть примерно 99%/1%

Задайте приоритет второму процессу так, чтобы он потреблял не менее 75% ресурсов первого ядра CPU, не изменяя при этом приоритет первого процесса. Так как первый процесс имеет максимальный приоритет в политике SCHED_OTHER, то для того чтобы второй процесс имел более высокий приоритет, нам необходимо изменить политику планирования второго процесса. Зададим для него политику планирования SCHED_RR с приоритетом 10:

```
localadmin@astra:~$ sudo chrt -r -p 10 $(ps -axo pid,command | grep resource-
↪demanding2 | tr -s ' '|cut -d" " -f2|head -n 1)
```

распределение использования процессора должно быть примерно 5%/95%

Завершите процессы, отправив первому сигнал SIGTERM, а второму SIGKILL:

```
localadmin@astra:~$ kill -s 2 $(ps -axo pid,command | grep resource-demanding1 | tr -
↪s ' '|cut -d " " -f2|head -n 1)
localadmin@astra:~$ kill -s 9 $(ps -axo pid,command | grep resource-demanding2 | tr -
↪s ' '|cut -d" " -f2|head -n 1)
```

Задание 5 (Ответ)

Задайте ограничение, превышающее текущее количество запущенных потоков пользователем на 3, запустите 3 раза новый процесс bash и убедитесь, что лимит действует:

Для начала необходимо определить текущее количество потоков, запущенных пользователем. Это можно сделать командой

```
localadmin@astra:~$ ps -eLo user | grep $USER | wc -l
179
```

Далее необходимо ввести команду для ограничения на fork():

```
localadmin@astra:~$ ulimit -u "$((179 + 3))"
localadmin@astra:~$ bash
localadmin@astra:~$ bash
localadmin@astra:~$ bash
localadmin@astra:~$ bash
localadmin@astra:~$ bash
bash: fork: retry: Ресурс временно недоступен
bash: fork: retry: Ресурс временно недоступен
^Cbash: fork: Прерван системный вызов
```

Ввести несколько раза команду `bash`, пока не получите сообщение «bash: fork: retry: Ресурс временно недоступен».

Как мы видим, лимиты на текущую сессию установились и теперь терминал можно закрыть.

Задание 6 (Ответ)

Получите информацию из псевдофайловой системы `/proc`.

О процессе syslog-ng:

Найдем PID процесса `ps -C syslog-ng -o pid= | head -1 | xargs` для подстановки в команду типа: `cat /proc/<PID>/<файл с информацией>`, затем будем выполнять команды по очереди:

- строка запуска процесса

```
localadmin@astra:~$ cat /proc/`ps -C syslog-ng -o pid= | head -1 | xargs`/cmdline
```

- окружение процесса

```
localadmin@astra:~$ sudo cat /proc/`ps -C syslog-ng -o pid= | head -1 | xargs`/environ
```

- сведения об объемах данных, прочитанных и записанных процессом в хранилище информации

```
localadmin@astra:~$ sudo cat /proc/`ps -C syslog-ng -o pid= | head -1 | xargs`/io
```

- лимиты процесса

```
localadmin@astra:~$ cat /proc/`ps -C syslog-ng -o pid= | head -1 | xargs`/limits
```

- информация о статусе процесса

```
localadmin@astra:~$ cat /proc/`ps -C syslog-ng -o pid= | head -1 | xargs`/status
```

Информация Об ОС:

- список параметров, которые были переданы ядру при загрузке

```
localadmin@astra:~$ cat /proc/cmdline
```

- сведения о всех установленных процессорах

```
localadmin@astra:~$ cat /proc/cpuinfo
```

- информация о состоянии памяти

```
localadmin@astra:~$ cat /proc/meminfo
```

- перечень устройств в системе

```
localadmin@astra:~$ cat /proc/devices
```

- перечень файловых систем, поддерживаемых ядром ОС

```
localadmin@astra:~$ cat /proc/filesystems
```

- перечень смонтированных файловых систем

```
localadmin@astra:~$ cat /proc/mounts
```

- список разделов подкачки

```
localadmin@astra:~$ cat /proc/swaps
```

13.7.2 Тест: Модуль 13. Процессы, потоки и управление их жизненным циклом

Пожалуйста, ответьте на несколько вопросов, чтобы закрепить полученные знания. Правильные ответы для самопроверки вы сможете найти в конце списка. Обратите внимание, что у некоторых вопросов может быть несколько верных вариантов ответов.

13.7.2.1 Вопросы

Вопрос 1. Закончите фразу «Проблема иерархической структуры процессов в современных Linux дистрибутивах заключается в. . . »

a. излишних накладных расходах при его порождении

b. вероятности появления зомби-процессов

c. невозможности корректно завершить процесс, если процесс родителя был аварийно завершен

d. сложности в распределении ресурсов между процессами

Вопрос 2. Сколько состояний есть у процесса?

a. 3

b. 4

c. 5

d. 6

Вопрос 3. Закончите фразу «Чем больше значение параметра приоритета процесса, тем. . . »

a. приоритет процесса ниже

b. приоритет процесса выше

Вопрос 4. Перечислите все команды, позволяющие управлять процессами в фоновом режиме ра-

боты:

a. Команда `tasks`

b. Команда `jobs`

c. Команда `fg`

d. Команда `fp`

e. Команда `bp`

f. Команда `bg`

Вопрос 5. Какая команда назначает политику SCHED_FIFO?

a. Команда `sudo chrt -f {priority} {PID}`

b. Команда `sudo chrt -o -p 0 {PID}`

c. Команда `sudo chrt -r -p {priority} {PID}`

Вопрос 6. Какая команда выведет процессы и потоки:

a. Команда `ps -ejH`

b. Команда `ps -eLf`

c. Команда `ps -axT`

d. Команда `ps -ejT`

Вопрос 7. Какой сигнал посылается в момент нажатия горячего сочетания клавиш `Ctrl + C` в консоли и означает завершение работы процесса?

a. SIGINT

b. SIGKILL

c. SIGCHILD

d. SIGSTOP

e. SIGSTP

f. SIGCONT

g. SIGTERM

Вопрос 8. Какая команда выполнит поиск по имени команды, использующейся при запуске про-

цесса:

a. Команда `ps -c имя_команды`

b. Команда `find -C имя_команды`

c. Команда `proc -c имя_команды`

d. Команда `ps -C имя_команды`

Вопрос 9. Что содержит в себе файл `/proc/cmdline/` :

a. Команду запуска первого процесса systemd

b. Команду запуска процесса ядра [kthreadd]

c. Список параметров, которые были переданы ядру при загрузке

d. Список команд, выполняемых в данный момент процессором

Вопрос 10. Что сделает команда `ulimit -u <число>` :

a. Ограничит максимальное количество пользователей в системе

b. Ограничит максимальное количество процессов пользователя в текущей оболочке

c. Ограничит максимальное количество процессов пользователя во всех оболочках

d. Ограничит максимальное количество активных сессий пользователей в системе

Ответы на вопросы

1. b
2. c
3. a
4. b, c, f
5. a
6. b
7. a
8. d
9. c

10. b

13.8 Заключение

В этом модуле мы узнали, что такое процессы и как устроен их жизненный цикл. Мы научились просматривать информацию о процессах и управлять ими. Теперь мы знаем, что такое зомби процессы и почему их не нужно бояться. Следующий модуль будет не настолько сложным. В нем мы познакомимся с вопросами архивирования.

13.9 Дополнительные источники информации

- Описание полей diskstat.

13.10 Обратная связь

Если остались вопросы, то их всегда можно задать в специальной теме.

Модуль 14. Архивирование и сжатие информации

14.1 Введение

Из этого модуля вы узнаете о еще одном принципиальном отличии Linux и Windows, только теперь в части работы с архивами. Тема несложная, но крайне прикладная. Приступим!

14.2 Архивация Linux vs Windows

Архивирование и сжатие файлов — это крайне любопытная тема, по которой сразу виден большой контраст между мирами Windows и Linux.

В системе Windows, как вы знаете, большую популярность приобрели Zip-архивы, поэтому Microsoft даже реализовала их поддержку на уровне системы под названием «сжатых папок». Данный формат включает одновременно и архивирование, и сжатие, в связи с чем эти слова в мире Windows произносят вместе и скороговоркой «архивированиеисжатиефайлов». . . , как будто в баню дровишки принес и высыпал.

В то же время в системе Linux граница между этими технологиями проведена настолько четко, что они реализованы в разных форматах и для работы с ними требуются разные приложения! Поэтому давайте уточним эти понятия:

- **Архивирование** — это процесс объединения нескольких файлов в один пакет, называемый архивом, для его дальнейшего хранения в безопасном месте и последующего восстановления с сохранением всей необходимой метаинформации об исходных файлах: дате создания и последнего изменения, владельцах, правах доступа и т.д.

  Отметим два существенных преимущества архивов. Во-первых, один большой архивный файл копируется быстрее, чем много маленьких, т.к. в этом случае данные движутся одним сплошным потоком, как доверху груженый товарняк. При копировании по сети снижаются накладные расходы на установление сессий, а при копировании на магнитные носители мы экономим на записи метаинформации. Во-вторых, сохранение метаинформации о файлах внутри архива позволяет не потерять ее при копировании архива на носитель с любым другим типом файловой системы.

  Для архивирования используют утилиты `tar`, `dd`, `cpio`. Обратное действие по извлечению исходных файлов из архива называют распаковкой, разархивированием или восстановлением данных из архива.

  Архивирование само по себе не включает сжатие данных, но используется вместе с ним.

- **Сжатие файлов** или компрессия — алгоритмическое преобразование данных, выполняемое с целью уменьшения их объема для более эффективного использования устройств хранения и передачи данных.

  Для сжатия файлов в Linux используют утилиты `gzip`, `bzip2`, `xz` и др. Обратное действие по извлечению исходных данных из сжатых файлов называют декомпрессией, восстановлением данных из сжатого файла.

Вот и получается, что когда нам нужно из Windows отправить несколько документов одним файлом, то это вопрос создания простого zip-архивчика. А стоит нам подключиться к веб-серверу по SSH, как отправка нескольких журналов в техподдержку превращается в двухэтапный процесс, сопряженный с фундаментальными идеями архивирования и сжатия информации.

Однако это не означает, что в Linux работа с архивами доступна только для продвинутых администраторов и не может быть столь же простой, как в Windows. Давайте начнем с возможностей графического интерфейса и уже после перейдем к обзору утилит.

14.3 Работа с архивами из графического интерфейса Fly

Архивы довольно часто используются в повседневной работе, поэтому всем пользователям нужно обладать хотя бы базовыми навыками по работе с ними. Файловый менеджер операционной системы Astra Linux позволяет делать это столь же просто, как мы к этому привыкли в Windows.

Для создания архива достаточно выделить в окне нужные объекты, кликнуть правой кнопкой мыши и в контекстном меню выбрать команду *Упаковать → Упаковать в архив...*, см. рис. 1.59.

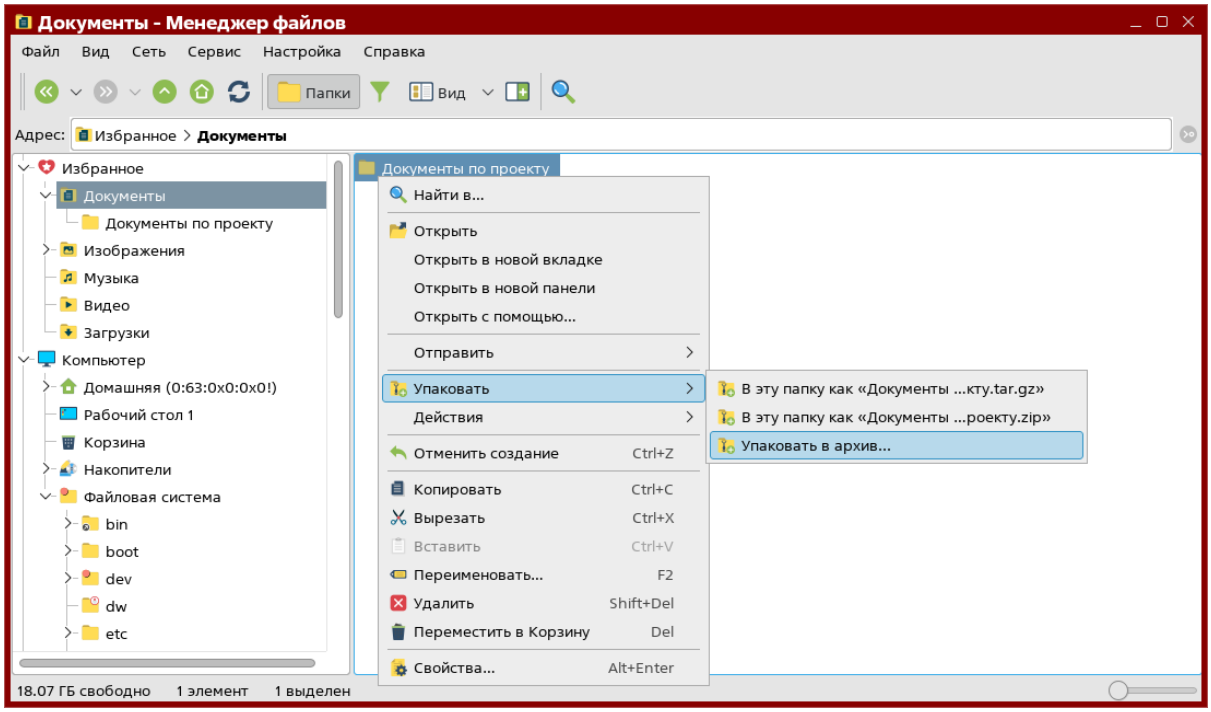


рис. 1.59 – Меню Упаковать: Упаковать в архив

Вам будет доступен большой перечень алгоритмов сжатия: 7-zip, jar, bzip, gzip и др., см. рис. 1.60.

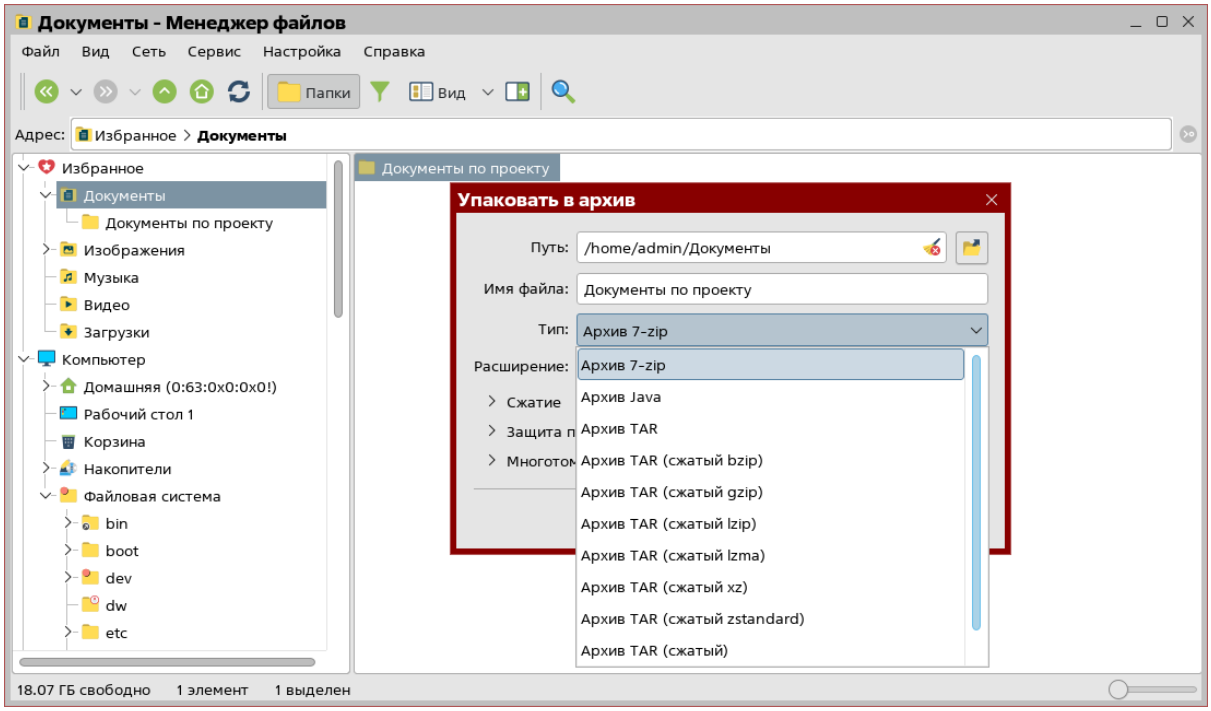


рис. 1.60 – Выбор способа сжатия

Приложение позволяет дополнительно установить степень сжатия и защитить архив паролем. Некоторые алгоритмы позволяют создавать многотомные архивы, когда архив состоит из нескольких частей фиксированного размера, что позволяет разместить его на нескольких носителях, размер которых не вмещает весь архив целиком.

Для просмотра и извлечения информации из архива вы можете легко «зайти в него», как в обычную папку, но этот способ не отличается высокой производительностью, поэтому подходит только в тех случаях, когда вам нужно извлечь всего пару файлов или просто ознакомиться с содержанием архива. Если же вы попробуете таким способом распаковать архив Apache Directory Studio, то вам потребуются минуты, на протяжении которых приложение будет пугать вас своей оценкой в несколько часов.

Поэтому для быстрого извлечения всех файлов из архива вам нужно кликнуть по нему правой кнопкой мыши и выбрать в контекстном меню действие *Распаковать → Распаковать в эту папку*. В этом случае задача будет решена даже быстрее, чем вы успеете прочитать скороговорку «архиватор архивировал самораспаковывающийся архив, да так и не заархивировал».

Примечание: О самораспаковывающихся архивах.

Некоторые архиваторы позволяют создавать так называемые самораспаковывающиеся архивы (*от англ. self-extracting archive, SFX*), которые представляют собой исполняемые файлы, способные извлекать заложенную в них информацию без дополнительных приложений. В Windows такие

архивы стали довольно популярны, а в Linux их саботируют, в том числе и по соображениям безопасности.

Однако в техническом плане для создания SFX-архивов в Linux есть все необходимые инструменты, и, чтобы поместить папку `myfolder` в исполняемый файл `myarchive.7z.sfx`, достаточно выполнить команду `7za a -sfx7zCon.sfx myarchive.7z.sfx myfolder`. Но мы бы не рекомендовали отправлять такой файл Linux-администратору и объяснять, что с ним нужно сделать, если вы не хотите услышать, куда вам нужно пойти.

В завершение стоит отметить, что функции архивации в файловом менеджере Fly реализованы с помощью приложения Ark, поэтому пакет этого архиватора должен быть установлен в системе, иначе соответствующие пункты контекстного меню окажутся недоступны. Если же вы захотите работать с приложением Ark напрямую, то запустите его через *Пуск → Утилиты → Работа с архивами Ark*.

14.4 Работа с архивами из командной строки

14.4.1 Архивирование и распаковка утилитой tar

В давние времена резервные копии хранились исключительно на магнитных лентах, поэтому для повышения скорости чтения, записи и поиска информации на этих типах носителей в конце 70-х годов была разработана первая версия программы `tar`, название которой стало сокращением от слов **t**ape **ar**chiver (с англ. ленточный архиватор).

Архивы tar, или как их еще называют тарболы (*от англ. tar ball, где ball переводится как «сворачивать в клубок»*), позволяют хранить всю необходимую метаинформацию о дате создания файлов, дате последнего изменения, владельцах, правах доступа и т.д., а программа `tar` умеет их правильно восстанавливать с учетом жестких и мягких ссылок. Именно поэтому администраторы Linux называют tar очень мощным инструментом для работы с архивами, а Windows-админы недоумевают, что в этом инструменте такого мощного, если сам по себе tar-формат даже сжатия данных не предполагает.

Примечание: В Linux кроме утилиты tar довольно часто используют еще `dd` и `cpio`. Утилита `cpio` (*от англ. copy input/output — копировать входной/выходной поток*) является предшественницей tar, но до сих пор еще часто используется несмотря на то, что tar покрывает все ее функциональные возможности. Утилита dd позволяет копировать блочное устройство в файл и обратно (*от англ. disk dump — снимок диска*).

Синтаксис утилиты tar следующий:

`tar опции файл_архива [список_файлов_для_архивации] [-C каталог_для_распаковки]`

Основные параметры и опции:

- Параметр `файл_архива` – задает имя файла архива. Параметр является обязательным и должен следовать сразу же после ключа `-f` (`--file`).

Примечание: Мы уже знаем, что расширения файлов не играют в Linux большого значения, но скрипты автоматического дополнения команд по нажатию клавиши Tab (из соображений повышения производительности) не анализируют сигнатуру файлов и опираются только на расширения файлов.

Поэтому, если вы хотите, чтобы при использовании tar у вас работало автодополнение имен файлов, не забывайте добавлять к именам тарболов суффикс .tar. А если (вернее, когда) вы станете счастливым обладателем архива, по имени которого ни джобса не понять, чем он был создан, воспользуйтесь утилитой `file <имя_архива>`.

- Ключ `-f` (`--file`) – позволяет задать имя архива, в который нужно упаковать указанные файлы или из которого их нужно извлечь. Опция является обязательной.

Примечание: Короткий ключ `-f` должен быть последним в группе опций, так как все последующие за ним символы будут восприняты как имя файла. Мы уже наблюдали такое поведение POSIX-ключей на примере команды `ps -aux`, которая работает правильно, но является некорректной, так как буква «x» после ключа `-u` в этом случае трактуется как имя пользователя.

- Ключ `-c` (`--create`) – создать новый архив и поместить в него указанные файлы.
- Ключ `-x` (`--extract`, `--get`) – извлечь файлы из архива. По умолчанию файлы извлекаются в ту же папку, где находится архив.
- Ключ `-t` (`--list`) – показать содержимое архива.

Примечание: Утилита `tar` с ключом `-t` позволяет просматривать содержимое как обычных, так и сжатых тарболов, поэтому вы без проблем сможете посмотреть содержимое архива с помощью команды `tar -tf myarchive.tar.gz`.

- Ключ `-v` (`--verbose`) – показывает подробную информацию о процессе работы.
- Параметр `список_файлов_для_архивации` – имена файлов и папок, которые должны быть помещены в архив. Параметр нужно задать, если используется опция `-c` (`--create`). Элементы списка отделяются друг от друга символом пробела.

При извлечении файлов из архива часто указывают каталог:

- Ключ `-C` (`--directory`) – позволяет задать каталог, в который нужно поместить извлекаемые из архива файлы.
- Параметр `каталог_для_распаковки` – путь к каталогу, в который нужно поместить извлекаемые из архива файлы.

  Параметр нужно задать, если используется опция `-C` (`--directory`).

Дополнительные опции, которые расширят ваши возможности по работе с архивами:

- Ключ `-r` (`--append`) – добавить файлы в конец архива.
- Ключ `-A` (`--catenate`, `--concatenate`) – добавить архив в конец другого архива.
- Ключ `--acls` – сохранить/восстановить списки управления доступом.
- Ключ `--xattrs` – сохранить/восстановить расширенные атрибуты файлов.
- Ключ `--exclude` – исключить файлы по шаблону, в котором допустимо использовать символы подстановки в стиле glob.
- Ключ `--strip-components` – извлечь файлы из вложенных папок архива, отбрасывая начало пути.

  Например, если у вас есть архив, созданный командой `tar -cf varfiles.tar /var`, то при извлечении файла `var/log/auth.log` в текущем каталоге будет создана папка var, в ней папка log, и только там вы найдете нужный вам файл.

  Если же вам нужна только игла без всех этих уток и зайцев, то можно отбросить два первых уровня и создать файл `auth.log` прямо в текущей директории командой:

  `tar -xf varbak.tar --strip=2 var/log/auth.log`
- Ключ `--wildcards` – использовать символы подстановки для выбора определенных файлов, удовлетворяющих критерию.
- Ключ `-u` (`--update`) – обновить архив более новыми файлами с тем же именем.
- Ключ `-h` (`--dereference`) – следовать за символическими ссылками, т.е. архивировать файлы,

на которые указывают символические ссылки. По умолчанию в архив добавляются именно ссылки, а не сами файлы.

Рассмотрим использование этих ключей на практических примерах.

Пример 1. Заархивируем файлы каталога в tar-архив.

Сначала нам нужно создать несколько тестовых файлов и каталогов, а чтобы посмотреть созданную нами структуру, мы воспользуемся утилитой `tree`. Если у вас её еще нет, поставьте ее командой `sudo apt-get install tree`.

```
localadmin@astra:~$ cd ~
localadmin@astra:~$ mkdir test_tar
localadmin@astra:~$ cd test_tar/
localadmin@astra:~/test_tar$ mkdir dir{1..3}
localadmin@astra:~/test_tar$ cat ~/.profile | tee dir{1..3}/file{1..2}
...
localadmin@astra:~/test_tar$ tree
.
├── dir1
│   ├── file1
│   └── file2
├── dir2
│   ├── file1
│   └── file2
└── dir3
    ├── file1
    └── file2
```

Заархивируем директорию `dir1`, используя ключ `-c` (`--create`). Для того чтобы увидеть список файлов, включенных в архив, воспользуемся ключом `-v` (`--verbose`). Обязательным параметром утилиты `tar` является ключ `-f` (`--file`), который задает имя файла архива. Дополнительным параметром идет список файлов для архивации. В нашем случае мы указали один каталог `dir1`.

```
localadmin@astra:~/test_tar$ tar -cvf archive.tar dir1
dir1/
dir1/file1
dir1/file2

localadmin@astra:~/test_tar$ ls -l
итого 24
-rw-r--r-- 1 localadmin localadmin 10240 авг  2 16:18 archive.tar
drwxr-xr-x 2 localadmin localadmin  4096 авг  2 16:16 dir1
drwxr-xr-x 2 localadmin localadmin  4096 авг  2 16:16 dir2
drwxr-xr-x 2 localadmin localadmin  4096 авг  2 16:16 dir3

localadmin@astra:~/test_tar$ tar -tf archive.tar
dir1/
dir1/file1
dir1/file2
```

Пример 2. Добавим каталог `dir2` к тарболу `archive.tar`:

```
localadmin@astra:~/test_tar$ tar -rvf archive.tar dir2
dir2/
dir2/file1
dir2/file2

localadmin@astra:~/test_tar$ tar -tf archive.tar
dir1/
dir1/file1
dir1/file2
dir2/
dir2/file1
dir2/file2
```

Пример 3. Распакуем содержимое тарбола `archive.tar` в новый каталог `extracted`:

```
localadmin@astra:~/test_tar$ mkdir extracted3
localadmin@astra:~/test_tar$ cd extracted3/
localadmin@astra:~/test_tar/extracted3$ tar -xvf ../archive.tar
dir1/
dir1/file2
dir1/file1
dir2/
dir2/file2
dir2/file1
localadmin@astra:~/test_tar/extracted3$ tree
.
├── dir1
│   ├── file1
│   └── file2
└── dir2
    ├── file1
    └── file2
2 directories, 4 files
```

Пример 4. Распакуем содержимое сжатого архива `archive.tar` в указанный каталог. Для выбора каталога необходимо использовать опцию `-C`, которая должна идти после имени архива:

```
localadmin@astra:~/test_tar/extracted3$ cd ..
localadmin@astra:~/test_tar$ mkdir extracted4
localadmin@astra:~/test_tar$ tar -xvf archive.tar -C ./extracted4/
localadmin@astra:~/test_tar$ tree ./extracted4/
./extracted4/
├── dir1
│   ├── file1
│   └── file2
└── dir2
    ├── file1
    └── file2

2 directories, 4 files
```

Пример 5. Извлечем из архива только папку `dir1`:

```
localadmin@astra:~/test_tar/extracted3$ cd ..
```

(continues on next page)

(продолжение с предыдущей страницы)

```
localadmin@astra:~/test_tar$ mkdir extracted5
localadmin@astra:~/test_tar$ cd extracted5
localadmin@astra:~/test_tar$ tar -xvf ../archive.tar dir1
localadmin@astra:~/test_tar/extracted5$ tree
.
└── dir1
    ├── file1
    └── file2
```

Пример 6. Извлечем из `archive.tar` только `file1`, используя опцию `--wildcards`:

```
localadmin@astra:~/test_tar/extracted3$ cd ..
localadmin@astra:~/test_tar$ mkdir extracted6
localadmin@astra:~/test_tar$ cd extracted6
localadmin@astra:~/test_tar$ tar -xvf ../archive.tar --wildcards '\*file1'
localadmin@astra:~/test_tar/extracted6$ tree
.
├── dir1
|    └── file1
└── dir2
  └── file1

2 directories, 2 files
```

14.4.2 Сжатие и восстановление данных утилитами gzip, bzip2, xz

Сжатие без потерь основано на устранении избыточности, которая присутствует в исходных данных. Например, можно заменить в тексте длинные слова на короткие числа, и размер файла сразу же уменьшится. Отношение размера исходного файла к размеру сжатого файла называют коэффициентом сжатия.

Примечание: При сжатии данных, которые обладают низкой избыточностью, например, изображений, аудио- и видеоконтента, обычные алгоритмы без потерь, такие как PNG и FLAC, не показывают хороших коэффициентов сжатия.

Однако ученым удалось разработать алгоритмы сжатия с потерями, такие как JPEG, MP3, AVI и др., которые учитывают особенности восприятия этих видов информации человеком и достигают выдающихся результатов без существенного снижения качества контента.

К сожалению, такие подходы не сработают с уже сжатыми данными, которые обладают избыточностью на уровне белого шума. Поэтому сжимать Zip-архив с помощью RAR будет не многим выгоднее, чем майнить криптовалюту Бузовой.

В мире Linux наибольшее распространение получили следующие форматы сжатых файлов:

- Формат **.gz** (**G**NU **Z**ip) использует алгоритм, который лучше всего подходит для сжатия текстовых данных. На текущий момент существует ряд алгоритмов, которые выигрывают и по скорости, и по степени сжатия, но тем не менее формат gz остается самым распространенным форматом для nix-подобных систем. Для работы с этим форматом предназначена утилита **gzip**.

- Формат **.bz2** использует алгоритм, который медленнее gz, но выдает ощутимо более высокие результаты по сжатию, см. рис. 1.61. Для работы с этим форматом предназначена утилита **bzip2**.

- Формат **.xz** сжимает данные еще лучше, чем .bz2, но делает это еще медленнее. Для работы с этим форматом применяется утилита **xz**.

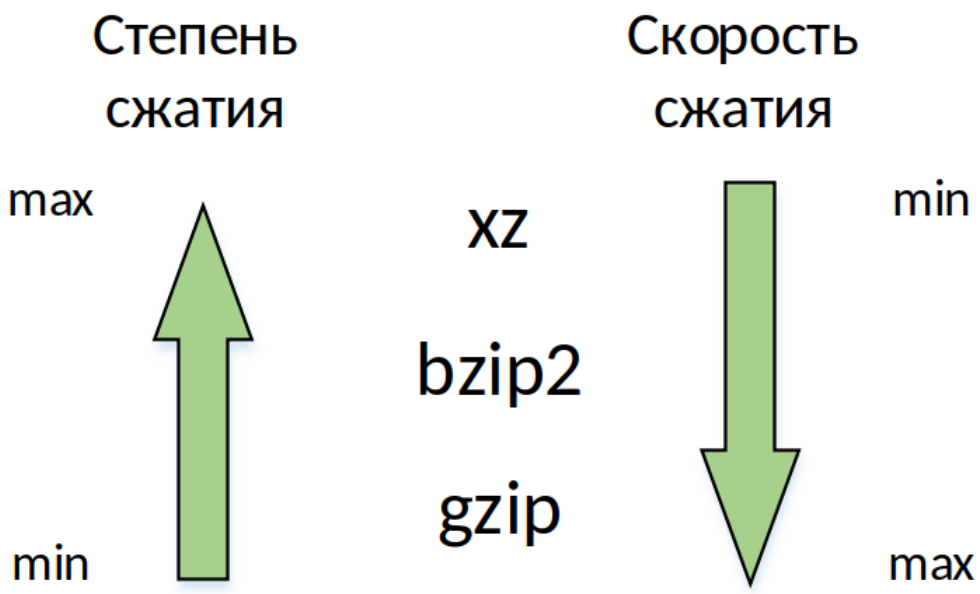


рис. 1.61 – Зависимость степени и скорости сжатия в современных алгоритмах

Утилиты `gzip`, `bzip2` и `xz` реализуют разные алгоритмы сжатия, но их объединяет то, что все они обеспечивают сжатие только одного файла, поэтому для создания сжатых архивов, в которых содержится множество файлов, их нужно использовать совместно с утилитой `tar`.

14.4.2.1 Утилита gzip

Формат gz является самым распространенным, и утилита `gzip` по умолчанию включена во все дистрибутивы Linux, так же как поддержка сжатых папок в Windows. Синтаксис утилиты:

`gzip <опции> <имя_файла>`

Основные параметры и опции:

- Параметр `«имя_файла»` – задает имя файла, для которого нужно выполнить сжатие или декомпрессию. Параметр является обязательным.
- `Без опций` – вызов команды без ключей выполняет сжатие указанного файла.
- Ключ `-d` (`--decompress`, `--uncompress`) – выполняет декомпрессию файла. Этот ключ использует Bash-скрипт **gunzip**.
- Ключ `-c` (`--stdout`) – указывает, что результаты утилиты требуется перенаправить в стандартный поток вывода. При этом исходный файл не удаляется.

  С помощью этой опции вы можете сохранить сжатые данные в файл с любым именем или извлечь их с выводом на экран без сохранения на диске, как делает Bash-скрипт **zcat**. Чтобы узнать подробнее, см. приведенный ниже пример 5.

Дополнительные опции расширят ваши возможности по работе с архивами:

- Ключ `-#` (`--fast`, `--best`) – управляет скоростью/коэффициентом сжатия, где `-1` (или `--fast`) соответствует **быстрому** сжатию с наихудшим коэффициентом, и `-9` (`--best`) соответствует медленному сжатию с **наилучшим** коэффициентом.
- Ключ `-t` – позволяет выполнить проверку целостности сжатого файла.
- Ключ `-l` (строчная L, `--list`) – показывает информацию о коэффициенте сжатия архива.
- Ключ `-f` (`--force`) – выполняет сжатие или распаковку, даже если такой файл уже существует. Если ключ не задан и команда вызвана в интерактивном режиме, то утилита будет спрашивать, нужно ли перезаписывать существующий файл.
- Ключ `-k` (`--keep`) – указывает, что исходный файл удалять не требуется. Данный параметр был добавлен позднее, поэтому его нет в `man gzip`, но вы можете найти информацию о нем в `gzip --help`.
- Ключ `-r` – интерпретирует имя файла как имя каталога и делает по нему полный обход, выполняя сжатие или декомпрессию всех дочерних файлов.

Рассмотрим использование этих ключей на практических примерах.

Пример 1. Выполним сжатие файла `archive.tar`, который мы создали ранее, а затем выведем информацию о сжатом файле. Вы увидите, что исходный файл будет заменен сжатым файлом, в имени которого появится суффикс .gz. По умолчанию используется уровень сжатия 6, что обеспечивает хороший коэффициент сжатия при средней скорости.

```
localadmin@astra:~/test_tar$ gzip archive.tar
localadmin@astra:~/test_tar$ ls
archive_dir3.tar.bz archive_dir3.tar.xz dir1  dir3       extracted2
archive_dir3.tar.gz archive.tar.gz      dir2  extracted  extract
localadmin@astra:~/test_tar$ gzip -l archive.tar.gz
        compressed        uncompressed  ratio uncompressed_name
               624               10240  94.2% archive.tar
```

Пример 2. Выполним декомпрессию файла. Для этого воспользуемся ключом `-d` (`--decompress`):

```
localadmin@astra:~/test_tar$ gzip -d archive.tar
localadmin@astra:~/test_tar$ ls
archive_dir3.tar.bz   archive_dir3.tar.xz  dir1    dir3         extracted2
archive_dir3.tar.gz   archive.tar          dir2    extracted    extract
```

Тот же самый эффект можно получить с помощью Bash-скрипта `gunzip`, который под капотом вызывает `gzip` с ключом `-d`. Еще раз выполним сжатие и декомпрессию файла:

```
localadmin@astra:~/test_tar$ gzip archive.tar
localadmin@astra:~/test_tar$ gunzip archive.tar
localadmin@astra:~/test_tar$ ls
archive_dir3.tar.bz   archive_dir3.tar.xz  dir1    dir3       extracted2
archive_dir3.tar.gz   archive.tar          dir2    extracted  extract
```

Пример 3. Посмотрим разницу между стандартным, минимальным и максимальным уровнем сжатия, используя перенаправление на стандартный вывод с опцией `-c`:

```
localadmin@astra:~/test_tar$ gzip -c archive.tar > archive.tar.gz
localadmin@astra:~/test_tar$ gzip -c -1 archive.tar > archive_min.tar.gz
localadmin@astra:~/test_tar$ gzip -c -9 archive.tar > archive_max.tar.gz
localadmin@astra:~/test_tar$ ls -l
итого 48
-rw-r--r-- 1 localadmin localadmin   712 авг  2 16:28 archive_dir3.tar.bz
-rw-r--r-- 1 localadmin localadmin   572 авг  2 16:28 archive_dir3.tar.gz
-rw-r--r-- 1 localadmin localadmin   604 авг  2 16:28 archive_dir3.tar.xz
-rw-r--r-- 1 localadmin localadmin   610 авг  3 10:36 archive_max.tar.gz
-rw-r--r-- 1 localadmin localadmin   671 авг  3 10:36 archive_min.tar.gz
-rw-r--r-- 1 localadmin localadmin 10240 авг  2 16:34 archive.tar
-rw-r--r-- 1 localadmin localadmin   624 авг  3 10:32 archive.tar.gz
```

Пример 4. Проверим в работе опцию рекурсивного сжатия/декомпрессии файлов.

Создадим тестовый каталог и файлы:

```
localadmin@astra:~/test_compress$ cd ~
localadmin@astra:~$ mkdir -p test_compress/dir_level1/dir_level2 && cd test_compress/
localadmin@astra:~$ echo {00001..99999} > file
localadmin@astra:~$ echo level1_file1 > dir_level1/file1
localadmin@astra:~$ echo level1_file2 > dir_level1/file2
localadmin@astra:~$ echo level2_file1 > dir_level1/dir_level2/file1
localadmin@astra:~$ echo level2_file2 > dir_level1/dir_level2/file2
localadmin@astra:~/test_compress$ tree
.
```

(continues on next page)

(продолжение с предыдущей страницы)

```
├──  dir_level1
│    ├──  dir_level2
│    │    ├── file1
│    │    └── file2
│    ├── file1
│    └── file2
└── file
```

Выполним команду `gzip` с ключом `-r` и увидим, что у всех дочерних файлов появился суффикс .gz, который указывает на то, что они были сжаты.

```
localadmin@astra:~/test_compress$ gzip -r dir_level1
localadmin@astra:~/test_compress$ tree
.
├──  dir_level1
│    ├──  dir_level2
│    │    ├──  file1.gz
│    │    └──  file2.gz
│    ├──  file1.gz
│    └──  file2.gz
└── file
```

Для декомпрессии дочерних файлов можно воспользоваться утилитой `gzip` с ключами `-rd`, но проще вызвать Bash-скрипт gunzip с одним ключом `-r`, так как скрипт под капотом вызовет утилиту `gzip` с ключом `-d` и добавит к списку опций ключ `-r`:

```
localadmin@astra:~/test_compress$ gunzip -r dir_level1
localadmin@astra:~/test_compress$ tree
.
├──  dir_level1
│    ├──  dir_level2
│    │    ├── file1
│    │    └── file2
│    ├── file1
│    └── file2
└── file
```

Пример 5. Выполним архивацию/декомпрессию данных с использованием потоков.

Сначала сожмем файл `/var/log/auth.log` с отправкой результатов в стандартный поток вывода, который далее перенаправим в файл `auth-2024-05-27.gz`:

```
localadmin@astra:~/test_compress$ sudo gzip --stdout /var/log/auth.log > auth-2024-05-
↪27.gz
localadmin@astra:~/test_compress$ file auth-2024-05-27.gz
auth-2024-05-27.gz: gzip compressed data, last modified: Mon May 27 13:58:12 2024,␣
↪from Unix,
original size 20249
```

Теперь выполним декомпрессию с отправкой результатов в стандартный поток вывода. В нашем случае мы выводим поток на экран, но его можно перенаправить на вход какой-нибудь утилиты,

например, less или grep. Таким образом, параметры --stdout и --decompress работают как cat, только применительно к сжатым файлам.

```
localadmin@astra:~/test_compress$ gzip --stdout --decompress auth-2024-05-27.gz
...
May 27 12:50:11 astra sudo[1520]: pam_unix(sudo:session): session opened for user␣
↪root by
localadmin(uid=0)
May 27 12:50:13 astra sudo[1520]: pam_unix(sudo:session): session closed for user root
...
```

Собственно, по этой причине и был создан вспомогательный Bash-скрипт zcat, который под капотом вызывает gzip с ключами -cd:

```
localadmin@astra:~/test_compress$ zcat auth-2024-05-27.gz
...
May 27 12:50:11 astra sudo[1520]: pam_unix(sudo:session): session opened for user␣
↪root by
localadmin(uid=0)
May 27 12:50:13 astra sudo[1520]: pam_unix(sudo:session): session closed for user root
...
```

14.4.2.2 Утилиты bzip2 и xz

В Linux стараются не изобретать велосипеды, которых и без того хватает, поэтому все архиваторы, разработанные после утилиты gzip, заимствуют ее набор ключей, да и она сама старается не отставать и перенимает полезные новинки у «коллег». В соответствии с той же логикой разработчики альтернативных алгоритмов предлагают аналоги команд gunzip и zcat — для популярных типов архивов они называются bunzip2, bzcat, unxz и xzcat соответственно.

Учитывая вышесказанное, для изменения алгоритма сжатия в большинстве случаев достаточно поменять только название команды, и результат останется прежним. Если же что-то пойдет не по плану, то открывайте мануал и проверяйте синтаксис.

Пример 1. В качестве практики предлагаем выполнить сравнение технических характеристик алгоритмов сжатия. Начнем с подготовки файлов для демонстрации.

```
localadmin@astra:~/test_compress$ cp file file_bz
localadmin@astra:~/test_compress$ cp file file_gz
localadmin@astra:~/test_compress$ cp file file_xz
localadmin@astra:~/test_compress$ ls -l
итого 2364
drwxr-xr-x 3 localadmin localadmin   4096 авг  3 11:09  dir_level1
-rw-r--r-- 1 localadmin localadmin    140 авг  3 11:13  dir_level1_recourcive.gz
-rw-r--r-- 1 localadmin localadmin 599994 авг  3 11:26 file
-rw-r--r-- 1 localadmin localadmin 599994 авг  3 11:34 file_bz
-rw-r--r-- 1 localadmin localadmin 599994 авг  3 11:35 file_gz
-rw-r--r-- 1 localadmin localadmin 599994 авг  3 11:35 file_xz
```

Выполним сжатие файлов `file_bz`, `file_gz` и `file_xz` соответствующими утилитами, оценим время выполнения (используя утилиту `time`) и размер сжатых файлов `ls -l`. Как видите, утилита `gzip` оказалась быстрее `xz` примерно в 5 раз, но уровень сжатия утилиты `xz` в 30 раз выше.

```
localadmin@astra:~/test_compress$ time gzip file_gz
real    0m0,115s
user    0m0,094s
sys     0m0,016s

localadmin@astra:~/test_compress$ time bzip2 file_bz
real    0m0,127s
user    0m0,114s
sys     0m0,009s

localadmin@astra:~/test_compress$ time xz file_xz
real    0m0,600s
user    0m0,567s
sys     0m0,017s

localadmin@astra:~/test_compress$ ls -l
итого 896
drwxr-xr-x 3 localadmin localadmin   4096 авг  3 11:09 dir_level1
-rw-r--r-- 1 localadmin localadmin     36 авг  3 11:16 dir_levelevel1_file
-rw-r--r-- 1 localadmin localadmin    140 авг  3 11:13 dir_level1_recourcive.gz
-rw-r--r-- 1 localadmin localadmin 599994 авг  3 11:26 file
-rw-r--r-- 1 localadmin localadmin  80831 авг  3 11:34 file_bz.bz2
-rw-r--r-- 1 localadmin localadmin 212174 авг  3 11:35 file_gz.gz
-rw-r--r-- 1 localadmin localadmin   6824 авг  3 11:35 file_xz.xz
```

14.4.3 Архивирование и сжатие в стиле Windows утилитами zip и 7z

Мы уже знаем, что в Linux архивирование и сжатие файлов – это две разные процедуры, а в Windows популярность приобрели форматы, которые действуют по принципу «all inclusive».

С момента реализации «сжатых папок» в 2000 году фактическим стандартом для Windows стал формат Zip, но сообщество продолжало активно пользоваться в том числе форматами RAR и 7z, которые предлагали более продвинутые алгоритмы сжатия, что очень ценилось во времена дискеток на 1.5 МБ. Но все эти форматы объединяет одно — они поддерживают одновременно и архивирование, и сжатие, не прибегая к помощи сторонних инструментов.

При портировании Zip и 7z на Linux разработчики не стали менять логику работы утилит, поэтому они так и остались «двое из ларца» и не могут использоваться для сжатия тарболов. Вернее, положить tar-файл внутрь zip-архива можно без проблем, но для того чтобы заглянуть внутрь этого тарбола, вам потребуется сначала выполнить полную декомпрессию zip-архива.

И в завершение осталось только добавить, что подходы, закрепившиеся в Windows, действительно оказались намного удобнее, поэтому даже в утилиту **tar** встроили возможность создания сжатых архивов в одно действие:

- Ключ `-z` (`--gzip`) – задействовать утилиту `gzip` для сжатия/декомпрессии данных.
- Ключ `-j` (`--bzip2`) – задействовать утилиту `bzip2` для сжатия/декомпрессии данных.
- Ключ `-J` (`--xz`) – задействовать утилиту `xz` для сжатия/декомпрессии данных.

Примечание: Кстати, интересные факты...

1. Название Zip не является аббревиатурой. Это просто английское слово, которое переводится как «двигаться с высокой скоростью». Разработчики хотели подчеркнуть, что их продукт был намного быстрее аналогов того времени.

2. Название 7z является сокращением от 7-Zip, а еще ранее это приложение называлось 777, где 7 по заявлению автора – это просто цифра, которая была добавлена для того, чтобы приложение находилось выше других в сортированных списках. Но любители НЛО говорят, что это все неспроста, и цифра 7 подозрительно совпадает с телефонным кодом России, ведь создателем этого формата является российский программист **Игорь Павлов**.

3. Название RAR является аббревиатурой и образовано от **R**oshal **AR**chiver. Создателем этого формата является еще один российский программист **Евгений Рошал**.

Мы не станем расписывать параметры вызова этих утилит, так как для zip в большинстве случаев будет достаточно команд `zip/unzip`, а для 7z — ключей **a (append)** и **d (decompress)** соответственно. Поэтому рассмотрим только несколько наиболее частых и полезных примеров.

Пример 1. Выполним архивирование и сжатие группы файлов с помощью утилиты zip:

```
localadmin@astra:~/test_compress$ zip file_archive.zip file_{gz,bz,xz}
 adding: file_gz (deflated 65%)
 adding: file_bz (deflated 65%)
 adding: file_xz (deflated 65%)
localadmin@astra:~/test_compress$ ls -l
итого 2988
drwxr-xr-x 3 localadmin localadmin   4096 авг  3  2023 dir_level1
-rw-r--r-- 1 localadmin localadmin    140 авг  3  2023 dir_level1_recourcive.gz
-rw-r--r-- 1 localadmin localadmin 599994 авг  3  2023 file
-rw-r--r-- 1 localadmin localadmin 636892 авг  3  2023 file_archive.zip
-rw-r--r-- 1 localadmin localadmin 599994 авг  3  2023 file_bz
-rw-r--r-- 1 localadmin localadmin 599994 авг  3  2023 file_gz
-rw-r--r-- 1 localadmin localadmin 599994 авг  3  2023 file_xz
```

Пример 2. Выполним восстановление файлов из архива. Для этого воспользуемся утилитой `unzip`:

```
localadmin@astra:~/test_compress$ unzip file_archive.zip
```

(continues on next page)

(продолжение с предыдущей страницы)

```
Archive:  file_archive.zip
replace file_gz? [y]es, [n]o, [A]ll, [N]one, [r]ename: A
 inflating: file_gz
 inflating: file_bz
 inflating: file_xz
localadmin@astra:~/test_compress$ ls -l
итого 2984
drwxr-xr-x 3 localadmin localadmin   4096 авг  3  2023 dir_level1
-rw-r--r-- 1 localadmin localadmin    140 авг  3  2023 dir_level1_recourcive.gz
-rw-r--r-- 1 localadmin localadmin 599994 авг  3  2023 file
-rw-r--r-- 1 localadmin localadmin 636892 авг  3  2023 file_archive.zip
-rw-r--r-- 1 localadmin localadmin 599994 авг  3  2023 file_bz
-rw-r--r-- 1 localadmin localadmin 599994 авг  3  2023 file_gz
-rw-r--r-- 1 localadmin localadmin 599994 авг  3  2023 file_xz
```

Пример 3. Посмотрим содержимое архива с помощью ключа `-l` (строчная L, от слова list):

```
localadmin@astra:~/test_compress$ unzip -l file_archive.zip
Archive: file_archive.zip
Length Date Time Name
--------- ---------- ----- ----
599994 2023-08-03 11:35 file_gz
599994 2023-08-03 11:34 file_bz
599994 2023-08-03 11:35 file_xz
--------- -------
1799982 3 files
```

Пример 4. Выполним декомпрессию данных в стандартный поток вывода, т.е. на экран, с помощью ключа `-p` (pipe) команды `unzip`:

```
localadmin@astra:~$ sudo zip auth-2024-05-27.zip /var/log/auth.log
  adding: var/log/auth.log (deflated 16%)
localadmin@astra:~$ unzip -p auth-2024-05-27.zip
...
May 27 12:50:11 astra sudo[1520]: pam_unix(sudo:session): session opened for user␣
↪root by
localadmin(uid=0)
May 27 12:50:13 astra sudo[1520]: pam_unix(sudo:session): session closed for user root
...
```

Пример 5. Выполним архивирование и сжатие с помощью утилиты `tar` в одно действие. Для этого воспользуемся ключом `-z`:

```
localadmin@astra:~$ sudo tar -czvf varfiles.tar.gz /var
...
/var/tmp/systemd-private-8cf669c8df1b48ba930a14a57441c918-upower.service-WFWd8H/
/var/tmp/systemd-private-8cf669c8df1b48ba930a14a57441c918-upower.service-WFWd8H/tmp/
/var/local/
/var/run
/var/mail/
localadmin@astra:~$
```

И тут же проведем обратную процедуру:

```
localadmin@astra:~$ tar -xzvf varfiles.tar.gz
...
/var/tmp/systemd-private-8cf669c8df1b48ba930a14a57441c918-upower.service-WFWd8H/
/var/tmp/systemd-private-8cf669c8df1b48ba930a14a57441c918-upower.service-WFWd8H/tmp/
/var/local/
/var/run
/var/mail/

localadmin@astra:~$ ls -l | grep var
drwxr-xr-x 12 localadmin localadmin 4096 мая 16 2023 var
-rw-r--r-- 1 root root 66761786 июн 4 06:44 varfiles.tar.gz
```

14.5 Практика и тестирование

14.5.1 Практическая работа: Модуль 14. Архивирование и сжатие информации

См. также *Требования, правила и цели выполнения практической работы*

14.5.1.1 Практические задания

Задание 1.

1. Создайте 3 текстовых файла:
   - `file1.txt` - Используя случайные символы `/dev/random`
   - `file2.txt` - Копируя один и тот же текст 10 раз:

     «Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.»

   - `file3.txt` - Используя бинарный файл в качестве текстового
2. Заархивируйте файлы, используя команду `tar`.

Задание 2.

Сожмите полученный архив и файл `file2.txt`, используя следующие методы сжатия/архивации – `gzip`, `bzip2`, `xz`, `cpio`.

Задание 3.

Не разжимая полученные сжатые файлы, поищите (zgrep) слово Lorem.

Задание 4.

Разархивируйте только файл `file2.txt`.

14.5.1.2 Ответы на практические задания (пошаговые инструкции)

Задание 1.

1. Создание файлов:

file1.txt: Для создания файла с использованием случайных символов из /dev/random можно воспользоваться утилитой dd следующим образом:

```
localadmin@astra:~$ dd if=/dev/random of=file1.txt bs=1024 count=1
1+0 записей получено
1+0 записей отправлено
1024 байт (1,0 kB, 1,0 KiB) скопирован, 0,000212911 s, 4,8 MB/
```

file2.txt: Для копирования текста 10 раз можно воспользоваться командой `echo` в цикле:

```
localadmin@astra:~$ for i in {1..10}; do echo "Lorem ipsum dolor sit amet,␣
↪consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore␣
↪magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris␣
↪nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in␣
↪voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint␣
↪occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id␣
↪est laborum."; done > file2.txt
```

file3.txt: Для создания файла из бинарного файла в качестве текстового можно воспользоваться утилитой cp:

```
localadmin@astra:~$ cp /usr/bin/zip file3.txt
```

2. Архивирование файлов:

Для архивирования файлов используем команду `tar`:

```
localadmin@astra:~$ tar -cvf archive.tar file1.txt file2.txt file3.txt
```

Задание 2.

Сжатие полученного архива.

gzip:

```
localadmin@astra:~$ gzip -k file2.txt
localadmin@astra:~$ gzip -k archive.tar
```

bzip2:

```
localadmin@astra:~$ bzip2 -k file2.txt
localadmin@astra:~$ bzip2 -k archive.tar
```

xz:

```
localadmin@astra:~$ xz -k file2.txt
localadmin@astra:~$ xz -k archive.tar
```

cpio:

```
localadmin@astra:~$ ls file2.txt | cpio -o > file2.txt.cpio
localadmin@astra:~$ ls archive.tar | cpio -o > archive.cpio
```

Задание 3.

Поиск слова «Lorem» без разжатия файла и распаковки архива:

```
localadmin@astra:~$ zgrep "Lorem" file2.txt.gz
localadmin@astra:~$ bzgrep "Lorem" file2.txt.bz2
localadmin@astra:~$ xzgrep "Lorem" file2.txt.xz
localadmin@astra:~$ cpio -i --to-stdout < file2.txt.cpio | grep "Lorem"
```

```
localadmin@astra:~$ zgrep "Lorem" -a archive.tar.gz
localadmin@astra:~$ bzgrep "Lorem" -a archive.tar.bz2
localadmin@astra:~$ xzgrep "Lorem" -a archive.tar.xz
localadmin@astra:~$ cpio -i --to-stdout < archive.cpio | grep -a "Lorem"
```

Примечание: Так как тут мы работаем с архивным файлом (бинарником), то утилите grep необходимо сказать, чтобы она работала с ним, как с текстом, используя опцию -a.

Задание 4.

Разархивация только файла file2.txt:

```
localadmin@astra:~$ tar -xvf archive.tar file2.txt
```

14.5.2 Тест: Модуль 14. Архивирование и сжатие информации

Пожалуйста, ответьте на несколько вопросов, чтобы закрепить полученные знания. Правильные ответы для самопроверки вы сможете найти в конце списка. Обратите внимание, что у некоторых вопросов может быть несколько верных вариантов ответов.

14.5.2.1 Вопросы

Вопрос 1. Какая команда используется для создания архива в формате tar?

a. Команда `gzip`

b. Команда `zip`

c. Команда `tar`

d. Команда `compress`

Вопрос 2. Какой из следующих форматов архивов является сжатым при помощи gzip?

a. Формат `.tar`

b. Формат `.zip`

c. Формат `.gz`

d. Формат `.bz2`

Вопрос 3. Какая команда используется для распаковки архива, сжатого при помощи gzip?

a. Команда `unzip`

b. Команда `gunzip`

c. Команда `tar`

d. Команда `uncompress`

Вопрос 4. Какая команда используется для просмотра содержимого архива без разархивирования?

a. Команда `zcat`

b. Команда `view`

c. Команда `zshow`

d. Команда `zlist`

Вопрос 5. Какая функция выполняется при архивировании данных?

a. Удаление малоиспользуемых данных из системы

b. Упаковка нескольких файлов в один файл

c. Разделение файла на несколько частей

d. Изменение формата файла

Вопрос 6. Какой формат файла обычно используется для архивирования файлов в Linux?

a. Формат `.exe`

b. Формат `.zip`

c. Формат `.tar`

d. Формат `.png`

Вопрос 7. Какие данные обычно подвергаются сжатию?

a. Только текстовые файлы

b. Только изображения

c. Любые данные

d. Только исполняемые файлы

Вопрос 8. Какой метод сжатия обычно используется для архивирования файлов в формате `.zip`?

a. Метод gzip

b. Метод bzip2

c. Метод LZMA

d. Метод Deflate

Вопрос 9. Какой формат архива может содержать несколько файлов и каталогов, но не использует сжатие?

a. Формат `.gz`

b. Формат `.tar`

c. Формат `.zip`

d. Формат `.bz2`

Вопрос 10. Какой формат архива поддерживает сжатие и шифрование?

a. Формат `.tar`

b. Формат `.gz`

c. Формат `.zip`

d. Формат `.7z`

Ответы на вопросы

1. c
2. c
3. b
4. a
5. b
6. c
7. c
8. d
9. b
10. d

14.6 Заключение

В этом модуле мы разобрались, как работать с архивированием и сжатием данных в Linux и почему для Windows-администраторов эти понятия являются синонимами. Граница между мирами, конечно, сильно истончается, поэтому мы уже видим, как zip и 7z портируют в Linux, а традиционно никсовые форматы наравне с rar и 7z включаются в нативную поддержку сжатых папок Windows (в том числе .tar, .tar.gz, .tar.bz2 и .tar.xz). Однако понимание концептуальных различий между форматами поможет вам уверенно использовать эти технологии в своей работе.

В следующем модуле мы познакомимся с крайне непростой темой – управлением программным

обеспечением в Linux. Тут-то нам и пригодятся знания текущего модуля, потому как deb-пакеты, используемые для установки программ в производных от Debian системах, представляют собой не что иное, как специализированные архивы.

14.7 Дополнительные источники информации

- Формат tar (wikipedia)
- Формат Gzip (wikipedia)
- Список архивов и архиваторов (wikipedia)
- Comparison of file archivers (wikipedia)

14.8 Обратная связь

Если остались вопросы, то их всегда можно задать в специальной теме.

Модуль 15. Управление ПО в Astra Linux

15.1 Введение

Сегодня мы посмотрим на приложения одновременно с двух позиций: как программисты и как системные администраторы. Из этого модуля вы узнаете, что такое deb-пакеты, зачем они нужны, из чего состоят, как работают пакетные менеджеры и зачем их так много. Информация этого модуля будет крайне важна для того, чтобы вы могли уверенно сопровождать ИТ-инфраструктуру под управлением ОС Linux.

15.2 Исполняемые файлы ELF

Для того чтобы эффективно управлять программным обеспечением, нужно понимать, как оно работает и из чего состоит.

В модуле 13 мы уже познакомились с процессами и знаем, что операционная система может выполнять инструкции из специальных бинарных файлов, которые называются исполняемыми файлами. В Windows такие файлы имеют расширение *.exe (от англ. executable files - исполняемые файлы), а в Linux это ELF (от англ. executable and linkable format — формат исполняемых и связываемых файлов).

С помощью утилиты hexdump мы можем легко заглянуть в начало этих файлов и найти специфичные для них сигнатуры, но если с ELF все понятно, то MZ в exe-файлах – это не что иное, как инициалы ведущего разработчика MS-DOS Марка Збиковски (ЧСВ у парня было... мое почтение).

```
localadmin@astra:~$ hexdump -C /usr/bin/cat | head -3
00000000 7f 45 4c 46 02 01 01 00 00 00 00 00 00 00 00 00 |.ELF............|
00000010 02 00 3e 00 01 00 00 00 50 2f 40 00 00 00 00 00 |..>.....P/@.....|
00000020 40 00 00 00 00 00 00 00 90 95 00 00 00 00 00 00 |@...............|

localadmin@astra:~$ file /usr/bin/cat
/usr/bin/cat: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynami...
```

В качестве подопытного возьмем WinBox от MikroTik:

```
localadmin@astra:~$ wget https://mt.lv/winbox64 -O winbox64.exe
localadmin@astra:~$ hexdump -C winbox64.exe | head -3
00000000 4d 5a 90 00 03 00 00 00 04 00 00 00 ff ff 00 00 |MZ..............|
00000010 b8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00 |........@.......|
00000020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |................|
localadmin@astra:~$ file winbox64.exe
winbox64.exe: PE32+ executable (GUI) x86-64 (stripped to external PDB), for MS Windows
```

Обратите внимание, что утилита file без проблем распознает сигнатуру exe-файлов даже без расширения.

Файлы EXE и ELF, конечно, отличаются по формату, но куда важнее то, что их программный код по-разному взаимодействует с операционной системой для получения доступа к ресурсам. Как вы помните, в Windows это взаимодействие осуществляется через API, а в Unix-подобных системах через прямые системные вызовы к ядру, поэтому просто так вы не запустите exe-файл на Linux и ELF-файл на Windows.

Для возможности использования одного и того же приложения на разных операционных системах разработчики выполняют его портирование, т.е. вносят необходимые изменения для возможности компиляции приложения под другую целевую операционную систему. Существуют также языки программирования, библиотеки и фреймворки, которые позволяют изначально писать полностью кроссплатформенный код.

Примечание: Можно отметить еще, что исполняемые файлы компилируются под разные архитектуры процессоров, отличающиеся набором инструкций на аппаратном уровне, поэтому у одной и той же версии приложения могут быть релизы для архитектуры x86-64 (она же AMD64, Intel64, /EM64T), ARM, Эльбрус и др. Компиляторы современных языков высокого уровня позволяют генерировать исполняемый код под разные архитектуры даже без внесения каких-либо изменений в программный код, поэтому этот нюанс не стоит большого внимания.

Для еще большего стирания границы между Windows и Linux был создан продукт Wine, который

умеет перехватывать вызовы exe-приложений к библиотекам операционной системы Windows и подменять их своими, обращаясь уже к системным вызовам системы Linux. Но следует понимать, что этот проект сталкивается с большими трудностями вследствие отсутствия документации по многим элементам Win32 API, поэтому «в бутылке» можно заставить работать не все Windows-приложения и далеко не сразу. Однако, если у вас на предприятии используются какие-то древние клиентские приложения, на переписывание которых нет ресурсов, то Wine вполне может стать для вас спасательным кругом, и такую возможность нужно обязательно проверить.

Примечание: Для установки Wine выполните команду `sudo apt install wine ia32-libs`, после чего приложение WinBox, например, можно будет запустить командой `wine winbox64.exe` или двойным кликом из файлового менеджера. Ну что, одну катку в Сапера или разложим Косынку?

А что насчет совместимости систем, основанных на Linux? Ядра этих систем используют общую кодовую базу и компилируются с небольшими отличиями, поэтому один и тот же ELF-файл может вполне успешно запускаться и на RHEL, и на Debian, и на Astra Linux. Собственно, именно поэтому мы можем запускать Linux-контейнеры разных систем на одном и том же ядре хостовой системы и portable файлы AppImage, которым для работы достаточно указать флаг запуска +x.

15.3 Пакеты приложений и менеджер dpkg

При создании сложных приложений разработчики редко могут обойтись одним исполняемым файлом, т.к. некоторые функции приложения бывает целесообразно вынести в отдельные утилиты, а общий для них программный код оформить в виде динамических библиотек. И не забываем еще про файлы настроек, файлы справочной системы и т.д.

Если же мы вспомним еще о том, что Linux придерживаются концепции единой иерархии файловой системы FHS, то окажется, что все файлы приложения при установке нужно еще и разложить по правильным папкам. Учитывая все вышесказанное, в целях упрощения установки приложений они распространяются в виде архивов, которые называют пакетами программ, а их установку в системе поручают пакетным менеджерам.

Разные операционные системы используют разные пакеты и пакетные менеджеры, например, в Debian это deb-пакеты и Debian package management system (DPKG), в RedHat — RedHat Package Manager (RPM), в Arch Linux — PAMCAN, в OpenSUSE — Zypper. По тому, какие пакеты используются в системе, разные дистрибутивы Linux иногда даже объединяют в группы deb-based, rpm-based и другие.

Есть свой пакетный менеджер и в Windows, который называется Windows Package Manager (winget). Он работает с MSI-пакетами и EXE программами установки. Winget имеет свой репозиторий ПО,

из которого он устанавливает пакеты на целевую систему, но можно также добавить сторонние репозитории или создать свой.

Примечание: Причина, по которой мы имеем дело с таким большим разнообразием пакетных менеджеров, очень проста: управление пакетами является крайне сложной задачей, к решению которой подступались уже неоднократно, но ни один из созданных инструментов нельзя считать идеальным, поэтому будут пробовать еще раз и еще.

Но есть и очевидный плюс от такого разнообразия. Учитывая, что разные линуксы немного по-разному трактуют FHS, наличие разных пакетов сильно снижает путаницу, а то пытались бы мы ставить пакеты от RedHat на Debian и очень удивлялись, почему лыжи не едут.

Система управления пакетами DPKG предоставляет набор динамических библиотек, реализующих программные интерфейсы, и одноименную утилиту командной строки dpkg, которая, используя эти интерфейсы, обеспечивает администраторам доступ ко всем возможностям системы. Приведем наиболее часто используемые команды:

- `sudo dpkg -i <пакет>.deb` – ключ `-i` (`--install`) означает, что нужно установить указанный пакет. Указывается имя deb-файла, включая абсолютный или относительный путь к нему.
- `sudo dpkg -r <имя пакета>` – ключ `-r` (`--remove`) указывает, что требуется удалить пакет, сохраняя конфигурационные файлы в системе. Указывается имя пакета, установленного ранее в системе.
- `sudo dpkg -P <имя пакета>` – ключ `-P` (`--purge`) указывает, что нужно удалить пакет полностью вместе с конфигурационными файлами.
- `dpkg -l` – ключ `-l` (строчная L, `--list`) позволяет получить список всех установленных в системе пакетов. Для получения этой информации привилегии суперпользователя не требуются.
- `dpkg -I <пакет>.deb` – ключ `-I` (заглавная i, `--info`) позволяет получить подробную информацию о пакете из deb-файла, включая краткое описание и список зависимостей.
- `dpkg -s <имя пакета>` – ключ `-s` (`--status`) позволяет получить подробную информацию о пакете, ранее установленном в системе, включая краткое описание.
- `dpkg -L <имя пакета>` – ключ `-L` (`--listfiles`) позволяет получить полный список всех файлов, которые были установлены в системе из этого пакета, включая справочные страницы.
- `dpkg -S <имя файла>` – ключ `-S` (`--search`) позволяет найти пакет, которому принадлежит указанный файл. Для поиска вы можете указать полный путь к файлу, но учитывайте, что

это может быть ссылка, которая не зарегистрирована в dpkg. Поэтому, если вы получите ошибку «не найден путь, подходящий под шаблон», попробуйте указать только имя файла и проанализировать все полученные результаты.

Рассмотрим несколько практических примеров:

С помощью ключа `-l` (строчная L, `--list`) получим список установленных пакетов и отфильтруем его:

```
localadmin@astra:~$ dpkg -l | grep sudo
ii  sudo  1.8.27-1+deb10u5+ci202309191157+astra6  amd64  Provide ...
```

Запросим детализированную информацию об установленном пакете:

```
localadmin@astra:~$ dpkg -s sudo
Package: sudo
Status: install ok installed
Priority: optional
Section: admin
Installed-Size: 3793
Maintainer: Bdale Garbee <bdale@gag.com>
Architecture: amd64
Version: 1.8.27-1+deb10u5+ci202303222055+astra3
Replaces: sudo-ldap
Depends: libaudit1 (>= 1:2.2.1), libc6 (>= 2.27), libpam0g (>= 0.99.7.1), ␣
↪libselinux1 (>= 1.32),
libpam-modules, lsb-
base
Conflicts: sudo-ldap
Conffiles:
/etc/init.d/sudo 1153f6e6fa7c0e2166779df6ad43f1a8
/etc/pam.d/sudo 4af0e4d91e635bebb96ca0eabdea1182
/etc/sudoers 45437b4e86fba2ab890ac81db2ec3606
/etc/sudoers.d/README 8d3cf36d1713f40a0ddc38e1b21a51b6
Description: Provide limited super user privileges to specific users
Sudo is a program designed to allow a sysadmin to give limited root
privileges to users and log root activity.  The basic philosophy is to give
as few privileges as possible but still allow people to get their work done.
This version is built with minimal shared library dependencies, use the
sudo-ldap package instead if you need LDAP support for sudoers.
Homepage: http://www.sudo.ws/
```

Отобразим список всех файлов, установленных из пакета acl:

```
localadmin@astra:~$ dpkg -L acl
/.
/bin
/bin/chacl
/bin/getfacl
/bin/setfacl
. . .
/usr/share/man/man5
/usr/share/man/man5/acl.5.gz
```

Скачаем deb файл Яндекс диска командой `wget`:

```
localadmin@astra:~$ wget https://repo.yandex.ru/yandex-disk/yandex-disk_latest_amd64.
↪deb
--2024-06-02 21:25:02-- https://repo.yandex.ru/yandex-disk/yandex-disk_latest_amd64.
↪deb
Распознаётся repo.yandex.ru (repo.yandex.ru)... 213.180.204.183, 2a02:6b8::183
Подключение к repo.yandex.ru (repo.yandex.ru)|213.180.204.183|:443... соединение␣
↪установлено.
HTTP-запрос отправлен. Ожидание ответа... 200 OK
Длина: 4796514 (4,6M) [application/octet-stream]
Сохранение в: «yandex-disk_latest_amd64.deb»
yandex-disk_latest_amd64.deb
100%[=============================================>]   4,57M 10,7MB/s за 0,4s
2024-06-02 21:25:02 (10,7 MB/s) - «yandex-disk_latest_amd64.deb» сохранён [4796514/
↪4796514]
```

Выведем информацию об этом deb-пакете:

```
localadmin@astra:~$ dpkg -I yandex-disk_latest_amd64.deb
new Debian package, version 2.0.
size 4796514 bytes: control archive=1740 bytes.
     51 байт(а),     1 строк      conffiles
    540 байт(а),    15 строк      control
    729 байт(а),    10 строк      md5sums
   1663 байт(а),    66 строк   *  postinst             #!/bin/sh
    371 байт(а),    24 строк   *  postrm               #!/bin/sh
     41 байт(а),     3 строк   *  prerm                #!/bin/sh
Package: yandex-disk
Version: 0.1.6.1080
Maintainer: Yandex LLC <support@disk.yandex.ru>
Architecture: amd64
Installed-Size: 12404
Depends: libc6, libgcc1, libstdc++6, zlib1g
Section: misc
Priority: optional
Description: Yandex.Disk keeps your files with you at all times.
 You can access photos, videos, and documents on Disk
 from any where in the world where there's internet.
 The Yandex.Disk client console lets you:
 * synchronize files and folders with your Disk
 * get public links to files and folders
 * customize folder syncing
```

Выполним установку Яндекс диска из deb-пакета утилитой `dpkg`:

```
localadmin@astra:~$ sudo dpkg -i yandex-disk_latest_amd64.deb
Выбор ранее не выбранного пакета yandex-disk.
(Чтение базы данных ... на данный момент установлено 191708 файлов и каталогов.)
Подготовка к распаковке yandex-disk_latest_amd64.deb ...
Распаковывается yandex-disk (0.1.6.1080) ...
Настраивается пакет yandex-disk (0.1.6.1080) ...
Using /usr/bin/wget
Обрабатываются триггеры для man-db (2.8.5-2) ...
Обрабатываются триггеры для xserver-xorg-core (2:1.20.14-1ubuntu1astra.se31) ...
update exec ids due to /usr/bin changed
```

Найдем, какому пакету принадлежит утилита `zcat`:

```
localadmin@astra:~$ dpkg -S zcat
gzip: /usr/share/man/man1/zcat.1.gz
xz-utils: /usr/share/man/man1/xzcat.1.gz
bzip2: /usr/share/man/man1/bzcat.1.gz
bzip2: /bin/bzcat
klibc-utils: /usr/lib/klibc/bin/zcat
gzip: /bin/zcat
xz-utils: /usr/bin/xzcat
```

Не забываем, что `/bin` является символьной ссылкой на `/usr/bin`:

```
localadmin@astra:~$ ls -l /bin
lrwxrwxrwx 1 root root 7 мая 16 2023 /bin -> usr/bin
```

При установке и удалении приложений пакетный менеджер dpkg обновляет информацию в своей локальной базе, которая находится в папке `/var/lib/dpkg/` и представляет собой обычные текстовые файлы. Например, когда мы запускаем команду `dpkg -S zcat`, она выполняет поиск всех файлов `*.list` в каталоге `/var/lib/dpkg/info/`, которые содержат имя *zcat*, что можно имитировать с помощью обычной команды `find`:

```
localadmin@astra:~$ find /var/lib/dpkg/info/ -name '*.list' -exec grep -iH 'zcat' {} \
↪;
/var/lib/dpkg/info/bzip2.list:/bin/bzcat
/var/lib/dpkg/info/bzip2.list:/usr/share/man/man1/bzcat.1.gz
/var/lib/dpkg/info/klibc-utils.list:/usr/lib/klibc/bin/zcat
/var/lib/dpkg/info/gzip.list:/bin/zcat
/var/lib/dpkg/info/gzip.list:/usr/share/man/man1/zcat.1.gz
/var/lib/dpkg/info/xz-utils.list:/usr/bin/xzcat
/var/lib/dpkg/info/xz-utils.list:/usr/share/man/man1/xzcat.1.gz
```

15.4 Управление зависимостями и приложения apt, aptitude, synaptic

Разработчики приложений могут писать код с нуля, а могут воспользоваться уже готовыми библиотеками, утилитами или даже целыми приложениями, что может существенно повысить скорость разработки и качество программного кода.

Методология разработки приложений с использованием частей других программных продуктов получила название «повторного использования кода» (*англ. code reuse*). Эта идея лежит в основе абсолютно всех языков программирования высокого уровня, платформ для разработки СУБД и всевозможных фреймворков – все они облегчают разработку за счет того, что предлагают повторное использование их кода.

Например, при разработке Системы управления ресурсами (СУР, устаревшее название ERP) мы можем написать собственную систему хранения данных на языке Си, воспользоваться библиотекой для создания встроенных баз данных Berkeley DB, системой управления базами данных PostgreSQL

или даже технологической платформой для разработки систем учета «1С:Предприятие». И чем более подходящий инструмент будет выбран, тем успешнее окажется продукт в будущем.

В системе Windows администраторы не так часто сталкиваются с ситуациями, когда для работы одного приложения нужно установить что-то еще. Ну, максимум потребуется обновить библиотеки VC++ Runtime и поставить .net framework, виртуальную машину Java (JVM) или кодеки. А в Linux идея повторного использования кода была возведена в абсолют, что стало одновременно ее благословением и проклятьем.

Для работы отдельных Linux-приложений требуется установить десятки или даже сотни других пакетов, и когда-то давно это приходилось делать исключительно вручную, на что уходило катастрофически много времени. Поэтому для упрощения процедуры управления пакетами были созданы продвинутые пакетные менеджеры, такие как APT (Advanced Packaging Tool), которые умеют автоматически разрешать зависимости и загружать все необходимые пакеты из сетевых репозиториев. Первым оказался Debian со своим APT, что обеспечило ему технологическое лидерство на несколько лет, а затем подтянулся и RedHat со своим Yum и DNF.

Например, если скачать с сайта Яндекса пакет `yandex-browser-stable_24.4.1.951-1_amd64.deb`, то при попытке установить его обычной командой `dpkg -i` вы получите ошибку «проблемы зависимостей». Но стоит в `/etc/apt/source.list` раскомментировать репозитории Astra Linux и выполнить команду `apt update`, как установка Яндекс браузера и множества других прикладных приложений станет доступна всего одной командой типа `apt install yandex-browser-stable`. Причем загружаться эти пакеты будут из проверенных репозиториев вендора, что намного безопаснее.

Проверьте, что в файле источников у вас раскомментированы репозитории **base** и **extended**

```
localadmin@astra:~$ cat /etc/apt/sources.list
deb https://download.astralinux.ru/astra/stable/1.7_x86-64/repository-base/ 1.7_x86-
↪64 main contrib non-free
deb https://download.astralinux.ru/astra/stable/1.7_x86-64/repository-extended/ 1.7_
↪x86-64 main contrib non-free
```

Обновите индекс доступных пакетов:

```
localadmin@astra:~$ apt-get update
Сущ:1 https://download.astralinux.ru/astra/stable/1.7_x86-64/repository-base 1.7_x86-
↪64 InRelease
Сущ:2 https://download.astralinux.ru/astra/stable/1.7_x86-64/repository-extended 1.7_
↪x86-64
InRelease
Чтение списков пакетов... Готово
Построение дерева зависимостей
Чтение информации о состоянии... Готово
Может быть обновлён 541 пакет. Запустите «apt list --upgradable» для показа
```

Установите Яндекс браузер из репозиториев Astra Linux:

```
localadmin@astra:~$ sudo apt-get install yandex-browser-stable
[sudo] пароль для localadmin:
Чтение списков пакетов... Готово
Построение дерева зависимостей
Чтение информации о состоянии... Готово
...
Следующие НОВЫЕ пакеты будут установлены:
  ffmpeg-plugin-browser gstreamer1.0-libav gstreamer1.0-plugins-bad ... mesa-vulkan-
↪drivers    yandex-browser-stable
Обновлено 0 пакетов, установлено 40 новых пакетов ...
Необходимо скачать 202 MB архивов.
После данной операции объём занятого дискового
пространства возрастёт на 545 MB.
Хотите продолжить? [Д/н]
...
```

Следует понимать, что DPKG и APT – это технологии, доступ к которым осуществляется через API динамических библиотек, а утилиты dpkg, apt-get и др. – это всего лишь инструменты командной строки, которые к ним обращаются. Утилиты apt-* вышли сразу вместе с APT в 1998 году. Чуть позднее в 2003 году появилась графическая оснастка Synaptic, а затем в 2005 году увидела свет утилита с псевдографическим интерфейсом Aptitude. Но значительным результатом переосмысления опыта работы с зависимостями стала утилита командной строки apt, первый релиз которой состоялся в 2014 году. Архитектура системы управления пакетами представлена на рис. 1.62.

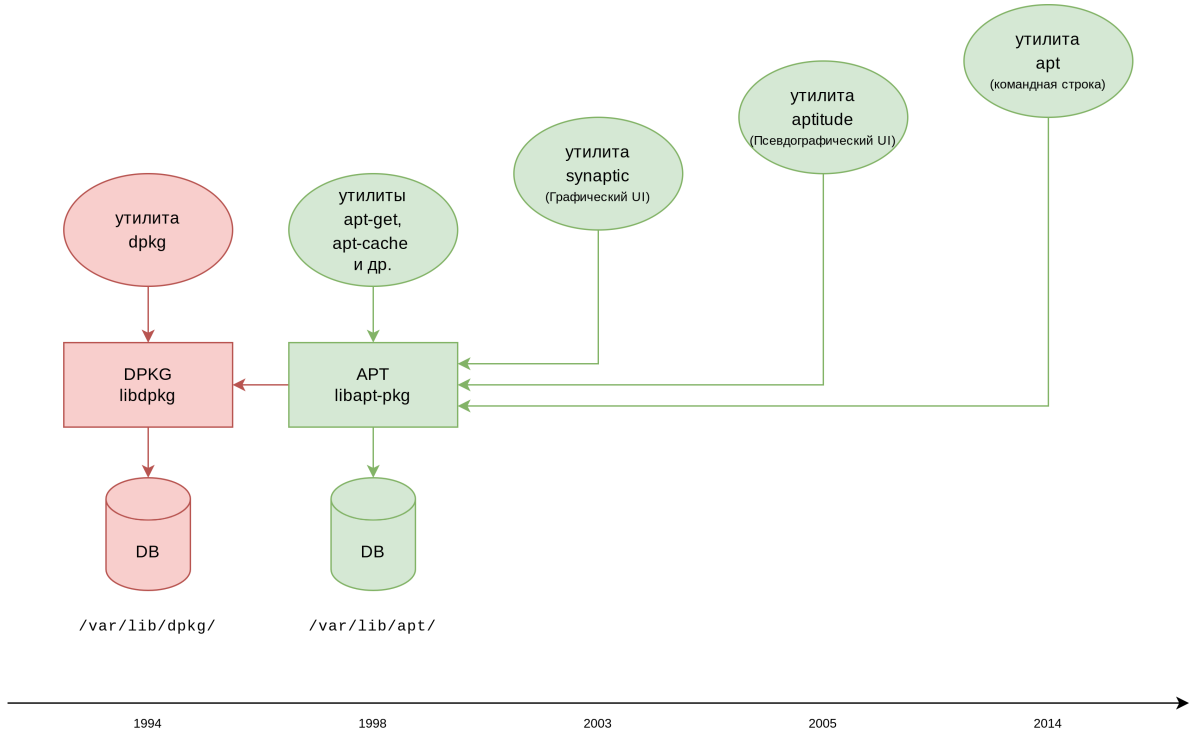


рис. 1.62 – Архитектура системы управления пакетами Debian-систем

Утилита apt предлагает комплексное разрешение зависимостей и разработана как интерфейс конечного пользователя, поэтому выводит более подробные комментарии о результатах выполнения

каждой команды `apt` и использует красивую подсветку вывода. В скриптах автоматизации предпочтение следует отдавать более простым специализированным инструментам APT, таким как apt-get, apt-cache и др. Именно по этой причине apt выдает предупреждение в командах типа `apt list | grep sssd`, когда вывод утилиты перенаправляется дальше по конвейеру: «ВНИМАНИЕ: apt не имеет стабильного интерфейса командной строки. Используйте его в скриптах автоматизации с осторожностью».

Примечание: Имя утилиты apt совпадает с именем пакетного менеджера APT в целом, поэтому вам крайне важно внимательно следить за контекстом, в котором используются эти названия. Например, когда говорят, что Synaptic или Aptitude работают через APT, то подразумевают, что они обращаются к API из libapt-pkg, а не просто вызывают утилиту apt в командной строке. Утилита apt появилась значительно позже и предназначена для работы конечного пользователя в интерактивном режиме.

15.5 Управление списком источников

Менеджер APT загружает пакеты из одного или нескольких репозиториев (источников) программного обеспечения и устанавливает их на ваш компьютер. В роли репозитория может выступать CD-ROM, папка на диске (локальная или смонтированная по сети), удаленный HTTP или FTP сервер.

Управление общим списком источников осуществляется через редактирование файла `/etc/apt/sources.list*`. Приложения могут также создавать свои собственные списки источников в папке `/etc/apt/sources.list.d*`, используя произвольное имя файла с суффиксом «.list». Редактировать файлы можно как обычным текстовым редактором, так и с помощью утилит.

Например, команда `sudo apt edit-sources` откроет файл `/etc/apt/sources.list` в редакторе по умолчанию и при его сохранении выполнит базовую проверку синтаксиса. Еще для редактирования источников вы можете воспользоваться утилитами Synaptic и Aptitude соответственно.

15.5.1 Подключение репозиториев Astra Linux

Ниже представлен пример правильно сконфигурированного файла `/etc/apt/sources.list` для ALSE 1.7.x.

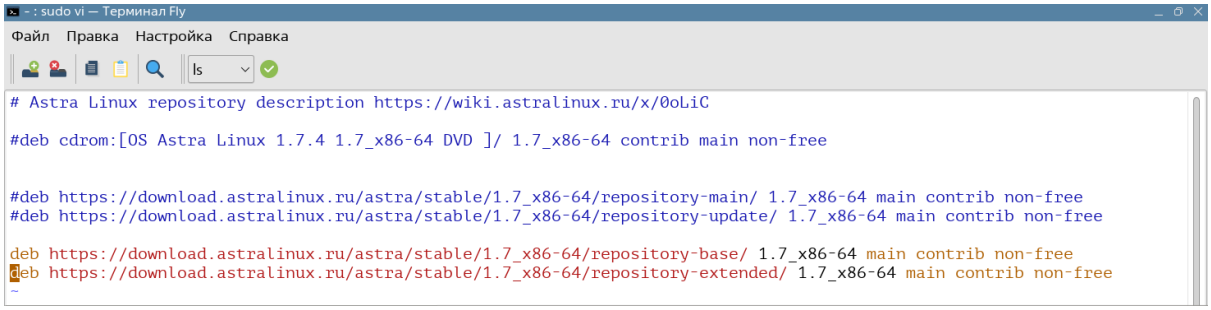


рис. 1.63 – Правильно настроенный список репозиториев sources.list

Для использования репозиториев Astra Linux достаточно подключить источник base (который уже включает main и update) и источник extended (который содержит дополнительное ПО, такое как Яндекс Браузер). Строки, указывающие на CD-ROM, и лишние репозитории можно не удалять, а просто закомментировать. После редактирования списка источников требуется обновить индекс пакетов командой `apt update`.

Разберем синтаксис строки источника:



рис. 1.64 – Синтаксис строки репозитория

Где:

- **deb** — указывает на то, что репозиторий соответствует репозиторию бинарных файлов с предварительно скомпилированными пакетами. Для репозиториев с исходными кодами используют «deb-src».

- https://download.astralinux.ru/astra/stable/1.7_x86-64/repository-main/ — задает адрес репозитория. У интернет-репозиториев адрес начинается с «http(s)://», адреса локальных репозиториев начинаются с «file:/». При добавлении репозитория с диска командой `apt-cdrom add` в файле появится строка «cdrom:[]/». В адресе вы видите, что используется поддерево репозиториев stable – это основная ветка для установки самых крайних оперативных обновлений, содержащая фиксированные ссылки, публикуемые в официальных бюллетенях безопасности. Установку оперативных обновлений следует выполнять по инструкциям, опубликованным в бюллетенях. Есть еще ветка frozen, которая позволяет зафиксировать пакетную базу на определенную версию операционной системы, которая используется для установки комплексных приложений, которые начинают поддержку новых релизов операционной системы с задержкой.

- **1.7_x86-64** - кодовое имя дистрибутива, которое дополняет адрес, уточняя необходимый релиз

продукта. В одном репозитории могут находиться пакеты сразу для нескольких релизов.

- **main contrib non-free** - это группа пакетов, объединенная по условиям использования:
  - **non-free** — группа содержит пакеты, которые не соответствуют принципам свободного ПО, имеют патенты или другие юридические ограничения;
  - **contrib** — группа содержит пакеты, которые сами по себе соответствуют принципам свободного ПО, но зависят от пакетов из группы «non-free» (т. е. не могут без них работать);
  - **main** — группа содержит пакеты свободного ПО, которые не зависят от пакетов из групп «contrib» и «non-free».

Информацию о пакетах для обновления индекса менеджер возьмет из файла Release или InRelease, ссылки на которые формируются по следующей схеме, которая отражена на рис. 1.65. Файлы InRelease отличаются от файлов Release только тем, что они содержат PGP-подписи. Содержание Packages-файлов после выполнения «apt-get update» кэшируется на локальном диске в папке `/var/lib/apt/lists`.



рис. 1.65 – Способ получения адреса на файл Release

Примечание: Обратите внимание, что сервера репозиториев Astra Linux не показывают вам все содержимое, если вы будете переходить по индексным страницам. Полная информация доступна только через Release/InRelease файлы.

15.5.2 Подключение сторонних репозиториев

Как уже было сказано ранее, сторонние репозитории можно вписать в файл `sources.list`, но правильнее создать отдельный файл `имя_репозитория.list` в каталоге `/etc/apt/sources.list.d/`, а также добавить в систему его открытый ключ, чтобы доверять этому репозиторию. Разберем этот процесс на примере добавления репозитория yandex-disk.

Создадим файл `/etc/apt/sources.list/yandex-disk.list`, добавим в него строку с репозиторием yandex-disk и обновим список пакетов в локальном кэше.

```
localadmin@astra:~$ echo "deb http://repo.yandex.ru/yandex-disk/deb/ stable main" |␣
↪sudo tee /etc/apt/sources.list.d/yandex-disk.list
```

(continues on next page)

(продолжение с предыдущей страницы)

```
deb http://repo.yandex.ru/yandex-disk/deb/ stable main
```

Обновим индекс пакетов `sudo apt update` и посмотрим на вывод:

```
localadmin@astra:~$ sudo apt update
Пол:1 http://repo.yandex.ru/yandex-disk/deb stable InRelease [3 900 B]
Игн:2 https://download.astralinux.ru/astra/stable/1.7_x86-64/repository-main 1.7_x86-
↪64 InRelease
Сущ:3 https://download.astralinux.ru/astra/stable/1.7_x86-64/repository-update 1.7_
↪x86-64 InRelease
Сущ:4 https://download.astralinux.ru/astra/stable/1.7_x86-64/repository-base 1.7_x86-
↪64 InRelease
Сущ:5 https://download.astralinux.ru/astra/stable/1.7_x86-64/repository-extended 1.7_
↪x86-64 InRelease
Ошб:1 http://repo.yandex.ru/yandex-disk/deb stable InRelease
   Следующие подписи не могут быть проверены, так как недоступен открытый ключ: NO_
↪PUBKEY
EFC4571D7C90E5AF
Сущ:6 https://download.astralinux.ru/astra/stable/1.7_x86-64/repository-main 1.7_x86-
↪64 Release
Чтение списков пакетов... Готово
W: Ошибка GPG: http://repo.yandex.ru/yandex-disk/deb stable InRelease: Следующие␣
↪подписи не могут быть проверены, так как недоступен открытый ключ: NO_PUBKEY␣
↪EFC4571D7C90E5AF
E: Репозиторий «http://repo.yandex.ru/yandex-disk/deb stable InRelease» не подписан.
N: Обновление из этого репозитория нельзя выполнить безопасным способом, поэтому по␣
↪умолчанию он отключён.
N: Информацию о создании репозитория и настройках пользователя смотрите в справочной␣
↪странице apt-secure(8).
```

Как видим, система не позволяет нам работать с репозиторием, для которого в системе не установлен публичный ключ. Давайте скачаем файл с открытым ключом этого репозитория и добавим его в систему.

```
localadmin@astra:~$ sudo wget http://repo.yandex.ru/yandex-disk/YANDEX-DISK-KEY.GPG
--2023-08-16 13:25:02-- http://repo.yandex.ru/yandex-disk/YANDEX-DISK-KEY.GPG
Распознаётся repo.yandex.ru (repo.yandex.ru)... 213.180.204.183, 2a02:6b8::183
Подключение к repo.yandex.ru (repo.yandex.ru) 213.180.204.183 :80... соединение ␣
↪восстановлено.
HTTP-запрос отправлен. Ожидание ответа... 200 OK
Длина: 1760 (1,7K) [application/octet-stream]
Сохранение в: «YANDEX-DISK-KEY.GPG»
  YANDEX-DISK-KEY.GPG
100% [=============================================>]   1,72K --.-  KB/s
2023-08-16 13:25:03 (206 MB/s) - «YANDEX-DISK-KEY.GPG» сохранён [1760/1760]
localadmin@astra:~$ sudo apt-key add YANDEX-DISK-KEY.GPG
OK
```

Теперь обновим локальный кэш пакетов и убедимся, что репозиторий был добавлен без ошибок и что пакет yandex-disk теперь доступен нам.

```
localadmin@astra:~$ sudo apt update
Пол:1 http://repo.yandex.ru/yandex-disk/deb stable InRelease [3 900 B]
```

(continues on next page)

(продолжение с предыдущей страницы)

```
Игн:2 https://download.astralinux.ru/astra/stable/1.7_x86-64/repository-main 1._x86-64
InRelease
Сущ:3 https://download.astralinux.ru/astra/stable/1.7_x86-64/repository-update 1._x86-
↪64
InRelease
Сущ:4 https://download.astralinux.ru/astra/stable/1.7_x86-64/repository-base 1._x86-
↪64 InRelease
Сущ:5 https://download.astralinux.ru/astra/stable/1.7_x86-64/repository-extended 1._
↪x86-64
InRelease
Сущ:6 https://download.astralinux.ru/astra/stable/1.7_x86-64/repository-main 1._x86-
↪64 Release
Пол:7 http://repo.yandex.ru/yandex-disk/deb stable/main amd64 Packages [561 B]
Пол:8 http://repo.yandex.ru/yandex-disk/deb stable amd64 Contents (deb) [244 B]
Получено 4 705 B за 1с (3 220 B/s)
Чтение списков пакетов... Готово
Построение дерева зависимостей
Чтение информации о состоянии... Готово
Все пакеты имеют последние версии.
localadmin@astra:~$ apt search yandex-disk
Сортировка... Готово
Полнотекстовый поиск... Готово
yandex-disk/неизвестно 0.1.6.1080 amd64
 Yandex.Disk keeps your files with you at all times.
```

Отключим репозиторий yandex-disk, закомментировав его в файле `/etc/apt/sources.list/yandex-disk.list`, обновим кэш и убедимся, что пакет yandex-disk больше не может быть найден.

```
localadmin@astra:~$ sudo sed -i 's/deb/#deb/g' /etc/apt/sources.list.d/yandex-disk.
↪list
localadmin@astra:~$ cat /etc/apt/sources.list.d/yandex-disk.list
#deb http://repo.yandex.ru/yandex-disk/deb/ stable main
localadmin@astra:~$ sudo apt update
Игн:1 https://download.astralinux.ru/astra/stable/1.7_x86-64/repository-main 1.7_x86-
↪64 InRelease
Сущ:2 https://download.astralinux.ru/astra/stable/1.7_x86-64/repository-update 1.7_
↪x86-64 InRelease
Сущ:3 https://download.astralinux.ru/astra/stable/1.7_x86-64/repository-base 1.7_x86-
↪64 InRelease
Сущ:4 https://download.astralinux.ru/astra/stable/1.7_x86-64/repository-extended 1.7_
↪x86-64
InRelease
Сущ:5 https://download.astralinux.ru/astra/stable/1.7_x86-64/repository-main 1.7_x86-
↪64 Release
Чтение списков пакетов... Готово
Построение дерева зависимостей
Чтение информации о состоянии... Готово
Все пакеты имеют последние версии.
localadmin@astra:~$ apt search yandex-disk
Сортировка... Готово
Полнотекстовый поиск... Готово
localadmin@astra:~$
```

Примечание: Если добавить строку *[trusted=yes]* между словом deb и путем до репозитория, например, «deb [trusted=yes] https://...», то APT будет использовать файл Release, доверяя этому источнику без проверки подписей. Однако это подходит только в том случае, если компьютеры не покидают пределов локальной сети. Отключение проверки подписей для источников на переносимых устройствах создает большую угрозу безопасности, т.к. злоумышленники смогут перехватить запрос компьютера и отдать ему пакет с каким-нибудь зловредом вместо запрашиваемого приложения.

15.6 Управление ПО с помощью APT

Как мы уже знаем, для управления в качестве интерфейсов для взаимодействия с APT можно использовать утилиты apt, apt-get, aptitude и synaptic. Рассмотрим работу с каждой из них подробнее.

15.6.1 Утилиты apt и apt-*

Для базового управления пакетами используются следующие команды утилиты apt:

- **update** — загружает информацию о пакетах из репозиториев и обновляет так называемый индекс пакетов, т.е. встроенную базу данных менеджера APT, в которой хранятся записи о доступных пакетах из репозиториев, включенных в системе.

- **install** — устанавливает один и более пакетов, перечисленные через пробел. Например, `sudo apt install package1 package2`. Стоит отметить, что нужно указывать не полное имя deb-файла, а его внутреннее имя (свойство Package), например, `sudo apt install htop`.

  С помощью знака равно можно выбрать конкретную версию пакета для установки, например, «sudo apt install package1=2.15», а с помощью суффикса «-» выполнить удаление пакета вместо установки, например: `sudo apt install package1 package2-`.

- **remove** — удаляет пакеты, сохраняя конфигурационные файлы. Если необходимо не удалить пакет, а установить, то можно воспользоваться суффиксом «+», например: `sudo apt remove package1 package2+`.

- **purge** — действует аналогично команде `remove`, но удаляет в том числе файлы конфигурации, относящиеся к этому пакету.

- **reinstall** — если по какой-то причине вы удалили один из файлов пакета или вам требуется откатить настройки приложения до заводских, то этой командой вы сможете выполнить

повторную установку пакета.

- **search** — выполняет поиск пакетов, в том числе по кратким описаниям пакетов.

  Например, с помощью команды `apt search 'file manager'` вы сможете найти все пакеты, которые имеют какое-то отношение к файловым менеджерам. Строка поиска позволяет использовать в том числе регулярные выражения.

- **show** — показывает информацию о пакете из файла control.

- **list** — выводит список пакетов по заданным критериям.

Для управления обновлениями используются команды:

- **upgrade** — устанавливает доступные обновления для всех пакетов в системе из всех доступных источников (sources.list).

Внимание: Использовать команду «apt upgrade» категорически запрещается, т. к. она не удаляет устаревшие пакеты, даже если это требуется для обновления приложений, что может нарушить работу операционной системы.

Вместо upgrade следует использовать хотя бы apt dist-upgrade», а лучше специальную утилиту «astra-update -a -r -T», где ключ `-a` включает интерактивный режим, ключ `-r` ограничивает обновление списком репозиториев из файла `/etc/apt/sources.list`, а ключ `-T` указывает, что выполнять проверку технологического диска не требуется, т.к. подключенные репозитории содержат все необходимые пакеты.

- **dist-upgrade** — действует аналогично upgrade, но при необходимости устанавливает новые зависимости или удаляет старые.

- **full-upgrade** — действует аналогично dist-upgrade, но может удалять уже установленные пакеты, если они больше не нужны или заменены другими пакетами.

- **autoremove** — используется для очистки системы от пакетов, которые были установлены для разрешения зависимостей и в настоящий момент более не используются, т.к. пакеты, зависимые от них, ранее были удалены из системы.

- **edit-sources** — позволяет редактировать файлы sources.list в редакторе по умолчанию и производит базовую проверку синтаксиса при его редактировании.

Рассмотрим работу с этими опциями на практике. Обновим список доступных пакетов командой `sudo apt update`.

```
localadmin@astra:~$ sudo apt update
Игн:1 https://download.astralinux.ru/astra/stable/1.7_x86-64/  repository-main 1.7_
↪x86-64 InRelease
Сущ:2 https://download.astralinux.ru/astra/stable/1.7_x86-64/  repository-update 1.7_
↪x86-64
InRelease
. . .
Пол:16 https://download.astralinux.ru/astra/stable/1.7_x86-64/  repository-main 1.7_
↪x86-64
amd64 Contents (deb) [1 621 kB]
Получено 19,2 MB за 10c (2 017 kB/s)

Чтение списков пакетов... Готово
Построение дерева зависимостей
Чтение информации о состоянии... Готово
Может быть обновлён 681 пакет. Запустите «apt list --upgradable» для показа.
```

Как видите, утилита запросила информацию из всех перечисленных в sources.list репозиториев и обновила список доступных пакетов. Кроме того, она предлагает нам посмотреть список доступных пакетов для обновления командой `apt list --upgradable`.

Опция search позволяет производить поиск пакетов по полям в информации о пакете (файле control), но для поиска пакетов по входящим в него файлам необходимо воспользоваться утилитой apt-file, которая не входит в состав пакета apt:

```
localadmin@astra:~$ sudo apt install apt-file
Чтение списков пакетов... Готово
Построение дерева зависимостей
Чтение информации о состоянии... Готово
Будут установлены следующие дополнительные пакеты:
 libapt-pkg-perl libexporter-tiny-perl liblist-moreutils-perl libregexp-assemble-perl
Следующие НОВЫЕ пакеты будут установлены:
 apt-file libapt-pkg-perl libexporter-tiny-perl liblist-moreutils-perl libregexp-
↪assemble-perl
Обновлено 0 пакетов, установлено 5 новых пакетов, для удаления отмечено 0 пакетов, и␣
↪681 пакетов
не обновлено.
Необходимо скачать 295 kB архивов.
После данной операции объём занятого дискового пространства возрастёт на 820 kB.
Хотите продолжить? [Д/н] y
Пол:1 https://download.astralinux.ru/astra/stable/1.7_x86-64/repository-main 1.7_x86-
↪64/main amd64
libapt-pkg-perl am
d64 0.1.34 [71,1 kB]
Пол:2 https://download.astralinux.ru/astra/stable/1.7_x86-64/repository-main 1.7_x86-
↪64/main amd64
libexporter-tiny-p
erl all 1.002001-1 [37,0 kB]
Пол:3 https://download.astralinux.ru/astra/stable/1.7_x86-64/repository-main 1.7_x86-
↪64/main amd64
liblist-moreutils-
perl amd64 0.416-1 [63,7 kB]
Пол:4 https://download.astralinux.ru/astra/stable/1.7_x86-64/repository-main 1.7_x86-
↪64/main amd64
```

(continues on next page)

(продолжение с предыдущей страницы)

```
libregexp-assemble
-perl all 0.36-1 [85,3 kB]
Пол:5 https://download.astralinux.ru/astra/stable/1.7_x86-64/repository-main 1.7_x86-
↪64/main amd64
apt-file all 3.2.2
[38,5 kB]
Получено 295 kB за 0с (782 kB/s)
Выбор ранее не выбранного пакета libapt-pkg-perl.
(Чтение базы данных ... на данный момент установлено 211162 файла и каталога.)
Подготовка к распаковке .../libapt-pkg-perl_0.1.34_amd64.deb ...
Распаковывается libapt-pkg-perl (0.1.34) ...
Выбор ранее не выбранного пакета libexporter-tiny-perl.
Подготовка к распаковке .../libexporter-tiny-perl_1.002001-1_all.deb ...
Распаковывается libexporter-tiny-perl (1.002001-1) ...
Выбор ранее не выбранного пакета liblist-moreutils-perl.
Подготовка к распаковке .../liblist-moreutils-perl_0.416-1_amd64.deb ...
Распаковывается liblist-moreutils-perl (0.416-1) ...
Выбор ранее не выбранного пакета libregexp-assemble-perl.
Подготовка к распаковке .../libregexp-assemble-perl_0.36-1_all.deb ...
Распаковывается libregexp-assemble-perl (0.36-1) ...
Выбор ранее не выбранного пакета apt-file.
Подготовка к распаковке .../apt-file_3.2.2_all.deb ...
Распаковывается apt-file (3.2.2) ...
Настраивается пакет libapt-pkg-perl (0.1.34) ...
Настраивается пакет libexporter-tiny-perl (1.002001-1) ...
Настраивается пакет libregexp-assemble-perl (0.36-1) ...
Настраивается пакет liblist-moreutils-perl (0.416-1) ...
Настраивается пакет apt-file (3.2.2) ...
The system-wide cache is empty. You may want to run 'apt-file update'
as root to update the cache.
Обрабатываются триггеры для xserver-xorg-core
(2:1.20.14-1ubuntu1astra.se17) ...
update exec ids due to /usr/bin changed
Обрабатываются триггеры для man-db (2.8.5-2) ...
```

При этом утилита построила дерево зависимостей, сформировала список дополнительных пакетов для установки, рассчитала объем необходимого места для их загрузки и установки и в интерактивном режиме запросила подтверждение на установку этих пакетов. После получения подтверждения она скачала все необходимые пакеты, распаковала их и настроила. На последнем шаге были автоматически обновлены страницы в man-db.

Воспользуемся этой утилитой, чтобы найти пакет, в состав которого входит утилита pstree. Для этого выполним команду `apt-file search pstree`.

```
localadmin@astra:~$ apt-file search pstree
psmisc: /usr/bin/pstree
psmisc: /usr/bin/pstree.x11
psmisc: /usr/share/man/man1/pstree.1.gz
psmisc: /usr/share/man/man1/pstree.x11.1.gz
psmisc: /usr/share/pixmaps/pstree16.xpm
psmisc: /usr/share/pixmaps/pstree32.xpm
python-psutil-doc: /usr/share/doc/python-psutil-doc/examples/pstree.py
```

Как видим, этот пакет называется psmisc, и командой `apt show psmisc` мы можем вывести о нем дополнительную информацию:

```
localadmin@astra:~$ apt show psmisc
Package: psmisc
Version: 23.2-1
Priority: optional
Section: admin
Maintainer: Craig Small <csmall@debian.org>
Installed-Size: 647 kB
Depends: libc6 (>= 2.14), libselinux1 (>= 1.32), libtinfo6 (>= 6)
Replaces: procps (<< 1:1.2)
Homepage: http://psmisc.sf.net/
Download-Size: 125 kB
APT-Manual-Installed: no
APT-Sources: https://download.astralinux.ru/astra/stable/1.7_x86-64/repository-main 1.
↪7_x86-
64/main amd64 Packages
Description: utilities that use the proc file system
This package contains miscellaneous utilities that use the proc FS:
...
```

В опции search утилиты apt можно применять регулярные выражения, например, для поиска пакета веб-сервера nginx необходимо ввести команду `apt search ^ngnix$`. Сравните с выводом команды без метасимвола $ `apt search "^ngnix"`.

```
localadmin@astra:~$ apt search "^nginx$"
Сортировка... Готово
Полнотекстовый поиск... Готово
nginx/stable 1.22.1-9 amd64
 small, powerful, scalable web/proxy server
```

Теперь установим какой-нибудь пакет в систему, например, анализатор диска, указывающий на наиболее заполненные каталоги - утилиту `ncdu`, командой `sudo apt install ncdu`.

```
localadmin@astra:~$ sudo apt install ncdu
[sudo] пароль для localadmin:
Чтение списков пакетов... Готово
Построение дерева зависимостей
Чтение информации о состоянии... Готово
Следующие НОВЫЕ пакеты будут установлены:
 ncdu
Обновлено 0 пакетов, установлено 1 новых пакетов, для удаления отмечено 0 пакетов, и␣
↪681 пакетов
не обновлено.
Необходимо скачать 45,3 kB архивов.
После данной операции объём занятого дискового пространства возрастёт на 102 kB.
Пол:1 https://download.astralinux.ru/astra/stable/1.7_x86-64/repository-extended 1.7_
↪x86-64/main
amd64 ncdu amd64 1.1
5.1-1 [45,3 kB]
Получено 45,3 kB за 0с (232 kB/s)
Выбор ранее не выбранного пакета ncdu.
(Чтение базы данных ... на данный момент установлено 211287 файлов и каталогов.)
```

(continues on next page)

(продолжение с предыдущей страницы)

```
Подготовка к распаковке .../ncdu_1.15.1-1_amd64.deb ...
Распаковывается ncdu (1.15.1-1) ...
Настраивается пакет ncdu (1.15.1-1) ...
Обрабатываются триггеры для man-db (2.8.5-2) ...
Обрабатываются триггеры для xserver-xorg-core (2:1.20.14-1ubuntu1astra.se17) ...
update exec ids due to /usr/bin changed
```

Теперь мы можем воспользоваться ей, чтобы оценить размер каталогов в домашней директории командой `ncdu ~`.

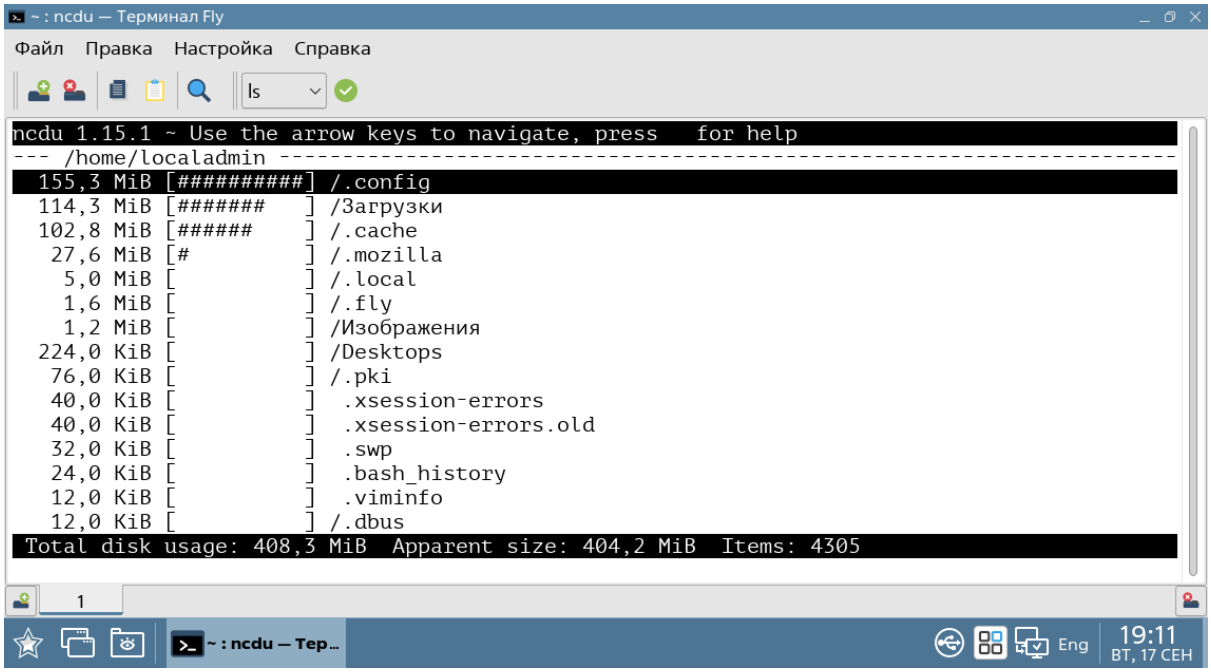


рис. 1.66 – Размер каталогов в домашней директории в ncdu

Переустановить эту утилиту (например, при повреждении её файлов) можно командой `sudo apt reinstall ncdu`.

```
localadmin@astra:~$ sudo apt reinstall ncdu
Чтение списков пакетов... Готово
Построение дерева зависимостей
. . .
Распаковывается ncdu (1.15.1-1) на замену (1.15.1-1) ...
Настраивается пакет ncdu (1.15.1-1) ...
Обрабатываются триггеры для man-db (2.8.5-2) ...
Обрабатываются триггеры для xserver-xorg-core (2:1.20.14-1ubuntu1astra.se17) ...
update exec ids due to /usr/bin changed
```

Решив, что эта утилита нам больше не потребуется, её можно удалить командой `sudo apt remove ncdu`.

```
localadmin@astra:~$ sudo apt remove ncdu
Чтение списков пакетов... Готово
```

(continues on next page)

(продолжение с предыдущей страницы)

```
Построение дерева зависимостей
Чтение информации о состоянии... Готово
Следующие пакеты будут УДАЛЕНЫ:
 ncdu
Обновлено 0 пакетов, установлено 0 новых пакетов, для удаления отмечено 1 пакетов, и␣
↪681 пакетов
не обновлено.
После данной операции объём занятого дискового пространства уменьшится на 102 kB.
Хотите продолжить? [Д/н] y
(Чтение базы данных ... на данный момент установлено 211293 файла и каталога.)
Удаляется ncdu (1.15.1-1) ...
Обрабатываются триггеры для man-db (2.8.5-2) ...
Обрабатываются триггеры для xserver-xorg-core (2:1.20.14-1ubuntu1astra.se17) ...
update exec ids due to /usr/bin changed
```

Утилита сообщит объем пространства, которое будет высвобождено после удаления пакета, и запросит подтверждение на удаление пакета.

Для вывода пакетов, которые могут быть обновлены, воспользуемся командой `apt list --upgradable` (а командой `apt list --installed` можно вывести список всех установленных пакетов в системе):

```
localadmin@astra:~$ apt list --upgradable
Вывод списка... Готово
acpi/stable,stable 1.7-1.2 amd64 [может быть обновлён с: 1.7-1.1]
ark/stable,stable 4:22.12.3-1astra1 amd64 [может быть обновлён с: 4:22.08.1-1astra2]
astra-admin-events/stable,stable 1.0.14+ci21 amd64 [может быть обновлён с: 0.0.1+ci2]
. . .
xserver-xorg/stable,stable 1:7.7+23astra3 amd64 [может быть обновлён с: 1:7.
↪7+23astra1]
xterm/stable,stable 379-1astra3 amd64 [может быть обновлён с: 366-1astra1
```

Команда `sudo astra-update -A -r -T` обновит все пакеты в системе автоматически. Если по каким-либо причинам вам необходимо обновить не всю систему, а только отдельный пакет, то можно воспользоваться командой `sudo apt-get install --only-upgrade astra-admin-events`.

```
localadmin@astra:~$ sudo apt-get install --only-upgrade astra-admin-events
Чтение списков пакетов... Готово
Построение дерева зависимостей
Чтение информации о состоянии... Готово
Будут установлены следующие дополнительные пакеты:
 astra-event-diagnostics libastraevents python-dateutil python-six python3-
↪astraevents python3-
distutils
 python3-lib2to3 python3-pkg-resources python3-setuptools syslog-ng-mod-astra
Предлагаемые пакеты:
 python-setuptools-doc
Следующие НОВЫЕ пакеты будут установлены:
 astra-event-diagnostics libastraevents python-dateutil python-six python3-
↪astraevents python3-
distutils
 python3-lib2to3 python3-pkg-resources python3-setuptools
Следующие пакеты будут обновлены:
```

(continues on next page)

(продолжение с предыдущей страницы)

```
 astra-admin-events syslog-ng-mod-astra
Обновлено 2 пакетов, установлено 9 новых пакетов, для удаления отмечено 0 пакетов, и␣
↪679 пакетов не  обновлено.
Необходимо скачать 2 019 kB архивов.
После данной операции объём занятого дискового пространства возрастёт на 4 959 kB.
Хотите продолжить? [Д/н] y
. . .
astra-event-diagnostics.service is a disabled or a static unit, not starting it.
Настраивается пакет python3-lib2to3 (3.7.3-1) ...
Настраивается пакет python3-distutils (3.7.3-1) ...
Настраивается пакет python3-setuptools (40.8.0-1) ...
Настраивается пакет syslog-ng-mod-astra (0.3.20+ci4) ...
Устанавливается новая версия файла настройки /etc/astra-syslog.conf ...
Устанавливается новая версия файла настройки /etc/audit/rules.d/astra-syslog.rules ...
Устанавливается новая версия файла настройки /etc/dbus-1/system.d/ru.astralinux.
↪events.conf ...
Устанавливается новая версия файла настройки /etc/syslog-ng/conf.d/mod-astra-unsigned-
↪binary-
file.conf ...
Настраивается пакет libastraevents (0.1.3) ...
Настраивается пакет python3-astraevents (0.1.3) ...
Настраивается пакет astra-admin-events (1.0.14+ci21) ...
Обрабатываются триггеры для libc-bin (2.28-10+deb10u1+ci202207192202+astra4) ...
Обрабатываются триггеры для xserver-xorg-core (2:1.20.14-1ubuntu1astra.se17) ...
update exec ids due to /usr/bin changed
Обрабатываются триггеры для dbus (1.12.20-2+astra.se7+ci7) ...
```

Теперь обновим систему командой `sudo astra-update -A -r -T` (этот процесс может занять некоторое время, и вы пока можете перейти к прочтению следующего менеджера пакетов aptitude).

```
localadmin@astra:~$ sudo astra-update -A -r -T
Параметры запуска: -A -r

Источники из /etc/apt/sources.list:
deb https://download.astralinux.ru/astra/stable/1.7_x86-64/repository-main/ 1.7_x86-
↪64 main
contrib non-free
deb https://download.astralinux.ru/astra/stable/1.7_x86-64/repository-update/ 1.7_x86-
↪64 main
contrib non-free
deb https://download.astralinux.ru/astra/stable/1.7_x86-64/repository-base/ 1.7_x86-
↪64 main
contrib non-free
deb https://download.astralinux.ru/astra/stable/1.7_x86-64/repository-extended/ 1.7_
↪x86-64 main
contrib non-free

Обнаруженные компоненты: contrib main non-free
Отключение функций безопасности не требуется
Игн:1 https://download.astralinux.ru/astra/stable/1.7_x86-64/repository-main 1.7_x86-
↪64 InRelease
Сущ:2 https://download.astralinux.ru/astra/stable/1.7_x86-64/repository-update 1.7_
↪x86-64
InRelease
Сущ:3 https://download.astralinux.ru/astra/stable/1.7_x86-64/repository-base 1.7_x86-
```

(continues on next page)

(продолжение с предыдущей страницы)

```
↪64 InRelease
Сущ:4 https://download.astralinux.ru/astra/stable/1.7_x86-64/repository-extended 1.7_
↪x86-64
InRelease
Сущ:5 https://download.astralinux.ru/astra/stable/1.7_x86-64/repository-main 1.7_x86-
↪64 Release
Чтение списков пакетов... Готово
Чтение списков пакетов... Готово
Построение дерева зависимостей
Чтение информации о состоянии... Готово
Расчёт обновлений... Готово
Следующие пакеты устанавливались автоматически и больше не требуются:
 coinor-libcbc3 coinor-libcgl1 coinor-libclp1 coinor-libcoinmp1v5 coinor-
↪libcoinutils3v5 coinor-
libosi1v5
 libepubgen-0.1-1 libetonyek-0.1-1 libgtkspell0 libkdecorations2private9␣
↪libkf5screen7 libplacebo7
libpoppler118
 libqxp-0.0-0 libstaroffice-0.0-0 libzmf-0.0-0 linux-headers-5.15.0-33-hardened linux-
↪headers-
5.15.0-33-lowlatency
Для их удаления используйте «sudo apt autoremove».
Следующие пакеты будут УДАЛЕНЫ:
 kmix
Следующие НОВЫЕ пакеты будут установлены:
 ant ant-contrib ant-optional astra-event-viewer ca-certificates-java default-jre␣
↪default-jre-
headless
. . .
 xserver-xorg-input-all xserver-xorg-input-evdev xserver-xorg-input-synaptics xserver-
↪xorg-video-
all
 xserver-xorg-video-amdgpu xserver-xorg-video-ati xserver-xorg-video-fbdev xserver-
↪xorg-video-
radeon xterm
Обновлено 679 пакетов, установлено 196 новых пакетов, для удаления отмечено 1 пакетов,
↪ и 0 пакетов
не обновлено.
Необходимо скачать 1 512 MB архивов.
После данной операции объём занятого дискового пространства возрастёт на 1 171 MB.
Пол:1 https://download.astralinux.ru/astra/stable/1.7_x86-64/repository-update 1.7_
↪x86-64/main
amd64 base-files amd64
10.3.astra11 [55,5 kB]
Пол:2 https://download.astralinux.ru/astra/stable/1.7_x86-64/repository-update 1.7_
↪x86-64/main
amd64 bsdutils amd64 1
:2.33.1-0.1.astra4se4 [121 kB]
. . .
Обновление успешно установлено
Для применения изменений в ядре или модулях нужна перезагрузка
Лог сохранен в /var/log/astra_update_202308151047.log
```

15.6.2 Менеджер пакетов aptitude

Менеджер пакетов `aptitude` позволяет выполнять те же операции с пакетами, что и `apt`, но обладает псевдографическим интерфейсом и поддерживает работу с мышью. В то же время вы можете использовать и командную строку, причем синтаксис команд такой же, как у `apt`.

Запускается приложение командой `aptitude`, а если необходимо не только просматривать информацию, но и устанавливать/обновлять/удалять пакеты, то `sudo aptitude`.

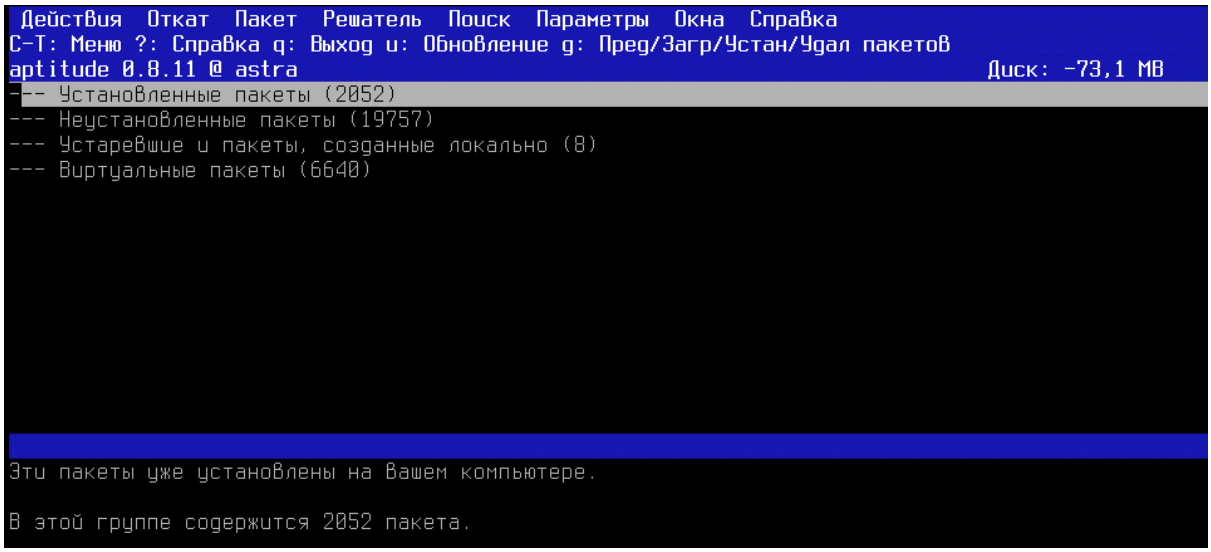


рис. 1.67 – Псевдо-графический менеджер пакетов aptitude

В главном окне приложения представлено дерево пакетов, свернутое до первого уровня. Для навигации по дереву используйте клавиши стрелок и мышь.

В верхней части окна расположено меню. Для его вызова с клавиатуры необходимо нажать сочетание `Ctrl + <Пробел>` или `F10`.

Для установки, обновления или удаления пакета, его необходимо пометить. Для этого необходимо воспользоваться меню «Пакет» или ввести:

- `+` (плюс) — для установки или обновления
- `-` (минус) — для удаления (аналог aptitude remove)
- `\_` (подчеркивание) — для полного удаления (аналог aptitude purge)

Если утилита была запущена без прав суперпользователя, а они потребовались в процессе работы, то в меню «Действия» можно выбрать пункт «Стать суперпользователем», и утилита будет перезапущена с повышенными привилегиями.

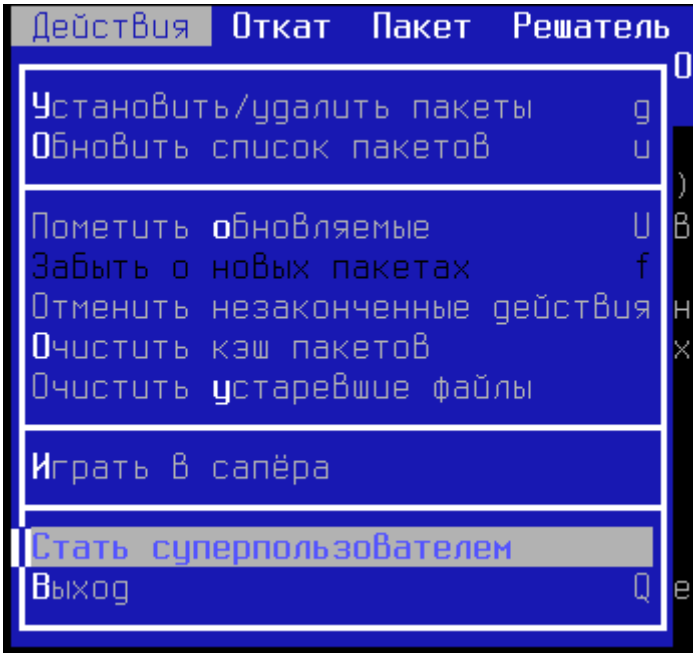


рис. 1.68 – Меню действия утилиты aptitude

Обновим локальный кэш пакетов нажатием клавиши u (аналог aptitude update) и найдем пакет alien. Для этого воспользуемся поиском (клавиша слеш «</>») или меню поиска.

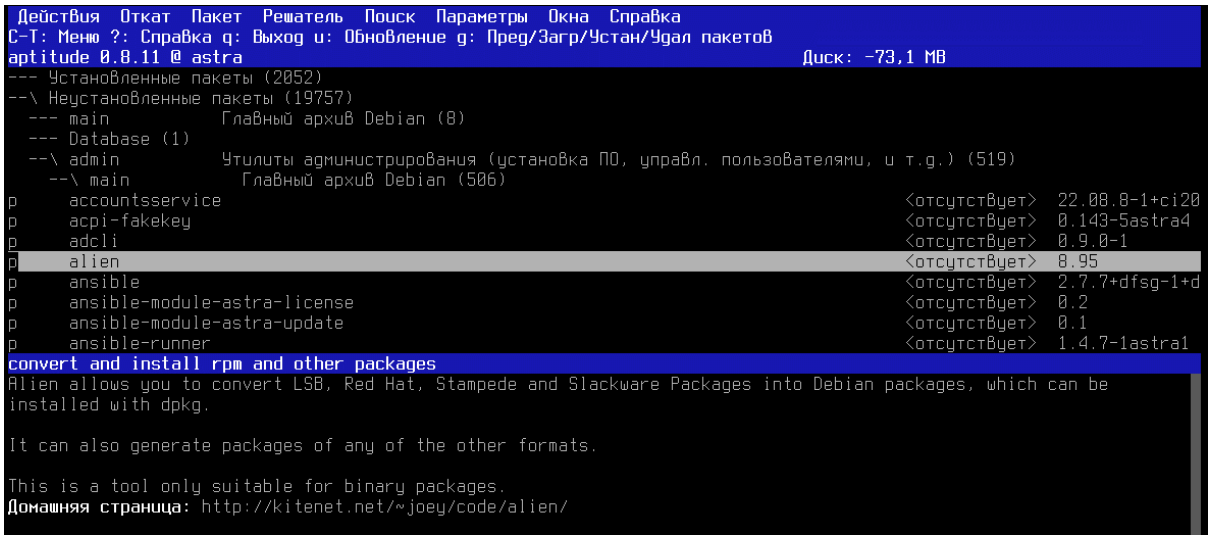


рис. 1.69 – Найденный пакет alien и информация о нем

Примечание: Утилита alien, входящая в базовый репозиторий, используется для конвертации rpm-пакетов (пакеты для семейства ОС, основанных на дистрибутивах Red Hat) в deb-пакеты и наоборот. Делать это стоит только в том случае, если deb-пакета для необходимой программы не существует. Конвертировать пакет гораздо проще, чем собирать его самому из исходников, однако стоит иметь в виду, что не каждый пакет может быть сконвертирован без ошибок — чем больше и

сложнее приложение, тем выше вероятность проблем.

Отметим этот пакет для установки символом плюс (Num+)» и установим его через меню *Действия → Установить/удалить пакеты(g)* или нажатием буквы (g).

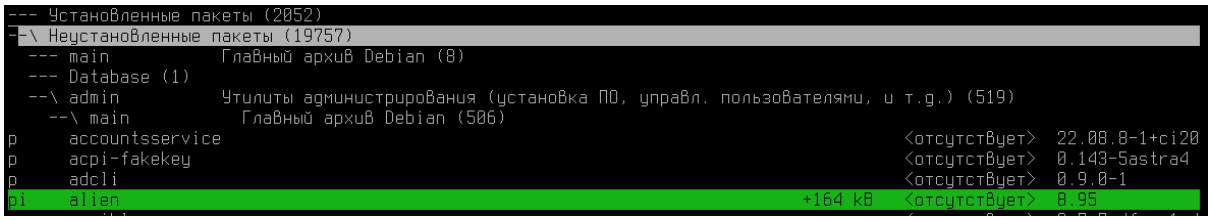


рис. 1.70 – После выделения пакет подсвечивается зеленым (для удаления красным)

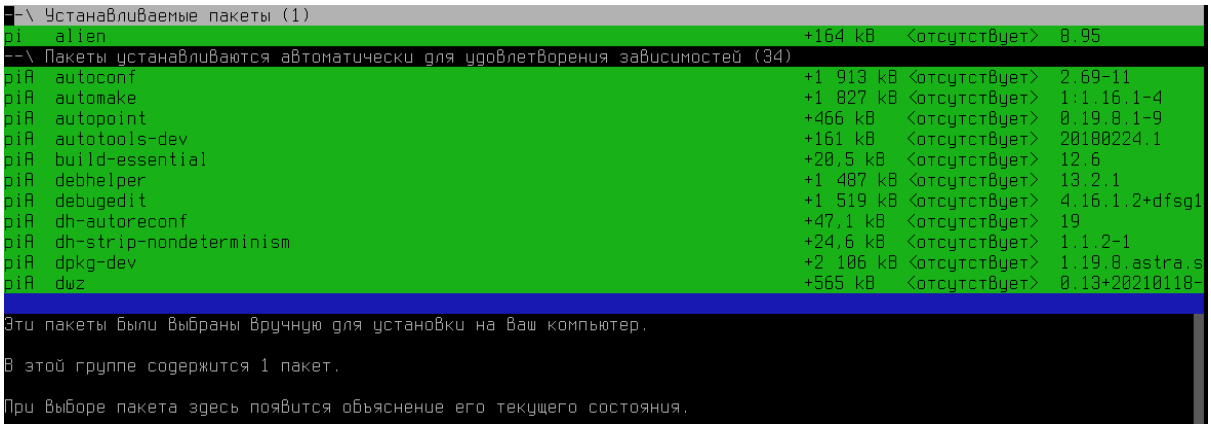


рис. 1.71 – Построение дерева зависимостей и предпросмотр

При этом произойдёт построение дерева зависимостей, и для подтверждения установки необходимо будет нажать (g) еще раз. В консоль будет выведен лог производимых операций.

Найдем пакет alien через поиск и убедимся, что он был установлен в системе.

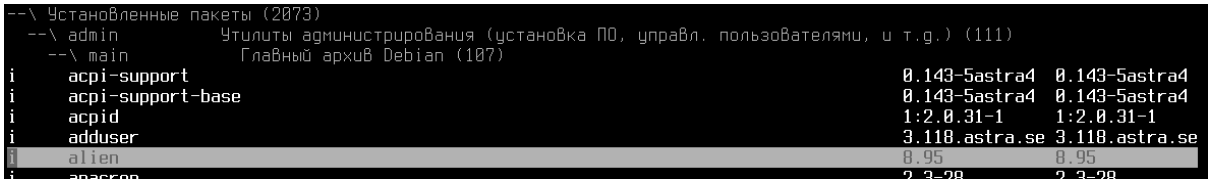


рис. 1.72 – Найден пакет alien в ветке установленные пакеты

Для просмотра информации необходимо открыть пакет двойным щелчком мыши или клавишей (Enter), при этом информация будет выведена в новом окне. Для закрытия окна необходимо нажать (q) или воспользоваться помощью меню.

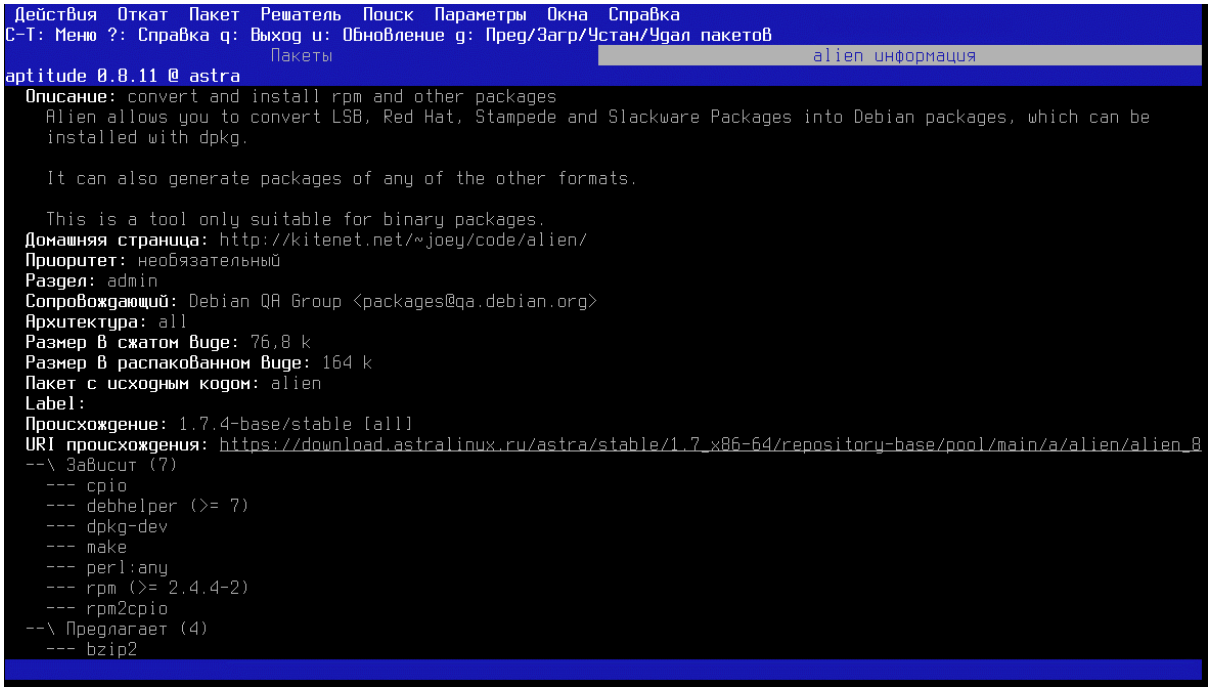


рис. 1.73 – Окно просмотра информации

Для перехода между окнами используйте клавиши F7 и F8.

Меню «Решатель» позволяет удобно разрешать конфликты зависимостей в интерактивном режиме.

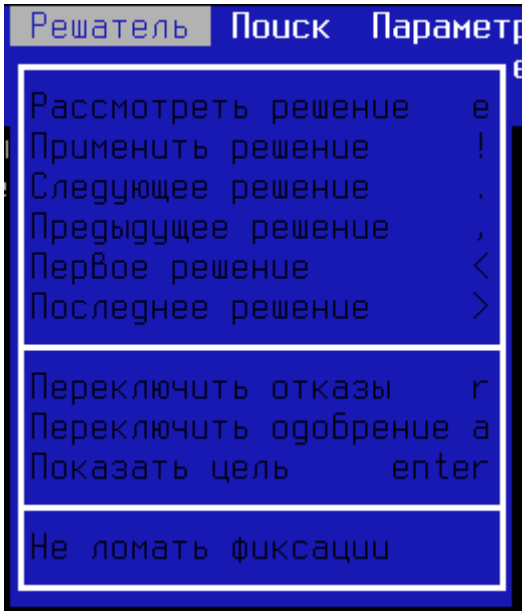


рис. 1.74 – Меню решатель конфликтов aptitude

Для закрытия программы необходимо нажать Q или воспользоваться командой из меню *Действия → Выход*.

15.6.3 Менеджер пакетов Synaptic

Менеджер пакетов Synaptic позволяет управлять ПО в Astra Linux через графический интерфейс. Запустить его можно из меню *Пуск → Системные → Менеджер пакетов Synaptic*. Для работы с ним нужны будут права суперпользователя.

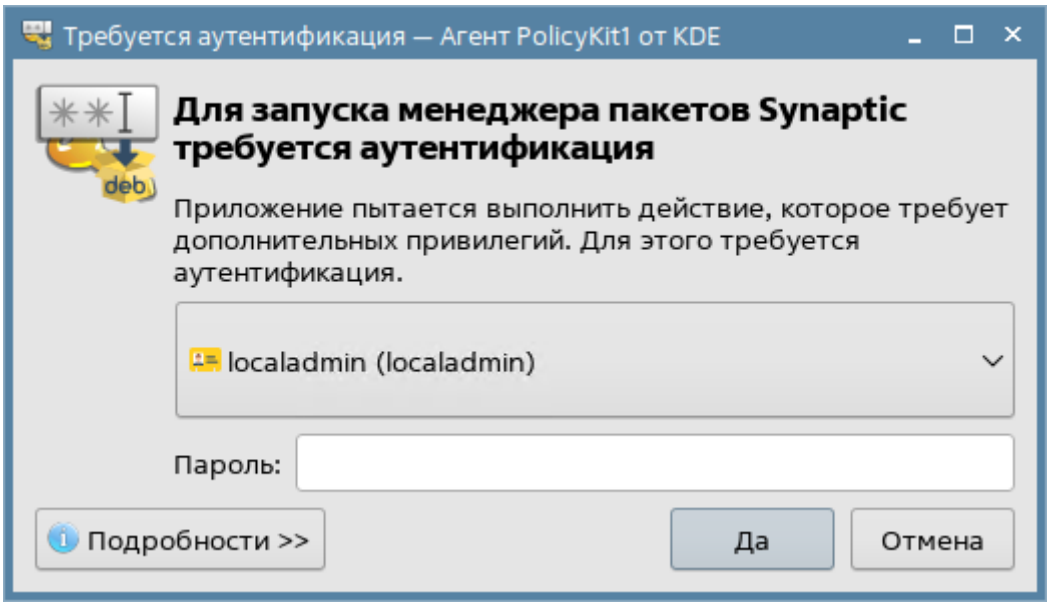


рис. 1.75 – Подтверждение прав суперпользователя

Приложение встретит вас краткой справкой по работе с ним. Но кто же ее читает :)

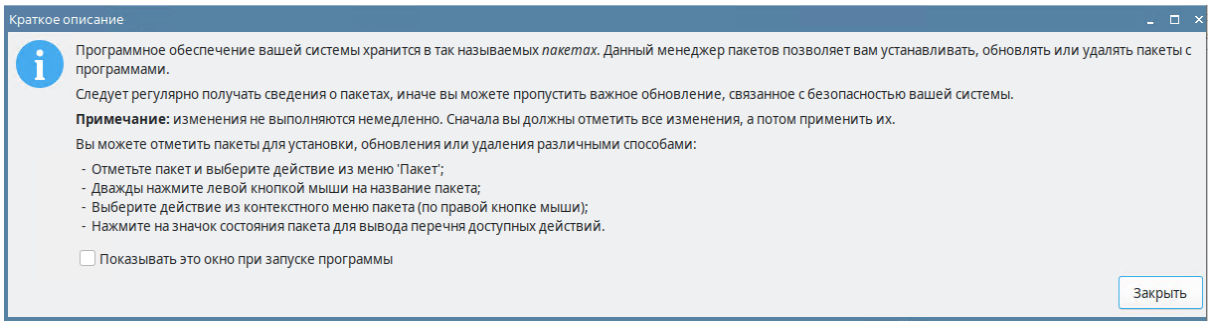


рис. 1.76 – Краткое описание менеджера пакетов

Synaptic имеет интуитивно понятный интерфейс работы. В левой части окна представлены категории пакетов (1) в зависимости от выбранного способа разделения (2). В правой части окна представлены список пакетов (3) и описание выделенного пакета (4). В нижней части окна расположена строка состояния.

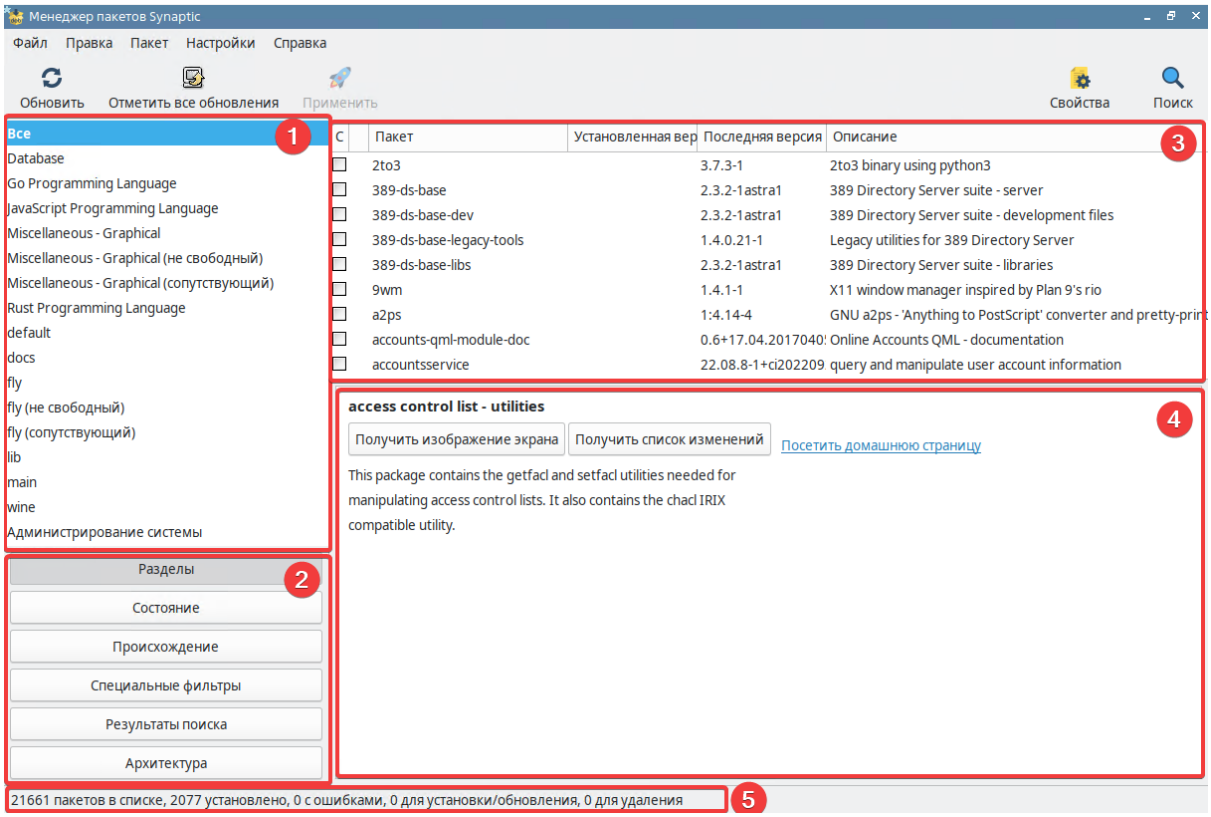


рис. 1.77 – Графический интерфейс Synaptic

Удалим установленный нами через aptitude пакет alien с помощью Synaptic.

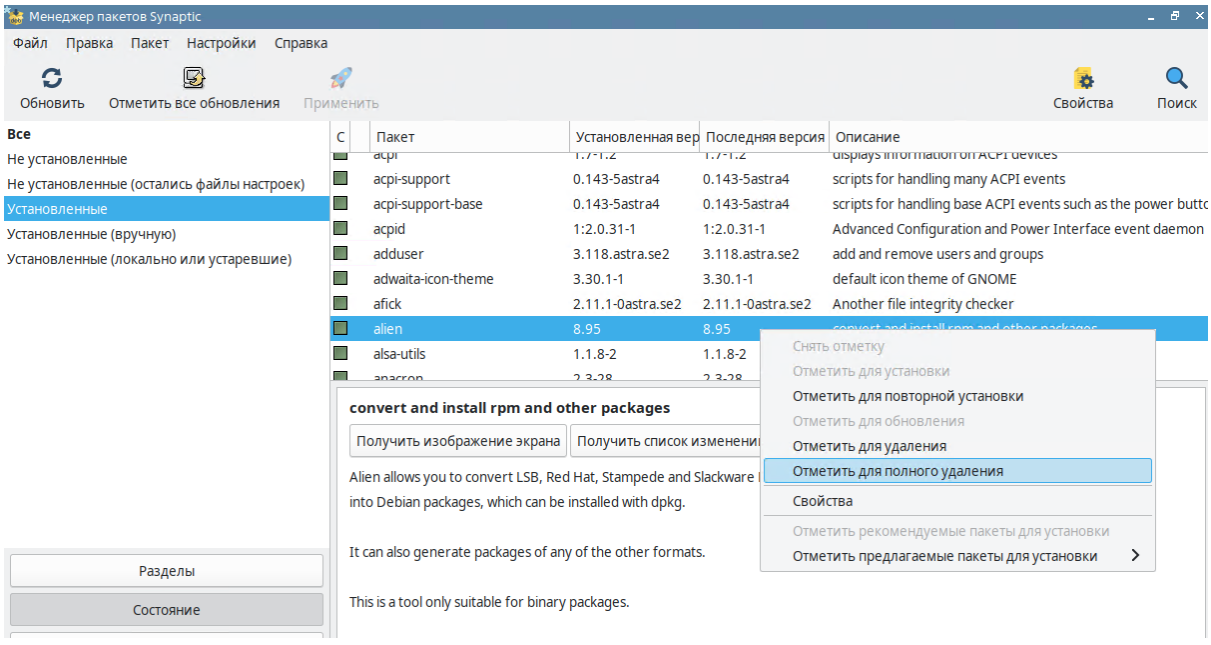


рис. 1.78 – Контекстное меню по выбранному пакету alien

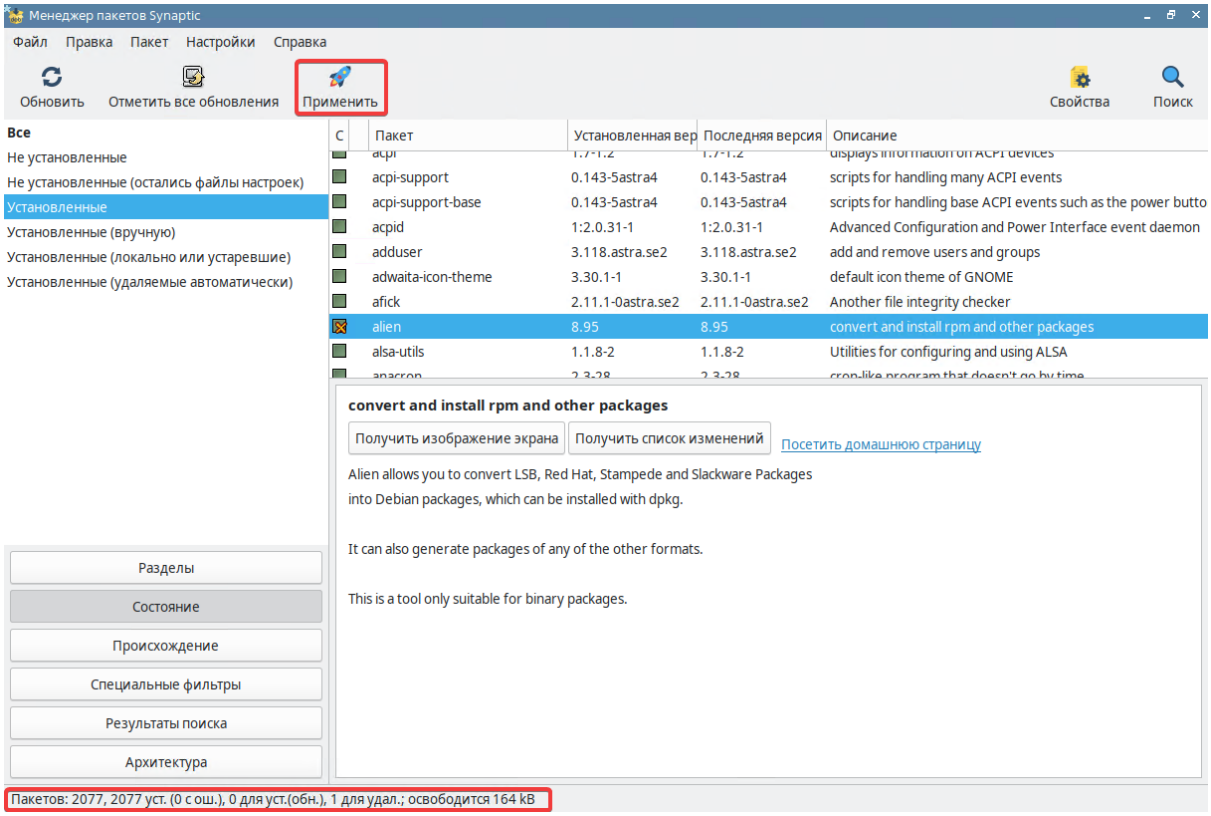


рис. 1.79 – Для применения удаления/установки нажимаем «Применить»

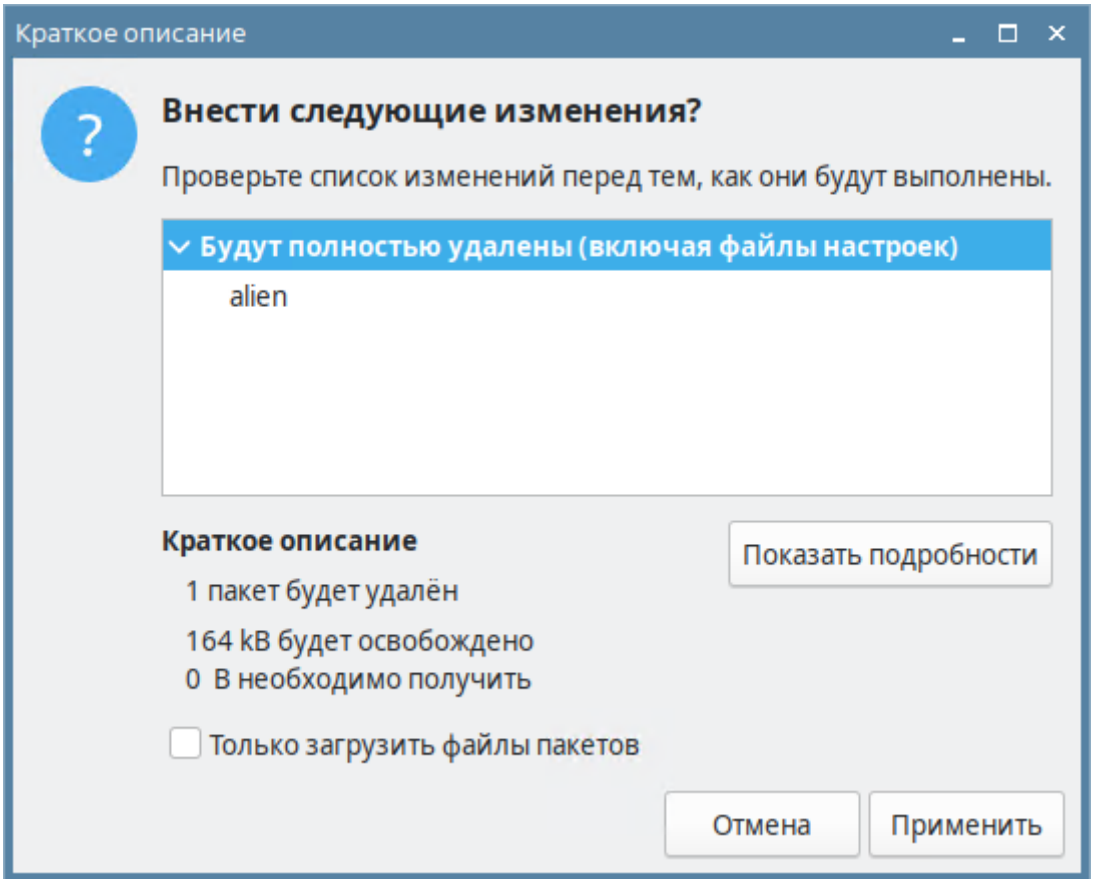


рис. 1.80 – Описание о то что будет сделано после подтверждения

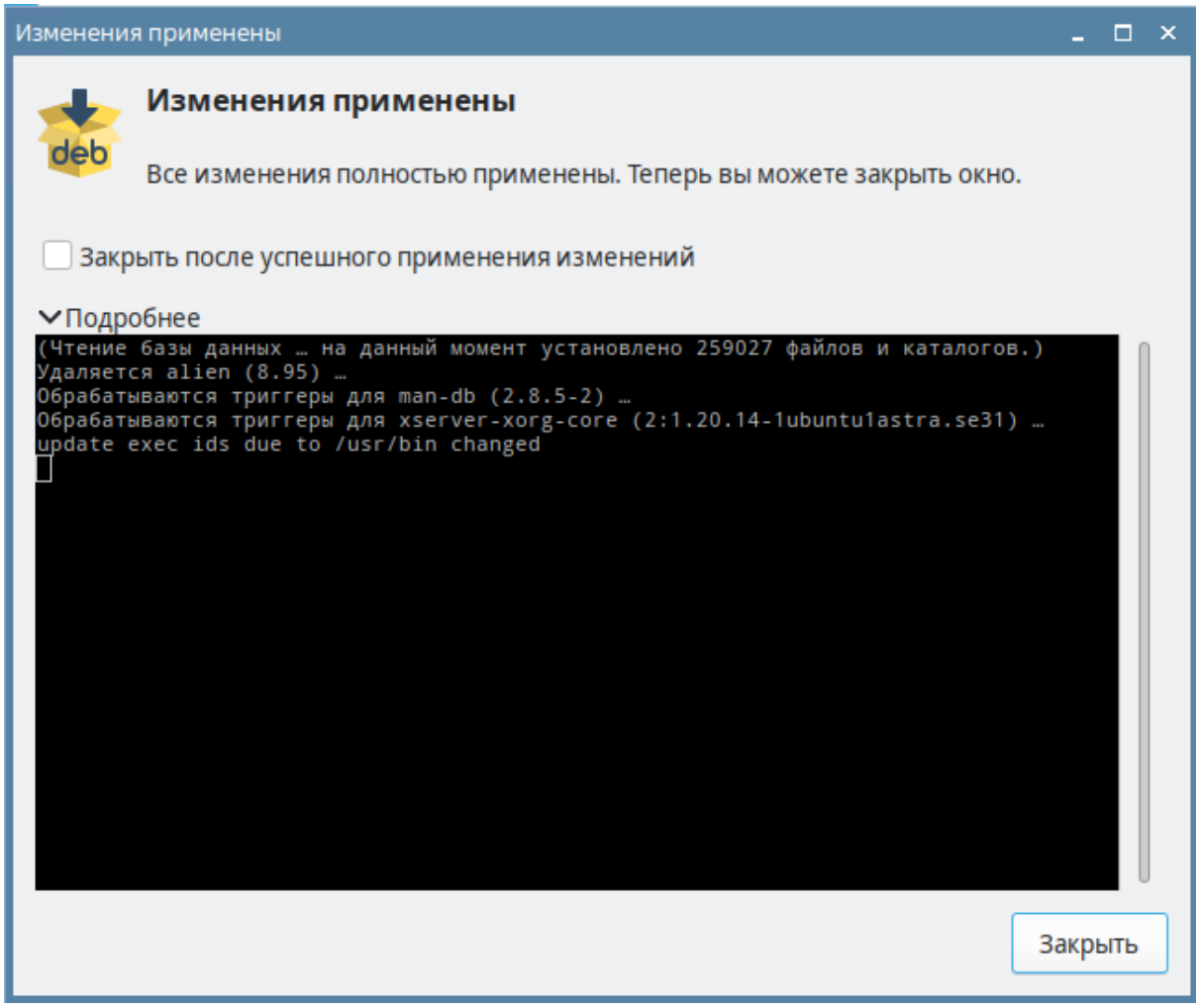


рис. 1.81 – Запуск API dpkg в терминале с просмотром о ходе выполнения

И удалим все пакеты из категории «Установленные (удаляемые автоматически)».

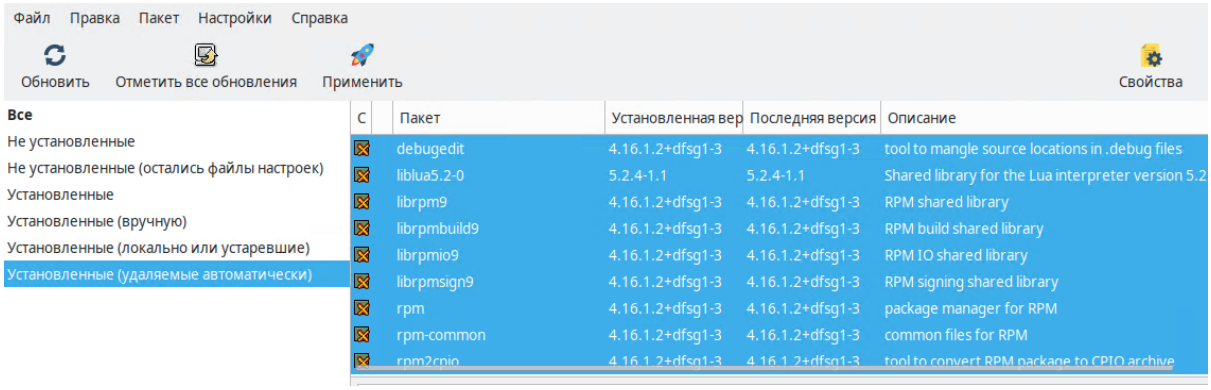


рис. 1.82 – Удалим пакеты которые можно удалить автоматически

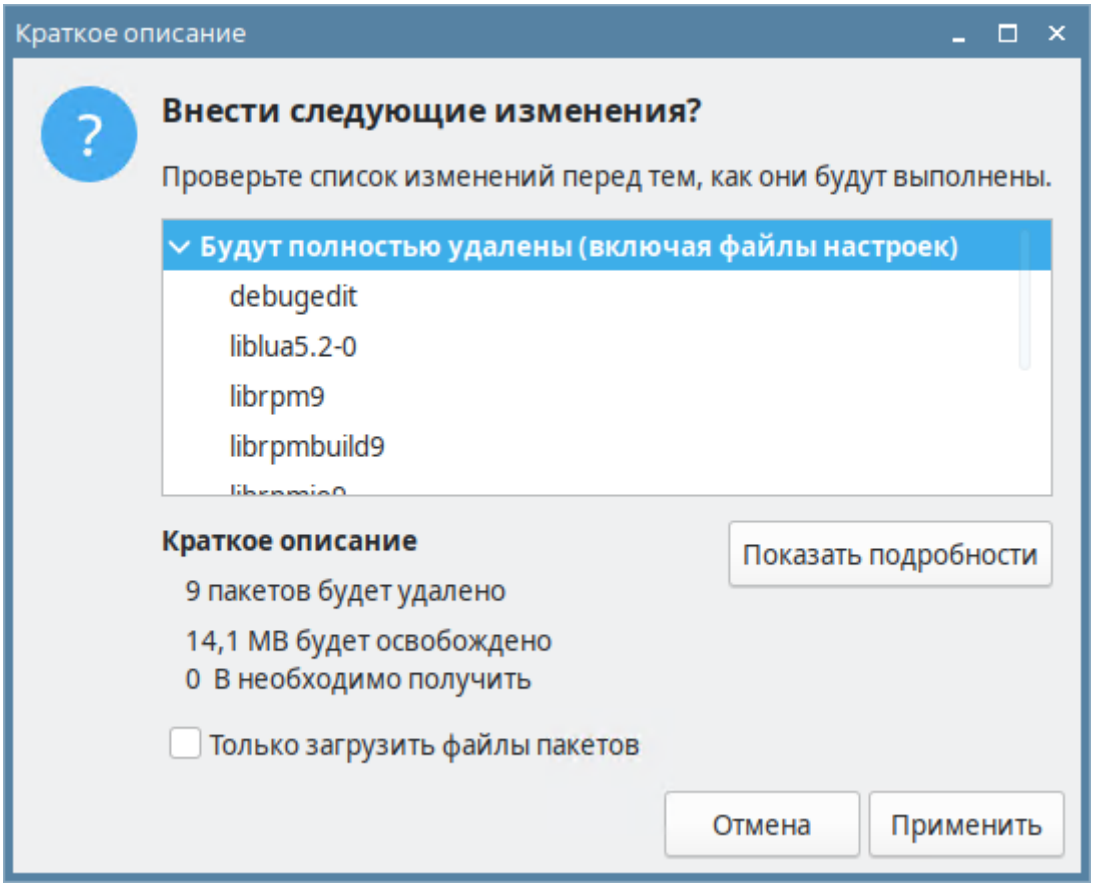


рис. 1.83 – Окно подтверждение с краткой информацией

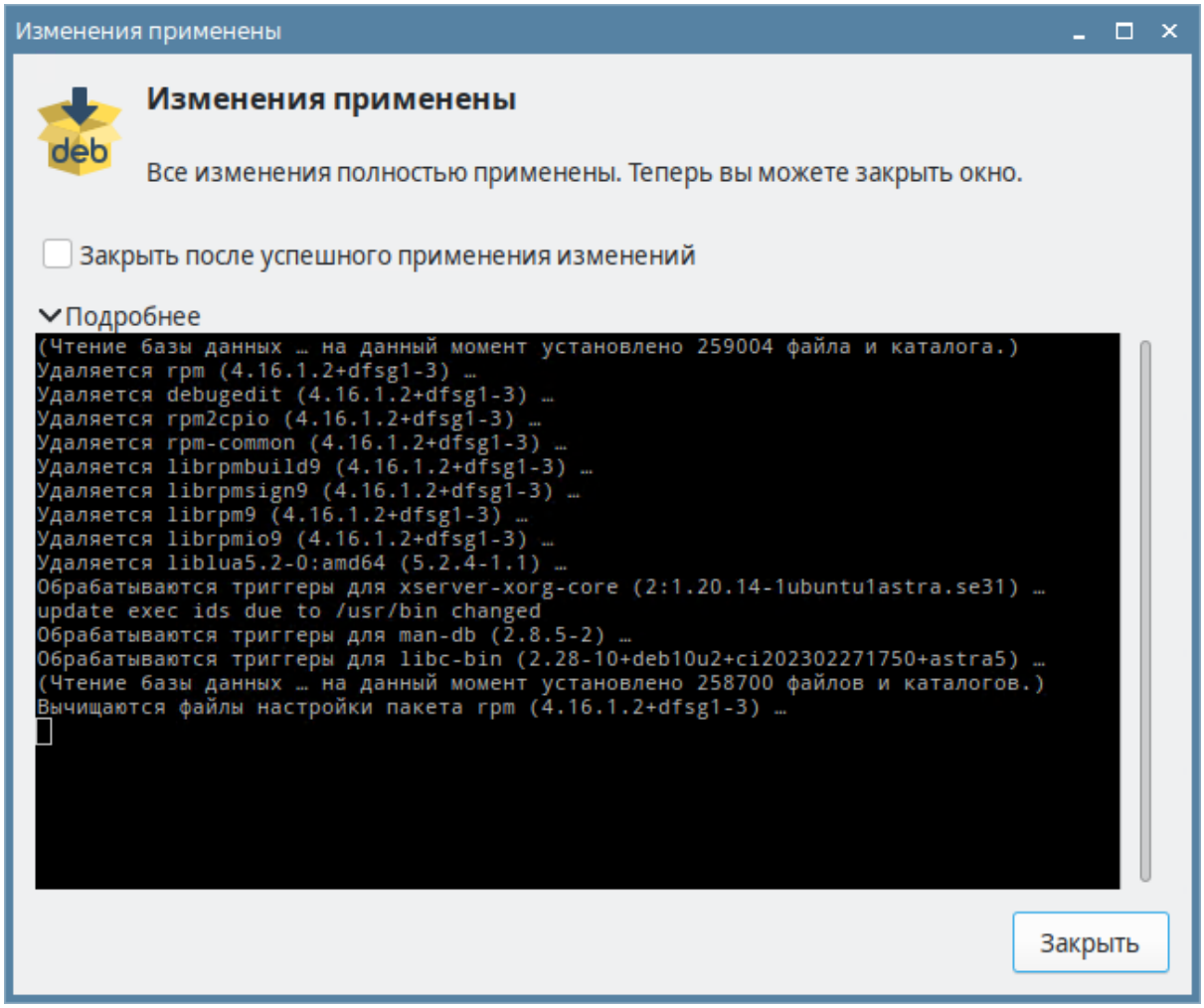


рис. 1.84 – Ход выполнения удаления в терминале

Источниками пакетов можно управлять в окне «Репозитории», вызвать которое можно из меню *Настройки* → *Репозитории* .

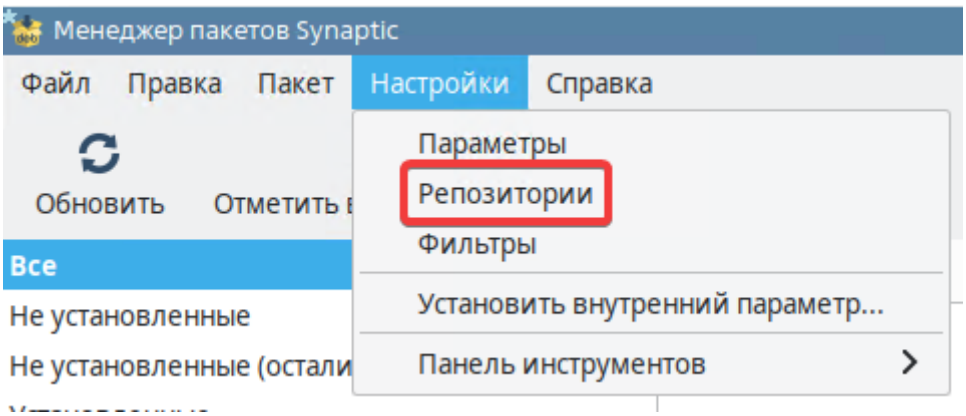


рис. 1.85 – Меню настроек Synaptic

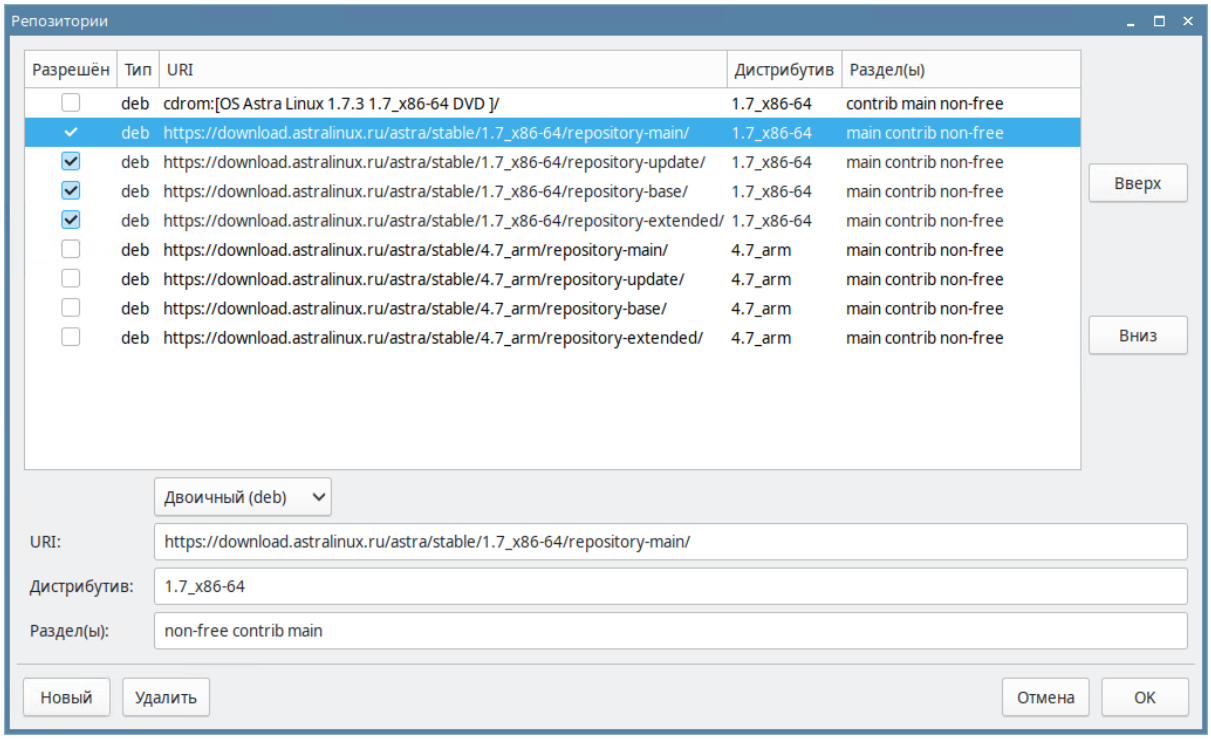


рис. 1.86 – Окно настроек репозиториев Synaptic

15.7 Внутреннее устройство deb-пакета

15.7.1 Обычные пакеты

Deb-пакет — это архив, созданный с помощью архиватора ar, внутри которого находятся не только файлы приложения, но и еще несколько служебных файлов, которые нужны для того, чтобы добавить гибкости.

Пакеты используют строгие правила именования, подчиняющиеся следующему синтаксису:

имя_пакета_версия_архитектура.deb

В качестве примера можно вспомнить пакет Яндекс диска, который мы уже загружали ранее `wget http://repo.yandex.ru/yandex-disk/yandex-disk_latest_amd64.deb`

С помощью утилиты file можно увидеть, что это Debian binary package:

```
localadmin@astra:~$ file yandex-disk_latest_amd64.deb
yandex-disk_latest_amd64.deb: Debian binary package (format 2.0)
```

Подробную информацию, включая зависимости, можно получить с помощью команды `apt info`:

```
localadmin@astra:~$ dpkg --info yandex-disk_latest_amd64.deb
 new Debian package, version 2.0.
 size 4796514 bytes: control archive=1740 bytes.
```

(continues on next page)

(продолжение с предыдущей страницы)

```
     51 байт(а),     1 строк      conffiles
    540 байт(а),    15 строк      control
    729 байт(а),    10 строк      md5sums
   1663 байт(а),    66 строк   *  postinst             #!/bin/sh
    371 байт(а),    24 строк   *  postrm               #!/bin/sh
     41 байт(а),     3 строк   *  prerm                #!/bin/sh
Package: yandex-disk
Version: 0.1.6.1080
...
```

15.7.2 Специальные типы deb-пакетов

Кроме обычных deb-пакетов существуют метапакеты и виртуальные пакеты.

Метапакеты — это пустые пакеты, которые только описывают зависимости. Они облегчают установку совместной группы программ, предварительно выбранных сопровождающим метапакета. Команда `apt install <метапакет>` автоматически установит все эти программы, используя зависимости метапакета. Пример такого пакета libreoffice.

Виртуальный пакет — это средство идентификации (своего рода группировки) реальных пакетов на основе общих логических критериев (например, предоставляемых услуг или совместимости со стандартной программой или уже существующим пакетом). Таких пакетов физически не существует. Для обозначения виртуального пакета используется поле Provides файла control, входящего в пакет.

15.7.3 Внутреннее устройство deb-пакета

Содержимое deb-пакета включает:

- Архив `control.tar.gz` содержит скрипты, которые используются при установке/удалении пакета, а также другие служебные файлы.
- Архив `data.tar.gz` содержит файлы программы для установки в систему. Их расположение в архиве соответствует положению в целевой системе. Например, файл `usr/share/doc/package/copyright` в deb-архиве после установки будет помещен в `/usr/share/doc/package/copyright`.
- Файл `debian-binary` содержит версию программы.
- Другие файлы, например, электронную цифровую подпись.

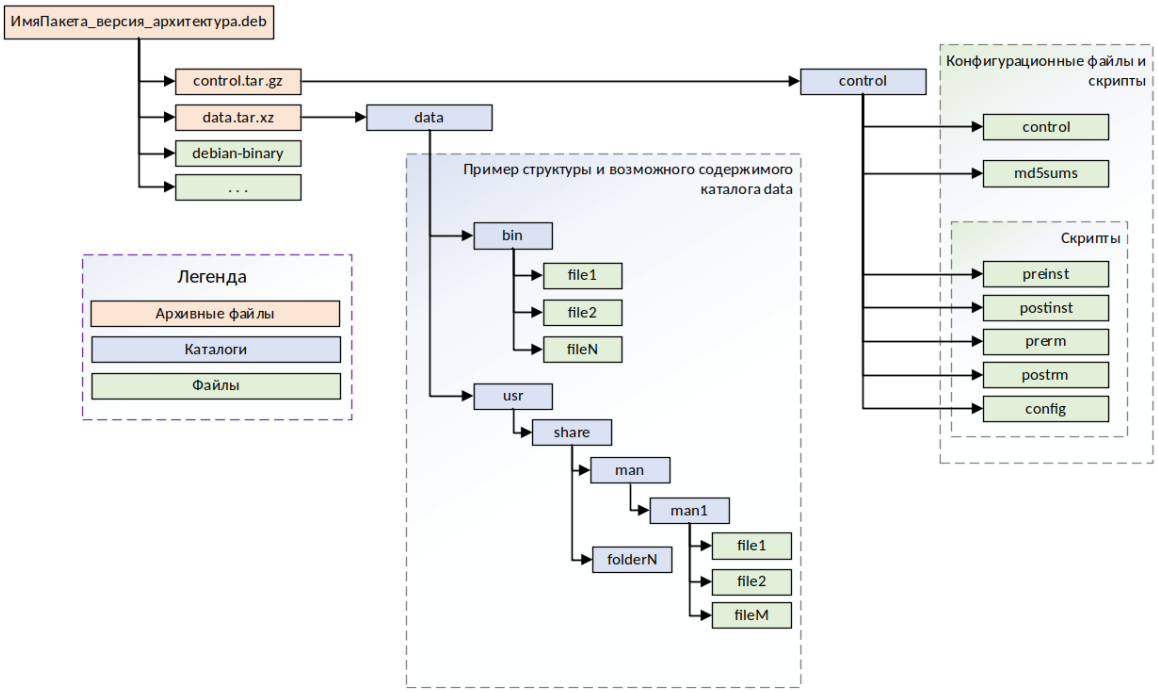


рис. 1.87 – Внутреннее устройство Deb-пакета

Командой `apt download acl` можно скачать текущую актуальную версию пакета acl и посмотреть его содержимое. Ниже представлен скриншот содержимого файла control в текстовом редакторе kate:

home > localadmin > packages > acl_2.2.53-4_amd64 > control > control

```
1  Package: acl
2  Version: 2.2.53-4
3  Architecture: amd64
4  Maintainer: Guillem Jover <guillem@debian.org>
5  Installed-Size: 206
6  Depends: libacl1 (= 2.2.53-4), libattr1 (>= 1:2.4.46-8), libc6 (>= 2.14)
7  Section: utils
8  Priority: optional
9  Multi-Arch: foreign
10 Homepage: https://savannah.nongnu.org/projects/acl/
11 Description: access control list - utilities
12  This package contains the getfacl and setfacl utilities needed for
13  manipulating access control lists. It also contains the chacl IRIX
14  compatible utility.
15
```

Содержимое файла control включает в себя следующие основные параметры:

- **Package** (обязательный) — имя пакета, под которым он будет зарегистрирован в системе.
- **Source** — имя и версия src-пакета, из которого собран данный deb-пакет (если есть таковой).

- **Version** — номер версии программы (пакета).
- **Section** (рекомендуемый) — имя раздела и секции, к которому принадлежит пакет, для его корректного размещения в репозитории в формате раздел/секция. Существуют 3 стандартных раздела: main, contrib и non-free. В каждом из этих разделов существует порядка 60 секций, например, admin, utils, devel, doc, base, net, mail и другие.
- **Priority** (рекомендуемый) - уровень важности пакета. Может принимать одно из 5 значений:
  - **Required** — пакет требуется в обязательном порядке для корректной работы любой Linux системы.
  - **Important** — пакет не влияет на работоспособность, но ожидается в любой Linux системе, например, man-db.
  - **Standard** — пакет входит в стандартный набор ПО для конкретного дистрибутива. Для разных дистрибутивов набор пакетов с уровнем важности Standard, как правило, разный.
  - **Optional** — пакеты, которые могут быть потенциально установлены в систему с данным дистрибутивом. Большинство пакетов относятся именно к этому уровню.
  - **Extra** — пакеты, которые конфликтуют с другими пакетами. Такие пакеты не устанавливаются по умолчанию пакетными менеджерами, но в случае необходимости такие пакеты может установить администратор, понимая при этом все связанные с этим риски.
- **Architecture (обязательный)** — архитектура, для которой данный пакет собран.
- **Depends** — перечень зависимостей между пакетами. Он включает в себя несколько видов зависимостей (могут быть использованы сразу несколько из них, каждая в новой строке):
  - **Depends** — перечень пакетов, перечисленных через запятую, от которых зависит данный пакет и без которых он не сможет нормально функционировать. Могут использоваться операторы сравнения для конкретизации необходимой версии пакета (<<,<=,=,>=,>>), например, **lib6 (>= 2.15)**, а также логическое «И» - запятая и логическое «ИЛИ» - вертикальная черта «|».
  - **Pre-Depends** — более строгая, чем depends, эти пакеты должны быть установлены еще до запуска скрипта установки данного пакета.
  - **Recommends** — перечень пакетов, которые значительно расширят функциональность, предлагаемую пакетом, но не являются обязательными для его работы.
  - **Suggests** — перечень пакетов, которые улучшают функциональность, но имеют второстепенное значение, как правило, отдельных утилит, или это может быть набор документации.

- **Enhances** — перечень пакетов, работу которых может улучшить данный пакет, своего рода противоположность suggests.
- **Conflicts** — перечень пакетов, которые не могут быть установлены одновременно с этим пакетом. Самой распространённой причиной является то, что оба пакета включают файлы с одинаковыми именами, предоставляют один и тот же сервис на одном и том же порту протокола управления передачей TCP.
- **Breaks** — схож с conflicts, но тут перечислены пакеты, работа которых (или их определенных версий) будет нарушена при установке данного пакета.
- **Provides** — имя виртуального пакета.

- **Installed-Size** — размер пакета, поле установки в систему.
- **Maintainer (обязательный)** — имя и координаты мейнтейнера, создавшего пакет.
- **Description (обязательный)** — подробное описание пакета, может представлять собой несколько абзацев текста.

Можно вывести информацию об установленном пакете acl в системе командой `apt show acl` и убедиться, что выводимая информация совпадает с содержимым файла control:

```
localadmin@astra:~$ apt show acl
Package: acl
Version: 2.2.53-4
Priority: optional
...
```

К необязательным файлам пакета относятся файл md5sums и скрипты для установки/удаления пакета. Файл md5sums содержит md5 хеши для всех файлов, кроме файлов, находящихся в архиве control.tar.xz. Данный файл необязателен для deb-пакета, однако программы верификации пакетов считают пакеты, не содержащие этот файл, ошибочными. Может использоваться некоторыми программами администрирования системы для верификации изменений в файловой системе.

Скрипты пакета могут вызываться при установке/удалении. Они расширяют возможности мейнтейнера по установке и настройке приложения. Например, этими скриптами могут создаваться/удаляться каталоги, в которых программа будет хранить свои временные файлы, может производиться добавление/удаление программы в меню вашего оконного менеджера, первичное конфигурирование программы и так далее.

Виды скриптов:

- **preinst** — скрипт, вызывающийся перед распаковкой пакета в файловую систему. Обычно при помощи этого скрипта пользователю задаются вопросы, позволяющие сконфигурировать программу прежде, чем начнется установка. Если скрипт завершится с кодом ошибки, то

пакет не будет установлен в систему.

- **postinst** — скрипт, вызывающийся после распаковки пакета. При помощи этого скрипта производится конфигурирование уже установленной программы. Конфигурирование происходит с использованием информации, полученной на предыдущем этапе от пользователя. Задавать пользователю вопросы из скрипта *postinst* не принято.

- **prerm** — скрипт, вызывающийся перед удалением пакета. Может задать пользователю некоторые вопросы, связанные с удалением. Например, вопрос о том, что делать с файлами журналов. Если скрипт завершится с кодом ошибки, пакет не будет удален.

- **postrm** — скрипт, вызывающийся после удаления файлов пакета.

- **config** — скрипт, вызывающийся на стадии конфигурирования и переконфигурирования пакета. Это дополнительный скрипт, который используется системой *Debconf*. Вызывается перед скриптом *postinst*, используется для того, чтобы задать пользователю вопросы в системе диалогов *debconf*.

15.8 Репозитории Astra Linux

Интернет-репозитории Astra Linux предназначены для упрощения процедуры обновления ОС и при наличии доступа в Интернет позволяют выполнять установку и обновление пакетов с помощью стандартных установщиков пакетов.

Для версии ALSE x.7 доступны 4 репозитория:

- Основной репозиторий (main)

- Оперативные обновления основного репозитория (update)

- Базовый репозиторий (base, включает main и update)

- Расширенный репозиторий (extended)

Рассмотрим назначение, состав и параметры этих репозиториев:

- **Основной репозиторий (репозиторий установочного диска, main):**

  - Содержит основной состав продукта, реализующий все функциональные возможности и функции безопасности.

  - Проходит сертификационные испытания, соответствует сертифицированному изделию.

  - Поставляется на установочном диске и в виде индивидуальной ссылки в личном кабинете пользователя.

- **Оперативные обновления основного репозитория (update):**
  - Содержит кумулятивные изменения основного репозитория (main), соответствующие выпуску оперативного обновления.
  - Подлежит сертификации.
  - Размещается в публичном доступе при выпуске бюллетеня безопасности.
- **Базовый репозиторий (base):**
  - Содержит все пакеты main и update.
  - Пакеты со средствами разработки.
  - Пакеты, необходимые для сборки репозитория main.
  - Не подлежит сертификации (если рабочие места требуют аттестации, то во избежание установки лишних пакетов не используйте base вместо main+update).
  - Поставляется в виде индивидуальной ссылки в личном кабинете пользователя.
- **Расширенный репозиторий (extended):**
  - Содержит дополнительное ПО, которое может функционировать в среде Astra Linux.
  - Не подлежит сертификации и не дорабатывается для реализации функций безопасности. Пакеты из этого репозитория могут быть несовместимы с последними оперативными обновлениями основного/базового репозиториев.
  - Поставляется в виде индивидуальной ссылки в личном кабинете пользователя.

Примечание: Подробно о составе, назначении и о работе с интернет-репозиториями Astra Linux, вы можете прочитать в статьях на wiki.astralinux.ru по ссылкам в дополнительных материалах:

15.9 Создание собственных репозиториев

Собственный репозиторий может понадобиться для уменьшения сетевого трафика, при массовом обновлении компьютеров или в случае, если по тем или иным причинам, интернет-канал не доступен на машине.

Создать собственный репозиторий (и управлять им) можно как полностью вручную (создать всю необходимую иерархию каталогов и метаданных), так и воспользовавшись специальным приложе-

нием **reprepro**, которое само создаст репозиторий согласно предоставленному конфигурационному файлу.

В домене ALD Pro есть подсистема «Репозиторий», которая основана как раз на базе reprepro. Подробнее о ней будет рассказано в отдельном модуле курса.

15.10 Практика и тестирование

15.10.1 Практическая работа: Модуль 15. Управление ПО в Astra Linux

См. также *Требования, правила и цели выполнения практической работы*

15.10.1.1 Практические задания

Задание 1.

Изучить утилиту dpkg и deb пакет.

1. Скачайте пакет Яндекс диска.
2. Проверьте его информацию.
3. Установите пакет.
4. Просмотрите список установленных пакетов.
5. Удалите пакет и все его конфигурации.
6. Получите список пакетов, установленных в системе, и отфильтруйте с помощью утилиты grep.
7. Получите детализированную информацию об установленном пакете sudo.
8. Выведите список файлов установленного пакета coreutils.
9. Найдите, какому пакету принадлежит указанный файл /bin/chacl.

Задание 2.

Изучить менеджер пакетов apt.

1. Настройте репозитории apt в файле source.list.
2. Обновите кэш apt.
3. Посмотрите список доступных пакетов для обновления командой.

4. Для поиска пакетов по входящим в него файлам необходимо воспользоваться утилитой apt-file.

5. Обновите кэш утилиты apt-file.

6. Воспользуемся этой утилитой, чтобы найти пакет, в состав которого входит утилита pstree.

Задание 3.

Изучить менеджер пакетов aptitude.

1. Установите с помощью менеджера пакетов aptitude утилиты alien.

2. Удалите с помощью менеджера пакетов aptitude утилиты alien.

Задание 4.

Изучить менеджер пакетов Synaptic.

1. Найдите через поиск утилиту htop.

2. Установите с помощью менеджера пакетов Synaptic утилиту htop.

3. Удалите с помощью менеджера пакетов Synaptic утилиту htop.

4. Отключите репозиторий https://download.astralinux.ru/astra/stable/1.7_x86-64/repository-extended/

15.10.1.2 Ответы на практические задания (пошаговые инструкции)

Задание 1.

Изучить утилиту dpkg и deb пакет.

1. Скачайте пакет Яндекс диска.

```
localadmin@astra:~$ wget repo.yandex.ru/yandex-disk/yandex-disk_latest_amd64.deb
```

2. Проверьте его информацию.

```
localadmin@astra:~$ dpkg -I yandex-disk_latest_amd64.deb
```

3. Установите пакет.

```
localadmin@astra:~$ sudo dpkg -i yandex-disk_latest_amd64.deb
```

4. Просмотрите список установленных пакетов.

```
localadmin@astra:~$ dpkg -l
```

5. Удалите пакет и все его конфигурации.

```
localadmin@astra:~$ sudo dpkg -P yandex-disk
```

6. Получите список пакетов, установленных в системе, и отфильтруйте с помощью утилиты grep.

```
localadmin@astra:~$ dpkg -l | grep sudo
```

7. Получите детализированную информацию об установленном пакете sudo.

```
localadmin@astra:~$ dpkg -s sudo
```

8. Выведите список файлов установленного пакета coreutils.

```
localadmin@astra:~$ dpkg -L coreutils
```

9. Найдите, какому пакету принадлежит указанный файл /bin/chacl.

```
localadmin@astra:~$ dpkg -S /bin/chacl
```

Задание 2.

Изучить менеджер пакетов apt.

1. Настройте репозитории apt в файле source.list.

```
localadmin@astra:~$ sudo nano /etc/apt/sources.list
```

```
#deb cdrom:[OS Astra Linux 1.7.6 1.7_x86-64 DVD ]/ 1.7_x86-64 contrib main non-free

#deb https://download.astralinux.ru/astra/stable/1.7_x86-64/repository-main/ 1.7_x86-
↪64 main contrib non-free
#deb https://download.astralinux.ru/astra/stable/1.7_x86-64/repository-update/ 1.7_
↪x86-64 main contrib non-free

deb https://download.astralinux.ru/astra/stable/1.7_x86-64/repository-base/ 1.7_x86-
↪64 main contrib non-free
deb https://download.astralinux.ru/astra/stable/1.7_x86-64/repository-extended/ 1.7_
↪x86-64 main contrib non-free
```

2. Обновите кэш apt.

```
localadmin@astra:~$ sudo apt update
```

3. Посмотрите список доступных пакетов для обновления командой.

```
localadmin@astra:~$ apt list --upgradable
```

4. Для поиска пакетов по входящим в него файлам необходимо воспользоваться утилитой apt-file. Установите ее с помощью команды:

```
localadmin@astra:~$ sudo apt install apt-file
```

5. Обновите кэш утилиты apt-file.

```
localadmin@astra:~$ sudo apt-file update
```

6. Воспользуемся этой утилитой, чтобы найти пакет, в состав которого входит утилита pstree.

```
localadmin@astra:~$ apt-file search pstree
```

Задание 3.

Изучить менеджер пакетов aptitude.

1. Установите с помощью менеджера пакетов aptitude утилиты alien.
2. Удалите с помощью менеджера пакетов aptitude утилиты alien.

Задание 4.

Изучить менеджер пакетов Synaptic.

1. Найдите через поиск утилиту htop.
2. Установите с помощью менеджера пакетов Synaptic утилиту htop.
3. Удалите с помощью менеджера пакетов Synaptic утилиту htop.
4. Отключите репозиторий https://download.astralinux.ru/astra/stable/1.7_x86-64/repository-extended/

15.10.2 Тест: Модуль 15. Управление ПО в Astra Linux

Пожалуйста, ответьте на несколько вопросов, чтобы закрепить полученные знания. Правильные ответы для самопроверки вы сможете найти в конце списка. Обратите внимание, что у некоторых вопросов может быть несколько верных вариантов ответов.

15.10.2.1 Вопросы

Вопрос 1. Какое расширение у библиотек Astra Linux?

a. Расширение `dll`

b. Расширение `mod`

c. Расширение `ocx`

d. Расширение `so`

Вопрос 2. Какое расширение у инсталляционных пакетов Astra Linux?

a. Расширение `rpm`

b. Расширение `deb`

c. Расширение `msi`

d. Расширение `run`

Вопрос 3. Какая утилита позволяет создать deb-пакет?

a. Утилита `apt`

b. Утилита `aptitude`

c. Утилита `dpkg`

d. Утилита `tar`

Вопрос 4. Какие существуют репозитории в ALSE?

a. repository-base

b. repository-extra

c. repository-extended

d. repository-main

e. repository-update

Вопрос 5. В каком файле прописываются репозитории?

a. Файл `/etc/apt.file`

b. Файл `/etc/apt/sources.list`

c. Файл `/etc/sources.list`

d. Файл `/sources/apt.list`

Вопрос 6. Какая команда позволяет посмотреть информацию о пакете?

a. Команда `apt show имя_пакета`

b. Команда `dpkg -i имя_пакета`

c. Команда `apt имя_пакета.deb`

d. Команда `apt-get имя_пакета`

Вопрос 7. Какой менеджер пакетов устанавливает пакет и зависимые пакеты?

a. apt

b. aptitude

c. synaptic

d. dpkg

Вопрос 8. Какой менеджер пакетов имеет графический интерфейс?

a. apt

b. dpkg

c. synaptic

d. apt-file

Вопрос 9. Какая команда позволяет создать свой репозиторий?

a. Команда `alien`

b. Команда `dpkg -repo`

c. Команда `apt-get`

d. Команда `reprepro`

Ответы на вопросы

1. b, d

2. b
3. c
4. a, c, d, e
5. b
6. a
7. a, b, c
8. c
9. d

15.11 Заключение

Мы стремительно приближаемся к завершению первой части курса, которая больше про Linux, но, как известно, чем дальше в лес, тем больше дров, и сегодняшний модуль тому пример. Тема управления пакетами в Linux крайне сложная, но вместе с тем и очень важная, поэтому требует вдумчивой проработки.

В следующем модуле будет немного попроще. Мы узнаем, как устроен диспетчер печати в Linux. До встречи!

15.12 Дополнительные источники информации

- Windows Package Manager
- Яндекс браузер Корпоративный
- Консольный клиент Яндекс Диска для Linux
- Интернет-репозитории Astra Linux Special Edition x.7
- Репозитории Astra Linux Special Edition x.7: структура, особенности подключения и использования
- Ключевые компоненты расширенного репозитория Astra Linux Special Edition x.7
- Использование стороннего программного обеспечения в аттестованных информационных си-

стемах, функционирующих под управлением Astra Linux Special Edition

15.13 Обратная связь

Если остались вопросы, то их всегда можно задать в специальной теме.

Модуль 16. Работа с подсистемой печати

16.1 Введение

Из этого модуля вы узнаете, почему служба печати в Linux называется CUPS и причем здесь Стив Джобс. Мы посмотрим, из каких компонентов она состоит и как они взаимодействуют между собой, познакомимся с основными настройками и сценариями ее использования, не выходя за рамки рабочих станций. Ну, что... разрыв строки, возврат каретки, погнали!

16.2 Служба печати в Linux

Как известно, после внедрения электронного документооборота сотрудники меньше печатать не стали, поэтому хотим мы того или нет, а со службой печати в Linux придется разбираться.

В операционной системе Astra Linux подсистема печати реализована на базе CUPS от англ. Common UNIX Printing System — общая система печати для UNIX. Разработчики этого продукта хотели создать универсальный кроссплатформенный инструмент с модульной архитектурой, которая бы упростила разработку драйверов для принтеров, и, учитывая тот факт, что CUPS принят сейчас большинством современных дистрибутивов Linux, можно сказать, что у них все удалось.

Первый релиз CUPS вышел в конце 90-х и за несколько лет был принят сразу несколькими nix-подобными дистрибутивами, включая Mac OS X. На первое время компании Apple удалось согласовать с разработчиками ряд оговорок в лицензионном соглашении, которые позволили использовать этот продукт в коммерческой операционной системе, но в дальнейшем они договорились о расширении сотрудничества, и в 2007 году Стив Джобс не только презентовал свой первый iPhone, но и приобрел права на исходные коды CUPS.

Более десяти лет продукт развивался под зонтиком Apple, по причине чего вы будете довольно часто встречать упоминания о компании в документации, но с 2020 года отделился под новым брендом OpenPrinting. В наши дни Apple CUPS — это версия, которая поставляется с macOS/iOS, а ветку OpenPrinting CUPS продолжают развивать для всех остальных nix-подобных операционных систем.

16.2.1 Архитектура CUPS

Службы печати для Linux и Windows развивались независимо, поэтому нельзя сказать, что они напрямую заимствовали идеи друг у друга, но обе этих разработки опирались на идеи предшествующих решений, поэтому во многом похожи. Если смотреть крупноблочно, то CUPS состоит из следующих компонентов, см. рис. 1.88:

- **Диспетчер печати**. В роли диспетчера выступает служба cupsd. Она упорядочивает задания для отправки на печать и выполняет те же функции, что и Print Spooler в операционной системе Windows.

  Однако, если в Windows приложения взаимодействуют с диспетчером печати по API Win32, то в Linux вместо программных интерфейсов выступает сетевой протокол IPP (Internet Printing Protocol), который расширяет возможности привычного протокола HTTP для организации взаимодействия с принтерами.

  Локальные приложения обращаются к диспетчеру печати напрямую через unix-сокет `/run/cups/cups.sock`, не используя сетевой стек, а при обращении к общим принтерам компьютера извне используется порт 631/TCP или 443/TCP (в случае SSL).

- **Система фильтров**. Каждый фильтр представляет собой приложение для преобразования направленных на печать данных в понятный для принтера набор простых инструкций. Кроме фильтров общего назначения существуют специальные фильтры, учитывающие особенности конкретной модели устройства, которые распространяются через специальные драйверы CUPS.

  Если у старых матричных принтеров набор инструкций включал всего несколько команд, таких как «напечатать строку, сделать разрыв и перевести каретку», то в современных принтерах используют язык описания страниц PostScript, который поддерживает настолько большой набор инструкций, что сопоставим по сложности с языками программирования высокого уровня. К счастью, обычным администраторам знать этот язык не требуется, и он остается уделом разработчиков драйверов.

  Если же вы станете счастливым обладателем устройства, которое не поддерживает набор инструкций PostScript, то в этом случае CUPS автоматически задействует универсальный фильтр Ghostscript, который преобразует инструкции PostScript в растровое изображение и отправит уже его на печать.

- **Система передачи данных** или бэкенд. Этот компонент отвечает за доставку данных до конкретного принтера, подключенного к устройству напрямую через USB (а может быть даже COM/LPT) или по сети. Если принтер использует нестандартный способ передачи данных, то соответствующий программный код этого бэкенда может быть установлен через специальный

драйвер CUPS.

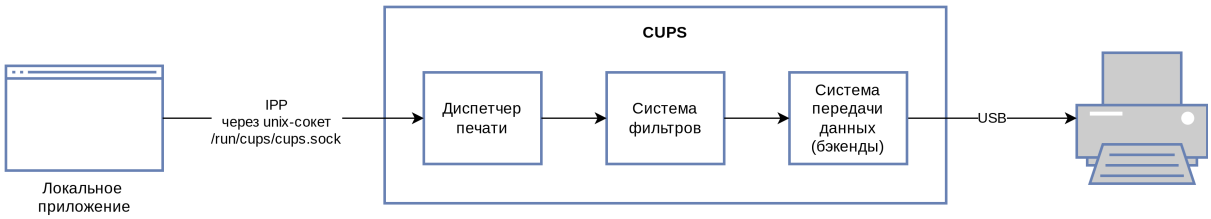


рис. 1.88 – Упрощенная схема печати из локального приложения

Следует понимать, что CUPS работает как клиент-серверное приложение даже при его использовании на рабочей станции. В этом случае клиентами выступают локальные приложения, которые обращаются к службе по локальному unix-сокету. Предлагаем подробнее разобрать работу CUPS на примере распечатки документа из LibreOffice.

16.2.1.1 Диспетчер печати

После того как мы в приложении LibreOffice нажали кнопку *Печать*, приложение подключается к сокету `/run/cups/cups.sock` и передает диспетчеру печати документ по протоколу IPP (HTTP).

Диспетчер печати — это серверное приложение cupsd, которое получает запросы по протоколу IPP и позволяет управлять принтерами, заданиями и другими сущностями системы печати. Протокол IPP использует POST-запросы протокола HTTP с типом контента «application/ipp» для расширения возможностей этого сетевого протокола прикладного уровня.

Диспетчер печати разработан как стандартный однопоточный серверный процесс. Он запускает внешние процессы для выполнения длительных операций, таких как печать документов, отправка уведомлений, сканирование устройств и драйверов, а также удалённый мониторинг принтеров. Служба работает от root, а внешние процессы выполняются из-под учётной записи lp с ограниченными правами доступа.

Максимальное количество одновременно поддерживаемых клиентов и заданий печати ограничивается только возможностями процессора, доступной оперативной памятью и свободными файловыми дескрипторами на сервере. Сам планировщик не накладывает каких-либо ограничений.

Все документы, которые отправляются на печать, в конечном итоге поступают к планировщику по протоколу IPP. Локальные приложения делают это через unix-сокет, а для внешних клиентов может быть открыт порт 631/TCP. При необходимости можно настроить безопасное подключение к службе CUPS по порту 443/TCP. Задания на печать, созданные планировщиком, проходят через фильтры и мониторы портов. Затем они преобразуются в подготовленные данные для печати и отправляются на принтеры или другие устройства. В обратном направлении передаются сведения о состоянии этих заданий.

16.2.1.2 Система фильтров

Для преобразования исходного файла задания в набор инструкций PostScript, понятный для принтера, используются так называемые фильтры. Фильтр считывает данные из стандартного ввода (или из файла, если он указан), выполняет обработку и передает дальше по цепочке через стандартный вывод. Дополнительные фильтры создаются разработчиками драйверов печатающих устройств и реализуют стандартный набор параметров, в том числе: имя принтера, идентификатор задания, имя пользователя, название задания, количество копий и другие параметры заданий.

В CUPS присутствуют фильтры для печати текста, PDF, HP-GL/2 и многих типов файлов изображений. CUPS также предоставляет фильтры драйверов принтеров для HP-PCL, ESC/P и для нескольких типов принтеров этикеток. Дополнительные фильтры регистрируются при установке драйверов через файлы mime.convs и PPD по следующим путям соответственно: `/usr/share/cups/mime/mime.convs` и `/etc/cups/ppd/`.

В рамках фильтров отдельно выделяют так называемые мониторы портов, но этот термин будет правильнее перевести как наставники портов. Эти специальные фильтры позволяют выполнить дополнительное форматирование данных для передачи конкретному устройству печати по конкретному каналу связи. По умолчанию доступно два фильтра:

- **монитор порта bcp**, который поддерживает протокол двоичной связи PostScript (*англ. Binary Communications Protocol, BCP).*
- **монитор порта tbcp**, который поддерживает протокол двоичной связи с тегами PostScript (*англ. Tagged Binary Communications Protocol, TBCP*).

Установка драйверов CUPS обычно выполняется с помощью стандартного пакетного менеджера APT, например, для управления устройствами Hewlett Packard предназначен пакет hplip. Некоторые драйвера, например, для принтеров Canon устанавливаются с помощью инсталляционных Bash-скриптов, см. страницу на wiki.astralinux.ru.

16.2.1.3 Система передачи данных (бэкенды)

Набор инструкций для принтера готов, осталось только передать их на принтер с учетом того, как он подключен к службе печати. И здесь в игру вступает бэкенд.

Бэкенд (как фильтры) получает информацию через входящий поток данных и передает их принтеру по установленному протоколу. Дополнительные типы бэкендов могут быть добавлены по мере необходимости через установку драйверов, а по умолчанию доступны:

- **USB** — подключение принтера через USB-порт.
- **IPP (Internet Printing Protocol)** — стандартный протокол для обмена данными между клиен-

том и сервером печати.

- **AppSocket (JetDirect)** — обеспечивает связь между приложением и устройством печати через порт 9100/TCP.
- **DNS-SD (mDNS)** и **SNMP** — механизмы обнаружения и управления принтерами в сети с использованием протоколов Zeroconf и Simple Network Management Protocol соответственно.
- **LPD (Line Printer Daemon)** — устаревший протокол печати.

В конечном счете наш документ попал на принтер и был распечатан, проделав большой путь от текста в приложении до текста на бумаге. Уточним несколько промежуточных этапов.

16.2.1.4 Файлы конфигураций

Диспетчер печати в своей работе использует настройки из конфигурационных файлов, которые находятся в каталоге `/etc/cups/`:

- Файл `cupsd.conf` – здесь хранятся основные настройки CUPS, например, какой порт прослушивается, уровень логирования, включена ли ротация логов и т.д.
- Файл `classes.conf` – группы (классов) принтеров. Группы принтеров используются для объединения нескольких принтеров в пул устройств, которые будут обслуживать одну очередь заданий печати. При отправке печати на группу принтеров CUPS начнет балансировать нагрузку (чередование принтеров), поэтому новое задание печати будет отправляться на следующий свободный принтер в группе.
- Файл `printers.conf` – настройки установленных принтеров.
- Файл `subscriptions.conf` – настройки используемых подписок.

16.2.1.5 Файлы заданий

Получив документы для распечатки, диспетчер печати не сразу передает их на вход фильтрам, а создает сначала задание в каталоге `/var/spool/cups`, что обеспечивает надежную работу диспетчера печати даже при сбоях. Файлы заданий бывают двух типов:

- **Управляющие файлы** — это сообщения IPP, основанные на исходных сообщениях IPP «Print-Job» или «Create-Job». Они начинаются с буквы «c» («c00001», «c99999», «c100000» и так далее).
- **Файлы данных** — это исходные файлы документов, отправленные на печать. Они начинаются с буквы «d» («d00001-001», «d99999-001», «d100000-001» и так далее).

Для каждого задания, известного системе, существует один управляющий файл и ноль или более файлов данных. Управляющие файлы обычно удаляются после отправки 500-го задания, а файлы данных удаляются сразу после успешной печати задания.

16.2.1.6 Файлы журналов

В случае каких-либо проблем очень помогают файлы журналов CUPS. Планировщик хранит три типа файлов журналов в каталоге `/var/log/cups`:

- В файле `access_log` перечислены все запросы HTTP и IPP, обработанные диспетчером.
- В файле `error_log` содержатся сообщения диспетчера и сопутствующих приложений. Этот файл можно использовать для выявления проблем.
- А в `page_log` перечисляются все распечатанные страницы, что может пригодиться для учета расходов.

Файлы журналов автоматически меняются планировщиком, когда они достигают установленного ограничениями размера. Если параметру MaxLogSize задано значение 0 (как в Astra Linux по умолчанию), то ротация в планировщике не выполняется. В Astra Linux для централизованной ротации журналов используется утилита `logrotate`, с которой мы познакомимся в *Модуль 21. Работа с журналами и отладка системы*.

```
localadmin@astra:~$ sudo ls /var/log/cups
access_log       access_log.4.gz  error_log       error_log.4.gz
access_log.1     access_log.5.gz  error_log.1     error_log.5.gz
access_log.2.gz  access_log.6.gz  error_log.2.gz  error_log.6.gz
access_log.3.gz  access_log.7.gz  error_log.3.gz  error_log.7.gz
```

По умолчанию файл `page_log` вестись не будет. Чтобы включить эту опцию, нужно в файл настроек CUPS `/etc/cups/cupsd.conf` к параметру PageLogFormat добавить следующее значение:

```
PageLogFormat %p %u %j %T %P %C %{job-billing} %{job-originating-host-name} %{job-
↪name} %{media} %{sides}
```

После внесения изменений в конфигурационный файл можно выполнить проверку. Для этого необходимо выполнить следующую команду:

```
localadmin@astra:~$ sudo cupsd -t
```

И перезапустить CUPS:

```
localadmin@astra:~$ sudo systemctl restart cups
```

На рисунке рис. 1.89 приведена более подробная схема взаимодействия компонентов CUPS, включая файлы конфигурации, заданий и журналов.

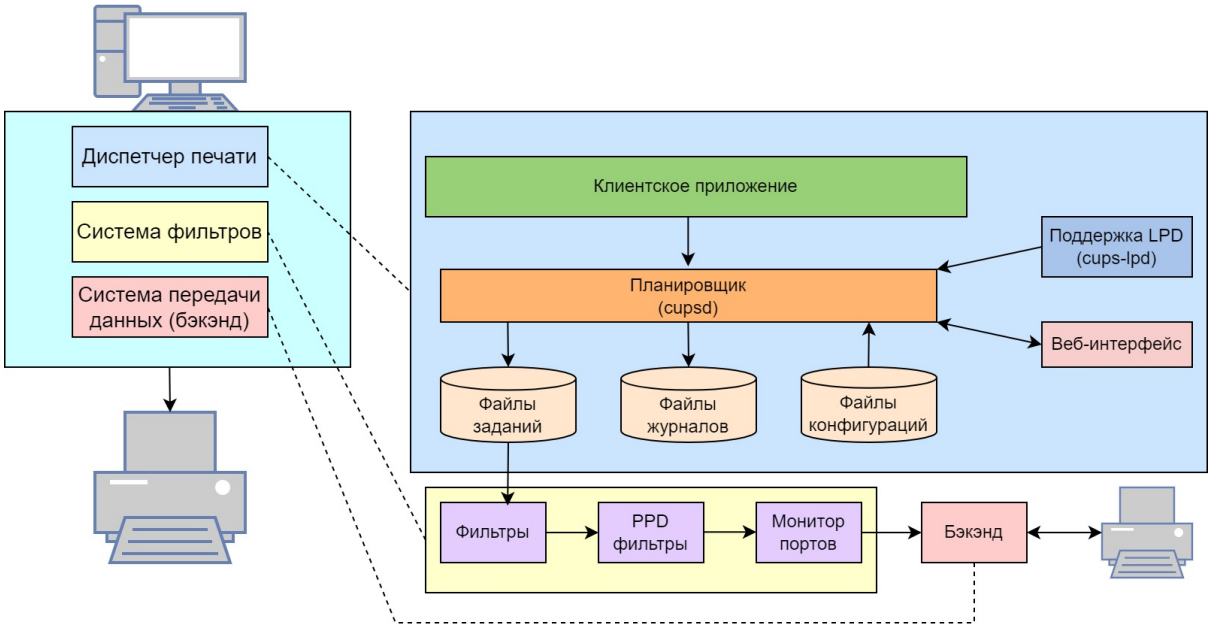


рис. 1.89 – Компоненты CUPS и их взаимодействие

16.2.1.7 Веб-интерфейс

Для управления CUPS в каждом дистрибутиве Linux есть свои десктопные утилиты, которые взаимодействуют с подсистемой напрямую по протоколу IPP. Однако CUPS предлагает и универсальный веб-интерфейс, который будет выглядеть одинаково на всех платформах. На обычных рабочих станциях веб-интерфейс доступен по адресу http://localhost:631, а на принт-серверах его обычно открывают с использованием защищенного соединения по 443 порту (https).

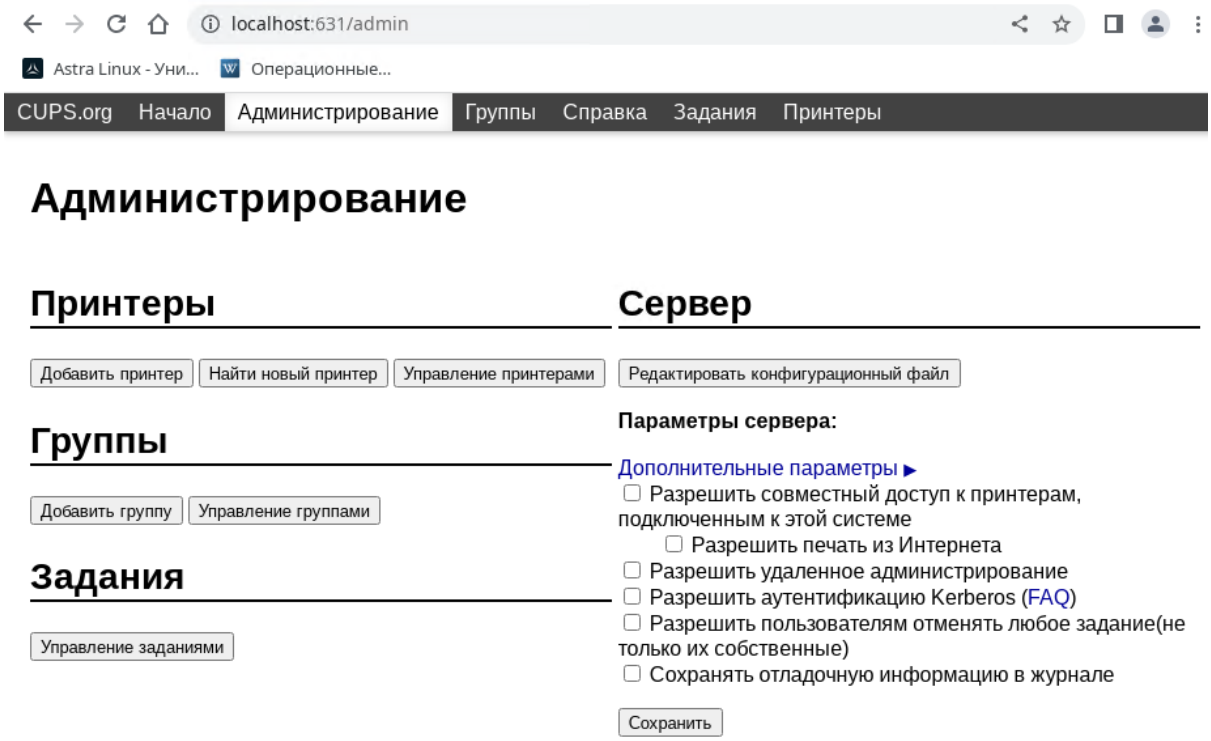


рис. 1.90 – Веб-интерфейс для управления CUPS

Веб-интерфейс управления CUPS реализован в виде нескольких CGI-приложений, которые расположены в папке `/usr/lib/cups/cgi-bin/`. Если обратите внимание, то каждая программа соответствует вкладке в браузере на странице CUPS:

Программа	Страница	Описание
admin.cgi	/admin	Содержит все инструменты для администрирования cupsd
classes.cgi	/classes	Управление группами (классами) принтеров и их список
help.cgi	/help	Доступ к онлайн документации (на английском языке)
jobs.cgi	/jobs	Управление заданиями и их список
printers.cgi	/printers	Управление принтерами и их список

16.2.1.8 Поддержка LPD

Протокол LPD (*англ. CUPS Line Printer Daemon, LPD*) уже давно устарел, поэтому в Astra Linux cups-lpd и inetd по умолчанию не устанавливаются. Но в других системах линукс вы можете обнаружить, что cups-lpd действует в связке со службой inetd, которая прослушивает порт 515/TCP и передает входящие LPD-запросы.

16.2.2 Управление CUPS

16.2.2.1 Способы управления CUPS

Управлять cupsd можно несколькими способами:

- Изменением настроек конфигурационного файла `/etc/cups/cupsd.conf`.
- Через веб-интерфейс, доступный (по умолчанию) по адресу http://localhost:631.
- С помощью консольных команд, например, `sudo lpinfo -v` – выводит список всех устройств.
- С помощью графической оснастки «Принтеры» (fly-admin-printer). В других Linux системах используются альтернативные приложения.

16.2.2.2 Управление с помощью графического приложения «Принтеры»

Графическую оснастку «Принтеры» для управления печатающими устройствами можно запустить из терминала командой `fly-admin-printer` или через меню *Пуск → Системные → Принтеры*. У приложения простой интуитивно понятный интерфейс, к которому не требуется мануал на 1000 страниц, но дополнительную информацию вы все же можете получить клавишей F1.

По следующим скриншотам вы можете увидеть, что управление печатью в Linux может быть не многим сложнее, чем в Windows. Технические подробности доступных настроек рассмотрим в следующих разделах.

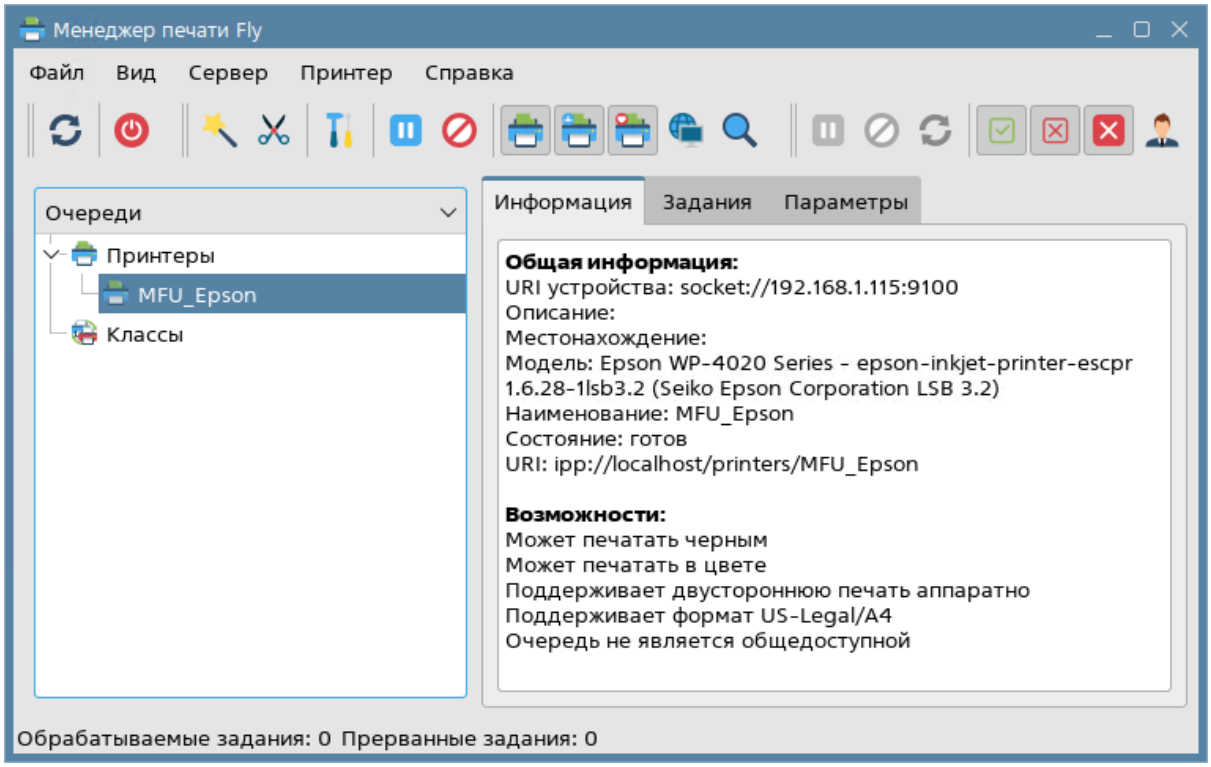


рис. 1.91 – Интерфейс графической оснастки «Принтеры»

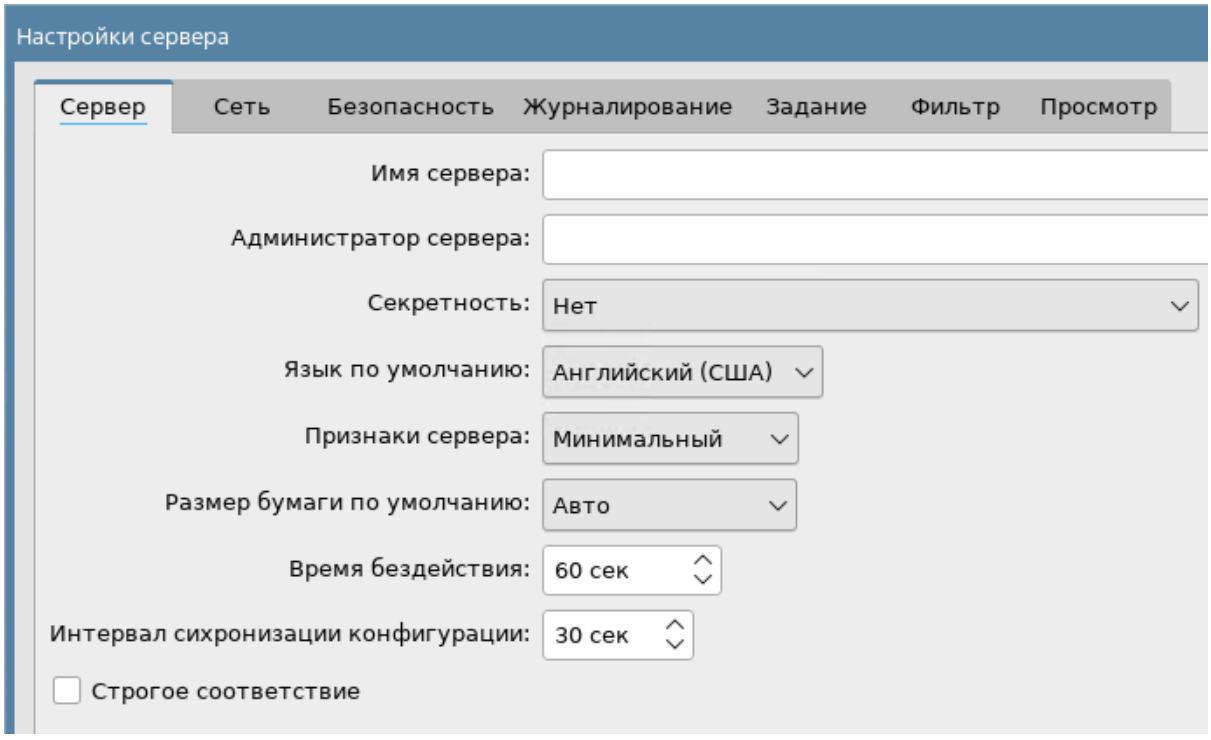


рис. 1.92 – Настройка сервера CUPS *Сервер → Настроить...*

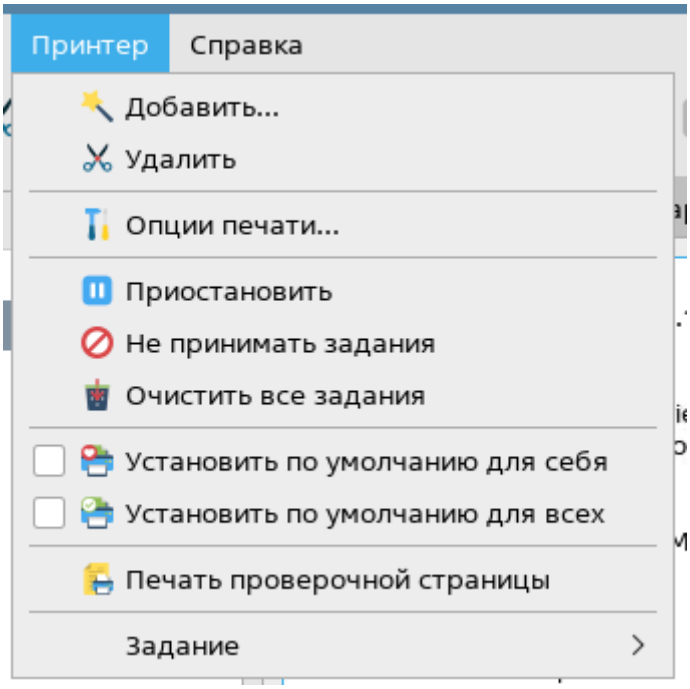


рис. 1.93 – Меню управления принтерами

16.2.2.3 Управление через веб-интерфейс

Веб-интерфейс доступен по адресу http://localhost:631 и предоставляет унифицированный доступ для любых Linux-систем вне зависимости от того, какие графические оснастки для них разработаны.

В разделе «Сервер» можно задать некоторые параметры или перейти к прямому редактированию конфигурационного файла `/etc/cups/cupsd.conf`.

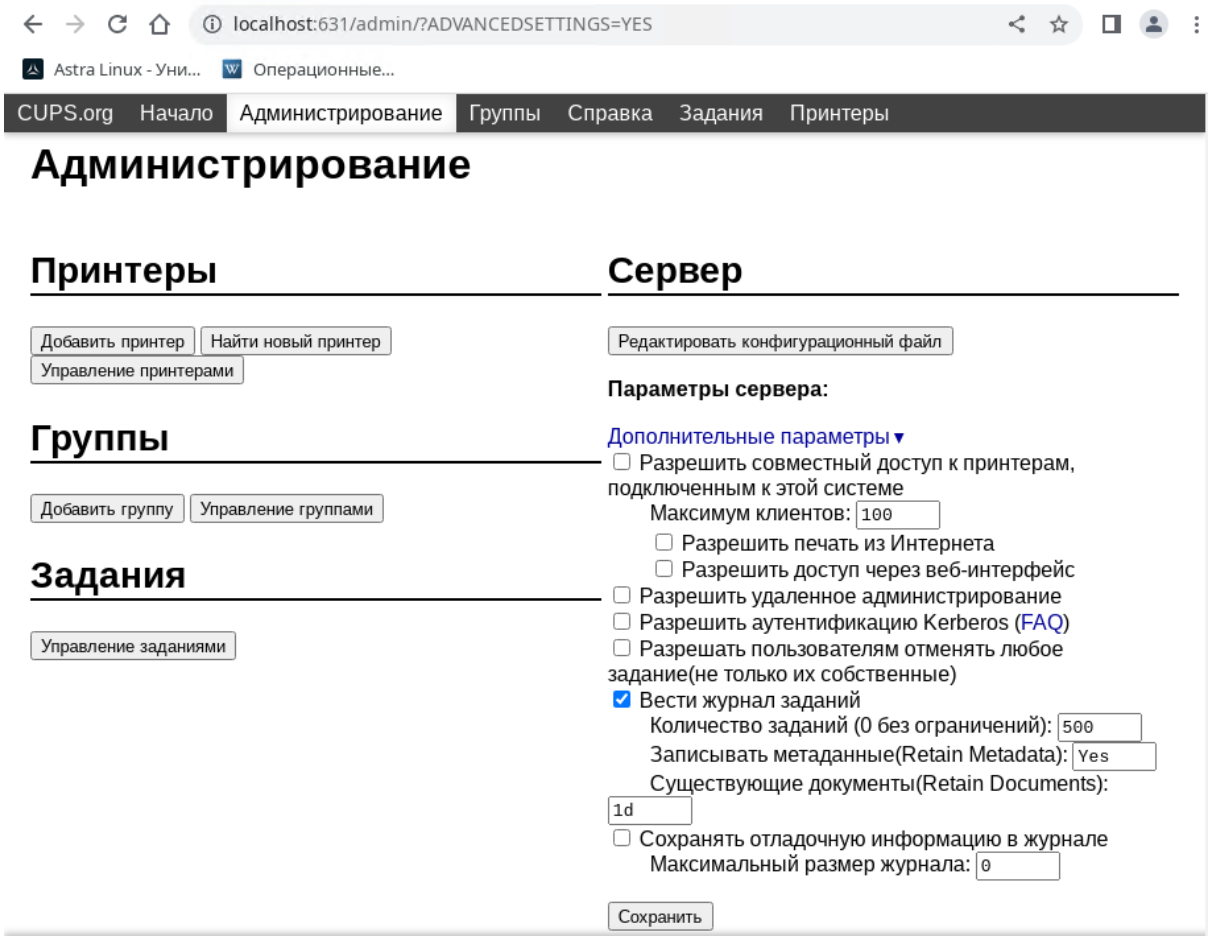


рис. 1.94 – Страница администрирования веб-портала CUPS

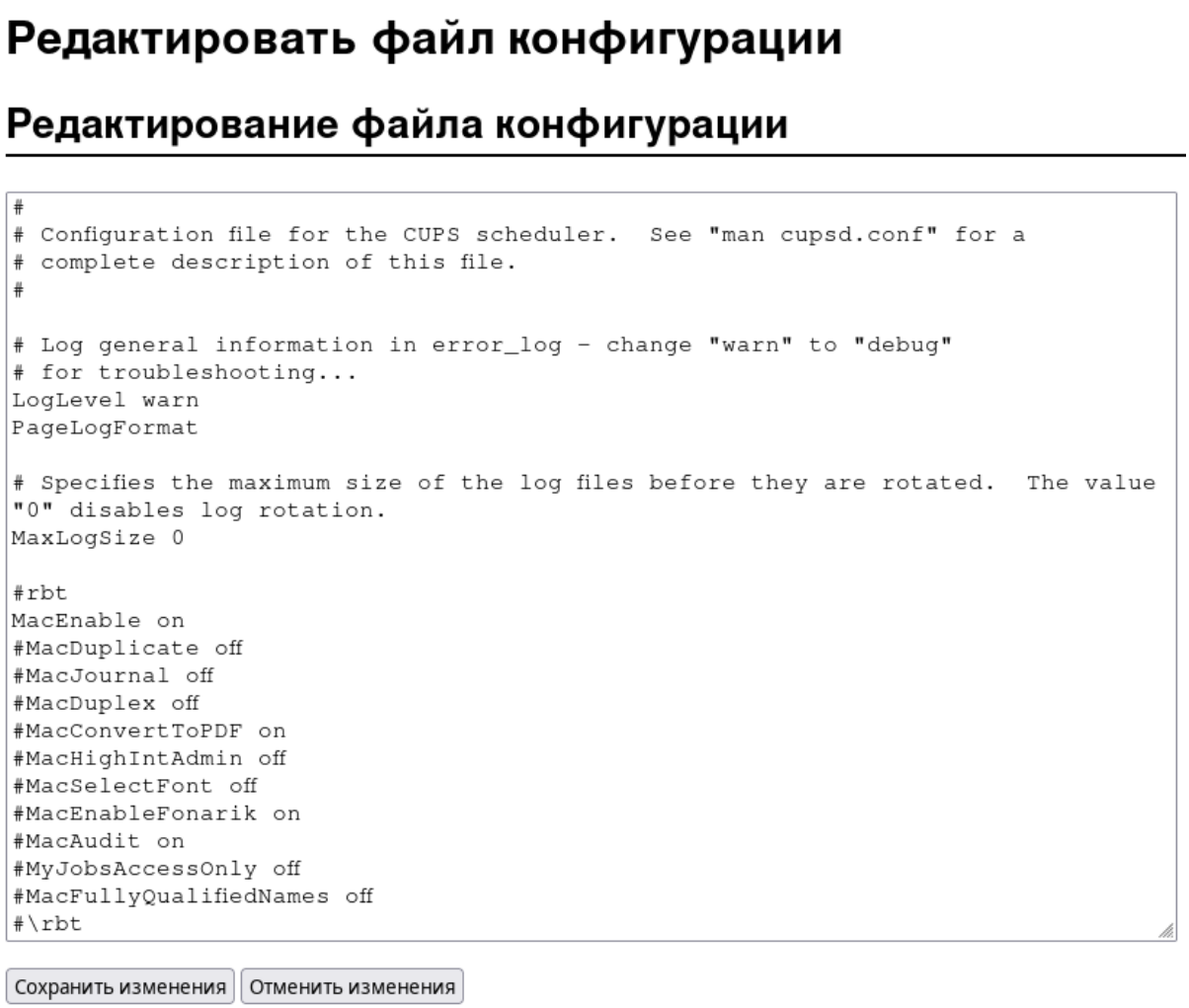


рис. 1.95 – Страница редактирования конфигурационного файла cupsd.conf

Добавление принтера, создание группы – все эти действия интуитивно понятны и не требуют уточнения. Технические подробности доступных настроек рассмотрим в следующих разделах.

16.2.2.4 Управление через параметры конфигурационного файла

Настройки CUPS могут быть выполнены через конфигурационный файл `/etc/cups/cupsd.conf`. Чтение конфигурации производится сервером cupsd только в момент запуска, поэтому для вступления изменений в силу необходимо перезапустить демон cupsd командой `sudo systemctl restart cups.service`.

```
localadmin@astra:~$ sudo systemctl restart cups.service
```

Опция «Listen»

По умолчанию на рабочих станциях служба cups слушает входящие подключения только на localhost. Проверить это можно командой `sudo netstat -plutn | grep cupsd`.

```
localadmin@astra:~$ sudo netstat -plutn | grep cupsd
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN      3468/cupsd
tcp6       0      0 ::1:631                 :::*                    LISTEN      3468/cupsd
```

Так как в этом модуле мы ведем речь в основном только о локальной печати, т.е. о локальном диспетчере печати, то прослушивание только localhost будет достаточным решением.

Но если вы захотите открыть доступ к локальному принтеру, как в Windows, то нужно:

1) Удалить или закомментировать строку «Listen localhost:631».

2) Добавить строку «Port 631» после строки «Listen /run/cups/cups.sock».

3) После строки комментария «# Only listen for connections from the local machine» можно добавить необходимые адреса/подсети, которые будут иметь доступ к cups.

```
#Listen localhost:631
Listen /run/cups/cups.sock
Port 631
```

Опция «Browsing»

Обнаружение сетевых и общих (*англ. shared*) принтеров управляется через опцию: «Browsing No». Если опция включена («On»), то компьютеры в сети будут видеть список всех принтеров, подключенных к серверу печати (что не очень удобно, если парк принтеров большой). Принцип работы заключается в рассылке широковещательных пакетов через avahi (кстати, именно поэтому мы не рекомендуем использовать доменную зону .local).

Каждый широковещательный пакет описывает один общий принтер или группу принтеров. Любая другая система CUPS в подсети (клиенты) получит широковещательную рассылку и сделает этот принтер доступным для локальных пользователей. Если клиент перестает получать широковещательные сообщения от сервера или если сервер отправляет специальное широковещательное сообщение «deleted», клиент удалит свою копию принтера.

Опция «DefaultAuthType»

Опция «DefaultAuthType Basic» определяет права доступа к веб-интерфейсу CUPS. Чтобы определенный пользователь имел доступ к администрированию CUPS, его нужно добавить в группу «lpadmin»: `sudo usermod -aG lpadmin <имя_пользователя>`. Группа «lpadmin» является группой по умолчанию для предоставления административного доступа, и ее можно поменять на любую другую. Для этого в файле `/etc/cups/cups-files.conf` в опции «SystemGroup» через пробел добавьте свою группу и перезапустите службу CUPS.

Разделы «Location»

Далее в файле идут разделы «Location», что актуально только при настройке общего доступа к принтерам для других компьютеров в сети.

```
# Restrict access to the server...
<Location />
 Order allow,deny
</Location>

# Restrict access to the admin pages...
<Location /admin>
 Order allow,deny
</Location>
. . .
```

Первый параметр «Order allow,deny» ограничивает все подключения, если они не разрешены явно. Ниже нужно добавить адреса, подсети, хосты которым разрешен доступ к серверу печати:

```
Allow from 10.1.10.*

Allow from localhost

Allow from printserver

Allow from 172.16.20.1
```

16.2.2.5 Управление через консольные команды

Установка/удаление виртуального принтера

Управление принтерами через консольные команды требуется только в заданиях автоматизации, но они помогут нам наиболее полно раскрыть механизмы работы CUPS. В случае если у вас не будет физического принтера, то для тестов достаточно будет установить пакет *printer-driver-cups-pdf*, который создаст виртуальный PDF-принтер:

```
localadmin@astra:~$ sudo apt install printer-driver-cups-pdf
```

Этот пакет установит нам принтер с именем PDF, убедимся в этом:

```
localadmin@astra:~$ lpstat -t
планировщик запущен
назначение системы по умолчанию: PDF
устройство для PDF: cups-pdf:/
PDF принимает запросы с момента Пн 27 мая 2024 15:28:07
принтер PDF свободен. Включен с момента Пн 27 мая 2024 15:28:07
```

Давайте создадим еще один PDF-принтер, чтобы иметь возможность объединить принтеры в группу (пул принтеров):

```
localadmin@astra:~$ sudo lpadmin -p ONE-MORE-PDF -v cups-pdf:/ -E -i /usr/share/ppd/
↪cups-pdf/CUPS-PDF_opt.ppd
```

Утилита lpadmin управляет принтерами и их настройками в Linux. Мы использовали следующие ключи:

- Ключ `-p` — указывает имя создаваемого принтера. В данном случае — «ONE-MORE-PDF».
- Ключ `-v` — указывает бэкенд CUPS для нового принтера. В данном случае — «cups-pdf».
- Ключ `-E` — если ключ указан после ключа `-p`, то разрешает принтеру принимать задания.
- Ключ `-i` — указывает путь к файлу PPD для нового принтера. В данном случае — `/usr/share/ppd/cups-pdf/CUPS-PDF_opt.ppd`. Этот файл содержит параметры и настройки для работы принтера. Он был установлен пакетом printer-driver-cups-pdf.

Убедимся, что наш новый принтер появился в системе:

```
localadmin@astra:~$ lpstat -p
принтер ONE-MORE-PDF свободен. Включен с момента Пн 27 мая 2024 15:53:06
принтер PDF свободен. Включен с момента Пн 27 мая 2024 15:28:07
```

Для удаления группы принтеров достаточно воспользоваться ключом -x утилиты lpadmin:

```
localadmin@astra-ald:~$ sudo lpadmin -x ONE-MORE-PDF
localadmin@astra-ald:~$ sudo lpadmin -x PDF
```

Создание/удаление группы принтеров

Следующая команда создаст группу принтеров по имени «PDF_Printer_Group» и поместит в неё принтер PDF:

```
localadmin@astra:~$ sudo lpadmin -p PDF -c PDF_Printer_Group
```

Поместим в эту группу второй созданный нами принтер «ONE-MORE-PDF»:

```
localadmin@astra:~$ sudo lpadmin -p ONE-MORE-PDF -c PDF_Printer_Group
```

Проверим, что группа действительно содержит оба наших принтера:

```
localadmin@astra:~$ lpstat -c PDF_Printer_Group
члены группы PDF_Printer_Group:
        PDF
        ONE-MORE-PDF
```

Сделаем группу общедоступной и разрешим ей принимать задания на печать:

```
localadmin@astra-ald:~$ sudo lpadmin -p PDF_Printer_Group -o printer-is-shared=true -E
```

Группа принтеров удаляется так же, как обычный принтер, ключом -x утилиты lpadmin:

```
localadmin@astra-ald:~$ sudo lpadmin -x PDF_Printer_Group
```

Печать заданий

Отправим файл `.profile` на печать из командной строки (из PowerShell мы такое не делали чуть чаще, чем никогда, никогда, никогда):

```
localadmin@astra:~$ lpr -P PDF_Printer_Group -T ‘‘profile’’ .profile
```

Где:

- Ключ `-P` – указывает, куда утилита lpr должна отправить файл. В нашем случае файл будет передан группе принтеров, а уже группа передаст первому свободному принтеру.
- Ключ `-T` – указывает имя задания.

Примечание: При отправке печати на группу принтеров, CUPS начнет балансировать нагрузку (чередование принтеров), тем самым новое задание печати будет отправляться на следующий принтер в группе.

Учитывая, что в группу входят виртуальные PDF-принтеры, при получении заданий они будут создавать pdf-файлы в папке PDF домашнего каталога пользователя:

```
localadmin@astra-ald:~$ ls ~/PDF
profile-job_1.pdf
```

Проверка статуса CUPS

Проверка статуса подсистемы печати выполняется командой `lpstat -t`.

```
localadmin@astra:~$ lpstat -t
планировщик запущен
Нет назначение системы по умолчанию
устройство для MFU_Epson: socket://192.168.1.115:9100
MFU_Epson принимает запросы с момента Пт 15 сен 2023 09:49:26
принтер MFU_Epson свободен. Включен с момента Пт 15 сен 2023 09:49:26
       Файл "/opt/epson-inkjet-printer-escpr/cups/lib/filter/epson-escpr-wrapper"␣
↪имеет
небезопасные права доступа  (0100755/uid=1000/gid=1000).
```

На приведенном выше примере система сообщает, что файл `/opt/epson-inkjet-printer-escpr/cups/lib/filter/epson-escpr-wrapper` (созданный в процессе установки драйверов) имеет небезопасные права доступа (0100755/uid=1000/gid=1000).

Для принтеров epson требуется дополнительно настраивать права доступа, но об этом всегда указывают в дополнительных инструкциях по подключению конкретных устройств:

```
localadmin@astra:~$ sudo chown root:root -R /opt/epson-inkjet-printer-escpr/cups/lib
```

(continues on next page)

(продолжение с предыдущей страницы)

```
localadmin@astra:~$ ls -l /opt/epson-inkjet-printer-escpr/cups/lib
итого 4
drwxr-xr-x 2 root root 4096 сен 15 09:44 **filter**
```

16.3 Практика и тестирование

16.3.1 Практическая работа: Модуль 16. Работа с подсистемой печати

См. также *Требования, правила и цели выполнения практической работы*

16.3.1.1 Практические задания

Задание 1.

Установите пакет printer-driver-cups-pdf любым удобным вам способом. В консоли выведите список доступных принтеров на компьютере и найдите название принтера.

Задание 2.

Откройте текстовый редактор (например, LibreOffice) и попробуйте распечатать любой документ на принтер, установленный в пункте 1.

Убедитесь, что документ «распечатался», используя приложение «Очередь печати».

Задание 3.

Зайдите на веб-интерфейс CUPS, создайте новый принтер PDF с именем **virtual_printer**. Сделайте его принтером по умолчанию. Убедитесь в том, что он стал принтером по умолчанию, используя консоль.

Задание 4.

Удалите принтер **virtual_printer**, используя графический интерфейс менеджера печати Fly.

Задание 5.

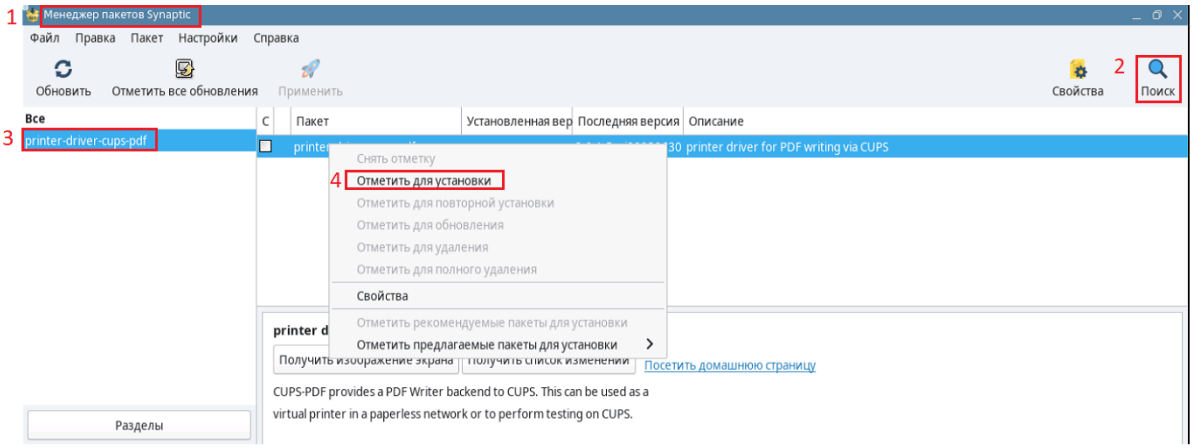
Вызовите электронную справку, посмотрите самостоятельно раздел Принтеры, в частности найдите информацию о PPD файлах.

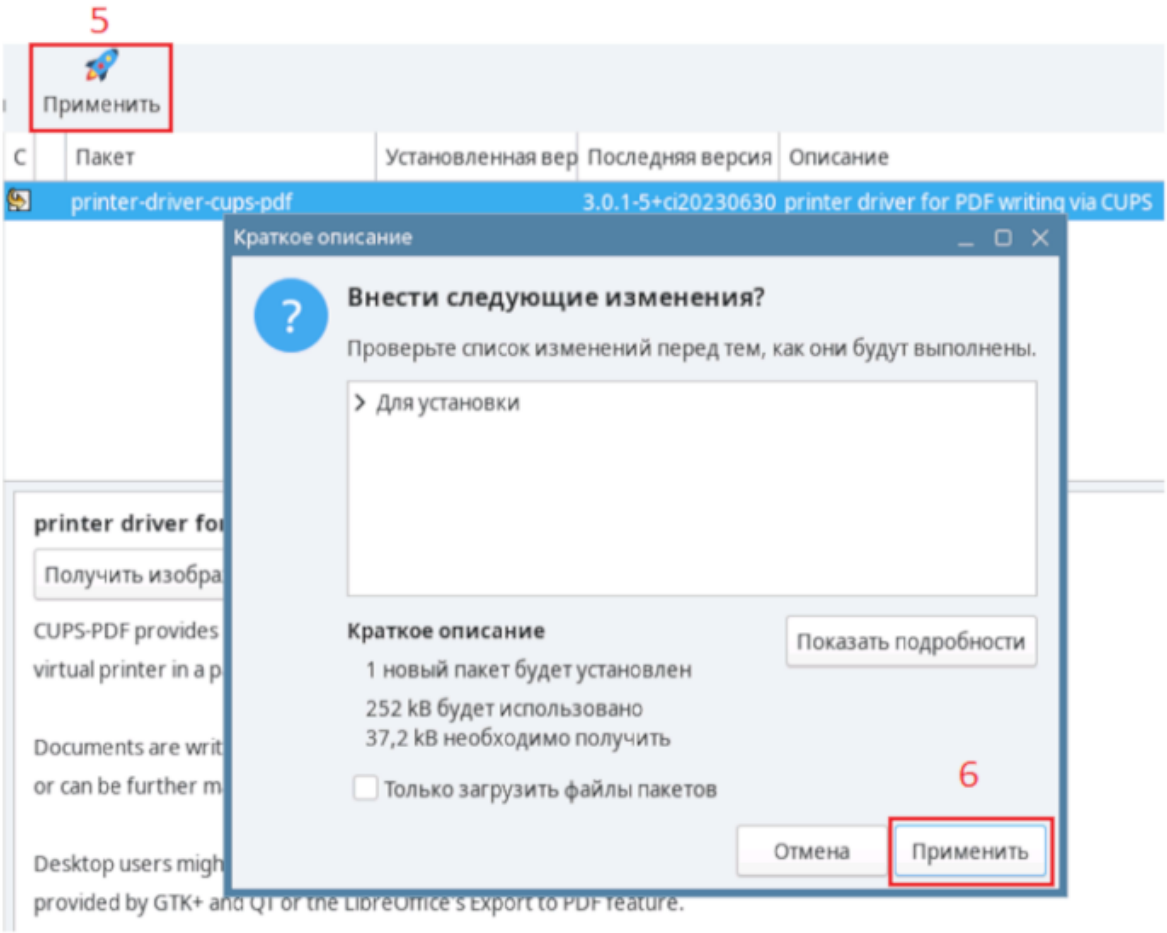
16.3.1.2 Ответы на практические задания (пошаговые инструкции)

Задание 1.

Установите пакет printer-driver-cups-pdf любым удобным вам способом. В консоли выведите список доступных принтеров на компьютере и найдите название принтера.

Установите пакет через менеджер пакетов Synaptic.





В консоли выполните команду `lpstat -p`.

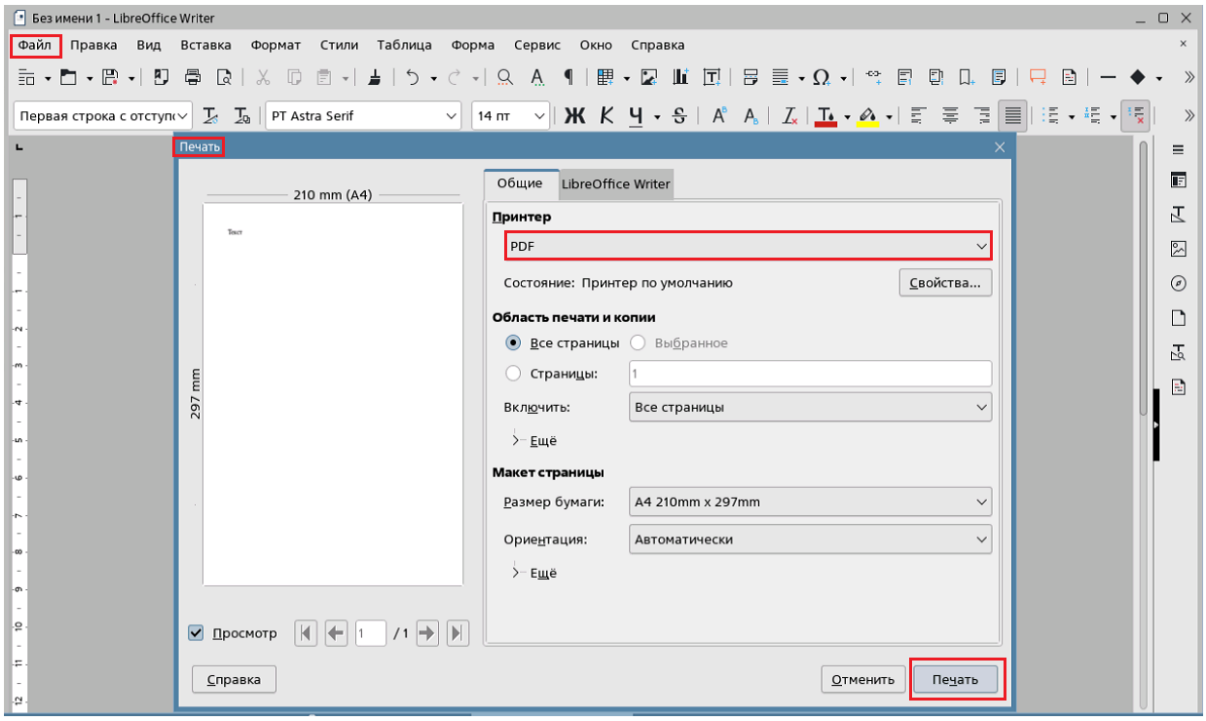
```
localadmin@astra:~$ lpstat -p
принтер PDF свободен. Включен с момента Чт 29 фев 2024 16:57:02
localadmin@astra:~$
```

Задание 2.

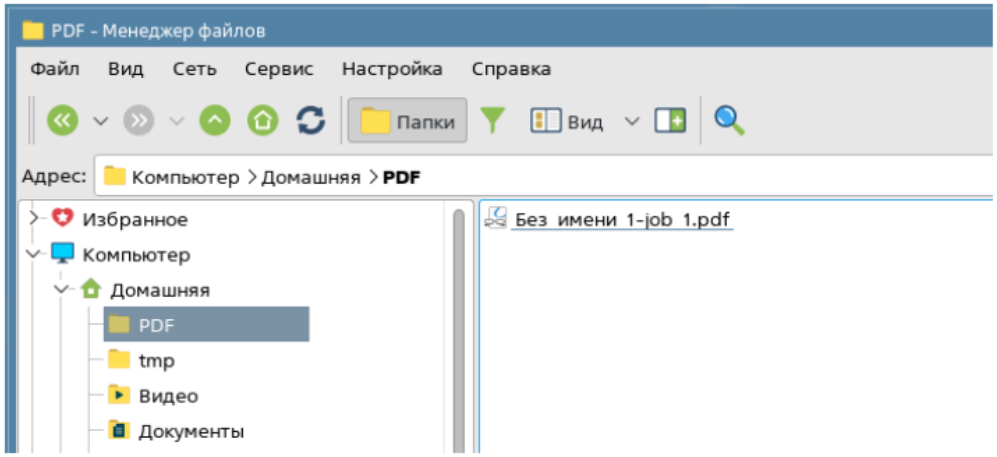
Откройте текстовый редактор (например, LibreOffice) и попробуйте распечатать любой документ на принтер, установленный в пункте 1.

Убедитесь, что документ «распечатался», используя приложение «Очередь печати».

Запустите LibreOffice, наберите текст, далее нажмите сочетание клавиш Ctrl + p или в строке меню выберите *Файл → Печать*.

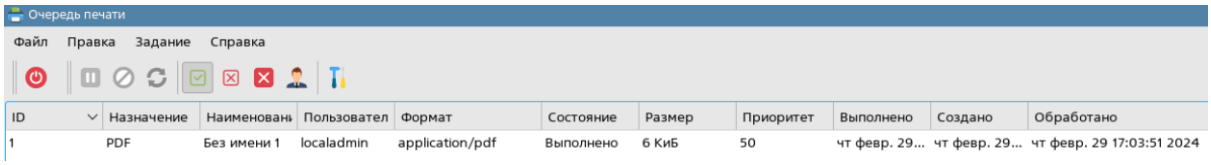


По умолчанию виртуальный принтер PDF печатает в каталог PDF в домашней директории пользователя, запустившего печать. Там вы сможете найти «распечатанный» документ.




```
localadmin@astra:~$ ls -la /home/localadmin/PDF/
итого 16
drwx------  2 localadmin localadmin 4096 фев 29 17:03 .
drwx------ 20 localadmin localadmin 4096 мар  1 09:47 ..
-rw-------  1 localadmin localadmin 4251 фев 29 17:03 Без_имени_1-job_1.pdf
```

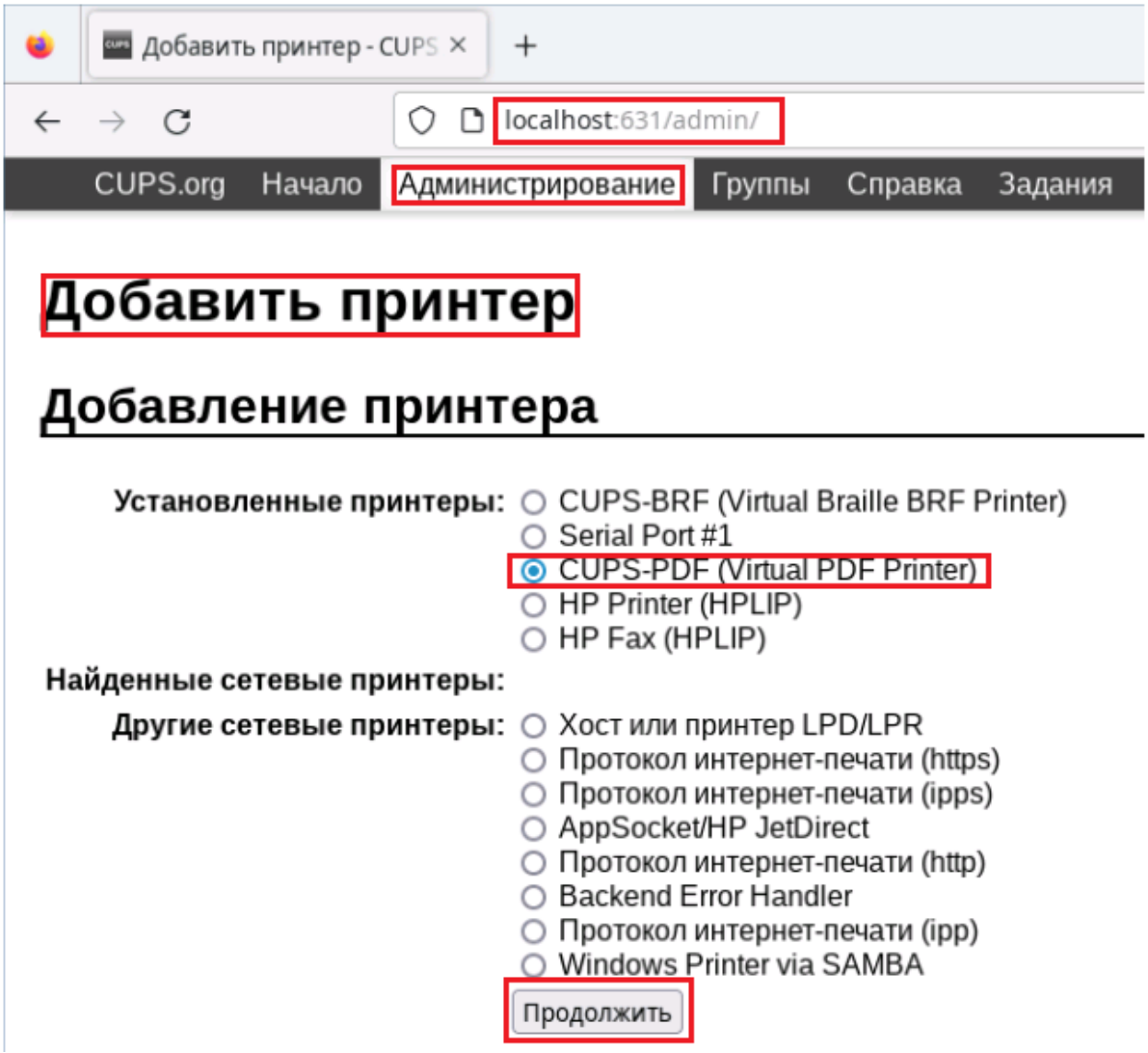
Также вы сможете увидеть его через очередь печати. Запустите **fly-jobviewer**, далее в строке меню выберите *Задание* и выделите *Показывать выполненные задания*.



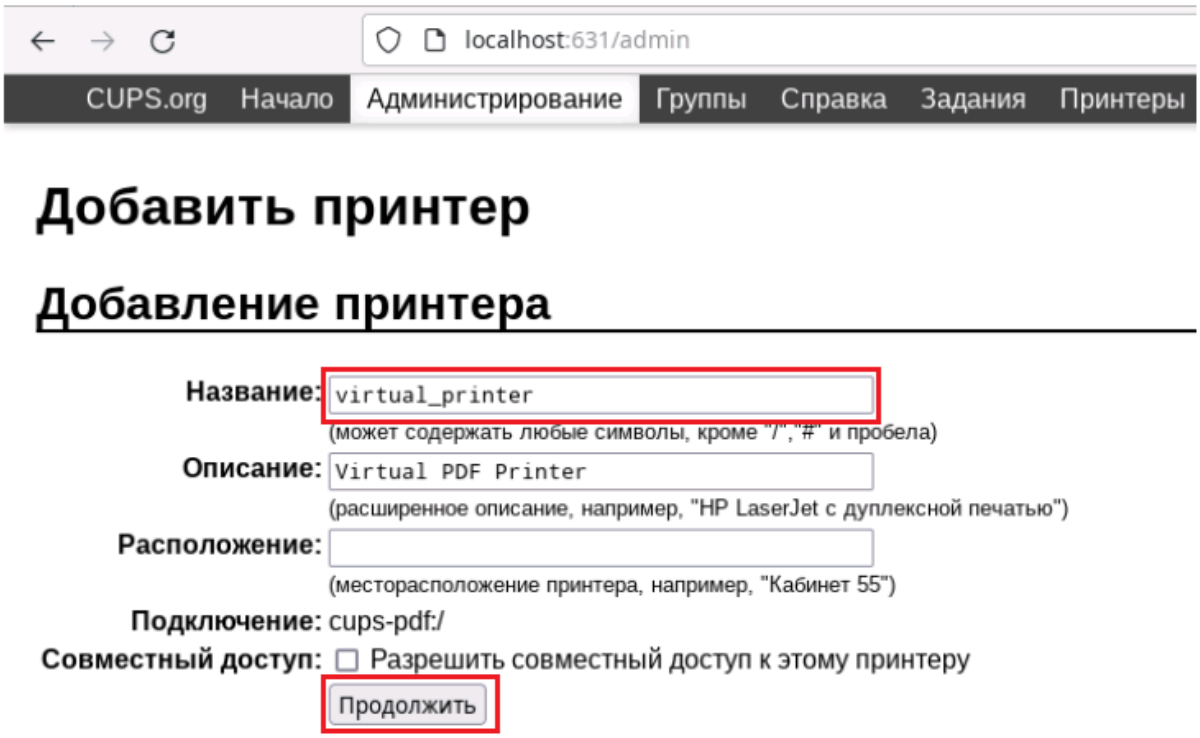
Задание 3.

Зайдите на веб-интерфейс CUPS, создайте новый принтер PDF с именем **virtual_printer**. Сделайте его принтером по умолчанию. Убедитесь в том, что он стал принтером по умолчанию, используя консоль.

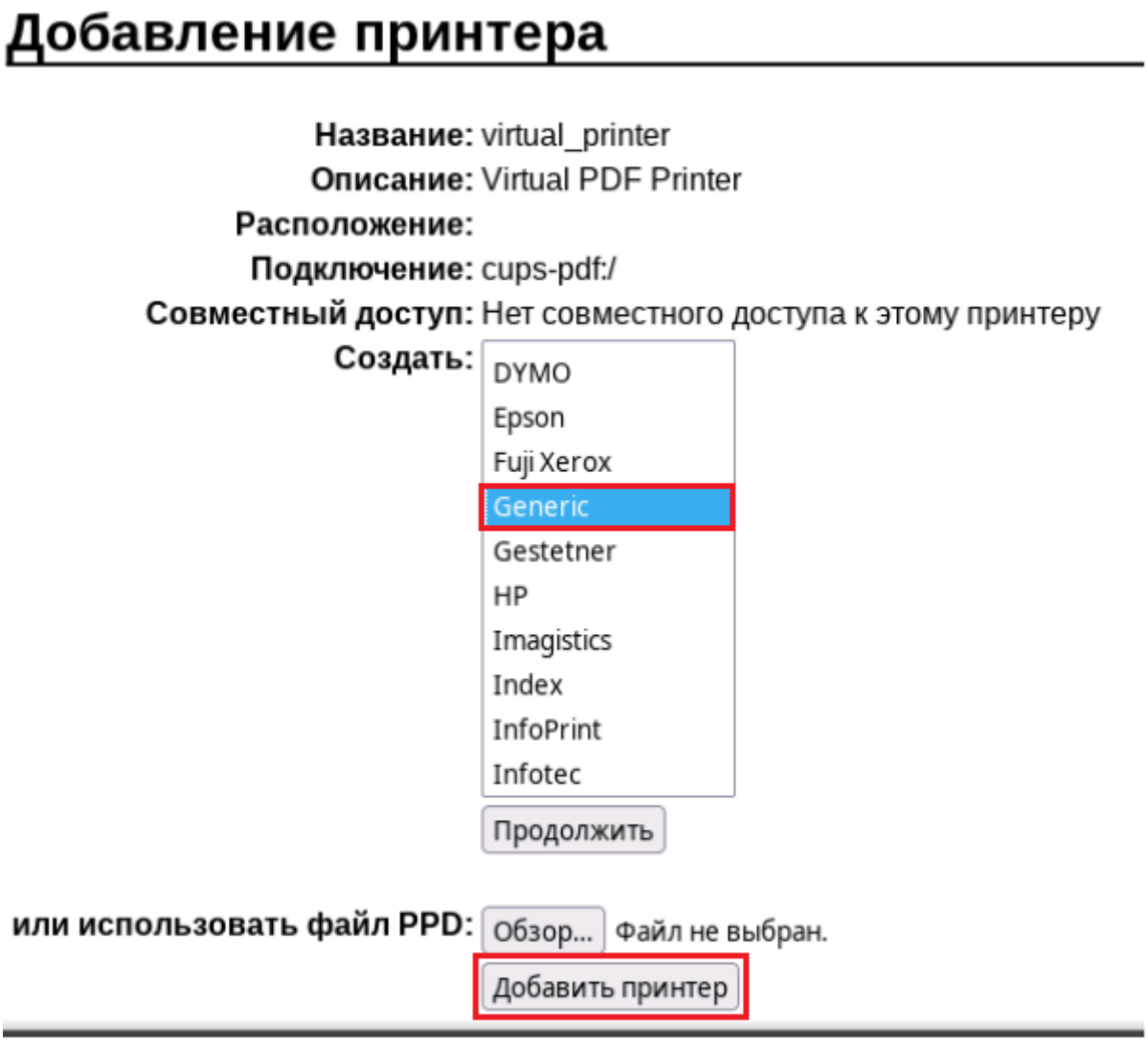
В браузере введите адрес http://localhost:631, далее откройте вкладку *Администрирование* → *Добавить принтер* далее выбрать Установленные принтеры: CUPS-PDF (Virtual PDF Printer) нажмем кнопку *Продолжить*:



Далее задайте имя принтера:



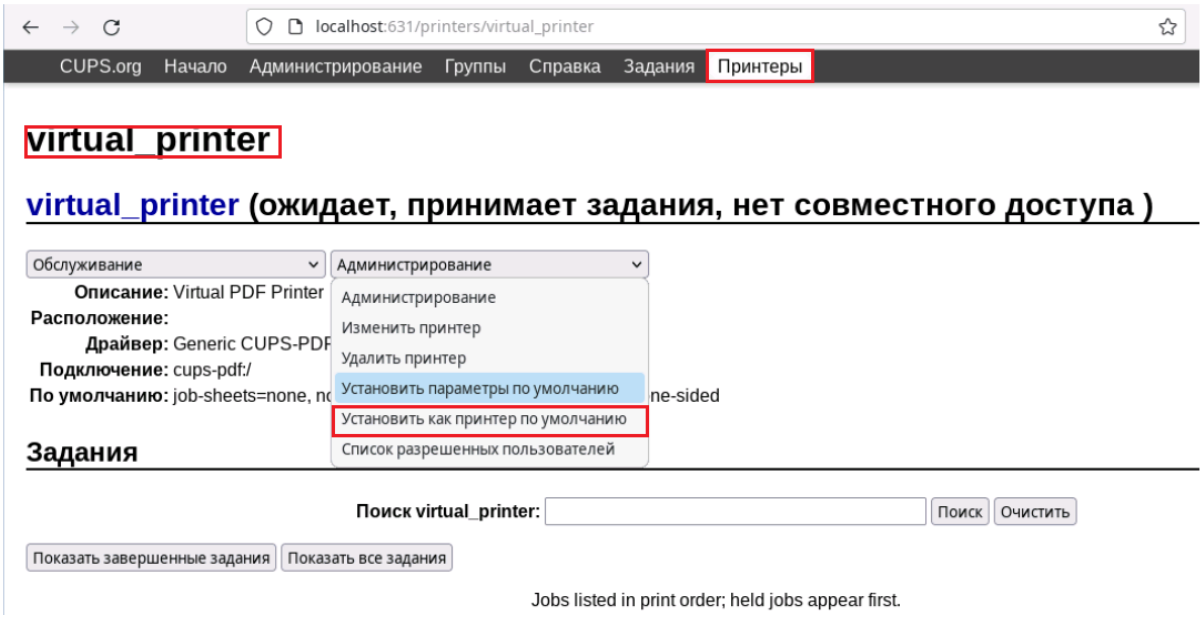
В следующем окне выберите «**Generic**» и нажмите на *Добавить принтер*.



Затем выберите модель принтера: **Generic CUPS-PDF Printer (w/ options) (en)** и нажмите на *Добавить принтер* .

В последнем окне оставьте параметры по умолчанию.

Следующим шагом сделайте принтер по умолчанию.

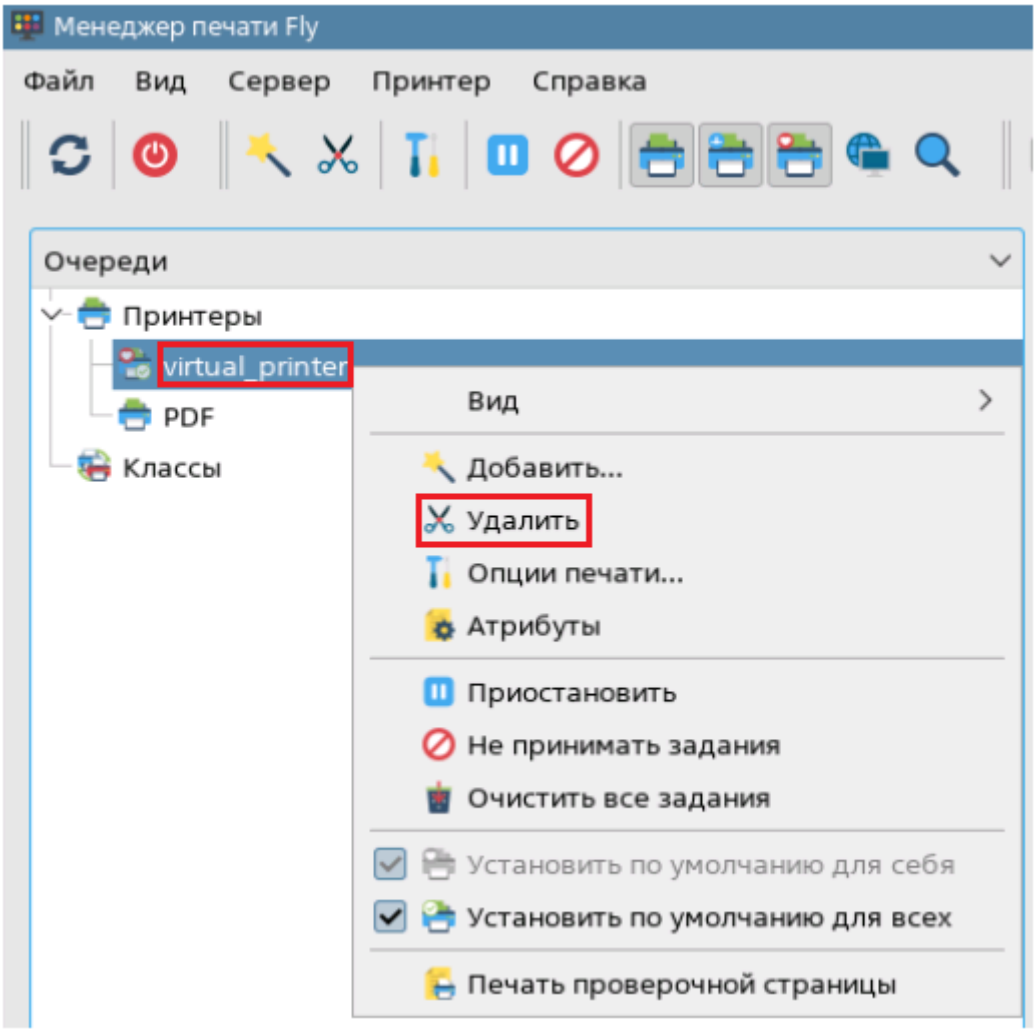


Чтобы убедиться в правильности настроек в консоли, можно выполнить команду `lpstat -d`.

```
localadmin@astra:~$ lpstat -d
назначение системы по умолчанию: virtual_printer
```

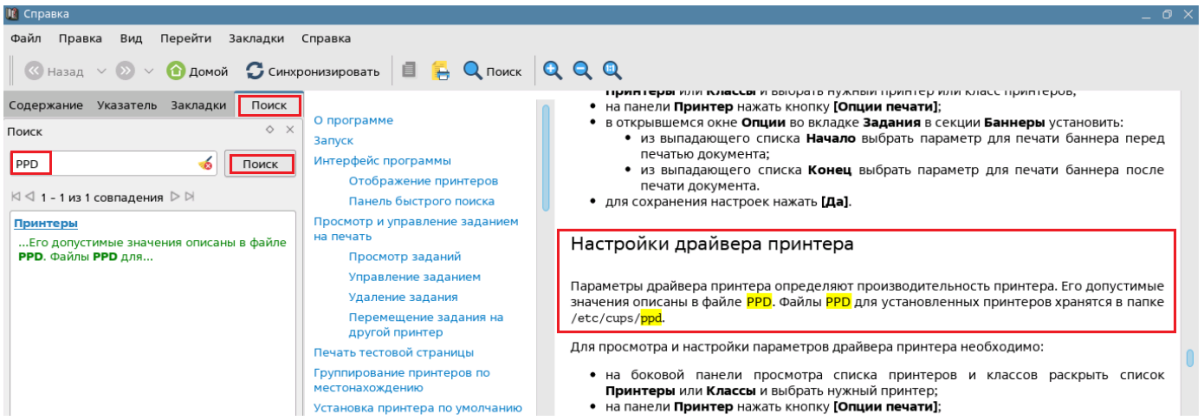
Задание 4.

Удалите принтер **virtual_printer**, используя графический интерфейс менеджера печати Fly.



Задание 5.

Вызовите электронную справку, посмотрите самостоятельно раздел Принтеры, в частности найдите информацию о PPD файлах.



16.3.2 Тест: Модуль 16. Работа с подсистемой печати

Пожалуйста, ответьте на несколько вопросов, чтобы закрепить полученные знания. Правильные ответы для самопроверки вы сможете найти в конце списка. Обратите внимание, что у некоторых вопросов может быть несколько верных вариантов ответов.

16.3.2.1 Вопросы

Вопрос 1. По какому порту доступен веб-интерфейс CUPS?

a. 80

b. 515

c. 631

d. 8080

Вопрос 2. Какой протокол не поддерживается системой CUPS?

a. IPP

b. LPD

c. SMB

d. NTP

Вопрос 3. Что означает значение `0` параметра `MaxLogSize` в файле `/etc/cups/cupsd.conf`?

a. Файлы логов выключены

b. Ротация логов выключена

c. Параметр не используется

Вопрос 4. Где хранятся файлы заданий печати?

a. Директория `/var/spool/cups`

b. Директория `/var/log/cups`

c. Директория `/etc/cups/`

d. Директория `/var/log/sssd`

Вопрос 5. По какому порту служба CUPS принимает запросы от внешних клиентов, если исполь-

зуется шифрование SSL?

a. 631

b. 443

c. 515

d. 445

Вопрос 6. Сервер CUPS принимает на печать только от localhost. Что нужно сделать, чтобы он начал принимать с любого адреса?

a. В файле `/etc/cups/cupsd.conf` добавить `Port 631`. Перезапустить службу cups.

b. В файле `/etc/cups/cupsd.conf` добавить `Port 631`.

c. В файле `/etc/cups/cupsd.conf` удалить строку `Listen localhost:631` и добавить `Port 631`.

d. В файле `/etc/cups/cupsd.conf` удалить строку `Listen localhost:631` и добавить `Port 631`. Перезапустить службу cups.

Ответы на вопросы

1. c
2. d
3. b
4. a
5. b
6. d

16.4 Заключение

В этом модуле мы познакомились с подсистемой печати CUPS и ее архитектурой, научились добавлять принтеры, создавать группы и управлять заданиями печати.

Следующий модуль будет посложнее. Он посвящен вопросам процесса запуска ОС и подсистеме

инициализации и управления службами systemd.

16.5 Дополнительные источники информации

- Руководство администратора CUPS от Easy Software Products, 2005
- Система печати CUPS в Astra Linux на wiki.astralinux.ru

16.6 Обратная связь

Если остались вопросы, то их всегда можно задать в специальной теме.

Модуль 17. Загрузка системы и управление службами systemd

17.1 Введение

В этом модуле мы вспомним о том, что происходит внутри компьютера после нажатия кнопки питания, рассмотрим этапы, через которые проходит система во время загрузки, и узнаем, как устранять проблемы, если в процессе загрузки что-то пойдет не по плану.

Отдельного внимания удостоится подсистема инициализации и управления службами systemd. Мы посмотрим, что представляют собой юниты и как управлять службами Linux. Узнаем, почему недолюбливают systemd, но при этом ей удалось вытеснить свою предшественницу init, в которой все было по фэншую.

После этого модуля мир для вас уже не будет прежним.

17.2 Общее представление о процессе загрузки ОС Linux

С одной стороны, Linux загружается так же, как Windows — и в том, и в другом случае компьютер тестирует оборудование, загружает ядро системы, стартует необходимые службы и запускает приложение для входа. С другой стороны, есть много существенных деталей, которые нужно знать, чтобы быстро устранять причины сбоев и с пониманием дела настраивать PXE-сервер для установки операционной системы по сети.

В конечном итоге вам решать, хотите ли вы как сеньор админ из Вилларибо за пару часов починить сервер, на котором ваш верный оруженосец в пятницу вечером сломал загрузочный сектор, или вы будете с ним на пару все выходные восстанавливать сервер из резервной копии, как какие-то джуны из Виллабаджо.

В общем виде загрузка системы выглядит следующим образом:

- Вы нажимаете кнопку включения, и **материнская плата** подает питание на центральный процессор.

- Как только питание поступает на **процессор,** он приступает к выполнению инструкций базовой системы BIOS/UEFI, программный код которой находится в постоянном запоминающем устройстве (ПЗУ) на материнской плате. Такое поведение материнской платы и процессора предопределено их аппаратным устройством.

- Получив управление, **базовая система** выполняет проверку оборудования и приступает к поиску загрузочного диска, для чего в настройках определяется перечень допустимых устройств и порядок их использования. Обычно загрузка выполняется с жесткого диска, но это может быть и флешка, и компакт диск, и даже сетевая карта.

  Загрузочный диск может быть размечен в старом формате MBR для BIOS или в новом формате GPT для UEFI, что определяет детали дальнейшей загрузки, но суть процесса от этого не меняется — базовая система передает управление загрузчику.

- Получив управление от BIOS/UEFI, **загрузчик** выполняет загрузку операционной системы самостоятельно или по цепочке передает управление другому загрузчику для решения поставленной задачи.

  Загрузчики представляют собой вспомогательные приложения, которые находятся между прошивкой и операционной системой, чтобы избавить ОС от технических подробностей реализации загрузочных секторов.

Примечание: Минутка Шелдона Купера: В старых компьютерах кнопка включения была фиксирующего типа и напоминала больше рубильник, который замыкал силовую цепь, но с переходом на ATX дежурное питание присутствует на материнской плате постоянно. Поэтому сейчас даже в выключенном состоянии компьютер все равно остается включен и может управлять блоком питания для подачи основного напряжения.

Благодаря такому устройству системы питания стала возможна реализация технологий Wake-on-LAN, iLO и IPMI. А чтобы полностью обесточить процессор, теперь кнопку включения нужно зажать на несколько секунд.

На рис. 1.96 представлен весь процесс загрузки системы Linux с краткими пояснениями по каждому из этапов, которые мы будем подробно разбирать далее.

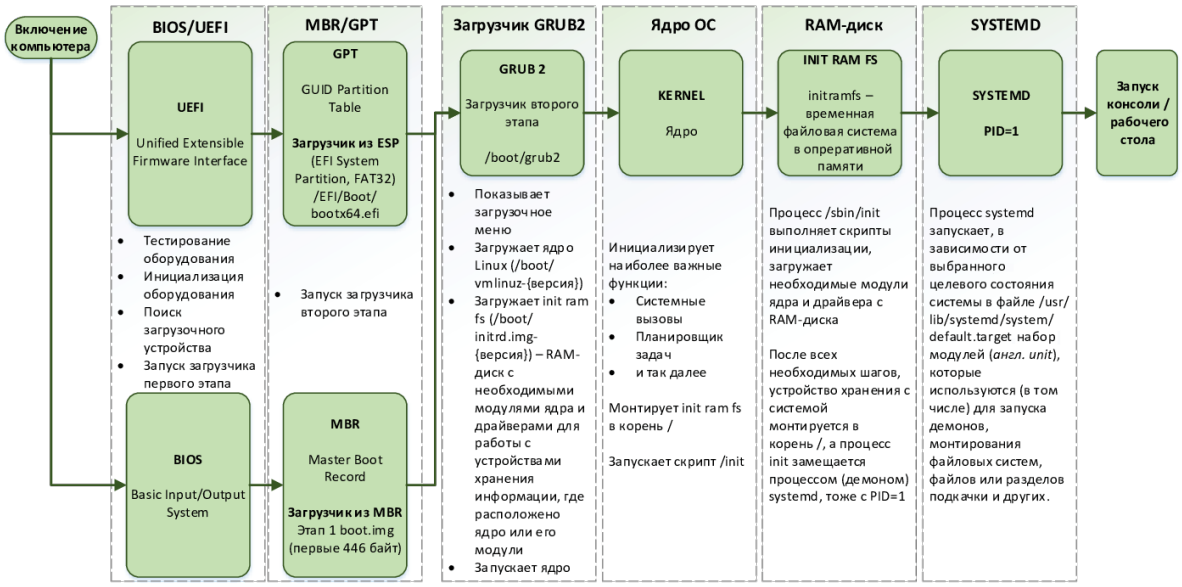


рис. 1.96 – Процесс загрузки ОС Linux

17.3 Базовые системы BIOS и UEFI

Базовая система BIOS (от англ. Basic Input-Output system — базовая система ввода-вывода) — это специальная программа, которая хранится в микросхеме на материнской плате и автоматически запускается при включении компьютера. На техническом сленге такие программы называют прошивками.

Основное предназначение базовой системы состоит в том, чтобы проверить аппаратное обеспечение (выполнить так называемый POST от англ. Power-On Self-Test — самотестирование при включении), инициализировать оборудование, найти загрузочное устройство и запустить загрузчик.

Однако BIOS морально устарел еще лет 20 назад, поэтому в 2007 году с подачи Intel и при активном участии Microsoft был утвержден новый индустриальный стандарт UEFI (от англ. Unified Extensible Firmware Interface — унифицированный интерфейс расширяемой прошивки). Предназначение UEFI то же самое, что и у BIOS, но теперь это не простая прошивка, а целая операционная система в миниатюре, которая обладает модульной архитектурой и позволяет разработчикам железа расширять ее возможности с помощью исполняемых EFI-приложений.

Например, с помощью UEFI-приложений можно реализовать шифрование диска, низкоуровневые инструменты диагностики железа и даже снимать PNG-скриншоты прямо из консоли. А для того чтобы защитить компьютер от вредоносных программ, в системе UEFI реализована технология Secure Boot, которая позволяет автоматически проверять цифровые подписи к приложениям.

Более того, вы можете выполнить команду `file /boot/efi/EFI/Boot/bootx64.efi` и убедиться, что загрузчик GRUB для UEFI тоже является EFI-приложением, которое не напрямую управляет про-

цессором, а пользуется API этой системы.

Прошивки для материнских плат персональных компьютеров разрабатываются несколькими компаниями, среди которых широко известны Award, AMI, Phoenix, но каждый производитель железа адаптирует их потом под себя и даже под конкретные модели плат, поэтому визуальный интерфейс и набор доступных настроек может сильно отличаться. Как говорится, в любой непонятной ситуации — RTBM!

17.4 Разметки диска MBR и GPT

Разметка диска MBR (от англ. Master Boot Record — основная загрузочная запись) пришла на замену BPB для MS-DOS, и ее появление было связано с увеличением емкости жестких дисков. Если раньше размер носителей исчислялся десятками мегабайт, то с начала 80-х они начали расти, как на стероидах, и разработчики задумались о немыслимом... что нужно иметь возможность устанавливать две, три и даже четыре операционные системы на один диск!

Запись MBR занимает один сектор в самом начале диска, поэтому ее размер составляет 512 байт, из которых 446 байт выделяются под программный код главного загрузчика, еще 4 блока по 16 байт составляют таблицу разделов диска, а последние 2 байта содержат сигнатуру 0x55AA, указывающую на то, что этот сектор является загрузочным.

Да, вы все правильно поняли, в таблице разделов MBR всего 4 записи, но это не так страшно, так как операционные системы на программном уровне научились работать с любым количеством расширенных разделов, а устанавливать больше 2-3 систем на один диск не так, чтобы очень нужно.

Гораздо сложнее с размерами этих записей. На описание смещения и размер диска выделяется всего по 4 байта, что при фиксированном размере сектора в 512 байт позволяет адресовать не более 2^{32} * 512 байт 00A0 2 ТБ, а это по современным меркам, прямо скажем, не много.

Чтобы решить ряд накопившихся проблем, в т.ч. обеспечить поддержку дисков размером более 2 ТБ, в рамках стандарта UEFI была разработана разметка GPT (от англ. GUID Partition Table — таблица разделов GUID). Эта разметка позволяет адресовать до 2^{64} секторов, что при стандартном размере сектора в 512 байт позволяет работать с дисками размером до 9.44 зеттабайт (1 ЗБ = 10^{21} байт), а при использовании секторов размером 4 096 байт максимальное значение увеличивается в 8 раз до 75.52 ЗБ.

Может показаться, что столько не надо, т.к. самые большие жесткие диски топчутся сейчас на отметке в несколько десятков терабайт. Но ведь можно объединить несколько таких дисков в один большой виртуальный диск, и на данный момент уже известно о создании виртуальных дисков на сотни петабайт, поэтому такие ограничения уже не видятся столь избыточными.

Еще одной отличительной особенностью GPT является то, что в части загрузки операционной

системы эта разметка опирается на расширенные возможности UEFI, поэтому загрузочных секторов больше нет, а загрузчик представляет собой EFI-приложение, которое лежит в виде файла в специальном разделе ESP (*англ. EFI System Partition — системный раздел EFI*), который отформатирован в привычной системе FAT32, см. рис 2. Кстати, проверьте, что у вас в системе покажет приложение GPart.

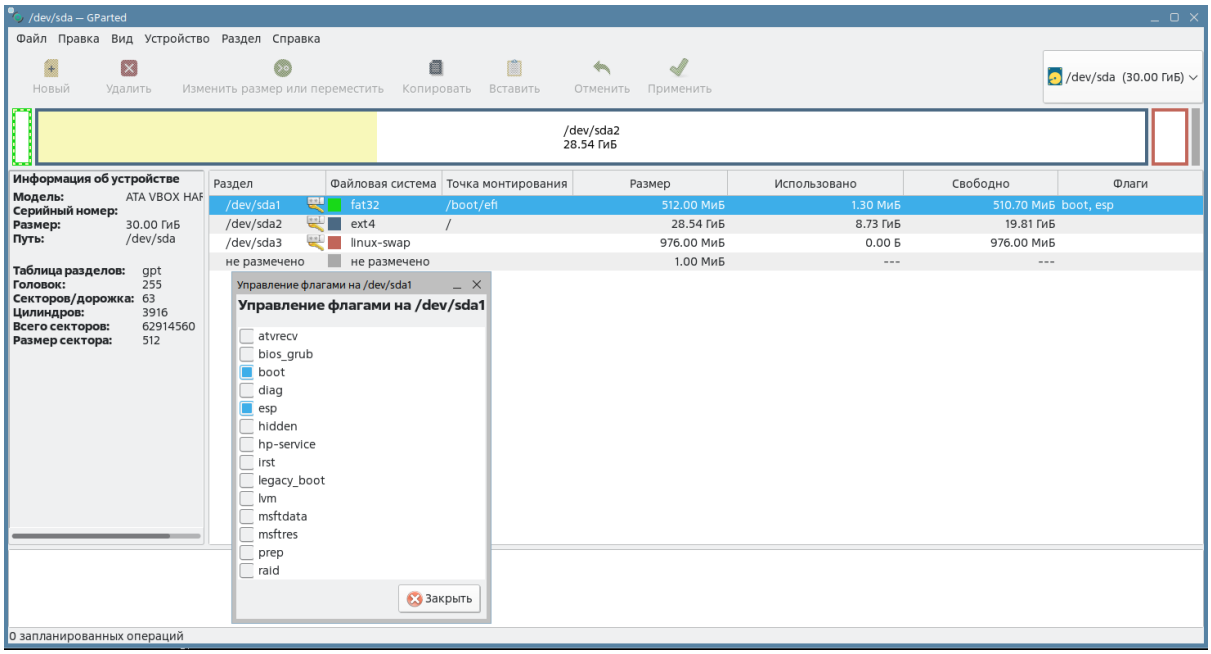


рис. 1.97 – Системный раздел EFI на диске с ALSE 1.7.x

Однако, несмотря на то, что загрузочные секторы теперь действительно больше не используются для защиты от случайного форматирования при работе пользователя со старыми приложениями, разметка GPT мимикрирует под MBR, размещая в нулевом секторе диска фейковую запись, которая указывает, что весь объем отдан под раздел GPT. Эту запись еще называют защитной (от англ. protective), а реальная таблица разделов находится дальше по диску, см. рис. 1.98.

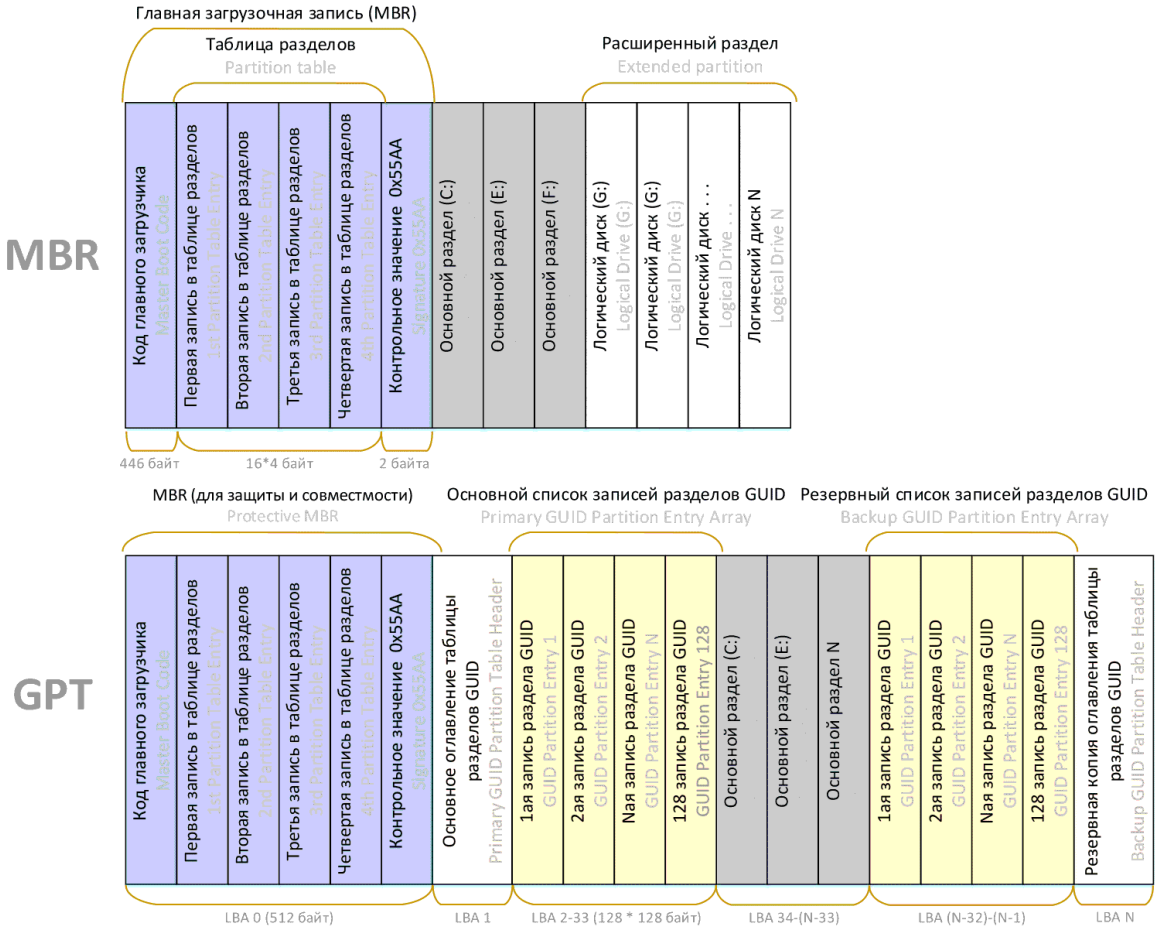


рис. 1.98 – Защитная MBR-запись разметки GPT

Понимая отличия MBR и GPT разметки, мы готовы перейти к рассмотрению работы загрузчиков.

17.5 Загрузчик GRUB2

Мы уже упоминали загрузчик GRUB, но он не является единственным — есть еще LILO, Syslinux и другие. Но в операционной системе ALSE выбор сделан именно в пользу GRUB2, который является эталонной реализацией загрузчика для Linux (от англ. GRand Unified Bootloader ver2 — главный универсальный загрузчик второй версии).

Этот загрузчик позволяет пользователю выбрать одну из доступных операционных систем и параметры ее запуска. Он был разработан с учетом опыта первой реализации, чтобы упростить поддержку кода и сделать его безопаснее, надежнее и чище, поэтому со временем смог полностью вытеснить старую версию. Если сейчас говорят GRUB, то подразумевают GRUB2, а предыдущую реализацию обычно называют теперь GRUB Legacy.

У загрузчика GRUB2 есть две версии: одна для BIOS (boot.img + core.img), которая управляет

процессором и взаимодействует с устройствами напрямую, и вторая для UEFI (bootx64.efi), которая использует API этой системы. Но у загрузчика в любом случае есть свои собственные драйверы файловых систем, шифрования дисков, взаимодействия с TPM и др. Обладая модульной архитектурой, GRUB2 позволяет подключать компоненты по мере надобности, что значительно расширяет его функциональность.

Учитывая поддержку спецификации Multiboot, GRUB2 может загрузить любую совместимую с ней операционную систему, среди которых: Linux, FreeBSD, Solaris и многие другие. Кроме того, GRUB2 может по цепочке передать управление другому загрузчику, что позволяет загружать Windows (через загрузчик NTLDR или bootmgr), MS-DOS, OS/2 и другие системы.

Загрузчик GRUB2 выполняет следующие задачи:

- Отображает меню для выбора операционной системы.
- Загружает выбранное ядро Linux, например, из файла `boot/vmlinuz-5.15.0-70-generic`
- Монтирует диск с модулями ядра и другими файлами системы в оперативную память с помощью initramfs, например, из файла `boot/initrd.img-5.15.0-70-generic`
- Запускает ядро.

Кроме этого, загрузчик позволяет изменять параметры загрузки ядра ОС и работать с командной строкой. Устройство загрузчика и работу с ним мы подробно разберем далее.

17.6 Запуск загрузчика

По историческим причинам сложилось много разных способов загрузки ОС. Какие-то из них чаще применяются в Windows, какие-то в Linux, но достаточно понять основной принцип, чтобы все встало на свои места.

17.6.1 Способ загрузки BIOS > MBR > PBR

Системы BIOS работают с разметкой диска MBR, в соответствии с которой они передают управление загрузчику из основной загрузочной записи, находящейся в нулевом секторе диска. И далее загрузка операционной системы выполняется в два этапа:

- **На первом этапе** загрузчик из MBR просматривает таблицу разделов и передает управление по цепочке загрузчику из нулевого сектора активного раздела, в котором находится загрузочная запись раздела PBR (от англ. Partition Boot Record, но в ряде источников ее называют Volume Boot Record, VBR).

Загрузчик из PBR загружает драйвер файловой системы и передает управление основному загрузчику операционной системы, который представлен уже в виде файлов, а не байтов управляющих инструкций, записанных в секторах на диске.

- **На втором этапе** основной загрузчик уже располагает всем необходимым, чтобы выполнить загрузку операционной системы, поэтому он начинает загружать ядро со всеми его модулями и передает ему управление.

Однако в этой схеме между первым и вторым этапами есть еще один этап, который называют «**этап 1.5**» (или полтора). Возникает он по причине того, что запись PBR должна укладываться в один сектор размером всего 512 байт, чего явно недостаточно для размещения сколь-нибудь существенной программы.

По указанной причине программный код загрузчика делят на две части, которые нужно устанавливать обязательно вместе, так как они являются фрагментами одной программы. Первую часть загрузчика (этап 1) размещают в PBR, и ее задача заключается лишь в том, чтобы загрузить первый сектор второй части загрузчика (этап 1.5) и передать управление этому коду.

Программный код второго загрузчика прячут между нулевым сектором раздела, в котором расположена загрузочная запись, и 63-м сектором, с которого начинаются данные. Эта область свободна и не является частью какого-либо раздела из соображений совместимости с DOS, поэтому может быть использована без каких-либо опасений.

Благодаря такому трюку загрузчик второго этапа может представлять собой уже довольно сложную программу, которая способна работать даже с файловыми системами FAT32. Например, старый загрузчик NTLDR (от англ. NT Loader) использовал параметры запуска ядра из файла `boot.ini` в корне системного диска, а современный загрузчик BOOTMGR (от англ. Boot Manager) хранит эти настройки в файле реестра /Boot/BCD на скрытом служебном разделе System.

Осторожно: Говорят, что эта схема настолько проста, что и ежу должно быть все понятно… Но откуда тогда в Интернете взялось 30к страниц с обсуждением проблемы загрузки Windows, которые ведут на статью kb 315261 «Компьютер не запускается после изменения активной секции с помощью средства управления дисками»?! В общем, будьте осторожнее с командой «Mark Partition as Active», а то у этого ежа какой-то повышенный IQ.

Ничто не мешает применить для загрузки Linux этот сценарий и установить GRUB в раздел диска, как загрузчик BOOTMGR или NTLDR, но так делают довольно редко. Обычно GRUB устанавливают напрямую в запись MBR, что будет рассмотрено далее.

17.6.2 Способ загрузки BIOS > MBR

Во времена, когда пользователями компьютеров были исключительно специалисты с очень высокой квалификацией, использовать флажок активности раздела для переключения между установленными операционными системами казалось вполне логичным решением. Компьютер, правда, можно было сломать, а так очень удобное решение.

Однако в дальнейшем из соображений удобства от такого подхода (внезапно) решили отказаться, и даже загрузчик NTLDR уже предлагал меню для выбора ОС. А раз уж информация об активности диска больше не требовалась, то стало возможным упростить цепочку и устанавливать основной загрузчик сразу в MBR. Именно так и поступает загрузчик GRUB, см. рис. 1.99.

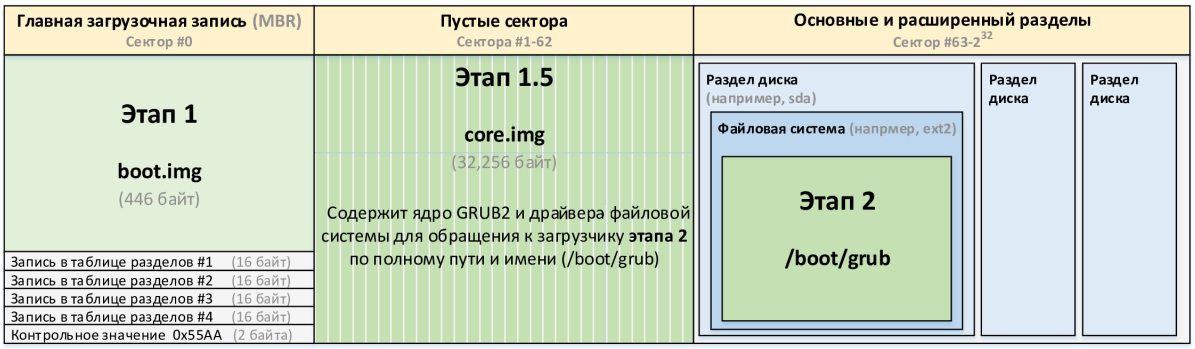


рис. 1.99 – Структура MBR в разрезе размещения загрузчиков при использовании BIOS

Загрузчик GRUB первого этапа так же, как в Windows, представлен в двух частях. Первая часть boot.img является простым загрузчиком, который размещается в основной загрузочной записи MBR (этап 1), и его единственная задача заключается в том, чтобы загрузить вторую часть core.img (этап 1.5) и передать ей управление. Программный код core.img, в свою очередь, загружает модули GRUB уже из папки /boot/grub (этап 2), с помощью которых выполняет дальнейшую загрузку операционной системы.

Это классический способ загрузки, который может применяться на устаревших системах.

Внимание: Установщик Windows конфликтует с grub-install, поскольку затирает программный код основной загрузочной записи и возвращает компьютер к использованию информации об активности разделов. Поэтому после установки Windows на тот же диск требуется восстановить код загрузчика GRUB, например, загрузившись через Live CD.

17.6.3 Способ загрузки BIOS > GPT и UEFI > MBR

Во время перехода индустрии с MBR на GPT появилось несколько Франкенштейнов, которых лучше избегать.

Во-первых, стандарт разметки GPT появился чуть позже UEFI, поэтому были релизы этой прошивки, которые поддерживали загрузку с MBR-дисков. И тут важно понимать, что подключить диск с разметкой MBR к современному компьютеру с UEFI не является проблемой, но вот использовать его в качестве загрузочного не стоит, даже если прошивка это позволяет.

Примечание: Допустимым считается загрузка с MBR-дисков на современных компьютерах с UEFI, если включена опция Legacy Boot режим (Compatibility Support Mode). В этом случае ваша прошивка работает как обычный BIOS.

Во-вторых, пользователям компьютеров с BIOS очень не нравилось, что систему можно было устанавливать только на диски объемом до 2Тб, иначе все остальное пространство становилось недоступным. Для снятия этого ограничения разработчики воспользовались обратной совместимостью GPT и MBR.

Как мы обсуждали выше, во избежание случайного удаления данных с GPT-диска в нулевом секторе располагается «защитная» MBR-запись, указывающая на то, что весь объем диска отдан под раздел GPT. В эту запись и стали записывать код загрузчика, которому BIOS передавал управление в штатном порядке. И оставалось только, чтобы код этого загрузчика и сама операционная система могли работать с диском в разметке GPT.

Еще раз отметим, что оба этих сценария не представляют практического интереса, поэтому не будут рассматриваться более подробно. Использовать разметку GPT с BIOS или MBR с UEFI нежелательно, а если вы переключите флажок «Включить EFI» для виртуальной машины с уже установленной системой, она просто не взлетит.

17.6.4 Способ загрузки UEFI > GPT

Как мы уже знаем, базовые системы UEFI могут работать с разделами диска, отформатированными в FAT32, поэтому позволяют полностью отказаться от каких-либо загрузочных секторов и загружать все нужные файлы напрямую из защищенного раздела ESP.

Так делает Windows, так поступает и GRUB. Например, EFI-файл, содержащий код загрузчика, можно найти по адресу `EFI/Boot/bootx64.efi` (копия `EFI/astralinux/grubx64.efi`), см. рис. 1.100.

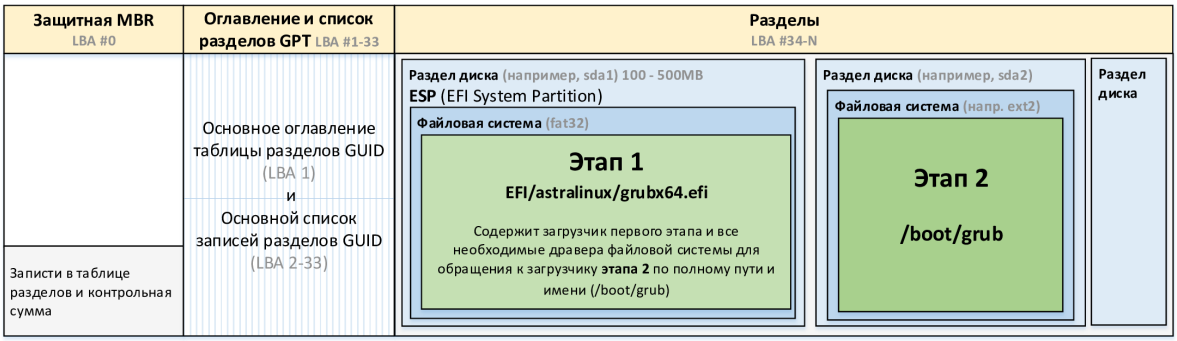


рис. 1.100 – Структура GPT в разрезе размещения загрузчиков при использовании EFI

Сейчас такой способ загрузки считается наиболее правильным и настоятельно рекомендуется всеми разработчиками операционных систем.

17.6.5 Способ загрузки PXE

Можно еще вспомнить про PXE (англ. Preboot eXecution Environment, предзагрузочная среда выполнения), когда загрузка начального исполняемого кода выполняется с сетевого хранилища, что используется в том числе для установки ОС по сети и обеспечения работы тонких клиентов.

В этом случае сетевая карта под капотом просто реализует один из представленных выше сценариев для BIOS или UEFI. Загрузчик первого этапа находится в ПЗУ сетевой карты и его задача состоит в том, чтобы сымитировать загрузочное устройство, используя файлы загрузчика с TFTP или HTTP-сервера. Адрес сервера и имя файла с загрузчиком сетевая карта получает от DHCP через опции 66 и 67 соответственно.

17.7 Работа загрузчика GRUB2

Рассмотрим работу загрузчика GRUB2 и этапы загрузки ОС, которые проходят под его управлением. Основные параметры, определяющие работу загрузчика это:

- Образ ядра и параметры его запуска
- Образ инициализационной файловой системы initramfs
- Корневая файловая система

17.7.1 Работа с меню загрузки

После начала своей работы GRUB2 выводит на экран загрузочное меню. По умолчанию в нем доступно только 2 пункта - загрузка системы и загрузка системы в режиме восстановления. Перезагрузим виртуальную машину и нажмем клавишу <Вверх> при появлении загрузочного меню, чтобы остановить таймер.

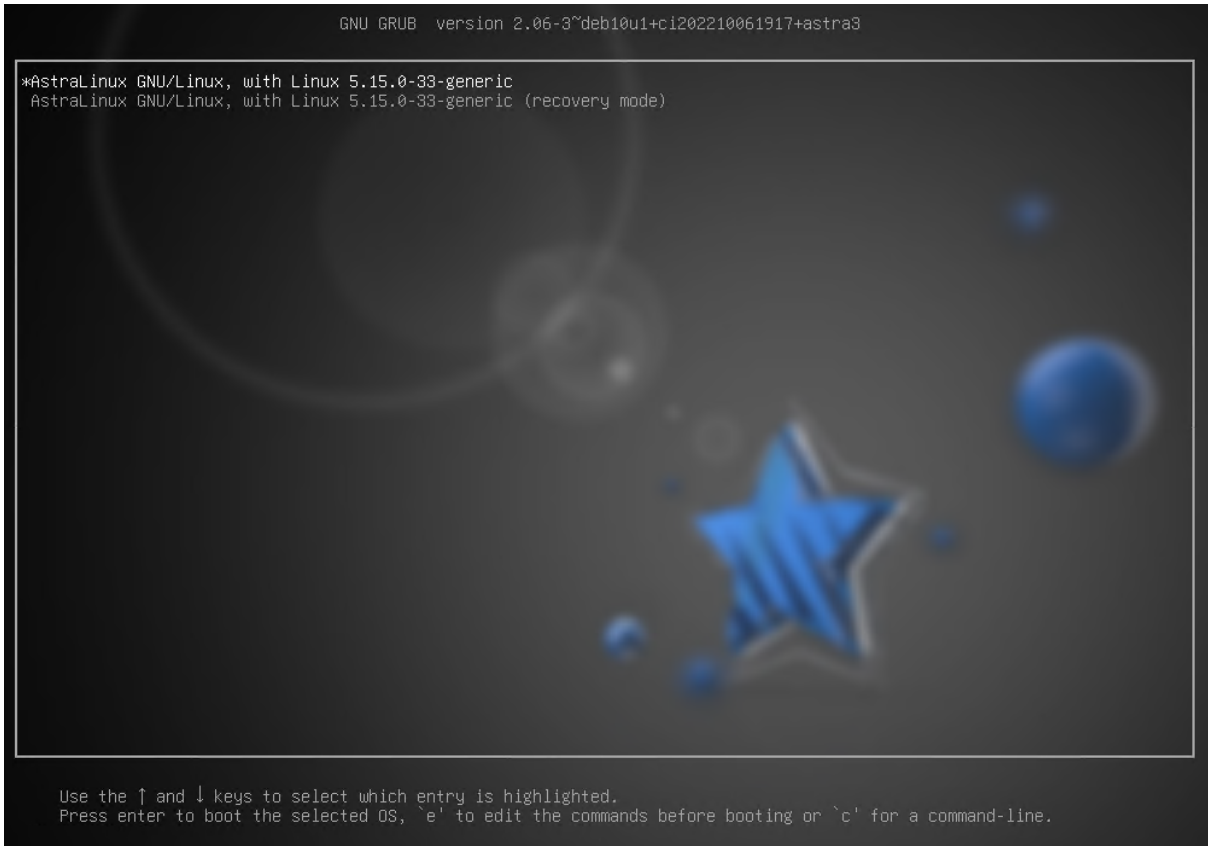


рис. 1.101 – Загрузочное меню GRUB2

Внизу страницы будет представлена справка по способам управления: с помощью стрелок можно переходить по пунктам меню, клавишей e можно отредактировать команды запуска перед загрузкой, а клавишей c - перейти в режим командной строки.

17.7.2 Просмотр и редактирование команд запуска

Перейдем в режим редактирования команд запуска. Система попросит ввести учетные данные пользователя, созданного при установке GRUB. Имя пользователя будет совпадать с именем локального администратора, в нашем случае это localadmin.

После ввода корректных учетных данных будут отображены команды запуска. Текст на экране доступен для редактирования, но пока в нем лучше ничего не менять (эти изменения не сохраняются и влияют только на текущий запуск ОС. Если была допущена ошибка и ОС не может загрузиться,

то перезагрузка решит эту проблему).



рис. 1.102 – Команды запуска ядра ОС для BIOS+MBR

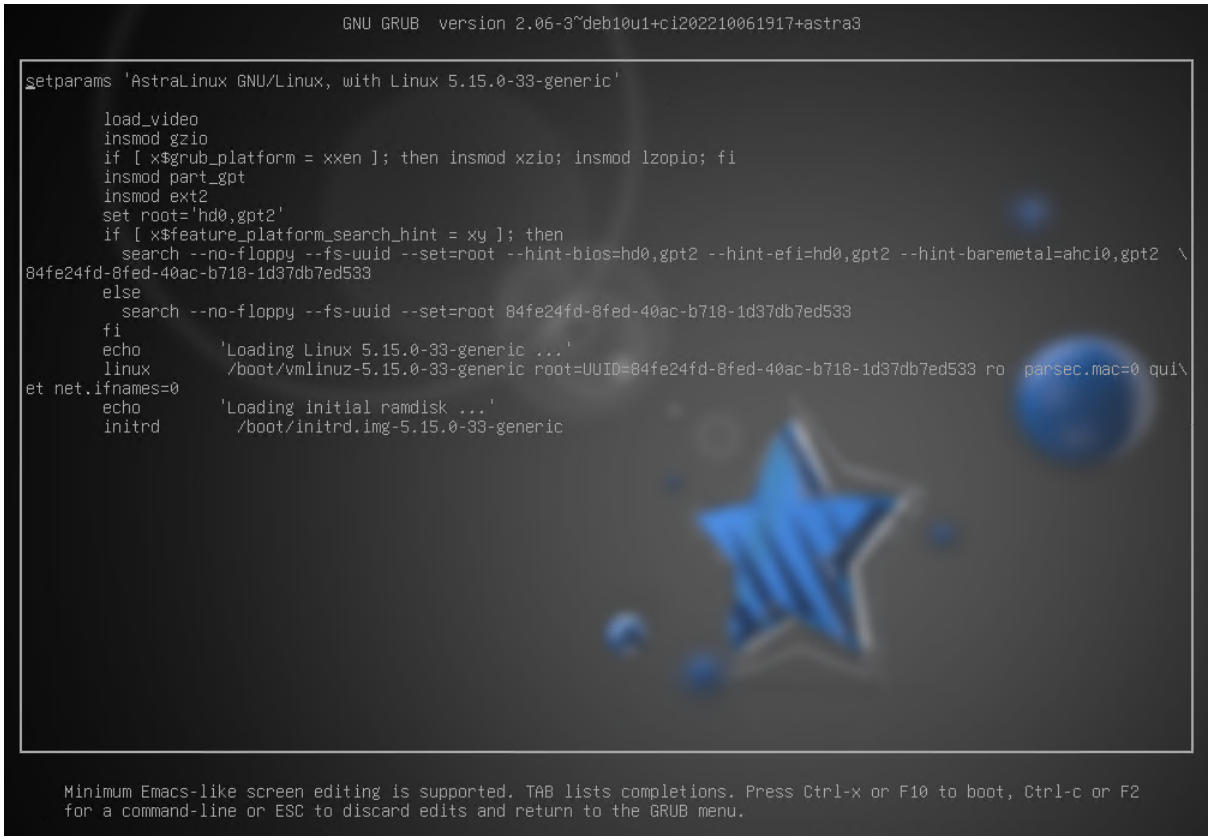


рис. 1.103 – Команды запуска ядра ОС для UEFI+GPT

Разберем, что означает каждая из этих команд:

- **load_video** – вызывает функцию, которая подгрузит видео-модули GRUB2.
- **insmod** {имя_модуля} – подгружает модуль по его имени. Модули GRUB2 хранятся в каталоге /boot/grub/x86_64-efi.
- **set/unset** {имя_переменной}={значение} – как и при работе в консоли - задает или удаляет переменную.
- **search** – ищет устройство по файлу (`--file`), по имени файловой системы (`--label`) или по UUID файловой системы (`--fs-uuid`). Параметр `--set` - задает переменную, параметр `--no-floppy` - не производит поиск дисководов дискет.
- **linux** {путь_до_образа_ядра_ОС и параметры запуска} – загружает ядро ОС с заданными параметрами
  - root=UUID=... – корневая файловая система.
  - ro – монтирование корневого каталога только на чтение (*англ. read-only*).
  - parsec.mac=0 – отключение мандатного управления доступом (1 - включено, 0- выклю-

чено).

- quite - «тихая» загрузка без вывода информационных сообщений ядра (процессы активации драйверов и модулей системы, проверки файловой системы).
- net.ifnames=0 - использование традиционного названия интерфейсов (eth0, eth1 . . .).

- **initrd** {путь_до_образа_RAM-диска} - загружает RAM-диск для ядра ОС.

Ниже представлен список некоторых других параметров, которые могут быть переданы ядру:

- **splash** - разрешить отображать графическую заставку
- **rw** - монтировать корневой каталог на чтение/запись (вместо ro)
- **s** - загрузка в режиме восстановления (single mode)
- **3** - загрузка в режиме multi-user (без графики)
- **5** - загрузка в режиме graphical (графический режим)

17.7.2.1 Командная строка GRUB2

Теперь переключимся в режим командной строки, нажав F2.

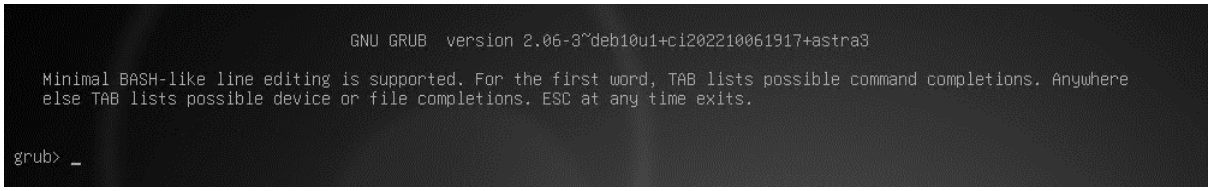


рис. 1.104 – Командная строка GRUB2

Нам доступна командная строка GRUB2. Командой `help` можно вывести список всех доступных команд. Перед её вызовом можно задать постраничный вывод командой `set pager=1` (перемотки или привычной команды `less` тут нет).



рис. 1.105 – Командная строка GRUB2

Возможности командой строки GRUB2 широки, но мы не будем их подробно рассматривать. Однако для примера можно воспользоваться командой `ls` и вывести информацию о разделе с системой и содержимым файловой системы:

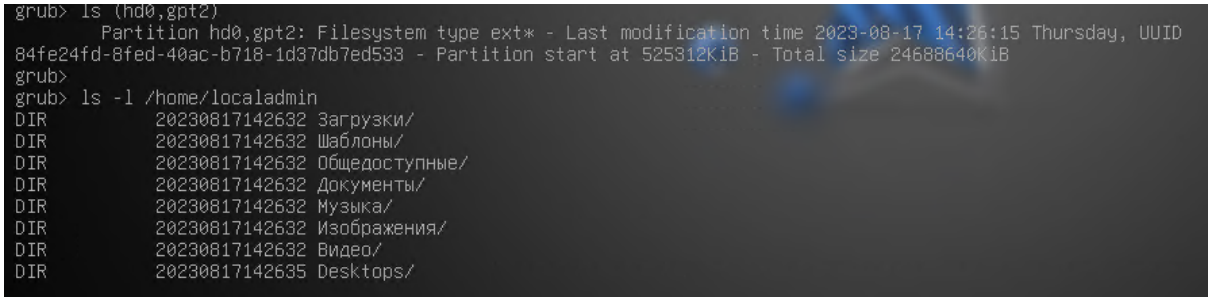
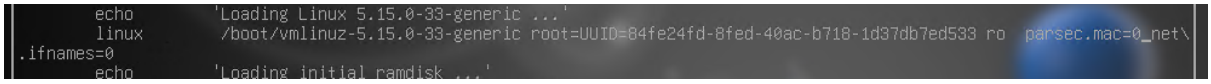


рис. 1.106 – Командная строка GRUB2

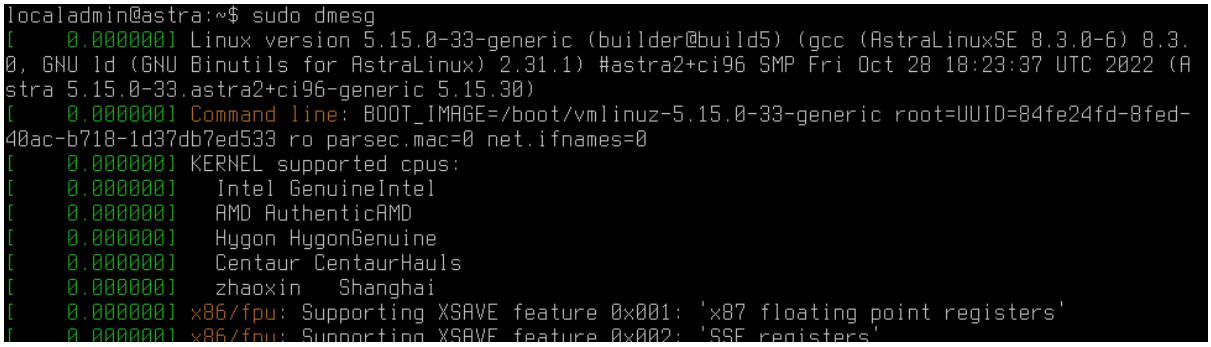
Командная строка GRUB2 может быть полезна при решении возникающих проблем, например, в том случае, когда конфигурационный файл был утрачен/повреждён или, как вариант, имя загрузочного устройства (hd0) было изменено в результате подключения дополнительных жёстких дисков.

Вернемся в окно просмотра и изменения команд загрузки, удалим параметр **quite** в команде `linux` и нажмем F10 для загрузки:



Как видите, в этот раз кроме статуса запуска демонов была выведена и другая информация о запуске ОС.

С помощью команды `dmesg` можно посмотреть все сообщения ядра ОС (от момента начала загрузки до текущего). После загрузки ОС откройте терминал, введите команду `sudo dmesg` и вы увидите подробный лог загрузки ОС.



Примечание: Инструкция по изменению параметров загрузчика доступна по ссылке: https://wiki.astralinux.ru/pages/viewpage.action?pageId=18874503

Инструкция по установке загрузчика GRUB2 в UEFI GPT доступна по ссылке: https://wiki.astralinux.ru/pages/viewpage.action?pageId=130424699

С полным руководством по GRUB и списку его команд на английском языке можно ознакомиться по ссылке: https://www.gnu.org/software/grub/manual/grub/grub.html

---

17.7.3 Загрузка ядра и RAM-диска

Файлы образа ядра `/boot/vmlinuz-`*версия_ядра* и инициализационной файловой системы initramfs `/boot/initrd.img-`*версия_ядра* – это сжатые архивные файлы.

GRUB производит распаковку и декомпрессию образа `initrd.img-`*версия_ядра*, а файл `vmlinuz-`*версия_ядра* распаковывает себя сам (подробнее об этом в разделе, посвященном запуску ядра ОС). После этого управление передается ядру ОС.

17.7.4 Настройка GRUB2

Работу кода загрузчика из образов `core.img` или `grubx64.efi` определяет файл с конфигурацией `/boot/grub/grub.cfg`. По своей сути, файл `/boot/grub/grub.cfg` – это довольно сложный sh-скрипт.

Если мы откроем этот файл, то увидим предупреждение о том, что этот файл автоматически создан командой `grub-mkconfig` на основе шаблонов из каталога `/etc/grub.d/` и файла с настройками `/etc/default/grub`.

Таким образом, внесение изменений напрямую в файл `/boot/grub/grub.cfg` – не только довольно сложная и сопряженная с риском ошибок задача, но и внесенные руками изменения будут перезаписаны при очередном запуске команды `update-grub`.

17.7.4.1 Шаблоны grub

Каталог `/etc/grub.d/` содержит шаблоны, которые могут быть использованы в качестве опций в загрузочном меню. Шаблоны — это сложные sh-скрипты. Цифра в начале шаблона определяет порядок его запуска при формировании нового конфигурационного файла `grub.cfg`. Как правило, добавление новых шаблонов и их редактирование не требуется. Для редактирования (в случае необходимости изменения загрузочного меню) предназначен файл `/etc/grub.d/40_custom`.

```
root@astra:/boot# ls -l /etc/grub.d
итого 92
-rwxr-xr-x 1 root root 10046 окт  6  2022 00_header
-rwxr-xr-x 1 root root  6260 авг  1  2022 05_debian_theme
-rwxr-xr-x 1 root root   353 мая 16 12:06 07_password
-rwxr-xr-x 1 root root 14282 окт  6  2022 10_linux
-rwxr-xr-x 1 root root 14180 окт  6  2022 20_linux_xen
-rwxr-xr-x 1 root root 12910 окт  6  2022 30_os-prober
-rwxr-xr-x 1 root root  1372 окт  6  2022 30_uefi-firmware
-rwxr-xr-x 1 root root   214 окт  6  2022 40_custom
-rwxr-xr-x 1 root root   215 окт  6  2022 41_custom
-rwxr-xr-x 1 root root   370 окт 15  2020 99_kcm_grub2
-rw-r--r-- 1 root root   483 окт  6  2022 README
```

17.7.4.2 Параметры grub

Файл `/etc/default/grub` содержит набор параметров (переменных) с комментариями, которыми можно управлять работой загрузчика. Посмотрим его содержимое:

```
# If you change this file, run 'update-grub' afterwards to update
# /boot/grub/grub.cfg.
# For full documentation of the options in this file, see:
#   info -f grub -n 'Simple configuration'

#GRUB_DEFAULT=0
```

(continues on next page)

(продолжение с предыдущей страницы)

```
GRUB_TIMEOUT=5
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
GRUB_CMDLINE_LINUX_DEFAULT="parsec.mac=0 quiet net.ifnames=0"
GRUB_CMDLINE_LINUX=""
GRUB_CMDLINE_LINUX_HARDENED="slub_debug=P page_poison=1 slab_nomerge pti=on
user.max_user_namespaces=0 kernel.kptr_re
strict=1 vsyscall=none"

# Uncomment to enable BadRAM filtering, modify to suit your needs
# This works with Linux (no patch required) and with any kernel that obtains
# the memory map information from GRUB (GNU Mach, kernel of FreeBSD ...)
#GRUB_BADRAM="0x01234567,0xfefefefe,0x89abcdef,0xefefefef"

# Uncomment to disable graphical terminal (grub-pc only)
#GRUB_TERMINAL=console

# The resolution used on graphical terminal
# note that you can use only modes which your graphic card supports via VBE
# you can see them in real GRUB with the command \`vbeinfo'
GRUB_GFXMODE=1024x768

# Uncomment if you don't want GRUB to pass "root=UUID=xxx" parameter to Linux
#GRUB_DISABLE_LINUX_UUID=true

# Uncomment to disable generation of recovery mode menu entries
#GRUB_DISABLE_RECOVERY="true"

# Uncomment to get a beep at grub start
#GRUB_INIT_TUNE="480 440 1"
GRUB_DISABLE_SUBMENU=y
GRUB_DEFAULT=gnulinux-5.15.0-70-generic-advanced-1864fbac-4e75-42ec-9a52-c0944d4fd6b9
```

Вот некоторые параметры:

- **GRUB_TIMEOUT** - определяет продолжительность отображения загрузочного меню;
- **GRUB_DEFAULT** - пункт загрузочного меню по умолчанию;
- **GRUB_CMDLINE_LINUX** - значение этого параметра будет передано в конец строки linux (в старой версии GRUB эти параметры использовались в строке kernel) как при нормальной загрузке, так и при загрузке в режиме восстановления;
- **GRUB_DISABLE_LINUX_RECOVERY=true** - для того, чтобы избежать появления в меню режима Recovery. Если нужен режим Recovery только для конкретно взятого ядра, нужно создать необходимую запись в скрипте /etc/grub/40_custom.

17.7.4.3 Просмотр новой конфигурации grub и обновление файла grub.cfg

Команда `grub-mkconfig` на основании шаблонов (скриптов) из `/etc/grub.d/` и файла с параметрами `/etc/default/grub` формирует текст конфигурационного файла и выводит его на экран.

Для создания (перезаписи) конфигурационного скрипта GRUB `/boot/grub/grub.cfg` необходимо выполнить команду `update-grub` (или `grub-mkconfig -o /boot/grub/grub.cfg`). Команда `update-grub` представляет собой скрипт:

```
#!/bin/sh
set -e
exec grub-mkconfig -o /boot/grub/grub.cfg "$@"
```

Примечание: Дополнительную информацию об изменении параметров загрузчика GRUB2 можно найти по ссылке: https://wiki.astralinux.ru/pages/viewpage.action?pageId=18874503.

По ссылке ниже вы можете ознакомиться с инструкцией по установке GRUB2 после замены таблицы разделов на устройстве хранения с MBR на GPT https://wiki.astralinux.ru/pages/viewpage.action?pageId=130424699.

17.8 Запуск ядра ОС

Файл образа ядра `/boot/vmlinuz-`*`версия_ядра`* представляет собой архивный загрузочный исполнимый образ. Если мы посмотрим на тип файла командой `file`, то мы увидим следующее:

```
localadmin@astra:~/initrd/initramfs$ sudo file /boot/vmlinuz-5.15.0-70-generic
[sudo] пароль для localadmin:
/boot/vmlinuz-5.15.0-70-generic: Linux kernel x86 boot executable bzImage, version 5.
↪15.0-70-
generic (builder@build5)  #astra2 SMP Mon Apr 10 13:35:08 UTC 2023, RO-rootFS, swap_
↪dev 0xA, Normal VGA
```

На самом деле это не совсем ядро, точнее не только оно. Содержимое файла `/boot/vmlinuz*` представляет собой заголовок + код настройки ядра + vmlinux (сжатое ядро), что превращает его в своего рода самораспаковывающийся архив, а название файла образа ядра состоит из **vm** - виртуальная память (наследие с зари разработки Linux), **linu** - часть linux, **z** - сокращение от «zipped» и номера версии ядра.

Ядро запускается функцией start_kernel(). При этом инициализируются наиболее важные функции, такие как системные вызовы, планировщик задач и другие, а также создаются все необходимые потоки ядра.

После загруженный в память образ initrd (за что отвечает GRUB) монтируется в корень в качестве временной файловой системы (за что отвечает ядро ОС), имеющей необходимый набор драйверов. Это позволяет работать с различным оборудованием, сохранять небольшой размер ядра и не пересобирать ядро под ту или иную конфигурацию аппаратного обеспечения. Такой механизм обеспечивается модульностью ядра Linux.

17.8.1 Содержимое RAM-диска

Содержимое RAM-диска представляет собой FHS-совместимую структуру каталогов, busybox, необходимые модули ядра и драйверы устройств. BusyBox - это набор UNIX-утилит командной строки, содержащихся в одном файле. Командным интерпретатором в нем выступает ash, совместимый с sh.

Посмотреть содержимое файла `initrd.img` можно командой `lsinitramfs /boot/initrd.img-$(uname -r)`, где: команда `$(uname -r)` выведет текущую версию ядра и имя ядра (generic).

```
localadmin@astra:$ lsinitramfs /boot/initrd.img-5.15.0-70-generic | head
.
bin
conf
conf/arch.conf
conf/conf.d
conf/conf.d/resume
conf/conf.d/splash
conf/initramfs.conf
conf/modules
etc
```

Для распаковки и изучения содержимого необходимо скопировать файл `initrd.img-$(uname -r)`, произвести декомпрессию и скопировать файлы из архива командами

```
localadmin@astra:~$ mkdir initrd && cd initrd && cp /boot/initrd.img-$(uname -r) .
localadmin@astra:~/initrd$ mkdir initramfs && cd initramfs
localadmin@astra:~/initrd/initramfs$ gzip -cd ../initrd.img-$(uname -r) | cpio -idmv
. . .
cpio: создана ссылка usr/sbin/watchdog, указывающая на usr/sbin/uevent
cpio: создана ссылка usr/sbin/watchdog, указывающая на usr/sbin/vconfig
usr/sbin/watchdog
usr/share
usr/share/plymouth
usr/share/plymouth/plymouthd.defaults
usr/share/plymouth/themes
usr/share/plymouth/themes/details
usr/share/plymouth/themes/details/details.plymouth
usr/share/plymouth/themes/text
usr/share/plymouth/themes/text.plymouth
611579 блоков

localadmin@astra:~/initrd/initramfs$ ls -l
итого 28
lrwxrwxrwx 1 localadmin localadmin    7 авг 21 16:39   bin   ->   usr/bin
drwxr-xr-x 3 localadmin localadmin 4096 авг 21 16:43   conf
drwxr-xr-x 7 localadmin localadmin 4096 авг 21 16:43   etc
-rwxr-xr-x 1 localadmin localadmin 6338 авг 23  2019   init
lrwxrwxrwx 1 localadmin localadmin    7 авг 21 16:39   lib   ->   usr/lib
lrwxrwxrwx 1 localadmin localadmin    9 авг 21 16:39   lib32   ->   usr/lib32
lrwxrwxrwx 1 localadmin localadmin    9 авг 21 16:39   lib64   ->   usr/lib64
lrwxrwxrwx 1 localadmin localadmin   10 авг 21 16:39   libx32   ->   usr/libx32
drwxr-xr-x 2 localadmin localadmin 4096 авг 21 16:43   run
```

(continues on next page)

(продолжение с предыдущей страницы)

```
lrwxrwxrwx 1 localadmin localadmin    8 авг 21 16:39   sbin   ->   usr/sbin
drwxr-xr-x 8 localadmin localadmin 4096 авг 21 16:43   scripts
drwxr-xr-x 9 localadmin localadmin 4096 авг 21 16:43   usr
```

17.8.2 Запуск первого процесса /sbin/init с PID=0

Ядро ОС запускает скрипт `/init`, который, вызывая скрипты из каталога `/scripts`, определяет тип устройств хранения информации и файловых систем на компьютере, производит различные проверки, подгружает необходимые для них драйверы, после чего перемонтирует в качестве корня реальную файловую систему с системного диска в режиме только на чтение. Файловая система монтируется только на чтение для исключения случаев её дальнейшей деградации, если она была повреждена.

После этого скрипт запускает файл `/sbin/init`, который становится первым процессом в дереве процессов пользовательского пространства с PID=1 (как мы выяснили ранее, это символическая ссылка на `/lib/system/systemd`, что связано с историческим наследием и обратной совместимостью с прошлой системой инициализации systemV). С этого момента и до появления приглашения на ввод учётных данных для входа в систему процессом будет управлять подсистема инициализации systemd.

17.9 Подсистема инициализации и управления службами systemd

При запуске файла `/lib/system/systemd` ядро искусственно (вспомним, что рождение первого процесса происходит особым образом) создаст в своих структурах одноименный процесс с PID=1 и PPID=0. Это родитель всех остальных процессов в пользовательском пространстве. Systemd - это подсистема инициализации и управления службами, представленная комплектом базовых компонентов Linux-системы.

Подсистема systemd пришла на смену системе systemV (system-пять), в которой первый процесс именовался init. SystemV – это очень старая модель, которая имеет целый ряд недостатков, главными из которых являются медленная работа и сложность в настройке зависимостей между демонами. Systemd, созданная Леннартом Поттерингом и другими сотрудниками компании Red Hat как свободное ПО под лицензией GNU, решила эти проблемы (при этом часть концепций systemd была взята у системы инициализации Mac OC launchd).

Systemd оперирует модулями (англ. unit) - универсальными единицами, отвечающими за различные типы объектов, такие как служба (*англ. service*), монтирование (*англ. mount*), цель (*англ. target*), устройство (*англ. device*) и другие. Реализованы механизмы иерархии целевых состояний и зависимостей внутри модулей, запуск службы по запросу (*англ. on-demand*), сокеты и D-bus акти-

вация запуска служб. Это позволило интенсивно распараллелить запуск служб в процессе загрузки системы и контролировать очередность их запуска. Кроме этого, были введены механизмы контроля и изоляции процессов служб - механизмы пространства имен (*англ. namespaces*) и контрольных групп Linux (*англ. cgroups*).

На производительность systemd наибольшее влияние оказало введение механизма модулей в целом, сокетов и автомонтирование файловых систем в частности. В системе systemV вместо модулей применялся набор скриптов, при выполнении которых могли вызываться множество команд и нерационально использоваться ресурсоёмкие утилиты, например, такие как awk, что негативно сказывалось на производительности. Введение модулей (стандартизированных конфигурационных файлов) позволило оптимизировать этот процесс. Кроме того, с распараллеливанием загрузки служб были проблемы как у systemV, так и у Upstart (система, призванная заменить systemV, которая их решала, но не в полном объеме). Созданные заранее сокеты с механизмами буферизации позволили убрать зависимости между сервисами, один из которых пишет информацию в сокет, а другой читает. Более того, этот механизм позволил перезапускать службы без потери данных - после запуска сервис сможет разобрать информацию, накопившуюся в очереди сокета. Монтирование файловых систем по запросу (при первом обращении к файловой системе) тоже позволило сократить время загрузки ОС.

Примечание: В systemd вместо классического для Linux названия «демон» применяется название служба, так привычное уху администратора систем Windows. Концепции служб systemd в Linux и служб в Windows с первого взгляда чем-то похожи. Мы можем управлять их запуском, настраивать зависимости от других служб, запускать службы по событиям, однако, различий, обусловленных разными архитектурными решениями в системах, довольно много.

На рисунке ниже представлена вся система инициализации systemd. Рассмотрим подробнее каждый из её компонентов.

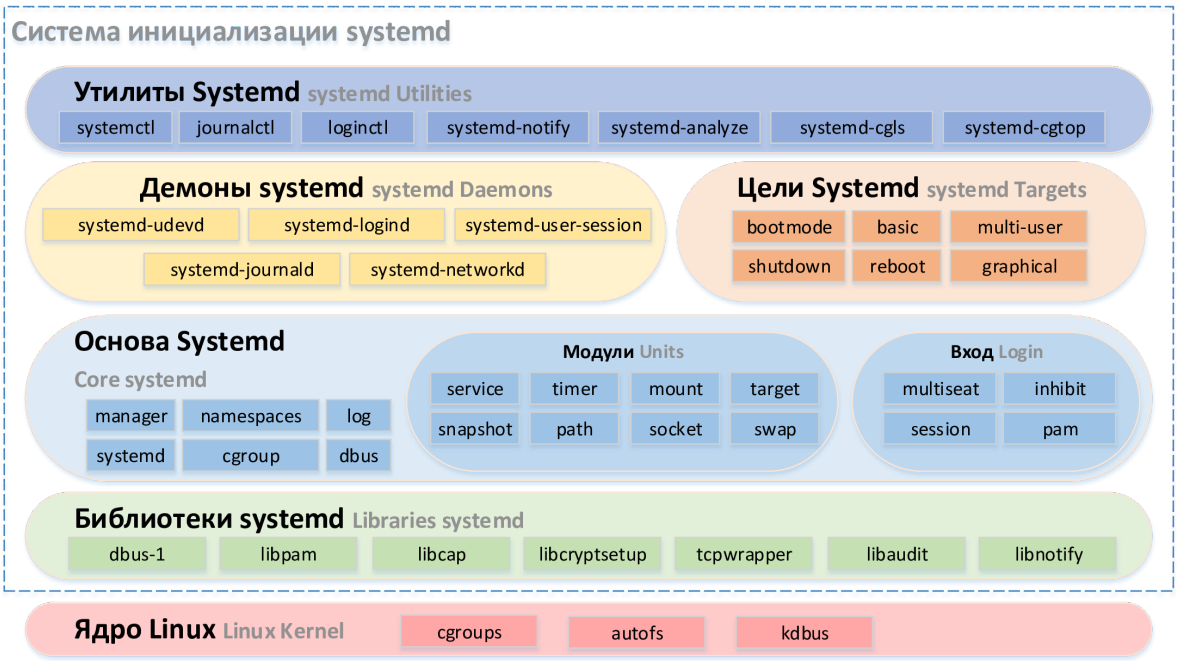


рис. 1.107 – Система инициализации systemd

17.9.1 Модули systemd

В systemd существуют следующие типы модулей:

- **.target** - позволяет группировать модули, воплощая концепцию уровней запуска;
- **.service** - отвечает за запуск сервисов (служб), также поддерживает вызов интерпретаторов для исполнения пользовательских скриптов;
- **.mount** - отвечает за монтирование файловых систем;
- **.automount** - позволяет отложить монтирование файловых систем до фактического обращения к точке монтирования;
- **.swap** - отвечает за подключение файла или устройства подкачки;
- **.timer** - позволяет запускать модули по расписанию;
- **.socket** - предоставляет службам поддержку механизма сокет-активации;
- **.slice** - отвечает за создание контейнера cgroups - группы иерархически организованных модулей, управляющих системными процессами;
- **.device** - позволяет реагировать на подключение устройств;
- **.path** - позволяет запускать модули при появлении пути в файловой системе;

- **.snapshot** - позволяет сохранять состояние менеджера systemd.

Каждый из таких модулей отвечает за отдельно взятый объект (службу, файловую систему, файл подкачки, сокет) или является вспомогательной частью системы systemd, позволяющей реализовать её преимущества (цель, устройство, таймер и другие).

Некоторые модули могут иметь в своем имени символ «@». Это означает, что этот файл является шаблоном - он не учитывается системой, а при создании нового модуля можно воспользоваться этим шаблоном, скопировав его, переименовав и изменив под свои задачи.

Модули в файловой системе представлены конфигурационными файлами. Основные места их размещения:

- Каталог `/usr/lib/systemd/system/` – модули, созданные при установке ОС;
- Каталог `/run/systemd/system/` — модули, созданные в рантайме. Этот каталог приоритетнее каталога с модулями, созданными при установке ОС;
- Каталог `/etc/systemd/system/` — модули, созданные и управляемые системным администратором. Этот каталог приоритетнее каталога модулей, созданных в рантайме.

Полный список возможных каталогов размещения модулей доступен в выводе команды `systemctl show --property=UnitPath`.

Ниже представлено содержимое каталогов `/usr/lib/systemd/system/` и `/etc/systemd/system/`

```
localadmin@astra:~$ ls -l /usr/lib/systemd/system
итого 984
-rw-r--r-- 1 root root  115 ноя 22  2018 acpid.path
-rw-r--r-- 1 root root  234 ноя 22  2018 acpid.service
-rw-r--r-- 1 root root  115 ноя 22  2018 acpid.socket
-rw-r--r-- 1 root root  157 апр 19  2020 acpi-support.service
-rw-r--r-- 1 root root  502 мар 30  2019 alsa-restore.service
-rw-r--r-- 1 root root  475 мар 30  2019 alsa-state.service
lrwxrwxrwx 1 root root    9 мар 30  2019 alsa-utils.service -> /dev/null
-rw-r--r-- 1 root root  776 мая 19  2019 anacron.service
. . .
lrwxrwxrwx 1 root root   15 апр  4 18:15 runlevel0.target -> poweroff.target
lrwxrwxrwx 1 root root   13 апр  4 18:15 runlevel1.target -> rescue.target
drwxr-xr-x 2 root root 4096 окт  3  2022 runlevel1.target.wants
lrwxrwxrwx 1 root root   17 апр  4 18:15 runlevel2.target -> multi-user.target
drwxr-xr-x 2 root root 4096 окт  3  2022 runlevel2.target.wants
lrwxrwxrwx 1 root root   17 апр  4 18:15 runlevel3.target -> multi-user.target
drwxr-xr-x 2 root root 4096 окт  3  2022 runlevel3.target.wants
lrwxrwxrwx 1 root root   17 апр  4 18:15 runlevel4.target -> multi-user.target
drwxr-xr-x 2 root root 4096 окт  3  2022 runlevel4.target.wants
lrwxrwxrwx 1 root root   16 апр  4 18:15 runlevel5.target -> graphical.target
drwxr-xr-x 2 root root 4096 окт  3  2022 runlevel5.target.wants
lrwxrwxrwx 1 root root   13 апр  4 18:15 runlevel6.target -> reboot.target
lrwxrwxrwx 1 root root    9 апр  4 18:15 sendsigs.service -> /dev/null
-rw-r--r-- 1 root root 1486 апр  4 18:15 serial-getty@.service
```

(continues on next page)

(продолжение с предыдущей страницы)

```
-rw-r--r-- 1 root root  442 фев 14  2019 shutdown.target
-rw-r--r-- 1 root root  402 фев 14  2019 sigpwr.target
lrwxrwxrwx 1 root root    9 апр  4 18:15 single.service -> /dev/null
. . .
-rw-r--r-- 1 root root  446 мая 16  2022 wpa_supplicant@.service
-rw-r--r-- 1 root root  478 мая 16  2022 wpa_supplicant-wired@.service
lrwxrwxrwx 1 root root    9 апр  4 18:15 x11-common.service -> /dev/null
-rw-r--r-- 1 root root  511 апр 10 09:35 zram-setup@.service

localadmin@astra:~$ ls -l /etc/systemd/system
итого 84
drwxr-xr-x 2 root root 4096 окт 18  2022 apache2.service.d
-rw-r--r-- 1 root root  207 мая 16 11:40 astra-mount-lock.service
-rw-r--r-- 1 root root  176 мая 16 12:06 astra-ptrace-lock.service
drwxr-xr-x 2 root root 4096 мая 16 12:06 astra-safepolicy.target.wants
drwxr-xr-x 2 root root 4096 авг 15 10:53 ceph-mgr@.service.d
drwxr-xr-x 2 root root 4096 авг 15 10:53 ceph-mon@.service.d
drwxr-xr-x 2 root root 4096 авг 15 10:53 ceph-osd@.service.d
lrwxrwxrwx 1 root root   42 мая 16 11:44 dbus-fi.w1.wpa_supplicant1.service -> /lib/
↪systemd/system/wpa_supplicant.service
lrwxrwxrwx 1 root root   40 мая 16 11:45 dbus-org.freedesktop.Avahi.service -> /lib/
↪systemd/system/avahi-daemon.service
lrwxrwxrwx 1 root root   53 мая 16 11:44 dbus-org.freedesktop.nm-dispatcher.service ->
↪ /lib/systemd/system/NetworkMan ager-dispatcher.service
lrwxrwxrwx 1 root root   34 авг 15 11:04 display-manager.service -> /lib/systemd/
↪system/fly- dm.service
drwxr-xr-x 2 root root 4096 окт 18  2022 dovecot.service.d
drwxr-xr-x 2 root root 4096 окт 18  2022 exim4.service.d
drwxr-xr-x 2 root root 4096 мая 16 11:29 getty.target.wants
drwxr-xr-x 2 root root 4096 мая 16 11:40 graphical.target.wants
drwxr-xr-x 2 root root 4096 мая 20 08:52 multi-user.target.wants
drwxr-xr-x 2 root root 4096 мая 16 11:44 network-online.target.wants
drwxr-xr-x 2 root root 4096 мая 16 11:44 printer.target.wants
drwxr-xr-x 2 root root 4096 мая 16 11:45 sockets.target.wants
-rw-r--r-- 1 root root 1107 окт  5  2022 suspend-to-hibernate.service
-rw-r--r-- 1 root root 1096 окт  5  2022 suspend-to-poweroff.service
drwxr-xr-x 2 root root 4096 мая 16 12:06 sysinit.target.wants
drwxr-xr-x 2 root root 4096 авг 14 14:04 timers.target.wants
-rw-r--r-- 1 root root  178 мая  5  2022 usbipd.service
drwxr-xr-x 2 root root 4096 авг 15 11:03 vboxadd.service.d
```

17.9.2 Цели systemd

Файлы целей systemd `*.target` предназначены для группировки модулей через цепочку зависимостей. Эта группировка позволяет оперировать не просто уровнями загрузки, как это было в init, а набором целевых состояний системы, которые можно создавать и изменять при необходимости. Это позволяет, с одной стороны, поделить классические уровни загрузки на отдельные цели, которые можно использовать отдельно друг от друга, а с другой - создавать свои собственные цели, при которых те или иные группы сервисов запускаются или останавливаются (учитывая все их зависимости).

Каждая цель может включать в себя зависимости от различных модулей (как правило это сервисы и цели). Эти зависимости устанавливаются через параметры модуля цели и через каталоги `<имя_цели>.target.wants`.

```
localadmin@astra:~$ ls -l /etc/systemd/system/multi-user.target.wants/
итого 0
lrwxrwxrwx 1 root root 35 мая 16 11:43 anacron.service -> /lib/systemd/system/anacron.
↪service
lrwxrwxrwx 1 root root 45 мая 16 11:40 astra-orientation.service -> /lib/systemd/
↪system/astra-orientation.service
. . .
lrwxrwxrwx 1 root root 42 мая 16 11:44 wpa_supplicant.service -> /lib/systemd/system/
↪wpa_supplicant.service
```

17.9.2.1 Целевые состояния системы и уровни загрузки

В классическом init было 6 уровней загрузки (*англ. runlevel*):

- **runlevel 0** – остановка системы;
- **runlevel 1** – загрузка в однопользовательском режиме, в системе выполняется минимальный набор программ. Также существует уровень S, который служит для ввода пароля root;
- **runlevel 2** – загрузка в многопользовательском режиме без поддержки сети;
- **runlevel 3** – загрузка в многопользовательском режиме с поддержкой сети;
- **runlevel 4** – не используется;
- **runlevel 5** – загрузка в многопользовательском режиме с поддержкой сети и графическим входом в систему;
- **runlevel 6** - перезагрузка.

Для перехода между этими уровнями использовались скрипты и команды, инициирующие переход - `runlevel <уровень_загрузки>` и `telinit <уровень_загрузки>`.

Systemd реализует эти уровни (для обратной совместимости) с помощью символических ссылок `runlevel<уровень_загрузки>.target` на цели со схожим назначением. Поэтому команды `runlevel` и `telinit`, используемые в init, по-прежнему доступны.

Целевые состояния системы, использующиеся в systemd, и их соответствие уровням загрузки:

- Файл `poweroff.target` – команда `runlevel 0`;
- Файл `rescue.target` – команда `runlevel 1`, `runlevel S`;
- Файл `multi-user.target` – команда `runlevel 3`;

- Файл `graphical.target` – команда `runlevel 5`;
- Файл `reboot.target` – команда `runlevel 6`.

```
localadmin@astra:~$ ls -l /usr/lib/systemd/system | grep runlevel
rwxrwxrwx 1 root root   15 сен 12  2023 runlevel0.target -> poweroff.target
lrwxrwxrwx 1 root root   13 сен 12  2023 runlevel1.target -> rescue.target
drwxr-xr-x 2 root root 4096 апр  4  2023 runlevel1.target.wants
lrwxrwxrwx 1 root root   17 сен 12  2023 runlevel2.target -> multi-user.target
drwxr-xr-x 2 root root 4096 апр  4  2023 runlevel2.target.wants
lrwxrwxrwx 1 root root   17 сен 12  2023 runlevel3.target -> multi-user.target
drwxr-xr-x 2 root root 4096 апр  4  2023 runlevel3.target.wants
lrwxrwxrwx 1 root root   17 сен 12  2023 runlevel4.target -> multi-user.target
drwxr-xr-x 2 root root 4096 апр  4  2023 runlevel4.target.wants
lrwxrwxrwx 1 root root   16 сен 12  2023 runlevel5.target -> graphical.target
drwxr-xr-x 2 root root 4096 апр  4  2023 runlevel5.target.wants
lrwxrwxrwx 1 root root   13 сен 12  2023 runlevel6.target -> reboot.target
-rw-r--r-- 1 root root  797 сен 12  2023 systemd-update-utmp-runlevel.service
```

Для управления целевым состоянием системы в systemd применяются команды:

- Команда `sudo systemctl isolate <имя>.target` – переключит систему в выбранное целевое состояние, например:
  - Команда `sudo systemctl isolate reboot.target` – перезагрузит систему,
  - Команда `sudo systemctl isolate multi-user.target` – переведет в режим работы без графического интерфейса.
- Команда `systemctl get-default` – выведет имя цели по умолчанию.
- Команда `systemctl set-default --force <имя>.target` – изменит цель по умолчанию на выбранную.

17.9.2.2 Устройство конфигурационного файла модуля службы

Синтаксис файла модуля напоминает формат конфигурационных файлов ini в Windows. Рассмотрим содержимое одного из модулей служб. Возьмем конфигурационный файл модуля `systemd-logind.service` и посмотрим его содержимое.

```
#  SPDX-License-Identifier: LGPL-2.1+
#
#  This file is part of systemd.
#
#  systemd is free software; you can redistribute it and/or modify it
#  under the terms of the GNU Lesser General Public License as published by
#  the Free Software Foundation; either version 2.1 of the License, or
#  (at your option) any later version.

[Unit]
```

(continues on next page)

(продолжение с предыдущей страницы)

```
Description=Login Service
Documentation=man:systemd-logind.service(8) man:logind.conf(5)
Documentation=https://www.freedesktop.org/wiki/Software/systemd/logind
Documentation=https://www.freedesktop.org/wiki/Software/systemd/multiseat
Wants=user.slice
After=nss-user-lookup.target user.slice
ConditionPathExists=/lib/systemd/system/dbus.service

# Ask for the dbus socket.
Wants=dbus.socket
After=dbus.socket

[Service]
BusName=org.freedesktop.login1
CapabilityBoundingSet=CAP_SYS_ADMIN CAP_MAC_ADMIN CAP_AUDIT_CONTROL CAP_CHOWN CAP_KILL
CAP_DAC_READ_SEARCH CAP_DAC_OV
ERRIDE CAP_FOWNER CAP_SYS_TTY_CONFIG
ExecStart=/lib/systemd/systemd-logind
FileDescriptorStoreMax=512
IPAddressDeny=any
LockPersonality=yes
MemoryDenyWriteExecute=yes
NoNewPrivileges=yes
Restart=always
RestartSec=0
RestrictAddressFamilies=AF_UNIX AF_NETLINK
RestrictNamespaces=yes
RestrictRealtime=yes
RestrictSUIDSGID=yes
SystemCallArchitectures=native
SystemCallErrorNumber=EPERM
SystemCallFilter=@system-service
WatchdogSec=3min

# Increase the default a bit in order to allow many simultaneous logins since
# we keep one fd open per session.
LimitNOFILE=524288
```

В файле в квадратных скобках обозначаются разделы. Для модуля службы существует 3 стандартных раздела:

- **[Unit]** – обычно его используют для определения метаданных и настройки отношения к другим модулям;
- **[Service]** – используется для предоставления конфигурации, которая применима только для служб;
- **[Install]** – Этот раздел является дополнительным и используется для определения условий запуска модуля.

Для модулей других типов вместо секции **[Service]** используются другие: **[Socket]**, **[Mount]**, **[Automount]**, **[Swap]**, **[Timer]**, **[Slice]**.

В каждом из этих разделов существует набор директив, с помощью которых определяется назначение и поведение модуля. Ниже представлены основные директивы.

Секция [Unit]

В разделе [Unit] можно использовать следующие директивы:

- **Description** – эта директива может использоваться для описания имени и основных функций модуля. Это значение возвращается различными инструментами systemd, поэтому полезно установить что-то короткое, конкретное и информативное.
- **Documentation** – эта директива предоставляет местоположения документации. Это могут быть внутренние доступные справочные страницы либо доступные в интернете (URL-адреса). Команда `systemctl status <имя_сервиса>` выведет эту информацию.
- **Requires** – в данной директиве перечислены все модули, от которых зависит этот модуль (служба или устройство). Если текущий модуль должен быть запущен, перечисленные здесь модули так же должны успешно запуститься. В противном случае это вызывает ошибку запуска текущего модуля. По умолчанию эти модули запускаются параллельно с текущим модулем.
- **Wants** – эта директива похожа на Requires=, но менее строгая. Systemd будет пытаться запустить любые модули, перечисленные здесь, когда запускается текущий модуль. Если эти модули не обнаружены или не запущены, текущий модуль просто продолжит запуск. Это рекомендуемый способ настройки большинства зависимостей. По умолчанию эти модули запускаются параллельно с текущим модулем.
- **BindsTo** – эта директива аналогична Requires=, но также приводит к остановке текущего модуля, когда соответствующий модуль останавливается.
- **Before** - модули, перечисленные в этой директиве, не будут запущены до тех пор, пока текущий модуль не будет отмечен как запущенный, если они будут запускаться одновременно.
- **After** - Модули, перечисленные в этой директиве, будут запущены до запуска текущего модуля.
- **Conflicts** - используется для задания модулей, которые нельзя запускать одновременно с текущим модулем. Запуск модуля с этой связью приведет к остановке других модулей.
- **Condition<условие>** - существует ряд директив, начинающихся с условия, которые позволяют администратору проверить некоторые состояния системы и модулей до запуска устройства, например: *ConditionFileNotEmpty* или *ConditionPathExists*. Если условие не выполнено, запуск модуля будет отменен без ошибки.

- **Assert<условие>** – подобен набору директив Condition..., но установленные директивы проверяют различные аспекты рабочей среды, чтобы решить, следует ли запустить модуль, например: *AssertPathExists*. Однако, в отличие от директив Condition..., отрицательный результат вызывает ошибку запуска текущего модуля.

Секция [Install]

В разделе [Install] можно использовать следующие директивы:

- **WantedBy** – Данная директива является наиболее распространенным способом определения того, как модуль должен быть запущен. Эта директива позволяет указать зависимость (аналогично директиве Wants в разделе [Unit]). Разница в том, что эта директива включена во вспомогательную секцию, позволяющую секции [Unit] оставаться относительно чистой. Systemd автоматически создает каталоги вида *имя_модуля.тип_модуля*.wants в каталоге с файлом модуля и добавляет эти зависимости в виде символических ссылок в каталоге ...wants.

- **RequiredBy** – эта директива очень похожа на директиву WantedBy, но это строгая зависимость. И в случае невозможности запуска соответствующего модуля запуск текущего модуля будет завершен с ошибкой.

- **Alias** – эта директива позволяет запустить модуль, используя другое имя.

- **Also** – эта директива позволяет включать или отключать сразу набор модулей. Здесь можно указать сопутствующие модули, которые всегда должны быть доступны, когда этот модуль запущен.

Секция [Service]

В разделе [Service] можно использовать следующие директивы:

- **Type** – тип службы:

  - **simple** – тип службы по умолчанию: systemd запустит эту службу незамедлительно. Процесс при этом не должен разветвляться (fork). Если после данной службы должны запускаться другие, то этот тип использовать не стоит (исключение — если служба использует сокетную активацию).

  - **forking** – systemd считает службу запущенной после того, как процесс разветвляется с завершением родительского процесса. Используется для запуска классических демонов за исключением тех случаев, когда в таком поведении процесса нет необходимости. Хорошо подходит, например, для запуска nginx, tomcat.

- **oneshot** - удобен для сценариев, которые выполняют одно задание и завершаются. Если задать параметр `RemainAfterExit=yes`, то systemd будет считать процесс активным даже после его завершения (например, при монтировании).

- **dbus** – служба считается находящейся в состоянии готовности после появления указанного BusName в системной шине DBus.

- **notify** – идентичен параметру `Type=simple`, но с уточнением, что демон пошлет systemd сигнал готовности. Реализация уведомления находится в библиотеке `libsystemd-daemon.so`.

- **idle** – systemd отложит выполнение двоичного файла службы до окончания запуска остальных («более срочных») задач. В остальном поведение аналогично Type=simple.

- **User** – указание пользователя, от имени которого будет запущена служба.

- **ExecStart** – директива, где нужно указать полный путь и аргументы команды, которая должна быть выполнена для запуска процесса. Эту директиву можно указать только один раз.

- **ExecStartPre** – директива, где нужно указать полный путь и аргументы команды, которые должны быть выполнены до запуска основного процесса. Эту директиву можно использовать несколько раз.

- **ExecStartPost** – аналогична ExecStartPre за исключением того, что данная директива указывает команды, которые будут запускаться после запуска основного процесса.

- **ExecReload** – эта необязательная директива содержит команду, необходимую для перезагрузки конфигурации службы (если она доступна).

- **ExecStop** – содержит команду, необходимую для остановки службы. Если она не указана, процесс будет просто уничтожен во время остановки службы.

- **ExecStopPost** – содержит команду, выполняемую после остановки службы.

- **Slice** – содержит имя контрольной группы.

- **KillMode** – оправляет способ завершения главного и дочерних процессов. Может принимать значения: **control-group** - (значение по умолчанию) завершит работу всех процессов контрольной группы, **mixed** - передаст сигнал SIGTERM главному и SIGKILL остальным процессам контрольной группы, **process** - завершит только главный процесс (не рекомендуется использовать).

- **RestartSec** – если автоматический перезапуск службы включен, задает время ожидания перед попыткой перезапуска службы.

- **Restart** – директива определяет условия, при которых systemd будет пытаться автоматически перезапустить службу. Может принимать значения: «always», «on-success», «on-failure», «on-abnormal», «on-abort» или «on-watchdog».

- **TimeoutSec** – устанавливает время, в течение которого система будет ждать старта или остановки службы, прежде чем пометить ее запуск как неудавшийся или остановку как принудительно убитую. Можете установить отдельные таймауты с помощью директив TimeoutStartSec и TimeoutStopSec.

17.9.2.3 Создание модуля службы

Создадим модуль службы для программы `/usr/bin/fly-message-monitor`, которая позволяет отправлять сообщения на рабочий стол пользователя. Для создания модуля службы запустим текстовый редактор с правами суперпользователя, чтобы открыть (создать) файл `/etc/systemd/system/fly-message-monitor.service`:

```
localadmin@astra:~$ sudo vim /etc/systemd/system/fly-message-monitor.service
```

и наполнить его следующим содержимым (**если ваш пользователь назван не localadmin, а как-то иначе, то необходимо указать имя учетной записи, используемой вами**):

```
[Unit]
Description=Сервис отправки уведомлений на рабочий стол
After=graphical.target

[Service]
ExecStart=/usr/local/bin/fly-message-monitor-localadmin
Type=simple
Restart=always
RestartSec=0
Environment="DISPLAY=:0" "XAUTHORITY=/home/localadmin/.Xauthority"

[Install]
WantedBy=graphical.target
```

Тут мы видим три характерные для файла службы секции Unit, Service и Install.

- В секции **[Unit]** мы используем две директивы:
  - **Description** для задания описания службы.
  - **After** для указания, что этот сервис должен запускаться только после загрузки сервисов, входящих в целевое состояние системы graphical.target, когда у нас будут доступны службы рабочего стола.
- В секции **[Service]**:

- через директиву **ExecStart** мы задаем путь до исполнимого файла.
- тип службы (Type) выберем Simple.
- директивы перезапуска службы **Restart** и **RestartSec** выберем always и 0 для немедленного запуска службы в случае её падения.
- директива **Environment** позволяет задавать или переопределять переменные для процесса сервиса. Переменные DISPLAY и XAUTHORITY необходимы, чтобы сообщения, отправляемые сервисом, доходили до рабочего стола сессии пользователя localadmin.

- В секции **[Install]** зададим только одну директиву.
  - **WantedBy** – говорим сервису, что для его работы необходимо целевое состояние системы graphical.target.

Создадим файл скрипта `/usr/local/bin/fly-message-monitor-localadmin` путем копирования файла `/usr/bin/fly-message-monitor` и последующего его изменения.

Скопируем файл `/usr/bin/fly-message-monitor`, отредактируем `/usr/local/bin/fly-message-monitor-localadmin`

```
localadmin@astra:~$ sudo cp /usr/bin/fly-message-monitor /usr/local/bin/fly-message-
↪monitor-
localadmin
```

и заменим цикл while в конце файла:

```
#wait changes in log file
while true ; do
    if [ -s $LOG_FILE ]; then
      MSG="empty"
      MSG=$(tail -10 $LOG_FILE)
      echo "$MSG" > ~/txt.txt
#     notify-send -u critical -t 10000 -i dialog-warning "$MESSAGE" "$MSG"
#     fly-dialog --caption "$MESSAGE" --msgbox "$MSG"
      fly-dialog --caption "$MESSAGE" --textbox ~/txt.txt
      rm ~/txt.txt
   fi

   #wait for changes
   inotifywait -qq -r -e modify $LOG_FILE &
   wait $!

   #if dir or file are deleted then wait for it again
   if [ ! -f $LOG_FILE ]; then
     wait_for_file
     #restart
     pkill inotifywait
   fi
done
```

на:

```
#wait changes in log file
while true ; do
   inotifywait -qq -r -e modify $LOG_FILE &
   wait $!
   MSG=""; TRAY_MSG=""
   URGENCY=$(head -1 $LOG_FILE)
   TYPE=$(sed -n '2p' $LOG_FILE)
   MSG=$(tail -1 $LOG_FILE)
   notify-send -u $URGENCY -t 4950 -i $TYPE "$MESSAGE" "$MSG"
done
```

тем самым позволив управлять не только выводимым сообщением, но и его типом, а также изменив продолжительность отображения сообщения на экране.

Сохраним файл службы, а затем перечитаем конфигурацию сервисов в системе командой

```
localadmin@astra:~$ sudo systemctl daemon-reload
```

Запустим нашу службу fly-message-monitor.service командой

```
localadmin@astra:~$ sudo systemctl start fly-message-monitor.service
```

Разрешим автозапуск службы при переходе между целевыми состояниями командой

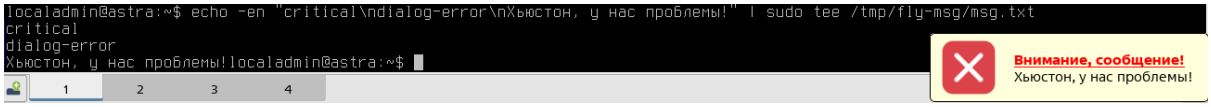
```
localadmin@astra:~$ sudo systemctl enable fly-message-monitor.service
Created symlink /etc/systemd/system/graphical.target.wants/fly-message-monitor.
↪service →
/etc/systemd/system/fly-message-monitor.s
```

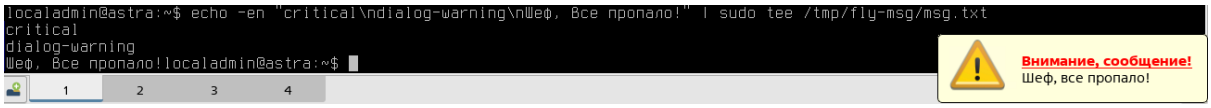
Посмотрим состояние службы командой

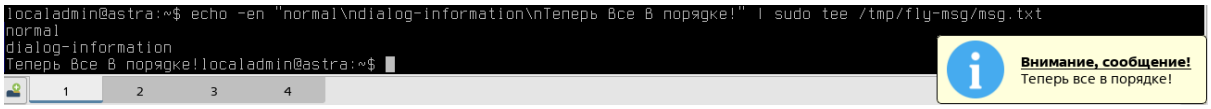
```
localadmin@astra:~$ systemctl status fly-message-monitor.service
o fly-message-monitor.service - Сервис отправки уведомлений на рабочий стол
  Loaded: loaded (/etc/systemd/system/fly-message-monitor.service; enabled; vendor␣
↪preset:
enabled)
  Active: active (running) since Mon 2024-03-04 18:29:05 MSK; 26min ago
Main PID: 21422 (fly-message-mon)
   Tasks: 2 (limit: 2250)
  Memory: 852.0K
     CPU: 65ms
  CGroup: /system.slice/fly-message-monitor.service
          ├─21422 /bin/bash /usr/bin/fly-message-monitor
          └─21560 inotifywait -qq -r -e modify /tmp/fly-msg/msg.txt
```

Проверим, работает ли служба и скрипт, как ожидалось. Отправим несколько сообщений:

```
localadmin@astra:~$ echo -en "critical \ndialog-error \nХьюстон, у нас проблемы!" |␣
↪sudo tee /tmp/fly-msg/msg.txt
localadmin@astra:~$ echo -en "critical \ndialog-warning\nШеф, все пропало!" | sudo␣
↪tee /tmp/fly-msg/msg.txt
localadmin@astra:~$ echo -en "normal\ndialog-information\nТеперь все в порядке!" |␣
↪sudo tee /tmp/fly-msg/msg.txt
```







Создадим еще одну службу `/etc/systemd/system/resource-demanding1.service`, которая будет запускать скрипт, воспроизводящий высокую нагрузку на CPU. Но перед этим подготовим скрипт, который будет запускать эта служба.

От имени суперпользователя создадим в каталоге `/usr/local/bin/` (который предназначен в том числе для хранения скриптов, запускаемых на уровне всей системы) скрипт `resource-demanding1.sh` и предоставим владельцу право на его выполнение.

```
#!/bin/bash
/usr/bin/sha256sum /dev/urandom > /dev/null
```

```
localadmin@astra:~$ sudo -i
localadmin@astra:~$ echo "/usr/bin/sha256sum /dev/urandom > /dev/null" > /usr/local/
↪bin/resource-demanding1.sh
localadmin@astra:~$ chmod u+x /usr/local/bin/resource-demanding1.sh
localadmin@astra:~$ exit
```

```
localadmin@astra:~$ sudo -i
[sudo] пароль для localadmin:
root@astra:~# echo "/usr/bin/sha256sum /dev/urandom > /dev/null" > /usr/local/bin/
↪resource-
demanding1.sh
root@astra:~# chmod u+x /usr/local/bin/resource-demanding1.sh
root@astra:~# exit
выход
localadmin@astra:~$ cat /usr/local/bin/resource-demanding1.sh
/usr/bin/sha256sum /dev/urandom > /dev/null
localadmin@astra:~$ ls -l /usr/local/bin/resource-demanding1.sh
-rwxr--r-- 1 root staff 88 мар  4 19:14 /usr/local/bin/resource-demanding1.sh
```

Создадим файл службы `/etc/systemd/system/resource-demanding1.service` со следующей конфигурацией:

```
[Unit]
Description=Сервис, воспроизводящий высокую нагрузку на CPU 0
After=graphical.target

[Service]
ExecStart=taskset 0x00000001 /usr/bin/bash /usr/local/bin/resource-demanding1.sh
Type=simple
```

(continues on next page)

(продолжение с предыдущей страницы)

```
[Install]
WantedBy=graphical.target
```

Этот сервис будет запускать скрипт `/usr/local/bin/resource-demanding1.sh` только на первом процессоре.

```
localadmin@astra:~$ sudo vim /etc/systemd/system/resource-demanding1.service
localadmin@astra:~$ sudo systemctl daemon-reload
localadmin@astra:~$ sudo systemctl start resource-demanding1.service
localadmin@astra:~$ systemctl status resource-demanding1.service
o resource-demanding1.service - Сервис воспроизводящий высокую нагрузку на CPU
  Loaded: loaded (/etc/systemd/system/resource-demanding1.service; disabled; vendor␣
↪preset:
enabled)
  Active: active (running) since Mon 2024-03-04 19:19:45 MSK; 3s ago
Main PID: 22979 (bash)
   Tasks: 1 (limit: 2250)
  Memory: 440.0K
     CPU: 3.938s
  CGroup: /system.slice/resource-demanding1.service
          ├─26319 /usr/bin/bash /usr/local/bin/resource-demanding1.sh
          └─26320 /usr/bin/sha256sum /dev/urandom
```

```
  PID USER      PRI  NI  VIRT   RES   SHR S CPU% MEM%   TIME+  Command
26320 root       20   0  5284   684   624 R 101.  0.0  1:52.26 /usr/bin/sha256sum /dev/urandom
```

Оставим службу `resource-demanding1.service` «disabled», то есть ей не разрешен автоматический запуск.

```
localadmin@astra:~$ sudo systemctl stop resource-demanding1.service
localadmin@astra:~$ sudo systemctl disabled resource-demanding1.service
```

17.9.2.4 Механизм пространства имен namespaces и контрольных групп в systemd

Пространство имен в Linux

Пространство имён (*англ. namespaces*) - это механизм ядра Linux, позволяющий изолировать и виртуализировать глобальные системные ресурсы множества процессов. Примеры ресурсов, которые можно виртуализировать: ID процессов, имена хостов, ID пользователей, доступ к сетям, межпроцессное взаимодействие и файловые системы. Одной из общих целей пространств имён является поддержка реализации контейнеров - инструмента для виртуализации на уровне операционной системы (а также других целей), обеспечивающего группу процессов иллюзией того, что они являются единственными процессами в системе. Поэтому одной из главных целей пространства имён является поддержка контейнеризации в Linux.

Linux-система при старте инициализирует один экземпляр каждого типа пространства имён, кроме пространства имён файловой системы. После инициализации можно создать или объединить

дополнительные пространства имён.

Все пространства имён поддерживают вложенность, то есть между ними можно установить связь «родитель — потомок». Таким образом, некоторые пространства наследуют все свойства от своего родительского пространства имён. Однако это верно не для всех пространств.

Функциональные возможности пространства имён одинаковы для всех типов: каждый процесс связан с пространством имён и может видеть или использовать только ресурсы, связанные с этим пространством имён, и, где это применимо, с его потомками. Таким образом, каждый процесс (или его группа) может иметь уникальное представление о ресурсе. Изолированный ресурс зависит от типа пространства имён, созданного для данной группы процессов.

В Linux существует 7 типов пространств имен:

- **PID** (*англ. PID namespace*) – изоляция ID процессов - позволяет процессам из разных пространств имн одинаковые PID;
- **сети** (*англ. net namespace*) – изоляция системных ресурсов, связанных с сетями: сетевое оборудование, стеки протоколов IPv4 и IPv6, таблицы IP-маршрутизации, фильтрации, номера портов (сокеты) и другие;
- **UTS** (*англ. UTS namespace, Unix Time Sharing*) – позволяет каждому контейнеру иметь свое имя хота и имя домена;
- **пользователя** (*англ. user namespace*) – изолируют ID пользователей и групп, корневой каталог, ключи;
- **файловой системы** (*англ. mnt namespace*) – обеспечивает независимость дерева файловой системы, ассоциированное с определённой группой процессов. Каждое пространство имён предоставляет уникальный ви файловой системы для всех процессов, принадлежащих этому пространству;
- **межпроцессорное взаимодействие** (*англ. IPC namespace*) – обеспечивает изоляцию процессов при обращении к разделяемым ресурсам, например, памяти. Процессам каждого пространства имен IPC видны только объекты, принадлежащие к этому пространству;
- **контрольные группы** (англ. cgroup namespace) – обеспечивают изоляцию и возможность накладывать ограничения на некоторые ресурсы компьютера: процессорные, сетевые, память, ввода-вывода. Механизм позволяет образовывать иерархические группы процессов с заданными ресурсными свойствами и обеспечивает программное управление ими.

Контрольные группы в Linux

Изначально этот механизм ядра Linux носил название контейнер процессов, но из-за путаницы с другими решениями по контейнеризации был переименован в механизм контрольных групп (*англ. cgroups*). Его предназначение - разделять и ограничивать некоторые ресурсы компьютера между процессами. Сейчас используется вторая версия этого механизма.

Сами контрольные группы находятся в псевдофайловой системе sysfs `/sys/fs/cgroup`. Одни контрольные группы можно включать в другие контрольные группы и тем самым агрегировать их. Контрольная группа представляет собой каталог с набором файлов, где имя файла — это имя параметра, а текст внутри файла — его значение.

Контрольные группы могут управлять доступом к следующим ресурсам:

- **cpu** – ограничение выделяемой доли процессорного времени по заданному значению, если система загружена. Не ограничивает процессорные ресурсы, если система свободна;
- **cpuset** – эту cgroup можно использовать для привязки процессов в контрольной группе к указанному наору ЦП и узлов NUMA;
- **freezer** – может приостанавливать и возобновлять работу всех процессов в контрольной группе. Заморозка контрольной группы так же влияет на её потомков;
- **io** - контролирует и ограничивает доступ к заданным блочным устройствам, применяет управление вводом-выводом посредством пропусков (throttling) и ограничения сверху листовых узлов и промежуточных узлов в иерархии хранилища;
- **memory** - контроллер памяти поддерживает учёт и ограничение памяти процесса, памяти ядра и подкачки, используемой контрольной группой;
- **pids** - контроллер позволяет ограничивать количество процессов, которые могут быть созданы в контрольной группе (и её потомках);
- **rdma** (remote direct memory access) - Контроллер RDMA позволяет ограничивать использование ресурсов RDMA/IB определённой контрольной группе.

Для получения информации о том, в каких контрольных группах находятся процессы, можно воспользоваться командой `ps xawf -eo pid,user,cgroup,args`.

```
localadmin@astra:~$ sudo ps xawf -eo pid,user,cgroup,args
 PID USER     CGROUP                     COMMAND
   2 root     -                          [kthreadd]
   3 root     -                           _ [rcu_gp]
   . . .
   1 root     0::/init.scope             /sbin/init
 265 root     0::/system.slice/systemd-jo /lib/systemd/systemd-journald
```

(continues on next page)

(продолжение с предыдущей страницы)

```
 298 root     0::/system.slice/systemd-ud /lib/systemd/systemd-udevd
 338 root     0::/system.slice/auditd.ser /sbin/auditd
 478 root     0::/system.slice/systemd-lo /lib/systemd/systemd-logind
 479 message+ 0::/system.slice/dbus.servi /usr/bin/dbus-daemon --system --
↪address=systemd: --no fork --nopidfile --systemd-activatio
 481 root     0::/system.slice/wpa_suppli /sbin/wpa_supplicant -u -s -O /run/wpa_
↪supplicant
 482 root     0::/system.slice/syslog-ng. /usr/sbin/syslog-ng -F --no-caps
 483 root     0::/system.slice/alsa-state /usr/sbin/alsactl -E HOME=/run/alsa -s -n␣
↪19 -c rdaemon
 489 root     0::/system.slice/udisks2.se /usr/lib/udisks2/udisksd
 495 avahi    0::/system.slice/avahi-daem avahi-daemon: running [astra.local]
 501 avahi    0::/system.slice/avahi-daem  _ avahi-daemon: chroot helper
 497 root     0::/system.slice/cron.servi /usr/sbin/cron -f
 503 root     0::/system.slice/NetworkMan /usr/sbin/NetworkManager --no-daemon
 563 root     0::/system.slice/polkit.ser /usr/lib/policykit-1/polkitd --no-debug
 611 root     0::/system.slice/libflygete /usr/bin/fly-getexe
 628 root     0::/system.slice/system-get /sbin/agetty -o -p -- \u --noclear tty1␣
↪linux
1114 root     0::/system.slice/fly-dm.ser /usr/bin/fly-dm vt7
. . .
```

А с помощью команды `systemd-cgls` можно увидеть контрольные группы и входящие в них процессы в древовидной структуре.

```
localadmin@astra:~$ systemd-cgls
Control group /:
-.slice
├─user.slice
│ ├─user-107.slice
│ │ ├─user@107.service
│ │ │ └─init.scope
│ │ │   ├─1156 /lib/systemd/systemd --user
│ │ │   └─1157 (sd-pam)
│ │ └─session-c1.scope
│ │   └─1122 /usr/lib/xorg/Xorg -br -novtswitch -quiet -keeptty :0 vt7 -logfile /var/
↪log/fly-dm/
│ └─user-1000.slice
│   ├─user@1000.service
│   │ └─init.scope
│   │   ├─1258 /lib/systemd/systemd --user
│   │   └─1259 (sd-pam)
│   └─session-3.scope
│     ├─ 1177 -:0
│     ├─ 1281 fly-wm
│     ├─ 1346 /usr/bin/dbus-launch --exit-with-session --sh-syntax
│     ├─ 1347 /usr/bin/dbus-daemon --syslog --fork --print-pid 5 --print-address 7 --
↪session
│     ├─ 1370 /usr/bin/kglobalaccel5
│     ├─ 1376 /usr/bin/VBoxClient --clipboard
│     ├─ 1377 /usr/bin/VBoxClient --clipboard
│     ├─ 1391 /usr/bin/VBoxClient -seamless
. . .
```

Посмотрим содержимое каталога `/sys/fs/cgroup/system.slice`.

```
localadmin@astra:~$ ls /sys/fs/cgroup/system.slice/
alsa-restore.service
alsa-state.service
astra-orientation.service
auditd.service
avahi-daemon.service
avahi-daemon.socket
cgroup.controllers
cgroup.events
cgroup.freeze
cgroup.kill
cgroup.max.depth
cgroup.max.descendants
. . .
```

Посмотрим содержимое каталога, относящегося к сервису аудита.

```
localadmin@astra:~$ ls /sys/fs/cgroup/system.slice/auditd.service/
cgroup.controllers      cgroup.subtree_control  memory.events.local  memory.stat
cgroup.events           cgroup.threads          memory.high          memory.swap.
↪current
cgroup.freeze           cgroup.type             memory.low           memory.swap.
↪events
cgroup.kill             cpu.pressure            memory.max           memory.swap.high
cgroup.max.depth        cpu.stat                memory.min           memory.swap.max
cgroup.max.descendants  io.pressure             memory.numa_stat     pids.current
cgroup.procs            memory.current          memory.oom.group     pids.events
cgroup.stat             memory.events           memory.pressure      pids.max
```

В этих файлах хранится статистика (например, cpu.stat) и ограничения использования ресурсов.

Управление контрольными группами сервисов

Контрольную группу для сервиса можно назначить прямо через его конфигурационный файл с помощью директивы `Slice: Slice={Имя_контрольной_группы}`. По умолчанию имя контрольной группы сервиса совпадает с именем сервиса.

Контрольную группу и параметры контроля можно задать с помощью встраиваемых файлов. Для этого:

- В каталоге `/etc/systemd/system/` необходимо создать одноименный сервису каталог с суффиксом «.d», например, для сервиса apache2 это будет `/etc/systemd/system/apache2.service.d/`.
- Внутри этого каталога создать файлы с расширением .conf (тут рассмотрен пример ограничения по процессорному времени):

Файл `00-slice.conf`

```
[Service]
Slice=AWESOME.slice
MemoryAccounting=yes
CPUAccounting=yes
```

Где:

- **Slice** задаёт – имя контрольной группы,
- **MemoryAccounting** и **CPUAccounting** – включает подсчет потребления ресурсов памяти и процессора соответственно.

Файл `10-CPUSettings.conf`

```
[Service]
CPUWeight=1000
```

Где:

- **CPUWeight=1000** – устанавливает вес процессов этой контрольной группы. Чем вес больше, тем больше приоритет (по умолчанию вес 100, а параметр может принимать значения от 1 до 10000).

Для применения этих настроек необходимо перечитать конфигурационные файлы службы командой `sudo systemctl daemon-reload` и перезапустить службу `sudo systemctl restart apache2`.

Установим ограничение на контрольную группу созданного нами сервиса resource-demanding1.service по использованию процессорного времени. Для этого создадим каталог `/etc/systemd/system/resource-demanding1.service.d` и создадим в нем файлы:

Файл `00-slice.conf` со следующим содержимым:

```
[Service]
MemoryAccounting=yes
CPUAccounting=yes
```

Файл `10-CPUSettings.conf` со следующим содержимым:

```
[Service]
CPUQuota=90%
```

Где:

- **CPUQuota** – устанавливает лимит потребления процессом ресурсов процессора.

```
localadmin@astra:~$ sudo mkdir /etc/systemd/system/resource-demanding1.service.d
localadmin@astra:~$ sudo vim /etc/systemd/system/resource-demanding1.service.d/00-
↪slice.conf
localadmin@astra:~$ sudo vim /etc/systemd/system/resource-demanding1.service.d/10-␣
```

(continues on next page)

(продолжение с предыдущей страницы)

```
↪CPUSettings.conf
localadmin@astra:~$ cat /etc/systemd/system/resource-demanding1.service.d/00- slice.
↪conf
[Service]
MemoryAccounting=yes
CPUAccounting=yes
localadmin@astra:~$ cat /etc/systemd/system/resource-demanding1.service.d/10-
↪CPUSettings.conf
[Service]
CPUQuota=90%
```

Перечитаем файлы конфигурации и перезапустим сервис resource-demanding1.service.

```
localadmin@astra:~$ sudo systemctl daemon-reload
localadmin@astra:~$ sudo systemctl restart resource-demanding1.service
localadmin@astra:~$ systemctl status resource-demanding1.service
o resource-demanding1.service - Сервис воспроизводящий высокую нагрузку на CPU 0
  Loaded: loaded (/etc/systemd/system/resource-demanding1.service; disabled; vendor␣
↪preset: enabled)
Drop-In: /etc/systemd/system/resource-demanding1.service.d
          └─00-slice.conf, 10-CPUSettings.conf
  Active: active (running) since Mon 2024-03-04 20:50:34 MSK; 2s ago
Main PID: 26483 (bash)
   Tasks: 2 (limit: 2250)
  Memory: 628.0K
     CPU: 2.263s
  CGroup: /system.slice/resource-demanding1.service
          ├─26483 /usr/bin/bash /usr/local/bin/resource-demanding1.sh
          └─26484 /usr/bin/sha256sum /dev/urandom
```

Проверим содержимое файла sys/fs/cgroup/system.slice/resource-demanding1.service/cpu.max.

```
localadmin@astra:~$ cat /sys/fs/cgroup/system.slice/resource-demanding1.service/cpu.
↪max
90000 100000
```

Как видим, введенное нами ограничение вступило в силу и теперь процессы в группе не могут потреблять более чем 90% процессорного времени, что подтверждает htop:



Больше информации о контрольных группах для служб можно найти в справке man 5 systemd.slice и man 5 systemd.resource-control.

Примечание: Подробнее о groups можно почитать по ссылке: https://manpages.debian.org/unstable/manpages-ru/cgroups.7.ru.html

---

17.9.2.5 Управление модулями и другими компонентами systemd

Для управления модулями systemd существует ряд утилит:

- Утилита `systemctl` - управления службами, модулями и целевыми состояниями системы;
- Утилита `journalctl` – утилита просмотра журналов, сформированных сервисом journald. Позволяет гибко настраивать вывод через параметры;
- Утилита `systemd-notify` - используется модулями .notify для оповещения службы systemd об их статусе;
- Утилита `systemd-analyze` – позволяет производить анализ времени запуска служб;
- Утилита `systemd-cgls` – выводит дерево контрольных групп (cgroups);
- Утилита `systemd-cgtop` - аналог утилиты top для контрольных групп;
- Утилита `loginctl` – инструмент управления менеджером systemd-logind.

Рассмотрим применение каждой из этих утилит подробнее.

Systemctl

Утилита предназначена для управления службами и целевыми состояниями системы.

Команды для управления целевыми состояниями системы:

- Команда `sudo systemctl isolate <имя>.target` – переключит систему в выбранное целевое состояние, например, `sudo systemctl isolate reboot.target` перезагрузит систему, а `sudo systemctl isolate multi-user.target` переведет в режим работы без графического интерфейса.
- Команда `systemctl get-default` – выведет имя цели по умолчанию.
- Команда `systemctl set-default --force <имя>.target` – изменит цель по умолчанию на выбранную.

Команды для просмотра списка модулей:

- Команда `systemctl` – выводит список всех запущенных модулей;
- Команда `systemctl -a` – выводит список всех модулей;
- Команда `systemctl --type=<тип_модуля>` – выводит список всех модулей соответствующего типа.

Команды для просмотра, редактирования и создания конфигурационного файла модулей:

- Команда `systemctl cat <имя_модуля>` – выводит содержимое конфигурационного файла модуля;
- Команда `systemctl --full edit <имя_модуля>` – редактирование конфигурационного файла модуля;
- Команда `systemctl --force --full edit <имя_модуля>` – создание нового модуля.

Команды для управления сервисами:

- Команда `systemctl status <имя_сервиса>.service` – выводит информацию о статусе службы;
- Команда `systemctl enable <имя_сервиса>.service` – добавляет службу в автозагрузку;
- Команда `systemctl disable <имя_сервиса>.service` – удаляет службу из автозагрузки;
- Команда `systemctl start <имя_сервиса>.service` – запускает службу;
- Команда `systemctl stop <имя_сервиса>.service` – останавливает службу;
- Команда `systemctl restart <имя_сервиса>.service` – перезапускает службу;
- Команда `systemctl reload <имя_сервиса>.service` – перечитывает конфигурационный файл службы;
- Команда `systemctl mask <имя_сервиса>.service` – маскирует службу. Делает невозможным запуск службы как в ручном режиме, так и через зависимость от другой службы. В каталоге `/etc/systemd/system` при этом создается символической ссылкой на `/dev/null`;
- Команда `systemctl unmask <имя_сервиса>.service` – отменяет маскировку службы. Возвращает возможность запуска службы. Ссылка на `/dev/null` при этом удаляется;
- Команда `systemctl reenable <имя_сервиса>.service` – выполняет disable, а затем enable службы;
- Команда `systemctl is-active <имя_сервиса>.service` – проверяет, запущена ли служба в настоящий момент;
- Команда `systemctl list-unit-files --type=service` – выводит список всех служб и их состояние.

```
localadmin@astra:~$ systemctl list-unit-files --type=service
UNIT FILE                                  STATE
acpi-support.service                       disabled
acpid.service                              disabled
alsa-restore.service                       static
alsa-state.service                         static
```

(continues on next page)

(продолжение с предыдущей страницы)

```
alsa-utils.service                          masked
anacron.service                             enabled
apt-daily-upgrade.service                   static
apt-daily.service                           static
astra-event-diagnostics.service             static
astra-mount-lock.service                    disabled
astra-orientation.service                   enabled

localadmin@astra:~$ systemctl status cups.service
o cups.service - CUPS Scheduler
  Loaded: loaded (/lib/systemd/system/cups.service; enabled; vendor preset: enabled)
  Active: active (running) since Tue 2023-08-22 00:00:51 MSK; 11h ago
   Docs: man:cupsd(8)
Main PID: 6485 (cupsd)
  Status: "Scheduler is running..."
   Tasks: 1 (limit: 2255)
  Memory: 4.2M
     CPU: 550ms
  CGroup: /system.slice/cups.service
          └─6485 /usr/sbin/cupsd -l

localadmin@astra:~$ sudo systemctl stop cups.service
localadmin@astra:~$ systemctl status cups.service
o cups.service - CUPS Scheduler
  Loaded: loaded (/lib/systemd/system/cups.service; enabled; vendor preset: enabled)
  Active: inactive (dead) since Tue 2023-08-22 11:53:58 MSK; 10s ago

localadmin@astra:~$ sudo systemctl mask cups.service
Created symlink /etc/systemd/system/cups.service → /dev/null.

localadmin@astra:~$ ls -l /lib/systemd/system/cups.service
-rw-r--r-- 1 root root 278 апр  5 17:04 /lib/systemd/system/cups.service

localadmin@astra:~$ ls -l /etc/systemd/system/cups.service
lrwxrwxrwx 1 root root 9 авг 22 11:56 /etc/systemd/system/cups.service -> /dev/null

localadmin@astra:~$ sudo systemctl start cups.service
Failed to start cups.service: Unit cups.service is masked.

localadmin@astra:~$ sudo systemctl unmask cups.service
Removed /etc/systemd/system/cups.service.

localadmin@astra:~$ ls -l /etc/systemd/system/cups.service
ls: невозможно получить доступ к '/etc/systemd/system/cups.service': Нет такого файла␣
↪или каталога

localadmin@astra:~$ sudo systemctl start cups.service
localadmin@astra:~$ systemctl status cups.service
o cups.service - CUPS Scheduler
  Loaded: loaded (/lib/systemd/system/cups.service; enabled; vendor preset: enabled)
  Active: active (running) since Tue 2023-08-22 11:59:00 MSK; 7s ago
    Docs: man:cupsd(8)
Main PID: 9169 (cupsd)
  Status: "Scheduler is running..."
   Tasks: 1 (limit: 2255)
  Memory: 5.5M
```

(continues on next page)

(продолжение с предыдущей страницы)

```
     CPU: 107ms
  CGroup: /system.slice/cups.service
          └─9169 /usr/sbin/cupsd -l

localadmin@astra:~$ sudo systemctl disable cups.service
Synchronizing state of cups.service with SysV service script with /lib/systemd/
↪systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install disable cups
Removed /etc/systemd/system/printer.target.wants/cups.service.
Removed /etc/systemd/system/sockets.target.wants/cups.socket.
Removed /etc/systemd/system/multi-user.target.wants/cups.path.

localadmin@astra:~$ sudo systemctl enable cups.service
Synchronizing state of cups.service with SysV service script with /lib/systemd/
↪systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable cups
Created symlink /etc/systemd/system/printer.target.wants/cups.service → /lib/systemd/
↪system/cups.service.
Created symlink /etc/systemd/system/sockets.target.wants/cups.socket → /lib/systemd/
↪system/cups.socket.
Created symlink /etc/systemd/system/multi-user.target.wants/cups.path → /lib/systemd/
↪system/cups.path.
```

Journalctl

Утилита предназначена для просмотра журналов, сформированных сервисом journald. Позволяет гибко настраивать вывод через параметры. Содержит большое количество параметров.

Примеры использования:

Команда `sudo journalctl -u cups.service -o short` – выводит информацию из журналов о сервисе печати cups в сокращённом виде.

```
localadmin@astra:~$ sudo journalctl -u cups.service -o short
-- Logs begin at Mon 2023-08-21 16:23:15 MSK, end at Tue 2023-08-22 12:11:25 MSK. --
авг 21 16:23:18 astra systemd[1]: Starting CUPS Scheduler...
авг 21 16:23:18 astra systemd[1]: Started CUPS Scheduler.
. . .
авг 22 11:53:58 astra systemd[1]: cups.service: Consumed 558ms CPU time.
авг 22 11:59:00 astra systemd[1]: Starting CUPS Scheduler...
авг 22 11:59:00 astra systemd[1]: Started CUPS Scheduler.
```

Команда `sudo journalctl -u cups.service -o json --since "2023-08-22 11:55:00" > cups.log.json` – записывает в файл информацию о событиях с сервисом cups, произошедших после 11:55 22 августа 2023 года в формате json.

```
localadmin@astra:~$ sudo journalctl -u cups.service -o json --since "2023-08-22␣
↪11:55:00" > cups.log.json
localadmin@astra:~$ cat cups.log.json
. . .
```

(continues on next page)

(продолжение с предыдущей страницы)

```
{"_COMM":"systemd","CODE_FUNC":"job_log_done_status_message","_MACHINE_ID":
↪"dc9b728287b64f9bbc71
7e6f3758ba29","SYSLOG_FACILITY":"3","_SYSTEMD_CGROUP":"/init.scope","_PID":"1","__
↪MONOTONIC_TIME
STAMP":"70550518702","_CAP_EFFECTIVE":"1fffffffff","MESSAGE_ID":
↪"39f53479d3a045ac8e11786248231f
bf","UNIT":"cups.service","JOB_RESULT":"done","INVOCATION_ID":
↪"b0b8216f5b114dd58a7a819aa9d6d802"
,"_EXE":"/usr/lib/systemd/systemd","JOB_ID":"2225","SYSLOG_IDENTIFIER":"systemd","_GID
↪":"0","_HO
STNAME":"astra","MESSAGE":"Started CUPS Scheduler.","_SYSTEMD_UNIT":"init.scope","_
↪SOURCE_REALTI
ME_TIMESTAMP":"1692694740380331","_SELINUX_CONTEXT":"unknown","_SYSTEMD_SLICE":"-.
↪slice","PRIORI
TY":"6","_TRANSPORT":"journal","_UID":"0","__CURSOR":
↪"s=519569fb055a42629f69fdc032ceda5b;i=109a;
b=b0bc07a3ea1740d98427f110c3cbb018;m=106d237bae;t=6037f34b6aad0;x=6bf8bc61b5f700","_
↪CMDLINE":"/s
bin/init","CODE_LINE":"950","JOB_TYPE":"start","CODE_FILE":"../src/core/job.c","__
↪REALTIME_TIMES
TAMP":"1692694740380368","_BOOT_ID":"b0bc07a3ea1740d98427f110c3cbb018"}
```

Команда `sudo journalctl --disk-usage` – выводит информацию о размерах файлов журналов.

```
localadmin@astra:~$ sudo journalctl --disk-usage
Archived and active journals take up 7.4M in the file system.
```

Команда `sudo journalctl -f` – выводит на экран информацию обо всех поступающих в реальном времени событиях.

Команда `sudo journalctl –f –u cups.service` – выводит на экран информацию обо всех поступающих в реальном времени событиях, связанных с сервисом cups.

Systemd-analyze

Утилита `systemd-analyze` позволяет производить анализ времени запуска служб.

```
localadmin@astra:~$ systemd-analyze
Startup finished in 4.919s (kernel) + 8.295s (userspace) = 13.214s
graphical.target reached after 8.245s in userspace
```

Вывод утилиты можно сохранить в файл с расширением. svg, просматривать, например, в браузере chromium:

```
localadmin@astra:~$ systemd-analyze plot > plot.svg
localadmin@astra:~$ chromium plot.svg
```

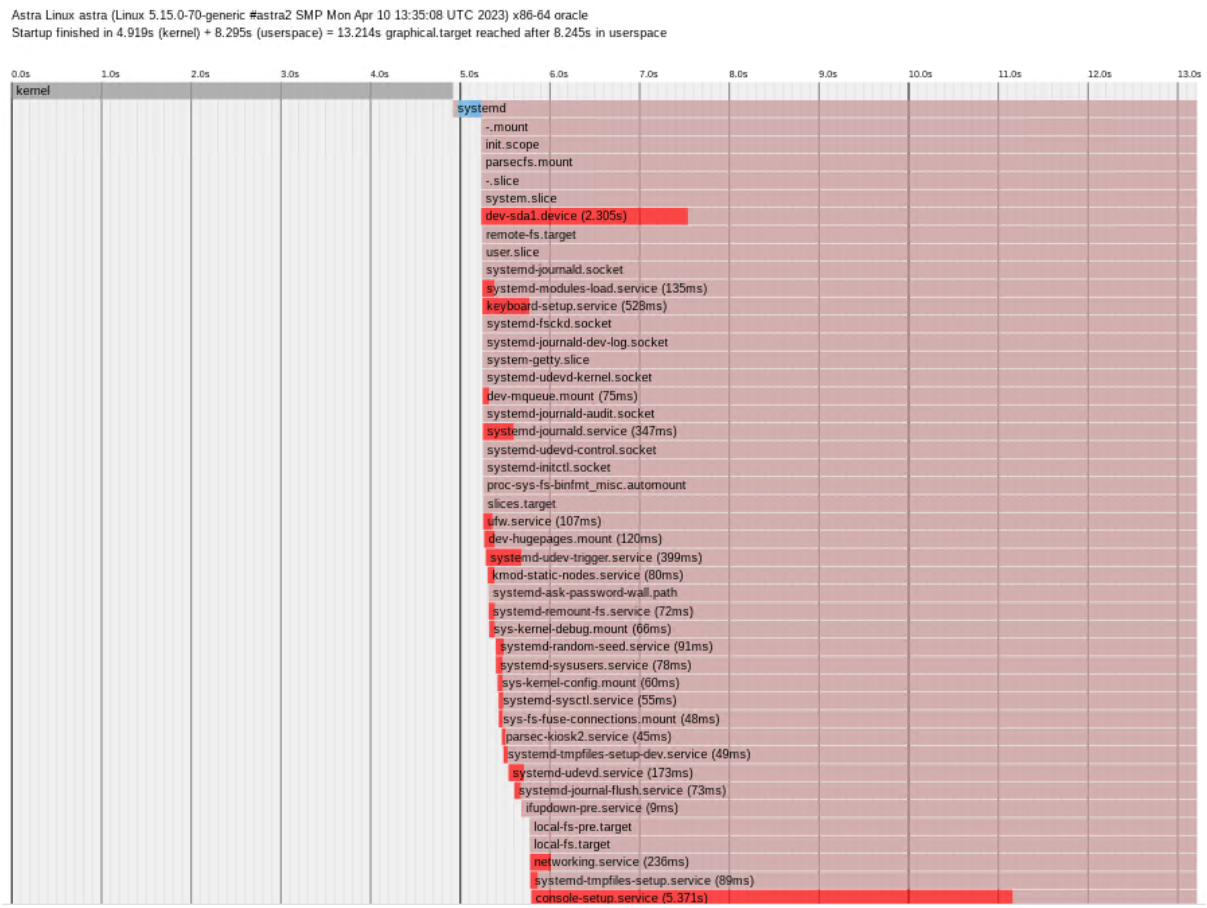


рис. 1.108 – Анализ времени запуска служб

Systemd-cgls

Утилита systemd-cgls выводит дерево контрольных групп (cgroups).

```
localadmin@astra:~$ systemd-cgls
Control group /:
-.slice
├─user.slice
│ ├─user-107.slice
│ │ ├─user@107.service
│ │ │ └─init.scope
│ │ │   ├─1156 /lib/systemd/systemd --user
│ │ │   └─1157 (sd-pam)
│ │ └─session-c1.scope
│ │   └─1122 /usr/lib/xorg/Xorg -br -novtswitch -quiet -keeptty :0 vt7 -logfile /var/
↪log/fly-dm/
│ └─user-1000.slice
│   ├─user@1000.service
│   │ └─init.scope
│   │   ├─1258 /lib/systemd/systemd --user
│   │   └─1259 (sd-pam)
│   └─session-3.scope
│     ├─1177 -:0
│     ├─1281 fly-wm
```

(continues on next page)

(продолжение с предыдущей страницы)

```
│      ├─1346 /usr/bin/dbus-launch --exit-with-session --sh-syntax
│      ├─1347 /usr/bin/dbus-daemon --syslog --fork --print-pid 5 --print-address 7 --
↪session
│      ├─1370 /usr/bin/kglobalaccel5
│      ├─1376 /usr/bin/VBoxClient -clipboard
. . .
```

Systemd-cgtop

Аналог утилиты `top` для контрольных групп.

```
Control Group                                                Tasks   %CPU   Memory  Input/s Output/s
/                                                              281    4,9    1.8G        -        -
/user.slice                                                    142    8,2  822.1M        -        -
/user.slice/user-1000.slice                                    138    6,9  710.7M        -        -
/user.slice/user-1000.slice/session-3.scope                    136    6,9  707.9M        -        -
/user.slice/user-107.slice                                       4    0,9  111.3M        -        -
/user.slice/user-107.slice/session-c1.scope                      2    0,9  103.8M        -        -
/system.slice                                                   59    0,1  910.6M        -        -
/system.slice/vboxadd-service.service                           14    0,1    2.3M        -        -
/system.slice/syslog-ng.service                                 11    0,0   41.6M        -        -
/init.scope                                                      1      -    6.8M        -        -
/system.slice/NetworkManager.service                             3      -    9.6M        -        -
/system.slice/alsa-state.service                                 1      -    1.2M        -        -
/system.slice/astra-orientation.service                          3      -    3.0M        -        -
```

loginctl

Утилита управления менеджером systemd-logind.

- Команда `loginctl` – выводит список сессий.

```
localadmin@astra:~$ loginctl
SESSION  UID USER       SEAT  TTY
     3 1000 localadmin seat0 tty7
    c1  107 fly-dm     seat0 tty7

2 sessions listed.
```

- Команда `loginctl session-status <номер_сессии>` – выводит статус сессии по её номеру.

```
localadmin@astra:~$ loginctl session-status 3
3 - localadmin (1000)
          Since: Mon 2023-08-21 16:23:37 MSK; 20h ago
         Leader: 1177 (fly-dm)
           Seat: seat0; vc7
            TTY: tty7
        Service: fly-dm; type x11; class user
          State: active
           Unit: session-3.scope
                 ├─1177 -:0
                 ├─1281 fly-wm
                 ├─1346 /usr/bin/dbus-launch --exit-with-session --sh-syntax
                 ├─1347 /usr/bin/dbus-daemon --syslog --fork --print-pid 5 --print-
```

(continues on next page)

(продолжение с предыдущей страницы)

```
↪address 7 --
                ├─1370 /usr/bin/kglobalaccel5
                ├─1376 /usr/bin/VBoxClient --clipboard
                ├─1377 /usr/bin/VBoxClient --clipboard
                ├─1391 /usr/bin/VBoxClient --seamless
```

- Команда `loginctl show-session` – выводит свойства самого менеджера (без аргументов) или указанной сессии.

```
localadmin@astra:~$ loginctl show-session
EnableWallMessages=no
NAutoVTs=6
KillUserProcesses=yes
RebootToFirmwareSetup=no
IdleHint=no
IdleSinceHint=0
IdleSinceHintMonotonic=0
BlockInhibited=handle-power-key:handle-suspend-key:handle-hibernate-key:handle-lid-
↪switch
DelayInhibited=shutdown:sleep
InhibitDelayMaxUSec=5s
UserStopDelayUSec=10s
HandlePowerKey=ignore
HandleSuspendKey=suspend
HandleHibernateKey=hibernate
HandleLidSwitch=suspend
HandleLidSwitchDocked=ignore
HoldoffTimeoutUSec=30s
IdleAction=ignore
IdleActionUSec=30min
PreparingForShutdown=no
PreparingForSleep=no
Docked=no
LidClosed=no
OnExternalPower=yes
RemoveIPC=yes
RuntimeDirectorySize=207409152
InhibitorsMax=8192
NCurrentInhibitors=5
SessionsMax=8192
NCurrentSessions=3
MaxIlevShutdown=no
```

- Команда `loginctl lock-session/unlock-session имя_сессии` – активирует/деактивирует блокировку экрана в указанной сессии.

- Команда `loginctl kill-session имя_сессии` – отправляет сигнал SIGKILL одному или нескольким процессам сессии.

```
localadmin@astra:~$ loginctl list-sessions
SESSION  UID USER       SEAT  TTY
     3 1000 localadmin seat0 tty7
    60 1000 localadmin seat0 tty2
```

(continues on next page)

(продолжение с предыдущей страницы)

```
    c1  107 fly-dm     seat0 tty7

3 sessions listed.
localadmin@astra:~$ loginctl kill-session 60
localadmin@astra:~$ loginctl
SESSION  UID USER       SEAT  TTY
     3 1000 localadmin seat0 tty7
    c1  107 fly-dm     seat0 tty7

2 sessions listed.
```

17.10 Практика и тестирование

17.10.1 Практическая работа: Модуль 17. Загрузка системы и управление службами systemd

См. также *Требования, правила и цели выполнения практической работы*

17.10.1.1 Практические задания

Задание 1.

1. Перезагрузите тестовую виртуальную машину и в момент её запуска измените команды запуска так, чтобы отключить «тихую» загрузку ОС.

2. Перенастройте загрузчик, чтобы при последующих запусках ОС «тихий» режим загрузки ОС был отключен, а тайм-аут отображения меню загрузки GRUB был 7 секунд. Убедитесь, что настройки применились.

3. Выведите на экран целевое состояние системы по умолчанию. Перезагрузите машину, используя целевое состояние systemd.

4. Выведите на экран сообщения ядра ОС от начала загрузки и ознакомьтесь с ними.

Задание 2.

Для успешного прохождения этого задания у виртуальной машины должны быть два CPU.

1. Если вы не создали модули службы fly-message-monitor.service и resource-demanding1.service (раздел 17.6) в процессе изучения материала модуля, то создайте их, в том числе настройте контрольную группу для службы resource-demanding1.service.

2. Создайте службу **resource-demanding2.service** по аналогии со службой resource-demanding1.service, но так, чтобы скрипт, который она запускает, назывался **resource-**

demanding2.sh и запускался на **втором процессоре**.

3. Создайте скрипт **/usr/local/bin/cpu_usage.sh** со следующим содержимым:

```
#!/bin/bash
CPU_THRESHOLD_WARNING=85 # %
CPU_THRESHOLD_CRITICAL=95 # %
CPU_THRESHOLD_PERIOD=5 # секунд
CPU_THRESHOLD_ITERATION=3 # раз
MSG_FILE="/tmp/fly-msg/msg.txt"

while :; do
# Get the first line with aggregate of all CPUs
cpu_now=($(head -n1 /proc/stat))
# Get all columns but skip the first (which is the "cpu" string)
cpu_sum="${cpu_now[@]:1}"
# Replace the column seperator (space) with +
cpu_sum=$((${cpu_sum// /+}))
# Get the delta between two reads
cpu_delta=$((cpu_sum - cpu_last_sum))
# Get the idle time Delta
cpu_idle=$((cpu_now[4]- cpu_last[4]))
# Calc time spent working
cpu_used=$((cpu_delta - cpu_idle))
# Calc percentage
cpu_usage=$((100 \* cpu_used / cpu_delta))

# Keep this as last for our next read
cpu_last=("${cpu_now[@]}")
cpu_last_sum=$cpu_sum

if [[ "$cpu_usage" -ge "$CPU_THRESHOLD_WARNING" ]]; then
   ((cpu_warning_count++))
   if [[ "$cpu_usage" -ge "$CPU_THRESHOLD_CRITICAL" ]]; then
      ((cpu_critical_count++))
   else
      cpu_critical_count=0
   fi
else
   cpu_warning_count=0
   cpu_critical_count=0
fi

msg_critical="critical\\ndialog-warning\\n<H1>Критически высокая нагрузка на
CPU!</H1><p>Использование CPU $cpu_usage%\\t$(date)</p>"
msg_warning="critical\\ndialog-warning\\n<H1>Высокая нагрузка на CPU!</H1><p>
↪Использование CPU
$cpu_usage%\\t$(date)</p>"
msg_ok="normal\\ndialog-information\\n<H1>Нагрузка на CPU в пределах нормы</H1><p>
↪Использование CPU
$cpu_usage%\\t$(date)</p>"


if [[ cpu_critical_count -ge CPU_THRESHOLD_ITERATION ]]; then
         echo -en $msg_critical > $MSG_FILE
         send_ok=1
elif [[ cpu_warning_count -ge CPU_THRESHOLD_ITERATION ]]; then
```

(continues on next page)

(продолжение с предыдущей страницы)

```
        echo -en $msg_warning > $MSG_FILE
        send_ok=1
elif [[ send_ok -eq 1 ]]; then
        echo -en $msg_ok > $MSG_FILE
        send_ok=0
fi

# Wait a second before the next read
sleep $CPU_THRESHOLD_PERIOD

echo "CPU usage at $cpu_usage%"
echo "Warning counter $cpu_warning_count"
echo "Critical counter $cpu_critical_count"
done
```

Не забудьте сделать файл исполнимым.

4. Создайте модуль службы и назовите его **cpu-monitor.service**, чтобы он запускал скрипт **/usr/local/bin/cpu_usage.sh** и стартовал автоматически при загрузке ОС только при цели **graphical.target**.

5. Выведите на экран содержимое конфигурационного файла модуля **cpu-monitor.service**, используя **systmctl**.

6. Отредактируйте файл модуля так, чтобы он запускался только после службы **fly-message-monitor.service**, используя утилиту **systemctl**.

7. Запустите службу **cpu-monitor.service** и убедитесь, что она работает.

8. Переведите систему в целевое состояние **multi-user.traget** и убедитесь, что службы **cpu-monitor.service** и **fly-message-monitor.service** не запущены.

9. Переведите систему в целевое состояние **graphical.target** и убедитесь, что службы **cpu-monitor.service** и **fly-message-monitor.service** запустились.

10. Запустите службу **resource-demanding1.service**.

11. Запустите службу **resource-demanding2.service** и проверьте, что при высокой (более 85%) нагрузке на CPU вы получаете сообщения от скрипта мониторинга.

Задание 3.

1. Измените настройки контрольной группы службы **resource-demanding1.service** так, чтобы **суммарная нагрузка** на два CPU была в пределах **85-94%.** Убедитесь, что сообщение от скрипта мониторинга изменилось.

2. Измените настройки контрольной группы службы **resource-demanding1.service** так, чтобы **суммарная нагрузка** на два CPU была ниже **85%.** Убедитесь, что сообщение от скрипта мониторинга изменилось и больше не появляется.

3. Выведите на экран дерево контрольных групп и найдите в нем контрольную группу для сервиса **resource-demanding1.service**.

Задание 4.

1. Выведите на экран все запущенные модули подсистемы systemd.

2. Выведите на экран все модули служб подсистемы systemd.

3. Выведите на экран в отдельном терминале информацию обо всех поступающих в реальном времени событиях подсистемы systemd.

4. Перезапустите службу **resource-demanding1.service**. Выведите на экран её статус.

5. Остановите службы **resource-demanding1.service**, **resource-demanding2.service** и **cpu-monitoring.service**.

6. Выведите на экран информацию из журналов о сервисе **resource-demanding1.service** в сокращённом виде.

17.10.1.2 Ответы на практические задания (пошаговые инструкции)

Задание 1.

1. Перезагрузите тестовую виртуальную машину и в момент её запуска измените команды запуска так, чтобы отключить «тихую» загрузку ОС.

   a. Перезагрузите машину и в момент появления меню загрузчика GRUB нажмите клавишу <**e**>.

   b. Введите имя пользователя и пароль от GRUB (который был задан в момент установки ОС).

   c. В третьей с конца строке, начинающейся с «linux», удалите слово «quiet» и нажмите <**F10**>.

2. Перенастройте загрузчик, чтобы при последующих запусках ОС «тихий» режим загрузки ОС был отключен, а тайм-аут отображения меню загрузки GRUB был 7 секунд.

   a. Откройте файл /etc/default/grub на редактирование от имени администратора.

```
sudo vim /etc/default/grub
```

b. Измените параметр GRUB_TIMEOUT=5 на GRUB_TIMEOUT=7.

c. Измените параметр GRUB_CMDLINE_LINUX_DEFAULT=»parsec.mac=0 quiet net.ifnames=0» на RUB_CMDLINE_LINUX_DEFAULT=»parsec.mac=0 net.ifnames=0».

d. Сохраните файл и выполните обновление конфигурации GRUB командой `sudo update-grub`.

e. Убедитесь, что настройки применились, изучив файл /boot/grub/grub.cfg.

```
less /boot/grub/grub.cfg
```

3. Выведите на экран целевое состояние системы по умолчанию. Перезагрузите машину, используя целевое состояние systemd.

```
systemctl get-default
sudo systemctl isolate reboot.target
```

4. Выведите на экран сообщения ядра ОС от начала загрузки и ознакомьтесь с ними.

```
sudo dmesg
```

Задание 2.

Для успешного прохождения этого задания у виртуальной машины должны быть два CPU.

1. Если вы не создали модули службы fly-message-monitor.service и resource-demanding1.service (раздел 17.6) в процессе изучения материала модуля, то создайте их, в том числе настройте контрольную группу для службы resource-demanding1.service. **Подробно об их создании изложено в тексте модуля**.

2. Создайте службу **resource-demanding2.service** по аналогии со службой resource-demanding1.service, но так, чтобы скрипт, который она запускает, назывался **resource-demanding2.sh** и запускался на **втором процессоре**.

   a. От имени суперпользователя создайте скрипт **resource-demanding2.sh** в каталоге */usr/local/bin/* (который предназначен в том числе для хранения скриптов запускаемых на уровне всей системы) и предоставьте владельцу право на его выполнение.

```
#!/bin/bash
/usr/bin/sha256sum /dev/urandom > /dev/null
```

```
sudo -i
```

(continues on next page)

(продолжение с предыдущей страницы)

```
echo "/usr/bin/sha256sum /dev/urandom > /dev/null" > /usr/local/bin/resource-
↪demanding2.sh
chmod u+x /usr/local/bin/resource-demanding2.sh
exit
```

b. Создайте файл службы */etc/systemd/system/resource-demanding2.service* со следующей конфигурацией:

```
[Unit]
Description=Сервис, воспроизводящий высокую нагрузку на CPU 1
After=graphical.target

[Service]
ExecStart=taskset 0x00000002 /usr/bin/bash /usr/local/bin/resource-
↪demanding2.sh
Type=simple

[Install]
WantedBy=graphical.target
```

```
sudo vim /etc/systemd/system/resource-demanding2.service
sudo systemctl daemon-reload
```

3. Создайте скрипт **/usr/local/bin/cpu_usage.sh** со следующим содержимым:

```
#!/bin/bash
CPU_THRESHOLD_WARNING=85 # %
CPU_THRESHOLD_CRITICAL=95 # %
CPU_THRESHOLD_PERIOD=5 # секунд
CPU_THRESHOLD_ITERATION=3 # раз
MSG_FILE="/tmp/fly-msg/msg.txt"

while :; do
# Get the first line with aggregate of all CPUs
cpu_now=($(head -n1 /proc/stat))
# Get all columns but skip the first (which is the "cpu" string)
cpu_sum="${cpu_now[@]:1}"
# Replace the column seperator (space) with +
cpu_sum=$((${cpu_sum// /+}))
# Get the delta between two reads
cpu_delta=$((cpu_sum - cpu_last_sum))
# Get the idle time Delta
cpu_idle=$((cpu_now[4]- cpu_last[4]))
# Calc time spent working
cpu_used=$((cpu_delta - cpu_idle))
# Calc percentage
cpu_usage=$((100 \* cpu_used / cpu_delta))

# Keep this as last for our next read
cpu_last=("${cpu_now[@]}")
cpu_last_sum=$cpu_sum


if [[ "$cpu_usage" -ge "$CPU_THRESHOLD_WARNING" ]]; then
```

(continues on next page)

(продолжение с предыдущей страницы)

```
  ((cpu_warning_count++))
  if [[ "$cpu_usage" -ge "$CPU_THRESHOLD_CRITICAL" ]]; then
    ((cpu_critical_count++))
  else
    cpu_critical_count=0
  fi
else
  cpu_warning_count=0
  cpu_critical_count=0
fi

msg_critical="critical\\ndialog-warning\\n<H1>Критически высокая нагрузка на CPU!
↪</H1><p>Использование CPU $cpu_usage%\\t$(date)</p>"
msg_warning="critical\\ndialog-warning\\n<H1>Высокая нагрузка на CPU!</H1><p>
↪Использование CPU
$cpu_usage%\\t$(date)</p>"
msg_ok="normal\\ndialog-information\\n<H1>Нагрузка на CPU в пределах нормы</H1>
↪<p>Использование CPU
$cpu_usage%\\t$(date)</p>"


if [[ cpu_critical_count -ge CPU_THRESHOLD_ITERATION ]]; then
        echo -en $msg_critical > $MSG_FILE
        send_ok=1
elif [[ cpu_warning_count -ge CPU_THRESHOLD_ITERATION ]]; then
        echo -en $msg_warning > $MSG_FILE
        send_ok=1
elif [[ send_ok -eq 1 ]]; then
        echo -en $msg_ok > $MSG_FILE
        send_ok=0
fi

# Wait a second before the next read
sleep $CPU_THRESHOLD_PERIOD

echo "CPU usage at $cpu_usage%"
echo "Warning counter $cpu_warning_count"
echo "Critical counter $cpu_critical_count"
done
```

Не забудьте сделать файл исполнимым.

```
sudo vim /usr/local/bin/cpu_usage.sh
chmod u+x /usr/local/bin/cpu_usage.sh
```

4. Создайте модуль службы и назовите его **cpu-monitor.service**, чтобы он запускал скрипт **/usr/local/bin/ cpu_usage.sh** и стартовал автоматически при загрузке ОС только при цели **graphical.target**. Создайте файл службы */etc/systemd/system/cpu-monitor.service* со следующей конфигурацией:

```
[Unit]
Description=Мониторинг нагрузки на CPU и отправка уведомлений при превышении␣
↪порога
```

(continues on next page)

(продолжение с предыдущей страницы)

```
[Service]
ExecStart=/usr/local/bin/cpu_usage.sh
Type=simple

[Install]
WantedBy=graphical.target
```

```
sudo vim /etc/systemd/system/cpu-monitor.service
```

Установите автоматический запуск службы:

```
sudo systemctl enable cpu-monitor.service
```

5. Выведите на экран содержимое конфигурационного файла модуля **cpu-monitor.service**, используя **systmctl**.

```
systemctl cat cpu-monitor.service
```

6. Отредактируйте файл модуля **cpu-monitor.service** так, чтобы он запускался только после службы **fly-message-monitor. service**, используя утилиту **systemctl**. Измените директиву WantedBy в секции [Install] на:

```
WantedBy=fly-message-monitor.service
```

Открыв файл модуля на редактирование с помощью команды:

```
sudo systemctl --full edit cpu-monitor.service
```

7. Запустите службу **cpu-monitor.service** и убедитесь, что она работает.

```
sudo systemctl start cpu-monitor.service
```

8. Переведите систему в целевое состояние **multi-user.traget** и убедитесь, что службы **cpu-monitor.service** и **fly-message-monitor.service** не запущены.

```
sudo systemctl isolate multi-user.target
systemctl status fly-message-monitor
systemctl status cpu-monitor
```

9. Переведите систему в целевое состояние **graphical.target** и убедитесь, что службы **cpu-monitor.service** и **fly-message-monitor.service** запустились.

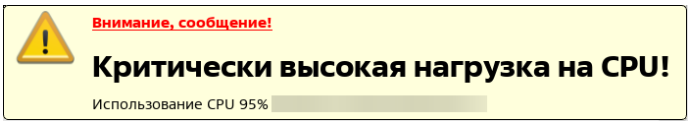
```
sudo systemctl isolate graphical.target
systemctl status fly-message-monitor
systemctl status cpu-monitor
```

10. Запустите службу **resource-demanding1.service**.

```
sudo systemctl start resource-demanding1.service
```

11. Запустите службу **resource-demanding2.service** и проверьте, что при высокой (более 85%) нагрузке на CPU вы получаете сообщения от скрипта мониторинга.

```
sudo systemctl start resource-demanding2.service
```



Задание 3.

1. Измените настройки контрольной группы службы **resource-demanding1.service** так, чтобы **суммарная нагрузка** на два CPU была в пределах **85-94%.** Убедитесь, что сообщение от скрипта мониторинга изменилось.

   a. Установите значение параметра **CPUQuota** в файле в 80%.

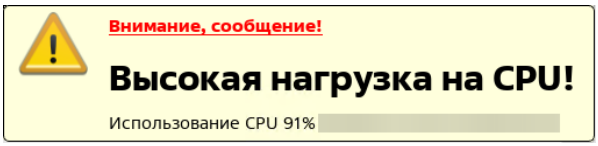
```
sudo vim /etc/systemd/system/resource-demanding1.service.d/10-CPUSettings.conf
```

   b. Перечитайте конфигурацию сервисов.

```
sudo systemctl daemon-reload
```

   c. Перезапустите сервис resource-demanding1.service.

```
sudo systemctl restart resource-demanding1.service
```



2. Измените настройки контрольной группы службы **resource-demanding1.service** так, чтобы **суммарная нагрузка** на два CPU была ниже **85%.** Убедитесь, что сообщение от скрипта мониторинга изменилось и больше не появляется.

   a. Установите значение параметра **CPUQuota** в файле в 50%.

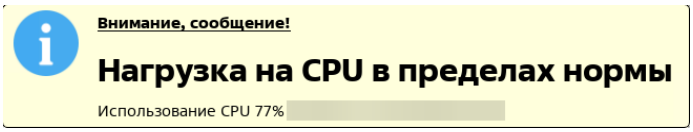
```
sudo vim /etc/systemd/system/resource-demanding1.service.d/10-CPUSettings.conf
```

   b. Перечитайте конфигурацию сервисов.

```
sudo systemctl daemon-reload
```

   c. Перезапустите сервис resource-demanding1.service.

```
sudo systemctl restart resource-demanding1.service
```



3. Выведите на экран дерево контрольных групп и найдите в нем контрольную группу для сервиса **resource-demanding1.service**.

```
systemd-cgls
```

Выполните поиск в открывшейся утилите less через ввод «/resource-demanding1.sh».

Задание 4.

1. Выведите на экран все запущенные модули подсистемы systemd.

```
systemctl
```

2. Выведите на экран все модули служб подсистемы systemd.

```
systemctl --type=service
```

3. Выведите на экран в отдельном терминале информацию обо всех поступающих в реальном времени событиях подсистемы systemd. Откройте новый терминал и введите команду:

```
sudo journalctl -f
```

4. Перезапустите службу **resource-demanding1.service**. Выведите на экран её статус.

```
sudo systemctl restart resource-demanding1.service
systemctl status resource-demanding1.service
```

5. Остановите службы **resource-demanding1.service**, **resource-demanding2.service** и **cpu-monitoring.service**.

```
sudo systemctl stop resource-demanding1.service
resource-demanding2.service cpu-monitor.service
```

6. Выведите на экран информацию из журналов о сервисе **resource-demanding1.service** в сокращённом виде.

```
sudo journalctl -u resource-demanding1.service -o short
```

17.10.2 Тест: Модуль 17. Загрузка системы и управление службами systemd

Пожалуйста, ответьте на несколько вопросов, чтобы закрепить полученные знания. Правильные ответы для самопроверки вы сможете найти в конце списка. Обратите внимание, что у некоторых вопросов может быть несколько верных вариантов ответов.

17.10.2.1 Вопросы

Вопрос 1. Какое сочетание технологий рекомендуется использовать в современных системах?

a. UEFI+MBR

b. BIOS+GPT

c. UEFI+GPT

d. BIOS+GPT

Вопрос 2. Расположите пункты запуска компьютера с ОС Linux от момента включения до момента запуска ядра в правильном порядке:

a. Тестирование и инициализация оборудования > Поиск загрузочного устройства > Запуск загрузчика первого этапа > Запуск загрузчика второго этапа > Вывод загрузочного меню > Загрузка ядра > Загрузка init ram fs > Запуск ядра

b. Тестирование и инициализация оборудования > Запуск загрузчика второго этапа > Запуск загрузчика первого этапа > Загрузка ядра > Загрузка init ram fs > Запуск ядра

c. Поиск загрузочного устройства > Запуск загрузчика первого этапа > Тестирование и инициализация оборудования > Запуск загрузчика второго этапа > Вывод загрузочного меню > Загрузка ядра > Загрузка init ram fs > Запуск ядра

Вопрос 3. Расположите пункты запуска компьютера с ОС Linux от момента запуска ядра до момента загрузки системы в выбранном целевом состоянии:

a. Инициализация наиболее важных функций ядра > Монтирование init ram fs в корень > Запуск скрипта /init > Монтирование устройства с системным диском в корень > Запуск процесса systemd > Загрузка модулей ядра и драйверов с RAM-диска > Запуск модулей systemd

b. Инициализация наиболее важных функций ядра > Монтирование init ram fs в корень > Запуск скрипта /init > Загрузка модулей ядра и драйверов с RAM-диска > Монтирование устройства с системным диском в корень > Запуск процесса systemd > Запуск модулей systemd

c. Монтирование init ram fs в корень > Инициализация наиболее важных функций ядра > Запуск скрипта /init > Загрузка модулей ядра и драйверов с RAM-диска > Монтирование устройства с системным диском в корень > Запуск процесса systemd > Запуск модулей systemd

Вопрос 4. В каком файле следует выполнять настройки GRUB?

a. Файл `/etc/grub.d/grub.cfg`

b. Файл `/boot/grub/grub.cfg`

c. Файл `/etc/default/grub`

d. Файл `/boot/default/grub`

Вопрос 5. Какая команда позволит применить новую конфигурацию GRUB?

a. Команда `update-grub`

b. Команда `grub-mkconfig`

c. Команда `grub-update`

d. Команда `renew-grub`

Вопрос 6. Какой модуль systemd не существует?

a. Модуль `.service`

b. Модуль `.device`

c. Модуль `.mount`

d. Модуль `.level`

e. Модуль `.target`

Вопрос 7. Какому уровню загрузки соответствует целевое состояние ОС `poweroff.target`:

a. runlevel 0

b. runlevel 1

c. runlevel 2

d. runlevel 3

e. runlevel 4

f. runlevel 5

g. runlevel 6

Вопрос 8. Какие секции могут присутствовать в конфигурационном файле модуля службы systemd:

a. Секция `[Install]`

b. Секция `[Unit]`

c. Секция `[System]`

d. Секция `[Remove]`

e. Секция `[Service]`

f. Секция `[Path]`

Вопрос 9. Какой командой можно отслеживать события со службами systemd в реальном времени?

a. Команда `sudo systemctl -f`

b. Команда `sudo journalctl -f`

c. Команда `sudo journalctl -r`

d. Команда `sudo systemctl journal --realtime`

Вопрос 10. Какая команда выведет список всех служб и их состояния:

a. Команда `systemctl list-unit-files --type=service`

b. Команда `systemctl service --status`

c. Команда `systemctl unit --status`

d. Команда `systemctl service --type=service`

Ответы на вопросы

1. c
2. a
3. b
4. c
5. a

6. d
7. a
8. a, b, e
9. b
10. a

17.11 Заключение

В этом модуле мы познакомились с порядком загрузки ОС Astra Linux, узнали, что такое GRUB, как его настраивать и как управлять параметрами загрузки ядра ОС. Мы также познакомились с системой инициализации и управления службами Systemd, её составом, модулями и контрольными группами. Научились создавать свои собственные модули и управлять ими, в том числе ограничивать потребление ресурсов через контрольные группы.

В следующем модуле мы познакомимся с модулями ядра ОС и подсистемой управления устройствами udev.

17.12 Дополнительные источники информации

- Практика загрузки. Изучение процесса загрузки Linux, Windows и Unix. Йогеш Барбара
- Инструкция по изменению параметров загрузчика доступна по ссылке: https://wiki.astralinux.ru/pages/viewpage.action?pageId=18874503
- Инструкция по установке загрузчика GRUB2 в UEFI GPT доступна по ссылке: https://wiki.astralinux.ru/pages/viewpage.action?pageId=130424699
- С полным руководством по GRUB и списку его команд на английском языке можно ознакомиться по ссылке: https://www.gnu.org/software/grub/manual/grub/grub.html
- Подробнее о groups можно почитать по ссылке: https://manpages.debian.org/unstable/manpages-ru/cgroups.7.ru.html

17.13 Обратная связь

Если остались вопросы, то их всегда можно задать в специальной теме.

Модуль 18. Управление устройствами и модулями ядра ОС

18.1 Введение

Из этого модуля вы узнаете, что такое служба systemd-udevd и как с помощью ее правил заставить работать 4G-модем, который упорно притворяется, что он USB-флешка. Мы посмотрим внимательнее, что такое модули ядра, и научимся не только загружать драйверы, но и намертво блокировать работу отдельных устройств, а то вдруг заклеить камеру стикером нам покажется недостаточной мерой.

18.2 Управление устройствами

В операционной системе Linux наиболее полная информация об устройствах доступна в псевдофайловой системе sysfs, которая монтируется в каталог /sys. Что такое псевдофайловые системы и как они работают мы уже рассматривали в модуле по управлению процессами, и, как вы помните, sysfs пришла на замену procfs, но вот порядка в ней на порядок больше.

По мере развития системы procfs ее файлы потеряли однообразие и используют теперь совершенно разные форматы, в то время как в системе sysfs одному параметру соответствует строго один файл, поэтому намусорить там значительно сложнее. Имя файла в sysfs является именем параметра, а содержимое файла — его значением, и в подавляющем большинстве случаев это действительно так.

18.2.1 Псевдофайловая система sysfs

В sysfs находятся следующие каталоги:

- Каталог `/sys/block` — тут содержатся символические ссылки на блочные устройства, найденные в системе, по одной ссылке на каждое устройство. Символические ссылки указывают на соответствующие каталоги в `/sys/devices`.
- Каталог `/sys/bus` — содержит каталоги для всех типов шин ядра. Один каталог на один тип шины. При этом каждый из этих каталогов содержит два подкаталога:
  - Каталог `/sys/bus/devices` — содержит символические ссылки на элементы в `/sys/devices`, соответствующие устройствам этого типа шины;

    - Каталог `/sys/bus/drivers` — содержит каталоги с драйверами устройств, подгруженных для работы с этой шиной. По одному каталогу на один драйвер.

- Каталог `/sys/class` — содержит структуру из подкаталогов, по одному на каждый класс устройств, зарегистрированных в системе. Внутри размещаются символические ссылки на элементы в `/sys/devices`, соответствующие этим устройствам.

- Каталог `/sys/dev` — содержит два подкаталога `block` и `char` для блочных и символьных устройств соответственно. Внутри этих подкаталогов находятся символические ссылки на каталоги в `/sys/devices`, соответствующие этим устройствам. Имена ссылок имеют вид X:Y, где X – это основной ID, а Y – дополнительный ID устройства.

- Каталог `/sys/devices` — содержит отражение структур device в ядре ОС в виде дерева каталогов и файлов.

- Каталог `/sys/digsig` — содержит данные, связанные с работой в замкнутой программной среде (ЗПИ).

- Каталог `/sys/firmware` — содержит интерфейсы для просмотра и изменение параметров прошивок, например, UEFI.

- Каталог `/sys/fs` — содержит подкаталоги для некоторых файловых систем, в том числе и cgroup. Этот каталог обычно используется как точка монтирования для tmpfs, в которой содержатся точки монтирования файловых систем контрольных групп.

- Каталог `/sys/hypervisor` — используется в случае работы гипервизора на компьютере.

- Каталог `/sys/kernel` — содержит интерфейсы для просмотра информации о работе ядра ОС.

- Каталог `/sys/module` — содержит подкаталоги для всех подгруженных модулей ядра, по одному на каждый модуль.

- Каталог `/sys/power` — содержит интерфейсы для просмотра и изменения параметров управления питанием и режимами энергосбережения.

18.2.2 Правила udev

Служба udev является диспетчером устройств ядра Linux, который обрабатывает все события пользовательского пространства, возникающие при добавлении/удалении аппаратных устройств и изменении их характеристик.

Обработчики событий службы udev называют также правилами, и с их помощью, например, служба управления сетевыми подключениями NetworkManager автоматически перенастраивает сетевые

интерфейсы при перемещении портативного устройства в другой сегмент сети. Вы можете создать собственные правила, с помощью которых можно выполнить переименование устройства или загрузку необходимых драйверов.

На рис. 1.109 представлена схема работы подсистемы udev, которая включает следующие шаги:

- Во время загрузки системы еще на самых ранних стадиях инициализации ядра создается псевдофайловая система **tmpfs**, которая монтируется в каталог `/dev`.
- В определенный момент стартует служба **systemd-udevd**, и в первую очередь она загружает в оперативную память базу данных правил (обработчиков), опираясь на файлы из каталогов `/usr/lib/udev/rules.d/` и `/etc/udev/rules.d/` (последние в приоритете).

  Файлы правил имеют суффикс `*.rules` и загружаются в алфавитном порядке, поэтому в названиях файлов первые два символа используют в качестве числа, которое задает приоритет.
- После загрузки правил служба отправляет ядру запрос на передачу всех событий **uevent**, возникших в системе еще до запуска **systemd-udevd**. Обмен информацией между ядром и пользовательским пространством осуществляется через сокет **netlink**.

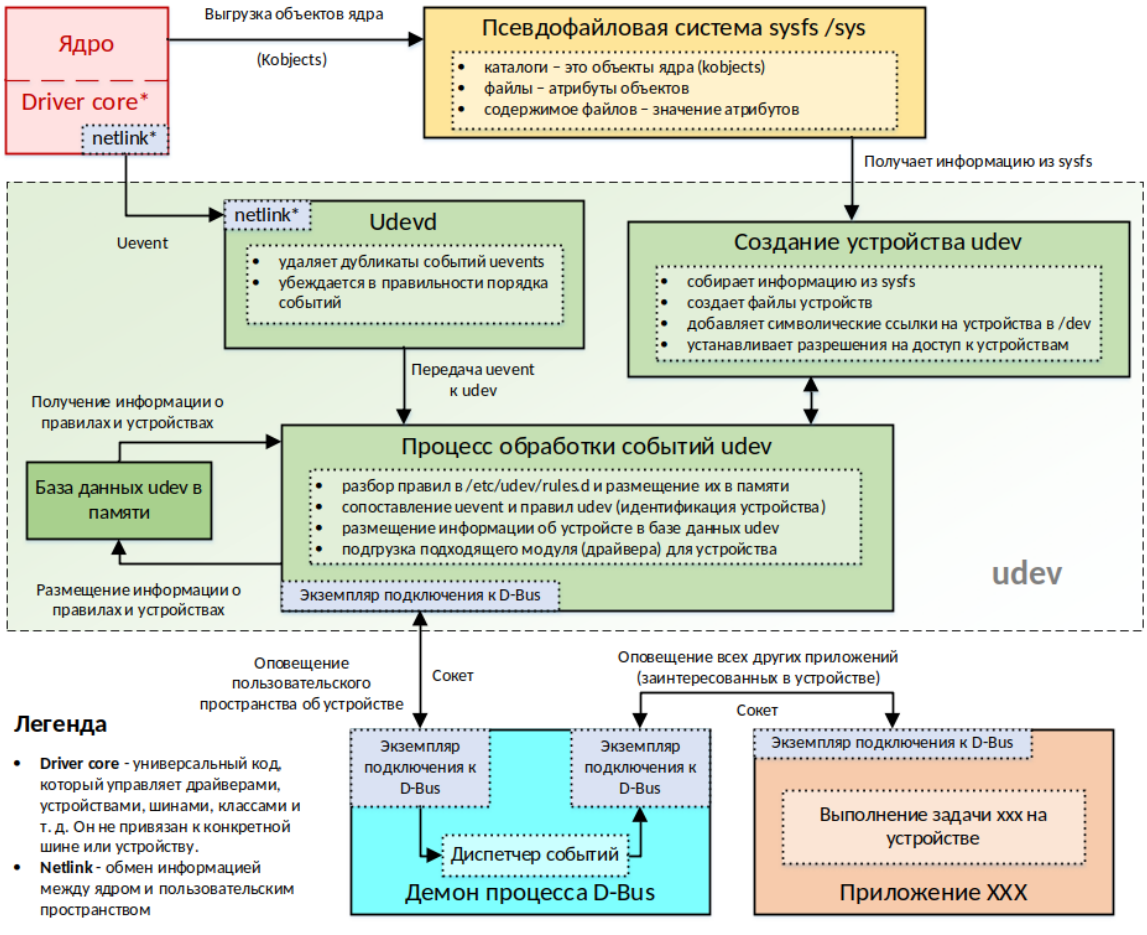


рис. 1.109 – Схема работы подсистемы udev

- При подключении/отключении периферийного устройства (например, usb-накопителя), ядро ОС производит обновление своих структур и отражает эти изменения в псевдофайловой системе **sysfs**. При этом происходит оповещение службы **systemd-udevd** о произошедшем событии (**uevent**).

- Служба **systemd-udevd** удаляет дубликаты событий uevent и обеспечивает их сортировку в правильном порядке, чтобы события о подключении родительских устройств (например, шины usb) обрабатывались раньше, чем события о подключении дочерних устройств (например, usb-накопителя).

- Далее в соответствии с информацией из базы **udev** проводит поиск и применение правил, соответствующих пришедшему событию. В правилах описаны действия (скрипты), которые должны быть выполнены.

  Кроме этого, udev собирает информацию об устройствах из **sysfs** и создает файлы устройств, выдает на них необходимые разрешения и добавляет символические ссылки в каталоге `/dev`. При необходимости udev отправляет ядру запросы на подгрузку необходимых модулей (драйверов устройств) и размещает информацию об устройствах в своей базе.

- На последнем этапе процесса служба **udev** через шину **D-Bus** оповещает о появлении нового устройства или изменении его физического состояния все приложения, заинтересованные в получении этой информации.

Примечание: Обработка событий в udev происходит параллельно, поэтому в файле `/etc/fstab` диски монтируются не по именам *sda* или *sdb*, а по их **UUID**, чтобы исключить ошибку, которая может возникнуть в связи с тем, что диски поменяются именами. Но в то же время с помощью правил udev имя устройства можно сделать постоянным.

18.2.2.1 Синтаксис правил

Правила **udev** хранятся в каталогах:

- Каталог `/usr/lib/udev/rules.d` — системные правила, созданные при установке системы или ПО.

- Каталог `/etc/udev/rules.d` — пользовательские правила, созданные администратором, более приоритетные (при совпадении имен файлов в этих двух каталогах).

Пример файла с правилами */usr/lib/udev/rules.d/50-udev-default.rules*:

```
# do not edit this file, it will be overwritten on update

# run a command on remove events
ACTION=="remove", ENV{REMOVE_CMD}!="", RUN+="$env{REMOVE_CMD}"
ACTION=="remove", GOTO="default_end"

SUBSYSTEM=="virtio-ports", KERNEL=="vport\*", ATTR{name}=="?\*", SYMLINK+="virtio-
↪ports/$attr{name }"

# select "system RTC" or just use the first one
SUBSYSTEM=="rtc", ATTR{hctosys}=="1", SYMLINK+="rtc"
SUBSYSTEM=="rtc", KERNEL=="rtc0", SYMLINK+="rtc", OPTIONS+="link_priority=-100"

...
```

Файлы правил имеют суффикс `*.rules` и состоят из множества строк, где каждая строка – это отдельное правило за исключением пустых строк с комментариями, начинающихся с символа решетки.

Каждое правило состоит из серии выражений `{ключ}{оператор}{значение}`, которые отделяются друг от друга запятыми. Выражения должны задавать как минимум один ключ соответствия и один ключ присвоения, где:

- Ключ соответствия — это условие, по которому определяется, соответствует ли конкретное устройство данному правилу или нет. Если ключей соответствия несколько, то правило применяется только в том случае, если устройство удовлетворяет сразу всем указанным критериям.
- Ключ присвоения — это действие, которое должно быть выполнено в системе, если текущее устройство удовлетворяет критериям ключей соответствия.

Ключи соответствия:

Ключи соответствия могут определять критерии как для самого устройства, так и для его родительских устройств. Например, для жесткого диска родителем будет устройство **SCSI**, а для него, в свою очередь, **шина BUS**).

Большинство ключей поддерживают шаблоны подстановки (*от англ. wildcard pattern*) в стиле оболочки: «*», «?», «[]», «|».

Основные ключи для критериев соответствия самого устройства:

- **KERNEL** — соответствие с именем устройства в ядре ОС;
- **SUBSYSTEM** — соответствие с подсистемой устройства;
- **DRIVER** — соответствие с именем драйвера, обслуживающего устройство;
- **DEVPATH** — соответствие пути до устройства в *sysfs*;

- **ATTR{атрибут}** — соответствие содержимому файла для этого атрибута в каталоге устройства *sysfs*;
- **ENV{переменная}** — соответствие значению переменной окружения.

Аналогичные ключи для определения соответствия как самому устройству, так и устройствам-предкам (родителям, дедушкам и т.д.):

- **KERNELS** — соответствие с именем устройства и/или его предка в ядре ОС;
- **SUBSYSTEMS** — соответствие с подсистемой устройства и/или его предка;
- **DRIVERS** — соответствие с именем драйвера, обслуживающего устройство и/или его предка;
- **ATTRS{атрибут}** — соответствие содержимому файла для этого атрибута в каталоге *sysfs* устройства и/или его предка.

С ключами соответствия используются два простых оператора:

- Оператор «==» — устройство соответствует критерию;
- Оператор «!=» — устройство не соответствует критерию.

Ключи присвоения (действий)

Ниже представлены основные ключи для действий, которые будут выполнены в случае, если правило применимо:

- **NAME** — устанавливает имя файла устройства в каталоге `/dev`;
- **SYMLINK** — создает символические ссылки, который действуют как альтернативные имена для этого устройства;
- **RUN** — задает список команд для запуска (это может быть имя скрипта с необходимыми операциями);
- **OWNER** — устанавливает владельца для файла устройства;
- **GROUP** — устанавливает группу-владельца для файла устройства;
- **MODE** — устанавливает права доступа на файл устройства.

Все вышеперечисленные ключи действий поддерживают шаблоны подстановки, подобные в команде `printf`:

- Шаблон `%k` — имя устройства, данное ему ядром,
- Шаблон `%n` — номер устройства (например, для sda2, номером является число 2),

- Шаблон `%p` — путь к файлу устройства,
- Шаблон `$name` — текущее имя устройства,
- Шаблон `$env{ключ}` — значение переменной окружения,
- Шаблон `$driver` — драйвер устройства.

С ключами присвоения используются следующие операторы:

- Оператор «=» — очистить список и добавить указанное действие;
- Оператор «+=» — добавить указанное действие без очистки списка;
- Оператор «-=» — удалить указанное действие из списка;
- Оператор «:=» — присвоить значение и запретить его изменение другими правилами.

18.2.2.2 Инструменты управления udev

Для управления udev предназначена утилита `udevadm`. Она управляет поведением **systemd-udevd** во время выполнения, запрашивает события ядра, управляет очередью событий и предоставляет простые механизмы отладки.

Основные опции утилиты udevadm:

- Команда `udevadm info <имя_устройства или путь_до_устройства>` — выводит информацию об устройстве из базы данных udev.
- Команда `udevadm test <путь_до_устройства_в_sysfs>` — имитирует запуск события udev для выбранного устройства и выводит отладочную информацию. Полезно для проверки работоспособности правил.
- Команда `udevadm monitor` — выводит события uevent в режиме реального времени.
- Команда `udevadm settle` — наблюдает за очередью событий udev. Если все события обработаны, через определенное время (по умолчанию 120 секунд) команда завершает свою работу.
- Команда `udevadm control` — управляет внутренним состоянием запущенного демона udev.

udevadm info

Для вывода информации об устройствах используется команда `udevadm info <имя_устройства/путь_до_устройства>`. Если необходимо вывести информацию и об устройствах-предках, то следует добавить параметр `-a` в команду `udevadm info -a <имя_устройства/путь_до_устройства>`.

Например, выведем информацию о диске `udevadm info /dev/sda`.

```
localadmin@astra:~$ udevadm info /dev/sda
P: /devices/pci0000:00/0000:00:0d.0/ata1/host0/target0:0:0/0:0:0:0/block/sda
N: sda
L: 0
S: disk/by-id/ata-VBOX_HARDDISK_VB6a0f90fb-c5f1dd19
S: disk/by-path/pci-0000:00:0d.0-ata-1
E: DEVPATH=/devices/pci0000:00/0000:00:0d.0/ata1/host0/target0:0:0/0:0:0:0/block/sda
E: DEVNAME=/dev/sda
...
```

udevadm test

Имитировать событие **udev** для выбранного устройства и вывести отладочную информацию можно командой `udevadm test <путь_до_устройства_в_sysfs>` (**DEVPATH**). Реальных изменений не происходит, полезно для проверки работоспособности правил.

```
localadmin@astra:~$ udevadm test /sys/block/sda
This program is for debugging only, it does not run any program
specified by a RUN key. It may show incorrect results, because
some values may be different, or not available at a simulation run.

Load module index
Network interface NamePolicy= disabled on kernel command line, ignoring.
Parsed configuration file /usr/lib/systemd/network/99-default.link
Created link configuration context.
Skipping empty file: /etc/udev/rules.d/10-usb-drive.rules
Reading rules file: /usr/lib/udev/rules.d/50-firmware.rules
Reading rules file: /usr/lib/udev/rules.d/50-udev-default.rules
. . .
Reading rules file: /usr/lib/udev/rules.d/99-systemd.rules
Rules contain 196608 bytes tokens (16384 \* 12 bytes), 26950 bytes strings
19410 strings (154764 bytes), 16958 de-duplicated (130267 bytes), 2453 trie nodes used
Invalid inotify descriptor.
Starting 'ata_id --export /dev/sda'
Process 'ata_id --export /dev/sda' failed with exit code 1.
Starting 'scsi_id --export --whitelisted -d /dev/sda'
'scsi_id --export --whitelisted -d /dev/sda'(out) 'ID_SCSI=1'
'scsi_id --export --whitelisted -d /dev/sda'(out) 'ID_VENDOR='
'scsi_id --export --whitelisted -d /dev/sda'(out) 'ID_VENDOR_ENC='
'scsi_id --export --whitelisted -d /dev/sda'(out) 'ID_MODEL='
'scsi_id --export --whitelisted -d /dev/sda'(out) 'ID_MODEL_ENC='
'scsi_id --export --whitelisted -d /dev/sda'(out) 'ID_REVISION='
'scsi_id --export --whitelisted -d /dev/sda'(out) 'ID_TYPE='
Process 'scsi_id --export --whitelisted -d /dev/sda' succeeded.
PDAC: PDPL is zero
PDAC: AUDIT is empty or invalid
DEVPATH=/devices/pci0000:00/0000:00:0d.0/ata1/host0/target0:0:0/0:0:0:0/block/sda
DEVNAME=/dev/sda
DEVTYPE=disk
DISKSEQ=9
MAJOR=8
MINOR=0
ACTION=add
```

(continues on next page)

(продолжение с предыдущей страницы)

```
SUBSYSTEM=block
ID_SCSI=1
ID_BUS=scsi
ID_PATH=pci-0000:00:0d.0-ata-1
ID_PATH_TAG=pci-0000_00_0d_0-ata-1
DEVLINKS=/dev/disk/by-path/pci-0000:00:0d.0-ata-1
TAGS=:systemd:
USEC_INITIALIZED=3952844
Unload module index
Unloaded link configuration context.
```

udevadm monitor

Команда `udevadm monitor` используется для вывода событий **uevent** в режиме реального времени. Если запустить утилиту и после этого вставить оптический диск в виртуальную машину, то мы увидим соответствующее событие.

```
localadmin@astra:~$ udevadm monitor
monitor will print the received events for:
UDEV - the event which udev sends out after rule processing
KERNEL - the kernel uevent

KERNEL[237008.787520] change  /devices/pci0000:00/0000:00:01.1/ata3/host2/target2:0:0/
↪2:0:0:0/b lock/sr0 (block)
UDEV [237008.836023]  change  /devices/pci0000:00/0000:00:01.1/ata3/host2/target2:0:0/
↪2:0:0:0/b lock/sr0 (block)
```

udevadm control

Для управления внутренним состоянием запущенного демона **udev** и диагностики его работы используются команды `sudo udevadm control [параметры]`.

- Команда `sudo udevadm control --ping` — проверяет, запущена ли служба **systemd-udevd** и отвечает ли она на запросы.
- Команда `sudo udevadm control --reload` — заставляет службу перечитать правила **udev** и других баз данных (например, индекс модуля ядра). Изменения не повлияют на уже существующие устройства. Новые правила будут применяться только к новым событиям **uevent**.
- Команда `sudo udevadm control --stop-exec-queue` — приостановит обработку новых событий службой **systemd-udevd**. События будут помещаться в очередь до востребования.
- Команда `sudo udevadm control --start-exec-queue` — возобновит обработку событий службой **systemd-udevd** после выполнения предыдущей команды `stop-exec-queue`.
- Команда `sudo udevadm control -log-priority=<уровень_журналирования>` — установит уровень журналирования службы **systemd-udevd**. Может принимать одно из следующих значений: **emerg, alert, crit, err, warning, notice, info, debug**.

Проверим работу команд на следующем примере:

- остановим обработку сообщений **uevent** командой `sudo udevadm control --stop-exec-queue`;
- Далее необходимо переподключить оптический диск;
- запустим `udevadm settle` в отдельном терминале и убедимся, что команда не завершила свою работу мгновенно, так как в очереди есть необработанные сообщения;
- возобновим обработку сообщений `sudo udevadm control --start-exec-queue` и
- убедимся, что команда `udevadm settle` завершила свою работу, т.к. все сообщения из очереди были обработаны.

18.2.2.3 Создание правила udev

Создадим правило **udev** для usb-накопителя, которое позволит логировать серийный номер и время подключения устройства к компьютеру.

Во-первых, создадим сам файл для правила в каталоге `/etc/udev/rules.d/` с помощью команды:

```
sudo touch /etc/udev/rules.d/99-usb-drive.rules
```

Во-вторых, создадим в текстовом редакторе скрипт `/usr/bin/usb_disk_log.sh`, который будет записывать необходимую нам информацию в файл.

```
#!/bin/bash
/bin/echo "В $(/bin/date) был вставлен usb-диск. Серийный номер: $1" >> /var/log/usb_
↪disk.log
```

Разрешим выполнение файла `usb_disk_log.sh` командой

```
localadmin@astra:~$ sudo chmod 744 /usr/bin/usb_disk_log.sh
```

Проверим работоспособность скрипта командой `sudo /usr/bin/usb_disk_log.sh 1234` — в файле `/var/log/usb_disk.log` должна появиться соответствующая строка.

```
localadmin@astra:~$ sudo /usr/bin/usb_disk_log.sh 1234
localadmin@astra:~$ cat /var/log/usb_disk.log
В Чт авг 24 11:28:46 MSK 2023 был вставлен usb-диск. Серийный номер: 1234
```

Теперь посмотрим параметры нашего usb-накопителя. Для этого подключим его к хостовой и гостевой машине и выведем информацию о пути до этого устройства `udevadm info /dev/sdb1 | grep DEVPATH | awk -F '=' '{print $2}`.

```
localadmin@astra:~$ udevadm info /dev/sdb1 | grep DEVPATH | awk -F '=' '{print $2}'
/devices/pci0000:00/0000:00:0b.0/usb1/1-1/1-1:1.0/host3/target3:0:0/3:0:0:0/block/sdb/
↪sdb1
```

И информацию о его атрибутах:

```
localadmin@astra:~$ udevadm info --attribute-walk /dev/sdb1 | grep -P '(usb|block)'
 looking at device '/devices/pci0000:00/0000:00:0b.0/usb1/1-1/1-1:1.0/host3/
↪target3:0:0/3:0:0:0 /block/sdb/sdb1':
   SUBSYSTEM=="block"
 looking at parent device '/devices/pci0000:00/0000:00:0b.0/usb1/1-1/1-1:1.0/host3/
↪target3:0:0/ 3:0:0:0/block/sdb':
   SUBSYSTEMS=="block"
 looking at parent device '/devices/pci0000:00/0000:00:0b.0/usb1/1-1/1-1:1.0/host3/
↪target3:0:0/ 3:0:0:0':
   ATTRS{device_blocked}=="0"
 looking at parent device '/devices/pci0000:00/0000:00:0b.0/usb1/1-1/1-1:1.0/host3/
↪target3:0:0' :
 looking at parent device '/devices/pci0000:00/0000:00:0b.0/usb1/1-1/1-1:1.0/host3':
 looking at parent device '/devices/pci0000:00/0000:00:0b.0/usb1/1-1/1-1:1.0':
   SUBSYSTEMS=="usb"
   DRIVERS=="usb-storage"
 looking at parent device '/devices/pci0000:00/0000:00:0b.0/usb1/1-1':
   SUBSYSTEMS=="usb"
   DRIVERS=="usb"
 looking at parent device '/devices/pci0000:00/0000:00:0b.0/usb1':
   KERNELS=="usb1"
   SUBSYSTEMS=="usb"
   DRIVERS=="usb"
```

В созданный нами файл правила `/etc/udev/rules.d/99-usb-drive.rules` добавим следующий текст:

```
SUBSYSTEM=="block", DRIVERS=="usb-storage", RUN+="/usr/bin/usb_disk_log.sh $env{ID_
↪SERIAL}"
```

Протестируем наше правило:

```
localadmin@astra:~$ udevadm test `udevadm info /dev/sdb1 | grep DEVPATH | awk -F '=' '
↪{print $2}'`
Load module index
Network interface NamePolicy= disabled on kernel command line, ignoring.
Parsed configuration file /usr/lib/systemd/network/99-default.link
Created link configuration context.
Reading rules file: /usr/lib/udev/rules.d/50-firmware.rules
Reading rules file: /usr/lib/udev/rules.d/50-udev-default.rules
Reading rules file: /usr/lib/udev/rules.d/55-dm.rules
. . .
USEC_INITIALIZED=46104098
run: '/bin/ln -f /dev/sdb1 /dev/sdb1_vfat'
run: '/usr/bin/usb_disk_log.sh Kingston_DataTraveler_2.0_5B791A91D481-0:0'
```

Теперь перезагрузим правила командой `sudo udevadm control --reload` и повторно подключим «флешку» к нашей гостевой машине. В журнале `usb_disk.log` появится новая информация:

```
localadmin@astra:~$ cat /var/log/usb_disk.log
В Чт авг 24 11:28:46 MSK 2023 был вставлен usb-диск. Серийный номер: 1234
В Thu Aug 24 12:34:46 MSK 2023 был вставлен usb-диск. Серийный номер: Kingston_
↪DataTraveler_2.0_
5B791A91D481-0:0
```

Еще несколько практических примеров:

Некоторые модели USB-модемов могут по ошибке определяться как USB-накопители. Чтобы исправить эту проблему, можно создать правило udev и настроить его на выполнение команды для смены режима работы модема. Для этого нужно:

a. Создать файл правила `/etc/udev/rules.d/10-modem_mode_switch.rules` с содержимым:

```
ACTION=="add",SUBSYSTEM=="usb",ATTRS{idVendor}=="12d1",ATTRS{idProduct}=="1f01",RUN+=
↪"/usr/sbin/usb_modeswitch --quiet --config-file /usr/share/usb_modeswitch/12d1:1f01"
```

b. Создать конфигурационный файл `/usr/share/usb_modeswitch/12d1:1f01` с содержимым:

```
DefaultVendor = 0x12d1
DefaultProduct = 0x1f01
TargetVendor = 0x12d1
TargetProduct = 0x14dc
# switch to 12d1:14dc (default HiLink CDC-Ether mode)
MessageContent="55534243123456780000000000000a11062000000000000100000000
000000"
```

После выполнения указанных действий при подключении устройства будет выполняться требуемая команда, и он будет работать как USB-модем.

Когда на сервере несколько сетевых карт, с помощью udev правил сетевым интерфейсам можно присвоить собственные имена, чтобы было проще ориентироваться. Например:

a. Создадим файл `/etc/udev/rules.d/10-interface_rename.rules` с содержимым:

```
SUBSYSTEM=="net", ACTION=="add", ATTR{address}=="08:00:27:48:15:07", NAME="localnet"
```

Это правило переименует сетевой интерфейс с MAC-адресом 08:00:27:48:15:07 в localnet, и вы будете точно знать, что интерфейс «смотрит» в локальную сеть.

Протестируем созданное нами правило. Для этого сначала найдем сетевое устройство в каталоге /sys

```
localadmin@astra:~$ find /sys -name *eth*
...
/sys/devices/pci0000:00/0000:00:03.0/net/eth0
...
```

Затем запустим тест и убедимся, что правило отработает:

```
localadmin@astra:~$ sudo udevadm test /sys/devices/pci0000:00/0000:00:03.0/net/eth0
...
SYSTEMD_ALIAS=/sys/subsystem/net/devices/localnet
...
```

Теперь можно либо перезаргузить машину, либо дать такой набор команд:

```
localadmin@astra:~$ sudo ip link set eth0 down
localadmin@astra:~$ sudo udevadm control --reload-rules
localadmin@astra:~$ sudo udevadm trigger
localadmin@astra:~$ sudo ip link set localnet up
```

Проверяем:

```
localadmin@astra:~$ sudo ip link show
```

b. По тому же принципу можно создать правило для интерфейса в Интернет. Не забудьте только изменить сетевые настройки в соответствии с новыми именами интерфейсов.

18.2.3 Просмотр информации об устройствах

Для просмотра информации об устройствах кроме утилиты udevadm и псевдофайловой системы sysfs существуют и другие способы.

18.2.3.1 Графический менеджер устройств

Менеджер устройств (fly-admin-device-manager) позволяет просматривать и управлять устройствами, которые зарегистрированы в системе, так же удобно, как мы это могли делать в диспетчере устройств Windows. Запустить его можно через меню пуск *Пуск → Системные → Менеджер устройств*.

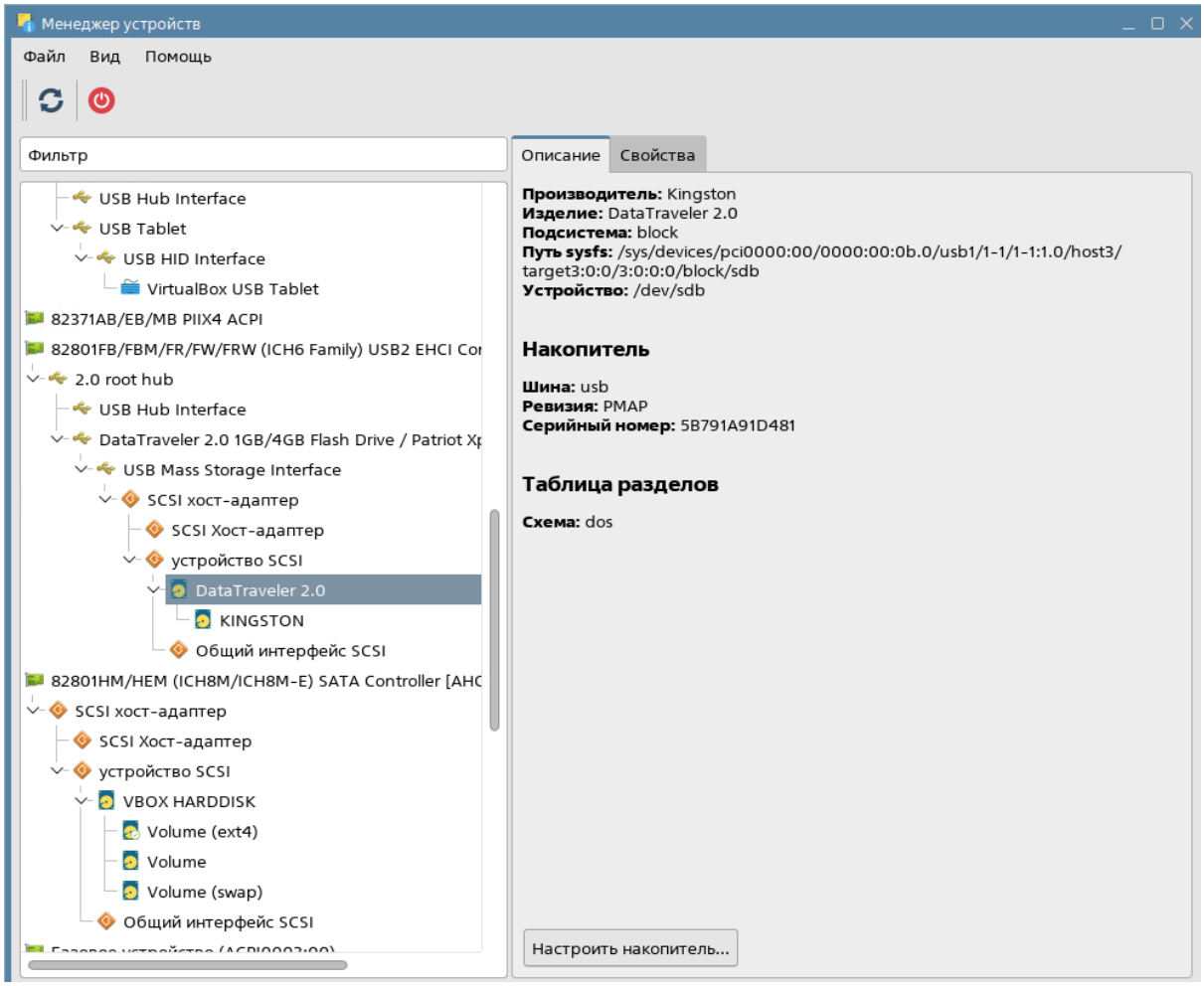


рис. 1.110 – Графический менеджер устройств

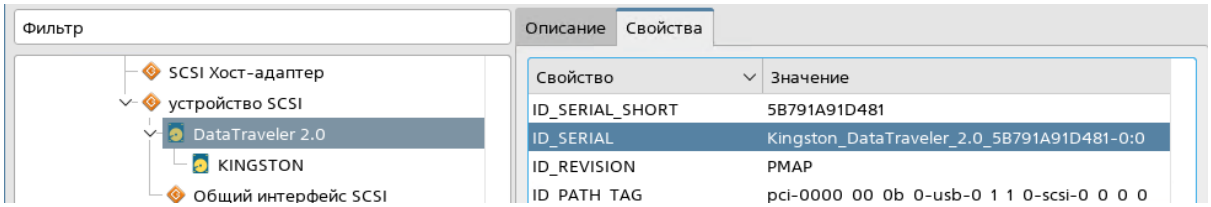


рис. 1.111 – Свойства выбранного устройства

18.2.3.2 Утилиты для просмотра информации об устройствах

Для командной строки тоже есть ряд удобных утилит.

Утилита `lspci` выводит информацию об устройствах, подключенных к шине PCI.

```
localadmin@astra:~$ lspci
00:00.0 Host bridge: Intel Corporation 440FX - 82441FX PMC [Natoma] (rev 02)
00:01.0 ISA bridge: Intel Corporation 82371SB PIIX3 ISA [Natoma/Triton II]
00:01.1 IDE interface: Intel Corporation 82371AB/EB/MB PIIX4 IDE (rev 01)
00:02.0 VGA compatible controller: VMware SVGA II Adapter
. . .
```

Утилита lsusb выводит информацию об устройствах, подключенных к шине USB.

```
localadmin@astra:~$ lsusb
Bus 004 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 003 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 001 Device 004: ID 13fe:1d00 Kingston Technology Company Inc. DataTraveler 2.0␣
↪1GB/4GB Flash
Drive / Patriot Xporter 4GB Flash Drive
. . .
```

Утилита lscpu выводит информацию о процессорах.

```
localadmin@astra:~$ lscpu
Архитектура: x86_64
CPU op-mode(s): 32-bit, 64-bit
Порядок байт: Little Endian
Address sizes: 46 bits physical, 48 bits virtual
CPU(s): 2
. . .
```

Утилита lshw выводит информацию о всех устройствах в системе. Доступна в расширенном репозитории для установки.

```
localadmin@astra:~$ sudo apt install lshw
Чтение списков пакетов... Готово
Построение дерева зависимостей
. . .
Обрабатываются триггеры для man-db (2.8.5-2) ...

localadmin@astra:~$ sudo lshw -short
H/W path              Device      Class        Description
============================================================
                                  system       VirtualBox
/0                                bus          VirtualBox
/0/0                              memory       128KiB BIOS
/0/1                              memory       2GiB System memory
/0/2                              processor    Intel(R) Xeon(R) CPU E5-2680 v4 @ 2.
↪40GHz
/0/100                            bridge       440FX - 82441FX PMC [Natoma]
/0/100/1                          bridge       82371SB PIIX3 ISA [Natoma/Triton II]
...
```

18.3 Модули ядра операционной системы

Модули ядра — это объектный файл, который содержит код, расширяющий функциональность базового ядра операционной системы. Модули бывают встроены в ядро или могут загружаться во время работы операционной системы. Самый простой пример модуля — это драйверы различных устройств.

18.3.1 Размещение модулей ядра

Все модули для установленных ядер хранятся директории `/lib/modules/`.

```
localadmin@astra:~$ ls /lib/modules
6.1.90-1-generic
```

Для каждой версии ядра создается свой подкаталог с необходимыми для него модулями. Содержимое каталога для текущий версии ядра можем посмотреть командой:

```
localadmin@astra:~$ ls /lib/modules/$(uname -r)
build   kernel  modules.alias      modules.builtin            modules.builtin.bin    ␣
↪ modules.dep      modules.devname  modules.softdep  modules.symbols.bin
initrd  misc    modules.alias.bin  modules.builtin.alias.bin  modules.builtin.modinfo␣
↪ modules.dep.bin  modules.order    modules.symbols  vdso
```

У файлов модулей расширение `*.ko`. Посмотреть все модули для текущей версии можно командой:

```
localadmin@astra:~$ find /lib/modules/$(uname -r) -name *.ko | head
/lib/modules/5.15.0-70-generic/misc/vboxguest.ko
/lib/modules/5.15.0-70-generic/misc/digsig_verif.ko
/lib/modules/5.15.0-70-generic/misc/parsec.ko
/lib/modules/5.15.0-70-generic/misc/vboxvideo.ko
/lib/modules/5.15.0-70-generic/misc/vboxsf.ko
/lib/modules/5.15.0-70-generic/misc/parsec-cifs.ko
/lib/modules/5.15.0-70-generic/kernel/net/nfc/nci/nci_spi.ko
/lib/modules/5.15.0-70-generic/kernel/net/nfc/nci/nci_uart.ko
/lib/modules/5.15.0-70-generic/kernel/net/nfc/nci/nci.ko
/lib/modules/5.15.0-70-generic/kernel/net/nfc/hci/hci.ko
```

При изменении состава модулей выполняется команда `depmod`, которая пересоздает базу данных установленных модулей и их зависимостей. Файл с базой можно посмотреть командой:

```
localadmin@astra:~$ cat /lib/modules/$(uname -r)/modules.dep | head
kernel/arch/x86/events/amd/amd-uncore.ko:
kernel/arch/x86/events/intel/intel-cstate.ko:
kernel/arch/x86/events/rapl.ko:
kernel/arch/x86/kernel/cpu/mce/mce-inject.ko:
kernel/arch/x86/kernel/msr.ko:
kernel/arch/x86/kernel/cpuid.ko:
kernel/arch/x86/crypto/twofish-x86_64.ko: kernel/crypto/twofish_common.ko
kernel/arch/x86/crypto/twofish-x86_64-3way.ko: kernel/arch/x86/crypto/twofish-x86_64.
↪ko
kernel/arch/x86/crypto/twofish-avx-x86_64.ko: kernel/crypto/crypto_simd.ko kernel/
↪crypto/cryptd.
ko kernel/arch/x86/crypto/twofish-x86_64-3way.ko kernel/arch/x86/crypto/twofish-x86_
↪64.ko kernel
/crypto/twofish_common.ko
...
```

18.3.2 Просмотр модулей ядра

Для просмотра модулей ядра, загруженных в данный момент, служит команда `lsmod`.

```
localadmin@astra:~$ lsmod
Module                  Size  Used by
binfmt_misc            24576  1
udp_diag               16384  0
tcp_diag               16384  0
inet_diag              24576  2 tcp_diag,udp_diag
. . .
psmouse               176128  0
hid                   147456  2 usbhid,hid_generic
usb_storage            77824  1 uas
e1000                 151552  0
i2c_piix4              28672  0
drm                   618496  8 vmwgfx,drm_kms_helper,vboxvideo,drm_ttm_helper,ttm
pata_acpi              16384  0
video                  61440  0
parsec                253952  0
digsig_verif          507904  1 parsec
```

Она выводит таблицу, в которой три столбца:

- **Module** – название модуля.
- **Size** – размер модуля в байтах.
- **Used by** – показывает количество экземпляров модуля, используемое в настоящее время, и список зависимостей, указываются через запятую. Значение ноль говорит о том, что модуль не используется.

Для вывода подробной информации о загруженном модуле, используется команда:

`modinfo <имя_модуля>`

Выведем информацию о модуле, который реализует драйвер сетевого адаптера e1000 с помощью команды:

```
localadmin@astra:~$ sudo modinfo e1000
[sudo] пароль для localadmin:
filename:       /lib/modules/5.15.0-70-generic/kernel/drivers/net/ethernet/intel/
↪e1000/e10
00.ko
license:        GPL v2
description:    Intel(R) PRO/1000 Network Driver
author:         Intel Corporation, <linux.nics@intel.com>
srcversion:     B4D2D3AB3E4A089C43DAE56
alias:          pci:v00008086d00002E6Esv*sd*bc*sc*i*
. . .
alias:          pci:v00008086d00001000sv*sd*bc*sc*i*
depends:
```

(continues on next page)

(продолжение с предыдущей страницы)

```
retpoline:      Y
intree:         Y
name:           e1000
vermagic:       5.15.0-70-generic SMP mod_unload modversions
parm:           TxDescriptors:Number of transmit descriptors (array of int)
parm:           RxDescriptors:Number of receive descriptors (array of int)
parm:           Speed:Speed setting (array of int)
parm:           Duplex:Duplex setting (array of int)
parm:           AutoNeg:Advertised auto-negotiation setting (array of int)
parm:           FlowControl:Flow Control setting (array of int)
parm:           XsumRX:Disable or enable Receive Checksum offload (array of int)
parm:           TxIntDelay:Transmit Interrupt Delay (array of int)
parm:           TxAbsIntDelay:Transmit Absolute Interrupt Delay (array of int)
parm:           RxIntDelay:Receive Interrupt Delay (array of int)
parm:           RxAbsIntDelay:Receive Absolute Interrupt Delay (array of int)
parm:           InterruptThrottleRate:Interrupt Throttling Rate (array of int)
parm:           SmartPowerDownEnable:Enable PHY smart power down (array of int)
parm:           copybreak:Maximum size of packet that is copied to a new buffer on␣
↪receive (uint)
parm: debug:Debug level (0=none,...,16=all) (int)
```

18.3.3 Управление модулями ядра

Для управления модулями ядра есть несколько инструментов:

- команды для загрузки и выгрузки модулей ядра;
- конфигурационные файлы загрузки модулей;
- конфигурационные файлы для автозагрузки модулей;
- конфигурационные файлы для запрета загрузки модулей (черные списки);
- конфигурационные файлы для задания параметров модулей.

Конфигурационные файлы находятся в следующих каталогах:

- Каталоги `/etc/modules-load.d/` и `/usr/lib/modules-load.d/` – каталоги с конфигурационными файлами *.conf для автозагрузки модулей при старте системы;
- Каталоги `/etc/modprobe.d/` и `/lib/modprobe.d/` – каталоги с конфигурационными файлами `*.conf` для блокировки модулей, создания псевдонимов и настройки параметров модулей.

18.3.3.1 Загрузка и выгрузка модулей ядра

Загрузка модулей ядра осуществляется командой:

`modprobe [параметры] <имя_модуля>` или `insmod <полный_путь_до_файла_модуля>`.

Выгрузка модулей осуществляется командой:

`modprobe -r <имя_модуля>` или `rmmode <имя_модуля>`.

Выгрузим модуль e1000 и загрузим его заново:

```
localadmin@astra:~$ sudo modprobe -r e1000 && sudo modprobe e1000
```

При выполнении команды вы могли заметить, что сетевое соединение прерывалось.

У команды `modprobe` кроме ключа `-r` есть еще много параметров. С полным списком вы можете ознакомиться в справке `man modprobe`. Рассмотрим только пару наиболее часто используемых:

- Ключ `--show-depends` – вывод зависимостей модуля (в том числе и файл самого модуля), например:

```
localadmin@astra:~$ sudo modprobe --show-depends ttm
insmod /lib/modules/5.15.0-70-generic/kernel/drivers/gpu/drm/drm.ko
insmod /lib/modules/5.15.0-70-generic/kernel/drivers/gpu/drm/ttm/ttm.ko
```

- Ключ `--showconfig` – вывод эффективных настроек модулей, например:

```
localadmin@astra:~$ sudo modprobe --showconfig | wc
   45962 137729 2162220
localadmin@astra:~$ sudo modprobe --showconfig | head
blacklist ath_pci
blacklist ohci1394
blacklist sbp2
blacklist dv1394
blacklist raw1394
blacklist video1394
blacklist aty128fb
blacklist atyfb
blacklist radeonfb
blacklist cirrusfb
localadmin@astra:~$ sudo modprobe --showconfig | tail
alias symbol:zgid ib_core
alias symbol:zl10036_attach zl10036
alias symbol:zl10039_attach zl10039
alias symbol:zl10353_attach zl10353
alias symbol:zpa2326_isreg_precious zpa2326
alias symbol:zpa2326_isreg_readable zpa2326
alias symbol:zpa2326_isreg_writeable zpa2326
alias symbol:zpa2326_pm_ops zpa2326
alias symbol:zpa2326_probe zpa2326
alias symbol:zpa2326_remove zpa2326
```

18.3.3.2 Автозагрузка модулей ядра

Для автозагрузки модулей ядра службой udev при старте системы их имена достаточно добавить в один из файлов с суффиксом .conf в каталоге /etc/modules-load.d/.

```
localadmin@astra:~$ ls -l /etc/modules-load.d/
итого 8
-rw-r--r-- 1 root root 123 авг 15 11:03 cups-filters.conf
lrwxrwxrwx 1 root root  10 апр  4 18:15 modules.conf -> ../modules
-rw-r--r-- 1 root root  30 мая  5  2022 usb-over-ip-load.conf
```

Пустые строки и комментарии игнорируются `/etc/modules-load.d/modules.conf`:

```
# /etc/modules: kernel modules to load at boot time.
#
# This file contains the names of kernel modules that should be loaded
# at boot time, one per line. Lines beginning with "#" are ignored.

localadmin@astra:~$ cat /etc/modules-load.d/cups-filters.conf
# Parallel printer driver modules loading for cups
# LOAD_LP_MODULE was not 'yes' in /etc/default/cups
lp
ppdev
parport_pc
```

Кроме этой директории существует еще директория `/usr/lib/modules-load.d/`, конфигурационные файлы которой тоже учитываются.

Модули, внесенные в эти конфигурационные файлы, будут автоматически загружены при следующем старте ОС.

18.3.3.3 Черный список модулей

Если требуется заблокировать работу устройства или избежать конфликтов между модулями, какие-то из них можно внести в так называемый черный список.

Черные списки находятся в файлах *.conf в следующих каталогах:

- Каталог `/lib/modprobe.d/` – содержит настройки, заданные при установке ОС или внесенные при установке ПО;
- Каталог `/etc/modprobe.d/` – в эту папку можно сохранять свои собственные настройки, которые потребовались вам в процессе эксплуатации системы.

```
localadmin@astra:~$ ls /lib/modprobe.d/
aliases.conf                            fbdev-blacklist.conf
blacklist_linux_5.15.0-33-generic.conf  systemd.conf
blacklist_linux_5.15.0-70-generic.conf

localadmin@astra:~$ ls /etc/modprobe.d/
blacklist-astra.conf    blacklist-firewire.conf      blacklist-rare-network.conf
blacklist-ath_pci.conf  blacklist-framebuffer.conf   iwlwifi.conf
blacklist.conf          blacklist.local.conf         parsec-cifs.conf

localadmin@astra:~$ cat /lib/modprobe.d/blacklist_linux_5.15.0-70-generic.conf | head
```

(continues on next page)

(продолжение с предыдущей страницы)

```
# Kernel supplied blacklist for linux 5.15.0-70-generic amd64
# modprobe.d/common.conf
# LP:1434842 -- disable OSS drivers by default to allow pulseaudio to emulate
blacklist snd-mixer-oss
blacklist snd-pcm-oss
# Autogenerated watchdog blacklist
blacklist acquirewdt
blacklist advantechwdt
blacklist alim1535_wdt
blacklist alim7101_wdt
```

Модули могут быть загружены как службой udev, так и на более раннем этапе из initramfs диска.

Для того чтобы запретить автозагрузку модуля, достаточно в каталоге `/etc/modprobe.d/` в файле с суффиксом `*.conf` создать строку, которая начиналась бы с директивы blacklist, за которой следовало бы имя модуля. Однако для соблюдения соглашения об именовании крайне желательно, чтобы имя такого файла начиналось со слова blacklist.

Для запрета автозагрузки модуля, загружаемого из initramfs, необходимо пересоздать initramfs после внесения изменений в конфигурационные файлы. Это можно сделать командой `sudo update-initramfs -u`.

Запрет, созданный с помощью директивы blacklist, запретит автозагрузку модуля. Однако модуль может быть все равно загружен, если от него зависят другие модули. Для полного запрета загрузки модуля в дополнение к директиве blacklist необходимо добавить команду `install <имя_модуля> /bin/true`.

Загрузку модулей можно запретить также с помощью параметров GRUB. Это может быть полезно, если один из модулей препятствует нормальной загрузке системы. Для этого необходимо добавить текст вида *module_blacklist=<имя_модуля1>,<имя_модуля2>* в строку с параметрами загрузки ядра.

Создадим файл blacklist-network.conf в /etc/modprobe.d/, добавим в него строку blacklist e1000 и перезагрузим систему.

```
localadmin@astra:~$ sudo vim /etc/modprobe.d/blacklist-network.conf
localadmin@astra:~$ sudo cat /etc/modprobe.d/blacklist-network.conf
blacklist e1000
localadmin@astra:~$ sudo reboot
```

Казалось бы, при перезагрузке системы драйвер сетевого адаптера не будет загружен. Выполним перезагрузку и убедимся в том, что драйвер по-прежнему загружается.

```
localadmin@astra:~$ lsmod | grep e1000
e1000                 151552  0
```

Дело в том, что этот драйвер загружается из initramfs. Выполним обновление образа Initrd, пере-

загрузим систему и убедимся, что теперь сеть нам не доступна:

```
localadmin@astra:~$ sudo update-initramfs -u
update-initramfs: Generating /boot/initrd.img-5.15.0-70-generic
localadmin@astra:~$ sudo reboot
```

После перезагрузки:

```
localadmin@astra:~$ lsmod | grep e1000
localadmin@astra:~$
```

Однако мы все еще можем загрузить этот модуль вручную командой:

```
localadmin@astra:~$ sudo modprobe e1000
localadmin@astra:~$ lsmod | grep e1000
e1000                  151552
```

При внесении дополнительной строки *install e1000 /bin/true* в конфигурационный файл `/etc/modprobe.d/blacklist-network.conf` мы полностью запретим загрузку этого модуля.

```
blacklist e1000
install e1000 /bin/true
```

Примечание: Строка *install e1000 /bin/true* указывает, что вместо загрузки модуля *e1000* надо выполнить команду */bin/true*, которая всегда завершает своё выполнение с кодом успеха (0). Таким образом, при попытке загрузить модуль *e1000* не возникает никаких ошибок — система считает, что модуль загружен успешно, хотя на самом деле он загружен не был.

Создайте файл `blacklist-network.conf` с содержимым, описанным выше, и перезагрузитесь:

```
localadmin@astra:~$ sudo vim /etc/modprobe.d/blacklist-network.conf
localadmin@astra:~$ sudo reboot
```

После перезагрузки проверим:

```
localadmin@astra:~$ sudo modprobe e1000
localadmin@astra:~$ lsmod | grep e1000
localadmin@astra:~$
```

Для восстановления работоспособности сети, удалим созданный нами файл черного списка `/etc/modprobe.d/blacklist-network.conf`, пересоздадим образ Initrd и перезагрузим систему.

```
localadmin@astra:~$ sudo rm /etc/modprobe.d/blacklist-network.conf
localadmin@astra:~$ sudo update-initramfs -u
update-initramfs: Generating /boot/initrd.img-5.15.0-70-generic
localadmin@astra:~$ sudo reboot
```

Как мы можем убедиться, сетевой адаптер снова работает, и сеть стала опять доступна:

```
localadmin@astra:~$ lsmod | grep e1000
e1000                  151552 0
```

18.3.3.4 Псевдонимы модулей

Псевдонимы модулей – это альтернативные имена модулей, которые могут быть созданы для удобства (если имя модуля, например, слишком длинное) или для нужд приложений.

Псевдонимы, как и черные списки, хранятся в файлах с суффиксом .conf в следующих каталогах:

- Каталог `/lib/modprobe.d/` - содержит настройки, заданные при установке ОС или внесенные при установке ПО.
- Каталог `/etc/modprobe.d/` - в эту папку можно сохранять свои собственные настройки, которые потребовались вам в процессе эксплуатации системы.

Однако в этом случае не нужно использовать слово blacklist в названии файлов.

Чтобы создать псевдоним, необходимо в файл с суффиксом `*.conf` (например, `aliases.conf` или `aliases-<что-то>.conf`) добавить строку вида:

`alias <псевдоним_модуля> <имя_модуля>`

В именах алиасов вам доступны также символы подстановки. Например, задав псевдоним через команду `alias my-mod* module_name`, к модулю можно будет обращаться через имена my-mod-1, my-mod-something и так далее.

18.3.3.5 Параметры модулей

Параметры модулей можно задавать как с помощью конфигурационных файлов, так и «на лету» с помощью утилиты `modprobe`:

- Параметры модулей, как и псевдонимы, задаются в файлах `*.conf` в каталогах `/lib/modprobe.d/` и `/etc/modprobe.d/`. Для этого необходимо добавить в файл строку вида:

  `options <имя_модуля> <имя_параметра>=<значение параметра>`

- Для управления параметрами модулей при работе системы можно воспользоваться командой:

  `modprobe <имя_модуля> <имя_параметра>=<значение параметра>`.

18.4 Практика и тестирование

18.4.1 Практическая работа: Модуль 18. Управление устройствами и модулями ядра ОС

См. также *Требования, правила и цели выполнения практической работы*

18.4.1.1 Практические задания

Задание 1.

1) По умолчанию группой файла /dev/tty8 является группа tty. Поменяйте группу на группу root для /dev/tty8.

2) После того как группа будет изменена, отмените свои изменения.

Задание 2.

1) Определите путь к сетевому устройству eth0 в каталоге /sys.

2) Выведите все атрибуты устройства eth0.

Задание 3.

1) Выгрузите модуль intel_rapl_msr.

2) Добавьте модуль intel_rapl_msr в черный список. Перезагрузите систему и убедитесь, что он не загружен.

3) Отмените свои изменения.

18.4.1.2 Ответы на практические задания (пошаговые инструкции)

Задание 1.

1) Создайте файл /etc/udev/rules.d/10-tty8.rules и добавьте в него следующую строку:

```
echo 'SUBSYSTEM=="tty", KERNEL=="tty8", GROUP="root"' | sudo tee /etc/rules.d/10-
↪tty8.rules
```

2) Проверьте, что нет ошибок:

```
udevadm test /devices/virtual/tty/tty8
```

3) Примените правило и убедитесь, что группа изменилась:

```
udevadm trigger /sys/devices/virtual/tty/tty8
ls -l /dev/tty8
crw------- 1 root root 4, 8 мар 7 06:37 /dev/tty8
```

4) Удалите файл 10-tty8.rules

```
rm /etc/udev/rules.d/10-tty8.rules
chown :tty /dev/tty8
```

Необязательно вручную менять разрешения на файл, можно перезагрузить компьютер.

Задание 2.

1) Так как устройство eth0 относится к классу сетевых устройств, посмотрите какие ссылки на устройства есть в /sys/net:

```
ls -l /sys/class/net/
lrwxrwxrwx 1 root root 0 мар 6 15:27 eth0 -> ../../devices/pci0000:00/0000:00:03.
↪0/net/eth0
lrwxrwxrwx 1 root root 0 мар 6 15:27 lo -> ../../devices/virtual/net/lo
```

Видно, что устройство находится по пути `/sys/devices/pci0000:00/0000:00:03.0/net/eth0.`

2) Зная путь к устройству, запустите команду udevadm:

```
udevadm info -a -p /sys/devices/pci0000:00/0000:00:03.0/net/eth0
```

Ключ `-a` указывает вывести все атрибуты устройства и всех его родительских устройств.

Задание 3.

1) Выполните команду:

```
modprobe -r intel_rapl_msr
```

2) Создайте файл /etc/modprobe.d/blacklist-rapl.conf:

```
sudo touch /etc/modprobe.d/blacklist-rapl.conf
echo "blacklist intel_rapl_msr" | sudo tee /etc/modprobe.d/blacklist-rapl.conf
```

После перезагрузки проверьте, что модуля intel_rapl_msr нет в выводе команды `lsmod`:

```
lsmod | grep intel_rapl_msr
```

3) Удалите файл /etc/modprobe.d/blacklist-rapl.conf и загрузите модуль intel_rapl_msr:

```
rm /etc/modprobe.d/blacklist-rapl.conf
modprobe intel_rapl_msr
```

Перезагрузите систему и убедитесь, что intel_rapl_msr снова загружен:

```
lsmod | grep intel_rapl_msr
```

18.4.2 Тест: Модуль 18. Управление устройствами и модулями ядра ОС

Пожалуйста, ответьте на несколько вопросов, чтобы закрепить полученные знания. Правильные ответы для самопроверки вы сможете найти в конце списка. Обратите внимание, что у некоторых вопросов может быть несколько верных вариантов ответов.

18.4.2.1 Вопросы

Вопрос 1. Какая директория отражает информацию об устройствах в ядре?

a. Директория `/sys`

b. Директория `/sys/devices`

c. Директория `/etc/modules-load.d`

d. Директория `/sys/class`

Вопрос 2. Есть два файла с одинаковыми названиями в директориях `/etc/udev/rules.d` и `/usr/lib/udev/rules.d/`. Какой из них будет иметь больший приоритет?

a. Тот, что в директории `/usr/lib/udev/rules.d/`

b. Они равнозначны

c. Тот, что в директории `/etc/udev/rules.d`

Вопрос 3. Какая команда позволяет посмотреть все PCI-устройства?

a. Команда `lspci`

b. Команда `lscpu`

c. Команда `lsusb`

d. Команда `lsdevice`

Вопрос 4. Какая команда служит для просмотра только USB-устройств?

a. Команда `lsblk`

b. Команда `lsipc`

c. Команда `lsmod`

d. Команда `lsusb`

Вопрос 5. Что делает команда `udevadm info -e`?

a. Выводит информацию о правилах udev

b. Показывает справку

c. Выгружает базу udev

d. Очищает базу udev

Вопрос 6. Какое расширение имеют файлы модулей ядра?

a. Расширение `so`

b. Расширение `ko`

c. Расширение `module`

d. Расширение `mo`

Вопрос 7. В какой директории находятся файлы модулей ядра?

a. Директория `/sys/module`

b. Директория `/lib64/modules`

c. Директория `/lib/modules/$(uname -r)`

d. Директория `/etc/modules-load.d`

Вопрос 8. В каком файле `modpobe` находит список всех встроенных модулей ядра?

a. Файл `/etc/modules`

b. Файл `/lib/modules/$(uname -r)/modules.order`

c. Файл `/lib/modules/$(uname -r)/modules.dep`

d. Файл `/lib/modules/$(uname -r)/modules.builtin`

Вопрос 9. Какая команда позволяет посмотреть список загруженных модулей ядра?

a. Команда `modprobe`

b. Команда `insmod`

c. Команда `lsmod`

d. Команда `rmmod`

Вопрос 10. Что делает команда `modprobe -r e1000`?

a. Выгружает модуль

b. Показывает информацию о модуле

c. Добавляет модуль

d. Выводит зависимости модуля e1000

Ответы на вопросы

1. b
2. c
3. a
4. d
5. c
6. b
7. c
8. d
9. c
10. a

18.5 Заключение

В этом модуле мы познакомились с содержимым псевдофайловой системы sysfs и правилами подсистемы udev, с помощью которых можно задавать действия, которые будут автоматически выполняться в системе при подключении/отключении устройств. Мы узнали немного больше о модулях ядра и о том, как ими управлять, поэтому нам станут теперь понятны сложные инструкции по настройке какого-нибудь специфического оборудования.

В следующем модуле мы подробнее познакомимся с файловыми системами в Linux и научимся работать с блочными устройствами и LVM.

18.6 Дополнительные источники информации

- Шаблон поиска (англ. wildcard pattern)
- Развернутую статью о правилах udev вы можете найти по ссылке https://www.arccomm.ru/\T2A\CYRP\T2A\cyrr\T2A\cyra\T2A\cyrv\T2A\cyri\T2A\cyrl\T2A\cyra-udev

18.7 Обратная связь

Если остались вопросы, то их всегда можно задать в специальной теме.

Модуль 19. Файловая система в Linux, работа с блочными устройствами и LVM

19.1 Введение

В этом модуле мы значительно расширим наши представления о работе с блочными устройствами и файловыми системами, а также познакомимся с менеджером логических томов LVM, который значительно расширяет возможности Linux-администраторов по управлению системами хранения.

Материала очень много, но вы не пугайтесь, не меньше половины страниц — это скриншоты, которые нужны для пояснения.

19.2 Подсистема хранения данных в Linux

На страницах этого курса мы уже обсуждали различные типы файловых систем: блочные (ext4, btrfs), псевдофайловые (proc, sys), в оперативной памяти (tmpfs, ramfs), сетевые (smb, nfs). Кроме

того, мы затрагивали также инструмент FUSE (*англ. Filesystem in USEr space*), с помощью которого можно монтировать диски без привилегий суперпользователя. Однако, подсистема хранения в Linux намного обширнее, и файловые системы – это только ее часть, см. рис. 1.112.

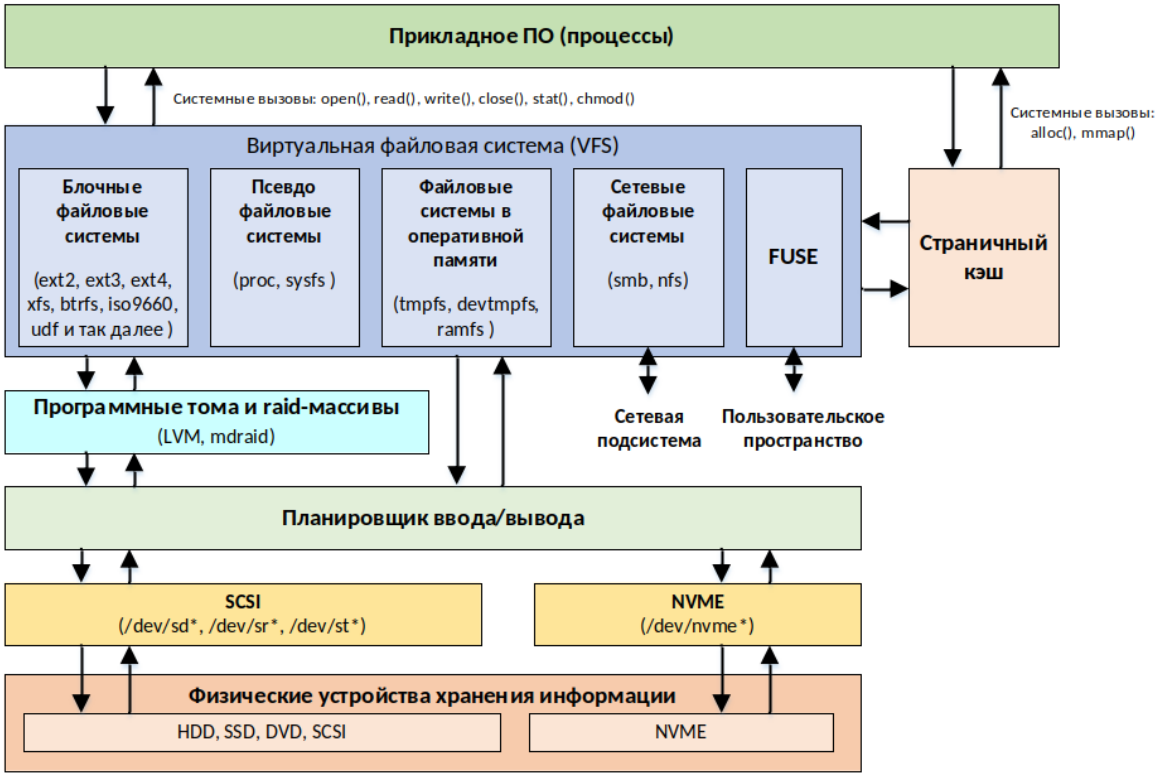


рис. 1.112 – Архитектура подсистемы хранения данных

Прикладное ПО

На верхнем уровне расположены прикладные программы. Они позволяют как управлять подсистемой хранения (создавать, изменять и удалять разделы дисков, форматировать разделы и так далее) так и работать с данными, хранящимися в файловых системах (создавать файлы, изменять их, менять права, просматривать атрибуты и так далее).

Через системные вызовы прикладное ПО взаимодействует с виртуальной файловой системой (*англ. Virtual File System, VFS*) и со страничным кэшем (*англ. Page Cache*).

Виртуальная файловая система и страничный кэш

Виртуальная файловая система (еще называемая виртуальным коммутатором файловой системы) — это прослойка, уровень абстракции, между системными вызовами и конкретной реализацией файловой системы. Она предназначена для единообразия доступа прикладного ПО к различным типам файловых систем (блочной, в оперативной памяти, сетевой и так далее). VFS декларирует программный интерфейс, позволяющий без внесения изменений в ядро ОС обеспечивать поддержку

новых файловых систем.

Страничный кэш – это набор страниц в оперативной памяти, в которые была записана информация с физического устройства хранения для обеспечения быстрого доступа к данным. При каждой операции страничного ввода-вывода, ядро ОС проверяет наличие страницы в кэше, и, по возможности, не использует затратную процедуру обращения к физическому устройству.

Программные тома и RAID-массивы

Виртуальная файловая система может взаимодействовать с планировщиком ввода/вывода непосредственно или через подсистему программных томов или RAID-массивов (таких как LVM или MDRAID), позволяющих гибко управлять разделами, расширять их, размещать их на нескольких физических устройствах и так далее, без необходимости прерывания работы (без простоя) и без необходимости переконфигурирования физических устройств хранения. Платой за это служит некоторое (небольшое) количество накладных расходов, за еще один уровень абстракции.

Планировщик ввода/вывода

Планировщик ввода/вывода – это компонент ядра, который выполняет оптимизацию запросов к диску для повышения производительности и пропускной способности, что особенно критично при работе с жесткими дисками, у которых перемещение считывающей головки является крайне затратной операцией.

Алгоритмов планирования существует несколько, и с помощью команды `sudo dmesg | grep "io scheduler"` вы можете посмотреть, какие из них были зарегистрированы в системе. Команда `cat /sys/block/sda/queue/scheduler` покажет, какой из них используется для конкретного диска.

По умолчанию ALSE использует планировщик mq-deadline, который хорошо подходит для операций асинхронной записи на HDD. Если же у вас SSD, то вам лучше использовать алгоритм kyber, который повышает приоритет операций чтения над записью, или отключить планировщик вовсе.

Физические устройства хранения информации и их файлы

На нижнем уровне находятся физические устройства хранения информации, а в качестве интерфейсов доступа к ним выступают файлы каталога `/dev`. Файлы блочных устройств, к примеру, именуются следующим образом:

- Файл `/dev/sd<N>` — дисковые устройства хранения информации с интерфейсами, поддерживающими последовательную передачу данных (SATA, SCSI).
- Файл `/dev/hd<N>` — дисковые устройства хранения с интерфейсами, поддерживающими параллельную передачу данных (PATA).

- Файл `/dev/st<N>` — ленточные накопители SCSI.
- Файл `/dev/nvme<N>` — NVMe накопители.

Для дисковых устройств sd и hd имя задается по маске sd/hd<номер_устройства>[<раздел>], где в качестве номера устройства выступают строчные буквы латинского алфавита (a,b,c,...), а номера раздела может быть как целым положительным числом (порядковый номер раздела на диске), так и отсутствовать (весь диск), например sdb1 – это первый раздел на втором накопителе SCSI.

Диски SSD могут именоваться как sd, если для подключения используют интерфейс SATA, и как nvme, если для подключения используется интерфейс PCI Express. Формат имен nvme довольно сильно отличается от sd/hd устройств, т.к. это не просто быстрые диски, а целая технология, которая была разработана специально для твердотельных накопителей. Имена NVMe устройств задаются в следующем формате:

```
nvme<номер_контроллера>n<пространство_имён>p<номер_раздела>
```

Где:

- **nvme<номер_контроллера>** — номер контроллера NVMe, начиная с нуля. Контроллер является символьным устройством.
- **n<пространство_имён>** — пространство имён (*англ. namespace*), нумерация начинается с единицы. Пространства имен напоминают RAID-массивы, так как могут выходить за пределы одного устройства и создавать логические тома, распределенные по нескольким контроллерам сразу. Это блочное устройство.
- **p<номер_раздела>** — номер раздела в пределах пространства имен, нумерация начинается с единицы. Эти разделы являются полной аналогией с разделами жестких дисков. Это блочное устройство.

Например, nvme0n1p1 – это первый раздел в первом пространстве имен на первом nvme-контроллере компьютера.

Примечание: Как вы знаете, в системе Windows во избежание проблем с именами дисков для логических томов, находящихся на дополнительных устройствах, рекомендуется использовать буквы с конца алфавита (X:\, Y:\, Z:\). Это связано это с тем, что автоматическая нумерация устройств зависит не только от того, к каким портам они подключены, но и от внешних факторов, например, как быстро они включаются.

Не следуя этой практике, вы вполне можете столкнуться с тем, что порядок дисков на сервере будет меняться между загрузками, и буквы станут прыгать в хаотичном порядке, как у дислектика, а вместе с ними начнут отваливаться сервисы, использующие эти данные.

В части автоматической нумерации дисков в Linux ситуация ровно такая же — буквы присваиваются дискам автоматически, и вы не можете полагаться на них, т.к. устройства вполне могут устроить день перемены имен.

Опираясь на информацию предыдущего модуля, вы конечно сможете закрепить конкретные имена за конкретными устройствами с помощью udev-правил. Но если вспомнить, что в Linux вместо букв логических дисков используется единое пространство имен файловой системы, то станет очевидно, что будет вполне достаточно в файле `/etc/fstab` примонтировать диски сразу по идентификаторам UUID. Как в ALSE, собственно, и сделано.

19.3 Виртуальная файловая система

Из модуля про работу с файлами в Linux мы уже знаем, что файловая система nix-подобных систем представлена единым пространством имен, что достигается через механизм монтирования различных файловых систем в единую структуру каталогов. Учитывая, что в единой структуре каталогов могут присутствовать совершенно разные файловые системы, потребовался унифицированный интерфейс, который бы обеспечил единообразное взаимодействие со всеми файловыми системами, и таким инструментом стала виртуальная файловая система (*от англ. Virtual File System, VFS*).

Система VFS позволяет с помощью одних и тех же системных вызовов взаимодействовать с любыми типами файловых систем, в том числе и сетевыми, как будто это обычная файловая система ext4 на локальном блочном устройстве. В состав VFS входят следующие типы объектов, которыми она оперирует:

- **суперблок** – метаданные файловой системы;
- **индексные дескрипторы** (*англ. inode*) – метаданные файла;
- **записи каталогов** (*англ. directory entry, dentry*) – специальные файлы, в которых хранятся жесткие ссылки, связывающие имена файлов с их индексными дескрипторами;
- **открытые файлы** – структуры, содержащие информацию об открытых файлах.

Для хранения информации о поддерживаемых типах файловых систем используется таблица, которая создается во время компиляции ядра. Каждая запись этой таблицы включает наименование типа и указатель на функцию, вызываемую во время монтирования этой файловой системы (если значение не указано, используется функция монтирования по умолчанию).

Функция монтирования используется для считывания суперблока, установки внутренних переменных и возврата дескриптора смонтированной системы обратно в VFS. После того, как система смонтирована, функции VFS используют открытый дескриптор для доступа к процедурам чтени-

я/записи этой файловой системы.

Для более эффективного управления памятью и устранения значительной фрагментации данных система VFS кэширует метаданные, используемые при монтировании (суперблок, индексные дескрипторы, записи каталогов и др.), с помощью специального механизма управления памятью, который называется «распределение slab» (*англ. slab allocation*). Секрет быстродействия этого алгоритма состоит в повторном использовании памяти, освобожденной одним объектом, для размещения другого объекта того же типа.

Управление кэшированием метаданных осуществляется с помощью переменной ядра ОС `vm.vfs_cache_pressure` в файле `/proc/sys/vm/vfs_cache_pressure`, значение которой представляет из себя целое число:

- `0` – ядро сохраняет (не освобождает) страничный кэш, может привести к нехватке оперативной памяти.
- `100` – ядро пытается сохранить справедливое распределение памяти между кэшем страниц (pagecache) и кэшем подкачки (swapcache). Используется по умолчанию.
- `>100` – ядро активно выгружает метаданные в VFS. При значении существенно выше 100 это может негативно сказаться на производительности VFS.

Текущую статистику по использованию кэшей slab можно посмотреть утилитой `sudo slabtop`

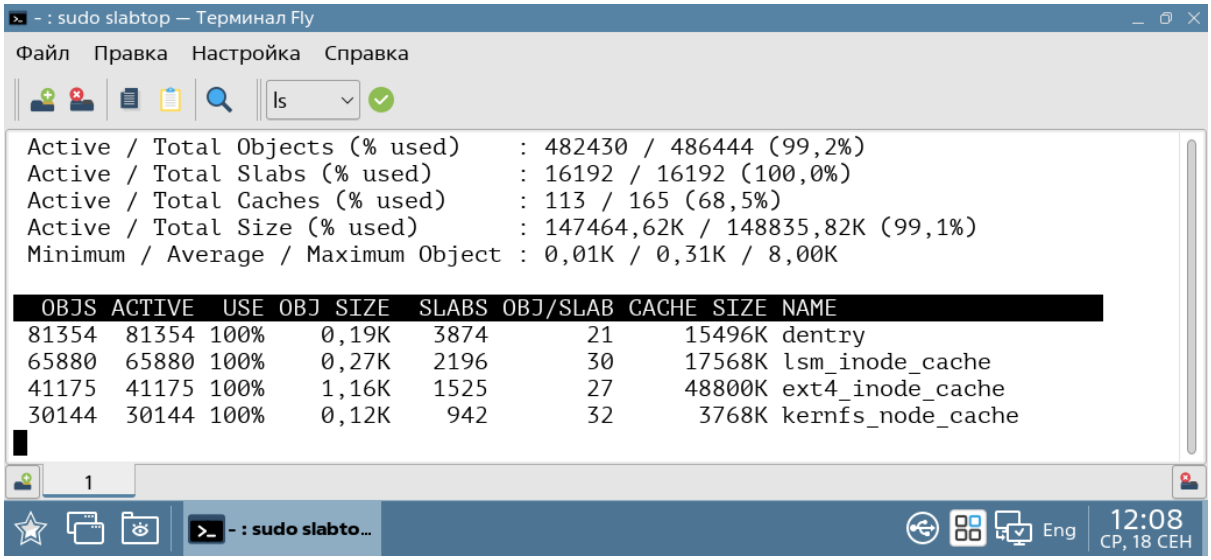


рис. 1.113 – Статистика по использованию кэшей slabtop

Для принудительного сохранения «грязных» страниц кэша (*англ. dirty pages*), которые были изменены после их помещения в кэш, можно воспользоваться утилитой `sync`. Она гарантирует, что все изменения в памяти будут записаны на диск, предотвращая потерю изменений в кэше при аварийном завершении работы системы.

Если же вам потребуется освободить память, зарезервированную под кэш, вы можете из-под суперпользователя выполнить команду:

`echo 3 | sudo tee /proc/sys/vm/drop_caches`

Где:

- `1` – освобождает pagecache,
- `2` – dentrie и inode кэши,
- `3` – все сразу.

```
localadmin@astra:~$ echo 3 | sudo tee /proc/sys/vm/drop_caches
[sudo] пароль для localadmin:
3
```

19.4 Файловая система ext4

Операционная система Linux способна работать с разными типами файловых систем, как поддерживающими концепцию индексных дескрипторов (ext4, xfs, BtrFS), так и совершенно чуждые им (exFAT, NTFS). Список поддерживаемых файловых систем определяется драйверами, то есть модулями, которые были включены в состав ядра и загружены в данный момент.

Список всех модулей ядра с драйверами файловых систем можно получить с помощью команды `ls /lib/modules/$(uname -r)/kernel/fs`.

```
localadmin@astra:~$ ls /lib/modules/$(uname -r)/kernel/fs
9p              btrfs       exfat     hpfs   nfs_common  overlayfs     sysv
adfs            cachefiles  f2fs      isofs  nfsd        pstore        ubifs
affs            ceph        fat       jffs2  nilfs2      qnx4          udf
afs             cifs        freevxfs  jfs    nls         qnx6          ufs
aufs            coda        fscache   ksmbd  ntfs        quota         vboxsf
autofs          cramfs      fuse      lockd  ntfs3       reiserfs      xfs
befs            dlm         gfs2      minix  ocfs2       romfs         zonefs
bfs             efs         hfs       netfs  omfs        shiftfs.ko
binfmt_misc.ko  erofs       hfsplus   nfs    orangefs    smbfs_common
```

Список файловых систем, поддерживаемых ядром в данный момент, можно получить командой `cat /proc/filesystems`. Словом, «nodev» (*англ. no device*) обозначаются файловые системы, которые не привязаны к устройствам, но это не означает, что данная система не используется, т.к. в эту категорию попадают псевдофайловые системы и системы, размещаемые в памяти.

```
localadmin@astra:~$ cat /proc/filesystems
nodev   sysfs
nodev   tmpfs
nodev   bdev
nodev   proc
```

(continues on next page)

(продолжение с предыдущей страницы)

```
nodev   cgroup
nodev   cgroup2
nodev   cpuset
nodev   devtmpfs
nodev   configfs
nodev   debugfs
nodev   tracefs
nodev   securityfs
nodev   sockfs
nodev   bpf
nodev   pipefs
nodev   ramfs
nodev   hugetlbfs
nodev   devpts
        ext3
        ext2
        ext4
        squashfs
        vfat
nodev   ecryptfs
        fuseblk
nodev   fuse
nodev   fusectl
nodev   mqueue
nodev   pstore
nodev   parsecfs
nodev   autofs
```

Для Linux родными считаются файловые системы из семейства Extended File System, EXT. Их разработка началась в начале 90-х для преодоления ограничений предшествующей им Minix File System как по длине имен файлов, так и по размеру поддерживаемых дисков.

К настоящему времени разработано несколько редакций, каждая из которых становилась значительно лучше предыдущей. В 2008 году была представлена четвертая и крайняя на текущий момент версия ext4, отличительными особенностями которой являются:

- Максимальный размер файла до 16 Тебибайт, 2^{40} байт (против 2 ТБ для ext3).
- Максимальный размер тома: до 1 Эксбибайта, 2^{60} байт (против 32 ТБ для ext3).
- Максимальное число подкаталогов в каждом каталоге 64 000 (против 32 000 для ext3).
- Максимальная длина имени файла 255 байт (или 127 русских символов в utf8).
- Журналируемая ФС, что помогает сохранить целостность файловой системы при сбоях.
- Использует контрольные суммы для автоматического обнаружения повреждения данных из-за аппаратных сбоев. КС добавлены для суперблока, inode, битовых карт блоков, блоков дерева экстентов, htree-узлов, журналов и др.

- Обратно-совместима к ext2 и ext3, что позволяет монтировать старые файловые системы через драйвер ext4.
- Поддерживает **пространственную запись файлов** – резервирование на диске всего требуемого пространства перед началом записи файла на диск.
- Поддерживает **экстенты** – хранение в дескрипторе адресов только первого и последнего блока из последовательного массива блоков данных. Размер экстента может достигать 128 МБ, что позволяет существенно повысить производительность ФС.
- Уменьшает фрагментацию данных за счет более рационального выделения блоков.
- Поддерживает отложенное выделение блоков памяти (непосредственно перед их записью на диск), что позволяет снизить нагрузку на кэш и увеличить производительность.
- Для работы с каталогами использует разновидность B-дерева, что позволяет без снижения производительности работать с большим количеством каталогов.
- Поддерживает быструю проверку занятых блоков, исключая из проверки индексные дескрипторы и свободные блоки данных.

К недостаткам ext4 можно отнести только отсутствие поддержки таких функций как дедупликация данных и снижение производительности при работе с разделами более 100 ТБ.

Хотя файловые системы семейства ext и отличаются друг от друга в деталях, но их структура практически идентична.

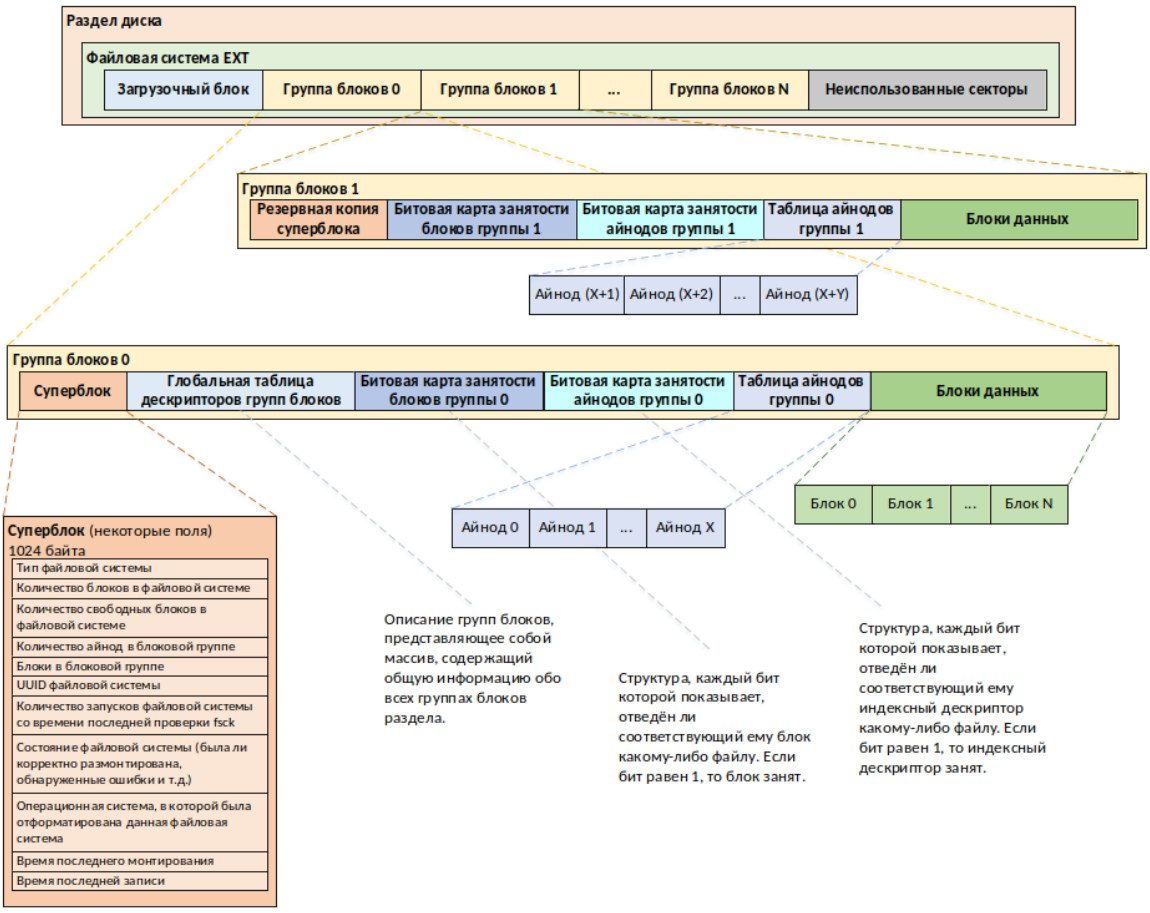


рис. 1.114 – Общая структура файловой системы семейства ext

В системе Linux процедура разметки раздела для работы с конкретной файловой системой называется созданием файловой системы, но более привычный для Windows-администраторов термин «форматирование диска» тоже используется.

Как мы помним, в начале раздела часть секторов выделяется под размещение загрузочного блока, а после него размещаются данные. Для уменьшения фрагментации и повышения производительности файловая система ext4 нарезает все доступное пространство на фрагменты, которые называются группами блоков.

В файловой системе NTFS нет аналога для групп блоков, так как каждая группа в какой-то степени является еще одной файловой системой, у которой есть собственные блоки данных, таблица файловых дескрипторов и дополнительная метаинформация. Маленькие файлы, конечно, не должны выходить за пределы одной группы, но если нам потребуется сохранить большой архив, то он без проблем оккупирует несколько групп сразу.

19.4.1 Состав группы блоков 0

Первая группа блоков, которая идет под номером ноль, немного отличается от всех остальных и содержит дополнительные метаданные. В нулевую группу блоков входят следующие разделы, см. рис. 1.114:

- **Суперблок** – это структура данных размером 1024 байта, в которой содержится информация о файловой системе. Для обеспечения надежности файловая система создаёт еще несколько резервных копий суперблока, и хранит их внутри других групп блоков в разных частях диска. Суперблок содержит следующую информацию:
  - Тип файловой системы (s_type).
  - Размер файловой системы в логических блоках, включая сам суперблок, ilist и блоки хранения данных (s_fsize).
  - Размер массива индексных дескрипторов (s_isize).
  - Число свободных блоков, доступных для размещения (s_tfree).
  - Число свободных индексных дескрипторов, доступных для размещения (s_t inode).
  - Состояние файловой системы (флаг модификации s_fmod, флаг режима монтирования s_fronly).
  - Размер логического блока в виде числа, обозначающего степень 2 (9 – 512 байт, 10 – 1КБ).
  - Количество логических блоков в группе блоков.
  - Количество индексных дескрипторов в группе блоков.
  - Идентификатор файловой системы (UUID).
  - Время последней записи.
  - Время последнего монтирования.
- **Глобальная таблица дескрипторов групп блоков** — это массив, в котором содержится информация о расположении всех групп блоков раздела.
- **Битовые карты занятости блоков и индексных дескрипторов** — помогают быстро определять, какие из блоков и индексных дескрипторов заняты, а какие свободны. Они представляют из себя массив двоичных данных, где каждый бит соответствует блоку данных или индексному дескриптору, а значение показывает занят он (1) или свободен (0).

- **Таблица индексных дескрипторов** — содержит массив индексных дескрипторов. Структура индексного дескриптора подробно рассматривалась в модуле про работу с файлами. Стоит отметить, что индексные дескрипторы создаются для всего раздела в целом и их нумерация уникальна для всех групп блоков.

- **Блоки данных** — объединяют группу секторов диска для хранения содержимого файлов, каталогов, косвенной адресации. Размер блоков данных кратен размеру сектора, который в большинстве случаев составляет 512 байт, но есть диски с секторами и по 4 Кб. Блоки данных по своей сути ближе всего к кластерам NTFS.

19.4.2 Состав групп блоков 1-N

Остальные группы блоков содержат следующие разделы:

- Битовая карта занятости блоков группы 0
- Битовая карта занятости индексных дескрипторов группы 0
- Таблица индексных дескрипторов группы 0
- Блоки данных

Некоторые группы блоков могут содержать резервную копию суперблока.

Посмотреть информацию суперблока можно, например, командой `sudo tune2fs -l /dev/sda1`.

```
localadmin@astra:~$ sudo tune2fs -l /dev/sda1
tune2fs 1.44.5 (15-Dec-2018)
Filesystem volume name:   <none>
Last mounted on:          /
Filesystem UUID:          1864fbac-4e75-42ec-9a52-c0944d4fd6b9
Filesystem magic number:  0xEF53
Filesystem revision #:    1 (dynamic)
Filesystem features:      has_journal ext_attr resize_inode dir_index filetype needs_
↪recovery extent 64bit flex_bg sparse_super large_file huge_file dir_nlink extra_
↪isize metadata_csum
Filesystem flags:         signed_directory_hash
Default mount options:    user_xattr acl
Filesystem state:         clean
Errors behavior:          Continue
Filesystem OS type:       Linux
Inode count:              987360
Block count:              3943936
Reserved block count:     197196
Free blocks:              1040563
Free inodes:              701427
First block:              0
Block size:               4096
Fragment size:            4096
Group descriptor size:    64
```

(continues on next page)

(продолжение с предыдущей страницы)

```
Reserved GDT blocks:      1024
Blocks per group:         32768
Fragments per group:      32768
Inodes per group:         8160
Inode blocks per group:   510
Flex block group size:    16
Filesystem created:       Tue May 16 11:29:02 2023
Last mount time:          Thu Aug 31 07:48:36 2023
Last write time:          Thu Aug 31 07:48:35 2023
Mount count:              15
Maximum mount count:      -1
Last checked:             Tue May 16 11:29:02 2023
Check interval:           0 (<none>)
Lifetime writes:          56 GB
Reserved blocks uid:      0 (user root)
Reserved blocks gid:      0 (group root)
First inode:              11
Inode size:               256
Required extra isize:     32
Desired extra isize:      32
Journal inode:            8
Default directory hash:   half_md4
Directory Hash Seed:      5a628ba0-789e-4649-a547-cce2c9147d81
Journal backup:           inode blocks
Checksum type:            crc32c
Checksum:                 0x9fbb256a
```

Чтобы получить информацию о файловых системах ext, можно воспользоваться утилитой dumpe2fs. Например, для вывода информации обо всех резервных копиях суперблока можно отфильтровать её вывод через grep:

```
localadmin@astra:~$ sudo dumpe2fs /dev/sda1 | grep superblock
dumpe2fs 1.44.5 (15-Dec-2018)
 Primary superblock at 0, Group descriptors at 1-2
 Backup superblock at 32768, Group descriptors at 32769-32770
 Backup superblock at 98304, Group descriptors at 98305-98306
 Backup superblock at 163840, Group descriptors at 163841-163842
 Backup superblock at 229376, Group descriptors at 229377-229378
 Backup superblock at 294912, Group descriptors at 294913-294914
 Backup superblock at 819200, Group descriptors at 819201-819202
 Backup superblock at 884736, Group descriptors at 884737-884738
 Backup superblock at 1605632, Group descriptors at 1605633-1605634
 Backup superblock at 2654208, Group descriptors at 2654209-2654210
```

19.5 Рекомендуемая схема разбиения

В Linux у дисков есть разделы, но мы с ними не работаем как с логическими дисками Windows. Файловая система раздела может быть смонтирована в любую точку единой файловой системы, которая соответствует системному диску, с которого была выполнена загрузка. Точно также монтируются внешние носители информации, такие как USB-флешки, DVD-диски.

Вы можете использовать для работы сервера один раздел диска, но в высоконагруженных системах некоторые каталоги рекомендуется выносить отдельно по следующим причинам:

Каталог	Для чего используется	Почему может быть выгодно перенести в другой раздел	Рекомендуемый размер
/home	Пользовательские файлы	Домашние каталоги пользователей, аналог `C:\Users` в Windows. Перенос на другой раздел диска помогает исключить переполнение системного диска и гарантирует сохранность файлов при переустановке.	Не менее 100 МБ
/tmp	Временные файлы	Каталог хранит временные файлы. Перенос на другой раздел диска помогает исключить переполнение системного диска.	Не менее 50 МБ
/usr, /opt	Файлы программ	В этих каталогах хранятся файлы программ, аналог `Program Files` Если система работает на быстром SSD, то программы можно вынести на HDD большого объема.	Не менее 8 ГБ
/var	Динамические данные	Этот каталог используется для динамических данных, таких как файлы журнала, кэшированные данные и т.д. Перенос на другой раздел диска помогает исключить переполнение системного диска	Не менее 400 МБ
swap	Раздел подкачки	Раздел подкачки необходим для увеличения объема доступной памяти, аналог файла подкачки в Windows. Перенос на другой раздел диска позволяет повысить производительность, если это SSSD диск, и уменьшить фрагментацию файла подкачки.	Выбирается исходя из объема оперативной памяти. Если необходима гибернация, то не менее объема оперативной памяти + небольшой запас в 1-2 ГБ.

Однако, следует учитывать, что при разбиении диска на разделы мы можем ошибиться в своих предположениях о том, сколько места потребуется на те или иные задачи, и поменять это решение «на лету» уже не получится.

Решением этой проблемы является использование дополнительного уровня абстракции в виде системы логических томов, работающих поверх дисковых разделов. Универсальным решением являются менеджер логических томов LVM, который будет рассмотрен далее в этом модуле.

19.6 Управление дисковыми разделами и файловыми системами

Перед практическим изучением работы утилит, нам необходимо добавить несколько виртуальных дисков в нашу виртуальную машину, см. рис. 1.115. Добавим 3 диска по 1ГБ и один диск, объемом 2ГБ.

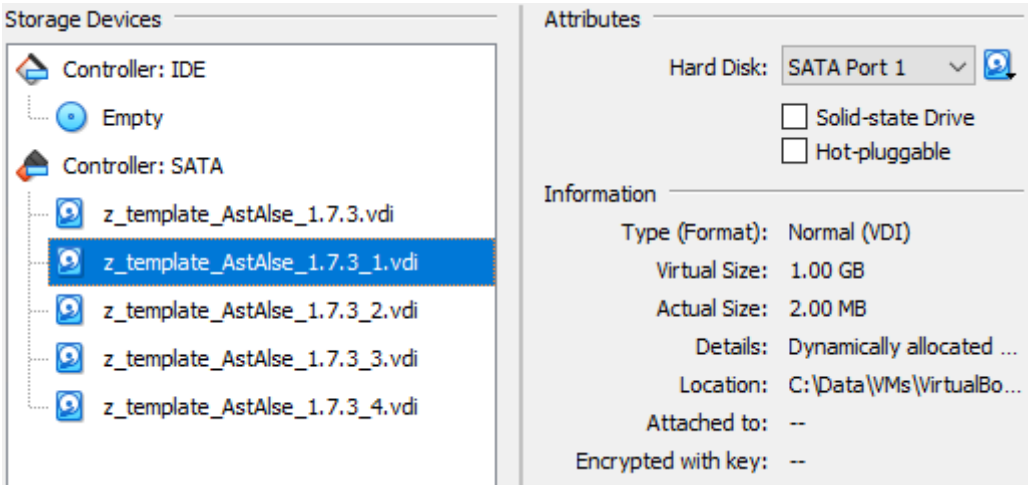


рис. 1.115 – Добавление дисков в VM в VirtualBox

19.6.1 Управление дисковыми разделами

Для управления дисковыми разделами, в Astra Linux доступно несколько утилит:

- Утилита `fdisk` – консольная утилита для разметки диска (создания и удаления разделов);
- Утилита `parted` – консольная утилита для разметки диска (создания, удаления и изменения разделов);
- Утилита `sfdisk` – утилита для использования в скриптах;
- Утилита `cfdisk` – псевдографическая утилита работы с разделами;
- Утилита `gparted` – графическая утилита разметки диска.

19.6.1.1 Утилита fdisk

Получение информации о дисках и разделах

После добавления дисков посмотрим список устройств хранения и их разделы с помощью команды `sudo fdisk -l` (L строчная).

```
localadmin@astra:~$ sudo fdisk -l
Диск /dev/sda: 16 GiB, 17179869184 байт, 33554432 секторов
Disk model: VBOX HARDDISK
Единицы: секторов по 1 * 512 = 512 байт
Размер сектора (логический/физический): 512 байт / 512 байт
Размер I/O (минимальный/оптимальный): 512 байт / 512 байт
Тип метки диска: dos
Идентификатор диска: 0x8892b118

Устр-во    Загрузочный  начало     Конец  Секторы    Размер  Идентификатор  Тип
/dev/sda1  *              2048  31553535  31551488      15G             83  Linux
/dev/sda2             31555582  33552383  1996802      975M              5  ␣
↪Расширенный
/dev/sda5             31555584  33552383  1996800      975M             82  Linux␣
↪своп / Solari


Диск /dev/sdb: 1 GiB, 1073741824 байт, 2097152 секторов
Disk model: VBOX HARDDISK
Единицы: секторов по 1 * 512 = 512 байт
Размер сектора (логический/физический): 512 байт / 512 байт
Размер I/O (минимальный/оптимальный): 512 байт / 512 байт


Диск /dev/sdc: 1 GiB, 1073741824 байт, 2097152 секторов
Disk model: VBOX HARDDISK
Единицы: секторов по 1 * 512 = 512 байт
Размер сектора (логический/физический): 512 байт / 512 байт
Размер I/O (минимальный/оптимальный): 512 байт / 512 байт


Диск /dev/sdd: 1 GiB, 1073741824 байт, 2097152 секторов
Disk model: VBOX HARDDISK
Единицы: секторов по 1 * 512 = 512 байт
Размер сектора (логический/физический): 512 байт / 512 байт
Размер I/O (минимальный/оптимальный): 512 байт / 512 байт


Диск /dev/sde: 2 GiB, 2147483648 байт, 4194304 секторов
Disk model: VBOX HARDDISK
Единицы: секторов по 1 * 512 = 512 байт
Размер сектора (логический/физический): 512 байт / 512 байт
Размер I/O (минимальный/оптимальный): 512 байт / 512 байт
```

Как мы видим, утилита вывела информацию о системном диске sda. Первые 4 раздела на нем являются основными, поэтому у третьего, расширенного раздела имя sda5. На новом созданном диске sdb разделов пока нет.

Для вывода информации только об одном диске необходимо к команде добавить имя диска, например, `sudo fdisk -l /dev/sdb`.

```
localadmin@astra:~$ sudo fdisk -l /dev/sdb
Диск /dev/sdb: 1 GiB, 1073741824 байт, 2097152 секторов
Disk model: VBOX HARDDISK
Единицы: секторов по 1 \* 512 = 512 байт
Размер сектора (логический/физический): 512 байт / 512 байт
Размер I/O (минимальный/оптимальный): 512 байт / 512 байт
```

Утилита fdisk может работать также в интерактивном режиме. Для этого необходимо просто указать имя диска `sudo fdisk /dev/sdb`.

```
localadmin@astra:~$ sudo fdisk /dev/sdb

Добро пожаловать в fdisk (util-linux 2.33.1).
Изменения останутся только в памяти до тех пор, пока вы не решите записать их.
Будьте внимательны, используя команду write.

Устройство не содержит стандартной таблицы разделов.
Создана новая метка DOS с идентификатором 0x68735955.
Команда (m для справки):
```

Введите `m` и нажмите Enter, чтобы утилита вывела справку по работе с ней:

```
Команда (m для справки): m

Справка:

DOS (MBR)
a    переключение флага загрузки
b    редактирование вложенной метки диска BSD
c    переключение флага dos-совместимости

Общие
d удалить раздел
F показать свободное неразмеченное пространство
l список известных типов разделов
n добавление нового раздела
p вывести таблицу разделов
t изменение типа раздела
v проверка таблицы разделов
i вывести информацию о разделе

Разное
m вывод этого меню
u изменение единиц измерения экрана/содержимого
x дополнительная функциональность (только для экспертов)

Сценарий
I загрузить разметку из файла сценария sfdisk
O записать разметку в файл сценария sfdisk

Записать и выйти
w запись таблицы разделов на диск и выход
```

(continues on next page)

(продолжение с предыдущей страницы)

```
q выход без сохранения изменений

Создать новую метку
g создание новой пустой таблицы разделов GPT
G создание новой пустой таблицы разделов SGI (IRIX)
o создание новой пустой таблицы разделов DOS
s создание новой пустой таблицы разделов Sun

Команда (m для справки):
```

Введем список известных типов разделов командой `l` и нажмите Enter:

```
...
Команда (m для справки): l

0 Пустой          24 NEC DOS         81 Minix / старый    bf    Solaris
1 FAT12           27 Скрытый NTFS Wi 82 Linux своп / So   c1    DRDOS/sec (FAT-
2 XENIX root      39 Plan 9          83 Linux             c4    DRDOS/sec (FAT-
3 XENIX usr       3c PartitionMagic  84 OS/2 hidden or    c6    DRDOS/sec (FAT-
4 FAT16 <32M      40 Venix 80286     85 Linux расширен    c7    Syrinx
5 Расширенный     41 PPC PReP Boot   86 NTFS набор томов  da    Данные не ФС
6 FAT16           42 SFS             87 NTFS набор томов  db    CP/M / CTOS / .
7 HPFS/NTFS/exFAT 4d QNX4.x          88 Linux plaintext   de    Dell Utility
8 AIX             4e QNX4.x 2-я част 8e Linux LVM         df    BootIt
9 AIX загрузочный 4f QNX4.x 3-я част 93 Amoeba            e1    DOS access
a OS/2 Boot-менед 50 OnTrack DM      94 Amoeba BBT        e3    DOS R/O
b W95 FAT32       51 OnTrack DM6 Aux 9f BSD/OS            e4    SpeedStor
c W95 FAT32 (LBA) 52 CP/M            a0 IBM Thinkpad hi   ea    Rufus alignment
e W95 FAT16 (LBA) 53 OnTrack DM6 Aux a5 FreeBSD          eb    BeOS фс
f W95 расшир. (LB 54 OnTrackDM6      a6 OpenBSD          ee    GPT
10 OPUS            55 EZ-Drive        a7 NeXTSTEP          ef    EFI (FAT-12/16/
11 Скрытый FAT12   56 Golden Bow      a8 Darwin UFS        f0    Linux/PA-RISC з
12 Compaq диагност 5c Priam Edisk     a9 NetBSD            f1    SpeedStor
14 Скрытый FAT16 < 61 SpeedStor       ab Darwin загрузоч   f4    SpeedStor
16 Скрытый FAT16   63 GNU HURD или Sy af HFS / HFS+        f2    DOS вторичный
17 Скрытый HPFS/NT 64 Novell Netware  b7 BSDI фс           fb    VMware VMFS
18 AST SmartSleep  65 Novell Netware  b8 BSDI своп         fc    VMware VMKCORE
1b Скрытый W95 FAT 70 DiskSecure Mult bb Boot Wizard скр   fd    Автоопределение
1c Скрытый W95 FAT 75 PC/IX           bc Acronis FAT32 L   fe    LANstep
1e Скрытый W95 FAT 80 Old Minix       be Solaris загр.     ff    BBT

Команда (m для справки):
```

Как видите, список не маленький, тут есть и FAT и даже NTFS. Эта таблица в дальнейшем нам еще пригодится.

Создание разделов

Создадим новый раздел, используя команду `n` (при этом на первые 3 вопроса ответы можно опустить – нас устроит значение по умолчанию):

n (создать раздел) -> [**p**] (первичный) -> [**1**] (номер раздела) -> [**2048**] (первый сектор раздела,

традиционно место в начале диска зарезервировано под код первичного загрузчика и «пустые» сектора).

В примере ниже Enter указан как кнопка, т.е. будет введены значения по умолчанию, вычисленные под размер вашего диска:

```
...
Команда (m для справки): n

Тип раздела
  p   основной    (0 первичный, 0 расширенный, 4 свободно)
  e   расширенный (контейнер для логических разделов)
Выберите (по умолчанию - p): <Enter>

Используется ответ по умолчанию p
Номер раздела (1-4, default 1): 1
Первый сектор (2048-2097151, default 2048): <Enter>
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-2097151, default 2097151):
```

Затем утилита попросит ввести последний сектор раздела (значение по умолчанию – самый последний сектор, весь диск). Кроме номера сектора, можно использовать выражения +/-<**сектора**> или +/-<**размер**>**{K,M,G,T,P}**, где K, M, G, T и P – это килобайты, мегабайты, и так далее.

А мы создадим раздел размером в 200 МБ. Для это введем +**200M**.

```
...
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-2097151, default 2097151): +200M

Создан новый раздел 1 с типом 'Linux' и размером 200 MiB

Команда (m для справки):
```

Введем команду p и выведем информацию о разделах на диске:

```
...
Команда (m для справки): p
Диск /dev/sdb: 1 GiB, 1073741824 байт, 2097152 секторов
Disk model: VBOX HARDDISK
Единицы: секторов по 1 \* 512 = 512 байт
Размер сектора (логический/физический): 512 байт / 512 байт
Размер I/O (минимальный/оптимальный): 512 байт / 512 байт
Тип метки диска: dos
Идентификатор диска: 0x68735955

Устр-во       Загрузочный начало Конец Секторы Размер Идентификатор Тип
/dev/sdb1                   2048 411647 409600    200M            83 Linux
```

Как видите, по умолчанию разделу был присвоен тип Linux (ему соответствует идентификатор 83).

Создадим еще пару разделов по 100МБ.

```
Устр-во   Загрузочный начало  Конец Секторы Размер Идентификатор Тип
/dev/sdb1                2048 411647 409600 200M              83 Linux
```

(continues on next page)

(продолжение с предыдущей страницы)

```
/dev/sdb2              411648 616447 204800 100M              83 Linux
/dev/sdb3              616448 821247 204800 100M              83 Linux
```

Изменение типа раздела

Изменим тип у второго раздела (/dev/sdb2) с Linux (83) на NTFS (87). Для этого введем команду `t`, выберем раздел и тип:

```
Команда (m для справки): t
Номер раздела (1-3, default 3): 2
Шестнадцатеричный код (введите L для получения списка кодов): 87

Тип раздела 'Linux' изменен на 'NTFS volume set'.
```

Выведем информацию о разделах ещё раз (команда `p`):

```
Команда (m для справки): p
Устр-во    Загрузочный начало Конец Секторы Размер Идентификатор Тип
/dev/sdb1                2048 411647 409600 200M              83 Linux
/dev/sdb2              411648 616447 204800 100M              87 NTFS набор томов
/dev/sdb3              616448 821247 204800 100M              83 Linux
```

Удаление раздела

Удалим второй раздел командой `d`.

```
...
Команда (m для справки): d
Номер раздела (1-3, default 3): 2

Раздел 2 был удален.
```

```
...
Команда (m для справки): p
Устр-во Загрузочный начало Конец Секторы Размер Идентификатор Тип
/dev/sdb1              2048 411647 409600 200M              83 Linux
/dev/sdb3            616448 821247 204800 100M              83 Linux
```

Завершение работы с утилитой fdisk

Завершим работу с утилитой fdisk командой `q` (без сохранения изменений).

Важно: Для того что бы сохранить изменения и завершить работу с утилитой, необходимо воспользоваться командой `w`, в противном случае, выполненные вами изменения с таблицей разделов не будут физически записаны на диск!

Создание таблицы разделов на диске

Для создания таблицы разделов на диске, запустим интерактивный режим работы утилиты с диском sdb воспользуемся командой `g`.

```
localadmin@astra:~$ sudo fdisk /dev/sdb

Добро пожаловать в fdisk (util-linux 2.33.1).
Изменения останутся только в памяти до тех пор, пока вы не решите записать их.
Будьте внимательны, используя команду write.

Устройство не содержит стандартной таблицы разделов.
Создана новая метка DOS с идентификатором 0x78840861.

Команда (m для справки): g
Created a new GPT disklabel (GUID: AE51D739-8831-1B4D-9461-40AE96E74B32).
```

Создадим один раздел размером 100МБ, выведем информацию о созданном разделе командой `<i>` и выйдем, сохранив изменения.

```
...
Команда (m для справки): n
Номер раздела (1-128, default 1):
Первый сектор (2048-2097118, default 2048):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-2097118, default 2097118): +100M

Создан новый раздел 1 с типом 'Linux filesystem' и размером 100 MiB.

Команда (m для справки): i
Номер раздела (1,2, default 2): 1

        Device: /dev/sdb1
         Start: 2048
           End: 206847
       Sectors: 204800
          Size: 100M
          Type: Файловая система Linux
     Type-UUID: 0FC63DAF-8483-4772-8E79-3D69D8477DE4
          UUID: 81A3DD77-3002-1F4B-A9D1-758073EC7E0A

Команда (m для справки): w
Таблица разделов была изменена.
Вызывается ioctl() для перечитывания таблицы разделов.
Синхронизируются диски.
```

После создания разделов с помощью `fdisk` стоит проверить, дошла ли информация о новых разделах до ядра ОС. Для этого воспользуемся командой `lsblk`.

```
localadmin@astra:~$ lsblk
NAME MAJ:MIN   RM  SIZE RO TYPE MOUNTPOINT
sda    8:0      0   16G 0 disk
```

(continues on next page)

(продолжение с предыдущей страницы)

```
├─sda1 8:1      0   15G 0 part /
├─sda2 8:2      0    1K 0 part
└─sda5 8:5      0  975M 0 part [SWAP]
sdb    8:16     0    1G 0 disk
└─sdb1 8:17     0  100M 0 part
sdc    8:32     0    1G 0 disk
sdd    8:48     0    1G 0 disk
sde    8:64     0    2G 0 disk
sr0   11:0      1 1024M 0 rom
```

Если информация не обновилась, необходимо воспользоваться командой `sudo partprobe`.

19.6.1.2 Утилита parted

Знакомство с утилитой `parted` стоит начать со справки. Запустим утилиту `sudo parted` и выведем список доступных команд используя команду (parted) `help`.

```
localadmin@astra:~$ sudo parted
GNU Parted 3.2
Using /dev/sda
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) help
 align-check TYPE N                        check partition N for TYPE(min  opt)␣
↪alignment
 help [COMMAND]                           print general help, or help on COMMAND
 mklabel,mktable LABEL-TYPE               create a new disklabel (partition table)
 mkpart PART-TYPE [FS-TYPE] START END     make a partition
 name NUMBER NAME                         name partition NUMBER as NAME
 print [devices  free  list,all|NUMBER]   display the partition table, available␣
↪devices,
       free space, all found partitions, or a particular partition
quit                                      exit program
rescue START END                          rescue a lost partition near START and END
resizepart NUMBER END                     resize partition NUMBER
rm NUMBER                                 delete partition NUMBER
select DEVICE                             choose the device to edit
disk_set FLAG STATE                       change the FLAG on selected device
disk_toggle [FLAG]                        toggle the state of FLAG on selected device
set NUMBER FLAG STATE                     change the FLAG on partition NUMBER
toggle [NUMBER [FLAG]]                    toggle the state of FLAG on partition NUMBER
unit UNIT                                 set the default unit to UNIT
version                                   display the version number and copyright
      information of GNU Parted
(parted)
```

Как видим, в отличии от утилиты `fdisk`, которая не умеет сама изменять размеры созданных разделов, утилита `parted`, напротив, позволяет это делать.

Важно: В отличии от `fdisk`, `gparted` вносит изменения сразу, перед некоторыми опасными операциями она просит подтвердить выполнение действия, например, при уменьшении размера диска.

Но при выполнении некоторых операций, она такого подтверждения не потребует, например, при удалении раздела.

Обратите внимание, что по умолчанию утилита будет использовать первое устройство (3 строка – Using /dev/sda). Для выбора определенного устройства при запуске утилиты необходимо указать его имя, например, `sudo parted /dev/sdb`.

И в том и в другом случае, утилита parted будет работать в интерактивном режиме, ожидая от пользователя ввода её внутренних команд (parted). Для запуска утилиты в не интерактивном режиме достаточно вызвать её с необходимыми параметрами, например, `sudo parted /dev/sdc mklabel gpt`.

```
localadmin@astra:~$ sudo parted /dev/sdc mklabel gpt
Information: You may need to update /etc/fstab.
```

Получение информации о дисках и разделах

Команда (parted) `print` выведет информацию о текущем диске.

```
(parted) print
Model: ATA VBOX HARDDISK (scsi)
Disk /dev/sda: 17,2GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number Start   End    Size   Type    File system  Flags
1      1049kB 16,2GB 16,2GB primary ext4         boot
2      16,2GB 17,2GB 1022MB extended
5      16,2GB 17,2GB 1022MB logical linux-swap(v1)
```

Выведем список устройств командой (parted) `print devices`.

```
(parted) print devices
/dev/sda (17,2GB)
/dev/sdb (1074MB)
/dev/sdc (1074MB)
/dev/sdd (1074MB)
/dev/sde (2147MB)
```

Кроме `print devices` есть и другие команды для вывода информации о дисках.

Команда (parted) `print free` выводит информацию о свободном месте разделах текущего диска.

```
(parted) print free
Model: ATA VBOX HARDDISK (scsi)
Disk /dev/sda: 17,2GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:
```

(continues on next page)

(продолжение с предыдущей страницы)

```
Number Start End     Size  Type      File system       Flags
       32,3kB 1049kB 1016kB           Free Space
1      1049kB 16,2GB 16,2GB primary   ext4              boot
       16,2GB 16,2GB 1048kB           Free Space
2      16,2GB 17,2GB 1022MB extended
5      16,2GB 17,2GB 1022MB logical   linux-swap(v1)
       17,2GB 17,2GB 1049kB           Free Space
```

Команда (parted) `print list` выводит информацию о дисках списком.

```
(parted) print list
Model: ATA VBOX HARDDISK (scsi)
Disk /dev/sda: 17,2GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number Start  End     Size     Type     File system     Flags
1      1049kB  16,2GB  16,2GB  primary   ext4           boot
2      16,2GB  17,2GB  1022MB  extended
5      16,2GB  17,2GB  1022MB  logical   linux-swap(v1)

**Model: ATA VBOX HARDDISK (scsi)**
Disk /dev/sdb: 1074MB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number Start End    Size File system Name Flags
1      1049kB 106MB 105MB
. . .
```

Выбор раздела

Для выбора раздела введем команду (parted):

`select <имя_устройства>`

Например, выберем второй диск командой `select /dev/sdb`.

Создание разделов

Для создания разделов воспользуемся командой (parted):

`mkpart <PART-TYPE> [<FS-TYPE>] <START> <END>`

Где:

- **<PART-TYPE>** — для MBR указывает тип раздела (primary, extended или logical), а для GPT

устанавливает метку раздела,

- <**FS-TYPE**> – определяет тип файловой системы,
- <**START**> и <**END**> – начало и конец раздела в процентном выражении от объема диска, в секторах (s), мегабайтах (MB), гигабайтах (GB) и других единицах.

Создадим раздел размером в 20% объёма диска /dev/sdb командой

```
(parted) mkpart Data ext4 106MiB 20%
(parted) print
Model: ATA VBOX HARDDISK (scsi)
Disk /dev/sdb: 1074MB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number Start  End    Size File system Name Flags
1      1049kB 106MB 105MB
2      111MB  215MB 104MB            Data
```

Изменение размера раздела

Для изменения размера раздела, воспользуемся командой (parted):

`resizepart <номер_раздела> <новый_конец_раздела>`

Например, увеличим второй раздел на 100МиБ `resizepart 2 315MiB`.

```
(parted) resizepart 2 315MiB
(parted) print
Model: ATA VBOX HARDDISK (scsi)
Disk /dev/sdb: 1074MB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number Start   End    Size   File system  Name  Flags
1      1049kB  106MB  105MB
2      111MB   330MB  219MB  ext4         Data
```

А теперь уменьшим размер второго раздела на 19 МиБ `resizepart 2 297MiB`.

```
(parted) resizepart 2 297MiB
Warning: Shrinking a partition can cause data loss, are you sure you want to continue?
Yes/No? Y
(parted) print
Model: ATA VBOX HARDDISK (scsi)
Disk /dev/sdb: 1074MB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:
```

(continues on next page)

(продолжение с предыдущей страницы)

```
Number  Start  End    Size    File system    Name    Flags
1       1049kB  106MB  105MB
2       111MB   311MB  200MB                  Data
```

При этом, утилита попросит нас подтвердить операцию, так как при ней возможно потеря данных.

Удаление раздела

Для удаления раздела служит команда (parted) `rm <номер_раздела>`. Удалим первый раздел на диске /dev/sdb командой `rm 1`.

```
(parted) select /dev/sdb
Using /dev/sdb
(parted) rm 1
(parted) print
Model: ATA VBOX HARDDISK (scsi)
Disk /dev/sdb: 1074MB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number  Start  End    Size    File system    Name    Flags
2       111MB  311MB  200MB                 Data
```

Завершим работу с утилитой командой (parted) `quit`.

```
(parted) quit
localadmin@astra:~$
```

После создания разделов утилитой `parted` стоит проверить, дошла ли информация о новых разделах до ядра ОС. Для этого воспользуемся командой `lsblk`.

```
localadmin@astra:~$ lsblk
NAME   MAJ:MIN  RM SIZE RO TYPE MOUNTPOINT
sda    8:0       0  16G  0 disk
├─sda1 8:1       0  15G  0 part /
├─sda2 8:2       0   1K  0 part
└─sda5 8:5       0 975M  0 part [SWAP]
sdb    8:16      0   1G  0 disk
└─sdb2 8:18      0 191M  0 part
sdc    8:32      0   1G  0 disk
sdd    8:48      0   1G  0 disk
sde    8:64      0   2G  0 disk
sr0   11:0       1 1024M 0 rom
```

Если информация не обновилась, необходимо воспользоваться командой `sudo partprobe`.

19.6.2 Управление файловыми системами

19.6.2.1 Создание и изменение файловой системы

Для управления файловыми системами есть несколько команд:

- `mkfs.<тип_файловой_системы> [<параметры>] <устройство>` – создание файловой системы
- `mkswap <устройство>` – создание раздела подкачки
- `resize2fs <устройство> <размер>` – изменение размера раздела файловой системы

Команда `mkfs` использует настройки по умолчанию, которые заданы в файле `/etc/mke2fs.conf`. Обратите внимание, что некоторые команды `mkfs` являются ссылками на другие утилиты:

```
localadmin@astra:~$ ls -l /usr/sbin | grep mkfs
lrwxrwxrwx 1 root root         8 мая 13  2018 mkdosfs -> mkfs.fat
-rwxr-xr-x 1 root root     14848 апр 24 13:31 mkfs
-rwxr-xr-x 1 root root     35336 апр 24 13:31 mkfs.bfs
-rwxr-xr-x 1 root root     39440 апр 24 13:31 mkfs.cramfs
lrwxrwxrwx 1 root root         9 ноя  1  2018 mkfs.exfat -> mkexfatfs
lrwxrwxrwx 1 root root         6 дек  5  2022 mkfs.ext2 -> mke2fs
lrwxrwxrwx 1 root root         6 дек  5  2022 mkfs.ext3 -> mke2fs
lrwxrwxrwx 1 root root         6 дек  5  2022 mkfs.ext4 -> mke2fs
-rwxr-xr-x 1 root root     39968 мая 13  2018 mkfs.fat
-rwxr-xr-x 1 root root    105048 апр 24 13:31 mkfs.minix
lrwxrwxrwx 1 root root         8 мая 13  2018 mkfs.msdos -> mkfs.fat
lrwxrwxrwx 1 root root         6 ноя 21  2022 mkfs.ntfs -> mkntfs
lrwxrwxrwx 1 root root         8 мая 13  2018 mkfs.vfat -> mkfs.fat
-rwxr-xr-x 1 root root    355768 фев 22  2019 mkfs.xfs
```

С параметрами утилиты можно ознакомиться в справке, но стоит выделить параметр `-i`, который задает количество байт информации на один индексный дескриптор, что в свою очередь определяет, сколько места на диске будет отдано под индексные дескрипторы. Так как для файловых систем ext количество индексных дескрипторов жестко задается при создании ФС, стоит сразу на этапе её создания определиться с их числом.

Создадим файловую систему ext4 на разделе диска `/dev/sdb` с параметрами по умолчанию:

```
localadmin@astra:~$ sudo mkfs.ext4 /dev/sdb2
mke2fs 1.44.5 (15-Dec-2018)
Creating filesystem with 195584 1k blocks and 48960 inodes
Filesystem UUID: 2205fa40-f724-4bcb-8ef2-d93e0786e4f4
Superblock backups stored on blocks:
       8193, 24577, 40961, 57345, 73729

Allocating group tables: done
Writing inode tables: done
Creating journal (4096 blocks): done
Writing superblocks and filesystem accounting information: done
```

Теперь немного расширим этот раздел (~100МБ) и расширим файловую систему.

```
localadmin@astra:~$ sudo parted /dev/sdb resizepart 2 400MiB
Information: You may need to update /etc/fstab.
localadmin@astra:~$ lsblk
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda      8:0    0   16G  0 disk
├─sda1   8:1    0   15G  0 part /
├─sda2   8:2    0    1K  0 part
└─sda5   8:5    0  975M  0 part [SWAP]
sdb      8:16   0    1G  0 disk
└─sdb2   8:18   0  294M  0 part
sdc      8:32   0    1G  0 disk
sdd      8:48   0    1G  0 disk
sde      8:64   0    2G  0 disk
sr0      11:0   1 1024M  0 rom
localadmin@astra:~$ sudo resize2fs /dev/sdb2
resize2fs 1.44.5 (15-Dec-2018)
Resizing the filesystem on /dev/sdb2 to 301056 (1k) blocks.
The filesystem on /dev/sdb2 is now 301056 (1k) blocks long.
```

19.6.2.2 Проверка целостности файловой системы

Для проверки целостности файловой системы служит команда `fsck` (*англ. file system check*). Для её успешной работы раздел не должен быть смонтирован. Запустим её для раздела `/dev/sdb2`:

```
localadmin@astra:~$ sudo fsck /dev/sdb2
fsck из util-linux 2.33.1
e2fsck 1.44.5 (15-Dec-2018)
/dev/sdb2: clean, 11/75480 files, 15692/301056 blocks
```

19.6.3 Графическая утилита GParted

С помощью графической утилиты `gparted`, имеющей интуитивно понятный интерфейс, можно как разметить диск, так и создать на его разделах файловые системы, а при необходимости изменить их размеры и атрибуты.

Утилита Gparted может быть запущена из меню *Пуск → Системные → Редактор разделов Gparted* или из командной строки (`sudo gparted`).

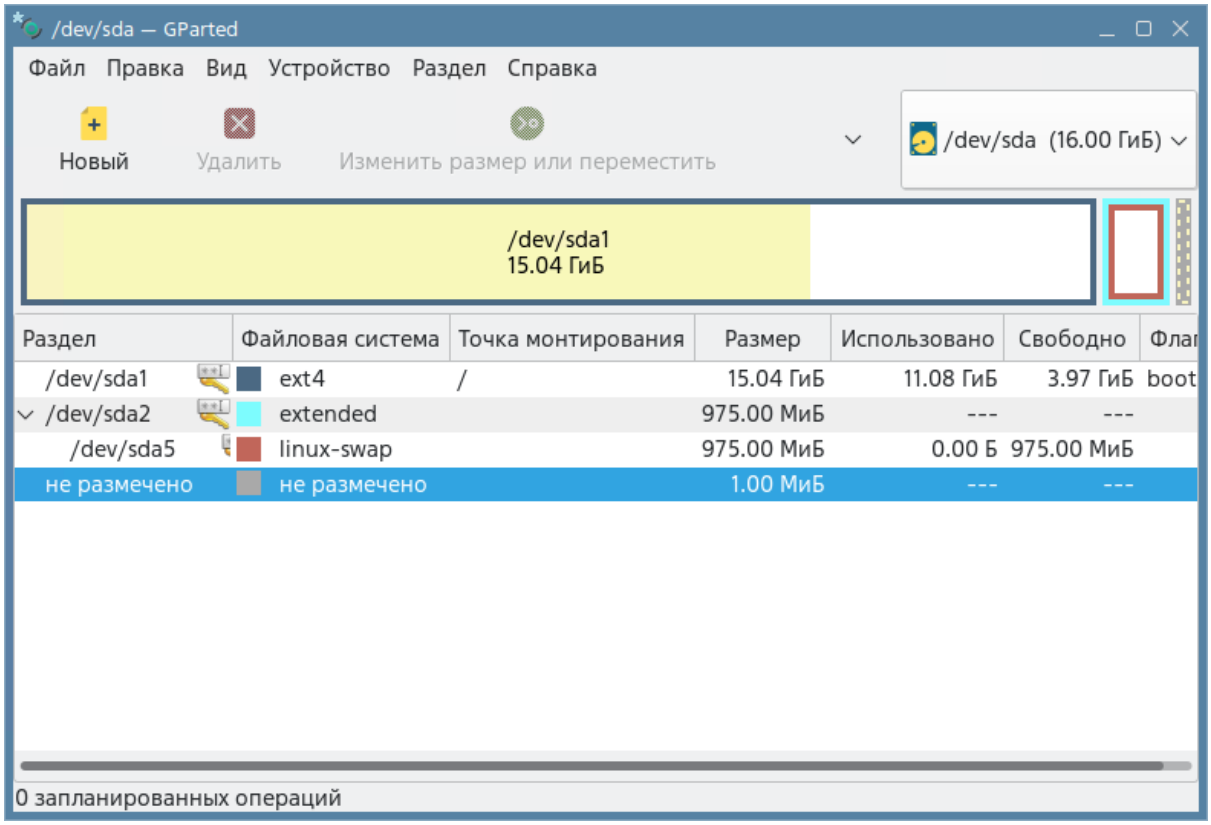
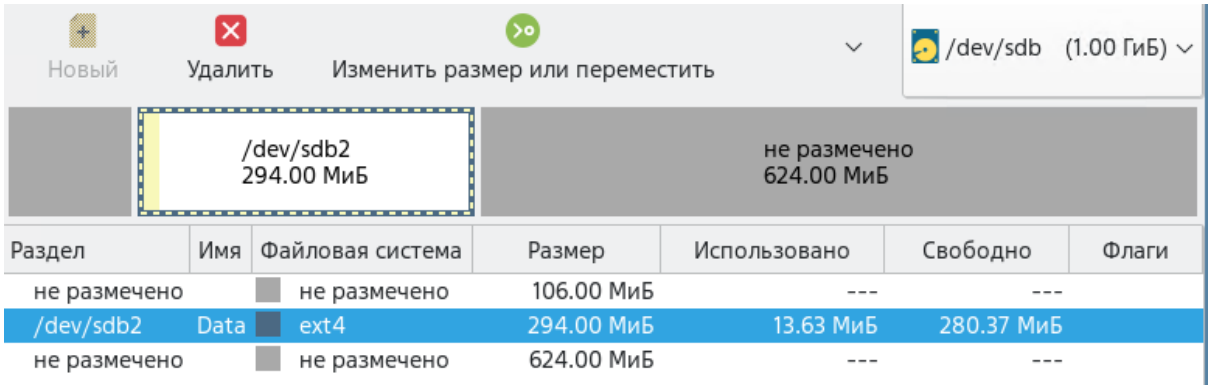
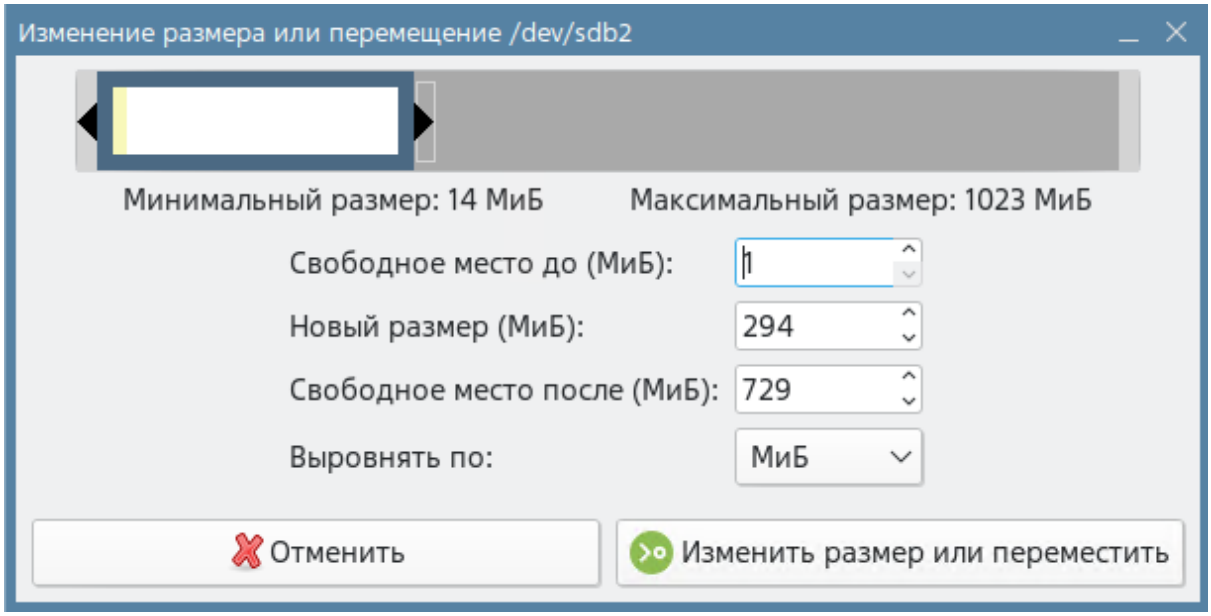


рис. 1.116 – Окно утилиты Gparted

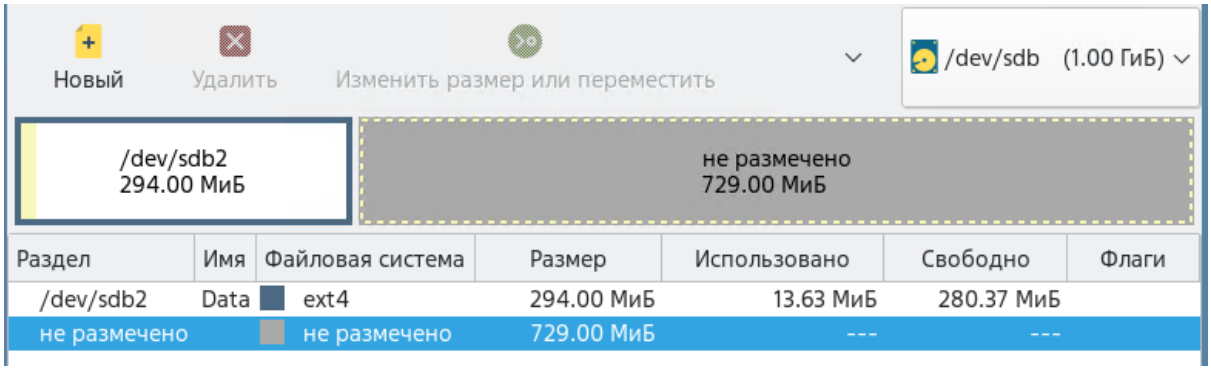
Выберем в правом верхнем углу диск `/dev/sdb`, в нем раздел sdb2, и применим действие *Изменить размер или переместить*



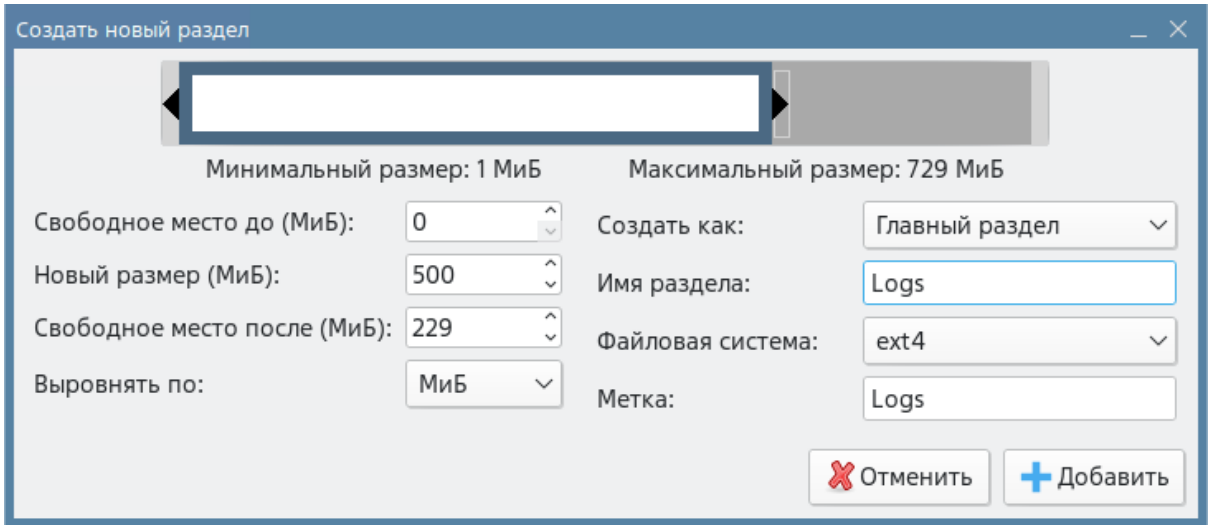
Переместим этот раздел в начало диска.



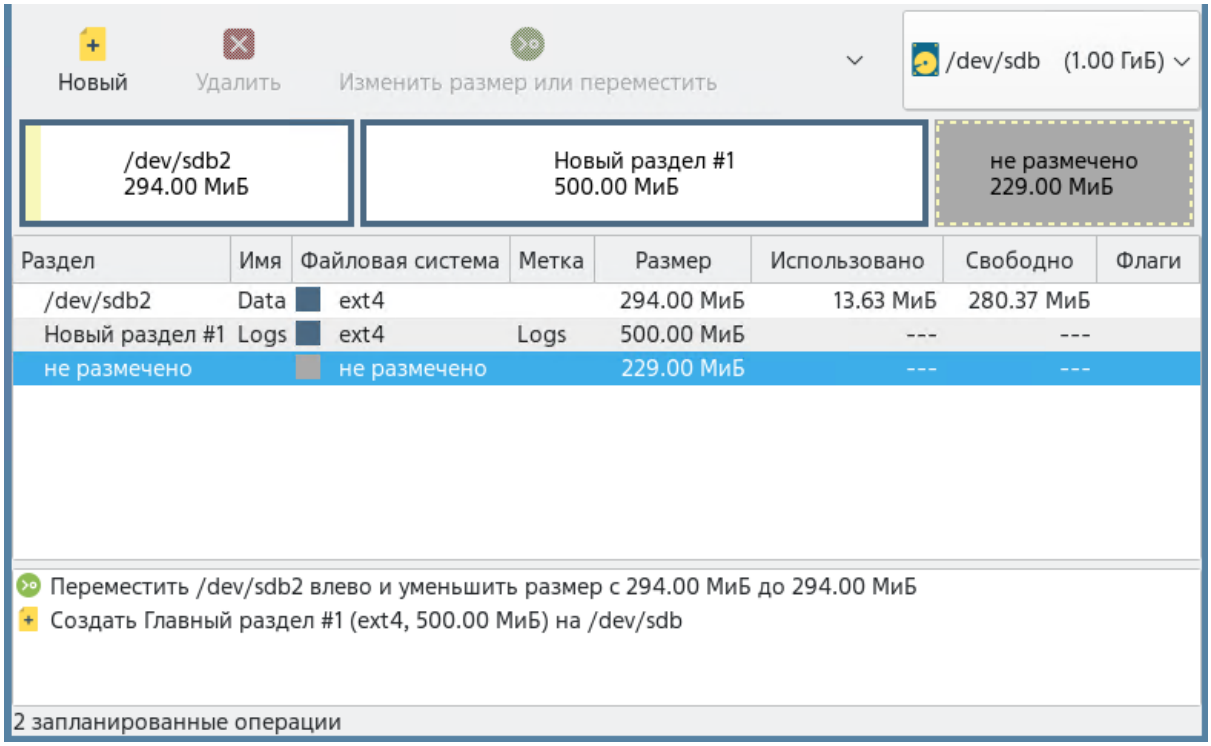
Согласимся с предупреждением утилиты.



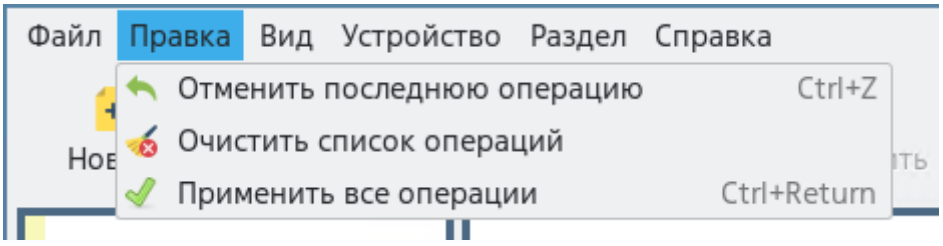
Теперь создадим новый раздел в свободной области диска размером в 500МБ.



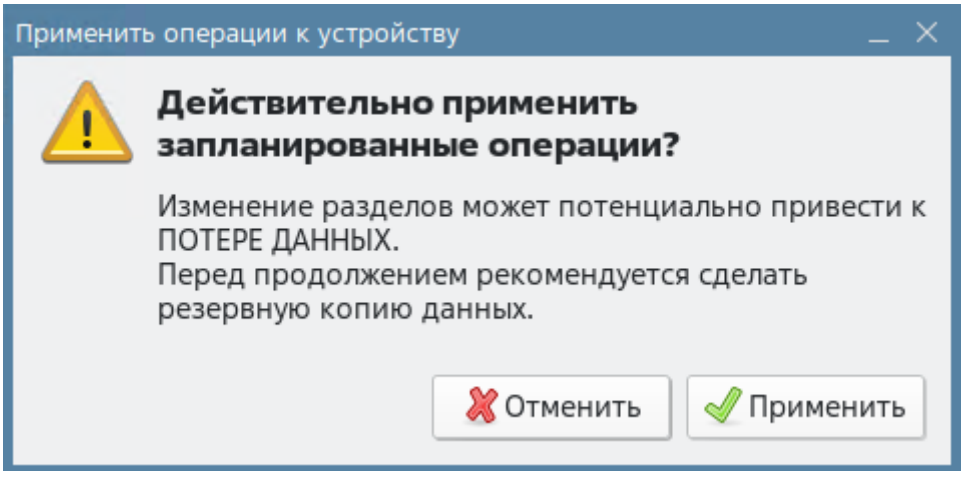
Раздел был создан и осталось свободное место.



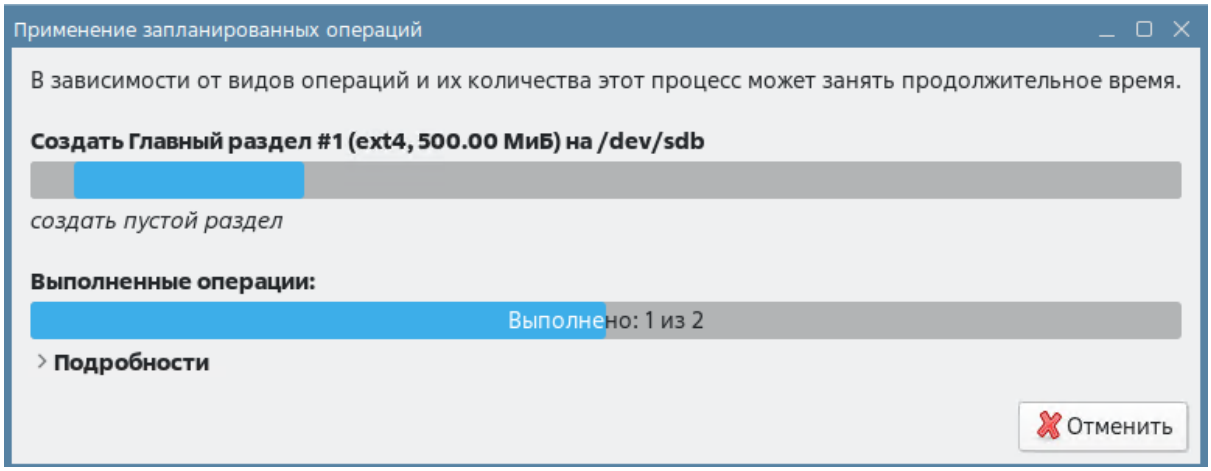
Заметьте, что фактических операций с диском еще не проводилось, они были только запланированы. Для применения изменений, необходимо выбрать пункт *Применить все операции* в меню *Правка*.

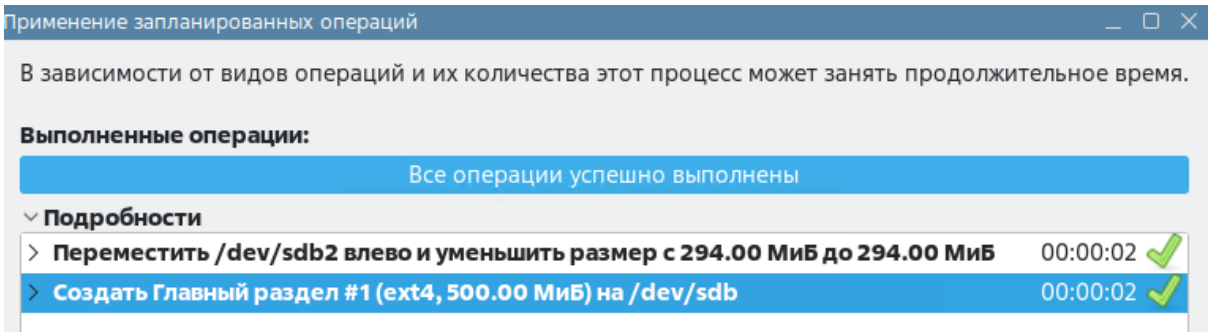


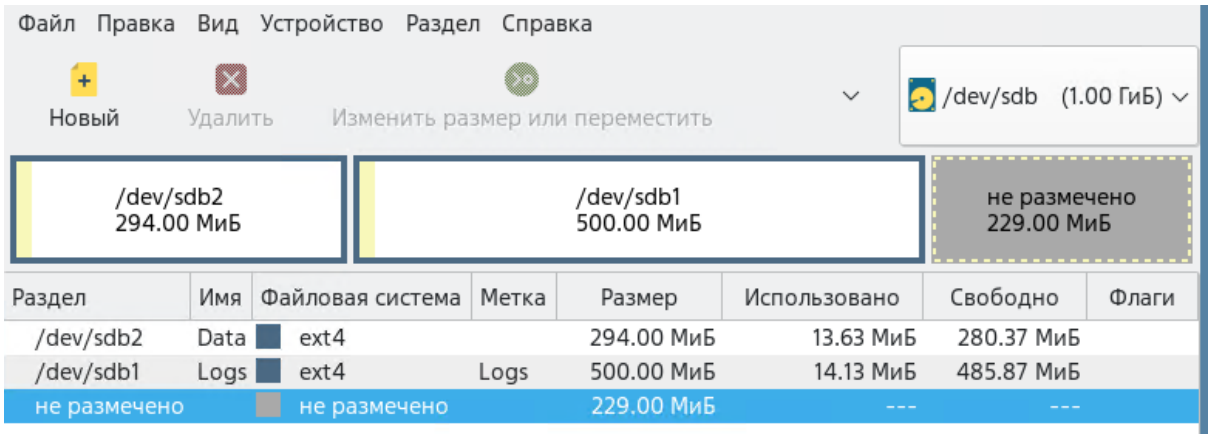
Применим операции, подтвердив их.



В процессе применения изменений вы будете видеть статус операции, а после их завершения отчет о выполнении.







Мы познакомились с тем, как в Linux можно разметить диски и создать на них файловые системы. Однако, для доступа к файловым системам этого недостаточно. Как в Windows мы не можем стандартными средствами обратиться к данным на диске, если ему не назначена буква, так и в Linux мы не можем обратиться к файловой системе, если она не была смонтирована. Поэтому, процессу монтирования ФС будет посвящен следующий раздел.

19.7 Монтирование файловых систем

Чтобы данные раздела диска стали доступны, этот раздел должен быть смонтирован в корневую файловую систему. Процесс монтирования файловой системы с точки зрения оператора ПК не очень сложен.

Существуют следующие виды монтирования:

- Временное через консольную команду `mount`.
- Постоянное через файл `/etc/fstab`.
- Монтирование с помощью службы systemd.

Опции монтирования:

- **auto** – файловая система монтируется автоматически.
- **ro** – монтируется в режиме «только чтение».
- **rw** – монтируется в режиме «чтение и запись».
- **dev** – файловая система может содержать файлы блочных и символьных устройств.
- **exec** – файловая система может содержать исполняемые файлы.
- **suid** – разрешено использование битов SUID и SGID.
- **user** – разрешено обычному пользователю размонтировать данную файловую систему и пр этом используются значения по умолчанию (noexec, nosuid, nodev).
- **defaults** – установки по умолчанию (rw, suid, dev, exec, nouser, async).
- **codepage=кодовая_страница** – применять указанную кодировку к именам файлов.
- **iocharset=набор_символов** – отображать имена файлов в соответствии с указанным набором символов.
- **noauto** – файловая система не может быть автоматически смонтирована.
- **nodev** – файловая система не может содержать файлы блочных и символьных устройств.
- **noexec** – файловая система не позволяет запускать исполняемые файлы.
- **nosuid** – запрещено использование битов SUID и SGID.
- **nouser** – обычному пользователю запрещено размонтировать данную файловую систему.

19.7.1 Временное монтирование

Временное монтирование выполняется утилитой `mount` и действует до очередной перезагрузки системы. Программа mount способна самостоятельно определить тип файловой системы, либо тип можно указать явно с помощью опции `-t`. Для определения типа файловой системы утилита mount использует библиотеку *blkid*.

19.7.1.1 Вывод информации о примонтированных ФС

Для вывода информации обо всех примонтированных в данный момент ФС и их параметров монтирования можно вызвать утилиту `mount` с ключом `-l` (строчная L) или без параметров вовсе.

```
localadmin@astra:~$ mount -l
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
udev on /dev type devtmpfs (rw,nosuid,relatime,size=962268k,nr_inodes=240567,mode=755,
↪inode64)
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000)
tmpfs on /run type tmpfs (rw,nosuid,noexec,relatime,size=202552k,mode=755,inode64)
/dev/sda1 on / type ext4 (rw,relatime,errors=remount-ro)
parsecfs on /parsecfs type parsecfs (rw,relatime,sync)
securityfs on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,relatime)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev,inode64)
tmpfs on /run/lock type tmpfs (rw,nosuid,nodev,noexec,relatime,size=5120k,inode64)
cgroup2 on /sys/fs/cgroup type cgroup2 (rw,nosuid,nodev,noexec,relatime,nsdelegate)
pstore on /sys/fs/pstore type pstore (rw,nosuid,nodev,noexec,relatime)
bpf on /sys/fs/bpf type bpf (rw,nosuid,nodev,noexec,relatime,mode=700)
systemd-1 on /proc/sys/fs/binfmt_misc type autofs (rw,relatime,fd=25,pgrp=1,timeout=0,
↪minp roto=5,maxproto=5,direct,pipe_ino=17710)
mqueue on /dev/mqueue type mqueue (rw,relatime)
hugetlbfs on /dev/hugepages type hugetlbfs (rw,relatime,pagesize=2M)
debugfs on /sys/kernel/debug type debugfs (rw,relatime)
configfs on /sys/kernel/config type configfs (rw,relatime)
fusectl on /sys/fs/fuse/connections type fusectl (rw,relatime)
tmpfs on /run/user/107 type tmpfs (rw,nosuid,nodev,relatime,size=202548k,mode=700,
↪uid=107, gid=118,inode64)
tmpfs on /run/user/107 type tmpfs (rw,nosuid,noexec,relatime,size=202552k,mode=755,
↪inode64)
tmpfs on /run/user/1000 type tmpfs (rw,nosuid,nodev,relatime,size=202548k,mode=700,
↪uid=100
0,gid=1000,inode64)
tmpfs on /run/user/1000 type tmpfs (rw,nosuid,noexec,relatime,size=202552k,mode=755,
↪inode64)
```

Для отображения информации только о реальных файловых системах можно воспользоваться фильтрацией вывода через утилиту `grep mount | grep "^/dev"`.

```
localadmin@astra:~$ mount | grep "^/dev"
/dev/sda1 on / type ext4 (rw,relatime,errors=remount-ro)
```

Более наглядный вывод дает утилита `findmnt`. Чтобы с ее помощью получить список реальных

файловых систем, можно использовать ее с ключом `findmnt --real`.

```
localadmin@astra:~$ findmnt --real
TARGET        SOURCE    FSTYPE    OPTIONS
/             /dev/sda1 ext4      rw,relatime,errors=remount-ro
├─/sys/fs/bpf bpf       bpf       rw,nosuid,nodev,noexec,relatime,mode=700
└─/parsecfs   parsecfs  parsecfs  rw,relatime,sync
```

```
localadmin@astra:~$ findmnt
TARGET                          SOURCE     FSTYPE        OPTIONS
/                               /dev/sda1  ext4          rw,relatime,errors=remount-ro
├─/sys                          sysfs      sysfs         rw,nosuid,nodev,noexec,relatime
│ ├─/sys/kernel/security        securityfs security      rw,nosuid,nodev,noexec,relatime
│ ├─/sys/fs/cgroup              cgroup2    cgroup2       rw,nosuid,nodev,noexec,
↪relatime,nsdelegat
│ ├─/sys/fs/pstore              pstore     pstore        rw,nosuid,nodev,noexec,relatime
│ ├─/sys/fs/bpf                 bpf        bpf           rw,nosuid,nodev,noexec,
↪relatime,mode=700
│ ├─/sys/kernel/debug           debugfs    debugfs       rw,relatime
│ ├─/sys/kernel/config          configfs   configfs      rw,relatime
│ └─/sys/fs/fuse/connections    fusectl    fusectl       rw,relatime
├─/proc                         proc       proc          rw,nosuid,nodev,noexec,relatime
│ └─/proc/sys/fs/binfmt_misc    systemd-1  autofs        rw,relatime,fd=25,pgrp=1,
↪timeout=0,minpro
├─/dev                          udev       devtmpfs      rw,nosuid,relatime,
↪size=962268k,nr_inodes
│ ├─/dev/pts                    devpts     devpts        rw,nosuid,noexec,relatime,
↪gid=5,mode=620,
│ ├─/dev/shm                    tmpfs      tmpfs         rw,nosuid,nodev,inode64
│ ├─/dev/mqueue                 mqueue     mqueue        rw,relatime
│ └─/dev/hugepages              hugetlbfs  hugetlbf      rw,relatime,pagesize=2M
├─/run                          tmpfs      tmpfs         rw,nosuid,noexec,relatime,
↪size=202552k,mo
│ ├─/run/lock                   tmpfs      tmpfs         rw,nosuid,nodev,noexec,
↪relatime,size=5120
│ ├─/run/user/1000              tmpfs      tmpfs         rw,nosuid,nodev,relatime,
↪size=202548k,mod
│ │ └─/run/user/1000            tmpfs[/user/private/1000/l0i0c0x0t0x0]
│ │                                        tmpfs         rw,nosuid,noexec,relatime,
↪size=202552k,mo
│ └─/run/user/107               tmpfs      tmpfs         rw,nosuid,nodev,relatime,
↪size=202548k,mod
│ └─/run/user/107               tmpfs[/user/private/107/l0i0c0x0t0x0]
│                                          tmpfs         rw,nosuid,noexec,relatime,
↪size=202552k,mo
└─/parsecfs                     parsecfs   parsecfs      rw,relatime,sync
```

Вызов команды `findmnt` с опцией -p позволит следить за процессом монтирования и размонтирования в реальном времени. Откроем новую вкладку в терминале и запустим там эту команду `findmnt -p`.

19.7.1.2 Создание точки монтирования

Создадим в каталоге `/mnt` подкаталог data и примонтируем в него устройство `/dev/sdb2` с опциями по умолчанию. Синтаксис команды `mount` при монтировании:

`sudo mount [<опции>] <устройство> <точка_монтирования>`.

```
localadmin@astra:~$ sudo mkdir /mnt/data
localadmin@astra:~$ sudo mount /dev/sdb2 /mnt/data
localadmin@astra:~$ ls /mnt/data
lost+found
```

Теперь создадим файл test.txt в /mnt/data и наполним его содержимым.

```
localadmin@astra:~$ sudo -i
root@astra:~# echo "Съешь еще этих булок, да выпей чаю." > /mnt/data/test.txt
root@astra:~# exit
выход
localadmin@astra:~$ cat /mnt/data/test.txt
Съешь еще этих булок, да выпей чаю.
```

19.7.1.3 Перемонтированные устройства

Теперь перемонтируем файловую систему с опцией только на чтение. Для этого выполним команду:

```
localadmin@astra:~$ sudo mount -o remount,ro /dev/sdb2 /mnt/data
```

Попробуем прочитать файл `/mnt/data/test.txt` и внести в него изменения: `sudo vim /mnt/data/test.txt`. Как мы можем убедиться, теперь даже из-под суперпользователя мы не можем вносить изменения в эту файловую систему. Однако, журнал для ФС ext3 и ext4 может все еще работать. Для отключения возможности записи и в журнал, необходимо добавить при монтировании еще одну опцию **noload**.

19.7.1.4 Размонтирование устройства

Для размонтирования устройства служит команда `sudo umount <точка_монтирования>` или `sudo umount <устройство>`. Размонтируем устройство /dev/sdb2 `sudo umount /dev/sdb2`.

```
localadmin@astra:~$ sudo umount /dev/sdb2
```

Откроем вкладку, на которой был запущен процесс `findmnt -p`, и ознакомимся с выполненными операциями.

```
localadmin@astra:~$ findmnt -p
ACTION          TARGET         SOURCE                                    FSTYPE  OPTIONS
перемонтировать /run/user/107  tmpfs[/user/private/107/l0i0c0x0t0x0]   tmpfs   rw,
↪nosuid,noe
```

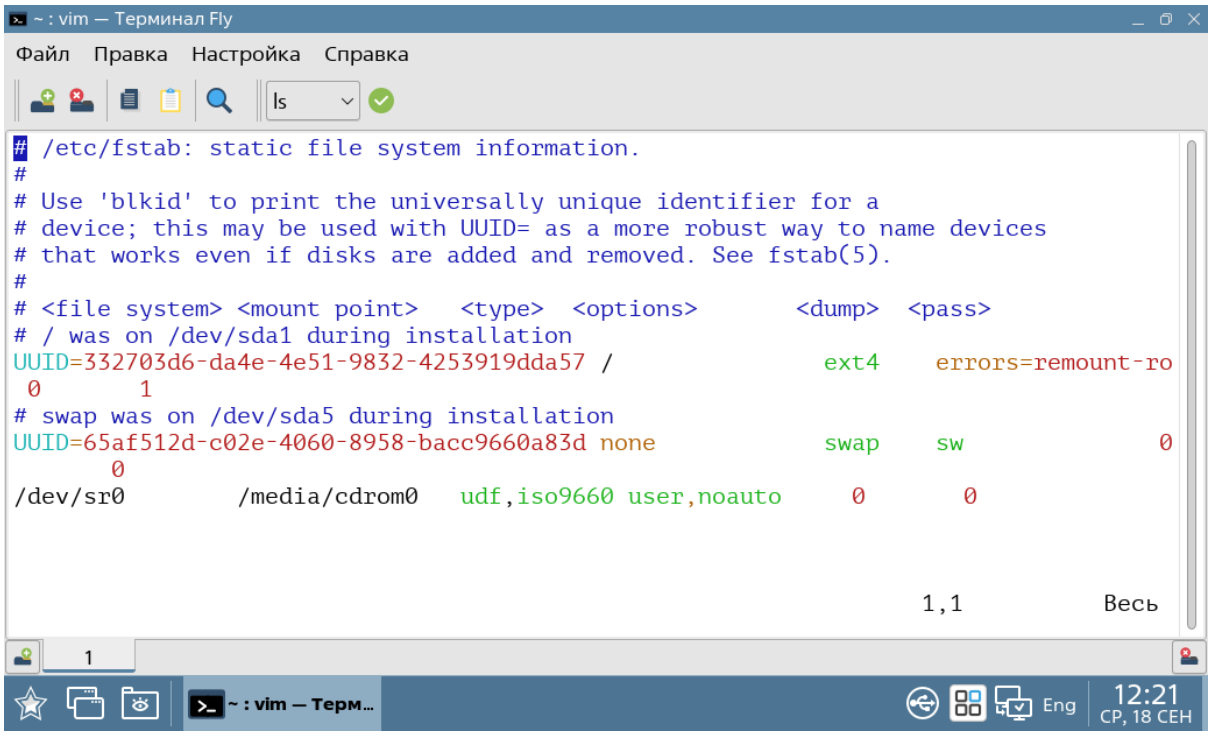
(continues on next page)

(продолжение с предыдущей страницы)

```
перемонтировать /run/user/1000 tmpfs[/user/private/1000/l0i0c0x0t0x0]  tmpfs   rw,
↪nosuid,noe
монтировать     /mnt/data      /dev/sdb2                               ext4    rw,
↪relatime
перемонтировать /run/user/107  tmpfs[/user/private/107/l0i0c0x0t0x0]   tmpfs   rw,
↪nosuid,noe
перемонтировать /run/user/1000 tmpfs[/user/private/1000/l0i0c0x0t0x0]  tmpfs   rw,
↪nosuid,noe
перемонтировать /mnt/data      /dev/sdb2                               ext4    ro,
↪relatime
перемонтировать /run/user/107  tmpfs[/user/private/107/l0i0c0x0t0x0]   tmpfs   rw,
↪nosuid,noe
перемонтировать /run/user/1000 tmpfs[/user/private/1000/l0i0c0x0t0x0]  tmpfs   rw,
↪nosuid,noe
размонтировать /mnt/data       /dev/sdb2                               ext4    ro,
↪relatime
```

19.7.2 Постоянное монтирование

Для постоянного монтирования (или долговременного) необходимо вносить изменения в конфигурационный файл `/etc/fstab`. Посмотрим его содержимое:



Формат файла представляет из себя таблицу с полями, разделенными пробелами. Строки, начинающиеся с символа решетки, являются комментариями, а пустые строки – игнорируются.

Всего используется шесть полей:

- **File system (fs_spec)** -– имя, метка (LABEL) или идентификатор (UUID) устройства. Указание

метки или идентификатора предпочтительнее, так как порядок обнаружения оборудования может быть изменен, что приведет к изменению имени устройства.

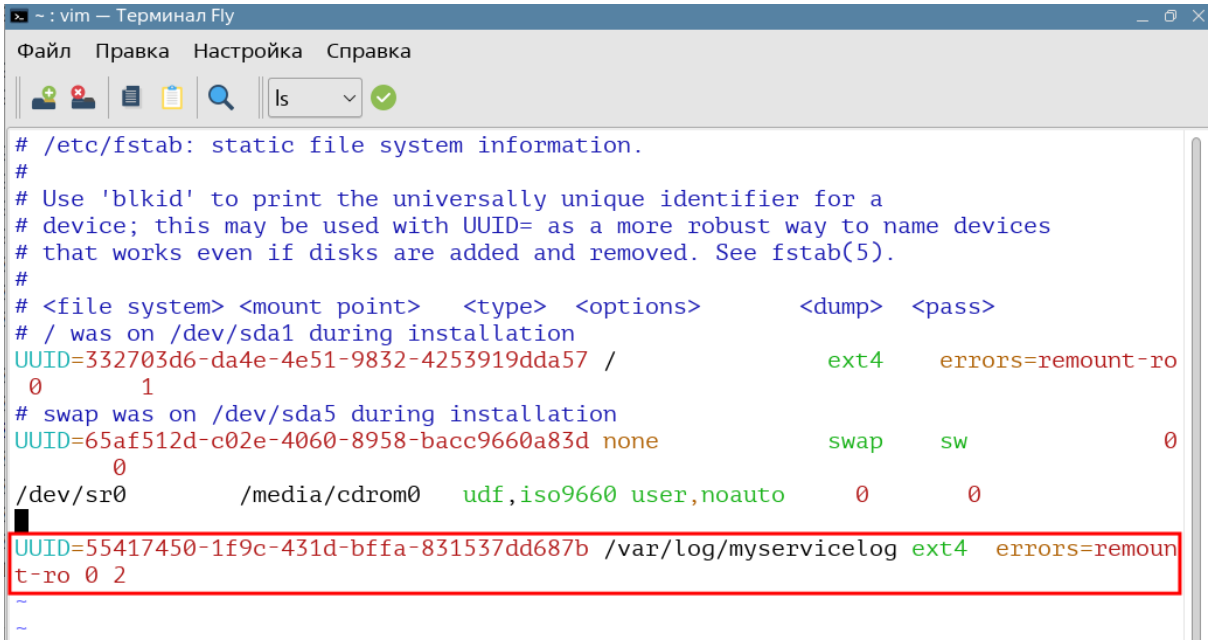
- **Mount point (fs_file)** –- точка монтирования (если нет, то значение none, как например у раздела подкачки). Если точка монтирования содержит пробелы или табуляции, их можно использовать как «\040» и «\011» соответственно.
- **Type (fs_vfstype)** – предполагаемый тип файловой системы. Можно указать несколько значений через запятую.
- **Options (fs_mntops)** -– параметры монтирования.
- **Dump (fs_freq)** –- используется утилитой dump. Определяет, какие файлы системы нужно выгружать при дампе. По умолчанию 0 (не выгружать).
- **Pass (fs_passno)** –- используется утилитой fsck. Определяет порядок проверки файловых систем при вызове утилиты fsck. Для корневой ФС этот параметр должен быть 1. Для остальных ФС равен 2 или 0 (не проверять). Системы на разных дисках будут проверяться параллельно, а на одном диске – последовательно.

Создадим постоянное монтирование для устройства `/dev/sdb1`. Для этого:

1. Создадим каталог `/var/log/myservicelog`, в который мы будем монтировать это устройство `sudo mkdir /var/log/myservicelog`.
2. Затем нам необходимо узнать UUID блочного устройства `/dev/sdb`. Это можно сделать, используя команду `lsblk -f | grep sdb1`.

```
localadmin@astra:~$ lsblk -f | grep sdb1
├─sdb1   ext4   Logs  55417450-1f9c-431d-bffa-831537dd687b
```

3. Добавим в файл `/etc/fstab` новую строку «*UUID=<UUID устройства> /var/log/myservicelog ext4 errors=remount-ro 0 2*»



4. Сохраним файл `/etc/fstab`.

5. Проверим, может ли система быть смонтирована, с помощью команды `sudo mount -a`, которая монтирует все файловые системы, перечисленные в файле `/etc/fstab`, за исключением тех, у которых задана опция noauto. Проверим, было ли примонтировано устройство `/dev/sdb1` `mount | grep "^/dev"`.

```
localadmin@astra:~$ sudo mount -a
localadmin@astra:~$ mount | grep "^/dev"
/dev/sda1 on / type ext4 (rw,relatime,errors=remount-ro)
/dev/sdb1 on /var/log/myservicelog type ext4 (rw,relatime,errors=remount-ro)
```

6. Как видим, устройство успешно смонтировалось. Теперь перезагрузим систему, и убедимся, что и после перезагрузки оно доступно.

19.8 Менеджер логических томов LVM

Менеджер логических томов LVM (*англ. Logical Volume Manager*) – это подсистема управления логическими томами, позволяющая использовать разные области одного жесткого диска и/или области с разных жестких дисков как один логический том.

С большой натяжкой аналогом LVM в Windows можно назвать динамические диски, которые тоже позволяют создавать программные RAID массивы и объединять несколько разделов на разных физических носителях в один логический раздел.

Примечание: LVM основан на Device mapper, которая используется также в реализации программных RAID-массивов, системы шифрования дисков dm-crypt и создания снимков файловой системы.

Основные преимущества LVM:

- Одну группу логических томов можно создавать поверх любого количества физических разделов.
- Размер логических томов можно легко менять прямо во время работы.
- LVM поддерживает механизм снимков состояний (*англ. snapshot*), копирование разделов «на лету» и зеркалирование, подобное RAID-1 и чередующуюся запись, подобно RAID-0.

К сожалению, получая преимущества приходиться чем-то жертвовать, и LVM не исключение. Удобство и гибкость при работе с LVM мы разменяваем на следующие недостатки:

- Работа дополнительно «прослойки» и уровня абстракции требует дополнительных накладных расходов, что может снизить производительность дисковой подсистемы (в среднем потери можно оценить в 5-10%)
- При использовании LVM на нескольких дисках, при потере одного из них, все данные хранимые на LVM будут утеряны.
- Начальная настройка LVM более сложна, чем просто разбиение диска. Необходимо понимать терминологию и модель LVM (логические тома, физические тома, группы томов), прежде чем вы сможете начать его использовать.

19.8.1 Структура и состав LVM

На рисунке ниже (рис. 1.117) представлена схема структуры LVM и её взаимодействие с физическими устройствами, разделами и файловыми системами.

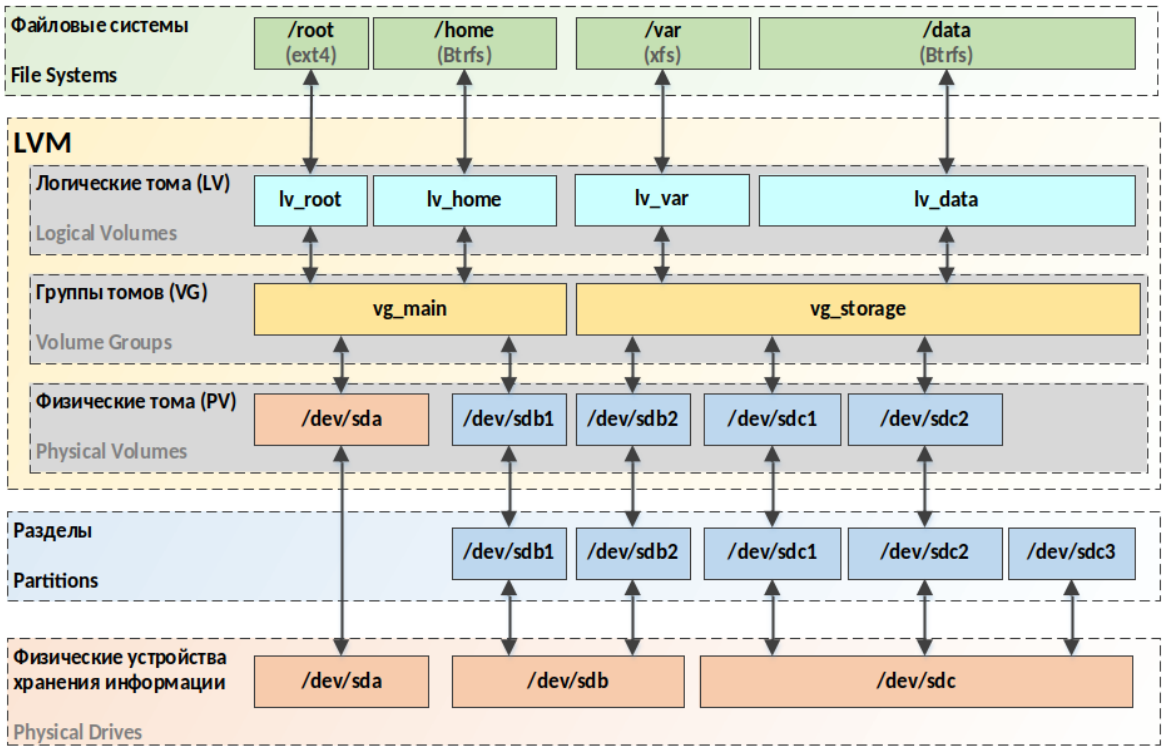


рис. 1.117 – Схема структуры LVM

В состав LVM входят следующие компоненты:

- **Физический том** (*англ. Physical Volume*, PV) –- раздел на диске или весь диск. В том числе, устройства программного и аппаратного RAID (которые уже могут включать в себя несколько физических дисков). Физические тома входят в группу томов.

- **Группа томов** (*англ. Volume Group, VG*) –- это самый верхний уровень абстрактной модели, используемой системой LVM. С одной стороны, группа томов состоит из физических томов, с другой – из логических и представляет собой единую административную единицу.

- **Логический том** (*англ. Logical Volume, LV*) –- раздел группы томов, эквивалентен разделу диска в не-LVM системе. Представляет собой блочное устройство и, как следствие, может содержать файловую систему.

Кроме указанных понятий существуют также логический (LE, Logical extent) и физический (PE, Physical extent) экстенты (порции данных). Физические тома делятся на порции данных, равные размеру физическим экстентам, а логические тома на порции данных равные размеру логическим экстентам, размер логического экстента не меняется в пределах группы томов и равен нескольким мегабайтам. При этом существует два алгоритма отображения физических экстентов на логическом:

- **Линейное отображение**, когда физические экстенты последовательно назначаются логическим

экстентам.

- **Расслоенное отображение** (*англ. stripped*), когда порции данных логических экстентов разделяют на определенное количество физических томов (похожая схема используется в RAID 0). Это может увеличить производительность работы логического тома, однако логический том с таким отображением не может быть расширен за пределы физических томов, на которых он изначально был создан.

19.8.2 Управление LVM

Для работы с LVM необходимо установить пакет lvm2, который по умолчанию не установлен: `sudo apt install lvm2 -y`. Как вы можете заметить, при этом создаётся новые сервисы и образ Initrd включающий необходимые модули для работы с LVM. В своих экспериментах мы воспользуемся ранее добавленными дисками `/dev/sdc`, `/dev/sdd` и `dev/sde`.

Полный процесс создания LVM и подготовки её к использованию можно представить шагами:

- Создание физических томов
- Создание групп томов
- Создание логических томов
- Создание ФС на логических томах
- Монтирование ФС

Так как ранее в модуле мы уже рассмотрели создание ФС и монтирование, сосредоточимся на первых 3 шагах.

Внимание: Стоит отметить, что все изменения с LVM, на логических томах которых созданы файловыми системами должны выполняться с размонтированными ФС.

19.8.2.1 Управление физическими томами

Создание физических томов

Для создания физического тома PV (инициализации целого диска или его раздела) используем команду:

`sudo pvcreate <устройство1> [<устройство2> <...> <устройствоN>]`

Где:

- **устройство1** .. **устройство<N>** — это имя устройства целиком (например, `/dev/sdb`) или его раздела (например, `/dev/sdb1`).

Инициализируем `/dev/sdc` командой `sudo pvcreate /dev/sdc`.

```
localadmin@astra:~$ sudo pvcreate /dev/sdc
Cannot use /dev/sdc: device is partitioned
```

Однако, мы получим ошибку, так как ранее разметили этот диск под GPT. Мы можем удалить таблицу разделов выполнив следующие команды:

```
localadmin@astra:~$ sudo dd if=/dev/zero of=/dev/sdc bs=1k count=1
1+0 записей получено
1+0 записей отправлено
1024 байт (1,0 kB, 1,0 KiB) скопирован, 0,020706 s, 49,5 kB/s
localadmin@astra:~$ sudo blockdev --rereadpt /dev/sdc
```

Повторим попытку `sudo pvcreate /dev/sdc`.

```
localadmin@astra:~$ sudo pvcreate /dev/sdc
 Physical volume "/dev/sdc" successfully created.
```

Теперь создадим два раздела на устройстве `/dev/sdd` по 500MB каждый и инициализируем их.

```
localadmin@astra:~$ sudo parted -s /dev/sdd mklabel gpt mkpart lvm1 ext4 1MiB 500MiB␣
↪mkpart lvm2 ext4 501MiB 1000MiB
localadmin@astra:~$ lsblk /dev/sdd
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sdd      8:48   0    1G 0  disk
├─sdd1   8:49   0  499M 0  part
└─sdd2   8:50   0  499M 0  part
localadmin@astra:~$ sudo pvcreate /dev/sdd1 /dev/sdd2
Physical volume "/dev/sdd1" successfully created.
Physical volume "/dev/sdd2" successfully created.
```

Вывод информации о физических томах

Для вывода информации о физических томах существуют следующие команды:

- Команда `pvs` –- позволяет настроить формат вывода, показывая по одному тому в каждой строке.
- Команда `pvdisplay` –- формирует подробный отчет для каждого физического тома, включая информацию о размере, экстентах, группе томов и пр. Формат вывода фиксирован.
- Команда `pvscan` –- проверяет все поддерживаемые блочные устройства в системе на предмет наличия физических томов.

```
localadmin@astra:~$ sudo pvs
PV          VG Fmt  Attr PSize   PFree
/dev/sdc       lvm2 ---    1,00g   1,00g
/dev/sdd1      lvm2 ---  499,00m 499,00m
/dev/sdd2      lvm2 ---  499,00m 499,00m

localadmin@astra:~$ sudo pvscan
PV /dev/sdc                       lvm2 [1,00 GiB]
PV /dev/sdd1                      lvm2 [499,00 MiB]
PV /dev/sdd2                      lvm2 [499,00 MiB]
Total: 3 [1,97 GiB] / in use: 0 [0 ] / in no VG: 3 [1,97 GiB]

localadmin@astra:~$ sudo pvdisplay
"/dev/sdc" is a new physical volume of "1,00 GiB"
--- NEW Physical volume ---
PV Name               /dev/sdc
VG Name
PV Size               1,00 GiB
Allocatable           NO
PE Size               0
Total PE              0
Free PE               0
Allocated PE          0
PV UUID               NO8oKl-ISSX-dfKy-Sdpz-KFIY-jPmF-oSr9Y4

"/dev/sdd1" is a new physical volume of "499,00 MiB"
--- NEW Physical volume ---
PV Name               /dev/sdd1
VG Name
PV Size               499,00 MiB
Allocatable           NO
PE Size               0
Total PE              0
Free PE               0
Allocated PE          0
PV UUID               5MFR0o-hFIE-x3w7-xLTm-1JLG-7WFa-xd99Bu

"/dev/sdd2" is a new physical volume of "499,00 MiB"
--- NEW Physical volume ---
PV Name               /dev/sdd2
VG Name
PV Size               499,00 MiB
Allocatable           NO
PE Size               0
Total PE              0
Free PE               0
Allocated PE          0
PV UUID               Uqh3LM-WaFA-8KLT-qNLx-goQ9-Jhj7-BKlYGo
```

Удаление физических томов

Для удаления физических томов используется команда `pvremove <имя_PV>`. Удалим PV `/dev/sdd2` `sudo pvremove /dev/sdd2`.

```
localadmin@astra:~$ sudo pvremove /dev/sdd2
 Labels on physical volume "/dev/sdd2" successfully wiped.
```

19.8.2.2 Управление группами томов

После создания физических томов мы можем создать группу томов. Соглашение об именовании предписывает начинать имя группы с префикса «vg_» и использовать либо порядковый номер, либо (что лучше) значащее имя, отражающее назначение этой группы.

Создание группы томов

Для создания группы томов служит команда:

`sudo vgcreate <имя_группы> <имя_физического_тома1> [<имя_физического_тома2> <имя_физического\_.`

Создадим группу томов «myservice» из томов `/dev/sdc` и `/dev/sdd1`.

```
localadmin@astra:~$ sudo vgcreate vg_myservice /dev/sdc /dev/sdd1
 Volume group "vg_myservice" successfully created
```

Вывод информации о группах томов

Для вывода информации о группах томов служат команды, аналогичные тем, что используются для вывода информации о физических томах:

- Команда `vgs` –- позволяет настроить формат вывода, показывая по одной группе в каждой строке.
- Команда `vgdisplay` -– формирует подробный отчет для каждой группы томов, включая информацию о размере, количестве физических и логических томов и пр. Формат вывода фиксирован.
- Команда `vgscan` –- проверяет все поддерживаемые дисковые устройства в системе на предмет наличия физических томов и групп томов.

```
localadmin@astra:~$ sudo vgs
VG            #PV #LV #SN Attr     VSize  VFree
vg_myservice    2   0   0 wz--n-   1,48g  1,48g

localadmin@astra:~$ sudo vgdisplay
--- Volume group ---
VG Name               vg_myservice
System ID
Format                lvm2
Metadata Areas        2
```

(continues on next page)

(продолжение с предыдущей страницы)

```
Metadata Sequence No   1
VG Access              read/write
VG Status              resizable
MAX LV                 0
Cur LV                 0
Open LV                0
Max PV                 0
Cur PV                 2
Act PV                 2
VG Size                1,48 GiB
PE Size                4,00 MiB
Total PE               379
Alloc PE / Size        0 / 0
Free PE / Size         379 / 1,48 GiB
VG UUID                oT3Rvd-e9po-ELF4-kpB9-VAZ2-0AVN-Oj1tk2

localadmin@astra:~$ sudo vgscan
Found volume group "vg_myservice" using metadata type lvm2
```

Изменение группы томов

Добавление томов в группу

Для добавления нового тома в группу томов используется команда:

`sudo vgextend <имя_группы> <имя_тома>`

Создадим раздел на устройстве `/dev/sde` и добавим его в группу томов «vg_myservice». При этом нам необязательно создавать физический том на устройстве, он будет создан автоматически.

```
localadmin@astra:~$ sudo parted -s /dev/sde mklabel gpt mkpart lvm1 ext4 1MiB 500MiB
localadmin@astra:~$ sudo vgextend vg_myservice /dev/sde1
 Physical volume "/dev/sde1" successfully created.
 Volume group "vg_myservice" successfully extended
```

Создадим еще один раздел и добавим его в группу.

```
localadmin@astra:~$ sudo parted /dev/sde mkpart lvm2 ext4 501MiB 1000MiB
Information: You may need to update /etc/fstab.

localadmin@astra:~$ sudo vgextend vg_myservice /dev/sde2
 Physical volume "/dev/sde2" successfully created.
  Volume group "vg_myservice" successfully extended
```

Удаление томов из группы

Для удаления тома из группы воспользуемся командой:

`sudo vgreduce <имя_группы> <имя_тома>`

Удалим из группы том `/dev/sde2` командой:

```
localadmin@astra:~$ sudo vgreduce vg_myservice /dev/sde2
 Removed "/dev/sde2" from volume group "vg_myservice
```

Переименование группы

Для переименования группы воспользуемся командой:

`vgrename <текущее_имя> <новое_имя>`

Переименуем группу в «vg_test»:

```
localadmin@astra:~$ sudo vgrename vg_myservice vg_test
 Volume group "vg_myservice" successfully renamed to "vg_test"
```

Удаление группы томов

Для удаления группы томов служит команда `sudo vgremove <имя_группы>`. Не будем удалять группу, она нам еще потребуется для создания логических томов.

19.8.2.3 Управление логическими томами

Создание логического тома

Для создания логического тома используется команда:

`sudo lvcreate [<опции>] <размер_тома> -n <имя_тома> <имя_группы>`

Где размер тома может быть задан несколькими способами:

- `-L<число>` - размер в мегабайтах;
- `-L<число><единицы_объема>` - размер в указанных единицах объема;
- `-l <число>` - размер в логических экстентах;
- `-l100%FREE` – весь объем группы.

Команда `sudo lvcreate -i2 -I4 -l100 -n <имя_тома> <имя_группы>` создаст логический том размером в 100 логических экстентов с расслоением по двум физическим томам и размером блока 4 KB.

Выведем информацию о количестве физических экстентов в созданной нами группе томов:

```
localadmin@astra:~$ sudo vgdisplay vg_test | grep "Total PE"
 Total PE 503
```

И создадим логический том «vl_test» размером в 300 экстентов:

```
localadmin@astra:~$ sudo lvcreate -l 300 -n vl_test vg_test
 Logical volume "vl_test" created.
```

Вывод информации о логических томах

И в этом случае команды аналогичны тем что использовались для физических томов и групп:

- Команда `lvs` – позволяет настроить формат вывода, показывая по одному тому в каждой строке.
- Команда `lvdisplay` – формирует подробный отчет для каждой группы томов, включая информацию о размере, количестве физических и логических томов и пр. Формат вывода фиксирован.
- Команда `lvscan` – проверяет все поддерживаемые дисковые устройства в системе на предмет наличия физических томов, групп томов и логических томов.

```
localadmin@astra:~$ sudo lvs
 LV      VG      Attr       LSize Pool Origin Data% Meta% Move Log Cpy%Sync Convert
 vl_test vg_test -wi-a----- 1,17g

localadmin@astra:~$ sudo lvscan
ACTIVE             '/dev/vg_test/vl_test' [1,17 GiB] inherit

localadmin@astra:~$ sudo lvdisplay
--- Logical volume ---
LV Path                /dev/vg_test/vl_test
LV Name                vl_test
VG Name                vg_test
LV UUID                L0OukQ-w3Ul-j6Tb-qdZ2-3Kqq-aToX-ibJGFn
LV Write Access        read/write
LV Creation host, time  astra, 2023-09-06 12:53:48 +0300
LV Status              available
# open                 0
LV Size                1,17 GiB
Current LE             300
Segments               2
Allocation             inherit
Read ahead sectors     auto
- currently set to     256
Block device           253:0
```

Изменение логического тома

Переименование тома

Допустим, мы заметили, что допустили ошибку в имени тома. Переименуем том «vl_test» в «lv_test» с помощью команды:

```
localadmin@astra:~$ sudo lvrename /dev/vg_test/vl_test lv_test
 Renamed "vl_test" to "lv_test" in volume group "vg_test"
```

Изменение размера тома

Для увеличения логического тома служит команда:

`sudo lvextend <размер>|<на_сколько_увеличить_размер> <полное_имя_логического_тома>`

Размер задается как при создании тома (опции -l и -L), а для указания того, насколько требуется увеличить размер, используется выражение типа -L+1G или -l +50.

Увеличим размер логического тома на 25 экстентов:

```
localadmin@astra:~$ sudo lvextend -l +25 /dev/vg_test/lv_test
 Size of logical volume vg_test/lv_test changed from 1,17 GiB (300 extents) to <1,27␣
↪GiB
(325 extents).
 Logical volume vg_test/lv_test successfully resized.
```

Если использовать опцию `-r`, то наряду с увеличением размера тома его файловая система тоже будет автоматически расширена:

`sudo lvextend –r -l +25 /dev/vg_test/lv_test`.

Для уменьшения размера тома служит команда `lvreduce`, синтаксис которой подобен lvextend:

`sudo lvreduce <размер>|<на_сколько_уменьшить_размер> <полное_имя_логического_тома>`

Кроме lvextend и lvreduce есть команда `lvresize`, которая может использоваться как для увеличения, так и для уменьшения размера логического тома.

Удаление логического тома

Для удаления логического тома служит команда `lvremove`:

`sudo lvremove <полное_имя_логического_тома>`.

19.9 Типовой сценарий переноса данных на другой носитель

19.9.1 Описание сценария

При правильном планировании на этапе установки ОС риск столкнуться с потребностью переноса данных на другой носитель информации уменьшаются. Однако в процессе эксплуатации машины сценарий её использования может измениться, как может измениться и потребление ресурсов, в

частности дискового пространства.

Рассмотрим типовой сценарий, когда на этапе установки ОС корневая ФС была установлена в обычный раздел, а не LVM, а место на диске заканчивается. В ходе анализа мы выяснили, что основным потребителем места является каталог `/home`. Мы решили перенести его на другой носитель информации. При разметке диска будем использовать LVM для повышения гибкости в управлении, справедливо полагая, что в будущем нам возможно придётся еще расширять доступное пространство для точки монтирования `/home`.

Примечание: В сценарии, для простоты, опустим некоторые детали переноса данных в продуктивной среде. В частности, не будем проверять активные сессии или выставлять запрет на изменение данных в каталоге `/home` после их копирования на новый раздел и его монтирования. В продуктивной среде такие работы следует выполнять в выделенное окно для обслуживания системы.

19.9.2 Перенос данных на внешнюю ФС

На нашей машине осталось свободное место на дисках sdd и sde, воспользуемся ими. Для начала перенесем данные на внешнюю ФС и удалим исходные данные на устройстве с корневой ФС:

1. Создадим логический том lv_home на разделах `/dev/sdd2` и `/dev/sde2` и создадим на нем файловую систему ext4:

- Используем раздел `/dev/sdd2` в качестве физического тома LVM:

```
localadmin@astra:~$ sudo pvcreate /dev/sdd2
Physical volume "/dev/sdd2" successfully created.
```

- Создадим на физических томах `/dev/sdd2` и `/dev/sde2` группу томов LVM vg_home:

```
localadmin@astra:~$ sudo vgcreate vg_home /dev/sdd2 /dev/sde2
Volume group "vg_home" successfully created
```

- Создадим логический том lv_home в группе vg_home используя весь объем:

```
localadmin@astra:~$ sudo lvcreate --extents 100%FREE -n lv_home vg_home
Logical volume "vl_home" created.
```

- Создадим файловую систему ext4 на логическом томе lv_home:

```
localadmin@astra:~$ sudo mkfs.ext4 /dev/vg_home/lv_home
mke2fs 1.44.5 (15-Dec-2018)
Creating filesystem with 253952 4k blocks and 63488 inodes
Filesystem UUID: cb7a7da2-70d6-4e67-86b7-76a503466577
Superblock backups stored on blocks:
```

(continues on next page)

(продолжение с предыдущей страницы)

```
   32768, 98304, 163840, 229376

Allocating group tables: done
Writing inode tables: done
Creating journal (4096 blocks): done
Writing superblocks and filesystem accounting information: done
```

2. Удалим содержимое каталога `/home/localadmin/Загрузки` и скопируем содержимое каталога `/home` на логический том lv_home:

- Удалим содержимое каталога `/home/localadmin/Загрузки`, что бы объем данных в `/home` не превышал 1 ГБ:

```
localadmin@astra:~$ sudo rm -rf /home/localadmin/Загрузки/*``
```

- Выполним временное монтирование ФС с устройства `/dev/mapper/vg_home-lv_home` в каталог `/mnt/home`:

```
localadmin@astra:~$ sudo mkdir /mnt/home
localadmin@astra:~$ sudo mount /dev/mapper/vg_home-lv_home /mnt/home
```

- Скопируем содержимое каталога `/home` в `/mnt/home` с сохранением прав доступа. Для этого воспользуемся утилитой `cp` с ключом `-a`, предварительно установив её:

```
localadmin@astra:~$ sudo cp -a /home /mnt/home
localadmin@astra:~$ ls -l /mnt/home
```

```
localadmin@astra:~$ ls -l /mnt/home/
итого 16
drwx------ 24 localadmin localadmin 4096 авг 15 13:05 localadmin
drwx------  3 user1      user1      4096 апр 18 20:51 user1
drwx------  3 user2      user2      4096 апр 18 20:33 user2
drwx------  3 user3      user3      4096 апр 18 20:33 user3
```

3. Выполним временное монтирование ФС `/dev/mapper/vg_home-lv_home` в `/home`, а затем создадим постоянное монтирование:

- Выполним временное монтирование `/dev/mapper/vg_home-lv_home` в `/home`:

```
localadmin@astra:~$ sudo mount /dev/mapper/vg_home-lv_home /home
localadmin@astra:~$ ls -l /home
итого 16
drwx------ 24 localadmin localadmin 4096 авг 16 10:38 localadmin
drwx------  3 user1      user1      4096 апр 18 20:51 user1
drwx------  3 user2      user2      4096 апр 18 20:33 user2
drwx------  3 user3      user3      4096 апр 18 20:33 user3
```

- Создадим постоянное монтирование используя файл `/etc/fstab`, добавив в него строку: «

`/dev/mapper/vg_home-lv_home /home ext4 defaults 0 2`

```
localadmin@astra:~$ echo "/dev/mapper/vg_home-lv_home /home ext4 defaults 0 2" | sudo␣
↪tee -a /etc/fstab
/dev/mapper/vg_home-lv_home /home ext4 defaults 0 2
localadmin@astra:~$ tail -n 1 /etc/fstab
/dev/mapper/vg_home-lv_home /home ext4 defaults 0 2
```

- Удостоверимся, что все работает:

```
localadmin@astra:~$ ls -l /home
итого 16
drwx------ 24 localadmin localadmin 4096 авг 16 10:38 localadmin
drwx------  3 user1      user1      4096 апр 18 20:51 user1
drwx------  3 user2      user2      4096 апр 18 20:33 user2
drwx------  3 user3      user3      4096 апр 18 20:33 user3
```

Примечание: После монтирования внешней ФС в корневую ФС все новые данные записанные в эту точку монтирования будут фактически храниться на внешней ФС. Однако, такой операцией мы не освободим место на диске корневой ФС. После успешного переноса данных на внешнюю ФС и проверки корректности монтирования, исходные данные могут быть удалены.

Что бы удалить данные, например, из каталога `/home` на устройстве `/dev/sda1`, необходимо предварительно примонтировать это устройство в корневую ФС.

---

4. Выполним удаление содержимого каталога `/home` с устройства `/dev/sda1`:

- Проверим, сколько места свободно на устройстве /dev/sda1:

```
localadmin@astra:~$ df | grep /dev/sda1
/dev/sda1                   15421320     11233904  3382248          77% /
```

- Примонтируем устройство с корневой ФС `/dev/sda1` в точку монтирования `/mnt/rootfs`:

```
localadmin@astra:~# ls -l /mnt/rootfs
итого 16
drwx------ 24 localadmin localadmin 4096 авг 15 13:05 localadmin
drwx------  3 user1      user1      4096 апр 18 20:51 user1
drwx------  3 user2      user2      4096 апр 18 20:33 user2
drwx------  3 user3      user3      4096 апр 18 20:33 user3
```

- Выполним удаление содержимого `/mnt/rootfs/home`:

```
localadmin@astra:~$ cd /mnt/rootfs/home
localadmin@astra:/mnt/rootfs/home$ sudo rm -rf ./*
```

- Проверим, сколько места на устройстве `/dev/sda1` мы освободили `df | grep /dev/sda1`:

```
localadmin@astra:~$ df | grep /dev/sda1
/dev/sda1           15421320     10475144  4141008      72% /
```

19.9.2.1 Добавление места логическому тому и расширение ФС

Наша машина работала какое-то время, пользователи генерировали новые данные и сохраняли их у себя в домашних папках. Мы столкнулись с необходимостью расширить пространство у логического тома lv_home. Для этого нам необходимо вначале добавить новый физический том LVM и расширить группу томов vg_home:

1. Добавим нашей виртуальной машине еще один диск размером 1 ГБ:

```
localadmin@astra:~$ sudo lsblk
NAME                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda                   8:0    0   16G  0 disk
├─sda1                8:1    0   15G  0 part /
. . .
└─sde2                8:66   0  499M  0 part
└─vg_home-lv_home 253:1    0  992M  0 lvm  /home
sdf                   8:80   0    1G  0 disk
sr0                  11:0    1 1024M  0 rom
```

2. Добавим новый физический том sdf в группу томов LVM vg_home:

```
localadmin@astra:~$ sudo vgextend vg_home /dev/sdf
Physical volume "/dev/sdf" successfully created.
Volume group "vg_home" successfully extended
```

3. Проверим текущий размер логического тома lv_home:

```
localadmin@astra:~$ sudo lvscan | grep lv_home
ACTIVE            '/dev/vg_home/lv_home' [992,00 MiB] inherit
```

4. Расширим логический том lv_home на все доступное пространство группы томов lv_home:

```
localadmin@astra:~$ sudo lvextend -l +100%FREE /dev/vg_home/lv_home
Size of logical volume vg_home/lv_home changed from 992,00 MiB (248 extents) to 1,96␣
↪GiB (503 extents).
Logical volume vg_home/lv_home successfully resized.
```

Как видим, логический том был расширен, и пользователи могут продолжать работать и сохранять свои данные в домашних директориях.

Примечание: Точно таким же образом, как мы сейчас переносили директорию `/home`, можно перенести каталог с общей папкой на файловом сервере Samba. Подробно о файловом сервере мы с вами будем говорить во второй части курса, в модуле, посвященном дополнительным подсистемам ALD Pro.

19.10 Практика и тестирование

19.10.1 Практическая работа: Модуль 19. Файловая система в Linux, работа с блочными устройствами и LVM

См. также *Требования, правила и цели выполнения практической работы*

19.10.1.1 Практические задания

Задание 1.

- Добавьте к существующей виртуальной машине новый жесткий диск размером 500 Мб;
- Установите метку диска msdos утилитой parted в режиме командной строки;
- Создайте на подключенном жестком диске 2 раздела по 100 Мб утилитой parted в режиме командной строки;
- Создайте на новых разделах файловые системы EXT3 и XFS и смонтируйте их в каталоги /mnt/part1 и /mnt/part2;
- Создайте постоянное монтирование для указанных разделов со следующими условиями:
  - /mnt/part1 монтируется в режиме только для чтения;
  - /mnt/part2 монтируется с запретом на запуск исполняемых файлов;
- Перезагрузите ОС для проверки созданных настроек;
- После проверки настроек удалите созданные точки монтирования и разделы;

Задание 2.

- На диске /dev/sdb создайте раздел размером 150Мб;
- Создайте LVM physical volume (PV) на созданном разделе;
- Создайте LVM volume group (VG) с именем TEST_VG на созданном PV;
- Создайте LVM logical volume (LV) с именем TEST_LV на созданном VG и занимающий весь доступный объем;
- Создайте файловую систему EXT4 на созданном LV;
- На диске /dev/sdb создайте еще один раздел размером 150Мб;

- Выполните процедуру расширения имеющегося LVM раздел на размер /dev/sdb2 с расширением файловой системы;
- Проверьте что LV и файловая система были расширены и удалите созданные разделы;

19.10.1.2 Ответы на практические задания (пошаговые инструкции)

Задание 1.

Добавьте к существующей виртуальной машине новый жесткий диск размером 500 Мб;

Установите метку диска msdos утилитой parted в режиме командной строки;

```
localadmin@astra:~$ sudo parted /dev/sdb mklabel msdos
```

Создайте на подключенном жестком диске 2 раздела по 100 Мб утилитой parted в режиме командной строки;

```
localadmin@astra:~$ sudo parted /dev/sdb mkpart primary 4MB 100MB
localadmin@astra:~$ sudo parted /dev/sdb mkpart primary 100MB 200MB
```

Создайте на созданных разделах файловые системы EXT3 и XFS и смонтируйте их в каталоги /mnt/part1 и /mnt/part2;

```
localadmin@astra:~$ sudo mkfs.ext3 /dev/sdb1
localadmin@astra:~$ sudo mkfs.xfs /dev/sdb2
localadmin@astra:~$ sudo mkdir /mnt/part1
localadmin@astra:~$ sudo mkdir /mnt/part2
localadmin@astra:~$ sudo mount /dev/sdb1 /mnt/part1/
localadmin@astra:~$ sudo mount /dev/sdb2 /mnt/part2/
```

Создайте постоянное монтирование для созданных разделов со следующими условиями:

- /mnt/part1 монтируется в режиме только для чтения;

```
localadmin@astra:~$ sudo echo "/dev/sdb1 /mnt/part1 ext3 ro 0 0" >> /ets/fstab
```

- /mnt/part2 монтируется с запретом на запуск исполняемых файлов;

```
localadmin@astra:~$ sudo echo "/dev/sdb2 /mnt/part2 xfs noexec 0 0" >> /ets/fstab
```

- Перезагрузите ОС для проверки созданных настроек;
- После проверки настроек удалите созданные точки монтирования из файла /etc/fstab и созданные разделы;

```
localadmin@astra:~$ sudo vim /etc/fstab
localadmin@astra:~$ sudo parted /dev/sdb rm 1
localadmin@astra:~$ sudo parted /dev/sdb rm 2
```

Задание 2.

На диске /dev/sdb создайте раздел размером 150Мб;

```
localadmin@astra:~$ sudo parted /dev/sdb mkpart primary 4MB 154MB
```

Создайте LVM physical volume (PV) на созданном разделе;

```
localadmin@astra:~$ sudo pvcreate /dev/sdb1
```

Создайте LVM volume group (VG) с именем TEST_VG на созданном PV;

```
localadmin@astra:~$ sudo vgcreate TEST_VG /dev/sdb1
```

Создайте LVM logical volume (LV) с именем TEST_LV на созданном VG и занимающий весь доступный объем;

```
localadmin@astra:~$ sudo lvcreate -n TEST_LV -l 100%FREE TEST_VG
```

Создайте файловую систему EXT4 на созданном LV;

```
localadmin@astra:~$ sudo mkfs.ext4 /dev/TEST_VG/TEST_LV
```

На диске /dev/sdb создайте еще один раздел размером 150Мб;

```
localadmin@astra:~$ sudo parted /dev/sdb mkpart primary 154MB 304MB
```

Выполните процедуру расширения, имеющегося LVM раздел на размер /dev/sdb2 с расширением файловой системы;

```
localadmin@astra:~$ sudo pvcreate /dev/sdb2
localadmin@astra:~$ sudo vgextend TEST_VG /dev/sdb2
localadmin@astra:~$ sudo lvextend -l +100%FREE -r /dev/TEST_VG/TEST_LV
```

Проверьте что LV и файловая система были расширены и удалите созданные разделы;

```
localadmin@astra:~$ sudo parted /dev/sdb rm 1
localadmin@astra:~$ sudo parted /dev/sdb rm 2
```

19.10.2 Тест: Модуль 19. Файловая система в Linux, работа с блочными устройствами и LVM

Пожалуйста, ответьте на несколько вопросов, чтобы закрепить полученные знания. Правильные ответы для самопроверки вы сможете найти в конце списка. Обратите внимание, что у некоторых вопросов может быть несколько верных вариантов ответов.

19.10.2.1 Вопросы

Вопрос 1. Какая файловая система используется для монтирования USB-накопителей без привилегий суперпользователя?

a. EXT

b. FUSE

c. SMB

d. VFS

Вопрос 2. Какая команда позволяет получить список файловых систем, поддерживаемых ядром в данный момент?

a. Команда `lsblk`

b. Команда `ls /dev/{v,d}d*`

c. Команда `cat /proc/filesystems`

d. Команда `vgdisplay`

Вопрос 3. Какая директория хранит программные файлы, доступные только для чтения, такие как исполняемые файлы, библиотеки и прочее?

a. Директория `/usr`

b. Директория `/boot`

c. Директория `/`

d. Директория `swap`

Вопрос 4. Какая команда создает новый раздел диска в утилите `fdisk`?

a. Команда `p`

b. Команда `i`

c. Команда `m`

d. Команда `n`

Вопрос 5. За что отвечает опция `noexec` в файле `/etc/fstab`?

a. Файловая система может содержать исполняемые файлы

b. Файловая система монтируется автоматически

c. Файловая система не позволяет запускать исполняемые файлы

d. Файловая система не может содержать файлы блочных и символьных устройств

Вопрос 6. Расположите в правильном порядке шаги создания дисковых разделов LVM.

a. Создание групп томов > Создание физических томов > Создание логических томов > Создание ФС на логических томах > Монтирование ФС

b. Создание физических томов > Создание логических томов > Создание групп томов > Создание ФС на логических томах > Монтирование ФС

c. Создание групп томов > Создание физических томов > Создание логических томов > Монтирование ФС > Создание ФС на логических томах

d. Создание физических томов > Создание групп томов > Создание логических томов > Создание ФС на логических томах > Монтирование ФС

Вопрос 7. При расширении раздела LVM, после выполнения расширения логического раздела, необходимо выполнить расширение файловой системы на этом разделе отдельной командой. Можно ли одной командой расширить и раздел и ФС?

a. Нет, нельзя

b. Да, нужно поставить команды друг за другом через точку с запятой

c. Да, нужно добавить в команду расширения логического раздела опцию `-r`

Ответы на вопросы

1. b
2. c
3. a
4. d
5. c
6. d
7. c

19.11 Заключение

В этом модуле мы углубили свои познания в вопросах устройства и администрирования файловых систем в Linux. Мы научились создавать, форматировать и монтировать разделы дисков, узнали о лучших практиках разнесения каталогов по разделам. И последнее, но не маловажное — мы научились работать с менеджером логических томов LVM.

В следующем модуле мы перейдем к вопросам использования TCP/IP стека, что очень важно для сопровождения работы домена, т.к. все компоненты системы взаимодействуют между собой преимущественно по сети.

19.12 Дополнительные источники информации

- Отсутствуют

19.13 Обратная связь

Если остались вопросы, то их всегда можно задать в специальной теме.

Модуль 20. Сеть и стек TCP/IP, управление сетевыми интерфейсами

20.1 Введение

В этом модуле мы разберем, как работает сетевой стек TCP/IP вообще и в Linux в частности, что крайне важно для уверенного сопровождения инфраструктуры домена, где большинство компонентов взаимодействуют между собой по компьютерной сети даже в тех случаях, когда находятся на одном хосте.

Тема эта крайне специфичная, так как на обывательском уровне все знают, что такое эти Интернеты, а в базовых вопросах маршрутизации ориентируются уже только два админа из десяти. И дело тут даже не в наличии или отсутствии сертификатов CCNA/MTNA, так как в стране достаточно ребят, у кого с бумажками все хорошо, а со знаниями не очень.

Времени у нас совсем немного, поэтому по принципу «20 на 80» будем рассматривать в основном IPv4, останавливаясь только на самых основных моментах, которые считаем принципиально важными для понимания этой технологии. Легкой прогулки не обещаем, но высокая в небе звезда уже

зовет в путь.

20.2 Основы компьютерных сетей

20.2.1 История вопроса

Когда компьютеры только зародились, их было очень мало и о сетях никто не думал. Но по мере того, как вычислительная техника становилась доступнее, идеи организации обмена данными между ЭВМ начали набирать популярность. Сначала это были соединения типа «точка-точка», затем задумались о более сложных топологиях.

Значительную роль в развитии этих технологий сыграл военный проект ARPA от наших западных партнеров, целью которого было, конечно же, не создание сети военной связи, устойчивой к ядерному нападению, а совместное использование ресурсов в сфере сотрудничества между университетами, частными технологическими компаниями... и ребятами в погонах.

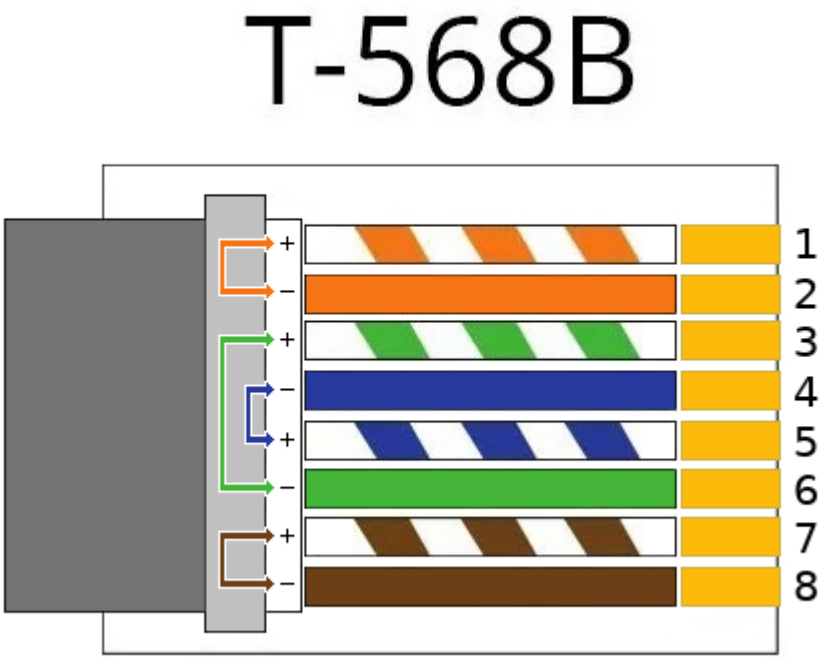
20.2.2 Объединяем два компьютера «точка-точка»

Для того чтобы связать два компьютера между собой, даже никакого сетевого оборудования не требуется, достаточно один конец патчкорда воткнуть в сетевую карту первого компьютера, а второй конец — в сетевую карту второго компьютера.

Раньше для соединения однотипных устройств нужен был хотя бы специальный кабель с перекрестной обжимкой, а теперь у всех сетевых карт порты Auto-MDI(X), поэтому они автоматически распознают тип кабеля и можно использовать обычные патчкорды с прямой обжимкой, которая предусматривает одинаковый порядок жил на обоих концах.

Примечание: Сисадмины иногда задаются вопросом: «А нужно ли запоминать правильный порядок цветов витой пары или достаточно обжать кабель одинаково с двух сторон?». И мы вам ответим, что правильный порядок, во-первых, уменьшает помехи, а во-вторых, исключает путаницу, когда на вашей стороне только один конец провода, а второй за стенкой у провайдера.

Для запоминания порядка цветов по стандарту 568B есть простое и универсальное мнемоническое правило: «Сто патчкордов обжимаем, и цвета запоминаем...». Работает не только с патчкордами)))



1. бело-оранжевый
2. оранжевый
3. бело-зеленый
4. синий
5. бело-синий
6. зеленый
7. бело-коричневый
8. коричневый

Теперь осталось только правильно выставить три основных сетевых настройки:

1. **IP-адрес** – это уникальный числовой идентификатор компьютера в сети, по которому один компьютер может обратиться к другому.

   По стандарту IPv4 адрес является последовательностью из 32 бит, которую обычно представляют в виде четырех целых чисел в диапазоне 0 — 255, разделенных точками, например, `192.168.1.1`. Учитывая, что эти числа кодируются восемью битами, их также называют иногда октетами.

2. **Маска** – это последовательность из 32-бит, с помощью которой компьютер определяет, находится ли «собеседник» в одной сети вместе с ним. Если IP-адрес пункта назначения принадлежит к той же сети, то компьютер будет отправлять пакеты напрямую в сетевой интерфейс. А если адрес будет из другой сети, то пакеты пойдут через шлюз.

   Особенность сетевой маски заключается в том, что это непрерывная последовательность единичек и нулей, например, `11111111 11111111 11111111 00000000`. Поэтому кроме полной записи `255.255.255.0`, к которой нас приучили в Windows, есть еще короткая запись **/24** (называемая префикс), где число соответствует количеству единичек в начале маски. В мире Linux маску часто указывают сразу вместе с IP-адресом, например, `192.168.1.1/24`.

   Учитывая, что единички маски указывают на то, какая часть IP-адреса будет определять его принадлежность к сети, короткую запись называют также префиксом сети. Для упрощения понимания приведем несколько примеров:

табл. 1.6 – Виды записей маски для разных подсетей

Маска сети	Префикс сети	Двоичная запись маски
0.0.0.0	/0	`00000000.00000000.00000000.00000000`
128.0.0.0	/1	`10000000.00000000.00000000.00000000`
192.0.0.0	/2	`11000000.00000000.00000000.00000000`
255.0.0.0	/8	`11111111.00000000.00000000.00000000`
255.255.0.0	/16	`11111111.11111111.00000000.00000000`
255.255.255.0	/24	`11111111.11111111.11111111.00000000`
255.255.255.255	/32	`11111111.11111111.11111111.11111111`

Примечание: Метод с использованием маски называют бесклассовой адресацией (англ. Classless Inter-Domain Routing, англ. CIDR), а до него использовалась классовая адресация, которая умерла еще в 90е годы ввиду своей неэффективности, но авторы некоторых статей и учебников до сих пор тащат эту «свежатину» про классы A, B, C, D, E, забивая людям голову ненужной информацией.

---

3. **Шлюз по умолчанию** – это адрес узла в компьютерной сети, через который компьютер может связываться с устройствами из других сетей, например, выходить в Интернет.

Для связи компьютер-компьютер маска и шлюз не имеют значения, т.к. шлюз вообще не используется, а маска по умолчанию будет длиной 24 бит, что вполне подойдет. Настроим указанную схему в графическом симуляторе сети GNS3 и проверим с помощью команды `ping` доступность второго компьютера, см. рис. 1.118.

Примечание: Кстати, приложение GNS3 является хорошей альтернативой симулятору Cisco Packet Tracer. С помощью этого приложения вы сможете создавать работоспособные модели сети, в том числе с использованием VLAN'ов, настраивать взаимодействие между виртуальными машинами, включать в состав сети программные маршрутизаторы, например, на базе RouterOS от Mikrotik. Если вы решите отточить свои навыки работы с сетями, то обязательно скачайте виртуальную машину с GNS3, и этот инструмент вам очень поможет!

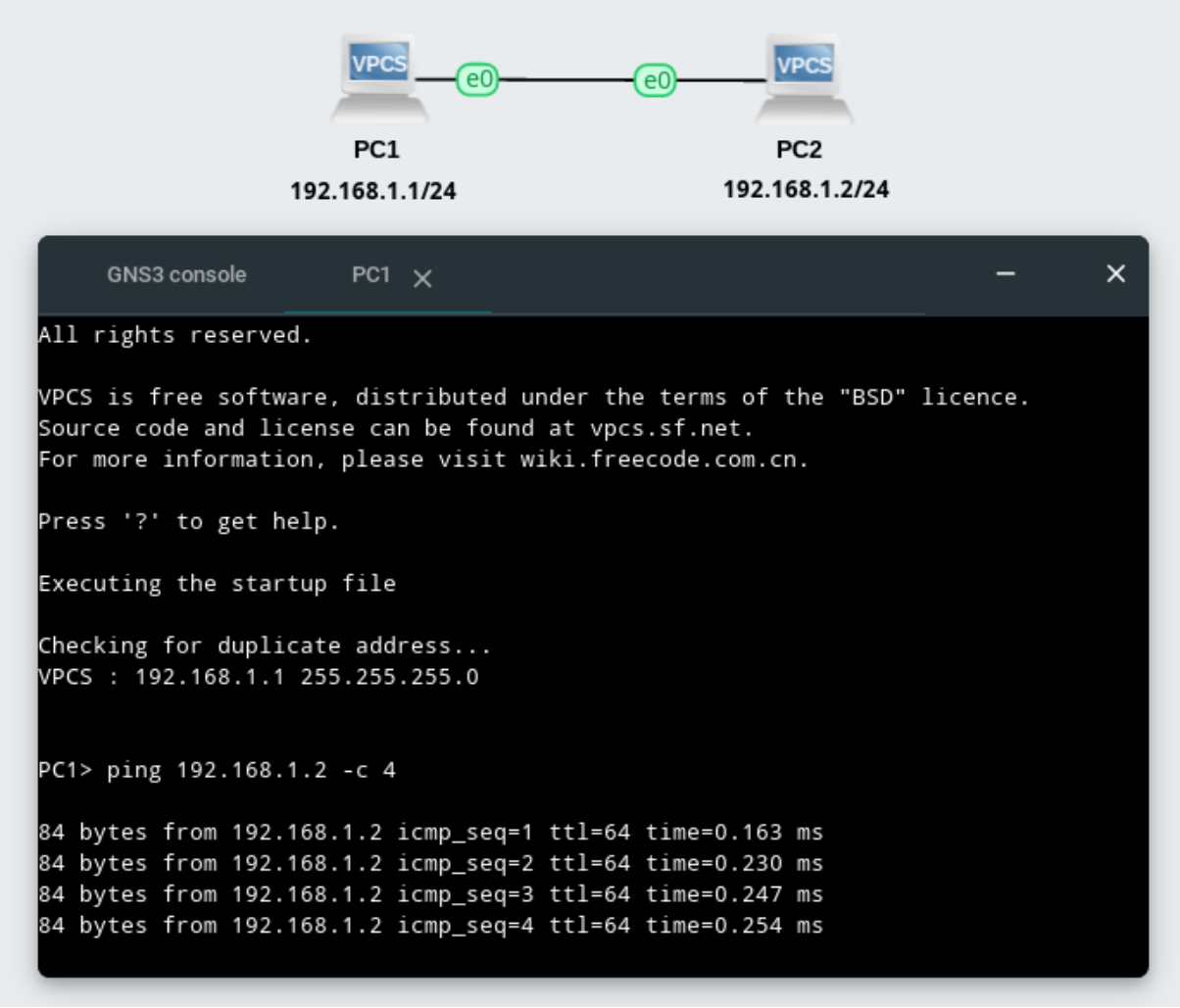


рис. 1.118 – Проверка доступности второго компьютера в симуляторе GNS3

Примечание: Для исключения конфликтов в использовании одних и тех же IP-адресов их выдачу поручили некоммерческой организации ICANN. Но выдача адресов – это дело не быстрое, а еще «белых» публичных адресов не так много, чтобы назначать их на каждый компьютер, поэтому для внутренних нужд предприятий определили несколько подсетей частных адресов:

- Сеть **192.168.0.0/16** – обычно используется Интернет-провайдерами для домашних устройств физических лиц.

- Сеть **10.0.0.0/8** – рекомендуется для использования на предприятиях.
- А для тех, кто хочет быть «не как все», есть еще диапазон **172.16.0.0** — **172.31.255.255**.

20.2.3 Объединяем несколько компьютеров в топологии «звезда»

Связывать компьютеры «один к одному» могли и раньше, но с появлением сетей стало возможно объединять их в цепочки, кольца и делать полную связанность «все со всеми». Но на предприятиях наибольшее распространение получила все же топология «звезда», которая предполагает, что компьютеры объединяются в сеть через один центральный узел, см. рис. 1.119.

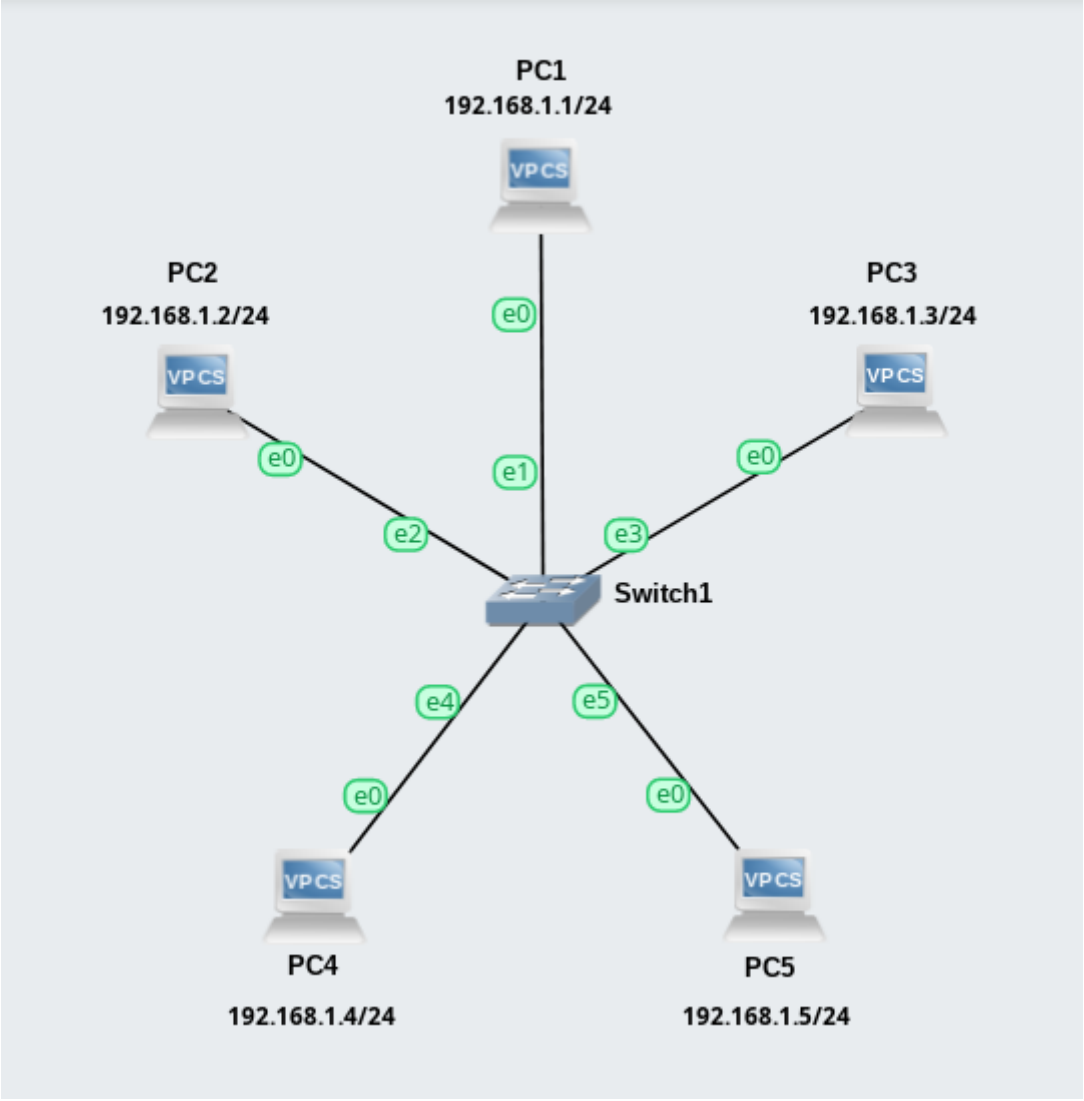


рис. 1.119 – Объединение компьютеров в топологии «звезда»

Топология звезда является очень простой, ее можно легко масштабировать, а при выходе из строя

обычного компьютера это никак не скажется на работе сети в целом.

В качестве центрального узла может выступать обычный компьютер, но в этом случае потребуется установить в него сразу несколько сетевых карт и сделать специфические настройки. Поэтому гораздо проще, дешевле и эффективнее воспользоваться специальными устройствами, которые называются **сетевыми коммутаторами**.

Первые коммутаторы назывались **хабами** и представляли собой примитивные устройства, которые умели только ретранслировать входящий сигнал на все остальные порты. В дальнейшем их вытеснили **свитчи**, которые научились запоминать, на каком порту какой MAC-адрес засветился, чтобы «не засорять эфир». И это не считая того, что свитчи стали оснащать так называемыми ASIC-микросхемами, которые способны решать задачу коммутации значительно эффективнее, чем центральный процессор обычного ПК.

20.2.4 Основы маршрутизации

Когда мы включаем на компьютере сетевой интерфейс, его IP-адрес, маска и шлюз используются для настройки таблицы маршрутизации. В системе Windows посмотреть ее можно с помощью команды `route print`, см. рис. 1.120.

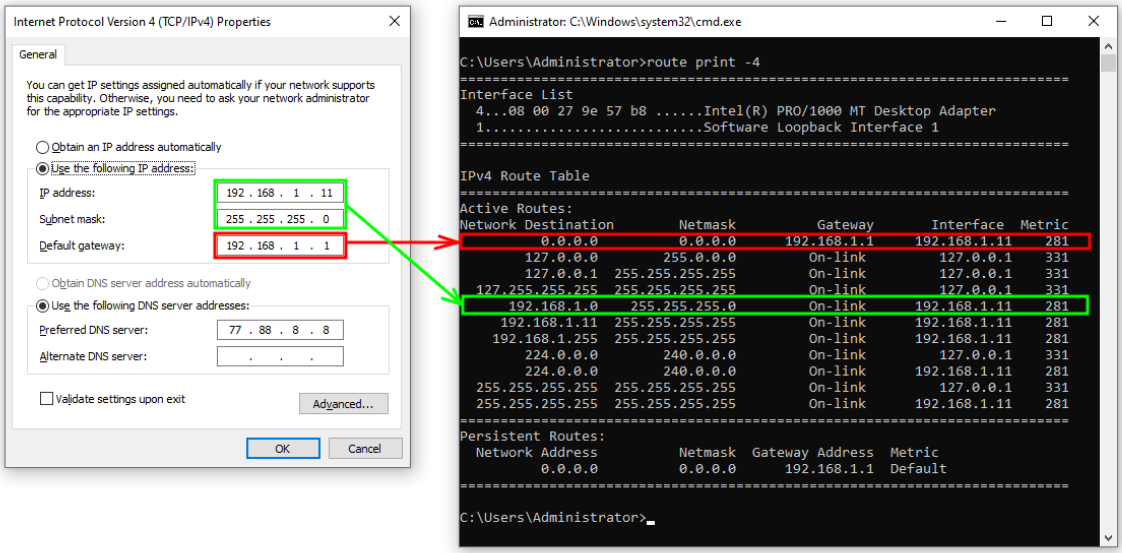


рис. 1.120 – Соответствие настроек сетевого интерфейса с записями в таблице маршрутизации в Windows

В системе Astra Linux для настройки сети в стиле Windows можно воспользоваться графическим апплетом приложения NetworkManager, а список маршрутов посмотреть утилитой `route`, см. рис. 1.121.

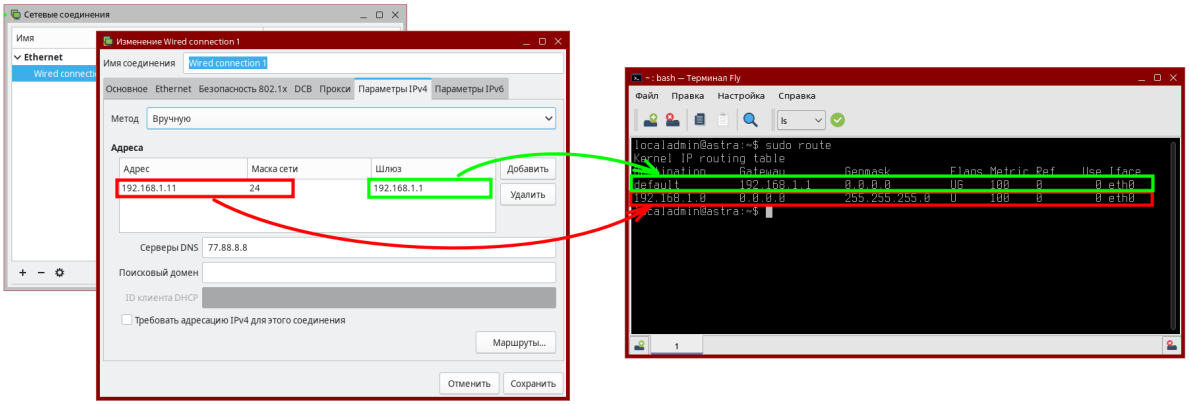


рис. 1.121 – Соответствие настроек сетевого интерфейса с записями в таблице маршрутизации в Astra Linux

Вне зависимости от того, где вы настраивали сеть, обратите внимание на следующие маршруты:

- IP-адрес и маска из настроек сетевого интерфейса определяют маршрут в локальную сеть 192.168.1.0/24, для которого в качестве шлюза указано значение On-link, т.е. пакеты нужно отправлять напрямую в сетевой интерфейс.
- Шлюз по умолчанию из настроек сетевого интерфейса определяет маршрут в сеть 0.0.0.0/0, для которого в качестве шлюза указан IP-адрес 192.168.1.1.

Когда мы обращаемся к какому-нибудь хосту по IP-адресу, наш компьютер отбирает из таблицы маршрутизации те маршруты, в сети которых попадает запрашиваемый адрес, а затем из найденных маршрутов выбирает наиболее подходящий.

20.2.4.1 Выбор маршрута при обращении к хосту из локальной сети

Если мы обратимся к хосту с IP 192.168.1.12, то этот адрес попадает в сеть 192.168.1.0/24, т.к. при побитовом умножении этого числа на маску 255.255.255.0 мы получим тот же самый адрес сети 192.168.1.0. Как вы помните «1 & 1 = 1» и «1 & 0 = 0», поэтому первые 24 бита останутся без изменений, а последние 8 бит будут обнулены, см. рис. 1.122.

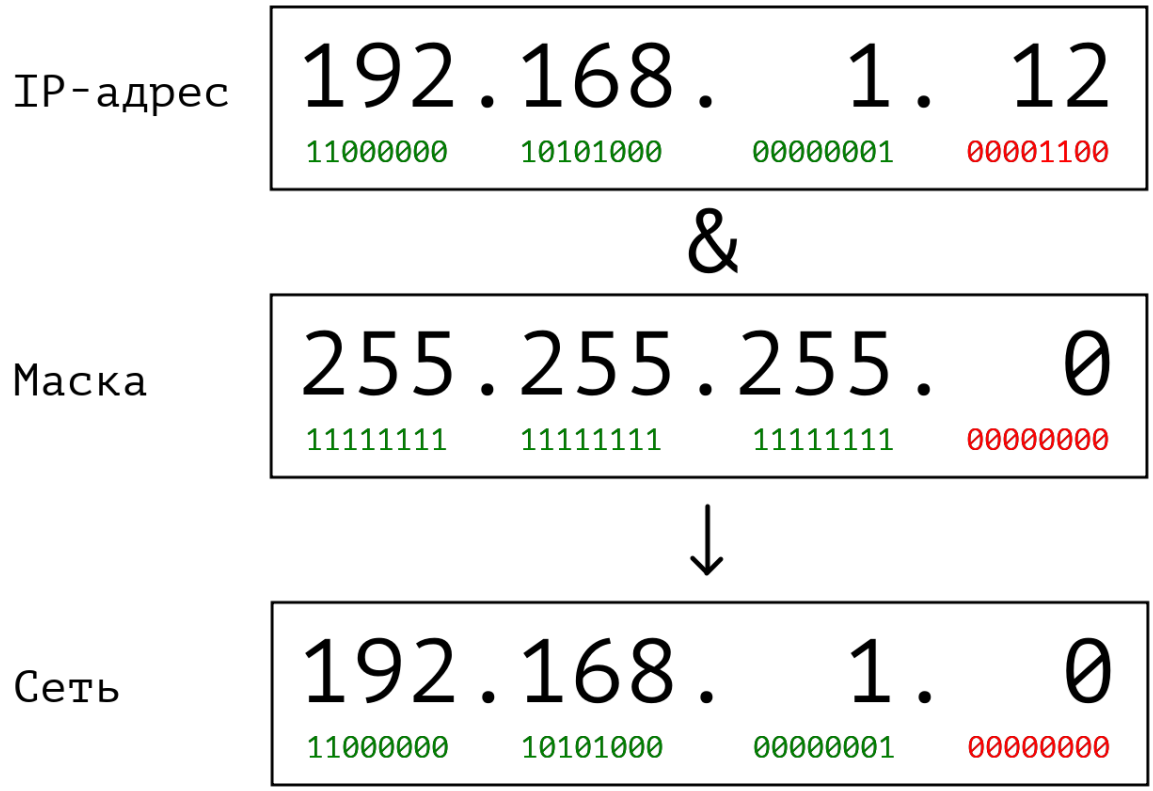


рис. 1.122 – Получение адреса сети путем побитового умножения IP-адреса на маску

Но обратите внимание, что крупные сети, у которых значение префикса меньше, как матрешки могут вбирать в себя IP-адреса более мелких сетей. А в сеть 0.0.0.0/0 так вообще попадают все возможные адреса IPv4, см. рис. 1.123, поэтому данный маршрут тоже подходит.

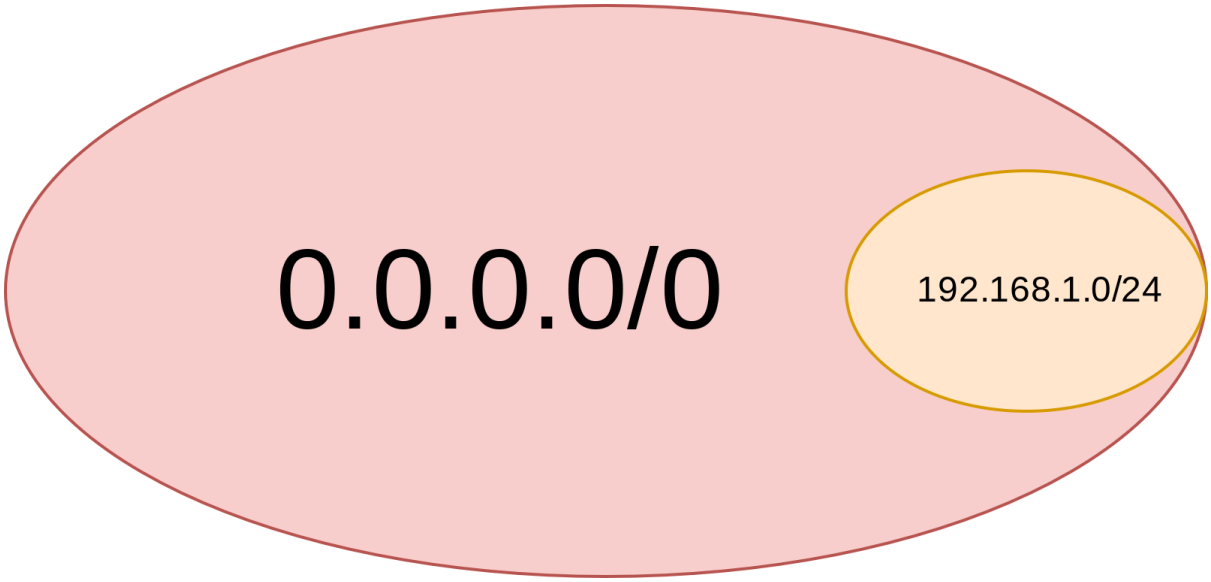


рис. 1.123 – Вложенность сетей с разной длиной префикса

Если IP-адресу назначения соответствует несколько маршрутов, то выбирается тот из них, который является более **специфичным**, т.е. у которого больше значение префикса сети. Именно поэтому мы пойдем по маршруту локальной сети (192.168.1.0/24), а не через шлюз по умолчанию (0.0.0.0/0), т.к. $24 > 0$.

Довольно часто администраторы ошибаются, полагая, что выбор маршрута осуществляется на основании **метрики**, которая учитывает количество промежуточных узлов, задержки и пропускную способность канала. Метрика тоже учитывается, но принимается во внимание только в тех случаях, когда в одну и ту же подсеть ведут несколько маршрутов с одинаковыми префиксами. Если же префиксы разные, то решение принимается исключительно по специфичности маршрута.

20.2.4.2 Выбор маршрута при обращении к хосту из сети Интернет

Стоит рассмотреть еще случай с обращением к хосту с IP 77.88.8.8, который находится в сети Интернет. С одной стороны, адресу назначения подходит только один маршрут с сетью 0.0.0.0/0, но в качестве шлюза у этого маршрута указан IP-адрес, а не физический интерфейс (On-link).

Чтобы определить физический интерфейс, куда нужно отправлять пакеты, чтобы они дошли до шлюза с IP-адресом 192.168.1.1, компьютеру нужно будет выполнить повторный проход по таблице маршрутизации, что называется рекурсивным поиском маршрута.

20.3 Стек протоколов TCP/IP

В предыдущем разделе мы рассматривали больше прикладные аспекты работы компьютерных сетей, а теперь начнем погружение в теорию, которая лежит в основе сетевых технологий.

20.3.1 Модель TCP/IP vs OSI

Стек протоколов TCP/IP был разработан в рамках проекта ARPA в конце 70-х годов, а 1 января 1983 сеть ARPANET окончательно перешла на новый стек, и эту дату принято считать официальным днем рождения Интернета.

Основная идея, заложенная в эту технологию, заключалась в разделении процесса сетевого взаимодействия на несколько стандартизированных уровней, каждый из которых должен был выполнять строго определенные функции, за счет чего можно было добиться совместимости приложений и оборудования, разрабатываемых независимыми производителями.

Идею многослойной архитектуры сетевого взаимодействия впервые сформулировали еще в конце 60-х годов, и ее развитием занималось параллельно сразу несколько команд. Например, широко известны усилия Международной организации по стандартизации (ISO) при поддержке Велико-

британии, которые привели к созданию модели OSI (англ. Open Systems Interconnection model — модель взаимодействия открытых систем). Однако победу в этом заплыве одержала все-таки команда в звездно-полосатых купальниках.

В наши дни OSI используют как эталонную модель для обучения, т.к. она является наиболее структурированной и проработанной, но мало кто знает, что у нее была вполне реальная практическая реализация. Например, в домене мы используем LDAP-протокол (Lightweight Directory Access Protocol — облегченный протокол доступа к каталогам), и его облегченность выражалась в том, что первый LDAP-сервер был просто шлюзом для доступа TCP/IP-клиентов к DAP-каталогам, которые работали на сетевом стеке OSI.

Несмотря на то, что модель OSI не используется сейчас на практике, коммутационное оборудование довольно часто относят к категориям L2 и L3, подразумевая канальный и сетевой уровни модели OSI соответственно. Например, L3-свитч – это сетевое устройство, функционирующие на третьем уровне OSI, которое предоставляет возможность коммутации и маршрутизации на базе IP-адресов. Поэтому будет полезно посмотреть, как уровни этих моделей соотносятся между собой, см. рис. 1.124.

<table>
<tr><th colspan="2">Модель OSI</th><th colspan="2">Модель TCP/IP</th></tr>
<tr><td>Прикладной уровень (application layer)</td><td>7</td><td rowspan="3">4</td><td rowspan="3">Прикладной уровень (application layer)</td></tr>
<tr><td>Уровень представления (presentation layer)</td><td>6</td></tr>
<tr><td>Сеансовый уровень (session layer)</td><td>5</td></tr>
<tr><td>Транспортный уровень (transport layer)</td><td>4</td><td>3</td><td>Транспортный уровень (transport layer)</td></tr>
<tr><td>Сетевой уровень (network layer)</td><td>3</td><td>2</td><td>Межсетевой уровень (internet layer)</td></tr>
<tr><td>Канальный уровень (data link layer)</td><td>2</td><td rowspan="2">1</td><td rowspan="2">Канальный уровень (link layer)</td></tr>
<tr><td>Физический уровень (physical layer)</td><td>1</td></tr>
</table>

рис. 1.124 – Сопоставление уровней OSI и TCP/IP

20.3.2 Уровни модели TCP/IP

Рассмотрим работу сетевой модели TCP/IP на примере HTTP-запроса к веб-серверу. Этот пример позволит пройти по всем уровням и рассмотреть каждый из них подробнее.

20.3.2.1 Прикладной уровень

На самом верхнем уровне сетевой модели TCP/IP находится прикладной уровень. Тут работают приложения, например, это может быть веб-сервер Apache, Nginx или что-то еще. Протоколы прикладного уровня определяют бизнес-логику взаимодействия пользователя с приложением.

Например, веб-сервер работает по протоколу прикладного уровня HTTP, в соответствии с которым в каждом запросе к серверу передается адрес запрашиваемой страницы, требуемый метод обработки запроса (GET, POST и т.д.), заголовки и при необходимости тело сообщения, в котором могут содержаться данные веб-формы. А в ответ веб-сервер возвращает нам, к примеру, контент страницы или код состояния 404, который указывает, что запрошенный нами контент не найден.

Примерно ту же роль выполняют прикладные протоколы SMTP, POP3 и другие — меняется только бизнес-логика приложений, а, следовательно, набор методов для взаимодействия с сервером, способы кодирования информации и т.п.

20.3.2.2 Транспортный уровень

Данные пользовательских приложений передаются по компьютерной сети внутри пакетов транспортного уровня. Основными протоколами тут являются:

- **TCP** (Transmission Control Protocol — протокол управления передачей) — устанавливает двустороннее соединение и гарантирует доставку сообщений.

  Этот протокол транспортного уровня используется подавляющим большинством протоколов прикладного уровня, например, HTTP, Kerberos, LDAP, SMB и другими, так как надежность обычно важнее скорости.

  Кстати, огромную роль этого протокола для всей технологии в целом разработчики отметили внесением его имени в общее название стека.

- **UDP** (User Datagram Protocol — протокол пользовательских датаграмм) — предполагает отправку сообщений в одну сторону без обратной связи.

  Этот протокол транспортного уровня широко используется в приложениях, для которых скорость важнее надежности или не требуется установление сессии, например, NTP, DNS, DHCP, потоковое аудио и видео (SIP и IPTV или, вернее, RTP).

Пакеты TCP/UDP, или как их еще называют сегменты, включают данные пользовательских приложений и добавляют к ним свой заголовок с необходимой метаинформацией – этот процесс называют инкапсуляцией, см. рис. 1.125.

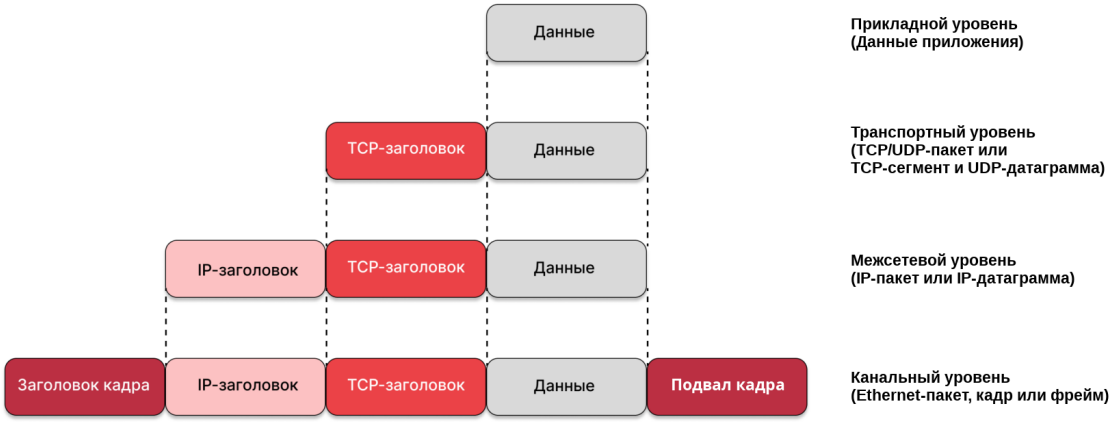


рис. 1.125 – Инкапсуляция пользовательских данных по стеку TCP/IP

Среди параметров, которые передаются в заголовке TCP/UDP-сегмента, наиболее важными являются **порты** источника и приемника. Порт представляет собой целое число в диапазоне от 0 до 65 535, с помощью которого операционная система обеспечивает одновременную работу сразу множества сетевых приложений.

Когда мы запускаем на компьютере сетевое приложение, оно открывает какой-то конкретный порт, то есть просит операционную систему пересылать ему все входящие пакеты, внутри которых порт получателя будет иметь указанное значение. Поэтому, когда говорят, что сервер был взломан через 22-й порт, подразумевают, что на этом сервере была запущена служба sshd, которая прослушивала 22-й порт, и злоумышленникам удалось реализовать один из известных векторов атак, например, методом брутфорса подобрать пароль администратора. Как говорится: «Дети, самый лучший пароль для администратора – это SSH-ключ»)))

Довольно часто у администраторов складывается ошибочное впечатление, что количество портов ограничивает количество входящих TCP-соединений. На самом деле тот же веб-сервер может принимать все входящие соединения на один 443-й порт, и у операционной системы не возникает никаких проблем с разбором входящих пакетов, для чего она использует дополнительно IP-адрес и порт отправителя. Таким образом, количество портов ограничивает не число соединений, а количество разных сетевых приложений, которые могут работать одновременно.

Отметим также, что правила привязки довольно гибкие, поэтому приложение может подписаться как на получение всех пакетов, поступающих на 443-й порт, так и конкретизировать, что это должны быть только TCP-пакеты или что эти пакеты должны поступать на интерфейс с конкретным IP-адресом. Поэтому в правилах межсетевого экрана для веб-приложения может быть указано требование открыть входящий порт в формате 443/TCP для конкретного IP.

20.3.2.3 Межсетевой уровень

Сегменты транспортного уровня передаются по компьютерной сети внутри пакетов межсетевого уровня, которые называются также датаграммами. Основным протоколом на этом уровне является, конечно же, протокол IP:

- **IP** (Internet Protocol — межсетевой протокол) – обеспечивает доставку пакетов между любыми хостами сети через произвольное количество промежуточных узлов (маршрутизаторов).

  Протокол не гарантирует надежной доставки до адресата, т.е. пакеты могут прийти в неправильном порядке, продублироваться, оказаться повреждёнными или вовсе потеряться по дороге. Гарантию безошибочной доставки дают некоторые протоколы более высокого уровня, например, TCP, которые используют IP в качестве транспорта.

  Протокол IP стал вторым протоколом, роль которого разработчики отметили внесением его имени в общее название стека.

Дополнительно можно упомянуть протоколы:

- **ICMP** (Internet Control Message Protocol — протокол межсетевых управляющих сообщений) — обеспечивает передачу служебных сообщений об ошибках и других исключительных ситуациях, возникших при передаче данных.

  Протокол используется, например, одной из самых популярных утилит ping, которая позволяет проверить доступность узла. Технически ICMP основан на протоколе IP, т.е. каждое ICMP-сообщение инкапсулируется непосредственно в пределах одного IP-пакета.

- **IGMP** (Internet Group Management Protocol — протокол управления группами Интернета) — используется для объединения сетевых устройств в группы, например, для организации широковещательных трансляций потокового видео или многоадресной рассылки при установке операционной системы по сети. Технически протокол IGMP основан на IP.

  Для работы протокола зарезервирована сеть 224.0.0.0/4 (224.0.0.0 — 239.255.255.255), каждый адрес которой определяет одну мультикастную группу. Серверное приложение в момент своего запуска посылает в сеть уведомление, что соответствующая группа доступна, а клиенты отвечают, что они хотят присоединиться, при этом роутеры и свитчи запоминают, что за соответствующим маршрутом/портом находится клиент соответствующей мультикастной группы. Добавление новых участников и удаление отключившихся также регулируется через протокол IGMP.

Следуя той же логике инкапсуляции, датаграммы включают сегменты транспортного уровня, добавляя к ним IP-заголовок со всей необходимой метаинформацией. Среди параметров, которые передаются в заголовке датаграммы, наиболее важным является IP-адрес.

Как мы обсуждали выше, адрес IPv4 представляет собой 32-битное целое число, которое обычно представляют в виде четырех октетов, разделенных символом точки. Используя значение IP-адреса, сетевой стек выбирает из таблицы маршрутизации наиболее подходящее направление для доставки пакета и посылает его в соответствующий сетевой интерфейс. Важно понимать, что значение маски этого маршрута не включается в IP-заголовок. Эта информация остается только на узлах сети и используется ими для правильной настройки маршрутизации.

20.3.2.4 Канальный уровень

На самом нижнем уровне сетевой модели TCP/IP находится канальный уровень, где под каналом подразумевается канал связи, через который осуществляется взаимодействие устройства с компьютерной сетью.

Датаграммы передаются по компьютерной сети внутри Ethernet-пакетов, которые называют также фреймами или кадрами. Основными протоколами на канальном уровне являются:

- Ethernet (ether — эфир и network — сеть) — самый распространенный протокол для передачи данных между устройствами, который использует MAC-адреса (Media Access Control) для идентификации отправителя и получателя данных.
- ARP (Address Resolution Protocol — протокол определения адреса) — предназначен для определения MAC-адреса другого компьютера по известному IP-адресу.
- RARP (Reverse Address Resolution Protocol — обратный протокол преобразования адресов) — предназначен для обратного отображения адресов, то есть определяет IP-адрес другого компьютера по известному MAC-адресу.

Следуя той же логике инкапсуляции, Ethernet-кадры включают датаграммы межсетевого уровня, добавляя к ним заголовок со всей необходимой метаинформацией, среди которой наиболее важным параметром является **MAC-адрес**.

Значение MAC-адреса представляет собой 12-разрядное шестнадцатеричное число, которое должно уникально идентифицировать сетевое устройство в пределах физического сегмента компьютерной сети. Это значение устанавливается сетевому устройству на заводе с использованием уникального префикса производителя, поэтому вероятность того, что в одной сети окажется два устройства с одним и тем же MAC-адресом, крайне мала. При необходимости MAC-адрес можно изменить в настройках сетевой карты на программном уровне.

Чтобы определить MAC-адрес устройства, которому нужно передать IP-датаграмму, компьютер отправляет в сеть широковещательный ARP-запрос. Полученные результаты записываются в заголовок Ethernet-кадра и кэшируются в ARP-таблице для повышения производительности работы сети.

К канальному уровню TCP/IP часто приписывают также физический уровень, т.к. модель TCP/IP (в отличие от OSI) не затрагивает и никак не стандартизирует аспекты физической передачи данных. А некоторые авторы так вообще добавляют к TCP/IP еще один уровень, наделяя этот стек свойствами пятиуровневой модели, что не только противоречит официальным RFC, но и создает дополнительную путаницу. Поэтому при ознакомлении с документацией, профильными статьями и учебной литературой вам следует обращать внимание на контекст, в котором авторы этих материалов излагают свои тезисы.

20.3.3 Протокол TCP в деталях

Продолжаем погружение. Теперь более подробно рассмотрим процесс установления соединения по протоколу транспортного уровня TCP.

20.3.3.1 Заголовок сегмента

Сегмент TCP (так называют TCP-пакеты) состоит из заголовка и данных, см. рис. 1.126.

Бит	0 – 3	4 – 6	7 – 15	16 – 31
0	Порт источника (**Source Port**)			Порт назначения (**Destination Port**)
32	Порядковый номер (**Sequence Number, SN**)			
64	Номер подтверждения (**Acknowledgment Number, ACK SN**)			
96	Длина заголовка (**Data offset**)	Зарезервировано	Флаги	Размер Окна (**Window size**)
128	Контрольная сумма (**Checksum**)			Указатель важности (**Urgent Point**)
160	**Опции** (необязательное, но используется практически всегда)			
160/192+	**Данные**			

рис. 1.126 – Структура заголовка сегмента TCP

Порт источника и назначения (по 16 бит) – поля, содержащие соответственно номер порта отправителя и получателя TCP-пакета.

Порядковый номер (32 бита) – поле, которое позволяет восстановить правильный порядок сегментов на стороне получателя. Нумерация необходима, т.к. при отправке большого объема данных для повышения эффективности протокола информация разбивается на небольшие сегменты размером 1500 байт, и нижележащие протоколы не гарантируют того, что пакеты придут к получателю в том же порядке. Размер сегментов можно регулировать, за это отвечает настройка MTU (maximum transmission unit — максимальная единица передачи).

Номер подтверждения (32 бита) – если установлен флаг ACK, то это поле содержит порядковый номер октета, который отправитель данного сегмента желает получить. Это означает, что все предыдущие октеты (с номерами от ISN+1 до ACK-1 включительно) были успешно получены.

Длина заголовка (смещение данных, 4 бита) – указывает значение длины заголовка, измеренное в

32-битовых словах. Минимальный размер составляет 20 байт (пять 32-битовых слов), а максимальный — 60 байт (пятнадцать 32-битовых слов). Длина заголовка определяет смещение полезных данных относительно начала сегмента.

Флаги (девять управляющих бит), из которых наиболее интересными являются SYN и ACK, которые используются в процедуре «рукопожатия» для установления соединения.

Размер окна (16 бит) – определяет количество байт данных (*англ. payload*), после передачи которых отправитель ожидает подтверждения от получателя, что данные получены. Иначе говоря, получатель пакета располагает для приёма данных буфером длиной «размер окна» байт.

Контрольная сумма (16 бит) – при расчёте контрольной суммы значение самого поля контрольной суммы принимается равным 0.

Указатель важности (16 бит) – это поле указывает порядковый номер октета, которым заканчиваются важные (urgent) данные. Поле принимается во внимание только для пакетов с установленным флагом URG.

Опции – могут применяться в некоторых случаях для расширения протокола.

20.3.3.2 Механизм работы протокола

Перед непосредственной отправкой пакетов протокол TCP (в отличии от UDP) устанавливает соединение, поэтому процесс передачи данных по протоколу TCP можно представить в трех актах: установка соединения, передача данных и завершение соединения.

Соединение TCP может находиться в одном из 11 состояний:

- **CLOSED** – начальное состояние узла. Фактически фиктивное.
- **LISTEN** – сервер ожидает запросы установления соединения от клиента.
- **SYN-SENT** – клиент отправил запрос серверу на установление соединения и ожидает ответа.
- **SYN-RECEIVED** – сервер получил запрос на соединение, отправил ответный запрос и ожидает подтверждения.
- **ESTABLISHED** – соединение установлено, идёт передача данных.
- **FIN-WAIT-1** – одна из сторон (узел-1) завершает соединение путем отправки сегмента, в котором установлен флаг FIN.
- **CLOSE-WAIT** – другая сторона (узел-2) переходит в это состояние, отправив, в свою очередь, сегмент ACK и продолжает одностороннюю передачу.

- **FIN-WAIT-2** – узел-1 получает ACK, продолжает чтение в ожидании получения сегмента с флагом FIN.
- **LAST-ACK** – узел-2 заканчивает передачу и отправляет сегмент с флагом FIN.
- **TIME-WAIT** – узел-1 получил сегмент с флагом FIN, отправил сегмент с флагом ACK и ждёт 2*MSL (максимальное временя жизни сегмента) секунд, перед окончательным закрытием соединения
- **CLOSING** – обе стороны инициировали закрытие соединения одновременно: после отправки сегмента с флагом FIN узел-1 также получает сегмент FIN, отправляет ACK и находится в ожидании сегмента ACK (подтверждения на свой запрос о разъединении).

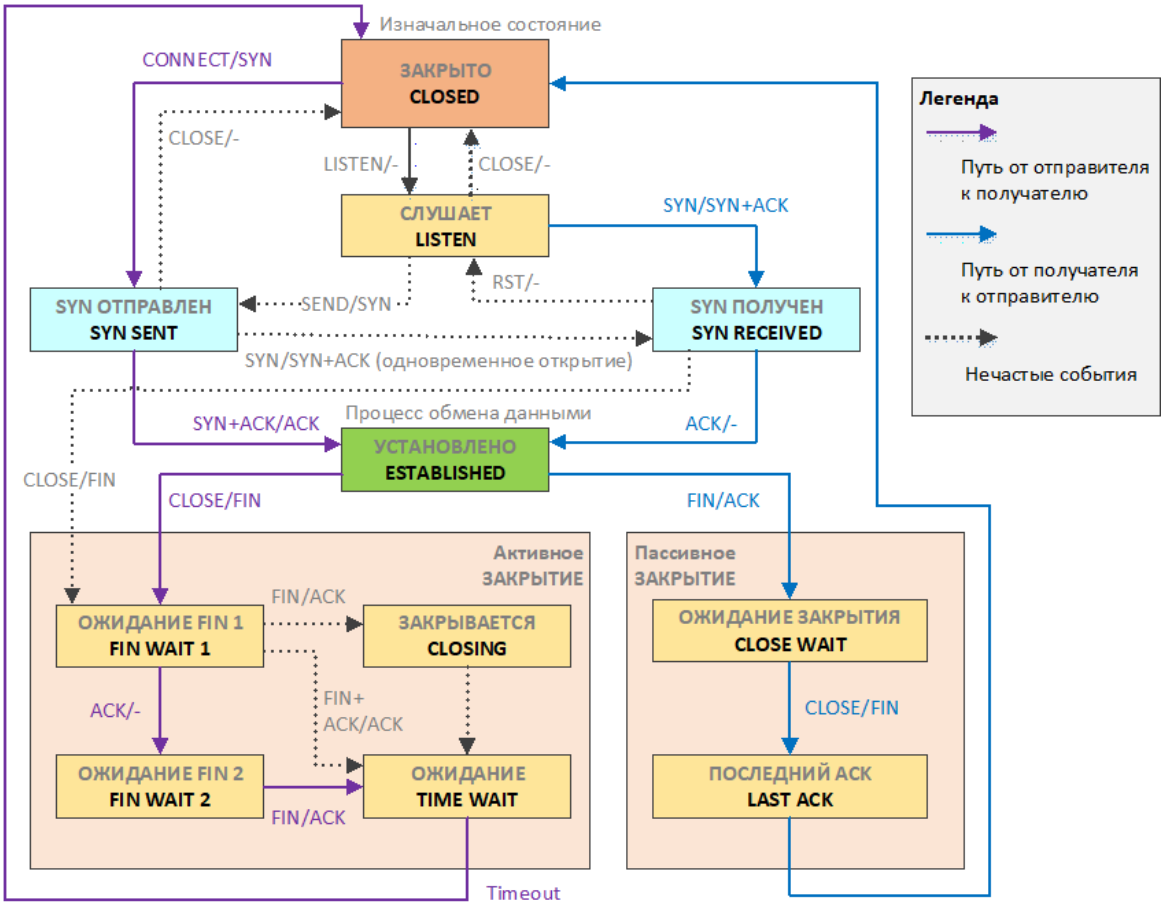


рис. 1.127 – Схема состояний TCP

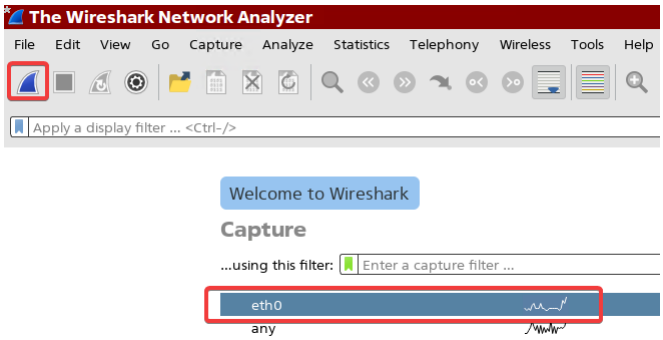
Установка утилиты wireshark для анализа сетевого трафика при TCP-соединении

Проанализируем сетевой трафик, генерируемый утилитой `dig` при разрешении имени astralinux.ru. Для этого воспользуемся утилитой wireshark.

Установим Wireshark и запустим её:

```
localadmin@astra:~$ sudo apt install wireshark
localadmin@astra:~$ sudo wireshark &
```

Запустим сбор трафика с интерфейса eth0:



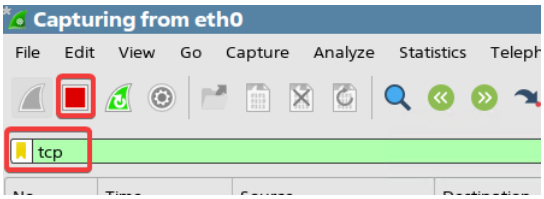
Выполним в терминале команду dig для разрешения имени «astralinux.ru»:

```
localadmin@astra:~$ dig astralinux.ru @77.88.8.8 +tcp
```

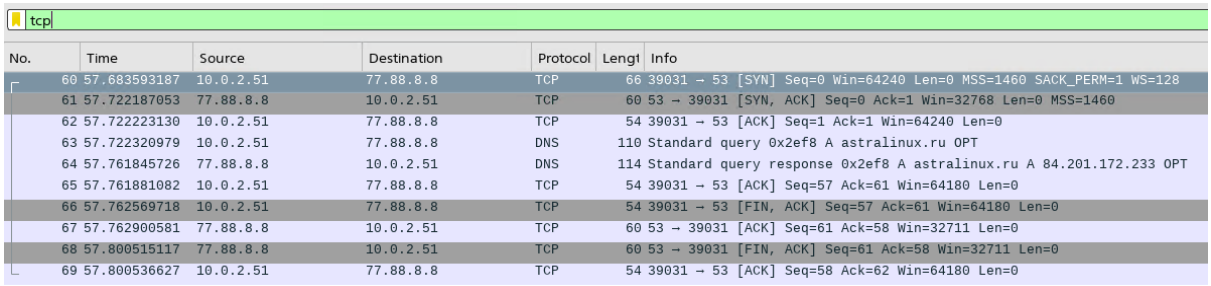
Где:

- **@77.88.8.8** – DNS-сервер Яндекса.
- **+tcp** – параметр, предписывающий команде использовать протокол TCP (по умолчанию используется UDP).

Остановим сбор трафика в окне программы wireshark и применим фильтр «tcp»:

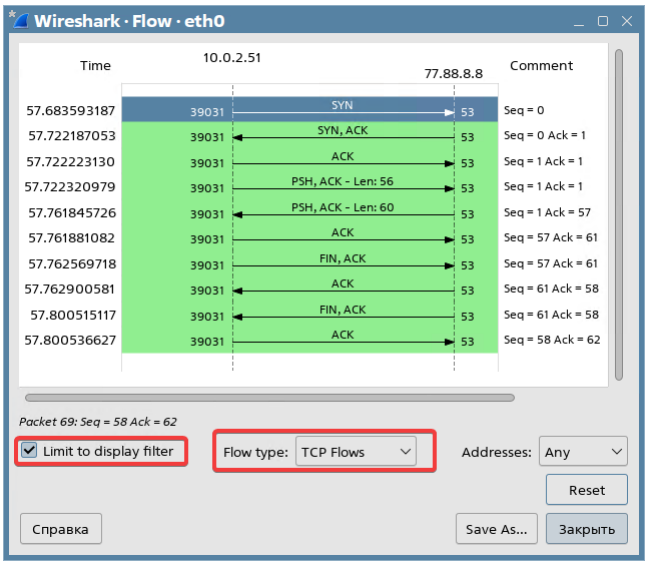


При успешном выполнении команды в окне программы wireshark мы увидим весь процесс обмена сообщениями:



Выбрав пункт меню *Statistics → Flow Graph*, можно открыть окно с графом обмена сообщениями.

Выберем опцию «Limit to display filter» и тип «TCP Flows»:



В следующих подразделах мы разберем эти сообщения, так что пока не закрывайте окно Wireshark.

Установка соединения

Процесс установки соединения называется также «рукопожатием» (*англ. handshake*) и проходит в три этапа:

1. Клиент, который намеревается установить соединение, посылает серверу сегмент с флагом SYN.

    - Сервер получает сегмент, запоминает номер последовательности и пытается создать сокет (буфер и управляющие структуры памяти) для обслуживания нового клиента.
    - В случае успеха сервер посылает клиенту сегмент с номером последовательности и флагами SYN и ACK и переходит в состояние SYN-RECEIVED.
    - В случае неудачи сервер посылает клиенту сегмент с флагом RST.

2. Если клиент получает сегмент с флагом SYN, то он запоминает номер последовательности и посылает сегмент с флагом ACK.

    - Если клиент одновременно получает и флаг ACK (что обычно и происходит), то он переходит в состояние ESTABLISHED.
    - Если клиент получает сегмент с флагом RST, то он прекращает попытки соединиться.
    - сли клиент не получает ответ в течение 10 секунд, то он повторяет процесс соединения заново.

3. Если сервер в состоянии SYN-RECEIVED получает сегмент с флагом ACK, то он переходит в состояние ESTABLISHED.

    - В противном случае после тайм-аута он закрывает сокет и переходит в состояние CLOSED.

Ниже приведен пример базового трехэтапного согласования:

	Узел-1 клиент		Узел-2 сервер
1	CLOSED		LISTEN
2	SYN-SENT –>	<SEQ=100><CTL=SYN>	–> SYN-RECEIVED
3	ESTABLISHED <–	<SEQ=300><ACK=101><CTL=SYN,ACK>	<– SYN-RECEIVED

1. Узел 1 передает сегмент с флагом SYN и номером последовательности 100.

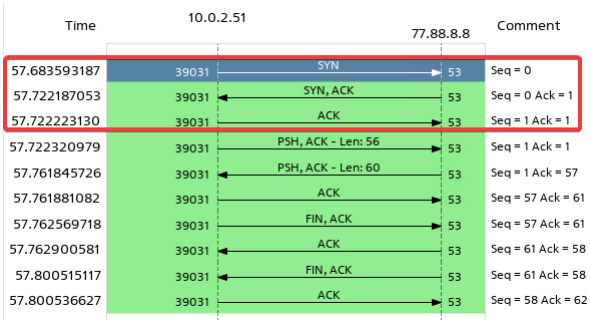
2. Узел 2 передает в ответ сегмент SYN с номером последовательности 300 и подтверждением ACK о принятом SYN (SEQ 101).

3. Узел 1 отвечает пустым сегментом с подтверждением ACK о принятом SYN (SEQ 301).

Номер последовательности SEQ – это счетчик, указывающий, сколько данных (в байтах) было отправлено в текущем сеансе TCP. Номер подтверждения ACK – это подтверждение о полученных данных и готовности к следующему сегменту.

С номером подтверждения ACK не все так просто. При передаче информации номер ACK равен сумме номера полученной последовательности и длины её сегмента TCP: $\text{ACK}_{\text{обмен сообщениями}} = \text{SEQ} + \text{LEN}$. Однако при установке и завершении соединения этот номер будет инкрементироваться на единицу: $\text{ACK}_{\text{установка/завершение соединения}} = \text{SEQ} + \text{LEN} + 1$ (длина при этом равна 0).

В нашем примере с DNS-запросом обратим внимание на первые 3 итерации – это и есть процесс «рукопожатия» в TCP:



Source	Destination	Protocol	Lengt	Info
10.0.2.51	77.88.8.8	TCP	66	39031 → 53 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 WS=128
77.88.8.8	10.0.2.51	TCP	60	53 → 39031 [SYN, ACK] Seq=0 Ack=1 Win=32768 Len=0 MSS=1460
10.0.2.51	77.88.8.8	TCP	54	39031 → 53 [ACK] Seq=1 Ack=1 Win=64240 Len=0

1. Клиент отправляет серверу сообщение с флагом SYN и номером последовательности 0 (SEQ=0).

2. Сервер отвечает клиенту сообщением с флагами SYN, ACK, номером последовательности 0 (SEQ=0) и подтверждением 1 (ACK=1).

3. Клиент отвечает серверу сообщением с флагом ACK, номером последовательности 1 (SEQ=1) и подтверждением 1 (ACK=1).

Примечание: По умолчанию Wireshark преобразует все порядковые номера и номера подтверждений в относительные номера. Это означает, что все номера SEQ и ACK всегда начинаются с 0 для первого пакета в каждой TCP-сессии.

Процесс передачи данных

При обмене данными приёмник использует номер последовательности, содержащийся в получаемых сегментах, для восстановления их исходного порядка. Приёмник уведомляет передающую сторону о номере последовательности, до которой он успешно получил данные, включая его в поле «номер подтверждения».

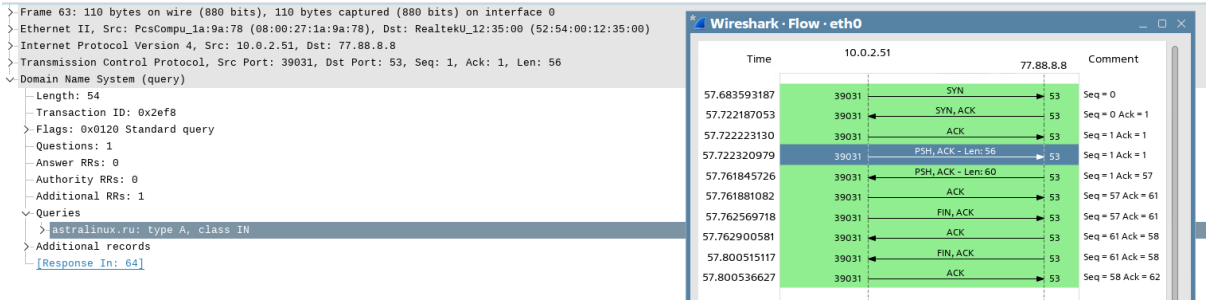
Если полученный сегмент содержит номер последовательности больший, чем ожидаемый, то данные из сегмента буферизируются, но номер подтверждённой последовательности не изменяется. Если впоследствии будет принят сегмент, относящийся к ожидаемому номеру последовательности, то порядок данных будет автоматически восстановлен исходя из номеров последовательностей в сегментах.

Поле «окно» позволяет принимающей стороне управлять интенсивностью потока данных. Сторона отправителя не будет посылать данных больше, чем указано в этом поле.

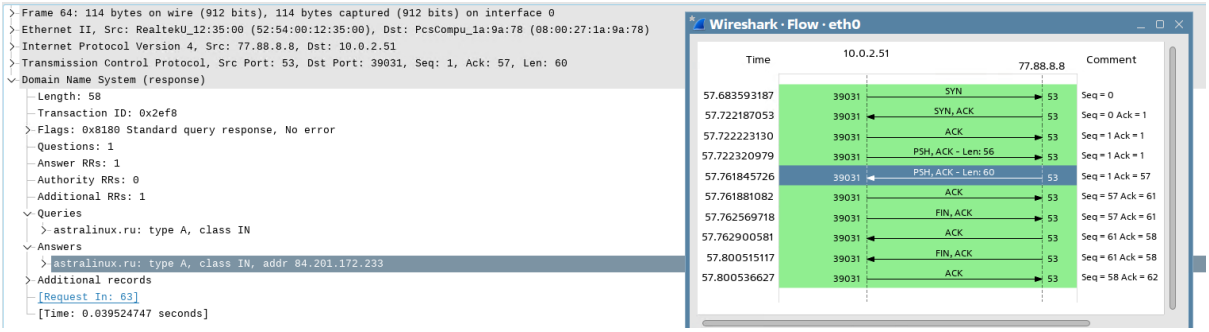
С помощью флага PSH передающая сторона может затребовать принять данные, не буферизуя их. Это «проталкивание» используется, например, в интерактивных приложениях.

Вернемся в окно программы Wireshark и рассмотрим обмен сообщениями по протоколу DNS:

1. Клиент отправляет серверу DNS-запрос на разрешения имени «astralinux.ru». Как видим, тут как раз используется флаг PSH:



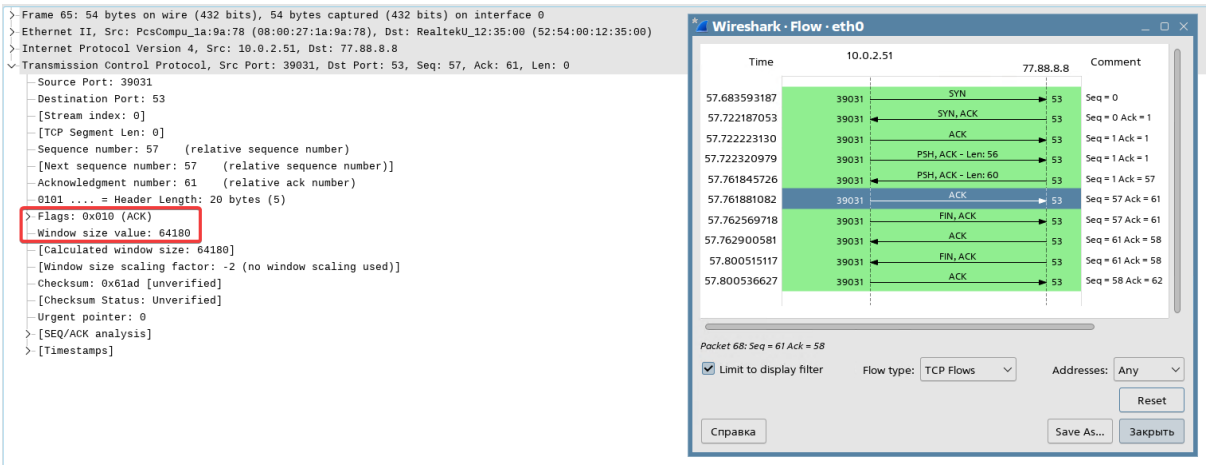
2. Сервер DNS отправляет клиенту ответ с найденным IP-адресом:



Обратите внимание, что номер подтверждения в этом случае будет ACK=57, так как это подтверждение на сообщение с номером последовательности SEQ=1 и длиной LEN=56.

3. Следующим сообщением с флагом ACK, номером последовательности 57 (SEQ=57) и подтверждением 61 (ACK=61) клиент уведомляет сервер, что он получил и принял отправленный ему пакет.

   В разделе «Transmission Control Protocol» можно посмотреть параметры передачи сообщений, в том числе и вышеупомянутое поле «окно» и флаги:



Завершение соединения

Завершение соединения можно рассмотреть в четыре этапа:

	Узел-1 клиент				Узел-2 сервер
	ESTABLISHED				ESTABLISHED
1.	(Close) FIN-WAIT-1	–>	<SEQ=100><ACK=300><CTL=FIN,ACK>		FIN-WAIT
2.	FIN-WAIT-2	<–	<SEQ=300><ACK=101><CTL=ACK>		CLOSE-WAIT
3.	(Close) TIME-WAIT	<–	<SEQ=300><ACK=101><CTL=FIN,ACK>		LAST-ACK
4.	TIME-WAIT	–>	<SEQ=101><ACK=301><CTL=ACK>		CLOSED
	(2 MSL) CLOSED				

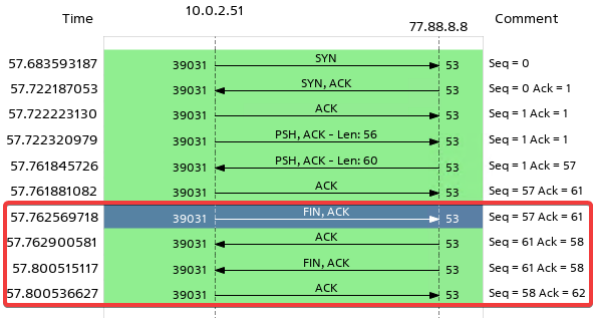
1. При завершении соединения клиент посылает серверу сообщение с флагами FIN и ACK, что означает, что клиент получил все необходимые ему данные и хочет закрыть соединение (SEQ=100, ACK=300).

2. В ответ сервер уведомляет клиента (ACK), что принял сообщение (SEQ=300, ACK=101).

3. В случае, если у сервера больше нет данных для клиента, он отправит ему сообщение с флагами FIN и ACK (SEQ=300, ACK=101).

4. Процесс закрытия сессии завершается отправкой клиентом серверу уведомления (ACK), что он принял сообщение (SEQ=101, ACK=301).

В нашем примере с DNS-запросом в Wireshark это выглядит следующим образом:




```
10.0.2.51       77.88.8.8       TCP     54 39031 → 53 [FIN, ACK] Seq=57 Ack=61 Win=64180 Len=0
77.88.8.8       10.0.2.51       TCP     60 53 → 39031 [ACK] Seq=61 Ack=58 Win=32711 Len=0
77.88.8.8       10.0.2.51       TCP     60 53 → 39031 [FIN, ACK] Seq=61 Ack=58 Win=32711 Len=0
10.0.2.51       77.88.8.8       TCP     54 39031 → 53 [ACK] Seq=58 Ack=62 Win=64180 Len=0
```

20.3.4 Протокол IP в деталях

Как мы помним, протокол объединяет сегменты сети (не путать с TCP-сегментами), обеспечивая доставку данных между любыми узлами сети через произвольное число промежуточных узлов (маршрутизаторов).

При доставке IP-пакета он проходит через разные каналы доставки. Возможно возникновение ситуации, когда размер пакета превысит возможности узла системы связи. В этом случае протокол предусматривает возможность дробления пакета, что называется IP-фрагментацией. В протоколе предусмотрена возможность запрета фрагментации конкретного пакета. Если такой пакет нельзя передать через сегмент связи целиком, то он уничтожается, а отправителю направляется ICMP-сообщение о проблеме.

Сейчас используются две актуальные версии протокола:

- **IPv4** — предусматривает использование 32-битных адресов, которых, как оказалось, не так уж и много, в связи с чем приходится придумывать разные ухищрения, такие как трансляция адресов NAT. Но этот протокол еще на долгие годы будет оставаться основной рабочей лошадкой.

- **IPv6** — разработан для решения проблемы нехватки адресов IPv4 и предусматривает использование адресного пространства емкостью 2^{128}, что очень-очень много, но это сделано для использования иерархии адресов в целях упрощения маршрутизации.

  Адрес в IPv6 представлен восемью четырехзначными шестнадцатеричными числами, разделенными двоеточиями, например, 2001:0db8:11a3:09d7:1f34:8a2e:07a0:765d, и это, пожалуй, все, что мы хотели бы рассказать.

Мы уже затрагивали вопрос того, что адреса IPv4 могут быть публичными (белыми) и частными, но добавим немного дополнительных деталей.

Частный IP-адрес (*англ. private IP address*), называемый также внутренним, внутрисетевым или локальным — это IP-адрес, принадлежащий к специальному диапазону, не используемому в сети Интернет. Такие адреса предназначены для применения в локальных сетях, и их распределение контролируется только внутри предприятия.

Для того чтобы компьютеры, использующие локальные адреса, могли выйти в Интернет, используется технология трансляции адресов NAT (*Network Address Translation — преобразование сетевых адресов*). Публичный адрес назначается только NAT-шлюзу, который принимает запросы на пересылку пакетов от клиентов локальной сети и подставляет в них свой собственный адрес в качестве IP-адреса получателя. При получении ответа NAT-шлюз выполняет обратное действие, чтобы клиенты считали, что они взаимодействуют с сервером напрямую. Таким образом, все пользователи предприятия ходят в сеть под одним и тем же белым IP-адресом.

Существуют следующие сети частных адресов IPv4:

- **Сеть 192.168.0.0/16** — включает всего 65к адресов, поэтому считается сетью для домашнего пользования. Эти адреса обычно используют Интернет-провайдеры для физических лиц, и они довольно часто назначены в простеньких роутерах по умолчанию.
- **Сеть 172.16.0.0/12** — включает порядка 1 млн. адресов, поэтому может использоваться в школах, университетах и на предприятиях малого бизнеса.
- **Сеть 10.0.0.0/8** — включает порядка 16 млн. адресов, поэтому подходит для построения сетей любого масштаба вплоть до предприятий крупного бизнеса.
- **Сеть 100.64.0.0/10** — рекомендована для использования провайдерами в качестве адресов CGN (Carrier-Grade NAT). Эта технология отличается от обычного NAT тем, что один публичный IP-адрес может использоваться несколькими маршрутизаторами клиента.
- **Сеть 169.254.0.0/16** — включает 65к link-local адресов, которые предназначены для коммуникаций только в пределах одного физического сегмента сети, т.е. маршрутизаторы не пересылают пакеты, если в качестве источника или приемника указан link-local адрес. Раньше эти адреса могли быть использованы в небольших локальных сетях без выхода в Интернет. Серверам они назначались вручную, а рабочие станции выбирали себе адрес автоматически, если не получали ответ от DHCP-сервера. В Microsoft такой способ настройки называли APIPA (англ. Automatic Private IP Addressing — автоматическая адресация внутри локальной сети).

  В настоящее время эта сеть практически не используется, и мы сталкиваемся с этими адресами только в том случае, если сервер DHCP оказался недоступен. Но если вам потребуется создать ну очень защищенный периметр без возможности выхода в Интернет на аппаратном уровне, теперь вы знаете, как это сделать)))
- **Сеть 127.0.0.0/8** — адреса локальной петлей, и включает порядка 16 млн. адресов. Они используется для организации межпроцессного взаимодействия между приложениями, которые запущены внутри одной операционной системы.

20.3.4.1 Заголовок датаграммы

Пакет IP — это форматированный блок информации, передаваемый по компьютерной сети, структура которого определена протоколом IP, см. рис. 1.128.

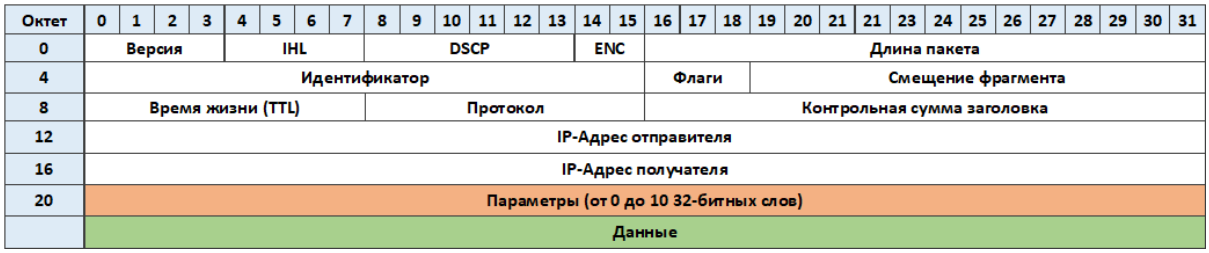


рис. 1.128 – Структура IP-пакета

- **Версия** — для IPv4 значение поля должно быть равно 4.
- **IHL** (Internet Header Length) длина заголовка IP-пакета в 32-битных словах (dword). Именно это поле указывает на начало блока данных (англ. payload — полезный груз) в пакете. Минимальное корректное значение для этого поля равно 5.
- **DSCP** (Differentiated Services Code Point, точка кода дифференцированных услуг) – элемент архитектуры компьютерных сетей, описывающий простой масштабируемый механизм классификации, управления трафиком и обеспечения качества обслуживания.
- **ECN** (Explicit Congestion Notification, явное уведомление о перегруженности) – расширение протокола IP, описанное в RFC 3168. ECN позволяет обеим сторонам в сети узнавать о возникновении затора на маршруте к заданному хосту или сети без отбрасывания пакетов.
- **Длина пакета** (Total Length) — длина пакета в октетах, включая заголовок и данные. Минимальное корректное значение для этого поля равно 20, максимальное — 65 535.
- **Идентификатор** (Identification) — значение, назначаемое отправителем пакета и предназначенное для определения корректной последовательности фрагментов при сборке пакета. Для фрагментированного пакета все фрагменты имеют одинаковый идентификатор.
- **Три бита флагов** — Первый бит должен быть всегда равен нулю, второй бит DF (don't fragment) определяет возможность фрагментации пакета и третий бит MF (more fragments) показывает, не является ли этот пакет последним в цепочке пакетов.
- **Смещение фрагмента** (Fragment Offset) — значение, определяющее позицию фрагмента в потоке данных. Смещение задается количеством восьмибайтовых блоков, поэтому это значение требует умножения на 8 для перевода в байты.
- **Время жизни** (TTL, time to live) — число маршрутизаторов, через которые может пройти этот пакет, прежде чем будет отброшен. При прохождении через маршрутизатор это число уменьшается на единицу. Если значение этого поля равно нулю, то пакет должен быть отброшен и отправителю будет послано сообщение Time Exceeded (ICMP тип 11 код 0).
- **Протокол** — идентификатор сетевого протокола следующего уровня указывает, данные какого

протокола содержит пакет, например, TCP, UDP, или ICMP (см. IANA protocol numbers и RFC 1700).

- **Контрольная сумма заголовка** (Header Checksum) — вычисляется по правилам RFC 1071.

20.3.4.2 Маршрутизация трафика

Маршрутизация — это одна из наиболее важных функций протокола IP. Она позволяет перенаправлять IP-пакеты между разными сетями, то есть управлять процессом объединения различных сетей в маршруты для прохождения пакетов от одного узла к другому.

Если сеть подключена к маршрутизатору непосредственно, он уже знает, как направить пакет в эту сеть. Если же сеть напрямую не подключена, администраторы должны настроить маршруты вручную (статическая маршрутизация) или включить динамическую маршрутизацию.

Динамическая маршрутизация — это протокол, определяющий взаимодействие маршрутизаторов между собой для автоматического распространения информации о маршрутах. Если на каком-то из маршрутизаторов в сети произойдет изменение в таблице маршрутов, он сможет по протоколу динамической маршрутизации автоматически проинформировать об этих изменениях своих соседей, а те передадут информацию дальше по цепочке. На небольших предприятиях используют в основном статическую маршрутизацию.

Рассмотрим процесс маршрутизации на примере проверки доступности узла с помощью утилиты ping, которая под капотом использует протокол ICMP. Все использованные в примере сети, IP и MAC-адреса являются вымышленными и любое совпадение с реальными адресами являются чистой случайностью)))

Предположим, у нас есть 2 узла с IP-адресами 192.168.10.2/24 и 192.168.20.2/24 соединённые кабелем с маршрутизатором, у которого один интерфейс подключен к сети 192.168.10.0/24, и его IP-адрес 192.168.10.1/24, а другой интерфейс подключён к сети 192.168.20.0/24 и его IP-адрес 192.168.20.1/24. На узлах 1 и 2 настроены шлюзы по умолчанию 192.168.10.1 и 192.168.20.1 соответственно.

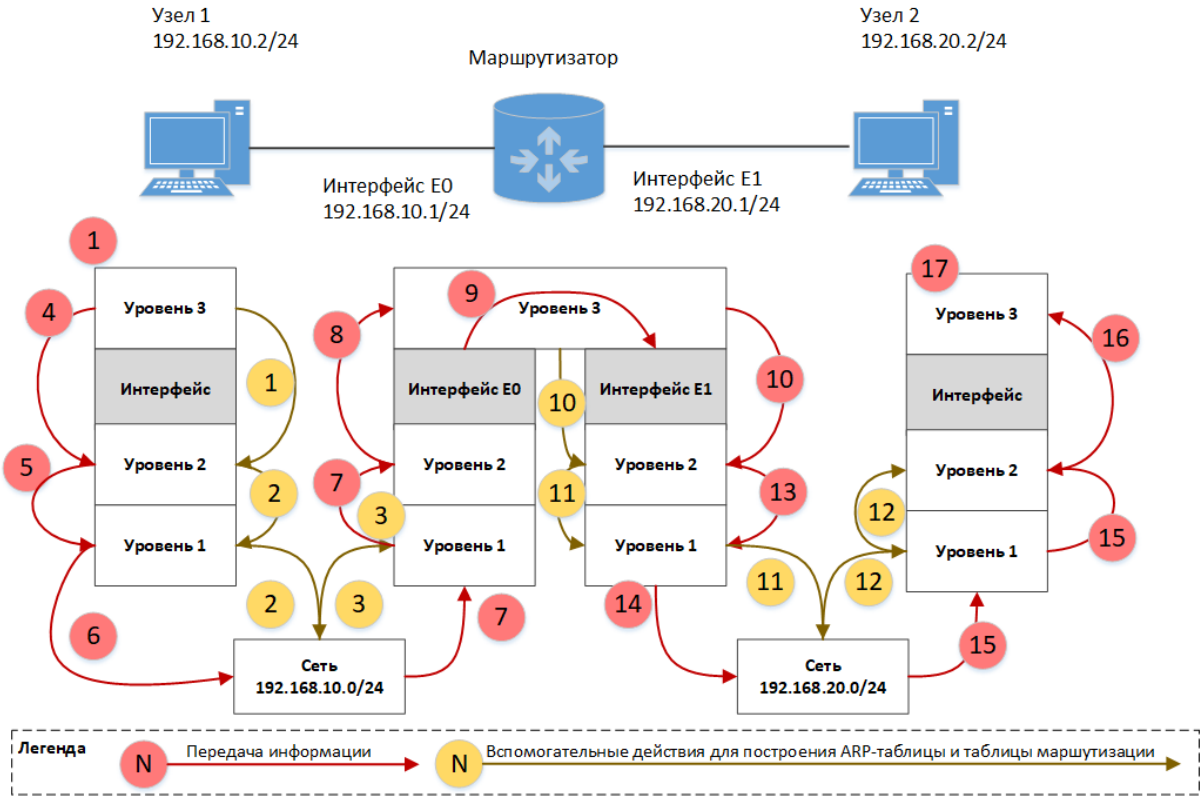


рис. 1.129 – Схема маршрутизации запроса ICMP на узел, находящийся в другой подсети

В таблице ниже приведен процесс маршрутизации ICMP пакета.

табл. 1.7 – Маршрутизация ICMP пакета

Шаг	Узел	Интерфейс	Действия
Шаг 1	Узел 1	192.168.10.2/24	ping 192.168.20.2. 1. Генерация пакета эхо-запроса ICMP к узлу 192.168.20.2/24. 2. Формируется IP-пакет, в который инкапсулирован ICMP-пакет. В заголовке пакета указаны IP-адреса источника и получателя. 3. Ядро операционной системы с учетом IP-адреса получателя выбирает маршрут из таблицы маршрутизации. В нашем случае это внешняя сеть (не присоединённая локально). 4. Протокол IP согласно своей таблице маршрутизации определяет следующий узел назначения. В нашем случае это шлюз по умолчанию 192.168.10.1/24 - интерфейс E0 маршрутизатора.

Процесс передачи ICMP пакета запускается в обратную сторону с той лишь разницей, что в ARP-кэшах устройств уже присутствуют все необходимые данные и выполнять ARP-запросы устройствам в этом случае не придется.

Если бы сети узла 1 и узла 2 не были бы «соседями» и на пути маршрута между ними были несколько сетей и маршрутизаторов, то логика прохождения пакетов все равно осталась бы прежней, увеличилось бы только количество промежуточных устройств и шагов обработки пакетов.

20.4 Работа сетевого стека в Linux

В современных операционных системах (таких как Linux, Windows, MacOS и далее по списку) основная часть стека сетевых протоколов реализована внутри ядра вплоть до транспортного уровня включительно. Работа в пространстве ядра позволяет получить прямой доступ к оборудованию и снизить накладные расходы на переключение контекста, что обеспечивает безопасную работу с сетевыми данными на высоких скоростях.

Функционирует это следующим образом: ядро Linux получает прерывания от сетевого адаптера и обрабатывает пакеты из очереди, пропуская их через канальный (Ethernet), сетевой (IP) и транспортный (TCP, UDP) уровни протоколов, перекладывая информацию в конечном итоге в буферы сокетов, откуда ее могут вычитать приложения с помощью стандартных системных вызовов, см. рис. 1.130.

К сетевым пакетам дополнительно применяются правила маршрутизации и фильтрации подсистемы Netfilter, которая позволяет подключать дополнительные модули ядра для управления сетевыми пакетами. Настройка таблиц маршрутизации выполняется, например, с помощью клиентских утилит ip и iproute2, а правила фильтрации и трансляции адресов можно настроить с помощью утилиты nft (ранее использовалась утилита iptables).

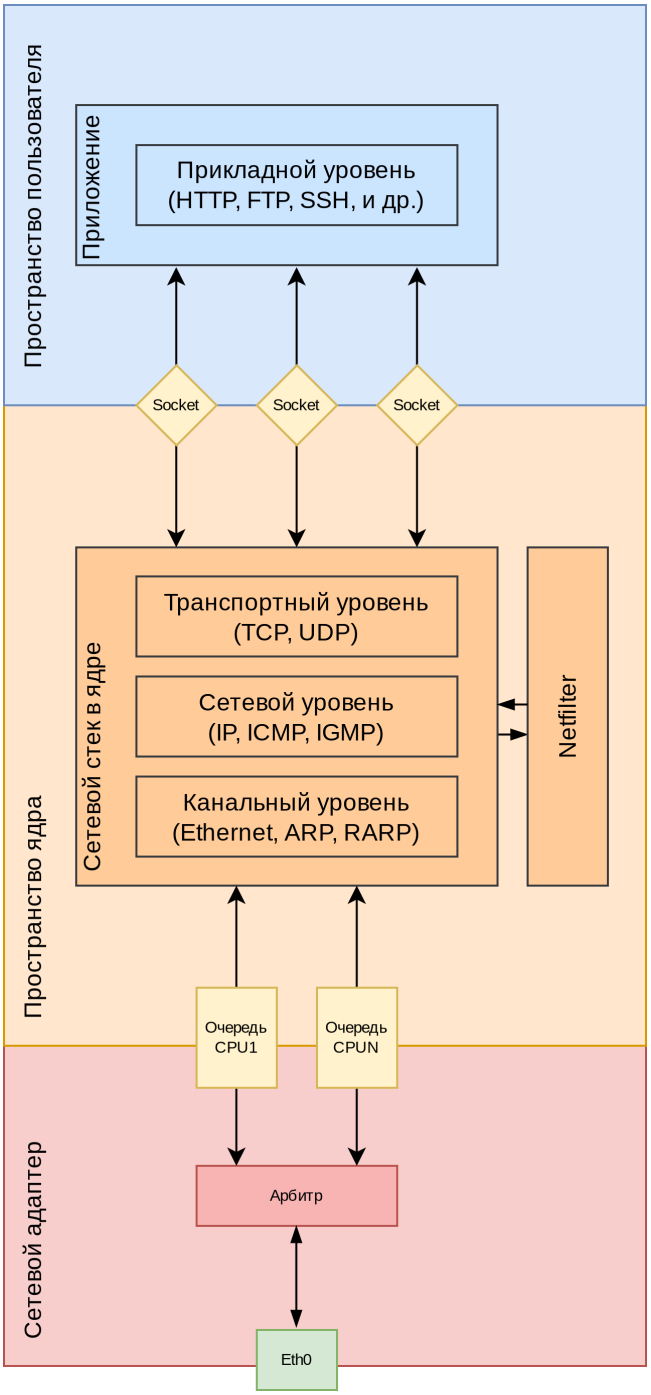


рис. 1.130 – Архитектура сетевого взаимодействия в системе Linux

Но чтобы не заниматься настройкой правил nftables на низком уровне, вы можете воспользоваться простым межсетевым экраном UFW (*англ. Uncomplicated Firewall*), для которого есть даже графический интерфейс GUFW, сопоставимый по функциональным возможностям с Windows Defender Firewall.

Кроме правил фильтрации, определенных пользователем, есть еще правила отбрасывания пакетов, предусмотренные протоколами. Например, если пакеты будут содержать IP-адрес, не принадлежащий устройству, они будут отброшены сетевым стеком ядра операционной системы, а если в них

будет MAC-адрес, не соответствующий тому, что настроен в адаптере, то они даже не дойдут до ядра, т.к. их срежет сетевой адаптер (если только на нем не включен «неразборчивый режим»).

Подобных оптимизаций в работе сетевого стека как на аппаратном, так и на программном уровне реализовано уже бессчетное количество, и они будут появляться дальше, так как требования бизнеса в части использования сетевых сервисов будут только расти. Но можно с уверенностью сказать, что сетевой стек Linux уже давно является зрелой технологией, способной обеспечить работу не только серверов, но и специализированных сетевых устройств.

Ядро Linux выбирают многие производители сетевого оборудования, например, D-Link (OpenWrt), MikroTik (RouterOS) и даже такие лидеры, как Cisco (IOS XE, IOS XR). Однако важна не только технология, но и умение ее настраивать. Поэтому продолжаем погружаться в технические детали.

20.4.1 Подготовка стека к работе

Чтобы на компьютере можно было использовать сетевой стек, сначала его нужно инициировать, поэтому в момент загрузки системы происходит следующее:

- Определяется модель сетевого адаптера и загружается соответствующий драйвер.
- Драйверу выделяется оперативная память для DMA (англ. direct memory access — прямой доступ к памяти), далее мы расскажем об этом подробнее.
- Создается поток для программных прерываний по одному на каждом из ЦПУ.
- В ядре создается обработчик программных прерываний.
- Создаются очереди для приема и передачи пакетов.
- На каждом из ЦПУ для NAPI (*англ. New API — новый API*) создается список опроса (*англ. pool list*), далее мы расскажем об этом подробнее.

После загрузки операционной системы требуется включить сетевой интерфейс в ядре. Сделать это можно вручную с помощью утилит ip и ifconfig или автоматически, используя один из следующих пакетов: ifupdown, NetworkManager, systemd.

20.4.1.1 Временное изменение настроек

Если вам нужно временно внести изменения в настройки соединений, вы можете воспользоваться командой `ip`.

Команда `ip link show` покажет краткую информацию о соединениях:

```
localadmin@astra:~$ ip link show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT␣
↪group default qlen 1000
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP mode␣
↪DEFAULT group default qlen 1000
   link/ether 08:00:27:19:4a:3a brd ff:ff:ff:ff:ff:ff
```

Команда `ip address` или просто `ip` а выведет основную информацию обо всех интерфейсах:

```
localadmin@astra:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default␣
↪qlen 1000
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
   inet 127.0.0.1/8 scope host lo
      valid_lft forever preferred_lft forever
   inet6 ::1/128 scope host
      valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group␣
↪default qlen 1000
   link/ether 08:00:27:19:4a:3a brd ff:ff:ff:ff:ff:ff
   inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic eth0
      valid_lft 85957sec preferred_lft 85957sec
   inet6 fe80::a00:27ff:fe19:4a3a/64 scope link
      valid_lft forever preferred_lft forever
```

Для временного изменения настроек интерфейса, например, для добавления еще одного IP-адреса с маской подсети можно воспользоваться командой `sudo ip address add 10.0.2.16/24 dev eth0`:

```
localadmin@astra:~$ sudo ip address add 10.0.2.16/24 dev eth0
localadmin@astra:~$ ip address show dev eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group␣
↪default qlen 1000
   link/ether 08:00:27:19:4a:3a brd ff:ff:ff:ff:ff:ff
   inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic eth0
      valid_lft 85391sec preferred_lft 85391sec
   inet 10.0.2.16/24 scope global secondary eth0
      valid_lft forever preferred_lft forever
   inet6 fe80::a00:27ff:fe19:4a3a/64 scope link
      valid_lft forever preferred_lft forever
```

Удалим IP-адрес вместе с маской командой `sudo ip address del 10.0.2.16/24 dev eth0`:

```
localadmin@astra:~$ sudo ip address del 10.0.2.16/24 dev eth0
```

Выключим и включим интерфейс eth0 командой `sudo ip link set eth0 down && sudo ip link set eth0 up`:

```
localadmin@astra:~$ sudo ip link set eth0 down && sudo ip link set eth0 up
```

Перезагрузим службу Networking командой `sudo systemctl restart networking.service`:

```
localadmin@astra:~$ sudo systemctl restart networking.service
```

20.4.1.2 Постоянное изменение настроек с помощью ifupdown

Пакет ifupdown (англ. **i**nter**f**ace up/down — включение/выключение интерфейса) — это проверенное временем традиционное решение для управления сетью, которое до сих пор используется во многих дистрибутивах на базе Linux.

Автоматическая настройка сети выполняется службой networking.service в соответствии с настройками из файла `/etc/network/interfaces`. Для конфигурирования интерфейсов внутри ядра служба использует утилиты ifup/ifdown, которые являются обертками поверх утилиты ifconfig. Каждый раз при включении/выключении службы она делает `ifup` / `ifdown` всех интерфейсов, отмеченных опцией auto.

Для автоматического получения IP-адреса служба *networking* использует клиент *isc-dhcp-client*. Для удобства управления настройками преобразования доменных имен дополнительно может устанавливаться пакет *resolvconf*. В этом случае настройки можно будет привязать к интерфейсам непосредственно в конфигурационном файле interfaces.

В системе Astra Linux пакет ifupdown устанавливается по умолчанию и может использоваться на серверах для настройки интерфейсов, имеющих постоянное подключение к компьютерной сети.

Файл `/etc/network/interfaces` используется командами `ifup` и `ifdown` для конфигурирования сетевых интерфейсов:

```
localadmin@astra:~$ cat /etc/network/interfaces
source /etc/network/interfaces.d/*

auto lo
iface lo inet loopback

allow-hotplug eth0
auto eth0
iface eth0 inet static
    address 10.0.1.11
    netmask 255.255.255.0
    gateway 10.0.1.1
```

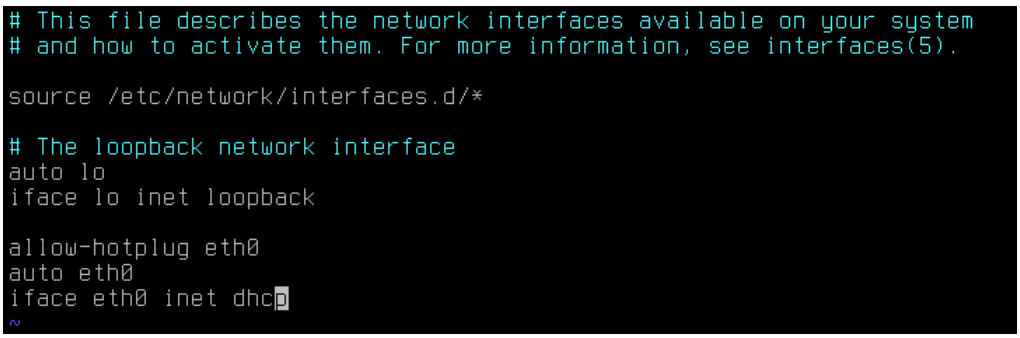
Комментарии по использованным инструкциям:

- **allow-hotplug eth0** — указывает, что нужно выполнять автоматический перезапуск интерфейса при его падении.
- **auto eth0** — строка, начинающаяся со слова «auto», задает интерфейс, который должен подниматься при вызове команды `ifup -a`, которая вызывается автоматически при запуске службы

networking.

- **iface eth0 inet static** — строка со словом «iface» начинает группу строк, отвечающих за настройку указанного интерфейса:
  - **inet** или **inet6** — указывает, какой протокол будет использоваться (IPv4 или IPv6).
  - **static** или **dhcp** — указывает способ назначения настроек (вручную или динамически).
  - **address**, **netmask**, **gateway** — задают IP-адрес, маску и шлюз по умолчанию для интерфейса, указанного в предшествующей ей строке «iface», если для него выбран способ назначения настроек «static».
  - Если вам требуется переопределить значение MAC-адреса, то новое значение можно задать строкой hwaddress ether 00:11:22:33:44:55

Внесем изменения в файл `/etc/network/interfaces` для настройки сетевого интерфейса eth0 через службу networking.



Сохраним файл и перезапустим службу networking.service `sudo systemctl restart networking.service`. Если настройки были внесены корректно, то запуск службы пройдет без ошибок. Проверим её статус командой `sudo systemctl status networking.service`.

```
localadmin@astra:~$ sudo systemctl restart networking.service
localadmin@astra:~$ sudo systemctl status networking.service
O networking.service - Raise network interfaces
  Loaded: loaded (/lib/systemd/system/networking.service; enabled; vendor preset:␣
↪enabled
  Active: active (exited) since Wed 2023-09-13 20:12:43 MSK; 16s ago
    Docs: man:interfaces(5)
 Process: 24569 ExecStart=/sbin/ifup -a --read-environment (code=exited, status=0/
↪SUCCESS
Main PID: 24569 (code=exited, status=0/SUCCESS)
   Tasks: 2 (limit: 2255)
  Memory: 5.7M
     CPU: 207ms
  CGroup: /system.slice/networking.service
          └─24589 /sbin/dhclient -4 -v -i -pf /run/dhclient.eth0.pid -lf /var/lib/
↪dhcp/dh
```

(continues on next page)

(продолжение с предыдущей страницы)

```
сен 13 20:12:43 astra dhclient[24589]: Sending on Socket/fallback
сен 13 20:12:43 astra dhclient[24589]: Created duid "\\000\\001\\000\\001,...
сен 13 20:12:43 astra dhclient[24589]: DHCPDISCOVER on eth0 to 255.255.255.255 port␣
↪67 int
сен 13 20:12:43 astra dhclient[24589]: DHCPOFFER of 10.0.2.15 from 10.0.2.2
сен 13 20:12:43 astra dhclient[24589]: DHCPREQUEST for 10.0.2.15 on eth0 to 255.255.
↪255.25
сен 13 20:12:43 astra dhclient[24589]: DHCPACK of 10.0.2.15 from 10.0.2.2
сен 13 20:12:43 astra ifup[24569]: RTNETLINK answers: File exists
сен 13 20:12:43 astra dhclient[24589]: bound to 10.0.2.15 -- renewal in 39567 seconds.
сен 13 20:12:43 astra ifup[24569]: bound to 10.0.2.15 -- renewal in 39567 seconds.
сен 13 20:12:43 astra systemd[1]: Started Raise network interfaces.
```

Не забудьте до перезапуска службы networking отключить проводное соединение в службе NetworkManager, а лучше отключить и замаскировать эту службу полностью:

```
localadmin@astra:~$ ls /etc/network/interfaces.d/
localadmin@astra:~$ sudo systemctl stop NetworkManager
localadmin@astra:~$ sudo systemctl disable NetworkManager
Removed /etc/systemd/system/dbus-org.freedesktop.nm-dispatcher.service.
Removed /etc/systemd/system/multi-user.target.wants/NetworkManager.service.
Removed /etc/systemd/system/network-online.target.wants/NetworkManager-wait-online.
↪service.
localadmin@astra:~$ sudo systemctl mask NetworkManager
Created symlink /etc/systemd/system/NetworkManager.service → /dev/null.
```

Теперь апплет NetworkManager не активен и не показывает соединения:

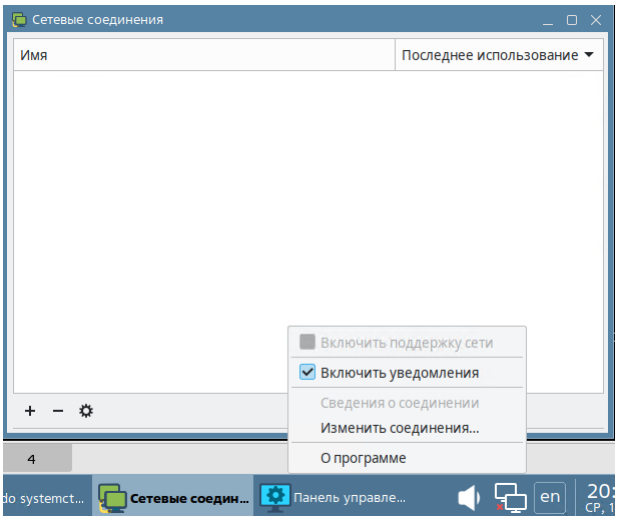


рис. 1.131 – Состояние апплета «выключено»

20.4.1.3 Постоянное изменение настроек с помощью NetworkManager

Служба NetworkManager — это современный высокоуровневый инструмент для управления сетевыми подключениями, который использует *libudev* и другие службы ядра Linux для автоматической перенастройки сетевых интерфейсов при перемещении портативных устройств между разными сетями. NetworkManager позволяет работать как с проводными, так и беспроводными сетевыми интерфейсами и некоторыми типами VPN-соединений.

Пакет network-manager состоит из одноименной службы, доступ к которой осуществляется по шине dbus (права суперпользователя не требуются), и ряда клиентских приложений, таких как:

- **nmcli** — командный интерфейс
- **nmtui** — псевдографический интерфейс
- **nm-applet** — графический интерфейс, который по функциональности не уступает приложению «Свойства сетевого подключения» от Windows.

Для автоматического получения IP-адреса служба NetworkManager по умолчанию использует свой собственный DHCP-клиент, но при необходимости ее можно переориентировать на использование других клиентов.

Служба NetworkManager автоматически управляет также содержимым файла `/etc/resolv.conf`, записывая туда значение, назначенное интерфейсу или полученное от DHCP-сервера.

```
localadmin@astra:~$ cat /etc/resolv.conf
# Generated by NetworkManager
nameserver 77.88.8.8
```

При установке Astra Linux с графической оболочкой Fly пакет network-manager будет инсталлирован автоматически. На серверах со статичными настройками сети это может показаться излишним, и вы захотите сэкономить пару десятков мегабайт оперативной памяти, но предоставляемые этим пакетом удобства настолько значительны, что RHEL рекомендует использовать NetworkManager в том числе и на серверах.

Конфигурационный файл NetworkManager.conf

Настройки NetworkManager задаются в файле `/etc/NetworkManager/NetworkManager.conf`.

```
[main]
plugins=ifupdown,keyfile

[ifupdown]
managed=false
```

На сервере Astra Linux с графическим интерфейсом все сетевые интерфейсы по умолчанию на-

ходятся под управлением NetworkManager за исключением lo – интерфейса обратной петли. Это обеспечивается параметром «managed=false» в секции «[ifupdown]», запрещающей управлять интерфейсами, которые находятся под управлением службы Networking.

```
localadmin@astra:~$ cat /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback
```

Утилита командной строки Nmcli

Утилита nmcli (*англ. Network Manager Command Line Interface*) позволяет настраивать NetworkManager из командной строки. Синтаксис этой утилиты:

```
$ nmcli <опции> <раздел> <действия>
```

Всего существуют восемь разделов со своими наборами команд (действиями):

- **help** – выдает справку о командах ncmcli и их использовании.
- **general** – возвращает статус NetworkManager и глобальную конфигурацию.
- **networking** – содержит команды для запроса состояния сетевого подключения и включения/отключения подключений.
- **radio** – содержит команды для запроса состояния подключения к сети WiFi и включения/отключения подключений.
- **monitor** – содержит команды для мониторинга активности NetworkManager и наблюдения за изменениями состояния сетевых подключений.
- **connection** – содержит команды для управления сетевыми интерфейсами, для добавления новых соединений и удаления существующих.
- **device** – используется в основном для изменения параметров, связанных с устройствами (например, имени интерфейса) или для подключения устройств с использованием существующего соединения.
- **secret** – регистрирует nmcli в качестве «секретного агента» NetworkManager, который прослушивает тайные сообщения. Эта секция используется редко, потому что nmcli при подключении к сетям по умолчанию работает именно так.

Посмотреть статус NetworkManager можно командой `nmcli general`.

```
localadmin@astra:~$ nmcli general
STATE       CONNECTIVITY  WIFI-HW   WIFI      WWAN-HW   WWAN
подключено  полностью     включен   включен   включен   включен
```

Посмотреть сводную информацию по соединению можно вывести командой `nmcli device show`.

```
localadmin@astra:~$ nmcli connection show
NAME                    UUID                                  TYPE            DEVICE
Проводное соединение 1  26c78985-a471-3663-98bb-9afbf978d59c  ethernet eth0
```

Подробные настройки интерфейса можно командой `nmcli device show eth0`.

```
localadmin@astra:~$ nmcli device show eth0
GENERAL.DEVICE:                         eth0
GENERAL.TYPE:                           ethernet
GENERAL.HWADDR:                         08:00:27:19:4A:3A
GENERAL.MTU:                            1500
GENERAL.STATE:                          100 (подключено)
GENERAL.CONNECTION:                     Проводное соединение 1
GENERAL.CON-PATH:                       /org/freedesktop/NetworkManager/
↪ActiveConnection/1
WIRED-PROPERTIES.CARRIER:               вкл.
IP4.ADDRESS[1]:                         10.0.2.15/24
IP4.GATEWAY:                            10.0.2.2
IP4.ROUTE[1]:                           dst = 0.0.0.0/0, nh = 10.0.2.2, mt = 100
IP4.ROUTE[2]:                           dst = 10.0.2.0/24, nh = 0.0.0.0, mt = 100
IP4.DNS[1]:                             8.8.8.8
IP6.ADDRESS[1]:                         fe80::2e8a:2a2:9c75:15b9/64
IP6.GATEWAY:                            --
IP6.ROUTE[1]:                           dst = fe80::/64, nh = ::, mt = 100
```

Подробные настройки соединения можно посмотреть командой `nmcli device show "Проводное соединение 1"`. Как вы можете заметить, у соединения настроек значительно больше, чем у устройства.

```
localadmin@astra:~$ nmcli connection show "Проводное соединение 1"
connection.id:                          Проводное соединение 1
connection.uuid:                        26c78985-a471-3663-98bb-9afbf978d59c
connection.stable-id:                   --
connection.type:                        802-3-ethernet
connection.interface-name:              --
connection.autoconnect:                 да
connection.autoconnect-priority:        -999
connection.autoconnect-retries:         -1 (default)
...
```

Добавим сетевое соединение командой `nmcli connection add type ethernet ifname eth1`, где type указывает тип соединения ethernet, а имя ifname задает имя интерфейса.

```
localadmin@astra:~$ sudo nmcli connection add type ethernet ifname eth1
Соединение «ethernet-eth1» (4b3a041f-027d-4fae-903d-d1d953f268a5) добавлено.
localadmin@astra:~$ nmcli connection show
```

(continues on next page)

(продолжение с предыдущей страницы)

```
NAME                    UUID                                  TYPE      DEVICE
Проводное соединение 1  26c78985-a471-3663-98bb-9afbf978d59c  ethernet  eth0
ethernet-eth1           4b3a041f-027d-4fae-903d-d1d953f268a5  ethernet  --
```

Задать значение IP-адреса можно командой `nmcli connection modify ethernet-eth1 ipv4.address 192.168.0.2/24`.

```
localadmin@astra:~$ nmcli connection modify ethernet-eth1 ipv4.address 192.168.0.2/24
```

Изменить способ назначения IP-адреса «вручную» можно командой `nmcli connection modify ethernet-eth1 ipv4.method manual`.

```
localadmin@astra:~$ nmcli connection modify ethernet-eth1 ipv4.method manual
```

Чтобы вернуть автоматическое назначение IP-адреса через DHCP, необходимо выполнить команду `nmcli connection modify ethernet-eth1 ipv4.method auto`

```
localadmin@astra:~$ nmcli connection modify ethernet-eth1 ipv4.method auto
```

Для того чтобы изменения настроек вступили в силу, нужно перезапустить соединение.

```
localadmin@astra:~$ nmcli dev reapply eth0
Соединение снова применено к устройству «eth0».
```

Можно также выключить и включить соединение командой

`nmcli connection down "Проводное соединение 1" && nmcli connection up "Проводное соединение 1"`

```
localadmin@astra:~$ nmcli connection down "Проводное соединение 1" && nmcli␣
↪connection up "Проводное соединение 1"
Соединение «Проводное соединение 1» успешно отключено (активный адрес D-Bus: /org/
↪freedesktop/NetworkManager/ActiveConnection/1)
Соединение успешно активировано (адрес действующего D-Bus: /org/freedesktop/
↪NetworkManager /ActiveConnection/2)
```

Утилита Nmtui

Псевдографическая утилита nmtui (англ. Network Manager Text User Interface) будет полезна при настройке соединения на сервере или на рабочей станции в режиме командной строки. Для запуска наберите в консоли команду `nmtui`.

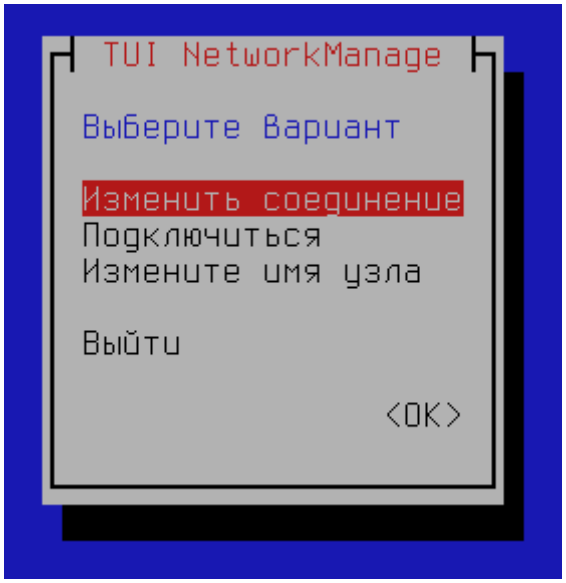


рис. 1.132 – Начало работы с утилитой nmtui

Выберем пункт *Изменить соединение → ethernet-eth1 → Изменить* .

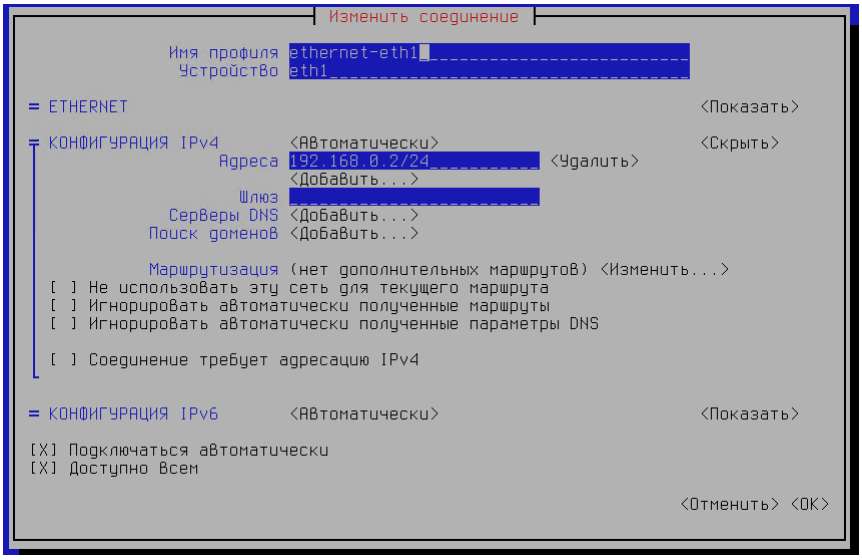


рис. 1.133 – Настройки соединения

Изменим IP-адрес на 192.168.0.3/24, добавим шлюз по умолчанию 192.168.0.1 и DNS-сервер Яндекса 77.88.8.8.

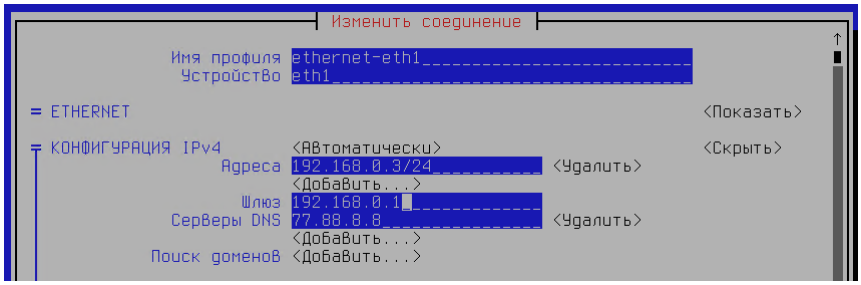


рис. 1.134 – Настройки соединения

Сохраним настройки и завершим работу с утилитой. Убедимся, что настройки были сохранены.

```
localadmin@astra:~$ nmcli connection show ethernet-eth1 | grep ipv4
ipv4.method:                            auto
ipv4.dns:                               77.88.8.8
ipv4.dns-search:                        --
ipv4.dns-options:                       --
ipv4.dns-priority:                      0
ipv4.addresses:                         192.168.0.3/24
ipv4.gateway:                           192.168.0.1
ipv4.routes:                            --
ipv4.route-metric:                      -1
ipv4.route-table:                       0 (unspec)
ipv4.routing-rules:                     --
ipv4.ignore-auto-routes:                нет
ipv4.ignore-auto-dns:                   нет
ipv4.dhcp-client-id:                    --
ipv4.dhcp-iaid:                         --
ipv4.dhcp-timeout:                      0 (default)
ipv4.dhcp-send-hostname:                да
ipv4.dhcp-hostname:                     --
ipv4.dhcp-fqdn:                         --
ipv4.dhcp-hostname-flags:               0x0 (none)
ipv4.never-default:                     нет
ipv4.may-fail:                          да
ipv4.dad-timeout:                       -1 (default)
ipv4.dhcp-vendor-class-identifier:      --
ipv4.dhcp-reject-servers:               --
```

Графический интерфейс nm-connection-editor

Запустить графический интерфейс для редактирования настроек соединений можно из консоли командой `nm-connection-editor`, через меню *Пуск → Панель управления → Сеть → Сетевые соединения* или контекстное меню апплета в области уведомлений.

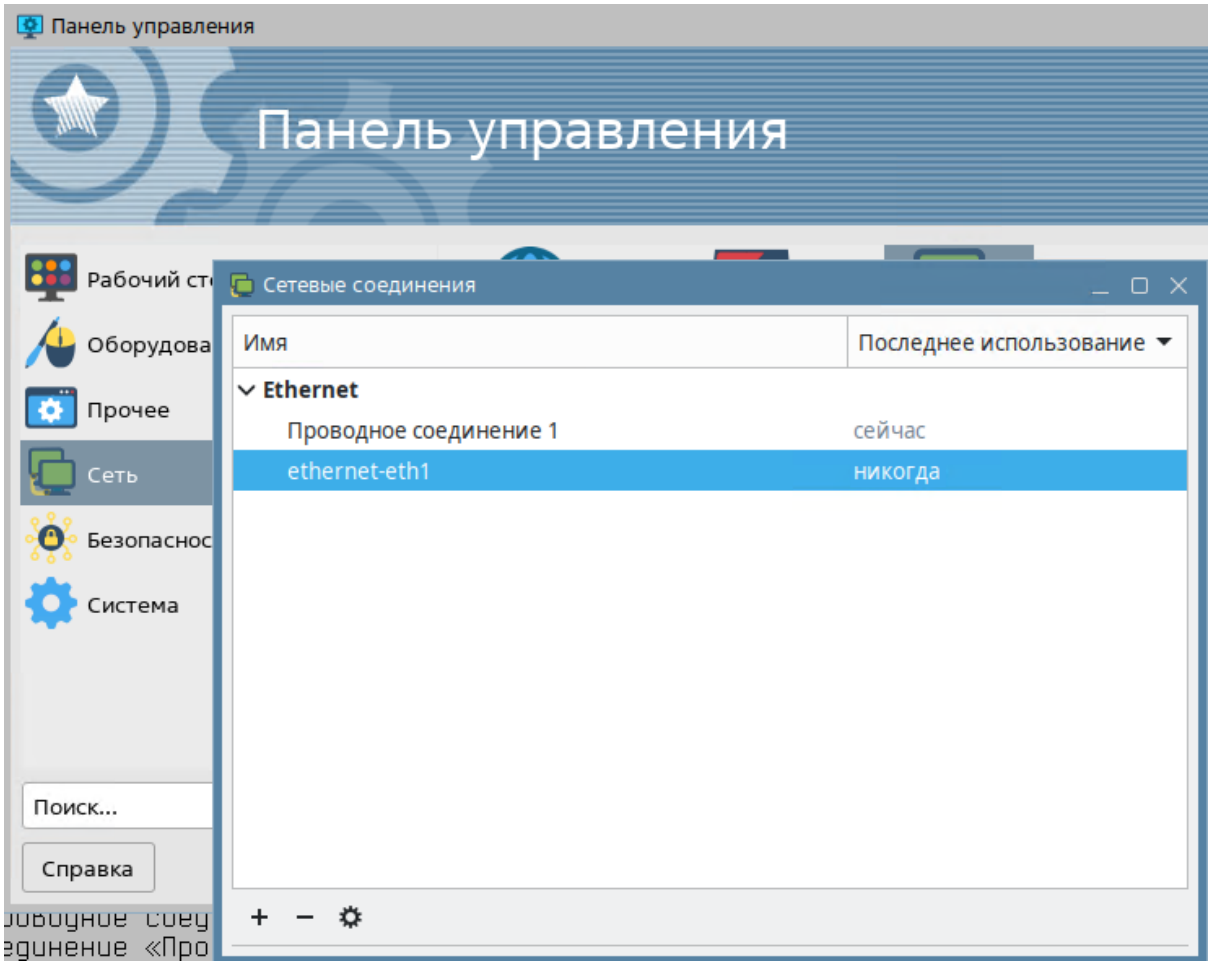


рис. 1.135 – Утилита «Сетевые соединения»

Утилита имеет интуитивно понятный интерфейс. Представлен список сетевых соединений, их статус и кнопки управления *добавить*, *удалить* и *настроить соединение*.

Дважды кликнем по соединению *Проводное соединение 1*. Перед нами появится окно «Изменение Проводное соединение 1», в котором есть несколько вкладок.

1. Вкладка **Основное** – общая настройка всех типов подключений и их приоритетов.

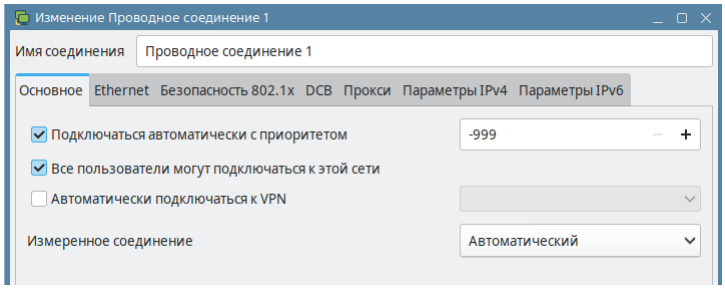


рис. 1.136 – Основные настройки проводного соединения

2. Вкладка **Ethernet** (название зависит от типа подключения) — доступны параметры MAC-

адрес, MTU и другие. Без необходимости их лучше не менять и использовать значения по умолчанию.

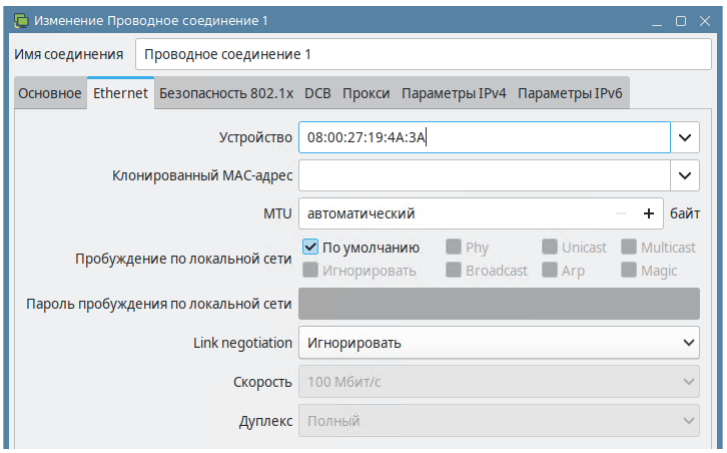


рис. 1.137 – Настройки Ethernet

3. Вкладка **Безопасность 802.1x** – подключение к сети с использованием сервера аутентификации.

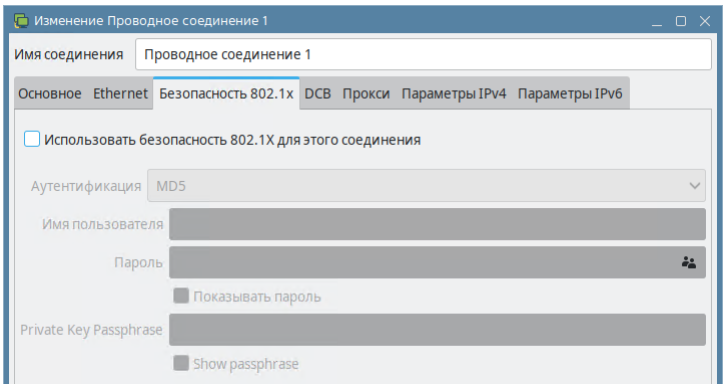


рис. 1.138 – Настройки безопасности

4. Вкладка **DCB** – подключение к сети хранения данных по протоколу Datacenter Bridging.

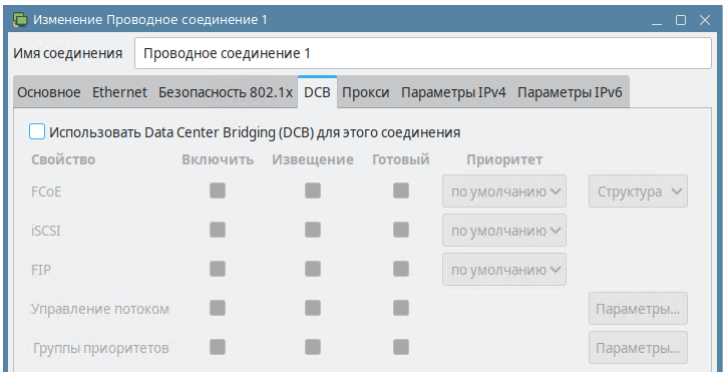


рис. 1.139 – Настройки DCB

5. Вкладка **Прокси** – настройка подключения через прокси-сервер.

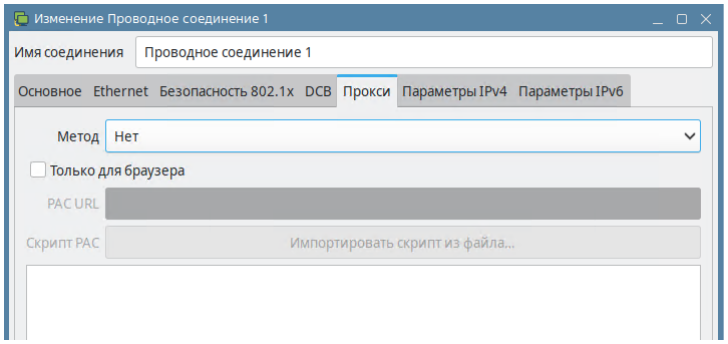


рис. 1.140 – Настройки прокси

6. Вкладка **Параметры IPv4** – выбор метода установки параметров протокола IPv4 и их установка. Доступные для изменения параметры меняются в зависимости от выбранного метода. На этой вкладке можно задавать IP-адреса сетевого интерфейса, маски подсетей, серверы DNS, поисковый домен и другое.

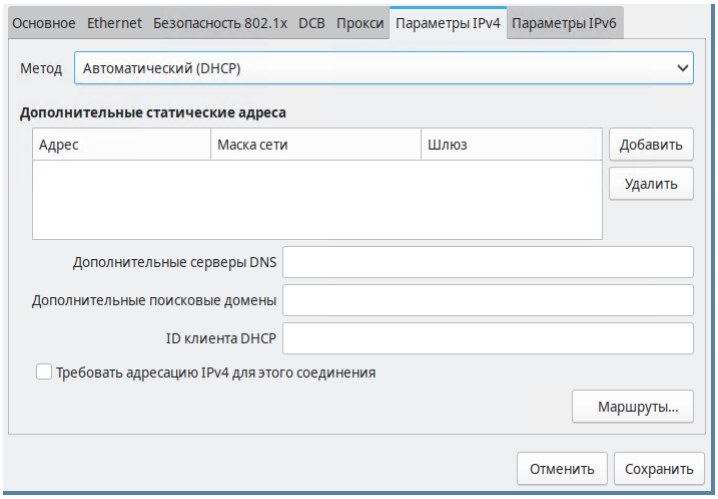


рис. 1.141 – Настройка IPv4

По нажатию на кнопку «Маршруты» открывает окно, в котором доступно создание и редактирование статических маршрутов для этого соединения, что особенно актуально для всякого рода VPN-соединений.

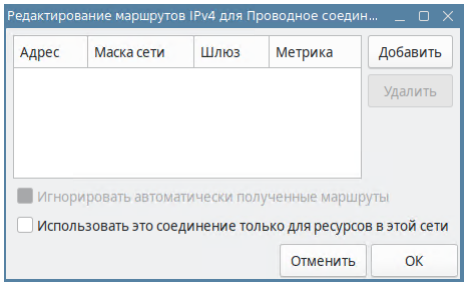


рис. 1.142 – Настройка маршрутов

Внимание: При изменении DNS-серверов через NetworkManager, служба вносит изменения в файл `/etc/resolv.conf`, перетирая все его содержимое.

7. Вкладка **Параметры IPv4** – выбор метода и параметров для протокола IPv6.

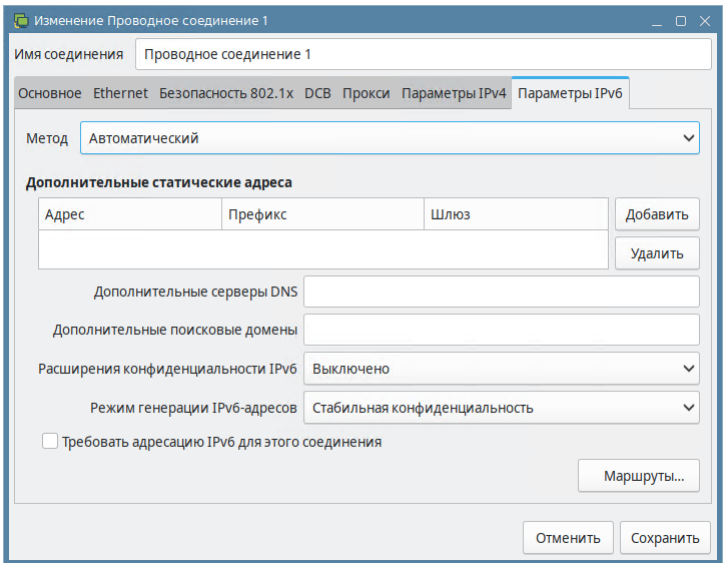


рис. 1.143 – Настройка IPv6

Чтобы открыть окно «Сведения о текущем соединении», выберите соответствующий пункт в контекстном меню апплета, иконка которого находится в трее в правом нижнем углу экрана.

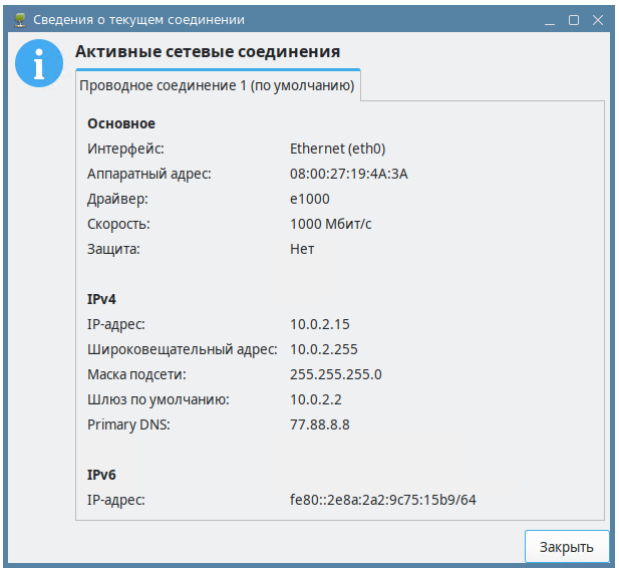


рис. 1.144 – Просмотр сведений о соединении

20.4.1.4 Постоянное изменение настроек с помощью systemd

В systemd есть служба systemd-networkd, с помощью которой можно вносить постоянные настройки конфигурации сети. Файлы настроек этой службы находятся в каталоге `/etc/systemd/network`.

Служба считается преемницей ifupdown и подходит для настройки как простых подключений, так и сложных конфигураций. Она обладает рядом преимуществ, например, поддерживает новые сетевые функции и не задерживает загрузку системы для проверки сетевых интерфейсов. Однако, несмотря на это, в простых сценариях настройки сети консервативный мир Linux продолжает активно использовать более простую службу networking.

В операционной системе Astra Linux служба systemd-networkd установлена, но по умолчанию деактивирована.

20.4.2 Физический уровень

На физическом уровне обработка входящих пакетов выполняется в следующем порядке:

- Все начинается с поступления цифровых сигналов на сетевую карту, где они преобразуются в двоичный поток данных. По сигнатуре Ethernet-кадра и его заголовку сетевая карта вычленяет кадр из потока.
- Сетевая карта проверяет контрольную сумму Ethernet-кадра и MAC-адрес устройства, которому он адресован. Если проверки пройдены успешно, то корректный кадр помещается в буфер сетевого устройства.
- Далее с использованием технологии DMA (*англ. direct memory access, прямой доступ к памяти*) кадр копируется в кольцевой буфер ядра или так называемую очередь. Размер выделяемой памяти, как правило, определяется двумя параметрами: количеством буферов (задается через настройки сетевой карты) и размером MTU (*англ. maximum transfer unit — максимальный размер передаваемого кадра*).
- Если обработка очереди находится в обычном режиме, то сетевая карта генерирует аппаратное прерывание IRQ (*англ. Interrupt Request — запрос на прерывание*), что вынуждает процессор переключиться на обработку новых данных. Если очередь переведена в режим опроса, то данные вычитываются процессором при следующем цикле обработки данных.

Вы можете изменить, например, MAC-адрес и значение MTU. Относительно MAC-адреса мы уже знаем, что это должно быть просто уникальное значение в пределах сети, а в части MTU обратим ваше внимание на одну довольно сложную проблему.

Если в вашей сети пинги и трассировки проходят «на ура», а сайты не загружаются, то можем вас

поздравить, вы столкнулись с так называемой черной дырой, которая ни разу не сверхмассивная звезда в космосе, а нарушение работы механизма согласования значения MTU.

Следует понимать, что вы можете выставить значение MTU на конкретном компьютере, но приемник и передатчик в сети должны действовать согласованно, поэтому фактическое значение MTU обычно определяется автоматически в момент установления связи, и эта процедура называется MTU discovery.

Для согласования значения MTU используется протокол ICMP: передатчик отправляет приемнику Ethernet-кадр с битом DF (не фрагментировать) в IP-заголовке и ждет, пока какой-то из узлов на пути следования кадра не отбросит его и не вернет ICMP-сообщение с требованием выполнить фрагментацию. Получив такое сообщение, передатчик уменьшает MTU и повторяет процедуру, пока значение MTU не станет достаточно маленьким, чтобы кадр прошел весь путь без фрагментации.

Но механизм согласования MTU может сломаться, если в сети присутствуют черные дыры (*Path MTU Discovery Black Hole*) — это такие узлы, которые отбрасывают большие кадры, не отправляя никаких ICMP-сообщений. Некоторые реализации автоматического согласования MTU пытаются обойти эту проблему и, не получив ответа, делают предположение, что это из-за MTU, а не по причине перегрузки канала. Но если в вашем случае это не сработало, то следует попытаться подобрать допустимое значение MTU вручную.

20.4.3 Уровень очередей

20.4.3.1 Получение пакетов

Когда скорость сетевого трафика исчислялась килобайтами в секунду, механизм обычных прерываний эффективно справлялся со своей задачей. Но прогресс целенаправленно вел индустрию к мега- и гигабитам, поэтому одной из оптимизаций стала технология NAPI (англ. New API — новый API), идея которой состояла в том, чтобы отложить обработку входящих сообщений до тех пор, пока их не наберется достаточное количество, чтобы сэкономить ресурсы на ненужных переключениях.

В технологии NAPI сохраняется баланс между двумя подходами: по умолчанию используется режим, управляемый прерываниями, а если скорость входящих пакетов превышает определенный порог, то сетевая карта переключается на режим опросов, и ядро начинает периодически обрабатывать входящие пакеты большими пачками, сокращая накладные расходы на прерывания, см. рис. 28. Когда нагрузка возвращается в норму, сетевая карта опять возвращается в режим прерываний, чтобы обеспечить более высокую скорость реакции.

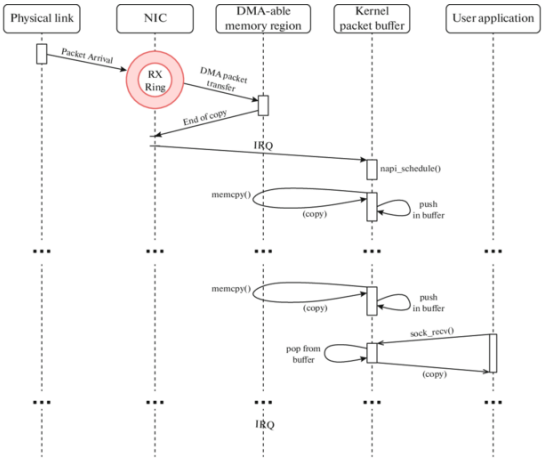


рис. 1.145 – Схема обработки пакетов NAPI

При использовании кольцевого буфера новые кадры не отклоняются, а замещают собой старые. При необходимости приложения запросят повторную отправку потерянных кадров. Для оптимизации потребления ресурсов и ускорения обработки применяются и другие технологии:

- Создание нескольких очередей, обрабатываемых на разных процессорах для распределения нагрузки (*англ. Receive Packet Steering, RPS — распределение принимаемых пакетов*).
- Распределение обработки пакетов между очередями RPS для повышения частоты попадания в кэш ЦПУ (*англ. Receive Flow Steering, RFS — распределение принимаемых потоков*).
- Направление пакетов в очередь, закрепленную за ядром ЦПУ, которое обрабатывает данные из этого потока (*англ. Accelerated Receive Flow Steering, aRFS — ускоренное распределение принимаемых потоков*).

На уровне очередей вы можете изменить длину очереди, включить/отключить использование оптимизаций RPS, RFS, aRFS.

20.4.3.2 Передача пакетов

При передаче пакетов каждый пакет обрабатывается процессором отдельно:

- После того как данные были упакованы в IP-пакет, в очередь помещается указатель на этот пакет. В случае, если очередь переполнена, пакеты отбрасываются.
- Драйверу сетевой карты сообщается, что пакеты готовы для отправки.
- Сетевая карта отправляет пакет или группу пакетов и сообщает об этом системе через генерацию прерывания.

20.4.4 Уровень протоколов

На уровне протоколов выполняется сборка фрагментированных IP и TCP-пакетов и проверка их целостности. Входящие пакеты проходят по сетевому стеку вверх, пока данные не достигнут буфера сокета, чтобы стать доступными пользовательскому приложению, а исходящие данные совершают тот же путь только в обратном направлении.

Статистику по работе протоколов можно получить из файлов `/proc/net/netstat`, `/proc/net/tcp`, `/proc/net/udp` и др., но проще будет воспользоваться командой `netstat -s`, которая предоставит подробную информацию по каждому из параметров.

```
localadmin@astra:~$ netstat -s
Ip:
    Forwarding: 2
    528616 total packets received
    0 forwarded
    0 incoming packets discarded
    528529 incoming packets delivered
    330271 requests sent out
    170 outgoing packets dropped
    37 dropped because of missing route
Icmp:
    1007 ICMP messages received
    0 input ICMP message failed
    ICMP input histogram:
```

Но самое интересное происходит, конечно, в части обработки пакетов правилами маршрутизации и фильтрации подсистемы Netfilter. Чтобы описать этот механизм в полном объеме, потребуется увеличить объем данного модуля в несколько раз, поэтому мы обозначим только общие моменты, которые помогут сориентироваться и задать общее направление, если эта тема вас заинтересует.

Для эффективной работы с правилами Netfilter вы должны четко понимать диаграмму движения пакетов, см. рис. 1.146.

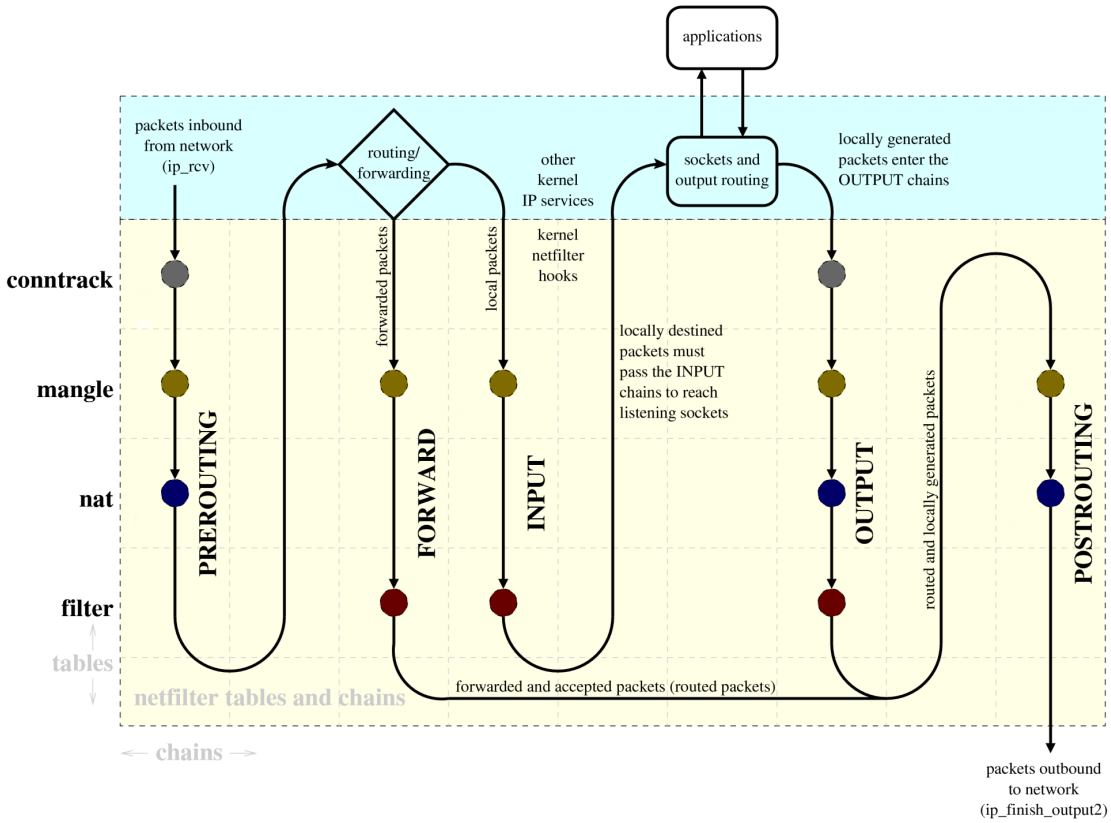


рис. 1.146 – Диаграмма потока пакетов Netfilter

Работа межсетевого экрана определяется правилами, каждое из которых состоит из критерия и действия. Правила объединяются в три основные цепочки:

- **INPUT** – входящие пакеты, которые были направленные одному из сетевых приложений, работающих на компьютере.
- **OUTPUT** – исходящие пакеты, которые были созданы одним из сетевых приложений, работающих на компьютере, для другого узла сети.
- **FORWARD** – входящие пакеты, которые компьютер должен передать дальше по сети, используя свои правила маршрутизации.

  Обратите внимание, что по умолчанию пересылка пакетов отключена и для ее включения нужно выполнить команду:

```
$ echo 1 | sudo tee /proc/sys/net/ipv4/ip_forward
```

Дополнительно есть цепочки PREROUTING и POSTROUTING, через которые пакеты проходят до и после фильтрации соответственно.

Независимые цепочки объединяются в таблицы, каждая из которых решает определенную задачу: цепочки таблицы filter отвечают за фильтрацию, цепочки таблицы nat — за преобразование сетевых адресов (NAT).

Сильной стороной Netfilter является то, что он имеет модуль conntrack, который позволяет делать stateful-фильтрацию, т.е. при обработке отдельных пакетов использовать информацию не только из заголовков (источник, получатель) или содержимого пакетов, но и данные о соединении. При настройке правил вы можете использовать следующие состояния соединений:

- **NEW** (новый) — означает, что пакет запрашивает установление нового соединения
- **ESTABLISHED** (установлено) — означает, что пакет связан с существующим соединением, в рамках которого пакеты ходили в обоих направлениях.
- **RELATED** (связанный) — означает, что пакет запрашивает установление нового подключения, но оно является частью существующего, как в случае с активными FTP-соединениями, когда подключение идёт к порту 20, а передача данных выполняется через динамический порт.

20.4.5 Механизм сокетов

При реализации сетевого стека в системе BSD (*англ. Berkeley Software Distribution*) разработчики создали новый вид программного интерфейса, который позволил связывать IP-адрес, тип транспортного протокола и номер порта, образуя таким образом «разъем для подключения».

Как физические разъемы делятся на вилки и розетки, так сокеты бывают клиентскими (исходящими) и серверными (входящими). Любое серверное приложение может функцией socket() создать «входящий» серверный сокет, связать его с неким IP-адресом и портом функцией bind(), после чего функцией *listen()* перейти в режим прослушивания для ожидания входящего соединения со стороны клиента, см. схему на рис. 1.147.

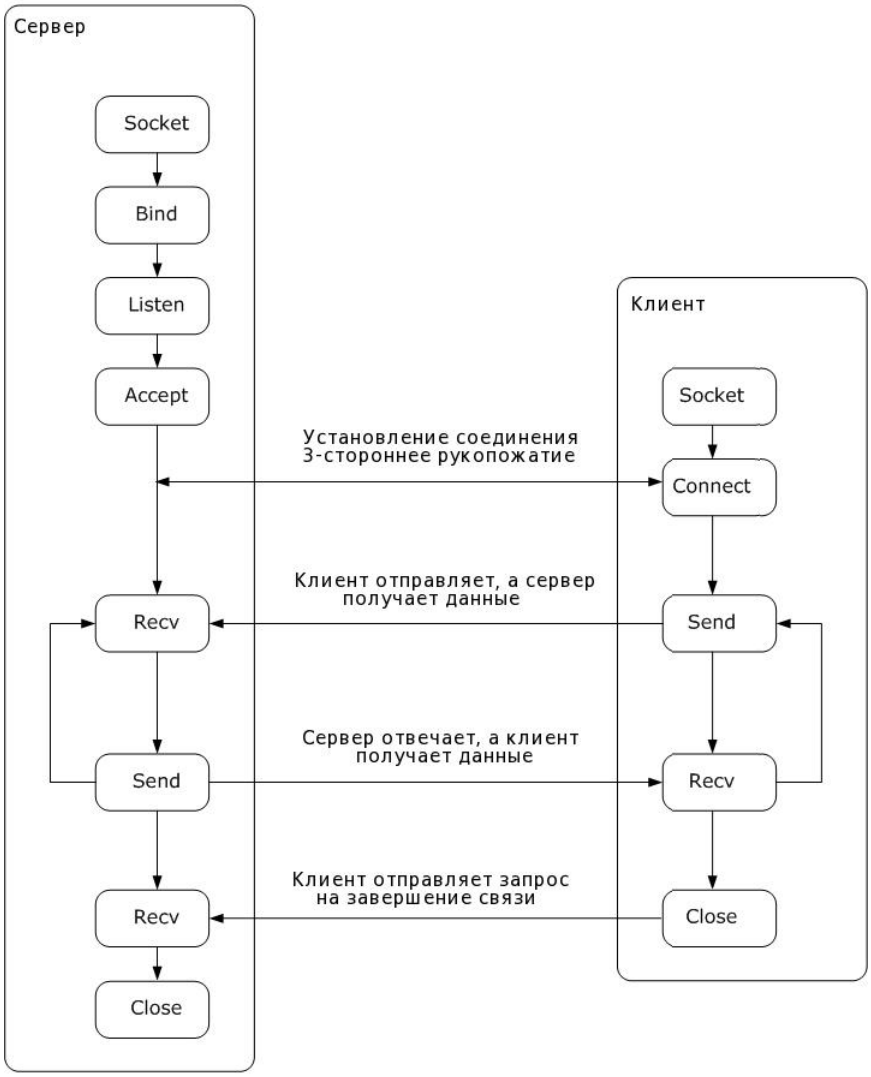


рис. 1.147 – Схема установления связи и передачи данных между клиентом и сервером по сокету

Технология получила название «Сокеты Беркли» (*англ. Berkley Sockets API*) и вошла в стандарт POSIX.1, поэтому поддерживается сейчас всеми современными nix-подобными системами, и ее реализация есть даже в Windows (*см. winsock*).

Изначально сокеты задумывались как программный интерфейс для доступа к транспортному уровню TCP/IP, но несмотря на доминирующее положение этого стека, разработчики реализовали интерфейс по модульному принципу, что позволило обеспечить поддержку и альтернативных семейств протоколов, например, Appletalk и X25. Такое универсальное исполнение позволило реализовать в том числе поддержку «локального семейства сокетов», с помощью которого удалось избежать ненужных накладных расходов при сетевом взаимодействии приложений, установленных в рамках одной операционной системы.

Каждое семейство протоколов использует свое собственное «семейство адресов сокетов» (*англ. socket address families*), которое называется также «доменом» (только не путайте эти «домены ад-

ресов сокетов» с DNS-доменами или Kerberos-доменами). Семейство адресов обычного IPv4-сокета называется AF_INET, и адресом сокета является связка из IP-адреса, типа транспортного протокола и номера порта. Семейство локальных сокетов называется AF_LOCAL (синоним AF_UNIX), и адресом сокета является полное имя файла специального типа, например, уже известный нам `/run/cups/cups.sock`.

Снижение накладных расходов при использовании локальных сокетов достигается за счет того, что приложения общаются друг с другом напрямую в обход стека протоколов TCP/IP, то есть данные не разбиваются на TCP-сегменты с их последующей упаковкой в IP-пакеты и Ethernet-кадры. Файл сокета в этом случае является только конечной точкой для подключения, а обмен данными идет без обращения к диску, так как системные вызовы работают только с буферами ядра.

Сейчас в Linux доступно три типа unix-сокетов, которые отличаются контролем потока данных:

1. **SOCK_STREAM** — устанавливает соединение и обеспечивает возможность надежного двустороннего обмена потоками данных (`/run/systemd/journal/stdout`, `/run/dbus/system_bus_socket`). В случае домена AF_INET этот тип используется для TCP.
2. **SOCK_DGRAM** — обеспечивает передачу блоков информации фиксированного размера (датаграмм) без предварительного установления соединения (`/run/systemd/journal/dev-log`, `/run/systemd/notify`). В случае домена AF_INET этот тип используется для UDP, но применительно к unix-сокетам передача датаграмм работает надежно, и их порядок не нарушается.
3. **SOCK_SEQPACKET** — сочетает в себе свойства двух предыдущих типов, т.к. является потоко-ориентированным как SOCK_STREAM, но поддерживает установку границ записей, как SOCK_DGRAM. Доменом AF_INET не используется.

Приложения, установленные на один хост, взаимодействуют по unix-сокетам на десятки процентов быстрее, чем по TCP-сокетам с использованием IP-адресов локальной петли, поэтому, к примеру, все популярные SQL и NoSQL базы данных реализуют и такой способ взаимодействия тоже.

При необходимости отладки таких приложений для перехвата трафика вы можете временно переключить их с использования unix-сокета на обычный TCP/IP или воспользоваться приложениями двух типов:

- Утилита `socat` — позволяет заменить сокет на свой и сменить транспорт данных на домен AF_INET, где данные можно успешно сохранить (аналог MITM).
- Утилита `sockdump` — использует механизм ядра BPF (Berkeley Packet Filters) и позволяет «слушать» сокет и сохранять данные по аналогии с сетевыми сокетами, см. https://github.com/mechpen/sockdump/tree/master.

Для примера работы с утилитой `socat` установим ее и напишем bash-скрипт:

```
$ sudo apt install socat tshark
$ nano ./dump-log_socat.sh
```

Вставим содержимое в скрипт dump-log_socat.sh:

```
#!/bin/bash

# Параметры
socket="/run/systemd/journal/dev-log"
dump="/tmp/dump_dev-log.pcap"

# Извлечение повторения
port=9876
source_socket="$(dirname "${socket}")/$(basename "${socket}").orig"

# Перемещение файлов сокета
mv "${socket}" "${source_socket}"
trap "{ rm '${socket}'; mv '${source_socket}' '${socket}'; }" EXIT

# Настройте канал через TCP, к которому мы можем подключиться
socat -t100 "TCP-LISTEN:${port},reuseaddr,fork" "UNIX-CONNECT:${source_socket}" &
socat -t100 "UNIX-LISTEN:${socket},mode=777,reuseaddr,fork" "TCP:localhost:${port}"␣
↪&

# Регистрируем трафик
tshark -i lo -w "${dump}" -F pcapng "dst port ${port} or src port ${port}"
```

Теперь осталось запустить его с правами sudo и записать данные с сокета, а для прерывания записи нажмите Ctrl+C:

```
$ sudo bash ./dump-log_socat.sh
...
^C
$ sudo wireshark /tmp/dump_dev-log.pcap
```

Применение sockdump (нужен установленный bcc):

```
$ sudo apt install bison clang-format cmake dpkg-dev \
ethtool flex libbpf-dev libclang-cpp-dev libclang-dev \
libedit-dev libelf-dev libfl-dev libluajit-5.1-dev \
libzip-dev llvm-dev luajit python3-pyroute2
$ mkdir ~/sources; cd ~/sources
$ git clone https://github.com/iovisor/bcc; mkdir ./bcc/build; cd ./bcc/build
$ cmake .. && make -j "$(nproc)"
$ make install
$ cd ~/sources/; git clone https://github.com/mechpen/sockdump.git
$ cd sockdump
$ sudo ./sockdump.py /run/systemd/journal/dev-log --format pcap --output dump
...
^C
$ wireshark -X lua_script:wireshark/dummy.lua dump
```

При кажущейся сложности использование sockdump имеет несколько преимуществ: нет необходимости перемещать сокет (что может вызвать ошибки в других компонентах), имеется lua-скрипт,

который позволяет декодировать бинарные данные в формат Wireshark.

20.4.5.1 Диагностика работы сети

Утилита ip

Команда ip neigh show выведет содержимое ARP-кэша:

```
localadmin@astra:~$ ip neigh show
10.0.2.2 dev eth0 lladdr 52:54:00:12:35:02 REACHABLE
```

Команда ip route show отобразит таблицу маршрутизации:

```
localadmin@astra:~$ ip route show
10.0.2.0/24 dev eth0 proto kernel scope link src 10.0.2.15
```

Для очистки ARP-кэша можно применить команду sudo ip neigh flush all:

```
localadmin@astra:~$ sudo ip neigh flush all
localadmin@astra:~$ ip neigh show
```

Утилита ping

Для проверки доступности узлов и времени прохождения пакетов можно воспользоваться командой:

```
localadmin@astra:~$ ping yandex.ru
PING yandex.ru (77.88.55.60) 56(84) bytes of data.
64 bytes from yandex.ru (77.88.55.60): icmp_seq=1 ttl=245 time=23.7 ms
64 bytes from yandex.ru (77.88.55.60): icmp_seq=2 ttl=245 time=25.5 ms
```

Если вы захотите повторить поведение одноименной утилиты из Windows, то укажите количество пакетов ping -c 4 yandex.ru.

Проверим ARP-кэш и убедимся, что в нем появилась запись:

```
localadmin@astra:~$ ip neigh show
10.0.2.2 dev eth0 lladdr 52:54:00:12:35:02 STALE
```

Утилита traceroute

Команда traceroute применяется для проверки прохождения пакетов до узла назначения и до каждого из промежуточных узлов (маршрутизаторов), например:

```
localadmin@astra:~$ sudo traceroute yandex.ru -I
traceroute to yandex.ru (5.255.255.70), 30 hops max, 60 byte packets
```

(continues on next page)

(продолжение с предыдущей страницы)

```
1 10.0.2.2 (10.0.2.2) 0.241 ms 0.210 ms 0.175 ms
2 192.168.1.1 (192.168.1.1) 4.690 ms 4.652 ms 4.825 ms
...
6 pe03.KK12.Moscow.gldn.net (79.104.235.215) 17.615 ms 16.136 ms 16.276 ms
7 yandex-gw.moscow.gldn.net (195.239.153.234) 16.265 ms 16.291 ms 16.232 ms
8 * * *
9 yandex.ru (5.255.255.70) 26.937 ms 28.007 ms 27.981 ms
```

Примечание: Поведение tracert в Windows и traceroute в Linux отличается. По умолчанию traceroute использует протокол UDP, а не ICMP как tracert. Для использования протокола ICMP необходимо использовать traceroute с опцией -I (i заглавная), но в этом случае утилита требует прав суперпользователя.

Утилита netstat

Утилита netstat выводит список сетевых соединений, таблицы маршрутизации, статистику интерфейсов, NAT-соединения и членство в мультикаст-группах. Мы не будем подробно на ней останавливаться, так как рекомендуем использовать более современную утилиту `ss`, о которой будет рассказано ниже.

```
localadmin@astra:~$ netstat | head -10
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags       Type       State         I-Node   Path
unix  2      [ ]         DGRAM                    27580    /run/user/107/systemd/
↪notify
unix  2      [ ]         DGRAM                    28371    /run/user/1000/systemd/
↪notify
unix  3      [ ]         DGRAM      CONNECTED     18032    /run/systemd/notify
unix  7      [ ]         DGRAM      CONNECTED     18044    /run/systemd/journal/
↪socket
unix  15     [ ]         DGRAM      CONNECTED     18054    /run/systemd/journal/dev-
↪log
unix  2      [ ]         DGRAM      CONNECTED     27519
```

Утилита ss

Утилита `ss` (*англ. socket stat*) – это инструмент командной строки, который отображает статистику сокетов системы Linux. Утилита может отображать статистику для сокетов TCP, UDP, DCCP, RAW, домена Unix и многого другого. Утилита позволяет отображать информацию, аналогичную netstat, при этом отображает больше, чем другие инструменты.

Опции утилиты:

- Команда `ss -t` — показывает активные TCP-соединения.
- Команда `ss -u` — показывает активные UDP-соединения.
- Команда `ss -l` — показывает только сокеты, ожидающие подключения.
- Команда `ss -p` — PID и имя программы, которой принадлежит каждый из сокетов.
- Команда `ss -n` — показывает числовые адреса, не пытаясь определить символьное имя узла, порт или имя пользователя.
- Команда `ss -tlp` — показывает TCP-соединения и их ассоциированные процессы.
- Команда `ss -ulp` — показывает UDP-соединения их ассоциированные процессы.
- Команда `ss -tulpn` — показывает TCP и UDP-соединения и их ассоциированные процессы.
- Команда `ss -s` — показывает общую статистику по типам сокетов.
- Команда `ss -a` — показывает сокеты, ожидающие подключения и не ожидающие подключения.
- Команда `ss -x` или `ss -f unix` — показывает unix-сокеты.

Показывает активные TCP и UDP соединения и их ассоциированные процессы:

```
localadmin@astra:~$ ss -tlp
State      Recv-Q     Send-Q   Local Address:Port     Peer Address:Port   Process
LISTEN     0          128            0.0.0.0:sunrpc         0.0.0.0:*
LISTEN     0          128          127.0.0.1:ipp            0.0.0.0:*
LISTEN     0          128               [::]:sunrpc            [::]:*
LISTEN     0          128              [::1]:ipp               [::]:*
```

Показывает TCP и UDP-соединения и их ассоциированные процессы:

```
localadmin@astra:~$ ss -tulp
Netid    State     Recv-Q    Send-Q  Local Address:Port    Peer Address:Port  Process
udp      UNCONN    0         0             0.0.0.0:56863        0.0.0.0:*
udp      UNCONN    0         0             0.0.0.0:sunrpc       0.0.0.0:*
udp      UNCONN    0         0             0.0.0.0:mdns         0.0.0.0:*
udp      UNCONN    0         0                [::]:33353           [::]:*
udp      UNCONN    0         0                [::]:sunrpc          [::]:*
udp      UNCONN    0         0                   *:xdmcp              *:*
udp      UNCONN    0         0                [::]:mdns            [::]:*
tcp      LISTEN    0         128           0.0.0.0:sunrpc       0.0.0.0:*
tcp      LISTEN    0         128         127.0.0.1:ipp          0.0.0.0:*
tcp      LISTEN    0         128              [::]:sunrpc          [::]:*
tcp      LISTEN    0         128             [::1]:ipp             [::]:*
```

Показывает TCP и UDP-соединения и их ассоциированные процессы, не пытаясь определить символьное имя узла, порт или имя пользователя:

```
localadmin@astra:~$ ss -tulpn
Netid    State    Recv-Q   Send-Q          Local Address:Port        Peer Address:Port␣
↪ Process
udp      UNCONN   0        0                     0.0.0.0:56863            0.0.0.0:*
udp      UNCONN   0        0                     0.0.0.0:111              0.0.0.0:*
udp      UNCONN   0        0                     0.0.0.0:5353             0.0.0.0:*
udp      UNCONN   0        0                        [::]:33353               [::]:*
udp      UNCONN   0        0                        [::]:111                 [::]:*
udp      UNCONN   0        0                           *:177                    *:*
udp      UNCONN   0        0                        [::]:5353                [::]:*
tcp      LISTEN   0        128                   0.0.0.0:111              0.0.0.0:*
tcp      LISTEN   0        128                 127.0.0.1:631              0.0.0.0:*
tcp      LISTEN   0        128                      [::]:111                 [::]:*
tcp      LISTEN   0        128                     [::1]:631                 [::]:*
```

Показывает общую статистику по типам сокетов:

```
localadmin@astra:~$ ss -s
Total: 420
TCP: 6 (estab 0, closed 2, orphaned 0, timewait 0)

Transport Total   IP        IPv6
RAW       0       0         0
UDP       7       3         4
TCP       4       2         2
INET      11      5         6
FRAG      0       0         0
```

20.5 Практика и тестирование

20.5.1 Практическая работа: Модуль 20. Сеть и стек TCP/IP, управление сетевыми интерфейсами

См. также *Требования, правила и цели выполнения практической работы*

20.5.1.1 Практические задания

Задание 1.

Просмотр сетевых интерфейсов в терминале:

- Выведите список всех сетевых интерфейсов на компьютере и их текущее состояние (активный/неактивный).
- Выведите только IPv4 адрес каждого интерфейса.
- Выведите таблицу ARP (Address Resolution Protocol) для конкретного сетевого интерфейса eth0.

- Выполните трассировку маршрута к ya.ru, используя ICMP-пакеты, с ограничением количества запросов до 1.
- Выведите с использованием команды netstat список всех открытых прослушивающих (listening) портов (т.е. портов, на которых приложения ожидают входящие соединения).

Задание 2.

Проверка DNS:

- Просмотрите текущие DNS-настройки, используя утилиту nmtui.
- Добавьте дополнительный DNS-сервер 1.1.1.1 и сделайте его первым в списке.
- Проверьте корректность работы новых DNS-настроек путем разрешения имен хостов в IP-адреса.

Задание 3.

Рассчитайте и получите последний IP-адрес хоста в сети 192.168.0.128/28.

Задание 4.

Настройка нового сетевого интерфейса в графической утилите NetworkManager:

- Создайте новый сетевой интерфейс.
- Сконфигурируйте IP-адрес и маску подсети на полученный в шаге 2.
- Убедитесь, что новые настройки применены успешно, используя команду nmcli.

20.5.1.2 Ответы на практические задания (пошаговые инструкции)

Задание 1.

Просмотр сетевых интерфейсов в терминале:

Выведите список всех сетевых интерфейсов на компьютере и их текущее состояние (активный/неактивный):

```
localadmin@astra:~$ ip link show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group
default qlen 1000
```

(continues on next page)

(продолжение с предыдущей страницы)

```
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00

2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP mode
DEFAULT group default qlen 1000

   link/ether 0e:9e:c9:92:f7:e0 brd ff:ff:ff:ff:ff:ff
```

Более читабельный вариант:

```
localadmin@astra:~$ ip -br link show
lo               UNKNOWN        00:00:00:00:00:00 <LOOPBACK,UP,LOWER_UP>
eth0             UP             0e:9e:c9:92:f7:e0 <BROADCAST,MULTICAST,UP,LOWER_UP>
```

Выведите только IPv4 адрес каждого интерфейса:

```
localadmin@astra:~$ ip -4 addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen
1000
   inet 127.0.0.1/8 scope host lo
      valid_lft forever preferred_lft forever

2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group
default qlen 1000
   inet 192.168.1.229/24 brd 192.168.1.255 scope global eth0

      valid_lft forever preferred_lft forever
```

Более читабельный вариант:

```
localadmin@astra:~$ ip -br -4 addr show

lo               UNKNOWN        127.0.0.1/8

eth0             UP             192.168.1.229/24
```

Выведите таблицу ARP (Address Resolution Protocol) для конкретного сетевого интерфейса eth0:

```
localadmin@astra:~$ sudo arp -i eth0
Address                  HWtype  HWaddress           Flags Mask     Iface
192.168.1.144            ether   6c:bf:b5:02:5a:c7   C              eth0
192.168.1.53             ether   28:16:ad:1e:52:2b   C              eth0
192.168.1.1              ether   50:ff:20:a1:4c:aa   C              eth0
```

Выполните трассировку маршрута к ya.ru, используя ICMP-пакеты, с ограничением количества запросов до 1:

```
localadmin@astra:~$ sudo traceroute ya.ru -q 1 -I
traceroute to ya.ru (5.255.255.242), 30 hops max, 60 byte packets
1 192.168.1.1 (192.168.1.1) 0.458 ms
2 100.76.128.1 (100.76.128.1) 3.804 ms
3 188.128.125.18 (188.128.125.18) 1.901 ms
4 95.167.92.165 (95.167.92.165) 19.464 ms
```

(continues on next page)

(продолжение с предыдущей страницы)

```
5 *
6 188.254.94.106 (188.254.94.106) 25.056 ms
7 *
8 ya.ru (5.255.255.242) 26.614 ms
```

Выведите с использованием команды netstat список всех открытых прослушивающих (listening) портов (т.е. портов, на которых приложения ожидают входящие соединения):

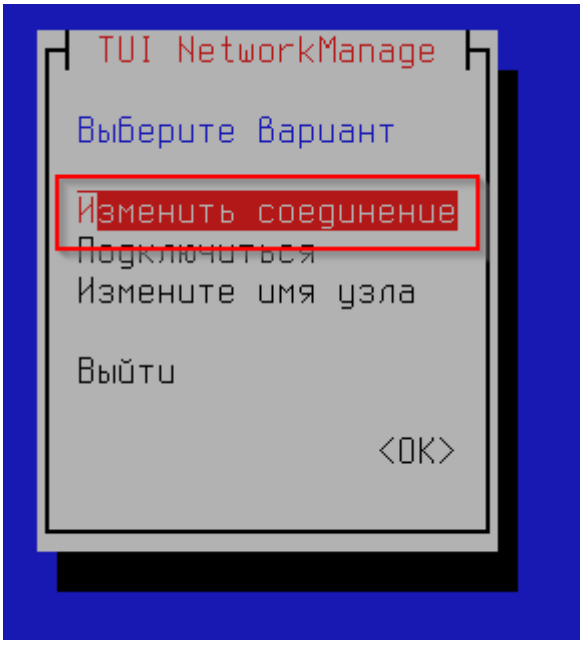
```
localadmin@astra:~$ netstat -tuln
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:22              0.0.0.0:\*              LISTEN
tcp6       0      0 :::22                   :::\*                   LISTEN
```

Задание 2.

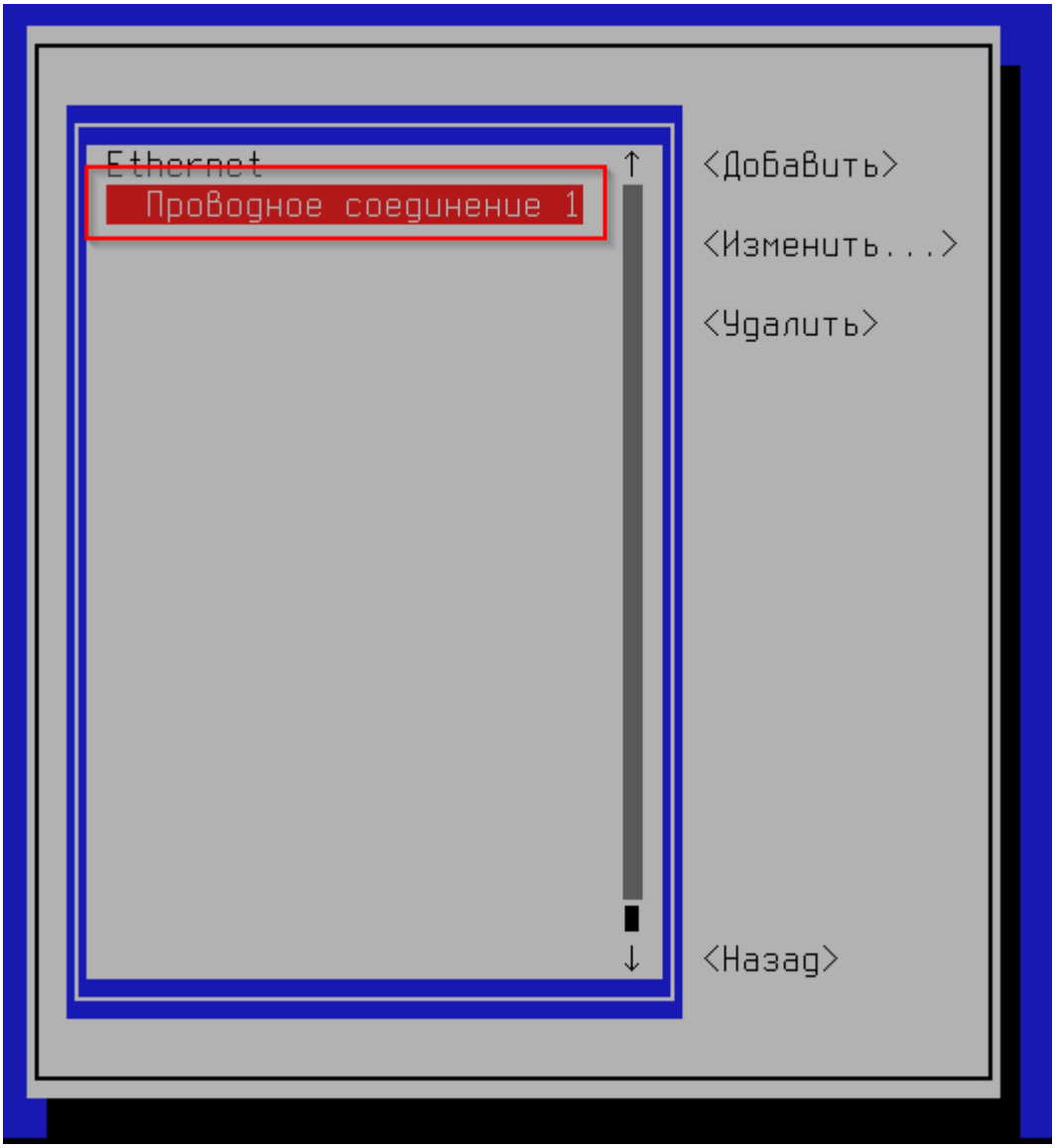
Проверка DNS:

- Просмотреть текущие DNS-настройки используя утилиту `nmtui`:

  1. Выбрать «Изменить соединение».



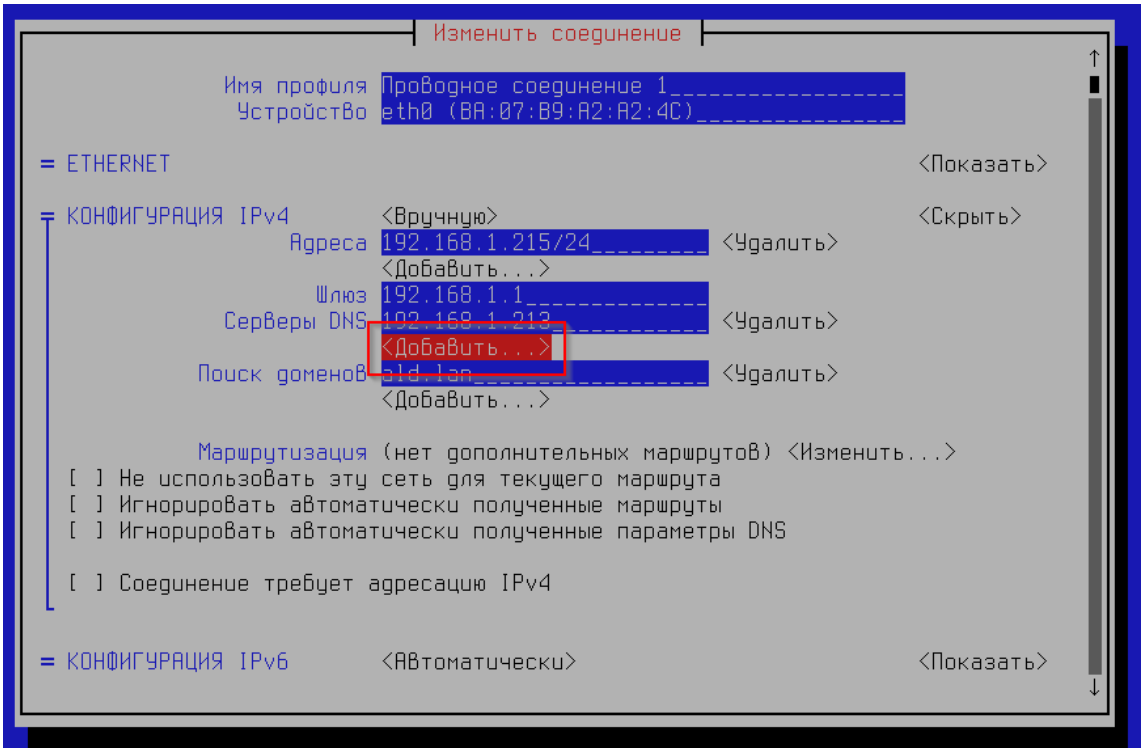
  2. Выбрать доступное сетевое соединение.



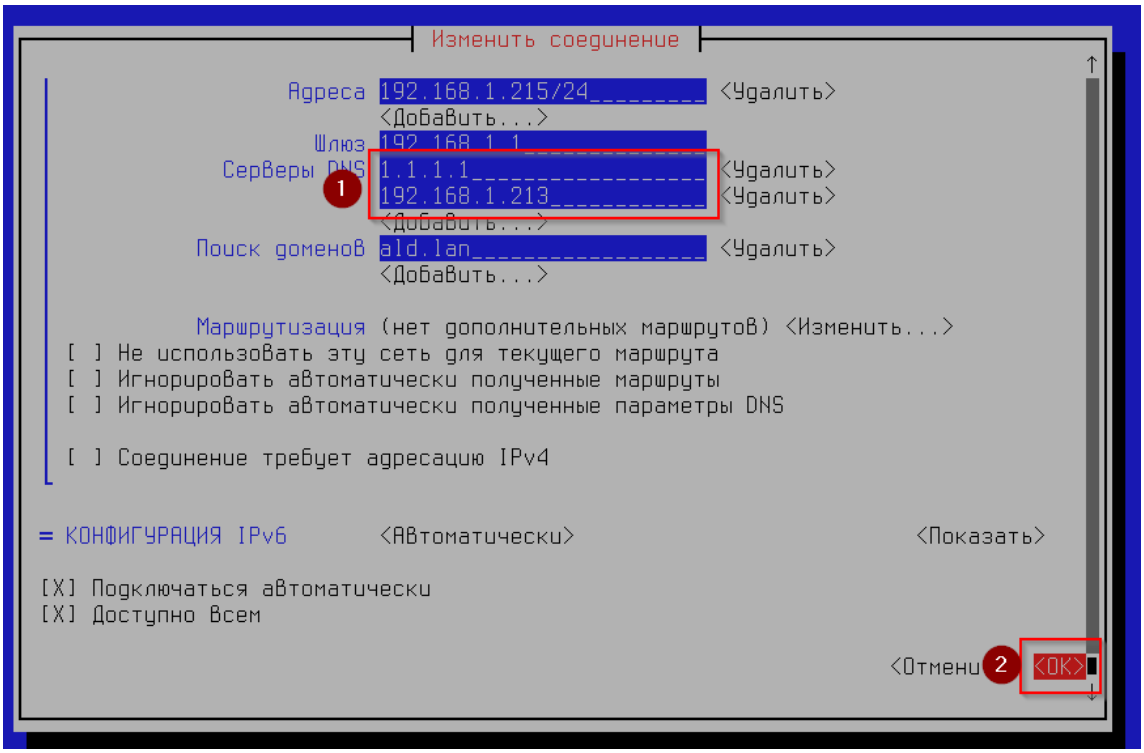
3. В поле «Серверы DNS» будет указаны текущие настройки.

- Добавьте дополнительный DNS сервер 1.1.1.1 и сделайте его первым в списке.

1. В разделе сервера DNS выбрать «Добавить».



2. В новое поле скопируйте значение первого DNS сервера, а в первое поле вписать 1.1.1.1 и нажать OK.



- Проверьте корректность работы новых DNS-настроек путем разрешения имен хостов в IP-адреса.

Откройте терминал и запустите команду nslookup или dig (если данные команды отсутствуют,

то их необходимо установить используя sudo apt-get install dnsutils).

```
localadmin@astra:~$ nslookup ya.ru
Server:         1.1.1.1
Address:        1.1.1.1#53


Non-authoritative answer:
Name:   ya.ru
Address: 77.88.55.242
Name:   ya.ru
Address: 5.255.255.242
Name:   ya.ru
Address: 2a02:6b8::2:242
```

Убедитесь, что в поле Server и Address используется ранее указанный DNS сервер 1.1.1.1

Задание 3

Рассчитайте и получите последний IP-адрес хоста в сети 192.168.0.128/28.

Для расчета последнего IP-адреса в сети 192.168.0.128/28 нужно выполнить следующие шаги

1. Найдите количество хостов в подсети.

   Для этого используется формула 2^(32-маска_подсети) - 2, где маска подсети равна 28. В данном случае это 2^(32-28) - 2 = 16 - 2 = 14 хостов.

2. Найдите последний IP-адрес хоста в подсети. Это будет адрес сети плюс количество хостов.

   - Адрес сети: 192.168.0.128
   - Количество хостов: 14
   - Последний IP-адрес: 192.168.0.128 + 14 = 192.168.0.142

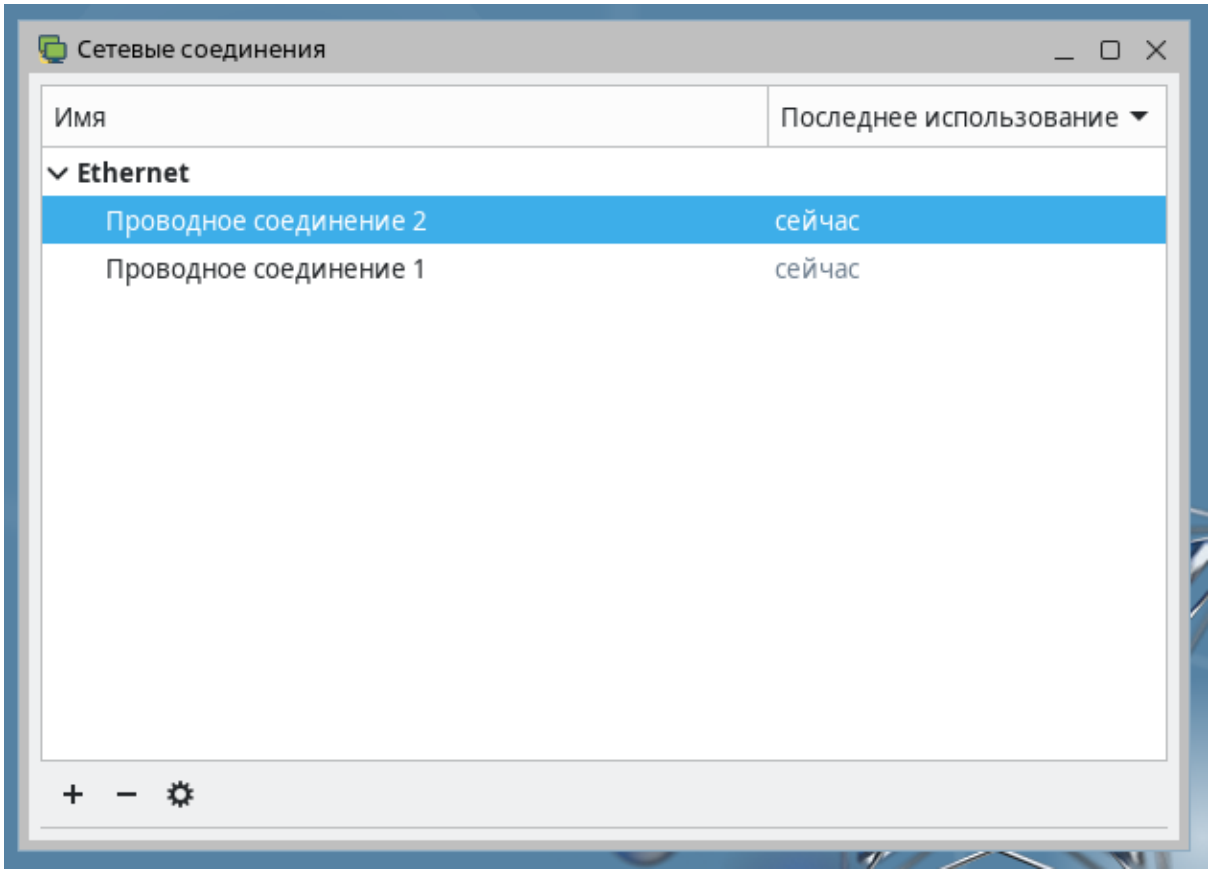
Таким образом, последний IP-адрес в сети 192.168.0.128/28 равен **192.168.0.142**

Задание 4.

Настройте новый сетевой интерфейс в графической утилите NetworkManager.

- Создать новый сетевой интерфейс.

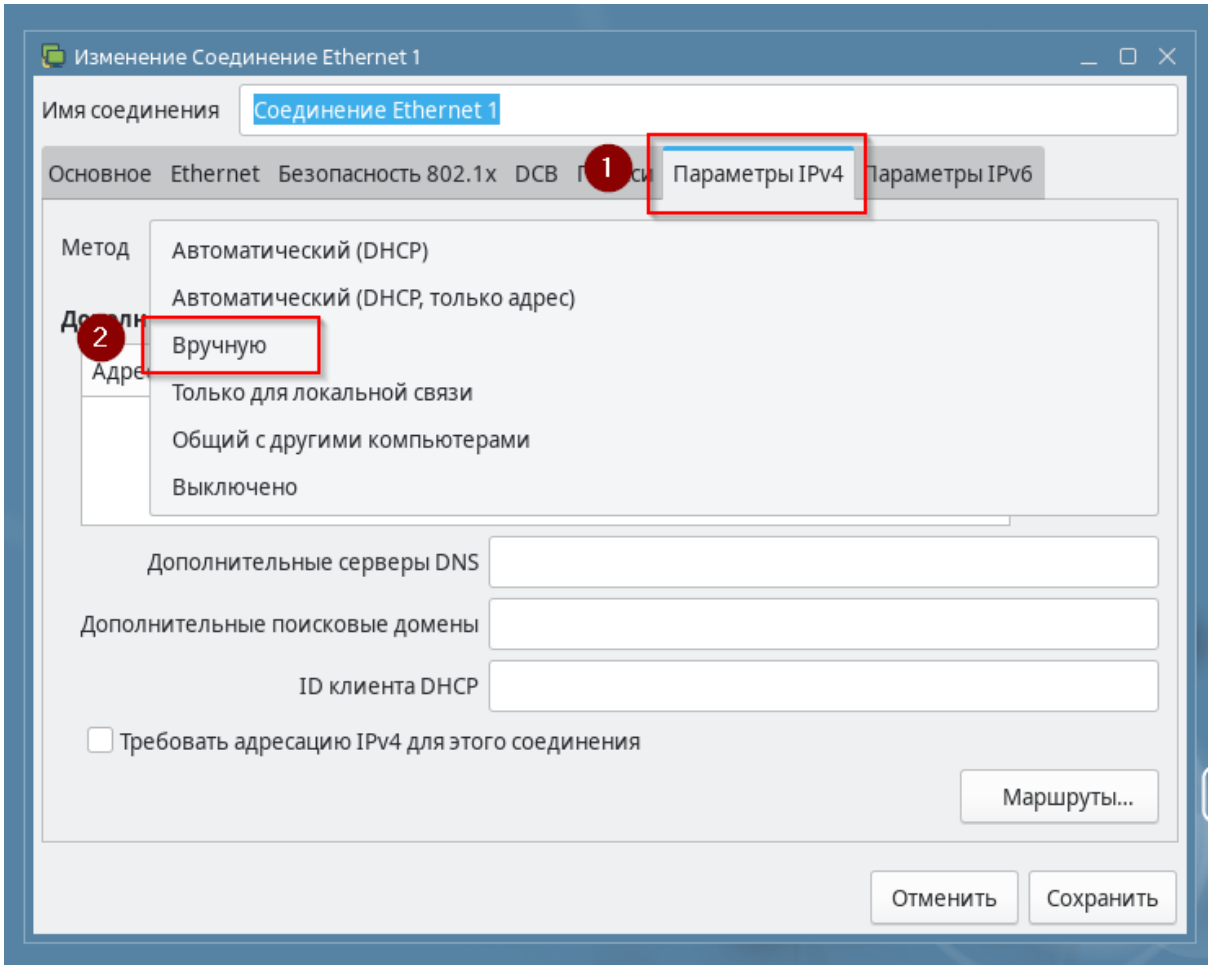
1. Добавьте новую сетевую карту (в VirtualBox, Proxmox и т.д.). В результате должно быть 2 сетевых соединения.



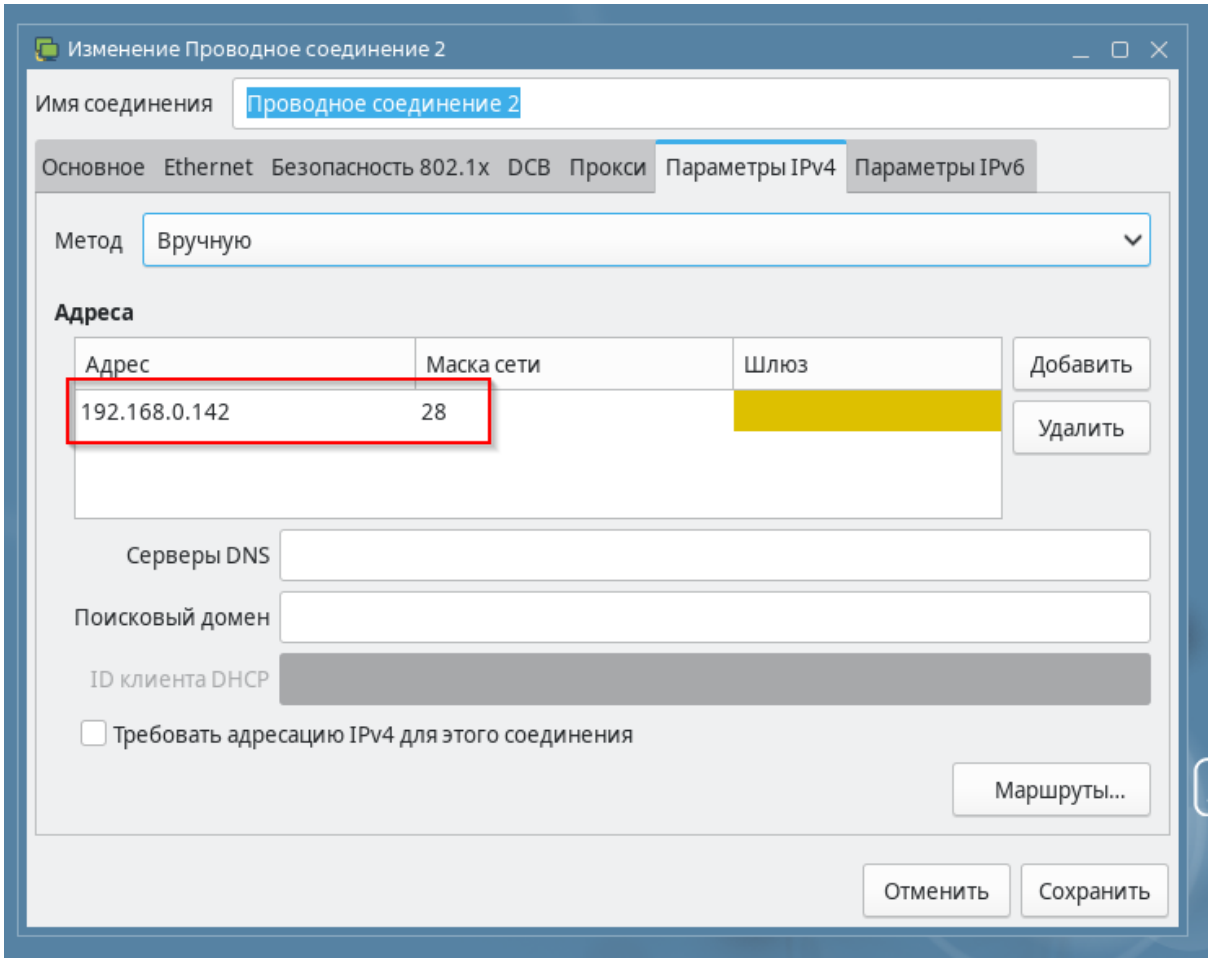
2. Откройте новое сетевое соединение

- Сконфигурируйте IP-адрес и маску подсети на полученный в шаге 2.

1. Перейдите на вкладку «Параметры IPv4», далее выберите метод конфигурации «Вручную».



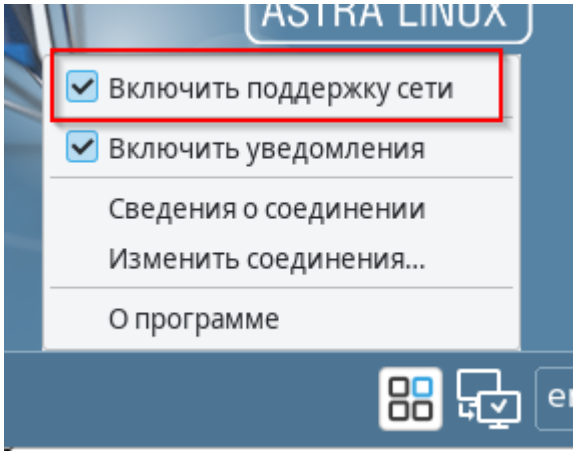
2. Пропишите IP-адрес и маску подсети (можно в любом формате)



3. Сохранить изменения

- Убедитесь, что новые настройки применены успешно, используя команду nmcli.

  Если внесенные изменения не применились, то выключите и включите заново «поддержку сети», либо используйте соответствующую команду в nmcli.




```
localadmin@astra:~$ nmcli
th0: подключено to Проводное соединение 1
      "Intel 82540EM"
```

(continues on next page)

(продолжение с предыдущей страницы)

```
      ethernet (e1000), BA:07:B9:A2:A2:4C, HW, MTU 1500
      ip4 default
      inet4 192.168.1.215/24
      route4 192.168.1.0/24
      route4 0.0.0.0/0
      inet6 fe80::6c95:4b18:1299:5e52/64
      route6 fe80::/64

eth1: подключено to Проводное соединение 2
      "Intel 82540EM"
      ethernet (e1000), DE:5F:5F:3F:4E:3E, HW, MTU 1500
      inet4 192.168.0.142/28
      route4 192.168.0.128/28
      inet6 fe80::a2c7:d881:d11d:429f/64
      route6 fe80::/64

lo: без управления
      "lo"
      loopback (unknown), 00:00:00:00:00:00, Ю3, MTU 65536

DNS configuration:
      servers: 1.1.1.1 192.168.1.213
      domains: ald.lan
      interface: eth0
```

20.5.2 Тест: Модуль 20. Сеть и стек TCP/IP, управление сетевыми интерфейсами

Пожалуйста, ответьте на несколько вопросов, чтобы закрепить полученные знания. Правильные ответы для самопроверки вы сможете найти в конце списка. Обратите внимание, что у некоторых вопросов может быть несколько верных вариантов ответов.

20.5.2.1 Вопросы

Вопрос 1. Какой протокол обеспечивает надежную передачу данных с установлением сессии?

a. TCP

b. UDP

c. ICMP

Вопрос 2. Какой протокол обеспечивает передачу данных без гарантии доставки и без установления соединения?

a. TCP

b. UDP

c. IP

d. FTP

Вопрос 3. Какой префикс соответствует маске подсети 255.255.255.0?

a. /8

b. /16

c. /24

d. /32

Вопрос 4. Какой файл содержит информацию о сетевой конфигурации интерфейсов в Astra Linux?

a. Файл `/etc/network/interfaces`

b. Файл `/etc/netconfig`

c. Файл `/etc/ethernet.conf`

d. Файл `/etc/network.conf`

Вопрос 5. Какой файл используется для настройки DNS-серверов в Linux?

a. Файл `/etc/resolve.conf`

b. Файл `/etc/dns.conf`

c. Файл `/etc/nameservers`

d. Файл `/etc/resolv.conf`

Вопрос 6. Какой сервис в Linux отвечает за разрешение DNS-запросов?

a. dnsd

b. bind

c. nslookup

d. systemd-resolved

Вопрос 7. Какой протокол работает на прикладном уровне модели OSI?

a. TCP

b. UDP

c. HTTP

d. ICMP

Вопрос 8. Сколько хостов может работать в сети с маской /27?

a. 30

b. 62

c. 126

d. 254

Вопрос 9. Какой протокол работает на транспортном уровне модели OSI?

a. HTTP

b. TCP

c. IP

d. DNS

Вопрос 10. Какой слой модели OSI отвечает за кадрирование данных?

a. Прикладной

b. Физический

c. Канальный

d. Сетевой

Ответы на вопросы

1. a
2. b
3. c
4. a
5. d
6. d
7. c

8. a
9. b
10. c

20.6 Заключение

В этом модуле мы разобрали, как работает сетевой стек TCP/IP, и научились его настраивать в Astra Linux с помощью нескольких инструментов. Как мы уже отмечали, навыки настройки сети крайне важны в домене, поэтому настоятельно рекомендуем несколько раз вернуться к материалам этого модуля и обязательно выполнить предлагаемые практические задания.

В следующем модуле мы познакомимся с системами журналирования в Astra Linux, затронем вопросы анализа производительности системы и сбора отладочной информации. Разберем виды нештатных ситуаций, которые могут возникнуть на ранних и поздних этапах загрузки ОС и способы их устранения.

20.7 Дополнительные источники информации

- Официальное руководство Cisco по подготовке к сертификационным экзаменам 100-101 или 100-105, Уэнделл Одом (ISBN 978-5-8459-1807-9)
- Графический симулятор сети GNS3

20.8 Обратная связь

Если остались вопросы, то их всегда можно задать в специальной теме.

Модуль 21. Работа с журналами и отладка системы

21.1 Введение

Говорят, что практика — это когда все работает, но непонятно как, а теория — это когда все понятно, но ничего не работает. Однако, когда теория совмещается с практикой, при этом ничего не работает и ничего не понятно, наступает время отладки.

Из этого модуля вы узнаете, как работает система журналирования в Linux и какие журналы можно использовать для быстрой отладки приложений. Мы обсудим эффективные подходы к выполнению отладки и детально разберем несколько часто встречающихся нештатных ситуаций.

21.2 Работа с журналами и сбор логов

На компьютерах Linux мы довольно часто используем приложения с открытыми исходными кодами, поэтому при выявлении проблемы, никто не мешает скомпилировать и запустить приложение в режиме отладки, но это все-таки далеко не самый простой способ.

Обычно системные администраторы ограничиваются внимательным изучением сообщений об ошибках, которые появляются в журналах в момент воспроизведения проблемы, поэтому нужно хорошо понимать, как работает эта система.

21.2.1 Система журналирования syslog

Разработчики каждого приложения вправе сами выбирать, какие именно события они будут логировать и куда будут их записывать, но все вендоры стараются придерживаться общих стандартов, т.к. понимают, что чем легче будет поддерживать их приложение, тем выше шансы, что их продукт окажется востребован на рынке. Поэтому среди разработчиков Linux признаком хорошего тона является запись событий в подсистему журналирования syslog.

Термином syslog (*англ. system log* — системный журнал) называют как саму службу централизованного ведения журналов событий, так и протокол, по которому она получает события от приложений. В операционной системе Windows аналогом syslog является подсистема, которая так и называется «Журнал событий» (*англ. Event Log*).

Сообщения syslog представляют собой обычные текстовые строки в формате:

```
<priority>timestamp hostname: message
```

Где:

- **priority** – определяет важность сообщения,
- **timestamp** – дату и время генерации события в системе
- **hostname** – позволяет организовать централизованное хранение событий со всех компьютеров в сети.

Приоритет включает в себя значение категории субъекта (facilities, см. табл. 1.8) и уровень важно-

сти события (severity, см. табл. 1.9) по формуле:

$$Priority = (facilities \cdot 8) + severity$$

Например, если нам потребуется зафиксировать событие Security (13) с уровнем Error (3), то нужно будет использовать приоритет:

$$Priority = (13 \cdot 8) + 3 = 107$$

В зависимости от значения приоритета и настроек syslog система может поместить событие в тот или иной файл, или вообще отбросить, что определяется фильтрами источников.

табл. 1.8 – Коды категорий субъектов (facilities) в протоколе syslog.

#	Категория	Описание
0	kern	ядро операционной системы
1	user	программное обеспечение пользователя
2	mail	почтовая система
3	daemon	системные службы (daemons)
4	auth	сообщения безопасности (авторизации)
5	syslog	собственные сообщения syslogd
6	lpr	подсистема печати
7	news	подсистема новостных групп (телеконференций, NNTP)
8	uucp	подсистема UUCP (Unix-to-Unix Copy Protocol)
9	cron	службы времени
10	authpriv	сообщения безопасности (авторизации)
11	ftp	служба FTP
12	ntp	подсистема NTP
13	security	журнал аудита (упразнен, используйте **auth**)
14	console	аварийный журнал
15	solaris-cron	службы времени
16	local0	локальный 0
17	local1	локальный 1
18	local2	локальный 2
19	local3	локальный 3
20	local4	локальный 4
21	local5	локальный 6
22	local6	локальный 6
23	local7	локальный 7

табл. 1.9 – Уровни важности (severity) в протоколе syslog.

#	Буквенный код*	Описание значения
0	**emerg**	Emergency – система не работоспособна
1	**alert**	Alert – система требует немедленного вмешательства
2	**crit**	Critical – состояние системы критическое
3	**err**	Error – сообщения об ошибках
4	**warning**	Warning – предупреждения о возможных проблемах
5	**notice**	Notice – сообщения о нормальных, но важных событиях
6	**info**	Informational – информационные сообщения
7	**debug**	Debug – отладочные сообщения

Примечание: Буквенный код применяется в фильтрах **syslog-ng** или утилите `logger`, подробнее см. справку `man logger`

Местом хранения системных журналов по стандарту FHS является каталог `/var/log`. Вот несколько наиболее важных из них:

- Файл `/var/log/syslog` — основной журнал, в котором хранятся все сообщения, кроме сообщений об аутентификации и сообщений почтовой службы.
- Файл `/var/log/messages` — сокращенная версия файла syslog. Сюда записываются несколько типов сообщений: почта, cron, сервисы systemd, сообщения от ядра, pam, аутентификация и др.
- Файл `/var/log/auth.log` — хранит сообщения об аутентификации служб и пользователей.

Стоит упомянуть также файлы:

- Файл `/var/log/boot.log` — хранит сообщения, сгенерированные во время загрузки ОС.
- Файл `/var/log/kern.log` — хранит сообщения от ядра ОС.
- Файл `/var/log/daemon.log` — хранит сообщения от различных служб (кроме почты, печати и планировщика cron).
- Файл `/var/log/user.log` – хранит сообщения от команд, выполняемых пользователями (NetworkManager, графическими программами и так далее).

Перечисляя файлы журналов, обычно упоминают еще `wtmp` и `lastlog`, но они хранят данные в бинарном формате и не имеют отношения к службе **syslog**:

- Файл `/var/log/lastlog` — информация о последних входах в систему пользователей.
- Файл `/var/log/wtmp` — хранит сообщения о сеансах пользователей, перезагрузках системы.

Для просмотра информации из этих файлов применяется утилита `who`, например, информацию о загрузке системы можно получить командой `who -b`.

Локальные приложения передают события в подсистему **syslog** через unix-сокет `/dev/log`, а для системных сообщений ядра отведены также сокет `/dev/kmsg` и псевдофайл `/proc/kmsg`. Но при написании кода разработчикам даже не требуется задумываться об этих сокетах, так как во всех современных языках программирования высокого уровня есть библиотеки, с помощью которых записать событие в журнал не сложнее, чем вывести строку на экран.

Более того, даже из скриптов bash событие в журнал можно записать одной строкой, например, вот так можно имитировать сообщение об ошибке уровня ядра:

```
localadmin@astra:~$ logger --priority kern.error 'Имитируем панику'
```

Это событие в соответствии с его приоритетом попадет в журнал `/var/log/syslog`.

```
localadmin@astra:~$ sudo tail -f /var/log/syslog
...
Jun 10 07:14:03 astra localadmin[4768]: Имитируем панику
```

Если вернуться к библиотекам логирования, то в них обычно реализуют множество дополнительных обработчиков, с помощью которых сообщения можно одновременно передавать в **syslog**, выводить на экран, записывать в собственный файл и многое другое. Например, служба **SSSD** позволяет сохранять события каждого компонента в отдельном файле.

21.2.2 Система журналирования syslog-ng

В системе Astra Linux Special Edition 1.7 для работы с системными журналами используется не старая служба syslog, а ее усовершенствованная версия syslog-ng (*от англ. Syslog Next Generation*) — это бесплатная реализация протокола syslog с открытым исходным кодом для nix-подобных систем.

Приложение расширяет исходную модель работы службы syslogd возможностями гибкой фильтрации на основе содержимого и добавляет такие важные функции, как возможность использования TCP для транспортировки сообщений. Передавать события на централизованный сервер журналирования можно не только на порт 514/UDP, но еще и 601/TCP, а в случае применения TLS дополнительно 6514/UDP и 6514/TCP.

Основным конфигурационным файлом службы является `/etc/syslog-ng/syslog-ng.conf`, в котором определяются объекты системы.

Приведем часть его содержимого:

```
localadmin@astra:~$ cat /etc/syslog-ng/syslog-ng.conf
@version: 3.19
@include "scl.conf"
...
```

В начале файла обозначается версия syslog-ng, и подключается файл `scl.conf`, который, в свою очередь, подключает все файлы с суффиксом `*.conf` из каталога `/etc/syslog-ng/scl/*`.

Аббревиатура SCL означает Syslog-ng Configuration Library — библиотека конфигураций syslog. Это набор файлов конфигураций, который по задумке разработчиков syslog-ng помогает избежать загромождения основного файла конфигурации и позволяет повторно использовать готовые фрагменты конфигурации на разных узлах.

В Astra Linux по умолчанию не существует каталога `/etc/syslog-ng/scl`, а все дополнительные файлы конфигурации располагаются непосредственно в каталоге `/etc/syslog-ng/conf.d`.

```
localadmin@astra:~$ tail /etc/syslog-ng/scl.conf -n 6
```

```
# This file is placed into /etc/syslog-ng in order to make it trivial to
# include in user written syslog-ng.conf files.  It sets up 'scl-root' and
# `include-path`, then includes all SCL supplied plugins.
#
@include 'scl/*/*.conf'
```

Далее следуют глобальные опции:

```
...
options {
   chain_hostnames(off); flush_lines(0); use_dns(no); use_fqdn(no);
   dns_cache(no); owner("root"); group("adm"); perm(0640);
   stats_freq(0); bad_hostname("^gconfd$");
};
...
```

Далее идет перечисление драйверов источников сообщений:

```
...
source s_src {
   system();
   internal();
};
...
```

Как видим, по умолчанию используются два источника:

- **system()** – отвечает за автоматическое определение платформы и сбор стандартных для этой платформы журналов.
- **internal()** – сообщения, генерируемые самой службой **syslog-ng**.

В операционной системе Astra Linux служба **syslog-ng** определяет, что рядом с ней работает служба **journald**, поэтому источник system() автоматически настраивает **syslog-ng** на чтение файла службы **journald** из каталога `/run/log/journal/`.

А служба **journald**, в свою очередь, получает все сообщения от приложений, т.к. файл `/dev/log` является обычной символической ссылкой на ее unix-сокет `/run/systemd/journal/dev-log`. Взаимодействие компонентов представлено на рис. 1.148.

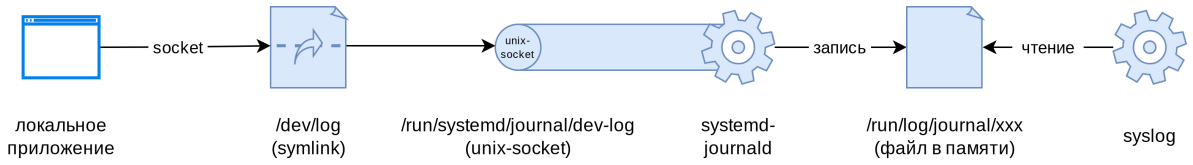


рис. 1.148 – Порядок получения сообщений службой syslog-ng

Далее следует раздел с драйверами приемников сообщений:

```
...
destination d_auth { file("/var/log/auth.log"); };
destination d_cron { file("/var/log/cron.log"); };
destination d_daemon { file("/var/log/daemon.log"); };
...
```

По умолчанию используются приемники двух типов:

- **file()** – запись сообщений в указанный файл;
- **usertty()** – отправка сообщений на терминал указанного пользователя, если сессия пользователя активна.

Следом идут фильтры:

```
...
filter f_dbg { level(debug); };
filter f_info { level(info); };
filter f_notice { level(notice); };
...
```

По умолчанию используются три типа фильтров:

- **level** – фильтрация сообщений по приоритету (severity), см. табл. 1.9;
- **program** – фильтрация сообщений по названию приложения;
- **facility** – фильтрация сообщений по субъекту (facility), см. табл. 1.8.

Затем в конфигурационном файле перечисляются пути логов (*англ. log paths*), которые связывают источники с фильтрами и получателями:

```
...
log { source(s_src); filter(f_auth); destination(d_auth); };
log { source(s_src); filter(f_cron); destination(d_cron); };
log { source(s_src); filter(f_daemon); destination(d_daemon); };
...
```

И уже в конце файла может быть настроена пересылка сообщений по сети:

```
# All messages send to a remote site
#
# log { source(s_src); destination(d_net); };
```

И подключение дополнительных файлов `*.conf` из каталога `/etc/syslog-ng/conf.d/`:

```
...
@include "/etc/syslog-ng/conf.d/*.conf"
...
```

Давайте посмотрим список дополнительных файлов в каталоге `/etc/syslog-ng/conf.d`:

```
localadmin@astra:~$ ls /etc/syslog-ng/conf.d
20-ufw.conf mod-astra-disk-space-running-out.conf  mod-astra-ram-running-out.conf
astraevents.conf mod-astra-previous-login.conf  mod-astra-unsigned-binary-file.conf
mod-astra.conf mod-astra-quota-depleted.conf
```

Как вы могли заметить, объекты задаются в следующем синтаксисе:

```
<Тип объекта> <Идентификатор объекта> {
   <параметр>(<опция> ..., ...) ...; ... };
```

Где:

- **Тип объекта** — драйвер источника и источник, драйвер приемника и приемник, путь, фильтр, анализатор, модификатор, шаблон, глобальная опция и др.
- **Идентификатор объекта** — уникальное имя, определяющее объект. Рекомендуется использовать имена с префиксами, указывающими на тип объекта, например, префикс «s_» для источников, префикс «d_» для приемников и т.д.

  Имена являются регистр-зависимыми. Желательно избегать имен, которые являются зарезервированными словами, но, если очень хочется, заключите это имя в кавычки. Повторное определение объектов не допускается, если не задан параметр: `@define allow-config-dups 1`.
- **Параметр** — список параметров объекта, которые заключаются в фигурные скобки. Параметры разделяются точкой с запятой.
- **Опция** — модификаторы параметров.

Определение объекта завершается символом точки с запятой.

21.2.2.1 Объекты syslog-ng

Система **syslog-ng** оперирует следующими объектами:

- **Путь (англ. log path)** – это комбинация источников, приемников и других объектов. Он указывает, откуда берется сообщение, как оно преобразуется и куда отправляется.
- **Драйвер источника** – метод коммуникации, используемый для получения сообщений.
- **Источник (англ. source)** – именованный набор сконфигурированных драйверов источников.
- **Драйвер приемника** – метод коммуникации, используемый для передачи протоколируемых сообщений.
- **Приемник (англ. destinations)** – именованный набор сконфигурированных драйверов приемников.
- **Фильтр (англ. filters)** – правило, применяемое для отбора сообщений.
- **Анализатор (англ. parsers)** – правило для анализа и разбора сообщений.
- **Модификатор правил (англ. rewriting rules)** – правило для изменения части сообщения.
- **Макрос** – идентификатор, для обращения к определенной части сообщения.
- **Шаблон (англ. template)** – набор макросов для реструктуризации сообщения.
- **Глобальные опции (англ. options)** – набор глобальных опций для детальной настройки syslog-ng

Источники и драйверы источников

Для источников служба syslog-ng использует следующий синтаксис:

```
source <идентификатор_объекта> { драйвер_источника(параметры); ... };
```

Например:

```
source s_src { system(); internal(); };
```

Это же определение объекта может быть записано в несколько строк:

```
source s_src {
      system();
      internal();
};
```

По правилам именования в названии источника следует использовать префикс «s_». Полный спи-

сок поддерживаемых драйверов представлен далее в табл. 1.10.

табл. 1.10 – Поддерживаемые типы драйверов источника

Название	Описание
syslog()	Прием сообщений по протоколу IETF-syslog.
internal()	Сообщения, генерируемые службой syslog-ng.
system()	Автоматическое определение платформы и сбор со стандартных для этой платформы журналов
systemd-journal()	Прямое получение сообщений от служб журналов на платформах, использующих systemd.
systemd-syslog()	Получение сообщений через сокет от служб журналов на платформах, использующих systemd.
file()	Получение сообщений из указанного файла.
pipe()	Чтение сообщений из указанного именованного потока.
unix-dgram()	Получение сообщений через указанный сокет SOCK_DGRAM.
unix-stream()	Получение сообщений через указанный сокет SOCK_STREAM.
network()	Прием сообщений от удаленного хоста по протоколу BSD-syslog в сетях IPv4 и IPv6. Поддерживает сетевые протоколы UDP, TCP, в том числе с TLS.
program()	Запуск указанного приложения и чтение сообщений из его стандартного вывода.

Приемники и драйверы приемников

Для приемников служба syslog-ng использует следующий синтаксис:

```
destination <идентификатор_объекта> { драйвер_источника(параметры); ... };
```

Например:

```
destination d_auth { file("/var/log/auth.log"); };
```

По правилам именования в названии приемника следует использовать префикс «d_». Полный список поддерживаемых драйверов представлен далее в табл. 1.11.

табл. 1.11 – Поддерживаемые типы драйверов приемника

Название	Описание
syslog()	Отправка сообщений на удаленный хост по протоколу IETF-syslog protocol. Поддерживает сетевые протоколы TCP, UDP и TLS.
usertty()	Отправка сообщений на терминал указанного пользователя, если сессия пользователя активна.
file()	Запись сообщений в указанный файл.
pipe()	Запись сообщений в указанный именованный поток.
unix-dgram()	Отправка сообщений в формате BSD через указанный сокет SOCK_DGRAM.
unix-stream()	Отправка сообщений в формате Linux через указанный сокет SOCK_STREAM.
network()	Отправка сообщений на удаленный хост по протоколу BSD через сети IPv4 IPv6. Поддерживает сетевые протоколы TCP, UDP и TLS.
program()	Запуск указанной программы и отправка сообщений на ее стандартный ввод.
sql()	Сохранение сообщений в СУБД SQL. Требует установки и настройки СУБД.
elasticsearch(), elasticsearch2()	Отправка сообщений на сервер Elasticsearch. Вариант драйвера elasticsearch2 поддерживает Elasticsearch версии 2 и новее.
hdfs()	Запись сообщений в указанный файл на узле Hadoop Distributed File System (HDFS).
kafka()	Публикация сообщений подписчикам через шину Apache Kafka.
loggly()	Отправка сообщения провайдеру Loggly (https://www.loggly.com/)
logmatic()	Отправка сообщения провайдеру Logmatic.io (https://logmatic.io/)
mongodb()	Сохранение сообщений в СУБД MongoDB.

Фильтры в файле конфигурации syslog-ng

Для фильтров служба syslog-ng использует следующий синтаксис:

```
filter <идентификатор_объекта> { тип_фильтра(фильтрующее_выражение); ... };
```

Например:

```
filter f_auth {
   facility(auth, authpriv) and not filter(f_debug) and not filter(f_audit);
};
```

Как вы можете заметить, допускается использование логических операндов «**and**» и «**not**». По правилам именования в названии фильтра следует использовать префикс «f_». Полный список поддерживаемых фильтров представлен далее в табл. 1.12.

табл. 1.12 – Поддерживаемые типы фильтров

Название	Описание
filter()	Вызов другого фильтра.
level(), priority()	Фильтрация сообщений по приоритету (severity, см. табл. 1.9). Допускает перечисление в виде массива, например: level(warn .. emerg) – все события с уровнем критичности от предупреждения до экстренных. .
facility()	Фильтрация сообщений по указанному объекту (facility, см. табл. 1.8).
host()	Фильтрация сообщений по имени хоста-отправителя.
message()	Фильтрация по регулярным выражениям Perl*, применяемым к содержимому сообщения.
source()	Фильтрация сообщений по указанному источнику.
program()	Фильтрация сообщений по названию приложения.
netmask()	Фильтрация сообщений по IP-адресу хоста-отправителя.
match()	Фильтрация по регулярным выражениям Perl, применяемым к заголовку и содержимому сообщения. Дополнительную информацию можно получить командой `man pcre`.
tags()	Фильтрация сообщений по наличию указанных тегов.
inlist()	Фильтрация сообщений по черным и белым спискам в файлах.

Пути в файле конфигураций syslog-ng

Для путей служба syslog-ng использует следующий синтаксис:

```
log{ source(...); filter(...); destination(...); };
```

Например:

```
log { source(s_src); filter(f_auth); destination(d_auth); };
```

21.2.2.2 Формат сообщений о событиях

В настоящее время существует два стандартных формата сообщений syslog:

- BSD-syslog (классический)
- IETF-syslog (новый)

Формат BSD-syslog

Общий размер сообщения не может быть больше 1024 байт, формат сообщения:

```
<PRIORITY><HEADER><MESSAGE>
```

Где:

- <**PRIORITY**> – как уже было сказано выше, приоритет кодирует значение категории субъекта, т.е. категорию подсистемы, от которой получено сообщение, и уровень важности события (*англ. Severity* — строгость). Используется формула $(facilities \cdot 8) + severity$.
- <**HEADER**> – содержит метку времени и имя хоста (без имени домена) или IP-адрес устройства.
- <**MESSAGE**> – содержит имя программы или процесса, сгенерировавшего сообщение, и текст самого сообщения.

Пример сообщения:

```
<165>Oct 11 22:14:15 mymachine auth[8710]: 'su root' failed for ivani on /dev/pts/8
```

Формат сообщения IETF-syslog

Формат сообщения:

```
<ЗАГОЛОВОК><СТРУКТУРИРОВАННЫЕ ДАННЫЕ><СООБЩЕНИЕ>
```

Где:

- <**ЗАГОЛОВОК**> состоит из:
  - **PRIORITY** — представляет Facility и Severity сообщения, рассчитывается по той же формуле: $(facilities \cdot 8) + severity$.
  - **VERSION** — номер версии стандарта протокола syslog. В настоящее время это значение может быть только 1.
  - **ISOTIMESTAMP** — время создания сообщения в формате ISO 8601 (yyyy-mm-ddThh:mm:ss+-ZONE).
  - **HOSTNAME** — устройство, которое первоначально отправило сообщение.
  - **APPLICATION** — устройство или приложение, которое сгенерировало сообщение.
  - **PID** – имя или идентификатор процесса приложения, отправившего сообщение.
  - **MESSAGEID** – идентификационный номер сообщения.
- <**СТРУКТУРИРОВАННЫЕ ДАННЫЕ**> – часть сообщения может содержать метаинформа-

цию о сообщении syslog или специфическую для приложения информацию, такую как счетчики трафика или IP-адреса.

Структурированные данные состоят из блоков данных, заключенных в скобки ([]). Каждый блок включает в себя идентификатор блока и одну или несколько пар имя=значение.

- <**СООБЩЕНИЕ**> – содержит текст самого сообщения в UTF-8.

Пример сообщения IETF-syslog:

```
<165>1 2003-10-11T22:14:15.003Z mymachine.example.com evntslog 4322 ID47␣
↪[exampleSDID@32473 iut="3" eventSource="Application" eventID="1011"] An application␣
↪event log entry...
```

21.2.2.3 Настройка централизованного журналирования

Служба syslog-ng позволяет отправлять сообщения от клиентов на сервер для возможности их централизованного хранения. Эта возможность задействована в продукте ALD Pro для построения подсистемы аудита. При выполнении настройки изменения нужно вносить как на стороне клиента, так и на сервере. Опишем общий порядок настройки.

Настройка syslog-ng на сервере

1. Создать сертификат и приватный ключ для сервера и разместить их в подкаталогах cert.d, key.d, ca.d каталога `/etc/syslog-ng`.
2. Изменить конфигурационный файл `/etc/syslog-ng/syslog-ng.conf` – определить в нем:
   - Источник, используя драйвер network(), указав в нем значения параметров port, transport, tls.
   - Приемники для логов, которые будут поступать на сервер с клиентов.
   - Пути сообщений, связав определённые источники и приемники и при необходимости фильтры, анализаторы и модификаторы.
3. Перезапустить службу syslog-ng.

Пример настройки сервера:

```
source s_tls_source {
network( ip("0.0.0.0") port(6514) transport("tls")
tls(
   key-file("/etc/syslog-ng/key.d/serverkey.pem")
   cert-file("/etc/syslog-ng/key.d/servercert.pem")
   ca-dir("/etc/syslog-ng/ca.d")));
```

(continues on next page)

(продолжение с предыдущей страницы)

```
};

destination d_tls_kern { file("/var/log/hosts/$HOST/kern.log"); };

log {source(s_tls_source); filter(f_kern); destination(d_tls_kern); };
```

Настройка syslog-ng на клиентах

1. Создать сертификат и приватный ключ для клиента и настроить доверие к сертификату сервера.
2. Изменить конфигурационный файл `/etc/syslog-ng/syslog-ng.conf` – определить в нем:
   - Приемники для логов, передаваемых на сервер журналирования, используя драйвер network(), указав в нем значения параметров IP-адрес, port, transport, tls.
   - Указать путь сообщений, связав локальные источники с приемниками на сервере.
3. Перезапустить службу syslog-ng.

Пример настройки клиента:

```
destination d_tls_destination {

network ("<ip_адрес>" port(6514) transport("tls")

tls(
   key-file("/etc/syslog-ng/key.d/clientkey.pem")
   cert-file("/etc/syslog-ng/key.d/clientcert.pem")
   ca-dir("/etc/syslog-ng/ca.d"))); };

log { source(s_src); destination(d_tls_destination); };
```

Управление службой с помощью утилиты syslog-ng-ctl

Для управления syslog-ng может применяться утилита `syslog-ng-ctl`, основные команды которой:

- Команда `syslog-ng-ctl stats` – вывести статистику по использованию источников и приемников.
- Команда `syslog-ng-ctl <verbose|trace|debug>` – изменить режим логирования, например, чтобы включить режим отладки первого уровня необходимо выполнить команду: `sudo syslog-ng-ctl debug --set=1`
- Команда `syslog-ng-ctl reload|stop` – перезапустить или остановить службу.

- Команда `syslog-ng -s` – проверить синтаксис конфигурационных файлов.

Дополнительную информацию читайте в справке `man syslog-ng` и `man syslog-ng-ctl`.

21.2.3 Система журналирования systemd-journald

Служба **systemd-journald** входит в состав подсистемы инициализации **systemd** и журналирует:

- сообщения от ядра;
- сообщения, отправляемые с помощью стандартной библиотечной функции syslog();
- сообщения, отправляемые с помощью «родного» для systemd-journald API;
- сообщения из стандартных потоков вывода (stdout, stderr) служб, запускаемых с помощью systemd;
- сообщения из подсистемы аудита.

Настройка **systemd-journald** осуществляется через конфигурационный файл `/etc/systemd/journald.conf` и с помощью утилиты `journalctl`.

Если заглянуть в этот файл, то можно увидеть, что все параметры закомментированы, но следует обратить внимание на параметр Storage, который может иметь следующие значения:

- **volatile** – в файл `/run/log/journal` (временная память);
- **persistent** – в файл `/var/log/journal` (постоянная память);
- **auto (по умолчанию)** – если есть каталог `/var/log/journal`, то в нем, а если нет, то в `/run/log/journal`.

В Astra Linux параметр Storage не задан, поэтому по умолчанию предполагается auto, и каталог `/run/log/journal` не существует, поэтому система сохраняет файлы журналов только во временной памяти.

Для работы с systemd-journald используется утилита `journalctl`. Например, при вызове `sudo journalctl` без параметров будут выведены все сообщения системы:

```
localadmin@astra:~$ sudo journalctl | head -n 3
-- Logs begin at Mon 2023-09-25 08:48:40 MSK, end at Tue 2023-09-26 11:37:34 MSK. --
сен 25 08:48:40 astra kernel: Linux version 5.15.0-70-generic (builder@build5) (gcc␣
↪(AstraLinuxSE 8.3.0-6) 8.3.0, GNU ld (GNU Binutils for AstraLinux) 2.31.1) #astra2␣
↪SMP Mon Apr 10 13:35:08 UTC 2023 (Astra 5.15.0-70.astra2-generic 5.15.92)
сен 25 08:48:40 astra kernel: Command line: BOOT_IMAGE=/boot/vmlinuz-5.15.0-70-
↪generic root=UUID=1864fbac-4e75-42ec-9a52-c0944d4fd6b9 ro parsec.mac=0 quiet net.
↪ifnames=0
...
```

Некоторые полезные опции утилиты:

- Команда `journalctl -f` – просмотр событий в реальном времени (аналог `tail -f <имя_журнала>`).
- Команда `journalctl -u <имя_юнита>` – сообщения от определенного юнита systemd.
- Команда `journalctl -k` – сообщения от ядра.
- Команда `journalctl -p <приоритет>` – вывод сообщений только определённого приоритета (уровня критичности).
- Команда `journalctl -n <число>` – вывод последних n сообщений (аналог `tail –n <число> <имя_журнала>`)
- Команда `journalctl <имя_поля=значение>` – вывод сообщений с фильтрацией по значению полей. Чаще всего используют поля _UID, _PID и _SYSTEMD_UNIT.
- Команда `journalctl -N` – вывод списка полей, используемых в сообщениях.
- Команда `journalctl -F <поле>` – вывод списка возможных значений для указанного поля.
- Команда `journalctl -o verbose` — вывод сообщения в расширенном формате, т.е. кроме сообщения выводится список полей и их значений.

Пример поиска ошибок в логах с применением команды `grep`:

```
localadmin@astra:~$ sudo journalctl --since "1 hour ago" | grep err
[sudo] пароль для localadmin:
авг 07 17:34:07 astra spice-vdagent[2195]:    error message: Cannot invoke method;␣
↪proxy is for a well-known name without an owner and proxy was constructed with the␣
↪G_DBUS_PROXY_FLAGS_DO_NOT_AUTO_START flag
авг 07 17:34:07 astra spice-vdagent[2195]:    error message: Cannot invoke method;␣
↪proxy is for a well-known name without an owner and proxy was constructed with the␣
↪G_DBUS_PROXY_FLAGS_DO_NOT_AUTO_START flag
```

Для отправки сообщений в **systemd-journald** можно использовать утилиты `systemd-cat` и `logger`. С помощью опции `-p` можно задавать приоритет сообщения. Примеры использования:

```
localadmin@astra:~$ echo "Что-то пошло не так!" | sudo systemd-cat -p 3
localadmin@astra:~$ sudo journalctl -p 3 -n 1
-- Logs begin at Mon 2023-09-25 08:48:40 MSK, end at Tue 2023-09-26  12:01:54 MSK. --
сен 26 12:01:45 astra cat[8361]: Что-то пошло не так!
```

21.2.4 Настройка ротации журналов

Система ротации журналов `logrotate` позволяет очищать или архивировать файлы журналов по достижению определенных лимитов, например, по размеру или возрасту. По умолчанию утилита

запускается ежедневно через планировщика заданий `cron`, см. файл `/etc/cron.daily/logrotate`.

Основной конфигурационный файл находится в `/etc/logrotate.conf`. Дополнительные файлы с суффиксом `*.conf` можно разместить в каталоге `/etc/logrotate.d/`.

Посмотрим настройки по умолчанию (основной файл конфигурации `/etc/logrotate.conf`):

```
# see "man logrotate" for details
# rotate log files weekly
weekly

# keep 4 weeks worth of backlogs
rotate 4

# create new (empty) log files after rotating old ones
create

# use date as a suffix of the rotated file
#dateext

# uncomment this if you want your log files compressed
#compress

# packages drop log rotation information into this directory
include /etc/logrotate.d

# system-specific logs may be also be configured here.
```

Где:

- **weekly** – период ротации. Файлы журналов подвергаются ротации weekly, если текущий день недели меньше дня недели последней ротации или если со времени последней ротации прошло больше недели. Возможные значения weekly, daily, monthly, yearly.
- **rotate** <**количество**> – файлы журнала ротируются указанное количество раз перед тем, как будут удалены или отправлены на адрес, указанный в директиве mail. Если количество равно нулю, прежние версии удаляются, а не ротируются. Количество по умолчанию – 4.
- **create** <**режим**> <**владелец**> <**группа**> – сразу после ротации (до запуска сценария postrotate) создается файл журнала (с таким же именем, как у только что ротированного файла журнала). Назначение полей:
  - <**режим**> — определяет права доступа для файла журнала в восьмеричном представлении (как в утилите chmod).
  - <**владелец**> — определяет пользователя, который будет назначен владельцем файла.
  - <**группа**> — определяет группу, которая будет назначена владельцем файла. Если какое-то из полей не задано, будут использоваться параметры исходного файла журнала. Эта опция может быть отключена с помощью опции **nocreate**.

Дополнительно используются следующие директивы:

- **compress** – сжимать старые журналы.
- **maxsize <размер>** – ротация по размеру.
- **maxage <количество_дней>** – предписывает удалять журналы, у которых возраст старше указанного.
- **postrotate <скрипт>** – указывает, какой скрипт должен быть выполнен после ротации.

21.2.5 Графическая утилита «Системный журнал»

В Astra Linux доступна графическая утилита для просмотра журналов и событий «Системный журнал» (KSystemlog), см. рис. 1.149. Запустить ее можно из командной строки `sudo ksystemlog` или из меню *Пуск → Панель управления → Безопасность → Журнал аудита*.

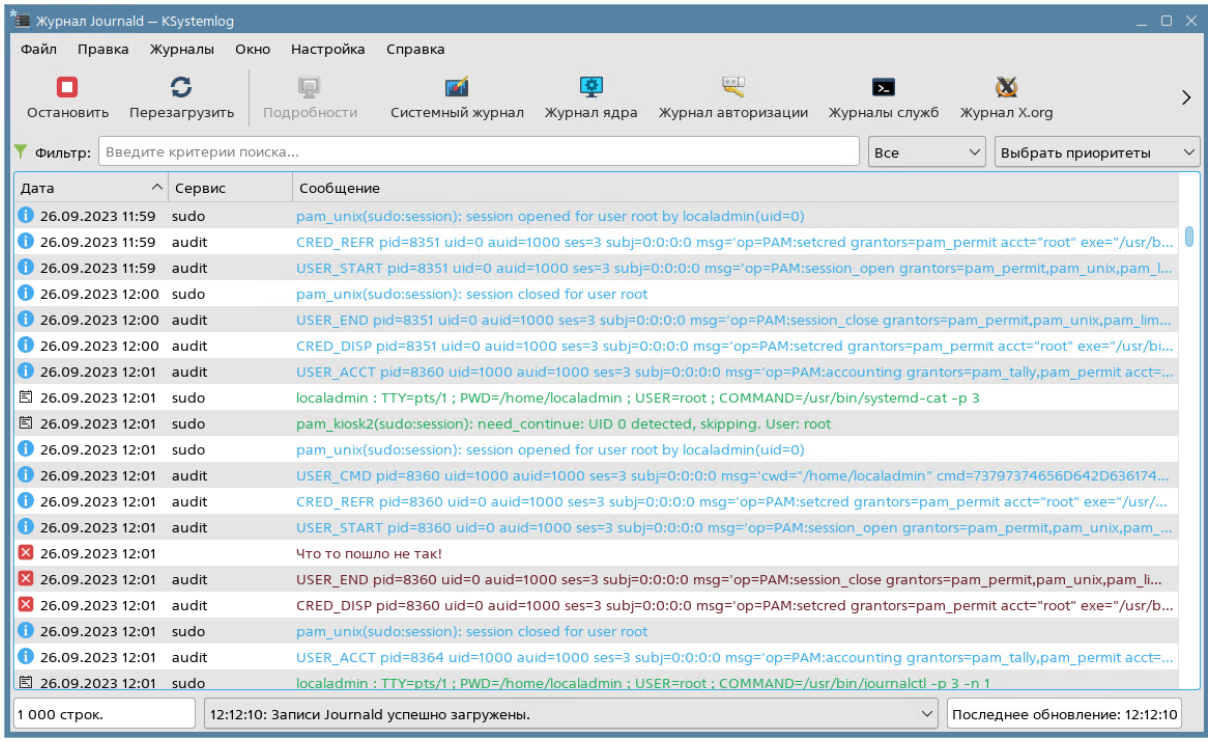


рис. 1.149 – Окно утилиты «Системный журнал»

Утилита «Системный журнал» позволяет просматривать события из основных системных журналов, журнала ядра, аудита, графической подсистемы, служб systemd. В том числе вы можете расширять список источников. Работая с утилитой, вам доступен также поиск сообщений по заданным критериям.

21.2.6 Инструмент astra-create-debug-logs

В состав Astra Linux входит утилита `astra-create-debug-logs`, с помощью которой можно автоматически собрать архив с файлами журналов системных служб.

Запускать утилиту нужно от имени суперпользователя, файл будет создан в каталоге `/tmp/`.

```
localadmin@astra:~$ sudo astra-create-debug-logs
[sudo] пароль для localadmin:
cur_ilev=0   sysmaxilev=0
Создание архива с системными журналами и конфигурацией ОС ASTRALINUX 1-7-4 ..
Архив с системными журналами ASTRALINUX-1-7-4:
-rw-r--r-- 1 root root 5,3M авг  7 17:51 /tmp/astra-logs-1-7-4-07_08_2024-17_51_32.
↪tar.xz
```

21.3 Сбор информации о работе ОС и ее производительности

Информацию о работе операционной системы и производительности отдельных ее компонентов можно получить множеством способов. Общую информацию об использовании ЦП, памяти, файла подкачки и сети можно посмотреть, например, с помощью графической утилиты «Системный монитор», см. рис. 1.150.

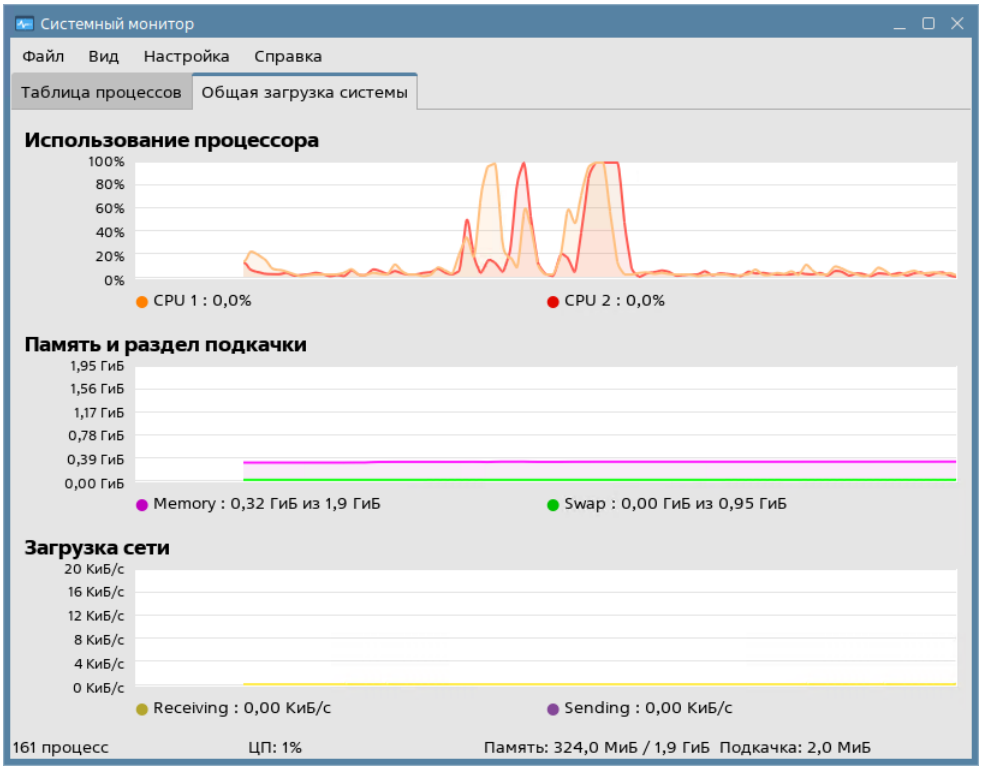


рис. 1.150 – Окно программы «Системный монитор»

Однако для получения более детализированной информации необходимо использовать специальные утилиты, см. рис. 1.151.

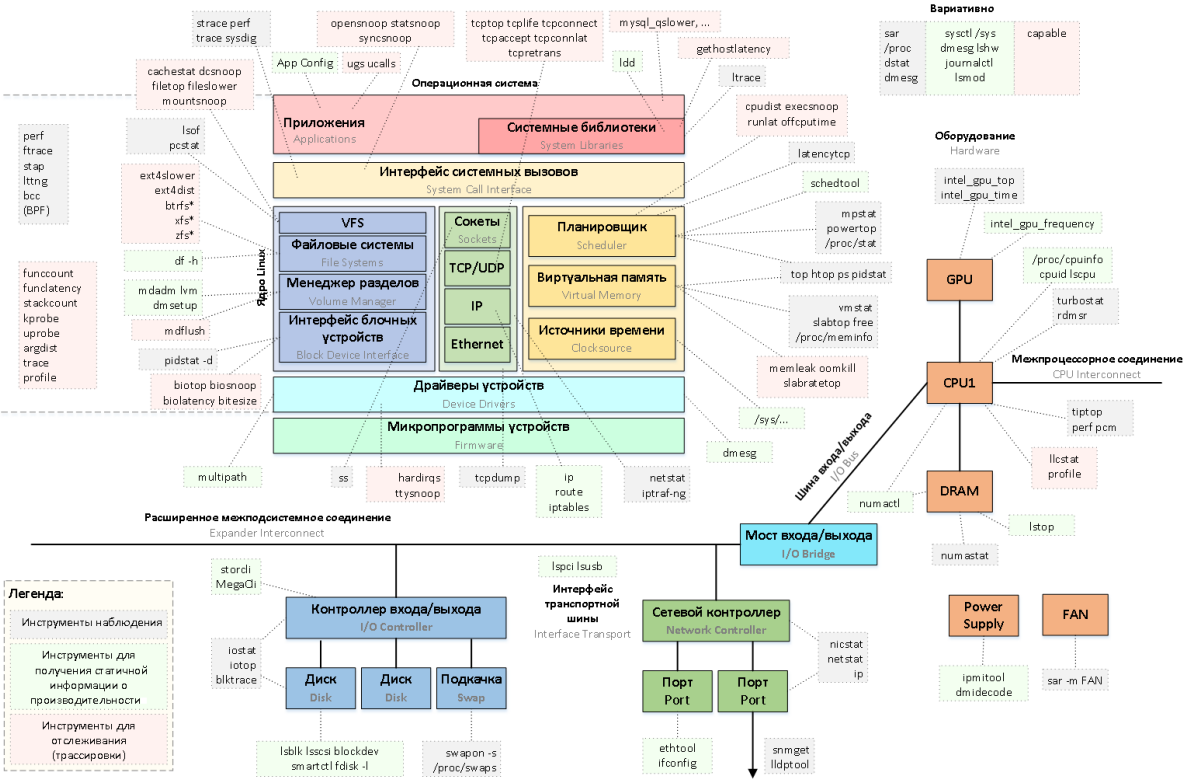


рис. 1.151 – Утилиты и источники информации для проверки и мониторинга подсистем Linux

21.3.1 Информация о процессах

Утилиты ps, top, htop служат для получения информации о процессах и потоках в системе. Работа с ними была рассмотрена в модуле, посвященном процессам.

Утилита vmstat (*от англ. Virtual Machine Statistic*) показывает информацию о процессах, памяти, страницах, I/O и о работе процессора. Пользователи vmstat могут указать интервал дискретизации, который дает возможность наблюдать работу системы в близком к реальному времени.

21.3.2 Информация о подсистеме хранения

Утилита iostat (*от англ. I/O Statistic*) входит в пакет sysstat и позволяет проанализировать загруженность системы. Она выводит основные параметры ввода/вывода данных, а также количество записанных или прочитанных данных. Кроме того, утилита выводит параметры загруженности процессора. Её можно использовать для оптимизации работы системы.

Утилита iotop (*от англ. I/O Top*) входит в пакет iotop и позволяет отслеживать в реальном времени загрузку дисковой подсистемы по процессам. Аналог top, но в разрезе нагрузки на дисковую подсистему.

Утилита lsof (*от англ. LiSt of Open Files*) позволяет просматривать информацию о том, какие

процессы и пользователи открыли файл.

21.3.3 Системные вызовы

Утилита `strace` (*от англ. System Trace*) используется для мониторинга и влияния на взаимодействия между процессами и ядром Linux, включая системные вызовы, доставку сигналов и изменения состояния процесса. Утилита использует функции ядра ptrace, поэтому для ее использования нужно отключить блокировку этого механизма командой `astra-ptrace-lock disable`.

Утилита `ltrace` (*от англ. Library Trace*) просто запускает и исполняет указанную команду до завершения. Она перехватывает и записывает вызовы динамических библиотек, вызываемые исполняемым процессом, и сигналы, полученные этим процессом. Она также может перехватывать и выводить системные вызовы, выполняемые программой.

Утилита `ldd` (*от англ. List Dynamic Dependencies*) выдает список зависимостей от общих объектов (списка разделяемых библиотек, используемых исполняемыми файлами или разделяемыми библиотеками).

Утилита `ftrace` (*от англ. Function Tracer*) позволяет узнать, что происходит внутри ядра. Ее можно использовать для отладки или анализа задержек и проблемы с производительностью, возникающие за пределами пользовательского пространства.

21.3.4 Сеть и еще раз сеть

Утилита `netstat` (*от англ. Network Statistic*) входит в пакет net-tools и позволяет просматривать состояние TCP-соединений (как входящих, так и исходящих), таблицы маршрутизации, число сетевых интерфейсов и сетевую статистику по протоколам.

Утилита `ss` (*от англ. Socket Statistic*) входит в пакет iproute2. Она является аналогом утилиты netstat, но работает быстрее и проще в использовании. Она даёт более подробные сведения о TCP-подключениях и о состояниях соединений, чем большинство других инструментов.

Утилита `route` входит в пакет iproute2, позволяет работать с таблицами IP-маршрутизации.

Утилита `iptables` позволяет настраивать правила фильтрации IP-пакетов на низком уровне.

Служба `ufw` (*от англ. Uncomplicated FireWall*) – упрощенный файрвол, который позволяет изменять настройки межсетевого экрана.

Утилита `tcpdump` позволяет захватывать сетевой трафик с учетом фильтров. Вывод может быть направлен как в файл, так и на стандартный вывод. Это наиболее часто используемый инструмент для устранения неполадок, связанных с работой сетевых приложений.

Приложение `wireshark` — это очень популярный инструмент для захвата и анализа сетевого трафика. Приложение поддерживает парсинг не только протоколов TCP/ IP, но и большинства протоколов прикладного уровня, включая DNS, NTP, HTTP, LDAP, Kerberos и др. А для возможности парсинга TLS-трафика вы можете предоставить приложению доступ к файлу с сессионными ключами.

Внимание: Подавляющее большинство проблем в домене возникает между компонентами, которые взаимодействуют друг с другом по сети, поэтому инструменты из этого раздела и конкретно Wireshark, если не серебряная пуля, то уж точно хлеб с маслом для каждого доменного администратора!

21.3.5 Универсальные программы и наборы утилит

Когда вы освоитесь с базовыми инструментами, можно будет посмотреть на универсальные программы сбора статистики и анализа производительности.

Утилита `perf` (*от англ. Performance*) является инструментом для сбора статистики и анализа производительности. Счетчики производительности для Linux — это подсистема на базе ядра, которая закладывает основу для анализа производительности. Она охватывает функции аппаратного уровня (CPU/PMU, блок мониторинга производительности), а также функции программного обеспечения (программные счетчики, точки трассировки).

Для использования необходимо установить пакет linux-tools-common и возможно специализированный пакет для текущей версии ядра linux-tools-<версия ядра>.

Документация: https://perf.wiki.kernel.org/index.php/Main_Page.

Утилита `stap` (*от англ. System Tap*) позволяет собирать и анализировать информацию о работающей Linux системе. В отличие от встроенных средств, таких как netstat, ps и top, это приложение было разработано с целью предоставить больше возможностей для сбора и предоставления информации.

Утилита представляет собой интерфейс командной строки и скриптовый язык программирования. Системные администраторы могут использовать SystemTap для мониторинга и анализа производительности системы, а разработчики программного обеспечения могут использовать SystemTap для анализа поведения приложения в работающей системе.

Документация доступна по ссылке http://sourceware.org/systemtap/.

Dtrace – это инструмент для динамической трассировки ядра системы и приложений в реальном

времени, главным образом, с целью их профайлинга и отладки.

eBPF (от англ. Enhanced Berkeley Packet Filter) – это механизм ядра, предоставляющий возможности трассировки и профайлинга как самого ядра, так и пользовательских приложений. Разработан как портированная и расширенная версия BPF (фильтр, умеющий, например, фильтровать пакеты для libpcap / tcpdump прямо в ядре).

Для использования необходимо установить пакет bpftrace.

Документация: https://github.com/iovisor/bpftrace.

Bpftrace (*от англ. Berkeley Packet Filter Trace*) — это инструмент, аналогичный DTrace, реализованный поверх eBPF. Считается стабильным, быстрым и пригодным для запуска в боевом окружении.

bcc (*от англ. BPF Compiler Collection*) – набор инструментов для создания программ, использующих возможности eBPF. При написании кода с использованием bcc в качестве языка программирования используется смесь Python и C. Также в bcc входит большой набор готовых утилит для профайлинга программ и мониторинга системы.

В состав bcc входят более 100 инструментов. Из них можно выделить следующие инструменты, полезные для администрирования:

- Утилита `bashreadline` — вывод введенных команд bash для всей ОС (во всех сеансах).
- Утилита `biotop` — top для дисков: итоговый ввод-вывод блочных устройств по процессам.
- Утилита `biosnoop` — отслеживание блокировок устройств ввода-вывода с помощью PID и задержек.
- Утилита `dirtop` — top для каталогов: операции чтения и записи по каталогам.
- Утилита `execsnoop` — отслеживание новых процессов с помощью системных вызовов exec().
- Утилита `exitsnoop` — отслеживание завершения процесса (сигналы выхода и сигналы о фатальных ошибках).
- Утилита `ext4dist` — итоговое распределение задержек операций ext4 в виде гистограммы.
- Утилита `ext4slower` — отслеживание медленных операций ext4.
- Утилита `filetop` — top для файлов: операции чтения и записи по файлам.
- Утилита `mountsnoop` — отслеживание системных вызовов монтирования и размонтирования по всей системе.
- Утилита `oomkill` — отследить убийцу процесса, чье убийство вызвано нехваткой памяти

(OOM).

- Утилита `tcptracer` — отслеживание установленных TCP-соединений (connect(), Accept(), close()).

Для использования необходимо установить пакеты: bpfcc-tools, libbpfcc и libbpfcc-dev

Документация: https://github.com/iovisor/bcc.

Пакет **sysstat** (*от англ. System Statistic*) — это набор системных утилит для контроля производительности активности использования операционной системы Linux и её подсистем. С помощью приложений можно получить информацию о системе, скорости операций ввода-вывода, использовании центрального процессора, памяти, подкачки, прерываний, сети, tty и о многом другом.

В состав пакета входят следующие утилиты:

- Утилита `iostat` — сообщает статистику ЦП и статистику ввода/вывода для блочных устройств.
- Утилита `mpstat` — сообщает статистику по отдельному или объединенному процессору.
- Утилита `pidstat` — сообщает статистику по задачам (процессам) Linux: ввод-вывод, процессор, память и т. д.
- Утилита `cifsiostat` — сообщает статистику CIFS.

Вы можете также запланировать выполнение некоторых утилит через cron или systemd для сбора данных о производительности в фоновом режиме:

- Утилита `sar` — собирает и хранит информацию о деятельности системы (см. ниже список показателей, собираемых `sar`).
- Утилита `sadc` — это сборщик данных о деятельности системы, используемый в качестве бэкенда для `sar`.
- Утилита `sa1` — собирает и сохраняет двоичные данные в файле ежедневных данных о деятельности системы. Это интерфейс sadc, предназначенный для запуска из cron или systemd.
- Утилита `sa2` — пишет сводный ежедневный отчет о деятельности. Это интерфейс `sar`, предназначенный для запуска из cron или systemd.
- Утилита `sadf` — отображает данные, собранные sar, в нескольких форматах (CSV, XML, JSON и т. д.) и может использоваться для обмена данными с другими программами. Эту команду также можно использовать для рисования графиков различных действий, собранных sar, в формате SVG (масштабируемая векторная графика).

Интервал выборки по умолчанию составляет 10 минут, но его, конечно, можно изменить (он может составлять всего 1 секунду).

Документация: https://github.com/sysstat/sysstat.

21.4 Поиск и устранение неисправностей

Термин «баг» вошел в инженерный жаргон еще во времена Эдисона, что было задолго до первых компьютеров, и означал небольшие неисправности в работе оборудования, а ремонтников называли соответственно «багхантерами». Очевидно, что термин перекочевал и к инженерам-программистам, поэтому, когда в компьютере Mark II была обнаружена проблема из-за мотылька, застрявшего в реле, эту историю в шутку назвали «первым случаем обнаружения реального бага».

Но как бы не назывался процесс устранения неисправностей — отладка или дебаггинг — он был, есть и остается крайне важным элементом обеспечения надежной работы технических систем.

21.4.1 Подходы к выполнению отладки

Как в программировании вычислительная сложность алгоритмов может отличаться в несколько раз, так при выполнении отладки вы можете использовать определенные подходы, которые значительно повысят ваши шансы на успех и сократят общее время решения проблемы. Сами по себе эти рекомендации не принесут вам удачи, но чем больше вы будете тренироваться в их использовании, тем чаще вам будет везти.

21.4.1.1 Воспроизведите ошибку

Как ни странно, но первый шаг на пути решения любой технической проблемы – это способность воспроизвести ее, то есть воссоздать условия, в которых эта ошибка возникает.

В первую очередь вам нужно понять, какие именно настройки или действия приводят к появлению ошибки, так как наблюдение за «жуком» в его естественной среде обитания даст возможность локализовать проблему, изменяя внешние условия ее воспроизведения. Это так называемый «динамический анализ».

Именно поэтому многие службы технической поддержки делают обязательным поле «Шаги по воспроизведению» при подаче заявки, но, как говорится, у правильной девушки в сумочке всегда должен быть загранпаспорт, купальник, фата и… губозакаточная машинка.

21.4.1.2 Локализуйте ошибку

Если вы научились воспроизводить ошибку, то вам нужно понять, в какой части системы она возникает. Наиболее популярной стратегией при этом, к сожалению, является метод «перебора», он же «грубой силы» или «метод тыка», так как он не требует глубоких познаний о работе системы. Как говорится: «Что тут думать? Прыгать надо!». Но журналов в системе слишком много, поэтому мы настоятельно рекомендуем хотя бы проанализировать процесс обработки данных, в рамках которого возникает ошибка, чтобы сократить круг подозреваемых.

Однако более эффективной стратегией является применение бинарного поиска при локализации ошибки — да, да, это про быстрый поиск по сортированному массиву данных, который имеет сложность *O (log(n))*. Если проанализировать любую ИТ-технологию, то это всегда окажется последовательный процесс обработки данных компонентами системы `A->B->C->D->E->F->G`, поэтому вы можете поделить эту цепочку пополам и внимательно проанализировать компонент `D`, который находится посередине. Если вы обнаружите, что до этого компонента данные успешно доходят, то можно будет отбросить первую половину цепочки и сконцентрироваться на изучении компонентов `E-F-G`, применяя тот же подход еще раз, а затем еще раз и еще. Но чем длиннее будет эта цепочка, тем больше времени вам сэкономит бинарный поиск.

Приведем еще один пример, в котором вы можете применить бинарный поиск. Допустим, у вас есть некоторая служба, которая не стартует, когда вы запускаете ее с определенным набором из 50 параметров в конфигурационном файле. В этом случае вы можете очень долго проверять каждый параметр по одному, а можете удалить вторую половину списка, а потом еще половину и... О, вы схватываете на лету!

21.4.1.3 Определите первопричину и устраните ее

А вот теперь наступает самый интересный этап решения проблем: вам нужно определить, что является первопричиной проблемы и как ее можно устранить.

Тут вам потребуется глубоко погрузиться в логику работы приложения, что делают найденные вами параметры и как они влияют на работу компонентов системы. Возможно, потребуется провести серию экспериментов, используя разные значения параметров. Ну и, конечно же, не забываем про «О'кей Гугл, где мои штаны!». Хотя сейчас популярнее уже Chat GPT, но с этим ИИ нужно быть построже, чтобы не купиться на восемь ног у лошади и не довести дело до «мне нужна твоя одежда, сапоги и мотоцикл»)))

21.4.1.4 Зафиксируйте ваши находки в документах

В завершение поговорим о том, что мы так часто забываем: обязательно задокументируйте ваши изыскания, чтобы упростить работу себе и своим коллегам в будущем.

Не поленитесь оставить подробный комментарий непосредственно в конфигурационном файле, в системе отслеживания изменений, а если у вас все по ITIL, то и в системе управления изменениями.

Делайте так, даже если вам кажется, что это была простая проблема, а то ежу, может быть, и понятно, но у него слишком высокий IQ.

21.4.2 Неисправности, возникающие на начальных стадиях загрузки

К неисправностям, возникающим на начальных стадиях загрузки, можно отнести повреждение загрузочного сектора, файлов загрузчика или недостаток места. Рассмотрим эти проблемы и способы их решения на примере виртуальной машины, загрузка которой выполняется по сценарию BIOS->MBR.

21.4.2.1 Создание резервной копии таблицы разделов

С помощью утилиты fdisk мы можем создать резервную копию разделов. Для этого необходимо запустить утилиту, указав необходимое устройство, выбрать пункт «записать разметку в файл сценария sfdisk», ввести команду «O» и указать имя файла `sda.bkp`:

```
localadmin@astra:~$ sudo fdisk /dev/sda
Добро пожаловать в fdisk (util-linux 2.33.1).
Изменения останутся только в памяти до тех пор, пока вы не
решите записать их. Будьте внимательны, используя команду write.

Команда (m для справки): O

Введите имя файла скрипта: sda.bkp

Сценарий записан успешно.

Команда (m для справки): q

localadmin@astra:~$ ls -l sda.bkp
-rw-r--r-- 1 root root 250 сен 26 21:35 sda.bkp
```

Запишем этот файл на другой диск (на один из тех, что были созданы в процессе изучения модуля про работу с файловыми системами и работе с блочными устройствами):

```
localadmin@astra:~$ sudo mount /dev/sdb2 /mnt/data
localadmin@astra:~$ sudo cp sda.bkp /mnt/data/
localadmin@astra:~$ ls /mnt/data
lost+found sda.bkp test.txt
```

21.4.2.2 Создание резервной копии загрузочного сектора

Мы можем восстановить загрузочный сектор путем повторной установки GRUB, а можем просто скопировать первый сектор размером 512 байт. Примонтируем диск и скопируем на него данные начальной области диска, используя утилиту `dd` :

```
localadmin@astra:~$ sudo mount /dev/sdb2 /mnt/data
localadmin@astra:~$ sudo dd if=/dev/sda of=/mnt/data/sda_boot.dd.bkp bs=512 count=4
4+0 записей получено
4+0 записей отправлено
2048 байт (2,0 kB, 2,0 KiB) скопирован, 0,000500627 s, 4,1 MB/s
localadmin@astra:~$ ls /mnt/data
lost+found sda.bkp sda_boot.dd.bkp test.txt
```

21.4.2.3 Создание загрузочного диска

Если политика безопасности компании позволяет, то вы можете воспользоваться готовым образом LiveCD, например, от gparted https://gparted.org/livecd.php. В противном случае вам нужно будет создать загрузочный образ самостоятельно, например, используя ALP-live.

ALP-live (Astra Linux Portable) — это образ операционной системы, предназначенный для работы сразу после загрузки со съемного носителя (CD, DVD, USB-Flash) без установки на жесткий диск. В статьях на wiki.astralinux.ru подробно рассказывается о том, как создать загрузочную USB-флешку, ссылки на статьи:

- Создание собственных ALP-live образов (Live-CD/Live-USB) в Astra Linux
- Создание загрузочных носителей USB flash drive

Создадим образ ALP-live, добавив в него пакет testdisk с одноименной утилитой, с помощью которой можно восстановить поврежденную таблицу разделов на диске. Перед созданием образа необходимо подключить новый диск размером не менее 20 ГБ, так как на дисках, подключённых к виртуальной машине, не хватит места для его размещения.

```
localadmin@astra:~$ sudo lsblk /dev/sdf
[sudo] пароль для localadmin:
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
sdf   8:80   0  20G  0 disk
localadmin@astra:~$ sudo fdisk /dev/sdf
Добро пожаловать в fdisk (util-linux 2.33.1).
Изменения останутся только в памяти до тех пор, пока вы не решите записать их.
Будьте внимательны, используя команду write.
Устройство не содержит стандартной таблицы разделов.
Создана новая метка DOS с идентификатором 0x30b1441c.
Команда (m для справки): g
Created a new GPT disklabel (GUID: 6B5EF8A8-EC37-BB4C-90A8-637820464532).
Команда (m для справки): n
Номер раздела (1-128, default 1): <enter>
```

(continues on next page)

(продолжение с предыдущей страницы)

```
Первый сектор (2048-41943006, default 2048): <enter>
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-41943006, default 41943006):
↪<enter>
Создан новый раздел 1 с типом 'Linux filesystem' и размером 20 GiB.
Команда (m для справки): w
Таблица разделов была изменена.
Вызывается ioctl() для перечитывания таблицы разделов. Синхронизируются диски.
localadmin@astra:~$ sudo mkfs /dev/sdf1
mke2fs 1.44.5 (15-Dec-2018)
Creating filesystem with 5242619 4k blocks and 1310720 inodes
Filesystem UUID: 010804eb-fe4a-44e8-86d9-a77e21a0973e
Superblock backups stored on blocks:
32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208, 4096000
Allocating group tables: done
Writing inode tables: done
Writing superblocks and filesystem accounting information: done
```

Смонтируем новый раздел в `/opt`, так как именно там будет создаваться Live_CD:

```
localadmin@astra:~$ sudo mount /dev/sdf1 /opt
localadmin@astra:~$ df
Файловая система 1K-блоков Использовано Доступно Использовано% Смонтировано в
udev                960260            0   960260             0% /dev
tmpfs               202552         5900   196652             3% /run
/dev/sda1         15421320     11601024  3015128            80% /
tmpfs              1012740           12  1012728             1% /dev/shm
tmpfs                 5120            4     5116             1% /run/lock
/dev/sdb1           485348           14   455638             1% /var/log/myservicelog
/dev/sdf1         20596416           24 19547872             1% /opt
```

Установим пакет live-build-astra командой:

```
localadmin@astra:~$ sudo apt install live-build-astra
```

Добавим «testdisk» в конфигурационный файл /usr/share/live-build-astra/customyze/astra_extend.list для включения этого пакета в образ:

```
localadmin@astra:~$ sudo vim /usr/share/live-build-astra/customyze/astra_extend.list
localadmin@astra:~$ cat /usr/share/live-build-astra/customyze/astra_extend.list
linux-astra-modules-5.4
linux-astra-modules-5.10
testdisk
```

Соберем образ командой `sudo live-build-astra`. Эта операция займет некоторое время.

```
localadmin@astra:~$ sudo live-build-astra
...
итого 3312804
-rw-r--r-- 1 root root 3388997632 сен 27 05:05 1.7_x86-64-1.7.0-27.09.2023_05.04.
↪livecd.iso
```

Скопируем созданный образ с виртуальной машины на хостовую. В случае с VirtualBox это можно сделать через общую папку (доступна в настройках виртуальной машины).

21.4.2.4 Вызов нештатной ситуации в демонстрационных целях

Предупреждение: Если вы хотите повторить эксперименты, предложенные далее, то настоятельно рекомендуем перед этим сделать снимок состояния виртуальной машины.

Для демонстрации намеренно испортим загрузочный сектор. Для этого запишем нули в первые 512 байт на диске с помощью команды:

```
localadmin@astra:~$ sudo dd if=/dev/zero of=/dev/sda bs=512 count=1
1+0 записей получено
1+0 записей отправлено
512 байт скопировано, 0,000190076 s, 2,7 MB/s
```

Теперь перезагрузим машину, чтобы зафиксировать ошибку **No bootable medium found**! Как вы заметили, мы столкнулись с нештатной ситуацией: система не может быть загружена, так как не видит загрузочный диск.

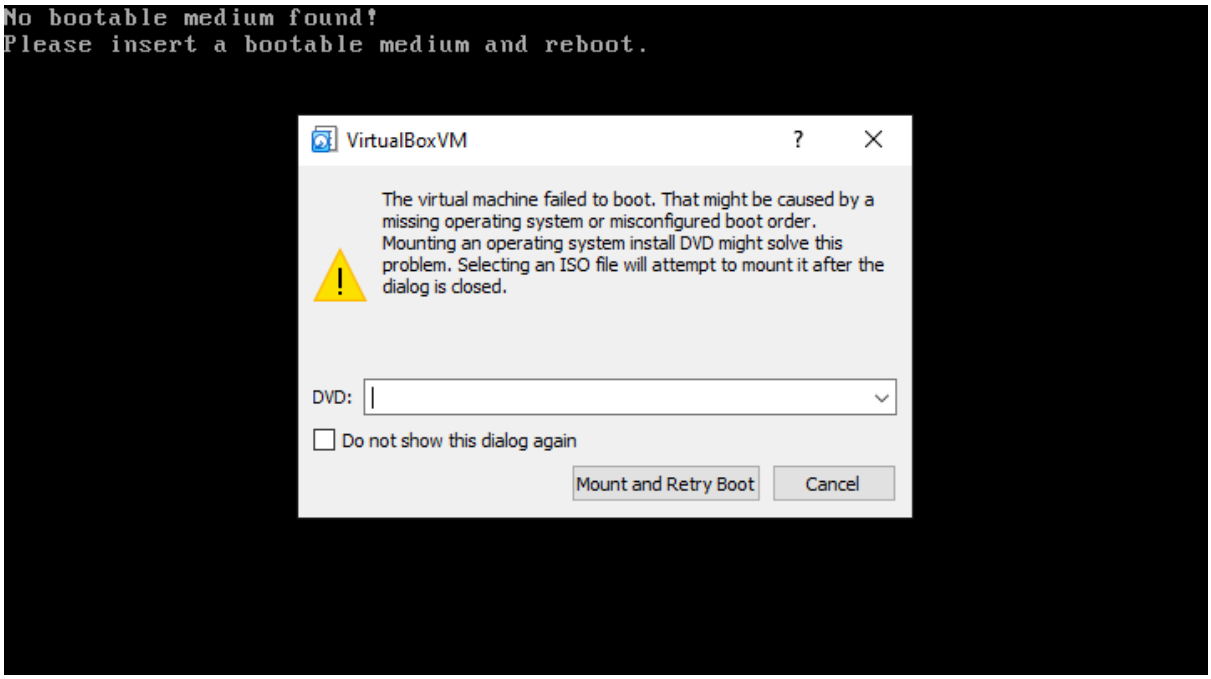
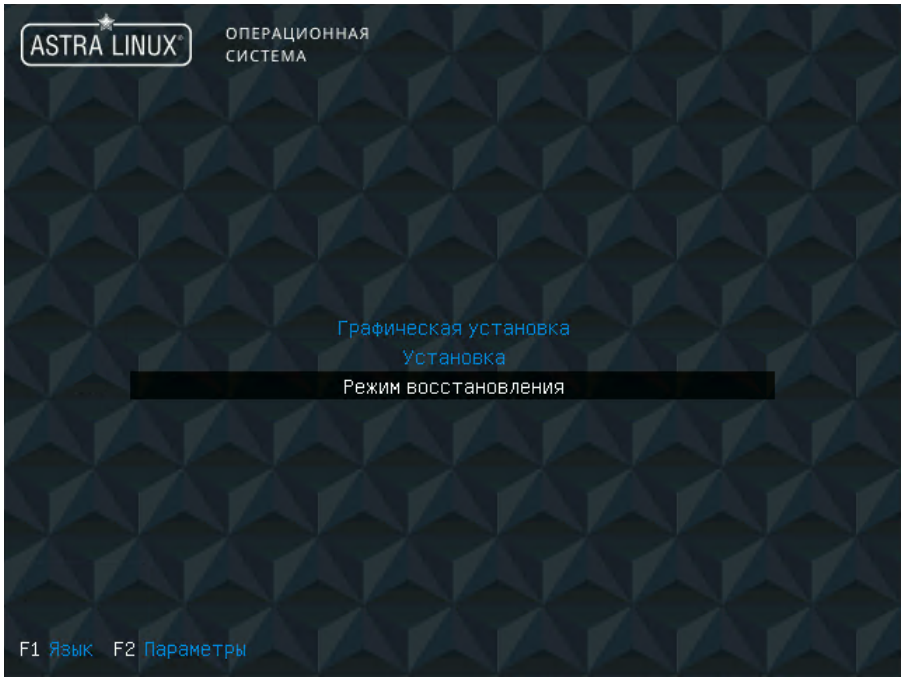


рис. 1.152 – Ошибка No bootable medium found

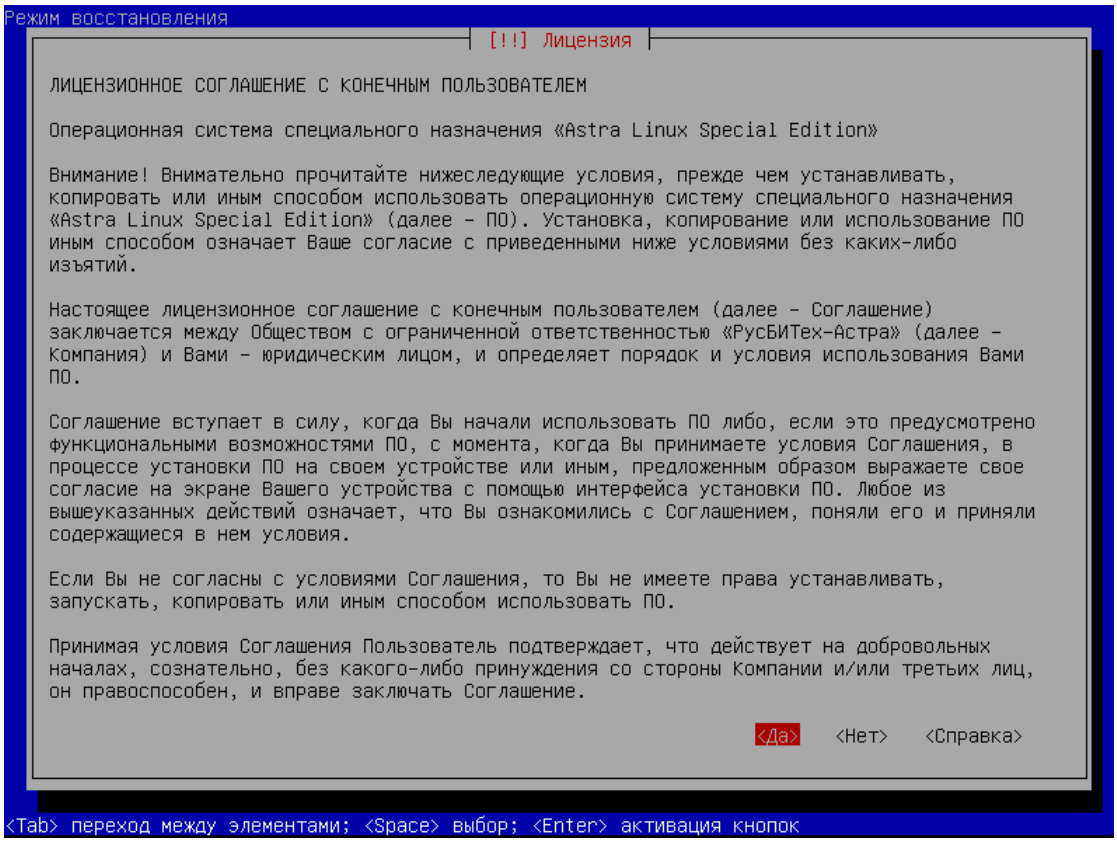
21.4.2.5 Восстановление таблицы разделов из резервной копии

Для восстановления поврежденной таблицы разделов необходимо выполнить ряд шагов:

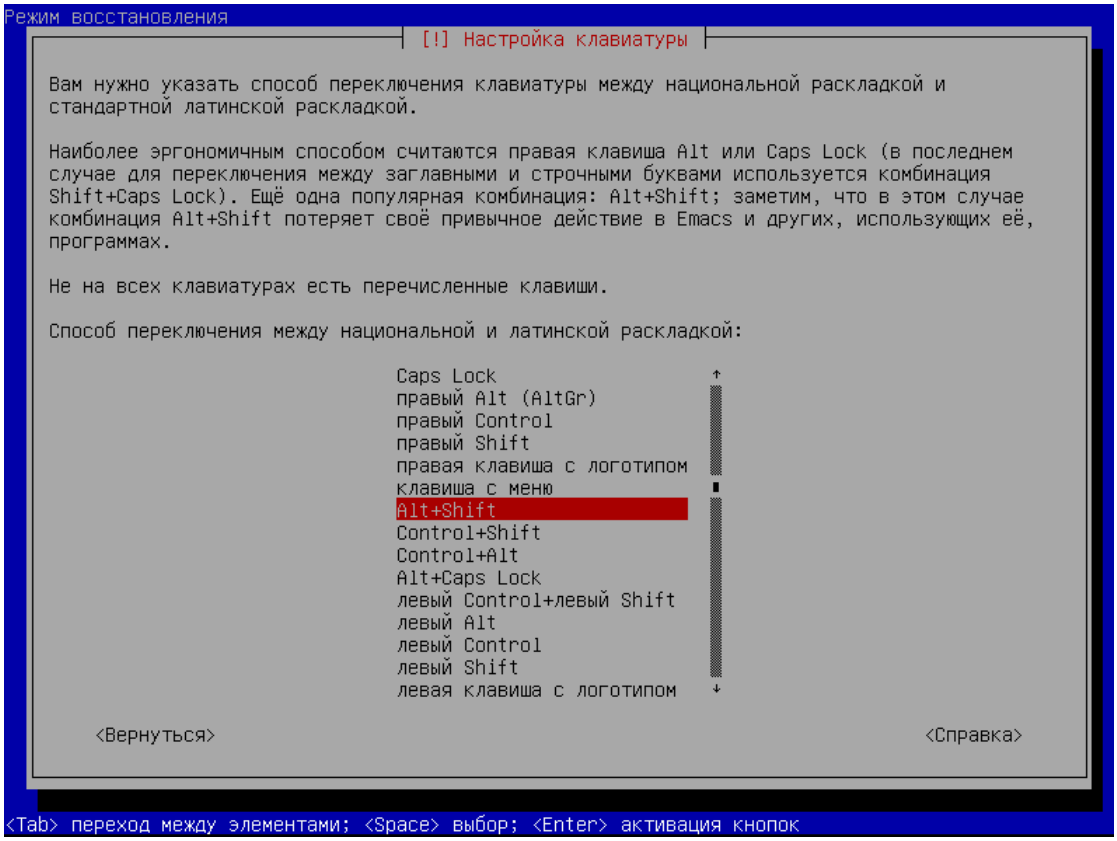
1. Загрузимся в режиме восстановления. Для этого подключим к компьютеру установочный диск и выберем пункт «Режим восстановления»



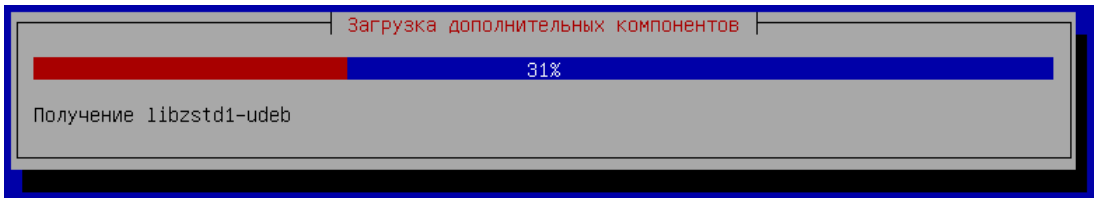
2. Согласимся с принятием лицензии.



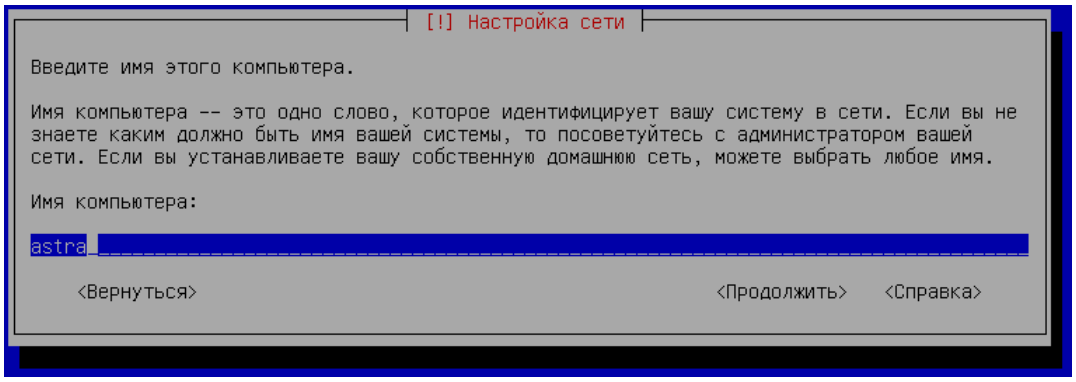
3. Выберем настройки клавиатуры.



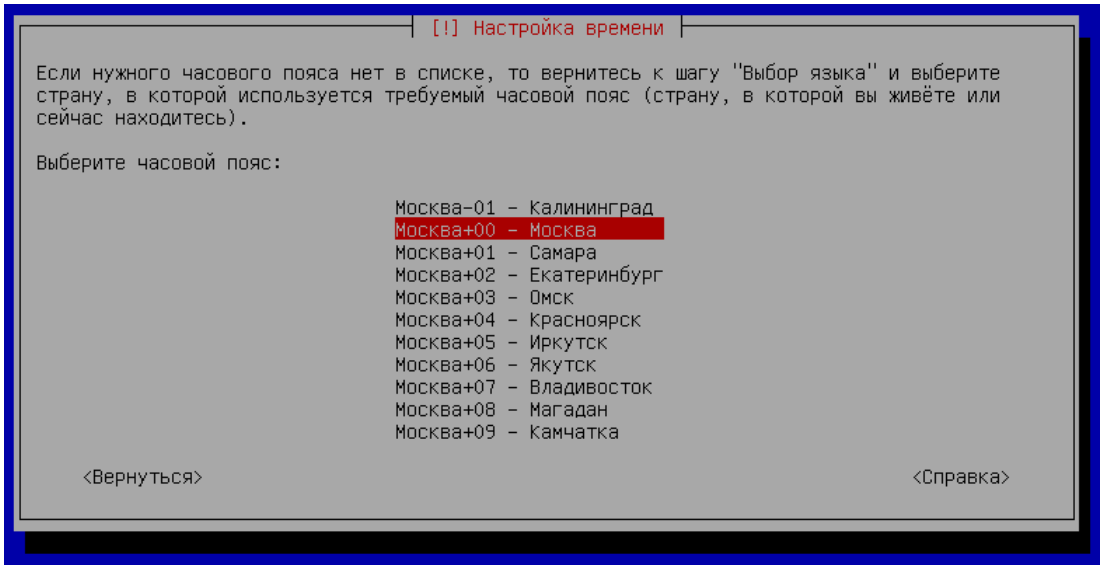
4. Необходимо подождать некоторое время, пока прогрузятся необходимые компоненты.



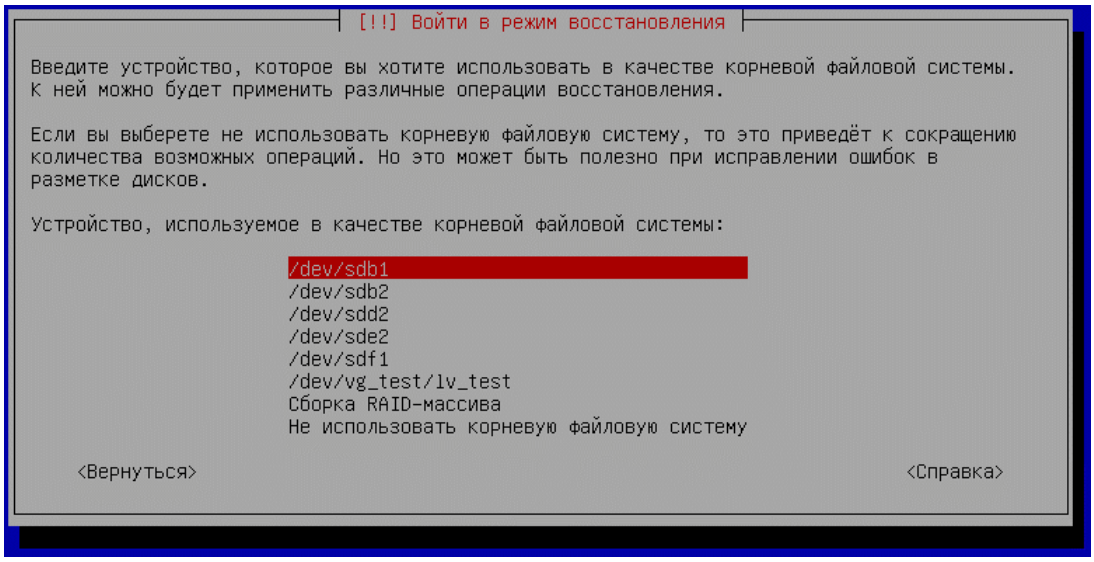
5. Далее настроим сеть.



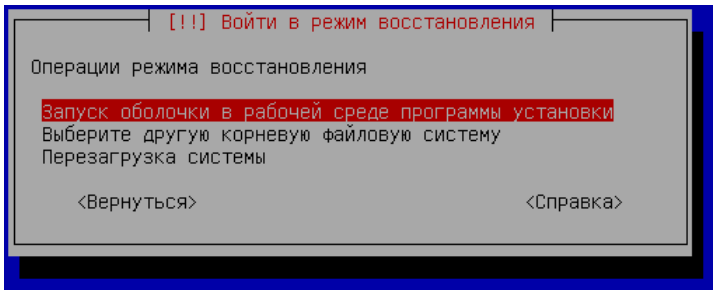
6. Настроим время.



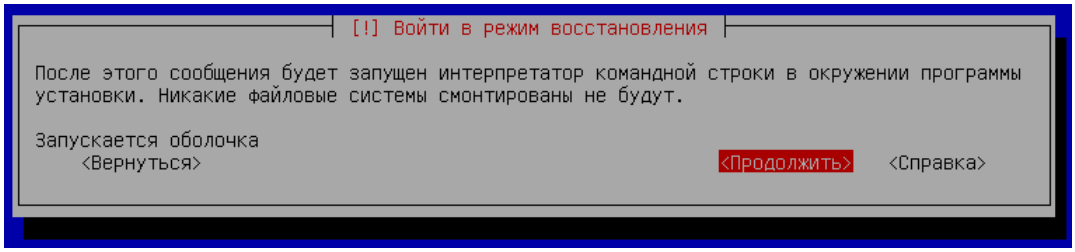
7. Запустим оболочку без монтирования ФС. В следующем диалоге необходимо выбрать пункт «Не использовать файловую систему», так как нам необходимо исправить ошибки разметки дисков.



8. И выберем «Запуск оболочки в рабочей среде программы установки».



9. Соглашаемся с предупреждением об отказе от монтирования файловых систем.



10. Примонтируем устройство с резервной копией:

    В нашем случае файл с резервной копией находится на жестком диске, подключенном к компьютеру, поэтому дополнительных действий не требуется. Если бы он находился на внешнем носителе информации, то его необходимо было бы подключить к компьютеру.

    В консоли введем команды для создания каталога `/mnt/data` и примонтируем в него диск:

```
~ # mkdir /mnt/data
~ # mount /dev/sdb2 /mnt/data
```

11. Восстановим таблицу разделов из резервной копии:

    - Введем команду `fdisk /dev/sda`.
    - Выберем пункт «загрузить разметку из файла сценария sfdisk», введя `I`, укажем файл `/mnt/data/sda.bkp` и запишем изменения, введя `w`.

```
BusyBox v1.30.1 (Debian 1:1.30.1-4+ci202207012308+astra1) built-in shell (ash)
Enter 'help' for a list of built-in commands.

~ # mkdir /mnt/data
~ # mount /dev/sdb2 /mnt/data
~ # ls /mnt/data
lost+found  sda.bkp     test.txt
~ # fdisk /dev/sda

Welcome to fdisk (util-linux 2.33.1).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table.
Created a new DOS disklabel with disk identifier 0xba5e09b5.

Command (m for help): I

Enter script file name: /mnt/data/sda.bkp

Created a new DOS disklabel with disk identifier 0x8892b118.
Created a new partition 1 of type 'Linux' and of size 15 GiB.
Created a new partition 2 of type 'Extended' and of size 975 MiB.
Created a new partition 5 of type 'Linux swap / Solaris' and of size 975 MiB.
Script successfully applied.

Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.
```

12. Примонтируем раздел `/dev/sda` и проверим его содержимое.

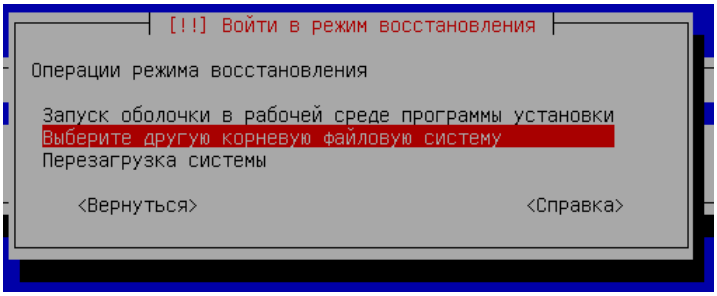
```
~ # fdisk -l /dev/sda
Disk /dev/sda: 16 GiB, 17179869184 bytes, 33554432 sectors
Disk model: VBOX HARDDISK
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x8892b118

Device     Boot    Start      End  Sectors  Size Id Type
/dev/sda1  *        2048 31553535 31551488   15G 83 Linux
/dev/sda2       31555582 33552383  1996802  975M  5 Extended
/dev/sda5       31555584 33552383  1996800  975M 82 Linux swap / Solaris
~ # mount /dev/sda1 /mnt/sda
~ # ls /mnt/sda
bin             initrd.img      libx32          parsec          sbin            usr
boot            initrd.img.old  lost+found      parsecfs        srv             var
dev             lib             media           proc            sys             vmlinuz
etc             lib32           mnt             root            tmp             vmlinuz.old
home            lib64           opt             run             users
```

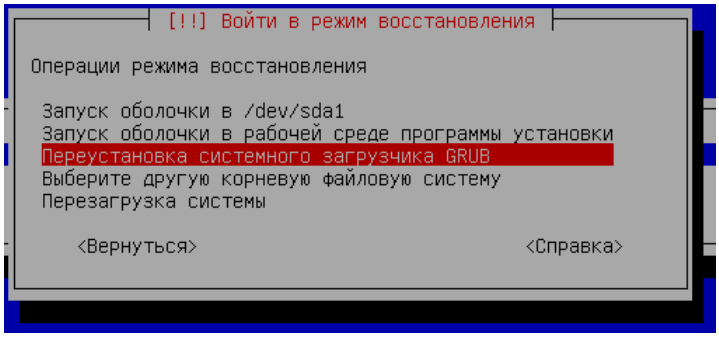
Команды для ввода:

```
~ # fdisk -l /dev/sda
~ # mkdir /mnt/sda
~ # mount /dev/sda1 /mnt/sda
~ # ls /mnt/sda
```

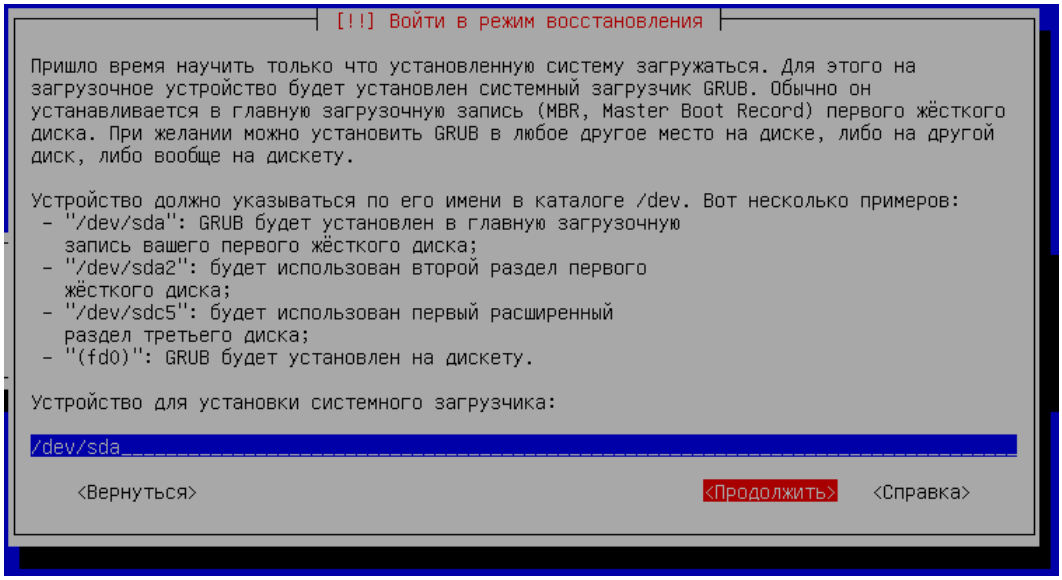
13. Запишем загрузочный сектор GRUB в MBR-запись, для этого выйдем из консоли командой `exit` и выберем другую ФС.



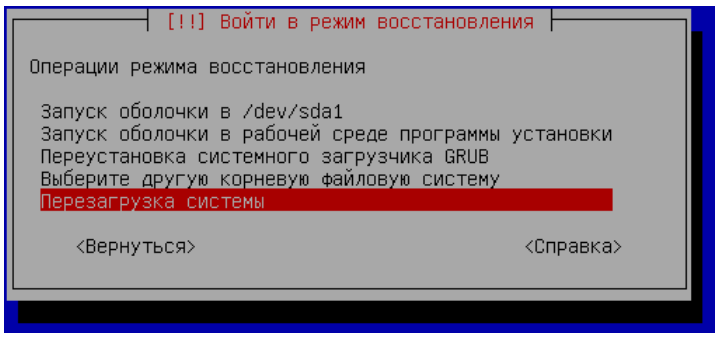
14. Выберем /dev/sda1 и затем пункт «Переустановка системного загрузчика GRUB».



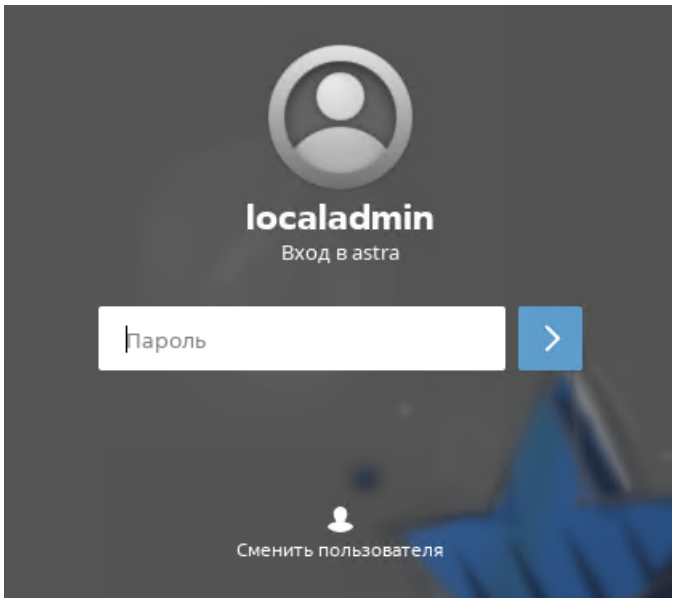
15. Переустановим его в главную загрузочную область устройства /dev/sda.



16. Перезагрузим компьютер.



17. Как видите, работа компьютера успешно восстановлена!



21.4.2.6 Восстановление загрузочной области диска из сохраненной копии

Первые шаги до 10-го пункта включительно аналогичны действиям из предыдущего раздела по восстановлению таблицы разделов из резервной копии. Поэтому продолжим с 11го шага.

11. Восстановим начальную область диска из резервной копии командой:

```
root@astra:~# dd if=/mnt/data/sda_boot.dd.bkp of=/dev/sda
root@astra:~# fdisk -l /dev/sda
```

```
~ # dd if=/mnt/data/sda_boot.dd.bkp of=/dev/sda
4+0 records in
4+0 records out
~ # fdisk -l /dev/sda
Disk /dev/sda: 16 GiB, 17179869184 bytes, 33554432 sectors
Disk model: VBOX HARDDISK
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x8892b118

Device     Boot    Start      End  Sectors  Size Id Type
/dev/sda1  *        2048 31553535 31551488   15G 83 Linux
/dev/sda2       31555582 33552383  1996802  975M  5 Extended
/dev/sda5       31555584 33552383  1996800  975M 82 Linux swap / Solaris
~ #
```

Как видите, разделы снова нам видны. В этом случае нам не требуется отдельно переустанавливать загрузчик GRUB в MBR – он был восстановлен.

12. Перезагрузим компьютер `exit` и в пункте меню перезапустить.

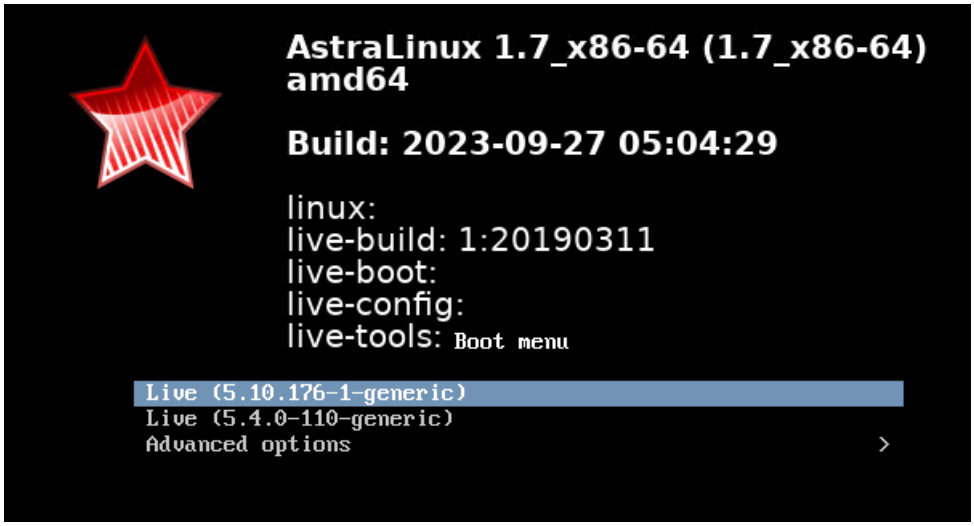
13. Как видите, работа компьютера снова успешно восстановлена!

21.4.2.7 Восстановление таблицы разделов с помощью утилиты testdisk

Для восстановления работоспособности этим способом:

1. Загрузим образ ALP-live с подготовленной ранее USB-флешки.

   - Подключим к системе созданную ранее загрузочную флешку.
   - Загрузим компьютер, выбрав один из доступных вариантов ядра.



- Войдем в систему с пустым паролем и откроем терминал горячим сочетанием клавиш Alt + t. Утилите testdisk необходимо минимум 24 строки для нормальной работы, поэтому откройте окно во весь экран.

2. Используя утилиту `testdisk`, восстановим таблицу разделов:

   Запустим утилиту `sudo testdisk /dev/sda`, где `/dev/sda` – устройство с поврежденной таблицей разделов, и подтвердим выбор устройства.

```
TestDisk 7.0, Data Recovery Utility, April 2015
Christophe GRENIER <grenier@cgsecurity.org>
http://www.cgsecurity.org

  TestDisk is free software, and
comes with ABSOLUTELY NO WARRANTY.

Select a media (use Arrow keys, then press Enter):
>Disk /dev/sda - 17 GB / 16 GiB - VBOX HARDDISK



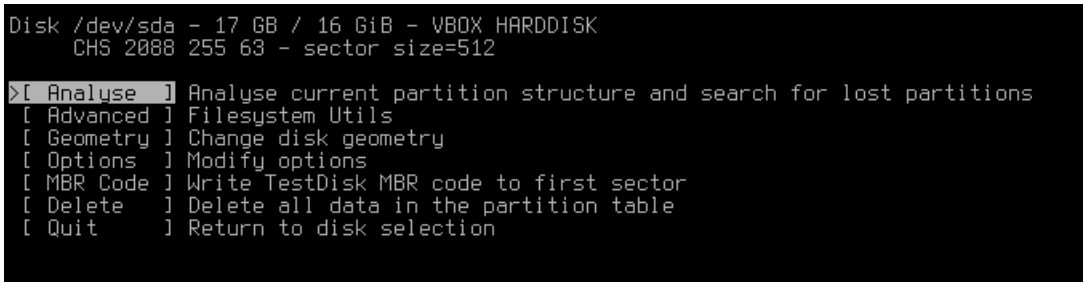
>[Proceed ]  [  Quit  ]

Note: Disk capacity must be correctly detected for a successful recovery.
If a disk listed above has incorrect size, check HD jumper settings, BIOS
detection, and install the latest OS patches and disk drivers.
```

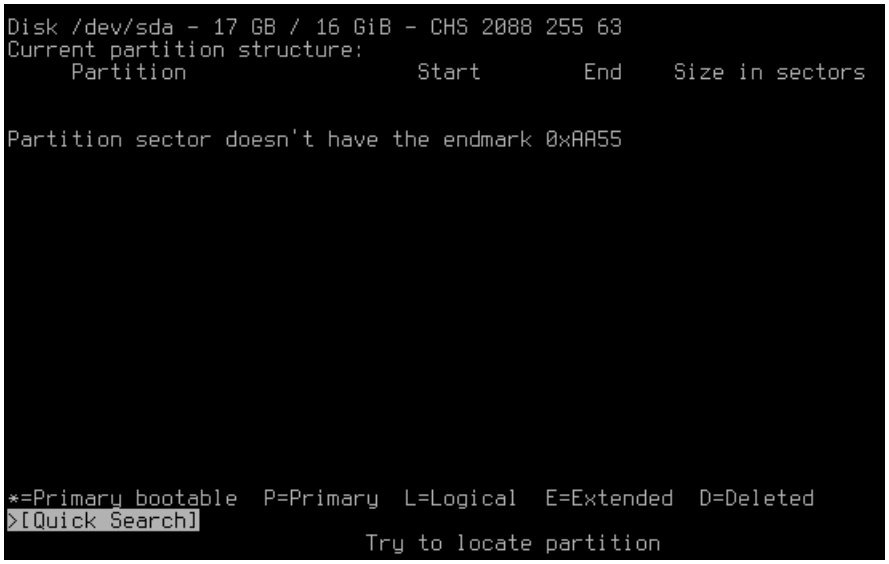
3. Выберем тип разделов. В нашем случае это Intel (MBR), однако это может быть и EFI GPT, если разметка у диска была GPT.



4. Далее выберем первый пункт – Analyse.



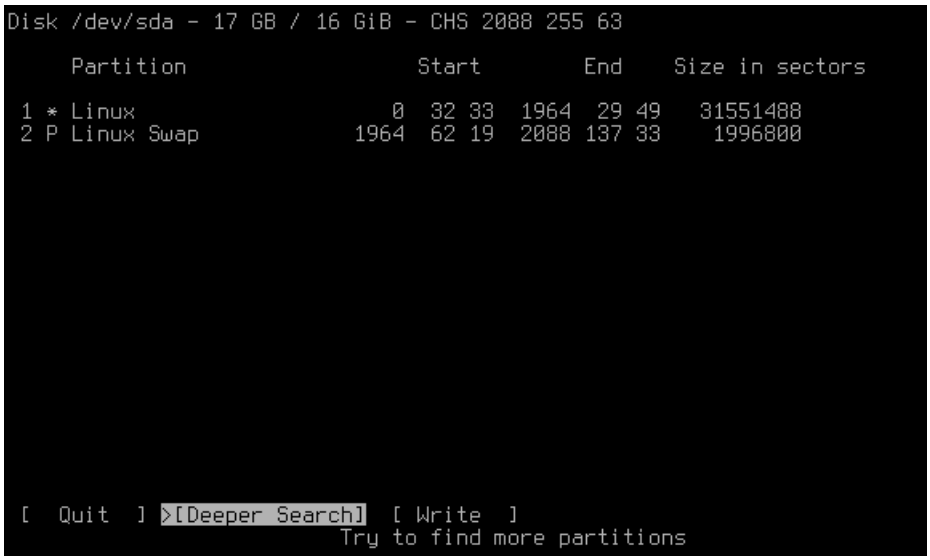
5. Подтвердим «быстрый поиск» (*англ. Quick Search*).



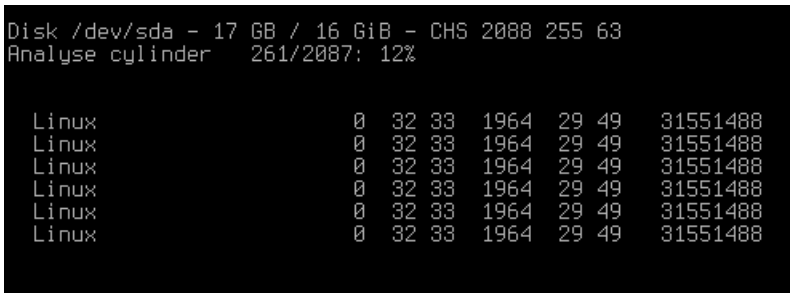
6. Ознакомимся с результатами поиска.



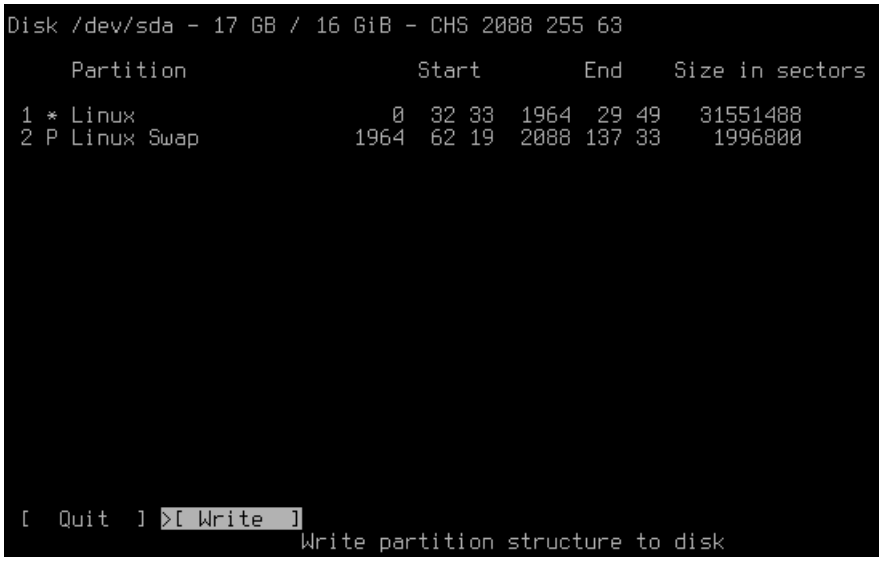
7. При необходимости воспользуемся «глубоким поиском».



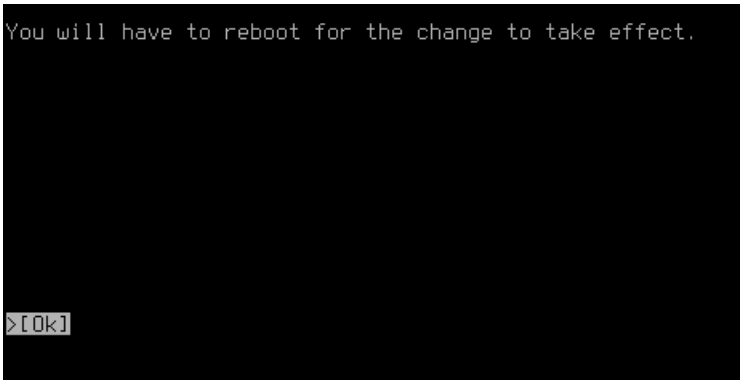
Процесс займет какое-то время:



8. После того, как мы убедились, что разделы были найдены, можно записать таблицу разделов на диск командой «Write». Подтвердим запись с помощью «Y».



9. Перезагружать систему не будем, выйдем из утилиты сочетанием Ctrl + c.



10. Проверим разделы на /dev/sda.

```
astra-live@Live-Astra:~$ sudo fdisk -l /dev/sda
Disk /dev/sda: 16 GiB, 17179869184 bytes, 33554432 sectors
Disk model: VBOX HARDDISK
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x00000000

Device     Boot    Start      End  Sectors  Size Id Type
/dev/sda1  *        2048 31553535 31551488   15G 83 Linux
/dev/sda2       31555584 33552383  1996800  975M 82 Linux swap / Solaris
```

Как видите, утилита не совсем верно восстановила разделы, так как их было три, а стало два. Раздел 2 был расширенным и включал раздел подкачки. Однако это не должно помешать загрузке с раздела /dev/sda1. Осталось восстановить загрузчик в главной загрузочной области.

11. Восстановим загрузчик в главной загрузочной области.

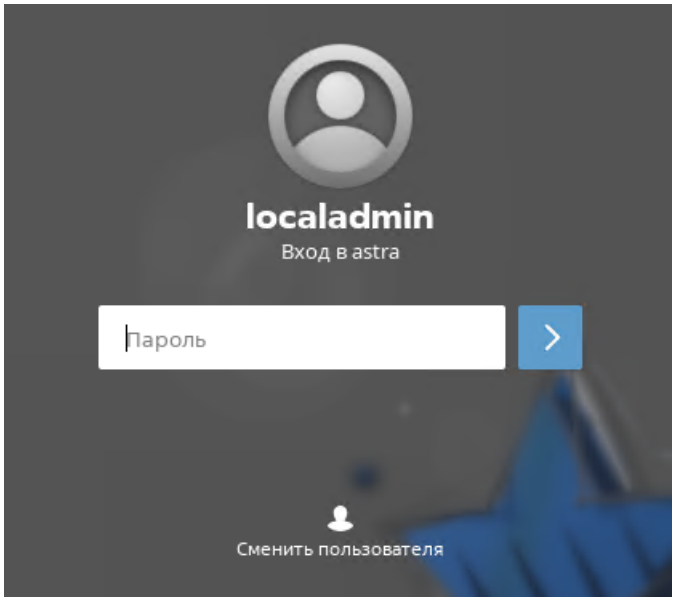
    Для восстановления загрузчика необходимо выполнить два действия. Во-первых, перемонтировать раздел с загрузчиком, а затем установить загрузчик, указав корень ФС и устройство:

```
astra-live@Live-Astra:~$ sudo grub-install --root-directory=/mnt/ dev/sda
```

```
astra-live@Live-Astra:~$ sudo mount /dev/sda1 /mnt
astra-live@Live-Astra:~$ sudo grub-install --root-directory=/mnt/ /dev/sda
Выполняется установка для платформы i386-pc.
Установка завершена. Ошибок нет.
astra-live@Live-Astra:~$
```

12. Извлечем загрузочную флешку и перезагрузим компьютер.

13. И снова удача! Говорят, что удача, как женщина: если за ней не бегать, то она обидится и сама придёт, чтобы все высказать.



21.4.2.8 Повреждение файлов загрузчика

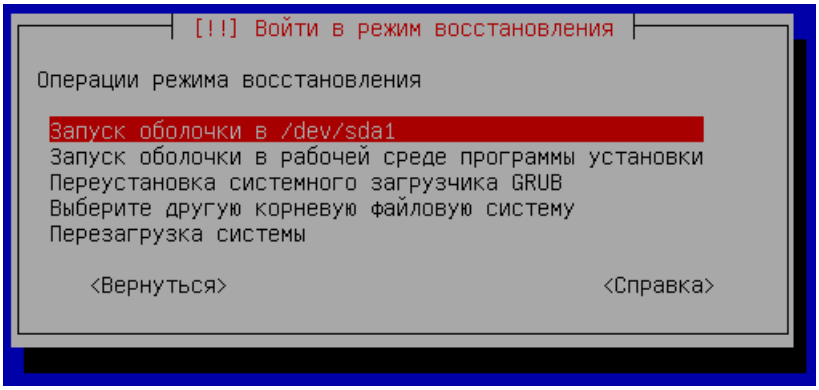
При повреждении файлов загрузчика GRUB или файлов из каталога `/boot` для восстановления работоспособности можно воспользоваться одной из следующих процедур.

21.4.3 Восстановление GRUB

Для восстановления загрузчика необходимо:

1. Загрузиться в режиме восстановления с установочного диска.

2. Запустить оболочку в разделе установки системы.



3. Перейти в консоль tty, например, горячим сочетанием клавиш:

Ctrl + Alt + F2

4. Смонтировать корневую ФС в /target, выполнив команду:

```
~ # chroot /target
```

5. Используя пакетный менеджер, проверить версию GRUB: grup-pc (Legacy) или grub-efi-amd64 (EFI). При необходимости заменить пакет на подходящий.

```
sh-5.0# apt list | grep grub
```

```
~ # chroot /target
sh-5.0# apt list | grep grub

WARNING: apt does not have a stable CLI interface. Use with caution in scripts.

fly-admin-grub2/stable,stable,now 1.0.2 all [installed]
grub-common/stable,stable,now 2.06-3~deb10u3+ci202304051618+astra4 amd64 [installed]
grub-coreboot-bin/stable 2.06-3~deb10u3+ci202304051618+astra4 amd64
grub-coreboot/stable 2.06-3~deb10u3+ci202304051618+astra4 amd64
grub-disk/stable 0.97-75 amd64
grub-doc/stable 0.97-75 all
grub-efi-amd64-bin/stable,stable 2.06-3~deb10u3+ci202304051618+astra4 amd64
grub-efi-amd64/stable,stable 2.06-3~deb10u3+ci202304051618+astra4 amd64
grub-efi-ia32-bin/stable,stable 2.06-3~deb10u3+ci202304051618+astra4 amd64
grub-efi-ia32/stable,stable 2.06-3~deb10u3+ci202304051618+astra4 amd64
grub-efi/stable,stable 2.06-3~deb10u3+ci202304051618+astra4 amd64
grub-emu/stable 2.06-3~deb10u3+ci202304051618+astra4 amd64
grub-firmware-qemu/stable 2.06-3~deb10u3+ci202304051618+astra4 amd64
grub-ieee1275-bin/stable 2.06-3~deb10u3+ci202304051618+astra4 amd64
grub-ieee1275/stable 2.06-3~deb10u3+ci202304051618+astra4 amd64
grub-legacy-doc/stable 0.97-75 all
grub-legacy/stable 0.97-75 amd64
grub-linuxbios/stable 2.06-3~deb10u3+ci202304051618+astra4 amd64
grub-pc-bin/stable,stable,now 2.06-3~deb10u3+ci202304051618+astra4 amd64 [installed,automatic]
grub-pc/stable,stable,now 2.06-3~deb10u3+ci202304051618+astra4 amd64 [installed]
grub-rescue-pc/stable 2.06-3~deb10u3+ci202304051618+astra4 amd64
grub-theme-starfield/stable 2.06-3~deb10u3+ci202304051618+astra4 amd64
grub-xen-bin/stable 2.06-3~deb10u3+ci202304051618+astra4 amd64
grub-xen-host/stable 2.06-3~deb10u3+ci202304051618+astra4 amd64
grub-xen/stable 2.06-3~deb10u3+ci202304051618+astra4 amd64
grub2-common/stable,stable,now 2.06-3~deb10u3+ci202304051618+astra4 amd64 [installed,automatic]
grub2/stable 2.06-3~deb10u3+ci202304051618+astra4 amd64
kcm-grub2/stable,stable,now 0.6.4+git20180511.699391e-0astra8 amd64 [installed,automatic]
```

6. Переустановить GRUB командой:

```
sh-5.0# grub-install /dev/sda
```

```
sh-5.0# grub-install /dev/sda
Installing for i386-pc platform.
Installation finished. No error reported.
```

7. Пересоздать конфигурационный файл GRUB командой:

```
sh-5.0# update-grub
```

```
sh-5.0# update-grub
Generating grub configuration file ...
Found background image: /usr/share/images/desktop-base/desktop-grub.png
Found linux image: /boot/vmlinuz-5.15.0-70-generic
Found initrd image: /boot/initrd.img-5.15.0-70-generic
Warning: os-prober will be executed to detect other bootable partitions.
Its output will be used to detect bootable binaries on them and create new boot entries.
done
```

8. Перезагрузить компьютер. Работоспособность GRUB восстановлена!



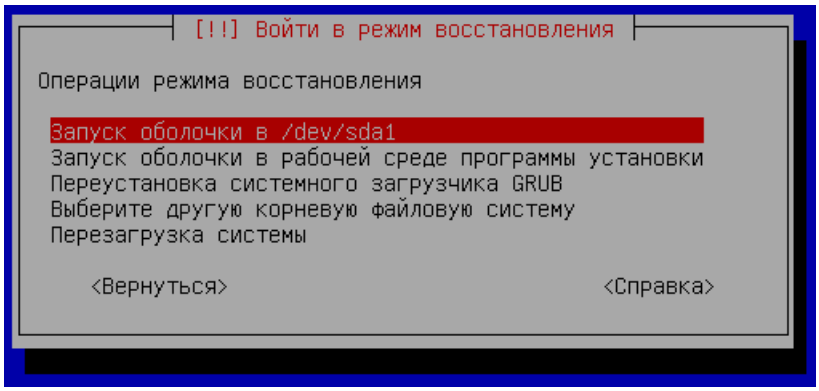
С дополнительными рекомендациями по замене таблицы разделов с MBR на GPT можно ознакомиться в статье Установка загрузчика GRUB2 в UEFI GPT

21.4.4 Пересоздание образа initrd

При повреждении образа initrd его можно легко восстановить. Для этого необходимо:

1. Загрузиться в режиме восстановления с установочного диска.

2. Запустить оболочку в разделе установки системы.



3. В консоли выполнить команду:

```
root@astra:/# update-initramfs -u
```

```
root@astra:/# update-initramfs -u
update-initramfs: Generating /boot/initrd.img-5.15.0-70-generic
root@astra:/#
```

Если необходимо пересоздать образ для определенной версии ядра, то нужно добавить ключ `-k` и указать версию, например:

```
root@astra:/# update-initramfs -u -k 5.15.0-70-generic
```

4. Перезагрузить компьютер. Работоспособность восстановлена!

Если для пересоздания образа Initrd в каталоге `/boot` не хватает места, то вы можете загрузиться в режиме восстановления и выполнить следующие действия:

- Удалить неиспользуемые данные, например, старые версии ядра или версии

```
localadmin@astra:~$ ls /boot
config-5.15.0-33-generic initrd.img-5.15.0-33-generic System.map-5.15.0-33-generic
config-5.15.0-70-generic initrd.img-5.15.0-70-generic System.map-5.15.0-70-generic
grub old-vmlinuz-5.15.0-33-generic vmlinuz-5.15.0-70-generic
```

- Смонтировать временную ФС в `/boot`, пересоздать образ Initrd и заменить его в оригинальном каталоге `/boot`

  1. Смонтируйте временную ФС `mount -t tmpfs none /boot`.
  2. Создайте образ Initrd для требуемой версии ядра, например:

     ```
     root@astra:~# update-initramfs -u -k 5.15.0-70-generic
     ```

  3. Сохраните созданный образ (при размонтировании временной ФС все её содержимое будет удалено).

     ```
     root@astra:~# cp -r /boot/initrd.img-5.15.0-70-generic /home
     ```

  4. Размонтируйте временную ФС.

     ```
     root@astra:~# umount /boot
     ```

  5. Замените старый образ на новый.

     ```
     root@astra:~# cp -r /home/initrd.img-5.15.0-70-generic /boot/
     ```

  6. Обновите конфигурацию загрузчика GRUB

     ```
     root@astra:~# update-grub
     ```

  7. Перезагрузите систему.

21.4.4.1 Проверка файловой системы на ошибки

Некоторые ошибки в работе диска можно исправить с помощью следующих утилит:

- **fsck** – утилита для проверки и восстановления файловой системы;
- **badblocks** – утилита для выявления сбойных блоков.

21.4.5 Утилита badblocks

Использование утилиты badblocks может выполняться в разных режимах:

- Режим по умолчанию. В этом режиме утилита проверяет чтение/запись блоков данных на устройстве без повреждения текущих данных. Для использования этого режима файловая система должна быть размонтирована

```
localadmin@astra:~$ sudo badblocks -v /dev/sda -o ~/bad_sectors.txt
Checking blocks 0 to 16777215
Checking for bad blocks (read-only test):
done
Pass completed, 0 bad blocks found. (0/0/0 errors)
```

- Если мы хотим выполнить проверку на смонтированной ФС, это возможно только в режиме чтения блоков, например:

```
localadmin@astra:~$ sudo badblocks -vn /dev/sdb2 -o ~/bad_sectors.txt
Checking for bad blocks in non-destructive read-write mode
From block 0 to 301055
Testing with random pattern: Pass completed, 0 bad blocks found. (0/0/0 errors)
```

- Если необходимо выполнить проверку с реальной записью данных на диск, то необходимо использовать ключ `-w`.

Предупреждение: Использование ключа `-w` приведет к уничтожению имеющихся данных на устройстве – будьте осторожны!

Результаты утилиты **badblocks** могут быть переданы утилите **fsck**, чтобы сообщить файловой системе о сбойных блоках и исключить их из использования, например, командой:

```
localadmin@astra:~$ sudo fsck -l ~/bad_sectors.txt /dev/sdb2
```

С другими опциями вы можете ознакомиться в справке `man badblocks`.

21.4.6 Утилита fsck

Для проверки диска утилитой fsck это устройство не должно быть смонтировано.

```
localadmin@astra:~$ sudo fsck /dev/sdb2
fsck из util-linux 2.33.1
e2fsck 1.44.5 (15-Dec-2018)
/dev/sdb2: clean, 14/75480 files, 15696/301056 blocks
```

Пример использования для смонтированного устройства:

```
localadmin@astra:~$ sudo fsck /dev/sdb1
fsck из util-linux 2.33.1  e2fsck 1.44.5 (15-Dec-2018)
Logs: clean, 11/128016 files, 26666/512000 blocks
localadmin@astra:~$ sudo mount /dev/sdb1
```

С другими опциями вы можете ознакомиться в справке `man fsck`.

21.4.7 Неисправности, возникающие на финальных стадиях загрузки

После загрузки системы проблем может быть тоже предостаточно, но вспомним анекдот и рассмотрим те, что связаны с паролями:

> Сидят два админа на работе, грустят, заходит третий и спрашивает:
>
> —Че такие грустные?
>
> — Да вчера пиво пили... и пароли меняли...

21.4.7.1 Обнуление счётчика неудачных попыток ввода пароля

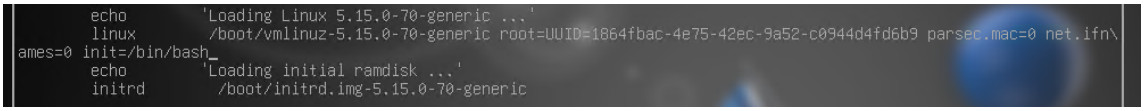
Если было превышено максимальное количество попыток ввода пароля, но пароль не утерян, то можно обнулить этот счетчик, но вам потребуется пароль от GRUB. Для этого необходимо:

1. Перезагрузить компьютер и при появлении меню загрузчика войти в режим редактирования нажатием клавиши e.
2. Вывести имя пользователя и пароль от GRUB.



3. В интерфейсе редактирования изменить строку, начинающуюся со слова linux:
   - удалить «ro» и «quite».

- добавить в конец строки `init=/bin/bash/`.



4. Выполнить загрузку нажатием клавиши: F10. Дождаться загрузки и появления приглашения командной строки.

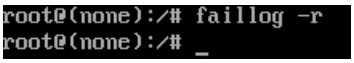
5. В случае, если ФС доступна только на чтение, выполнить ее перемонтирование командой:

```
root@(none):/# mount -o rw,remount /
```



6. Сбросить счетчик неудачных попыток ввода пароля командой:

```
root@(none):/# faillog -r
```



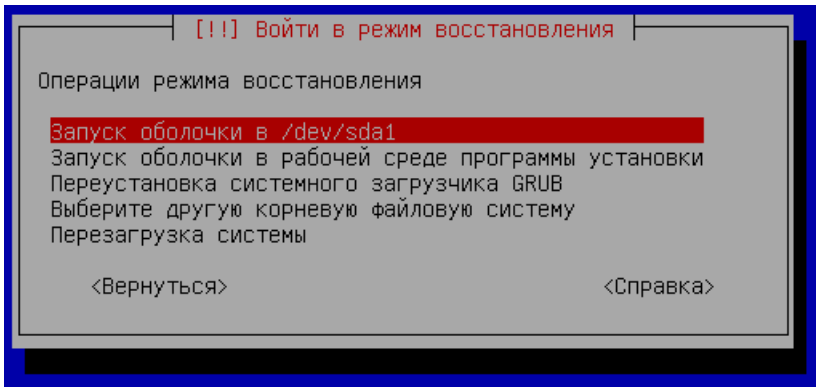
7. Перезагрузить компьютер и войти в систему под своими учетными данными.

21.4.7.2 Сброс пароля учетной записи администратора

Все мы помним, что обычному пользователю пароль можно сбросить командой **passwd**, но что делать, если вам требуется восстановить пароль от администратора? Для этого нужно:

1. Загрузиться в режиме восстановления с установочного диска.

2. Запустить оболочку в разделе установки системы (например, `/dev/sda1`).



3. Перейти в консоль tty, например, горячим сочетанием клавиш:

Ctrl + Alt + F2

4. Смонтировать корневую ФС в /target, выполнив команду:

```
~ # chroot /target
```

5. С помощью текстового редактора удалить хеш пароля в файлах `/etc/shadow`, `/etc/shadow-`.

6. Перезагрузить машину и выполнить вход без пароля с последующей его установкой.

21.5 Практика и тестирование

Пришло время проверить себя пройти материал на практике и ответить на тестовые вопросы

21.5.1 Практическая работа: Модуль 21. Работа с журналами и отладка системы

См. также *Требования, правила и цели выполнения практической работы*

21.5.1.1 Практические задания

Задание 1.

Просмотрите все логи командой less в папке `/var/log`, чтобы понять какие данные записываются

Задание 2.

- Просмотрите логи wtmp
- Просмотрите логи last
- Просмотрите журнал загрузки

Задание 3.

- Запустим просмотр журнала в реальном времени в первом окне
- Отправим событие для теста в другом окне терминала через `logger` приоритет kern.error
- Попробуйте разные источники и уровни важности, см. man logger или curl cht.sh/logger

Задание 4.

Проверьте, какие процессы используют файлы в директории `/var/log`

Задание 5.

Откройте графическую утилиту журнала и с помощью фильтра найдите ваши сообщения.

Задание 6.

Соберите информацию о процессах, используя команды из лекции

Задание 7.

Узнайте свободное место и занятые Inode

Задание 8.

Соберите информацию о сетевых соединениях

Задание 9.

- Получите информацию о настройках межсетевого экрана
- Разрешите ssh прежде чем включать firewall

Задание 10.

- Сделайте дамп сетевых соединений
- Изучите основные часто используемые ключи tcpdump через cht.sh

Задание 11.

- Сделайте трассировку bash
- Сделайте трассировку logger чтобы выяснить какой сокет открывает logger
- Изучите основные часто используемые ключи strace через cht.sh:
- Просмотрите список библиотек logger

Задание 12.

Восстановление работы загрузчика с загрузочного диска

21.5.1.2 Практическая работа (с ответами)

Задание 1.

Просмотрите все логи командой less в папке `/var/log`, чтобы понять какие данные записываются:

Основной журнал, в котором хранятся все сообщения, кроме сообщений об аутентификации и сообщений почтовой службы.

```
$ less /var/log/syslog
```

Сокращенная версия файла syslog, сюда записываются несколько типов сообщений: почта, cron, сервисы systemd, сообщения от ядра, pam, аутентификация и др.:abbr:

```
$ sudo less /var/log/messages
```

Сообщения об аутентификации служб и пользователей:

```
$ sudo less /var/log/auth.log
```

Сообщения, сгенерированные во время загрузки ОС:

```
$ sudo less /var/log/boot.log
```

Сообщения от ядра ОС(*):

```
$ sudo less /var/log/kern.log
```

Сообщения от различных служб (кроме почты, печати и планировщика cron).

```
$ sudo less /var/log/daemon.log
```

Сообщения от команд, выполняемых пользователями, например: (NetworkManager):

```
$ sudo less /var/log/user.log
```

Задание 2.

- Просмотрите `/var/log/wtmp` командой `who`
- Просмотрите `/var/log/last` командой `last`
- Просмотрите журнал загрузки командой `sudo dmesg`

Задание 3.

Запустим просмотр журнала в реальном времени в первом окне:

```
$ sudo journalctl -f -o verbose
```

Отправим событие для теста в другом окне терминала:

```
$ logger --priority kern.error Test error
```

Попробуйте разные источники и уровни важности, см. `man logger` или `curl cht.sh/logger`

Задание 4.

Проверьте, какие процессы используют файлы в директории /var/log

```
$ sudo lsof +d /var/log
$ sudo watch -n 1 lsof +d /var/log
```

Задание 5.

Откройте графическую утилиту журнала и с помощью фильтра найдите ваши сообщения.

: *Пуск → Панель управления → Безопасность → Журнал аудита*

Задание 6.

Соберите информацию о процессах, используя команды:

`cat /proc/1/status`

```
$ ps
$ ps aux
$ sudo ps aux
$ top
$ htop
$ vmstat
$ uptime
$ mpstat
$ pidstat 2 5
```

Задание 7.

Узнайте свободное место и занятые Inode

```
$ df -h
$ df -i
$ sudo iostat
$ sudo iotop
```

Задание 8.

Соберите информацию о сетевых соединениях:

```
$ ss
$ curl cht.sh/ss
$ ss -tlp
$ sudo ss -tlp
$ sudo lsof -i4
$ ip a
$ ip r
```

Задание 9.

Получите информацию о настройках межсетевого экрана:

```
$ sudo ufw
$ sudo ufw status
```

Разрешите ssh прежде чем включать firewall

```
$ sudo ufw allow openssh
$ sudo ufw enable
$ sudo ufw status
$ curl cht.sh/ufw
```

Задание 10.

Сделайте дамп сетевых соединений

```
$ tcpdump
$ sudo tcpdump
$ sudo tcpdump -D
```

Изучите основные часто используемые ключи `tcpdump` через cht.sh:

`curl cht.sh/tcpdump`

Задание 11.

Сделайте трассировку bash: `strace bash -c "echo test"`

Сделайте трассировку logger чтобы выяснить какой сокет открывает logger

```
$ strace logger test
$ strace logger test | grep /dev/log
```

Изучите основные часто используемые ключи strace через cht.sh:

`curl cht.sh/strace`

Просмотрите список библиотек logger:

`ldd "$(which logger)"`

Задание 12.

Восстановление работы загрузчика

1. создайте копию в dd

```
$ sudo dd if=/dev/sda of=/mnt/data/sda_boot.bkp bs=512 count=4
```

2. затрите диск первые 512 байт

```
$ sudo dd if=/dev/zero of=/dev/sda bs=512 count=1
$ sudo reboot
```

3. Загрузитесь установочного с диска и зайдите в busybox

4. При Монтируем наш второй диск

```
$ mount /dev/sdb /mnt
```

5. Восстановим наш загрузчик

```
$ dd if=/mnt/sda_boot.bkp of=/dev/sda
```

6. Перезагрузим виртуальную машину

** Дополнительно попробуйте восстановить все неработающие состояния из текста лекции

21.5.2 Тест: Модуль 21. Работа с журналами и отладка системы

Пожалуйста, ответьте на несколько вопросов, чтобы закрепить полученные знания. Правильные ответы для самопроверки вы сможете найти в конце списка. Обратите внимание, что у некоторых вопросов может быть несколько верных вариантов ответов.

21.5.2.1 Вопросы

Вопрос 1. Как называется протокол системного журнала в Linux-системах?

a. varlog

b. statlog

c. logged

d. syslog

Вопрос 2. Какие субъекты событий верные?

a. user10

b. emerg

c. kern

d. auth

Вопрос 3. Сколько уровней важности в syslog?

a. 5

b. 8

c. 25

d. 4

Вопрос 4. Какие службы журналирования используются в ALSE x.7 по умолчанию?

a. sysklogd

b. syslog-ng

c. rsyslog

d. journald

Вопрос 5. Какой программой ротируются логи?

a. logrotate

b. rotlogs

c. cronrotate

d. syslog-ng

Вопрос 6. Какие существуют файлы логов в директории `/var/log`?

a. syslog

b. boot.log

c. messages.log

d. zip.log.gz

Вопрос 7. Какая команда позволяет посмотреть последние входы и перезагрузки?

a. Команда `past`

b. Команда `second`

c. Команда `last`

d. Команда `dd`

Вопрос 8. Какой файл содержит журнал загрузки?

a. Файл `messages`

b. Файл `boot.log`

c. Файл `user.log`

d. Файл `dpkg.log`

Вопрос 9. Какой формат сообщений содержит UTF-8 кодировку?

a. RFC 3164

b. RFC 5424

c. Оба верные

Вопрос 10. Какой параметр нужно настроить, чтобы journald сохранял журнал?

a. Параметр `Storage=persistent`

b. Параметр `Storage=auto`

c. Параметр `Storage=/var/log/journald`

d. Параметр `Storage=/dev/sdb`

Вопрос 11. Какая команда позволяет следить за журналом в реальном времени?

a. Команда `sudo journalctl -p`

b. Команда `sudo journalctl -f`

c. Команда `sudo journalctl -N`

d. Команда `sudo journalctl -o`

Вопрос 12. Какая команда позволяет отправить сообщение в службу журналирования?

a. Команда `syslog-ng -s`

b. Команда `systemd-cat`

c. Команда `logger`

d. Команда `journald`

Вопрос 13. Какая служба запускает ротацию логов ежедневно?

a. cron

b. systemd-journald

c. ss

d. logger

Вопрос 14. Какой инструмент системы Astra Linux создает архив логов и конфигураций?

a. astra-debug-logs

b. astra-create-debug-logs

c. astra-create-logs

d. astra-logs

Вопрос 15. Какая команда позволяет проверить среднюю загрузку процессора?

a. Команда `uptime`

b. Команда `top`

c. Команда `htop`

d. Все варианты

Вопрос 16. Какая команда позволяет снять дамп сетевого трафика?

a. Команда `netstat`

b. Команда `ss`

c. Команда `tcpdump`

d. Команда `iostat`

Вопрос 17. Какая команда позволяет создать резервную копию разделов диска?

a. Команда `du`

b. Команда `df`

c. Команда `dd`

d. Команда `ncdu`

Ответы на вопросы

1. d
2. b, c, d
3. b
4. b, d
5. a
6. a, b
7. c
8. b
9. b
10. a
11. b
12. b, c
13. a
14. b
15. d
16. c
17. c

21.6 Заключение

Мы заканчиваем курс молодого бойца по общим вопросам администрирования Linux, так как считаем такой объем уже достаточным для освоения доменных технологий.

Далее мы перейдем к изучению централизованной системы управления ИТ-инфраструктурой ALD Pro, рассмотрим ее устройство и принцип работы отдельных компонентов технологического стека, основными из которых являются: служба каталога FreeIPA, агентская часть доменного компьютера SSSD, групповые политики на базе системы управления конфигурациями SaltStack.

21.7 Дополнительные источники информации

- Журналы работы системных служб
- LiveCD от gparted
- Создание собственных ALP-live образов (Live-CD/Live-USB) в Astra Linux
- Создание загрузочных носителей USB flash drive

21.8 Обратная связь

Если остались вопросы, то их всегда можно задать в специальной теме.

Часть 2. ALD Pro

Вторая часть курса посвящена вопросам администрирования домена ALD Pro на базе службы каталога FreeIPA, но мы затронем и другие подсистемы продукта, такие как сервер репозиториев (Reprepro), файловый сервер (Samba), принт-сервер (CUPS), сервер аудита (syslog-ng) и сервер мониторинга (Zabbix).

Стоит заметить, что служба каталога – это крайне непростая система, состоящая из множества компонентов, использующих разные технологии и протоколы, но мы постараемся охватить все основные моменты, чтобы вы чувствовали себя уверенно при администрировании домена.

Обращаем также ваше внимание, что для успешного освоения этого материала следует обязательно изучить *Часть 1. Astra Linux*.

Модуль 1. Система ALD Pro и ее компоненты

1.1 Введение

Программный комплекс ALD Pro – это в первую очередь служба каталога, которая позволит при переходе от Windows к Linux сохранить лучшие практики системного администрирования, доступные ранее при эксплуатации Active Directory.

Из этого модуля вы узнаете об истории развития служб каталога, архитектуре ALD Pro и рекомендациях по планированию ресурсов. «Буков» будет много, но будет интересно!

1.2 Система ALD Pro

Используя продукты ведущих вендоров, мы не всегда задумываемся о том, что в них реализованы лучшие бизнес-практики, позволяющие экономить на администрировании, повышать безопасность инфраструктуры и обеспечивать удобство для работы пользователей. Но эти детали, как шило в... мешке, не дают покоя, и при попытке заменить проверенные инструменты свободным ПО

оказывается, что ни одно из альтернативных решений неспособно без существенных доработок обеспечить сопоставимые показатели по безопасности, удобству и эффективности.

Сегодня нужно заменить большое количество программных продуктов, а времени совсем мало, поэтому приходится расставлять приоритеты и начинать с самого важного. Так, одним из стратегических направлений компании Астра стала разработка службы каталога, которая должна заменить Microsoft Active Directory и стать сердцем обновленной ИТ-инфраструктуры российских предприятий.

Система централизованного управления ИТ-инфраструктурой ALD Pro позволит при переходе от Windows к Linux сохранить лучшие практики системного администрирования, доступные ранее при эксплуатации Active Directory.

Эта служба каталога построена на базе FreeIPA и позволяет объединить Linux-компьютеры в один домен, обеспечивая единую точку входа, централизованное управление правами доступа, установку и настройку программного обеспечения — все то, без чего невозможно поддерживать инфраструктуру на том же уровне удобства и безопасности, сохраняя прежний размер команды системных администраторов.

Контроллеры домена ALD Pro обеспечивают те же сервисы, что и в Active Directory:

- Центр распространения ключей на базе MIT Kerberos обеспечивает возможность централизованной аутентификации по безопасному протоколу Kerberos V5.
- Система доменных имён на базе ISC Bind9 тесно интегрирована с доменом и обеспечивает взаимодействие клиентов друг с другом, контроллерами домена и другими сервисами.
- LDAP-каталог на базе 389 Directory Server обеспечивает хранение и репликацию информации между контроллерами домена. В LDAP каталоге хранятся учетные записи пользователей и компьютеров, DNS записи и другая информация.
- Механизм групповых политик обеспечивает возможность централизованной настройки доменных компьютеров с доставкой параметров через LDAP.
- Служба синхронизации времени на базе Chrony обеспечивает возможность синхронизации времени с контроллером домена по стандартному протоколу NTP, что крайне важно для корректной работы Kerberos-аутентификации.

Продукт ALD Pro (Astra Linux Directory Pro) быстро развивается, и с даты первого релиза было опубликовано уже несколько сотен улучшений, что позволило стать лидером на рынке доменных решений. У нас уже сотни инсталляций по всей стране, и с каждым днем их становится все больше.

1.2.1 Возможности службы каталога

Служба каталога (*англ. Directory Service*) — это в первую очередь **средство централизованного управления ИТ-ресурсами** предприятия, такими как пользователи и компьютеры. В ALD Pro на каждом контроллере домена установлен веб-портал, с помощью которого можно получить доступ к базе данных LDAP-каталога, чтобы просматривать список пользователей, редактировать их атрибуты, сбрасывать пароли, создавать новые учетные записи, см. рис. 2.1.

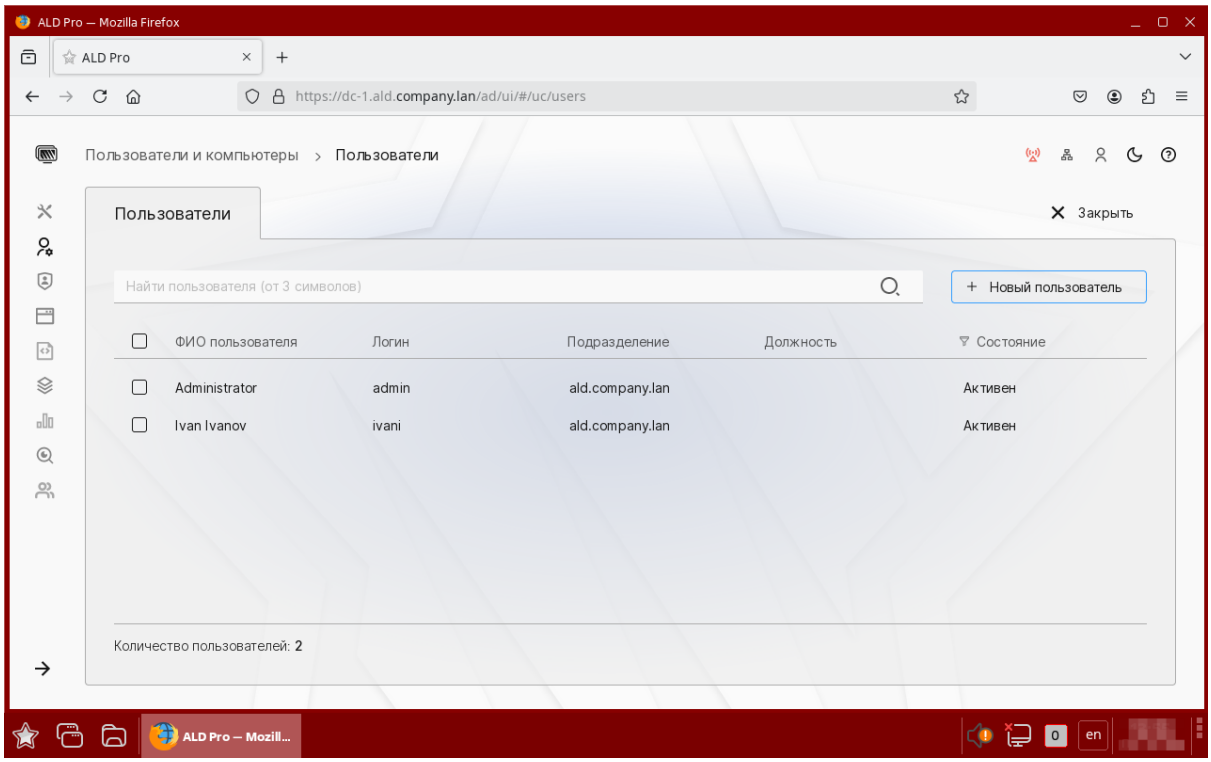


рис. 2.1 – Централизованное редактирование пользователей

Каталог содержит также учетные записи компьютеров, которые нужны для возможности организации единой точки входа с аутентификацией по безопасному протоколу Kerberos V5. Для того чтобы Linux-компьютер мог быть участником домена, на него устанавливается агентская часть ALD Pro, ключевым компонентом которой является служба SSSD. Эта служба встраивает свои модули в стандартные стеки PAM и NSS, за счет чего она может перехватывать запросы на аутентификацию и авторизацию и обрабатывать их через контроллеры домена.

Поиск доступного контроллера домена выполняется в рамках процедуры автоматического обнаружения доступных сервисов. Когда пользователь в первый раз входит в компьютер, его аутентификация возможна только через контроллер домена, но при получении успешного статуса PAM_SUCCESS служба SSSD сохранит хеш в локальной базе, что гарантирует возможность автономной аутентификации этого пользователя даже в условиях отсутствия связи с сервером. В случае строгих правил безопасности кэширование паролей можно отключить.

Еще одной сильной стороной службы каталога ALD Pro являются **групповые политики**, часть из которых реализована непосредственно в службе FreeIPA на связке с SSSD. Аббревиатура IPA вообще-то является сокращением от слов «Identity, Policy, Audit» и указывает на то, что эта служба является средством централизованного управления идентификацией, политиками и аудитом. Базовых политик FreeIPA немного, но все они крайне полезные – это политики паролей, правила HBAC и SUDO, политики SELinux и монтирования дисков.

Как вы знаете, пароли являются самым простым, но вместе с тем и не самым безопасным способом аутентификации пользователей. Поэтому в службе каталога реализованы политики паролей, которые позволяют снизить риск взлома привилегированных учетных записей за счет предъявления к ним повышенных требований, см. рис. 2.2.

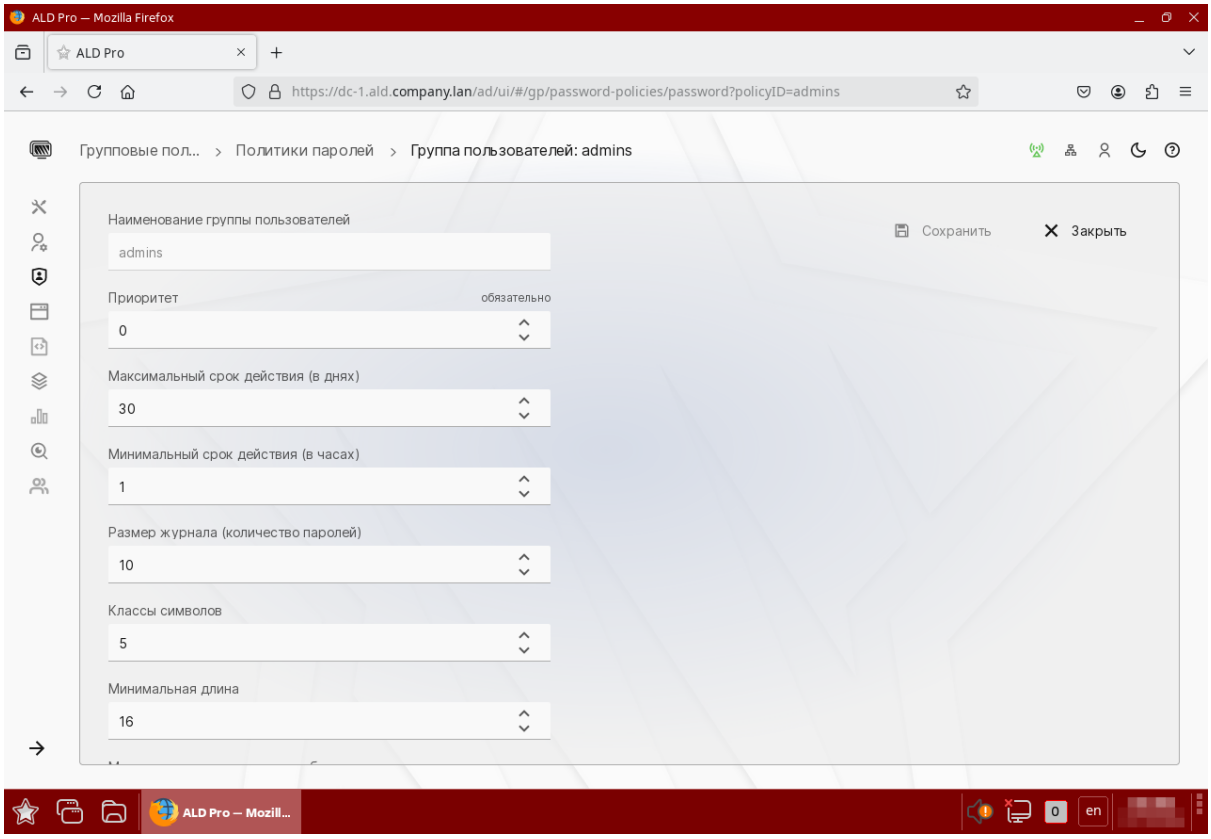


рис. 2.2 – Политика паролей для участников группы администраторов

Механизм политик паролей позволяет для каждой группы пользователей создать отдельный набор правил, которые поддерживают стандартные требования к паролям MIT Kerberos. Система будет сверяться с этим списком правил каждый раз, когда пользователь захочет установить себе новый пароль. В случае если пользователь входит сразу в несколько групп, будет применена самая строгая политика с наивысшим приоритетом. Если на пользователя не распространяется действие ни одной политики, ему будет назначена глобальная политика по умолчанию (global_policy).

Политики HBAC (англ. Host-based access control — контроль доступа к узлу) задают набор правил

для настройки доступа пользователей или групп пользователей к определенным хостам с использованием конкретных PAM-сервисов. Механизм реализован средствами PAM-модуля службы SSSD. Например, после установки первого контроллера в домене будет правило allow_all (см. рис. 2.3), которое рекомендуется назначить только на администраторов, а для обычных пользователей создать еще одно правило, ограничивающее их доступом только к обычным рабочим станциям.

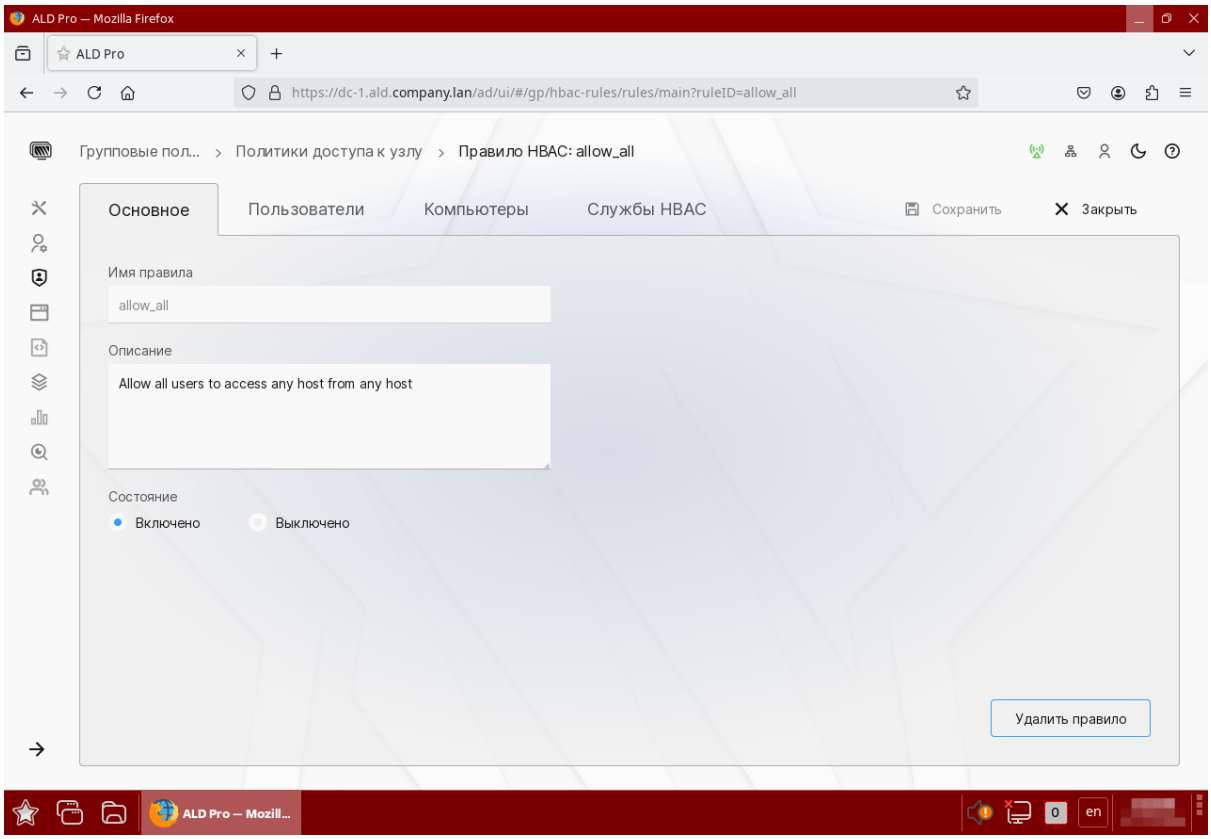


рис. 2.3 – Политика доступа к узлу

Похожим образом работают политики повышения привилегий (правила SUDO), которые позволяют дать пользователю или группе пользователей возможность выполнять на определенных хостах заданные команды через утилиту sudo. Например, вы можете дать возможность контент-менеджеру сайта при необходимости перезапускать веб-сервер с использованием команды `sudo systemctl restart apache2.service»`.

Базовые политики FreeIPA очень полезны, но они не позволят вам изменить окружение пользователей и компьютеров, установить и настроить дополнительное программное обеспечение, поэтому в продукте ALD Pro реализованы дополнительные групповые политики за счет интеграции с системой конфигурирования SaltStack, см. рис. 2.4.

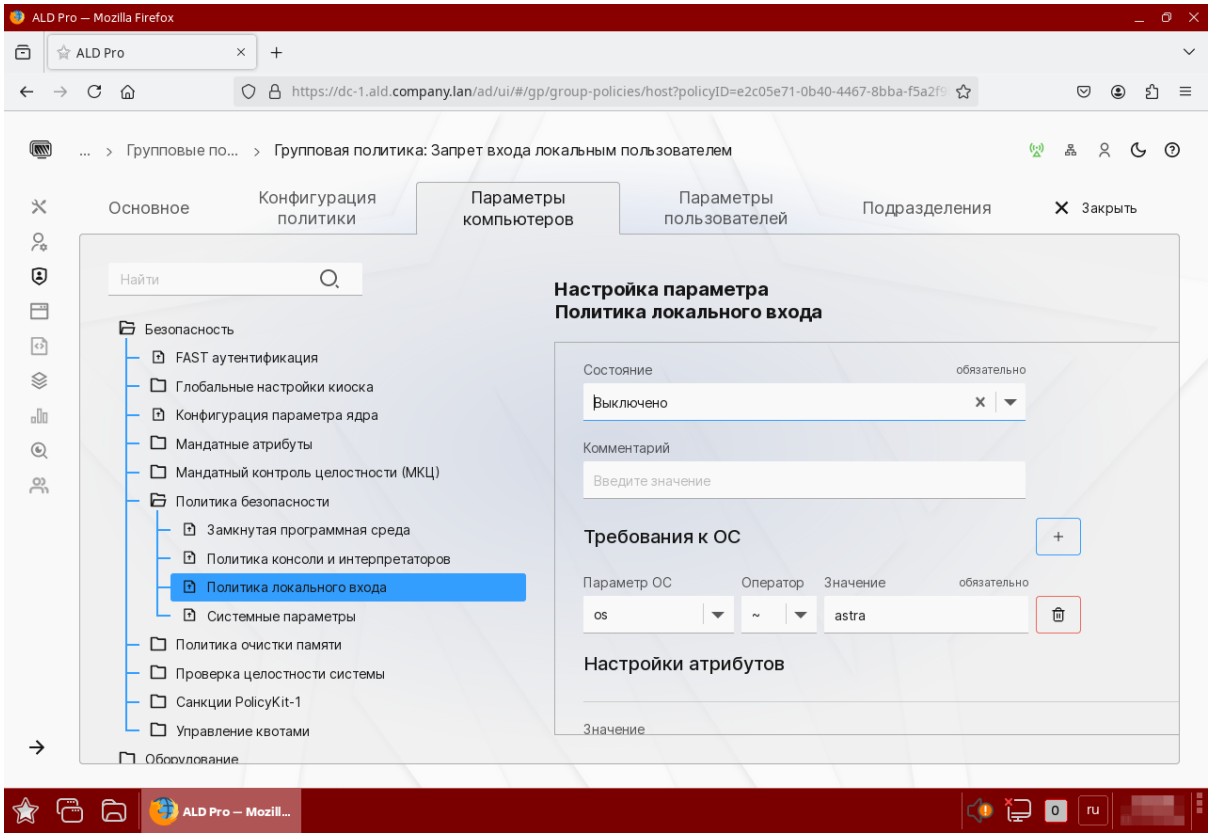


рис. 2.4 – Групповые политики ALD Pro

Групповые политики ALD Pro работают так же, как в MS AD, по методу Pull. Параметры извлекаются из домена по протоколу LDAP с учетом положения пользователя/компьютера в иерархии структурных подразделений. Для обеспечения безопасности при подключении к серверу клиентская служба aldpro-salt-minion выполняет аутентификацию по протоколу Kerberos V5 с использованием учетной записи компьютера из файла /etc/krb5.keytab. Для снижения нагрузки с контроллеров домена суммирование параметров выполняется на стороне клиента. Для повышения надежности применения параметров групповых политик изменения в операционную систему компьютера вносятся с помощью скриптов одной из лучших систем конфигурирования SaltStack, которая предоставляет множество проверенных модулей для управления содержимым конфигурационных файлов различных форматов.

Продуктовая команда ведет непрерывную работу по расширению перечня параметров групповых политик, доступных из коробки, но при необходимости системные администраторы могут самостоятельно разработать дополнительные параметры, с помощью которых можно обеспечить централизованное управление любыми корпоративными приложениями.

В завершение следует отметить, что базовые возможности службы каталога значительно расширяются дополнительными подсистемами, которые построены на хорошо известных компонентах с открытым исходным кодом:

- Подсистема «Репозиторий программного обеспечения» (Reprepro)
- Подсистема «Общий доступ к файлам» (Samba)
- Подсистема «Служба печати» (CUPS)
- Подсистема «Динамическая настройка узла» (ISC DHCP)
- Подсистема «Установка ОС по сети» (TFTP HPA + PXE)
- Подсистема «Мониторинг» (Zabbix, Grafana)
- Подсистема «Аудит» (Syslog-NG)

Установка и настройка подсистем выполняется в несколько кликов из единого портала управления, что существенно упрощает развертывание и управление инфраструктурой.

1.2.2 Почему FreeIPA, а не Samba AD

Выбор компонентов технологического стека является одним из наиболее важных моментов в разработке программного обеспечения, поэтому мы хотели бы раскрыть причины, по которым мы сделали ставку на службу каталога FreeIPA.

На данный момент работу Linux-компьютера в домене можно обеспечить с помощью нескольких клиентских компонентов с открытым исходным кодом: это служба SSSD от проекта FreeIPA, служба Winbind от проекта Samba и отдельные PAM/NSS модули. В качестве бэкенда у этих компонентов могут выступать FreeIPA, MS Active Directory, Samba AD, MIT KDC или даже простые LDAPv3 каталоги.

Выбирая простые бэкенды, такие как MIT KDC или LDAPv3-каталоги, потребуется направить излишне много усилий на решение задач репликации или интеграции с DNS и Kerberos. Поэтому выбор фактически лежит между FreeIPA и Samba AD. Давайте посмотрим на эти службы каталога внимательнее.

1.2.2.1 Служба каталога Samba AD

Когда речь заходит о Samba (https://www.samba.org), в первую очередь нужно определиться, в каком контексте обсуждается продукт, т. к. этот швейцарский нож может предоставить и общий доступ к файлам, и совместное использование принтеров, и доменные сервисы, причем как в формате Active Directory, так и в устаревшей реализации NT4.

Работу над созданием службы совместного доступа к файлам Эндрю Триджелл (Andrew Tridgell) начал еще в конце 1991 года, когда обучался в аспирантуре Австралийского национального универ-

ситета и по совместительству отвечал за сетевое администрирование. Образовательное учреждение закупило программное обеспечение PC X server eXcursion от компании DEC (Digital Equipment Corporation), и Триджеллу понадобилось организовать доступ с eXcursion к файлам, размещаемым на других серверах под управлением операционной системы Sun.

Программного продукта, который бы решал эту задачу, не было, и Триджелл (как настоящий «тыжпрограммист») путем реверс-инжиниринга сетевого протокола, просто анализируя трафик, повторил реализацию одного из наиболее успешных продуктов компании DEC, что принесло автору заслуженное признание. Впоследствии, конечно, оказалось, что у протокола все-таки есть открытая спецификация, и этот RevEng не имел большого практического смысла, но, как говорится, бешеной собаке семь верст не крюк.

Важно: Я немного покопался, немного поспрашивал и обнаружил, что нужная мне спецификация предназначена для протокола SMB и что она доступна через ftp. Тогда скачал ее и начал удалять все эти ужасные константы из кода, когда они стали понятны. Я был шокирован, увидев реальную спецификацию SMB, и понял, как мне повезло, что мой первоначальный код вообще работал.

– *Эндрю Триджелл,*

Samba-Unix Talking with PCs, Linux Journal выпуск #7, ноябрь 1994

Изначально проект так и назывался — SMB Server, то есть по названию протокола Server Message Block. Но это имя оказалось уже занято, и, по легенде, Триджелл выбрал новое название из слов, начинающихся с буквы «S» и содержащих «M» и «B» в том же порядке, получив их командой `grep` по системному словарю:

```
$ grep -i '^s.*m.*b' /usr/share/dict/words
samba
```

В дальнейшем танцевать в ритме Samba стало для Триджелла по большому счету его основной деятельностью. С 1994 года он возглавляет в университете проект, в рамках которого создавалась распределенная файловая система HiDIOS для компьютеров Fujitsu, и за это время выходит несколько релизов Samba, в том числе вторая версия, где впервые появилась поддержка службы каталога в стиле NT4.

С 2001 года Триджелл работает в VA Linux Systems над устройствами сетевого хранения данных и развивает Samba в части оптимизации под ядро Linux. Работая в Quantum с октября 2001 года над

корпоративными системами хранения, Триджелл добавляет в Samba возможность работы Linux-компьютера в составе домена Active Directory. С января 2003 года Триджелл работал над развитием систем хранения в исследовательском центре IBM, и в том же году выходит Samba 3, в которой была реализована поддержка протоколов SMB2.2 и SMB3.

В части функций Active Directory переломным моментом для проекта Samba стал 2004 год, когда комиссия Европейского союза признала Microsoft виновной в злоупотреблении монопольным положением и наложила штраф в 500 млн евро с требованием предоставить полную информацию о протоколах для обеспечения совместимости. Триджелл работал в исследовательском центре IBM, там решили воспользоваться ситуацией и запросить необходимые спецификации, но теперь этот запрос уже соответствовал интересам корпорации Microsoft. Для усиления своих позиций в судах несколько лет спустя Microsoft даже наняла разработчика Samba Криса Хартела (Chris Hertel) для написания документации.

Открывшиеся возможности определили дальнейший вектор развития продукта, и с 2005 года Триджелл полностью переключился на разработку Samba 4 как участник OSDL (Open Source Development Laboratory). Эту некоммерческую организацию IBM учредила вместе с другими технологическими гигантами — Intel, Hitachi, Hewlett-Packard, Fujitsu, NEC, Computer Associates — для ускорения проникновения Linux в корпоративную среду. Через нее финансировались ключевые проекты, именно там работал сам Линус Торвальдс и несколько других разработчиков ядра операционной системы.

При этом важно понимать, что директора этих ИТ-компаний не были какими-то наивными альтруистами, коммунистами или сторонниками культуры хиппи. Просто к тому моменту общество уже пришло к осознанию, что, закрывая исходные коды, коммерческие организации создают точно такие же барьеры для развития индустрии, как это делала AT&T несколькими десятилетиями ранее, ограничивая доступ к использованию своих патентов. Поэтому совместная работа над системными решениями может значительно способствовать развитию и выгодна всем.

Благодаря беспрецедентной открытости компании Microsoft (по известным причинам) через семь лет удалось выпустить релиз Samba 4, в котором появилась возможность работы в роли контроллера Active Directory с Heimdal Kerberos и анонсировалась возможность промышленной эксплуатации, см. рис. 2.5. На тот момент было реализовано только 40% от необходимого функционала, но этого хватило, чтобы ряд компаний стал делать на этом бизнес.

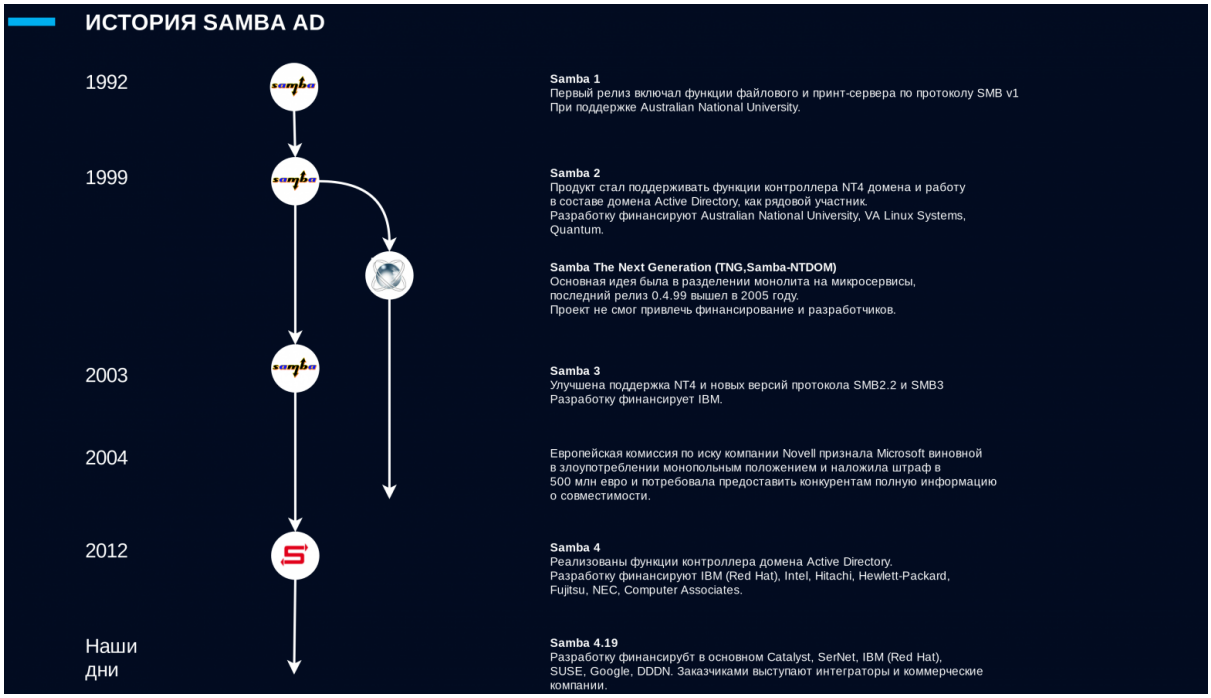


рис. 2.5 – История развития проекта Samba

В вопросах внедрения Samba AD дальше всех продвинулась французская компания Tranquil IT (https://www.tranquil.it), которой на данный момент удалось реализовать более 300 проектов, в самом крупном из которых было порядка 140 контроллеров домена. Начинали они с развертывания доменов NT4 на Samba 3, поэтому часть клиентов им нужно было просто обновить. Выступая на конференции SambaXP в 2017 году, основатели компании Tranquil IT Денис и Винсент Кардон (Denis Cardon, Vincent Cardon) отметили, что для миграции небольших инфраструктур на Samba AD препятствий практически нет, а проблемы в основном связаны с недостаточной квалификацией сотрудников: незнанием компьютерных сетей, устройства Active Directory, отсутствием навыков администрирования Linux.

Опытные инженеры в основном хвалят продукт, но предостерегают, что потребуется проштудировать большой объем плохо написанной документации, перечитать сотни тематических подписок, выполнить бессчетное количество экспериментов, и все равно что-то будет работать не так, как нужно, или не работать вовсе, а претензии можно будет предъявить только самому себе. Кстати, в части документации обязательно обратите внимание на сайт Tranqil IT, на который ссылаются даже на официальном сайте wiki.samba.org.

Важно: Берите продукт, используйте его. А если он не решит ваших задач, то вспомните, сколько вы за него заплатили. И не забудьте отправить мне отчет об ошибке.

– Эндрю Триджелл

Samba-Unix Talking with PCs, Linux Journal выпуск #7, ноябрь 1994

Если с маленькими инфраструктурами понятно, то крупные проекты внедрения возможны только при значительных доработках продукта, которые Tranquil IT воплощала через коммерческие заказы в Catalyst, например:

- При внедрении продукта в центральных банках западноафриканских государств была реализована поддержка последнего входа (4.4.0).
- На проекте министерства окружающей среды Франции потребовалось реализовать поддержку дополнительных протоколов шифрования (4.7.0), журналов в формате JSON (4.9.0), функций экспорта/импорта объектов групповой политики (4.9.0).
- Внедрение в министерстве культуры и коммуникаций Франции не обошлось без реализации Read Only Domain Controller (4.7.0).
- Внедрение в министерстве государственных финансов Франции стало возможным благодаря устранению ограничения в 4 ГБ на размер базы данных (4.9.0), разработке графических инструментов анализа топологии (4.9.0), улучшению управления DNS (4.9.0) и работы prefork (4.10.0).
- Для национального агентства по безопасности информационных систем значительно доработана документация в части функций безопасности.

Проект Samba развивается не быстро, но за последние годы в направлении Enterpise было сделано уже достаточно много, чтобы даже крупные проекты стали принципиально возможны. Конечно, нет поддержки лесов, штатной репликации SYSVOL, не стоит рассчитывать на поддержку некоторых приложений, таких как Exchange, имеющих сильную привязку к расширенным функциям Active Directory, но при необходимости без всего этого можно обойтись. Основные причины, по которым мы не сделали выбор в пользу Samba, заключаются в другом.

Проект Samba AD изначально был направлен на то, чтобы можно было создать отдельный лес из одного домена полностью на контроллерах Samba для обслуживания исключительно Windows-компьютеров. Для управления Linux-инфраструктурой домены Samba ничем не лучше, чем MS Active Directory, а возможности Samba по внедрению в состав существующих лесов Active Directory (тем более при наличии дочерних доменов) крайне ограничены. Более того, такие попытки приводят к большому количеству трудно отслеживаемых и зачастую нерешаемых проблем, что не только

не дает никаких преимуществ в реализации проектов замещения, но и ставит их под угрозу.

Например, всем известно, что Windows-компьютеры загружают шаблоны групповых политик из общей папки SYSVOL, но мало кто обращает внимание на то, что DFS-клиент доменного компьютера обслуживается ближайшим контроллером за счет того, что DNS-сервер Microsoft в зависимости от IP-адреса клиента поднимает A-записи соответствующих контроллеров в начало списка. Учитывая, что DNS-сервер от Samba не реализует такую логику, работа групповых политик Windows в больших инфраструктурах окажется невозможна.

Учитывая все вышесказанное, мы настоятельно рекомендуем создавать новую службу каталога изначально под Linux на базе FreeIPA и интегрировать ее с заменяемой службой MS AD только по открытым протоколам, чтобы обеспечить стабильную работу гибридной инфраструктуры.

1.2.2.2 Служба каталога FreeIPA

Возможность вводить Linux-компьютеры в домен Active Directory впервые появилась в службе Winbind из состава компонентов Samba, что позволило сократить разрыв между двумя мирами и начать проникновение Linux в корпоративную среду, где в то время безгранично доминировала компания Microsoft со своей активной директорией. Но если Samba искала простые способы интеграции с Windows, то команда FreeIPA направила свои усилия на то, чтобы создать новое решение с оглядкой на лучшие практики Microsoft, которое бы в полной мере соответствовало потребностям Linux, как сделала сама Microsoft в конце 90-х, когда создала свою Active Directory, используя передовые технологии UNIX, такие как LDAP, Kerberos и DNS.

Проект IPA (https://www.freeipa.org) официально стартовал в 2007 году, но готовиться к этому компания Red Hat начала сильно раньше, когда в 2004 году приобрела у компании AOL (America Online) права на код Netscape Directory Server, а заодно и команду разработчиков, которые развивали этот продукт последнее десятилетие. Если внимательно приглядеться к истории LDAP, то за сухими формулировками из пресс-релизов откроется настоящая детективная история, полная экшена, конфликтов интересов, провалов и свержений, по которой можно снять настоящий блокбастер, см. рисунок рис. 2.6.

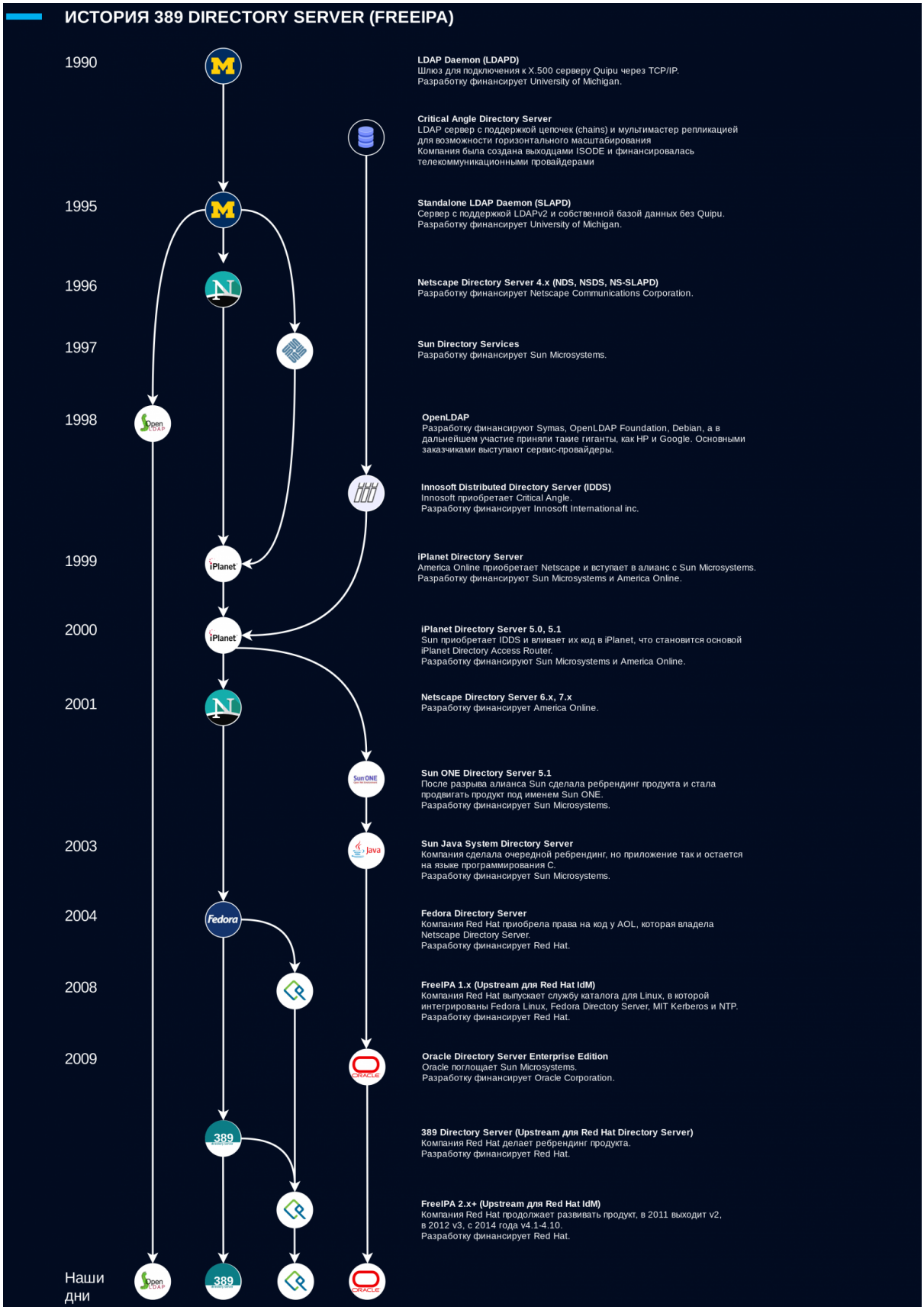


рис. 2.6 – История развития FreeIPA

Изначально стандарт электронных каталогов X.500 был разработан международным союзом электросвязи (International Telecommunication Union, ITU), и драйвером развития технологии были телекоммуникационные компании, которым нужно было предоставлять авторизованные услуги связи миллионам абонентов. К слову, в инфраструктуре российских операторов из большой тройки LDAP тоже использовался.

Основную роль в разработке технологии сыграли исследователи из Мичиганского университета (University of Michigan), Университетского колледжа Лондона (University College London), компаний PSI, Critical Angle, Netscape, Sun, America Online, HP и др. Компания Microsoft тоже участвовала в обсуждении стандарта, но ее роль заключалась в большей степени в том, что она показала альтернативный способ применения технологии для построения корпоративных доменов, и далее эстафетную палочку у телекоммуникационных компаний перехватили уже ИТ-гиганты.

На момент покупки Netscape их программный код вобрал в себя последние разработки ведущих игроков рынка, поэтому в LDAP-сервере Red Hat до сих пор таится большой потенциал, который пока еще не задействован в их IdM-решении. Например, 389 Directory Server позволяет выстраивать трехуровневую топологию, состоящую из Мастеров, Хабов и Потребителей, в которой практически нет ограничений по горизонтальному масштабированию для построения систем, испытывающих колоссальные нагрузки по операциям чтения.

В фокусе внимания FreeIPA находится разработка функций централизованной идентификации, политик и аудита (от англ. identity, policy, audit). Для этого разработчики интегрировали с LDAP-сервером эталонную реализацию Kerberos SSO (MIT Kerberos) и одну из лучших реализаций DNS-службы (ISC Bind9). А для взаимодействия с Active Directory команда Red Hat воспользовалась наработками проекта Samba AD, в создании которых принимала и продолжает принимать непосредственное участие.

Еще большее значение имеют усилия команды FreeIPA в развитии клиентской части. Переосмысление потребностей Linux в части аутентификации привело к созданию новой службы SSSD, работа над которой в 2009 году была выделена в отдельный проект. Данная аббревиатура официально расшифровывается как System Security Services Daemon, но, как рассказывает один из разработчиков продукта в кулуарах, это на самом деле первые буквы имен четырех основателей: Simo Sorce, Sumit Bose, Stephen Gallagher и Dmitri Pal.

Для становления проекта SSSD потребовалось некоторое время, но на сегодняшний день эта служба практически полностью вытеснила Winbind и является самым правильным ответом на вопрос о том, как обеспечить работу компьютера под управлением операционной системы из семейства Linux в составе домена. С помощью этой службы компьютер Linux можно ввести в домен Microsoft Active Directory, Samba AD и даже использовать в качестве бэкенда простой LDAP v3 - сервер. Но в полном объеме продукт раскрывается, конечно же, если в роли бэкенда выступает FreeIPA.

В случае FreeIPA служба SSSD использует идентификаторы безопасности POSIX, хоторые хранят-

ся напрямую в службе каталога, а не рассчитывает их с использованием вероятностей из идентификаторов SID. Вам становятся доступны такие функции, как правила HBAC и SUDO, политики монтирования AutoFS. На компьютерах под управлением Astra Linux, оснащенных подсистемой безопасности PARSEC, через FreeIPA можно централизованно управлять мандатным доступом для защиты конфиденциальной информации.

Для того чтобы окончательно развеять сомнения, сравним продукты через анализ программного кода. Разработка Samba AD выполнялась в условиях нехватки ресурсов, поэтому исходные коды предыдущей версии, отвечающие за работу SMB-протокола, были добавлены на скорую руку практически без изменений, и эти проблемы остаются в проекте до сих пор: достаточно открыть репозиторий с исходными кодами, и вы увидите папку с названием «Samba 3». И так было всегда. Например, еще в начале 2000-х Люк Лейтон (Luke Leighton) предлагал переписать продукт на микросервисную архитектуру, но привлечь финансирование ему не удалось, поэтому форк Samba TNG (The Next Generation) в 2005 году окончательно умер. И только 20 лет спустя в версии 4.16 его идеи наконец-то были реализованы, и службы MS-RPC были разнесены по отдельным приложениям с взаимодействием через сокеты unix. В проекте FreeIPA дела обстоят совсем иначе: разработчики вкладывают значительные ресурсы в рефакторинг программного кода. Например, с выходом Python3 весь программный код был переписан под новый язык, чтобы задействовать его новые возможности.

Если посмотреть на объем кода командой `wc -l $(git ls-files)`, то окажется, что в Samba сейчас почти 5.5 млн строк, в то время как в проектах FreeIPA суммарно только 3 млн, но, как известно, больше не значит лучше, и следует учесть еще два обстоятельства:

1. Проект Samba 4 вобрал в себя порядка 2,5 млн строк Samba 3, где была реализована поддержка протоколов SMB, что не относится напрямую к функциям службы каталога. И с учетом этой поправки объем кода будет уже сопоставим.

2. Продукт FreeIPA использует программный код Samba AD для интеграции с Active Directory, и, более того, именно сотрудники Red Hat помогают проекту Samba в развитии этой функциональности.

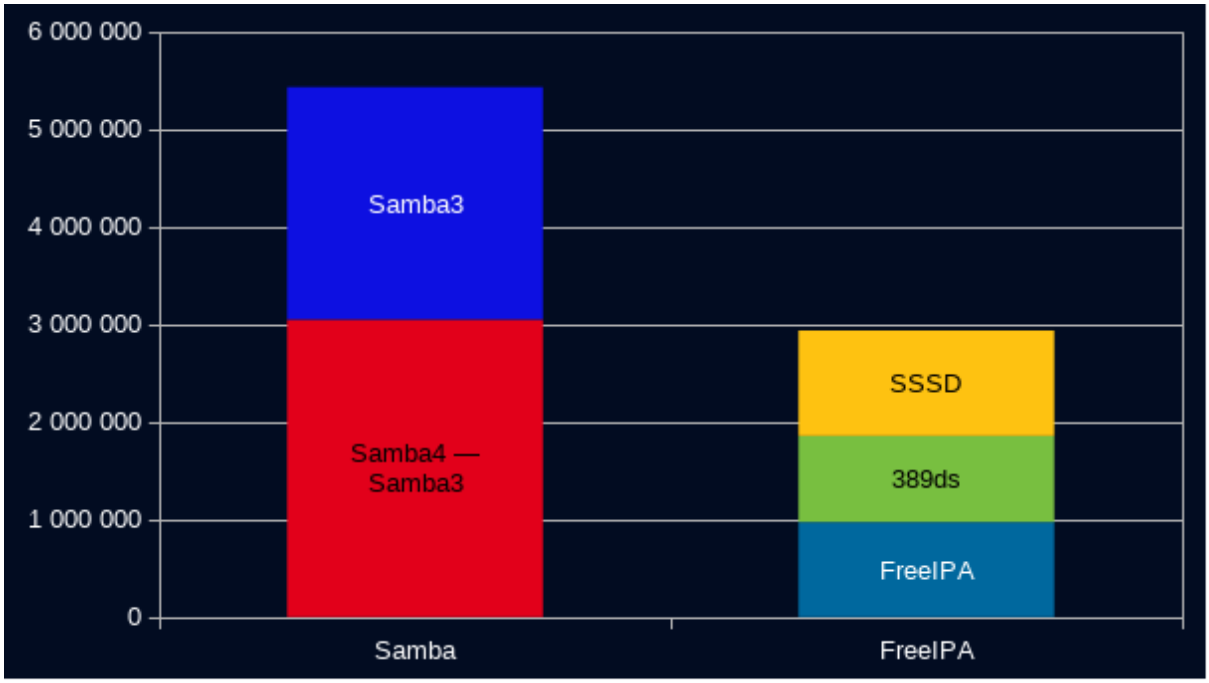


рис. 2.7 – Сравнение объема кода по количеству строк

Еще более показательным окажется сравнение динамики изменений, которую можно получить командой `shortstat`.

```
$ cd samba/
$ git log --shortstat --pretty='^ "%h", "%as", "%an", ' | tr "\n" " " | tr "^" "\n"
 "4c291514a9e", "2023-10-17", "Joseph Sutton", 2 files ...
 "d209cdf4f0c", "2023-10-17", "Joseph Sutton", 2 files ...
 "37594035547", "2023-10-17", "Joseph Sutton", 1 file ...
```

Если сравнить показатели по добавлению/удалению строк, то станет очевидным, что в последние годы компания Red Hat инвестировала в проект значительно больше усилий, чем Samba Team. И это не считая того, что часть изменений Samba внесена теми же сотрудниками Red Hat.

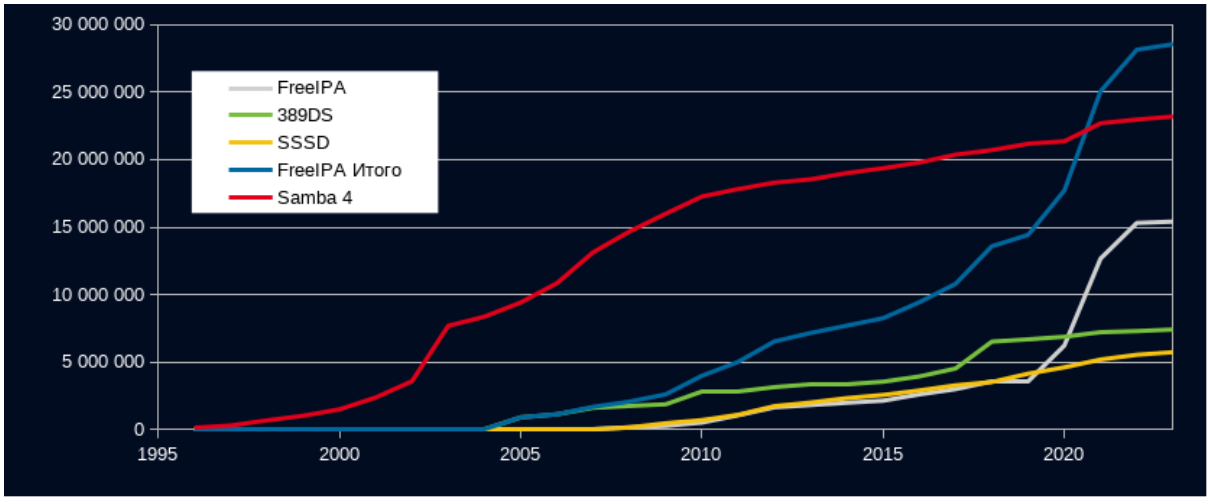


рис. 2.8 – Динамика изменений по проектам

1.2.2.3 Выводы

По нашему мнению, при планировании полного перехода на отечественные операционные системы, использующие в подавляющем большинстве ядро Linux, стратегически правильным выбором будут, конечно же, решения на базе службы каталога FreeIPA. Именно с этим бэкендом клиентская служба SSSD реализует весь свой потенциал, именно эту службу каталога будет активно развивать открытое сообщество.

По результатам наших нагрузочных тестов мы с уверенностью можем сказать, что служба каталога FreeIPA сможет закрыть потребности предприятий любого масштаба. Мы поднимали домены, содержащие 400 контроллеров, и загружали в базу миллионы учетных записей пользователей — во всех тестах FreeIPA по производительности показывает сопоставимые результаты с Active Directory. Также значительный прирост производительности бэкенда ожидается с переводом движка встроенной СУБД с Berkeley DB на LMDB, официальная поддержка которого уже появилась в LDAP-каталоге 389 Directory Server.

Возвращаясь к службе каталога Samba AD, мы можем сказать, что она подойдет тем предприятиям, которые планируют заменить только контроллеры домена, оставляя всю инфраструктуру рабочих мест под управлением операционной системы Windows. Сильной стороной Samba AD является ее возможность мимикрировать под Active Directory.

Но даже в случае выбора Samba AD нельзя ожидать, что можно будет просто заменить одни контроллеры на другие, и уж тем более не стоит рассчитывать на надежную работу домена, в составе которого будут одновременно контроллеры Samba AD и Microsoft Active Directory: эксплуатация такой инфраструктуры потребует от сотрудников запредельных компетенций по устройству активной директории вообще и каждой реализации в частности, если такие вообще есть на рынке труда. Как говорится, кто считает, что трудно найти настоящую любовь, тот не искал разработчиков на Scala.

1.2.3 Система ALD Pro и ее компоненты

Продукт ALD Pro состоит из нескольких подсистем, каждую из которых нужно устанавливать в отдельной операционной системе во избежание проблем с зависимостями пакетов и нежелательной конкуренции за ресурсы. Поэтому вместе с лицензией на контроллер домена ALD Pro предоставляется восемь лицензий на ALSE, одна из которых потребуется для установки контроллера, а остальные можно использовать для установки любых подсистем. Типовая схема развертывания продукта в отказоустойчивом режиме с использованием двух лицензий на контроллеры ALD Pro представлена на рис. 2.9.

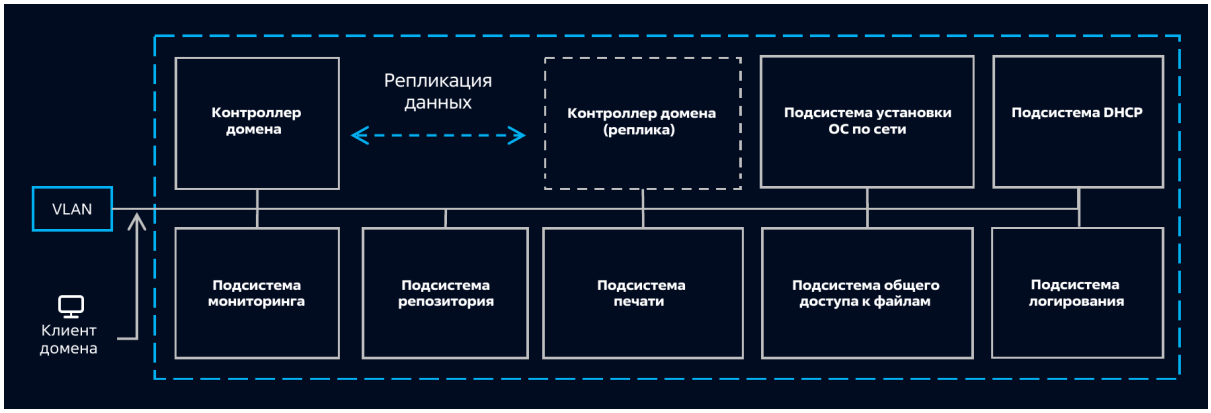


рис. 2.9 – Типовая схема развертывания ALD Pro

1.2.3.1 Контроллер домена

В ALD Pro мы придерживаемся концепции домена Microsoft, поэтому контроллер домена – это сервер, управляющий доступом к сетевым ресурсам. На контроллер домена устанавливаются следующие компоненты:

- NTP — для синхронизации времени была выбрана служба Chrony.
- Служба каталога — в основу была положена служба FreeIPA, в состав которой входят:
  - KDC — используется MIT Kerberos.
  - DNS-сервер — используется ISC Bind9.
  - LDAP-каталог — используется 389 Directory Server.
- Портал управления ALD Pro — реализован на Django с использованием дополнительных компонентов RabbitMQ и Celery.
- Через портал управления настраиваются в том числе объекты групповых политик, параметры которых предоставляются компьютерам через LDAP. Применение параметров на рабочих станциях выполняется с использованием Salt-скриптов.

Служба синхронизации времени chrony

Синхронизация времени необходима в домене для корректной работы следующих механизмов: протокола аутентификации Kerberos, двухфакторной аутентификации на основе синхронизированных по времени одноразовых паролей, разрешения конфликтов при репликации, возможности сопоставления журналов событий с разных серверов.

Мы проанализировали лучшие практики реализации синхронизации времени в MS Active Directory

и реализовали трехуровневую модель, в состав которой входят корневые серверы времени (аналог ePDC), рядовые контроллеры домена, все остальные серверы и рабочие станции.

Управление синхронизацией времени осуществляется через службу Chrony, которая является открытой реализацией протокола сетевого времени NTP (англ. Network Time Protocol) и рассчитана на работу даже в перегруженных сетях. Для настройки Chrony используется параметр групповой политики ALD Pro. Как уже было сказано выше, по умолчанию реализуется трехуровневая модель, но администраторы могут реализовать и более сложные схемы, настраивая этот параметр групповой политики вручную.

Служба каталога FreeIPA

Продукт FreeIPA является службой каталога, которая объединяет в себе LDAP-сервер на базе 389 Directory Server, центр распространения ключей Kerberos от MIT и ISC Bind9 в качестве DNS-сервера. Для интеграции с Active Directory на контроллерах FreeIPA устанавливается также компонент Samba AD, который позволяет контроллеру формировать корректный PAC-сертификат и отвечать на RPC-вызовы по протоколу SMB.

Эта служба каталога представляет собой не просто какой-то набор скриптов, с помощью которых можно установить и настроить отдельные компоненты. FreeIPA тесно интегрирует эти компоненты между собой и оборачивает их своим программным интерфейсом JSON-RPC API, через который работает веб-портал, утилита командной строки ipa и другие средства автоматизации.

Единая точка входа реализована средствами MIT Kerberos, а DNS-служба построена на базе ISC Bind9, но в качестве базы данных оба компонента используют LDAP-каталог службы 389 Directory Server. Служба FreeIPA автоматически расширяет схему, добавляя необходимые атрибуты и объектные классы, поэтому при вводе компьютера в домен для него автоматически создаются все необходимые DNS-записи, а при продвижении дополнительного контроллера автоматически настраивается репликация.

Если вернуться к MIT KDC, то этот продукт является эталонной реализацией протокола Kerberos V5, который изначально был разработан Массачусетским технологическим институтом и принят в MS Active Directory в качестве основного протокола аутентификации. До сих пор еще довольно часто можно встретить Heimdal Kerberos, но эта реализация была создана для обхода экспортных ограничений USA, поэтому она обладает меньшей совместимостью и не рекомендуется к использованию в гибридных средах.

Служба DNS построена с использованием одного из самых популярных и проверенных временем компонентов — ISC Bind9 (англ. Berkeley Internet Name Domain, BIND). Этот продукт способен выдерживать нагрузку Интернет-провайдеров, и 8 из 10 корневых серверов DNS работают именно на BIND, поэтому потребности даже самых крупных предприятий ему совершенно не страшны.

Портал управления

Система централизованного управления ИТ-инфраструктурой ALD Pro значительно расширяет возможности службы каталога FreeIPA, добавляя групповые политики на базе SaltStack и интеграцию с наиболее востребованными сервисами, такими как файловый сервер, принт-сервер, установка ОС по сети и т.д. Так же как FreeIPA, система ALD Pro оборачивает все свои функции с помощью программного интерфейса JSON API, через который работает портал управления, и этот интерфейс можно использовать для решения задач автоматизации.

В техническом плане портал управления представляет собой серверное веб-приложение, которое устанавливается на всех контроллерах в домене. Это Django-приложение, у которого есть собственная SQL база данных на PostgreSQL, но вся основная информация находится в LDAP-каталоге, поэтому не имеет значения, к порталу какого контроллера домена вы будете обращаться.

1.2.3.2 Подсистема «Репозиторий программного обеспечения»

Репозиторий ПО является одним из самых важных компонентов ALD Pro, так как в Linux повторное использование кода доведено до абсолюта и без репозиториев невозможно устанавливать и обновлять программное обеспечение. Подсистема реализована на базе продукта RepRepro, который поддерживает создание локальных Debian-совместимых репозиториев.

С помощью подсистемы вы можете создать локальный репозиторий с пакетами операционной системы и частные репозитории для размещения пакетов, которые требуется устанавливать через групповые политики.

1.2.3.3 Подсистема «Общий доступ к файлам»

Для организации общего доступа к файлам команда ALD Pro выбрала набор компонентов Samba, т.к. в нем реализована поддержка SMB-протокола, через который можно предоставить доступ в том числе клиентам Active Directory с рабочих станций Windows в гибридных сценариях развертывания.

Из коробки Samba максимально приближается к тому, как Linux работает с моделями RWX и UGO. Несовместимые DOS атрибуты, с помощью которых в Windows файл отмечается как «архивный», «скрытый» или «системный», можно отбрасывать (что снижает совместимость с Windows), сопоставлять со стандартными битами Execute (что снижает совместимость с Linux) или хранить в расширенных атрибутах файлов, если это поддерживается файловой системой.

Однако с помощью подключаемых модулей виртуальной файловой системы (англ. Virtual File System, VFS) служба Samba способна имитировать одну из следующих моделей безопасности:

- Плагин acl_xattr — позволяет значительно приблизиться к имитации Windows ACL, сохраняя

часть информации в расширенных атрибутах, но продолжая активно использовать POSIX ACL, что дает возможность сохранить пользователям Linux прямой доступ к файлам.

- Плагин nfs4acl_xattr — переводит файловый сервер в режим максимальной совместимости с Windows ACL, сохраняя всю необходимую информацию в расширенных атрибутах файлов, но это исключает возможность работы с файлами напрямую из Linux.

1.2.3.4 Подсистема «Служба печати»

В ходе реализации проектов импортозамещения на предприятиях крайне важно сохранить привычный уровень удобства в работе с наиболее востребованными сервисами, такими как печать. В продукте ALD Pro подсистема печати реализована с использованием CUPS (*англ. Common UNIX Printing System*).

Служба CUPS обеспечивает стандартизированный способ взаимодействия между компьютером и принтером, обрабатывает очередь печати, управляет настройками принтера и выполняет другие задачи, связанные с печатью. Она поддерживает различные протоколы, такие как IPP (*англ. Internet Printing Protocol*), SMB (*англ. Server Message Block*) и другие, что позволяет подключать принтеры через локальные сети или сети Интернет.

1.2.3.5 Подсистема «Динамическая настройка узла»

Для поддержки сценария установки операционных систем по сети в продукте ALD Pro реализована подсистема динамической настройки узлов (*англ. Dynamic Host Configuration Protocol, DHCP*) на базе ISC DHCP.

При необходимости вы можете использовать этот продукт для настройки рабочих станций во всей сети, но вам нужно понимать, что веб-портал имеет ограниченные средства управления и для управления инфраструктурой вам потребуется обращаться к утилитам командной строки.

1.2.3.6 Подсистема «Установка ОС по сети»

Для установки ОС по сети используется технология PXE (*англ. Preboot eXecution Environment, произносится «пикси»*), которая создает клиент-серверную среду для загрузки компьютера с сетевой карты без локальных носителей данных, таких как HDD или USB-накопители.

Если в настройках BIOS/UEFI в качестве загрузочного устройства выбрана сетевая карта, то после включения компьютера базовая система передаст управление загрузчику из постоянного запоминающего устройства сетевой карты (ПЗУ). Загрузчик сетевой карты получит настройки с помощью встроенного DHCP-клиента, качает необходимые файлы с PXE-сервера по TFTP или HTTP-протоколу и передаст управление загрузчику операционной системы из этих файлов.

Первоначально для раздачи файлов был выбран упрощенный протокол TFTP (англ. Trivial File Transfer Protocol), работающий поверх UDP, т.к. программный код, обеспечивающий его поддержку на стороне клиентов, можно было уместить на дискетах и даже в ПЗУ сетевых устройств. В продукте ALD Pro используется улучшенная реализация этого сервера от **H**ans **P**eter **A**nvin, которая так и называется TFTP-HPA.

По мере удешевления микросхем и увеличения доступного объема памяти функциональность загрузчиков стали расширять, поэтому современные прошивки сетевых карт поддерживают в том числе и более продвинутый протокол HTTP, который работает поверх TCP. В продукте ALD Pro для раздачи файлов по HTTP используется веб-сервер от Apache.

1.2.3.7 Подсистема «Мониторинг»

Для отслеживания состояния серверной группировки в продукте реализована подсистема мониторинга на базе открытой системы Zabbix. Веб-интерфейс системы написан на PHP, для хранения данных используется база PostgreSQL.

В продукте используется агентская схема, поэтому после продвижения хоста до одной из подсистем серверной группировки на него устанавливается агент Zabbix и включается отслеживание ключевых метрик.

1.2.3.8 Подсистема «Аудит»

Для возможности интеграции продукта с SIEM системами реализована подсистема аудита, которая выступает в роли коллектора и способна принимать события безопасности от всех участников домена. Подсистема использует syslog-ng – одного из лучших решений. Из коробки вы можете настроить сбор трех простых событий, дополнительная настройка подсистемы возможна через механизм дополнительных параметров групповых политик.

Совместно с коллегами из компании Positive Technologies был разработан пакет экспертизы, который позволяет анализировать события службы ALD Pro для автоматического выявления инцидентов безопасности. В продуктовой команде ALD Pro мы сформировали экспертные компетенции по FreeIPA и ее технологическим компонентам: 389 Directory Server, MIT Kerberos, ISC Bind 9. Но знать, как извлечь максимум информации о событиях безопасности из системы и как правильно построить корреляции для точного выявления инцидентов безопасности, — это две большие разницы. Поэтому мы рады нашему сотрудничеству с Positive Technologies (с PT Expert Security Center — командой MaxPatrol SIEM) и благодарны за то, что эксперты компании применили свой опыт работы с Active Directory для выявления подозрительной активности в FreeIPA. Уверены, что новые правила корреляции — заметный вклад в развитие этого направления.

1.2.3.9 Клиентская часть ALD Pro

В роли клиентской части на доменных машинах в ALD Pro выступает набор из следующих компонентов:

- **chrony** – клиент сервера времени.
- **aldpro-client** – клиентская часть ALD Pro, через которую устанавливаются и настраиваются остальные клиентские приложения и сервисы. Содержит в том числе скрипты для автоматического обнаружения сервисов, например, aldpro-client-service-discovery и aldpro-samba-discovery.
- **astra-freeipa-client** — утилита от Astra Linux для внесения дополнительных настроек в конфигурацию sssd и других служб в соответствии с требованиями операционной системы.
- **freeipa-client** – клиентская часть подсистемы FreeIPA, настраивает службу SSSD.
- **krb5-user** — поддержка работы с билетами Kerberos, включает утилиты kinit, kdestroy и др.
- **sssd** — набор служб для работы машины в составе домена.
- **sssd-ldap** — LDAP бэкенд службы SSSD.
- **ldap-utils** — утилиты ldapsearch, ldapmodify и другие.
- **freeipa-admintools** — набор утилит администрирования ipa CLI.
- **aldpro-salt-minion** и **salt-aldpro-salt-common** — пакеты кастомизированной службы salt-minion, которая обеспечивает работу групповых политик ALD Pro.
- **syslog-ng** — клиентская часть подсистемы журналирования.
- **zabbix-agent** — клиентская часть подсистемы мониторинга.

Подробнее об этих компонентах будет рассказано в следующих модулях, которые посвящены соответствующим подсистемам и работе компьютера в домене ALD Pro в целом.

1.3 Перед развертыванием ALD Pro

1.3.1 Минимальные требования к инфраструктуре для развертывания ALD Pro

Для обеспечения базовой функциональности ALD Pro нужен только один контроллер домена, но для обеспечения отказоустойчивости рекомендуется устанавливать не менее двух серверов, поскольку такая конфигурация позволит обновлять контроллеры и выполнять другие задачи обслуживания без прерывания сервисов аутентификации и авторизации. Минимальная конфигурация

оборудования для развёртывания подсистем приведена в нижеследующей таблице.

табл. 2.1 – Минимальные характеристики для установки подсистем ALD Pro

Компонент	Ядер ЦПУ	RAM	SSD	Кол-во
Контроллер домена				
для обслуживания до 3000 пользователей и 3000 компьютеров из расчета потребностей службы каталога и механизма групповых политик	4 шт.	8 ГБ	>= 50 ГБ	2 шт.
для обслуживания 10000 дополнительных компьютеров	10 шт.	-	-	-
для хранения 10000 дополнительных пользователей в каталоге	-	0.5 ГБ	0.5 ГБ	-
Подсистема «Репозитории ПО»	2 шт.	2 ГБ	>=100 ГБ	1 шт.
Подсистема «Общий доступ к файлам»	2 шт.	2 ГБ	>=30 ГБ	1 шт.
Подсистема «Служба печати»	2 шт.	2 ГБ	>=30 ГБ	1 шт.
Подсистема «Динамическая настройка узла» DHCP	2 шт.	2 ГБ	>=30 ГБ	1 шт.
Подсистема «Установка ОС по сети»	2 шт.	2 ГБ	>=30 ГБ	1 шт.
Подсистема «Мониторинг»	2 шт.	2 ГБ	>=30 ГБ	1 шт.
Подсистема «Аудит»	2 шт.	2 ГБ	>=30 ГБ	1 шт.

1.3.2 Рекомендации по планированию дополнительных ресурсов службы каталога

От работы доменных служб зависит надежность работы всей ИТ-инфраструктуры, поэтому при планировании домена администраторы должны действовать проактивно и применять различные способы вертикального и горизонтального масштабирования, чтобы исключить проблемы недостаточной производительности. В данном разделе представлены рекомендации, которые помогут вам в планировании ресурсов службы каталога.

Вертикальным масштабированием называют повышение производительности системы за счет увеличения производительности отдельного узла путем выделения серверу дополнительных вычислительных ресурсов: оперативной памяти, потоков центрального процессора и т. п.

1.3.2.1 Объем памяти

В работе службы каталога преобладают операции чтения, поэтому в производительности контроллера решающую роль играет достаточный объем оперативной памяти, чтобы контроллер мог об-

рабатывать запросы без обращения к медленным дискам.

Файлы каталога расположены в папке `/var/lib/dirsrv`. После развертывания первого контроллера размер базы составляет порядка 45 Мб и увеличивается по мере создания объектов, в среднем по 50 КБ на каждую дополнительную учетную запись пользователя. Таким образом, размер файлов на диске можно рассчитать математически: для нужд самой операционной системы контроллеру следует выделить не менее 50 ГБ, а под хранение каталога для упрощения расчетов возьмем по 1 ГБ на каждые 20 тысяч объектов. Расчетное значение должно составлять не более 40% от доступного пространства.

$$HDD, GB = 50 + N \text{ объектов}/20000$$

Для загрузки каталога требуется чуть больше оперативной памяти, т.к. служба **ns-slapd** индексирует отдельные атрибуты каталога, но в среднем можно использовать те же нормативы. Пустая база данных занимает в памяти не менее 65 Мб, и это значение растет по мере увеличения числа объектов, на каждую дополнительную учетную запись в среднем добавляется 50 КБ. Таким образом, минимально необходимый объем оперативной памяти можно рассчитать математически: для нужд самой операционной системы контроллеру следует выделить порядка 2-3 ГБ, а для работы с каталогом в целях упрощения расчетов возьмем по 1ГБ на каждые 20 тысяч объектов:

$$RAM, GB = 3 + N_{\text{объектов}}/20\ 000$$

Приведем несколько примеров:

- Для работы с каталогом, в котором содержится 10 000 пользователей и 100 групп, контроллеру нужно выделить порядка 4 ГБ ОЗУ:

$$RAM, GB = 3 + 10100/2000000A04 \text{ Гб}$$

- Для работы с каталогом, в котором 100 000 пользователей и 30 000 групп, контроллеру нужно выделить до 16 ГБ ОЗУ:

$$RAM, GB = 3 + 130000/2000000A010 \text{ Гб}$$

Внимание: При создании учетных записей контроллер домена потребляет в два раза больше оперативной памяти, т.е. примерно 100 КБ на каждую новую запись, которая высвобождается после перезапуска службы **dirsrv**. Эта особенность работы не проявляется в сценарии репликации, но ее следует учитывать при планировании начальной миграции.

Примечание: Для обеспечения надёжной работы контроллеров домена и сопутствующих сервисов, а также для исключения увеличения времени репликации из-за дополнительных операций чтения данных с диска, контроллерам домена рекомендуется выделять не менее 8 ГБ оперативной памяти.

1.3.2.2 Ресурсы центрального процессора

Если объем потребляемой оперативной памяти зависит от количества записей в каталоге, то количество ресурсов центрального процессора определяется тем, сколько пользователей (компьютеров) должен обслуживать конкретный контроллер. Например, в базе может быть 100 тысяч пользователей, но нагрузка по их обслуживанию может распределяться между 20 репликами, и в этом случае на каждый контроллер будет приходиться не более 5 тысяч обслуживаемых пользователей.

Примечание: Учетные записи уволенных сотрудников, которые просто находятся в каталоге, напрямую не влияют на то, сколько ядер процессора потребуется контроллеру для нормальной работы. Деградация сервиса при увеличении количества пользователей в каталоге возможна только в тех сценариях, когда используется перебор всех записей, особенно по неиндексированным атрибутам.

Корпоративные приложения очень по-разному взаимодействуют с каталогом. Например, Kerberos нагружает процессор значительно больше LDAP-аутентификации, т.к. использует сложные криптографические алгоритмы, а использование кеширования помогает снизить нагрузку в разы, если не на порядок. Поэтому предложить какой-то универсальный нормативный показатель на одного сотрудника не представляется возможным, и системным администраторам следует рассчитать его для своей организации самостоятельно, достигая использования ЦПУ на уровне 40% от максимума.

Отправной точкой в расчетах может быть выделение одного потока процессора для работы операционной системы и еще по одному потоку на каждую тысячу обслуживаемых пользователей (компьютеров):

$$\text{ЦПУ} = (1 + N\ users/1000)$$

Приведем несколько примеров.

Пример 1.

Для обслуживания 2 - 3 тысяч пользователей контроллеру домена нужно выделить порядка 4

потоков:

$$\text{ЦПУ} = (1 + 3000/1000) = 4$$

Компания RedHat в своей документации рекомендует использовать контроллеры домена именно такой производительности, но эта рекомендация не подойдет крупным организациям, в штате которых работает несколько сотен тысяч сотрудников, т. к. контроллер, обслуживающий 5-10 тысяч сотрудников, будет в 2-3 раза эффективнее использовать оперативную память в пересчете на активного пользователя, чем контроллер, обслуживающий 3 тысячи сотрудников.

Пример 2.

Для обслуживания 10 тысяч пользователей контроллеру домена нужно выделить 11 потоков.

$$\text{ЦПУ} = (1 + 10000/1000) = 11$$

Повторимся, что данные оценки сильно зависят от сценария использования службы каталога, поэтому на такое количество пользователей мы рекомендуем для начала выделять порядка 8-12 потоков, а затем подбирать точное значение, опираясь на целевой показатель в 40% нагрузки от максимума.

Пример 3.

Если под контроллер домена выделить физический сервер с двумя процессорами Xeon Silver (48 потоков), то теоретически он сможет обслуживать 47 тысяч пользователей. При планировании таких контроллеров следует принимать во внимание пропускную способность сети (потребуется не менее гигабитного интерфейса) и другие ограничения системы.

$$\text{ЦПУ}_{\text{потоков}} = (1 + 47000/1000) = 48$$

Внимание: При изменении количества потоков процессора (замена процессора или изменение настроек виртуальной машины) необходимо обновить переменную DAEMON_ARGS в файле `/etc/default/krb5-kdc` и перезапустить службу **krb5-kdc**.

Примечание: Если нагрузка на ЦПУ не превышает 40%, то добавлять ядра не имеет смысла и дополнительный прирост в скорости обработки запросов можно получить только за счет увеличения частоты центрального процессора. Мы рекомендуем использовать процессоры с базовой тактовой частотой ниже 1,8 ГГц.

Примечание: До версии 2.4.0 на контроллерах домена работала служба Salt Master, которой нужно было выделять столько же ресурсов, сколько и службе каталога. После перехода на автономную службу aldpro-salt-minion требования к ресурсам были существенно снижены, что отражено в приведенных выше расчетах.

1.3.2.3 Пропускная способность сети

В процессе обслуживания клиентов контроллеры домена, как правило, получают небольшие входящие запросы на предоставление относительно больших объемов данных, поэтому на контроллерах обычно преобладает исходящий трафик.

Любые оценки являются субъективными, но в качестве общих рекомендаций следует исходить из того, что при количестве пользователей до 5 000 будет достаточно интерфейса с пропускной способностью в 100 Мбит/с, а если предполагается обслуживать больше пользователей, установите на контроллере интерфейс с пропускной способностью в 1 Гбит/с. В результате оптимизации нужно добиться, чтобы показатель использования сети не выходил за пределы 40% от максимума.

Учитывая то, что внешние каналы связи редко достигают гигабитных скоростей и активно используются пользователями для выхода в Интернет, для каждого офиса рекомендуется создавать отдельный сайт (локацию) и размещать в нем один-два контроллера для локальной обработки клиентских запросов. Такой подход к организации домена позволит разгрузить VPN-туннели между офисами и использовать эти каналы связи только для репликации.

1.3.2.4 Модуль синхронизации

Основные ресурсы, потребляемые при работе модуля синхронизации — это дисковое пространство, размер которого будет прямо пропорционален количеству синхронизируемых объектов. Дисковое пространство потребляется БД PostgreSQL, где хранятся метаданные всех синхронизированных объектов. Для корректной работы модуля синхронизации необходимо выделить дополнительно 15-20% дискового пространства от расчётного.

С настройками по умолчанию PostgreSQL хранит кеш объектов в оперативной памяти. Учитывая, что около 80% обрабатываемых запросов являются запросами на чтение, будет достаточно выделить дополнительно 10-15% от расчетного объёма ОЗУ.

1.3.2.5 Прочие службы

Начиная с версии ALD Pro 2.4.0, служба Salt Master больше не используется на контроллерах домена, поэтому выделение дополнительных ресурсов на ее работу не требуется.

Начиная с версии ALD Pro 2.3.0, брокер сообщений RabbitMQ не использует кластеризацию, поэтому его накладными расходами можно пренебречь, отнеся их к базовым потребностям операционной системы.

1.3.2.6 Горизонтальное масштабирование

Горизонтальным масштабированием называют повышение производительности системы за счет увеличения количества вычислительных узлов, обслуживающих клиентов, без изменения их производительности. Например, если предполагается использовать контроллеры с 8-12 потоками ЦПУ, которые могут обслуживать по 10 тысяч пользователей, то в организации со штатом в 30 тысяч сотрудников потребуется не менее 3 контроллеров такой производительности:

$$N_{controllers} = 30000_{users}/10000_{percontroller} = 3$$

Синхронизация данных между контроллерами обеспечивается процессом репликации, и в службе каталога FreeIPA она происходит в мультимастер-режиме, т.е. изменения можно вносить на любом контроллере. Балансировка нагрузки и отказоустойчивость реализована на стороне клиента и обеспечивается через DNS.

К механизму репликации мы вернемся, когда будем изучать *Модуль 8. Управление топологией в домене*.

1.3.3 Рекомендации по именованию домена

В документации и обучающих курсах применяются домены **contoso.com** или **ad.example.test**, которые используют для обучающих целей, но не приемлемы для развёртывания продуктивных сред. Поэтому мы рекомендуем придерживаться следующих правил:

- Лучший вариант – это использовать свой домен третьего уровня. Например, если организация приобрела домен **mycompany.ru** для своего веб-сайта и почты, то для ALD Pro можно выделить доменное имя третьего уровня **ald.mycompany.ru**.

  Крайне важно не использовать DNS-зоны, которые принадлежат другим субъектам или могут быть ими заняты.

- Если вы не хотите использовать публичный домен, то можно задействовать зоны **«.lan»** или

«.internal», например, **ald.mycompany.lan**.

Microsoft активно продвигает зону **«.local»**, поэтому она упоминается во многих инструкциях, но при использовании этой зоны вы можете столкнуться с конфликтами Multicast DNS (RFC6762) и невозможностью использования протокола **zeroconf**, который реализован в Linux службой Avahi, поэтому зону .local мы не рекомендуем.

- Не пытайтесь использовать для домена однокомпонентные имена (Single Label Domain, SLD), такие как **.yandex**, **.google** или **.microsoft**, даже если ваша организация (вдруг) приобрела себе такое имя. В домене MS AD конфигурация SLD для новых доменов больше не поддерживается, и в службе каталога FreeIPA на развертывание домена с таким именем тоже установлен явный запрет.

Важно: Зоны «.lan» и «.internal» не зарегистрированы в глобальном списке Top-Level Domains, поэтому чисто теоретически их могут ввести в эксплуатацию, что приведет к конфликту в будущем, но вероятность этого события крайне мала.

Выбор префикса для домена:

- Выберите префикс, который вряд ли устареет, т. е. избегайте имен, таких как линейка продуктов или операционная система, которые могут измениться в будущем. Рекомендуется использовать универсальные имена, такие как corp, ent или ald.

- Выберите префикс, содержащий только стандартные символы Интернета a–z, 0–9 и (-), но не полностью числовые.

- Выберите префикс длиной не более 15 символов, потому что первый компонент доменного имени будет использован в качестве NetBIOS-имени.

- При выборе префикса учитывайте, что два домена не смогут работать в доверительных отношениях, если у них будут совпадать NetBIOS-имена, например, если у вас текущий домен AD DS имеет имя **corp.mycompany.com**, то не стоит для домена ALD Pro использовать имя **corp.mycompany.ru**. Иначе вы не сможете создать между этими доменами доверительные отношения.

Важно: В стандарте доменных имен (RFC1034) разрешается использовать заглавные буквы латинского алфавита A-Z, но служба каталога FreeIPA налагает дополнительные требования и обязует использовать только нижний регистр символов.

Если вы в команде ввода в домен укажите имя хоста большими буквами, то процедура завершится ошибкой. Если поменять имя хоста на заглавные (например, **FS-01.ald.company.lan**) после ввода в домен, то вы столкнетесь с ошибками при установке подсистем ALD Pro на такие хосты.

Для именования хостов на предприятиях обычно используют определенные соглашения. Например, в соответствии с правилом `<type>-<number>.<суффикс домена>` сервер может называться **dc-18.ald.company.lan**. Это существенно упрощает работу с большой инфраструктурой и позволяет обрабатывать имена в заданиях автоматизации. Указанное имя можно легко разделить на части по символу «-» с помощью утилиты `cut`. Результат выполнения команды покажет, что выбранный узел является контроллером домена, а не сервером мониторинга:

```
admin@dc-1:~$ echo "dc-18.ald.company.lan" | cut -d "." -f 1 | cut -d "-" -f 1
dc
```

Понятное дело, что на имена не стоит сильно полагаться, т.к. сервер dc-18 может быть пока еще просто сервером, который только предполагается сделать контроллером домена. Но мы надеемся, что основная суть понятна.

1.4 Практика и тестирование

1.4.1 Тест: Модуль 1. Система ALD Pro и ее компоненты

Пожалуйста, ответьте на вопросы, чтобы проверить свои знания. Ответы можно найти в конце списка. Учтите, что у некоторых вопросов есть несколько правильных ответов или их можно менять местами.

1.4.1.1 Вопросы

Вопрос 1. Что такое ALD Pro?

a. Программное обеспечение для управления базами данных (СУБД)

b. Система централизованного управления ИТ-инфраструктурой

c. Антивирусная программа для Astra Linux

d. Средство разработки веб-приложений

Вопрос 2. Какие основные функции выполняет ALD Pro?

a. Управление пользователями и компьютерами, централизованная аутентификация, применение групповых политик

b. Мониторинг сетевого трафика и обнаружение вторжений

c. Резервное копирование и восстановление данных

d. Управление виртуальными машинами

Вопрос 3. На какой из реализаций служб каталога основана система централизованного управления ИТ-инфраструктурой ALD Pro?

a. MS Active Directory

b. Samba AD

c. FreeIPA

d. Apache Directory

e. OpenLDAP

Вопрос 4. Какие компоненты входят в состав контроллера домена ALD Pro?

a. Веб-сервер nginx, база данных MySQL, FTP-сервер

b. Служба синхронизации времени chrony, FreeIPA, портал управления, ISC BIND

c. Почтовый сервер, DNS-сервер PowerDNS, DHCP-сервер

d. Система мониторинга, система резервного копирования, файловый сервер NFS

Вопрос 5. Какие подсистемы дополнительно могут быть установлены в ALD Pro?

a. Веб-сервер lighttpd, почтовый сервер, система управления проектами

b. Репозиторий программного обеспечения, общий доступ к файлам, мониторинг и аудит

c. Система виртуализации, контейнеризация, оркестрация k8s

d. Служба печати, динамическая настройка узла и установка ОС по сети

e. База данных MongoDB, система кэширования memcache, брокер сообщений Kafka

Вопрос 6. Какая из систем управления конфигурацией используется в ALD Pro для реализации механизма групповых политик?

a. Ansible community

b. Open Source Puppet

c. Open Source Chef

d. Salt-Stack

Вопрос 7. Вы планируете создать в домене еще 300 000 объектов. Какой объем ОЗУ нужно будет дополнительное выделить контроллерам домена ALD Pro?

a. 8 ГБ

b. 15 ГБ

c. 30 ГБ

d. 4 ГБ

Вопрос 8. Выберите максимально приемлемый вариант наименования домена.

a. ALD.mycompany.internal

b. ald.mycompany.local

c. ald.mycompany.lan

d. 773.mycompany.ru

Ответы на вопросы

1. b
2. a
3. c
4. b
5. b, d
6. d
7. b
8. c

1.5 Заключение

В этом модуле мы познакомились со службой каталога, а в следующем будем уже устанавливать первый контроллер домена и, конечно же, вводить клиентскую машину в домен.

1.6 Дополнительные источники информации

- Документация по службе каталога FreeIPA
- Документация по клиентской службе SSSD
- Информация о протоколе аутентификации Kerberos
- Документация по Samba AD от Tranquil IT

1.7 Обратная связь

Если остались вопросы, то их всегда можно задать в специальной теме.

Модуль 2. Развертывание ALD Pro

2.1 Введение

Из этого модуля вы узнаете о том, как установить первый контроллер и присоединить рабочую станцию к домену. Приготовьтесь, что будет очень много практики!

При установке продукта мы затрагиваем много вопросов по администрированию Astra Linux, которые уже рассматривали в первой части курса, но мы постарались привести в этом модуле небольшие емкие пояснения, чтобы он мог служить кратким руководством (Quick Start Guide) для тех системных администраторов, кому доведется знакомиться с учебным курсом, начиная именно с этого модуля, как точки входа.

2.2 Подготовка виртуальной среды для прохождения этого курса

2.2.1 Описания тестового стенда

В рамках курса мы установим пару контроллеров домена и несколько подсистем, в том числе рабочих станций. Все виртуальные машины будут работать в одной натированной сети **10.0.1.0/24**, см. рис. 2.10.

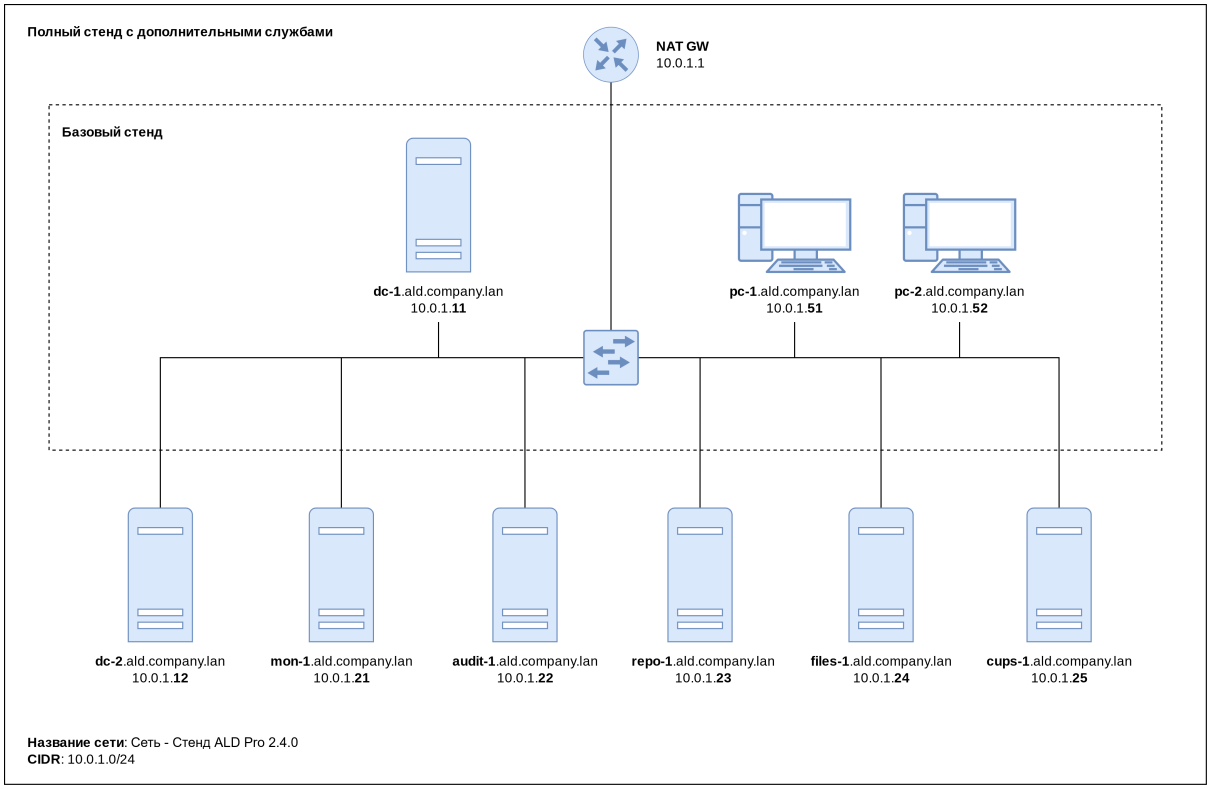


рис. 2.10 – Схема тестового стенда ALD Pro

В качестве доменного имени возьмем значение **ald.mycompany.lan** в соответствии с рекомендациями из предыдущего модуля, см. *Рекомендации по именованию домена*.

Для упрощения развертывания тестового стенда в ходе установки первого контроллера мы сделаем копию виртуальной машины и будем использовать ее в качестве шаблона для установки остальных подсистем и рабочих станций, поэтому на всех хостах у нас будет операционная система с максимальным уровнем защищенности **«Смоленск»** и динамическими дисками на **50 ГБ**, хотя для клиентов ALD Pro уровень защищенности не имеет значения.

Примеры мы будем показывать на простом кроссплатформенном инструменте виртуализации VirtualBox и эту же среду рекомендуем использовать для выполнения практических заданий, так как в этом случае вам будет значительно проще ориентироваться в предоставленных рекомендациях.

2.2.2 Минимальные требования для тестовой среды

Тестовый стенд будет работать без реальной нагрузки, поэтому мы можем существенно снизить минимальные требования к производительности виртуальных машин, но при планировании ресурсов реальной инфраструктуры ориентируйтесь, пожалуйста, на рекомендации, которые были изложены ранее, см. *Модуль 1. Система ALD Pro и ее компоненты*, раздел *Минимальные требования к*

инфраструктуре для развертывания ALD Pro.

табл. 2.2 – Минимальные требования по ресурсам для выполнения практических заданий

Компонент системы	Ядер CPU	RAM	SSD	Кол-во
Контроллер домена	2	4 ГБ	50 ГБ	1 шт.
Резервный контроллер домена	2	4 ГБ	50 ГБ	1 шт.
Подсистема «Репозиторий ПО»	1	2 ГБ	50 ГБ	1 шт.
Подсистема «Общий доступ к файлам»	1	2 ГБ	50 ГБ	1 шт.
Подсистема «Служба печати»	1	2 ГБ	50 ГБ	1 шт.
Подсистема «Мониторинг»	1	2 ГБ	50 ГБ	1 шт.
Подсистема «Аудит»	1	2 ГБ	50 ГБ	1 шт.
Клиентские машины	1	2 ГБ	50 ГБ	2 шт.

В ходе выполнения практических заданий вам не потребуется включать все эти виртуальные машины сразу, поэтому для прохождения второй части курса достаточно будет ноутбука со следующими характеристиками:

- CPU: **8 ядер**.
- RAM: **16 ГБ**.
- SSD: **150 ГБ** (желательно, чтобы диск был SSD, т.к. производительность работы контроллеров домена сильно зависит от скорости дисковой подсистемы).

2.2.3 Настройка натированной сети

Взаимодействие между хостами в домене осуществляется через компьютерную сеть TCP/IP по протоколам KRB5, DNS, LDAP(S), RPC, HTTP(S), NTP, SMB и многим другим. В рамках курса мы будем устанавливать все виртуальные машины в одной общей натированной сети VirtualBox, для создания которой нужно выполнить следующие действия:

- Открыть менеджер сети из меню *Файл → Инструменты*.
- Перейти на вкладку *Сеть NAT* и нажать кнопку *Создать*.
- В настройках сети задать следующие параметры:
  - Имя сети: **Сеть - Стенд ALD Pro 2.4.0**.
  - CIDR сети (диапазон адресов): **10.0.1.0/24**.
  - Поддержка DHCP: **выключить**, т.к. IP-адреса на хостах стенда будем настраивать вруч-

ную, см. рис. 2.11.

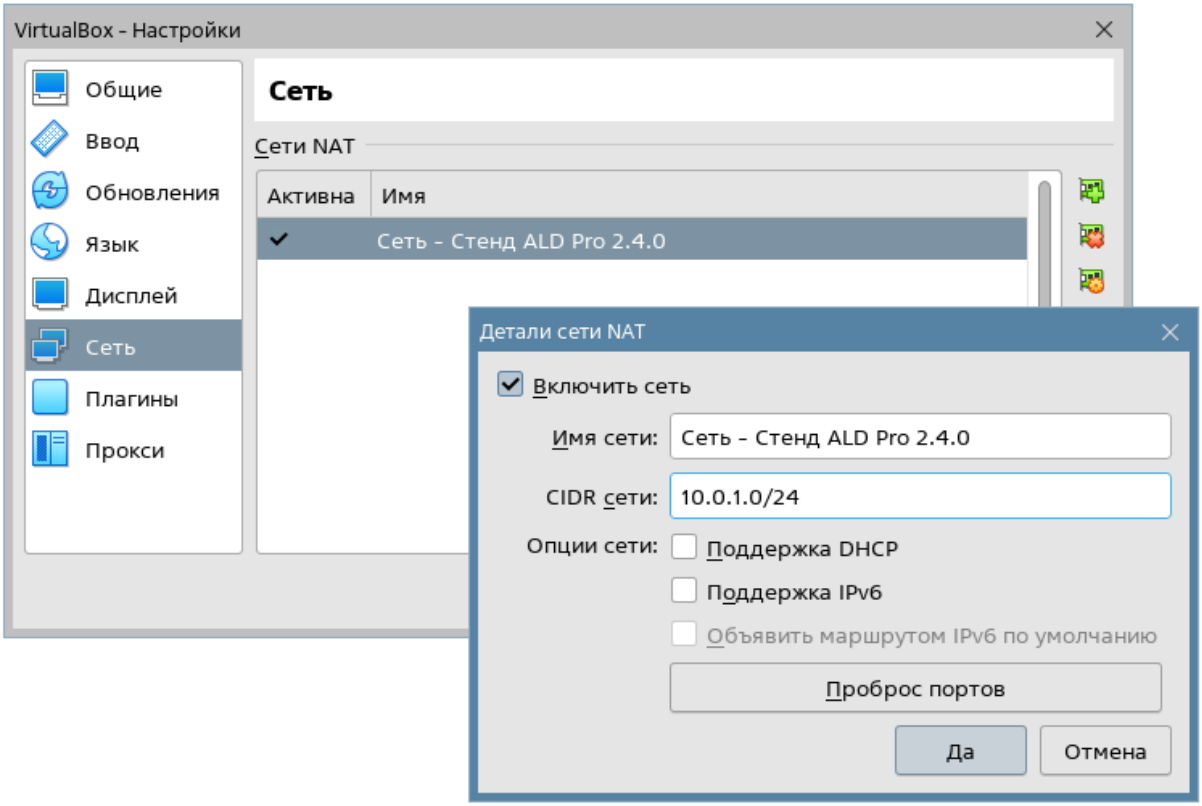


рис. 2.11 – Создание сети NAT

В натированной сети средствами VirtualBox будет создан **шлюз**, через который виртуальные машины смогут выходить в Интернет. Адресом шлюза будет первый IP-адрес сети **10.0.1.1**.

Примечание: Учитывая, что все хосты будут находиться в одной сети, они смогут взаимодействовать друг с другом без каких-либо ограничений, но в реальной жизни компьютерную сеть предприятия обычно нарезают на множество сегментов, взаимодействие между которыми ограничивают правилами межсетевого экрана для повышения безопасности в соответствии с концепцией нулевого доверия (Zero Trust). Чтобы правильно настроить сетевое оборудование, ознакомьтесь с разделом *Правила межсетевого экрана*.

2.3 Развертывание первого контроллера домена

2.3.1 Подготовка виртуальной машины

2.3.1.1 Создание виртуальной машины

В окне VirtualBox Manager выполните команду *Машина → Создать* и укажите следующие параметры:

- Имя: **dc-1.ald.company.lan - Стенд ALD Pro 2.4.0**.
- Папка машины: **по умолчанию**.
- Тип: **Linux**.
- Версия: **Other Linux (64-bit)**.
- Объем памяти: **4096 МБ**.
- Жесткий диск: **Создать новый виртуальный жесткий диск**.
- Тип файла: **VDI (VirtualBox Disk Image)**.
- Формат хранения: **Динамический виртуальный жесткий диск**.
- Путь: **по умолчанию**.
- Размер: **50 ГБ**.

После создания виртуальной машины в ее настройках задайте следующие параметры:

- На вкладке *Система → Процессор* укажите **4 ЦП** для ускорения установки ОС, далее можно уменьшить этот параметр до **2 ЦП**.
- На вкладке *Система → Ускорение* укажите интерфейс паравиртуализации **KVM**.
- На вкладке *Носители* для компакт-диска укажите установочный файл installation-1.7.6.11-26.08.24_17.26.iso.
- На вкладке *Сеть* выберите тип подключения **Сеть NAT** и имя **Сеть - Стенд ALD Pro 2.4.0**, см. рис. 2.12.

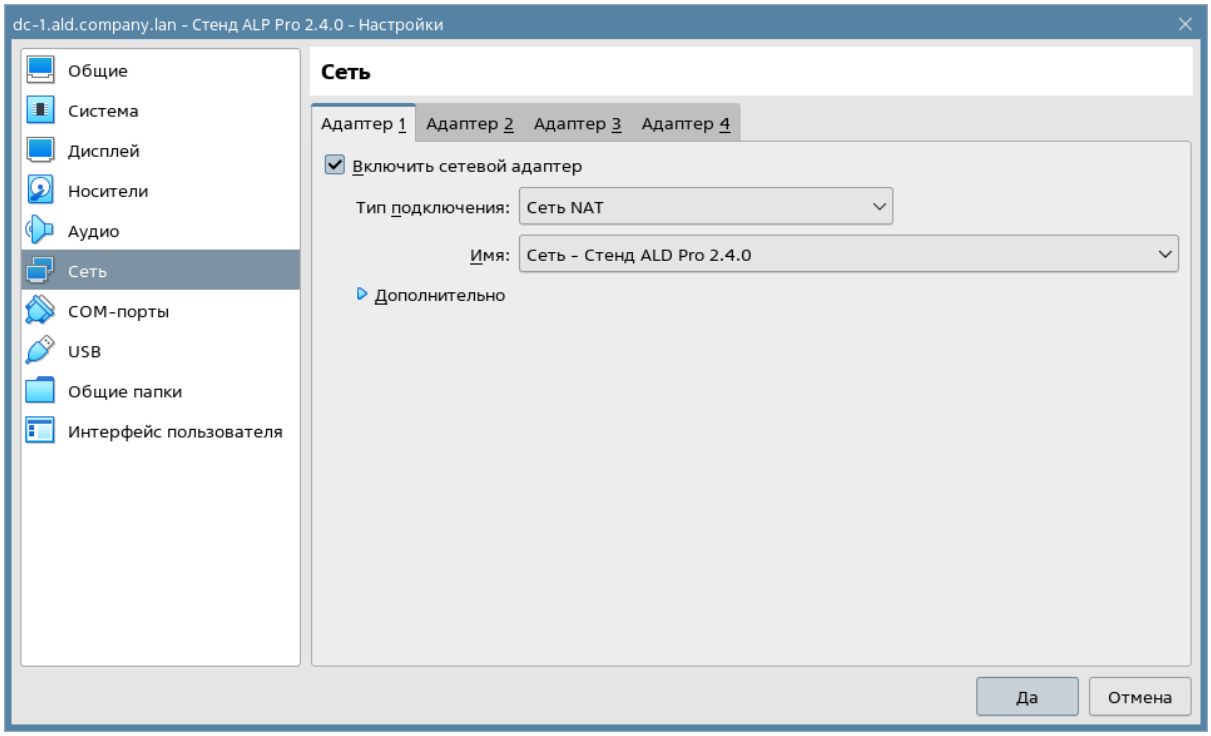


рис. 2.12 – Настройки сети VM

Примечание: Для установки операционной системы необходимо будет загрузить образ диска ALSE версии 1.7.6 из официального репозитория https://dl.astralinux.ru/astra/frozen/1.7_x86-64/1.7.6/iso/installation-1.7.6.11-26.08.24_17.26.iso

Мы не рекомендуем загружать ISO-образ через торрент-сети или сайты в Интернете, т.к. такие файлы могут быть изменены злоумышленниками и содержать вредоносное программное обеспечение. Для обеспечения безопасности используйте только официальный источник **dl.astralinux.ru** или ссылку из личного кабинет Астра.

2.3.1.2 Установка операционной системы

Загрузите виртуальную машину и установите операционную систему с графическим окружением и максимальным уровнем защищенности «Смоленск»:

- Именем компьютера оставьте **astra** по умолчанию.
- Локальным администратором укажите пользователя **localadmin**.
- В качестве пароля локального администратора установите значение **Pa$$w0rd**.
- Настройку сети пропустите.

- При разметке диска выберите пункт **Авто – использовать весь диск и настроить LVM** (потребуется в будущем при обновлении ОС до версии ALSE 1.8+) и схему разметки **Все файлы в одном разделе**.

- Версию ядра оставьте **6.1-generic** по умолчанию, варианты hardened (с усиленной самозащитой) и lowlatency (с увеличенной до 1000 Гц частотой переключения задач) продуктом ALD Pro не поддерживаются.

- Из пакетов программ по умолчанию можно исключить, например, **Средства работы с графикой**, но обязательно оставьте **Офисные приложения**, так как в дальнейшем нам потребуется LibreOffice для просмотра CSV-файлов в табличном виде. Добавьте также пакет **Средство удаленного подключения SSH**, если вы хотите подключаться к компьютеру удаленно.

- При выборе дополнительных параметров укажите уровень защищенности **Смоленск** и оставьте включенными флажки **Мандатный контроль целостности** и **Мандатное управление доступа**. В тестовых средах для удобства работы можно отключить флажок **Запрос пароля для команды sudo**.

- При настройке GRUB используйте тот же пароль **Pa$$w0rd**.

Для получения дополнительной информации об установке системы см. *Модуль 2. Установка ОС Astra Linux*.

После первой загрузки системы вы можете проверить ее версию и уровень защищенности, используя следующие команды:

```
cat /etc/astra/build_version
sudo astra-modeswitch getname
```

Результат выполнения команд выделен цветом:

```
localadmin@astra:~$ cat /etc/astra/build_version
1.7.6.11

localadmin@astra:~$ sudo astra-modeswitch getname
[sudo] пароль для localadmin:
maximum(smolensk)
```

Кратко о работе утилиты sudo

Утилита sudo похожа на команду `Run as...` из Windows и позволяет запускать команды из-под пользователя с повышенными привилегиями.

По умолчанию при первом обращении к утилите `sudo` система потребует ввести пароль, а после успешной аутентификации внесет необходимую информацию в кэш и будет хранить ее на протя-

жении последующих **15 минут**.

В тестовых средах вы можете отключить запрос пароля для команды sudo, увеличить время `timestamp_timeout` в файле `/etc/sudoers` или запустить новую оболочку от имени суперпользователя командой `sudo -i`:

```
localadmin@astra:~$ sudo -i
[sudo] пароль для localadmin:
root@astra:~#
```

После выполнения команды `sudo -i` вы увидите, что имя пользователя сменилось на `root`, а строка приглашения заканчивается символом решетки `#` — это означает, что теперь вы можете запускать команды от имени `root` без использования утилиты `sudo`.

Чтобы выйти из привилегированной сессии, достаточно выполнить команду `exit`:

```
root@astra:~# exit
выход
localadmin@astra:~$
```

О том, что сессия вернулась в обычный режим, будет указывать символ `$` в конце строки приглашения.

Для получения дополнительной информации о работе утилиты sudo см. *Модуль 10. Система аутентификации PAM и управление правами SUDO*.

2.3.1.3 Временная настройка сетевого интерфейса

Для возможности установки пакетов, необходимых для работы гостевых дополнений, сделаем временную настройку сетевого интерфейса через апплет NetworkManager:

- Сделайте клик правой кнопкой мыши по значку сетевого соединения в трее.
- В контекстном меню апплета выберите пункт *Параметры соединений*.
- В окне сетевых соединений двойным кликом по строке *Проводное соединение 1* откройте настройки этого подключения.
- На вкладке *Параметры IPv4* укажите следующие настройки:
  - IP-адрес: **10.0.1.10**.
  - Маска: **24** или **255.255.255.0**.
  - Шлюз: **10.0.1.1**.

- Сервер DNS: **77.88.8.8**.

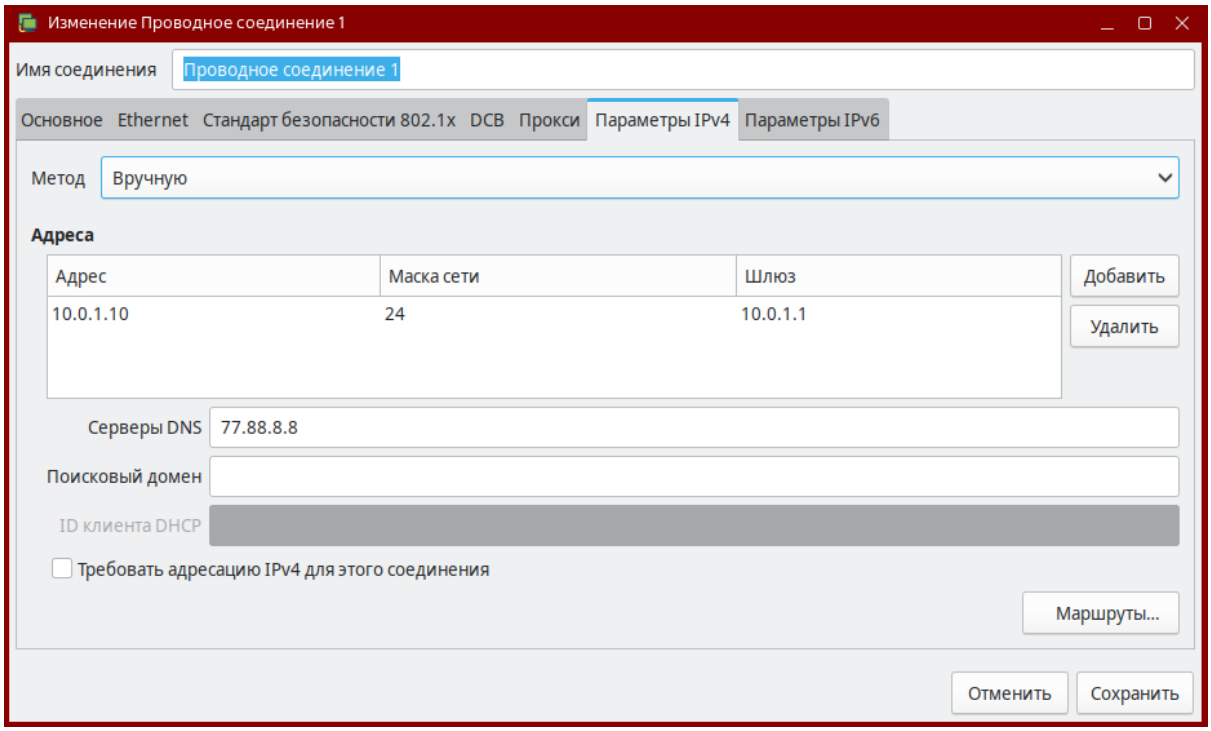


рис. 2.13 – Настройка сетевого интерфейса с помощью апплета Network Manager

Чтобы изменения вступили в силу, выключите и включите обратно поддержку сети в контекстном меню апплета NetworkManager.

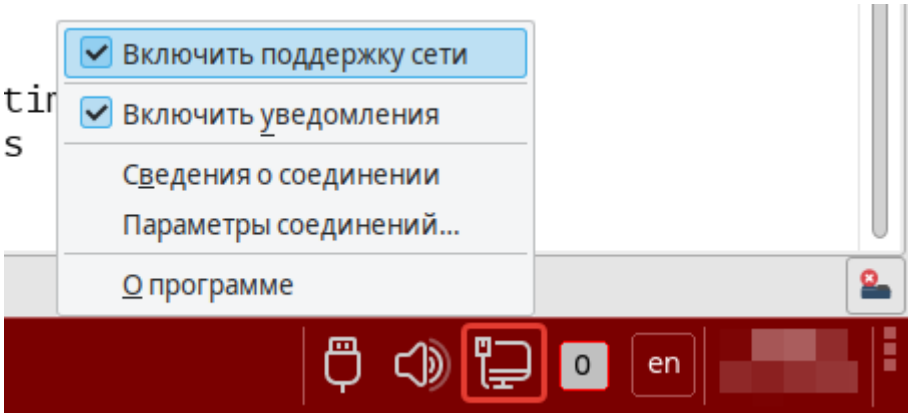


рис. 2.14 – Включение/отключение поддержки сети

Чтобы убедиться в работе сетевого стека, выполните следующие команды:

```
ip a show dev eth0
ping dl.astralinux.ru -c 4
```

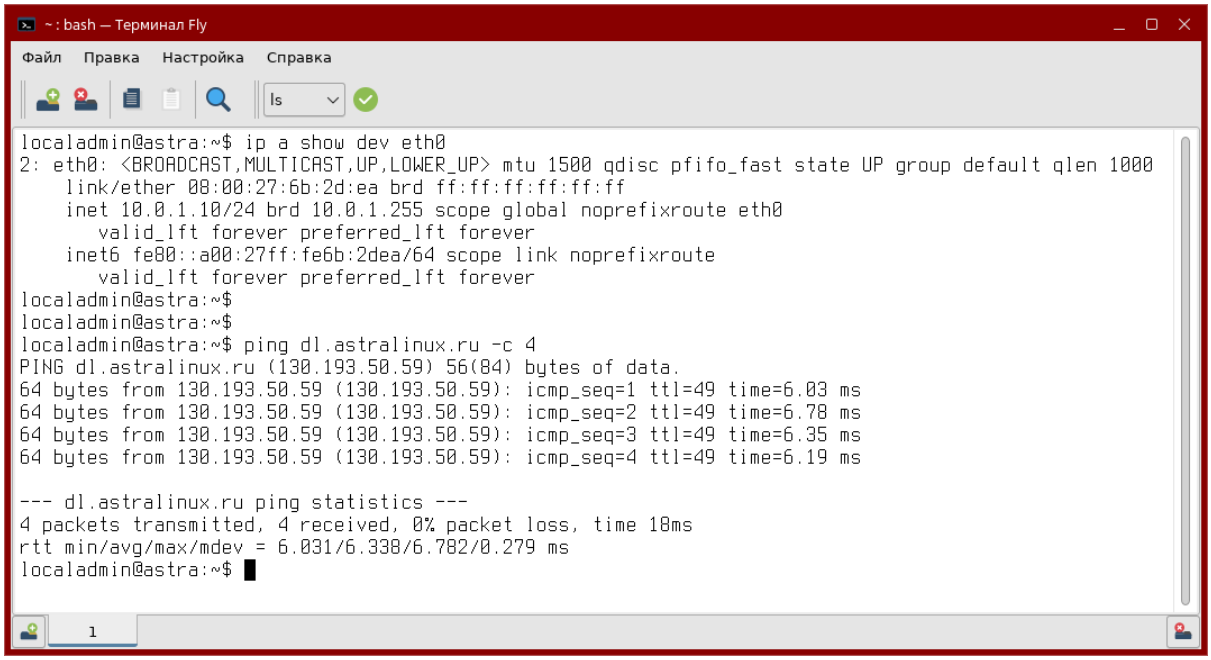


рис. 2.15 – Проверка работы сети

Для получения дополнительной информации по управлению сетевыми интерфейсами см. *Модуль 20. Сеть и стек TCP/IP, управление сетевыми интерфейсами*.

2.3.1.4 Настройка репозиториев операционной системы

Файлы программ для Linux объединяются в пакеты, а пакеты объединяются в репозитории, которые могут распространяться как в виде ISO-образов на диске, так и размещаться на веб-серверах в локальной сети или Интернет. Мы будем использовать два репозитория **base** и **extended**, поэтому напомним их назначение:

- `base` – это базовый репозиторий, который объединяет репозитории `main` и `update`, а также содержит пакеты со средствами разработки, которые потребуются нам, например, для установки гостевых дополнений (gcc и linux-headers).
- `extended` – содержит дополнительное программное обеспечение, которое может функционировать в среде Astra Linux.

Примечание: Следует принимать во внимание, что пакеты из репозиториев `base` и `extended` собираются из исходных кодов и для них устраняются известные уязвимости, но действие сертификатов соответствия на эти репозитории не распространяется, поэтому в продуктивной среде рекомендуется использовать только репозитории `main` и `update`.

С пакетами из репозиториев работает менеджер **apt**, поэтому основным файлом для хранения списка доступных репозиториев является `/etc/apt/sources.list`, а дополнительные списки могут находиться в файлах `*.list` из директории `/etc/apt/sources.list.d/`.

Вы можете редактировать файл `/etc/apt/sources.list` напрямую или запустить редактор с помощью следующей команды, чтобы воспользоваться дополнительными проверками при сохранении изменений:

```
sudo apt edit-sources
```

Для операционной системы ALSE 1.7.6.11 установите следующее содержимое файла `/etc/apt/sources.list`:

```
deb https://dl.astralinux.ru/astra/frozen/1.7_x86-64/1.7.6/repository-base/ 1.7_x86-
↪64 main contrib non-free
deb https://dl.astralinux.ru/astra/frozen/1.7_x86-64/1.7.6/repository-extended/ 1.7_
↪x86-64 main contrib non-free
```

В source-листах каждая строка указывает параметры подключаемого репозитория в следующем формате:

`deb <путь_к_каталогу_репозитория> <код_дистрибутива> <компонент1> <компонентN>`

Где:

- **deb** — указывает на то, что репозиторий соответствует репозиторию бинарных файлов с предварительно скомпилированными пакетами. Для репозиториев с исходными кодами используют «deb-src».
- **uri** — задает адрес репозитория. У сетевых репозиториев адрес начинается с `http(s)://`, адреса локальных копий репозиториев начинаются с `file:/`, а при добавлении репозитория с диска командой `apt-cdrom add` в файле появится строка `cdrom:[]/`.
- **код дистрибутива** — дополняет uri, уточняя необходимый релиз продукта. В одном репозитории могут находиться пакеты сразу для нескольких релизов.
- **компонент** — это группа пакетов, объединенная по условиям использования:
  - **non-free** — группа содержит пакеты, которые не соответствуют принципам свободного ПО, имеют патенты или другие юридические ограничения;
  - **contrib** — группа содержит пакеты, которые сами по себе соответствуют принципам свободного ПО, но зависят от пакетов из группы **non-free** (т. е. не могут без них работать);
  - **main** — группа содержит пакеты свободного ПО, которые не зависят от пакетов из групп **contrib** и **non-free**.

Информацию о пакетах репозитория менеджер **apt** возьмет из файлов Release или InRelease, адреса которых формируются по шаблону `<uri>/dists/<код дистрибутива>/<Release или InRelease>`.

Как вы помните, в адресе **uri** разработчики Astra Linux предлагают использовать ветку **stable**, которая соответствует последней версии операционной системы, но для работы с ALD Pro нужно **обязательно переключиться на ветку frozen** конкретной версии, чтобы гарантировать полную совместимость пакетов. Информация о поддержке продуктом ALD Pro очередных обновлений операционной системы Astra Linux и возможности обновления публикуется в release notes по продукту.

Дополнительная информация о настройке репозиториев для 1.7.6 UU1 и 1.7.6 UU2

При использовании операционной системы Astra Linux нежелательно устанавливать программное обеспечение из репозиториев предыдущих версий, поэтому у каждой версии ОС, включая срочные обновления UU, есть своя собственная ветка frozen:

- Если вы установили Astra Linux SE 1.7.6 UU1 (1.7.6.14) и хотите зафиксировать эту версию, то содержимое файла должно быть следующим:

```
deb https://dl.astralinux.ru/astra/frozen/1.7_x86-64/1.7.6/uu/1/repository-base/ 1.7_
↪x86-64 main contrib non-free
deb https://dl.astralinux.ru/astra/frozen/1.7_x86-64/1.7.6/uu/1/repository-extended/␣
↪1.7_x86-64 main contrib non-free
```

- Если вы установили Astra Linux SE 1.7.6 UU2 (1.7.6.15) и хотите зафиксировать эту версию, то содержимое файла должно быть следующим:

```
deb https://dl.astralinux.ru/astra/frozen/1.7_x86-64/1.7.6/uu/2/repository-base/ 1.7_
↪x86-64 main contrib non-free
deb https://dl.astralinux.ru/astra/frozen/1.7_x86-64/1.7.6/uu/2/repository-extended/␣
↪1.7_x86-64 main contrib non-free
```

После изменения состава репозиториев следует обновить индекс доступных пакетов с помощью команды `sudo apt update`. Эта команда извлекает список пакетов и зависимостей, но еще не обновляет само программное обеспечение. Вот результат ее выполнения:

```
localadmin@astra:~$ sudo apt update
Пол:1 https://dl.astralinux.ru/astra/frozen/1.7_x86-64/1.7.6/repository-base 1.7_x86-
↪64 InRelease [5 306 B]
Пол:2 https://dl.astralinux.ru/astra/frozen/1.7_x86-64/1.7.6/repository-extended 1.7_
↪x86-64 InRelease [9 772 B]
...
Получено 6 589 kB за 1с (5 402 kB/s)
Чтение списков пакетов... Готово
Построение дерева зависимостей
```

(continues on next page)

(продолжение с предыдущей страницы)

```
Чтение информации о состоянии... Готово
Может быть обновлён 1 пакет. Запустите «apt list --upgradable» для показа.
```

Если обратить внимание на последнюю строку в выводе предыдущей команды, то вы увидите, что **может быть обновлён 1 пакет**, который можно посмотреть с помощью команды `sudo apt list --upgradable`:

```
localadmin@astra:~$ sudo apt list --upgradable
Вывод списка... Готово
astra-version/stable 8.1.46+v1.7.6.ext2.1 amd64 [может быть обновлён с: 8.1.43+v1.7.6.
↪11]
N: Есть 1 дополнительная версия. Используйте «-a» для просмотра.
```

Если бы у нас был подключен только репозиторий `base`, то все пакеты имели бы последние версии и ничего обновлять не требовалось бы, но так как мы подключили еще репозиторий `extended`, то менеджер APT предлагает нам обновить пакет `astra-version` из расширенного репозитория. Установка этого пакета не приведет к каким-либо существенным изменениям в системе, просто появится man-справка на саму утилиту astra-version и еще несколько вспомогательных файлов.

Для установки обновлений в Debian и Ubuntu обычно предлагают команду `sudo apt upgrade`, но для системы Astra Linux использовать ее **категорически запрещается**, т.к. эта команда не удаляет устаревшие пакеты, даже если это требуется для обновления приложений, что **может нарушить работу операционной системы**. Поэтому вместо команды `upgrade` следует использовать команду `dist-upgrade` в стратегии `--force-confold`, т.е. при обновлении пакетов менеджер apt должен сохранять содержимое старых конфигурационных файлов, если они были изменены из скриптов установки или вручную:

```
sudo apt dist-upgrade -y -o Dpkg::Options::=--force-confold
```

Результат выполнения команды:

```
localadmin@astra:~$ sudo apt dist-upgrade -y -o Dpkg::Options::=--force-confold
Чтение списков пакетов... Готово
Построение дерева зависимостей
Чтение информации о состоянии... Готово
Расчёт обновлений... Готово
Следующие пакеты устанавливались автоматически и больше не требуются:
libbluetooth3 libmm-glib0 libndp0 libnma-common libnma0 libpkcs11-helper1␣
↪libteamdctl0 libxsettings-client0 openvpn
Для их удаления используйте «sudo apt autoremove».
Следующие пакеты будут обновлены:
astra-version
Обновлено 1 пакетов, установлено 0 новых пакетов, для удаления отмечено 0 пакетов, и␣
↪0 пакетов не обновлено.
Необходимо скачать 8 376 B архивов.
После данной операции объём занятого дискового пространства возрастёт на 20,5 kB.
Пол:1 https://dl.astralinux.ru/astra/frozen/1.7_x86-64/1.7.6/repository-extended 1.7_
↪x86-64/main amd64 astra-version amd64 8.1.46+v1.7.6.ext2.1 [8 376 B]
```

(continues on next page)

(продолжение с предыдущей страницы)

```
Получено 8 376 B за 0c (87,3 kB/s)
(Чтение базы данных ... на данный момент установлено 220649 файлов и каталогов.)
Подготовка к распаковке .../astra-version_8.1.46+v1.7.6.ext2.1_amd64.deb ...
Распаковывается astra-version (8.1.46+v1.7.6.ext2.1) на замену (8.1.43+v1.7.6.11) ...
Настраивается пакет astra-version (8.1.46+v1.7.6.ext2.1) ...
Обрабатываются триггеры для man-db (2.8.5-2+b1) ...
Обрабатываются триггеры для xserver-xorg-core (2:21.1.7-1ubuntu4astra.se28) ...
update exec ids due to /usr/bin changed
```

Примечание: Кстати, продукт ALD Pro для обновления пакетов использует стратегию `--force-confnew`, то есть пакетный менеджер должен переопределять содержимое конфигурационных файлов, даже если они были изменены из скриптов установки или вручную, поэтому для корректного обновления продукта обратитесь к соответствующему разделу руководства администратора.

Чтобы упростить обновление операционной системы, разработчики Astra Linux создали специальную утилиту `astra-update`, которая под капотом запускает обновление с использованием правильной стратегии, и предлагает целый ряд дополнительных функций, поэтому вместо `dist-upgrade` вы можете запустить эту команду со следующими ключами:

```
sudo astra-update -A -r -T
```

Где:

- Ключ `-A` – указывает, что обновления нужно устанавливать полностью автоматически `-I`, включая `-e` и отключая `-d` функции безопасности по мере необходимости.
- Ключ `-r` – указывает, что установка обновлений должна быть ограничена списком репозиториев из файла `/etc/apt/sources.list`.
- Ключ `-T` – указывает, что выполнять проверку технологического диска не требуется, т.к. подключенные репозитории содержат все необходимые пакеты.

Для получения дополнительной информации по работе с пакетами и репозиториями программного обеспечения см. *Модуль 15. Управление ПО в Astra Linux*.

2.3.1.5 Установка гостевых дополнений

Для возможности переноса команд в виртуальную машину через буфер обмена нам потребуется установить гостевые дополнения (guest additions). Подключите и примонтируйте диск с гостевыми дополнениями, выбрав меню *Устройства → Подключить образ диска Дополнений гостевой ОС*:

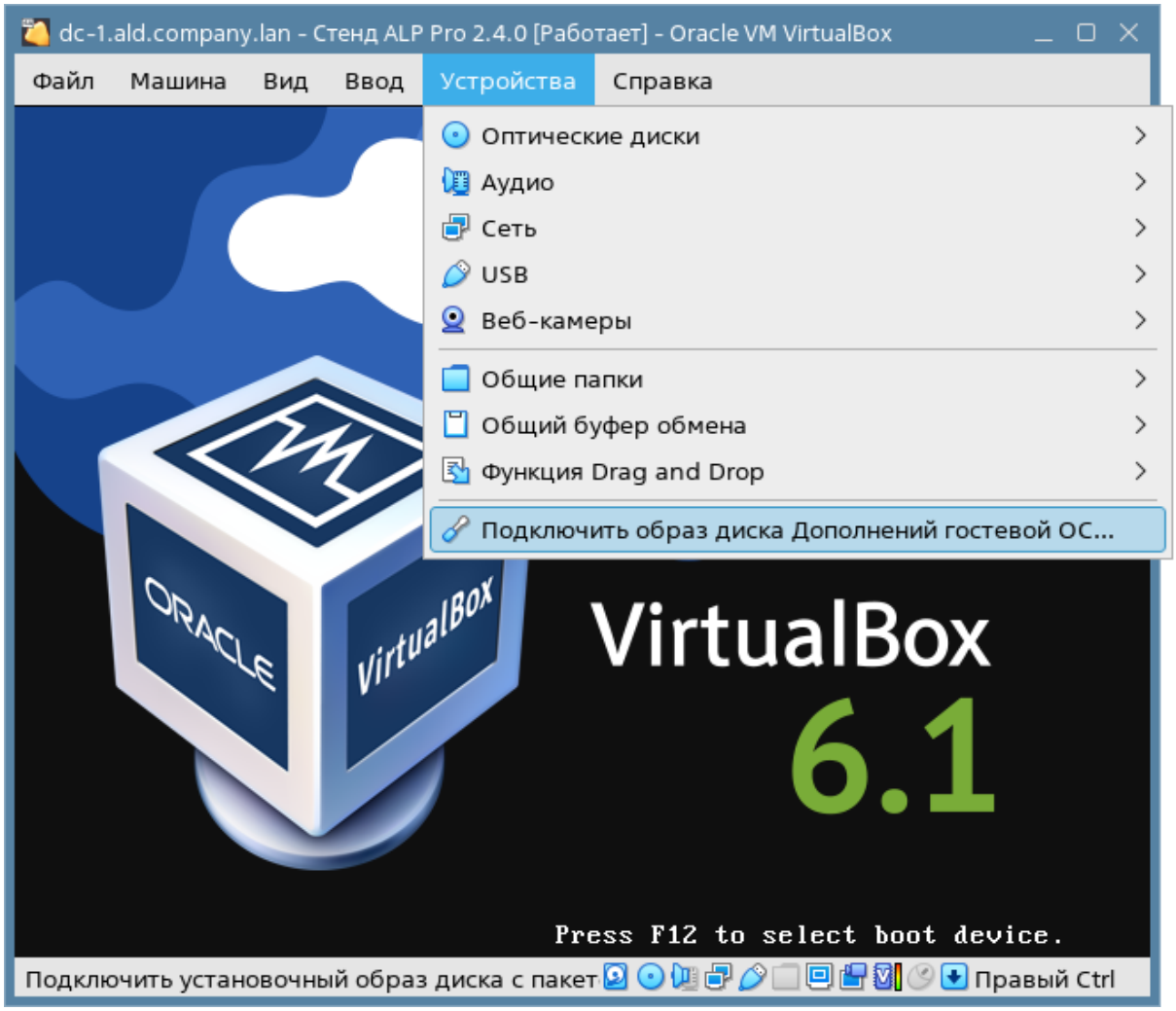


рис. 2.16 – Подключение диска с гостевыми дополнениями

Примечание: Диск с гостевыми дополнениями можно скачать по ссылке с официального сайта.

Установите **VBoxLinuxAdditions** следующими командами:

```
sudo apt install gcc make linux-headers-6.1
sudo sh /media/cdrom0/VBoxLinuxAdditions.run
```

Осталось в настройках машины включить двунаправленный буфер обмена и после перезагрузки системы вы сможете с помощью горячего сочетания клавиш Shift + Ins переносить в окно терминала вашей виртуальной машины текст конфигурационных файлов и команды через общий буфер обмена.

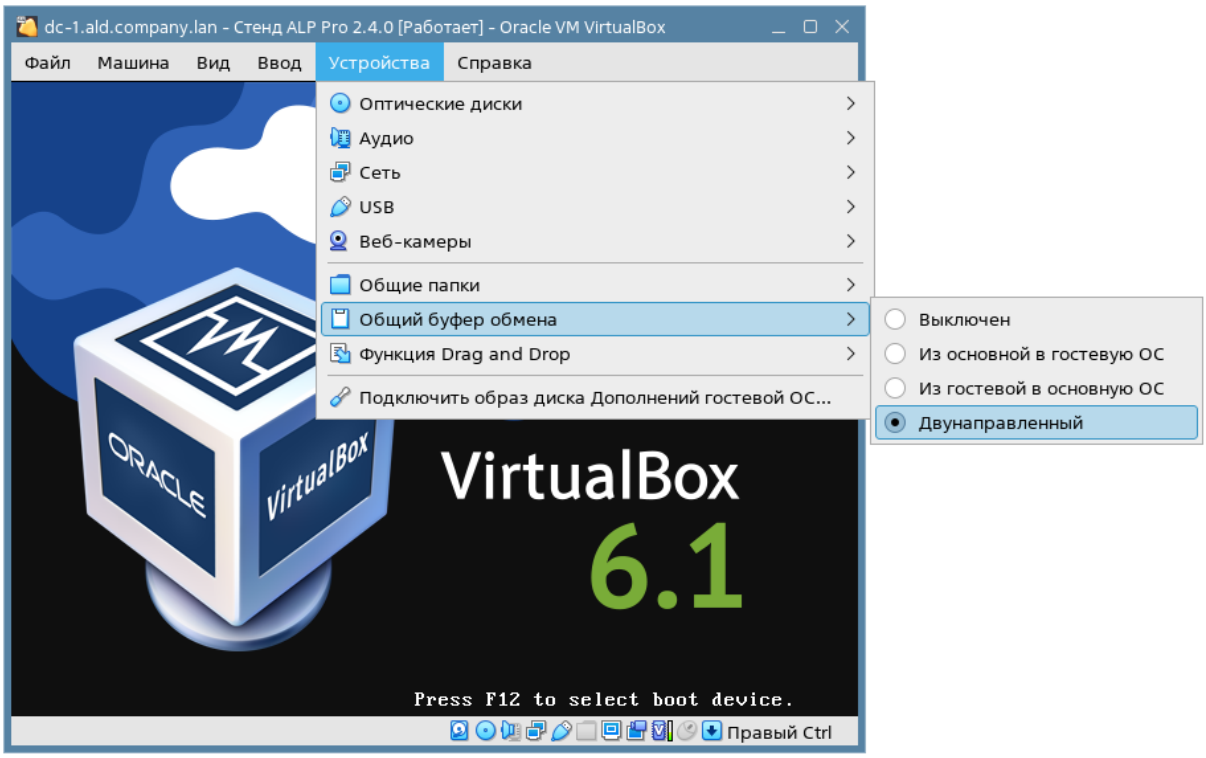


рис. 2.17 – Включение двунаправленного буфера обмена

2.3.1.6 Клонирование шаблона виртуальной машины

Чтобы упростить в дальнейшем развертывание остальных подсистем и рабочих станций, сделаем копию виртуальной машины сразу после установки гостевых дополнений. Перед выполнением клонирования рекомендуется завершить работу операционной системы из меню *Пуск → Завершение работы → Выключение* или командой `poweroff` утилиты `systemctl`:

```
systemctl poweroff
```

Запустите мастер клонирования командой *Машина → Клонировать* из окна VirtualBox. При выполнении клонирования укажите следующие параметры:

- Имя: **Шаблон - Стенд ALD Pro 2.4.0**.
- Путь: **Оставьте по умолчанию**.
- Тип клонирования: **Полное клонирование**.
- Цель клонирования: **Состояние машины**.
- Политика MAC-адреса: **Сгенерировать новые MAC-адреса всех сетевых адаптеров**. Мы, конечно, не планируем загружать виртуальную машину непосредственно из шаблона, но лучше сразу сгенерировать новый MAC-адрес, чтобы полностью исключить вероятность конфликта адресов в будущем.

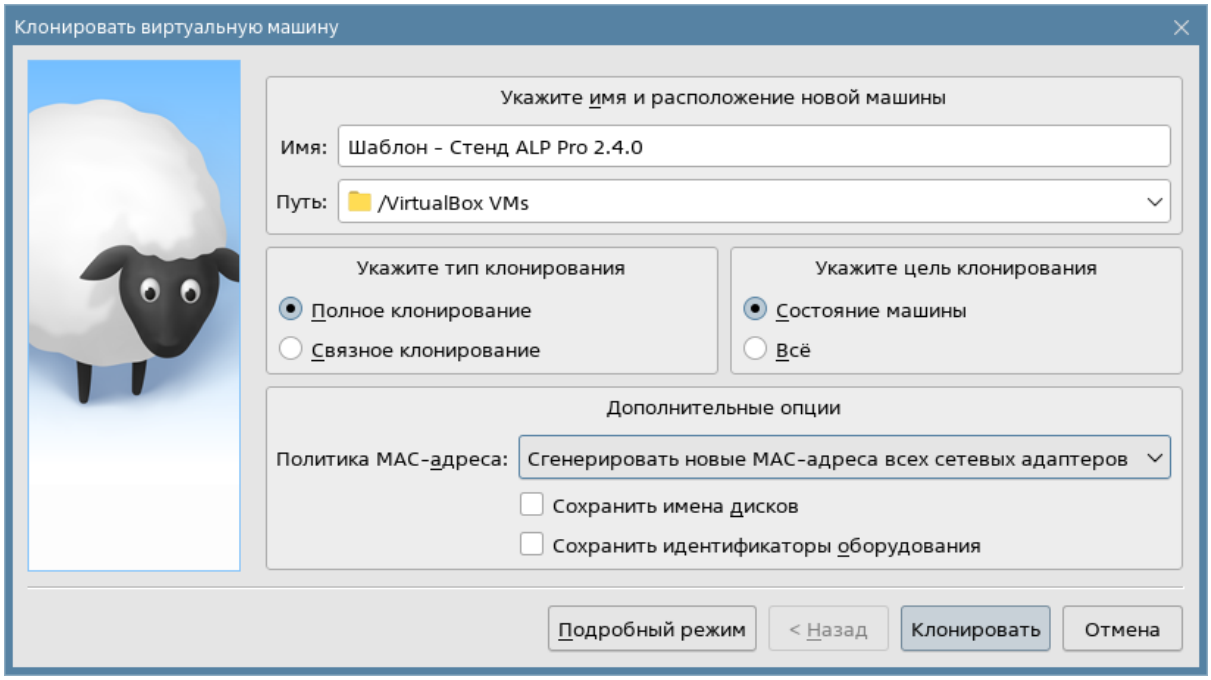


рис. 2.18 – Клонирование шаблона виртуальной машины

2.3.2 Настройка сетевого интерфейса для постоянной работы

Ранее мы уже настраивали сетевой интерфейс для установки гостевых дополнений виртуальной машины, но теперь нужно выполнить настройку для постоянной работы контроллера домена.

Операционная система была установлена с графическим интерфейсом **Fly**, поэтому в качестве менеджера сетевых подключений по умолчанию предлагается использовать службу **NetworkManager**, которая отлично подходит для управления сетевыми настройками на рабочих станциях пользователей, т.к. позволяет автоматически переключаться на доступные Wi-Fi сети и устанавливать VPN-соединения, но на серверах ее использование считается излишним. Отключить службу можно следующими командами:

```
sudo systemctl stop NetworkManager
sudo systemctl disable NetworkManager
sudo systemctl mask NetworkManager
systemctl status NetworkManager
```

После выполнения последней команды `status` в терминале отобразится информация о том, что служба NetworkManager остановлена (inactive), и запустить ее не получится, т.к. она замаскирована (masked):

```
...
localadmin@astra:~@ systemctl status NetworkManager
 o NetworkManager.service
   Loaded: masked (Reason: Unit NetworkManager.service is masked.)
   Active: inactive (dead) since Mon 2024-01-15 11:49:50 MSK; 1s ago
```

(continues on next page)

(продолжение с предыдущей страницы)

```
 Main PID: 509 (code=exited, status=0/SUCCESS)
      CPU: 264ms
```

Старое название службы до версии ALSE 1.7.4

В AstraLinux Special Edition до версии 1.7.4 служба называлась `network-manager` в нижнем регистре через дефис, поэтому для отключения службы нужно было выполнять следующие команды:

```
sudo systemctl stop network-manager
sudo systemctl disable network-manager
sudo systemctl mask network-manager
systemctl status network-manager
```

Отключение апплета NetworkManager

После выключения службы Network Manager значок апплета в трее начнет вводить вас в заблуждение, что у компьютера нет подключения к сети. Вы можете завершить работу аплета и удалить пакет network-manager следующими командами:

```
sudo kill $(pidof nm-applet)
sudo apt remove network-manager
```

Теперь сетевые настройки нужно задавать вручную в файлах `/etc/network/interfaces` и `/etc/resolv.conf`. Файл `interfaces` определяет настройки сетевых интерфейсов для утилит `ifup` и `ifdown`, которые используются службой `networking`. Файл `resolv.conf` определяет настройки для разрешения DNS-имен.

Для настройки сетевого интерфейса вы можете открыть файл `/etc/network/interfaces` напрямую:

```
sudo nano /etc/network/interfaces
```

Содержимое файла после редактирования должно стать следующим (новые строки отмечены цветом):

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*
```

(continues on next page)

(продолжение с предыдущей страницы)

```
# The loopback network interface
auto lo
iface lo inet loopback

# Настройка eth0
auto eth0
iface eth0 inet static
  address 10.0.1.11
  netmask 255.255.255.0
  gateway 10.0.1.1
```

Вот комментарии по использованным инструкциям:

- **auto eth0** — строка, начинающаяся со слова «**auto**», указывает интерфейс, который будет подниматься автоматически при вызове команды `ifup -a`. Посмотреть список доступных интерфейсов можно командой `ip a`, первый сетевой интерфейс обычно имеет идентификатор `eth0`.

- **iface eth0 inet static** — строка со словом «**iface**» начинает группу строк, отвечающих за настройку указанного интерфейса. Следующее слово «**inet/inet6**» указывает, какой протокол будет использоваться: IPv4 или IPv6 соответственно. Параметр «**static**» или «**dhcp**» указывает на способ назначения настроек: вручную или динамически.

  На контроллерах домена настоятельно рекомендуется использовать статичные IP-адреса, потому что контроллер выступает не только в роли LDAP-сервера и центра распространения Kerberos-ключей, но и как DNS-сервер. Если же вам все-таки потребуется перенести контроллер в другую подсеть, то нужно будет поменять A-запись в прямой зоне, PTR-запись в обратной зоне, указать новый IP-адрес на всех компьютерах, где он был задан вручную, в том числе в настройках DHCP-сервера, а затем перезагрузить все хосты, которым этот IP-адрес был выдан в качестве DNS по протоколу DHCP.

- **address**, **netmask**, **gateway** — задают IP-адрес, маску и шлюз по умолчанию для интерфейса, который указан в строке «**iface**» (при условии, что для него выбран способ назначения настроек «**static**»).

Примечание: В некоторых инструкциях в файле `interfaces` задают такие параметры, как **dns-nameservers** и **dns-search**, но они имеют силу только в том случае, если в системе работает служба `resolvconf`, которая переносит эти настройки в файл `/etc/resolv.conf` соответствующим образом. В системе Astra Linux служба `resolvconf` по умолчанию не устанавливается.

Чтобы применить новые настройки, достаточно перезапустить службу networking командой

`systemctl restart networking`. Может потребоваться также очистить старое соединение командой `ip addr flush dev <имя устройства>`:

```
sudo ip addr flush dev eth0
sudo systemctl restart networking
```

Проверить работу сети можно с помощью утилиты `ping`:

```
localadmin@astra:~$ ping 77.88.8.8 -c 4
PING 77.88.8.8 (77.88.8.8) 56(84) bytes of data.
64 bytes from 77.88.8.8: icmp_seq=1 ttl=54 time=91.3 ms
64 bytes from 77.88.8.8: icmp_seq=2 ttl=54 time=91.6 ms
. . .
```

Для возможности обращения компьютера к публичным репозиториям по DNS-именам следует настроить файл `/etc/resolv.conf` для библиотеки **glibc**, которая используется сетевыми приложениями, в том числе утилитами `ping`, `dig` и др. В нашем случае файл уже был настроен ранее службой *NetworkManager* до ее удаления с сервера, поэтому содержимое файла должно быть следующим:

```
localadmin@astra:~$ cat /etc/resolv.conf
# Generated by NetworkManager
nameserver 77.88.8.8
```

Вы можете удалить комментарий от NetworkManager и указать любой другой сервер по своему усмотрению. Корректность настройки файла `/etc/resolv.conf` можно проверить с помощью той же утилиты `ping`, обращаясь к серверу по имени:

```
localadmin@astra:~$ ping dl.astralinux.ru -c 4
PING dl.astralinux.ru (51.250.6.116) 56(84) bytes of data.
64 bytes from 51.250.6.116 (51.250.6.116): icmp_seq=1 ttl=52 time=31.9 ms
64 bytes from 51.250.6.116 (51.250.6.116): icmp_seq=2 ttl=52 time=27.2 ms
. . .
```

Примечание: Если вам потребуется установить продукт в закрытом периметре без доступа в Интернет, то вы можете воспользоваться установочным диском или создать локальный репозиторий.

Напомним еще раз, что для получения дополнительной информации по управлению сетевыми интерфейсами см. *Модуль 20. Сеть и стек TCP/IP, управление сетевыми интерфейсами*.

2.3.3 Настройка репозиториев продукта ALD Pro

Ранее мы уже настроили репозитории для операционной системы ALSE 1.7.6.11, поэтому содержимое файла `/etc/apt/sources.list` должно быть следующим:

```
deb https://dl.astralinux.ru/astra/frozen/1.7_x86-64/1.7.6/repository-base/ 1.7_x86-
↪64 main contrib non-free
deb https://dl.astralinux.ru/astra/frozen/1.7_x86-64/1.7.6/repository-extended/ 1.7_
↪x86-64 main contrib non-free
```

Из основного репозитория операционной системы продукт ALD Pro берет целый ряд системных компонентов, таких как FreeIPA, 389 Directory Server, SSSD и другие, но для установки программного комплекса потребуется настроить так же и собственный репозиторий продукта.

В рамках настоящего курса мы предлагаем устанавливать релиз ALD Pro 2.4.0, который рекомендован к использованию продуктивной среде, так как для него определен длительный период поддержки (Long-Term Support, LTS), что сильно уменьшает накладные расходы команд эксплуатации на своевременную установку обновлений безопасности. Но для поддержания актуальности материалов мы будем делать соответствующие уточнения по новым функциям продукта, которые стали доступны в версиях после 2.4.0.

Дополнительная информация о версионировании ALD Pro

Нумерация релизов ALD Pro ведется в формате «X.Y.Z», который в соответствии с правилами семантического версионирования соответствует следующим требованиям:

- X — очередные обновления (мажорные версии) — включают изменения без обратной совместимости. Переход на следующее очередное обновление может потребовать новой установки продукта с последующим импортом данных из предыдущей версии.

- Y — оперативные обновления (минорные версии) — расширяют функциональность без нарушения обратной совместимости. Обновление с одной оперативной версии на другую возможно без полной переустановки продукта, но в силу архитектурных и функциональных изменений процесс обновления может содержать ряд специфичных действий, которые должны быть отражены в документации.

  Выпуск оперативных обновлений, начиная с релиза 2.0, гарантируется на срок не менее 36 месяцев с даты выхода очередного обновления.

- Z — срочные оперативные обновления (патчи, включенные в продукт) — вносят исправления, связанные с устранением критичных дефектов и угроз информационной безопасности без нарушения обратной совместимости. Установка срочных оперативных обновлений выполняется типовым образом и не требует значительных временных затрат со стороны команды, отвечающей за эксплуатацию системы в продуктивной среде.

  Выпуск срочных оперативных обновлений, начиная с релиза 2.4.0, осуществляется только в рамках оперативных обновлений с длительным сроком поддержки (Long-Term Support, LTS)

и гарантируется на срок не менее 6 месяцев с даты выхода следующего LTS-релиза. Первым LTS-релизом стала версия 2.4, второй LTS-релиз запланирован на версию 2.8, выход которой ожидается в конце 2025 года.

Чтобы подключить репозиторий ALD Pro версии 2.4.0 нужно открыть еще несуществующий файл `/etc/apt/sources.list.d/aldpro.list` на редактирование командой:

```
sudo nano /etc/apt/sources.list.d/aldpro.list
```

И сохранить в нем следующую строку:

```
deb https://dl.astralinux.ru/aldpro/frozen/01/2.4.0 1.7_x86-64 main base
```

После изменение настроек репозиториев не забудьте обновить индекс пакетов:

```
sudo apt update
```

Дополнительная информация о поддержке операционных систем в ALD Pro 2.4.0

Программный комплекс ALD Pro версии 2.4.0 поддерживает следующий перечень операционных систем:

- Серверная часть: ALSE 1.7.5 (включая UU1), 1.7.6 (включая UU1 и UU2).
- Клиентская часть: ALSE 1.7.1, 1.7.2, 1.7.3, 1.7.4, 1.7.5 (включая UU1), 1.7.6 (включая UU1 и UU2) и 1.8.1.

В рамках стратегии мультивендорности в 2025 году планируется реализация клиентской части для операционных систем альтернативных вендоров, таких как РЕД ОС и Альт Линукс, чтобы обеспечить возможность управления конфигурациями этих систем через штатный механизм групповых политик. На данный момент уже доступны опытные образцы (Minimum Viable Product, MVP), которые вы можете запросить через техническую поддержку.

Дополнительная информация по совместимости разных версий продукта

Серверная часть:

- Установку контроллеров домена и подсистем продукта необходимо выполнять только на чистую операционную систему, в которой ранее не были установлены другие сервисы, что связано с большим количеством зависимостей пакетов.

- В нормальном состоянии все контроллеры домена и подсистемы должны работать на одной и той же версии продукта и операционной системы. Если домен находится в процессе обновления, то управление подсистемами должно осуществляться через портал управления обновленного контроллера.

- Пакеты операционной системы и продукта можно обновлять как вместе, так и по отдельности. Обновление операционной системы существенно влияет на функциональные возможности продукта, т.к. ключевые компоненты FreeIPA и 389 Directory Server поставляются через базовый репозиторий операционной системы.

- Схема LDAP-каталога будет изменена во всем домене уже после обновления операционной системы или продукта на первом контроллере домена. Вместе с тем, функциональные возможности нового API будут доступны только при обращении клиентов к обновленным контроллерам и подсистемам.

Клиентская часть:

- Установку клиентской части можно выполнять на любой компьютер без оглядки на другое используемое программное обеспечение. Конфликт зависимостей пакетов минимален.

- На рядовых серверах и рабочих станциях версия продукта и операционной системы может не совпадать с той версией, которая используется на контроллерах домена.

- Если домен находится в процессе обновления, клиент (начиная с версии 2.4.0) загрузит актуальную версию скриптов policy manager из LDAP-каталога и только в части API будет ограничен функциональными возможностями того контроллера домена, который его обслуживает.

Дополнительная информация по приоритетам пакетов

Разработчики Astra Linux предупреждают, что установка пакетов из сторонних репозиториев (включая репозитории Debian) небезопасно и может привести к нарушению работы системы, поэтому, подключая такие репозитории, системные администраторы принимают ответственность полностью на себя. Вместе с тем, для уменьшения вероятности замещения компонентов, реализующих сертифицированные функции безопасности ОС, пакетами из сторонних репозиториев, в Astra Linux устанавливается повышенный приоритет 900 для собственных пакетов:

```
localadmin@astra:~$ cat /etc/apt/preferences.d/smolensk
Package: *
Pin: release n=1.7_x86-64
```

(continues on next page)

(продолжение с предыдущей страницы)

```
Pin-Priority: 900
```

Изначально продукт ALD Pro использовал идентификатор релиза `generic`, поэтому нужно было создавать дополнительный файл для настройки приоритетов собственных пакетов, но с версии 2.2.0 мы перешли на такой же идентификатор `1.7_x86-64`, как у операционной системы, что позволяет использовать для пакетов ALD Pro такой же приоритет, как у пакетов операционной системы, поэтому никаких дополнительных файлов создавать больше не требуется.

2.3.4 Настройка имени контроллера домена

Имя хоста должно быть задано в формате полного имени FQDN (*от англ. Fully Qualified Domain Name*), например, **dc-1.ald.company.lan**. Такой подход принят в Red Hat, и он наследуется при использовании службы каталога ALD Pro на базе FreeIPA.

Для изменения имени хоста в Astra Linux мы рекомендуем использовать утилиту `hostnamectl`, так как после прямого редактирования файла `/etc/hostname` изменения вступают в силу не сразу, а только после перезагрузки.

```
sudo hostnamectl set-hostname dc-1.ald.company.lan
```

Для того, чтобы сервер всегда мог преобразовать свое имя **dc-1.ald.company.lan** в IP-адрес **10.0.1.11**, даже при недоступности DNS-сервиса, в файл `/etc/hosts` нужно добавить строку, связывающую IP-адрес с именем. Вы можете указать не только полное, но и короткое имя **dc-1**, но первым по списку **обязательно** должно идти полное имя, т. к. первое значение считается каноническим именем и будет возвращаться командой `hostname -f`, что требуется для корректной работы скриптов автоматизации.

Еще крайне важно удалить из файла строку, связывающую имя хоста с адресом **127.0.0.1**, т. к. в соответствии с настройками `/etc/gai.conf` адреса локальной петли имеют более высокий приоритет, а нам крайне важно, чтобы имя хоста разрешалось в IP-адрес локальной сети, а не loopback, потому что какие-то из служб в будущем могут прослушивать сетевые порты только на этом адресе.

С учетом указанных замечаний, содержимое файла `/etc/hosts` должно быть следующим:

```
127.0.0.1 localhost.localdomain localhost
10.0.1.11 dc-1.ald.company.lan dc-1
```

Примечание: Обратите внимание, что имя хоста нужно настроить до установки пакетов продукта, т.к. служба Rabbit MQ по завершению установки возьмет текущее имя хоста в качестве имени кла-

стера и последующее повышение роли сервера будет завершаться ошибкой из-за несовпадения имен. В этом случае вы можете, конечно, переустановить пакет `rabbitmq-server` и повторить установку контроллера домена, но лучше все-таки выдержать правильную последовательность действий.

2.3.5 Установка пакетов контроллера домена

Теперь система готова к установке серверной части ALD Pro. Выполните следующую команду:

```
sudo DEBIAN_FRONTEND=noninteractive \
  apt-get install -y -q aldpro-mp aldpro-gc aldpro-syncer
```

Комментарии по использованным инструкциям и параметрам:

- `DEBIAN_FRONTEND` — переменная окружения, которая позволяет изменить режим взаимодействия с пользователем при установке пакетов менеджером APT. Многие приложения на стадии установки уточняют необходимые настройки для последующей работы, что станет помехой для автоматического развертывания. Переключение менеджера пакетов в режим noninteractive позволяет избежать уведомлений от пактов Kerberos, OpenDNSSec и PAM.
- `\` — позволяет перенести текст длинной команды на следующую строку, чтобы исключить появление горизонтальной прокрутки в веб-версии и неправильных переносов при экспорте в PDF-формат.
- `-y` — параметр позволяет автоматически ответить «Да» на все возможные вопросы в ходе установки.
- `-q` — параметр позволяет скрыть сообщения о прогрессе установки, делая журнал более читаемым.
- Устанавливаемые deb-пакеты:
  - `aldpro-mp` — пакет портала управления (management portal), который позволяет через зависимости установить все остальные пакеты, необходимые для работы основных функций контроллера домена ALD Pro.
  - `aldpro-gc` — пакет модуля глобального каталога, который необходим, если планируется использовать двусторонние доверительные отношения с лесом доменов Microsoft Active Directory. Глобальный каталог позволяет обрабатывать запросы от стандартных оснасток Microsoft Windows на поиск пользователей и групп из домена ALD Pro при назначении им прав доступа в домене MS AD.
  - `aldpro-syncer` — пакет модуля синхронизации, который необходим, если планируется использовать расширенные функции интеграции с доменом Microsoft Active Directory.

Он позволяет получить один и тот же состав пользователей, групп и подразделений в двух доменах, синхронизируя все основные атрибуты объектов, включая пароли пользователей. Это дает возможность предоставить пользователям доступ по логину/паролю к любым информационным системам, вне зависимости от того, через контроллер какого домена информационная система проверяет аутентичность пользователей.

Прежде чем продолжить, проверьте журнал пакетного менеджера `/var/log/apt/term.log` на предмет ошибок. Если установка прошла корректно, то следующая команда не должна показать никаких строк:

```
sudo grep 'error:' /var/log/apt/term.log
```

2.3.6 Повышение роли сервера до контроллера домена

Теперь все готово для продвижения контроллера (*англ. domain controller promotion*) или, как обычно говорят, для повышения роли сервера до контроллера домена. В ходе этой процедуры выполняется настройка всех компонентов, чтобы сервер начал выполнять функции контроллера домена. В домене MS AD повышение роли сервера выполняется с помощью мастера настройки доменных служб или средства командной строки `Adprep.exe`, а в домене ALD Pro для этого предназначена утилита `aldpro-server-install`.

Утилита ALD Pro представляет собой неинтерактивное консольное приложение, поэтому все параметры являются обязательными. Перед вызовом утилиты рекомендуется отключать запись истории команд, так как в одном из параметров передается пароль администратора в открытом виде:

```
set +o history # Отключаем запись истории команд

sudo aldpro-server-install -n dc-1 -d ald.company.lan -p 'Pa$$w0rd' \
  --ip 10.0.1.11 --setup_gc --setup_syncer --no-reboot

set -o history # Включаем запись истории команд
```

Комментарии по использованным ключам:

- Параметр `-n` (`--host`) — короткое имя хоста без указания полного имени домена, например, **dc-1**. Значение должно быть указано в нижнем регистре.
- Параметр `-d` (`--domain`) — полное имя домена, например, **ald.company.lan**.
- Параметр `-p` (`--admin_pwd`) — пароль, который нужно установить администратору домена `admin` и суперпользователю LDAP-каталога `cn=Directory Manager`. Пароль должен быть длиной не менее 8 символов, текст пароля нужно заключать в одинарные кавычки, чтобы экранировать специальные символы, включая знак доллара, используемый в bash для подстановки значений переменных.

- Параметр `--ip` — IP-адрес контроллера домена, например, **10.0.1.11**. Значение требуется указывать явно, так как на сервере может быть несколько сетевых интерфейсов.
- Ключ `--setup_gc` — указывает, что нужно настроить установленный ранее модуль глобального каталога.
- Ключ `--setup_syncer` — указывает, что нужно настроить установленный ранее модуль синхронизации.
- Ключ `--no-reboot` — отменяет перезагрузку после завершения процедуры настройки. Выполнение скрипта занимает некоторое время, поэтому мы рекомендуем выполнить перезагрузку вручную после ознакомления с результатами его работы в терминале.

Стандартное описание всех параметров скрипта вы можете получить с помощью ключа `-h`.

Примечание: Чтобы при выполнении команды предотвратить ее запись в историю, вам не обязательно полностью отключать механизм регистрации команд, достаточно поставить в начале команды дополнительный символ пробела.

Если же команда все-таки попала в историю, то эту строку можно удалить с помощью другой команды `history -d $(history 1)` или прямым редактированием файла `/root/.bash_history`.

Для получения дополнительной информации по управлению историей см. *Модуль 4. Работа с терминалом и оболочкой Bash*, раздел *Работа с историей команд Bash*.

По завершению установки вы увидите сообщение об успешном выполнении операции:

```
...
------------
Succeeded: 5 (changed=5)
Failed:    0
------------
Total states run:     5
Total run time: 331.382 ms
localadmin@dc-1:~$
```

После повышения роли сервера до контроллера домена вы можете проверить содержимое файла `/etc/resolv.conf` и убедиться, что теперь DNS-запросы обрабатываются собственной службой Bind9:

```
cat /etc/resolv.conf
# auto-generated by IPA installer
search ald.company.lan
nameserver 127.0.0.1
```

Где:

- Строка `search ald.company.lan` — задает DNS-суффикс, используемый при разрешении имен, поэтому при обращении к хосту по короткому имени `dc-1` система будет дополнять его до полного имени `dc-1.ald.company.lan`.
- Строка `nameserver 127.0.0.1` — указывает, что разрешение имен должно выполняться через локальную службу ISC Bind, которая обрабатывает запросы, поступающие на 53-й порт сервера.

Контроллер домена готов к работе, осталось только перезагрузить сервер:

```
sudo reboot
```

После загрузки контроллера выберите в первом выпадающем списке значение *ald.company.lan* вместо *Этот компьютер* и войдите в систему из-под учетной записи доменного администратора `admin` с высоким уровнем целостности `63`, который будет предложен по умолчанию:

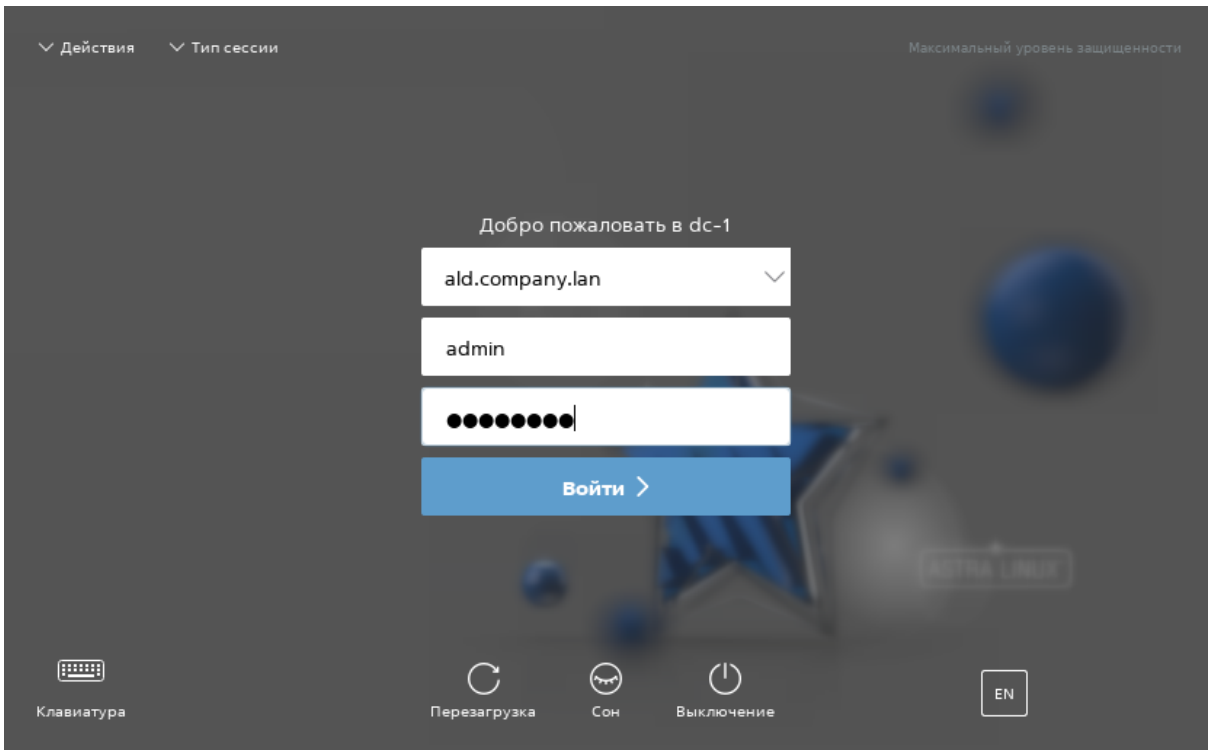


рис. 2.19 – Вход в систему под доменной учетной записью

2.3.7 Проверка базовых функций домена

2.3.7.1 Проверка доменных служб

Обратите внимание, что во время загрузки контроллера домена окно для входа пользователя в систему может отобразиться раньше, чем станут доступны доменные сервисы, поэтому:

- Если в первом выпадающем списке формы для входа вместо имени домена `ald.company.lan`

отображается строка `Ожидание ответа домена...`, то пытаться входить в систему доменным пользователем преждевременно. Подождите пару минут и попробуйте еще раз.

- Если при входе доменного пользователя в систему вы получите ошибку «Вход не выполнен», то подождите пару минут и попробуйте войти в систему еще раз, прежде чем начинать отладку.

В случае, если вход в операционную систему доменным пользователем окажется все-таки невозможен, войдите в систему локальным администратором `localadmin` и проверьте в первую очередь доступность клиентской службы SSSD, которая обеспечивает работу операционной системы в составе домена:

```
systemctl status sssd
```

Результат проверки в нормальном состоянии:

```
localadmin@dc-1:~$ systemctl status sssd
o sssd.service - System Security Services Daemon
   Loaded: loaded (/lib/systemd/system/sssd.service; enabled; vendor preset: enabled)
   Active: active (running) since Sat 2025-04-07 09:49:16 MSK; 17min ago
...
```

При необходимости перезапустите службу командой `sudo systemctl restart sssd`, повторите вход в систему доменным пользователем еще раз и проверьте новые записи в журналах `/var/log/auth.log`` и `/var/log/messages`.

Далее проверьте на сервере состояние доменных служб утилитой `ipactl`:

```
sudo ipactl status
```

Результат проверки в нормальном состоянии:

```
localadmin@dc-1:~$ sudo ipactl status
Directory Service: RUNNING
krb5kdc Service: RUNNING
kadmin Service: RUNNING
named Service: RUNNING
httpd Service: RUNNING
ipa-custodia Service: RUNNING
smb Service: RUNNING
winbind Service: RUNNING
ipa-otpd Service: RUNNING
ipa-dnskeysyncd Service: RUNNING
ipa: INFO: The ipactl command was successful
```

Примечание: Для управления базовыми сервисами домена утилиты `ipactl` будет вполне достаточно, но в ALD Pro есть также и дополнительная утилита `aldproctl`, которая позволяет управлять состоянием собственных служб, например, модулем глобального каталога или синхронизации. С

помощью этой утилиты вы можете отправлять команды в том числе и конкретным службам, для этого нужно использовать ключ `--service`:

```
sudo aldproctl status --service=globalcatalog
```

При необходимости перезапустите доменные службы командой `sudo ipactl restart`, повторите вход в систему доменным пользователем еще раз и проверьте новые записи в журналах `/var/log/auth.log` и `/var/log/messages`.

2.3.7.2 Проверка Kerberos-аутентификации

При успешном входе в систему доменным пользователем в связке ключей должен появиться **TGT-билет**, что можно проверить с помощью команды `klist`:

```
admin@dc-1:~$ klist
Ticket cache: KEYRING:persistent:1902000000:krb_ccache_BUoysxG
Default principal: admin@ALD.COMPANY.LAN

Valid starting       Expires              Service principal
07.04.2025 12:29:42  08.04.2025 12:01:56  krbtgt/ALD.COMPANY.LAN@ALD.COMPANY.LAN
```

Для уничтожения билетов используется команда `kdestroy`, после выполнения которой мы увидим, что в связке ключей нет больше кэша:

```
admin@dc-1:~$ kdestroy
admin@dc-1:~$ klist
klist: Credentials cache keyring 'persistent:1902000000:krb_ccache_BUoysxG' not found
```

Для запроса **нового TGT-билета** на доменного пользователя `admin` воспользуемся утилитой `kinit`. Как мы видим, после прохождения успешной Kerberos-аутентификации нам был выдан новый билет `krbtgt/ALD.COMPANY.LAN`:

```
admin@dc-1:~$ kinit admin
Password for admin@ALD.COMPANY.LAN: ********

admin@dc-1:~$ klist
Ticket cache: KEYRING:persistent:1902000000:krb_ccache_BUoysxG
Default principal: admin@ALD.COMPANY.LAN

Valid starting       Expires              Service principal
07.04.2025 12:31:42  08.04.2025 11:59:45  krbtgt/ALD.COMPANY.LAN@ALD.COMPANY.LAN
```

Дополнительная информация об алиасе root принципала admin

Обратите внимание, что при запуске команды `kinit` без указания имени пользователя утилита

попытается взять имя пользователя сначала из связки ключей, а если кэш будет удален командой `kdestroy`, то переключится на имя текущего пользователя Linux.

По указанной причине начинающие администраторы иногда выполняют аутентификацию из-под принципала `root@ALD.COMPANY.LAN`, который является алиасом для учетной записи `admin@ALD.COMPANY.LAN`, что может вызывать различные коллизии. Например, до версии 2.4.0 это приводило к тому, что администратор не мог выполнить прозрачную Kerberos-аутентификацию на портале.

Чаще всего это происходит, когда администратор после команды `kdestroy` запускает аутентификацию командой `kinit` без указания имени пользователя, но с добавлением `sudo`:

```
admin@dc-1:~$ kdestroy
admin@dc-1:~$ sudo kinit
Password for root@ALD.COMPANY.LAN:

admin@dc-1:~$ klist
Ticket cache: KEYRING:persistent:1965600000:krb_ccache_o2glwGp
Default principal: root@ALD.COMPANY.LAN

Valid starting       Expires              Service principal
07.04.2025 13:16:02  08.04.2025 12:31:18  krbtgt/ALD.COMPANY.LAN@ALD.COMPANY.LAN
```

Эта ошибка возникает также и при выполнении `kinit` без `sudo`, когда администратор работает в сессии суперпользователя `root`, в которую можно войти с помощью команды `sudo -i`.

2.3.7.3 Проверка портала управления

Для входа на портал управления откройте на контроллере домена страницу https://dc-1.ald.company.lan в браузере Mozilla Firefox. Вы сможете войти в систему как по паролю, так и через прозрачную Kerberos-аутентификацию, если в связке ключей будет действительный TGT-билет.

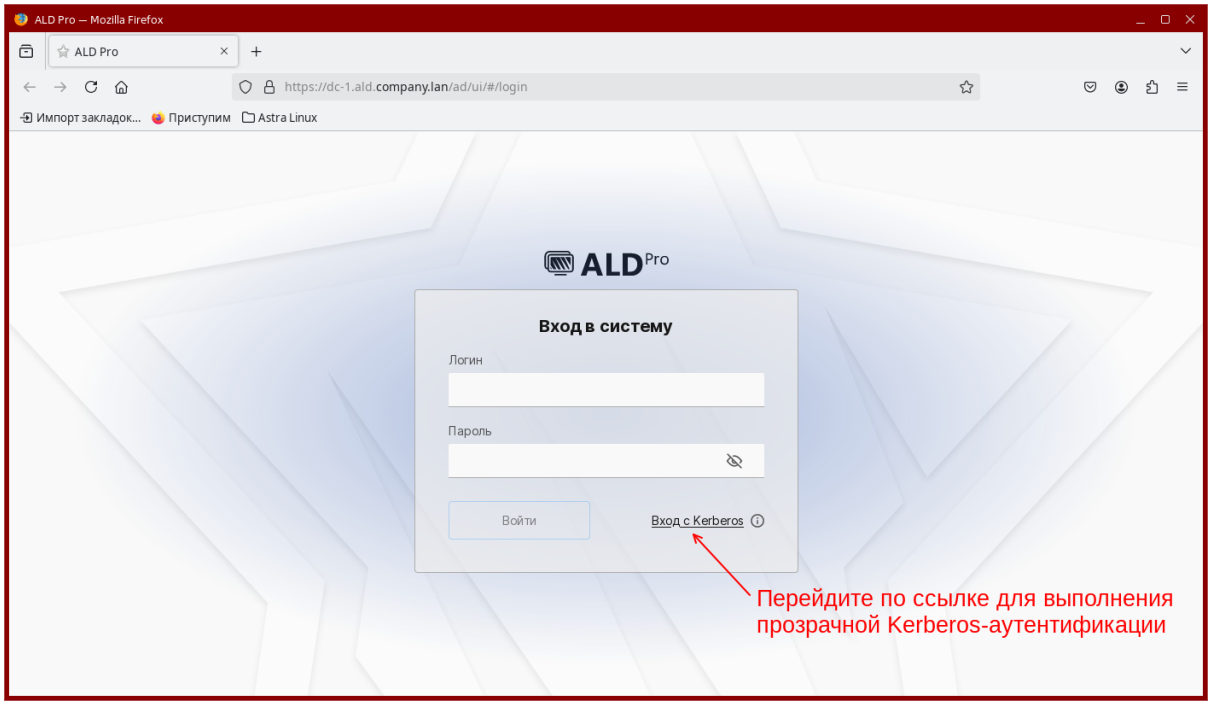


рис. 2.20 – Прозрачная Kerberos-аутентификация на портале

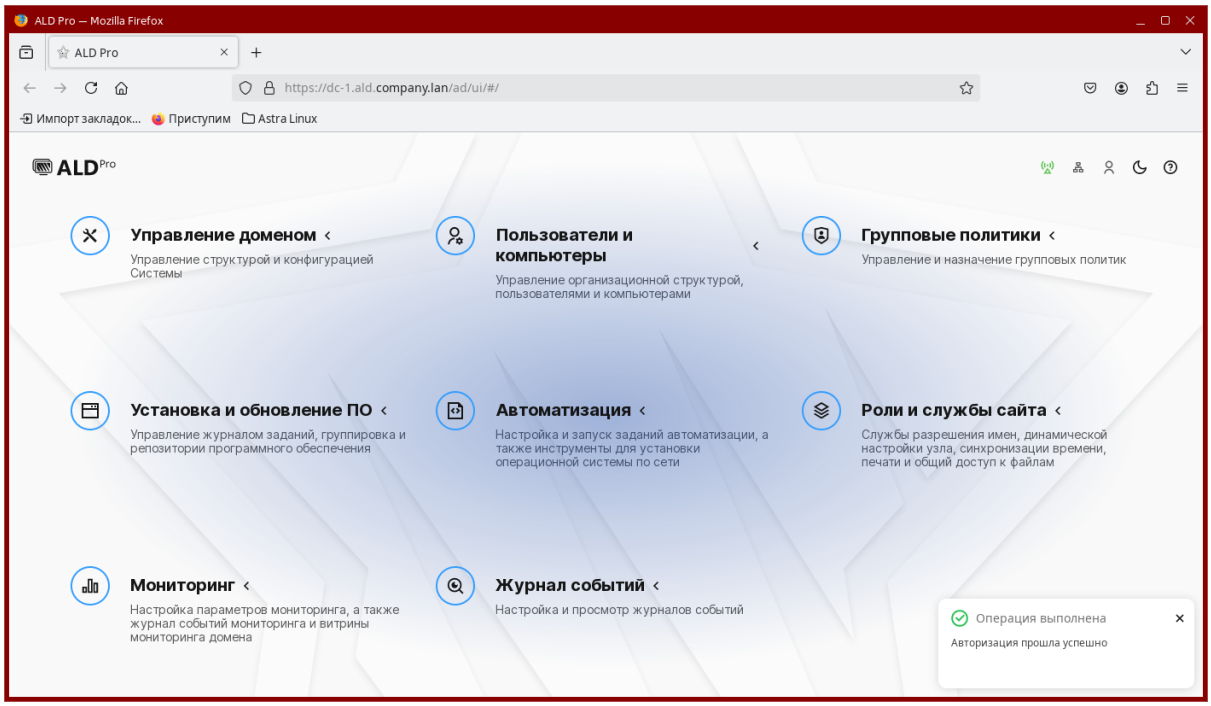


рис. 2.21 – Портал управления ALD Pro после успешного входа

Чувствительность к регистру в имени пользователя

В версии 2.4.0 веб-форма для входа на портал чувствительна к регистру символов в части имени пользователя, поэтому вам нужно вводить значение `admin` и не использовать другие варианты

написания, такие как `Admin` или `ADMIN`. Учитывая, что в службе каталога FreeIPA все учетные записи создаются в нижнем регистре, в версии ALD Pro 2.5.0 указанное ограничение было снято.

Дополнительная информация для настройки домашней страницы

В версии 2.4.0 домашней страницей FireFox по умолчанию устанавливается короткое имя хоста `https://dc-1`, которое не рекомендуется использовать для работы с порталом, т.к. в этом случае веб-страница не сможет пройти проверку SSL-сертификата и не будет работать Kerberos-аутентификация.

В версии ALD Pro 2.5.0 указанная настройка уже исправлена, но для любой предыдущей версии вы можете сделать это вручную. Для этого достаточно будет исправить содержимое файла `/usr/lib/firefox/distribution/policies.json`, например, следующей командой:

```
sudo sed -i 's/dc-1\|dc-1.ald.company.lan/dc-1.ald.company.lan/g' \
  /usr/lib/firefox/distribution/policies.json
```

Содержимое файла после коррекции должно быть следующим:

```
admin@dc-1:~$ cat /usr/lib/firefox/distribution/policies.json
{
        "policies": {
              "BlockAboutAddons": true,
              "BlockAboutConfig": true,
              "Certificates": {
                       "ImportEnterpriseRoots": true,
                       "Install": ["/etc/ipa/ca.crt"]
              },

              "Homepage": {
                       "URL": "https://dc-1.ald.company.lan/",
                       "Locked": true,
                       "StartPage": "homepage-locked"
              }
        }
}
```

Дополнительная информация о входе на портал с компьютеров вне домена

Если вы откроете портал управления на компьютере, который не является участником домена ALD Pro, то вам потребуется согласиться с риском использования самоподписанного сертификата, и Kerberos-аутентификация работать не будет, потребуется вводить логин и пароль вручную.

Если в момент прозрачной аутентификации в связке ключей не будет действительного TGT-билета, то появится сообщение «Аутентификация с Kerberos не удалась» и нужно будет сначала выполнить полную Kerberos-аутентификацию в терминале с помощью утилиты `kinit`, а затем повторить попытку еще раз.

После успешной прозрачной аутентификации на портале управления в связке ключей текущего пользователя появится сервисный билет на доступ к службе `HTTP/dc-1.ald.company.lan`.

```
admin@dc-1:~$ klist
Ticket cache: KEYRING:persistent:1902000000:krb_ccache_BUoysxG
Default principal: admin@ALD.COMPANY.LAN

Valid starting       Expires              Service principal
07.04.2025 12:32:23  08.04.2025 11:59:45  HTTP/dc-1.ald.company.lan@ALD.COMPANY.LAN
07.04.2025 12:31:42  08.04.2025 11:59:45  krbtgt/ALD.COMPANY.LAN@ALD.COMPANY.LAN
```

При повторном входе на портал, если cookie не были удалены и срок жизни предыдущей пользовательской сессии еще не истек, доступ к порталу будет предоставлен без повторной Kerberos-аутентификации, поэтому валидный TGT-билет не потребуется и сервисный билет на доступ к HTTP-службе в связке ключей не появится.

2.3.8 Дополнительная настройка DNS-службы

2.3.8.1 Отключение DNSSEC и запрета рекурсивных запросов

В качестве DNS-сервера на контроллере домена выступает служба BIND9 (named-pkcs11), основным конфигурационным файлом которой является `/etc/bind/named.conf`. Дополнительные настройки для службы каталога подключаются из файла `/etc/bind/ipa-options-ext.conf` с помощью параметра `include`. В этом файле нам нужно будет отключить проверку dnssec и разрешить рекурсивные DNS-запросы.

По умолчанию служба BIND9 для проверки ответов использует механизм DNSSEC, но его лучше отключить, т. к. технология все еще не получила широкого распространения, и ошибки в настройках зон могут приводить к невозможности разрешения имен. Для этого в файле `/etc/bind/ipa-options-ext.conf` параметру «dnssec-validation» нужно установить значение «no».

Еще одной особенностью настроек BIND9 по умолчанию является запрет на обработку рекурсивных DNS-запросов от клиентов из других сетей, что сделано для предотвращения DDoS-атак с DNS-усилением. Мы можем настроить доверенные сети для BIND9 в файле `ipa-ext.conf` или просто установить значение «any» для параметров «allow-recursion» и «allow-query-cache» в файле `ipa-options-ext.conf`.

Чтобы настроить BIND9 в соответствии с указанными выше рекомендациями, откройте файл `/etc/bind/ipa-options-ext.conf` на редактирование командой:

```
sudo nano /etc/bind/ipa-options-ext.conf
```

И приведите его содержание к следующему виду:

```
dnssec-validation no;
allow-recursion { any; };
allow-query-cache { any; };
```

Для применения изменений необходимо перезапустить DNS-службу на контроллере домена:

```
sudo systemctl restart bind9-pkcs11.service
```

2.3.8.2 Настройка глобального перенаправления DNS-запросов

Служба BIND9 может самостоятельно выполнять рекурсивное разрешение имен, последовательно обращаясь ко всем DNS-серверам, обслуживающим зону, начиная с корневых серверов (см. файлы `/etc/bind/named.conf.default-zones` и `/usr/share/dns/root.hints`), но этот механизм потребляет дополнительные ресурсы, поэтому для быстрого разрешения публичных DNS-имен рекомендуется использовать глобальный перенаправитель. Эту опцию можно настроить на портале управления в меню *Роли и службы сайта → Служба разрешения имен → Глобальная конфигурация DNS*.

Чтобы добавить перенаправитель, нажмите кнопку *Добавить* в списке глобальных перенаправителей и введите адрес DNS-сервера от Яндекс **77.88.8.8**. В качестве политики перенаправления выберите «**Только перенаправлять**», в этом случае BIND9 будет запрашивать ответ только у перенаправителя. Если же вы укажите «**Сначала перенаправлять**», то в случае неуспешного обращения к серверам перенаправления BIND9 будет пататься разрешить запрос самостоятельно. По завершению редактирования настроек нажмите кнопку *Сохранить*, чтобы изменения вступили в силу, см. рис. 2.22 (**1**).

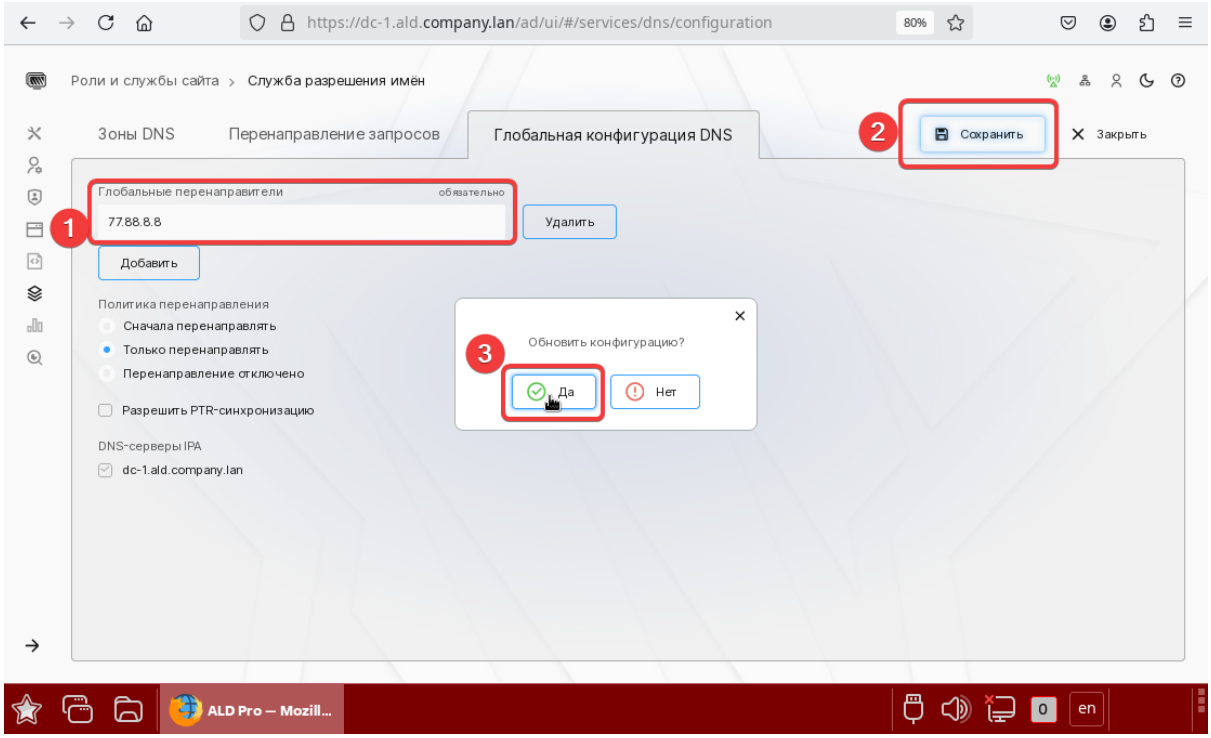


рис. 2.22 – Глобальная конфигурация DNS

Проверить настройки DNS-службы можно из командной строки:

```
ipa dnsconfig-show
```

Результат выполнения команды:

```
admin@dc-1:~$ ipa dnsconfig-show
Глобальные перенаправители: 77.88.8.8
Политика перенаправления: only
Разрешить PTR-синхронизацию: FALSE
DNS-серверы IPA: dc-1.ald.company.lan
```

Для проверки того, как именно BIND9 выполняет разрешение имен (самостоятельно или через глобальный перенаправитель), нам потребуется какой-нибудь сниффер сетевого трафика, например, Wireshark или TCP Dump.

Установим пакет tcpdump командой `sudo apt install tcpdump` и запустим в отдельном окне терминала перехват всех пакетов, в которых фигурирует 53-й порт:

```
sudo tcpdump port 53
```

Осталось проверить, что будет происходить при преобразовании DNS-имени **dl.astralinux.ru** в IP-адрес, например, с помощью утилит `host`, `ping`, `nslookup` или `dig`:

```
host dl.astralinux.ru
```

В окне сниффера можно увидеть, что запрос был направлен на глобальный перенаправитель **dns.yandex.ru**, и в ответ был получен IP-адрес **130.193.50.59**:

```
admin@dc-1:~$ sudo tcpdump port 53
...
15:36:12.154912 IP dc-1.ald.company.lan.41938 > dns.yandex.ru.domain: 5409+% [1au]␣
↪PTR? 8.8.88.77.in-addr.arpa. (63)
15:36:12.160096 IP dns.yandex.ru.domain > dc-1.ald.company.lan.40646: 62590 1/0/1 A␣
↪130.193.50.59 (61)
...
```

2.3.9 Установка глобального каталога

Как уже было сказано ранее, модуль глобального каталога необходим, если планируется использовать двусторонние доверительные отношения с лесом доменов Microsoft Active Directory. Глобальный каталог позволяет обрабатывать запросы от стандартных оснасток Windows на поиск пользователей и групп из домена ALD Pro при назначении им прав доступа в домене MS AD.

Мы установили модуль глобального каталога во время продвижения контроллера домена, для этого достаточно было установить пакет `aldpro-gc` и добавить ключ `--setup_gc` при запуске утилиты `aldpro-server-install`. Если же вы не установили модуль глобального каталога сразу, то его можно добавить позднее на любой контроллер домена через портал управления на странице *Роли и службы сайта → Служба глобального каталога*. Контроллер, которому будет назначена новая роль, автоматически обновит свою конфигурацию в течении получаса.

Чтобы не ждать до следующего обновления конфигурации, вы можете форсировать установку роли на сервере с помощью следующих команд:

```
sudo DEBIAN_FRONTEND=noninteractive apt-get install -q -y aldpro-gc
sudo aldpro-gc --action install
```

Но обратите внимание на следующие два тезиса:

- Просто выполнения команды `aldpro-gc --action install` для установки модуля глобального каталога недостаточно и нужно обязательно назначить роль глобального каталога через портал управления, иначе при очередном обновлении конфигурации сервера модуль будет автоматически удален из системы.

- Модуль глобального каталога можно установить также в порядке обновления командой `aldpro-server-install --update` с помощью ключа `--setup_gc`. Но такой способ не рекомендуется, т.к. установка в этом случае будет выполняться значительно дольше, и при отсутствии должной внимательности вы можете по ошибке удалить модуль синхронизации, если не укажете его ключ в команде.

Управлять службой глобального каталога можно с помощью команд `start`, `stop`, `restart` и

`status` утилиты `aldproctl`:

```
admin@dc-1:~$ sudo aldproctl status --service=globalcatalog
Сервис dirsrv@GLOBAL-CATALOG: ЗАПУЩЕН

admin@dc-1:~$ sudo aldproctl restart --service=globalcatalog
Сервис dirsrv@global-catalog: ПЕРЕЗАПУЩЕН
```

Просмотреть записи глобального каталога можно следующим запросом:

```
admin@dc-1:~$ ldapsearch -QLLL -H ldap://dc-1.ald.company.lan:3268 cn
...
dn: cn=admin,cn=users,dc=ald,dc=company,dc=lan
cn: admin
...
```

Более подробно работу с модулем глобального каталога мы рассмотрим в модуле *Модуль 9. Миграция и гибридное развертывание с MS AD*.

2.3.10 Установка модуля синхронизации

Как уже было сказано ранее, модуль синхронизации необходим, если планируется использовать расширенные функции интеграции с доменом Microsoft Active Directory.

Он позволяет получить один и тот же состав пользователей, групп и подразделений в двух доменах, синхронизируя все основные атрибуты объектов, включая пароли пользователей. Это дает возможность предоставить пользователям доступ по логину/паролю к любым информационным системам, вне зависимости от того, через контроллер какого домена информационная система проверяет аутентичность пользователей.

Вы можете установить модуль синхронизации сразу во время продвижения контроллера домена, для этого достаточно установить пакет `aldpro-syncer` и добавить ключ `--setup_syncer` при запуске утилиты `aldpro-server-install`. Если вы не установили модуль синхронизации сразу, то это можно сделать в порядке обновления. Для этого сначала нужно установить необходимые deb-пакеты:

```
sudo DEBIAN_FRONTEND=noninteractive apt-get install -q -y aldpro-mp aldpro-syncer
```

Затем требуется выполнить обновление системы командой `aldpro-server-install --update` со всеми необходимыми ключами:

```
set +o history # Отключаем запись истории команд

sudo aldpro-server-install --update -n dc-1 -d ald.company.lan -p 'Pa$$w0rd' \
  --ip 10.0.1.11 --setup_syncer --no-reboot

set -o history # Включаем запись истории команд
```

Если на контроллере домена был установлен также модуль глобального каталога, то при обнов-

лении сервера вам нужно будет добавить ключ `--setup_gc`, иначе подсистема обновления удалит этот модуль с сервера.

2.4 Присоединение компьютера к домену

Из этого раздела вы узнаете, как настроить рабочую станцию и присоединить ее к домену ALD Pro. Для упрощения развертывания стенда мы создали шаблон виртуальной машинры в процессе установки первого контроллера домена, но если у вас нет такой копии виртуальной машины, просто обратитесь к разделу *Подготовка виртуальной машины* выше.

2.4.1 Подготовка виртуальной машины

Для создания рабочей станции выберите виртуальную машину **Шаблон - Стенд ALD Pro 2.4.0** и запустите мастер клонирования командой *Машина → Клонировать* из меню VirtualBox Manager. При выполнении клонирования укажите следующие параметры:

- Имя: **pc-1.ald.company.lan - Стенд ALD Pro 2.4.0**.
- Путь: **Оставьте по умолчанию**.
- Тип клонирования: **Полное клонирование**.
- Цель клонирования: **Состояние машины**.
- Политика MAC-адреса: **Сгенерировать новые MAC-адреса всех сетевых адаптеров**.

После клонирования машины оптимизируйте объем вычислительных ресурсов, изменив требуемые параметры RAM и CPU.

Для возможности работы клонированной машины в одной компьютерной сети с другими копиями нам нужно обязательно назначить машине уникальный MAC-адрес сетевого интерфейса. Мы можем сделать это как в момент клонирования машины, так и после. При использовании NetworkManager иногда может потребоваться также выбрать новое устройство в настройках сетевого подключения после загрузки операционной системы.

Однако настройка машины на этом не заканчивается. Как вы знаете, в Windows нужно обязательно выполнить еще процедуру подготовки sysprep (*англ. system preparation — подготовка системы*), в ходе которой из системы удаляются все сведения, относящиеся к конкретному экземпляру, включая идентификатор безопасности компьютера (SID). Для систем Linux это действие не является обязательным, но лучше все-таки удалить некоторые данные, относящиеся к конкретному экземпляру операционной системы.

Сгенерируем новый идентификатор машины, чтобы при необходимости использования динамической настройки от DHCP она получала собственный IP-адрес:

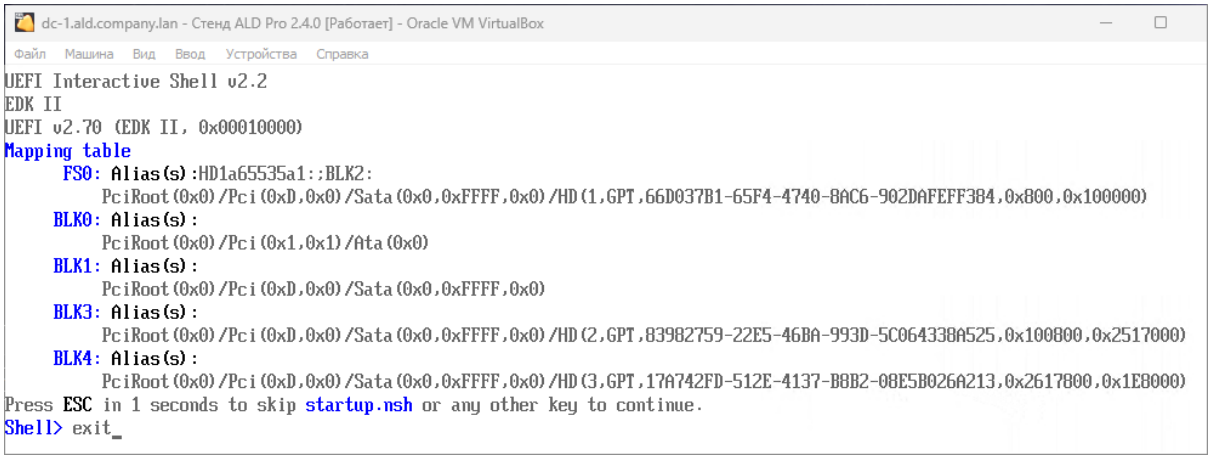
```
sudo rm -fv /etc/machine-id
sudo rm -fv /var/lib/dbus/machine-id
sudo dbus-uuidgen --ensure
sudo systemd-machine-id-setup
```

Сгенерируем новые ssh-ключи, чтобы у всех компьютеров они были разными:

```
sudo rm -fv /etc/ssh/ssh_host_*
sudo ssh-keygen -A -v
```

Исправление порядка загрузки для клонированной машины

При первом включении клонированной машины вы можете столкнуться с ошибкой, возникающей из-за неправильного порядка загрузки:



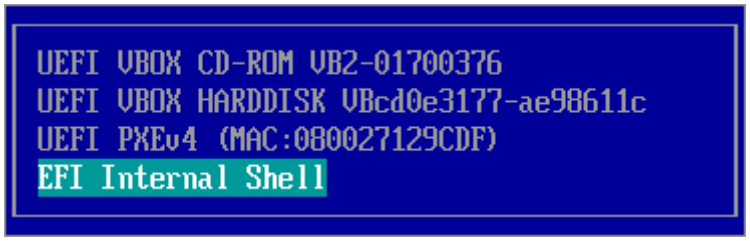
Для устранения проблемы выполните следующие действия:

1. Введите команду `exit` и нажмите клавишу Enter.

2. В открывшемся интерфейсе перейдите в меню *Boot Maintenance Manager* → *Boot Options* → *Change Boot Order* и нажмите Enter.



3. Клавишами + и - задайте правильный порядок устройств для загрузки, при котором CD-ROM будет на первом месте, а системный жесткий диск на втором:



4. Подтвердите свой выбор нажатием клавиши Enter, сохраните изменения клавишами F10 и Y, вернитесь в главное меню клавишей Esc и загрузите машину, выбрав пункт меню *Continue* .

2.4.2 Настройка сетевого интерфейса для постоянной работы

Для управления сетевыми подключениями на рабочих станциях рекомендуется использовать службу NetworkManager, т.к. она позволяет автоматически переключаться на доступные Wi-Fi сети и устанавливать VPN-соединения. Откройте настройки сети в апплете NetworkManager и на вкладке «Параметры IPv4» установите значения:

- Метод: **Вручную**.

  Рабочие станции обычно получают сетевые настройки от DHCP-сервера, но в целях упрощения мы будем назначать адрес вручную.

- Адрес: **10.0.1.51** (в соответствии с описанием тестового стенда).

- Маска: **24** или **255.255.255.0**.

- Шлюз: **10.0.1.1** (адресом шлюза VirtualBox является первый IP-адрес сети).

- Серверы DNS: **10.0.1.11** (адрес dc-1.ald.company.lan).

  Обратите внимание, что при использовании DNS-сервера от Microsoft, он должен резолвить записи не только из прямой, но и обратной зоны тоже. Если перенаправить обратную зону невозможно, требуется добавить вручную PTR-записи контроллеров домена, иначе ввод машин в домен будет невозможен из-за дополнительной проверки безопасности на стороне клиента.

- Поисковый домен: **ald.company.lan** (это значение будет добавляться к коротким именам).

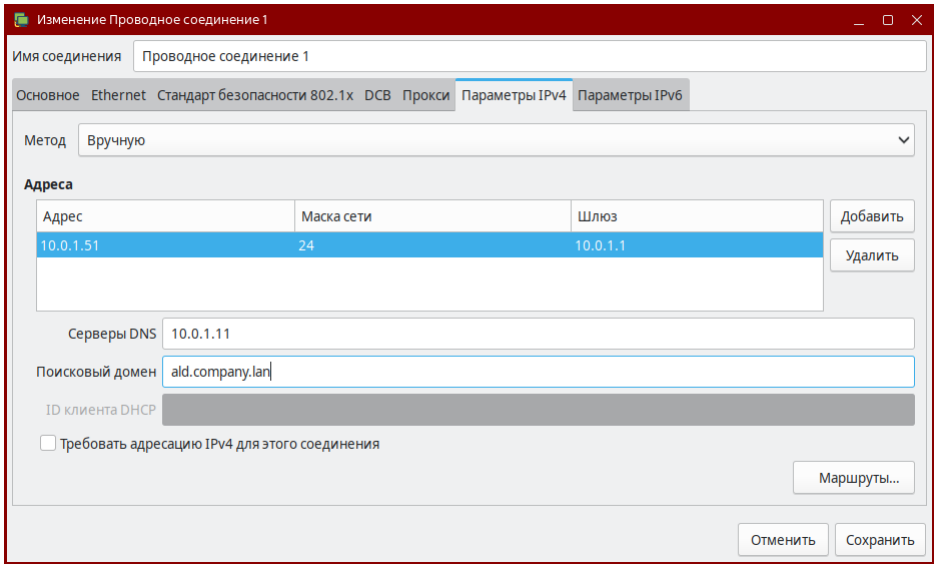


рис. 2.23 – Настройка сети на рабочей станции

Для применения новых настроек необходимо выключить и повторно включить поддержку сети в контекстном меню апплета.

2.4.3 Настройка репозиториев продукта ALD Pro

Для установки клиентской части ALD Pro используются те же репозитории, что и для установки серверной части, поэтому содержимое файлов `/etc/apt/sources.list` и `/etc/apt/sources.list.d/aldpro.list` должно быть аналогичным.

Установите для файла `/etc/apt/sources.list` следующее содержимое:

```
deb https://dl.astralinux.ru/astra/frozen/1.7_x86-64/1.7.6/repository-base/ 1.7_x86-
↪64 main contrib non-free
deb https://dl.astralinux.ru/astra/frozen/1.7_x86-64/1.7.6/repository-extended/ 1.7_
↪x86-64 main contrib non-free
```

Установите для файла `/etc/apt/sources.list.d/aldpro.list` следующее содержимое:

```
deb https://dl.astralinux.ru/aldpro/frozen/01/2.4.0 1.7_x86-64 main base
```

После изменение настроек репозиториев не забудьте обновить индекс пакетов:

```
sudo apt update
```

2.4.4 Установка пакетов клиентской части

Теперь система готова к установке клиентской части ALD Pro. Выполните следующую команду:

```
sudo DEBIAN_FRONTEND=noninteractive \
  apt-get install -y -q aldpro-client
```

Для получения дополнительной информации по использованным ключам см. раздел *Установка пакетов контроллера домена*.

Если перезагрузить пользовательский компьютер прямо сейчас, то в сообщениях ядра можно будет увидеть ошибки запуска службы sssd и других зависящих от нее сервисов (журнал загрузки можно найти в файле `/var/log/boot.log`). Это происходит по причине того, что служба еще не настроена соответствующим образом (журнал службы sssd можно найти в файле `/var/log/sssd/sssd.log`).

2.4.5 Проверка имени хоста на уникальность

Как вы знаете, имя компьютера должно быть уникальным в домене. В системе Windows имена компьютеров задаются в верхнем регистре символов и по умолчанию представляют собой случайную строку в формате DESKTOP-S16CV34. В системе Astra Linux имена задаются в нижнем регистре и по умолчанию предлагается использовать имя astra. Чтобы сгенерировать случайную строку как в Windows, вы можете воспользоваться следующей командой:

```
$ echo "desktop-$(expr $RANDOM$(date +%s) | md5sum | head -c 7)"
desktop-711eaaf
```

Проверить желаемое имя на уникальность можно с помощью утилиты `host`. По приведенному далее примеру можно заметить, что имя **dc-1** уже используется хостом с IP-адресом **10.0.1.11**, а имя **pc-1** еще свободно:

```
admin@dc-1:~$ host dc-1.ald.company.lan
dc-1.ald.company.lan has address 10.0.1.11

admin@dc-1:~$ host pc-1.ald.company.lan
Host pc-1.ald.company.lan not found: 3(NXDOMAIN)
```

В предыдущих редакциях продукта имя хоста на клиентских компьютерах нужно было устанавливать вручную до присоединения компьютера к домену, но с версии 2.4.0 это уже не требуется и оно будет изменено автоматически.

2.4.6 Консольная утилита aldpro-client-installer

2.4.6.1 Ввод компьютера в домен

Вводом компьютера в домен или присоединением к домену (*англ. Domain Join*) называют процедуру, в ходе которой для компьютера создается учетная запись в домене и выполняется настройка компьютера, чтобы он мог выполнять аутентификацию/авторизацию пользователей и получать

параметры групповых политик из домена. После присоединения компьютера его называют участником домена (*англ. Domain Members*) или просто доменным компьютером.

В домене MS AD присоединение компьютеров к домену выполняется с помощью оснастки `Свойства компьютера` или командлета `Add-Computer`, а в домене ALD Pro для этого предназначена утилита `aldpro-client-installer`. Утилита имеет графический интерфейс, но используется преимущественно в неинтерактивном консольном режиме, в котором все параметры являются обязательными. Перед вызовом утилиты рекомендуется отключать запись истории команд, так как в одном из параметров передается пароль администратора в открытом виде.

Выполним присоединение компьютера к домену из консоли следующими командами:

```
set +o history # Отключаем запись истории команд

sudo /opt/rbta/aldpro/client/bin/aldpro-client-installer --host pc-1 \
  --domain ald.company.lan --account admin --password 'Pa$$w0rd' \
  --gui --force

set -o history # Включаем запись истории команд
```

Комментарии по использованным ключам:

- Параметр `host` — короткое имя хоста без доменной части, например, **pc-1**. Значение должно быть указано в нижнем регистре.
- Параметр `domain` — полное имя домена, к которому нужно присоединить хост, например, **ald.company.lan**.
- Параметр `account` — логин администратора домена или другой учетной записи, обладающей достаточными привилегиями для присоединения компьютеров к домену.
- Параметр `password` — пароль от указанной привилегированной учетной записи.
- Ключ `gui` — указывает, что присоединение компьютера нужно выполнить в консоли без использования графического интерфейса.
- Ключ `force` — указывает, что компьютер следует ввести в домен, даже если в домене уже есть учетная запись для указанного хоста. Требуется в тех случаях, когда администратор переустанавливает операционную систему и хочет ввести компьютер в домен с тем же именем.

Описание параметров скрипта можно получить с помощью ключа `-h`. Также доступны короткие псевдонимы, например, в место `--domain` можно указать `-c`, но эти сокращения, на наш взгляд, являются менее запоминающимися.

В ходе присоединения компьютера к домену для него создается учетная запись, с помощью которой он сможет проверять аутентичность доменных пользователей и выполнять авторизованные

запросы в домене. На компьютерах Windows учетные данные компьютера хранятся в защищенной части реестра, в так называемом менеджере учетных записей безопасности (*англ. Security Accounts Manager, SAM*). На компьютерах Linux они находятся в файле `/etc/krb5.keytab`, и их можно посмотреть с помощью утилиты `klist`:

```
localadmin@pc-1:~$ sudo klist -ket
Keytab name: FILE:/etc/krb5.keytab
KVNO Timestamp           Principal
---- ------------------- ------------------------------------------------------
   1 07.04.2025 14:36:35 host/pc-1.ald.company.lan@ALD.COMPANY.LAN (aes256-cts-hmac-
↪sha1-96)
   1 07.04.2025 14:36:35 host/pc-1.ald.company.lan@ALD.COMPANY.LAN (aes128-cts-hmac-
↪sha1-96)
```

Для того чтобы изменения вступили в силу, требуется выполнить перезагрузку компьютера:

```
sudo reboot
```

После перезагрузки войдите в систему, используя учетную запись администратора домена. Для первого входа в систему доменной учетной записью нужен будет доступ к контроллеру домена, а в дальнейшем проверка аутентичности возможна будет через кэш службы **sssd**. Подробно о том, как служба SSSD встраивается в PAM и NSS для обеспечения работы компьютера в домене, смотрите *Модуль 5. Работа Linux-компьютера в домене*.

2.4.6.2 Вывод компьютера из домена

Вывод компьютера из домена (*англ. Unjoin from the Domain*) выполняют, когда требуется изменить назначение компьютера или полностью вывести его из эксплуатации, чтобы пользователи этого компьютера не могли получить доступ к ресурсам домена, участником которого он являлся. Мы рассмотрим особенности этой процедуры как со стороны рабочей станции, так и со стороны домена

Действия на рабочей станции

В мире Linux очень высока степень повторного использования программного кода, поэтому для правильной работы приложения ему могут требоваться десятки или даже сотни вспомогательных пакетов. Чтобы упростить работу с зависимостями предназначены специальные пакетные менеджеры, но даже они не всегда справляются с корректным удалением сложного системного программного обеспечения, поэтому для полного приведения системы в исходное состояние довольно часто рекомендуют восстановливать виртуальную машину из резервной копии или выполнять повторную установку операционной системы. Тем не менее, в случае более простых клиентских приложений эта задача вполне решаема, и в данном разделе мы представим несколько рекомендаций, с помощью которых можно максимально приблизиться к желаемому результату.

Чтобы вывести компьютер из домена, на рабочей станции в первую очередь нужно откатить на-

стройки клиента FreeIPA, то есть привести содержание общих конфигурационных файлов, включая настройки PAM стека, в исходное состояние. Для этого войдите в систему под учетной записью локального администратора, например, **localadmin**, и выполните команду `astra-freeipa-client` с ключом `-U`:

```
# Выполните команду на pc-1 под локальным администратором localadmin
sudo astra-freeipa-client -U -y
```

Выполнять эту команду нужно из-под учетной записи локального администратора, т.к. после ее выполнения доменный администратор **admin** потеряет возможность запускать команды с повышенными привилегиями из-под `sudo`. В ходе выполнения команды в системе происходят следующие изменения:

- В службе каталога снимается отметка о том, что хост зарегистрирован в домене, т.е. учетной записи хоста удаляется Kerberos-пароль. Действие выполняется авторизованно из-под учетной записи самого хоста.
- Ключи хоста удаляются из файла `/etc/krb5.keytab`.
- Содержимое конфигурационных файлов `/etc/krb5.conf`, `/etc/ldap/ldap.conf` и др. приводится в исходное состояние.
- К имени файла конфигурации `/etc/sssd/sssd.conf` добавляется суффикс `.deleted`.

Далее следует удалить пакеты приложений, указав явно **aldpro**, **freeipa**, **sssd** и **krb5** (все остальные пакеты будут удалены через зависимости):

```
sudo apt purge 'aldpro*' 'freeipa*' 'sssd*' 'krb5*'
sudo apt autoremove --purge
```

Если не удалить приложения, то после перезагрузки останутся включенными, но будут работать некорректно такие службы как aldpro-salt-minion, aldpro-client-service-discovery, aldpro-client-rdm.

Далее следует удалить домашние директории доменных пользователей, в нашем случае доменного администратора.

```
sudo rm -rf /home/admin
```

Если в домене используются механизмы мандатного управления доступом, не забудьте удалить также домашние директории доменных пользователей из папки `/home/.pdp`.

Далее следует удалить каталоги клиентских приложений подсистем ALD Pro и другие артефакты, созданные из скриптов, потому что пакетный менеджер ничего не знает об этих файлах и не сможет взять на себя ответственность за их удаление.

```
sudo rm -rf /var/lib/sss/pubconf/krb5.include.d/
```

(continues on next page)

(продолжение с предыдущей страницы)

```
sudo rm -rf /etc/krb5.conf.d/
sudo rm -rf /var/lib/sss/
sudo rm -rf /etc/sssd/
sudo rm -rf /opt/rbta/aldpro/
sudo rm -rf /etc/syslog-ng/aldpro/
sudo rm -rf /etc/salt/minion.d/
sudo rm -rf /var/log/salt/
sudo rm -rf /var/lib/certmonger/
sudo rm -rf /var/tmp/aldpro-salt.ccache
```

Далее следует удалить корневой сертификат домена из файла `/etc/ssl/certs/cacertificates.crt`, для этого нужно выполнить команду `update-ca-certificates`:

```
sudo update-ca-certificates
```

В завершение компьютеру нужно установить короткое имя **astra**, чтобы при выполнении команд `sudo` не получать предупреждений о невозможности разрешения полного имени **astra.ald.company.lan**, которое было установлено скриптом `astra-freeipa-client` при выводе машины из домена:

```
sudo hostnamectl set-hostname astra
```

После перезагрузки войдите в систему, используя учетную запись локального администратора.

Действия на контроллере домена

Обычно при выведении компьютера из домена его аккаунт и DNS-записи удаляются из LDAP-каталога не сразу, а только по истечению некоторого периода, продолжительность которого определяется политикой удержания.

Для того, чтобы никто не смог воспользоваться аккаунтом компьютера, который выведен из домена, при выполнении этой процедуры на клиентском компьютере утилита `astra-freeipa-client` автоматически удаляет из LDAP-каталога Kerberos-пароль для учетной записи этого компьютера.

Если выполнение команд на стороне клиента невозможно, то для удаления Kerberos-пароля хоста из LDAP-каталога вы можете воспользоваться командой `host-disable` утилиты `ipa`:

```
admin@dc-1:~$ ipa host-disable pc-1.ald.company.lan
ipa: ERROR: Эта запись уже отключена
```

Как можно заметить, запись хоста уже отключена в домене, поэтому ничего больше не требуется. Запускать эту команду можно с любого компьютера в домене, совсем не обязательно, чтобы это был контроллер. Вам будет достаточно, чтобы в связке ключей был валидный TGT-билет доменного пользователя, обладающего достаточными привилегиями.

Если же вы захотите привести LDAP-каталог в исходное состояние, как до ввода компьютера в домен, то нужно будет полностью удалить аккаунт компьютера и его DNS-записи. Сделать это можно из терминала следующими командами:

```
kinit admin
ipa dnsrecord-del ald.company.lan. pc-1 --del-all
ipa host-del pc-1.ald.company.lan
```

Результаты выполнения команд:

```
admin@dc-1:~$ kinit admin

admin@dc-1:~$ ipa dnsrecord-del ald.company.lan. pc-1 --del-all
---------------------
Удалена запись "pc-1"
---------------------

admin@dc-1:~$ ipa host-del pc-1.ald.company.lan
-----------------------------------
Удалён узел "pc-1.ald.company.lan"
-----------------------------------
```

Информация для устаревших редакций продукта

В устаревших редакциях продукта при выводе компьютера из домена нужно было удалить Salt-ключи этого компьютера на всех контроллерах домена командой `sudo salt-key -y -d pc-1.ald.company.lan`. Начиная с версии 2.4.0, управление подсистемами переведено полностью на автономную службу `aldpro-salt-minion`, поэтому компонент Salt Master более не используется и удалять Salt-ключи при выводе компьютера из домена теперь не требуется.

2.4.7 Графическая утилита

Если запустить утилиту `aldpro-client-installer` без параметров, то откроется окно графического интерфейса, см. рис. 2.24. Однако следует принять во внимание, что графический режим стандартной утилиты предоставляет только базовые возможности для интерактивного заполнения полей и считается уже устаревшим.

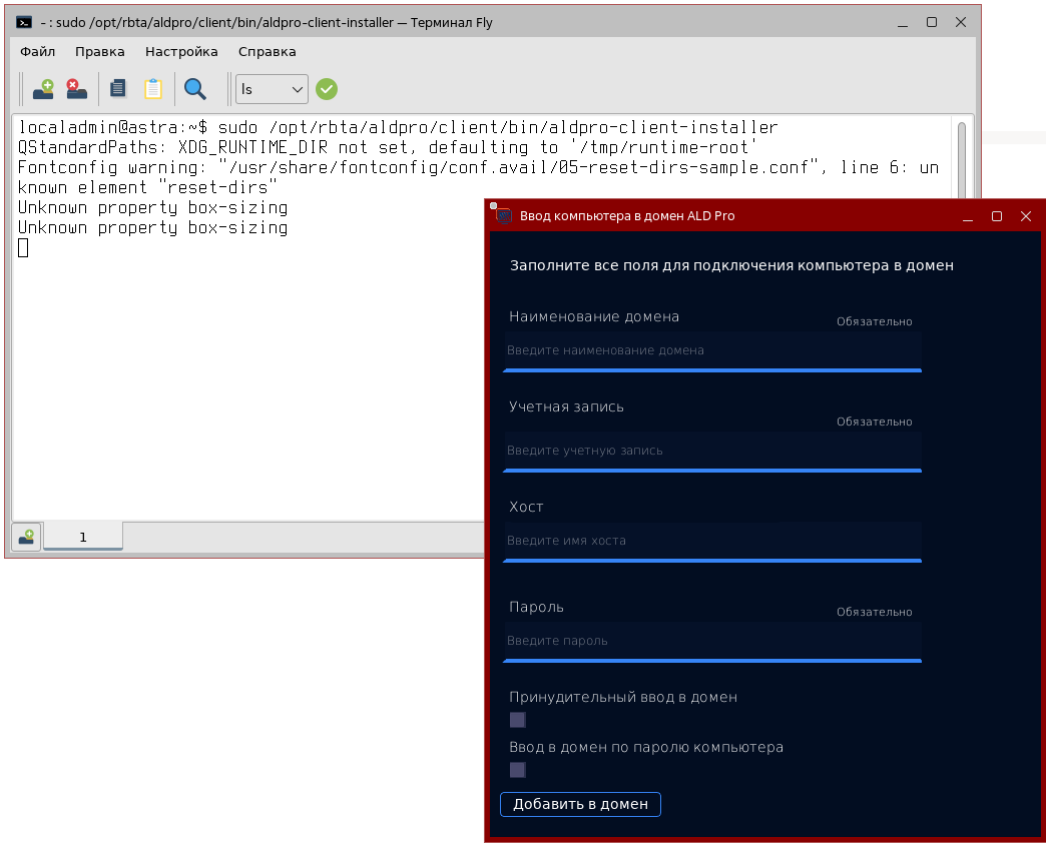


рис. 2.24 – Устаревший графический интерфейс для ввода компьютера в домен

Запуск в графическом режиме после изменения имени хоста

Если при запуске утилиты в графическом режиме возникнет ошибка `qt.qpa.xcb: could not connect to display :0`, значит имя хоста было изменено после входа в систему. В этом случае вам нужно просто перезапустить графическую сессию следующей командой:

```
sudo systemctl restart fly-dm
```

Для удобства присоединения компьютеров Astra Linux к домену ALD Pro была разработана дополнительная графическая утилита `aldpro-join`, которая предоставляет Windows-администраторам знакомый интерфейс и дополнительные преимущества, в том числе возможность присоединения компьютера в заданное структурное подразделение. Утилита опубликована через личный кабинет в составе релиза 2.5.0, но поддерживает работу в том числе и с предыдущей LTS-версией 2.4.0. В последующих релизах утилита будет включена уже в состав репозитория продукта.

2.4.7.1 Ввод компьютера в домен

Для присоединения компьютера к домену с помощью графической утилиты все предыдущие этапы по подготовке виртуальной машины должны быть точно такими же, включая настройку сетевого интерфейса и добавление репозиториев продукта. Устанавливать пакеты клиентской части продукта необязательно, так как они будут установлены автоматически по зависимостям при установке пакета графической утилиты.

Далее нужно выполнить следующие действия:

1. Скачайте deb-пакет по прямой ссылке или из личного кабинета и установите его следующей командой:

```
sudo DEBIAN_FRONTEND=noninteractive \
  apt-get install -y -q ./aldpro-join_2.0-2_amd64.deb
```

2. Запустите утилиту из меню *Пуск → ALD Pro → Ввод компьютера в домен ALD Pro v2.0*, этот ярлык использует команду aldpro-join --gui.

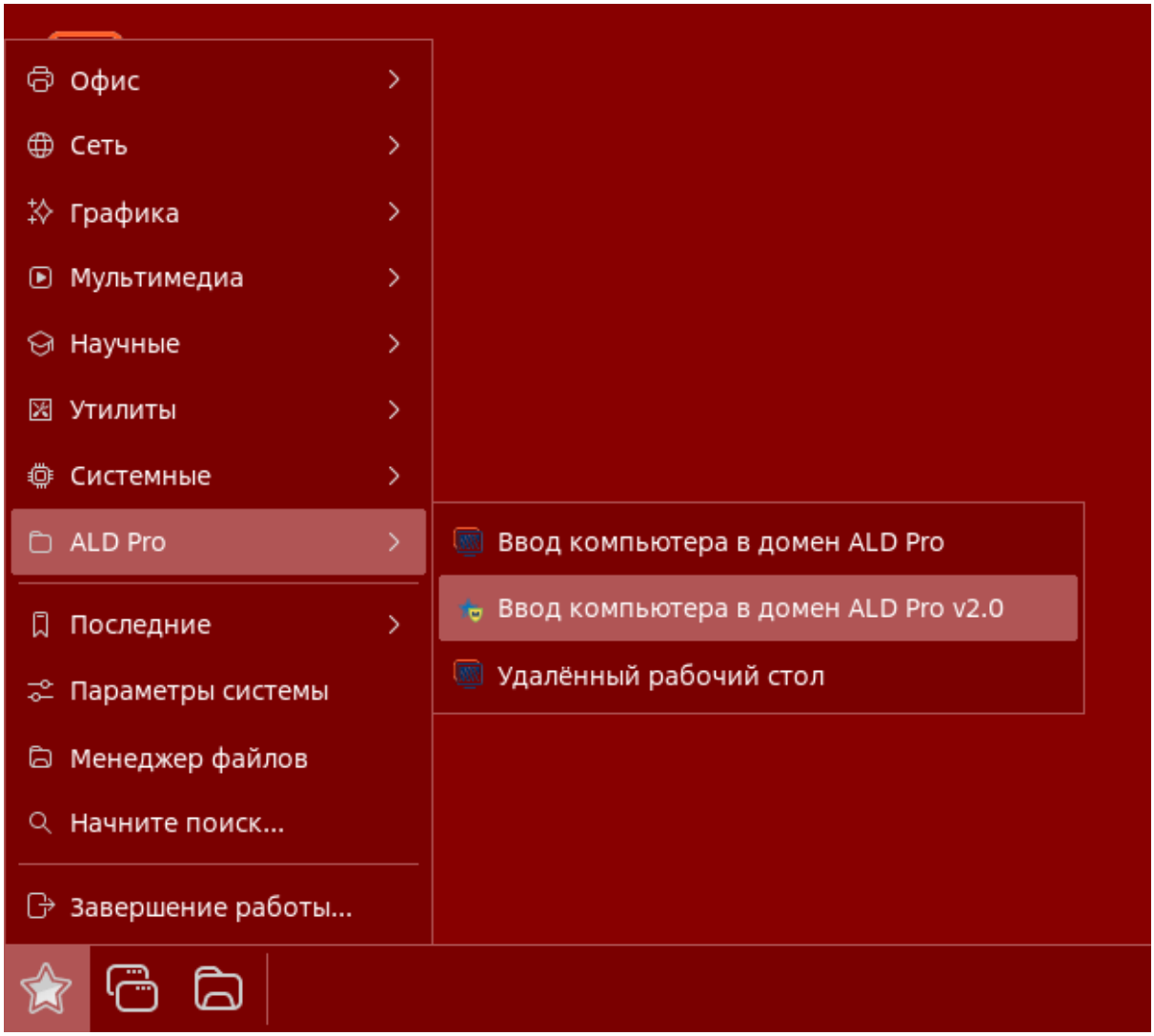


рис. 2.25 – Запуск приложения для присоединения к домену v2.0

3. Чтобы изменить имя компьютера, нажмите кнопку *Изменить...* напротив имени компьютера и установите желаемое значение, например, pc-14.

   Обратите внимание, что после изменения имени компьютера вы сможете продолжить работу с графической утилитой, но если вы закроете ее, то повторный запуск приложения будет возможен только после перезапуска графической оболочки, для чего достаточно выйти из системы и зайти еще раз.

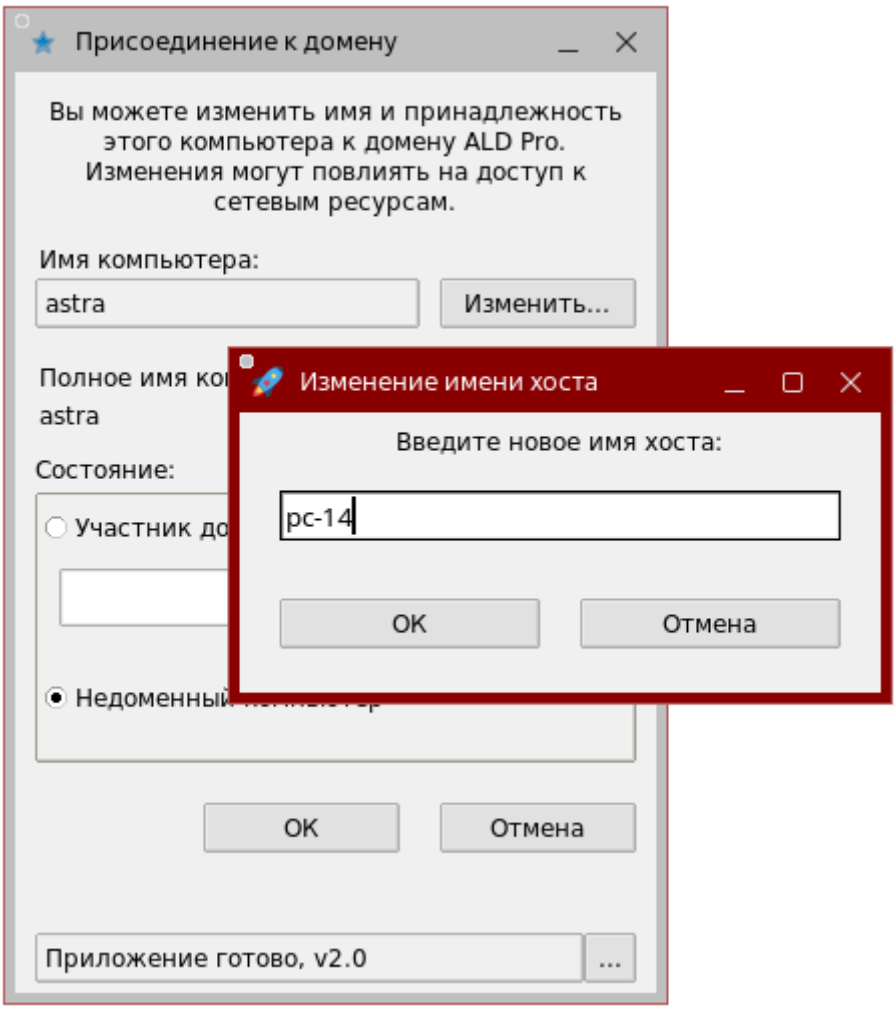


рис. 2.26 – Изменение имени компьютера

4. Установите переключатель в положение *Участник домена ALD Pro* и введите имя домена, например, **ald.company.lan**. Нажмите кнопку *OK*, чтобы начать присоединение компьютера к домену.

5. В окне аутентификации введите учетные данные доменного администратора. Это может быть пользователь admin, но можно создать отдельную учетную запись, привилегий которой будет достаточно только для ввода машин в домен. Информация представлена в руководстве администратора.

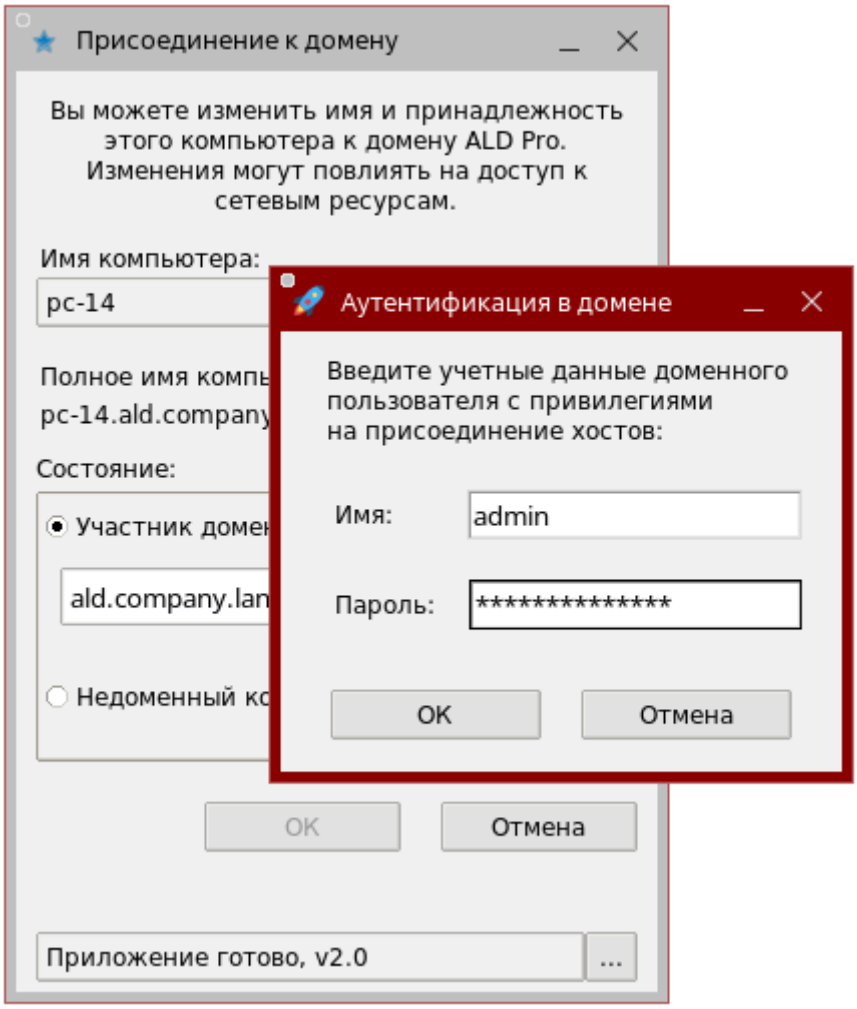


рис. 2.27 – Ввод имени администратора домена для присоединения компьютера

Примечание: В нашем сообществе профессионалов мы регулярно проводим фокус-группы по обсуждению новых функций, которые планируется включить в состав продукта. В рамках работы одной из таких групп в графической утилите был обнаружен дефект, который препятствует возможности использования паролей с символом доллара. Указанное ограничение будет устранено уже в следующем релизе утилиты.

---

6. Выберите организационное подразделение, в котором требуется создать учетную запись компьютера.

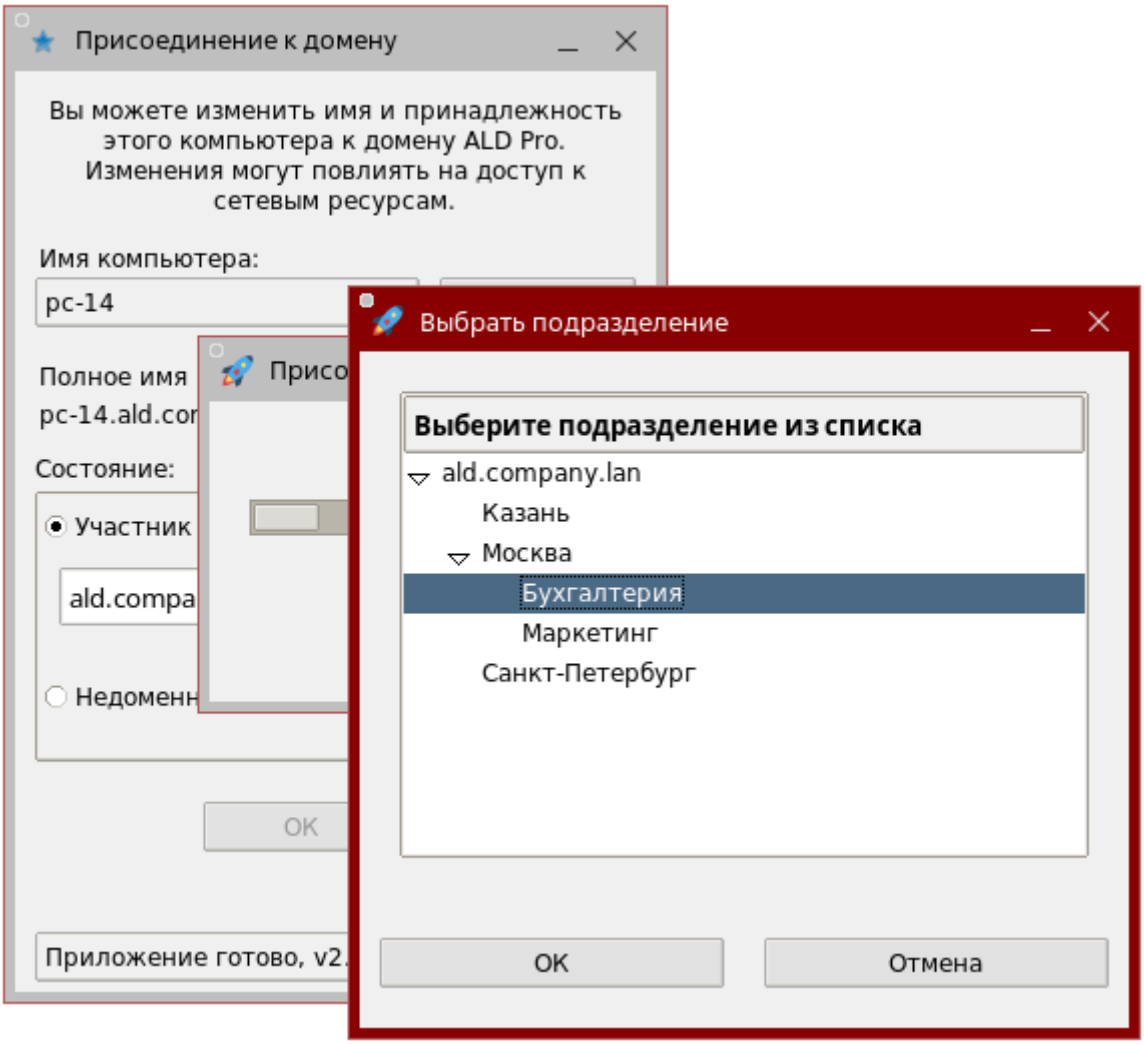


рис. 2.28 – Выбор организационного подразделения

7. Если процедура присоединения компьютера ALSE к домену прошла успешно, вы увидите соответствующее уведомление. Обратите внимание, что утилита сразу применяет групповые политики, политики ПО и задания автоматизации, поэтому процедура может занять некоторое время, наберитесь терпения.

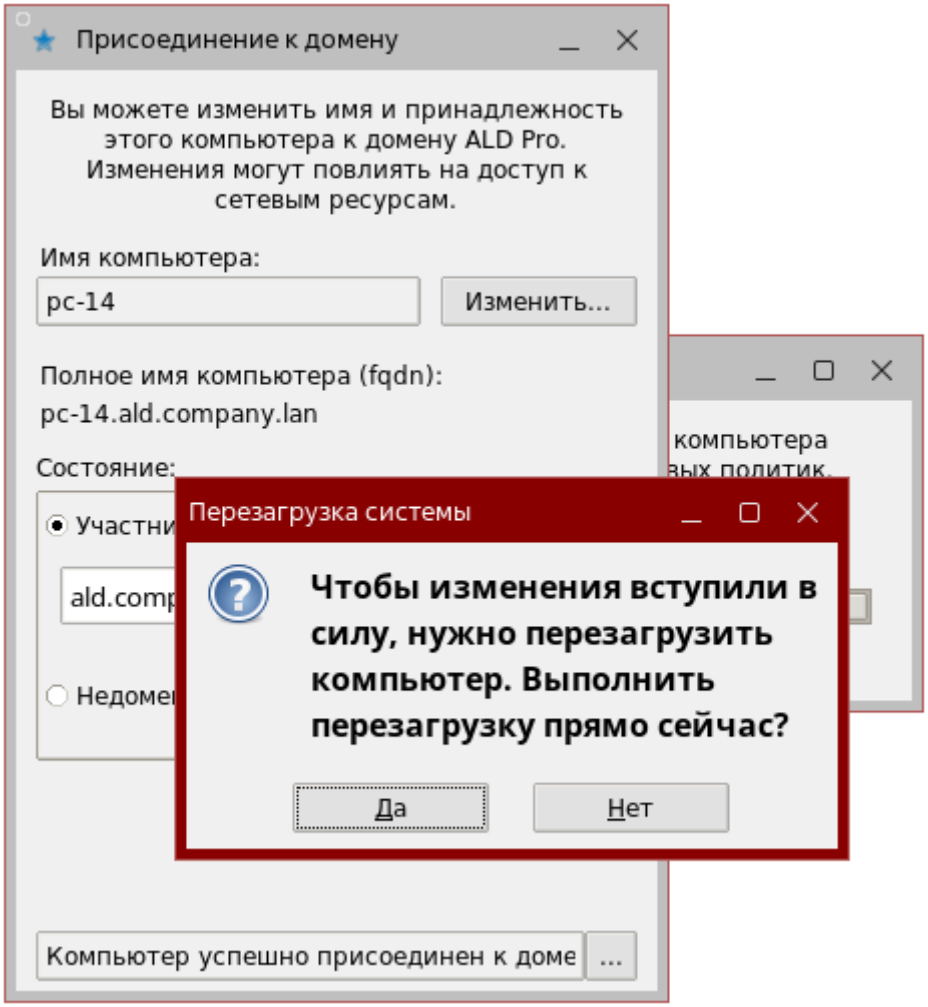


рис. 2.29 – Выполните перезагрузку системы

Журнал событий вы сможете посмотреть в файле `/var/log/.aldpro-join.log`. После перезагрузки компьютера вам станет доступен вход в систему доменным пользователем.

2.4.7.2 Вывод компьютера из домена

Если вы захотите вывести компьютер из домена, вам достаточно будет запустить приложение еще раз из-под локального администратора `localadmin`, установить переключатель в положение *Недоменный компьютер* и нажать кнопку *OK*. Учетная запись компьютера и его DNS-записи будут удалены из домена.

После вывода машины из домена рекомендуется сразу выполнить перезагрузку.

2.4.7.3 Проверка доверия с доменом

При вводе компьютера в домен для него создается учетная запись, пароль которой сохраняется в файле `/etc/krb5.keytab`. Если компьютер располагает актуальным паролем от своей учетной записи в домене, говорят, что между компьютером и доменом установлены доверительные отношения (*англ. Trust relationship between a domain-joined device and its domain*).

Срок действия пароля компьютера не ограничен, но в домене MS AD он обновляется каждые 30 дней, поэтому при восстановлении компьютера из старой резервной копии вы можете столкнуться с тем, что он потеряет доверие с доменом. В домене FreeIPA пароль компьютера автоматически не обновляется, но вы можете настроить доп. параметр групповой политики ALD Pro, с помощью которого реализовать ту же логику, поэтому в утилиту `aldpro-join` встроен механизм для проверки и восстановления доверия с доменом.

Чтобы проверить актуальность пароля компьютера, нажмите кнопку *Проверить...* напротив имени домена. В поле *Версия ключа и дата выдачи* будет отображать значение в следующем формате: <**Версия пароля**> <**Дата изменения пароля**> <**Время изменения пароля**>.

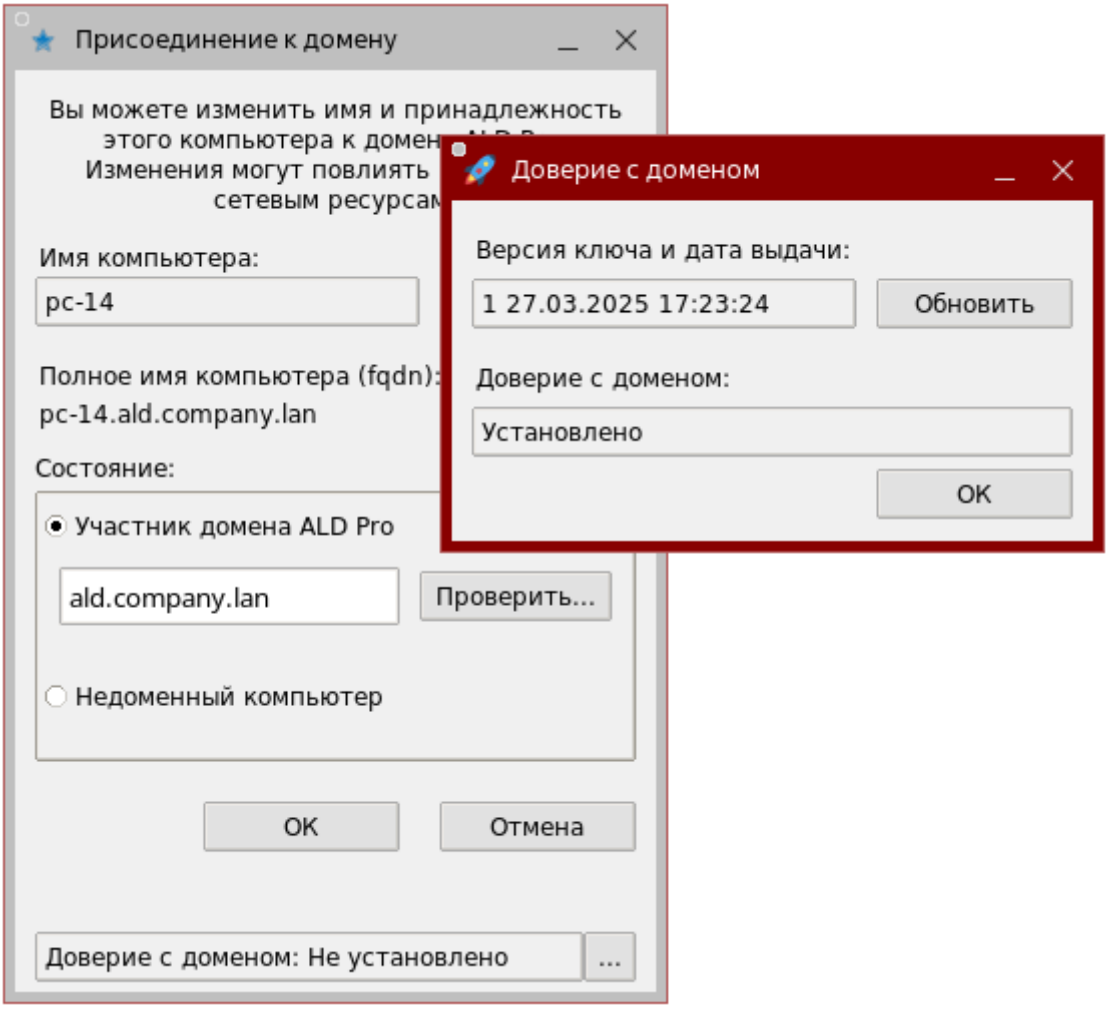


рис. 2.30 – Проверка доверия с доменом

2.4.7.4 Обновление пароля компьютера и восстановление доверия с доменом

Если нажать кнопку *Обновить*, то приложение установит компьютеру новый пароль и сохранит его в файл `/etc/krb5.keytab`.

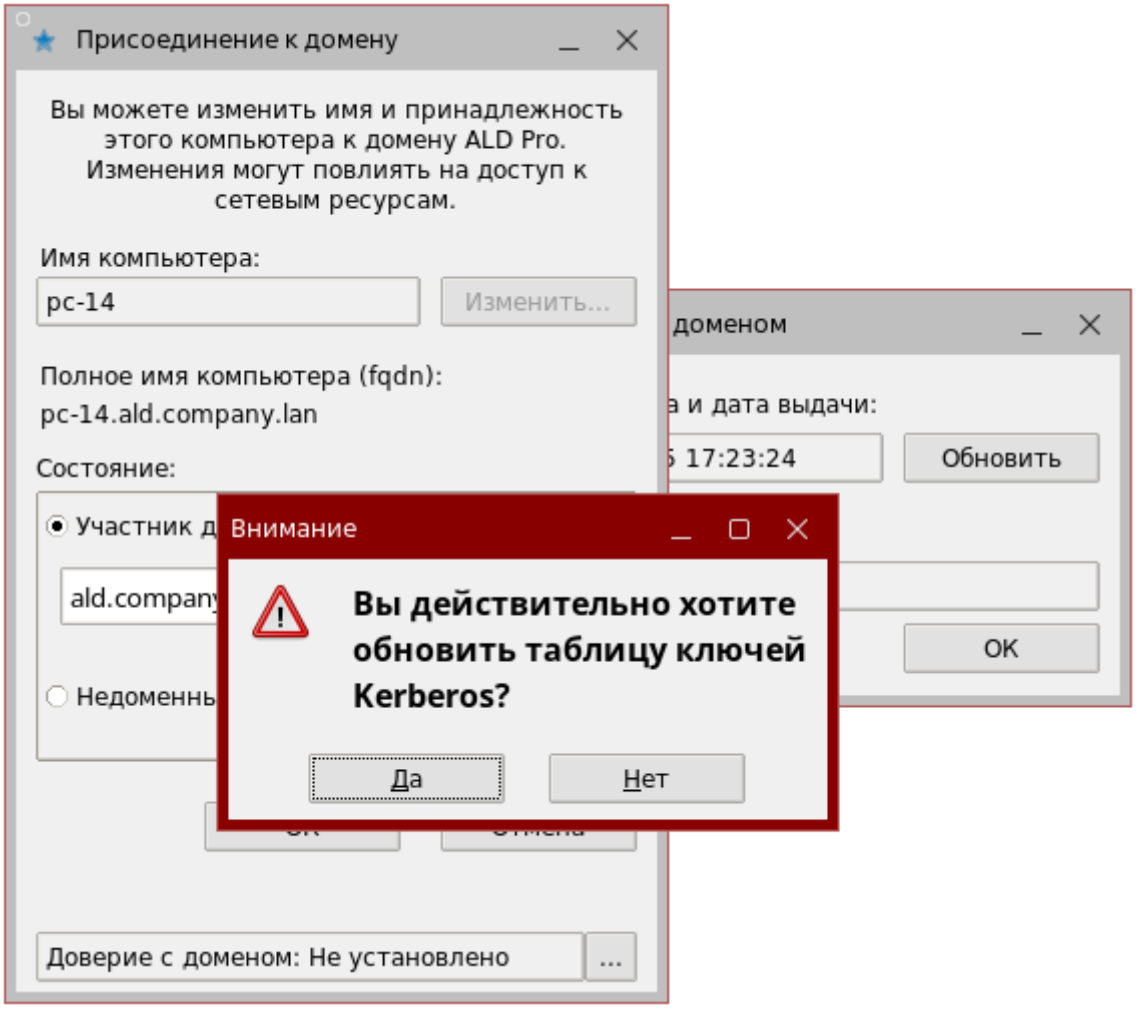


рис. 2.31 – Обновление пароля компьютера

Если в момент обновления пароля у компьютера будет доверие с доменом, то пароль будет обновлен из-под учетной записи компьютера автоматически. Если компьютер не будет располагать актуальным паролем, то для восстановления доверия потребуется ввести учетные данные администратора домена или любого другого пользователя, который обладает достаточными привилегиями.

2.4.8 Повышение безопасности при вводе компьютеров в домен

При вводе машины в домен происходит выполнение утилиты `ipa-getkeytab`, которая устанавливает Kerberos-пароль для учетной записи компьютера в домене, используя привилегированную учетную запись. В небольших компаниях для этого используют учетную запись `admin` или вклю-

чают персональную учетную запись сотрудника в группу admins, но это не подходит для крупных организаций с большим штатом ИТ-специалистов, где эту задачу поручают сотрудникам, не имеющим отношения к администрированию серверной группировки. Рассмотрим несколько способов повышения безопасности процедуры ввода компьютера в домен.

2.4.8.1 Присоединение к домену из-под учетной записи с ограниченными правами

Хорошей практикой считается предоставлять сотрудникам только те права, которые действительно необходимы им для выполнения должностных обязанностей.

Права доступа в службе каталога FreeIPA делегируются с помощью трехуровневой модели безопасности, которая включает в себя Роли (*Roles*), Привилегии (*Privileges*) и Разрешения (*Permissions*). На базовом уровне разрешениям соответствуют инструкции управления доступом 389 Directory Server (*Access Control Instructions, ACI*), с помощью которых можно предоставить доступ на чтение, запись, поиск и другие действия применительно ко всему каталогу, его ветке или конкретной записи. Разрешения группируются в привилегии, привилегии группируются в роли, а роли уже назначаются пользователям.

В службе каталога FreeIPA уже есть роль Enrollment Administrator, но у нее нет разрешения на создание хостов System: Add Hosts, поэтому она подходит для присоединения к домену только тех компьютеров, у которых уже есть учетная запись в домене:

```
admin@dc-1:~$ ipa role-show 'Enrollment Administrator'
  Имя роли: Enrollment Administrator
  Описание: Участники роли имеют привилегии, необходимые для регистрации компьютеров␣
↪в домене: - регистрация компьютеров в домене; - управление аутентификационными␣
↪данными компьютеров.
  Privileges: Host Enrollment

admin@dc-1:~$ ipa privilege-show 'Host Enrollment'
  Имя привилегии: Host Enrollment
  Описание: Host Enrollment
  Разрешения: System: Add krbPrincipalName to a Host, System: Enroll a Host, System:␣
↪Manage Host Certificates, System: Manage Host Enrollment Password, System: Manage␣
↪Host Keytab,
              System: Manage Host Principals
  Предоставление привилегий ролям: Enrollment Administrator
```

Чтобы с помощью этой роли можно было присоединять к домену новые компьютеры, которых еще нет в службе каталога, мы могли бы добавить разрешение System: Add Hosts к привилегии Host Enrollment, которой обладает роль Enrollment Administrator, но лучше будет создать новую роль, которая будет вбирать в себя все необходимые разрешения.

Создадим новую роль Enrollment Manager, которой в дополнение к существующей привилегии Host Enrollment добавим новую привилегию Host Creation с разрешением System: Add Hosts:

```
ipa privilege-add 'Host Creation' \
  --desc='Создание новых хостов в домене'

ipa privilege-add-permission 'Host Creation' \
  --permissions='System: Add Hosts'

ipa role-add 'Enrollment Manager' \
  --desc='Участники имеют право создавать хосты и присоединять их к домену'

ipa role-add-privilege 'Enrollment Manager' \
  --privileges='Host Creation' \
  --privileges='Host Enrollment'
```

Результат выполнения команд:

```
admin@dc-1:~$ ipa privilege-add 'Host Creation' \
>       --desc='Создание новых хостов в домене'
-----------------------------------
Добавлена привилегия "Host Creation"
-----------------------------------
Имя привилегии: Host Creation
Описание: Создание новых хостов в домене

admin@dc-1:~$ ipa privilege-add-permission 'Host Creation' \
>       --permissions='System: Add Hosts'
Имя привилегии: Host Creation
Описание: Создание новых хостов в домене
Разрешения: System: Add Hosts
-----------------------------------
Количество добавленных разрешений 1
-----------------------------------

admin@dc-1:~$ ipa role-add 'Enrollment Manager' \
>       --desc='Участники имеют право создавать хосты и присоединять их к домену'
-----------------------------------
Добавлена роль "Enrollment Manager"
-----------------------------------
Имя роли: Enrollment Manager
Описание: Участники имеют право создавать хосты и присоединять их к домену

admin@dc-1:~$ ipa role-add-privilege 'Enrollment Manager' \
>       --privileges='Host Creation' \
>       --privileges='Host Enrollment'
Имя роли: Enrollment Manager
Описание: Участники имеют право создавать хосты и присоединять их к домену
Privileges: Host Enrollment, Host Creation
-----------------------------------
Количество добавленных привилегий 2
-----------------------------------
```

Создадим нового пользователя `Tech Support Engineer` и назначим ему роль `Enrollment Manager`:

```
ipa user-add tse --first='Tech Support' --last='Engineer' --random

ipa role-add-member 'Enrollment Manager' --users=tse
```

Результат выполнения команд:

```
admin@dc-1:~$ ipa user-add tse --first='Tech Support' --last='Engineer' --random
---------------------------
Добавлен пользователь "tse"
---------------------------
  Имя учётной записи пользователя: tse
  Имя: Tech Support
  Фамилия: Engineer
  Полное имя: Tech Support Engineer
  ...

admin@dc-1:~$ ipa role-add-member 'Enrollment Manager' --users=tse
  Имя роли: Enrollment Manager
  Описание: Участники имеют право создавать хосты и присоединять их к домену
  Пользователи-участники: tse
  Privileges: Host Enrollment, Host Creation
-----------------------------------
Количество добавленных участников 1
-----------------------------------
```

Возможности по управлению системными ролями и привилегиями на портале несколько ограничены, но вы можете назначать эти роли через веб-интерфейс на вкладке *Роли* страницы *Пользователи и компьютеры* → *Пользователи* → *Пользователь*, см. рис. 2.32. Для этого достаточно выполнить следующие три действия:

1. Найти нужную запись в списке всех ролей с помощью поиска по подстроке «enroll».

2. Кликнуть по стрелке «влево», чтобы перенести найденную запись в список назначенных (выбранных) ролей.

3. Нажать кнопку *Сохранить*, чтобы зафиксировать изменения.

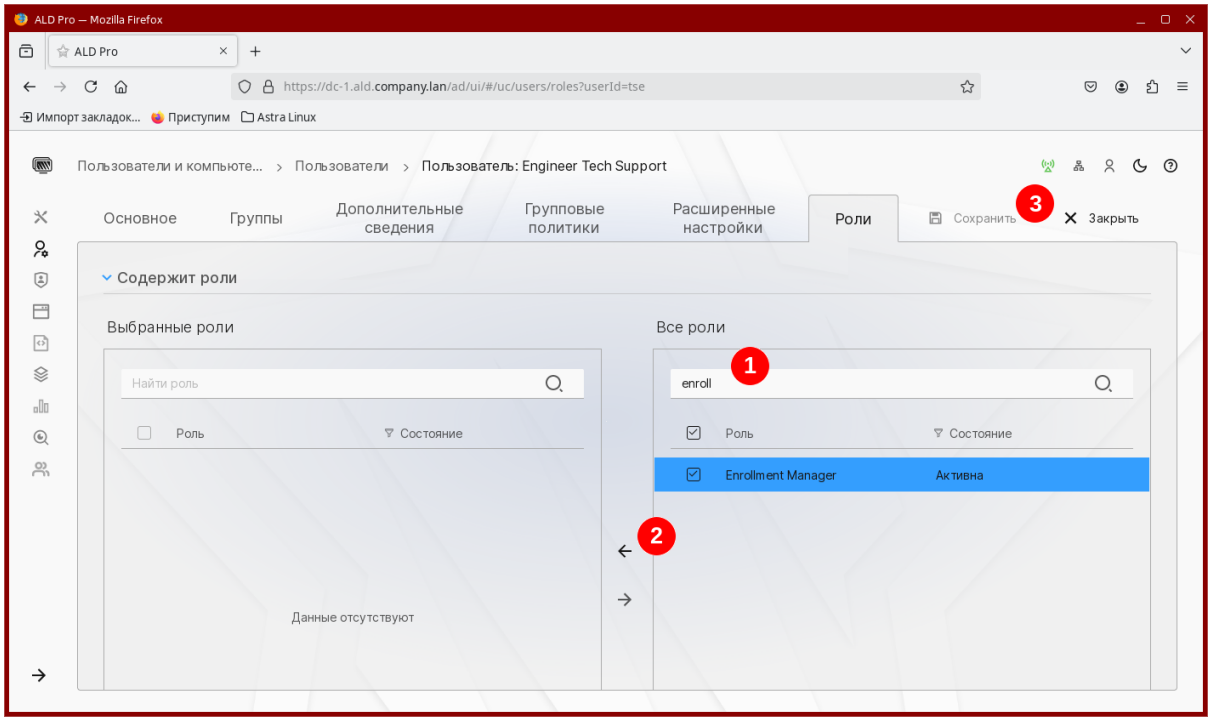


рис. 2.32 – Назначение роли «Enrollment Manager»

После назначения роли Enrollment Manager пользователь сможет создавать компьютеры в домене и присоединять их к домену, не имея при этом полных административных прав в домене. Но следует принимать во внимание, что кроме прав в домене администратору потребуются также права суперпользователя на локальной машине.

2.4.8.2 Присоединение к домену с помощью одноразового пароля

Чтобы уменьшить риск разглашения паролей доменных администраторов при вводе компьютеров домен, в FreeIPA есть возможность присоединения машин по собственному паролю компьютера. В качестве такого пароля может выступать одноразовый LDAP-пароль или Kerberos-ключи компьютера. В этом курсе мы рассмотрим способ с использованием временного LDAP-пароля.

Этот способ обладает несколькими преимуществами:

- Учетная запись хоста может быть создана сразу в правильном организационном подразделении, чтобы на компьютер распространялось действие необходимых групповых политик.
- Учетную запись хоста можно сразу включить во все группы, чтобы на него распространялось действие необходимых правил HBAC и SUDO.
- Для ввода машины в домен на стороне рабочей станции не нужно будет использовать пароль привилегированной учетной записи.

Продемонстрируем работу данного сценария на примере утилиты `aldpro-client-install`.

Сначала администратор домена должен создать учетную запись хоста с помощью команды `ipa host-add`, используя ключ `--random` для генерации одноразового пароля:

```
ipa host-add pc-2.ald.company.lan --random --ip-address=10.0.1.52 \
  --force --department='ou=Московский офис,ou=ald.company.lan,cn=orgunits,cn=accounts,
↪dc=ald,dc=company,dc=lan'
```

Где:

- Параметр `pc-2.ald.company.lan` — имя компьютера в формате FQDN.
- Ключ `random` — это опция указывающая на требование сгенерировать случайный одноразовый пароль.
- Параметр `ip-address` — адрес компьютера для создания DNS-записи в домене.
- Ключ `force` — опция, которая позволяет принудительно установить значение имени узла, даже если такого имени нет в DNS.
- Параметр `--department` — аргумент, который позволяет создать хост в нужном подразделении.

В результате выполнения команды будет создана учетная запись компьютера, которой будет назначен одноразовый пароль для присоединения к домену:

```
admin@dc-1:~$ ipa host-add pc-2.ald.company.lan --random --ip-address=10.0.1.52 \
>       --force --department='ou=Московский офис,ou=ald.company.lan,cn=orgunits,
↪cn=accounts,dc=ald,dc=company,dc=lan'
-----------------------------------
Добавлен узел "pc-2.ald.company.lan"
-----------------------------------
   Имя узла: pc-2.ald.company.lan
   Случайный пароль: 2LyXwGJItweZbvJP3LLqFCT
   Link to department: ou=Московский
         офис,ou=ald.company.lan,cn=orgunits,cn=accounts,dc=ald,dc=company,dc=lan
   Link to head department: Московский офис
   Пароль: True
   Таблица ключей: False
   Managed by: pc-2.ald.company.lan
```

С помощью такой учетной записи компьютера мы можем присоединить его к домену, не используя учетных данных администратора. В этом случае вместо ключа `--account` нам нужно будет указать ключ `--pc`, а в качестве пароля ввести одноразовый пароль самой машины:

```
sudo /opt/rbta/aldpro/client/bin/aldpro-client-installer --host pc-2 \
  --domain ald.company.lan --pc --password '2LyXwGJItweZbvJP3LLqFCT' \
  --gui --force
```

2.5 Практика и тестирование

2.5.1 Требования, правила и цели выполнения практической работы

Практическая работа в рамках этого курса проводится на виртуальной машине (ВМ) в гипервизоре VirtualBox. Допускается использование других систем виртуализации, но примеры будут приведены именно для VirtualBox ввиду его относительной легковесности и универсальности.

2.5.1.1 Требования к размещающей (хостовой) машине:

- Процессор с архитектурой x86_64 с 8 ядрами/потоками.
- Объем оперативной памяти не менее 16 ГБ.
- Объем свободного дискового пространства не менее 150 ГБ.
- Хостовая операционная система Astra Linux или Windows.
- Гипервизор VirtualBox 7.0 или новее.
- Доступ в Интернет.

Необходимые материалы:

- Образ для установки ОС Astra Linux SE 1.7.6

Требования к гостевым (тестовым) машинам:

- Виртуальные машины для прохождения практических заданий по этому курсу подготавливаются слушателем курса самостоятельно в рамках выполнения некоторых практических работ.

Правила и методика выполнения практической работы:

Для достижения наилучшего результата практическую работу рекомендуется выполнять без помощи раздела с ответами. Алгоритм должен быть следующий:

1. Прочитать и понять суть задания.
2. Выполнить задание.
3. В случае затруднений выполнить задание, используя лекционный материал и примеры.
4. В случае затруднений выполнить задание, используя информационные ресурсы интернет.
5. В случае затруднений выполнить задание, используя раздел с ответами.
6. Сравнить ваш способ решения с информацией из раздела с ответами.

Цели практической работы:

- Закрепление знаний, полученных в ходе изучения теоретического материала на практике.

Оценочное время на выполнение: 60-180 минут.

2.5.2 Практическая работа: Модуль 2. Развертывание ALD Pro

См. также *Требования, правила и цели выполнения практической работы*.

2.5.2.1 Практические задания (без ответов)

Задание 1. Подготовьте шаблон виртуальной машины для тестового стенда

1.1. Настройте натированную сеть для тестового стенда:

- Имя сети: **Сеть - Стенд ALD Pro 2.4.0**.
- CIDR сети (диапазон адресов): **10.0.1.0/24**.
- Поддержка DHCP: **Отключаем DHCP**.

1.2. Создайте виртуальную машину для первого контроллера домена:

- Имя: **dc-1.ald.company.lan - Стенд ALD Pro 2.4.0**.
- Папка машины: **по умолчанию**.
- Тип: **Linux**.
- Версия: **Other Linux (64-bit)**.
- Объем памяти: **4096 МБ**.
- Жесткий диск: **Создать новый виртуальный жесткий диск**.
- Тип файла: **VDI (VirtualBox Disk Image)**.
- Формат хранения: **Динамический виртуальный жесткий диск**.
- Путь: **по умолчанию**.
- Размер: **50 ГБ**.

В настройках задайте следующие параметры:

- Процессор: **4 ЦП**.

- Интерфейс паравиртуализации: **KVM**.
- Оптический привод: installation-1.7.6.11-26.08.24_17.26.iso.
- Тип сети: **Сеть NAT**.
- Имя сети: **Сеть - Стенд ALD Pro 2.4.0**.

1.3. Установите операционную систему с графическим окружением и максимальным уровнем защищенности «Смоленск»:

- Имя компьютера оставьте по умолчанию **astra**.
- Локальным администратором укажите пользователя **localadmin**.
- В качестве пароля локального администратора установите значение **Pa$$w0rd**.
- Настройку сети пропустите.
- При разметке диска выберите пункт **Авто – использовать весь диск и настроить LVM** (потребуется в будущем при обновлении ОС до версии ALSE 1.8+) и схему разметки **Все файлы в одном разделе**.
- Оставьте версию ядра по умолчанию **6.1-generic**. Ядра hardened (усиленная самозащита) и lowlatency (с увеличенной до 1000 Гц частотой переключения задач) продукт ALD Pro не поддерживает.
- Из пакетов программ по умолчанию можно исключить, например, **Средства работы с графикой**, но обязательно оставьте **Офисные приложения**, так как в дальнейшем нам потребуется LibreOffice для просмотра CSV-файлов в табличном виде. Добавьте также пакет **Средство удаленного подключения SSH**, если вы хотите подключаться к компьютеру удаленно.
- При выборе дополнительных параметров укажите уровень защищенности **Смоленск** и оставьте включенными флажки **Мандатный контроль целостности** и **Мандатное управление доступа**. В тестовых средах для удобства работы можно отключить флажок **Запрос пароля для команды sudo**.
- При настройке GRUB используйте тот же пароль **Pa$$w0rd**.

1.4. Сделайте временную настройку сетевого интерфейса через апплет NetworkManager со следующими параметрами:

- IP-адрес: **10.0.1.10**.
- Маска: **24** или **255.255.255.0**.

- Шлюз: **10.0.1.1**.
- Сервер DNS: **77.88.8.8**.

1.5. Настройте `base` и `extended` репозитории операционной системы.

1.6. Установите гостевые дополнения.

1.7. Клонируйте шаблон виртуальной машины:

- Имя: **Шаблон - Стенд ALD Pro 2.4.0**.
- Путь: **Оставьте по умолчанию**.
- Тип клонирования: **Полное клонирование**.
- Цель клонирования: **Состояние машины**.
- Политика MAC-адреса: **Сгенерировать новые MAC-адреса всех сетевых адаптеров**.

Задание 2. Разверните первый контроллер домена

2.1. Настройте сетевой интерфейс для постоянной работы:

- IP-адрес: **10.0.1.11**.
- Маска: **24** или **255.255.255.0**.
- Шлюз: **10.0.1.1**.
- Сервер DNS: **77.88.8.8**.

2.2. Настройте репозитории продукта ALD Pro.

2.3. Настройте имя контроллера домена.

2.4. Установите пакеты контроллера домена.

2.5. Выполните повышение роли сервера до контроллера домена.

2.6. Проверьте базовые функции домена.

2.6.1. Проверьте доменные службы sssd и ipactl.

2.6.2. Проверьте Kerberos-аутентификацию.

2.5.3. Проверьте портал управления и настройте домашнюю страницу.

2.6. Выполните дополнительную настройку DNS-службы.

2.6.1. Отключите DNSSEC и запретите рекурсивные запросы.

2.6.2. Настройте глобальный перенаправитель DNS-запросов.

Задание 3. Присоедините компьютер к домену

3.1. Подготовьте виртуальную машину:

- Имя: **pc-1.ald.company.lan - Стенд ALD Pro 2.4.0**.
- Путь: **Оставьте по умолчанию**.
- Тип клонирования: **Полное клонирование**.
- Цель клонирования: **Состояние машины**.
- Политика MAC-адреса: **Сгенерировать новые MAC-адреса всех сетевых адаптеров**.

3.2. Настройте сетевой интерфейс для постоянной работы:

- Метод: **Вручную**.
- Адрес: **10.0.1.51** (в соответствии с описанием тестового стенда).
- Маска: **24** или **255.255.255.0**.
- Шлюз: **10.0.1.1** (адресом шлюза VirtualBox является первый IP-адрес сети).
- Серверы DNS: **10.0.1.11** (адрес dc-1.ald.company.lan).
- Поисковый домен: **ald.company.lan** (это значение будет добавляться к коротким именам).

3.3. Настройте репозитории продукта ALD Pro.

3.4. Установите пакеты клиентской части.

3.5. Проверьте имя хоста на уникальность.

3.6. Выполните ввод компьютера в домен.

Задание 4. Выполните вывод компьютера из домена

- Откатите настройки клиента FreeIPA.
- Удалите пакеты приложений.
- Удалите домашнюю директорию доменного администратора.

- Удалите каталоги клиентских приложений подсистем ALD Pro и другие артефакты, созданные из скриптов.
- Удалите корневой сертификат домена.
- Верните исходное имя хоста и перезагрузить систему.
- На контроллере домена удалите аккаунт компьютера и его DNS-записи.

Задание 5. Присоедините компьютер к домену графической утилитой

- Скачайте deb-пакет по прямой ссылке или из личного кабинета.
- Запустите утилиту из меню *Пуск → ALD Pro → Ввод компьютера в домен ALD Pro v2.0*.
- Измените имя компьютера, например, pc-14.
- Введите компьютер в домен **ald.company.lan**.

Задание 6. Проверьте доверие с доменом и сбросьте пароль

- Проверьте наличие доверия с доменом.
- Обновите пароль компьютера.

Задание 7. Присоедините компьютер к домену учетной записью с ограниченными правами

- Создайте новую роль и добавьте ей все необходимые права доступа.
- Создайте пользователя tse и назначьте ему новую роль.
- Проверьте возможность присоединения компьютера к домену с использованием учетной записи пользователя tse.

Задание 8. Присоедините компьютер к домену с помощью одноразового пароля

- Создайте учетную запись хоста с одноразовым паролем.
- Присоедините компьютер pc-2 к домену, используя одноразовый пароль самой машины.

2.5.2.2 Практические задания (с ответами)

Для отработки практических навыков мы предлагаем вам подготовить стенд, который был описан в материалах модуля. В этом документе мы собрали очень краткие подсказки, чтобы вы могли проверить себя, получится ли у вас без расширенных материалов модуля вспомнить все существенные технические детали.

Задание 1. Подготовьте шаблон виртуальной машины для тестового стенда

1.1. Настройте натированную сеть для тестового стенда

Для создания натированной сети VirtualBox нужно выполнить следующие действия:

- Открыть «Менеджер сети» из меню *Файл → Инструменты*.
- Перейти на вкладку «Сеть NAT» и нажать кнопку *Создать*.
- В настройках сети задать следующие параметры:
  - Имя сети: **Сеть - Стенд ALD Pro 2.4.0**.
  - CIDR сети (диапазон адресов): **10.0.1.0/24**.
  - Поддержка DHCP: **Отключаем DHCP**.

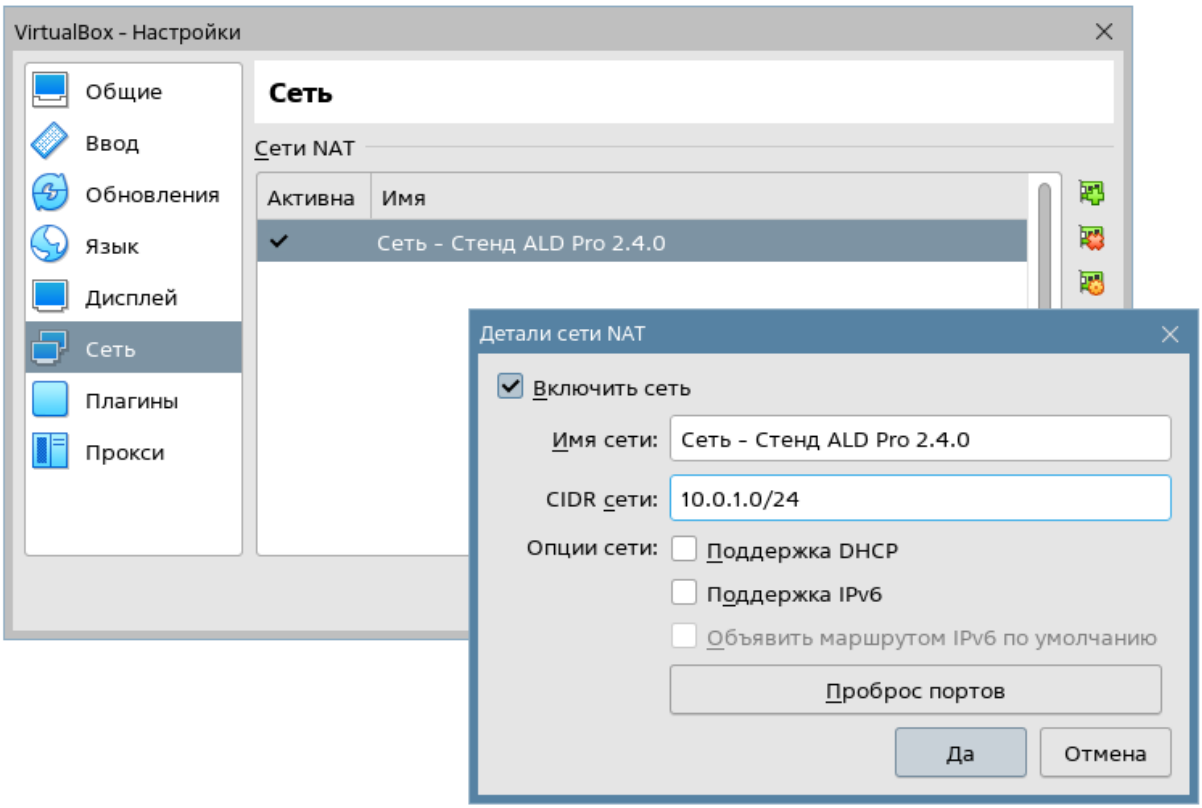


рис. 2.33 – Создание сети NAT

1.2. Создайте виртуальную машину для первого контроллера домена

В окне VirtualBox Manager выполните команду *Машина → Создать* и укажите следующие параметры:

- Имя: **dc-1.ald.company.lan - Стенд ALD Pro 2.4.0**.
- Папка машины: **по умолчанию**.
- Тип: **Linux**.
- Версия: **Other Linux (64-bit)**.
- Объем памяти: **4096 МБ**.
- Жесткий диск: **Создать новый виртуальный жесткий диск**.
- Тип файла: **VDI (VirtualBox Disk Image)**.
- Формат хранения: **Динамический виртуальный жесткий диск**.
- Путь: **по умолчанию**.
- Размер: **50 ГБ**.

После создания виртуальной машины в ее настройках задайте следующие параметры:

- На вкладке *Система → Процессор* укажите **4 ЦП** для ускорения установки ОС, далее можно уменьшить этот параметр до **2 ЦП**.
- На вкладке *Система → Ускорение* укажите **KVM**.
- На вкладке *Носители* для компакт-диска укажите установочный файл installation-1.7.6.11-26.08.24_17.26.iso.
- На вкладке *Сеть* выберите тип подключения **Сеть NAT** и имя **Сеть - Стенд ALD Pro 2.4.0**.

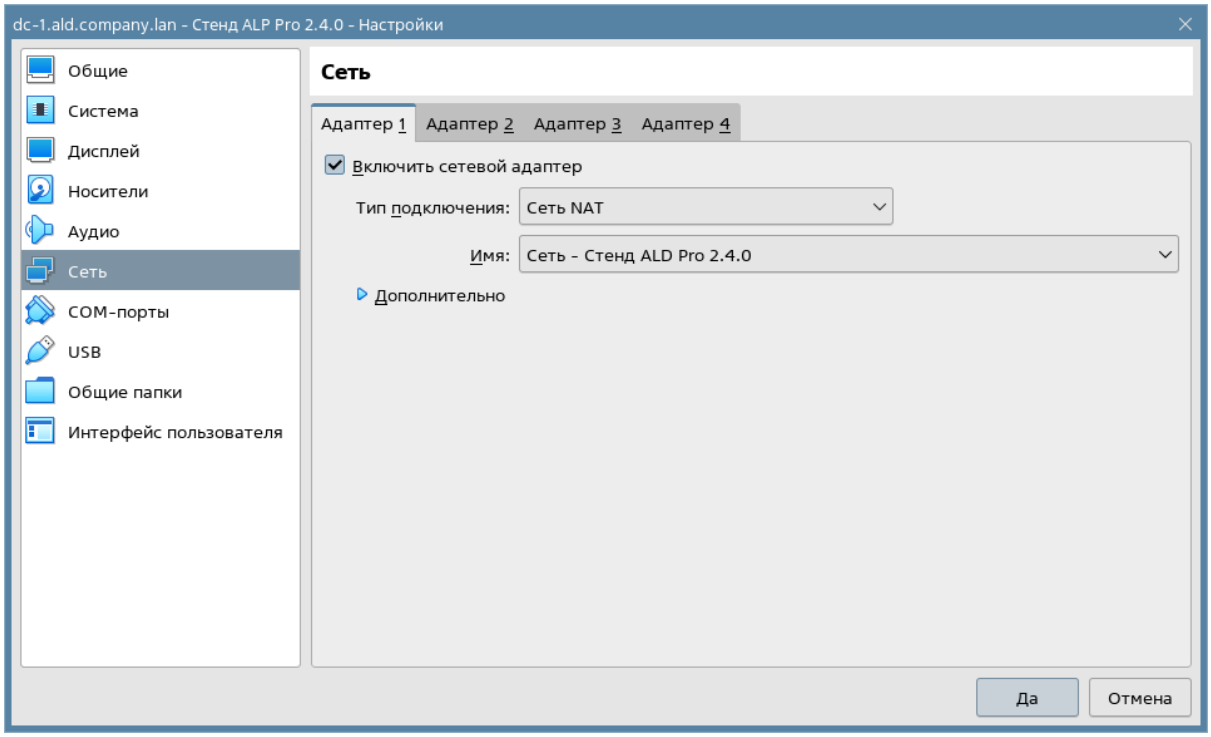


рис. 2.34 – Настройки сети VM

1.3. Установите операционную систему

Загрузите виртуальную машину и установите операционную систему с графическим окружением и максимальным уровнем защищенности «Смоленск»:

- Имя компьютера оставьте по умолчанию **astra**.
- Локальным администратором укажите пользователя **localadmin**.
- В качестве пароля локального администратора установите значение **Pa$$w0rd**.
- Настройку сети пропустите.
- При разметке диска выберите пункт **Авто – использовать весь диск и настроить LVM** (потре-

буется в будущем при обновлении ОС до версии ALSE 1.8+) и схему разметки **Все файлы в одном разделе**.

- Оставьте версию ядра по умолчанию **6.1-generic**. Ядра hardened (усиленная самозащита) и lowlatency (с увеличенной до 1000 Гц частотой переключения задач) продукт ALD Pro не поддерживает.

- Из пакетов программ по умолчанию можно исключить, например, **Средства работы с графикой**, но обязательно оставьте **Офисные приложения**, так как в дальнейшем нам потребуется LibreOffice для просмотра CSV-файлов в табличном виде. Добавьте также пакет **Средство удаленного подключения SSH**, если вы хотите подключаться к компьютеру удаленно.

- При выборе дополнительных параметров укажите уровень защищенности **Смоленск** и оставьте включенными флажки **Мандатный контроль целостности** и **Мандатное управление доступа**. В тестовых средах для удобства работы можно отключить флажок **Запрос пароля для команды sudo**.

- При настройке GRUB используйте тот же пароль **Pa$$w0rd**.

После первой загрузки системы вы можете проверить ее версию и уровень защищенности, используя следующие команды:

```
cat /etc/astra/build_version
sudo astra-modeswitch getname
```

Результат выполнения команд выделен цветом:

```
localadmin@astra:~$ cat /etc/astra/build_version
1.7.6.11

localadmin@astra:~$ sudo astra-modeswitch getname
[sudo] пароль для localadmin:
maximum(smolensk)
```

1.4. Сделайте временную настройку сетевого интерфейса

Сделайте временную настройку сетевого интерфейса через апплет NetworkManager со следующими параметрами:

- IP-адрес: **10.0.1.10**.

- Маска: **24** или **255.255.255.0**.

- Шлюз: **10.0.1.1**.

- Сервер DNS: **77.88.8.8**.

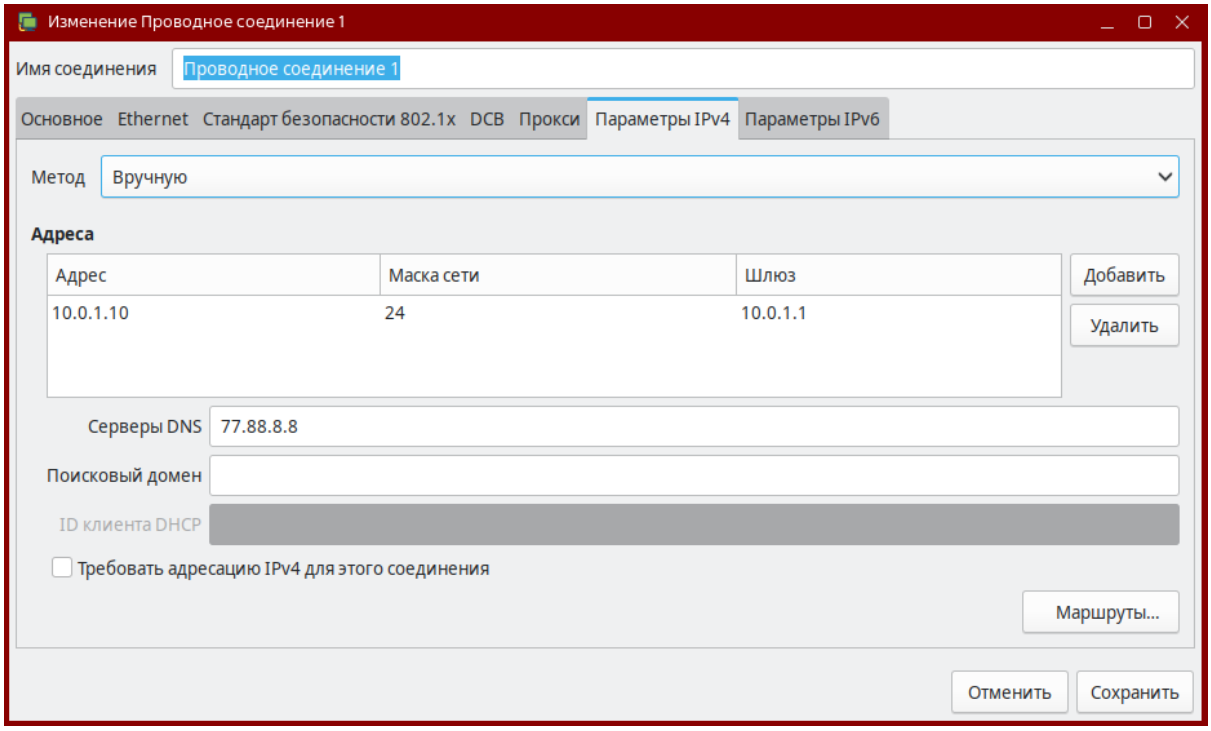


рис. 2.35 – Настройка сетевого интерфейса с помощью апплета Network Manager

Если сетевые настройки не применились, то выключите и включите обратно поддержку сети для применения изменений.

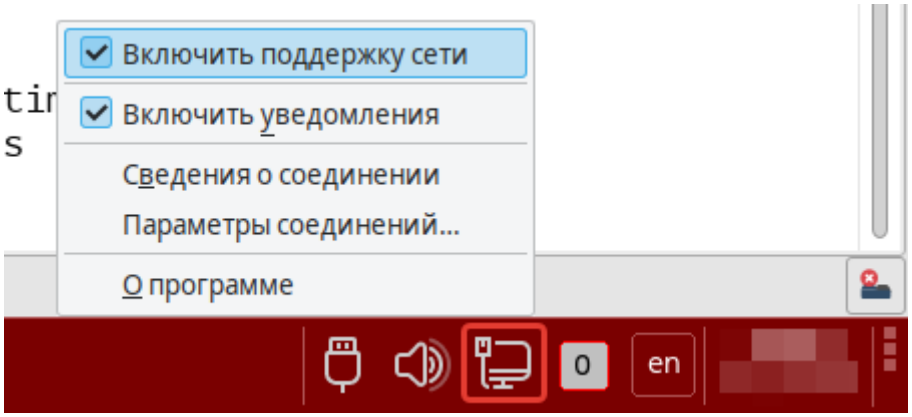


рис. 2.36 – Включение/отключение поддержки сети

Чтобы убедиться в работе сетевого стека, выполните следующие команды:

```
ip a show dev eth0
ping dl.astralinux.ru -c 4
```

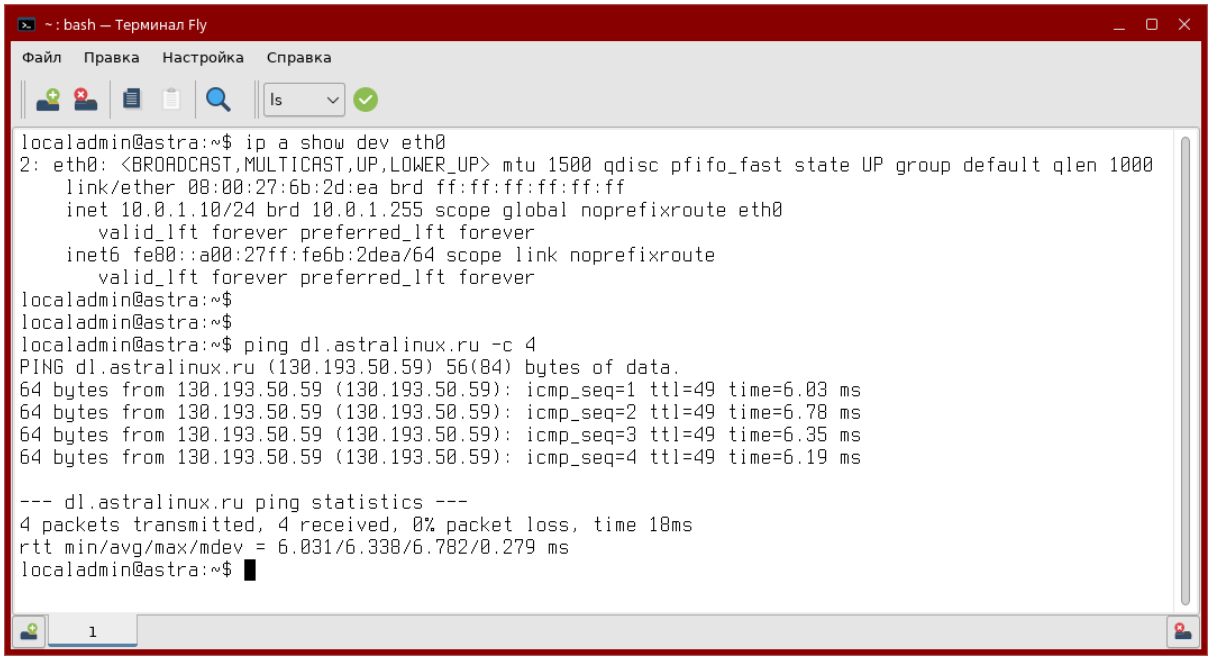


рис. 2.37 – Проверка работы сети

1.5. Настройте репозитории операционной системы

Вы можете редактировать файл `/etc/apt/sources.list` напрямую или запустить редактор с помощью следующей команды, чтобы воспользоваться дополнительными проверками при сохранении изменений:

```
sudo apt edit-sources
```

Для операционной системы ALSE 1.7.6.11 установите следующее содержимое файла `/etc/apt/sources.list`:

```
deb https://dl.astralinux.ru/astra/frozen/1.7_x86-64/1.7.6/repository-base/ 1.7_x86-
↪64 main contrib non-free
deb https://dl.astralinux.ru/astra/frozen/1.7_x86-64/1.7.6/repository-extended/ 1.7_
↪x86-64 main contrib non-free
```

После изменения состава репозиториев следует обновить индекс доступных пакетов с помощью команды `sudo apt update`. Эта команда извлекает список пакетов и зависимостей, но еще не обновляет само программное обеспечение. Вот результат ее выполнения:

```
localadmin@astra:~$ sudo apt update
Пол:1 https://dl.astralinux.ru/astra/frozen/1.7_x86-64/1.7.6/repository-base 1.7_x86-
↪64 InRelease [5 306 B]
Пол:2 https://dl.astralinux.ru/astra/frozen/1.7_x86-64/1.7.6/repository-extended 1.7_
↪x86-64 InRelease [9 772 B]
...
Получено 6 589 kB за 1с (5 402 kB/s)
Чтение списков пакетов... Готово
```

(continues on next page)

(продолжение с предыдущей страницы)

```
Построение дерева зависимостей
Чтение информации о состоянии... Готово
Может быть обновлён 1 пакет. Запустите «apt list --upgradable» для показа.
```

Выполните обновление пакетов через dist-upgrade или sudo astra-update -a -r :

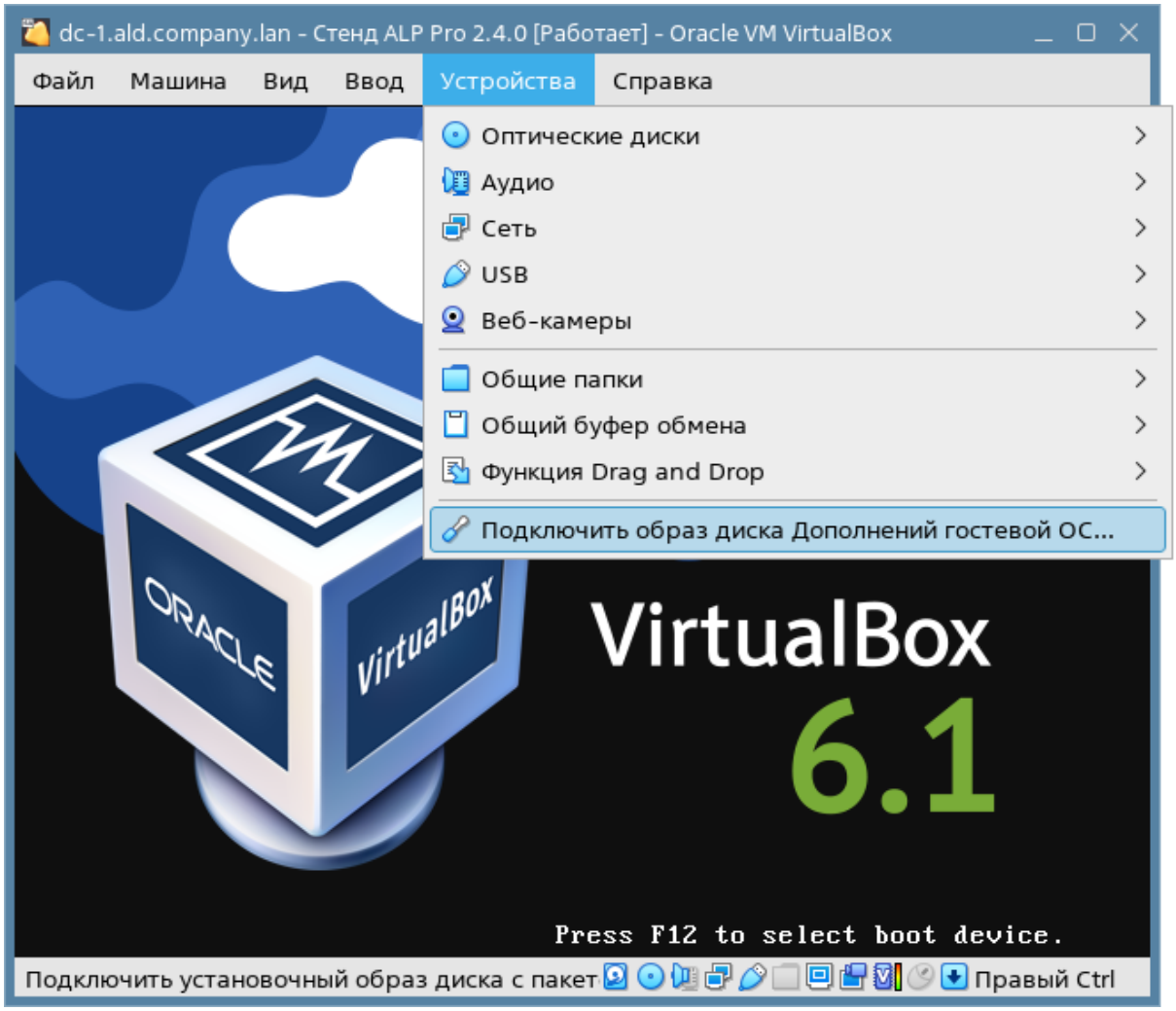
```
sudo apt dist-upgrade -y -o Dpkg::Options::=--force-confold
```

Результат выполнения команды:

```
localadmin@astra:~$ sudo apt dist-upgrade -y -o Dpkg::Options::=--force-confold
Чтение списков пакетов... Готово
Построение дерева зависимостей
Чтение информации о состоянии... Готово
Расчёт обновлений... Готово
Следующие пакеты устанавливались автоматически и больше не требуются:
libbluetooth3 libmm-glib0 libndp0 libnma-common libnma0 libpkcs11-helper1␣
↪libteamdctl0 libxsettings-client0 openvpn
Для их удаления используйте «sudo apt autoremove».
Следующие пакеты будут обновлены:
astra-version
Обновлено 1 пакетов, установлено 0 новых пакетов, для удаления отмечено 0 пакетов, и␣
↪0 пакетов не обновлено.
Необходимо скачать 8 376 B архивов.
После данной операции объём занятого дискового пространства возрастёт на 20,5 kB.
Пол:1 https://dl.astralinux.ru/astra/frozen/1.7_x86-64/1.7.6/repository-extended 1.7_
↪x86-64/main amd64 astra-version amd64 8.1.46+v1.7.6.ext2.1 [8 376 B]
Получено 8 376 B за 0c (87,3 kB/s)
(Чтение базы данных ... на данный момент установлено 220649 файлов и каталогов.)
Подготовка к распаковке .../astra-version_8.1.46+v1.7.6.ext2.1_amd64.deb ...
Распаковывается astra-version (8.1.46+v1.7.6.ext2.1) на замену (8.1.43+v1.7.6.11) ...
Настраивается пакет astra-version (8.1.46+v1.7.6.ext2.1) ...
Обрабатываются триггеры для man-db (2.8.5-2+b1) ...
Обрабатываются триггеры для xserver-xorg-core (2:21.1.7-1ubuntu4astra.se28) ...
update exec ids due to /usr/bin changed
```

1.6. Установите гостевые дополнения

Подключите и примонтируйте диск с гостевыми дополнениями, выбрав меню *Устройства* → *Подключить образ диска Дополнений гостевой ОС* :

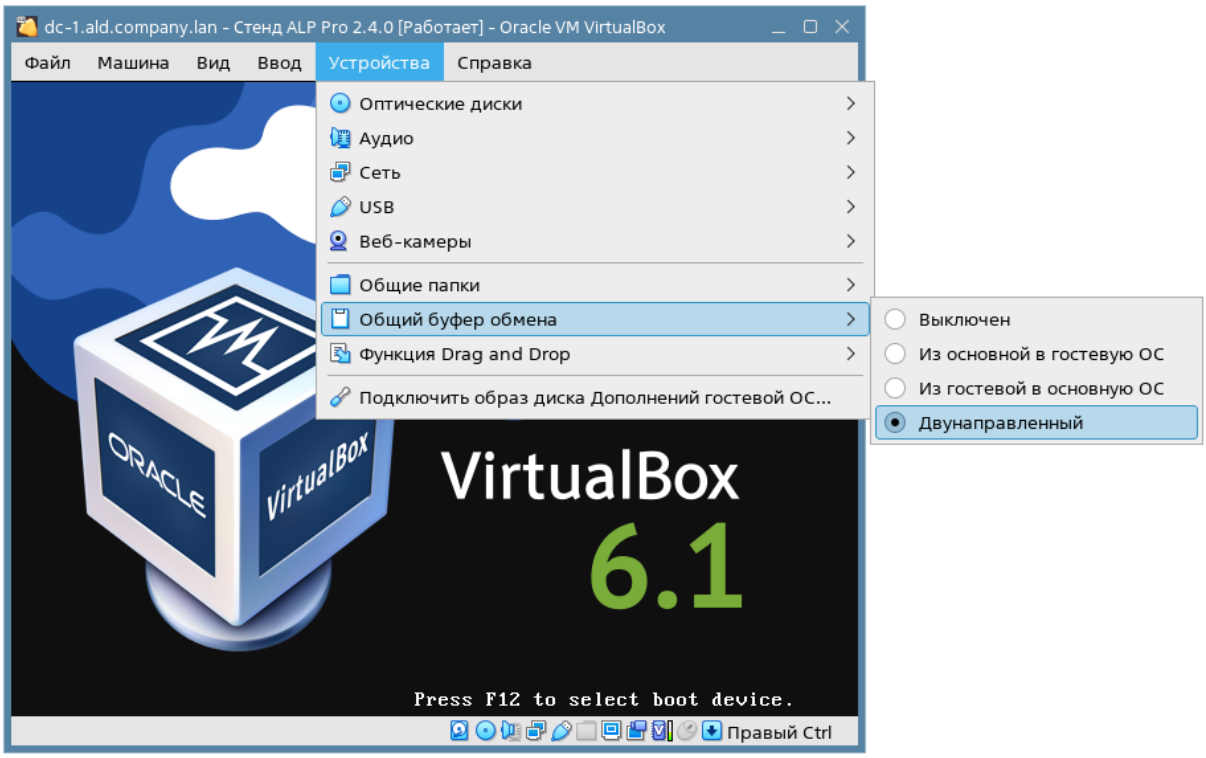


Примечание: Диск с гостевыми дополнениями можно скачать по ссылке с официального сайта.

Установите **VBoxLinuxAdditions** следующими командами:

```
sudo apt install gcc make linux-headers-6.1
sudo sh /media/cdrom0/VBoxLinuxAdditions.run
```

Осталось в настройках машины включить двунаправленный буфер обмена и дополнения заработают при следующей загрузке системы.



1.7. Клонируйте шаблон виртуальной машины

Перед выполнением клонирования рекомендуется завершить работу операционной системы из меню *Пуск → Завершение работы → Выключение* или командой `poweroff` утилиты `systemctl`:

```
systemctl poweroff
```

Запустите мастер клонирования командой *Машина → Клонировать* из VirtualBox Manager. При выполнении клонирования укажите следующие параметры:

- Имя: **Шаблон - Стенд ALD Pro 2.4.0**.
- Путь: **Оставьте по умолчанию**.
- Тип клонирования: **Полное клонирование**.
- Цель клонирования: **Состояние машины**.
- Политика MAC-адреса: **Сгенерировать новые MAC-адреса всех сетевых адаптеров**. Мы, конечно, не планируем загружать виртуальную машину непосредственно из шаблона, но лучше сразу сгенерировать новый MAC-адрес, чтобы полностью исключить вероятность конфликта адресов в будущем.

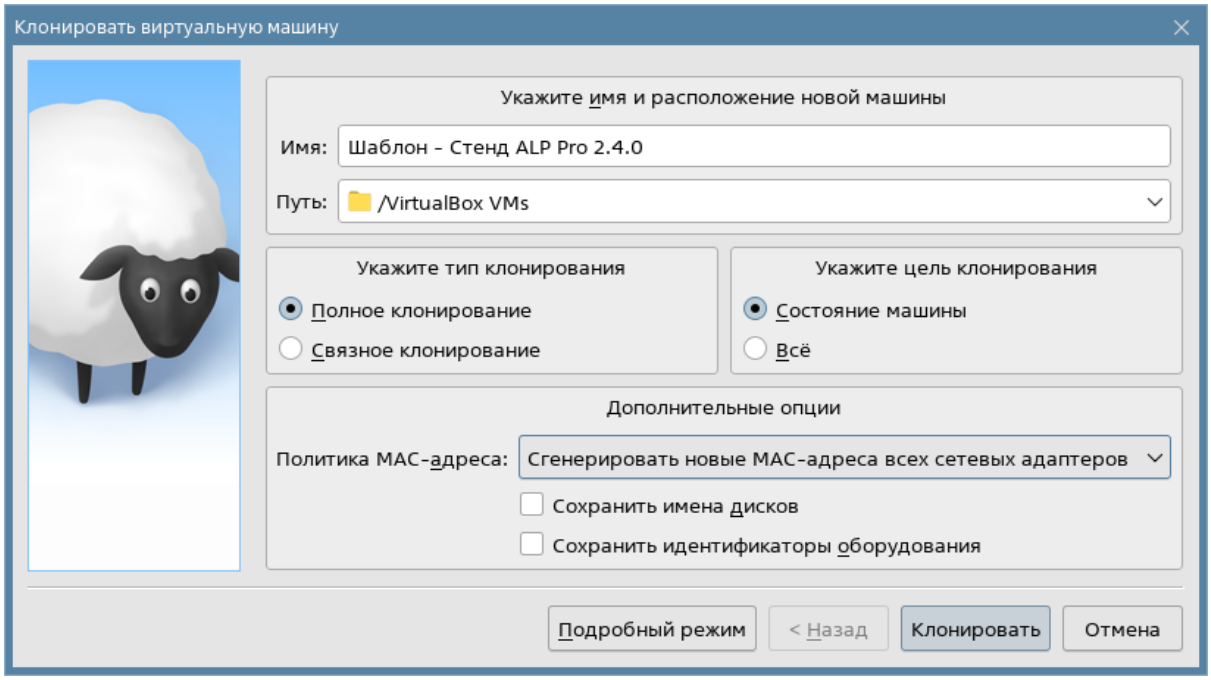


рис. 2.38 – Клонирование шаблона виртуальной машины

Задание 2. Разверните первый контроллер домена

2.1. Настройте сетевой интерфейс для постоянной работы

Отключить службу **NetworkManager** следующими командами:

```
sudo systemctl stop NetworkManager
sudo systemctl disable NetworkManager
sudo systemctl mask NetworkManager
systemctl status NetworkManager
```

После выполнения последней команды `status` в терминале отобразится информация о том, что служба NetworkManager остановлена (inactive), и запустить ее не получится, т.к. она замаскирована (masked):

```
...
localadmin@astra:~@ systemctl status NetworkManager
 o NetworkManager.service
   Loaded: masked (Reason: Unit NetworkManager.service is masked.)
   Active: inactive (dead) since Mon 2024-01-15 11:49:50 MSK; 1s ago
 Main PID: 509 (code=exited, status=0/SUCCESS)
      CPU: 264ms
```

Завершить работу аплета и удалить пакет network-manager следующими командами:

```
sudo kill $(pidof nm-applet)
sudo apt remove network-manager
```

Для настройки сетевого интерфейса вы можете открыть файл `/etc/network/interfaces` напрямую:

```
sudo nano /etc/network/interfaces
```

Содержимое файла после редактирования должно стать следующим (новые строки отмечены цветом):

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# Настройка eth0
auto eth0
iface eth0 inet static
  address 10.0.1.11
  netmask 255.255.255.0
  gateway 10.0.1.1
```

Чтобы применить новые настройки, достаточно перезапустить службу networking командой `systemctl restart networking`. Может потребоваться также очистить старое соединение командой `ip addr flush dev <имя устройства>`:

```
sudo ip addr flush dev eth0
sudo systemctl restart networking
```

Проверить работу сети можно с помощью утилиты `ping`:

```
localadmin@astra:~$ ping 77.88.8.8 -c 4
PING 77.88.8.8 (77.88.8.8) 56(84) bytes of data.
64 bytes from 77.88.8.8: icmp_seq=1 ttl=54 time=91.3 ms
64 bytes from 77.88.8.8: icmp_seq=2 ttl=54 time=91.6 ms
. . .
```

Удалить комментарий от NetworkManager из файла `/etc/resolv.conf`, содержимое файла должно быть следующим:

```
localadmin@astra:~$ cat /etc/resolv.conf
nameserver 77.88.8.8
```

Проверьте разрешение имен с помощью утилиты `ping`, обращаясь к серверу по имени:

```
localadmin@astra:~$ ping dl.astralinux.ru -c 4
PING dl.astralinux.ru (51.250.6.116) 56(84) bytes of data.
64 bytes from 51.250.6.116 (51.250.6.116): icmp_seq=1 ttl=52 time=31.9 ms
64 bytes from 51.250.6.116 (51.250.6.116): icmp_seq=2 ttl=52 time=27.2 ms
. . .
```

2.2. Настройте репозитории продукта ALD Pro

Чтобы подключить репозиторий ALD Pro версии 2.4.0 нужно открыть файл `/etc/apt/sources.list.d/aldpro.list` на редактирование командой:

```
sudo nano /etc/apt/sources.list.d/aldpro.list
```

И сохранить в нем следующую строку:

```
deb https://dl.astralinux.ru/aldpro/frozen/01/2.4.0 1.7_x86-64 main base
```

После изменение настроек репозиториев не забудьте обновить индекс пакетов:

```
sudo apt update
```

2.3. Настройте имя контроллера домена

Выполните команду:

```
sudo hostnamectl set-hostname dc-1.ald.company.lan
```

Приведите содержимое файла `/etc/hosts` к следующему состоянию:

```
127.0.0.1 localhost.localdomain localhost
10.0.1.11 dc-1.ald.company.lan dc-1
```

2.4. Установите пакеты контроллера домена

Выполните следующую команду для установки контроллера домена с модулями глобального каталога и синхронизации:

```
sudo DEBIAN_FRONTEND=noninteractive \
  apt-get install -y -q aldpro-mp aldpro-gc aldpro-syncer
```

Прежде чем продолжить, ознакомьтесь с журналом пакетного менеджера в файле `/var/log/apt/term.log` на наличие ошибок:

```
sudo grep 'error:' /var/log/apt/term.log
```

2.5. Выполните повышение роли сервера до контроллера домена

Выполните повышение роли сервера следующими командами:

```
set +o history # Отключаем запись истории команд

sudo aldpro-server-install -n dc-1 -d ald.company.lan -p 'Pa$$w0rd' \
```

(continues on next page)

(продолжение с предыдущей страницы)

```
  --ip 10.0.1.11 --setup_gc --setup_syncer --no-reboot

set -o history # Включаем запись истории команд
```

По завершению установки вы увидите сообщение об успешном выполнении операции:

```
...
------------
Succeeded: 5 (changed=5)
Failed:    0
------------
Total states run:     5
Total run time: 331.382 ms
localadmin@dc-1:~$
```

Убедитесь, что теперь DNS-запросы обрабатываются собственной службой Bind9:

```
$ cat /etc/resolv.conf
# auto-generated by IPA installer
search ald.company.lan
nameserver 127.0.0.1
```

Контроллер домена готов к работе, осталось только перезагрузить сервер:

```
sudo reboot
```

После загрузки контроллера выберите в первом выпадающем списке значение **ald.company.lan** вместо **Этот компьютер** и войдите в систему из-под учетной записи доменного администратора `admin` с высоким уровнем целостности `63`, который будет предложен по умолчанию:

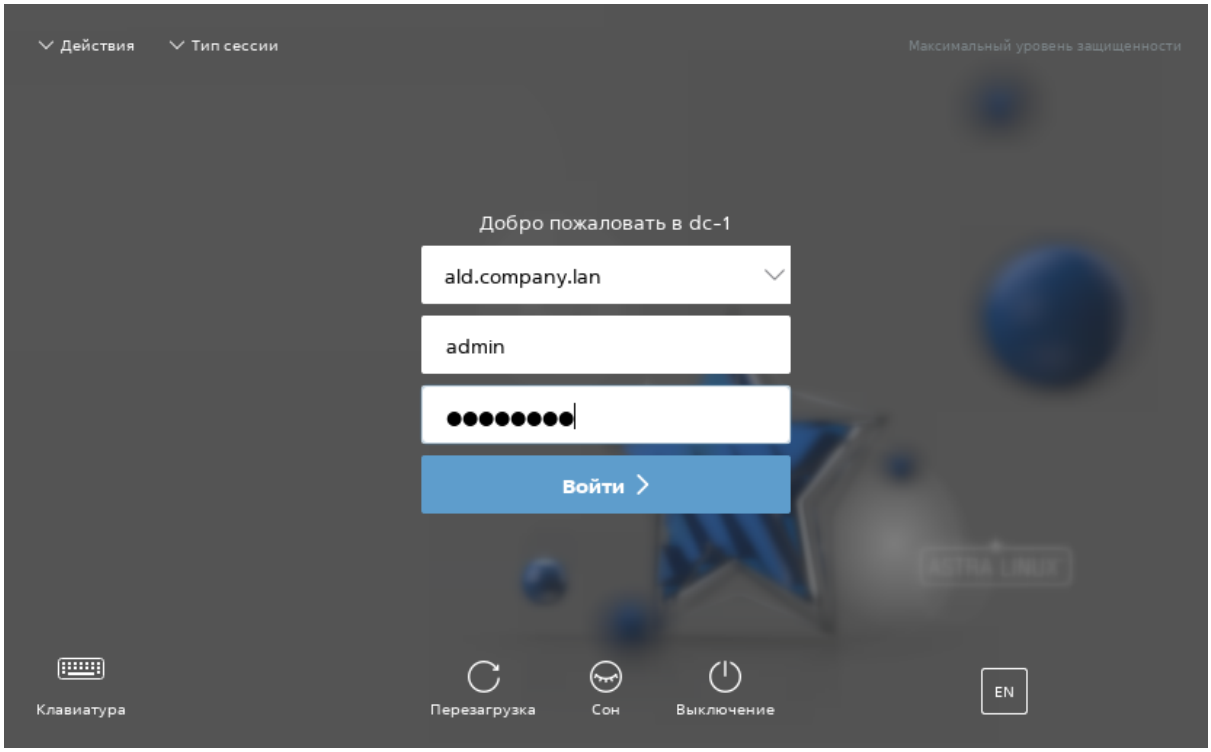


рис. 2.39 – Вход в систему под доменной учетной записью

2.6. Проверьте базовые функции домена

2.6.1. Проверьте доменные службы

Проверьте в первую очередь доступность клиентской службы SSSD:

```
systemctl status sssd
```

Результат проверки:

```
localadmin@dc-1:~$ systemctl status sssd
o sssd.service - System Security Services Daemon
   Loaded: loaded (/lib/systemd/system/sssd.service; enabled; vendor preset: enabled)
   Active: active (running) since Sat 2025-04-07 09:49:16 MSK; 17min ago
...
```

При необходимости перезапустите службу командой `sudo systemctl restart sssd`, повторите вход в систему доменным пользователем еще раз и проверьте новые записи в журналах `/var/log/auth.log`` и `/var/log/messages`.

Далее проверьте на сервере состояние доменных служб утилитой `ipactl`:

```
sudo ipactl status
```

Результат проверки:

```
localadmin@dc-1:~$ sudo ipactl status
Directory Service: RUNNING
krb5kdc Service: RUNNING
kadmin Service: RUNNING
named Service: RUNNING
httpd Service: RUNNING
ipa-custodia Service: RUNNING
smb Service: RUNNING
winbind Service: RUNNING
ipa-otpd Service: RUNNING
ipa-dnskeysyncd Service: RUNNING
ipa: INFO: The ipactl command was successful
```

При необходимости перезапустите доменные службы командой `sudo ipactl restart`, повторите вход в систему доменным пользователем еще раз и проверьте новые записи в журналах `/var/log/auth.log`` и `/var/log/messages`.

2.6.2. Проверьте Kerberos-аутентификацию

При успешном входе в систему доменным пользователем в связке ключей должен появиться **TGT-билет**, что можно проверить с помощью команды `klist`:

```
admin@dc-1:~$ klist
Ticket cache: KEYRING:persistent:1902000000:krb_ccache_BUoysxG
Default principal: admin@ALD.COMPANY.LAN

Valid starting       Expires              Service principal
07.04.2025 12:29:42  08.04.2025 12:01:56  krbtgt/ALD.COMPANY.LAN@ALD.COMPANY.LAN
```

Для уничтожения билетов используется команда `kdestroy`, после выполнения которой мы увидим, что в связке ключей кэш больше не найден:

```
admin@dc-1:~$ kdestroy
admin@dc-1:~$ klist
klist: Credentials cache keyring 'persistent:1902000000:krb_ccache_BUoysxG' not found
```

Для запроса нового **TGT-билета** на доменного пользователя `admin` воспользуемся утилитой `kinit`. Как мы видим, после прохождения успешной Kerberos-аутентификации нам был выдан новый **TGT-билет**:

```
admin@dc-1:~$ kinit admin
Password for admin@ALD.COMPANY.LAN: ********

admin@dc-1:~$ klist
Ticket cache: KEYRING:persistent:1902000000:krb_ccache_BUoysxG
Default principal: admin@ALD.COMPANY.LAN

Valid starting       Expires              Service principal
07.04.2025 12:31:42  08.04.2025 11:59:45  krbtgt/ALD.COMPANY.LAN@ALD.COMPANY.LAN
```

Выполните аутентификацию через алиас root, используя пароль пользователя admin:

```
admin@dc-1:~$ kdestroy
admin@dc-1:~$ sudo kinit
Password for root@ALD.COMPANY.LAN:

admin@dc-1:~$ klist
Ticket cache: KEYRING:persistent:1965600000:krb_ccache_o2glwGp
Default principal: root@ALD.COMPANY.LAN

Valid starting       Expires              Service principal
07.04.2025 13:16:02  08.04.2025 12:31:18  krbtgt/ALD.COMPANY.LAN@ALD.COMPANY.LAN
```

2.6.3. Проверьте портал управления и настройте домашнюю страницу

Для входа на портал управления откройте на контроллере домена страницу https://dc-1.ald.company.lan в браузере Mozilla Firefox и перейдите по ссылке для выполнения прозрачной Kerberos-аутентификации.

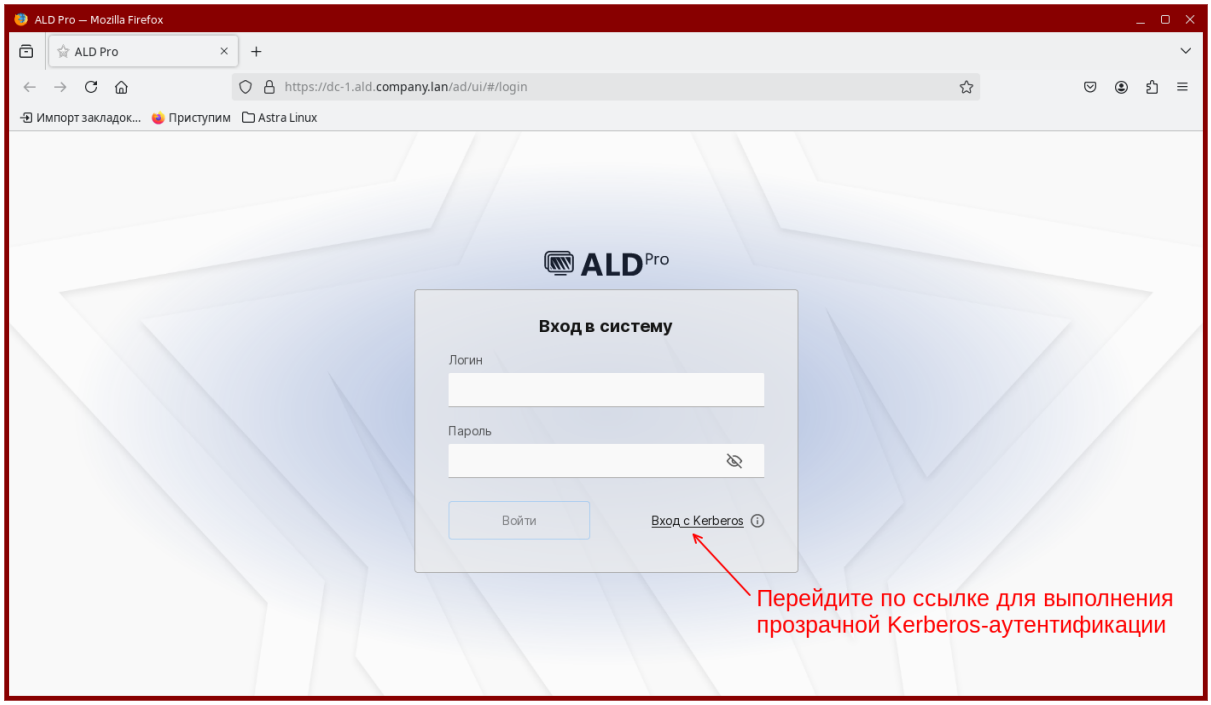


рис. 2.40 – Прозрачная Kerberos-аутентификация на портале

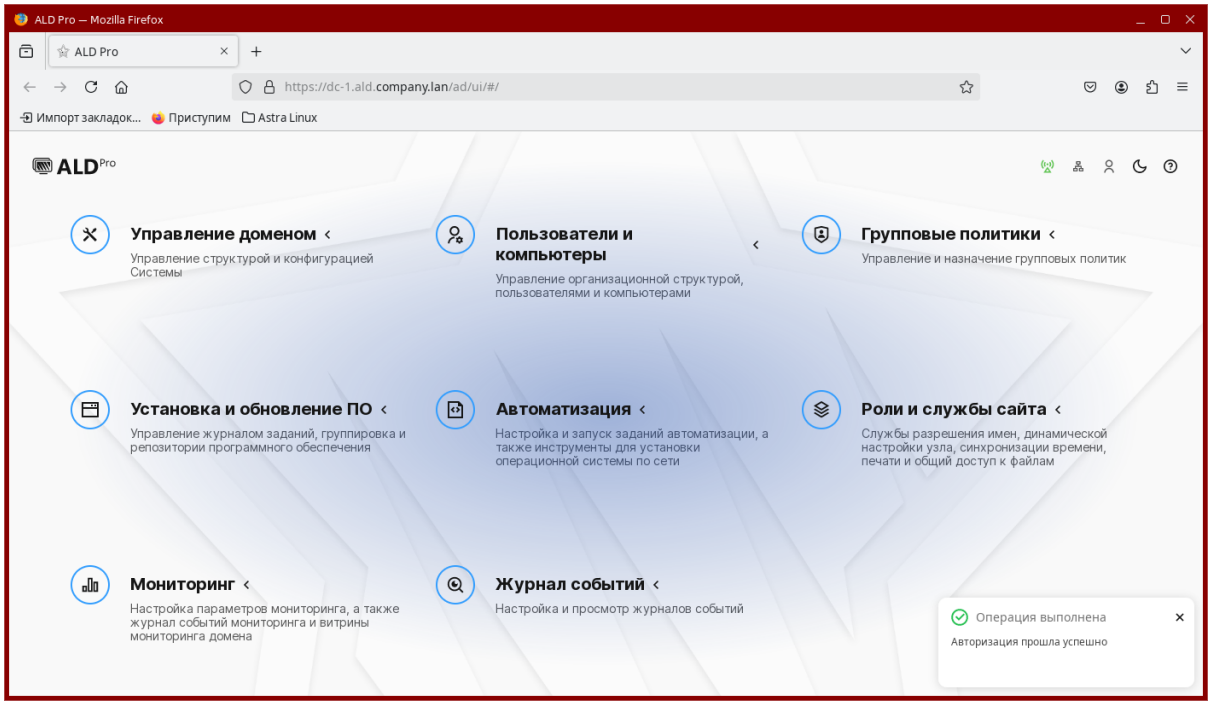


рис. 2.41 – Портал управления ALD Pro после успешного входа

Установите правильный адрес домашней страницы для браузера FireFox следующей командой:

```
sudo sed -i 's/dc-1\|dc-1.ald.company.lan/dc-1.ald.company.lan/g' \
    /usr/lib/firefox/distribution/policies.json
```

Содержимое файла после коррекции должно быть следующим:

```
admin@dc-1:~$ cat /usr/lib/firefox/distribution/policies.json
{
        "policies": {
              "BlockAboutAddons": true,
              "BlockAboutConfig": true,
              "Certificates": {
                      "ImportEnterpriseRoots": true,
                      "Install": ["/etc/ipa/ca.crt"]
              },

              "Homepage": {
                      "URL": "https://dc-1.ald.company.lan/",
                      "Locked": true,
                      "StartPage": "homepage-locked"
              }
        }
}
```

Проверьте, что в связке ключей текущего пользователя появится сервисный билет на доступ к службе `HTTP/dc-1.ald.company.lan`.

```
admin@dc-1:~$ klist
Ticket cache: KEYRING:persistent:1902000000:krb_ccache_BUoysxG
```

(continues on next page)

(продолжение с предыдущей страницы)

```
Default principal: admin@ALD.COMPANY.LAN

Valid starting       Expires              Service principal
07.04.2025 12:32:23  08.04.2025 11:59:45  HTTP/dc-1.ald.company.lan@ALD.COMPANY.LAN
07.04.2025 12:31:42  08.04.2025 11:59:45  krbtgt/ALD.COMPANY.LAN@ALD.COMPANY.LAN
```

Выполните очистку кэша командой `kdestroy`, откройте еще одно окно браузера FireFox и убедитесь, что доступ к порталу будет предоставлен без повторной Kerberos-аутентификации, поэтому валидный TGT-билет не потребуется и сервисный билет на доступ к HTTP-службе в связке ключей не появится.

2.7. Выполните дополнительную настройку DNS-службы

2.7.1. Отключите DNSSEC и запретите рекурсивные запросы

Откройте файл `/etc/bind/ipa-options-ext.conf` на редактирование командой:

```
sudo nano /etc/bind/ipa-options-ext.conf
```

И приведите его содержание к следующему виду:

```
dnssec-validation no;
allow-recursion { any; };
allow-query-cache { any; };
```

Для применения изменений необходимо перезапустить DNS-службу на контроллере домена:

```
sudo systemctl restart bind9-pkcs11.service
```

2.7.2. Настройте глобальный перенаправитель DNS-запросов

Откройте на портале управления страницу *Роли и службы сайта → Служба разрешения имен → Глобальная конфигурация DNS*.

Нажмите кнопку *Добавить* в списке глобальных перенаправителей и введите адрес DNS-сервера от Яндекс **77.88.8.8**.

В качестве политики перенаправления выберите «**Только перенаправлять**», в этом случае BIND9 будет запрашивать ответ только у перенаправителя.

По завершению редактирования настроек нажмите кнопку *Сохранить*, чтобы изменения вступили в силу.

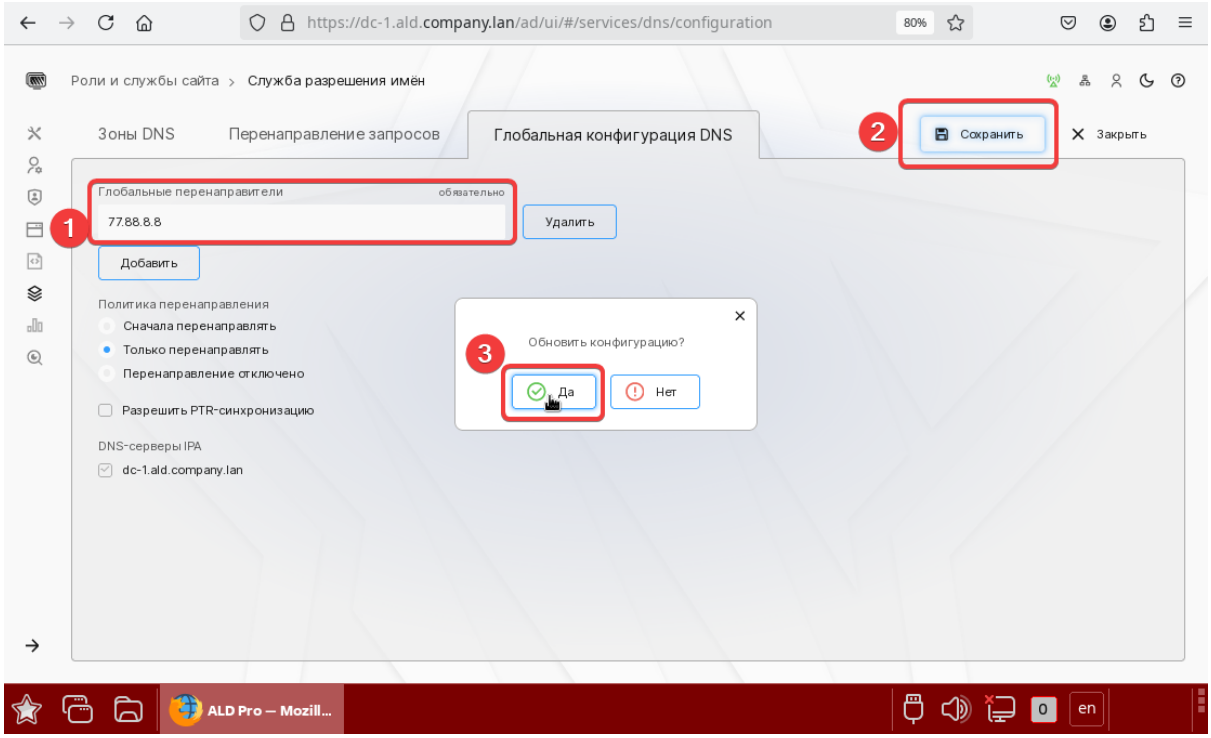


рис. 2.42 – Глобальная конфигурация DNS

Проверить настройки DNS-службы можно из командной строки:

```
ipa dnsconfig-show
```

Результат выполнения команды:

```
admin@dc-1:~$ ipa dnsconfig-show
Глобальные перенаправители: 77.88.8.8
Политика перенаправления: only
Разрешить PTR-синхронизацию: FALSE
DNS-серверы IPA: dc-1.ald.company.lan
```

Установим пакет tcpdump командой `sudo apt install tcpdump` и запустим в отдельном окне терминала перехват всех пакетов, в которых фигурирует 53-й порт:

```
sudo tcpdump port 53
```

Осталось проверить, что будет происходить при преобразовании DNS-имени **dl.astralinux.ru** в IP-адрес, например, с помощью утилит `host`, `ping`, `nslookup` или `dig`:

```
host dl.astralinux.ru
```

В окне сниффера можно увидеть, что запрос был направлен на глобальный перенаправитель **dns.yandex.ru**, и в ответ был получен IP-адрес **130.193.50.59**:

```
$ sudo tcpdump port 53
```

(continues on next page)

(продолжение с предыдущей страницы)

```
...
15:36:12.154912 IP dc-1.ald.company.lan.41938 > dns.yandex.ru.domain: 5409+% [1au]␣
↪PTR? 8.8.88.77.in-addr.arpa. (63)
15:36:12.160096 IP dns.yandex.ru.domain > dc-1.ald.company.lan.40646: 62590 1/0/1 A␣
↪130.193.50.59 (61)
...
```

Задание 3. Присоедините компьютер к домену

3.1. Подготовьте виртуальную машину

Для создания рабочей станции выберите виртуальную машину **Шаблон - Стенд ALD Pro 2.4.0** и запустите мастер клонирования командой *Машина → Клонировать* из меню VirtualBox Manager. При выполнении клонирования укажите следующие параметры:

- Имя: **pc-1.ald.company.lan - Стенд ALD Pro 2.4.0**.
- Путь: **Оставьте по умолчанию**.
- Тип клонирования: **Полное клонирование**.
- Цель клонирования: **Состояние машины**.
- Политика MAC-адреса: **Сгенерировать новые MAC-адреса всех сетевых адаптеров**.

После клонирования машины оптимизируйте объем вычислительных ресурсов, изменив требуемые параметры RAM и CPU.

Сгенерируем новый идентификатор машины, чтобы при необходимости использования динамической настройки от DHCP она получала собственный IP-адрес:

```
sudo rm -fv /etc/machine-id
sudo rm -fv /var/lib/dbus/machine-id
sudo dbus-uuidgen --ensure
sudo systemd-machine-id-setup
```

Сгенерируем новые ssh-ключи, чтобы у всех компьютеров они были разными:

```
sudo rm -fv /etc/ssh/ssh_host_*
sudo ssh-keygen -A -v
```

3.2. Настройте сетевой интерфейс для постоянной работы

Откройте настройки сети в апплете NetworkManager и на вкладке «Параметры IPv4» установите значения:

- Метод: **Вручную**.
- Адрес: **10.0.1.51** (в соответствии с описанием тестового стенда).
- Маска: **24** или **255.255.255.0**.
- Шлюз: **10.0.1.1** (адресом шлюза VirtualBox является первый IP-адрес сети).
- Серверы DNS: **10.0.1.11** (адрес dc-1.ald.company.lan).
- Поисковый домен: **ald.company.lan** (это значение будет добавляться к коротким именам).

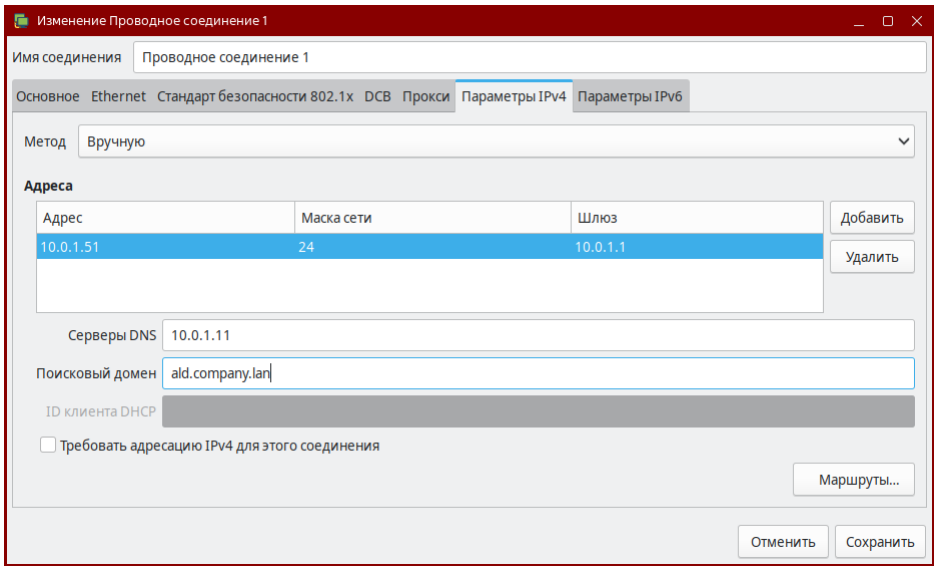


рис. 2.43 – Настройка сети на рабочей станции

Для применения новых настроек необходимо выключить и повторно включить поддержку сети в контекстном меню апплета.

3.3. Настройте репозитории продукта ALD Pro

Установите для файла `/etc/apt/sources.list` следующее содержимое:

```
deb https://dl.astralinux.ru/astra/frozen/1.7_x86-64/1.7.6/repository-base/ 1.7_x86-
↪64 main contrib non-free
deb https://dl.astralinux.ru/astra/frozen/1.7_x86-64/1.7.6/repository-extended/ 1.7_
↪x86-64 main contrib non-free
```

Установите для файла `/etc/apt/sources.list.d/aldpro.list` следующее содержимое:

```
deb https://dl.astralinux.ru/aldpro/frozen/01/2.4.0 1.7_x86-64 main base
```

После изменение настроек репозиториев не забудьте обновить индекс пакетов:

```
sudo apt update
```

3.4. Установите пакеты клиентской части

Теперь система готова к установке пакетов клиентской части ALD Pro. Для этого выполните команду:

```
sudo DEBIAN_FRONTEND=noninteractive \
  apt-get install -y -q aldpro-client
```

3.5. Проверьте имя хоста на уникальность

Проверьте желаемое имя на уникальность можно с помощью утилиты `host`:

```
admin@dc-1:~$ host pc-1.ald.company.lan
Host pc-1.ald.company.lan not found: 3(NXDOMAIN)
```

3.6. Выполните ввод компьютера в домен

Выполним присоединение компьютера к домену из консоли следующими командами:

```
set +o history # Отключаем запись истории команд

sudo /opt/rbta/aldpro/client/bin/aldpro-client-installer --host pc-1 \
  --domain ald.company.lan --account admin --password 'Pa$$w0rd' \
  --gui --force

set -o history # Включаем запись истории команд
```

Посмотрите ключи хоста из файла `/etc/krb5.keytab` с помощью утилиты `klist`:

```
localadmin@pc-1:~$ sudo klist -ket
Keytab name: FILE:/etc/krb5.keytab
KVNO Timestamp           Principal
---- ------------------- ------------------------------------------------------
   1 07.04.2025 14:36:35 host/pc-1.ald.company.lan@ALD.COMPANY.LAN (aes256-cts-hmac-
↪sha1-96)
   1 07.04.2025 14:36:35 host/pc-1.ald.company.lan@ALD.COMPANY.LAN (aes128-cts-hmac-
↪sha1-96)
```

Для того чтобы изменения вступили в силу, требуется выполнить перезагрузку компьютера:

```
sudo reboot
```

После перезагрузки войдите в систему, используя учетную запись администратора домена. Для первого входа в систему доменной учетной записью нужен будет доступ к контроллеру домена, а в дальнейшем проверка аутентичности возможна будет через кэш службы **sssd**. Подробно о том, как

служба SSSD встраивается в PAM и NSS для обеспечения работы компьютера в домене, смотрите *Модуль 5. Работа Linux-компьютера в домене*.

Задание 4. Выполните вывод компьютера из домена

Откатите настройки клиента FreeIPA командой `astra-freeipa-client` с ключом `-U`:

```
# Выполните команду на pc-1 под локальным администратором localadmin
sudo astra-freeipa-client -U -y
```

Удалите пакеты приложений:

```
sudo apt purge 'aldpro*' 'freeipa*' 'sssd*' 'krb5*'
sudo apt autoremove --purge
```

Удалите домашнюю директорию доменного администратора:

```
sudo rm -rf /home/admin
```

Далее следует удалить каталоги клиентских приложений подсистем ALD Pro и другие артефакты, созданные из скриптов, потому что пакетный менеджер ничего не знает об этих файлах и не сможет взять на себя ответственность за их удаление.

```
sudo rm -rf /var/lib/sss/pubconf/krb5.include.d/
sudo rm -rf /etc/krb5.conf.d/
sudo rm -rf /var/lib/sss/
sudo rm -rf /etc/sssd/
sudo rm -rf /opt/rbta/aldpro/
sudo rm -rf /etc/syslog-ng/aldpro/
sudo rm -rf /etc/salt/minion.d/
sudo rm -rf /var/log/salt/
sudo rm -rf /var/lib/certmonger/
sudo rm -rf /var/tmp/aldpro-salt.ccache
```

Далее следует удалить корневой сертификат домена из файла `/etc/ssl/certs/cacertificates.crt`, для этого нужно выполнить команду `update-ca-certificates`:

```
sudo update-ca-certificates
```

В завершение компьютеру нужно вернуть исходное имя, например, **astra**, и перезагрузить систему:

```
sudo hostnamectl set-hostname astra
sudo reboot
```

На контроллере домена удалите аккаунт компьютера и его DNS-записи из терминала следующими командами:

```
kinit admin
ipa dnsrecord-del ald.company.lan. pc-1 --del-all
ipa host-del pc-1.ald.company.lan
```

Результаты выполнения команд:

```
admin@dc-1:~$ kinit admin

admin@dc-1:~$ ipa dnsrecord-del ald.company.lan. pc-1 --del-all
---------------------
Удалена запись "pc-1"
---------------------

admin@dc-1:~$ ipa host-del pc-1.ald.company.lan
-----------------------------------
Удалён узел "pc-1.ald.company.lan"
-----------------------------------
```

Задание 5. Присоедините компьютер к домену графической утилитой

Для присоединения компьютера к домену выполните следующие действия:

1. Скачайте deb-пакет по прямой ссылке или из личного кабинета и установите пакет приложения следующей командой:

```
sudo DEBIAN_FRONTEND=noninteractive \
  apt-get install -y -q ./aldpro-join_2.0-2_amd64.deb
```

2. Запустите утилиту из меню *Пуск → ALD Pro → Ввод компьютера в домен ALD Pro v2.0*, ярлык выполняет команду `aldpro-join --gui`.

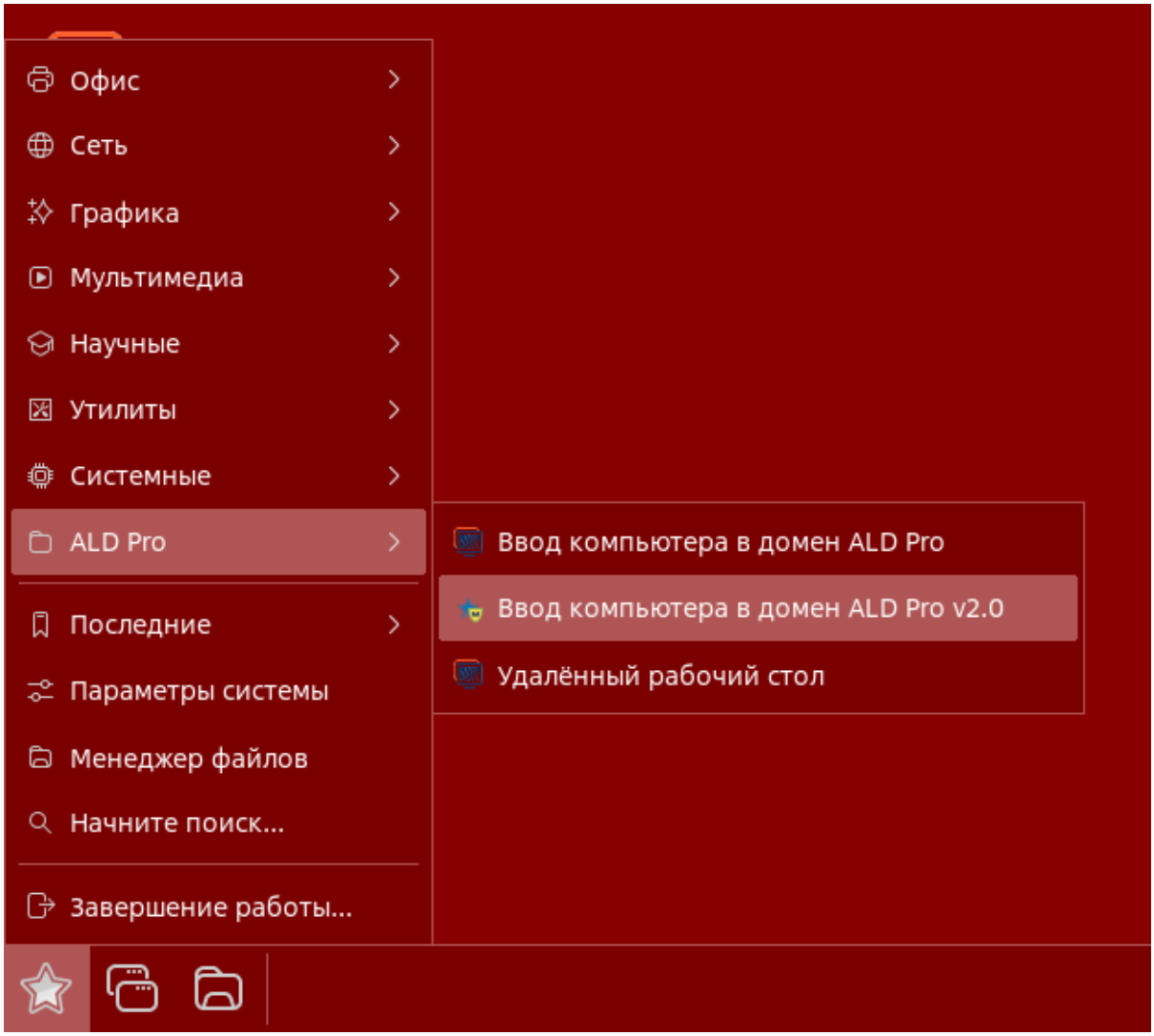


рис. 2.44 – Запуск приложения для присоединения к домену v2.0

3. Чтобы изменить имя компьютера, нажмите кнопку *Изменить...* напротив имени компьютера и установите желаемое значение, например, pc-14.

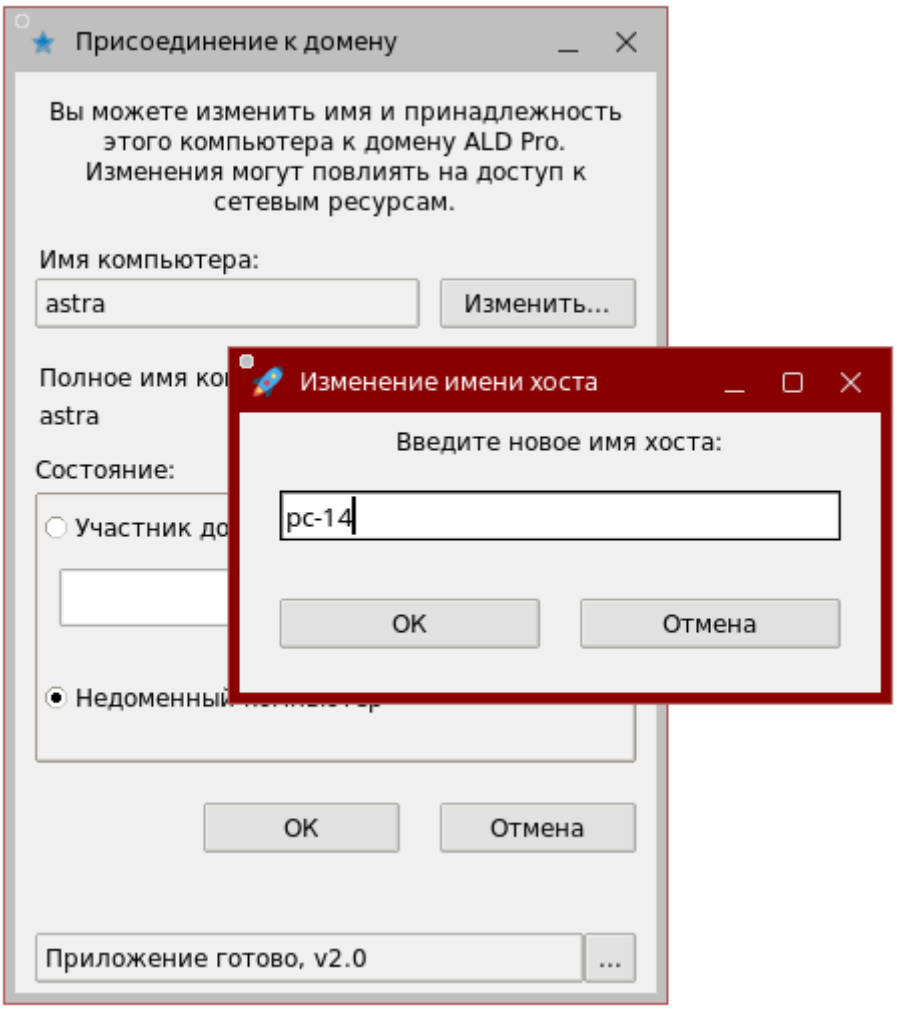


рис. 2.45 – Изменение имени компьютера

4. Установите переключатель в положение «**Участник домена ALD Pro**» и введите имя домена, например, **ald.company.lan**. Нажмите кнопку *OK*, чтобы начать присоединение компьютера к домену.

5. В окне аутентификации введите учетные данные доменного администратора. Это может быть пользователь admin, но можно создать отдельную учетную запись, привилегий которой будет достаточно только для ввода машин в домен.

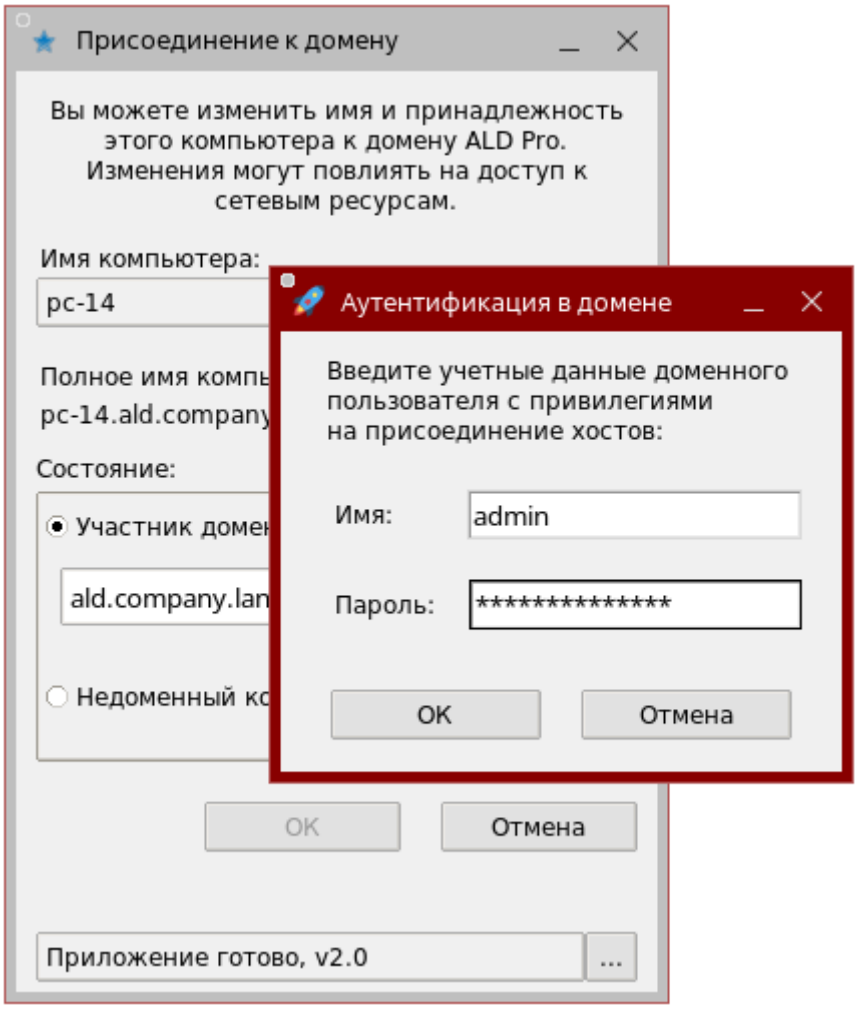


рис. 2.46 – Ввод имени администратора домена для присоединения компьютера

6. Выберите организационное подразделение, в котором требуется создать учетную запись компьютера.

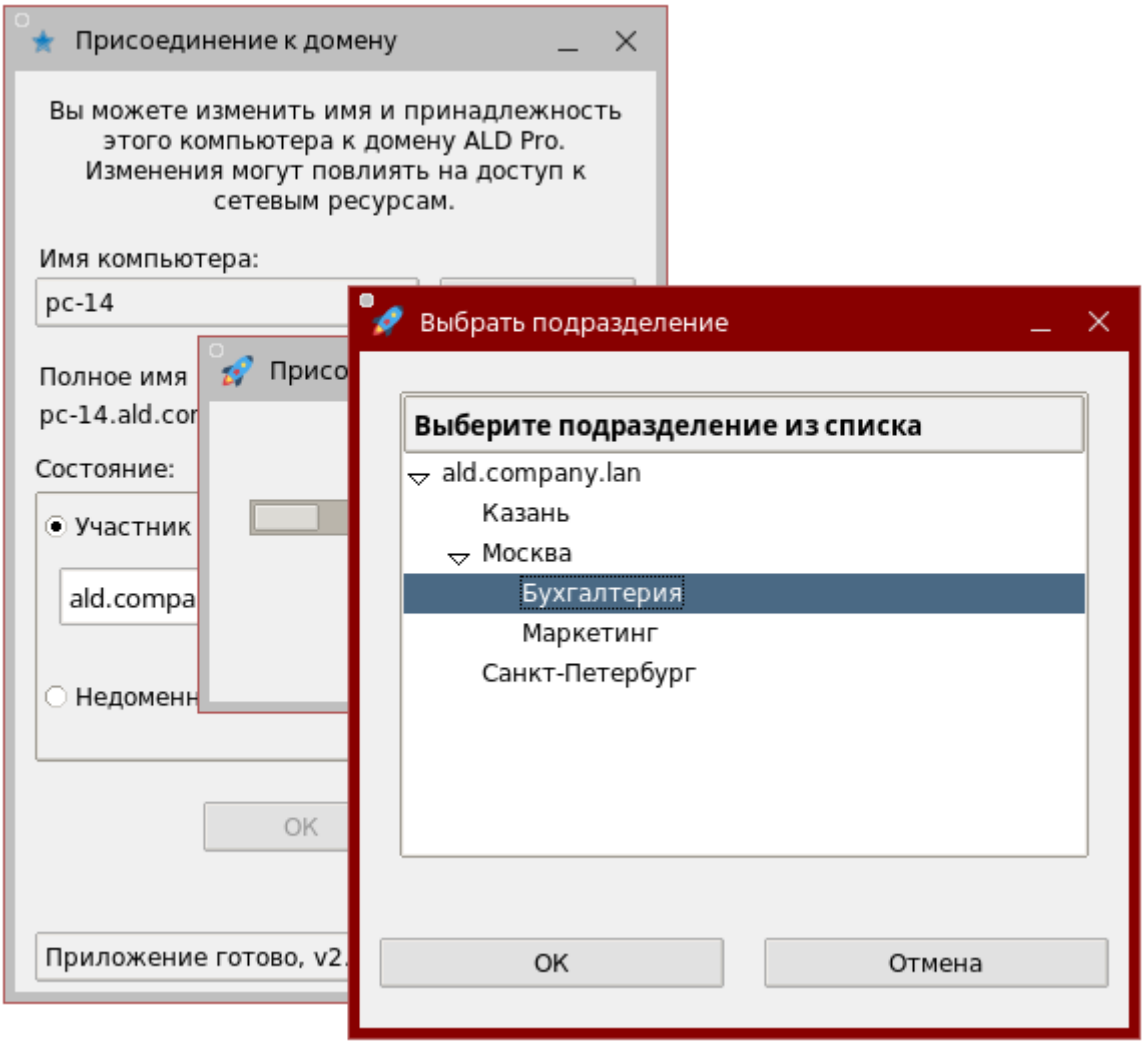


рис. 2.47 – Выбор организационного подразделения

7. Если процедура присоединения компьютера ALSE к домену прошла успешно, вы увидите соответствующее уведомление. Обратите внимание, что утилита сразу применяет групповые политики, политики ПО и задания автоматизации, поэтому процедура может занять некоторое время, наберитесь терпения.

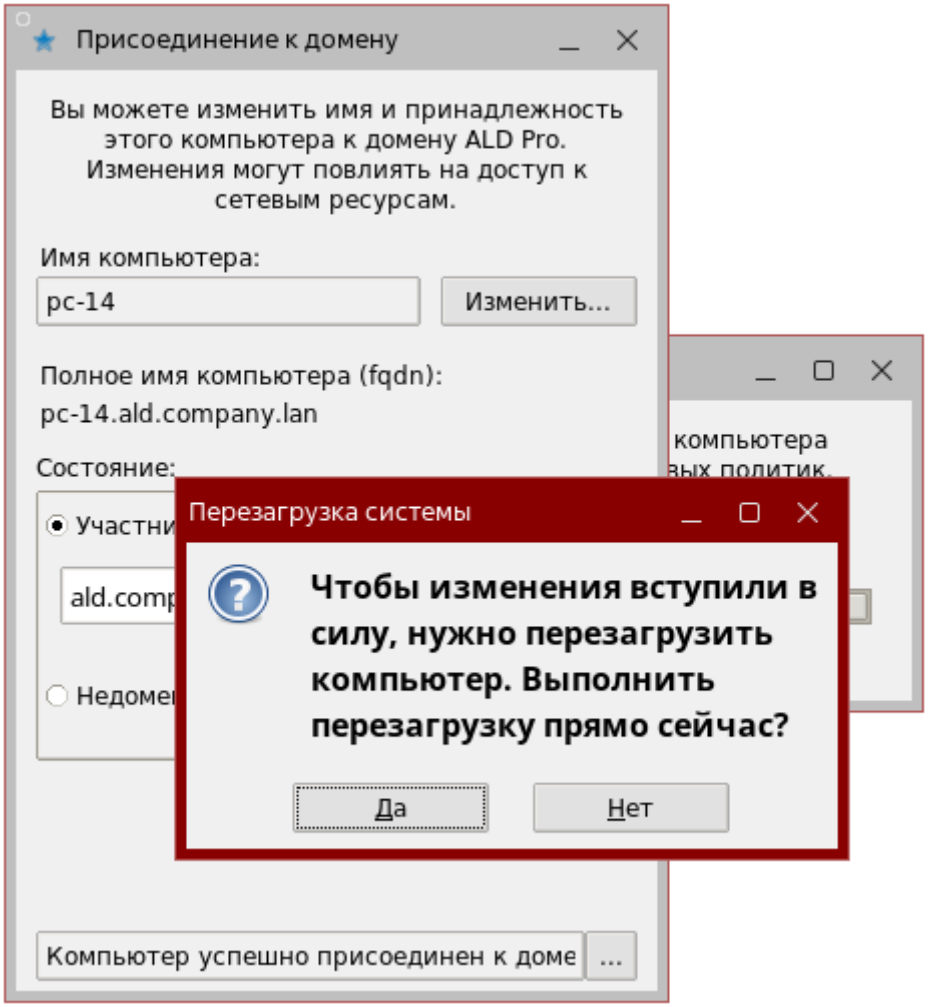


рис. 2.48 – Выполните перезагрузку системы

После перезагрузки компьютера вам станет доступен вход в систему доменным пользователем.

Задание 6. Проверьте доверие с доменом и сбросьте пароль

Чтобы проверить актуальность пароля, нажмите кнопку *Проверить...* напротив имени домена. В поле *Версия ключа и дата выдачи* будет отображать значение в следующем формате: <**Версия пароля**> <**Дата изменения пароля**> <**Время изменения пароля**>.

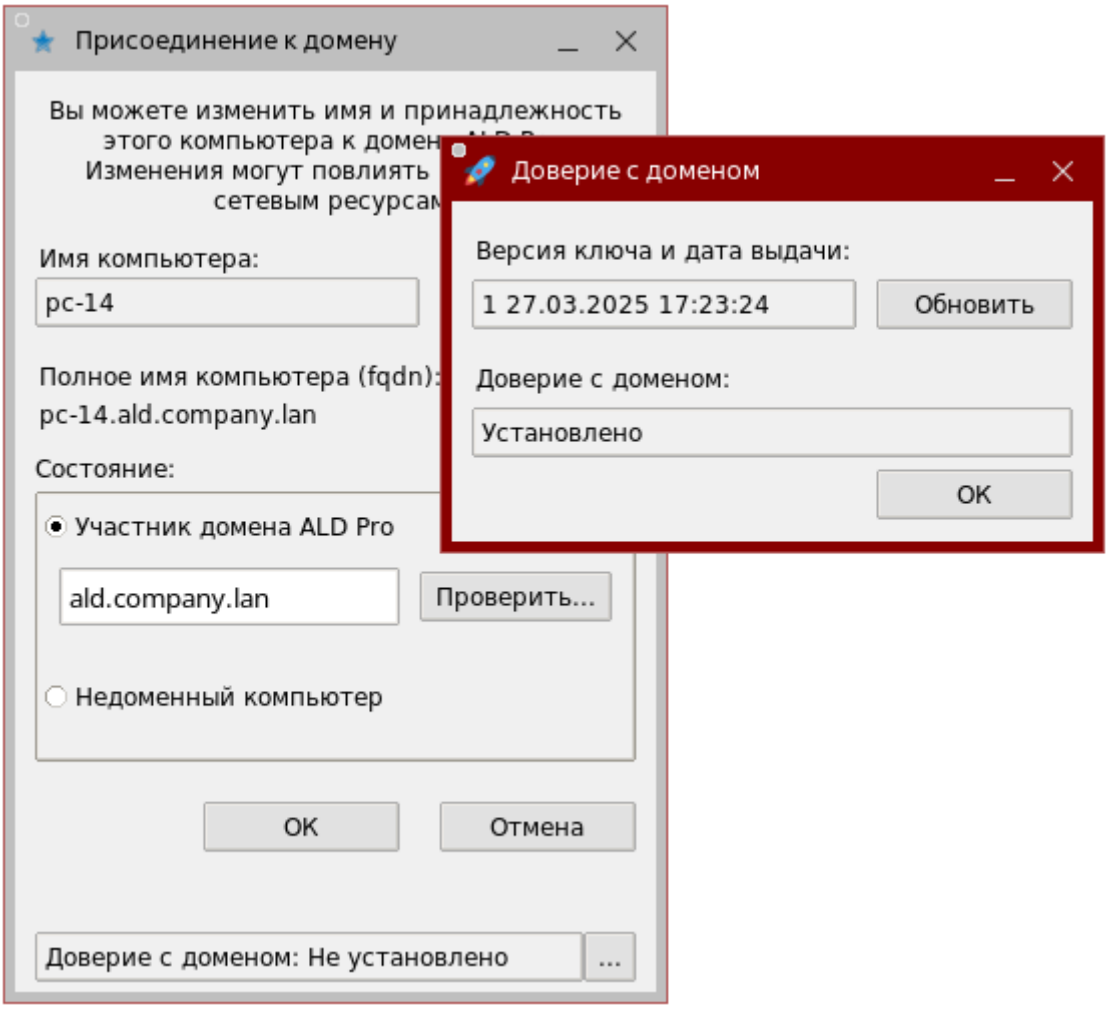


рис. 2.49 – Проверка доверия с доменом

Нажмите кнопку *Обновить*, чтобы установить компьютеру новый пароль и сохранит его в файл `/etc/krb5.keytab`.

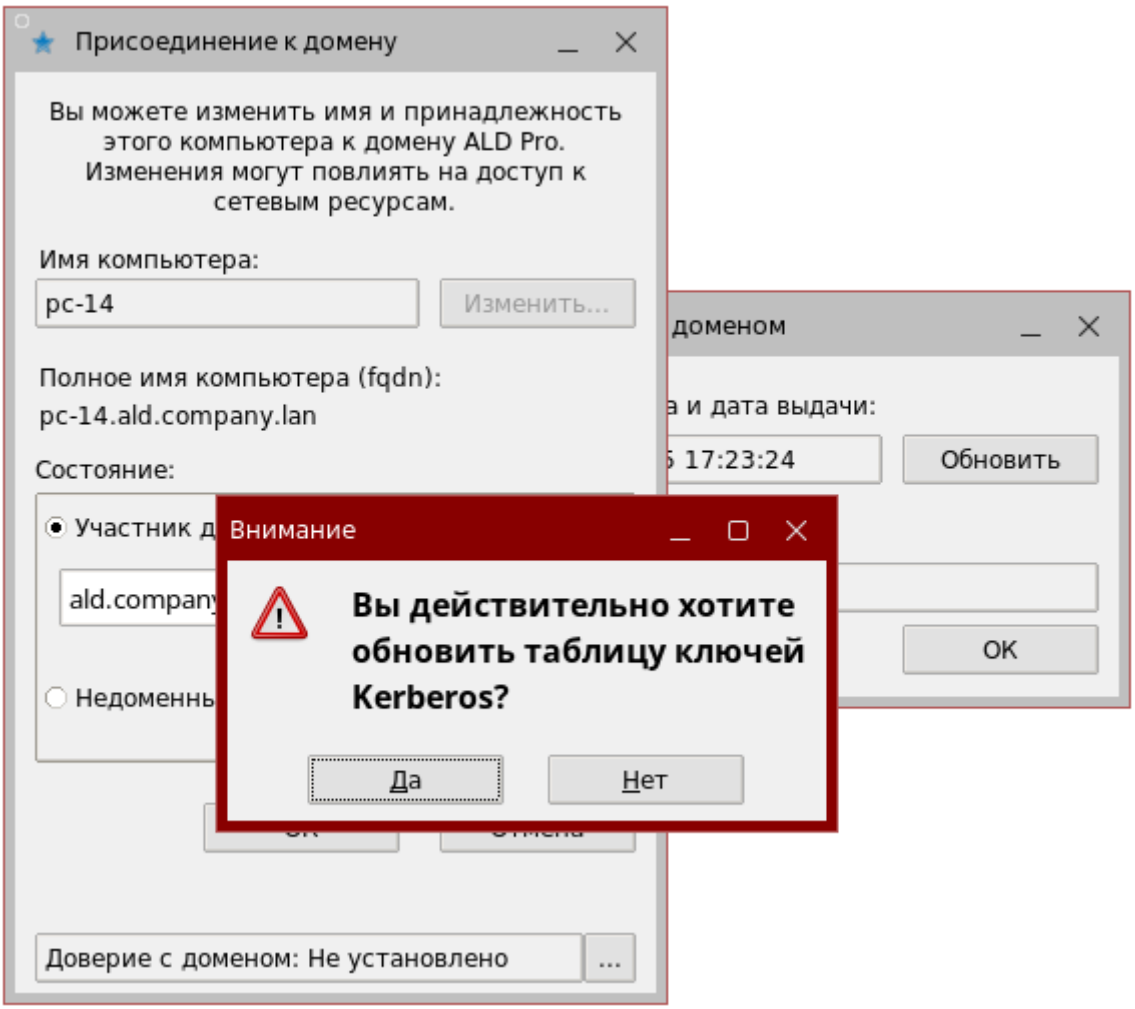


рис. 2.50 – Обновление пароля компьютера

Задание 7. Присоедините компьютер к домену учетной записью с ограниченными правами

Создайте новую роль и добавьте ей все необходимые права доступа:

```
ipa privilege-add 'Host Creation' \
  --desc='Создание новых хостов в домене'

ipa privilege-add-permission 'Host Creation' \
  --permissions='System: Add Hosts'

ipa role-add 'Enrollment Manager' \
  --desc='Участники имеют право создавать хосты и присоединять их к домену'

ipa role-add-privilege 'Enrollment Manager' \
  --privileges='Host Creation' \
  --privileges='Host Enrollment'
```

Создайте пользователя пользователя `Tech Support Engineer` и назначьте ему роль `Enrollment Manager`:

```
ipa user-add tse --first='Tech Support' --last='Engineer' --random

ipa role-add-member 'Enrollment Manager' --users=tse
```

А можно назначить роль и через веб-интерфейс на вкладке *Роли* страницы *Пользователи и компьютеры → Пользователи → Пользователь*:

1. Найдите нужную запись в списке всех ролей с помощью поиска по подстроке «enroll».

2. Кликните по стрелке «влево», чтобы перенести найденную запись в список назначенных (выбранных) ролей.

3. Нажать кнопку *Сохранить*, чтобы зафиксировать изменения.

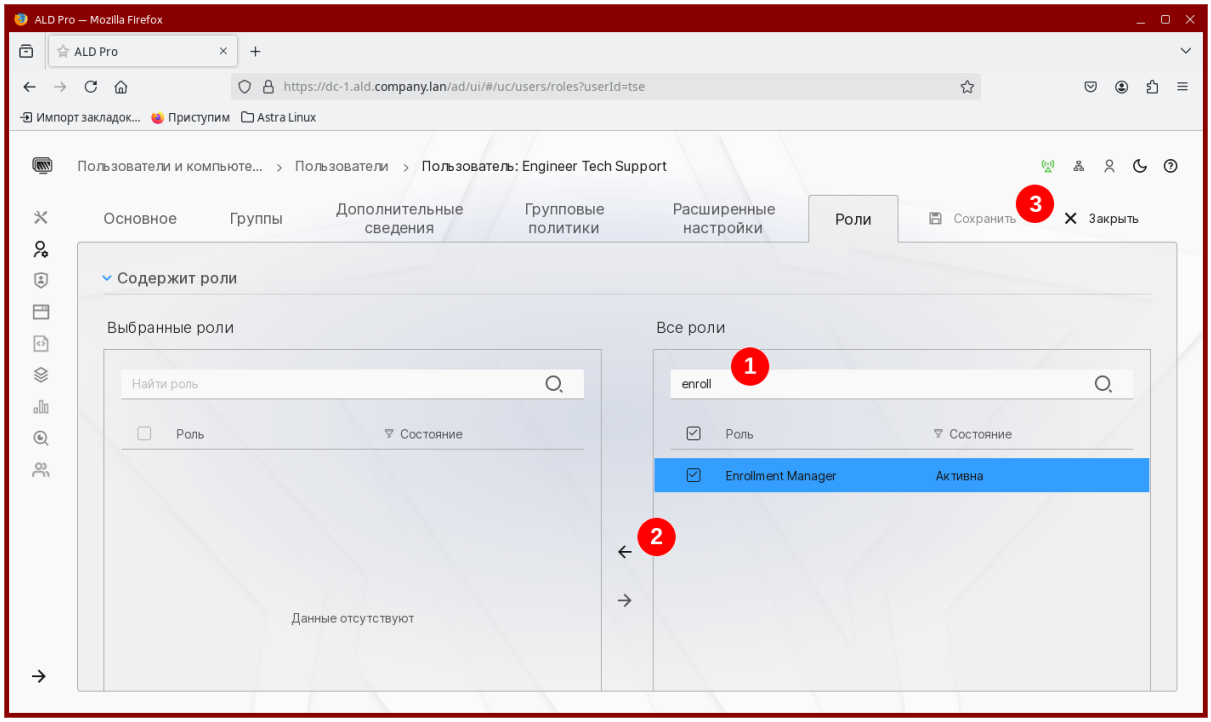


рис. 2.51 – Назначение роли «Enrollment Manager»

Проверьте возможность присоединения компьютера к домену с использованием учетной записи пользователя `tse`.

Задание 8. Присоедините компьютер к домену с помощью одноразового пароля

Создайте учетную запись хоста с помощью команды `ipa host-add`, используя ключ `--random` для генерации одноразового пароля:

```
ipa host-add pc-2.ald.company.lan --random --ip-address=10.0.1.52 \
  --force --department='ou=Московский офис,ou=ald.company.lan,cn=orgunits,cn=accounts,
↪dc=ald,dc=company,dc=lan'
```

В результате выполнения команды будет создана учетная запись компьютера, которой будет назначен одноразовый пароль для присоединения к домену:

```
admin@dc-1:~$ ipa host-add pc-2.ald.company.lan --random --ip-address=10.0.1.52 \
>       --force --department='ou=Московский офис,ou=ald.company.lan,cn=orgunits,
↪cn=accounts,dc=ald,dc=company,dc=lan'
----------------------------------
Добавлен узел "pc-2.ald.company.lan"
----------------------------------
   Имя узла: pc-2.ald.company.lan
   Случайный пароль: 2LyXwGJItweZbvJP3LLqFCT
   Link to department: ou=Московский
         офис,ou=ald.company.lan,cn=orgunits,cn=accounts,dc=ald,dc=company,dc=lan
   Link to head department: Московский офис
   Пароль: True
   Таблица ключей: False
   Managed by: pc-2.ald.company.lan
```

Присоедините компьютер `pc-2` к домену, используя одноразовый пароль самой машины:

```
sudo /opt/rbta/aldpro/client/bin/aldpro-client-installer --host pc-2 \
  --domain ald.company.lan --pc --password '2LyXwGJItweZbvJP3LLqFCT' \
  --gui --force
```

2.5.3 Тест: Модуль 2. Развертывание ALD Pro

Пожалуйста, ответьте на вопросы, чтобы проверить свои знания. Ответы можно найти в конце списка. Учтите, что у некоторых вопросов есть несколько правильных ответов или их можно менять местами.

2.5.3.1 Вопросы

Вопрос 1. На каком уровне защищенности работают контроллеры домена ALD Pro?

a. Базовый «Орел»

b. Усиленный «Воронеж»

c. Максимальный «Смоленск»

d. Не имеет значения

Вопрос 2. На каком уровне защищенности могут быть клиенты домена ALD Pro?

a. Базовый «Орел»

b. Усиленный «Воронеж»

c. Максимальный «Смоленск»

d. Не имеет значения

Вопрос 3. Какую ветку репозитория ALSE необходимо использовать на компьютерах ALD Pro?

a. stable

b. frozen

c. latest

d. fixed

Вопрос 4. Установка какого приложения запустит установку всех пакетов, необходимых для работы контроллера домена ALD Pro без учета глобального каталога и модуля синхронизации?

a. aldpro

b. aldpro-mp

c. aldpro-dc

d. aldprodc

Вопрос 5. Установка какого приложения запустит установку всех пакетов, необходимых для ввода машины в домен ALD Pro?

a. aldpro

b. aldpro-minion

c. aldpro-client

d. aldproclient

Вопрос 6. В каком случае вы сможете ввести компьютер в домен ALD Pro?

a. Есть учетная запись администратора домена и привилегии суперпользователя

b. Есть учетная запись пользователя, которому делегированы права на ввод компьютеров в домен, и привилегии суперпользователя

c. Есть одноразовый пароль компьютера и привилегии суперпользователя

d. Есть учетная завись любого доменного пользователя и привилегии суперпользователя

e. Есть привилегии суперпользователя

Вопрос 7. Выберите правильную команду для продвижения контроллера домена ALD Pro.

a. Команда `sudo aldpro-server-promotion <параметры>`

b. Команда `sudo aldpro-dc-install <параметры>`

c. Команда `sudo aldpro-dc-promotion <параметры>`

d. Команда `sudo aldpro-server-install <параметры>`

Вопрос 8. Какую настройку рекомендуется выполнить после продвижения первого контроллера домена?

a. Отключить DNSSEC

b. Разрешить рекурсивные запросы и кэширование

c. Настроить глобальные перенаправители DNS-запросов

d. Все вышеперечисленное

Вопрос 9: Какие инструменты используются для управления доменом ALD Pro?

a. Портал управления

b. Командная строка

c. Поисковик Yandex

d. Запросы к LDAP-каталогу

e. Чат-бот телеграм

Вопрос 10: Где нужно внести изменения, чтобы вывести компьютер из домена ALD Pro?

a. Изменения нужно внести только на рабочей станции

b. Изменения нужно внести только в LDAP-каталоге домена

c. Изменения нужно внести и на рабочей станции и в LDAP-каталоге домена

d. Компьютер выйдет из домена автоматически, если его не включать более 30 дней

Ответы на вопросы

1. c
2. d
3. b

4. b
5. c
6. a, b, c
7. d
8. d
9. a, b, d
10. c

2.6 Заключение

В этом модуле мы с вами установили первый контроллер и научились вводить рабочую станцию в домен. Далее мы рассмотрим работу служб разрешения имен и синхронизации времени, научимся ими управлять.

2.7 Дополнительные источники информации

- Astra Linux 1.7.6.11-26.08.24_17.26.iso
- Рекомендуем ознакомиться также с видеороликами нашего хорошего товарища на канале «ИТ-Проповедник». Учитывайте, что ролики записаны для версии 2.2.1, поэтому часть информации потеряла актуальность, но в целом они могут быть крайне полезны.
  - ALD PRO 2.2.1 - развертывание первого контроллера домена
  - Сервер разрешения имен BIND 9 в ALD Pro 2.2.1
  - ALD PRO 2.2.1 - ввод клиента в домен

2.8 Обратная связь

Если остались вопросы, то их всегда можно задать в специальной теме.

Модуль 3. Базовые функции и настройка ALD Pro

3.1 Введение

В этом модуле мы будем ближе знакомиться с порталом управления и разбираться, как в домене работают базовые сетевые сервисы NTP и DNS. Если вы ранее администрировали домен Active Directory, то эти технологии и концепции должны быть вам вполне знакомы и нужно будет только понять принципиальные отличия.

Хотелось бы уже захватить работу групповых политик, но ввиду важности этой темы мы все-таки лучше посвятим ей отдельный модуль. Как говорится, лучше меньше да лучше, а слишком хорошо – тоже нехорошо.

3.2 Портал управления ALD Pro

3.2.1 Архитектура портала управления

Основным интерфейсом администратора в ALD Pro является **Портал управления**, который покрывает функции таких оснасток Windows, как Server Manager, Users and Computers, Group Policy Management Editor, Domains and Trusts и многих других.

Портал управления представляет собой веб-приложение, разработанное на языке Python с использованием фреймворка Django. Запросы пользователей принимает веб-сервер Apache и передает Python-приложению через WSGI (*англ. Web Server Gateway Interface*), а клиентская часть написана на React и взаимодействует с сервером через JSON API.

Программные интерфейсы бэкенда могут быть использованы сторонними разработчиками для интеграции своих собственных приложений со службой каталога. Например, с помощью API был написан *скрипт policy.py для импорта дополнительных параметров групповых политик*. А еще в плейбуках продукта Astra Automation через обращение к API выполняется установка дополнительных контроллеров домена.

В соответствии с диаграммой развертывания портал управления устанавливается на каждом контроллере домена ALD Pro, и вы можете обращаться к любому из этих экземпляров, так как все они работают с данными из общего LDAP-каталога, см. рис. 2.52. Любые изменения в части пользователей, групп, настроек домена будут автоматически реплицированы на все контроллеры в домене средствами LDAP-каталога 389 Directory Server.

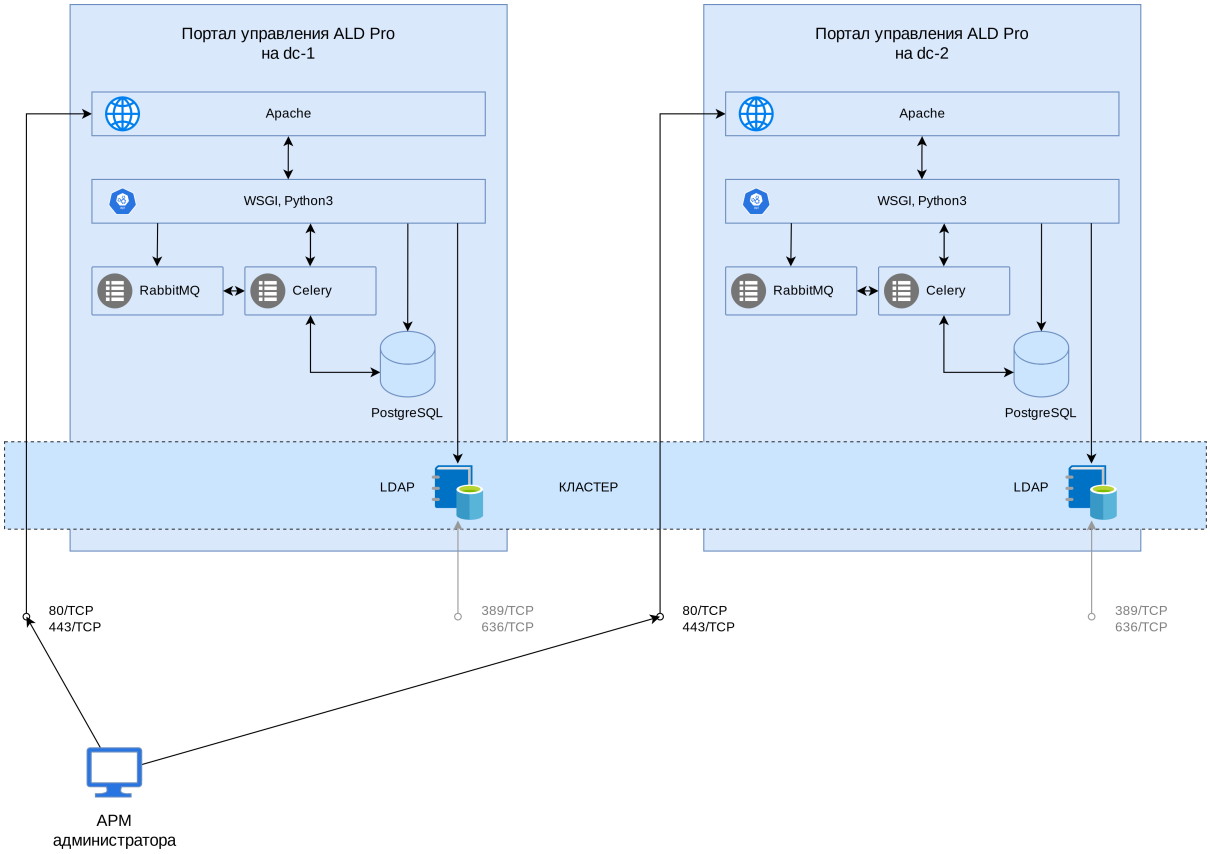


рис. 2.52 – Архитектура портала управления

Можно сказать, что контроллеры домена образуют кластер, который обеспечивает согласованность данных **в конечном итоге**. Это означает, что создание одноименных объектов и редактирование одних и тех же атрибутов на нескольких контроллерах домена при отсутствии связи между ними приведет к созданию конфликтующих изменений, но, когда связь между контроллерами восстановится, эти конфликты будут устранены автоматически в ходе репликации.

Вместе с тем, у каждого экземпляра веб-портала есть и собственная СУБД реляционного типа, в роли которой выступает PostgreSQL, но с версии ALD Pro 2.4.x в ней хранятся только вспомогательные данные, поэтому любой портал управления покажет вам актуальную информацию, которая уже стала доступна контроллеру через механизм репликации.

Начиная с версии ALD Pro 2.4.x, продукт не использует больше компонент Salt Master на контроллере домена и установка дополнительных подсистем выполняется автономной службой `aldpro-salt-minion`, которая получает всю необходимую информацию через LDAP.

При назначении доменному компьютеру какой-то из ролей ALD Pro, бэкенд портала управления вносит соответствующие изменения в его учетную запись. Для этого учетным записям серверов назначается класс `rbta-subsystem`, который объединяет целый ряд атрибутов с префиксом `rbtaSubsystem*` (`rbtaSubsystemConfig`, `rbtaSubsystemLog`, `rbtaSubsystemMetainfo`, `rbtaSubsystemRole`, `rbtaSubsystemSite`, `rbtaSubsystemState` и `rbtaSubsystemVersionAld`).

Через некоторое время после внесения изменения на портале управления служба aldpro-salt-minion, работающая на целевом хосте, обратится в LDAP-каталог и проверит значения этих атрибутов, чтобы привести сервер к указанному состоянию. Таким образом выполняется установка, обновление или удаление подсистемы, более детально об этом процессе, см. *Модуль 7. Дополнительные подсистемы ALD Pro*.

3.2.2 Вход в портал управления

Портал управления поддерживает два способа входа: по логину/паролю и через прозрачную Kerberos-аутентификацию.

3.2.2.1 Аутентификация по паролю

Портал управления реализует собственную веб-форму для входа по логину/паролю, которая передает данные на сервер через POST-запрос в открытом виде, поэтому при использовании такого способа аутентификации нужно обязательно защищать соединение шифрованием SSL, см. рис. 2.53.

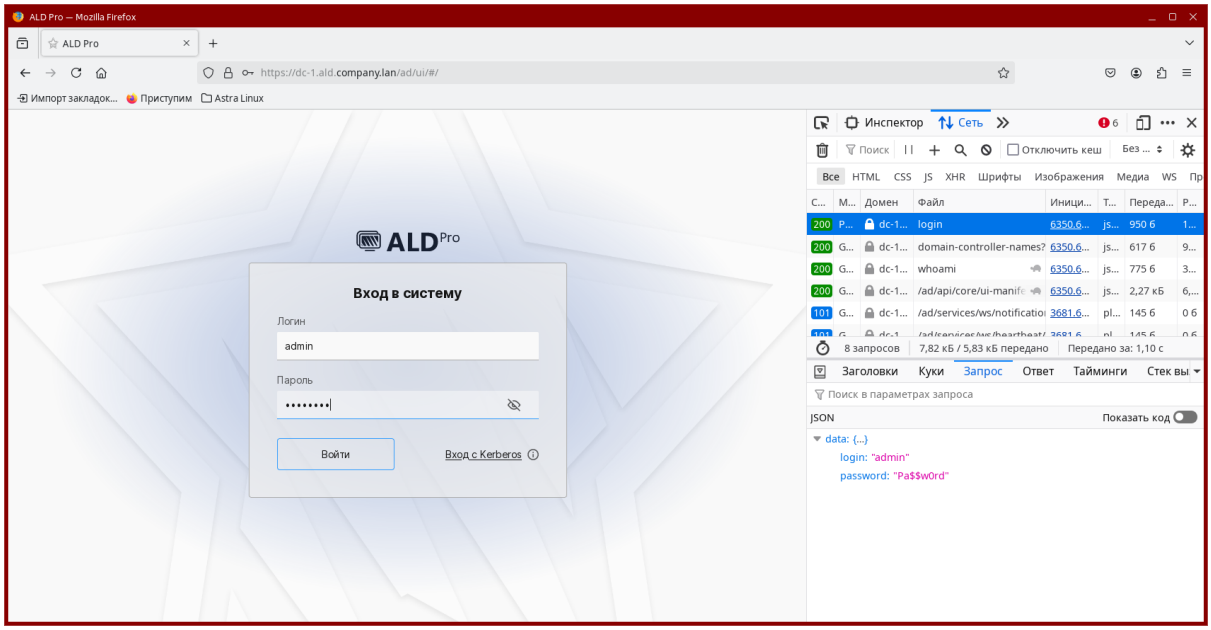


рис. 2.53 – Данные при аутентификации по паролю передаются POST-запросом в открытом виде, поэтому требуется защищенное HTTPS-соединение

Шифрование нужно еще и по той причине, что после аутентификации в каждом запросе к серверу будет передаваться CSRF-токен, который является временным аутентификатором пользователя, поэтому для обеспечения безопасности портал ALD Pro запрещает работу по протоколу HTTP и автоматически перенаправляет пользователей на протокол HTTPS, защищенный шифрованием.

3.2.2.2 Прозрачная Kerberos-аутентификация

Если вы откроете портал управления с доменного компьютера, то у вас будет возможность пройти прозрачную аутентификацию по протоколу Kerberos V5 без пароля. Поток данных этой аутентификации отражен на рис. 2.54:

1. Пользователь нажимает на кнопку *Вход с Kerberos*, которая расположена в правом нижнем углу формы аутентификации.

2. Браузер клиента отправляет POST-запрос на адрес https://dc-1.ald.company.lan/ad/api/ds/login/kerberos, для которого в настройках веб-сервера включена Kerberos-аутентификация.

3. Веб-сервер видит, что для запрашиваемой страницы требуется Kerberos-аутентификация, поэтому возвращает ответ со статусом 401 Unauthorized и заголовком «WWW-Authorization: Negotiate», который указывает на необходимость аутентификации с использованием механизма SPNEGO (Simple and Protected GSSAPI Negotiation Mechanism) через NTLM или Kerberos.

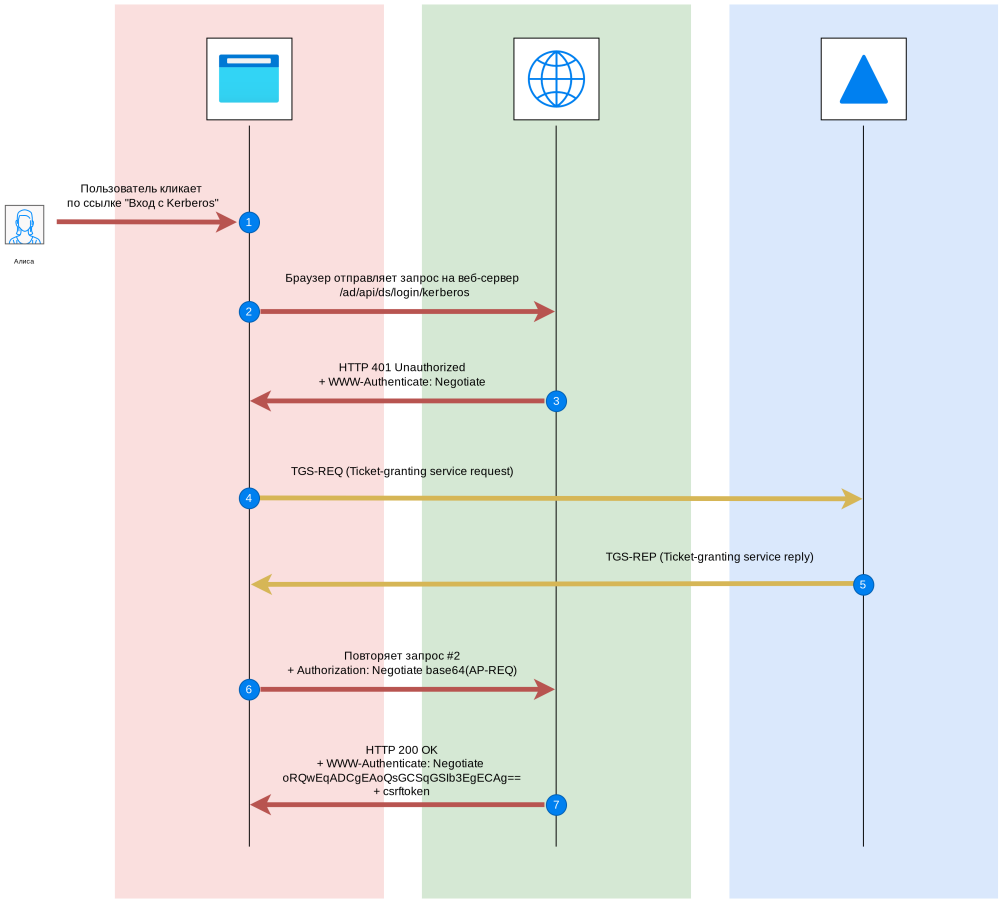


рис. 2.54 – Аутентификация на портале через SPNEGO по Kerberos V5

4. Браузер проверяет, что на веб-сайте с адресом https://dc-1.ald.company.lan можно выполнять аутентификацию по механизму SPNEGO, и запрашивает через GSSAPI билет для аутентификации в сервисе HTTP/dc-1.ald.company.lan. Если в связке ключей нет требуемого TGS-

билета, но есть TGT-билет, то GSSAPI сделает запрос TGS-REQ на контроллер домена для получения запрашиваемого билета.

5. Контроллер домена, получив валидный запрос TGS-REQ, возвращает ответ TGS-REP, внутри которого находится TGS-билет и сессионный ключ из этого билета для аутентификации в сервисе.

6. Браузер, получив TGS-билет от GSSAPI, формирует запрос AP-REQ, кодирует его в base64 и передает веб-серверу в заголовке «WWW-Authorization: Negotiate base64(AP-REQ)». Адрес запроса остается прежний, как на втором шаге `https://dc-1.ald.company.lan/ad/api/ds/login/kerberos`.

7. Веб-сервер извлекает AP-REQ из заголовка и передает его службе GSS-Proxy, которая имеет доступ к keytab-файлу службы HTTP/dc-1.ald.company.lan и может проверить валидность запроса. Служба GSS-Proxy подтверждает веб-серверу, что запрос валиден, тогда веб-сервер генерирует новый CSRF-токен и передает его в ответе с успешным статусом 200 OK.

   Служба GSS-Proxy создает дополнительный уровень защиты на контроллере домена, а для более простых приложений веб-серверу Apache можно дать прямой доступ к keytab-файлу, чтобы он проверял аутентификацию самостоятельно.

Обратите внимание, что в инспекторе браузера на вкладке *Сеть* шаги #3 и #6 отображаться не будут. На запрос #2 мы сразу получим ответ #7, так как запросы SPNEGO обрабатываются на уровне GSSAPI и не доходят до веб-страницы.

Как было сказано ранее, на шаге #4 браузер должен был проверить, что сайту разрешено запрашивать аутентификацию по механизму SPNEGO. На контроллерах домена штатным браузером является Firefox, и эта настройка задается в файле `/usr/lib/firefox/distribution/policies.json` следующим образом:

```
cat /usr/lib/firefox/distribution/policies.json
{
   "policies": {
      "BlockAboutAddons": true,
      "BlockAboutConfig": true,
      "Authentication": {
         "SPNEGO": ["ald.company.lan"]
      },
      "Certificates": {
         "ImportEnterpriseRoots": true,
         "Install": ["/etc/ipa/ca.crt"]
      },
      "Homepage": {
         "URL": "https://dc-1.ald.company.lan/",
         "Locked": true,
         "StartPage": "homepage-locked"
      }
```

(continues on next page)

(продолжение с предыдущей страницы)

```
    }
}
```

Если вы захотите централизованно управлять содержанием файла policies.json, например, чтобы настроить прозрачную Kerberos-аутентификацию на всех компьютерах в домене, воспользуйтесь дополнительным параметром групповой политики common_firefox_policy из архива, который можно найти в личном кабинете клиента, см. *Импорт дополнительных параметров*.

На том же четвертом шаге программный интерфейс GSSAPI должен был запросить TGS-билет, но сделать он это сможет только в том случае, если в связке ключей будет валидный TGT-билет. Проверить связку ключей пользователя можно с помощью команды `klist`. Если билета нет, его можно получить командой `kinit <имя_пользователя>`.

Команда просмотра связки ключей:

```
klist
```

Команда очистки кэша связки ключей:

```
kdestroy
```

Команда получения TGT-билета для пользователя `admin`:

```
kinit admin
```

Продемонстрируем полную очистку кэша и повторную аутентификацию, в результате выполнения команд видим новый krbtgt-билет:

```
admin@dc-1:~$ kdestroy
Other credential caches present, use -A to destroy all
admin@dc-1:~$ klist
klist: Credentials cache keyring 'persistent:491200000:krb_ccache_djyB8Wd' not found`
admin@dc-1:~$ kinit admin
Password for admin@ALD.COMPANY.LAN:
admin@dc-1:~$ klist
Ticket cache: KEYRING:persistent:491200000:krb_ccache_djyB8Wd
Default principal: admin@ALD.COMPANY.LAN

Valid starting        Expires              Service principal
02.04.2025 20:13:22   03.04.2025 20:13:18  krbtgt/ALD.COMPANY.LAN@ALD.COMPANY.LAN`
```

При наличии действительного TGT-билета в связке ключей вы можете на портале управления кликнуть по ссылке *Вход с Kerberos*. После успешного входа в связке ключей вы увидите TGS-билет для аутентификации в службе `HTTP/dc-1.ald.company.lan`.

```
admin@dc-1:~$ klist
Ticket cache: KEYRING:persistent:491200000:krb_ccache_djyB8Wd
Default principal: admin@ALD.COMPANY.LAN
```

(continues on next page)

(продолжение с предыдущей страницы)

```
Valid starting       Expires Service      principal
02.04.2025 20:13:29  03.04.2025 20:13:18  HTTP/dc-1.ald.company.lan@ALD.COMPANY.LAN
02.04.2025 20:13:22  03.04.2025 20:13:18  krbtgt/ALD.COMPANY.LAN@ALD.COMPANY.LAN`
```

Как уже было сказано, дальнейшая проверка аутентичности пользователя подтверждается CSRF-токеном, который отправляется на веб-сервер в каждом запросе, поэтому Kerberos-ключи больше не потребуются, и их можно даже удалить.

3.2.3 Интерфейс портала управления

Итак, вы открыли портал управления, см. рис. 2.55.

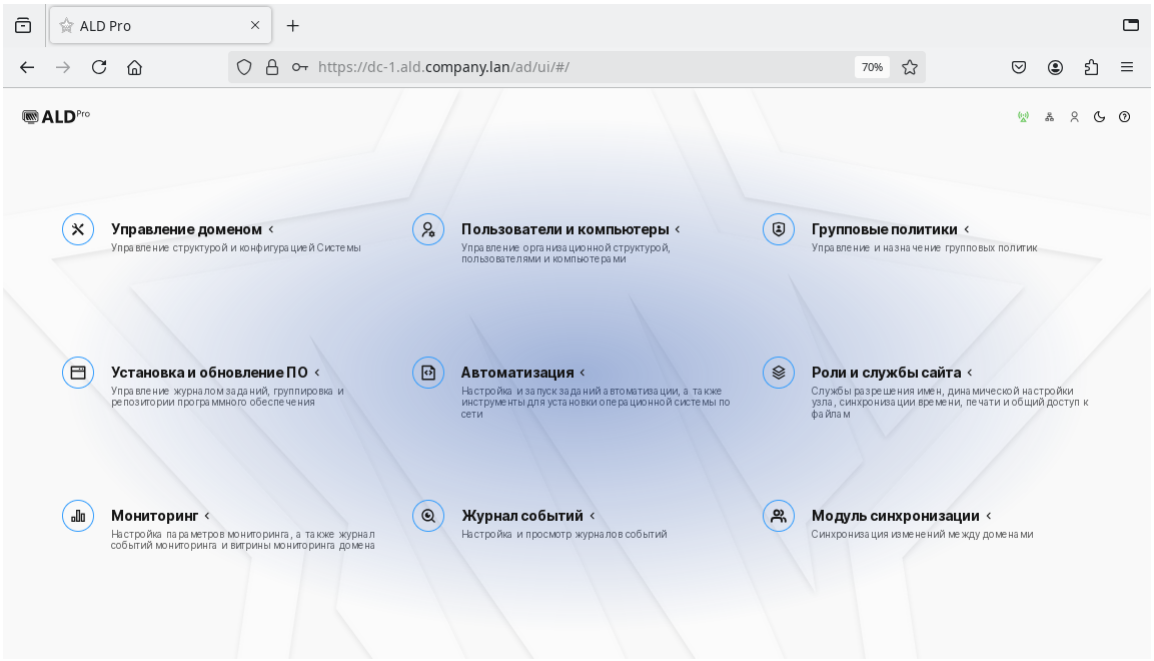


рис. 2.55 – Портал управления ALD Pro

В правом верхнем углу есть пять иконок:



Отображает статус подключения. Информация может отображаться с задержкой, что не влияет на работу системы. В последующих редакциях индикатор будет упразднен.



Позволяет быстро перейти на портал управления любого контроллера домена без необходимости вводить адрес вручную.

Отображает имя текущего пользователя, позволяет перейти к личной карточке сотрудника или выйти из системы.

Позволяет переключить интерфейс портала управления между темной и светлой темами оформления.

Позволяет перейти к справочному центру с рабочей документацией.

В основной области представлено девять плиток, каждая из которых группирует элементы управления по функциональному признаку.

3.2.3.1 Управление доменом

- **Общая информация**
  - **Работоспособность домена** — отображает основные показатели серверной группировки, если установлена и работает подсистема мониторинга.
  - **Состав системы** — отображает список хостов, роль которых была повышена до одной из подсистем. В версии 2.4.0 появилась активная ссылка, которая ведет на карточку подсистемы, см. рис. 2.56.

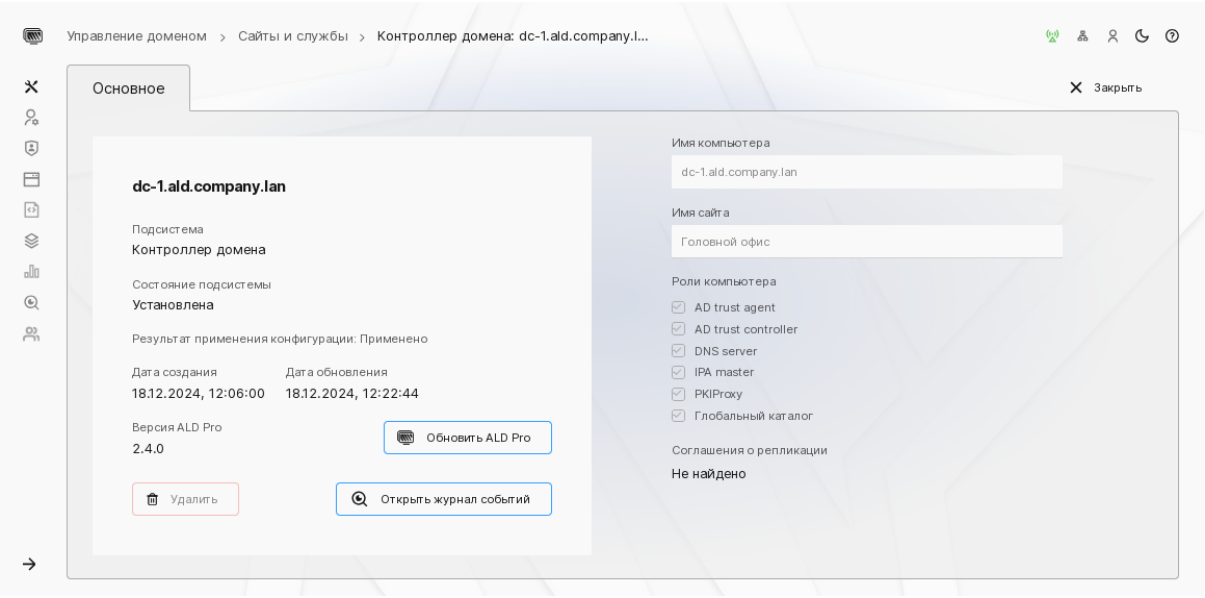


рис. 2.56 – Карточка контроллера dc-1.ald.company.lan

- **Граф топологии** — показывает принадлежность контроллеров домена к сайтам (локациям) и как они связаны соглашениями о репликации. По умолчанию отображаются контроллеры только из сайта «Головной офис». Чтобы изменить критерии отбора, воспользуйтесь кнопкой *Фильтр*.

- **Параметры домена** — тут вы найдете версию ALD Pro, имя Kerberos-области и уровень домена FreeIPA.

- **Сайты и службы**

  - **Сайты** — на этой странице можно создать сайты (локации), с помощью которых настраивается географическая балансировка нагрузки. Механизм работает иначе, чем в Active Directory, поэтому привязывать сайты к сетям не требуется.

    При обращении клиента к SRV-записям `_kerberos._tcp.полное_имя_домена` DNS-сервер из сайта `hq` перенаправит запрос на `_kerberos._tcp.hq._locations.полное_имя_домена` с помощью CNAME-записи, где для серверов из того же сайта `hq` приоритет будет равен нулю, поэтому клиент отдаст им предпочтение.

  - **Контроллеры домена** — на этой странице можно увидеть полный перечень всех контроллеров домена и выполнить повышение роли сервера до контроллера домена.

  - **Соглашения о репликации** — на этой странице отображается список сегментов топологии, которые связывают пары контроллеров соглашениями о репликации. Если открыть карточку сегмента, то можно выполнить повторную инициализацию одного из узлов этой связи.

- **Доп. параметры групповых политик** — раздел позволяет создавать кастомные параметры групповых политик, которые являются аналогом ADMX-шаблонов Microsoft Active Directory. Вы можете определить перечень атрибутов и задать Salt-скрипт, с помощью которого этот параметр будет применяться на рабочих станциях.

- **Каталог заданий автоматизации** — раздел позволяет создавать пользовательские шаблоны заданий автоматизации, которые отличаются от групповых политик только тем, что при назначении на группу компьютеров будут выполнены на каждом хосте только один раз.

- **Службы и параметры Kerberos**

  - **Службы** — на этой странице вы можете создать служебную учетную запись для Kerberos-аутентификации пользователей в вашем приложении. По умолчанию в домене есть учетные записи для DNS, HTTP, CIFS, LDAP, ipa-dnskeysyncd.

    Если в представленном списке нет нужной вам службы или вам потребуется создать учетную запись для сервера, который не предполагается вводить в домен, то вы сможете

создать учетную запись из командной строки.

- **Политика билетов Kerberos** — на этой странице вы можете задать срок жизни Kerberos-билетов (по умолчанию 24 часа или 86400 секунд), и максимальный срок обновления, на протяжении которого клиент сможет продлевать их действие (по умолчанию 7 дней или 604 800 секунд).

  Обратите внимание, что на доменных компьютерах в файле `/etc/sssd.conf` параметры `krb5_lifetime`, `krb5_renewable_lifetime` и `krb5_renew_interval` не заданы, поэтому служба SSSD по умолчанию запрашивает необновляемые билеты.

- **Пользователи и группы**

  - **Параметры пользователей** — на этой странице вы можете определить параметры для новых пользователей. Будьте осторожны с назначением дополнительных объектных классов — если вы добавите объектный класс, который определяет обязательный атрибут, то создание и редактирование пользователей через веб-интерфейс станет невозможным.

  - **Параметры групп** — на этой странице вы можете определить параметры для новых групп. Опять же, будьте осторожны с назначением дополнительных объектных классов.

  - **POSIX идентификаторы** — на этой странице вы можете настроить диапазоны идентификаторов для пользователей и групп. По умолчанию в домене создается диапазон емкостью в 200k идентификаторов, поэтому при превышении этого значения нужно будет создать дополнительный диапазон вручную.

  - **Атрибуты пользователей** — на этой странице вы сможете, не меняя схему каталога, добавить дополнительные атрибуты к карточке пользователя, которые будут доступны на вкладке *Дополнительные сведения*.

    Фактические значения будут храниться в атрибуте *user-custom-attr* в json-формате. В момент создания атрибута вы можете определить тип данных, который определит внешний вид поля на карточке пользователя: логическое значение, дата и время, строка, номер телефона, число.

- **Интеграция с MS AD** — на этой странице вы можете настроить доверительные отношения, миграцию и синхронизацию с доменом MS AD. Обратите внимание, что это разные инструменты, которые могут использоваться вместе и по отдельности в зависимости от сценария гибридного развертывания.

  Модуль миграции похож на инструмент ADMT (Active Directory Migration Tool) и рекомендуется для выполнения разовых миграций небольших организаций, в которых не более 2k сотрудников. Модуль синхронизации обеспечивает поддержание целостности информации в

двух доменах, включая пароли пользователей, и рассчитан на длительное гибридное развертывание инфраструктур любого масштаба.

- **Роли и права доступа** — на этой странице вы можете создавать «метароли» системы ALD Pro, привилегии которых поддерживают привязку к структурным подразделениям. При активации метароли создаются соответствующие привилегии и разрешения системы FreeIPA, которые регулируют права доступа к объектам LDAP-каталога на уровне инструкций доступа ACI (*англ. Access Control Instructions*).

3.2.3.2 Пользователи и компьютеры

- **Организационная структура** — на этой странице вы можете создавать подразделения (*англ. organizational unit, OU*), выстраивая организационно-штатную структуру компании. Структурные подразделения являются доработкой ALD Pro, этой функции нет в службе каталога FreeIPA. Через структурные подразделения назначаются групповые политики и выполняется разделение зон ответственности между системными администраторами.

- **Пользователи, группы пользователей, компьютеры, группы компьютеров** — на этих страницах вы можете создавать объекты каталога, редактировать их атрибуты и выполнять с ними ряд других действий. Эти вкладки выполняют функции, аналогичные тем, что предоставляет оснастка Users and Computers в Active Directory.

- **Корзина** — на этой странице вы найдете учетные записи удаленных пользователей, которые еще доступны для восстановления. Данная функция реализована с использованием возможностей жизненного цикла пользователей FreeIPA, пользователи корзины соответствуют хранимым пользователям (*англ. preserved*).

3.2.3.3 Групповые политики

- **Групповые политики** — в этом разделе вы можете создавать объекты групповых политик ALD Pro, наполнять их параметрами и назначать на структурные подразделения. Механизм наследования и суммирования параметров приближен к Active Directory. На стороне рабочих станций параметры применяются с использованием Salt-скриптов.

- **Политики доступа к узлу** — в этом разделе вы можете настраивать HBAC-правила FreeIPA для ограничения доступа к узлам. Механизм работает на основе PAM.

- **Политики повышения привилегий** — на этой странице вы можете настраивать SUDO-правила FreeIPA для предоставления пользователям прав на выполнение отдельных команд с повышенными привилегиями через утилиту sudo.

- **Политики паролей** — в этом разделе вы можете настраивать политики паролей FreeIPA, которые позволяют определить более строгие требования к паролям для определенных групп пользователей.

3.2.3.4 Установка и обновление ПО

- **Политики ПО** — в этом разделе вы можете создавать объекты групповых политик ALD Pro для установки программного обеспечения, наполнять их параметрами из каталога ПО и назначать их на структурные подразделения.

- **Каталог ПО** — раздел позволяет создавать параметры групповых политик для установки программного обеспечения. Вы можете добавить deb-пакеты для установки, определить перечень атрибутов и задать шаблоны конфигурационных файлов.

  Для полноценного использования политик ПО требуется заранее установить сервер репозиториев и загрузить в него необходимые deb-пакеты, но вы можете использовать эти политики и просто для доставки конфигурационных файлов на рабочие станции без установки deb-пакетов.

- **Репозитории ПО** — в этом разделе можно выполнить повышение роли сервера до подсистемы репозиториев программного обеспечения (Reprepro) и увидеть перечень таких серверов.

- **Политики обновления ALD Pro** — в этом разделе вы можете создавать объекты групповых политик ALD Pro для обновления клиентской части продукта до последней версии из указанных репозиториев. Политики обновления можно назначать на компьютеры или группы компьютеров.

3.2.3.5 Автоматизация

- **Задания автоматизации** — в этом разделе вы можете запускать новые задания автоматизации и просматривать список заданий, запущенных ранее.

  Задания автоматизации могут быть назначены на компьютеры, группы компьютеров или структурные подразделения. Они отличаются от групповых политик только тем, что будут выполнены на каждом хосте только один раз.

  Для возможности использования заданий автоматизации вам требуется сначала настроить пользовательские шаблоны в каталоге заданий автоматизации.

- **Установка ОС по сети** — в этом разделе можно выполнить повышение роли сервера до подсистемы установки ОС по сети (TFTP HPA + PXE) и увидеть перечень таких серверов. Вы сможете создавать профили загрузки и назначать их на устройства с указанием MAC-адреса. Как

только устройство с указанным MAC-адресом обратится к PXE-серверу, ему будет передан индивидуальный конфигурационный файл, содержание которого определяется настройками профиля.

3.2.3.6 Роли и службы сайта

- **Служба разрешения имен** — в этом разделе вы можете настроить работу встроенного DNS-сервера (BIND9). По умолчанию в домене создаются как минимум прямая и обратная зоны, вы можете настроить также перенаправление зон для интеграции с MS AD и разрешение публичных имен через внешний глобальный перенаправитель.

- **Служба динамической настройки узла** — в этом разделе можно выполнить повышение роли сервера до подсистемы динамической настройки узлов (ISC DHCP) и увидеть перечень таких серверов.

  Данная роль была добавлена для поддержки установки ОС по сети, поэтому через веб-портал вы сможете настроить только конфигурационные файлы. Для управления выданными адресами вам потребуется использовать утилиты командной строки.

- **Служба синхронизации времени** — в этом разделе можно настроить внешний источник времени и перечень контроллеров, которые будут выполнять функцию корневых NTP-серверов домена (NTP Chrony). В части синхронизации времени корневые NTP-серверы работают как ePDC, т.е. берут время только от внешнего источника. Настройки синхронизации времени будут доставлены на рабочие станции через групповую политику «Настройки сервиса сетевого времени Chrony».

- **Служба печати** — в этом разделе можно выполнить повышение роли сервера до подсистемы печати (CUPS) и увидеть перечень таких серверов. Вы можете создавать принтеры, загружать драйверы, и т.д.

- **Общий доступ к файлам** — в этом разделе можно выполнить повышение роли сервера до подсистемы общего доступа к файлам (Samba) и увидеть перечень таких серверов. Вы можете создавать общие папки и определять права доступа на уровне SMB-подключения.

- **Служба глобального каталога** — в этом разделе на контроллер домена можно установить модуль глобального каталога (англ. Global Catalog, GC), который является собственной разработкой ALD Pro. Модуль GC необходим для возможности полноценного использования двусторнонних доверительных отношений.

3.2.3.7 Мониторинг

В этом разделе можно выполнить повышение роли сервера до подсистемы мониторинга (Zabbix) и увидеть перечень таких серверов. Вы сможете просматривать журнал событий и витрины мониторинга.

3.2.3.8 Журнал событий

В этом разделе можно выполнить повышение роли сервера до подсистемы аудита (syslog-ng) и увидеть перечень таких серверов. Вы сможете создавать правила сбора событий, но для просмотра журналов вам нужно будет перейти на сервер и воспользоваться, например, графическим приложением Ksystemlog или настроить интеграцию с SIEM.

3.2.3.9 Модуль синхронизации

На этой странице настраивается модуль синхронизации. Данный раздел портала появляется, если был установлен модуль синхронизации. Тут можно настроить контроллеры домена AD, контроллеры домена ALD Pro и параметры синхронизации паролей.

3.2.4 Управление организационной структурой

Перейдем на портал управления в раздел *Пользователи и компьютеры* далее *Организационная структура* и выберем корневое подразделение *ald.company.lan*, как показано на рисунке ниже:

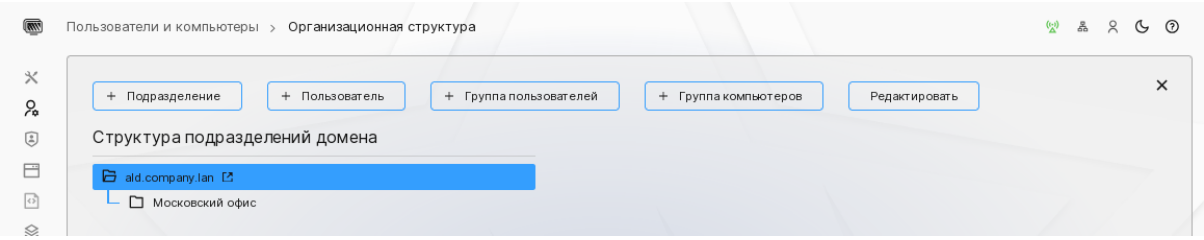


рис. 2.57 – Организационная структура компании

Если выбрано какое-то подразделение, вам будут доступны следующие кнопки:

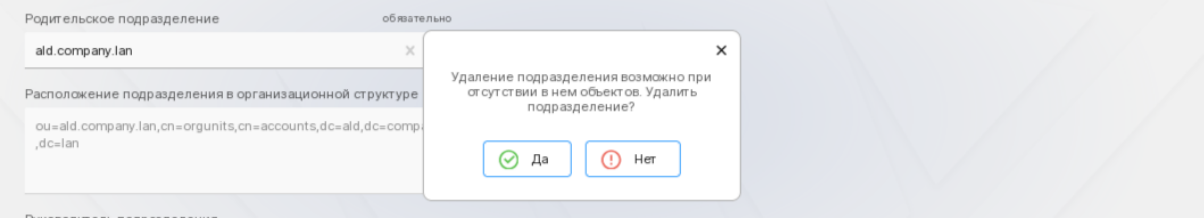
- Кнопка *+ Подразделение* — добавить подразделение.
- Кнопка *+ Пользователь* — создать пользователя в подразделении.
- Кнопка *+ Группа пользователей* — создать группу пользователей в подразделении.
- Кнопка *+ Группа компьютеров* — создать группу компьютеров в подразделении.

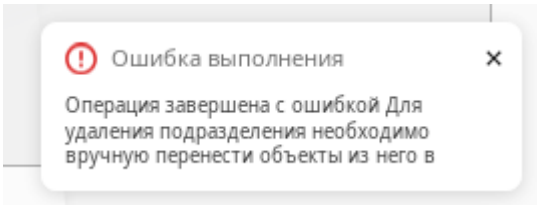
- Кнопка *Редактировать* — перейти к карточке подразделения для ее редактирования.

Мы уже отмечали, что в службе каталога FreeIPA нет подразделений, в этой системе все объекты расположены в плоской структуре и для назначения групповых политик используются обычные группы пользователей и компьютеров. Для возможности назначения групповых политик по схеме LSDOU (*англ. Local, Site, Domain, OU*), т.е. на группы объектов в соответствии с их логическим размещением, в ALD Pro была реализована иерархия организационных подразделений.

Физически учетные записи пользователей, компьютеров и групп остаются в тех же контейнерах FreeIPA, а принадлежность к организационному подразделению определяется через ссылку, которая задается в атрибуте rbtadp (*англ. RusBITech Astra Division Path — путь до организационного подразделения*).

На практике эта разница выражается, например, в том, что интерфейс ALD Pro не даст вам удалить подразделение, к которому привязаны объекты.





Если же вы удалите запись организационного подразделения в обход интерфейса ALD Pro, то будут удалены только дочерние подразделения, а пользователи, компьютеры и группы, принадлежащие этим подразделениям, останутся нетронутыми. Вы сможете найти их в списках соответствующих типов объектов или через командную строку, например, `ipa user-find <подстрока>`.

Настроим простую организационную структуру, как показано на рис. 2.58.

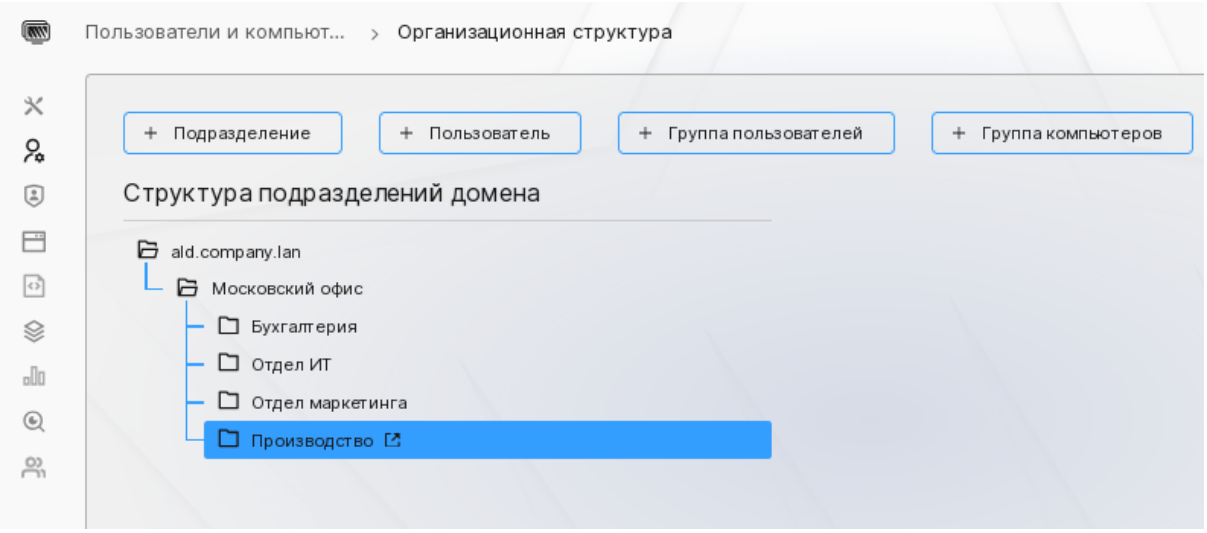


рис. 2.58 – Пример организационной структуры

Для этого с помощью кнопки *+ Подразделение* создайте несколько структурных подразделений и не забудьте сохранить изменения.

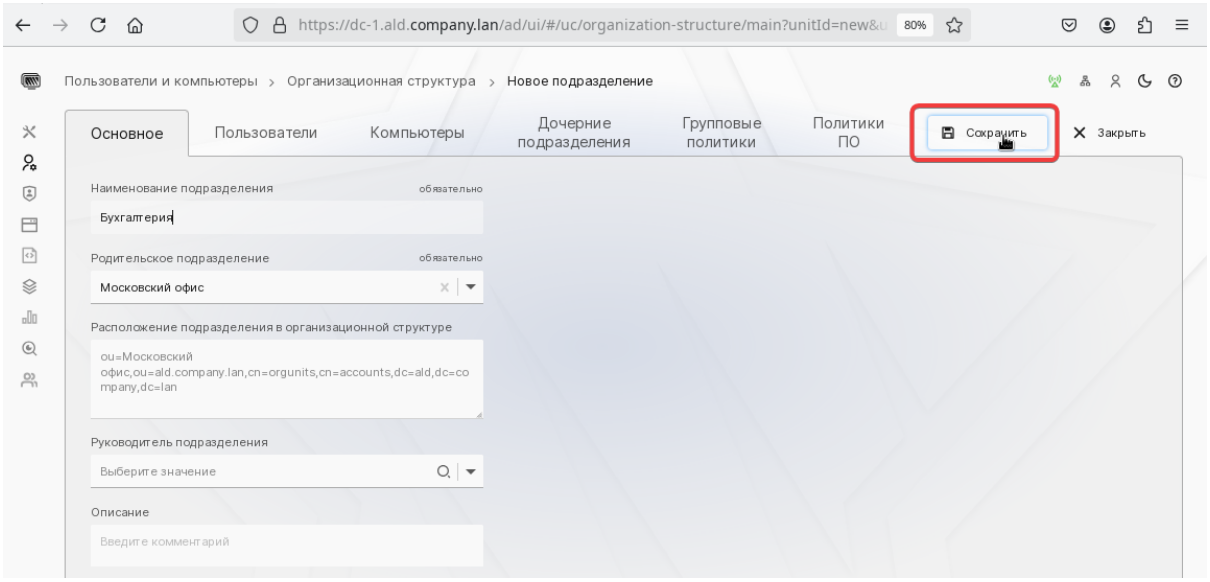


рис. 2.59 – Сохранение нового организационного подразделения

Теперь рассмотрим подробнее карточку организационного подразделения.

- **Основное** — на этой вкладке вы можете изменить основные атрибуты организационного подразделения и применить к нему определенные действия. Кнопка *Сохранить* доступна в правом верхнем углу страницы всегда (см. рис. 2.59), а кнопка *Удалить подразделение* находится в правом нижнем углу на вкладке с основными сведениями.

  - **Описание** — позволяет сделать уточнение, какие объекты каталога следует размещать в этом организационном подразделении.

- **Руководитель подразделения** — позволяет выбрать сотрудника из списка, см. рис. 2.60. Данная информация носит справочный характер и не влияет на какие-либо права доступа. Вы можете использовать ее в заданиях автоматизации для построения отчетов или рассылки уведомлений об истечении срока действия паролей.

- **Расположение** — объединяет группу атрибутов, с помощью которых можно уточнить физическое местоположение организационного подразделения.

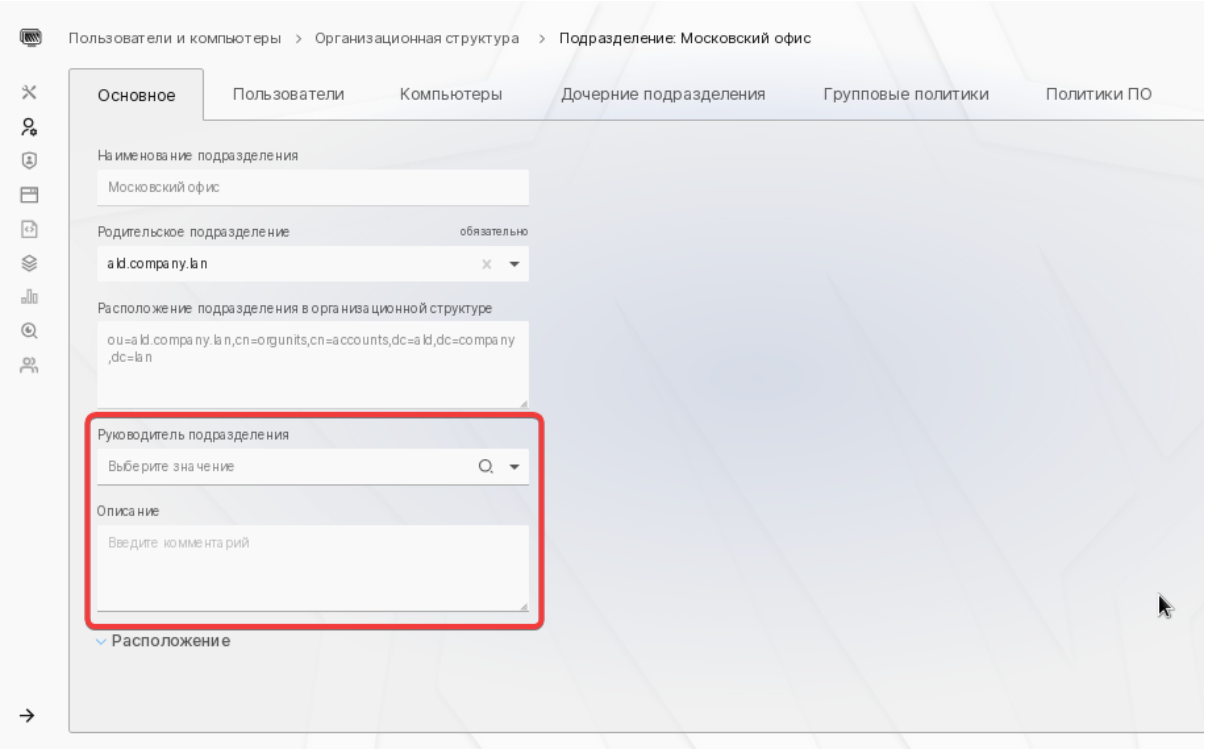


рис. 2.60 – Основные свойства подразделения

- **Пользователи** — на этой вкладке вы можете изменить список пользователей и групп пользователей, принадлежащих этому организационному подразделению.

- **Компьютеры** — аналогично предыдущей вкладке тут вы можете изменить список компьютеров и групп компьютеров, принадлежащих этому организационному подразделению.

- **Дочерние подразделения** — на этой вкладке вы можете посмотреть список существующих дочерних подразделений и создать новые.

- **Групповые политики** — на этой вкладке вы можете создавать новые, просматривать и удалять существующие назначения объектов групповой политики, которые эквивалентны ссылкам GPO из MS AD (*англ. GPO Link*).

- **Политики ПО** — аналогично предыдущей вкладке тут вы можете создавать новые, просматривать и удалять существующие назначения объектов групповой политики по установке программного обеспечения.

3.2.5 Управление пользователями

Перейдем в раздел *Пользователи и компьютеры → Пользователи*, в котором доступен список всех пользователей домена без служебных учетных записей, и откроем карточку администратора.

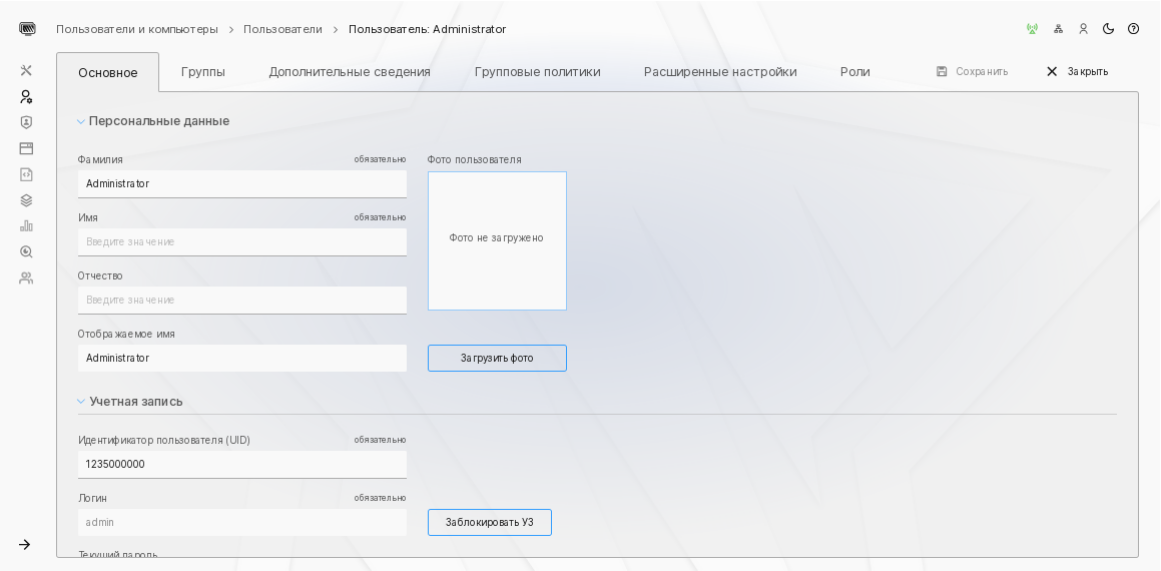


рис. 2.61 – Карточка сотрудника

Вы увидите ряд вкладок, с помощью которых можно редактировать следующую информацию:

- **Основные** — на этой вкладке доступны основные атрибуты пользователя.
  - **Персональные данные** — раздел позволяет задать ФИО, отображаемое имя и фотографию сотрудника.
  - **Учетная запись** — раздел позволяет задать UID учетной записи или заблокировать ее. UID является аналогом SID, и в домене FreeIPA его действительно можно изменить. В этом же разделе вы можете посмотреть срок действия пароля и сбросить его на новое значение.

    Если администратор назначит новый пароль не себе, а любому другому пользователю, то такой пароль будет считаться с истекшим сроком действия, как будто был установлен флажок «Требовать смены пароля при следующем входе в систему» (*англ. User must change password at next logon*).

    Если вам потребуется «снять этот флажок», чтобы пароль был сразу активен, то вам потребуется продлить срок действия пароля вручную из окна терминала, например, командой `ipa user-mod kuznetsov --password-expiration 20250101115110Z`.
  - **Организационные данные** — раздел позволяет указать должность, определить руководителя и задать принадлежность к организационному подразделению.

Информация о руководителе носит справочный характер и не влияет на какие-либо права доступа. Вы можете использовать ее в заданиях автоматизации для построения отчетов или рассылки уведомлений об истечении срока действия паролей.

- **Контактная информация и расположение** — позволяют задать дополнительную справочную информацию о сотруднике.

- **Сертификат пользователя** — позволяет добавить сертификат, который будет использоваться для двухфакторной аутентификации через PKINIT (*англ. Public Key Cryptography for Initial Authentication in Kerberos*) — расширения протокола Kerberos, которое позволяет при аутентификации использовать криптографию с открытым ключом вместо пароля.

  На данный момент реализована интеграция с USB-токенами JaCarta и приложением для входа SecureLogon, которое заменяет стандартное окно, чтобы предоставить дополнительные возможности для удобной работы с токенами.

- **Открытые ключи SSH** — позволяют обеспечить вход на рабочие станции по протоколу SSH с аутентификацией по ключу. Работают через PAM-модуль службы SSSD, которая установлена на всех компьютерах в домене.

- **Группы** — на этой вкладке вы можете посмотреть и изменить информацию об участии пользователя в группах.

  Напоминаем, что в Linux (как и в Windows) локальные группы не поддерживают вложенности, поэтому в состав их участников нельзя включать другие группы. Доменные группы FreeIPA (как MS AD) снимают это ограничение, поэтому на данной странице есть раздел *Состоит в группах непрямое наследование*, в котором указаны те группы, участником которых пользователь является опосредованно через механизм наследования.

- **Дополнительные сведения** — на этой вкладке вы можете просматривать и редактировать значения кастомных атрибутов, список которых можно определить на странице *Управление доменом → Пользователи и группы → Атрибуты пользователей*. Значения этих атрибутов хранятся в атрибуте user-custom-attr в json-формате.

- **Групповые политики** — на этой вкладке вы найдете список параметров групповых политик, которые должны быть применены к данному пользователю. Это чем-то похоже на инструмент RSoP из MS AD, который выдает прогнозируемый результат суммирования параметров в зависимости от настроек групповых политик и расположения пользователя в иерархии организационных подразделений.

- **Расширенные настройки** – на этой вкладке вы можете задать основной и дополнительные адреса SMTP, x500 и SIP для интеграции с почтовыми системами, например, РуПост.

- **Роли** – на этой вкладке вы можете посмотреть и изменить информацию об участии пользователя в ролях, через которые предоставляются права доступа к объектам каталога.

  В службе FreeIPA реализована трехуровневая модель, в которой есть роли, привилегии и разрешения. Разрешения соответствуют инструкциям доступа ACI (*англ. Access Control Instructions*). Подробнее о ролях и их назначении вы можете узнать в справочном центре https://dc-1.ald.company.lan/ad/help/ui/73.

  Система ALD Pro предоставляет несколько десятков преднастроенных ролей, например:

  - **ALDPRO - Main Administrator** — администратор домена обладает полными правами.
  - **ALDPRO - Domain Viewer** — участники роли получают набор привилегий для просмотра всех разделов портала ALD Pro.
  - **ALDPRO - RuPost Service Integrations** — эта роль необходима для того, чтобы предоставить учетной записи почтовой службы Рупост необходимые права доступа.
  - и другие.

  Но под капотом системы есть еще несколько десятков ролей FreeIPA, которые можно назначать пользователям, чтобы они могли воспользоваться соответствующими полномочиями при работе через утилиту ipa или веб-портал FreeIPA, например:

  - **Enrollment Administrator** — администратор, который может вводить компьютеры в домен, если для них уже есть учетная запись в службе каталога.
  - **helpdesk** — сотрудник технической поддержки, который может выполнять простые заявки на обслуживанию, например, сброс паролей.
  - **Security Architect** — архитектор безопасности, который обладает правами на администрирование политик паролей и др.
  - **User Administration** — администратор, который обладает правами для управления группами и учетными записями пользователей.

Создадим несколько пользователей, которые потребуются нам в последующих упражнениях и демонстрациях:

Атрибут	Пользователь1	Пользователь2	Пользователь3	Пользователь4
Логин	iivanov	ppetrov	asidorov	vsidorova
Фамилия	Иванов	Петров	Сидоров	Сидорова
Имя	Иван	Петр	Александр	Виктория
Отчество	Иванович	Петрович	Александрович	Александровна
Подразделение	Отдел ИТ	Отдел маркетинга	Производство	Бухгалтерия
Пароль	Pa$$w0rd	Pa$$w0rd	Pa$$w0rd	Pa$$w0rd

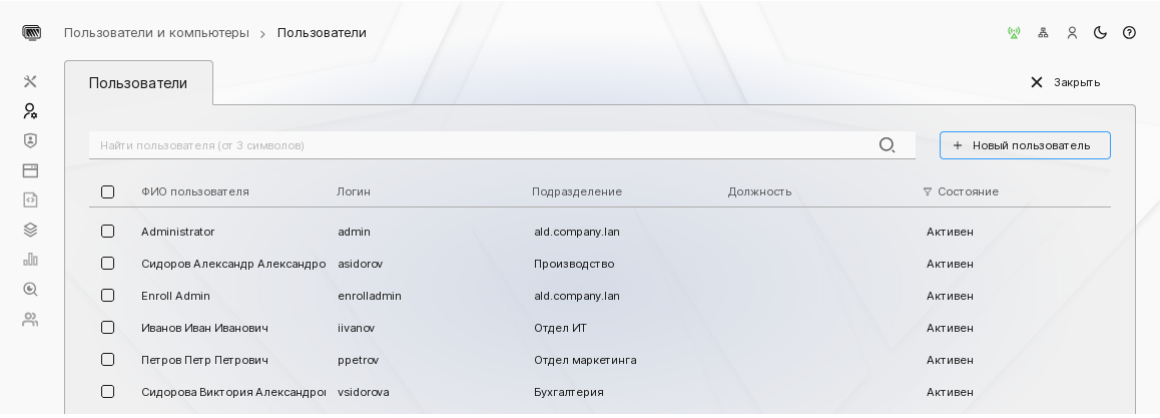


рис. 2.62 – Список сотрудников

3.2.6 Управление компьютерами

Перейдем на страницу *Пользователи и компьютеры → Компьютеры* и откроем карточку объекта с именем pc-1.ald.company.lan.

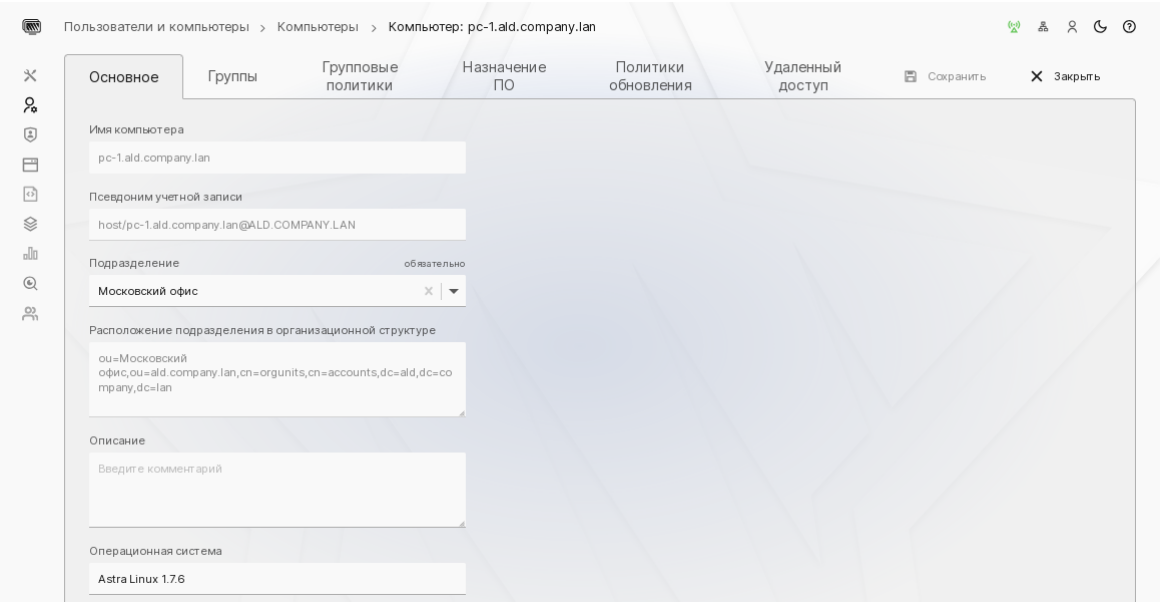


рис. 2.63 – Карточка компьютера pc-1.ald.company.lan

Вы увидите ряд вкладок, с помощью которых можно редактировать следующую информацию:

- **Основные** – в этом разделе представлены атрибуты с основной информацией о компьютере. Вы можете изменить принадлежность компьютера к организационному подразделению, задать дополнительное описание, добавить уточнения по платформе и установленной операционной системе. Поля, за исключением «имя компьютера», «псевдоним учетной записи» и «расположение подразделения в организационной структуре» доступны для редактирования.

- **Группы** — на этой вкладке вы можете посмотреть и изменить информацию об участии компьютера в группах, с помощью которых можно назначать базовые политики FreeIPA, например, политики доступа к узлу (HBAC-правила) и политики повышения привилегий (SUDO-правила).

- **Групповые политики и Назначение ПО** — так же как в карточке пользователя, на этой вкладке вы найдете список параметров групповых политик и политик ПО, которые должны быть применены к данному компьютеру.

- **Удаленный доступ** – интерфейс для подключения к рабочему столу пользователя для оказания технической поддержки. Подробнее в следующем разделе.

3.2.7 Удаленный доступ для оказания поддержки

Портал управления ALD Pro предоставляет инструмент удаленного доступа к рабочему столу по протоколу VNC, который вы можете использовать для оказания помощи сотрудникам. Для того чтобы включить удаленный доступ, пользователь должен запустить приложение *Пуск → ALD Pro → Удаленный рабочий стол* и сообщить администратору временный пароль, см. рис. 2.64.

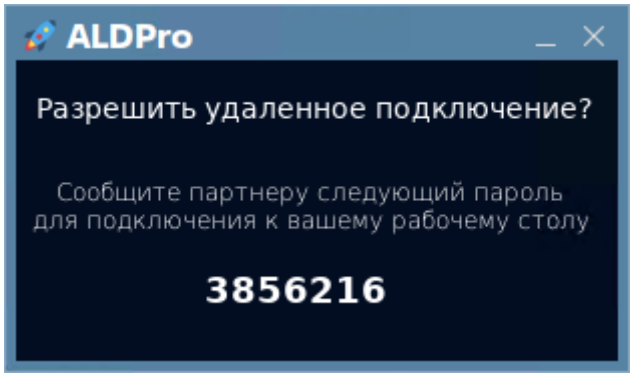


рис. 2.64 – Временный пароль для удаленного подключения

Администратор сможет выбрать активную сессию из списка на вкладке *Удаленный доступ* карточки компьютера и подключиться к рабочему столу пользователя по предоставленному паролю, см. рис. 2.64. Информацию об активных сессиях портал управления получает, обращаясь к рабочей станции по порту 30000/TCP. Удаленный доступ работает по протоколу VNC, подключение

выполняется по портам, начиная с 6080/TCP.

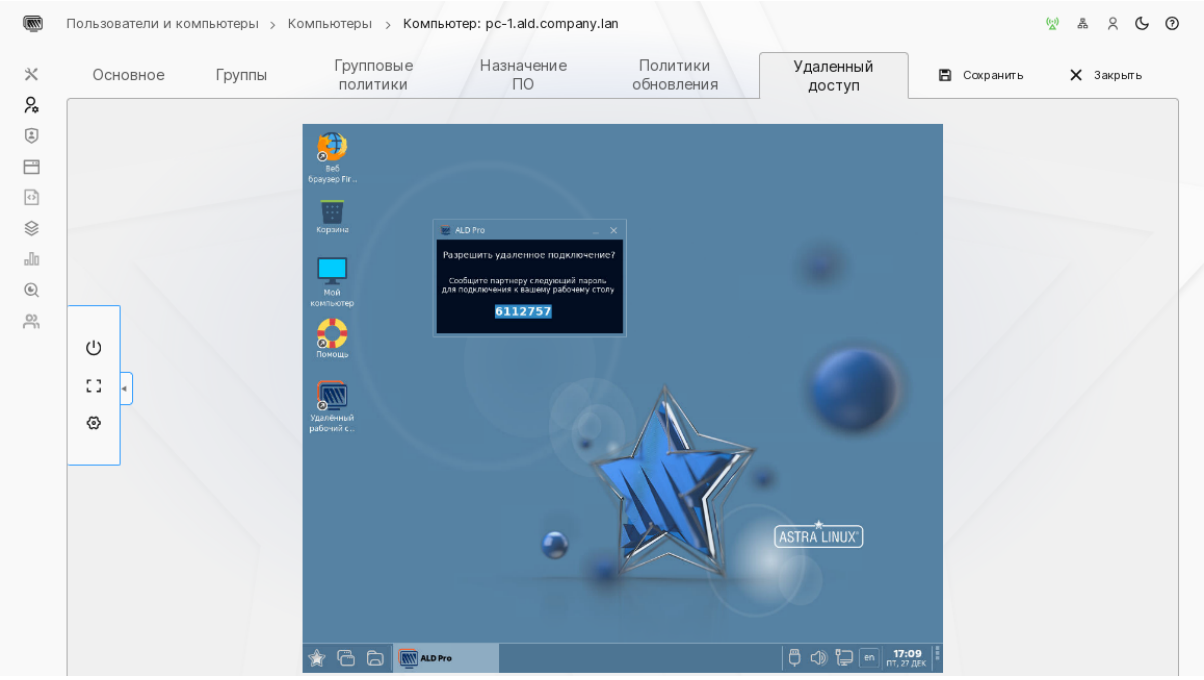


рис. 2.65 – Управление удаленным рабочим столом пользователя с портала управления ALD Pro

Попробуйте сами, как это работает. Включите виртуальную машину с pc-1, войдите в операционную систему под учетной записью пользователя Петр Петров (ppetrov) и запустите приложение для удаленного доступа. После этого перейдите на портал управления и воспользуйтесь этим паролем для удаленного подключения, см. рис. 2.66.

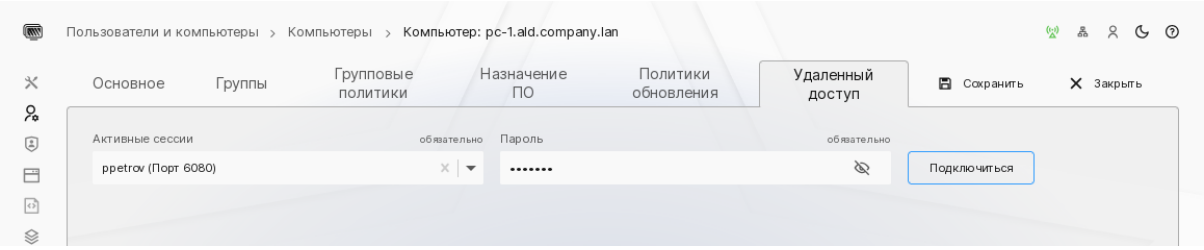


рис. 2.66 – Подключение к удаленному рабочему столу пользователя с портала управления ALD Pro

В левой части окна будет панель с кнопками, которые позволят вам перейти в полноэкранный режим и завершить сессию, см. рис. 2.67.

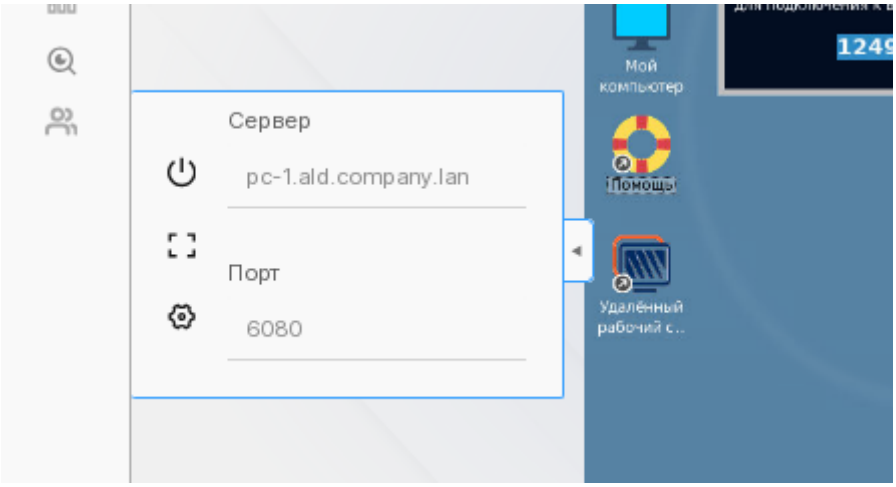


рис. 2.67 – Панель управления сессией удаленного подключения

Если вы по ошибке закроете карточку компьютера, то удалённый доступ не прервётся, и вы сможете вернуться назад, чтобы продолжить оказание технической поддержки. Пользователь может прервать удаленное подключение по своему усмотрению, для этого ему достаточно закрыть окно приложения «Удаленный рабочий стол».

3.2.8 Управление группами

С точки зрения интерфейса управление группами пользователей и компьютеров выполняется одинаково, поэтому достаточно будет рассмотреть группы пользователей. Откроем на портале управления на страницу *Пользователи и компьютеры → Группы пользователей*.

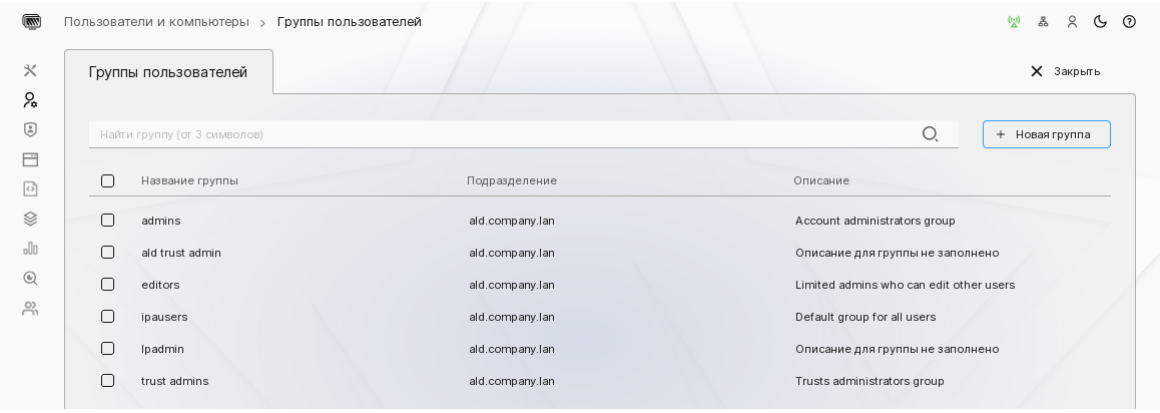


рис. 2.68 – Группы пользователей

В домене ALD Pro по умолчанию создаётся несколько предустановленных групп пользователей, имена которых используются в программном коде системы напрямую. Давайте рассмотрим эти группы:

- **admins** — специальная группа администраторов, которой напрямую назначены ACI в каталоге,

поэтому она обладает полными правами в домене.

Обратите внимание, что пользователь `admin` по умолчанию включен в локальную группу `astra-admin` на всех компьютерах ALSE в домене, что дает ему право локального администратора системы. Другие участники группы `admins` таких прав иметь не будут, т.к. участником локальной группы могут быть только конкретные пользователи системы.

- **editors** — специальная группа, которая позволяет дать пользователям права на редактирование учетных записей других пользователей, но без предоставления полных прав администратора.

- **ipausers** — специальная группа, в которую автоматически включаются все новые пользователи домена, что делает ее похожей на группу `Domain Users` из Active Directory.

  По соображениям повышения производительности группа `ipausers` по умолчанию является non-POSIX группой, то есть у нее нет идентификатора GID, и через нее не получится назначать права доступа на целевых системах.

- **lpadmin** — группа, имя которой совпадает с именем локальной группы `lpadmin`, поэтому, назначая эту группу, мы фактически предоставляем пользователю права на управление локальной подсистемой печати.

- **Trust admins** и **ald trust admin** — имена этих групп используются в скриптах `ipa-adtrust-install` и `ipa-aldtrust-install`. Если при повышении роли сервера до контроллера домена пользователь не будет включен в группу `ald trust admin`, то контроллер не сможет стать контроллером доверия и придется запускать утилиту `ipa-aldtrust-install` вручную еще раз.

Рассмотрим атрибуты, доступные на карточке группы:

- **Основные** — на этой вкладке вы можете посмотреть имя и идентификатор группы, перенести группу в другое подразделение, задать дополнительное описание и удалить группу.

- **Пользователи и Группы** – на этих вкладках вы можете изменить состав участников группы и участие самой группы в других группах.

- **Роли** — на этой вкладке вы можете назначить группе какие-нибудь роли, через которые участники группы получат права доступа на управление объектами каталога.

Подробнее о группах вы можете узнать в справочном центре https://dc-1.ald.company.lan/ad/help/ui/29.

3.3 Служба синхронизации времени

Синхронизация времени в домене необходима для корректной работы следующих механизмов: аутентификации по протоколу Kerberos, двухфакторной аутентификации на основе синхронизированных по времени одноразовых паролей, разрешения конфликтов при репликации, возможности сопоставления журналов событий с разных серверов.

3.3.1 Механизм синхронизации времени в ALD Pro

Функцию сервера времени выполняет каждый контроллер домена. Источники времени задаются через параметры групповой политики «Настройки сервиса сетевого времени Chrony» и, если параметр не задан, то по умолчанию используется иерархия, как в домене Active Directory, которая работает следующим образом:

- Корневым контроллерам домена, которые являются аналогом ePDC, в качестве источника времени назначается внешний пул NTP-серверов. Первый контроллер домена автоматически становится корневым NTP-сервером.

  Обратите внимание, что внешний пул должен обязательно содержать доступные источники, иначе корневой NTP посчитает себя недоверенным источником времени и синхронизация времени в домене работать не будет.

- Обычным контроллерам домена в качестве источника времени устанавливаются корневые NTP-серверы домена.

- Остальным серверам и рабочим станциям в качестве источника времени устанавливается любой контроллер домена.

Скрипт политики времени и даты находится в папке `rbta_ldap_date_time_h` в директории политик хоста /srv/aldpro-salt/roots/states/policies/host-policies/ (до версии ALD Pro 2.4.0 политика находилась в каталоге /srv/salt/standalone/roots/states/policies/host-policies/rbta_ldap_date_time_h).

Изменения в работе NTP в версии 2.5.0

Начиная с версии 2.5.0, система дополнительно использует параметры `minstratum` и `stratumweight`, которые обеспечивают поддержку сайтов. В этом случае рабочим станциям и рядовым серверам в домене назначаются следующие группы источников времени:

- **minstratum 12** - корневые источники времени из своего сайта;

- **minstratum 13** - обычные контроллеры из своего сайта;

- **minstratum 14** - корневые источники времени из других сайтов;
- **minstratum 15** - обычные контроллеры из других сайтов.

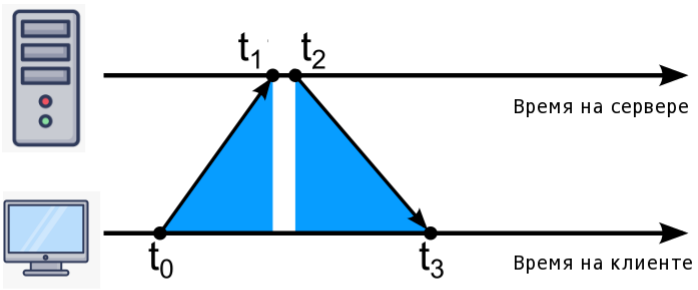
Группы источников времени задаются с помощью динамических DNS-записей, настройка которых выполняется службой **aldpro-ntp-management** автоматически. Обновление записей происходит при следующих изменениях в домене:

- изменение состава корневых NTP-серверов;
- добавление или удаление контроллера домена;
- изменение привязки контроллера к сайту.

Служба `aldpro-ntp-management` устанавливается на все контроллеры домена, но должна быть запущена только на одном из них. Она выполняет проверку healthcheck NTP каждые 2 минуты, в ходе которой получает данные из LDAP-каталога, выполняет расчет того, какие DNS-записи должны присутствовать в домене, после чего сверяет полученные результаты с фактическим состоянием. Если проверка обнаружит какие-то расхождения, то служба автоматически внесет соответствующие изменения в каталог.

3.3.2 Протокол синхронизации времени

Протокол NTP (*англ. Network Time Protocol*) — это протокол для синхронизации времени между вычислительными устройствами по компьютерным сетям с переменной задержкой. Для вычисления точного времени выполняется расчет задержки и смещения, см. рис. 2.69.



Задержка:

$$\delta = (t_3 - t_0) - (t_2 - t_1)$$

Смещение:

$$\theta = \frac{(t_1 - t_0) - (t_3 - t_2)}{2}$$

рис. 2.69 – Алгоритм синхронизации NTP

Клиент регулярно опрашивает один и более NTP-серверов, рассчитывает задержку и смещение для

каждого из них и пропускает полученные значения через статистические фильтры для смягчения последствий ошибок. Чем более симметричной окажется задержка на исходящей и входящей части маршрута, тем выше будет точность определения времени. Однако следует понимать, что NTP-клиент не использует время одного «лучшего» сервера — для повышения точности он использует данные сразу нескольких серверов.

Серверы NTP выстраиваются в иерархию, где каждый последующий уровень или слой (*англ. stratum*) обладает меньшей точностью, см. рис. 2.70:

- **stratum 0** — это авторитетные источники времени, такие как атомные часы. Серверы NTP не соответствуют этому уровню по определению, поэтому значение 0 интерпретируется как «уровень не задан».

- **stratum 1** — это серверы, которые напрямую подключены к авторитетному источнику времени уровня 0, например, к атомным часам (точность time-a.nist.gov составляет триллионные доли секунды), GPS-приемнику (точность ntpx.imvp.ru составляет миллиардные доли секунды) и др. Серверы первого уровня могут взаимодействовать между собой для проверки работоспособности.

- **stratum 2** — вторичные серверы, которые получают время от серверов stratum 1. Серверы stratum 2 обычно запрашивают время у нескольких серверов stratum 1 и могут взаимодействовать с другими серверами stratum 2, чтобы обеспечить более стабильное и достоверное время для всех устройств в одноранговой группе. При правильной настройке и выборе серверов-источников точное время имеет погрешность менее 1 миллисекунды.

- **stratum 3** — серверы, синхронизированные с серверами stratum 2. Они используют те же алгоритмы для пиринга и выборки данных, что и stratum 2, и сами могут выступать в качестве серверов для компьютеров stratum 4.

- и так далее до stratum 15 включительно.

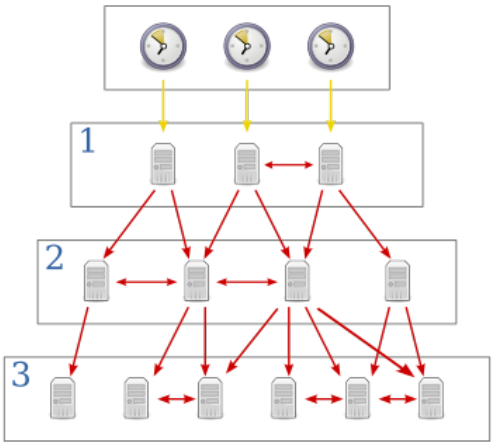


рис. 2.70 – Иерархия NTP-серверов

В качестве источника внешнего времени можно выбрать сервера stratum 2 от проекта ntp.org. По состоянию на январь 2024 года пулы содержат порядка 200 публичных серверов, из которых ~150 работают по протоколу IPv4 и ~50 по протоколу IPv6. Российские пулы доступны по следующим адресам:

- 0.ru.pool.ntp.org
- 1.ru.pool.ntp.org
- 2.ru.pool.ntp.org
- 3.ru.pool.ntp.org

Есть так же проект эталонного времени от ФГУП «ВНИИФТРИ», где находится аппаратный комплекс эталонного времени, с помощью которого формируется независимая национальная шкала времени РФ. По состоянию на 2022 год около полумиллиарда потребителей ежесуточно синхронизировали время через NTP-серверы этого проекта. Доменные адреса этих серверов:

- Москва:
  - ntp1.vniiftri.ru
  - ntp2.vniiftri.ru
  - ntp3.vniiftri.ru
  - ntp4.vniiftri.ru
- Иркутск:
  - ntp1.niiftri.irkutsk.ru
  - ntp2.niiftri.irkutsk.ru
- Хабаровск:
  - vniiftri.khv.ru
  - vniiftri2.khv.ru
- Новосибирск:
  - ntp.sstf.nsk.ru

3.3.3 Пакет chrony и служба chronyd

При установке ALD Pro (как клиентской, так и серверной части) в системе появляется служба chrony, содержание конфигурационного файла которой устанавливается автоматически через механизм групповых политик в соответствии с текущими настройками домена из раздела *Роли и службы сайта → Служба синхронизации времени*.

В состав пакета chrony входят:

- **chronyd** — это служба, которая может работать как в режиме NTP-клиента, так и NTP-сервера, обеспечивая синхронизацию времени в инфраструктуре предприятия;
- **chronyc** — это утилита, с помощью которой можно управлять службой chronyd, в том числе по сети, если эта служба работает на удаленном сервере;
- **chrony-helper** — вспомогательный скрипт `/usr/libexec/chrony/chrony-helper`, который позволяет, например, создать в операционной системе службу, которая будет извлекать имена NTP-серверов из SRV-записей.

Приведем несколько советов по работе с этой службой:

- Текущие настройки службы синхронизации времени на хосте можно посмотреть в файле `/etc/chrony/chrony.conf`.
- Текущее значение времени можно узнать в приложении «Дата и Время» или из терминала командой `timedatectl`.
- Принудительно запустить синхронизацию времени можно перезапуском службы с помощью команды `systemctl restart chrony`.
- Для взаимодействия со службой chronyd во время ее работы предназначен интерфейс командной строки `chronyc`. Чтобы увидеть, с какими серверами служба устанавливает соединение, через него можно отправить команду sources. Символом звездочки будет отмечен сервер, который был выбран в качестве основного источника времени.

```
chronyc sources -v
```

Результат выполнения команды:

```
root@dc-1:~# chronyc sources -v
...
^\* dc-1.ald.company.lan 2 6 377 51 +4571ns[ -48us] +/- 19ms
...
```

- Система ALD Pro использует параметр `makestep`, поэтому при синхронизации времени компьютер сразу устанавливает требуемое значение. Форсировать переход к целевому значению

можно вызовом команды `chronyc makestep`.

Если бы параметр `makestep` не был задан, то служба крайне медленно подтягивала бы текущее время к требуемому значению (по несколько секунд в минуту), и вам казалось бы, что синхронизация времени не работает.

- Система ALD Pro использует параметр `rtcsync`, поэтому клиенты сверяют часы каждые 11 минут, но что более важно, при синхронизации времени служба `chrony` сбрасывает флаг `STA_UNSYNC`, поэтому в приложении «Дата и время» у вас нет предупреждения об отсутствии синхронизации.

- Если вам требуется проверить работу NTP на сервере, вы можете воспользоваться командой `ntpdate -qvd dc-1.ald.company.lan`. Ключ `-q` (*англ. query only — только запросить*), говорит о том, что нужно только отправить запрос на сервер, но не менять значение системных часов. Крайне полезными являются также ключи `-v` и `-d`, включающие подробный вывод (verbose) и отладку (debugging) соответственно.

- После синхронизации команда `timedatectl` может показывать расхождение между системным временем операционной системы (Universal time) и значением времени в BIOS (RTC time, real time clock), так как запись в BIOS происходит только при выключении компьютера. Форсировать запись в BIOS можно командой `hwclock --systohc`.

И не забывайте, что при значительном изменении времени, например, после восстановления виртуальной машины из горячего снимка, все ранее выданные билеты Kerberos могут оказаться недействительными.

3.4 Служба разрешения имен

В домене основным протоколом аутентификации является Kerberos V5, который для удобства пользователей оперирует не IP-адресами, а доменными именами, поэтому для его работы нужна надежная и правильно настроенная служба разрешения имен.

В качестве DNS-сервера в FreeIPA взяли проверенную временем службу BIND9 (*англ. Berkeley Internet Name Domain version 9*), которая тесно интегрирована с каталогом, что` выражается в следующем:

- При присоединении компьютера к домену или повышении роли сервера до контроллера домена все необходимые DNS-записи создаются скриптами FreeIPA автоматически.

- Все DNS-записи хранятся в LDAP-каталоге, поэтому синхронизируются между серверами в рамках штатной процедуры репликации контроллеров домена. Доступ к каталогу BIND9 получает через плагин bind-dyndb-ldap.

- Средствами BIND9 реализована географическая балансировка нагрузки.

3.4.1 Протокол DNS

Система доменных имен (*англ. Domain Name System, DNS*) — это не что иное, как глобальная телефонная книга Интернета, в которой в колонке с номерами телефонов указаны IP-адреса, а в колонке с именами абонентов — доменные имена серверов, или правильнее сказать ресурсов, которые они предоставляют, поэтому DNS-записи называются так же ресурсными.

Когда системный администратор легко ориентируется в IP-адресах своей инфраструктуры, это традиционно уважается. Но адресов все-таки чуть больше, чем желания их запоминать, поэтому уже с первых дней ARPANET существовал файл `/etc/hosts`, сопоставляющий имена узлов с числовыми адресами.

Файл hosts был очень хорошим решением — простой, удобный, в общем, очень хороший файл. Правда, для внесения изменений в глобальную копию этого файла нужно было в рабочие часы дозвониться до Стэнфорда, а в остальном очень хороший файл. Проблема с внесением изменений стала действительно ПРОБЛЕМОЙ уже очень скоро, поэтому в дополнение к hosts разработали систему доменных имен.

На уровне клиента DNS-протокол довольно прост. Информация является общедоступной, поэтому никакая аутентификация не требуется и можно ограничиться UDP-протоколом. Достаточно отправить запрос к DNS-серверу на порт 53/UDP, и тут же получишь ответ, см. рис. 2.71.

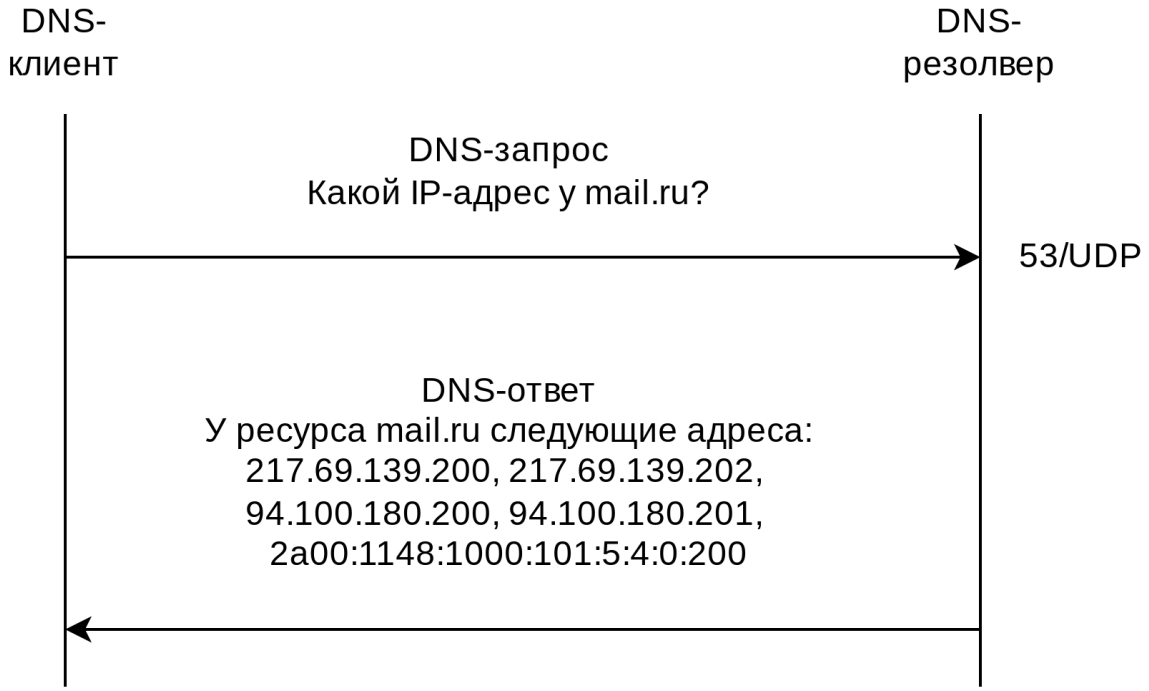


рис. 2.71 – Пример обычного DNS-запроса с получением данных по UDP

Но процесс немного усложняется, когда размер сообщения выходит за пределы 512 байт, так как в этом случае DNS-сервер вместо нормального ответа вернет пакет с флагом TC (*англ. truncated — усеченный*), и клиенту нужно будет повторить запрос, но уже на порт 53/TCP.

Например, при исследовании SRV-записей с помощью nslookup в доменах на десять контроллеров и больше вы будете сталкиваться с предупреждением о каком-то усечении (*англ. Truncated, retrying in TCP mode — усечено, повторная попытка в режиме TCP*):

```
nslookup -q=srv _kerberos._tcp.ald.company.lan
```

Результат выполнения команды:

```
admin@dc-1:~$ nslookup -q=srv _kerberos._tcp.ald.company.lan
;; Truncated, retrying in TCP mode.
Server:         127.0.0.1
Address:        127.0.0.1#53

_kerberos._tcp.ald.company.lan canonical name = _kerberos._tcp.hq._locations.ald.
↪company.lan.
_kerberos._tcp.hq._locations.ald.company.lan   service = 0 100 88 dc-5.ald.company.
↪lan.
_kerberos._tcp.hq._locations.ald.company.lan   service = 0 100 88 dc-3.ald.company.
↪lan.
_kerberos._tcp.hq._locations.ald.company.lan   service = 0 100 88 dc-1.ald.company.
↪lan.
_kerberos._tcp.hq._locations.ald.company.lan   service = 0 100 88 dc-9.ald.company.
↪lan.
_kerberos._tcp.hq._locations.ald.company.lan   service = 0 100 88 dc-4.ald.company.
↪lan.
_kerberos._tcp.hq._locations.ald.company.lan   service = 0 100 88 dc-7.ald.company.
↪lan.
_kerberos._tcp.hq._locations.ald.company.lan   service = 0 100 88 dc-2.ald.company.
↪lan.
_kerberos._tcp.hq._locations.ald.company.lan   service = 0 100 88 dc-8.ald.company.
↪lan.
_kerberos._tcp.hq._locations.ald.company.lan   service = 0 100 88 dc-6.ald.company.
↪lan.
```

Размер полезной нагрузки для ответа по UDP был ограничен лимитом в 512 байт для оптимизации работы по сети, так как в то время преобладали сети X.25, у которых MTU был 576 байт, и такой размер позволял гарантированно избежать фрагментации UDP-пакетов. После переключения на TCP-протокол клиент сможет получить ответ любого размера, даже если это будут огромные записи с сертификатами.

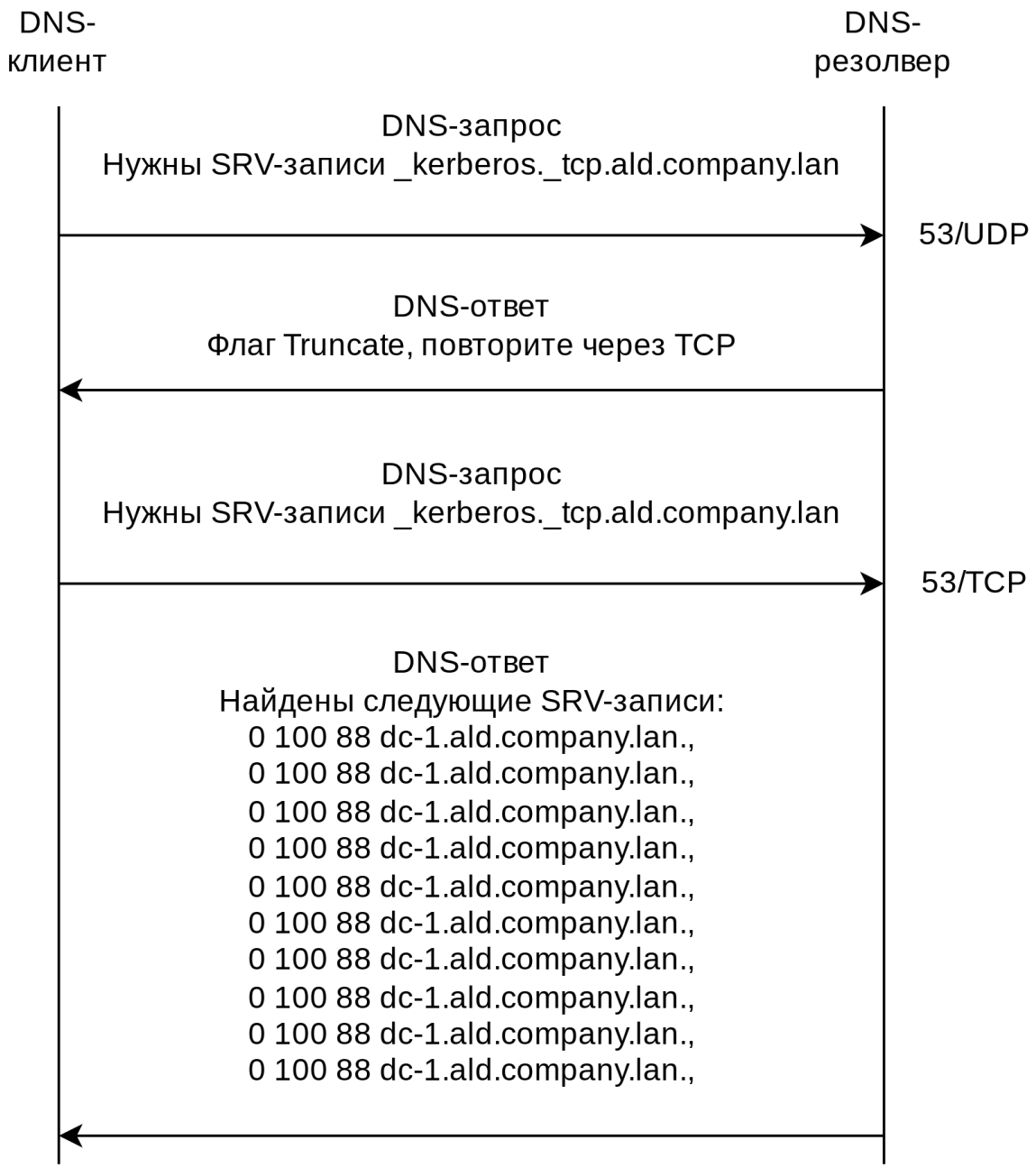


рис. 2.72 – Пример DNS-запроса с переключением на TCP

Со стороны сервера протокол выглядит значительно сложнее, чем со стороны клиента. Система DNS представляет собой распределенную базу данных, за работу которой отвечает множество серверов, которые находятся под управлением различных организаций. Эта база определяет иерархическое пространство имен, см. рис. 2.73.

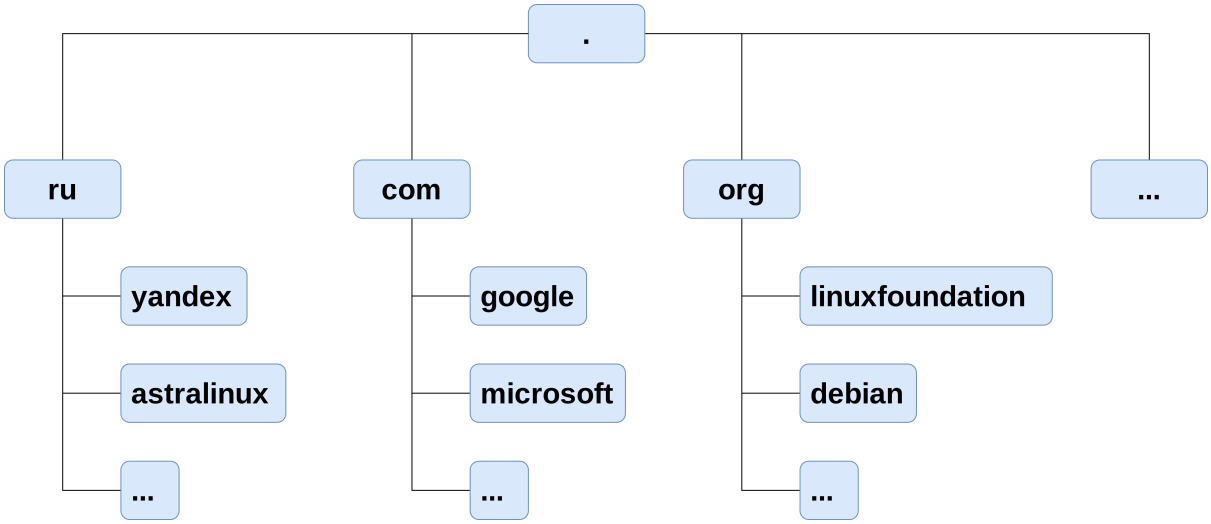


рис. 2.73 – Иерархическое пространство DNS-имен

В самом верху иерархии находится корневая (root) зона, которая обозначается символом точки. Обычно мы не пишем адреса сайтов с точкой на конце «yandex.ru.», но эта последняя точка как бы предполагается.

За работу корневой зоны отвечают тринадцать корневых серверов, имена которых имеют вид {a..m}.root-servers.net, но клиентам, которым требуется разрешать DNS-имена самостоятельно, нужно знать их «по IP-адресам». Например, в том же BIND9 эта информация находится в файле /usr/share/dns/root.hints:

```
cat /usr/share/dns/root.hints | grep ROOT-SERVERS | grep A
```

Результат выполнения:

```
admin@dc-1:~$ cat /usr/share/dns/root.hints | grep ROOT-SERVERS | grep A
.                       3600000       NS    A.ROOT-SERVERS.NET.
A.ROOT-SERVERS.NET.     3600000       A     198.41.0.4
A.ROOT-SERVERS.NET.     3600000       AAAA  2001:503:ba3e::2:30
B.ROOT-SERVERS.NET.     3600000       A     199.9.14.201
B.ROOT-SERVERS.NET.     3600000       AAAA  2001:500:200::b
C.ROOT-SERVERS.NET.     3600000       A     192.33.4.12
C.ROOT-SERVERS.NET.     3600000       AAAA  2001:500:2::c
...
```

В целях упрощения обычные DNS-клиенты не имеют информации root.hints и отправляют запросы не напрямую к NS-серверам, которые обслуживают соответствующие зоны, а через так называемые DNS-резолверы, которые выполняют всю эту работу за них. Однако, несмотря на то, что BIND9 является резолвером и может рекурсивно обходить все NS-сервера, начиная с корневой зоны, в домене для снижения нагрузки рекомендуется все-таки использовать глобальные перенаправители, например, DNS-сервер от Яндекса с IP-адресом 77.88.8.8.

Совсем сложным процесс становится, когда мы хотим защитить доменные имена от MITM-атак

(*англ. Man in the middle — человек посередине*) и добавляем технологию DNSSEC, которая позволяет подтвердить подлинность DNS-информации с использованием цифровых подписей.

Очень упрощенно это выглядит примерно так: есть корневая зона «.», которая содержит в себе записи о доменах первого уровня, таких как «ru.», «com.» и т.д. Каждая запись подписывается закрытым ключом, а открытый ключ предоставляется всем желающим, что позволяет легко проверить получаемые ответы. Например, с помощью следующего запроса вы можете получить RRSIG-запись с цифровой подписью, которая подтверждает, что зону «ru.» действительно обслуживают сервера ripn.net:

```
dig -t any +dnssec @k.root-servers.net ru.
```

Результат выполнения:

```
admin@dc-1:~$ dig -t any +dnssec @k.root-servers.net ru.
...
ru.                     172800 IN      NS      d.dns.ripn.net.
ru.                     172800 IN      NS      a.dns.ripn.net.
ru.                     172800 IN      NS      b.dns.ripn.net.
ru.                     172800 IN      NS      f.dns.ripn.net.
ru.                     172800 IN      NS      e.dns.ripn.net.
ru.                     86400  IN      DS      43786 8 2
3C59747544090BC74419D5F69E32D8C9B18EA48CBDAA33C094356191 20CED431
ru. 86400 IN RRSIG DS 8 1 86400 20240707170000 20240624160000 5613 .
HaAdFkvkNMV8gHrgX6dTTR+MLAWwy2HOSJTJTbu1dTR9m8DgnAPgUhS4
uMr61zbORe8hOFlqnUTXuEtjH1gmVaIPEmW76aGMiG07LgnKVsMDVSq9
1K2A8J4cm4/t+wru/sAidVINcyxZ7YInLhLccMR+wBRIcusvq9dlpqtA
TcT8FAFT7uxZ2j+JvnwRC2vmxLdOCXW25GPJ2OmFklp8ipQZ+kBUHdTc
sBHbmyVBXSyK63T6/3wdHsWBt33z2OCnZrSm/1koMVsZ4l8/2BEMZY6S
MpZ4o22NLOkP+RHwBp7LYOmoYn5n21QQ3cKUcsXh/JyweBlIShc0Yvx8 KMp3PQ==...
```

Однако домены второго уровня обслуживаются другими серверами, и ответы этих серверов должны быть тоже проверяемыми, поэтому ключи приходится раздавать им тоже. И далее процесс может повторяться еще несколько раз в зависимости от числа участников, обслуживающих конкретное доменное имя.

Технология DNSSEC является крайне непростой и в локальной сети, которая защищена от внешнего воздействия, использование этой технологии видится излишним, поэтому мы не будем на ней сильно останавливаться. Лучше мы расскажем немного о том, как выполнить базовую проверку работы DNS.

Самый простой способ проверить работу системы разрешения имен – это сделать пинг по доменному имени сервера. Утилита предназначена, конечно, для проверки целостности и качества соединений, но и в этом случае отлично подходит:

```
ping -c 4 yandex.ru
```

Результат выполнения:

```
admin@dc-1:~$  ping -c 4 yandex.ru
PING yandex.ru (77.88.55.88) 56(84) bytes of data.
64 bytes from yandex.ru (77.88.55.88): icmp_seq=1 ttl=247 time=9.34 ms
64 bytes from yandex.ru (77.88.55.88): icmp_seq=2 ttl=247 time=9.21 ms
64 bytes from yandex.ru (77.88.55.88): icmp_seq=3 ttl=247 time=9.19 ms
64 bytes from yandex.ru (77.88.55.88): icmp_seq=4 ttl=247 time=9.26 ms

--- yandex.ru ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 7ms
rtt min/avg/max/mdev = 9.187/9.247/9.336/0.112 ms
```

Гораздо больше информации можно получить с помощью утилиты `nslookup`, причем и в Linux, и в Windows интерфейс этой утилиты практически одинаковый. Для получения A-записей достаточно передать утилите просто имя сайта:

```
nslookup yandex.ru
```

Результат выполнения команды:

```
admin@dc-1:~$  nslookup yandex.ru
Server: 192.168.1.1
Address: 192.168.1.1#53

Non-authoritative answer:
Name: yandex.ru
Address: 77.88.44.55
Name: yandex.ru
Address: 77.88.55.88
Name: yandex.ru
Address: 5.255.255.77
Name: yandex.ru
Address: 2a02:6b8:a::a
```

И тут возникает вопрос: а что значит неавторитетный ответ (*англ. Non-authoritative answer*)? Вспоминаем, что мы обращаемся к NS-серверу не напрямую, а через вспомогательный резолвер, поэтому ответ от резолвера утилита считает недостаточно авторитетным. То есть авторитетным ответ будет только от серверов, которые указаны в NS-записях.

Как мы уже сказали, по умолчанию `nslookup` возвращает A-записи, и поменять тип записи можно с помощью ключа `-q` (query):

```
nslookup -q=ns yandex.ru
```

Результат выполнения команды:

```
admin@dc-1:~$  nslookup -q=ns yandex.ru
Server: 192.168.1.1
Address: 192.168.1.1#53

Non-authoritative answer:
yandex.ru       nameserver = ns1.yandex.ru.
```

(continues on next page)

(продолжение с предыдущей страницы)

```
yandex.ru       nameserver = ns2.yandex.ru.

Authoritative answers can be found from:
ns1.yandex.ru internet address = 213.180.193.1
ns2.yandex.ru internet address = 93.158.134.1
ns1.yandex.ru has AAAA address 2a02:6b8::1
ns2.yandex.ru has AAAA address 2a02:6b8:0:1::1
```

Как мы видим, зону yandex.ru обслуживает в том числе сервер ns1.yandex.ru с IP-адресом 213.180.193.1, поэтому если мы спросим про яндекс у него, то ответ уже будет считаться авторитетным. Для этого нужно указать адрес DNS-сервера после доменного имени:

```
nslookup yandex.ru 213.180.193.1
```

Результат выполнения команды:

```
admin@dc-1:~$  nslookup yandex.ru 213.180.193.1
Server: 213.180.193.1
Address: 213.180.193.1#53

Name: yandex.ru
Address: 77.88.55.88
Name: yandex.ru
Address: 77.88.44.55
Name: yandex.ru
Address: 5.255.255.77
Name: yandex.ru
Address: 2a02:6b8:a::a
```

Утилита `nslookup` может порадовать вас также интерактивным режимом, если вы запустите ее без параметров. Чтобы переключиться на другой DNS-сервер, вам нужно воспользоваться командой `server`, а чтобы запрашивать другой тип записей, установите переменную `type` с помощью команды `set`. Вот пример работы с утилитой в интерактивном режиме:

```
admin@dc-1:~$ nslookup
> yandex.ru
Server: 192.168.1.1
Address: 192.168.1.1#53

Non-authoritative answer:
Name: yandex.ru
Address: 77.88.44.55
Name: yandex.ru
Address: 77.88.55.88
Name: yandex.ru
Address: 5.255.255.77
Name: yandex.ru
Address: 2a02:6b8:a::a
>
> server 213.180.193.1
Default server: 213.180.193.1
Address: 213.180.193.1#53
```

(continues on next page)

(продолжение с предыдущей страницы)

```
>
> set type=mx
>
> mail.yandex.ru
Server: 213.180.193.1
Address: 213.180.193.1#53

Non-authoritative answer:
*** Can't find mail.yandex.ru: No answer

Authoritative answers can be found from:
mail.yandex.ru nameserver = ns4.yandex.ru.
mail.yandex.ru nameserver = ns3.yandex.ru.
> exit
```

Вроде бы все работает, но почему полученный результат опять не является авторитетным, если мы запрашиваем информацию напрямую у сервера Yandex? Ответ на самом деле прост: почтовому сервису Яндекса делегирована вся зона целиком и обслуживается она на других серверах, чтобы команда проекта могла управлять ей самостоятельно. Поэтому повторите запрос к серверу ns4.yandex.ru, и все будет уже вполне авторитетно:

```
ping -c 1 ns4.yandex.ru
nslookup -q=mx mail.yandex.ru 77.88.21.1
```

Результат выполнения команды:

```
admin@dc-1:~$ ping -c 1 ns4.yandex.ru
PING ns4.yandex.ru (77.88.21.1) 56(84) bytes of data.
64 bytes from ns4.yandex.ru (77.88.21.1): icmp_seq=1 ttl=56 time=3.18 ms

--- ns4.yandex.ru ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 3.179/3.179/3.179/0.000 ms

admin@dc-1:~$ nslookup -q=mx mail.yandex.ru 77.88.21.1
Server: 77.88.21.1
Address: 77.88.21.1#53
mail.yandex.ru mail exchanger = 0 mx.yandex.ru.
```

Утилита `nslookup` отлично подходит для решения большинства вопросов, т.к. через нее можно получить основную информацию о работе доменной службы, но гораздо больше информации вы сможете получить с помощью утилиты `dig`. Многие считают, что название утилиты `dig` означает «копать», что недалеко от истины, но на самом деле это название является аббревиатурой от Domain Information Groper (сборщик информации о домене). Мы настоятельно рекомендуем переходить к утилите `dig` только после того, как вы станете уверенно работать с утилитой `nslookup`.

3.4.2 Конфигурационные файлы службы разрешения имен

В каталоге `/etc/bind` на контроллере домена содержатся следующие конфигурационные файлы:

- Файл `named.conf` — основной конфигурационный файл с настройками BIND9. Как раз в этом файле подключается плагин `bind-dyndb-ldap`.
- Файл `named.conf.options` — содержит параметры DNS-сервера.
- Файл `/etc/bind/named.conf.default-zones` — содержит зоны по умолчанию. Тут нас в первую очередь должны интересовать настройки корневой зоны «.», где указано, что нужно использовать файл `/usr/share/dns/root.hints`.

  В файле `root.hints` перечислены IP-адреса всех корневых DNS-серверов, обслуживающих Интернет, поэтому BIND9 даже при отсутствии глобального перенаправителя способен самостоятельно разрешать доменные имена.
- Файл `/etc/bind/zones.rfc1918` — содержит список частных обратных зон (10.x.x.x и так далее).
- Файл `/etc/bind/bind.keys` — файл с ключами для обеспечения работы DNSSEC. Технология DNSSEC еще не получила большого распространения и требует отдельного рассмотрения, поэтому ее предлагается не использовать.
- Файл `/etc/bind/db.*` — файлы зон DNS.
- Файл `/etc/bind/rndc.key` — файл с ключами для аутентификации DHCP-сервера (если мы захотим, чтобы DHCP мог автоматически обновлять DNS-записи хостов).

Кроме стандартных файлов в каталоге есть два дополнительных файла, предназначенных для работы FreeIPA:

- Файл `/etc/bind/ipa-ext.conf` — подключается в `named.conf` и предназначен для определения дополнительных настроек службы BIND9, которые не будут изменены при обновлении сервера FreeIPA. Например, в этом файле вы можете определить перечень доверенных сетей. По умолчанию все строки в этом файле закомментированы:

```
admin@dc-1:~$ sudo cat /etc/bind/ipa-ext.conf
/* User customization for BIND named
*
* This file is included in /etc/bind/named.conf and is not modified during IPA
* upgrades.
* "options" settings must be configured in /etc/bind/ipa-options-ext.conf.
*
* Example: ACL for recursion access:
*
* acl "trusted_network" {
```

(continues on next page)

(продолжение с предыдущей страницы)

```
*   localnets;
*   localhost;
*   234.234.234.0/24;
*   2001::co:ffee:babe:1/48;
*  };
*/
```

- Файл `/etc/bind/ipa-options-ext.conf` — подключается в секции `options` файла `named.conf` и предназначен для определения дополнительных опций службы BIND9, которые не будут изменены при обновлении сервера FreeIPA.

  Этот файл мы уже изменяли, чтобы отключить DNSSEC и разрешить рекурсивные запросы, поэтому его содержимое в нашем случае выглядит следующим образом:

```
admin@dc-1:~$ sudo cat /etc/bind/ipa-options-ext.conf
dnssec-validation no;
allow-recursion { any; };
allow-query-cache { any; };
```

Содержимое файла `/etc/bind/named.conf` управляется службой FreeIPA. Тут наибольший интерес представляют следующие настройки:

- **Секция logging** — указывает, что журнал работы BIND9 будет доступен по адресу `/var/cache/bind/named.run`;
- **Инструкции include** — подключают дополнительные конфигурационные файлы.
- **Секция dyndb** — настраивает LDAP-плагин для подключения к каталогу по unix-сокету.

```
/* WARNING: This config file is managed by IPA.
*
* DO NOT MODIFY! Any modification will be overwritten by upgrades.
*
*
* - /etc/bind/ipa-options-ext.conf (for options)
* - /etc/bind/ipa-ext.conf (all other settings)
*/

options {
      // Put files that named is allowed to write in the data/ directory:
      directory "/var/cache/bind"; // the default
      dump-file               "cache_dump.db";
      statistics-file         "named_stats.txt";
      memstatistics-file      "named_mem_stats.txt";

      tkey-gssapi-keytab "/etc/bind/named.keytab";

      pid-file "/run/named/named.pid";

      managed-keys-directory "/var/cache/bind/dynamic";
```

(continues on next page)

(продолжение с предыдущей страницы)

```
      /* user customizations of options */
      include "/etc/bind/ipa-options-ext.conf";

      /* crypto policy snippet on platforms with system-wide policy. */
      // not available
};

/* If you want to enable debugging, eg. using the 'rndc trace' command,
* By default, SELinux policy does not allow named to modify the /var/named directory,
* so put the default debug log file in data/ :
*/
logging {
      channel default_debug {
               file "named.run";
               severity dynamic;
               print-time yes;
      };
};

//zone "." IN {
//      type hint;
//      file "named.ca";
//};

include "/etc/bind/named.conf.default-zones";
include "/etc/bind/bind.keys";

/* user customization */
include "/etc/bind/ipa-ext.conf";

dyndb "ipa" "/usr/lib/bind/ldap.so" {
      uri "ldapi://%2fvar%2frun%2fslapd-ALD-COMPANY-LAN.socket";
      base "cn=dns,dc=ald,dc=company,dc=lan";
      server_id "dc-1.ald.company.lan";
      auth_method "sasl";
      sasl_mech "GSSAPI";
      sasl_user "DNS/dc-1.ald.company.lan";
};
```

Если заглянуть в LDAP-каталог с помощью Apache Directory Studio, см рис. 2.74, то вы сможете увидеть, что в атрибутах записи с DN «cn=dns,dc=ald,dc=company,dc=lan», хранятся настройки, которые мы задаем на странице *Роли и службы сайта → Служба разрешения имен* вкладка *Глобальная конфигурация DNS*, см. рис. 2.75.

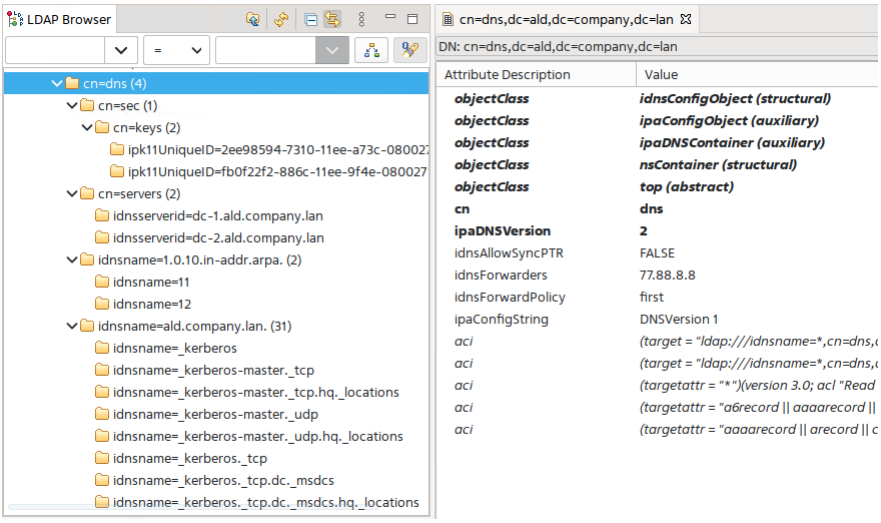


рис. 2.74 – Содержимое записи «cn=dns,dc=ald,dc=company,dc=lan»

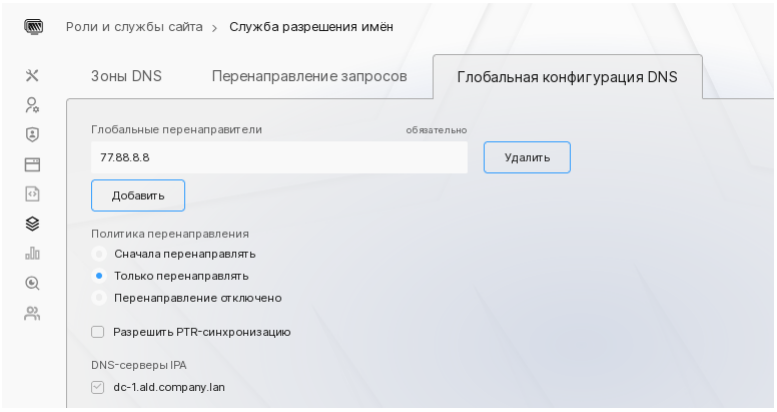


рис. 2.75 – Глобальная конфигурация DNS

3.4.3 Зоны DNS и параметры зоны

На вкладке «Зоны DNS» представлен список прямых и обратных зон, которые вы можете создавать, удалять и редактировать, см. рис. 2.76. Однако следует учитывать, что встроенный DNS-сервер ориентирован в первую очередь на потребности домена и целый ряд настроек BIND9 вам будет недоступен. Поэтому, если вам потребуются зоны со специфическими настройками, рекомендуется развернуть отдельную DNS-инфраструктуру, а в домене настроить перенаправление.

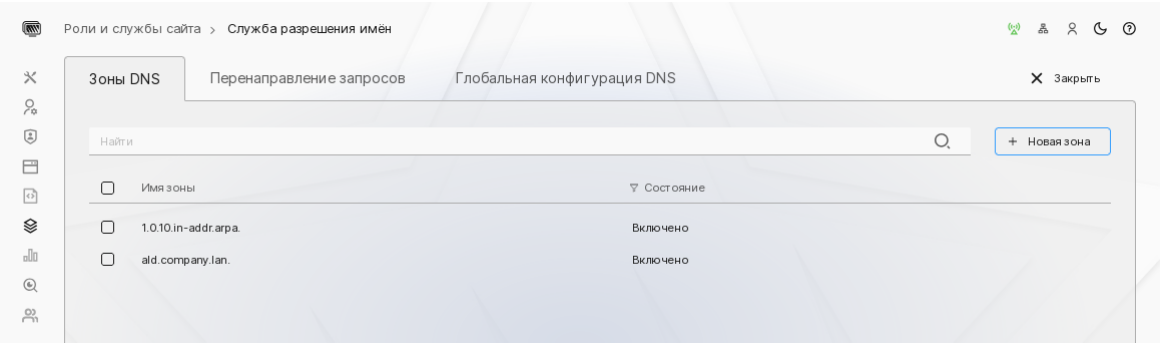


рис. 2.76 – Список DNS-зон

Откроем прямую зону «ald.company.lan» и перейдем на вкладку «Параметры зоны», см. рис. 2.77:

- **Полномочный сервер имен** (primary name server) — сервер, который является владельцем этой DNS-зоны.
- **Адрес электронной почты администратора** (Responsible Person) — значение является обязательным, но не имеет значения для доменной зоны предприятия.
- **Номер SOA** (Serial Number) – серийный номер зоны, который может быть использован подчиненными DNS-серверами для синхронизации.

  Обратите внимание, что в домене FreeIPA серийный номер исключен из репликации во избежание «состояния гонок», поэтому это значение формируется каждым сервером индивидуально. Это не является проблемой для контроллеров, т.к. они реплицируют DNS-записи через LDAP, но, если вы будете настраивать репликацию Master-Slave со сторонней DNS-инфраструктурой, вам следует учитывать, что при переключении на другой DNS-сервер нужно будет принудительно выполнить загрузку зоны во избежание потерь информации.
- **Обновление SOA** (Refresh) – периодичность отправки запросов к владельцу зоны для обновления DNS-записей, значение указывается в секундах.
- **Повторный запрос SOA** (Retry) – время в секундах, через которое будет выполнен повторный запрос информации к серверу-владельцу, если он окажется недоступен.
- **Окончание действия SOA** (Expire) – время (в секундах), в течение которого вторичный сервер будет пытаться завершить синхронизацию зоны с первичным DNS-сервером.
- **Минимальный срок жизни SOA** (Minimum TTL) – время жизни кэша для негативного ответа на запрос в зоне.
- **Срок жизни по умолчанию** (Default TTL) – время по умолчанию, в течение которого информация будет кэшироваться другими DNS-серверами.

- **Срок жизни** (TTL) – время, в течение которого информация будет кэшироваться другими DNS-серверами.
- **Динамическое обновление** (Allow-update) – позволяет клиентам и DHCP-серверу автоматически обновлять DNS-записи для поддержания их в актуальном состоянии. Может быть разрешено или запрещено.
- **Разрешить запрос** (Allow-query) – определяет, кому разрешено запрашивать записи из этой зоны. По умолчанию разрешено всем («any»).
- **Разрешить перенос** (Allow-transfer**)** – определяет, каким Slave-серверам разрешен перенос зоны, т.е. её копирование. По умолчанию не разрешено никому («none»).
- **Глобальные перенаправители** (Global forwarders) и **Политика перенаправления** (Update-policy) – тут можно указать один или более глобальных перенаправителей и определить политику перенаправления.
- **Состояние зоны DNS** – позволяет отключить зону, не удаляя ее. Это может потребоваться в процессе отладки.

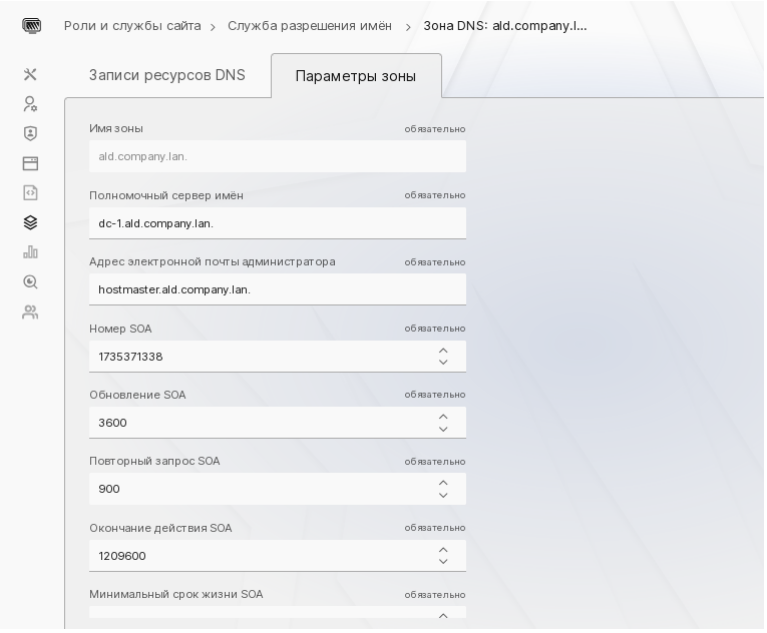


рис. 2.77 – Параметры зоны ald.company.lan

Как мы можем убедиться, в LDAP-каталоге информация зоны хранится в записи с DN «idnsname=ald.company.lan,cn=dns,dc=ald,dc=company,dc=lan».

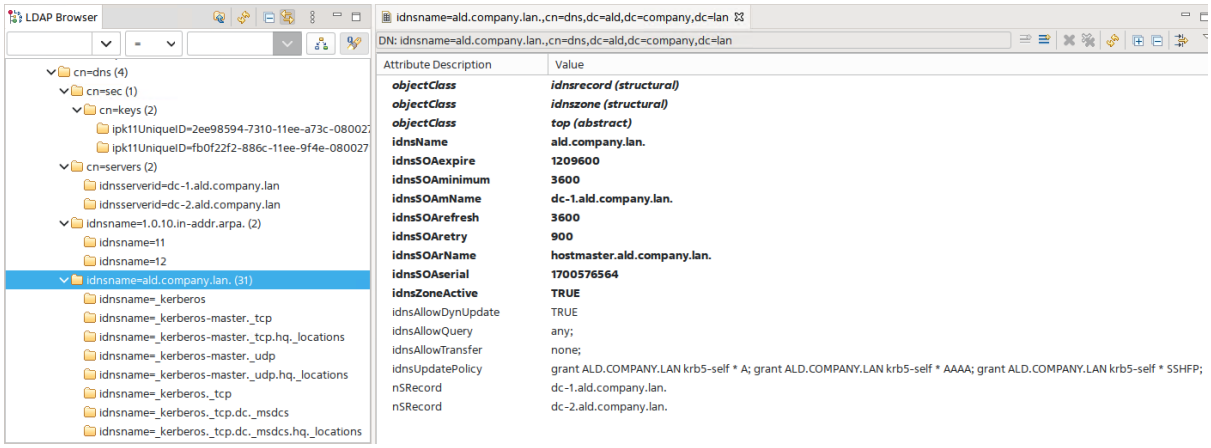


рис. 2.78 – Запись «idnsname=ald.company.lan,cn=dns,dc=ald,dc=company,dc=lan»

3.4.4 Записи в доменной зоне

На вкладке *Записи ресурсов DNS* мы увидим все DNS-записи этой зоны, см. рис. 2.79.

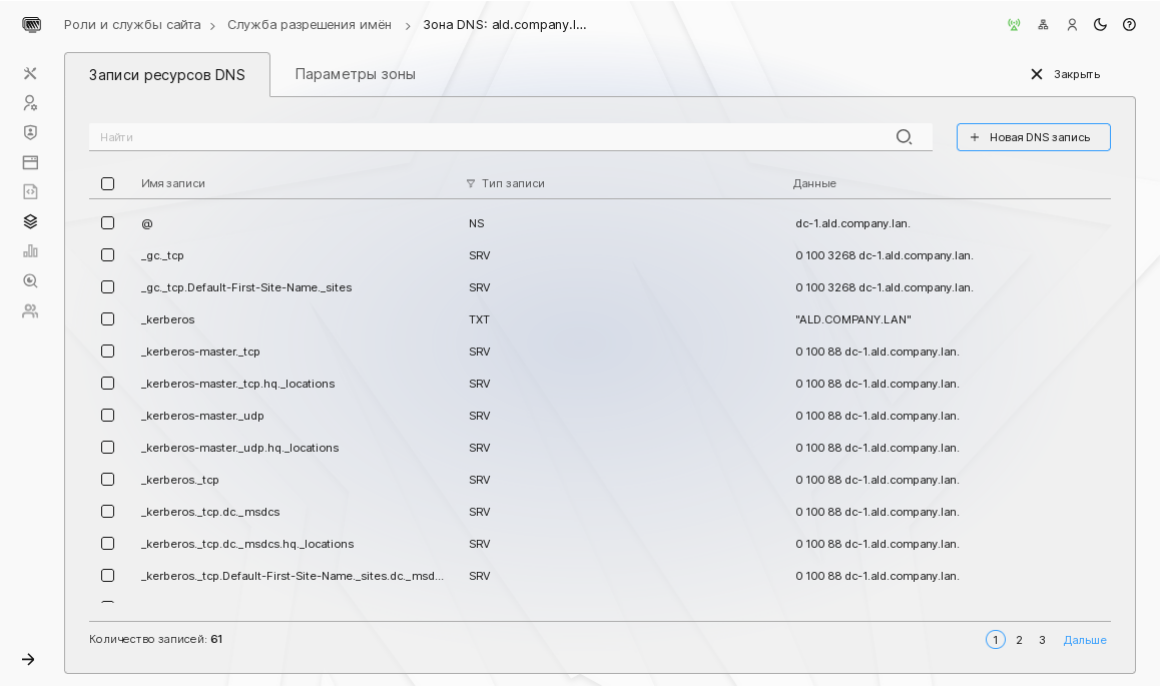


рис. 2.79 – Записи в зоне ald.company.lan

Каждая ресурсная запись содержит следующие атрибуты:

- **Имя (NAME)** — адрес, к которому «принадлежит» данная ресурсная запись.
- **Тип (TYPE)** — задает тип ресурсной записи, который определяет ее назначение.
- **Поле данных (RDATA)** — данные ресурсной записи, формат которых зависит от типа.

Если открыть одну из SRV-записей _kerberos._tcp, которые используются для обнаружения центров распространения ключей Kerberos, когда к ним нужно обратиться по протоколу TCP, см. рис. 2.80, то в поле данных мы видим приоритет, вес, порт и цель.

В интерфейсе отображается только три основных атрибута, но следует понимать, что есть еще:

- **Класс (CLASS)** — определяет тип сетей, добавлен из предположения, что DNS может использоваться не только в TCP/IP, но и с другими типами сетей.
- **TTL (Time To Live)** — допустимое время хранения данной ресурсной записи в кэше неответственного DNS-сервера.
- **Длина поля данных (RDLEN)** — дополнительная служебная информация.

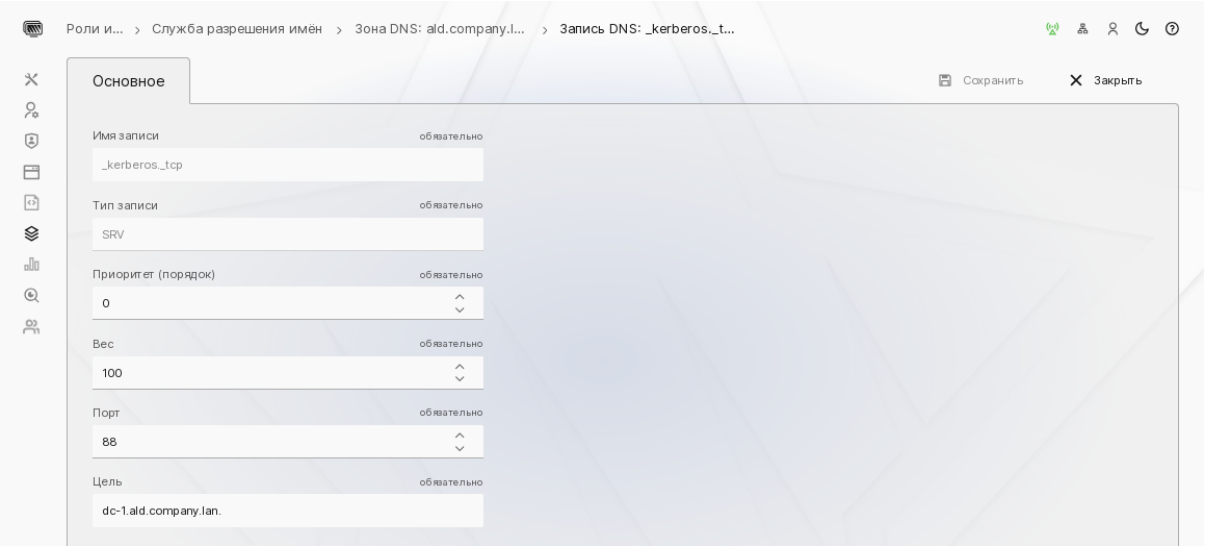


рис. 2.80 – DNS-запись _kerberos._tcp

Приведем список наиболее важных DNS-записей:

- **Запись A** (address record) или запись адреса — связывает имя хоста с адресом протокола IPv4, то есть запрос A-записи для имени dc-1.ald.company.lan вернет IPv4-адрес сервера dc-1 в домене ald.company.lan.
- **Запись AAAA** (IPv6 address record) — работает как A-запись только с протоколом IPv6.
- **Запись CNAME** (canonical name record) или каноническая запись имени (псевдоним) — используется для перенаправления на другое имя.
- **Запись MX** (mail exchange) или почтовый обменник — указывает сервер, который принимает входящую почту для домена.
- **Запись NS** (name server) — делегирует обслуживание зоны другому DNS-серверу.

- **Запись PTR** (pointer), обратная DNS-запись или запись указателя — позволяет по IP-адресу получить каноническое FQDN имя хоста.

  В домене Kerberos PTR-записи используются как средство защиты от различных атак. Например, при вводе машины в домен недостаточно в качестве DNS-сервера указать DNS-сервер из инфраструктуры MS AD, где настроено перенаправление прямой зоны. Нужно настроить перенаправление обратной зоны тоже или создать PTR-записи для контроллеров домена ALD Pro на DNS-сервере Microsoft.

- **SRV-запись** (server selection) — позволяет задать список серверов, которые предоставляют определенные сервисы в сети, например, LDAP, Kerberos, NTP. Эти записи необходимы для работы механизмов автоматического обнаружения сервисов и далее мы рассмотрим их подробнее.

- **TXT-запись** (text) — позволяет хранить в DNS любые настройки, например, политики обработки спама (Sender Policy Framework, SPF), публичный сертификат для проверки писем получателями с использованием цифровых подписей (DomainKeys Identified Mail, DKIM). С помощью текстовых записей, например, провайдеры могут запрашивать подтверждение того, что вы владеете доменным именем.

Вернемся к SRV-записям т.к. они очень активно используются в домене. Эти записи определяют местоположение определенных служб, то есть имя хоста и номер порта. SRV-запись имеет такой формат:

```
_<service>._<proto>.<name> <TTL> <class> <SRV> <priority> <weight> <port> <target>
```

Где:

- **service** — символьное имя сервиса (например, _ldap).

- **proto** — транспортный протокол, используемый сервисом (TCP или UDP).

- **name** — доменное имя, для которого эта запись действует.

- **TTL** — время жизни.

- **class** — поле класса (это всегда IN).

- **SRV** — тип записи (для SRV-записей всегда SRV).

- **priority** — приоритет целевого хоста, более низкое значение означает, что этот сервер более предпочтителен.

- **weight** — позволяет распределить нагрузку среди серверов с одинаковым приоритетом. Более высокое значение означает, что этот сервер более предпочтителен.

- **port** — номер порта TCP или UDP, на котором работает сервис.
- **target** — каноническое имя машины, предоставляющей сервис.

Вы можете добавить в домене свои собственные SRV-записи, но следует понимать, что служба каталога FreeIPA для своих сервисов создает немного специфичные SRV-записи. Этим записям в каталоге назначен класс idnsTemplateObject и задано дополнительное свойство «idnsTemplateAttribute;cnamerecord», которое определяет шаблон подстановки для поддержки технологии сайтов (локаций), см. рис. 2.81.

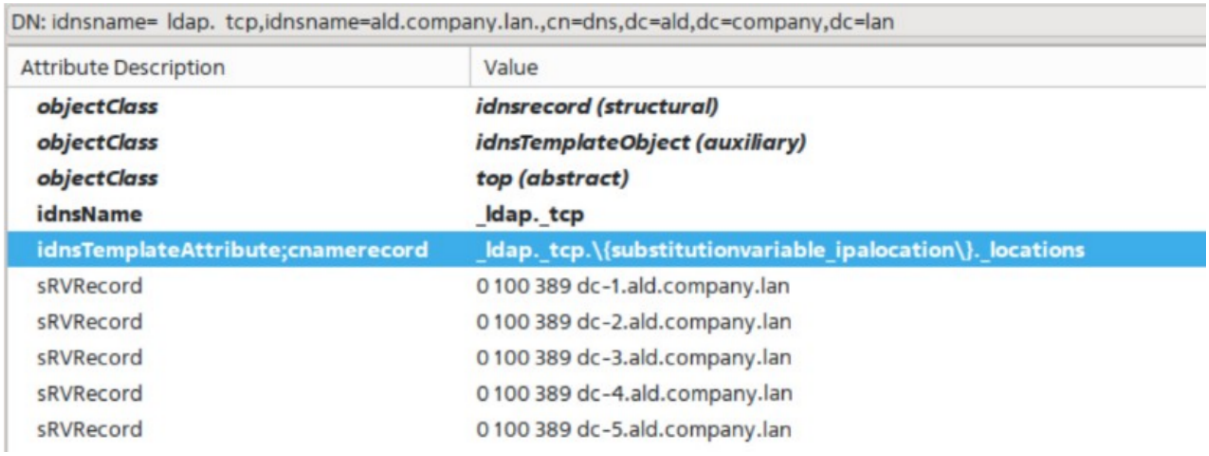


рис. 2.81 – Шаблон idnsTemplateAttribute в SRV-записях службы FreeIPA

Для привязки серверов к сайтам у записей DN «cn=servers,cn=dns,dc=ald,dc=company,dc=lan» задано значение атрибута «idnsSubstitutionVariable;ipalocation», соответствующее имени сайта, в котором этот сервер находится, см. рис. 2.82.

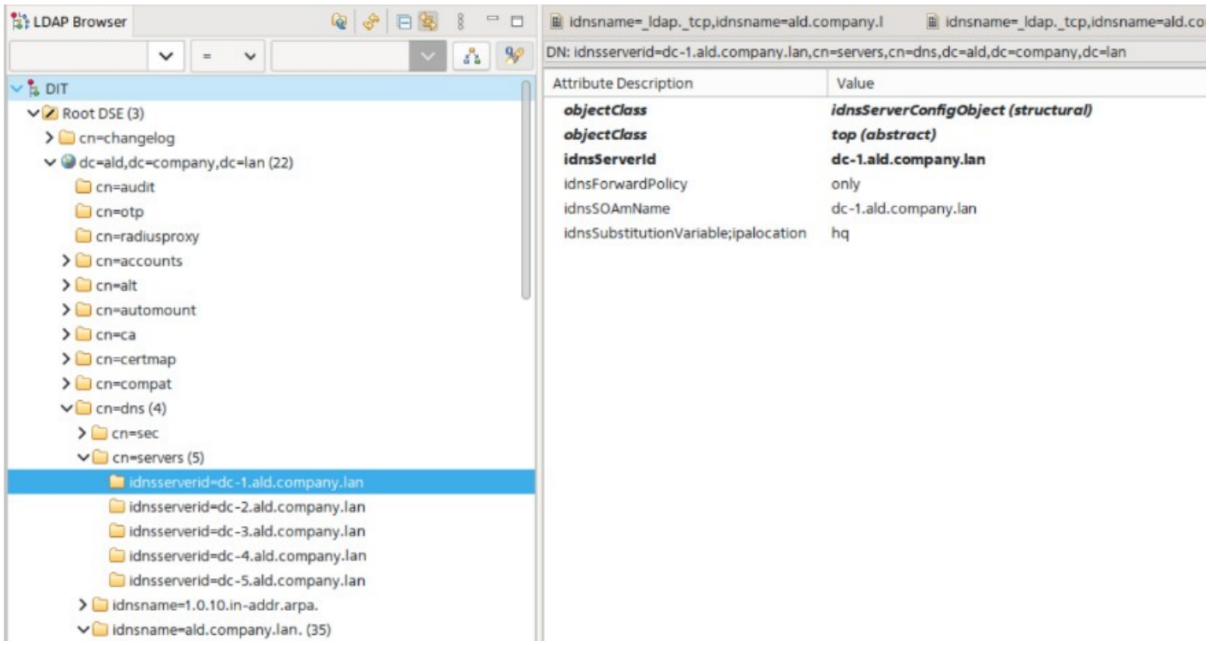


рис. 2.82 – Атрибут idnsSubstitutionVariable для контроллера домена

Например, при запросе _ldap._tcp.ald.company.lan у DNS-службы на контроллере 10.0.1.11 (dc-1.ald.company.lan) сервис BIND9 выполнит подстановку и вернет результат для адреса _ldap._tcp.hq_locations.ald.company.lan, что соответствует сайту головного офиса hq (*англ. head quarters*).

```
nslookup -q=SRV _ldap._tcp.ald.company.lan
```

Результат выполнения:

```
admin@dc-1:~$ nslookup -q=SRV _ldap._tcp.ald.company.lan
Server: 127.0.0.1
Address: 127.0.0.1#53
_ldap._tcp.ald.company.lan canonical name = _ldap._tcp.hq._locations.ald.company.lan.
_ldap._tcp.hq._locations.ald.company.lan service = 50 100 389 dc-4.ald.company.lan.
_ldap._tcp.hq._locations.ald.company.lan service = 0 100 389 dc-1.ald.company.lan.
_ldap._tcp.hq._locations.ald.company.lan service = 0 100 389 dc-2.ald.company.lan.
_ldap._tcp.hq._locations.ald.company.lan service = 50 100 389 dc-3.ald.company.lan.
_ldap._tcp.hq._locations.ald.company.lan service = 50 100 389 dc-5.ald.company.lan.
```

Как вы можете заметить, в список будут включены все контроллеры домена, но для серверов из сайта hq приоритет будет равен 0, а у всех остальных 50, поэтому служба SSSD будет в первую очередь обращаться к контроллерам из «своего» сайта. Таким образом, привязка хостов к сайтам выполняется через DNS-сервер, который их обслуживает.

3.5 Практика и тестирование

3.5.1 Практическая работа: Модуль 3. Базовые функции и настройка ALD Pro

См. также *Требования, правила и цели выполнения практической работы*

3.5.1.1 Практические задания

Для отработки практических навыков мы предлагаем вам подготовить стенд, который был описан в материалах модуля. В этом документе мы собрали очень краткие подсказки, чтобы вы могли проверить себя, получится ли у вас без расширенных материалов модуля вспомнить все существенные технические детали.

Задача 1

- Изучите конфигурацию службы `chrony` на dc-1 в файле `/etc/chrony/chrony.conf`.
- Посмотрите состояние синхронизации в окне приложения `Дата и время` на dc-1.
- Изучите конфигурацию `chrony` на pc-1.

- Откройте вкладку в графическом интерфейсе даты и время на pc-1.

Задача 2

- Проверьте источники на dc-1 командой `sudo chronyc sources -v`.
- Проверьте источники на pc-1 командой `sudo chronyc sources -v`.

Задача 3

- Проверьте работу NTP-сервера с pc-1 командой `sudo ntpdate -qvd dc-1.ald.company.lan`.

Задача 4

- Проверьте расхождение во времени на dc-1 и pc-1.
- Синхронизируйте время на dc-1 и pc-1 командой `sudo chronyc makestep`.
- Сохраните текущее время в BIOS командой `sudo hwclock --systohc`.

Задача 5

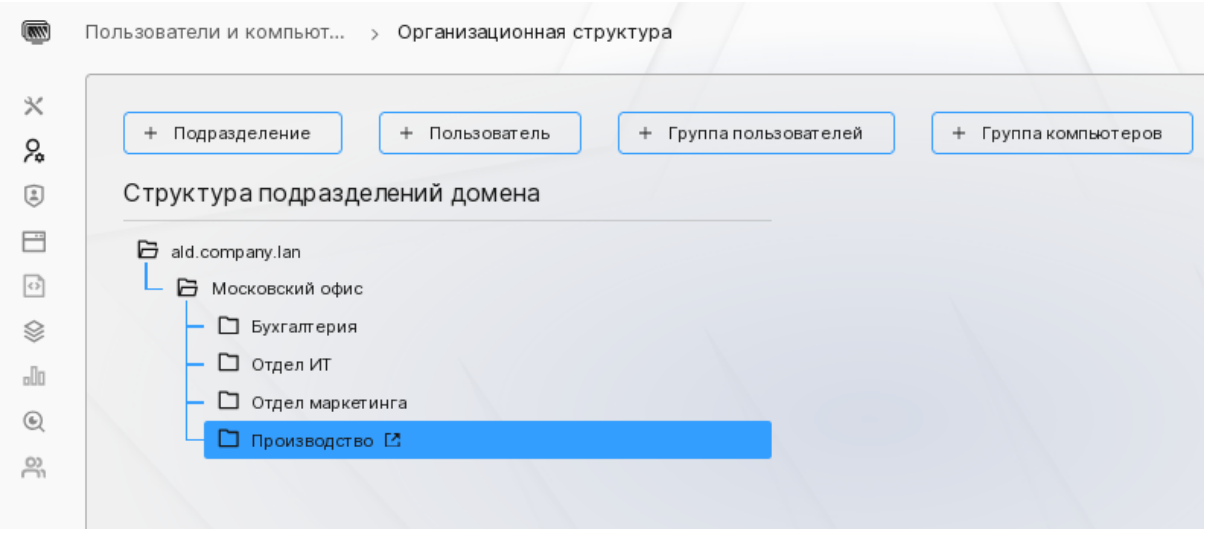
- Посмотрите на содержимое каталога `/etc/bind` на dc-1.
- Изучите конфигурационные файлы bind.

Задача 6

- Установите Apache Directory Studio и изучите В DNS-записи по пути `cn=dns,dc=ald,dc=company,dc=lan`.
- Создайте подключение, используя для подключения учетную запись администратора `uid=admin,cn=users,cn=accounts,dc=ald,dc=company,dc=lan`.

Задача 7

- Настройте простую организационную структуру, как показано на рисунке ниже:



Задача 8

Создайте четырех пользователей со следующими значениями атрибутов:

Атрибут	Пользователь 1	Пользователь 2	Пользователь 3	Пользователь 4
Логин	iivanov	ppetrov	asidorov	vsidorova
Фамилия	Иванов	Петров	Сидоров	Сидорова
Имя	Иван	Петр	Александр	Виктория
Отчество	Иванович	Петрович	Александрович	Александровна
Подразделение	Отдел ИТ	Отдел маркетинга	Производство	Бухгалтерия
Пароль	Pa$$w0rd	Pa$$w0rd	Pa$$w0rd	Pa$$w0rd

Задача 9

- Включите (если выключена) виртуальную машину pc-1 и войдите в нее под новой учетной записью `ppetrov`.
- Откройте через портал управления удаленным рабочим столом для оказания технической поддержки сотруднику `ppetrov`.

3.5.2 Тест: Модуль 3. Базовые функции и настройка ALD Pro

Пожалуйста, ответьте на несколько вопросов, чтобы закрепить полученные знания. Правильные ответы для самопроверки вы сможете найти в конце списка. Обратите внимание, что у некоторых вопросов может быть несколько верных вариантов ответов.

3.5.2.1 Вопросы

Вопрос 1. Что такое серверы stratum 2?

a. Протокол службы синхронизации

b. Вторичные NTP-серверы, которые получают точное время от серверов stratum 1

c. Команда обновления времени

d. Параметр нужен для синхронизации времени каждые 2 минуты

Вопрос 2. Какой командой можно форсировать переход к целевому значению времени?

a. Команда `sudo hwclock --systohc`

b. Команда `timedatectl`

c. Команда `sudo chronyc makestep`

d. Команда `sudo chronyc sources -v`

Вопрос 3. В каком файле хранится конфигурация службы chrony?

a. Файл `/etc/chrony.conf`

b. Файл `/etc/chrony/ald-chrony.conf`

c. Файл `/etc/chrony/ipa-chrony.conf`

d. Файл `/etc/chrony/chrony.conf`

Вопрос 4. Какой командой можно записать текущее время в BIOS?

a. Команда `timedatectl`

b. Команда `sudo hwclock --systohc`

c. Команда `sudo chronyc makestep`

d. Команда `sudo ntpdate`

Вопрос 5. В каком файле находятся опции DNS для FreeIPA?

a. Файл `/etc/bind/ipa-options-ext.conf`

b. Файл `/etc/bind/ald-options-ext.conf`

c. Файл `/etc/bind/ipa-params.*`

d. Файл `/etc/bind/ipa.rfc1918`

Вопрос 6. Какая запись связывает имя хоста с адресом протокола IPv4?

a. AAAA-запись

b. A-запись

c. PTR-запись

d. SRV-запись

Вопрос7. По какому DN в LDAP-каталоге находится информация о записях DNS?

a. dns=dns,dc=ald,dc=company,dc=lan

b. cn=dns,dc=ald,dc=company,dc=lan

c. cn=dns,dc=ald.company.lan

d. dns=ald.company.lan,dc=ald,dc=company,dc=lan

Вопрос 8. Какой тип записи в доменах FreeIPA и Active Directory используется для того, чтобы задать настройки для подключения к LDAP, Kerberos и другим службам?

a. TTL-запись

b. SOA-запись

c. SRV-запись

d. AAAA-запись

Вопрос 9. Какая SRV-запись используется для поиска сервиса KDC?

a. Запись `_kerberos._tcp`

b. Запись `_ntp._udp`

c. Запись `_ldap._tcp`

d. Запись `ttl 9600`

Вопрос 10. Какая запись указывает на сервис NTP?

a. Запись `_kerberos._tcp`

b. Запись `_ntp._udp.ald.lan.`

c. Запись `_ldap._tcp`

d. Запись `ttl 9600`

Вопрос 11. Какой специальный атрибут для ссылки на структурные подразделения использует система ALD Pro?

a. Атрибут `rbta-dp`

b. Атрибут `rbtahq`

c. Атрибут `rbtadp`

d. Атрибут `alddp`

Ответы на вопросы

1. b
2. c
3. d
4. b
5. a
6. b
7. b
8. c
9. a
10. b
11. c

3.6 Заключение

В этом модуле мы познакомились поближе с архитектурой портала управления и двумя сетевыми службами контроллера домена: NTP и DNS. В следующем модуле мы перейдем к таким компонен-

там, как KDC, LDAP, и узнаем, зачем нужна Samba AD на контроллерах FreeIPA.

3.7 Дополнительные источники информации

- Проект эталонного времени от ФГУП «ВНИИФТРИ»
- Рекомендуем ознакомиться также с видеороликами нашего хорошего товарища на канале «ИТ-Проповедник». Учитывайте, что ролики записаны для версии 2.2.1, поэтому часть информации потеряла актуальность, но в целом они могут быть крайне полезны.
    - Установка Apache Directory Studio на Astra Linux 1.7 в домене ALD Pro

3.8 Обратная связь

Если остались вопросы, то их всегда можно задать в специальной теме.

Модуль 4. Работа службы каталога ALD Pro

4.1 Введение

В этом модуле мы рассмотрим общую архитектуру службы каталога ALD Pro, устройство LDAP-каталога, функции KDC-сервера и даже узнаем, зачем на контроллерах домена FreeIPA устанавливается Samba.

Крайне полезными будут практические материалы по управлению компонентами системы и информация про идентификаторы безопасности.

4.2 Архитектура службы каталога ALD Pro

Система ALD Pro (Astra Linux Directory Pro) — это набор сетевых служб сервера Astra Linux для организации централизованного управления ИТ-инфраструктурой на базе широко известных компонентов с открытым исходным кодом.

В основе системы лежит служба каталога FreeIPA, которая централизованно хранит информацию о пользователях, компьютерах и сервисных учётных записях. Доступ к этим данным осуществляется через открытые протоколы.

Система ALD Pro отвечает за работу службы NTP в домене и расширяет возможности службы каталога FreeIPA благодаря интеграции с системой конфигурирования SaltStack. Эта интеграция позволяет централизованно настраивать рабочие станции в домене, применяя концепцию групповых политик.

Мы уже разобрали работу служб NTP, DNS и портала управления ALD Pro. Сегодня мы продолжим знакомство с архитектурой и рассмотрим взаимосвязь таких компонентов, как 389 Directory Server, MIT Kerberos, Samba. Общая архитектура решения представлена на рис. 2.83.

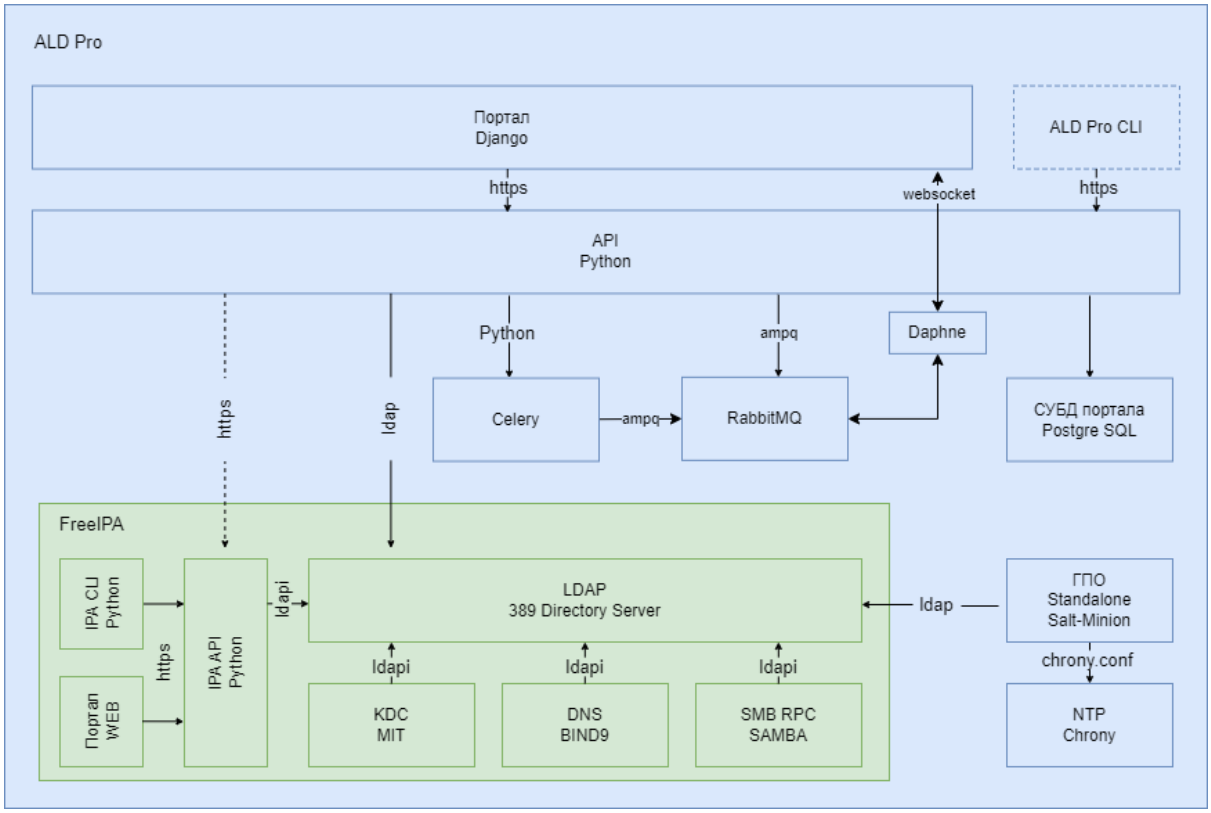


рис. 2.83 – Архитектура службы каталога

Ключевым компонентом FreeIPA является LDAP-каталог 389 Directory Server, который представляет собой специализированную нереляционную систему хранения и управления данными. Чтобы интегрировать компоненты системы с LDAP-каталогом разработчики FreeIPA создали порядка 15 специальных плагинов, например, для управления топологией, динамического назначения номеров идентификаторов безопасности и др.

Структура LDAP-каталога подобна файловой системе, где папкам соответствуют записи, а файлам – атрибуты записей. Подробнее об этом мы расскажем в одном из следующих разделов.

За работу каталога могут отвечать десятки и даже сотни серверов, которые объединяются в кластер, обеспечивающий устойчивость к разделению (*англ. Partition Tolerance*) и согласованность данных в конечном итоге (*анг. Eventually Consistent*) по теореме CAP.

Внешние клиенты, например, агент групповых политик Standalone Salt-minion, взаимодействуют с контроллерами домена по протоколу LDAP на портах 389/TCP и 636/TCP (LDAP + TLS). Локальные компоненты, такие как KDC, DNS, Samba, обращаются к базе LDAP через специальные плагины по LDAPI, используя unix-сокет (англ. LDAP over IPC, LDAPI), что повышает безопасность и сокращает накладные расходы на сетевой стек.

4.3 Сервер FreeIPA

Название службы каталога FreeIPA является акронимом от Free Identity, Policy and Audit, что указывает на основные приоритеты в развитии продукта — централизованное управление идентификацией, политиками и аудитом.

Также как в 90-е годы Microsoft заимствовала из UNIX-мира технологии DNS, Kerberos, LDAP для построения своего Windows-домена, команда разработчиков проекта FreeIPA ставит сейчас своей целью заимствовать у Microsoft лучшие бизнес-практики для построения Linux-домена. Именно по этой причине система унаследовала от Active Directory многие концепции, такие как:

- **Домен** — совокупность пользователей, компьютеров и других сетевых объектов, информация о которых хранится в общей базе данных.
- **Контроллер домена** — сервер, обеспечивающий работу распределенной базы данных домена (службы каталога) и предоставляющий всем участникам сервисы аутентификации и авторизации.
- **Доменный пользователь** — пользователь, которому создана учетная запись в домене, с помощью которой он может проходить аутентификацию на контроллерах домена и получать авторизованный доступ при обращении к доменным сервисам.
- **Доменный компьютер** — компьютер, которому создана учетная запись в домене, с помощью которой он может получать информацию из LDAP-каталога и проверять аутентичность пользователей по протоколу Kerberos V5.
- **Керберизированный сервис** — сетевое приложение, которому создана учетная запись в домене, которую он может использовать как компьютер для получения информации из LDAP-каталога и проверки аутентичности пользователей по протоколу Kerberos V5.

Вместе с тем некоторые функции реализованы иначе, чем в AD, или даже не планируются к реализации. Например, в части централизованного управления рабочими станциями FreeIPA предлагает хороший базовый набор политик безопасности, но все остальные функции предлагает решать с помощью отдельных систем конфигурирования. Поэтому в продукте ALD Pro эта потребность закрыта интеграцией с одной из лучших систем управления конфигурациями SaltStack.

В состав компонентов FreeIPA входят:

- Сервер LDAP на базе 389 Directory Server.
- Служба для управления DNS на базе BIND9.
- Единая точка входа по протоколу Kerberos V5 на базе MIT Kerberos KDC.
- Дополнительно с помощью Samba AD может быть активирована интеграция с Active Directory (в ALD Pro компонент устанавливается по умолчанию).
- Дополнительно с помощью DogTag может быть реализована работа с сертификатами (в ALD Pro планируется реализация внешних плагинов к PKI Proxy для интеграции со сторонними удостоверяющими центрами).

4.3.1 Управление сервером FreeIPA

Для управления службой каталога можно использовать различные инструменты, которые взаимодействуют с ней через программные интерфейсы JSON-RPC или XML-RPC, которыми она оборачивает свои компоненты, или напрямую через LDAP. Основными клиентскими интерфейсами являются:

- Веб-портал, который доступен по адресу `https://<fqdn имя контроллера>/ipa/ui/` и работает через JSON-RPC.

  Вы можете обращаться к этому инструменту для настройки тех функций, которые не заявлены в продукте ALD Pro, но поддерживаются компонентом. Например, для настройки правил автоматического монтирования сетевых дисков.

- Утилита `ipactl`, используется для управления службами FreeIPA, их корректного включения, выключения и перезапуска.
- Утилита командной строки `ipa`, которая установлена на всех доменных компьютерах и работает через JSON-RPC. Утилита имеет собственный механизм автоматического обнаружения контроллеров и поддерживает прозрачную Kerberos-аутентификацию.
- Дополнительные утилиты, такие как `ipa-backup`, `ipa-restore`, `ipa-getkeytab` и др.

4.3.1.1 Веб-интерфейс управления

Мы не будем подробно останавливаться на веб-интерфейсе и скажем только, что он есть, и вы можете получить к нему доступ, например, по адресу https://dc-1.ald.company.lan/ipa/ui.

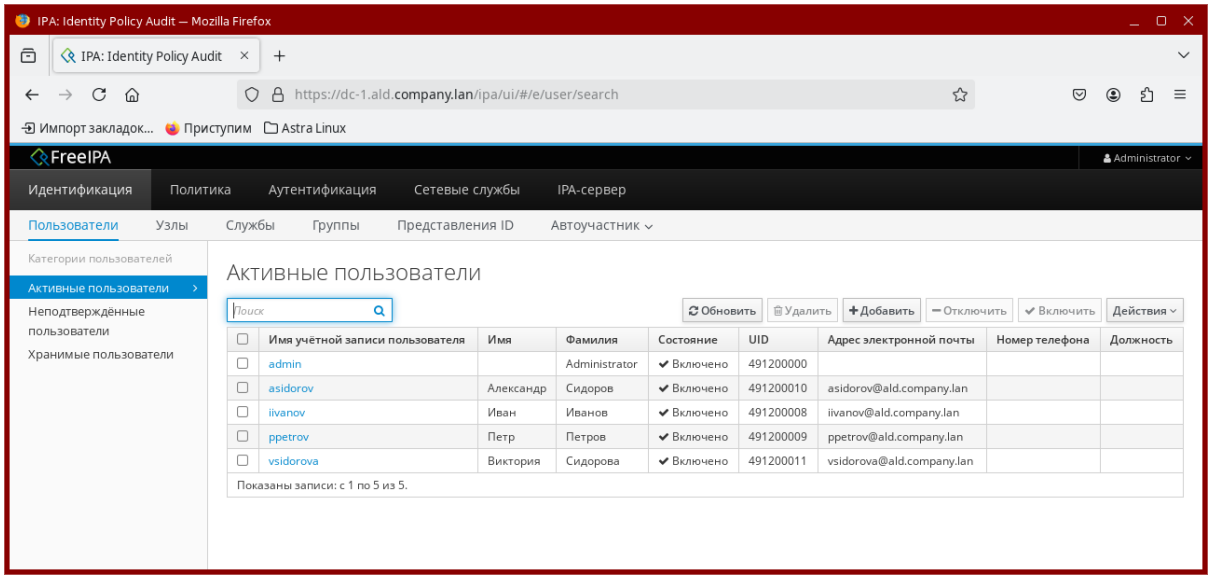


рис. 2.84 – Интерфейс веб-портала FreeIPA

Для входа на портал по умолчанию используется прозрачная Kerberos-аутентификация, поэтому, если у вас в связке ключей есть действующий TGT-билет, то вы попадете в систему без необходимости ввода пароля.

В горизонтальном меню представлено пять вкладок:

- **Идентификация** – управление объектами каталога (пользователи, узлы, службы, группы).
- **Политика** – политики паролей, билетов Kerberos, HBAC-правила, SUDO-правила, Parsec (интеграция с функциями мандатного контроля от Astra Linux).
- **Аутентификация** – управление аутентификацией через серверы RADIUS, одноразовые ключи и сертификаты.
- **Сетевые службы** – управление службой DNS и автомонтированием папок.
- **IPA-сервер** – управление топологией репликации, настройка ролей, диапазонов идентификаторов и др. параметров.

4.3.1.2 Утилита ipactl

Все компоненты контроллера домена представляют собой службы, которыми можно управлять через утилиту `systemctl`. Например, вы можете перезапустить веб-сервер командой `systemctl restart apache2.service`. Но, учитывая, что запуск и выключение служб требуется выполнять в определенном порядке, была разработана специальная утилита `ipactl`, которая позволяет управлять всеми этими службами сразу.

Основные команды утилиты:

- Команда `ipactl start` – запуск.
- Команда `ipactl stop` – остановка.
- Команда `ipactl restart` – перезапуск.
- Команда `ipactl status` – вывод статуса.

Для выполнения команд нужно обладать привилегиями суперпользователя, поэтому добавляйте к ним sudo, например:

```
sudo ipactl status
```

Результат выполнения команды:

```
admin@dc-1:~$ sudo ipactl status
Directory Service: RUNNING
krb5kdc Service: RUNNING
kadmin Service: RUNNING
named Service: RUNNING
httpd Service: RUNNING
ipa-custodia Service: RUNNING
smb Service: RUNNING
winbind Service: RUNNING
ipa-otpd Service: RUNNING
ipa-dnskeysyncd Service: RUNNING
ipa: INFO: The ipactl command was successful
```

4.3.1.3 Утилита ipa

Утилита `ipa` написана на языке Python и входит в состав пакета freeipa-client, который устанавливается на всех компьютерах в домене. Для ее использования ничего дополнительно устанавливать не требуется.

Для взаимодействия с сервером утилита отправляет запросы по протоколу JSON-RPC. Аутентификация выполняется по протоколу Kerberos V5 с использованием билетов из связки ключей текущего пользователя.

Чтобы проверить, от имени какого пользователя выполняются запросы на сервере, можно использовать команду `ipa user-find` с параметром `--whoami`. После выполнения запроса в связке ключей должен появиться билет на доступ к HTTP-службе того контроллера, через который был обработан запрос.

```
ipa user-find --whoami
```

Результат выполнения команды:

```
admin@dc-1:~$ ipa user-find --whoami
--------------------
найден 1 пользователь
--------------------
  Имя учётной записи пользователя: admin
  Фамилия: Administrator
  Домашний каталог: /home/admin
  Оболочка входа: /bin/bash
  Псевдоним учётной записи: admin@ALD.COMPANY.LAN, root@ALD.COMPANY.LAN
  UID: 491200000
  ID группы: 491200000
  Учётная запись отключена: False
---------------------------------
Количество возвращённых записей 1
---------------------------------
```

Параметры для бэкенда и для клиентской части определяются в общем конфигурационном файле `/etc/ipa/default.conf`. На контроллере домена содержимое файла имеет следующий вид:

```
admin@dc-1:~$ cat /etc/ipa/default.conf
[global]
host = dc-1.ald.company.lan
basedn = dc=ald,dc=company,dc=lan
realm = ALD.COMPANY.LAN
domain = ald.company.lan
xmlrpc_uri = https://dc-1.ald.company.lan/ipa/xml
ldap_uri = ldapi://%2Frun%2Fslapd-ALD-COMPANY-LAN.socket
mode = production
enable_ra = False
ra_plugin = None
```

На обычном доменном компьютере этот файл выглядит немного иначе:

```
admin@pc-1:~$ cat /etc/ipa/default.conf
[global]
basedn = dc=ald,dc=company,dc=lan
realm = ALD.COMPANY.LAN
domain = ald.company.lan
server = dc-1.ald.company.lan
host = pc-1.ald.company.lan
xmlrpc_uri = https://dc-1.ald.company.lan/ipa/xml
enable_ra = False
```

Параметры, актуальные больше для клиента:

- **xmlrpc_uri** — адрес сервера FreeIPA, к которому должен подключаться клиент для взаимодействия с доменом. По умолчанию в параметрах `xmlrpc_uri` и `server` будет указан адрес того контроллера, через который компьютер был присоединен к домену.

  Этот адрес используется как утилитой `ipa`, так и дополнительными инструментами, например, утилитой `ipa-getcert` и др. Обратите также внимание, что современные версии клиентских приложений используют JSON-RPC, поэтому заменяют в адресе значение `/ipa/xml` на

`/ipa/json`.

Если параметр `xmlrpc_uri` не задан, то клиент попытается сформировать URI-адрес, используя значение параметра `server`.

- **server** — полное имя (FQDN) сервера FreeIPA для взаимодействия клиента через XML-RPC. Этот параметр считается устаревшим и используется, только если не задан параметр `xmlrpc_uri`.

- **domain** — полное имя домена для автоматического обнаружения активного контроллера по SRV-записям вида `_ldap._tcp.ald.company.lan`. Клиент переходит на механизм автоматического обнаружения сервисов, только если параметры `xmlrpc_uri` и `server` не будут заданы явным образом.

  Если параметр `domain` не задан, то клиент попытается взять вместо него значение параметра `realm` в нижнем регистре символов.

- **enable_ra** — указывает, действует ли центр сертификации (англ. Certification Authority, CA) в качестве агента центра регистрации (англ. Registration Authority), как в случае интеграции с DogTag. В продукте ALD Pro центр сертификации DogTag не используется, поэтому параметр имеет значение `False`.

Параметры, актуальные больше для сервера:

- **host** — имя хоста, которое используется при формировании URL-адресов.

- **realm** — область Kerberos, которая используется, например, для формирования имени службы `dirsrv@ALD-COMPANY-LAN`. Параметр критичен для сервера, а на клиенте значение `realm` может быть использовано вместо `domain`, если этот параметр не будет задан явным образом.

- **ldap_uri** — адрес LDAP-сервера, который используется бэкендом API, поэтому данная настройка задается только на контроллерах домена. По умолчанию бэкенд использует протокол LDAPI для обращения к серверу через UNIX-сокет.

- **basedn** — отличительное имя (DN) базовой записи для поиска записей в LDAP-каталоге. Если параметр не задан, то `basedn` формируется по значению параметра `domain`. Параметр критичен для сервера, а на клиенте используется, например, утилитой `ipa-client-automount`.

- **mode** — определяет режим работы сервера, поддерживаются значения `production` и `development`. В продуктивном режиме, например, для повышения производительности не выполняются некоторые дополнительные самопроверки.

- **ra_plugin** — определяет название плагина для взаимодействия бэкенда с центром регистрации

(англ. Registration Authority), как в случае интеграции с DogTag. В продукте ALD Pro центр сертификации DogTag не используется, поэтому параметр имеет значение `None`.

Как вы могли заметить, для обеспечения быстрой работы и высокой отказоустойчивости утилита `ipa` использует довольно сложный порядок выбора активного сервера FreeIPA:

1. В первую очередь утилита `ipa` проверяет доступность того сервера, который указан в параметре `xmlrpc_uri`.

2. Если параметр `xmlrpc_uri` не задан, то утилита формирует URI по значению параметра `server` и проверяет доступность API по этому адресу.

3. Если параметр `server` не задан, то утилита выполняет автоматическое обнаружение сервера через DNS по SRV-записям вида `_ldap._tcp.<полное имя домена>`, используя в качестве полного имени домена значение параметра `domain`.

4. Если параметр `domain` не задан, то вместо него утилита попробует использовать значение `realm` в нижнем регистре символов.

Доступность сервера FreeIPA можно проверить командой `ping`. С помощью ключа `-d` можно запросить вывод дополнительной отладочной информации:

```
ipa -d ping
```

Результат выполнения команды:

```
admin@dc-1:~$ ipa -d ping
...
ipa: DEBUG: importing plugin module ipaclient.plugins.vault
ipa: DEBUG: failed to find session_cookie in persistent storage for principal
↪'admin@ALD.COMPANY.LAN'
ipa: DEBUG: trying https://dc-1.ald.company.lan/ipa/json
ipa: DEBUG: New HTTP connection (dc-1.ald.company.lan)
ipa: DEBUG: received Set-Cookie (<class 'list'>)'['ipa_session=MagBearerToken=9Q6FI..
↪.ZMCHCKsExQ%3d%3d;path=/ipa;httponly;secure;']'
ipa: DEBUG: storing cookie 'ipa_session=MagBearerToken=9Q6FINpIhPV...MCHCKsExQ%3d%3d;
↪' for principal admin@ALD.COMPANY.LAN
ipa: DEBUG: Created connection context.rpcclient_140561608787896
ipa: DEBUG: raw: ping(version='2.239')
ipa: DEBUG: ping(version='2.239')
ipa: DEBUG: [try 1]: Forwarding 'ping/1' to json server 'https://dc-1.ald.company.
↪lan/ipa/json'
ipa: DEBUG: HTTP connection keep-alive (dc-1.ald.company.lan)
ipa: DEBUG: received Set-Cookie (<class 'list'>)'['ipa_
↪session=MagBearerToken=0r2cXXlnOKMtVYcu....%3d%3d;path=/ipa;httponly;secure;']'
ipa: DEBUG: storing cookie 'ipa_session=MagBearerToken=0r2...Q%3d%3d;' for principal␣
↪admin@ALD.COMPANY.LAN
ipa: DEBUG: Destroyed connection context.rpcclient_140561608787896
--------------------------------------------
IPA server version 4.8.10. API version 2.239
--------------------------------------------
```

Прокомментируем некоторые строки:

- **ipa: DEBUG** — это отладочные данные, из которых можно извлечь много полезной информации о порядке обработки запроса;
- **ipa: DEBUG: trying https://dc-1.ald.company.lan/ipa/json** — указывает, на какой URL-адрес был направлен текущий запрос;
- **ipa: DEBUG: raw: ping(version="2.239")** — указывает используемую версию API и выполняемую команду.

Для получения расширенной информации по командам утилиты и их параметрам можно воспользоваться командой `ipa help topics`. Для выхода из справки нажмите клавишу `q`, как при работе с утилитой `less`.

```
ipa help topics
```

Результат выполнения команды:

```
admin@dc-1:~$ ipa help topics
dnarange                        dnarange
dns                             Система доменных имён (DNS)
domainlevel                     Повысить уровень домена IPA.
group                           Groups of users
hbac                            Команды управления доступом на основе узла
hbactest                        Simulate use of Host-based access controls
host                            Узлы/Компьютеры
hostgroup                       Groups of hosts.
idrange                         Диапазоны идентификаторов
...
```

Чтобы просмотреть справку по управлению пользователями, введите команду:

```
ipa help user
```

Результат выполнения команды:

```
admin@dc-1:~$ ipa help user
Пользователи

Управление записями пользователей. Все пользователи являются POSIX-пользователями.

IPA поддерживает широкий спектр форматов имён пользователей, но следует учитывать те␣
↪ограничения, которые могут существовать в конкретной среде.
...
```

Утилита IPA имеет большое количество команд. В рамках модуля мы рассмотрим только управление пользователями, см. табл. 2.3, но вы можете обратиться также к более полному курсу по автоматизации средствами утилиты ipa.

табл. 2.3 – Список команд утилиты IPA для управления пользователями

Команда	Выполняемое действие
`ipa user-add`	Добавить нового пользователя. Если вы хотите добавить нового «неподтвержденного» пользователя, воспользуйтесь командой `ipa stageuser-add`.
`ipa user-add-manager`	Отметить в учетной записи, кто является руководителем этого сотрудника.
`ipa user-add-principal`	Добавить новый псевдоним Kerberos к учетной записи пользователя.
`ipa user-del`	Удалить пользователя. По умолчанию используется `--no-preserve`, поэтому запись удаляется безвозвратно. Если использовать ключ `--preserve`, то записи будут перемещены в контейнер `cn=deleted users,cn=accounts,cn=provisioning` — в ALD Pro называется «Корзина», а в интерфейсе FreeIPA — «Хранимые пользователи». Если учетная запись уже находится в корзине, она может быть окончательно удалена повторным выполнением команды `ipa user-del` без ключа `--preserve`.
`ipa user-disable`	Деактивировать учетную запись пользователя.
`ipa user-enable`	Активировать учетную запись пользователя.
`ipa user-find`	Выполнить поиск учетных записей пользователей.
`ipa user-mod`	Изменить учетную запись пользователя.
`ipa user-remove-manager`	Удалить отметку о руководителе сотрудника.
`ipa user-remove-principal`	Удалить псевдоним Kerberos учетной записи.
`ipa user-show`	Показать сведения о пользователе.
`ipa user-stage`	Переместить удаленную запись в список неподтвержденных пользователей.
`ipa user-status`	Состояние блокировки учетной записи пользователя.
`ipa user-undel`	Восстановить удаленную учетную запись пользователя из корзины.
`ipa user-unlock`	Разблокировать учетную запись пользователя.
`ipa passwd`	Установить новый пароль.

Поиск пользователей (ipa user-find)

Для поиска пользователей в каталоге используется команда `ipa user-find`, которая выведет список записей по заданным критериям. Синтаксис команды:

```
ipa [global-options] user-find [CRITERIA] [options]
```

Где:

- global-options — глобальные параметры, описание которых можно просмотреть командой ipa -h, например, ключ -d позволяет включить вывод сообщений для отладки.
- CRITERIA — критерий поиска по атрибутам uid, givenName, sn, telephoneNumber, ou, title. Допускает указывать не только полное значение, но и часть подстроки.
- options — дополнительные параметры фильтрации, см. табл. 2.4.

табл. 2.4 – Основные критерии поиска пользователей

Команда	Выполняемое действие
--login=STR	Поиск по полю «Логин», только точное совпадение
--first=STR	Поиск по полю «Имя», только точное совпадение
--last=STR	Поиск по полю «Фамилия», только точное совпадение
--cn=STR	Поиск по полю «Полное имя» (common name), только точное совпадение
--displayname=STR	Поиск по полю «Отображаемое имя», только точное совпадение
--middlename=STR	Поиск по полю «Отчество», только точное совпадение
--initials=STR	Поиск по полю «Инициалы», только точное совпадение
--homedir=STR	Поиск по полю «Домашний каталог», только точное совпадение
--gecos=STR	Поиск по полю GECOS, только точное совпадение
--shell=STR	Поиск по полю «Оболочка входа», только точное совпадение
--principal=PRINCIPAL	Поиск по полю «Псевдоним учётной записи», только точное совпадение
--principal-expiration	Не работает в версии 4.8.10, поэтому для поиска используйте ldapsearch
--password-expiration	Не работает в версии 4.8.10, поэтому для поиска используйте ldapsearch
--email=STR	Поиск по полю «Адрес электронной почты», только точное совпадение
--uid=INT	Поиск по полю «ID пользователя», только точное совпадение
--gidnumber=INT	Поиск по полю «ID группы», только точное совпадение
--street=STR	Поиск по полю «Адрес», только точное совпадение
--city=STR	Поиск по полю «Город», только точное совпадение
--state=STR	Поиск по полю «Область/республика», только точное совпадение
--postalcode=STR	Поиск по полю «Индекс», только точное совпадение
--phone=STR	Поиск по полю «Номер телефона», только точное совпадение

continues on next page

табл. 2.4 – продолжение с предыдущей страницы

Команда	Выполняемое действие
--mobile=STR	Поиск по полю «Номер мобильного телефона», только точное совпадение
--pager=STR	Поиск по полю «Номер пейджера», только точное совпадение
--fax=STR	Поиск по полю «Номер факса», только точное совпадение
--orgunit=STR	Поиск по полю «Отдел», только точное совпадение
--title=STR	Поиск по полю «Должность», только точное совпадение
--manager=STR	Поиск по полю «Руководитель», только точное совпадение
--carlicense=STR	Поиск по полю «Номер автомобиля», только точное совпадение
--class=STR	Поиск по полю «Категория пользователей», только точное совпадение
--departmentnumber=STR	Поиск по полю «Номер отдела», только точное совпадение
--employeenumber=STR	Поиск по полю «Номер сотрудника», только точное совпадение
--employeetype=STR	Поиск по полю «Тип сотрудника», только точное совпадение
--country=STR	Поиск по полю «Страна», только точное совпадение
--department=DNPARAM	Поиск по полю «Подразделение»
--disabled=BOOL	Поиск «деактивированных» учетных записей
--preserved=BOOL	Поиск пользователей в корзине
--timelimit=INT	Ограничение времени поиска в секундах («0» — отсутствие ограничения)
--sizelimit=INT	Максимальное количество возвращенных записей («0» — отсутствие ограничения)
--whoami	Показать запись пользователя для текущей учетной записи Kerberos
--all	Получить и вывести все атрибуты, возвращаемые сервером
--raw	Вывести записи в том виде, в котором они хранятся на сервере
--pkey-only	Вывести только первичный ключ login
--in-groups=STR	Поиск пользователей, находящихся в искомой группе
--not-in-groups=STR	Поиск пользователей, не находящихся в искомой группе
--in-roles=STR	Поиск пользователей, имеющих искомую роль
--not-in-roles=STR	Поиск пользователей, не имеющих искомую роль

Запросы к серверу можно отследить в журнале доступа командой `sudo tail -f /var/log/dirsrv/slapd-ALD-COMPANY-LAN/access`, см. рис. 2.85.

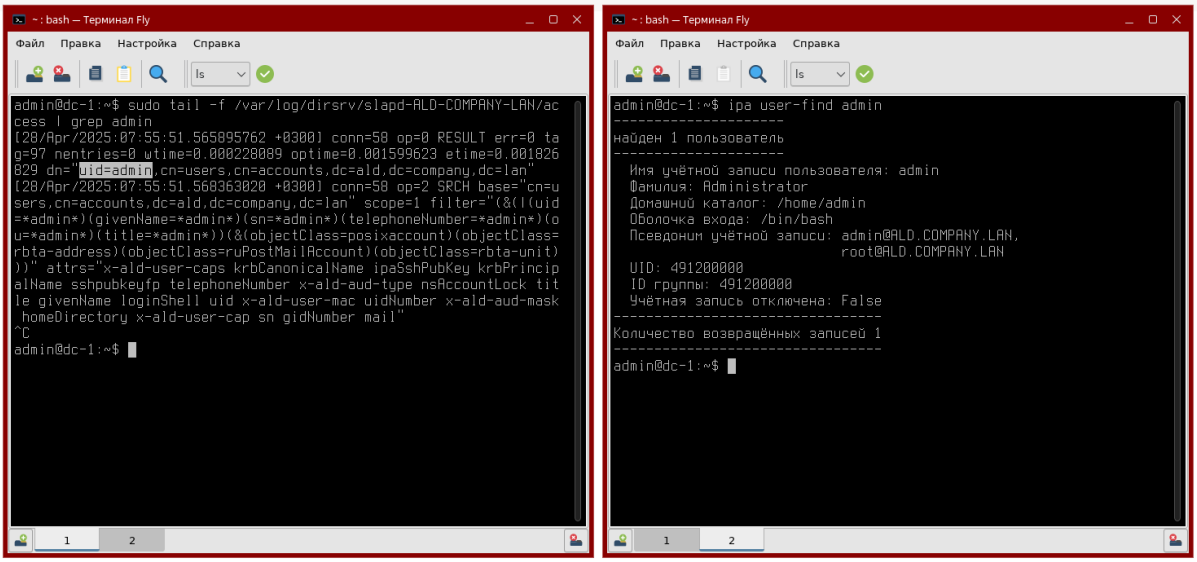


рис. 2.85 – Проверка запроса на поиск в LDAP-каталоге по журналу доступа

Давайте запросим информацию о пользователе с логином admin. Чтобы можно было использовать вывод утилиты `ipa`, в скриптах автоматизации можно добавить ключ `--raw`. В этом случае вы получите данные в формате LDIF (*от англ. LDAP, Data Interchange Format — формат представления записей службы каталога или их изменений в текстовой форме*):

```
admin@dc-1:~$ ipa user-find --all --raw --login admin
---------------------
найден 1 пользователь
---------------------
  dn: uid=admin,cn=users,cn=accounts,dc=ald,dc=company,dc=lan
  uid: admin
  sn: Administrator
  cn: Administrator
  homedirectory: /home/admin
  gecos: Administrator
  loginshell: /bin/bash
  krbprincipalname: admin@ALD.COMPANY.LAN
  krbprincipalname: root@ALD.COMPANY.LAN
  uidnumber: 296400000
  gidnumber: 296400000
  x-ald-user-mic-level: 63
  nsaccountlock: FALSE
  rbtadp: ou=ald.company.lan,cn=orgunits,cn=accounts,dc=ald,dc=company,dc=lan
  rbtaou: ald.company.lan
  ipaNTSecurityIdentifier: S-1-5-21-2487214781-2721300798-1427603078-500
  ...
----------------------------------
Количество возвращённых записей 1
----------------------------------
```

Удалить поясняющие записи и лишние пробелы можно командами `grep` и `sed`:

```
ipa user-find --all --raw --login admin \
```

(continues on next page)

(продолжение с предыдущей страницы)

```
 | grep -E '(^\s\s|^$)' \
 | sed 's/^\s\s//g' \
 | grep -iE '(^dn|^sn|^uid|nsaccountlock|^$)'
```

Где:

- `grep -E '(^\s\s|^$)'` — выбирает строки, начинающиеся с двух пробелов (\s символ пробела в регулярных выражениях) или пустые строки (^$).
- `sed 's/^\s\s//g'` — заменяет двойной пробел в начале строки на пустой символ.
- `grep -iE '(^dn|^sn|^uid|nsaccountlock|^$)` — выбирает несколько полей с помощью логического утверждения `dn ИЛИ sn ИЛИ uid ИЛИ nsaccountlock`. Обратите внимание, что в конце также выбираются и пустые строки `^$`. Дело в том, что они нужны для разделения нескольких записей согласно спецификации формата LDIF.

Результат выполнения команды:

```
dn: uid=admin,cn=users,cn=accounts,dc=ald,dc=company,dc=lan
uid: admin
sn: Administrator
uidnumber: 491200000
nsaccountlock: FALSE
```

Чтобы не допустить высокой нагрузки на сервер, бэкенд IPA возвращает не более 100 записей, а на обработку запросов должно уходить не более 2 секунд. Если ваш запрос не уложится в эти ограничения, то вы получите ошибку. Текущие лимиты можно посмотреть с помощью команды `ipa config-show`:

```
admin@dc-1:~$ ipa config-show
  Максимальная длина имени пользователя: 32
  Максимальная длина имени хоста: 64
  Основа домашних каталогов: /home
  Оболочка по умолчанию: /bin/bash
  Группа пользователей по умолчанию: ipausers
  Почтовый домен по умолчанию: ald.company.lan
  Ограничение времени поиска: 2
  Ограничение размера поиска: 100
  ...
```

Изменить конфигурацию можно с портала FreeIPA или из командной строки:

```
admin@dc-1:~$ ipa config-mod --searchrecordslimit=500 --searchtimelimit=5
  Максимальная длина имени пользователя: 32
  Максимальная длина имени хоста: 64
  Основа домашних каталогов: /home
  Оболочка по умолчанию: /bin/bash
  Группа пользователей по умолчанию: ipausers
  Почтовый домен по умолчанию: ald.company.lan
```

(continues on next page)

(продолжение с предыдущей страницы)

```
  Ограничение времени поиска: 5
  Ограничение размера поиска: 500
  ...
```

Детальный просмотр пользователя (ipa user-show)

Чтобы получить подробную информацию о пользователе, воспользуйтесь командой `ipa user-show`. Например, чтобы получить информацию о пользователе `petrov.av` выполните следующую команду:

```
admin@dc-1:~$ ipa user-show petrov.av
  Имя учётной записи пользователя: petrov.av
  Имя: Александр
  Фамилия: Петров
  Домашний каталог: /home/petrov.av
  Оболочка входа: /bin/bash
  Имя учётной записи: petrov.av@ALD.COMPANY.LAN
  Псевдоним учётной записи: petrov.av@ALD.COMPANY.LAN
  Адрес электронной почты: petrov.av@ald.company.lan
  UID: 491200007
  ID группы: 491200007
  Учётная запись отключена: False
  Link to department: ou=ald.company.lan,cn=orgunits,cn=accounts,dc=ald,dc=company,
↪dc=lan
  Link to head department: ald.company.lan
  Пароль: True
  Участник групп: ipausers
  Indirect Member of role: ALDPRO - Organization units
  Доступные ключи Kerberos: True
```

Выполняя команду `ipa user-show` с ключами `--rights` и `--all`, вы можете увидеть свои права доступа (effective rights) на изменение атрибутов выбранного пользователя. Однако примите во внимание, что параметр `--all` в этом случае обязателен, иначе вывод будет аналогичен команде `ipa user-show` без ключей.

```
ipa user-show iivanov --rights --all
```

Результат команды:

```
  admin@dc-1:~$ ipa user-show iivanov --rights --all
    dn: uid=iivanov,cn=users,cn=accounts,dc=ald,dc=company,dc=lan
    Имя учётной записи пользователя: iivanov
    Имя: Иван
    Фамилия: Иванов
    ...
    Indirect Member of role: ALDPRO - Organization units
    Доступные ключи Kerberos: True
    attributelevelrights: {'objectclass': 'rscwo', 'aci': 'rscwo', 'sn': 'rscwo', ...
↪'ipanthomedirectorydrive': 'rscwo', 'nsaccountlock': 'rscwo'}
    ipantsecurityidentifier: S-1-5-21-3838112765-3762782768-2146355126-1008
```

(continues on next page)

(продолжение с предыдущей страницы)

```
    ipauniqueid: 904f6e98-2caf-11f0-bbdb-08002725086e
    ...
```

Права пользователя отображены в области attributelevelrights. Описание возможных значений вы найдете в табл. 2.5

табл. 2.5 – Права на атрибуты для выбранного пользователя

Разрешение	Описание команды
r	Право на чтение атрибута для пользователя, запустившего команду
s	Право на поиск по атрибуту для пользователя, запустившего команду
w	Право записи в атрибут для пользователя, запустившего команду
o	Право на удаление атрибута для пользователя, запустившего команду
c	Право на сравнение атрибутов для пользователя, запустившего команду
W	Право на запись для целевого пользователя, атрибуты которого просматриваются
O	Право на удаление для целевого пользователя, атрибуты которого просматриваются

Создание пользователя (ipa user-add)

Для создания пользователя используется команда `ipa user-add` с параметрами, см. табл. 2.6. Все доступные команды можно просмотреть в справке, вызвав команду:

```
admin@dc-1:~$ ipa help user-add
```

табл. 2.6 – Список атрибутов для добавления пользователя

Атрибут	Описание
`--first=STR`	Имя пользователя. По умолчанию не задано.
`--last=STR`	Фамилия пользователя. По умолчанию не задана.
`--cn=STR`	Общее, распространенное имя. Если параметр не задан, автоматически формируется из имени и фамилии пользователя, например, Петр Иванов.

continues on next page

табл. 2.6 – продолжение с предыдущей страницы

Атрибут	Описание
--displayname=STR	Отображаемое имя. Если не задано, создается из имени и фамилии пользователя, например, Петр Иванов.
--initials=STR	Инициалы. Если не задано, определяется автоматически из имени и фамилии, например, П.И.
--homedir=STR	Домашняя директория, по умолчанию /home/<user_login>, например, /home/ivanov.
--gecos=STR	Дополнительная информация о пользователе, которая может содержать номер телефона, домашний адрес, факс и др.
--shell=STR	Оболочка пользователя. Значение по умолчанию /bin/bash задается глобально в настройках контроллера домена.
--principal=PRINCIPAL	Уникальное имя пользователя, которое используется при аутентификации в домене. По умолчанию имеет значение user_login@{DOMAIN_SUFFIX}. Например, ivanov@ALD.COMPANY.LAN.
--principal-expiration=DATETIME	Окончание срока действия Kerberos-записи. Задается в формате GeneralizedTime ASN.1.
--password-expiration=DATETIME	Окончание срока действия Kerberos-пароля. Задается в том же формате, что и principal-expiration. Рассчитывается автоматически в момент изменения пароля в соответствии с действующей политикой паролей. По умолчанию срок действия паролей составляет 90 дней.
--email=STR	Электронная почта. По умолчанию равна user_login@{domain_suffix}. Например, ivanov.p@ald.company.lan.
--password	Запросить пароль пользователя в момент создания учетной записи.
--random	Сгенерировать пароль автоматически. Требования к паролю задаются политикой паролей контроллера домена. Пароль будет выведен после успешного выполнения команды по созданию пользователя.

continues on next page

табл. 2.6 – продолжение с предыдущей страницы

Атрибут	Описание
--uid=INT	Назначить пользователю конкретный номер uidNumber. По умолчанию генерируется автоматически DNA-плагином.
--gidnumber=INT	Назначить пользователю конкретный номер первичной группы GidNumber. По умолчанию идентификатор первичной группы совпадает с идентификатором пользователя gidNumber=uidNumber.
--street=STR	Улица. По умолчанию значение не задано.
--city=STR	Город. По умолчанию значение не задано.
--state=STR	Область. По умолчанию значение не задано.
--postalcode=STR	Почтовый индекс. По умолчанию значение не задано.
--phone=STR	Рабочий телефон. По умолчанию значение не задано.
--mobile=STR	Мобильный телефон. По умолчанию значение не задано.
--pager=STR	Номер пейджера. По умолчанию значение не задано.
--fax=STR	Номер факса. По умолчанию значение не задано.
--orgunit=STR	Отдел. По умолчанию значение не задано.
--title=STR	Должность. По умолчанию значение не задано.
--manager=STR	Руководитель. По умолчанию значение не задано. Параметр должен совпадать с логином существующей учетной записи другого пользователя.
--carlicense=STR	Номер водительского удостоверения. По умолчанию значение не задано.
--sshpubkey=STR	Публичный ключ ssh. Для подключения к хостам по ssh-ключу. По умолчанию значение не задано.
--user-auth-type=['password', 'radius', 'otp', 'pkinit', 'hardened']	Тип авторизации пользователя. По умолчанию — по паролю (password).
--class=STR	Категория пользователя. Это поле может использоваться для автоматического назначения пользователю групп через механизм автоучастия (плагин automembership). По умолчанию значение не задано.

continues on next page

табл. 2.6 – продолжение с предыдущей страницы

Атрибут	Описание
--radius=STR	Прокси-RADIUS-сервер. Для авторизации по RADIUS. По умолчанию значение не задано.
--radius-username=STR	Имя пользователя RADIUS. По умолчанию не задано.
--departmentnumber=STR	Служебная информация о департаменте. По умолчанию не задана.
--employeenumber=STR	Служебная информация о пользователе. По умолчанию не задана.
--employeetype=STR	Служебная информация о пользователе. По умолчанию не задана.
--preferredlanguage=STR	Предпочитаемый язык. Дополнительная информация о пользователе. По умолчанию значение не задано.
--certificate=CERTIFICATE	Сертификат пользователя для двухфакторной аутентификации. По умолчанию значение не задано.
--macmin=STR	Целое число для назначения минимального уровня конфиденциальности. По умолчанию равно 0.
--macmax=STR	Целое число для назначения максимального уровня конфиденциальности. По умолчанию равно 0.
--miclevel=INT	Целое число, задающее маску уровней целостности. По умолчанию равно 0.
--middlename=STR	Отчество пользователя. По умолчанию значение не задано.
--photo=BYTES	Фотография пользователя, можно загрузить через портал управления. По умолчанию значение не задано.
--country=STR	Код страны по стандарту ISO 3166, например RU. По умолчанию значение не задано.
--proxyaddresses=STR	Полный список адресов пользователя. Может содержать SMTP-, SIP-, X.500-адреса. Например, SMTP: ivanov@ald.company.lan. По умолчанию значение не задано.

continues on next page

табл. 2.6 – продолжение с предыдущей страницы

Атрибут	Описание
`--department=DNPARAM`	Задает подразделение, к которому прикреплен пользователь. По умолчанию значение будет равно ou={domain_sufix},cn=orgunits,cn=accounts,{root_suffix}, например, ou=ald.company.lan,cn=orgunits,cn=accounts,dc=ald,dc=co
`--setattr=STR`	Присвоить значение атрибуту пользователя в формате «атрибут=значение». Если атрибут многозначный, он будет перезаписан.
`--addattr=STR`	Добавить атрибут для пользователя в формате «атрибут=значение».
`--noprivate`	Не создавать первичную группу пользователя. По умолчанию при создании пользователя создается одноименная группа, gidNumber которой равен uidNumber пользователя. POSIX-пользователь не может быть создан без POSIX-группы.
`--all`	Показать все атрибуты пользователя. Используется при выводе данных.
`--raw`	Показать всю информацию о пользователе. Используется при выводе данных.
`--no-members`	Не показывать группы, в которых состоит пользователь (атрибуты memberOf).

Уточним также:

- **{root_suffix}** — это корневой суффикс LDAP-каталога, состоит из нескольких компонентов доменного имени (Domain Component, DC), например, `dc=ald,dc=company,dc=lan`.
- **{domain_suffix}** — это суффикс домена в формате полного имени домена, например, `ald.company.lan`.

Рассмотрим создание пользователя на следующем примере. Для удобства каждый параметр будем задавать в отдельной строке с помощью символа переноса строки `\` backslash:

```
ipa user-add isemenov \
  --first="Иван" --last="Семенов" --middlename="Иванович" \
  --cn="Семенов И.И." --displayname="Семенов И.И." --initials="И.И." \
  --principal-expiration="$(date +'%Y%m%d000000Z' -d '+90 days')" \
  --password-expiration="$(date +'%Y%m%d000000Z' -d '+90 days')" \
  --random --email="semenov.ii@ald.company.lan" \
  --addattr="memberof=cn=ipausers,cn=groups,cn=accounts,dc=ald,dc=company,dc=lan" \
  --country="RU" --street="Ленина, 1" --city="Москва" --state="Московская область" \
  --postalcode="101" --phone="+71234567890" --mobile="+71234567890" \
```

(continues on next page)

(продолжение с предыдущей страницы)

```
 --fax="+71234567890" --orgunit="Отдел АСУ" --title="Инженер первой категории" \
 --departmentnumber="Департамент 1" --employeenumber="1" --employeetype="Категория 1"
```

Результат выполнения команды:

```
--------------------------------
Добавлен пользователь "isemenov"
--------------------------------
  Имя учётной записи пользователя: isemenov
  Имя: Иван
  Фамилия: Семенов
  Полное имя: Семенов И.И.
  Отображаемое имя: Семенов И.И.
  Инициалы: И.И.
  Домашний каталог: /home/isemenov
  GECOS: Иван Семенов
  Оболочка входа: /bin/bash
  Имя учётной записи: isemenov@ALD.COMPANY.LAN
  Псевдоним учётной записи: isemenov@ALD.COMPANY.LAN
  Окончание действия учётной записи Kerberos: 20250807000000Z
  Окончание действия пароля пользователя: 20250807000000Z
  Адрес электронной почты: semenov.ii@ald.company.lan
  Случайный пароль: 3Eg@Bx[G6m|aPP@KBgf3L9
  UID: 491200012
  ID группы: 491200012
  Адрес: Ленина, 1
  Город: Москва
  Область/республика: Московская область
  Индекс: 101
  Номер телефона: +71234567890
  Номер мобильного телефона: +71234567890
  Номер факса: +71234567890
  Отдел: Отдел АСУ
  Должность: Инженер первой категории
  Номер отдела: Департамент 1
  Номер сотрудника: 1
  Тип сотрудника: Категория 1
  Минимальный уровень конфиденциальности: 0
  Максимальный уровень конфиденциальности: 0
  Учётная запись отключена: FALSE
  nsuniqueid: 83778601-2cb011f0-81d3c33d-ecd7a769
  Middle name: Иванович
  Contry: RU
  proxy addresses: SMTP:SEMENOV.II@ALD.COMPANY.LAN
  Link to department: ou=ald.company.lan,cn=orgunits,cn=accounts,dc=ald,dc=company,
↪dc=lan
  Link to head department: ald.company.lan
  Пароль: True
  Участник групп: ipausers
  Роли: ALDPRO - Organization units
  Indirect Member of group: ipausers
  Доступные ключи Kerberos: True
```

Обратите внимание на случайный пароль `3Eg@Bx[G6m|aPP@KBgf3L9`, который был сгенерирован системой автоматически. Он будет нужен пользователю для первого входа в систему.

Если вам нужно будет назначить пароль пользователя вручную, то вместо ключа `--random` используйте ключ `--password`. А если потребуется назначить пароль из командной строки, то воспользуйтесь перенаправлением вывода:

```
echo 'Pa$$w0rd' | ipa user-add apetrov \
  --first="Александр" --last="Петров" --middlename="Владимирович" \
  --password --password-expiration="$(date +'%Y%m%d000000Z' -d '+90 days')"
```

Результат выполнения команды:

```
 ------------------------------
 Добавлен пользователь "apetrov"
 ------------------------------
   Имя учётной записи пользователя: apetrov
   Имя: Александр
   Фамилия: Петров
   Полное имя: Александр Петров
   Отображаемое имя: Александр Петров
   Инициалы: АП
   Домашний каталог: /home/apetrov
   GECOS: Александр Петров
   Оболочка входа: /bin/bash
   Имя учётной записи: apetrov@ALD.COMPANY.LAN
   Псевдоним учётной записи: apetrov@ALD.COMPANY.LAN
   Окончание действия пароля пользователя: 20250807000000Z
   Адрес электронной почты: apetrov@ald.company.lan
   UID: 491200013
   ID группы: 491200013
   Минимальный уровень конфиденциальности: 0
   Максимальный уровень конфиденциальности: 0
   Учётная запись отключена: FALSE
   nsuniqueid: cafe1201-2cb011f0-81d3c33d-ecd7a769
   Middle name: Владимирович
   proxy addresses: SMTP:APETROV@ALD.COMPANY.LAN
   Link to department: ou=ald.company.lan,cn=orgunits,cn=accounts,dc=ald,dc=company,
↪dc=lan
   Link to head department: ald.company.lan
   Пароль: True
   Участник групп: ipausers
   Роли: ALDPRO - Organization units
   Доступные ключи Kerberos: True
```

Изменение параметров пользователя (ipa user-mod)

Сотрудники меняют номера телефонов и фамилии, получают повышения в должности, поэтому вам потребуется менять отдельные атрибуты. Для этого вам потребуется команда `ipa user-mod`. Параметры команды практически совпадают с параметрами команды `ipa user-add`, см. табл. 2.6.

В качестве примера: если пользователь перевелся из московского офиса в казанский, вам потребуется изменить параметр *department* в свойствах пользователя. С точки зрения LDAP-каталога это равносильно изменению атрибута *rbtadp*. Создайте департамент *ou=kazan* на портале ALD Pro и

выполните команду:

```
ipa user-mod apetrov \
  --department="ou=kazan,ou=ald.company.lan,cn=orgunits,cn=accounts,dc=ald,dc=company,
↪dc=lan"
```

Результат выполнения команды:

```
 ------------------------------
 Изменён пользователь "apetrov"
 ------------------------------
   Имя учётной записи пользователя: apetrov
   Имя: Александр
   Фамилия: Петров
   ...
   Link to department: ou=kazan,ou=ald.company.lan,cn=orgunits,cn=accounts,dc=ald,
↪dc=company,dc=lan
   Link to head department: kazan
   ...
```

Пример изменения контактных данных пользователя:

```
ipa user-mod apetrov \
  --phone "+71122334455" --email "apetrov@kaz.ald.company.lan"
```

Результат выполнения команды:

```
 ------------------------------
 Изменён пользователь "apetrov"
 ------------------------------
   Имя учётной записи пользователя: apetrov
   Имя: Александр
   Фамилия: Петров
   ...
   Адрес электронной почты: apetrov@kaz.ald.company.lan
   ...
   Номер телефона: +71122334455
   ...
```

Аналогичным образом можно изменить любые другие атрибуты пользователя.

Установка нового пароля (ipa passwd)

Пользователь может сменить себе пароль с помощью команды passwd без параметров:

```
admin@dc-1:~$ ipa passwd
Текущий пароль:
Новый пароль:
Введите Новый пароль ещё раз для проверки:
---------------------------------------
Изменён пароль "admin@ALD.COMPANY.LAN"
---------------------------------------
```

С помощью этой же команды администраторы могут назначить новый пароль любой другой учетной записи, если у них достаточно прав доступа:

```
admin@dc-1:~$ ipa passwd apetrov
Новый пароль:
Введите Новый пароль ещё раз для проверки:
----------------------------------------
Изменён пароль "apetrov@ALD.COMPANY.LAN"
----------------------------------------
```

Аналогичный результат можно получить командой `ipa user-mod` с ключом `--password`:

```
admin@dc-1:~$ ipa user-mod --password apetrov
Пароль:
Введите Пароль ещё раз для проверки:
--------------------------------
Изменён пользователь "apetrov"
--------------------------------
Имя учётной записи пользователя: apetrov
...
```

Если администратор воспользуется ключом `--random`, то система автоматически сгенерирует и назначит новый пароль пользователю:

```
ipa user-mod --random apetrov
```

Результат выполнения команды:

```
 ---------------------------------
 Изменён пользователь "apetrov"
 ---------------------------------
 Имя учётной записи пользователя: apetrov
 ...
 Случайный пароль: Vf}7-NsaWNnV.OlM18]q^E
 ...
```

При назначении пользователю нового пароля из веб-интерфейса или из командной строки срок действия Kerberos-пароля автоматически проставляется равным текущей дате, поэтому пароль считается истекшим и требует изменения при следующем входе. Изменить значение этого атрибута можно, например, с помощью утилиты `ipa`, используя ключ `--password-expiration`:

```
ipa user-mod apetrov --password-expiration 20260101115110Z
```

Если пользователь сменит пароль самостоятельно, ему будет назначен срок действия в соответствии с установленными политиками (по умолчанию — 90 дней).

Активация/деактивация пользователя (ipa user-disable/user-enable)

С помощью команды `ipa user-disable` администратор может деактивировать учетную запись, чтобы заблокировать пользователю доступ, не удаляя учетную запись из домена. В этом случае атри-

буту nsAccountLock устанавливается значение True:

```
admin@dc-1:~$ ipa user-disable apetrov
-----------------------------------------------------
Учётная запись пользователя "apetrov" отключена
-----------------------------------------------------
```

Текущее состояние учетной записи можно посмотреть командой ipa user-show:

```
admin@dc-1:~$ ipa user-show apetrov --all --raw | grep nsaccountlock
  nsaccountlock: TRUE
```

Состояние блокировки показывает так же команда ipa user-status, но ее лучше не использовать, т.к. она больше предназначена для получения информации об автоматической временной блокировки пользователя со всех контроллеров домена, что занимает значительно больше времени.

```
  admin@dc-1:~$ ipa user-status apetrov
  ----------------------
  Account disabled: True
  ----------------------
  ...
```

Деактивированный пользователь не сможет войти в операционную систему доменного компьютера, пока не будет активирован с помощью команды ipa user-enable:

```
admin@dc-1:~$ ipa user-enable apetrov
-----------------------------------------------------
Учётная запись пользователя "apetrov" включена
-----------------------------------------------------
```

Снятие блокировки пользователя (ipa user-unlock)

Кроме активации/деактивации в домене есть еще автоматическая временная блокировка по количеству ошибок, которая позволяет исключить перебор паролей.

В соответствии с настройками по умолчанию, если пользователь в течение 1 минуты 6 раз подряд введет неправильно пароль, то его учетная запись будет автоматически заблокирована на следующие 10 минут. Указанные параметры можно изменить с помощью политик паролей.

Текущее состояние учетной записи можно узнать с помощью команды ipa user-status. Следует учитывать, что количество неправильных попыток записывается в атрибуте krbLoginFailedCount, который для повышения производительности исключен из репликации, поэтому бэкенд команды ipa user-status для сбора необходимой информации опрашивает все контроллеры в домене, что может занять некоторое время:

```
admin@dc-1:~$ ipa user-status apetrov
-----------------------
Account disabled: False
```

(continues on next page)

(продолжение с предыдущей страницы)

```
----------------------
  Сервер: dc-1.ald.company.lan
  Неудачные попытки входа: 6
  Последняя успешная аутентификация: N/A
  Последняя ошибка при аутентификации: 2025-04-04T09:15:18Z
  Время сейчас: 2025-04-04T09:19:01Z
...
  Сервер: dc-5.ald.company.lan
  Неудачные попытки входа: 0
  Последняя успешная аутентификация: N/A
  Последняя ошибка при аутентификации: 2025-04-04T09:15:18Z
  Время сейчас: 2025-04-04T09:19:01Z

--------------------------------
Количество возвращённых записей 5
--------------------------------
```

Можно увидеть, что число попыток на dc-1 равно 6. Поэтому пользователь не сможет пройти Kerberos-аутентификацию с помощью этого контроллера домена, но на работу других контроллеров домена эта блокировка никак не повлияет.

Чтобы разблокировать учетную запись пользователя, нужно воспользоваться командой `ipa user-unlock`. В этом случае выполняется сброс счетчика `krbLoginFailedCount`, а в атрибут `krbLastAdminUnlock` записывается текущее время системы:

```
admin@dc-1:~$ ipa user-unlock apetrov
----------------------------------------
Учётная запись "apetrov" разблокирована
----------------------------------------
```

Для снятия блокировки бэкенд команды `ipa user-unlock` обращается ко всем контроллерам в домене, поэтому выполнение команды может занять некоторое время.

Удаление пользователя (ipa user-del)

Удалить учетные записи пользователей из домена можно командой `ipa user-del`. По умолчанию используется ключ `--no-preserve`, поэтому запись удаляется безвозвратно. Если использовать ключ `--preserve`, то запись будет перемещена в контейнер хранимых пользователей `cn=deleted users,cn=accounts,cn=provisioning`.

```
admin@dc-1:~$ ipa user-del --preserve apetrov
--------------------------------
Удален пользователь "apetrov"
--------------------------------
```

В системе ALD Pro удаленные учетные записи можно будет найти в корзине, см. рис. 2.86, а в интерфейсе FreeIPA этот раздел называется «Хранимые пользователи».

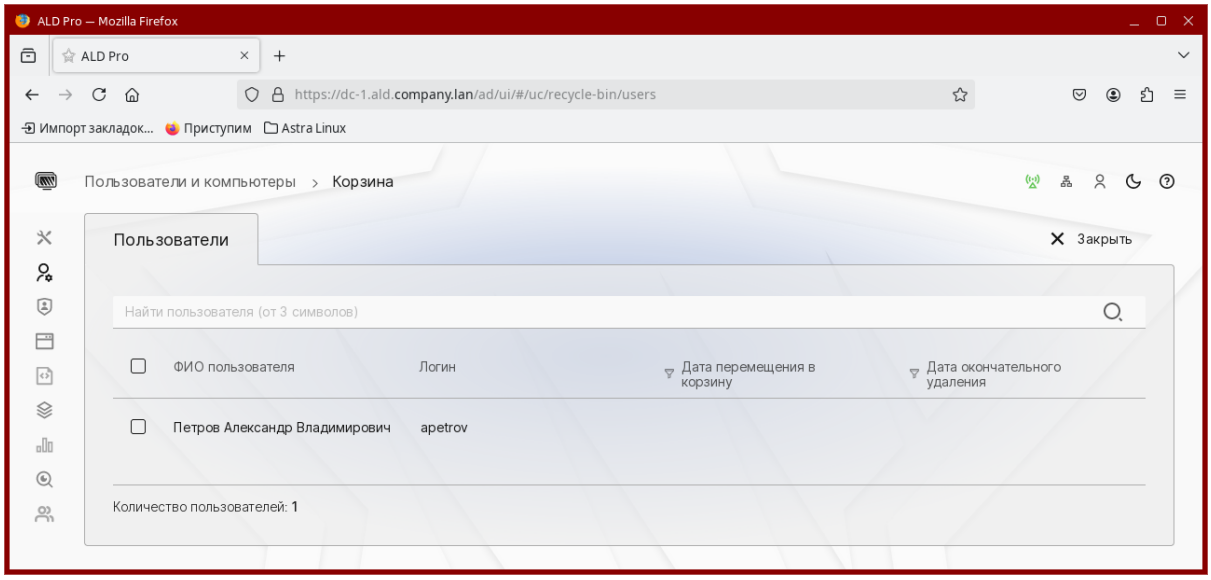


рис. 2.86 – Корзина с удаленными пользователями

Для поиска пользователей, помещенных в корзину, в команде `ipa user-find` нужно использовать ключ `--preserved true`:

```
admin@dc-1:~$ ipa user-find --preserved true
---------------------
найден 1 пользователь
---------------------
  Имя учётной записи пользователя: apetrov
  Имя: Александр
  Фамилия: Петров
  Домашний каталог: /home/apetrov
  Оболочка входа: /bin/bash
  Имя учётной записи: apetrov@ALD.COMPANY.LAN
  Псевдоним учётной записи: apetrov@ALD.COMPANY.LAN
  Адрес электронной почты: apetrov@kaz.ald.company.lan
  UID: 491200013
  ID группы: 491200013
  Номер телефона: +71122334455
  Учётная запись отключена: True
  Хранимый пользователь: True
----------------------------------
Количество возвращённых записей 1
----------------------------------
```

Для восстановления удаленной учетной записи пользователя из корзины используйте команду `ipa user-undel`:

```
admin@dc-1:~$ ipa user-undel apetrov
-------------------------------------------------------------
Учётная запись пользователя "apetrov" возвращена после удаления
-------------------------------------------------------------
```

Пользователи, находящиеся в корзине, блокируются автоматически, поэтому атрибут

nsAccountLock для этих учетных записей устанавливается равным True. После восстановления учетной записи из корзины указанный атрибут также автоматически удаляется. Если на момент помещения в корзину учетная запись уже была заблокирована, то после восстановления такой записи она так и останется в заблокированном состоянии, для нее атрибут nsAccountLock удаляться не будет.

4.3.1.4 Утилита ipa-getkeytab

Для аутентификации по протоколу Kerberos V5 система использует хеши паролей, которые называются также Kerberos-ключами. Для повышения совместимости клиент и сервер могут хранить сразу несколько разных хешей для одного и того же пароля и договариваться об использовании наиболее безопасного их них в момент аутентификации.

Внутри LDAP-каталога Kerberos-ключи хранятся в атрибуте krbPrincipalKey, а на диске в keytab-файлах, которые называют также таблицами ключей (*англ. Key Table*). Следует понимать, что внутри одного keytab-файла может храниться сразу несколько ключей для одного и того же пароля, или несколько ключей для разных версий пароля одного и того же пользователя или даже несколько ключей для разных пользователей.

Чтобы сбросить пароль доменной учетной записи и выгрузить Kerberos-ключи в keytab-файл, можно воспользоваться утилитой ipa-getkeytab. Содержимое keytab-файла представляет собой бинарные данные, посмотреть которые можно утилитой klist.

Создадим в домене учетную запись для нового сервиса **HTTP/pc-1.ald.company.lan** и опробуем на нем работу этих утилит:

```
admin@dc-1:~$ sudo ipa service-add HTTP/pc-1.ald.company.lan
--------------------------------------------------------------
Добавлена служба "HTTP/pc-1.ald.company.lan@ALD.COMPANY.LAN"
--------------------------------------------------------------
  Имя учётной записи: HTTP/pc-1.ald.company.lan@ALD.COMPANY.LAN
  Псевдоним учётной записи: HTTP/pc-1.ald.company.lan@ALD.COMPANY.LAN
  Managed by: pc-1.ald.company.lan

admin@dc-1:~$ sudo ipa-getkeytab -s dc-1.ald.company.lan \
>  -p HTTP/pc-1.ald.company.lan -k /etc/http.keytab
Таблица ключей успешно получена и сохранена в: /etc/http.keytab

admin@dc-1:~$ sudo klist -ke /etc/http.keytab
Keytab name: FILE:/etc/http.keytab
KVNO Principal
---- --------------------------------------------------------------------------
   1 HTTP/pc-1.ald.company.lan@ALD.COMPANY.LAN (aes256-cts-hmac-sha1-96)
   1 HTTP/pc-1.ald.company.lan@ALD.COMPANY.LAN (aes128-cts-hmac-sha1-96)
```

При вызове утилиты ipa-getkeytab используются следующие ключи:

- `-s dc-1.ald.company.lan` — задает имя KDC-сервера, к которому следует отправить запрос.
- `-p HTTP/pc-1.ald.company.lan` — определяет имя сервиса.
- `-k /etc/http.keytab` — задает путь для сохранения keytab-файла.

При вызове утилиты `klist` используются следующие ключи:

- `-k` — задает путь до keytab-файла, который нужно прочитать.
- `-e` — указывает, что нужно отобразить ключи с использованным алгоритмом хеширования.

Получить keytab-файл можно как для сервиса, так и для учетной записи обычного пользователя. Следует только учитывать, что в момент выгрузки keytab-файла система генерирует новый пароль, но не отображает его на экране, поэтому аутентифицироваться такой пользователь после этого сможет только по keytab-файлу.

Создадим keytab-файл для пользователя `iivanov` и пройдем аутентификацию в домене с помощью ключей из этого файла:

```
admin@dc-1:~$ sudo ipa-getkeytab -s dc-1.ald.company.lan \
>  -p apetrov@ALD.COMPANY.LAN -k /home/admin/apetrov.keytab
  Таблица ключей успешно получена и сохранена в: /home/admin/apetrov.keytab

admin@dc-1:~$ sudo kinit apetrov -kt /home/admin/apetrov.keytab

admin@dc-1:~$ sudo klist
Ticket cache: KEYRING:persistent:491200000:krb_ccache_9andJvL
Default principal: apetrov@ALD.COMPANY.LAN

Valid starting       Expires              Service principal
02.05.2025 11:51:35  03.05.2025 11:39:08  krbtgt/ALD.COMPANY.LAN@ALD.COMPANY.LAN
```

4.3.1.5 Утилиты ipa-backup и ipa-restore

Для резервного копирования и восстановления информации каталога предназначены утилиты `ipa-backup` и `ipa-restore`. Следует понимать, что на контроллере домена кроме службы каталога есть еще целый ряд сервисов, поэтому данная копия не позволит вам восстановить контроллер целиком, но зато вы можете застраховать себя от полной потери данных.

Следующей командой можно создать резервную копию каталога:

```
admin@dc-1:~$ sudo ipa-backup --data --online --verbose --log-file freeipa-backup.log
ipalib.sysrestore: DEBUG: Loading StateFile from '/var/lib/ipa/sysrestore/sysrestore.
↪state'
ipalib.sysrestore: DEBUG: Loading StateFile from '/var/lib/ipa/sysrestore/sysrestore.
↪state'
. . .
ipaserver.install.ipa_backup: INFO: Backing up userRoot in ALD-COMPANY-LAN to LDIF
```

(continues on next page)

(продолжение с предыдущей страницы)

```
ipaserver.install.ipa_backup: INFO: Waiting for LDIF to finish
ipaserver.install.ipa_backup: INFO: Backing up ALD-COMPANY-LAN
ipaserver.install.ipa_backup: INFO: Waiting for BAK to finish
ipapython.ipautil: DEBUG: Starting external process
ipapython.ipautil: DEBUG: args=['tar', '--xattrs', '--selinux', '-czf', '/var/lib/ipa/
↪backup/ipa- data-2024-03-22-21-26-19/ipa-data.tar', '.']
ipapython.ipautil: DEBUG: Process finished, return code=0
ipapython.ipautil: DEBUG: stdout=
ipapython.ipautil: DEBUG: stderr=
ipaserver.install.ipa_backup: INFO: Backed up to /var/lib/ipa/backup/ipa-data-2024-01-
↪22-21-26-19
ipapython.admintool: INFO: The ipa-backup command was successful
```

Где:

- Ключ `--data` – указывает, что нужно выполнить резервную копию только данных. С помощью ключа `--logs` можно включить еще журналы изменений. Если не определять никакой ключ, то в резервную копию будут включены все данные.
- Ключ `--online` – требует выполнить резервную копию без остановки служб.
- Ключ `--verbose` – указывает, что нужно вывести отладочную информацию.
- Ключ `--log-file` – задает имя файла, в который будет сохранен журнал о выполнении процедуры резервного копирования.

А вот так вы можете восстановить данные из копии:

```
admin@dc-1:~$ sudo ipa-restore --data --online --instance=ALD-COMPANY-LAN /var/lib/
↪ipa/backup/ipa-data-2024-01-22-21-26-19
Directory Manager (existing master) password:
Preparing restore from /var/lib/ipa/backup/ipa-data-2024-01-22-21-26-19 on dc-1.ald.
↪company.lan
Performing DATA restore from DATA backup
Temporary setting umask to 022
Restoring data will overwrite existing live data. Continue to restore? [no]: yes
Each master will individually need to be re-initialized or
re-created from this one. The replication agreements on
masters running IPA 3.1 or earlier will need to be manually
re-enabled. See the man page for details.
Disabling all replication.
Disabling replication agreement on dc-2.ald.company.lan to dc-1.ald.company.lan
Starting Directory Server
Restoring from userRoot in ALD-COMPANY-LAN
Waiting for LDIF to finish
Restoring umask to 18
The ipa-restore command was successful
```

Где:

- Ключ `--data` – указывает, что требуется восстановить только данные.

- Ключ `--online` – требует выполнить восстановление без остановки служб.
- Ключ `--instance` – задает имя экземпляра службы каталога, т.к. на одном сервере их может быть несколько. Например, в ALD Pro реализован модуль глобального каталога, который представляет собой еще один экземпляр LDAP-каталога, который доступен по порту 3268/TCP.

  Глобальный каталог нужен для того, чтобы при работе в доверительных отношениях с MS AD вы могли из стандартных интерфейсов Windows назначать права доступа на объекты ALD Pro, выбирая их из списка. Без глобального каталога поиск по объектам доверенного домена работать не будет.

Однако при восстановлении каталога из резервной копии в домене из нескольких контроллеров вам потребуется сделать также полную реинициализацию остальных реплик, т.к. у восстановленного контроллера будет новое значение генератора CSN и он выпадет из инкрементальной репликации. Детали этой процедуры можно будет понять после подробного ознакомления с алгоритмом работы репликации, см. *Модуль 7. Дополнительные подсистемы ALD Pro*.

4.4 Сервер LDAP 389 Directory Server

Для распределенного хранения данных в домене используется LDAP-каталог 389 Directory Server (ранее Fedora Directory Server, а до этого Netscape Directory Server). Эта служба соответствует уровню предприятий и позволяет реплицировать данные на сотни серверов, обеспечивая сотни тысяч операций чтения/записи в секунду.

Служба 389 DS позволяет хранить в базе миллионы записей, имеет мощный механизм разграничения доступа вплоть до уровня отдельных атрибутов, поддерживает протокол LDAP v3, в том числе с шифрованием SSL/TLS и SASL (при выполнении Kerberos-аутентификации).

В качестве встроенной базы данных 389 DS использует очень популярную библиотеку BDB (Berkeley DB), но в предстоящих релизах ожидается его замена на LMDB (Lightning Memory-Mapped Database), что повысит производительность отдельных операций в несколько раз.

4.4.1 Каталог LDAP

Каталог - это специализированная иерархическая нереляционная база данных для хранения информации о людях, компьютерах, группах и других объектах.

Изначально по стандарту X.500 доступ к каталогам предполагался через компьютерную сеть OSI, а разработанный в дальнейшем облегченный протокол доступа к данным каталога (Lightweight Directory Access Protocol, LDAP) дал возможность взаимодействия с каталогами по сетям TCP/IP.

Функции LDAP-каталога на контроллере домена реализованы средствами 389 Directory Server, файлы которого вы найдете в папке `/var/lib/dirsrv/slapd-ALD-COMPANY-LAN/db`. Информация каталога представлена в виде древовидной структуры, которую также называют Directory Information Tree или сокращенно DIT, см. рис. 2.87.

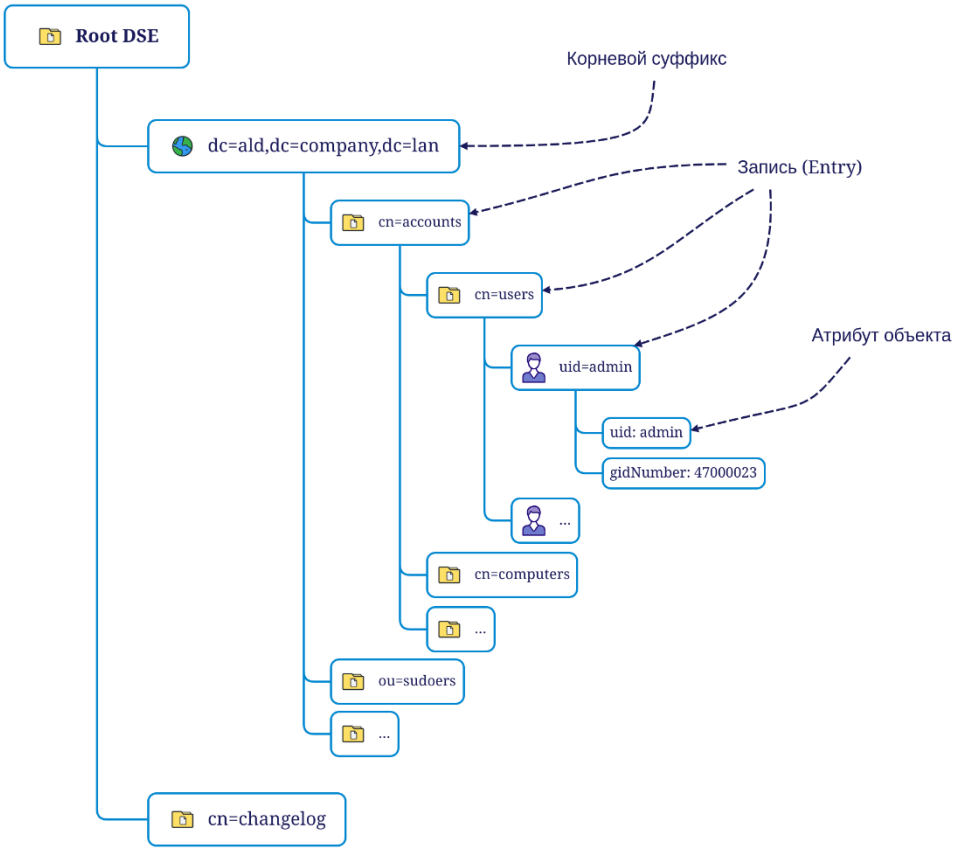


рис. 2.87 – Структура каталога Directory Information Tree

Корень каталога называется Root DSE, где DSE означает Directory System Agent Specific Entry, то есть специализированная запись агента системы директорий. Эта запись не имеет родителя, и она описана в файле `/etc/dirsrv/slapd-ALD-COMPANY-LAN/dse.ldif`, где slapd-ALD-COMPANY-LAN - это директория сервиса `slapd` для домена `ALD-COMPANY-LAN`.

Корень Root DSE является родителем для записей `dc=ald,dc=company,dc=lan` и `cn=changelog`, которые называют также базовыми записями, корневыми суффиксами или контекстами именования (*от англ. base entry, root suffix, naming context*). В контексте `dc=ald,dc=company,dc=lan` хранятся все записи каталога, а в `cn=changelog` — журнал изменений для работы плагина Retro Changelog Plugin. Все контексты, определенные в каталоге, указаны в операционном атрибуте `namingContexts` записи Root DSE, однако спецификация LDAP позволяет также использовать служебные контексты. Например, в записи `cn=config` представлены настройки каталога, а через запись `cn=monitor` можно получить доступ к информации о состоянии сервера в режиме реаль-

ного времени.

От корневого суффикса `dc=ald,dc=company,dc=lan` ответвляются дочерние записи (*от англ. entry*), которые, в свою очередь, могут быть контейнерами для других записей, за счет чего и образуется древовидная структура. Таким образом, записи каталога можно сравнить с директориями файловой системы.

Записи каталога называют так же объектами или контейнерами. У каждого объекта есть имя, которое должно быть уникальным в пределах родительского контейнера, поэтому это имя называется относительно уникальным именем (*англ. relative distinguished name или кратко RDN*). Учетные записи пользователей, например, имеют имена `uid=admin`, `uid=ivan.kuznetsov` и т.д. Особенность имен записей в LDAP заключается в том, что они хранят не конкретные значения, а только ссылки на хранимые атрибуты записей, которые используются для идентификации записей. Например, для идентификации учетных записей пользователей используют атрибут `uid`, для учетных записей компьютеров — `fqdn` (*от англ. fully qualified domain name, полное доменное имя хоста*), а в именах контейнеров обычно присутствует `cn` (*от англ. common name, общее имя*).

В приведенном примере RDN учетной записи доменного администратора `uid=admin` состоит из названия атрибута `uid`, после которого идет символ присвоения `=` и далее значение атрибута `admin`. Вся эта запись вместе называется определением значения атрибута (*от англ. attribute value assertion, AVA*). Обычно имена записей задаются значением одного атрибута, но могут использоваться и несколько, тогда в имени RDN эти определения AVA будут объединяться знаком сложения `+`. Например, при конфликтах репликации запись переименовывается в формате `nsuniqueid=0a950601...+uid=Ivan`.

Для идентификации записи в пределах всего каталога используют уникальное отличительное имя (*англ. distinguished name или сокращенно DN*). Отличительное имя представляет собой цепочку RDN, которые записывают через запятую слева направо, начиная с целевой записи и до корневого суффикса вверх по иерархии, см. рис. 2.88.

Если привести аналогию с объектами файловой системы, то RDN будет соответствовать относительному пути, а DN — полному абсолютному пути до объекта. И так же как на одном диске не может быть двух директорий с одинаковыми полными именами, в каталоге LDAP не может быть двух записей с одинаковыми DN. Описание формата DN можно найти в RFC 4514.

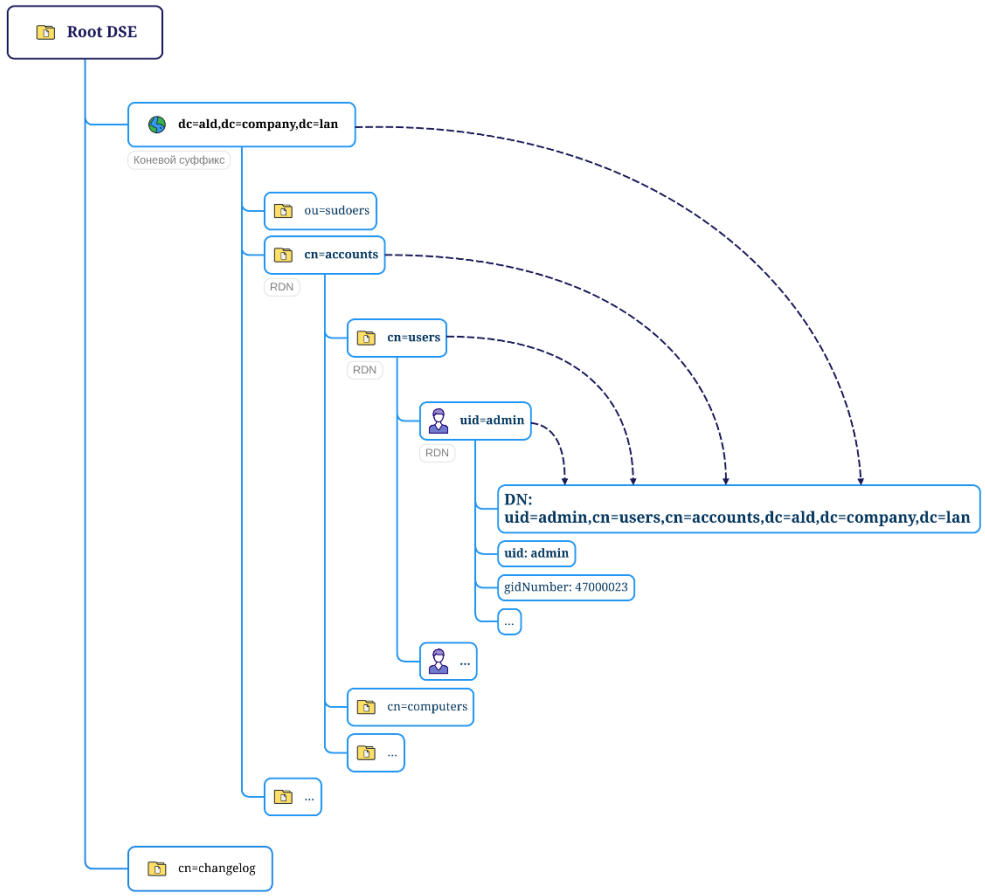


рис. 2.88 – Пример формирования уникального отличительного имени записи DN

Запись каталога может хранить не только дочерние записи, но и набор атрибутов, описывающий свойства записи. Например, для учетной записи пользователя атрибутами могут быть Ф.И.О., должность, номер телефона и т.д. Продолжая аналогию с файловой системой, можно сказать, что запись соответствует папке на диске, а атрибут - файлу, поэтому внутри папки могут быть другие папки (записи), а файлы (атрибуты) являются конечными листьями в этом дереве.

Обычные атрибуты называются «хранимыми», но в LDAP-каталоге кроме них есть еще служебные «операционные» атрибуты, которые содержат вспомогательную информацию для обеспечения работы каталога и связи между записями. Например, чтобы не пересчитывать вручную количество дочерних записей, вы можете воспользоваться операционным атрибутом `numSubordinates`, который кэширует для вас это значение.

Рассмотрим несколько интересных операционных атрибутов:

- `entrydn` или `dsEntryDN` — позволяют получить уникальное отличительное имя записи в секции атрибутов, что очень требуется иногда для подключения к каталогу внешних информационных систем. Например, сервису VK Почта для интеграции очень нужно видеть в

атрибутах полное значение отличительного имени пользователя.

- `nsUniqueId` и `entryUUID` — определяют уникальный идентификатор записи в пределах домена. Для каталога 389 Directory Server, как потомка Netscape Directory Server (NS), нативным атрибутом является `nsUniqueId`, а дополнительный атрибут `entryUUID` был добавлен упрощения миграции с OpenLDAP на 389 Directory Server после того, как два основных корпоративных дистрибутива Linux (SUSE и Red Hat) решили удалить OpenLDAP из своих репозиториев.

- `entryid` и `parentid` — хранят идентификатор текущей и родительской записи, который уникален для каждого экземпляра 389 Directory Server.

- `entryusn` — номер обновления записи в последовательности изменений (англ. Update Sequence Number, USN), который уникален для каждого экземпляра 389 Directory Server.

- `creatorsName`, `modifiersName`, `createTimestamp` и `modifyTimestamp` — хранят соответственно имена тех пользователей, которые создали и изменили запись в каталоге, а так же дату и время этих событий. Значения будут одинаковыми на всех серверах в домене.

  Обратите внимание, если вам потребуется забирать из каталога последние изменения простым LDAP-запросом, то правильно будет сортировать и фильтровать записи по атрибуту `entryusn` и не использовать `modifyTimestamp`, т.к. изменения, сделанные на других серверах, будут реплицироваться с задержками.

- `numSubordinates` — хранит текущее значение количества дочерних записей. Следует учитывать, что значение показывает общее количество всех записей, включая и скрытые операционные записи, отмеченные классом `ldapsubentry`.

Данные в LDAP-каталоге должны быть строго структурированы, поэтому при включенной проверке целостности новые атрибуты должны быть определены в схеме каталога заранее. Перечень атрибутов, доступных для конкретного объекта, определяется списком назначенных ему классов, см. атрибут `objectClass`. Например, учетные записи пользователей могут иметь атрибуты `uidNumber` и `gidNumber`, так как им присвоен класс объектов `posixAccount`, см. рис. 2.89.

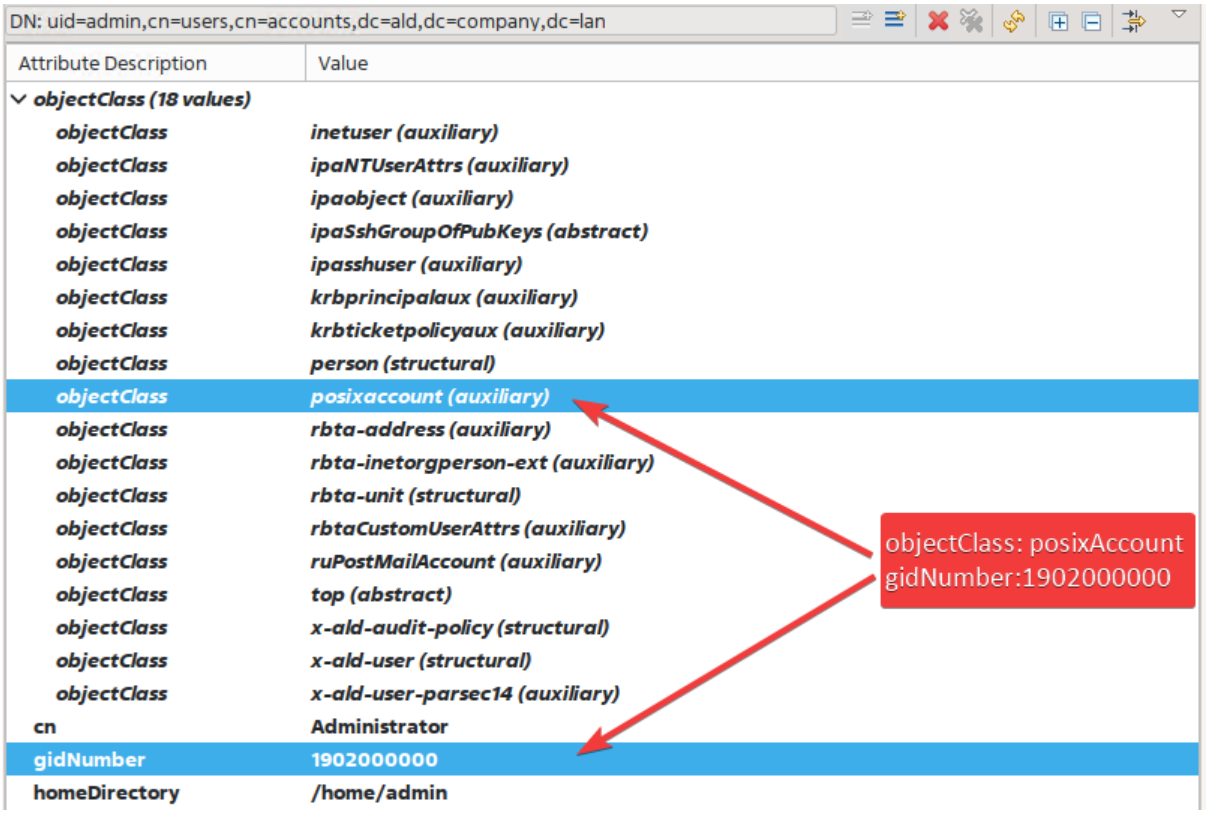


рис. 2.89 – Наличие атрибута gidNumber определяется классом объекта posixAccount

Всем объектам каталога прямо или через наследование назначен как минимум один класс `top`, т.к. в нем описан атрибут `objectClass`, с помощью которого работает механизм назначения классов. Но практической пользы от объектов с одним классом немного, поэтому обычно в каталоге у объектов два и более классов.

В качестве примера рассмотрим класс объектов `posixGroup`, который определен в схеме следующим образом:

```
( 1.3.6.1.1.1.2.2 NAME 'posixGroup' DESC 'Standard LDAP objectclass' SUP top␣
↪STRUCTURAL MUST ( cn $ gidNumber ) MAY ( userPassword $ memberUid $ description ) X-
↪ORIGIN 'RFC 2307' )
```

Где:

- `1.3.6.1.1.1.2.2` – это идентификатор объекта (object id, OID). Глобальные идентификаторы присваиваются международными организациями (IANA, ISO, ITU-T, ANSI, BSI).

  Для расширения схемы в прикладных системах используется пространство номеров с префиксом «1.3.6.1.4.1.<X>», где <**X**> – это внутренний номер организации. Например, объекты РусБИТех-Астра имеют префикс «1.3.6.1.4.1.52616.*».

- `NAME '...'` – инструкция NAME задает имя класса.

- DESC '...' – инструкция *DESC* задает описание класса.
- SUP '...' – инструкция SUP указывает родительский класс. В приведенном примере наследование идет от класса top, поэтому объектам будет доступен его атрибут objectClass.
- STRUCTURAL – инструкция указывает, что класс будет относиться к виду структурных. Существуют также абстрактные (ABSTRACT) и вспомогательные (AUXILIARY) классы.
- MUST и MAY – инструкции, которые позволяют задать списки обязательных и дополнительных атрибутов соответственно. Полный перечень атрибутов, доступных объекту, расширяется атрибутами, которые наследуются от всех родительских классов.
- X-ORIGIN '...' – инструкция, которая позволяет задать комментарий с ссылкой на документацию, из которой можно почерпнуть дополнительную информацию об этом классе. В приведенном примере информацию следует искать в документе RFC 2307.

Обратите внимание на то, что один и тот же атрибут может быть использован в нескольких классах, поэтому пользователям можно задать значение gidNumber, т.к. им назначен класс posixAccount, и в то же время он может быть задан у групп, т.к. им назначен класс posixGroup, см. рис. 2.90.

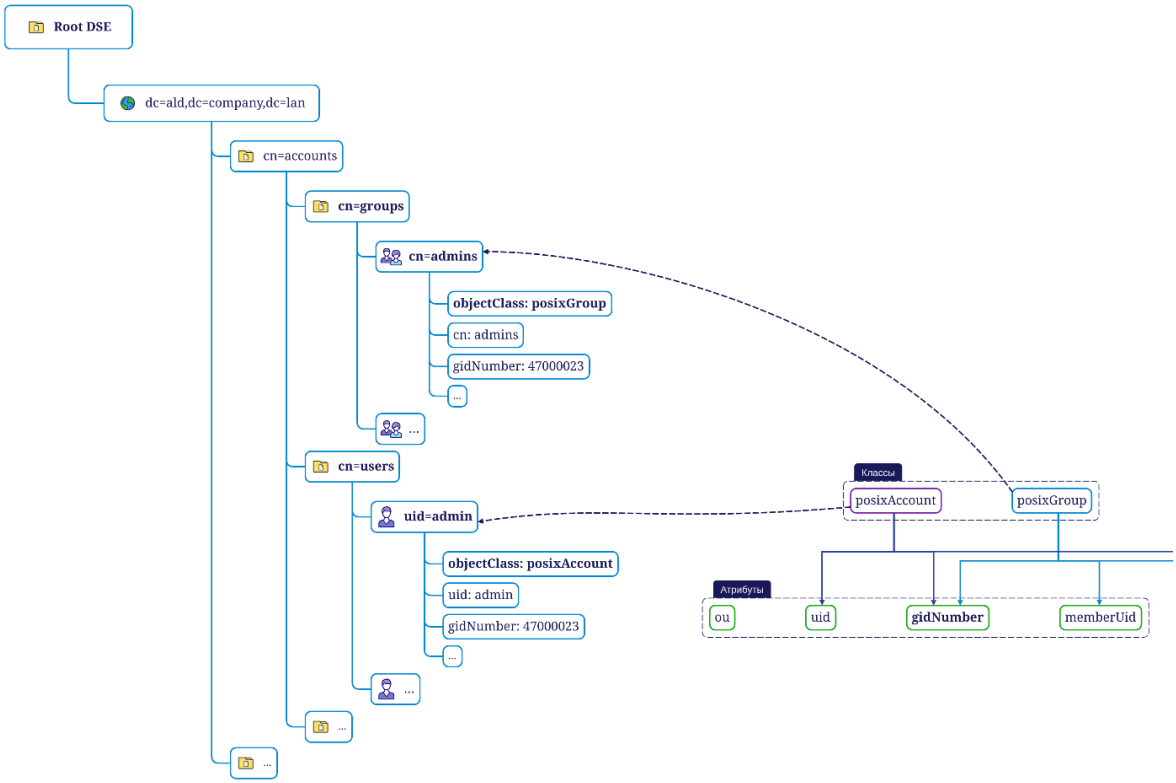


рис. 2.90 – Использование атрибута gidNumber классами posixAccount и posixGroup

Теперь рассмотрите описание атрибута gidNumber подробнее:

```
( 1.3.6.1.1.1.1.1 NAME 'gidNumber' DESC 'Standard LDAP attribute type' SYNTAX 1.3.6.1.
↪4.1.1466.115.121.1.27 SINGLE-VALUE X-ORIGIN 'RFC 2307' )
```

Где:

- `1.3.6.1.1.1.1.1` – это уникальный идентификатор атрибута.
- `NAME '...'` – инструкция NAME задает имя атрибута.
- `DESC '...'` – инструкция DESC задает описание атрибута.
- `SYNTAX '...'` – инструкция задает тип хранимого в атрибуте значения. В приведенном примере 1.3.6.1.4.1.1466.115.121.1.27 соответствует целым числам. Описание всех типов данных можно найти в RFC 4517.
- `SINGLE-VALUE` – инструкция указывает, что атрибут может хранить только одно значение. Если этой инструкции не будет, то объектам можно будет присваивать несколько значений этого атрибута.
- `X-ORIGIN '...'` – инструкция, которая позволяет задать комментарий со ссылкой на документацию.

Если вы хотите подробнее ознакомиться с доступными классами и атрибутами, то при обращении к каталогу по LDAP информацию можно получить из операционного DIT *cn=schema*. На диске эта информация хранится в директориях:

- `/usr/share/dirsrv/schema/`
- `/etc/dirsrv/schema/`
- `/usr/share/dirsrv/updates/`
- `/etc/dirsrv/slapd-ALD-COMPANY-LAN/`

Вот полная информация обо всех файлах 389 Directory Server на диске:

- Папка `/usr/sbin/` содержит исполнимые файлы:
  - Файл `/usr/sbin/ns-slapd` – исполнимый файл сервиса.
  - Файл `/usr/sbin/dsctl` – утилита управления сервисом.
  - Файл `/usr/sbin/dsconf` – утилита настройки и мониторинга.
- Папка `/usr/lib/systemd/system/` содержит файлы сервиса:
  - Файл `/usr/lib/systemd/system/dirsrv.target` – это “целевой” файл, используемый для

совместной работы со всеми экземплярами dirsrv.

- Файл `/usr/lib/systemd/system/dirsrv@<название-Экземпляра>.service` (если экземпляр только один, то имя сервиса будет dirsrv@.service).

- Папка `/var/lock/dirsrv/slapd-<название-Экземпляра>/` содержит файлы блокировок базы данных.

- Папка `/var/lib/dirsrv/slapd-<название-Экземпляра>/`` содержит файлы резервных копий, базы данных, журналов изменений и LDIF-файлы:

  - Папка `/var/lib/dirsrv/slapd-<название-Экземпляра>/bak/` содержит резервные копии LDAP-каталога, созданные командой `dsconf -D "cn=Directory Manager" ldap://localhost backup create`.

  - Папка `/var/lib/dirsrv/slapd-<название-Экземпляра>/db/` содержит базу данных каталога.

  - Папка `/var/lib/dirsrv/slapd-<название-Экземпляра>/ldif/` содержит LDIF-файлы, которые создаются, например, при резервном копировании каталога командой `sudo dsctl ALD-COMPANY-LAN db2ldif userRoot`.

- Папка конфигураций `/etc/dirsrv/` содержит:

  - Папка `/etc/dirsrv/config/` содержит конфигурационные файлы.

  - Папка `/etc/dirsrv/schema/` содержит файлы файлы схемы LDAP-каталога.

  - Папка `/etc/dirsrv/ssca/` содержит сертификаты и ключи самоподписанного центра сертификации (англ. Simple Self-Signed Certificate Authority, SSCA).

  - Папка `/etc/dirsrv/slapd-<название-Экземпляра>/schema/` содержит файлы схемы LDAP-каталога для конкретного экземпляра сервера. В этом каталоге находится в том числе файл `99user.ldif` с изменениями схемы каталога, которые вносятся через LDAP-запросы и реплицируются автоматически между всеми контроллерами в домене.

- Папка `/usr/share/dirsrv/schema/` содержит схему каталога по умолчанию.

В нашем случае корневой суффикс каталога был задан как `dc=ald,dc=company,dc=lan`, а название экземпляра, соответственно, `ALD-COMPANY-LAN`. Рассмотрите самостоятельно содержимое папок `/var/lib/dirsrv/slapd-ALD-COMPANY-LAN/` и `/etc/dirsrv/slapd-ALD-COMPANY-LAN/`

4.4.2 Управление LDAP-каталогом

Для управления каталогом по протоколу LDAP можно использовать различные инструменты:

- **Apache Directory Studio** — графический инструмент, который позволяет просматривать и редактировать записи и атрибуты записей;
- **ldapsearch** — утилита, которая позволяет просматривать и редактировать записи и атрибуты записей;
- **dsctl** – утилита управления сервисом;
- **dsconf** – утилита настройки и мониторинга.

4.4.2.1 Работа с каталогом из Apache Directory Studio

Для работы с LDAP-каталогом из графического интерфейса — просмотра структуры каталога, редактирования записей, импорта и экспорта данных — можно воспользоваться таким бесплатным инструментом, как Apache Directory Studio, см. рис. 2.91.

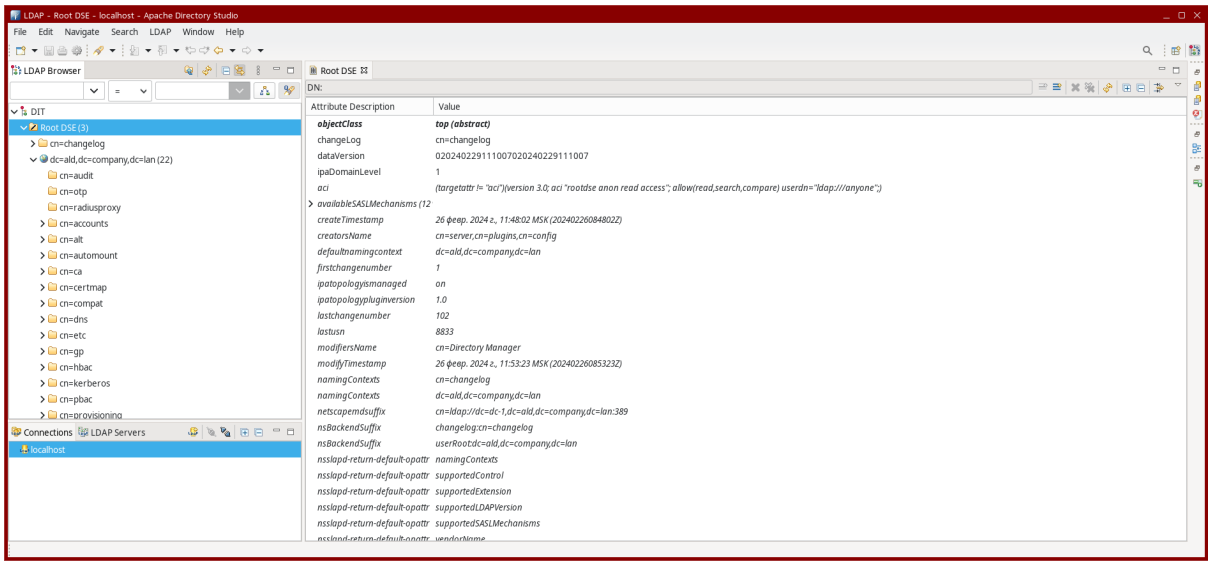


рис. 2.91 – Интерфейс Apache Directory Studio

Загрузить приложение можно с официального сайта. На контроллерах запускать приложение можно будет сразу, а на клиентской машине потребуется установить виртуальную машину Java (англ. Java Runtime Environment, JRE).

Для установки **Apache Directory Studio** необходимо:

1) Получить актуальную ссылку на архив с программой на сайте разработчика и скачать этот архив на клиентскую виртуальную машину, например, с помощью утилиты `wget`.

```
wget https://dlcdn.apache.org/directory/studio/2.0.0.v20210717-M17/
↪ApacheDirectoryStudio-2.0.0.v20210717-M17-linux.gtk.x86_64.tar.gz
```

2) Распаковать архив в текущий каталог:

```
tar -xvzf ApacheDirectoryStudio-2.0.0.v20210717-M17-linux.gtk.x86_64.tar.gz
```

3) Установите виртуальную машину Java:

```
sudo apt install default-jre
```

Для запуска Apache Directory Studio просто кликните дважды по исполняемому файлу программы в окне файлового менеджера или выполните в окне терминала следующую команду:

```
~/ApacheDirectoryStudio/ApacheDirectoryStudio
```

Настройка соединения

Чтобы подключиться к LDAP-каталогу, нужно создать новое подключение через меню *LDAP* → *New connection...* . Откроется окно для создания нового подключения, см. рис. 2.92.

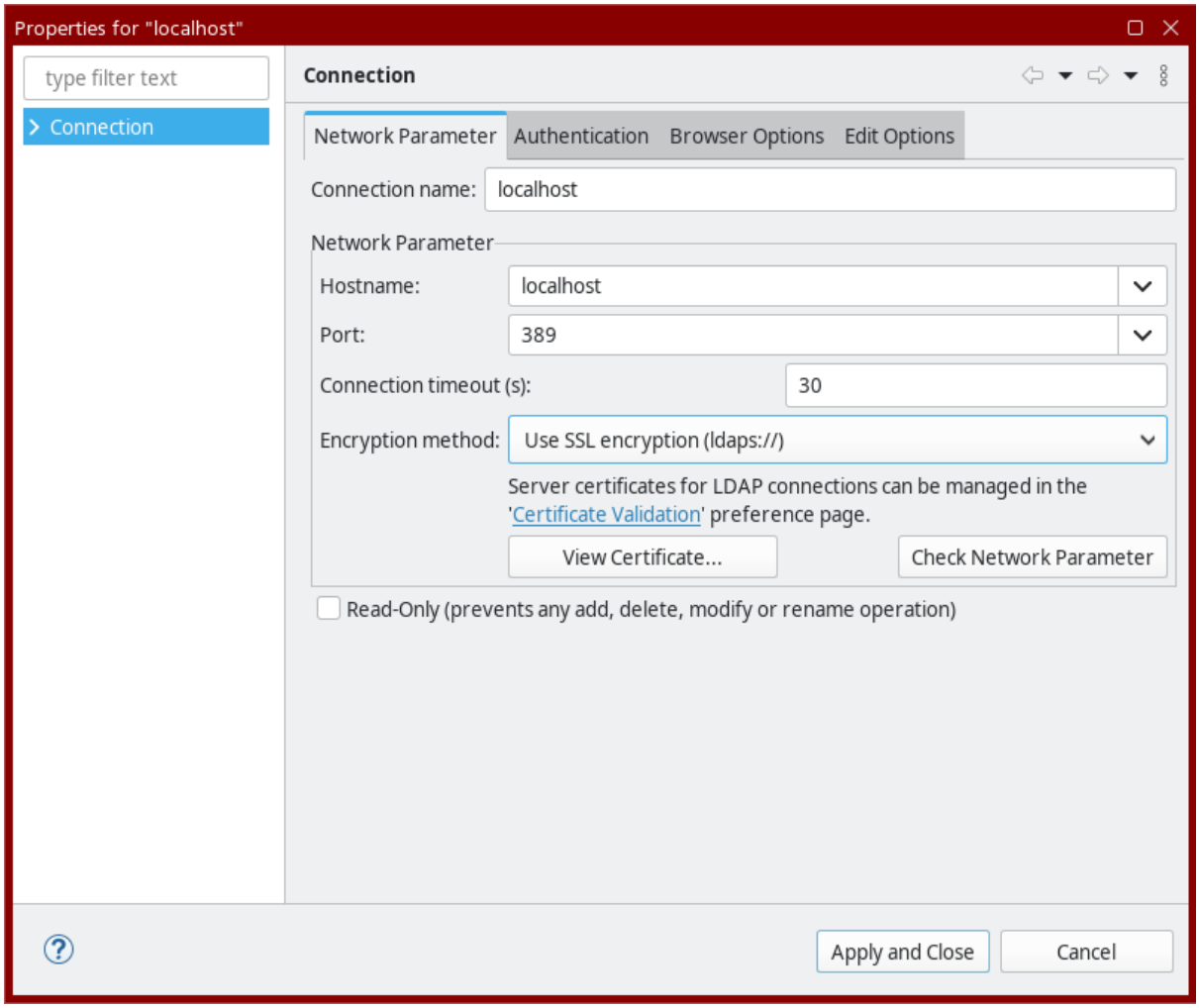


рис. 2.92 – Настройка нового LDAP-подключения

Важно: По умолчанию Apache Directory Studio предлагает создать незащищенное подключение на порт 389, что допустимо только при обращении к каталогу по localhost, т.е. приложение должно быть установлено непосредственно на контроллере домена, т.к. при подключении пароль будет передаваться в открытом виде.

Если вы подключаетесь к каталогу с другого компьютера в сети, обязательно используйте порт 636 и метод шифрования SSL, чтобы перехват пароля был невозможен.

Аутентификация по паролю называется также привязкой (*от англ. bind*), поэтому поле для ввода имени пользователя называется *Bind DN or user*, см. рис. 2.93. В этом поле можно указать как имя суперпользователя каталога **cn=Directory Manager**, так и полное отличительное имя пользователя, например, доменного администратора **uid=admin,cn=users,cn=accounts,dc=ald,dc=company,dc=lan**.

Перед закрытием окна проверьте настройку текущего подключения нажатием кнопки *Check Authentication*, см. рис. 2.93.

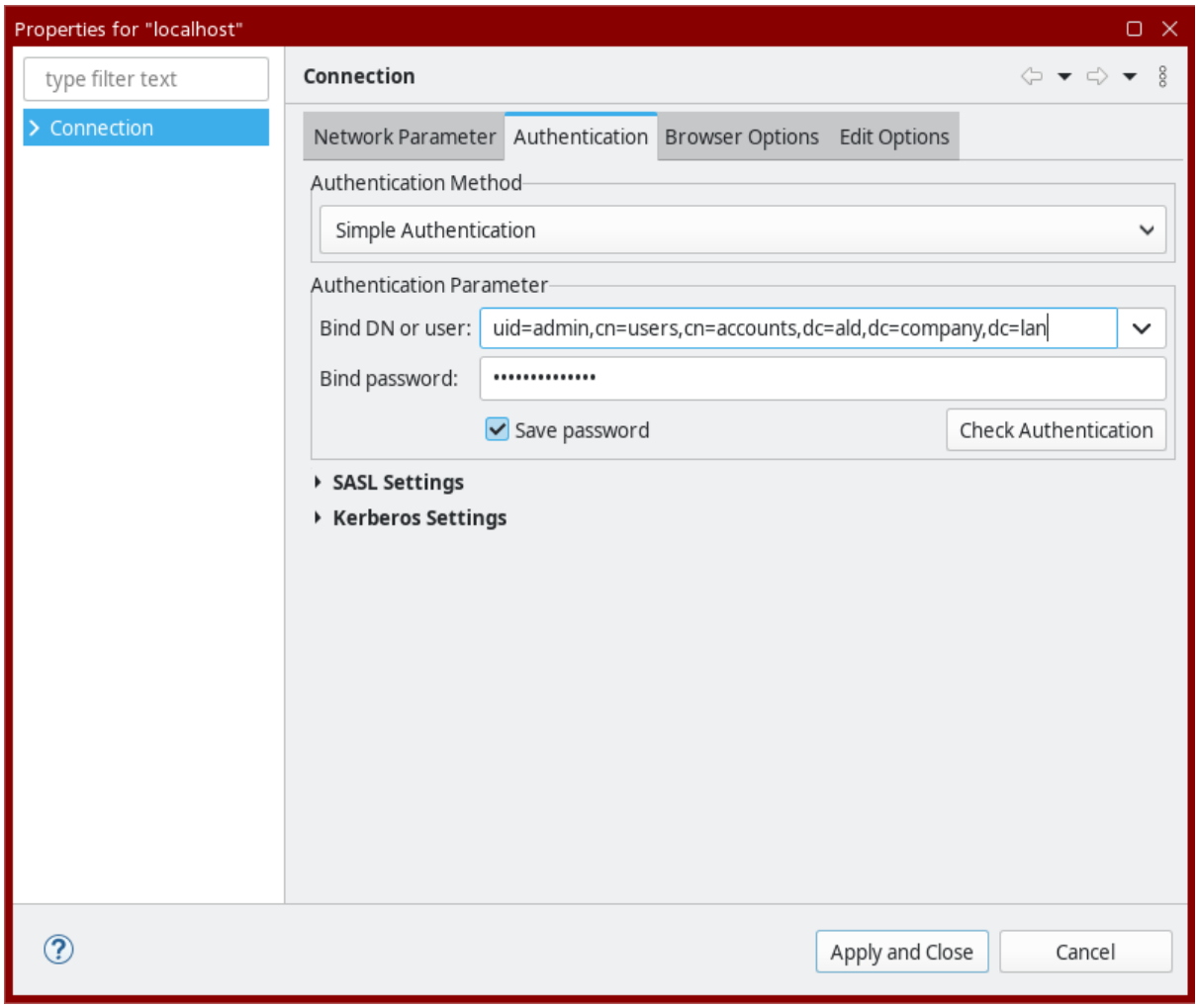


рис. 2.93 – Настройка аутентификации для нового LDAP-подключения

После добавления подключения новый сервер появится в списке *Connections*. Вы сможете подключиться к нему, кликнув дважды по строке с названием подключения. В зависимости от выбранной учетной записи у вас будут разные права на доступ к записям и атрибутам.

Давайте разберем три типовых операции: просмотр и экспорт объектов каталога, просмотр схемы каталога.

Просмотр и экспорт объектов каталога

Вы можете просмотреть записи в дереве каталога, выбрав нужный RDN на вкладке *LDAP Browser*. Например, если вы выберите в дереве узел *cn=accounts*, то основные атрибуты этой записи отобразятся в центральном окне с заголовком *cn=accounts,dc=ald,dc=company,dc=lan*, см. рис. 2.94.

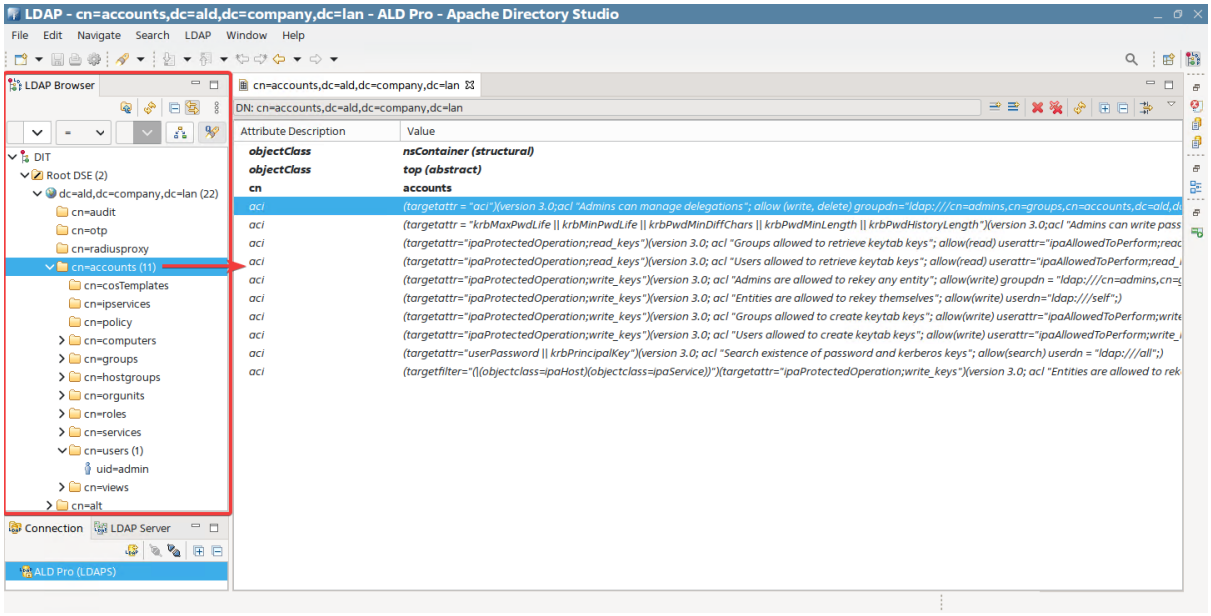


рис. 2.94 – *Окно LDAP Browser*

Если вы знаете полное имя записи, то сможете перейти к ней сразу из меню *Navigate → Go to DN*. Например, если вы подключились к каталогу с помощью учетной записи суперпользователя cn=Directory Manager, то сможете таким образом открыть ветку *cn=config* с настройками сервера, которая по умолчанию скрыта.

Рассмотрим несколько полезных DN:

- cn=users,cn=accounts,суффикс_домена — содержит учетные записи пользователей домена. У каждого пользователя есть атрибуты uid (логин), uidNumber (идентификатор безопасности), gidNumber (идентификатор безопасности первичной группы), userPassword (хеш пароля для LDAP-аутентификации), krbPrincipalKey (ключи для Kerberos-аутентификации), ipaNTHash (хеш пароля для NTLM-аутентификации).

- cn=groups,cn=accounts,суффикс_домена — содержит записи групп пользователей. Список участников группы задается множественным атрибутом member. В учетных записях пользователей эта информация кэшируется в атрибуте memberof, что позволяет существенно ускорить получение информации об участии пользователя в группах.

  В службе каталогов MS AD у пользователей тоже есть атрибут memberof для ускорения запросов, но он содержит только группы прямого участия, то есть не учитывает механизм наследования участия для вложенных групп. Полным аналогом будет атрибут tokenGroups, который содержит идентификаторы SID всех групп, участником которых является пользователь прямо или опосредованно. Учтите только, что ни один из этих атрибутов не отображается на вкладке *Attribute Editor* в окне свойств пользователя и вы не сможете увидеть атрибут tokenGroups в Apache Directory Studio.

- `cn=computers,cn=accounts,суффикс_домена` — содержит учетные записи компьютеров, которые используют атрибут `krbPrincipalKey` так же как пользователи для хранения Kerberos-ключей, а атрибут `userPassword` в качестве одноразового пароля для присоединения компьютеру к домену без использования учетной записи привилегированного пользователя домена.
- `cn=hostgroups,cn=accounts,суффикс_домена` — содержит записи групп компьютеров, например, ipaservers.
- `cn=orgunits,cn=accounts,суффикс_домена` — содержит записи структурных подразделений ALD Pro, ссылки на которые есть у пользователей, компьютеров и групп, см. атрибут rbtadp.
- `cn=dns,суффикс_домена` — содержит записи службы доменных имен.

Используя Apache Directory Studio, вы можете также экспортировать объекты каталога, см. рис. 2.95:

1. Щелкните по RDN *cn=users* в дереве правой кнопкой мыши.
2. В контекстном меню выберите пункт *Export*.
3. В дополнительном меню выберите нужный формат — *CSV Export*.

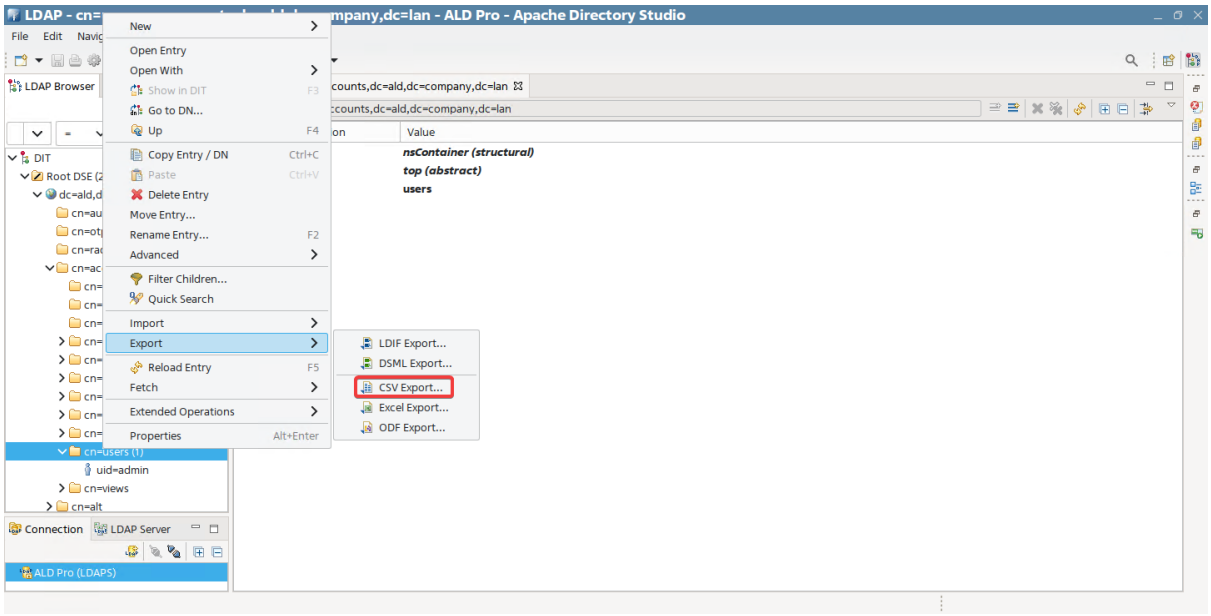


рис. 2.95 – Экспорт объектов в CSV

Просмотр схемы каталога

В программе Apache Directory Studio можно легко посмотреть структуру каталога, которую называют схемой. Выберите Root DSE в дереве LDAP Browser и выполните команду меню *LDAP → Open Schema Browser*, см. рис. 2.96:

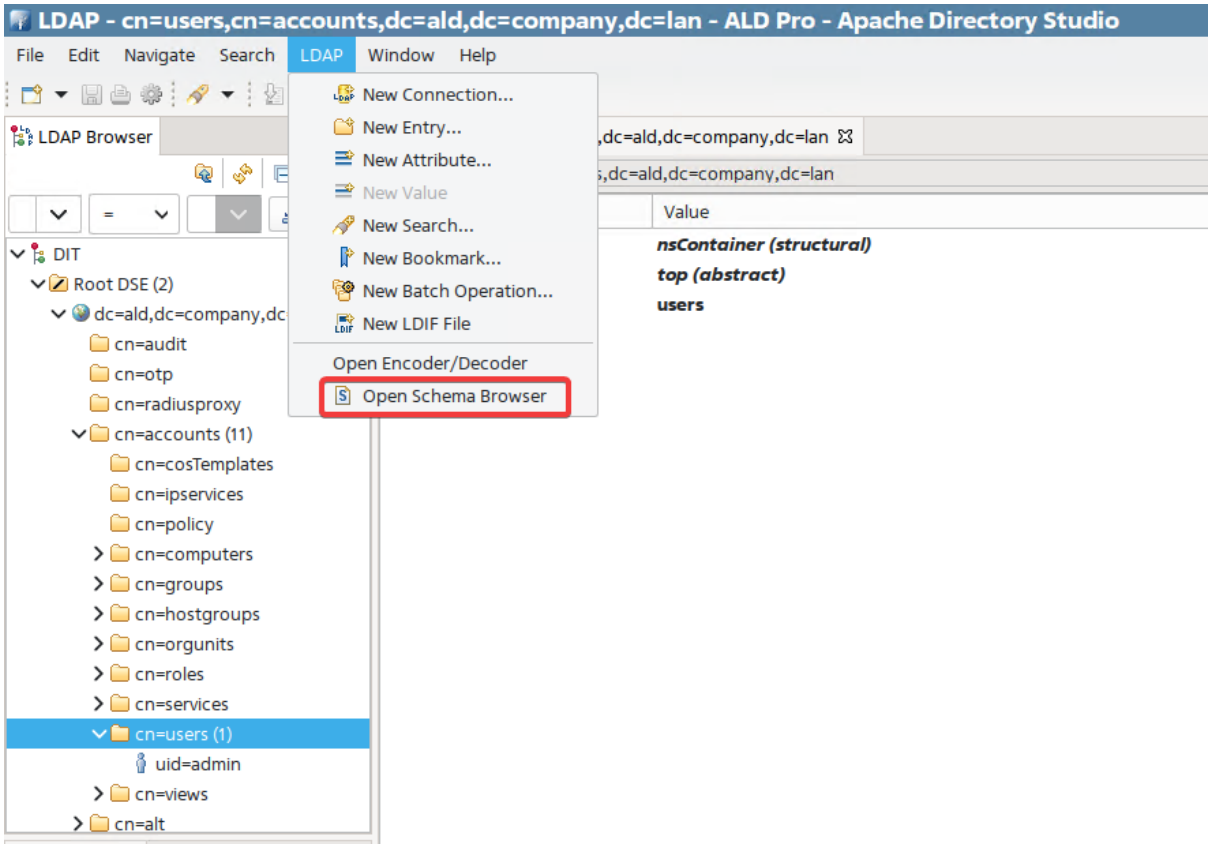


рис. 2.96 – Меню Open Schema Browser

По умолчанию в Schema Browser открывается страница для просмотра классов объектов (Object Classes). Вы можете делать поиск и просматривать детали (Details). Например, введите в фильтр «posix», и вы найдете обсуждаемые ранее классы posixAcccount и posixGroup, как показано на рисунке рис. 2.97:

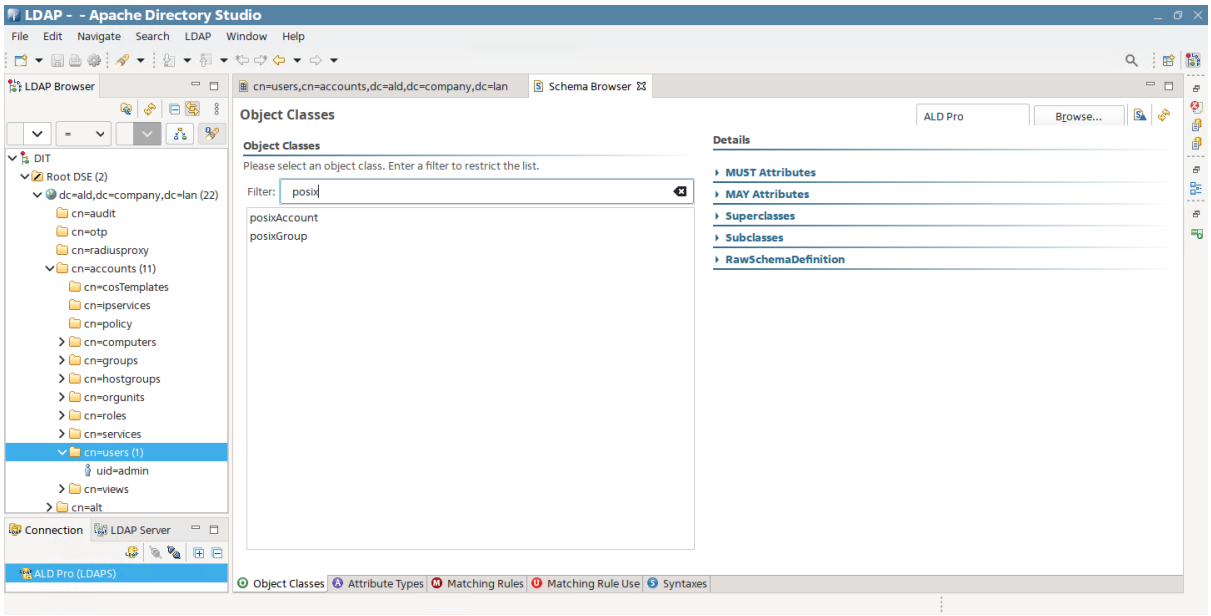


рис. 2.97 – Schema Browser и вкладка Object Classes

Для перехода к другим группам объектов используйте ярлычки в нижней части вкладки. Для просмотра атрибутов откройте страницу :menuselection: Attribute Types . В этом окне можно искать по списку всех атрибутов и просматривать детали выбранного атрибута. Например, введите в поле фильтра «gid», и вы увидите несколько атрибутов, в именах которых содержится эта подстрока, см. рис. 2.98:

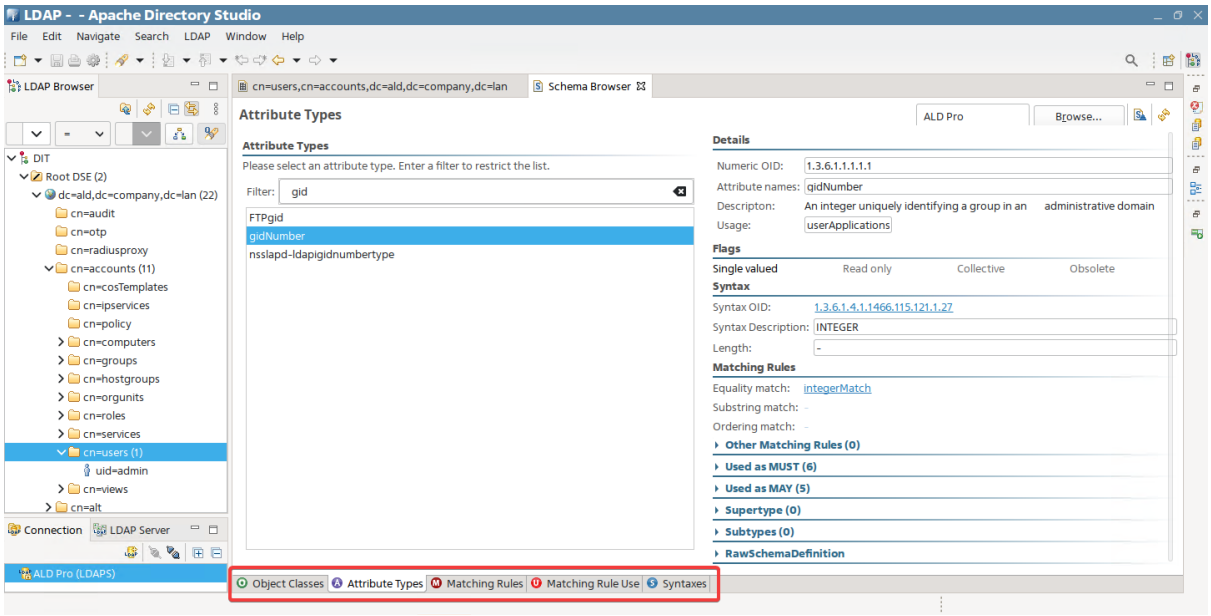


рис. 2.98 – Schema Browser и вкладка Attribute Types

4.4.2.2 Утилиты ldap-utils

Для работы с LDAP-каталогом из командной строки используются утилиты пакета ldap-utils:

- Утилита ldapwhoami — выполняет подключение к каталогу и возвращает имя текущего пользователя.
- Утилита ldapsearch — выполняет поиск по каталогу с указанными параметрами.
- Утилита ldapadd — позволяет создать новые записи в каталоге.
- Утилита ldapdelete — позволяет удалить записи из каталога.
- Утилита ldapmodify — позволяет изменить атрибуты существующей записи в каталоге.
- Утилита ldapcompare — позволяет сравнить текущее значение атрибута конкретной записи с желаемым значением и получить результат в формате TRUE/FALSE.
- Утилита ldapexop — позволяет выполнять расширенные операции, добавленные разработчиками конкретного LDAP-сервера. Расширенные операции подобны хранимым процедурам SQL и позволяют расширять возможности взаимодействия с сервером без внесения изменений

в LDAP-протокол.

- Утилита `ldappasswd` — позволяет сбросить пароль для учетной записи.
- Утилита `ldapmodrdn` — позволяет переименовывать существующие объекты каталога.

Подключение к LDAP-каталогу

Подключение к LDAP-каталогу происходит в рамках каждого вызова указанных выше утилит. Вы можете задавать параметры подключения в явном виде либо полагаться на настройки по умолчанию, описанные в файле `/etc/ldap/ldap.conf`.

```
admin@dc-1:~$ cat /etc/ldap/ldap.conf
...
TLS_CACERT      /etc/ssl/certs/ca-certificates.crt

URI ldaps://dc-1.ald.company.lan
BASE dc=ald,dc=company,dc=lan
SASL_MECH GSSAPI
```

Где:

- `TLS_CACERT` — путь к сертификату для установления безопасного соединения по протоколу LDAPS или для включения шифрования STARTTLS внутри небезопасного соединения.
- `URI` — адрес сервера для подключения по умолчанию.
- `BASE` — отличительное имя базовой записи по умолчанию.
- `SASL_MECH` — механизм для простого уровня аутентификации и безопасности (англ. Simple Authentication and Security Layer, SASL). По умолчанию используется .

При вводе компьютера в домен файл `/etc/ldap/ldap.conf` настраивается автоматически. По умолчанию подключение будет выполняться по протоколу LDAPS с использованием Kerberos-аутентификации через общий программный интерфейс сервисов безопасности (англ. Generic Security Services API, GSSAPI). В качестве сервера для подключения будет использоваться тот контроллер, через который хост был введен в домен.

Удалите кэш из связки ключей и проверьте возможность подключения к каталогу командой `ldapwhoami`:

```
admin@dc-1:~$ kdestroy
admin@dc-1:~$ ldapwhoami
SASL/GSSAPI authentication started
ldap_sasl_interactive_bind_s: Local error (-2)
        additional info: SASL(-1): generic failure: GSSAPI Error: Unspecified GSS␣
↪failure.  Minor code may provide more information (No Kerberos credentials␣
↪available (default cache: KEYRING:persistent:491200000))
```

Ошибка «No Kerberos credentials available» говорит о том, что вы не имеете Kerberos-билета в связке ключей. Выполните Kerberos-аутентификацию и попробуйте запустить команду `ldapwhoami` еще раз:

```
admin@dc-1:~$ kinit admin
Password for admin@ALD.COMPANY.LAN:******
admin@dc-1:~$ ldapwhoami
SASL/GSSAPI authentication started
SASL username: admin@ALD.COMPANY.LAN
SASL SSF: 256
SASL data security layer installed.
dn: uid=admin,cn=users,cn=accounts,dc=ald,dc=company,dc=lan
```

Теперь у нас получилось выполнить запрос к LDAP-серверу, т.к. в связке ключей был валидный TGT-билет, и система смогла с его помощью успешно получить сервисный билет на доступ к LDAP-серверу. Проверить кэш можно командой `klist` без параметров:

```
admin@dc-1:~$ klist
Ticket cache: KEYRING:persistent:491200000:krb_ccache_JxM2PBY
Default principal: admin@ALD.COMPANY.LAN

Valid starting      Expires             Service principal
02.05.2025 09:58:25 03.05.2025 09:58:01 ldap/dc-1.ald.company.lan@ALD.COMPANY.LAN
02.05.2025 09:58:08 03.05.2025 09:58:01 krbtgt/ALD.COMPANY.LAN@ALD.COMPANY.LAN
```

Как вы видите, в кэше появился билет `ldap/dc-1.ald.company.lan`, поэтому вы можете обращаться к LDAP-каталогу с помощью утилит из пакета ldap-utils без необходимости ввода учетных данных.

При подключении к серверу вы можете переопределить любой из параметров по умолчанию, задав его в явном виде:

```
ldapwhoami -H ldap://dc-1.ald.company.lan -ZZ -x -W \
  -D "uid=admin,cn=users,cn=accounts,dc=ald,dc=company,dc=lan"
```

Где:

- Ключ `-H` — позволяет указать хост для подключения и протокол в явном виде, например, `ldaps://dc-1.ald.company.lan`.

  В качестве протокола можно использовать нешифрованный LDAP или шифрованный LDAPS. Вы можете выполнять подключение к LDAP-каталогу также через UNIX-сокет, если утилита запускается на том же сервере. Тогда нужно указать `ldapi://%2fvar%2frun%2fslapd-ALD-COMPANY-LAN.socket`.

- Ключ `-ZZ` — позволяет включить шифрование внутри небезопасного протокола LDAP. Устанавливает требование использовать только шифрованное соединение, поэтому если сервер не поддерживает TLS, то соединение будет разорвано.

  Есть еще ключ `-Z`, который означает необходимость отправки серверу запроса STARTTLS,

но если сервер не поддерживает TLS, обмен продолжится по небезопасному протоколу.

Рекомендуется использовать ключ `-ZZ` или явно указывать протокол LDAPS.

- Ключ `-x` — указывает на необходимость выполнить простую аутентификацию по логину/-паролю вместо SASL.
- Ключ `-W` — указывает, что пароль должен быть предоставлен в интерактивном режиме (а если нужно передать пароль в параметре, то это можно сделать с помощью ключа `-w`).
- Ключ `-D` — задает отличительное имя DN пользовательской записи, которую нужно использовать для аутентификации, например:
  - для администратора домена `uid=admin,cn=users,cn=accounts,dc=ald,dc=company,dc=lan`;
  - для суперпользователя каталога `cn=Directory Manager`.

Поиск по каталогу ldapsearch

Для поиска информации вы можете использовать утилиту ldapsearch, которая извлекает из каталога набор записей с указанными атрибутами в соответствии с заданными критериями фильтрации. Синтаксис команды в общем виде выглядит следующим образом:

`ldapsearch [options] [filter [attributes...]]`

Где:

- `options` — определяют параметры подключения, для простоты мы будем использовать параметры по умолчанию.
- `filter` — задает точные критерии поиска записей.
- `attributes` — список атрибутов, которые нужно извлечь из каталога.

Для примера напишем запрос, который будет выдавать список всех пользователей домена из записи `cn=users`, которая, в свою очередь, является потомком записи `cn=accounts`, и все они находятся в пространстве корневого суффикса `dc=ald,dc=company,dc=lan`. Таким образом, полным уникальным именем записи, в которой хранятся пользователи, будет `cn=users,cn=accounts,dc=ald,dc=company,dc=lan`.

Выполните поиск по каталогу с помощью команды `ldapsearch`:

```
ldapsearch -Q -s sub -b "cn=users,cn=accounts,dc=ald,dc=company,dc=lan" \
  '(uid=*)' uid givenName sn
```

Где:

- Ключ `-Q` — включает тихий режим SASL, в котором не выводится информация о SASL-подключении. Режим полезен при автоматизации для того, чтобы исключить техническую информацию.

  Можно дополнительно указать ключ `-LLL`, который определяет формат вывода: первый ключ `-L` ограничит формат до LDIFv1, второй отключит комментарии, а третий уберет вывод версии LDIF.

- Ключ `-s` — определяет область поиска (*scope*), параметр может принимать следующие значения:

  - `one` — поиск выполняется по дочерним записям на один уровень ниже в иерархии;
  - `sub` — поиск выполняется по всем дочерним записям (значение по умолчанию);
  - `children` — то же, что и sub, но ограничивает поиск только прямыми потомками базовой записи;
  - `base` — ограничивает поиск базовой записью, которая задана параметром `-b`.

- Ключ `-b` — задает базовую запись (*base*), которая будет использоваться в качестве начального узла для поиска по дереву.

- Параметр `'(uid=*)'` — определяет критерий поиска. В приведенном примере запрашиваются все записи, у которых задано значение атрибута `uid`. Фильтры и составные фильтры мы рассмотрим далее.

- Параметры `uid givenName sn` — определяют список атрибутов, которые нужно извлечь из каталога для найденных записей. Если этот список не задать явно, то по умолчанию будут извлечены все хранимые атрибуты, как если бы был указан символ `'*'`. Чтобы извлечь все операционные атрибуты, нужно указать символ `'+'`. Если вы хотите извлечь сразу и хранимые и операционные атрибуты, то нужно указать оба символа подряд `'*' '+'`.

Дополнительно можно использовать ключ `-d` для изменения уровня отладки, предствляющий собой битовую маску от 0 до 65535, в которой каждый бит соответствует определенной категории сообщений отладки:

- 1 — LDAP_DEBUG_TRACE
- 2 — LDAP_DEBUG_PACKETS
- 4 — LDAP_DEBUG_ARGS
- 8 — LDAP_DEBUG_CONNS
- 16 — LDAP_DEBUG_BER

- 32 — LDAP_DEBUG_FILTER
- 64 — LDAP_DEBUG_CONFIG
- 128 — LDAP_DEBUG_ACL
- 256 — LDAP_DEBUG_STATS
- 512 — LDAP_DEBUG_STATS2
- 1024 — LDAP_DEBUG_SHELL
- 2048 — LDAP_DEBUG_PARSE
- 16384 — LDAP_DEBUG_SYNC
- 32768 — LDAP_DEBUG_NONE

После выполнения запроса вы получите следующий результат:

```
# extended LDIF
#
# LDAPv3
# base <cn=users,cn=accounts,dc=ald,dc=company,dc=lan> with scope subtree
# filter: (uid=*)
# requesting: uid givenName sn
#

# admin, users, accounts, ald.company.lan
dn: uid=admin,cn=users,cn=accounts,dc=ald,dc=company,dc=lan
uid: admin
sn: Administrator

# iivanov, users, accounts, ald.company.lan
dn: uid=iivanov,cn=users,cn=accounts,dc=ald,dc=company,dc=lan
uid: iivanov
givenName:: 0JjQstCw0LO=
sn:: 0JjQstCw0L3QvtCy

# ppetrov, users, accounts, ald.company.lan
dn: uid=ppetrov,cn=users,cn=accounts,dc=ald,dc=company,dc=lan
uid: ppetrov
givenName:: 0J/QtdGC0YA=
sn:: 0J/QtdGC0YDQvtCy

#...

# search result
search: 4
result: 0 Success

# numResponses: 8
# numEntries: 7
```

Результат выполнения запроса выводится в поток stdout в формате LDIF, и его можно передавать по конвейеру (pipeline) или перенаправить в файл для последующей обработки. Подробнее о формате LDIF мы поговорим далее.

Как уже упоминалось ранее, кроме обычных хранимых атрибутов, есть еще служебные атрибуты, которые встроены в LDAP-сервер и называются операционными атрибутами. Большинство операционных атрибутов доступны только для чтения, и, чтобы их увидеть, необходимо указать `'+'` в списке атрибутов. Например, с помощью следующего запроса вы можете увидеть, что запись `cn=users` имеет операционный атрибут `numSubordinates`, в котором содержится число дочерних записей:

```
ldapsearch -Q -LLL -s base \
  -b "cn=users,cn=accounts,dc=ald,dc=company,dc=lan" '+'
```

Результат выполнения команды:

```
  admin@dc-1:~$ ldapsearch -Q -LLL -s base \
  >  -b "cn=users,cn=accounts,dc=ald,dc=company,dc=lan" '+'
  dn: cn=users,cn=accounts,dc=ald,dc=company,dc=lan
  creatorsName: cn=Directory Manager
  modifiersName: uid=admin,cn=users,cn=accounts,dc=ald,dc=company,dc=lan
  createTimestamp: 20250428191035Z
  modifyTimestamp: 20250428192224Z
  nsUniqueId: 7662c78e-246411f0-81d3c33d-ecd7a769
  parentid: 2
  entryid: 3
  entryUUID: b969ca8f-78f1-43f7-bf07-662a779383bf
  entryusn: 11700
  numSubordinates: 7
  ...
```

О том, как составлять фильтры, добавлять записи, изменять атрибуты, вы сможете узнать в нашем расширенном курсе «Автоматизация задач администрирования через LDAP-запросы».

4.4.2.3 Утилита управления сервисом dsctl

Как мы уже говорили, для запуска, остановки и перезагрузки служб FreeIPA следует использовать утилиту `ipactl`. Но для управления отдельными экземплярами LDAP-каталога вам нужно будет воспользоваться утилитой `dsctl`. С ее помощью вы сможете:

- Запускать, перезапускать, останавливать, получать статус и уничтожать экземпляр каталога (команды `start`, `restart`, `stop`, `status`, `remove`).
- Управлять базой данных: выполнять переиндексацию базы данных, резервное копирование и восстановление, выгружать дамп в LDIF-файл и восстанавливать каталог из дампа, выполнять проврку целостности каталога (`db2index`, `db2bak`, `bak2db`, `db2ldif`, `ldifdb2`, `dbverify`).

- Производить проверку здоровья экземпляра (healthcheck).
- И многое другое.

Произведем проверку здоровья нашего экземпляра ALD-COMPANY-LAN :

```
admin@dc-1:~$ sudo dsctl ALD-COMPANY-LAN healthcheck
Beginning lint report, this could take a while ...
Checking config:hr_timestamp ...
Checking config:passwordscheme ...
Checking backends:changelog:cl_trimming ...
Checking backends:changelog:mappingtree ...
...
No issues found.
```

Сравним db2bak и db2ldif . Обе команды для создания резервной копии требуют остановки службы каталога. Файлы по умолчанию будут сохранены в подкаталогах /bak/ и /ldif/ директории /var/lib/dirsrv/slapd-ALD-COMPANY-LAN/ . Принципиальное отличие между командами только в формате бэкапа: команда db2bak создает полную резервную, в которую попадают в том числе бинарные файлы индексов, а команда db2ldif создает один ldif-файл просто с данными.

Создадим резервную копию и дамп:

```
admin@dc-1:~$ sudo ipactl stop
Stopping ipa-dnskeysyncd Service
...
ipa: INFO: The ipactl command was successful

admin@dc-1:~$ sudo dsctl ALD-COMPANY-LAN db2bak
...
db2bak successful

admin@dc-1:~$ sudo dsctl ALD-COMPANY-LAN db2ldif userRoot
...
db2ldif successful

admin@dc-1:~$ sudo ipactl start
...
ipa: INFO: The ipactl command was successful
```

Проверим созданные файлы:

```
admin@dc-1:~$ sudo ls -l /var/lib/dirsrv/slapd-ALD-COMPANY-LAN/bak
итого 4
drwx------ 4 dirsrv dirsrv 4096 ср  2 11:23 ALD-COMPANY-LAN-2025_04_02_11_23_51

admin@dc-1:~$ sudo ls -l /var/lib/dirsrv/slapd-ALD-COMPANY-LAN/ldif
итого 6104
-rw------- 1 dirsrv dirsrv 3120670 ноя  2 11:35 ALD-COMPANY-LAN-userroot-2025_04_02_
↪11_35_21.ldif
```

4.4.2.4 Утилита настройки и мониторинга dsconf

Утилита dsconf — это мощное средство для настройки и мониторинга сервера 389 Directory Server, у которого только справка под 7 тысяч строк, см. man dsconf. Мы не будем подробно разбирать все возможности этой утилиты, но покажем пару полезных приемов.

Резервное копирование

Утилита dsconf в отличии от dsctl позволяет выполнять резервное копирование без остановки службы каталога:

```
admin@dc-1:~$ sudo dsconf ALD-COMPANY-LAN backup create
The backup create task has finished successfully
```

Проверим созданный нами файл:

```
admin@dc-1:~$ sudo ls -l /var/lib/dirsrv/slapd-ALD-COMPANY-LAN/bak
итого 8
drwx------ 4 dirsrv dirsrv 4096 ср  2 11:23 ALD-COMPANY-LAN-2025_11_02_11_23_51
drwx------ 4 dirsrv dirsrv 4096 ср  2 11:48 ALD-COMPANY-LAN-2025_11_02_11_48_56
```

Просмотр и изменение настроек сервера 389ds

Для просмотра настроек сервера можно воспользоваться командой config get. Настроек довольно много, поэтому выведем на экран только те из них, которые относятся к управлению паролями:

```
admin@dc-1:~$ sudo dsconf ALD-COMPANY-LAN config get | grep password
nsslapd-allow-hashed-passwords: on
passwordAdminDN:
passwordBadWords:
passwordChange: on
passwordCheckSyntax: off
passwordDictCheck: off
passwordDictPath:
passwordExp: off
passwordGraceLimit: 0
passwordInHistory: 6
passwordIsGlobalPolicy: off
passwordLegacyPolicy: on
passwordLockout: off
passwordLockoutDuration: 3600
passwordMaxAge: 8640000
passwordMaxClassChars: 0
passwordMaxFailure: 3
passwordMaxRepeats: 0
passwordMaxSeqSets: 0
passwordMaxSequence: 0
passwordMin8bit: 0
passwordMinAge: 0
passwordMinAlphas: 0
```

(continues on next page)

(продолжение с предыдущей страницы)

```
passwordMinCategories: 3
passwordMinDigits: 0
passwordMinLength: 8
passwordMinLowers: 0
passwordMinSpecials: 0
passwordMinTokenLength: 3
passwordMinUppers: 0
passwordMustChange: off
passwordPalindrome: off
passwordResetFailureCount: 600
passwordSendExpiringTime: off
passwordStorageScheme: PBKDF2-SHA512
passwordTPRDelayExpireAt: -1
passwordTPRDelayValidFrom: -1
passwordTPRMaxUse: -1
passwordTrackUpdateTime: off
passwordUnlock: on
passwordUserAttributes:
passwordWarning: 86400
```

Следует учитывать, что политики паролей в службе каталога FreeIPA реализованы собственными средствами, поэтому не все параметры 389 Directory Server будут работать так, как ожидается. Например, фильтром запрещенных слов `passwordBadWords` воспользоваться у вас не получится, но эту функцию для FreeIPA команда ALD Pro планирует адаптировать уже в одном из следующих релизов.

Давайте изменим схему хранения паролей `passwordStorageScheme` с безопасного, но ресурсоемкого алгоритма PBKDF2-SHA512 на алгоритм попроще и побыстрее SSHA512, чтобы существенно увеличить скорость LDAP Bind операции.

Текущее значение настройки можно посмотреть следующей командой:

```
admin@dc-1:~$ sudo dsconf ALD-COMPANY-LAN config get passwordStorageScheme
passwordStorageScheme: PBKDF2-SHA512
```

Установить новое значение можно следующей командой:

```
admin@dc-1:~$ sudo dsconf ALD-COMPANY-LAN config replace passwordStorageScheme=SSHA512
selinux is disabled, will not relabel ports or files.
Successfully replaced "passwordStorageScheme"
```

С помощью следующей команды вы можете убедиться, что изменение схемы хранения не повлияло на пароль пользователя `admin`, который был установлен ранее:

```
ldapsearch -H ldaps://dc-1.ald.company.lan \
  -o ldif-wrap=no -D 'cn=Directory Manager' -W \
  -b 'uid=admin,cn=users,cn=accounts,dc=ald,dc=company,dc=lan' \
  | grep 'userPassword::' | cut -d" " -f 2 | base64  -d
```

Результат выполнения:

```
{PBKDF2-SHA512}10000$Dtq6lpYFcu1uLuUMs3lYFswkAermaOln
↪$KRCzVfS3IUIv7zN+jvlbEe7DIXLDfsShEod4H5FADXDfZUv1UUBVqCpTxSzlXlMoMg5LHzlbWShOrTONdh1ZaA==
```

Смените пароль пользователю, например, с помощью утилиты `kpasswd`, повторите выполнение команды и убедитесь, что способ хеширования действительно изменился на SSHA512:

```
{SSHA512}3vQIJJDlA5mmj/
↪ouIFjY1uuyHcuiDuQdiNTaEt9thneV5ZK6seJAWdaibAZpfcmeB63poARbQTRanckARmYzEOU5FB3YWFNu
```

Кроме команд `get` и `replace` доступны также `add` и `delete`. Дополнительную справку по командам можно получить с помощью ключа `-h`:

```
admin@dc-1:~$ sudo dsconf ALD-COMPANY-LAN config -h
usage: dsconf instance config [-h] {get,add,replace,delete} ...

positional arguments:
{get,add,replace,delete}
                      action
  get                 get
  add                 Add attribute value to configuration
  replace             Replace attribute value in configuration
  delete              Delete attribute value in configuration
```

4.5 Сервер MIT Kerberos krb5-kdc

Основным протоколом аутентификации в службе каталога FreeIPA является Kerberos V5, работу которого обеспечивает эталонная реализация центра распространения ключей (*англ. Key Distribution Center, KDC*) от Массачусетского технологического института (агл. Massachusetts Institute of Technology, MIT).

Когда пользователь делает вход в операционную систему доменного компьютера или выполняет аутентификацию утилитой `kinit`, запрос отправляется службе KDC. При успешной проверке аутентичности клиенту выдается TGT-билет, с помощью которого он может в дальнейшем проходить прозрачную аутентификацию на любом контроллере домена без повторного ввода пароля, чтобы получать сервисные билеты для аутентификации в керберизированных сервисах.

В основе протокола Kerberos лежит **идея доверия**:

- Контроллер доверяет клиенту, т.к. они оба владеют общим ключом, в роли которого выступает пароль пользователя.
- Сервис доверяет контроллеру, т.к. они оба владеют общим ключом, в роли которого выступает пароль сервиса.
- Сервис транзитивно доверяет клиенту, т.к. ему доверяет контроллер, которому доверяет сам сервис.

В дальнейшем это доверие может быть расширено на взаимодействие между доменами или, как говорят, областями действия Kerberos, тогда мы получаем следующую схему:

- Контроллер из домена А доверяет клиенту из домена А, т.к. они оба владеют общим ключом, в роли которого выступает пароль пользователя.
- Сервис из домена Б доверяет контроллеру из домена Б, т.к. они оба владеют общим ключом, в роли которого выступает пароль сервиса.
- Сервис из домена Б доверяет клиенту из домена А, т.к. оба домена владеют общим ключом, в роли которого выступает пароль объекта доверительных отношений, который создается в момент установления доверительных отношений между доменами.

Запутанно? А в лесу доменов MS AD может быть вообще несколько уровней доменов, между которыми устанавливаются транзитивные доверительные отношения, тогда возникает целая цепочка рекурсивных Kerberos-запросов, как в детской песенке:

> *Вот дом, который построил Джек.*
>
> *А это пшеница, которая в тёмном чулане хранится в доме, который построил Джек.*
>
> *А это весёлая птица-синица, которая часто ворует пшеницу, которая в тёмном чулане хранится в доме, который построил Джек.*
>
> ...

Центр распространения ключей MIT KDC тесно интегрирован с LDAP-каталогом с помощью плагина ipa-kdb, который позволяет извлекать из службы каталога не только аутентификационную информацию, но и поддерживает генерацию авторизационной структуры данных MS-PAC с информацией об участии пользователя в группах, что требуется для интеграции с MS AD.

В архитектуре KDC иногда выделяют две отдельные службы:

- Сервер аутентификации (Authentication Server, AS), который отвечает за аутентификацию пользователей и выдачу TGT-билетов, поэтому должен иметь доступ к паролям пользователей.
- Сервер выдачи разрешений (Ticket Granting Server, TGS), который отвечает за выдачу сервисных TGS-билетов, поэтому должен иметь доступ к паролям сервисов.

Однако на практике обе службы обычно реализованы вместе под общим названием Центра распространения ключей (KDC) и такие детали можно считать несущественными.

Если не вдаваться в детали, которых будет предостаточно в следующем модуле, то процедура аутентификации по протоколу Kerberos состоит из следующих шагов, см. рис. 2.99:

1. Клиент отправляет контроллеру запрос на аутентификацию (AS_REQ).

2. Контроллер выдает клиенту TGT-билет (AS_REP). Билет зашифрован мастер-ключом KDC, поэтому может быть проверен любым контроллером в домене.

3. Клиент отправляет контроллеру запрос на получение сервисного билета для аутентификации, например, на файловом сервере (TGS_REQ), прилагая полученный ранее TGT-билет. Запрос может быть обработан любым контроллером в домене, т.к. все они располагают мастер-ключом, которым зашифрован TGT-билет.

4. Контроллер выдает клиенту TGS-билет (TGS_REP). Билет зашифрован паролем сервиса, поэтому может быть проверен сервисом самостоятельно без обращения к контроллеру.

5. Клиент обращается к сервису (AP_REQ), прилагая полученный ранее TGS-билет. Запрос может быть обработан сервисом, т.к. он располагает своим ключом, которым зашифрован TGS-билет.

6. Сервис подтверждает аутентичность пользователя и устанавливает с ним сессию (AP_REP).

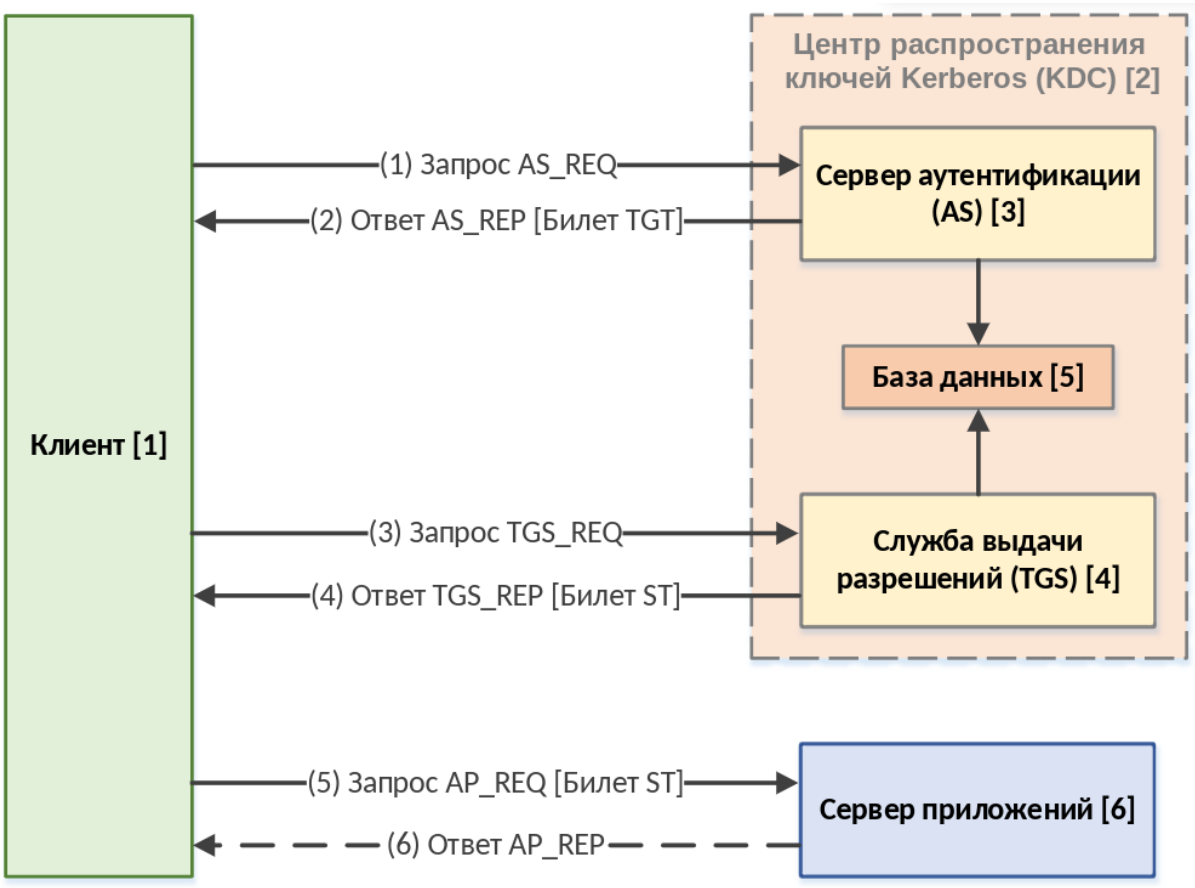


рис. 2.99 – Взаимодействие клиента с компонентами KDC в ходе аутентификации по протоколу Kerberos V5

Как вы можете заметить, ответственность за передачу билетов по протоколу Kerberos V5 возла-

гается на клиента, а для проверки аутентичности пользователя сервису не требуется обращаться к контроллеру домена. Керберизированные сервисы могут, конечно, обращаться к контроллеру по протоколу LDAP для извлечения авторизационной информации об участии пользователя в группах, но при использовании MS-PAC даже этого не требуется — пользователь приходит с билетом, в котором уже содержится вся необходимая информация.

Подробно Kerberos-аутентификацию мы рассмотрим в следующем модуле, а сейчас давайте сведем все новые технические термины в один список:

- **Аутентификация** — процедура проверки подлинности, т.е аутентичности пользователя, например, на основании того, что обеим сторонам известен общий секрет.

- **Авторизация** — процедура предоставления пользователю прав доступа на выполнение определённых действий. Обычно права доступа назначаются не на конкретных пользователей, а на группы, поэтому информацию об участии пользователей в группах считают авторизационной информацией.

- **Область действия Kerberos** (*англ. Realm*) — совокупность пользователей и компьютеров, пароли которых хранятся в единой базе данных. Совпадает с DNS-именем домена, но пишется заглавными буквами, например, `ALD.COMPANY.LAN`.

- **Имя принципала** (*англ. Principal*) — это строка, уникально идентифицирующая учетную запись пользователя, компьютера или керберизированной службы в домене. Принадлежность принципала к области Kerberos задается как в адресе электронной почты с помощью символа собачки `@`, например, `admin@ALD.COMPANY.LAN`.

- **Имя сервиса** (*англ. Service Principal Name, SPN*) — это имя принципала для сервиса, которое уникально идентифицирует учетную запись сервиса в домене. В именах сервисов может указываться имя хоста, на котором эта служба работает в домене, например, `HTTP/dc-1.ald.company.lan@ALD.COMPANY.LAN`.

- **TGT-билет (Ticket-Granting Ticket)** — билет, который позволяет клиенту проходить прозрачную аутентификацию на контроллерах домена без ввода пароля для получения сервисных билетов, которые необходимы для аутентификации в керберизированных сервисах. Называется также билетом для получения других билетов, удостоверением или мандатом.

- **TGS-билет (Ticket-Granting Service Ticket)** — билет, который позволяет клиенту проходить прозрачную аутентификацию без ввода пароля в керберизированном сервисе, для которого этот билет был выдан. Называется так же сервисным билетом, удостоверением или мандатом сервиса.

- **AS_REQ/AS_REP** – запрос к серверу аутентификации (Authentication Server, AS) центра распространения ключей Kerberos для получения TGT-билета и ответ от него соответственно.

- **TGS_REQ/TGS_REP** – запрос к серверу выдачи разрешений (Ticket Granting Server, TGS) центра распространения ключей Kerberos для получения TGS-билета и ответ от него соответственно.

Как мы уже сказали, более подробно процесс Kerberos-аутентификации мы рассмотрим в следующем модуле, см. *Модуль 5. Работа Linux-компьютера в домене*.

4.6 Сервер Samba

Разработчики компании Red Hat изначально работали над развитием службы каталога Samba AD и только когда поняли, что, подстраиваясь под Windows, хорошую службу каталога для Linux не сделать, запустили проект FreeIPA. В то же время, когда им потребовалось реализовать интеграцию с MS AD, они задействовали свои старые наработки, поэтому в контроллер FreeIPA интегрирована служба Samba AD, которая обеспечивает NTLM-аутентификацию и обслуживает SMB RPC вызовы.

Протокол SMB позволяет выполнять RPC-вызовы, поэтому с его помощью можно предоставить доступ не только к файлам и принтерам, но и выполнять другие типы запросов, которые представляются как доступ к «файлам» через общий ресурс `IPC$`. Существует несколько интерфейсов, которые предоставляются контроллерами домена, наиболее известными из которых являются локальная служба безопасности (*англ. Local Security Authority, LSA*) и служба аутентификации пользователей NETLOGON.

В момент установления доверительных отношений с MS AD контроллер FreeIPA через SMB RPC согласовывает общий секрет и другие настройки. В дальнейшем Samba участвует в формировании структуры данных MS-PAC для Kerberos-билетов, в которой содержится информация об участии пользователя в группах.

Удаленные вызовы LSA используются также для преобразования имен. Например, когда вы в Windows открываете свойства папки, в списке доступа ACL хранятся только идентификаторы безопасности SID, но они автоматически преобразуются в имена пользователей через обращение к контроллеру по SMB протоколу на порт 445/TCP. Если контроллер будет в этот момент недоступен, то в списке так и останутся только числа.

На рис. рис. 2.100 показана схема взаимодействия компонентов при обработке запросов из домена Active Directory при интеграции через механизм доверительных отношений.

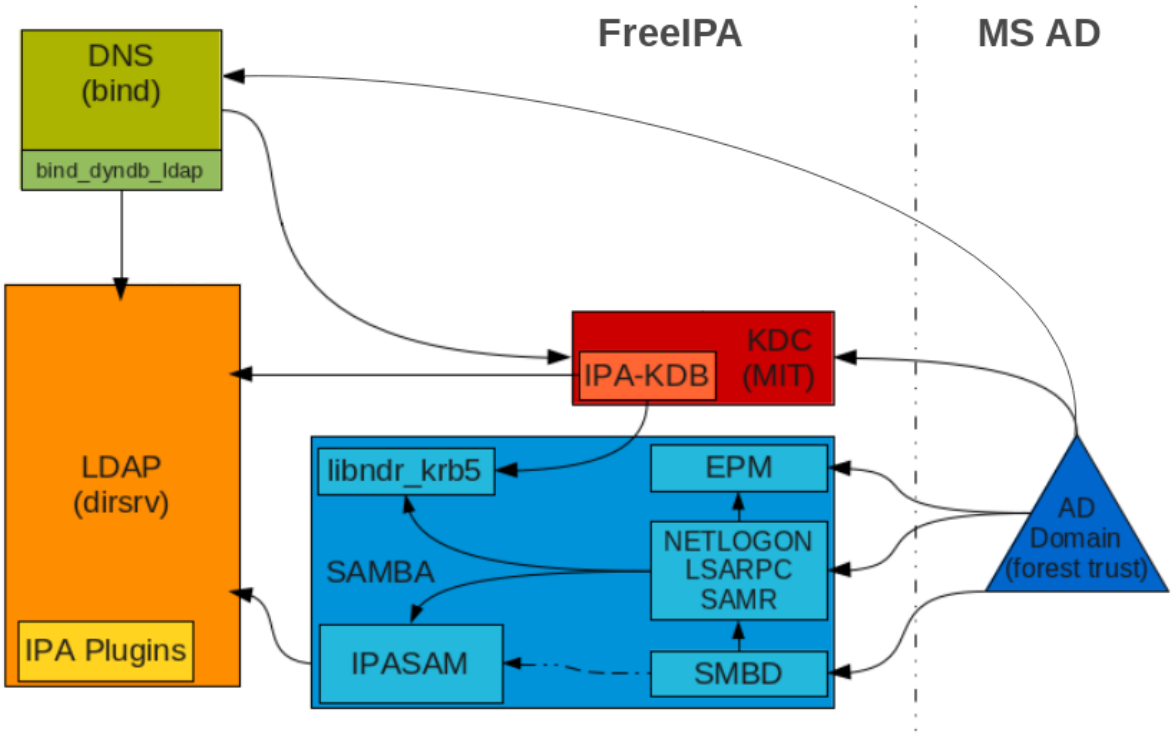


рис. 2.100 – Схема взаимодействия компонентов FreeIPA при работе в доверительных отношениях с доменом Active Directory

Приведем краткое описание назначения компонентов:

- **DNS (bind)** — служба, которая обрабатывает DNS-запросы (порты 53/UDP и 53/TCP). Взаимодействие между доменами осуществляется в первую очередь по доменным именам.
- **KDC (MIT)** — служба, которая обрабатывает запросы на Kerberos-аутентификацию (порт 88/TCP).
- **IPA-KDB** - плагин к MIT KDC от разработчиков FreeIPA для получения информации из LDAP-каталога с сопоставлением атрибутов и записей по схеме Kerberos.
- **EPM (Endpoint Mapper)** - служба, которая предоставляет клиентам RPC-сервис для определения параметров дальнейшего подключения к другим RPC-сервисам, таким как NETLOGON, LSA, SAMR (порт 135/TCP).
- **NETLOGON/LSARPC/SAMR** — сервисы RPC для обработки запросов клиентов на NTLMSSP-аутентификацию, преобразование идентификаторов и другие операции (динамические порты).

  В преобразовании идентификаторов участвует также служба **winbind**, которая является клиентской частью Samba и обеспечивате работу Linux-клиента в домене. В службе каталога FreeIPA большую часть этих задач для Samba выполняет уже собственная служба sssd, с

которой winbind взаимодействует через плагины.

- **IPASAM** — плагин к Samba от разработчиков FreeIPA для получения информации из LDAP-каталога с сопоставлением атрибутов по схеме Active Directory.
- **libndr_krb5** - библиотека для формирования структуры данных MS-PAC с авторизационной информацией об участии пользователя в группах.
- **SMBD** — служба, которая предоставляет клиентам сервисы по протоколам SMB на порту 445/TCP. Клиенты могут выполнять RPC-запросы по протоколу SMB через обращение к общему ресурсу IPC$.

Более подробно мы обсудим роль компонентов Samba в модуле, который будет посвящен интеграции с Microsoft Active Directory.

4.7 Идентификаторы безопасности

4.7.1 Диапазоны идентификаторов

Мы уже знаем, что для идентификации субъектов безопасности в Linux используются идентификаторы POSIX, а в Windows эту роль выполняют идентификаторы SID (*англ. Security Identifiers*). Однако, в то время как в Windows идентификаторы пользователей и групп находятся в одном пространстве SID, идентификаторы POSIX представляют собой два непересекающихся множества: идентификаторы пользователей находятся в пространстве UID (*англ. User Identifier*), а идентификаторы групп в пространстве GID (*англ. Group Identifier*) соответственно.

Если взять SID с номером S-1-5-21-1491017894-2377586105-2170988794-500, то по нему никак не понять, принадлежит ли он пользователю или группе, если только вы не знаете хорошо известные идентификаторы (*англ. well-known SIDs*), в соответствии с которыми идентификатор 500 всегда назначается администратору.

Если разложить SID на составляющие, то последнее число 500 однозначно идентифицирует субъект относительно домена, поэтому это число называется относительным идентификатором (*англ. Relative Identifier, RID*), а предшествующие ему три числа 1491017894-2377586105-2170988794 идентифицируют домен, поэтому для всех SID в домене эти числа имеют одно и то же значение.

Идентификаторы POSIX, так же как RID'ы, представляют собой натуральные числа в диапазоне от 0 до 2^{32} (4 млрд), но не имеют уникальной доменной части, поэтому получается, что все компьютеры работают как бы в одном общем пространстве идентификаторов. Чтобы решить обозначенную проблему, в FreeIPA используют диапазоны идентификаторов (*англ. ID Range*), определяющие часть общего пространства идентификаторов, которые будут использоваться конкретным доменом.

Половину диапазона из 4 млрд служба каталога FreeIPA резервирует на подчиненные (subordinate) идентификаторы, и мы не будем их затрагивать, т.к. они не используются в повседневных задачах. Вторую половину служба использует для обычных пользователей и групп, но делает это крайне экономно. При установке первого контроллера система случайным образом выбирает себе ID Range емкостью 200к идентификаторов, поэтому вероятность пересечения с другим доменом будет крайне мала, всего 1 к 10 тысячам. А если 200к вам будет мало, вы сможете создать себе дополнительный диапазон любой емкости.

Как вы знаете, в MS AD одному из контроллеров назначается FSMO-роль «RID Master», после чего он получает в свое управление весь диапазон свободных идентификаторов и выдает их другим контроллерам пачками по 500 штук. Как только у кого-то из контроллеров свободных идентификаторов становится меньше 100 единиц, этот контроллер обращается к Мастеру RID, и тот выдает ему следующую пачку из 500 идентификаторов.

В службе каталога FreeIPA любой контроллер может стать Мастером RID, для этого достаточно с помощью этого контроллера создать пользователя или группу, чтобы ему потребовался идентификатор безопасности. В этом случае контроллер обратится к соседу, заберет у него половину свободного диапазона и сможет использовать эти идентификаторы для своих нужд и передавать другим контроллерам в домене по запросу.

За работу этого механизма отвечает плагин распределенного назначения идентификаторов (*англ. Distributed Numeric Assignment, DNA*), а часть диапазона ID Range, которым владеет конкретный контроллер домена, называется DNA ID Range. Таким образом, в домене FreeIPA мы работаем с двумя типами диапазонов идентификаторов:

- **ID Range** — диапазон идентификаторов, который зарезервирован за доменом в целом.
- **DNA ID Range** — диапазон распределенного назначения идентификаторов, который является подмножеством диапазона ID Range и закреплен за конкретным контроллером домена, чтобы тот мог создавать пользователей и группы автономно.

Для совместимости с Active Directory в FreeIPA всем пользователям и группам назначаются также идентификаторы SID в стиле Windows. Их значения рассчитываются математически. Сложность этих расчетов состоит в том, что в Linux мы имеем дело с двумя пространствами идентификаторов, а в Windows пространство общее. Поэтому для идентификаторов групп, которые могут совпадать с идентификаторами пользователей, резервируется дополнительно еще один диапазон RID, который называется вторичным, см. рис. 2.101.

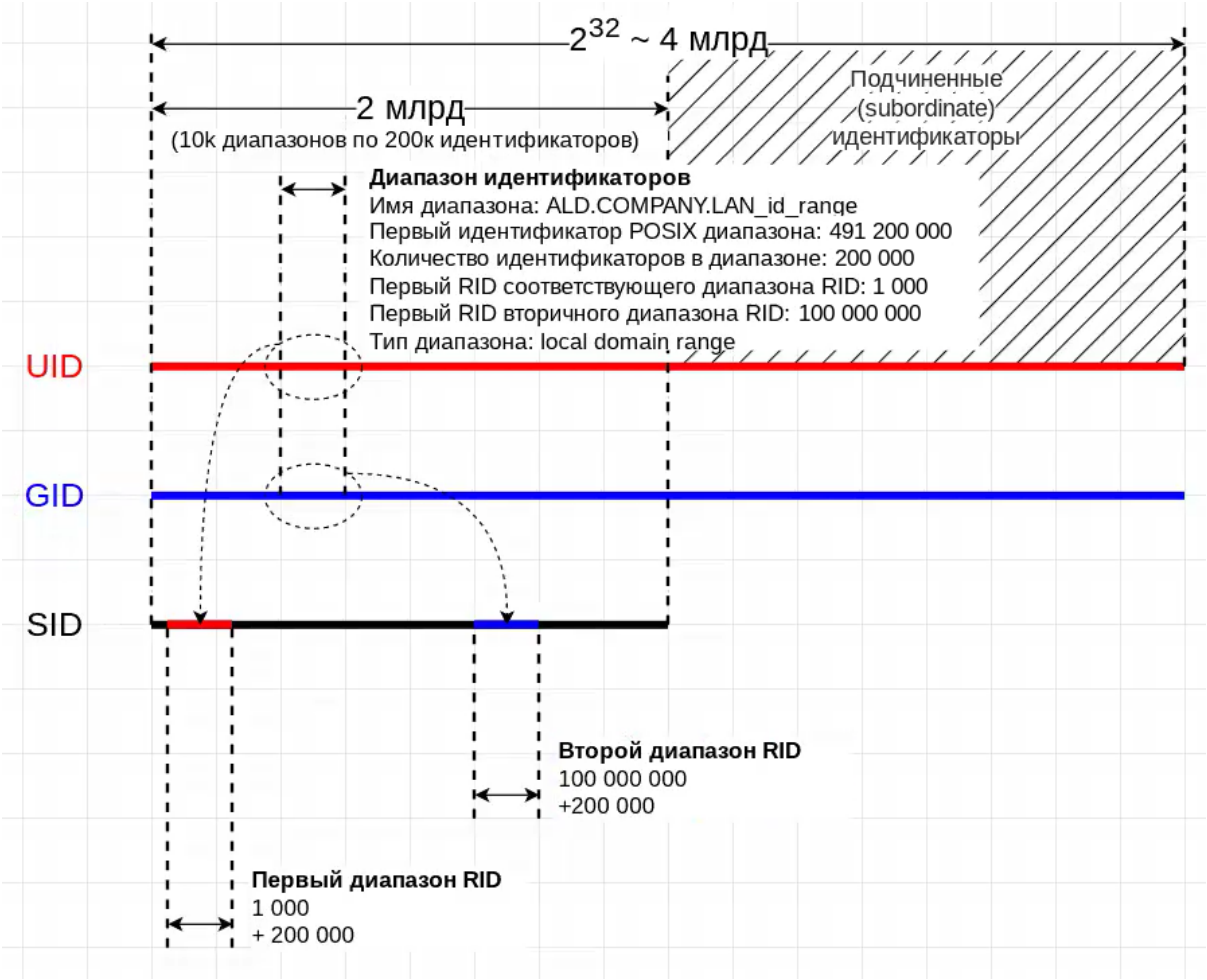


рис. 2.101 – Преобразование POSIX-идентификаторов (UID и GID) в идентификаторы безопасности Windows (SID)

Примерно так же работает пересчет идентификаторов SID из доверенного домена MS AD в идентификаторы POSIX. В момент установления доверительных отношений в домене FreeIPA для каждого домена из леса MS AD создается ID Range, настройки которого используются для пересчета. По умолчанию емкость этих диапазонов тоже 200к идентификаторов, но, в отличие от обычного ID Range, диапазон для доверенного домена MS AD можно расширить.

Примечание: Если в лесу MS AD после установления доверительных отношений с доменом FreeIPA будут созданы новые поддомены, то на стороне FreeIPA нужно будет повторить процедуру извлечения информации о поддоменах следующей командой:

```
ipa trust-fetch-domains <имя_корневого_домена_AD>
```

Это позволит создать для новых поддоменов диапазоны ID Range и сделает их доступными для клиентов службы каталога FreeIPA.

4.7.1.1 Просмотр диапазонов идентификаторов

Посмотреть текущие диапазоны идентификаторов домена (ID Range) можно как из веб-интерфейса ALD Pro, так и через утилиту командной строки ipa.

Для просмотра через веб-интерфейс необходимо открыть страницу *Управление доменом → Пользователи и группы → POSIX идентификаторы*. Как вы помните, существующий диапазон невозможно расширить, можно только добавить новый.

https://dc-1.ald.company.lan/ad/ui/#/domainmgmt/users-and-group-settings/id-ranges

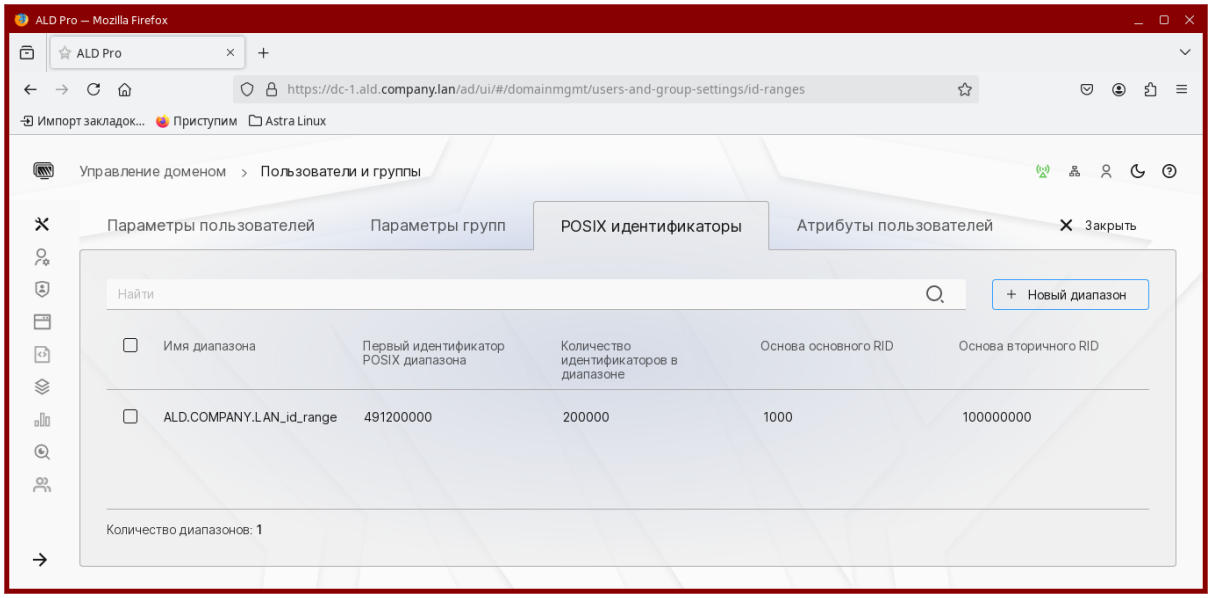


рис. 2.102 – Настройка диапазонов в веб-интерфейсе

Список всех диапазонов идентификаторов домена и их параметры можно посмотреть из окна терминала, используя команду `ipa idrange-find`:

```
admin@dc-2:~$ ipa idrange-find
-----------------
найден 1 диапазон
-----------------
  Имя диапазона: ALD.COMPANY.LAN_id_range
  Первый идентификатор POSIX диапазона: 491200000
  Количество идентификаторов в диапазоне: 200000
  Первый RID соответствующего диапазона RID: 1000
  Первый RID вторичного диапазона RID: 100000000
  Тип диапазона: local domain range
---------------------------------
Количество возвращённых записей 1
---------------------------------
```

Чтобы вывести параметры определенного диапазона идентификаторов домена, следует использовать команду `ipa idrange-show <Имя_Диапазона>`:

```
admin@dc-2:~$ ipa idrange-show ALD.COMPANY.LAN_id_range
  Имя диапазона: ALD.COMPANY.LAN_id_range
  Первый идентификатор POSIX диапазона: 491200000
  Количество идентификаторов в диапазоне: 200000
  Первый RID соответствующего диапазона RID: 1000
  Первый RID вторичного диапазона RID: 100000000
  Тип диапазона: local domain range
```

Просмотреть список диапазонов распределенного назначения идентификаторов (DNA ID Range) на всех контроллерах домена можно командой `sudo ipa-replica-manage dnarange-show`:

```
admin@dc-1:~$ sudo ipa-replica-manage dnarange-show
dc-1.ald.company.lan: 491200009-491399999
dc-2.ald.company.lan: No range set
...
```

Вывести список диапазонов только на конкретном контроллере можно так:

```
admin@dc-1:~$ sudo ipa-replica-manage dnarange-show dc-1.ald.company.lan
dc-1.ald.company.lan: 491200009-491399999
```

4.7.1.2 Выделение диапазона идентификатора вручную

В некоторых случаях вы можете захотеть создать еще один диапазон идентификаторов домена в дополнение к исходному. Например, когда исходный диапазон идентификаторов уже исчерпан. Для этого нужно выполнить следующие команды:

1. Создадим новый диапазон идентификаторов командой `sudo ipa idrange-add` на dc-1:

```
ipa idrange-add ALD.COMPANY.LAN_second_id_range \
  --base-id=1000000 --range-size=200000 --rid-base 201000 \
  --secondary-rid-base 100200000
```

Результат выполнения команды

```
admin@dc-1:~$ ipa idrange-add ALD.COMPANY.LAN_second_id_range \
>   --base-id=1000000 --range-size=200000 --rid-base 201000 \
>   --secondary-rid-base 100200000
---------------------------------------------------------------------
Добавлен диапазон идентификаторов "ALD.COMPANY.LAN_second_id_range"
---------------------------------------------------------------------
  Имя диапазона: ALD.COMPANY.LAN_second_id_range
  Первый идентификатор POSIX диапазона: 1000000
  Количество идентификаторов в диапазоне: 200000
  Первый RID соответствующего диапазона RID: 201000
  Первый RID вторичного диапазона RID: 100200000
  Тип диапазона: local domain range
```

2. Проверим доступные диапазоны идентификаторов домена:

```
admin@dc-1:~$ ipa idrange-find
------------------
найдено 2 диапазона
------------------
  Имя диапазона: ALD.COMPANY.LAN_id_range
  Первый идентификатор POSIX диапазона: 491200000
  Количество идентификаторов в диапазоне: 200000
  Первый RID соответствующего диапазона RID: 1000
  Первый RID вторичного диапазона RID: 100000000
  Тип диапазона: local domain range

  Имя диапазона: ALD.COMPANY.LAN_second_id_range
  Первый идентификатор POSIX диапазона: 1000000
  Количество идентификаторов в диапазоне: 200000
  Первый RID соответствующего диапазона RID: 201000
  Первый RID вторичного диапазона RID: 100200000
  Тип диапазона: local domain range
--------------------------------
Количество возвращённых записей 2
--------------------------------
```

При создании диапазона идентификаторов ID Range соответствующий ему динамический диапазон DNA ID Range не создается автоматически, поэтому его нужно будет создать вручную. Для управления диапазонами DNA ID Range предназначены команды утилиты `ipa-replica-manage`.

Текущие настройки диапазона можно посмотреть командой `dnarange-show`:

```
admin@dc-1:~$ sudo ipa-replica-manage dnarange-show dc-1.ald.company.lan
dc-1.ald.company.lan: 491200009-491399999
```

Давайте назначим dc-1 новый диапазон, для этого воспользуемся командой `dnarange-set`:

```
admin@dc-1:~$ sudo ipa-replica-manage dnarange-set dc-1.ald.company.lan 1000000-
↪1199999
```

У реплик может быть не только основной, но и дополнительный диапазон идентификаторов DNA ID Range, который они начнут использовать, когда в текущем диапазоне закончатся свободные идентификаторы. Давайте в качестве следующего диапазона назначим оставшиеся идентификаторы из диапазона, который был создан при установке системы. Чтобы назначить следующий диапазон идентификаторов DNA конкретному серверу, используйте команду `dnanextrange-set`.

```
admin@dc-1:~$ sudo ipa-replica-manage dnanextrange-set dc-1.ald.company.lan 491200009-
↪491399999
```

Посмотрим, что у нас получилось:

```
admin@dc-1:~$ sudo ipa-replica-manage dnarange-show dc-1.ald.company.lan
dc-1.ald.company.lan: 1000000-1199999
admin@dc-1:~$ sudo ipa-replica-manage dnanextrange-show dc-1.ald.company.lan
dc-1.ald.company.lan: 491200009-491399999
```

Примечание: Как правило, служба sssd может работать с новым диапазоном службы каталога FreeIPA сразу без выполнения дополнительных действий. Если у вас появятся какие-либо проблемы с новыми пользователями, то выполните очистку кэша службы sssd на проблемном компьютере командой `sudo sssctl cache-remove` (потребуется установить пакет `apt install sssd-tools`).

4.8 Практика и тестирование

4.8.1 Практическая работа: Модуль 4. Работа службы каталога ALD Pro

См. также *Требования, правила и цели выполнения практической работы*

4.8.1.1 Практические задания

Задача 1.

- Используя утилиту `ipactl`, получите статус сервисов FreeIPA.
- Используя утилиту `aldproctl`, получите статус сервисов ALD Pro.

Задача 2.

Создайте резервную копию данных FreeIPA с помощью утилиты `ipa-backup`.

Задача 3.

Используя портал управления ALD Pro, создайте нового пользователя *testadmin* и добавьте его в группу администраторов `admins`.

Задача 4.

Сгенерируйте keytab-файл для только что созданного пользователя `testadmin` с помощью утилиты `ipa-getkeytab`.

Задача 5.

С помощью keytab-файла и команды `user-status` утилиты `ipa` получите статус пользователя `admin`.

Задача 6.

Используя утилиту `dsctl`, создайте отчет `healthcheck` в обычном формате и JSON.

Задача 7.

Используя `dsconf`, получите статистику по использованию сервера.

4.8.1.2 Ответы на практические задания (пошаговые инструкции)

Задача 1.

- Используя утилиту `ipactl`, получите статус сервисов FreeIPA сервера:

```
  sudo ipactl status

Результат выполнения команды:
```

```
admin@dc-1:~$ sudo ipactl status
Directory Service: RUNNING
krb5kdc Service: RUNNING
kadmin Service: RUNNING
named Service: RUNNING
httpd Service: RUNNING
ipa-custodia Service: RUNNING
smb Service: RUNNING
winbind Service: RUNNING
ipa-otpd Service: RUNNING
ipa-dnskeysyncd Service: RUNNING
ipa: INFO: The ipactl command was successful
```

- Используя утилиту `aldproctl`, получите статус сервисов ALD Pro сервера:

```
  sudo aldproctl status

Результат выполнения команды:
```

```
admin@dc-1:~$ sudo aldproctl status
.....Сервисы ALD Pro.....
Сервис aldpro-mp-services: ЗАПУЩЕН
Сервис globalcatalog: ЗАПУЩЕН
Сервис ipa-gcsyncd: ЗАПУЩЕН
.....Сервисы FreeIPA.....
Сервис Directory Service: ЗАПУЩЕН
Сервис krb5kdc: ЗАПУЩЕН
Сервис kadmin: ЗАПУЩЕН
Сервис named: ЗАПУЩЕН
Сервис httpd: ЗАПУЩЕН
Сервис ipa-custodia: ЗАПУЩЕН
Сервис smb: ЗАПУЩЕН
Сервис winbind: ЗАПУЩЕН
```

(continues on next page)

(продолжение с предыдущей страницы)

```
Сервис ipa-otpd: ЗАПУЩЕН
Сервис ipa-dnskeysyncd: ЗАПУЩЕН
.....Другие сервисы.....
Сервис celery: ЗАПУЩЕН
Сервис celerybeat: ЗАПУЩЕН
```

Задача 2.

Создайте резервную копию данных FreeIPA с помощью утилиты `ipa-backup`:

```
sudo ipa-backup
```

Результат выполнения команды:

```
admin@dc-1:~$ sudo ipa-backup
Preparing backup on tdc01.ald.lan
Local roles match globally used roles, proceeding.
Stopping IPA services
Backing up userRoot in ALD-LAN to LDIF
Backing up ALD-LAN
Backing up files
Starting IPA service
Backed up to /var/lib/ipa/backup/ipa-full-2025-04-02-16-54-32
The ipa-backup command was successful
```

Задача 3.

Используя портал управления ALD Pro, создайте нового пользователя `testadmin` и добавьте его в группу администраторов `admins`.

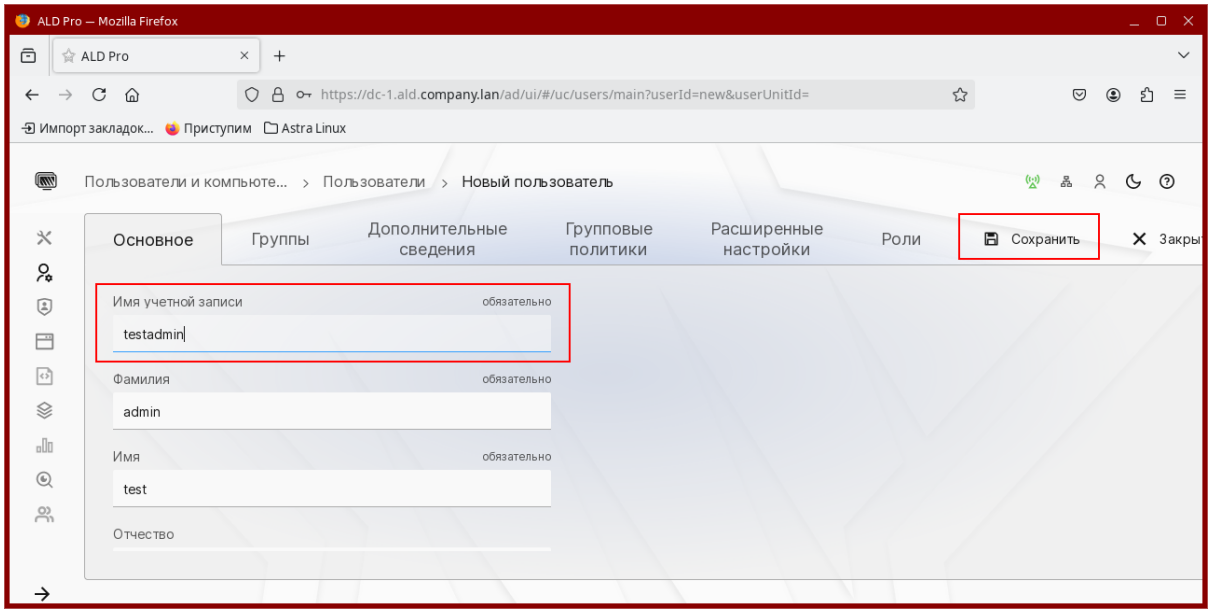


рис. 2.103 – Создание пользователя testadmin

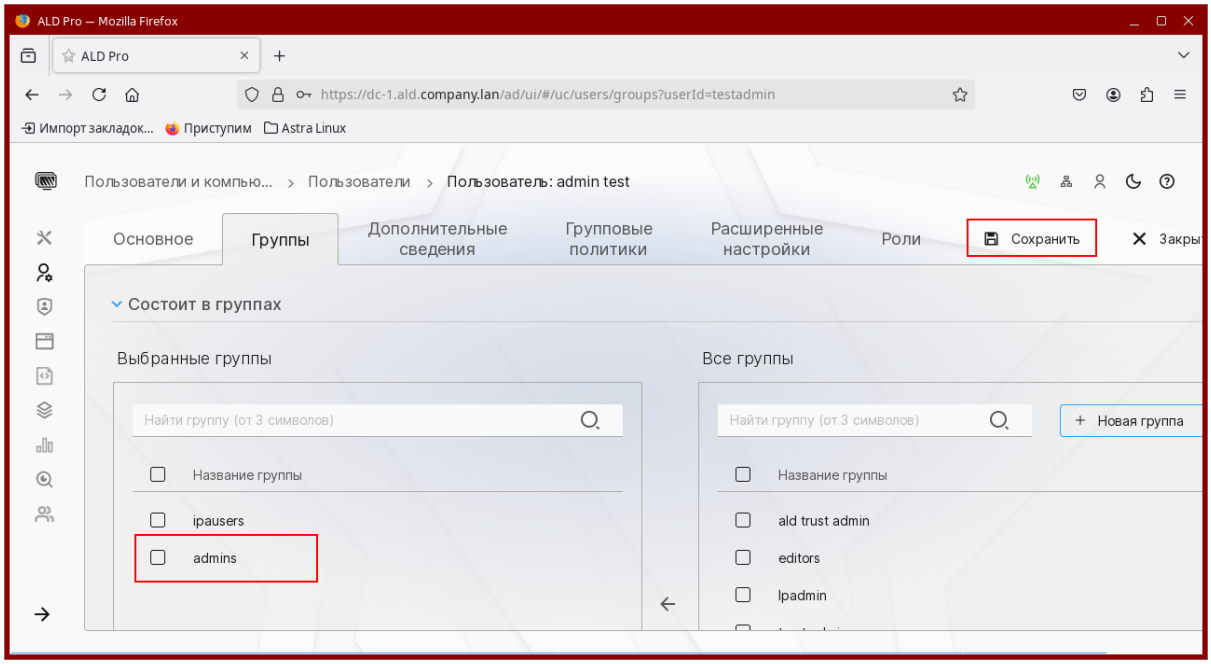


рис. 2.104 – Добавление пользователя в группу admins

Задача 4.

Получите keytab для только что созданного пользователя `testadmin` с помощью утилиты `ipa-getkeytab`:

```
kinit admin

ipa user-mod testadmin \
  --password-expiration="$(date +'%Y%m%d000000Z' -d '+90 days')"

sudo ipa-getkeytab -s dc-1.ald.company.lan -p testadmin -k /tmp/testadmin.keytab
```

Результат выполнения команды:

```
admin@dc-1:~$ kinit admin
Password for admin@ALD.COMPANY.LAN:

admin@dc-1:~$ ipa user-mod testadmin \
>       --password-expiration="$(date +'%Y%m%d000000Z' -d '+90 days')"
--------------------------------
Изменён пользователь "testadmin"
--------------------------------
...

admin@dc-1:~$ sudo ipa-getkeytab -s dc-1.ald.company.lan -p testadmin -k /tmp/
↪testadmin.keytab
Таблица ключей успешно получена и сохранена в: /tmp/testadmin.keytab
```

Задача 5.

С помощью keytab-файла и команды `user-status` утилиты `ipa` получите статус пользователя `admin`.

```
sudo kinit testadmin -kt /tmp/testadmin.keytab
ipa user-status admin
```

Результат выполнения команд:

```
admin@dc-1:~$ sudo kinit testadmin -kt /tmp/testadmin.keytab

admin@dc-1:~$ ipa user-status admin
-----------------------
Account disabled: False
-----------------------
Сервер: dc-1.ald.company.lan
Неудачные попытки входа: 0
Последняя успешная аутентификация: N/A
Последняя ошибка при аутентификации: N/A
Время сейчас: 2025-05-09T10:40:14Z
--------------------------------
Количество возвращённых записей 1
--------------------------------
```

Задача 6

Используя утилиту `dsctl`, создайте отчет `healthcheck` в обычном формате и JSON:

```
sudo dsctl --json slapd-ALD-COMPANY-LAN healthcheck
```

Результат выполнения команды:

```
admin@dc-1:~$ sudo dsctl --json slapd-ALD-COMPANY-LAN healthcheck
Enter password for cn=Directory Manager on ldap://dc-1.ald.company.lan:
[]
```

Пустые квадратные скобки означают что нет проблем, проверим это командой:

```
sudo dsctl slapd-ALD-COMPANY-LAN healthcheck
```

Результат выполнения команды:

```
admin@dc-1:~$ sudo dsctl slapd-ALD-COMPANY-LAN healthcheck
Enter password for cn=Directory Manager on ldap://dc-1.ald.company.lan:
Beginning lint report, this could take a while ...
Checking config:hr_timestamp ...
Checking config:passwordscheme ...
Checking backends:changelog:cl_trimming ...
...
Checking logs:notes ...
```

(continues on next page)

(продолжение с предыдущей страницы)

```
Healthcheck complete.
No issues found.
```

Задача 7

Используя dsconf, получите статистику по использованию сервера.

```
sudo dsconf slapd-ALD-COMPANY-LAN monitor server
```

Результат выполнения команды:

```
admin@dc-1:~$ sudo dsconf slapd-ALD-COMPANY-LAN monitor server
Enter password for cn=Directory Manager on ldap://dc-1.ald.company.lan:
dn: cn=monitor
version: 389-Directory/2.3.2 B2023.038.0907
threads: 17
connection: 6:20240321135430Z:667:667:-:cn=Directory
Manager:0:0:0:6:ip=local
connection: 7:20240321135430Z:757:757:-:cn=Directory
Manager:0:0:0:7:ip=local
connection: 10:20240321135430Z:8:8:-:cn=Directory
Manager:0:0:0:10:ip=local
connection:
...
currenttime: 20240321143733Z
starttime: 20240321135429Z
nbackends: 2
```

4.8.2 Тест: Модуль 4. Работа службы каталога ALD Pro

Пожалуйста, ответьте на несколько вопросов, чтобы закрепить полученные знания. Правильные ответы для самопроверки вы сможете найти в конце списка. Обратите внимание, что у некоторых вопросов может быть несколько верных вариантов ответов.

4.8.2.1 Вопросы

Вопрос 1. Что такое FreeIPA?

a. Система управления идентификацией и доступом

b. Система мониторинга

c. Система резервного копирования

d. Система управления виртуальными машинами

Вопрос 2. Какой порт используется по умолчанию для незащищенного LDAP?

a. 389/TCP

b. 636/TCP

c. 88/TCP

d. 53/UDP

Вопрос 3. Какой компонент FreeIPA является центральным хранилищем данных?

a. Apache HTTP Server

b. 389 Directory Server

c. MIT Kerberos

d. Dogtag Certificate System

Вопросы 4. Что такое ipactl?

a. Утилита для управления FreeIPA сервером

b. Утилита для установки FreeIPA клиента

c. Утилита для просмотра логов FreeIPA

d. Утилита для репликации FreeIPA

Вопрос 5. Что такое DN в LDAP?

a. Уникальный идентификатор записи

b. Атрибут записи

c. Фильтр поиска

d. Индекс

Вопрос 6. Что такое Kerberos realm?

a. Клиент Kerberos

b. Сервер аутентификации Kerberos

c. Область действия Kerberos

d. База данных Kerberos

Вопрос 7. Какой компонент KDC отвечает за выдачу TGT-билетов?

a. Database

b. Ticket-Granting Service

c. Key Distribution Center

d. Authentication Service

Вопрос 8. Что такое keytab в контексте Kerberos?

a. Файл с ключами для аутентификации служб

b. База данных Kerberos

c. Журнал событий Kerberos

d. Конфигурационный файл Kerberos

Вопрос 9. Что такое базовый DN (Base DN) в LDAP?

a. Фильтр поиска

b. Уникальный идентификатор записи

c. Атрибут записи

d. Корневой уровень иерархии каталога

Вопрос 10. Что такое схема (schema) в LDAP?

a. Фильтр поиска

b. Атрибут записи

c. Определение структуры данных

d. Модуль для хранения данных

Ответы на вопросы

1. a
2. a
3. b
4. a

5. a
6. c
7. d
8. a
9. d
10. c

4.9 Заключение

В этом модуле мы познакомились с составом службы каталога ALD Pro, рассмотрели ее компоненты и научились с ними работать. Мы узнали, что такое идентификаторы безопасности и диапазоны идентификаторов, и в общих чертах посмотрели на процесс Kerberos-аутентификации.

В следующем модуле мы познакомимся со способами аутентификации и с устройством службы sssd, обеспечивающей работу компьютера в домене.

4.10 Дополнительные источники информации

- Курс «Автоматизация задач администрирования через утилиту ipa»
- Курс «Автоматизация задач администрирования через LDAP-запросы»
- Lightweight Directory Access Protocol (LDAP): String Representation of Distinguished Names RFC 4514;
- Lightweight Directory Access Protocol (LDAP): Syntaxes and Matching Rules RFC 4517.

4.11 Обратная связь

Если остались вопросы, то их всегда можно задать в специальной теме.

Модуль 5. Работа Linux-компьютера в домене

5.1 Введение

Из этого модуля вы узнаете, что такое служба sssd и как работает компьютер Linux в составе домена. Мы подробно рассмотрим работу протокола аутентификации Kerberos, утилит для работы с билетами и доступные журналы, что очень поможет вам в решении любых нештатных ситуаций.

5.2 Служба sssd

5.2.1 Краткий обзор

За работу компьютера в домене отвечает служба sssd, которая была создана в рамках проекта FreeIPA и является на сегодняшний день лучшим решением для обеспечения работы компьютера Linux в составе доменов на базе FreeIPA, MS Active Directory, Samba AD, MIT KDC или даже простого LDAPv3. Однако максимально свои возможности sssd раскрывает, конечно же, когда на бэкенде у нее служба каталога FreeIPA, и продукт ALD Pro построен как раз на базе этой службы каталога, поэтому в конфигурационных файлах `/etc/sssd/sssd.conf` на доменных компьютерах будет указан поставщик «ipa»:

```
admin@pc-1:~$ sudo cat /etc/sssd/sssd.conf | grep id_provider
id_provider = ipa
```

Как рассказывал Александр Боковой, один из ведущих разработчиков FreeIPA, официально аббревиатура sssd расшифровывается как *System Security Services Daemon* (*Демон служб системной безопасности*), но на самом деле эти слова были подобраны под имена четырех основателей: Simo Sorce, Sumit Bose, Stephen Gallagher и Dmitri Pal. Так что еще один наш бывший соотечественник Дмитрий Пал внес огромный вклад в развитие систем идентификации Linux.

Основная задача службы sssd заключается в предоставлении доступа к информации удаленной службы каталога и механизмам централизованной аутентификации. Для обработки запросов на аутентификацию служба sssd на доменных компьютерах встраивает свой модуль **pam_sss**.so в PAM-стек, см. файлы `common-*` из папки `/etc/pam.d`:

```
admin@dc-1:~$ ls /etc/pam.d
chfn                            common-session.pam-old  runuser
chpasswd                        cron                    runuser-l
chsh                            cups                    samba
common-account                  fly-dm                  sshd
common-account.pam-old          fly-dm-greeter          sssd-shadowutils
common-auth                     fly-dm-np               su
common-auth.pam-old             fly-wm                  sudo
common-password                 login                   su-l
```

(continues on next page)

(продолжение с предыдущей страницы)

```
common-password.pam-old                 newusers                sumac.xauth
common-session                          other                   systemd-user
common-session-noninteractive           passwd
common-session-noninteractive.pam-old   polkit-1
```

Для обработки запросов на идентификацию служба sssd встраивает свои модули в NSS-стек, см. файл `/etc/nsswitch.conf`.

```
admin@dc-1:~$ cat /etc/nsswitch.conf
...
passwd: files sss
group: files sss
shadow: files sss
gshadow: files
...
```

Управление службой осуществляется через `systemctl` с помощью стандартного набора команд `start`, `stop`, `restart` и `status`, но не пытайтесь проверять команду `stop` из непривилегированной сессии, поскольку остановить службу вы сможете, а запустить повторно уже нет, т.к. выполнение команд из-под `sudo` вам станет недоступно и придется ждать пару минут, пока служба sssd будет запущена автоматически.

```
admin@pc-1:~$ systemctl status sssd
sssd.service - System Security Services Daemon
   Loaded: loaded (/lib/systemd/system/sssd.service; enabled; vendor preset: enabled)
   Active: active (running) since Apr 2025-04-07 22:38:35 MSK; 15s ago
 Main PID: 3740 (sssd)
    Tasks: 4 (limit: 4597)
   Memory: 10.1M
   CGroup: /system.slice/sssd.service
           ├─3740 /usr/sbin/sssd -i --logger=files
           ├─3741 /usr/lib/x86_64-linux-gnu/sssd/sssd_be --domain ald.company.lan --
↪uid 0 --gid 0 --logger=files
           ├─3742 /usr/lib/x86_64-linux-gnu/sssd/sssd_ifp --uid 0 --gid 0 --
↪logger=files
           └─3743 /usr/lib/x86_64-linux-gnu/sssd/sssd_pac --uid 0 --gid  --
↪logger=files
```

Модули службы sssd и другие клиентские приложения реализуют поддержку протокола Kerberos не напрямую, а через общий программный интерфейс сервисов безопасности (*англ. Generic Security Services API, GSSAPI*). Это позволяет заменять библиотеки, используемые GSS-API, без необходимости переписывать все приложения.

Для поддержки протокола Kerberos V5 через GSSAPI используются библиотеки **libgssapi-krb5**, **librkb5** и др. В соответствии с настройкой `default_ccache_name` из конфигурационного файла `/etc/krb5.conf` для хранения полученных билетов библиотека использует связку ключей Linux (KEYRING). Управлять билетами можно с помощью утилит из пакета krb5-user, таких как `kinit`, `klist`, `kdestroy`, `kvno` и др.

Интерфейс GSSAPI позволяет не только устанавливать безопасный контекст, но и выполнять шифрование сообщений. Поэтому при выполнении LDAP-запросов с аутентификацией по Kerberos данные будут зашифрованы внутри средствами GSSAPI.

5.2.2 Архитектура службы sssd

При первом входе доменного пользователя в систему служба sssd сохраняет информацию о нем в локальном кэше, что ускоряет работу службы и позволяет обеспечить возможность входа в автономном режиме, даже когда нет связи с контроллером домена. Извлечение информации из домена и поддержание кэша в актуальном состоянии является довольно сложной задачей, поэтому sssd состоит из множества компонентов, которые общаются друг с другом при помощи различных методов межпроцессного взаимодействия, см. рис. 2.105.

В состав службы входят следующие компоненты:

- **монитор** (*англ. Monitor*);
- **серверные части** или Бэкенды (*англ. Backends*);
- **ответчики** (*англ. Responders*);
- **клиентские библиотеки и приложения**.

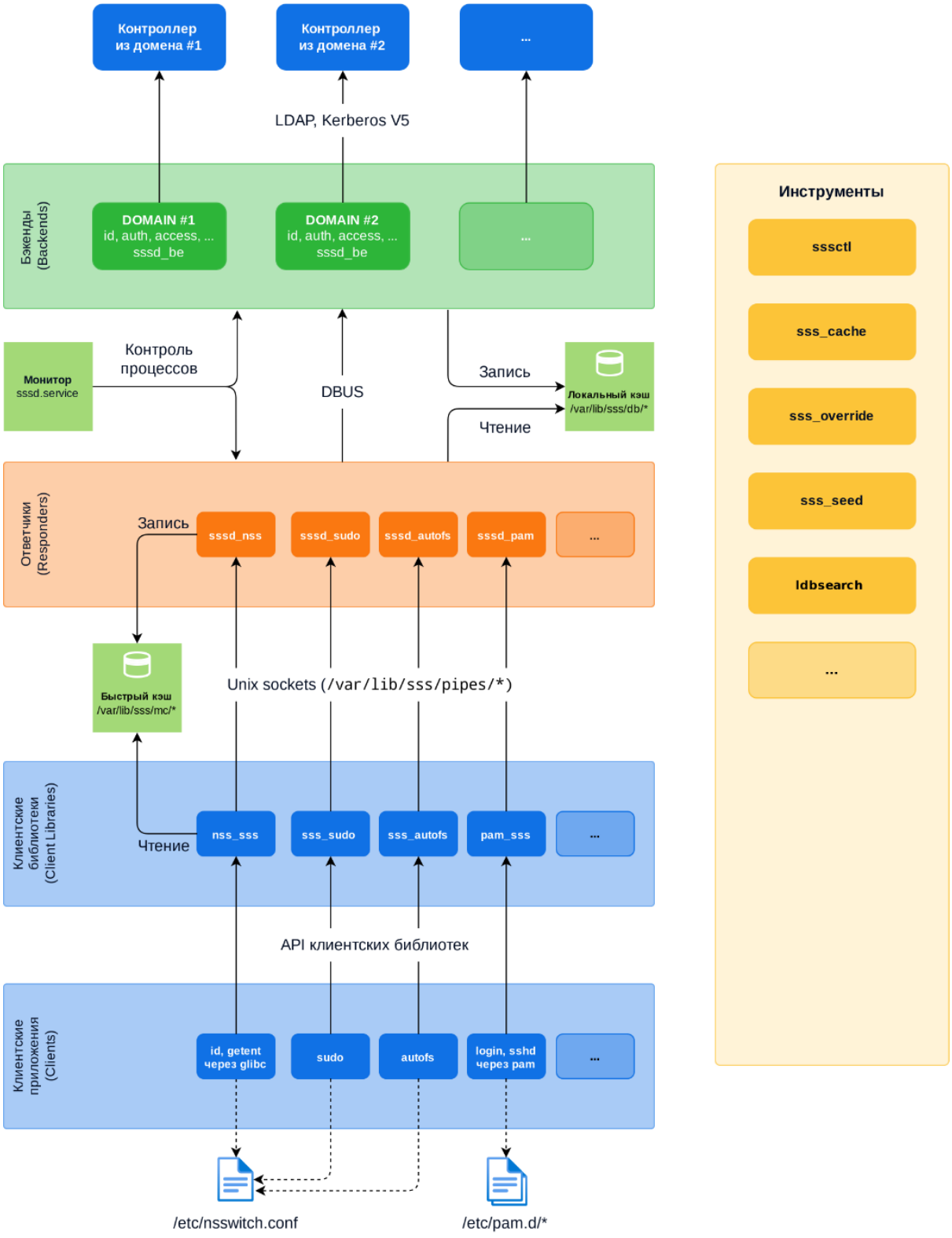


рис. 2.105 – Архитектура службы sssd

5.2.2.1 Монитор (Monitor)

Служба **sssd.service** порождает процессы всех других компонентов и управляет ими, обеспечивая их слаженную работу, поэтому службу называют «Монитором», т.к. в английском языке это слово имеет также значение «Наставник», что довольно точно отражает роль данного компонента.

Монитор взаимодействует с остальными компонентами по протоколу S-Bus, который является собственной реализацией протокола D-Bus, т.е. компоненты не регистрируют себя в системной шине D-Bus и обратиться к ним напрямую не получится. Для возможности управления службой по D-Bus предназначен ответчик IFP (*англ. info pipe*), через него, например, утилита `sssctl` получает информацию об активном контроллере, который обслуживает хост.

На рис. 2.106 представлена диаграмма процесса загрузки службы:

- В момент запуска Монитор анализирует файл конфигурации `sssd.conf` (1) и загружает его в кэш `/var/lib/sss/db/config.ldb` (2).
- Далее Монитор, используя системные вызовы, порождает процесс Бэкенда (3), который загружает информацию из config (4, 5) и регистрирует себя в Мониторе (6).
- На следующем шаге Монитор порождает Ответчиков NSS и PAM (7), которые считывают информацию из config (8, 9) и регистрируются в Мониторе (10) и Бэкенде (11).

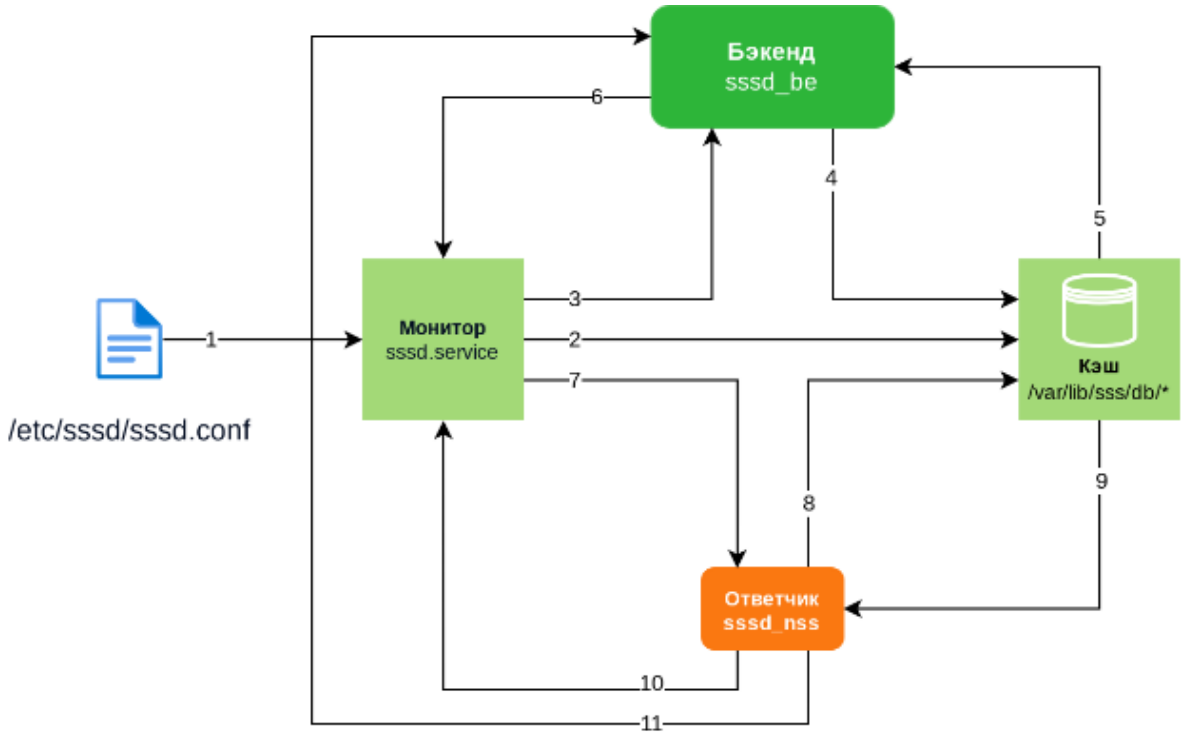


рис. 2.106 – Диаграмма загрузки службы sssd

Помимо внутреннего взаимодействия через SBus, служба sssd использует UNIX-сигналы, напри-

мер, в тех случаях, когда процесс завис и не отвечает через SBus. Обработчики сигналов обычно интегрированы с главным циклом событий **tevent** с использованием вызова **tevent_add_signal**:

- **SIGTERM** — сигнал на завершение работы процесса.
- **SIGKILL** — сигнал для принудительного завершения работы процесса, который направляется монитором, если процесс не завершается после получения SIGTERM.
- **SIGUSR1** / **SIGUSR2** — сигналы на переключение Бэкенда в автономный режим и обратно. Если сигнал будет направлен Монитору, то он перешлет его всем своим дочерним процессам sssd_be.
- **SIGHUP** — может быть передан Монитору, после чего тот начнет новый лог-файл и вызовет метод сброса журналов управляемым процессам, что крайне удобно при отладке. Кроме того, процессы sssd_be перечитают resolv.conf, а процесс sssd_nss выполнит очистку быстрого кэша.

5.2.2.2 Серверные части или Бэкенды (Backends)

Бэкенд (*англ. Backend, BE*) – это процесс приложения `/usr/lib/x86_64-linux-gnu/sssd/sssd_be`, который запускается Монитором для взаимодействия с доменом в соответствии с настройками секции `[domain/<имя_домена>]` конфигурационного файла `/etc/sssd/sssd.conf`. Бэкенд не является монолитом и в зависимости от настроек подключает один и более Поставщиков данных (*англ. Data Provider, DP*), каждый из которых представляет собой совместно используемую библиотеку (*англ. Shared Object, SO*).

```
admin@dc-1:~$ sudo ps -aux | grep sssd_be
[sudo] пароль для admin:
root       724  0.0  0.5 118844 23728 ?        S    12:05   0:00 /usr/lib/x86_64-
↪linux-gnu/sssd/sssd_be --domain ald.company.lan --uid 0 --gid 0 --logger=files
admin     5921  0.0  0.0   6168   812 pts/0    S+   12:16   0:00 grep sssd_be
```

Рассмотрим описание бэкендов в конфигурационном файле командой:

```
admin@pc-1:~$ sudo cat /etc/sssd/sssd.conf
[domain/ald.company.lan]
id_provider = ipa
ipa_server = _srv_, dc-1.ald.company.lan
ipa_domain = ald.company.lan
ipa_hostname = pc-1.ald.company.lan
auth_provider = ipa
chpass_provider = ipa
access_provider = ipa
cache_credentials = True
ldap_tls_cacert = /etc/ipa/ca.crt
krb5_store_password_if_offline = True
...
```

Библиотеки поставщиков службы sssd находятся в каталоге `/usr/lib/x86_64-linux-gnu/sssd/`, и за работу компьютера в домене ALD Pro (FreeIPA), например, отвечает Поставщик **ipa**, которому соответствует библиотека `libsss_ipa.so`. Если же ввести компьютер Linux напрямую в домен Active Directory, то будет использоваться `libsss_ad.so`.

Каждый Поставщик данных может оказывать Бэкенду sssd следующие услуги:

- **Идентификация (id_provider)** — предоставление информации о пользователях, служебных учетных записях, группах и так далее.
- **Аутентификация (auth_provider)** — проверка аутентичности пользователей и служб.
- **Авторизация (access_provider)** — проверка прав доступа на запуск служб на уровне HBAC.
- **Поставщик sudo (sudo_provider)** — предоставление информации о правилах SUDO на повышение привилегий.
- И другие.

Разные модули могут использовать разные настройки, например, для Поставщика `ipa` требуется задать параметр `ipa_server`, который используется для определения контроллера домена FreeIPA, с которым требуется поддерживать соединение, а для Поставщика `ad` аналогичным параметром будет `ad_server`.

По умолчанию в конфигурационном файле `sssd.conf` для домена явно задается только несколько провайдеров (*id*, *auth*, *chpass*, *access*), а для других провайдеров подразумевается, что их имя совпадает с именем поставщика идентификационных данных (*id_provider*). Для удобства из соображений сокращения размера конфигурационного файла служба sssd руководствуется следующими правилами:

- Если параметр `ipa_domain` не указан, то используется полное имя домена из секции `[domain/<fqdn домена>]`.
- Если параметр `ipa_server` не указан, то поиск активного контроллера домена будет выполняться автоматически через DNS, как будто параметр `ipa_server = _srv_`.
- Если провайдер не указан, то используется значение из парамтера `id_provider`. Исключением является `access_provider`. Если его значение не указано, то по умолчанию параметр принимает значение `permit`, что означает «всем пользователям разрешен доступ». Для управления доступом параметр `access_provider` должен быть задан явно.

Таким образом, стандартный файл конфигурации `/etc/sssd/sssd.conf` может быть упрощен до следующего вида:

```
. . .
[domain/ald.company.lan]
id_provider = ipa
access_provider = ipa
cache_credentials = True
ldap_tls_cacert = /etc/ipa/ca.crt
krb5_store_password_if_offline = True
. . .
```

5.2.2.3 Ответчики (Responders)

Ответчик служит посредником между пользовательским приложением (таким как `id` или `getent`), кэшем sssd и бэкендом. Когда приложение запрашивает данные, например, ищет пользователя по имени, Ответчик выполняет поиск в локальном кэше и, если данные не найдены или устарели, обращается к Бэкенду, чтобы тот совершил запрос к серверу службы каталога.

Клиентские библиотеки взаимодействуют с Ответчиком через UNIX-сокет (например, `/var/lib/sss/pipes/nss`), а внутри sssd между Ответчиками, Бэкендами и Монитором взаимодействие осуществляется через DBus.

Каждый Ответчик работает в отдельном процессе и выполняет строго определенные задачи:

- **Ответчик NSS** (sssd_nss) — обеспечивает данными диспетчер службы имен (*англ. Name Service Switch, NSS*).
- **Ответчик PAM** (sssd_pam) — предоставляет данные для работы подключаемых модулей аутентификации (*англ. Pluggable Authentication Modules, PAM*).
- **Ответчик SUDO** (sssd_sudo) — обеспечивает правилами утилиту `sudo`.
- И так далее.

5.2.2.4 Клиентские библиотеки и приложения

Для взаимодействия с Ответчиком приложению нужно обращаться к соответствующему UNIX-сокету по специфичному сетевому протоколу, поэтому для упрощения разработки клиентских приложений были созданы вспомогательные совместно используемые библиотеки, такие как `libnss_sss.so.2`, `pam_sss.so` и др., которые можно найти в каталоге `/usr/lib/x86_64-linux-gnu`.

Например, если в конфигурационном файле `/etc/nsswitch.conf` для базы `sudoers` источником данных будет указан `sss`, то утилита `sudo` для взаимодействия с Ответчиком `sssd_sudo` воспользуется функциями из библиотеки `libsss_sudo.so`. Аналогичным образом работает служба `autofs`.

Однако есть и такие приложения, которые могут даже не знать о существовании sssd, если ис-

пользуют более универсальные функции из библиотек *pam* и *glibc*, которые взаимодействуют с клиентскими библиотеками sssd от имени приложения.

Например, утилиты `id`, `getent`, `ls` вызывают стандартные функции NSS, которые извлекают информацию из нескольких баз данных имен (например, `passwd`, `group`, `service`, `netgroup` и так далее). На рис. 2.107 отражен поток данных, создаваемый клиентским приложением `id`, выполняющим NSS запрос через службу sssd:

- Клиентское приложение `/usr/bin/id` вызывает функцию `getpwnam()` из библиотеки *glibc* для получения информации о пользователе (1).
- Библиотека *glibc* загружает конфигурационный файл `nsswitch.conf` (2, 3), затем в соответствии с его настройками загружает совместно используемую библиотеку `libnss_sss.so.2` и вызывает ее функцию `_nss_sss_getpwnam_r()` (4).
- Для получения данных от Ответчика `/usr/lib/x86_64-linux-gnu/sssd/sssd_nss` библиотека *libnss_sss.so.2* отправляет запрос через UNIX-сокет `/var/lib/sss/pipes/nss` по специализированному протоколу обмена данными (5).
- Ответчик NSS проверяет наличие данных в кэше (6, 7) и в случае отсутствия данных или истекшего срока действия отправляет запрос `getAccountInfo` к Бэкенду, запрашивая необходимые данные (8).
- Бэкенд из-под учетной записи хоста `/etc/krb5.ketyab` направляет LDAP-запрос для получения данных от службы каталога (9). Если запрашиваемый пользователь принадлежит тому же домену, то это будет обычный LDAP-запрос на поиск информации, в противном случае sssd выполнит расширенную LDAP-операцию.

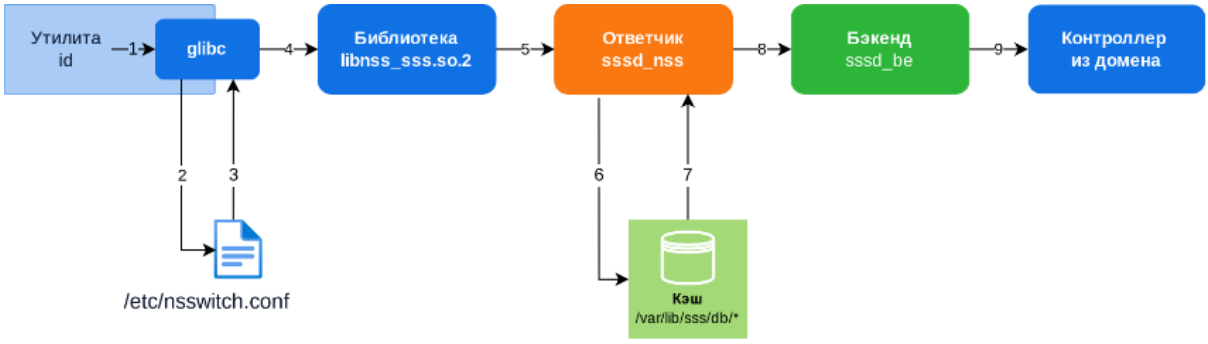


рис. 2.107 – Диаграмма потока данных при вызове команды id

Ответчик PAM работает немного иначе. Локальный кэш будет использоваться только в том случае, если служба находится в режиме офлайн. А если связь с сервером есть, то при входе пользователя в систему Ответчик запросит проверку аутентичности у контроллера и обновит информацию об участии пользователя в группах, даже если срок действия кэша еще не истек. В качестве примеров

клиентских приложений, использующих PAM, можно привести `login`, `su` и `sshd`.

Несмотря на то, что клиентские приложения и библиотеки не являются синонимами, в рабочей документации и программном коде службы sssd под суффиксом sss_cli (*англ. SSSD Client*) часто подразумеваются именно клиентские библиотеки, например, сетевой протокол, с помощью которого клиентские библиотеки общаются с Ответчиками службы sssd.

5.2.3 Механизм кэширования службы sssd

Служба sssd в своей работе использует кэш, что обеспечивает возможность работы в автономном режиме без подключения к серверу и позволяет исключить ненужные обращения к серверу за информацией, когда в этом нет необходимости. Система кэширования довольно сложна и включает в себя несколько взаимодополняющих механизмов.

5.2.3.1 Локальный кэш (local cache, cache)

Локальный кэш – это база данных, в которую Бэкенды записывают результаты, получаемые из службы каталога, чтобы Ответчики могли считывать их напрямую. При помещении объектов в кэш им назначается срок жизни, и до тех пор, пока он не истечет, все последующие запросы к этим данным обрабатываются без обращения к серверу. Устаревшие данные могут использоваться только в том случае, если служба находится в автономном режиме.

Файлы локального кэша находятся на диске в каталоге `/var/lib/sss/db/`. Для работы с этим кэшем служба sssd использует встроенную библиотеку баз данных облегченного доступа (*англ. Light Weight Embedded Database Library, ldb*), и это та же библиотека, с помощью которой реализована база данных службы каталога Samba AD, когда этот набор программного обеспечения используется как самостоятельная служба каталога.

Если установить пакет **ldb-tools**, то вам станет доступна утилита `ldbsearch`, с помощью которой можно заглянуть внутрь ldb-файлов. Структура этой базы данных подобна LDAP-каталогу, а параметры ldbsearch во многом повторяют параметры утилиты ldapsearch.

Установим пакет ldb-tools и выполним поиск:

```
sudo apt install ldb-tools

sudo ldbsearch -H /var/lib/sss/db/cache_ald.company.lan.ldb \
  -b  name=admin@ald.company.lan,cn=users,cn=ald.company.lan,cn=sysdb \
  fullName uidNumber cachedPassword
```

```
asq: Unable to register control with rootdse!
# record 1
dn: name=admin@ald.company.lan,cn=users,cn=ald.company.lan,cn=sysdb
```

(continues on next page)

(продолжение с предыдущей страницы)

```
fullName: Administrator
uidNumber: 491200000
cachedPassword: $6$u4YejIUZ9cxnfqeh$bxH/g8FNJ7gpz/...E/U/fotvnnimCKnjGWsueB/

 # returned 1 records
 # 1 entries
 # 0 referrals
```

В данном запросе ключ `-b` (`--basedn`) задет отличительное имя базовой записи для поиска. По результатам выполнения запроса можно заметить, информация в базе есть, и пароль закэширован, поэтому пользователь может успешно выполнить вход даже в автономном режиме.

Теперь зайдем в сессию супер-пользователя, выполним удаление кэша очисткой каталога `/var/lib/sss/db/`, перезапустим службу sssd и повторим поиск:

```
sudo rm -rf /var/lib/sss/db/*
sudo systemctl restart sssd
sudo ldbsearch -H /var/lib/sss/db/cache_ald.company.lan.ldb \
  -b name=admin@ald.company.lan,cn=users,cn=ald.company.lan,cn=sysdb \
  fullName uidNumber cachedPassword
```

```
asq: Unable to register control with rootdse!
# returned 0 records
# 0 entries
# 0 referrals
```

Как видите, теперь в кэше пусто. Для прогрева кэша мы запросим информацию о пользователе *admin* с помощью утилиты `id`, после чего посмотрим в локальный кэш еще раз:

```
admin@dc-1:~$ id admin
uid=491200000(admin) gid=491200000(admins) группы=491200000(admins),1001(astra-admin),
↪491200003(lpadmin),1005611117(ald trust admin)

admin@dc-1:~$ sudo ldbsearch -H /var/lib/sss/db/cache_ald.company.lan.ldb \
  -b name=admin@ald.company.lan,cn=users,cn=ald.company.lan,cn=sysdb \
  fullName uidNumber cachedPassword
```

Результат поиска в кэше:

```
asq: Unable to register control with rootdse!
# record 1
dn: name=admin@ald.company.lan,cn=users,cn=ald.company.lan,cn=sysdb
fullName: Administrator
uidNumber: 491200000
# returned 1 records
# 1 entries
# 0 referrals
```

Как видите, теперь информация о пользователе на месте, но хеш пароля мы все-таки потеряли, поэтому автономный вход пользователю будет недоступен, пока он хотя бы один раз не войдет в систему по паролю.

На низком уровне ldb использует библиотеку Trivial DataBase, поэтому сырые данные в формате ключ-значение можно извлекать из этих файлов с помощью утилит из пакета **tdb-tools**:

```
admin@pc-1:~$ sudo apt install tdb-tools
...

admin@pc-1:~$ sudo tdbtool /var/lib/sss/db/config.ldb keys
key 18 bytes: DN=@INDEX:CN:SSSD
key 21 bytes: DN=CN=SSSD,CN=CONFIG
key 18 bytes: DN=@INDEX:CN:SUDO
key 17 bytes: DN=@INDEX:CN:PAC
key 13 bytes: DN=CN=CONFIG
. . .
```

В папке `/var/lib/sss/db/` находятся следующие ldb-файлы локального кэша:

- Файл `config.ldb` — кэш для хранения настроек службы, загружаемых из конфигурационного файла `/etc/sssd/sssd.conf`.
- Файл `sssd.ldb` — база данных локального домена (local provider), который позволял использовать возможности вложенных групп без централизованной службы каталога. Для администрирования использовались команды sss_useradd, sss_groupadd и др. Поддержка функции прекращена в sssd с версии 2.0.0, но файл пока еще остался.
- Файл `cache_ald.company.lan.ldb` — кэш для хранения критически важной информации, получаемой из домена. Сохранение таких данных на диск требуется выполнять сразу при получении новых данных.
- Файл `timestamps_ald.company.lan.ldb` — кэш для хранения некритичной и часто обновляемой информации. Сохранение данных на диск выполняется в фоновом режиме по усмотрению операционной системы.

Кроме файлов локального кэша в этой же папке находятся файлы кэша учетных данных Kerberos, которые можно просматривать с помощью утилиты `klist` с ключом `-c`:

- Файл `ccache_ALD.COMPANY.LAN` — кэш для хранения Kerberos-билетов службы sssd, с помощью которых она выполняет LDAP-запросы к службе каталога.
- Файл `fast_ccache_ALD.COMPANY.LAN` — кэш для хранения еще одного TGT-билета, с помощью которого обеспечивается дополнительное шифрование запросов по технологии FAST (*англ. Flexible Authentication Secure Tunneling – гибкая аутентификация через защищенное туннелирование*), которая позволяет исключить передачу по сети посылок, зашифрованных только простыми паролями пользователей. В службе каталога MS AD эту технологию называют также Kerberos Armoring.

Удаление всех файлов из директории db является самым простым, но в то же время и самым

полным способом очистки локального кэша службы sssd:

```
sudo systemctl stop sssd
sudo rm -rf /var/lib/sss/db/*
sudo systemctl start sssd
```

Того же эффекта (но явно безопаснее, чем удаление французского языка `rm -fr`) можно достичь с помощью команды `cache-remove` утилиты `sssctl` из состава пакета sssd-tools:

```
sudo apt install sssd-tools
sudo sssctl cache-remove
```

В ходе очистки кэша вам будет задано несколько вопросов:

```
admin@pc-1:~$ sudo sssctl cache-remove
SSSD must not be running. Stop SSSD now? (yes/no) [yes]
Creating backup of local data...
Removing cache files...
SSSD needs to be running. Start SSSD now? (yes/no) [yes]
```

Однако для отладки sssd в продуктивной среде рекомендуют использовать утилиту `sss_cache` из того же пакета **sssd-tools**. С ее помощью можно не удалять записи из кэша, а отметить их недействительными, что сильно повышает производительность службы, т.к. обновление информации о записях в кэше происходит значительно быстрее, чем их создание.

Вот так можно отметить недействительными все записи в кэше:

```
sudo sss_cache -E
```

С помощью ключа `-u` можно отметить недействительными записи конкретного пользователя, что бывает крайне полезно при выполнении отладки на контроллерах домена в продуктивной среде:

```
sudo sss_cache -u admin
```

5.2.3.2 Быстрый кэш (in-memory cache, memcache)

Ответчик службы sssd является однопоточным приложением и обрабатывает запросы со скоростью одного ядра, поэтому на контроллерах домена он мог бы стать узким местом и ограничить возможности вертикального масштабирования.

В то же время по соображениям безопасности клиентским библиотекам нельзя давать прямой доступ к локальному кэшу, т.к. в нем содержатся в том числе конфиденциальные данные, например, хеши паролей. Поэтому для устранения противоречия был создан дополнительный быстрый кэш, который доступен клиентским библиотекам напрямую, но содержит при этом только общую информацию о пользователях и группах.

Быстрый кэш представляет собой хеш-таблицы, которые находятся в каталоге `/var/lib/sss/mc/`

и отображаются на память (Memory-mapped). Ответчик NSS записывает информацию в быстрый кэш, и до тех пор, пока эти данные остаются актуальными, клиентской библиотеке не требуется связываться с sssd для их извлечения. Если нужная запись будет отсутствовать в кэше или срок ее жизни истек, запрос будет перенаправлен Ответчику NSS, который извлечет данные из локального кэша или обратится к контроллеру домена через Бэкенд.

По умолчанию время хранения записей в быстром кэше составляет 300 секунд, и это значение может быть изменено с помощью параметра `memcache_timeout` в файле `/etc/sssd/sssd.conf`. Для отключения быстрого кэша нужно установить `memcache_timeout=0`, а для очистки, как и в случае локального кэша, просто удалить файлы и перезапустить службу:

```
sudo systemctl stop sssd
sudo rm -rf /var/lib/sss/mc/*
sudo systemctl start sssd
```

Вам следует учитывать следующие два момента:

- Утилита `sss_cache` отмечает недействительными одновременно записи и локального и быстрого кэша.
- Команда `cache-remove` утилиты `sssctl` удаляет только локальный кэш и не влияет на работу быстрого кэша, поэтому физически удалить записи быстрого кэша вы можете только вручную.

5.2.3.3 Негативный или безрезультатный кэш (negative cache, ncache)

Для уменьшения нагрузки на сервер при обращении к несуществующим объектам каталога внутри Ответчика NSS и ряда других компонентов реализован так называемый негативный или безрезультатный кэш.

Этот кэш находится в оперативной памяти и очищается каждые 15 секунд, изменить эту настройку можно с помощью параметра `entry_negative_timeout` в конфигурационном файле `/etc/sssd/sssd.conf`. Для полной очистки негативного кэша достаточно просто перезапустить службу sssd обычной командой:

```
sudo systemctl restart sssd
```

5.2.3.4 Алгоритм использования кэша (cache lookup)

Служба sssd кэширует пользователей, группы, правила HBAC/SUDO, SSH-ключи, карты монтирования дисков и другую информацию, но вне зависимости от типов записей поиск в кэше выполняется очень похожим образом. В упрощенном виде алгоритм поиска показан на рис. 2.108.

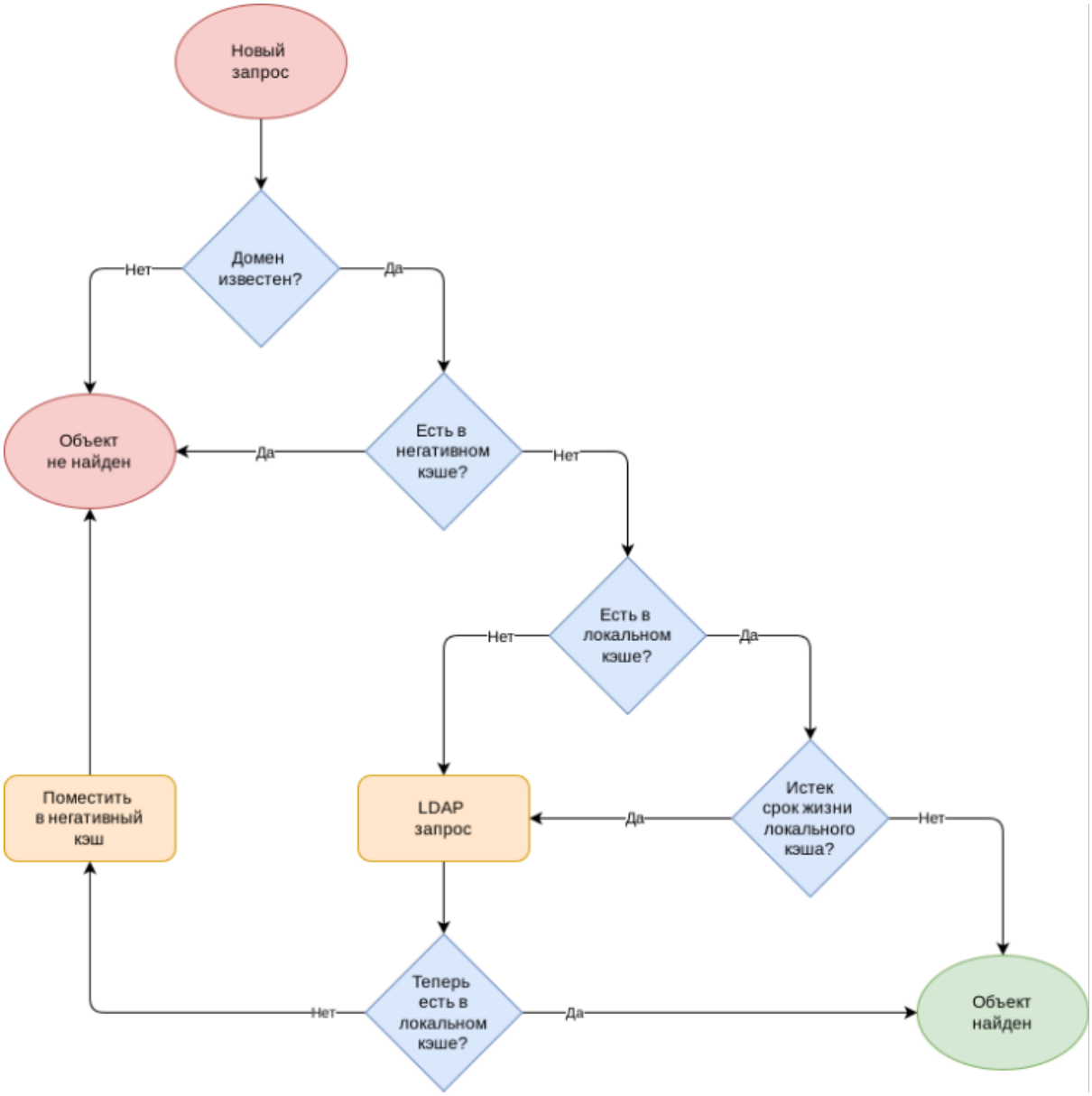


рис. 2.108 – Упрощенный алгоритм поиска в кэше sssd

В действительности алгоритм немного сложнее, и выполняется чуть больше проверок и действий (например, если мы уже знаем, к какому домену относится учетная запись пользователя, то эта информация используется, как подсказка), но приведенной схемы достаточно для получения общих представлений.

На рисунке рис. 2.109 отражен поток данных при поиске информации с использованием кэша. Рассмотрим последовательность взаимодействия компонентов чуть подробнее:

- Клиентское приложение вызывает функцию `getpwnam()` из библиотеки *glibc*, которая в соответствии с настройками из `nsswitch.conf` загружает Клиентскую библиотеку *libnss_sss.so.2* и вызывает ее функции, подробнее см. в разделе *Клиентские библиотеки и приложения*.

- Клиентская библиотека проверяет, есть ли информация о запрашиваемом объекте в быстром кэше (memcache), и только после этого обращается к Ответчику sssd_nss.
- Ответчик проверяет, есть ли информация о запрашиваемом объекте в локальном кэше, и только после этого обращается к Бэкенду sssd_be.
- Бэкенд выполняет запрос к LDAP-каталогу, сохраняет информацию в локальный кэш и сообщает Ответчику о готовности.
- Ответчик повторно обращается к локальному кэшу, извлекает оттуда необходимую информацию, записывает ее в быстрый кэш и передает Клиентской библиотеке в качестве ответа.
- Клиентская библиотека возвращает ответ в функцию библиотеки *glibc*, и на этом поиск завершается.

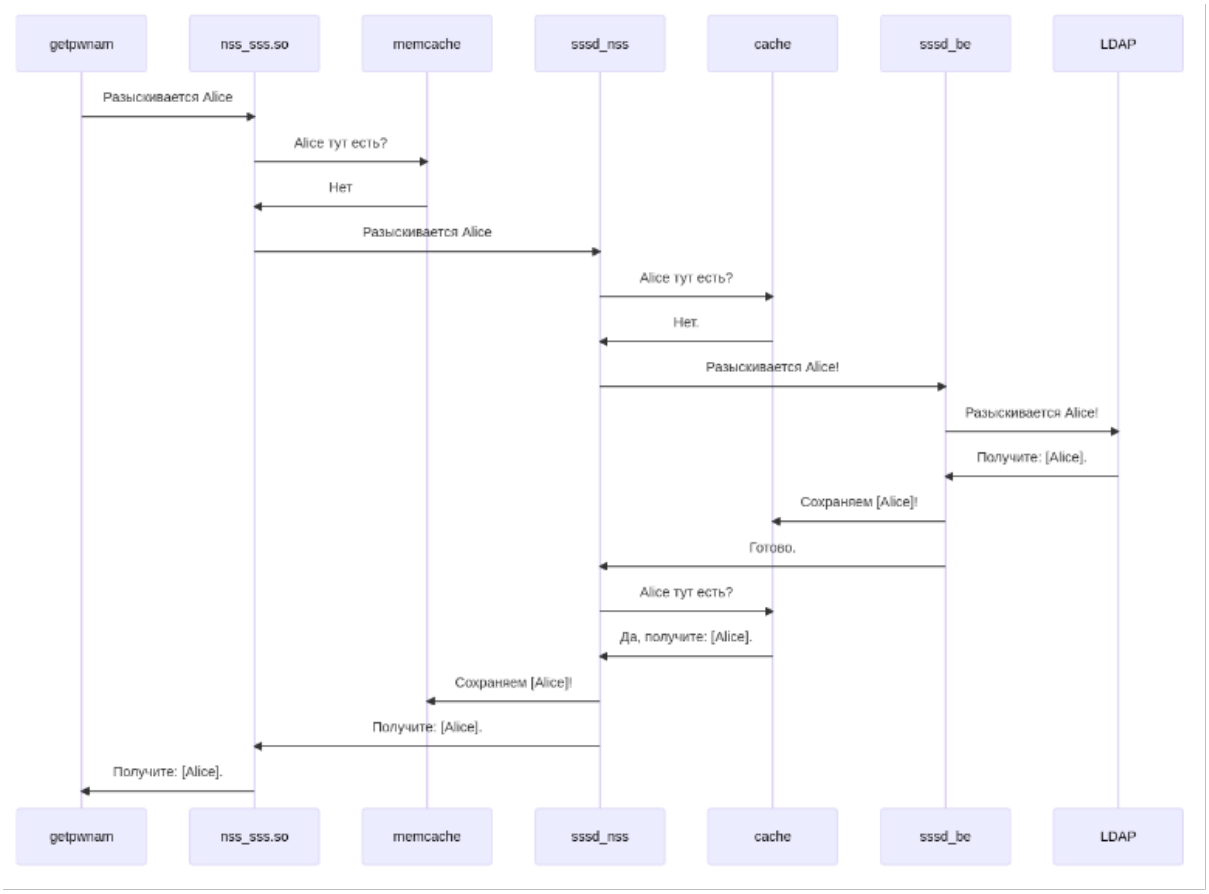


рис. 2.109 – Поток данных при поиске информации о пользователе с использованием информации из кэша

5.3 Способы аутентификации в домене

Служба каталога ALD Pro построена на базе FreeIPA и поддерживает три вида аутентификации: LDAP Bind, Kerberos и NTLM.

5.3.1 Аутентификация по протоколу LDAP

Простая аутентификация или привязка (*англ. Bind*) является самым распространенным, но при этом не самым безопасным способом аутентификации, т.к. пароль на сервер нужно передать открытым текстом. Во избежание разглашения учетных данных требуется обязательно использовать защищенный протокол LDAPS или включать шифрование с помощью StartTLS.

Примечание: Когда вы подключаетесь к LDAP-каталогу, например, с помощью Apache Directory Studio, использовать аутентификацию по паролю вполне допустимо, и вам нужно позаботиться только о том, чтобы канал связи был зашифрован.

Если же у вас серверное приложение для аутентификации доменных пользователей использует пароли, то через такое приложение пароли проходят в открытом виде, а значит его должны администрировать те же люди, которые управляют доменом. Поэтому количество таких приложений в инфраструктуре настоятельно рекомендуется сводить к минимуму.

Чтобы выполнить подключение по LDAPS, этот протокол можно указать явно в способе подключения к серверу:

```
ldapwhoami -H ldaps://dc-1.ald.company.lan \
  -xD 'uid=admin,cn=users,cn=accounts,dc=ald,dc=company,dc=lan' -W -d 1
```

Результат выполнения запроса:

```
admin@dc-1:~$ ldapwhoami -H ldaps://dc-1.ald.company.lan \
>   -xD 'uid=admin,cn=users,cn=accounts,dc=ald,dc=company,dc=lan' -W -d 1
...
ldap_connect_to_host: TCP dc-1.ald.company.lan:636
...
ldap_parse_result
...
dn: uid=admin,cn=users,cn=accounts,dc=ald,dc=company,dc=lan
...
```

Чтобы активировать StartTLS, требуется добавить ключ `-ZZ`:

```
ldapwhoami -H ldap://dc-1.ald.company.lan \
  -xD 'uid=admin,cn=users,cn=accounts,dc=ald,dc=company,dc=lan' -W -d 1 -ZZ
```

Результат выполнения запроса:

```
admin@dc-1:~$ ldapwhoami -H ldap://dc-1.ald.company.lan \
>   -xD 'uid=admin,cn=users,cn=accounts,dc=ald,dc=company,dc=lan' -W -d 1 -ZZ
...
ldap_connect_to_host: TCP dc-1.ald.company.lan:389
...
```

(continues on next page)

(продолжение с предыдущей страницы)

```
res_errno: 0, res_error: <Start TLS request accepted.Server willing to negotiate SSL.>
↪, res_matched: <>
...
ldap_parse_result
...
dn: uid=admin,cn=users,cn=accounts,dc=ald,dc=company,dc=lan
...
```

Напомним назначение ключей, используемых в утилитах из пакета ldap-utils:

- Ключ `-H` — позволяет задать адрес LDAP-сервера и протокол подключения LDAP, LDAPS или LDAPI. Если в качестве протокола указать LDAPS, то будет устанавливаться сразу шифрованное подключение на порт 636/TCP.
- Ключ `-x` — указывает на необходимость выполнить простую аутентификацию по логину/паролю вместо SASL.
- Ключ `-D` — задает отличительное имя DN пользовательской записи, которую нужно использовать для аутентификации.
- Ключ `-W` — указывает, что пароль должен быть предоставлен в интерактивном режиме.
- Ключ `-d` — устанавливает уровень отладки. Значение представляет собой битовую маску от 0 до 65535, в которой каждый бит соответствует определенной категории сообщений отладки. Мы используем значение 1 (LDAP_DEBUG_TRACE), которое соответствует трассировке входов и выходов из процедур.
- Ключ `-ZZ` — позволяет включить шифрование внутри небезопасного протокола LDAP. Устанавливает требование использовать только шифрованное соединение, поэтому если сервер не поддерживает TLS, то соединение будет разорвано.

Опасно: Для лучшего понимания угрозы приведем пример, как делать категорически запрещается:

```
ldapsearch -H ldap://dc-1.ald.company.lan \
  -xD 'uid=admin,cn=users,cn=accounts,dc=ald,dc=company,dc=lan' -w 'Pa$$w0rd'
```

С помощью программы Wireshark вы можете увидеть, что злоумышленник, у которого будет доступ к просмотру сетевых пакетов, может легко извлечь значение `Pa$$w0rd` из запроса, если канал связи не будет зашифрован:

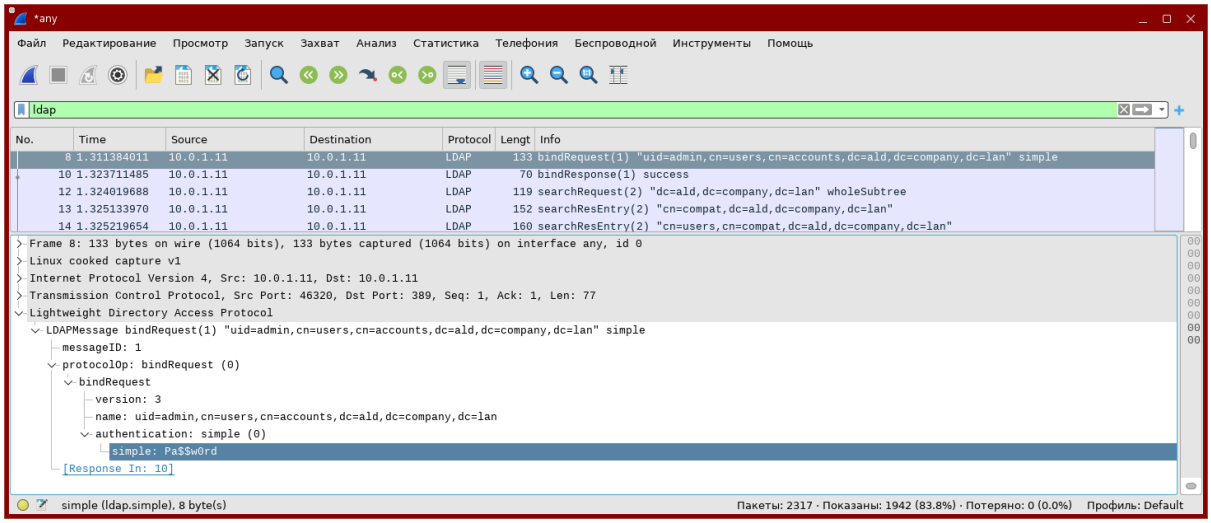


рис. 2.110 – Пароль пользователя открытым текстом в запросе bindRequest

---

Для проведения простой аутентификации LDAP-каталог хранит хеш пароля в атрибуте `userPassword`. Алгоритм проверки аутентичности работает следующим образом:

- Пользователь передает на сервер свое имя и пароль в открытом виде.
- Сервер извлекает из каталога хеш пароля для указанного пользователя.
- Пароль, полученный в запросе на аутентификацию, хешируется тем же алгоритмом и сравнивается со значением, полученным из каталога. Если значения совпали, аутентификация считается успешной.

Хеширование паролей выполняется автоматически в момент изменения значения атрибута `userPassword`. По умолчанию служба каталога используется устойчивый к взлому алгоритм PBKDF2_SHA256, в чем можно убедиться с помощью следующей команды:

```
ldapsearch -H ldaps://dc-1.ald.company.lan \
  -o ldif-wrap=no -D 'cn=Directory Manager' -W \
  -b 'uid=admin,cn=users,cn=accounts,dc=ald,dc=company,dc=lan' \
  | grep 'userPassword::' | cut -d" " -f 2 | base64  -d
```

Результат выполнения:

```
Enter LDAP Password:
{PBKDF2_SHA512}10000$jx2z0Yf0Ak+uJafLT38yMfePgEc$aU6G9cQ/D...
```

Поясним ключи и используемые команды:

- Ключ `-o ldif-wrap=no` — требует выводить текст без переносов в одну строку, если она не

помещается в 80 символов, что полезно в скриптах автоматизации.

- Командой `grep` мы делаем поиск строки, в которой содержится значение «userPassword::» с хешем пароля. Значение атрибута отделено от его названия двумя символами двоеточия, т.к. значение закодировано в base64.
- Командой `cut` мы разрезаем строку, используя символ пробела в качестве разделителя, и с помощью ключа `-f2` извлекаем второй элемент из массива.
- Командой `base64 -d` мы декодируем текст из base64.

Результатом выполнения команды будет строка с паролем администратора, где в фигурных скобках указан способ хеширования, далее до символа $ идет соль пароля. Все остальное – это хеш, полученный с использованием указанной соли и способа хеширования.

```
{PBKDF2_SHA512}10000$jx2z0Yf0Ak+uJafLT38yMfePgEc$aU6G9cQ/D...
```

Если включить режим миграции пользователей, то при создании новых учетных записей в атрибут `userPassword` можно будет записать хеш пароля, при этом поддерживается довольно большой перечень алгоритмов: CRYPT, CRYPT-MD5, CRYPT-SHA256, CRYPT-SHA512, MD5, SHA, SHA256, SHA384, SHA512, SMD5, SSHA, SSHA256, SSHA384, SSHA512.

Включим режим миграции и создадим пользователя `migrated_user` с хешем пароля `{MD5}PMMc0kYUmuxoB5JB5x6Y9g==`, который соответствует значению `Pa$$w0rd`, и проверим возможность LDAP-аутентификации пользователя под этим паролем:

```
ipa config-mod --enable-migration=true

ipa user-add migrated_user \
   --first="migrated" --last="user" \
   --setattr userPassword='{MD5}PMMc0kYUmuxoB5JB5x6Y9g=='

ldapwhoami -H ldaps://dc-1.ald.company.lan \
   -xD 'uid=migrated_user,cn=users,cn=accounts,dc=ald,dc=company,dc=lan' -w 'Pa$$w0rd'
```

Результат выполнения команд:

```
   admin@dc-1:~$ ipa config-mod --enable-migration=true
   Максимальная длина имени пользователя: 32
   Максимальная длина имени хоста: 64
   Основа домашних каталогов: /home
   Оболочка по умолчанию: /bin/bash
   Группа пользователей по умолчанию: ipausers
   Почтовый домен по умолчанию: ald.company.lan
   Ограничение времени поиска: 2
   Ограничение размера поиска: 100
   Поля поиска пользователей: uid,givenname,sn,telephonenumber,ou,title
   Поля поиска групп: cn,description
   Включить режим миграции: TRUE
   Основа субъекта сертификата: O=ALD.COMPANY.LAN
```

(continues on next page)

(продолжение с предыдущей страницы)

```
  Уведомление об окончании действия пароля (в днях): 4
  Возможности подключаемого модуля паролей: AllowNThash, KDC:Disable Last Success
  Порядок применения списка пользователей SELinux: guest_u:s0$xguest_u:s0$user_u:s0
↪$staff_u:s0-s0:c0.c1023$sysadm_u:s0-s0:c0.c1023$unconfined_u:s0-s0:c0.c1023
  Пользователь SELinux по умолчанию: unconfined_u:s0-s0:c0.c1023
  Типы PAC по умолчанию: MS-PAC, nfs:NONE
  Главные IPA-серверы: dc-1.ald.company.lan
  Главный IPA-сервер с поддержкой PKINIT: dc-1.ald.company.lan
  DNS-серверы IPA: dc-1.ald.company.lan

  admin@dc-1:~$ ipa user-add migrated_user \
  >    --first="migrated" --last="user" \
  >    --setattr userPassword='{MD5}PMMc0kYUmuxoB5JB5x6Y9g==='
  ----------------------------------
  Добавлен пользователь "migrated_user"
  ----------------------------------
  Имя учётной записи пользователя: migrated_user
  Имя: migrated
  Фамилия: user
  Полное имя: migrated user
  Отображаемое имя: migrated user
  Инициалы: mu
  Домашний каталог: /home/migrated_user
  GECOS: migrated user
  Оболочка входа: /bin/bash
  Имя учётной записи: migrated_user@ALD.COMPANY.LAN
  Псевдоним учётной записи: migrated_user@ALD.COMPANY.LAN
  Адрес электронной почты: migrated_user@ald.company.lan
  UID: 491200009
  ID группы: 491200009
  Минимальный уровень конфиденциальности: 0
  Максимальный уровень конфиденциальности: 0
  Учётная запись отключена: FALSE
  nsuniqueid: dd687f07-2e5b11f0-81d3c33d-ecd7a769
  proxy addresses: SMTP:MIGRATED_USER@ALD.COMPANY.LAN
  Link to department: ou=ald.company.lan,cn=orgunits,cn=accounts,dc=ald,dc=company,
↪dc=lan
  Link to head department: ald.company.lan
  Пароль: True
  Участник групп: ipausers
  Роли: ALDPRO - Organization units
  Доступные ключи Kerberos: False

  admin@dc-1:~$ ldapwhoami -H ldaps://dc-1.ald.company.lan \
  >    -xD 'uid=migrated_user,cn=users,cn=accounts,dc=ald,dc=company,dc=lan' -w 'Pa$
↪$w0rd'
  dn: uid=migrated_user,cn=users,cn=accounts,dc=ald,dc=company,dc=lan
```

Необходимое значение атрибута `userPassword` можно получить следующими командами на языке Python:

```
admin@dc-1:~$ python3
Python 3.7.3 (default, Apr  8 2024, 13:48:44)
[GCC 8.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
```

(continues on next page)

(продолжение с предыдущей страницы)

```
>>> import hashlib
>>> import binascii
>>> import base64
>>>
>>> password = 'Pa$$w0rd'
>>> md5_hash_string = hashlib.md5(password.encode()).hexdigest()
>>> md5_hash_bytes = binascii.unhexlify(md5_hash_string)
>>> md5_hash_base64_string = base64.b64encode(md5_hash_bytes).decode('utf-8')
>>>
>>> print(md5_hash_base64_string)
PMMc0kYUmuxoB5JB5x6Y9g==
>>> exit()
```

Для завершения миграции пользователю потребуется один раз выполнить простую LDAP-аутентификацию. Для этого на сервере существует даже специальная страница миграции, доступная по адресу https://dc-1.ald.company.lan/ipa/migration/, см. рис. 2.111. В случае успешной аутентификации серверу будет доступен пароль пользователя в открытом виде и он сможет сгенерировать недостающие ключи.

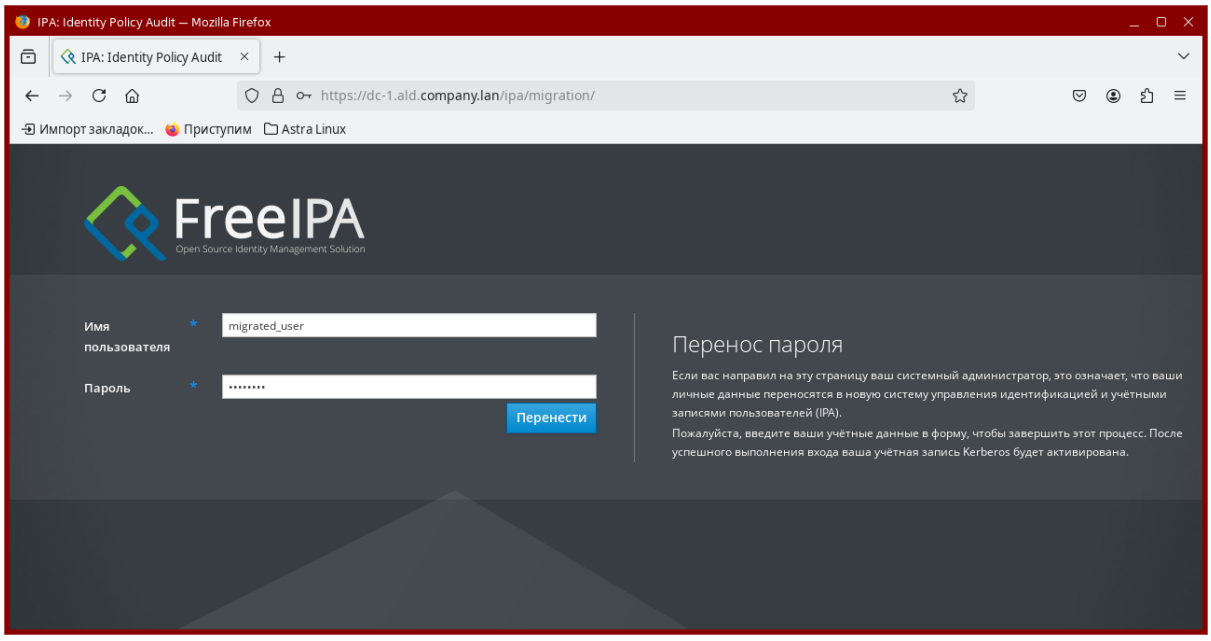


рис. 2.111 – Страница для завершения миграции пользователей

5.3.2 Аутентификация по протоколу Kerberos

5.3.2.1 Общая информация

Основным протоколом аутентификации в домене является Kerberos V5, который был назван так по имени треглавой собаки, охраняющей по древнегреческой мифологии выход из царства мертвых. Каждая голова Цербера соответствует одному из трех участников процедуры аутентификации:

- **Клиент (Client)** — субъект, желающий получить доступ к ресурсу.
- **Сервер приложения (Application Server, AP)** — служба, к ресурсу которой клиент хочет получить доступ.
- **Центр распространения ключей (Key Distribution Center, KDC)** — доверенная сторона, отвечающая за аутентификацию пользователей и выдачу билетов для доступа к сетевым службам в домене.

  Мы уже отмечали, что в рамках KDC часто выделяют два отдельных компонента, отвечающих за каждую из операций (сервер аутентификации AS и сервер выдачи билетов TGS), но на практике такие детали можно считать несущественными.

Служба KDC работает по порту 88/TCP и отвечает только за выдачу билетов. Для управления учетными записями Kerberos на сервере устанавливается так же служба kadmin, которая использует порты 464/TCP и 749/TCP.

5.3.2.2 Поток данных при Kerberos-аутентификации

Рассмотрим процесс аутентификации в упрощенном виде, опуская несущественные детали, такие как фаза предварительной аутентификации, использование nonce и др. Для упрощения понимания протокола вам следует учесть, что он был разработан для эксплуатации в незащищенных компьютерных сетях, когда сетевые пакеты могут быть подслушаны и изменены злоумышленником.

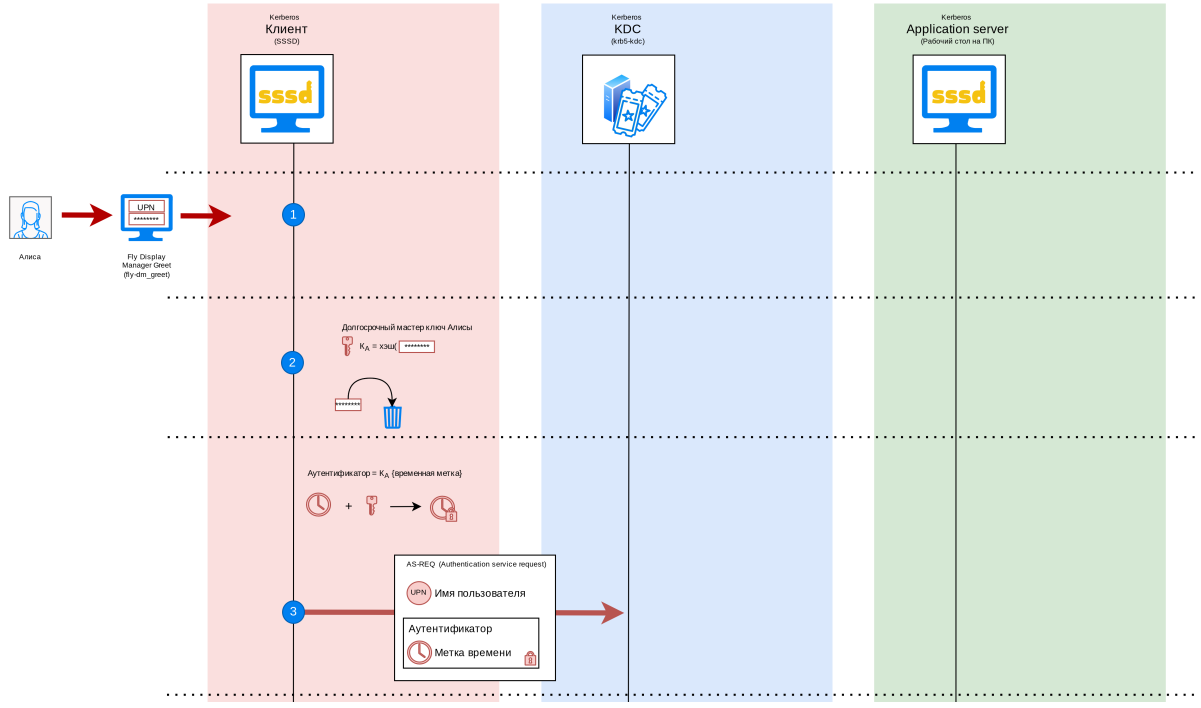


рис. 2.112 – Запрос от клиента на получение TGT-билета

Шаг #1 Пользователь Алиса через приложение графического входа fly-dm_greet (*англ. Fly Display Manager Greet - приложение для входа экранного менеджера Fly*) передает логин и пароль в открытом виде стеку модулей аутентификации (Pluggable Authentication Modules, PAM). За аутентификацию в домене отвечает модуль pam_sss.so, который подключается библиотекой libpam в соответствии с настройками из файла common-auth :

```
admin@pc-1:~$ cat /etc/pam.d/common-auth
...
auth    [success=1 default=ignore]      pam_sss.so use_first_pass
...

admin@pc-1:~$ file /usr/lib/x86_64-linux-gnu/security/pam_sss.so
/usr/lib/x86_64-linux-gnu/security/pam_sss.so: ELF 64-bit LSB shared object, x86-64,␣
↪version 1 (SYSV), dynamically linked,␣
↪BuildID[sha1]=178d1f9af2fafaf8dd6e4d46bd6244804acf31b5, stripped
```

Шаг #2 Функции Клиентской библиотеки pam_sss.so обращаются к Ответчику sssd_pam. Ответчик обращается к Бэкенду, который порождает процесс krb5_child для аутентификации в домене по протоколу Kerberos V5.

```
admin@pc-1:~$ file /usr/lib/x86_64-linux-gnu/sssd/sssd_pam
/usr/lib/x86_64-linux-gnu/sssd/sssd_pam: ELF 64-bit LSB executable, x86-64, version 1␣
↪(SYSV), dynamically linked, interpreter /lib64/ld-linux-x86-64.so.2, for GNU/Linux␣
↪3.2.0, BuildID[sha1]=102efb73ab48e0a7db70e62d76ab3451a2f8c1ba, stripped

admin@pc-1:~$ file /usr/lib/x86_64-linux-gnu/sssd/krb5_child
/usr/lib/x86_64-linux-gnu/sssd/krb5_child: setuid ELF 64-bit LSB executable, x86-64,␣
↪version 1 (SYSV), dynamically linked, interpreter /lib64/ld-linux-x86-64.so.2, for␣
↪GNU/Linux 3.2.0, BuildID[sha1]=0e99bfee28c8d22ef2507670610399a795ab7714, stripped
```

Клиент рассчитывает хеш пароля, который называют также долгосрочным ключом (*UPN long-term key*) или просто Kerberos-ключом. После этого Клиент может удалить пароль в открытом виде из памяти компьютера для повышения устойчивости системы к взлому.

Шаг #3 Клиент отправляет запрос к Центру распространения ключей (*Key Distribution Center, KDC*) для получения TGT-билета.

В четвертой версии протокола на этом шаге никакие учетные данные не требовались, т. к. ответ будет зашифрован долгосрочным ключом пользователя, и способность расшифровать ответ является подтверждением того, что пользователь является тем, за кого себя выдает. Но с пятой версии протокола сервер отклонит запрос и потребует предварительную аутентификацию, поэтому в журнале /var/log/auth.log на сервере можно увидеть строки с текстом «Additional pre-authentication required»:

```
Apr 4 07:27:11 dc-1 krb5kdc[1922]: AS_REQ (8 etypes {aes256-cts-hmac-sha1-96(18),␣
↪aes128-cts-hmac-sha1-96(17), aes256-cts-hmac-sha384-192(20), aes128-cts-hmac-sha256-
↪128(19), DEPRECATED:des3-cbc-sha1(16), DEPRECATED:arcfour-hmac(23), camellia128-cts-
↪cmac(25), camellia256-cts-cmac(26)}) 10.0.1.11: NEEDED_PREAUTH: admin@ALD.COMPANY.
↪LAN for krbtgt/ALD.COMPANY.LAN@ALD.COMPANY.LAN, Additional pre-authentication␣
```

(continues on next page)

(продолжение с предыдущей страницы)

```
↪required
```

В качестве аутентификатора предварительной аутентификации выступает метка времени, зашифрованная симметричным алгоритмом с помощью долгосрочного ключа клиента. Указанная проверка существенно затрудняет кибератаки, так как лишает злоумышленника возможности запросить TGT-билет на любого пользователя в системе.

Почему клиент не отправляет запрос сразу с аутентификатором

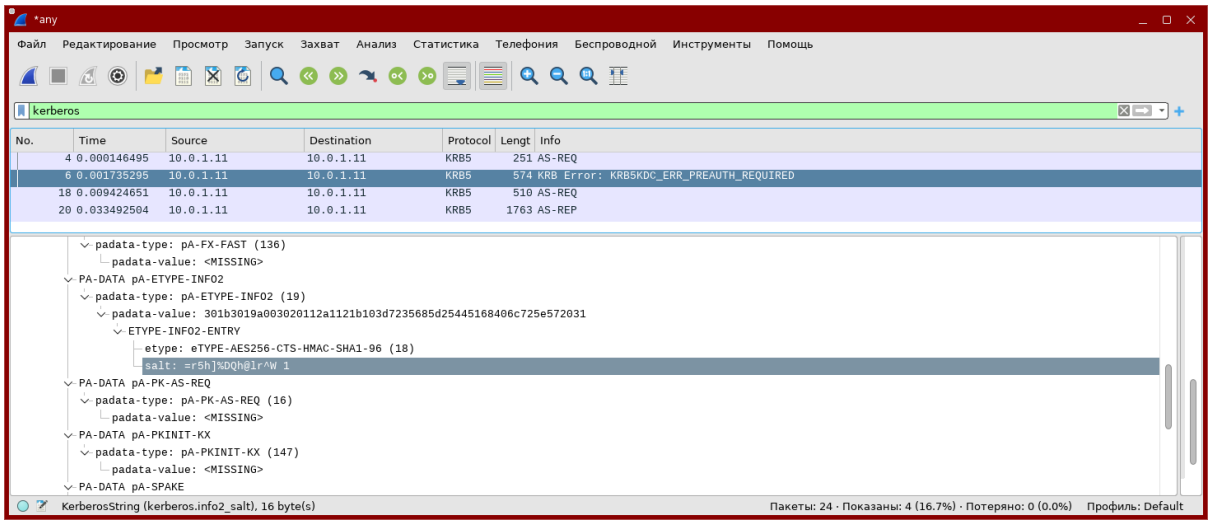
Может показаться, что, отправляя запрос без аутентификатора, клиент совершает лишнее действие, но это не совсем так. Для расчета долгосрочного ключа пользователя нужно знать не только пароль в открытом виде, но и значение соли, с использованием которой этот ключ был сгенерирован.

В самом простом случае соль может быть производным значением от имени принципала, но у пользователя может быть несколько псевдонимов, да и безопаснее, чтобы значение соли было случайным. Поэтому разработчики протокола решили эту задачу крайне интересным способом: в ответе с требованием предварительной аутентификации сервер сам сообщает клиенту, какую соль нужно использовать.

Вы можете увидеть значение соли при выполнении аутентификации с помощью утилиты `kinit`, если включить режим отладки:

```
admin@dc-1:~$ env KRB5_TRACE=/dev/stdout kinit admin <<< 'Pa$$w0rd' | grep salt
[76745] 1746980758.290401: Selected etype info: etype aes256-cts, salt "=r5h]%DQh@lr^
↪W 1", params ""
```

В сетевых пакетах эта информация передается в открытом виде:



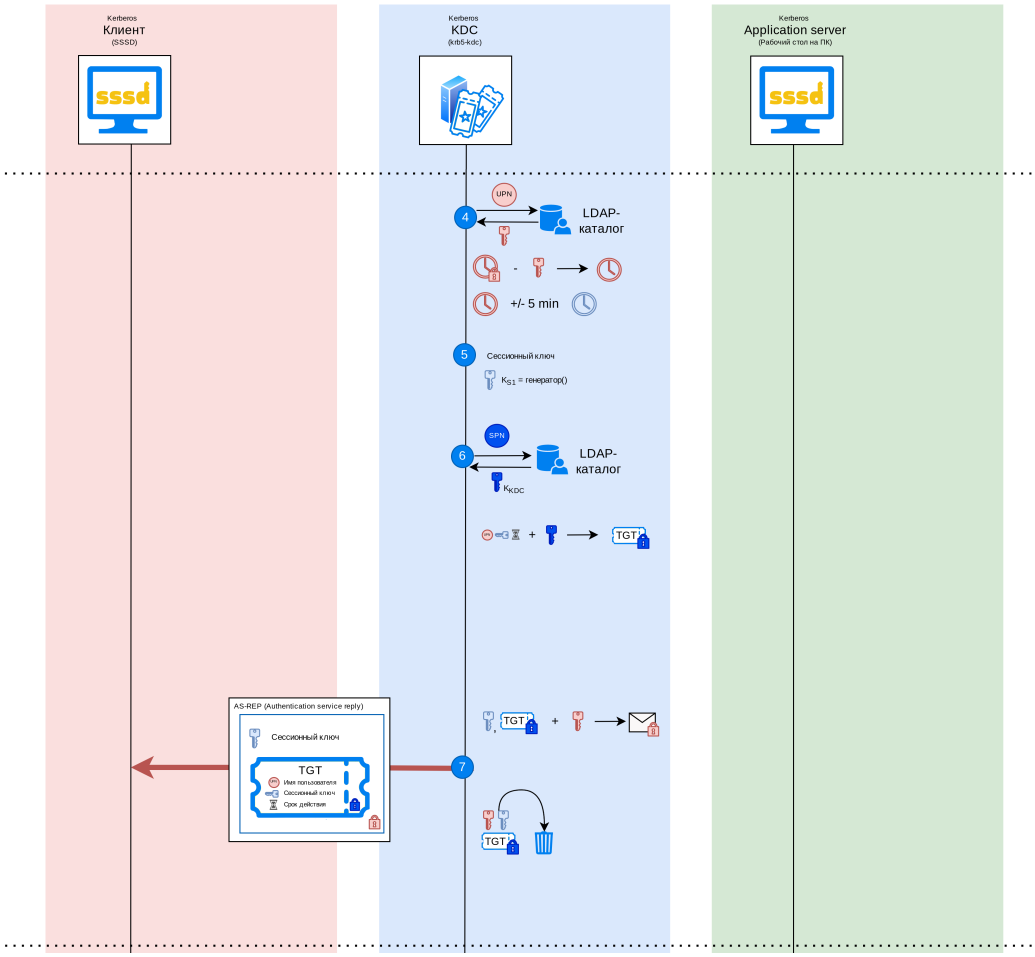


рис. 2.113 – Создание и отправка TGT-билета на сторону клиента

Шаг #4 Служба KDC расшифровывает аутентификатор, используя долгосрочный ключ Алисы из LDAP-каталога. Ключи для аутентификации по протоколу Kerberos хранятся в атрибуте `krbPrincipalKey`, который представляет собой бинарный объект, зашифрованный мастер-ключом KDC (krbMKey). Если процедура расшифровки завершилась успешно и полученная метка времени расходится с временем сервера не более, чем на 5 минут, то считается, что предварительная аутентификация пройдена успешно. По указанной причине для корректной работы протокола Kerberos V5 важна синхронизация времени между всеми участниками домена.

Вы можете уменьшить допустимое отклонение времени, изменив значение параметра `clockskew` в секции `libdefaults` файла `/etc/krb5.conf`, но увеличивать его по соображениям безопасности лучше не стоит. Правильнее будет корректно настроить синхронизацию времени.

Шаг #5 Для повышения безопасности служба KDC генерирует временный сессионный ключ S1, чтобы использовать его в дальнейшем для шифрования сообщений между клиентом и контроллерами домена вместо долгосрочного ключа пользователя.

Обратите внимание, что в названии **Центр распространения ключей** под ключами подразумеваются как раз временные сессионные ключи S1 и S2, которые безопасно передаются между участниками

домена внутри зашифрованных Kerberos-билетов, а не долгосрочные ключи, которые представляют собой хеши паролей.

Шаг #6 Сессионный ключ был сгенерирован конкретным сервером, но в домене может быть несколько контроллеров, и Клиент вправе обратиться с последующим запросом к любому из них. Учитывая случайный характер сессионного ключа, другой контроллер домена не сможет рассчитать его математически, поэтому сессионный ключ следует передать ему в явном виде. По протоколу Kerberos ответственность за передачу сессионного ключа возлагается на клиента, для чего ему выдается зашифрованный билет на выдачу билетов (*Ticket-granting ticket, TGT*), который он должен предъявлять в KDC при последующих обращениях.

Внутри TGT-билета содержится имя пользователя (*User Principal Name, UPN*), сессионный ключ (S1) и информация по сроку действия билета (*Time to Live, TTL*). Билет зашифрован симметричным алгоритмом с помощью долгосрочного ключа KDC (хеш пароля служебной учетной записи KRBTGT, Key Distribution Center Service Account), поэтому расшифровать его можно только на контроллере, и подделать билет на стороне Клиента невозможно.

Шаг #7 Для передачи сессионного ключа и TGT-билета на сторону клиента они шифруются симметричным алгоритмом с помощью долгосрочного ключа клиента, поэтому только клиент сможет расшифровать сообщение, подтверждая этим фактом, что является тем, за кого себя выдает. Данная проверка аутентичности считается основной.

Теперь сервер может удалить долгосрочный ключ клиента, сессионный ключ и билет из памяти за ненадобностью для повышения безопасности системы.

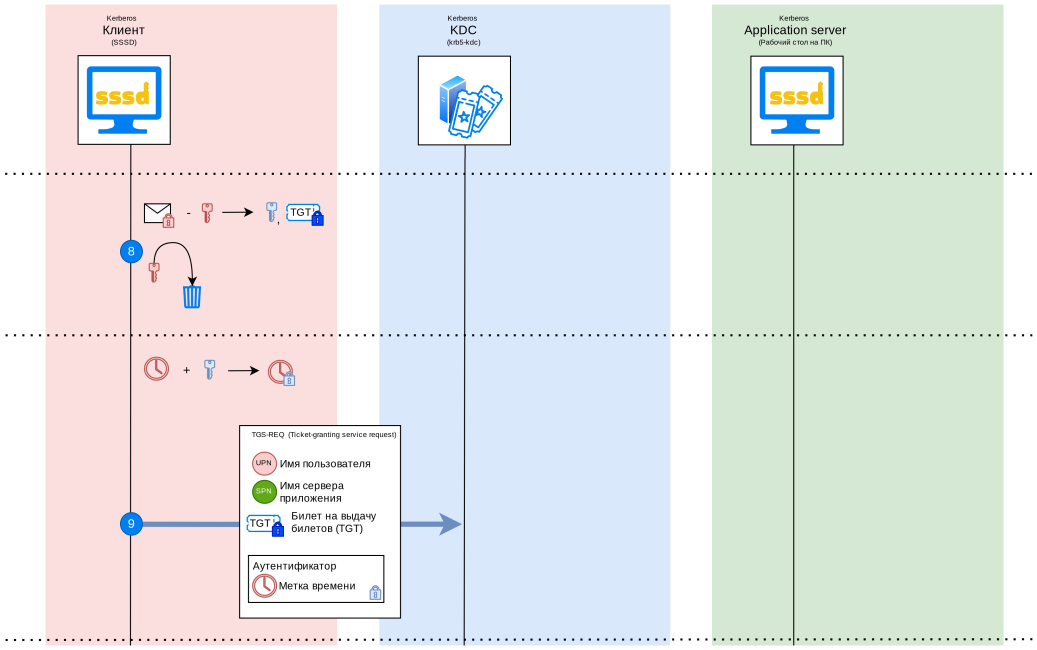


рис. 2.114 – Запрос от клиента на получение сервисного билета с помощью TGT-билета

Шаг #8 Клиент расшифровывает посылку, внутри которой находится сессионный ключ и TGT-билет, своим долгосрочным ключом. Возможность использования этих данных в последующих запросах означает, что Клиент является тем, за кого себя выдает. Если результат расшифровки окажется недействительным, значит, ответ был получен от подставного Центра распределения ключей.

После расшифровки посылки компьютер может удалить долгосрочный ключ пользователя из памяти за ненадобностью для повышения безопасности системы.

Шаг #9 Процесс `krb5_child` сохраняет TGT-билет в связке ключей Linux и передает результаты по цепочке «Бэкенд => Ответчик => Клиентская библиотека».

Получив подтверждение об успешной выдачи TGT-билета, Клиентская библиотека отправляет запрос на получение сервисного билета для доступа к серверу приложения. В запросе содержится имя пользователя (UPN), имя сервера приложения (*Service Principal Name, SPN*), билет на выдачу билетов (TGT) и аутентификатор. В качестве имени приложения выступает имя учетной записи хоста `host/pc-1.ald.company.lan`, аутентификатором является метка времени, зашифрованная симметричным алгоритмом с помощью сессионного ключа S1.

На этом шаге TGT-билет закодирован только долгосрочным ключом KDC, но этого достаточно, чтобы злоумышленник, перехвативший сообщение, не смог подделать запросы, т. к. он не располагает сессионным ключом S1.

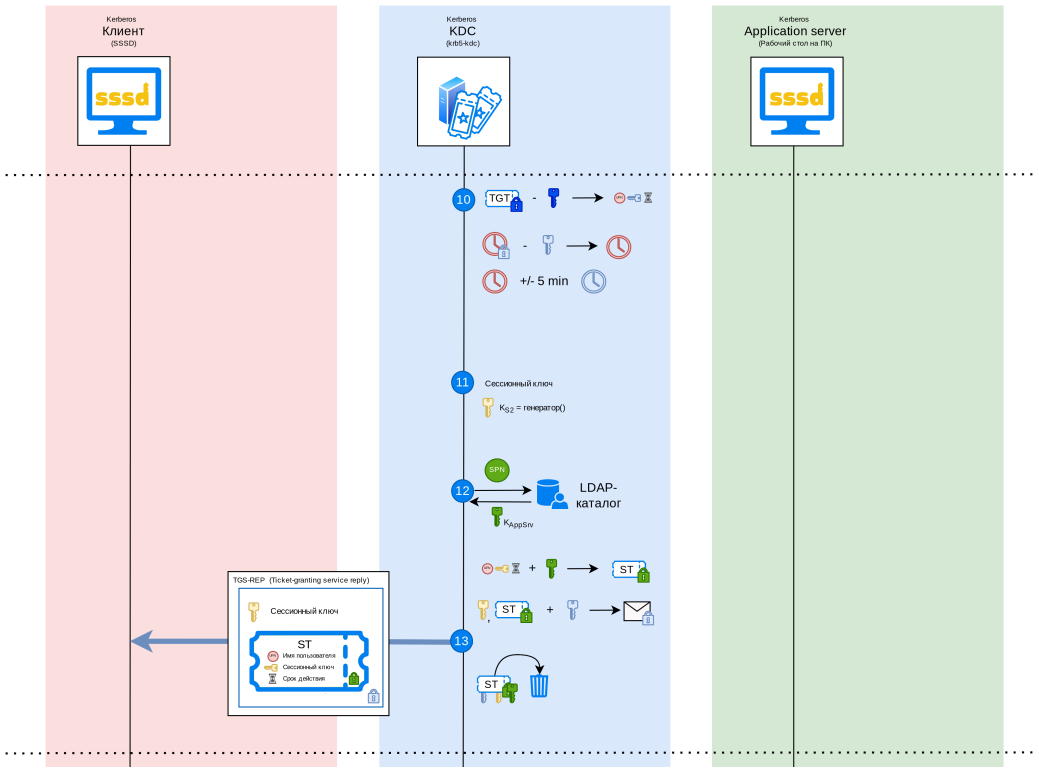


рис. 2.115 – Создание и отправка сервисного TGS-билета на сторону клиента

Шаг #10 Служба KDC расшифровывает информацию из TGT-билета, используя долгосрочный ключ KDC из LDAP-каталога, после чего ей становится доступна следующая информация: имя пользователя, сессионный ключ S1 и срок действия билета.

Далее служба расшифровывает аутентификатор, используя сессионный ключ из TGT-билета, и, если полученное значение расходится с временем сервера не более, чем на 5 минут, то считается, что аутентификация пройдена успешно.

Шаг #11 Для повышения безопасности протокола аутентификации KDC генерирует новый сессионный ключ S2, чтобы его можно было использовать в дальнейшем для шифрования сообщений между клиентом и сервером приложения вместо ключа S1.

Шаг #12 Ключ S2 был сгенерирован сервером KDC, и его следует передать Серверу приложения. По протоколу Kerberos ответственность за передачу ключа возлагается на клиента, для чего ему выдается зашифрованный сервисный билет (*Service ticket, ST*), который он должен предъявлять Серверу приложения. Сервисный билет называют также ST-билетом или TGS-билетом.

Внутри TGS-билета содержится имя пользователя (UPN), сессионный ключ S2 и информация по сроку действия билета (TTL). Билет зашифрован симметричным алгоритмом с помощью долгосрочного ключа Сервера приложения, поэтому расшифровать его сможет только Сервер приложения, и подделать билет на стороне Клиента невозможно.

Сервером приложения в нашем случае является `fly-dm_greet`, поэтому для расшифровки TGS-билета используются ключи от учетной записи компьютера из файла `/etc/krb5.keytab`. Для повышения безопасности каждому керберизированному приложению обычно создается собственная учетная запись в домене.

Шаг #13 После передачи сервисного билета Клиенту информация о ключах на сервере больше не требуется и может быть удалена из памяти для повышения безопасности системы.

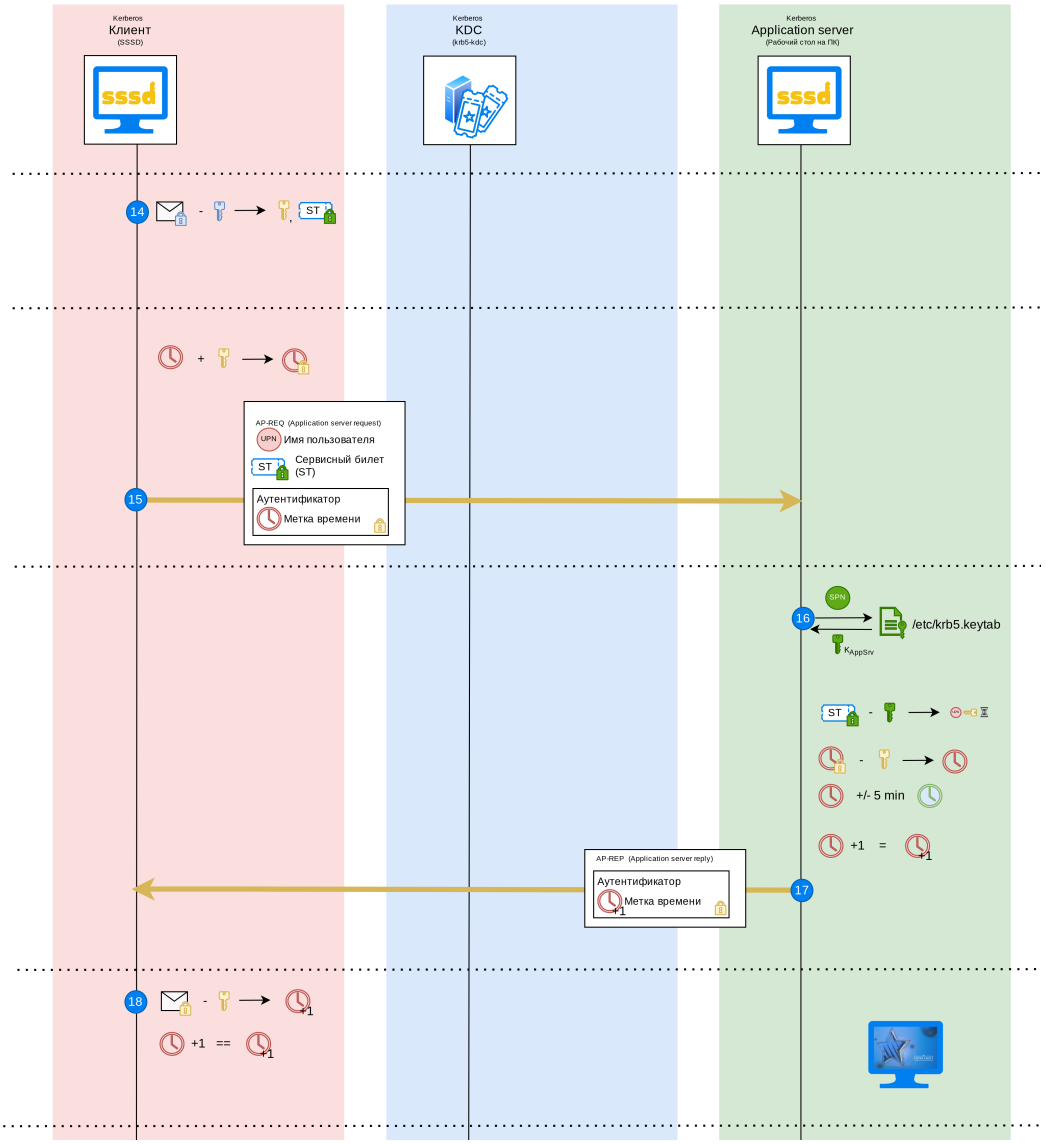


рис. 2.116 – Аутентификация клиента на сервере приложения с помощью TGS-билета

Шаг #14 Клиент расшифровывает сессионный ключ S2 и сервисный билет ST известным ему сессионным ключом S1. Возможность использования этих данных в последующих запросах означает, что Клиент является тем, за кого себя выдает.

Шаг #15 Клиент отправляет Серверу приложения запрос на аутентификацию, в котором содержится имя пользователя (UPN), сервисный билет (ST) и аутентификатор. В качестве аутентификатора выступает метка времени, зашифрованная симметричным алгоритмом с помощью сессионного ключа S2.

На этом шаге ST-билет закодирован только долгосрочным ключом Сервера приложения, но этого достаточно, чтобы злоумышленник, перехвативший сообщение, не смог подделать запросы.

Шаг #16 Сервер приложения, в роли которого выступает служба sssd на пользовательском компьютере, извлекает долгосрочный ключ (хеш пароля) из файла `/etc/krb5.keytab` для расшифровки

запроса. Используя этот ключ, служба sssd может расшифровать информацию из сервисного билета (ST), после чего ей становится доступна следующая информация: имя пользователя (UPN), сессионный ключ S2 и срок действия билета (TTL). Служба sssd расшифровывает аутентификатор, используя сессионный ключ S2 из сервисного билета, и, если полученное значение расходится со временем компьютера не более, чем на 5 минут, то считается, что аутентификация пройдена успешно.

Шаг #17-#18 Если бы в качестве керберизированного сервиса выступала не локальная служба sssd, а какое-нибудь приложение, расположенное на другом хосте, то Сервер приложения мог бы взаимно подтвердить Клиенту свою аутентичность, отправив в качестве аутентификатора метку времени, увеличенную на одну секунду и зашифрованную сессионным ключом S2.

В случае локальной службы sssd взаимная аутентификация не требуется, поэтому служба сразу отдает подтверждение, что Алиса успешно прошла аутентификацию, и приложение `fly-dm_greet` запускает рабочий стол `fly-wm` (*Fly Windows Manager*) от имени этого пользователя.

После проверки аутентичности пользователя TGS-билет на доступ к хосту более не требуется, поэтому компьютер не сохранит его в связке ключей и увидеть этот билет командой `klist` не получится. Но на контроллере домена в журнале `/var/log/auth.log` вы сможете найти подтверждение, что такой билет действительно запрашивался и был предоставлен рабочей станции.

5.3.2.3 Управление Kerberos-аутентификацией

Для работы с Kerberos-билетами на компьютере Linux используются утилиты из пакета krb5-user, такие как `kinit`, `klist`, `kdestroy`, `kvno`, `kpasswd` и другие.

Аутентификация

Для выполнения аутентификации по протоколу Kerberos (т.е. для получения TGT-билета) используется утилита `kinit`. Обратите внимание, что при выполнении повторной аутентификации тем же пользователем все ранее выданные билеты удаляются.

```
admin@dc-1:~$ kinit
Password for admin@ALD.COMPANY.LAN:
```

Имя конкретного пользователя можно указать с помощью параметра:

```
admin@dc-1:~$ kinit ivan.kuznetsov@ALD.COMPANY.LAN
Password for ivan.kuznetsov@ALD.COMPANY.LAN:
```

При этом вы можете использовать как полные имена пользователей, так и короткие, поскольку система будет автоматически дополнять короткие имена значением параметра `default_realm` из файла `/etc/krb5.conf`:

```
admin@dc-1:~$ cat /etc/krb5.conf | grep default_realm
 default_realm = ALD.COMPANY.LAN

admin@dc-1:~$ kinit ivan.kuznetsov
Password for ivan.kuznetsov@ALD.COMPANY.LAN:
```

Примечание: Как мы уже обсуждали ранее (см. примечание *Дополнительная информация об алиасе root принципала admin*, *Модуль 2. Развертывание ALD Pro*), если не указать имя пользователя, то утилита `kinit` попытается сначала взять его из связки ключей, а если кэш будет удален командой `kdestroy`, то переключится на имя текущего пользователя Linux из-за чего при выполнении команды `sudo kinit` мы иногда можем выполнить аутентификацию от имени принципала `root@ALD.COMPANY.LAN`, который является алиасом для учетной записи `admin@ALD.COMPANY.LAN`.

По указанной причине обязательно следите за тем, в каком контексте выполняется команда `kinit`, если используете ее без параметров.

С помощью утилиты `kinit` вы можете выполнить аутентификацию не только по паролю, но и по keytab-файлу, в котором хранятся уже готовые ключи. Для этого достаточно задействовать ключи `-kt` и указать полный путь к файлу:

```
sudo kinit -kt /etc/krb5.keytab
```

Если путь к файлу не указывать с помощью ключа `-t`, то по умолчанию будет использован как раз системный файл `/etc/krb5.keytab`.

Просмотр билетов

Для просмотра полученных Kerberos-билетов используется утилита `klist`. При обращении к утилите без параметров она отобразит информацию о билетах из кэша по умолчанию.

```
admin@dc-1:~$ klist
Ticket cache: KEYRING:persistent:491200000:krb_ccache_8KcsvR7
Default principal: admin@ALD.COMPANY.LAN

Valid starting       Expires              Service principal
02.04.2025 06:33:10  03.04.2025 06:33:07  krbtgt/ALD.COMPANY.LAN@ALD.COMPANY.LAN
```

С помощью ключа `-c` можно прочитать билеты из кэша, который хранится в файле:

```
admin@dc-1:~$ sudo klist -c /var/lib/sss/db/ccache_ALD.COMPANY.LAN
Ticket cache: FILE:/var/lib/sss/db/ccache_ALD.COMPANY.LAN
Default principal: host/dc-1.ald.company.lan@ALD.COMPANY.LAN
```

(continues on next page)

(продолжение с предыдущей страницы)

```
Valid starting       Expires              Service principal
02.04.2025 06:21:24  03.04.2025 06:21:24  krbtgt/ALD.COMPANY.LAN@ALD.COMPANY.LAN
02.04.2025 06:21:24  03.04.2025 06:21:24  ldap/dc-1.ald.company.lan@ALD.COMPANY.LAN
```

Утилита klist позволяет читать не только файлы кэшей учетных данных, но и keytab-файлы. Для того чтобы указать имя keytab-файла, нужно воспользоваться ключом `-k` (keytab). Полезно также использовать ключи `-e` (encryption) и `-t` (timestamp), чтобы увидеть дату выдачи и используемые алгоритмы шифрования.

```
admin@dc-1:~$ sudo klist -ket /etc/krb5.keytab
Keytab name: FILE:/etc/krb5.keytab
KVNO Timestamp           Principal
---- ------------------- ------------------------------------------------------
   2 28.03.2025 10:39:47 host/dc-1.ald.company.lan@ALD.COMPANY.LAN (aes256-cts-hmac-
↪sha1-96)
   2 28.03.2025 10:39:47 host/dc-1.ald.company.lan@ALD.COMPANY.LAN (aes128-cts-hmac-
↪sha1-96)
   2 28.03.2025 10:39:47 host/dc-1.ald.company.lan@ALD.COMPANY.LAN (aes128-cts-hmac-
↪sha256-128)
   2 28.03.2025 10:39:47 host/dc-1.ald.company.lan@ALD.COMPANY.LAN (aes256-cts-hmac-
↪sha384-192)
   2 28.03.2025 10:39:47 host/dc-1.ald.company.lan@ALD.COMPANY.LAN (camellia128-cts-
↪cmac)
   2 28.03.2025 10:39:47 host/dc-1.ald.company.lan@ALD.COMPANY.LAN (camellia256-cts-
↪cmac)
```

Удаление билетов

Для того чтобы удалить билеты Kerberos из кэша, воспользуйтесь утилитой `kdestroy` без параметров. Проверим, что билеты есть в кэше:

```
admin@dc-1:~$ klist
Ticket cache: KEYRING:persistent:491200000:krb_ccache_8KcsvR7
Default principal: admin@ALD.COMPANY.LAN

Valid starting       Expires              Service principal
02.04.2025 06:33:10  03.04.2025 06:33:07  krbtgt/ALD.COMPANY.LAN@ALD.COMPANY.LAN
```

Выполним удаление билетов и проверим кэше еще раз:

```
admin@dc-1:~$ kdestroy
admin@dc-1:~$ klist
klist: Credentials cache keyring 'persistent:491200000:krb_ccache_8KcsvR7' not found`
```

Получение сервисных билетов

Для аутентификации в керберизированных сервисах приложения автоматически запрашивают сервисные билеты через GSSAPI. Все проходит без участия пользователя, поэтому такой метод аутен-

тификации называют прозрачным. Например, если вы зайдете на портал управления с аутентификацией по Kerberos, в связке ключей появится билет на доступ к службе HTTP.

Однако во время отладки, особенно при работе с доверительными отношениями, нам иногда требуется получить билет на какой-то конкретный сервис вручную. Сделать это можно, например, с помощью утилиты `kvno` (*англ. key version number — номер версии ключа*).

Основное назначение этой утилиты состоит в том, чтобы получить билет на указанный сервис и показать номер версии пароля, который хранится на сервере KDC. Это позволяет, например, быстро установить, что keytab-файл был взят из старой резервной копии и там хранятся неактуальные ключи. Но каким бы ни было изначальное назначение утилиты, она очень удобна для отладки.

Чтобы получить билет на требуемый сервис, его название нужно передать утилите `kvno` в качестве параметра. После выполнения команды в связке ключей появится соответствующий билет:

```
admin@dc-1:~$ kinit admin
Password for admin@ALD.COMPANY.LAN:

admin@dc-1:~$ kvno HTTP/dc-1.ald.company.lan
HTTP/dc-1.ald.company.lan@ALD.COMPANY.LAN: kvno = 1

admin@dc-1:~$ klist
Ticket cache: KEYRING:persistent:491200000:krb_ccache_xOCrrRF
Default principal: admin@ALD.COMPANY.LAN

Valid starting       Expires              Service principal
02.04.2025 07:33:57  03.04.2025 07:33:52  HTTP/dc-1.ald.company.lan@ALD.COMPANY.LAN
02.04.2025 07:33:55  03.04.2025 07:33:52  krbtgt/ALD.COMPANY.LAN@ALD.COMPANY.LAN
```

Примечание: Посмотреть названия сервисов можно в портале управления на странице *Управление доменом → Службы и параметры Kerberos*.

Работа с коллекцией ключей

Выполняя Kerberos-аутентификацию под новым пользователем, вы не удаляете ранее полученные билеты, а просто переключаетесь на другое хранилище в рамках вашей коллекции ключей. Посмотреть все билеты в вашей коллекции можно командой `klist` с ключом `-A` (*англ. all - все*):

```
admin@dc-1:~$ kinit admin
Password for admin@ALD.COMPANY.LAN:

admin@dc-1:~$ kinit semenov.ii
Password for semenov.ii@ALD.COMPANY.LAN:

admin@dc-1:~$ klist -A
Ticket cache: KEYRING:persistent:491200000:krb_ccache_oR4OXEO
```

(continues on next page)

(продолжение с предыдущей страницы)

```
Default principal: semenov.ii@ALD.COMPANY.LAN

Valid starting       Expires              Service principal
03.04.2025 06:54:06  03.04.2025 06:54:03  krbtgt/ALD.COMPANY.LAN@ALD.COMPANY.LAN

Ticket cache: KEYRING:persistent:491200000:krb_ccache_8KcsvR7
Default principal: admin@ALD.COMPANY.LAN

Valid starting       Expires              Service principal
02.04.2025 06:53:58  03.04.2025 06:53:55  krbtgt/ALD.COMPANY.LAN@ALD.COMPANY.LAN
```

Это позволяет вам быстро вернуться к нужному кэшу ключей без повторного ввода пароля. Для переключения между хранилищами используют команду `kswitch`, которой нужно передать имя пользователя ключом `-p` (principal). Вы можете использовать короткое имя пользователя без указания Kerberos-реалма.

```
admin@dc-1:~$ kswitch -p admin
admin@dc-1:~$ klist
Ticket cache: KEYRING:persistent:491200000:krb_ccache_8KcsvR7
Default principal: admin@ALD.COMPANY.LAN

Valid starting       Expires              Service principal
02.04.2025 06:53:58  03.04.2025 06:53:55  krbtgt/ALD.COMPANY.LAN@ALD.COMPANY.LAN

admin@dc-1:~$ kswitch -p semenov.ii
admin@dc-1:~$ klist
Ticket cache: KEYRING:persistent:491200000:krb_ccache_oR4OXEO
Default principal: semenov.ii@ALD.COMPANY.LAN

Valid starting       Expires              Service principal
02.04.2025 06:54:06  03.04.2025 06:54:03  krbtgt/ALD.COMPANY.LAN@ALD.COMPANY.LAN
```

Если вам потребуется очистить все хранилища коллекции сразу, воспользуйтесь утилитой `kdestroy` с ключом `-A` (all). После этого klist с ключом `-A` не покажет никаких данных.

```
admin@dc-1:~$ kdestroy -A
admin@dc-1:~$ klist -A
admin@dc-1:~$
```

Администрирование учетных записей Kerberos

Сервер Kerberos может не только выдавать билеты, но и предоставляет интерфейс для управления учетными записями. За это отвечает служба kadmind, которая обрабатывает запросы на портах 464/TCP и 749/TCP.

На контроллерах домена FreeIPA служба не имеет собственной базы данных и взаимодействует с общим LDAP-каталогом через специальный плагин, поэтому при выполнении через нее любых административных действий служба каталога выполняет все проверки в соответствии с установ-

ленными инструкциями доступа.

Служба kadmind принимает удаленные запросы от таких программ, как kadmin, kpasswd, fly-passwd и даже Windows-компьютеры в составе домена ALD Pro (FreeIPA) могут с помощью этой службы предоставить доменному пользователю возможность сменить пароль штатными средствами через Ctrl + Alt + Del.

Например, если вам потребуется сменить пароль пользователя из командной строки, вы можете воспользоваться утилитой kpasswd. При вызове утилиты без параметров она использует текущее имя. Чтобы задать имя пользователя явным образом, передайте его в качестве параметра:

```
admin@dc-1:~$ kpasswd semenov.ii
Password for semenov.ii@ALD.COMPANY.LAN:
Enter new password:
Enter it again:
Password changed.
```

Но для смены пароля мы рекомендуем все-таки установить и использовать графическую утилиту fly-passwd, т.к. она позволяет не только изменить пароль в домене, но и обновить кэш службы sssd, что делает ее незаменимым инструментом для удаленных пользователей, которые подключаются к корпоративной сети через VPN.

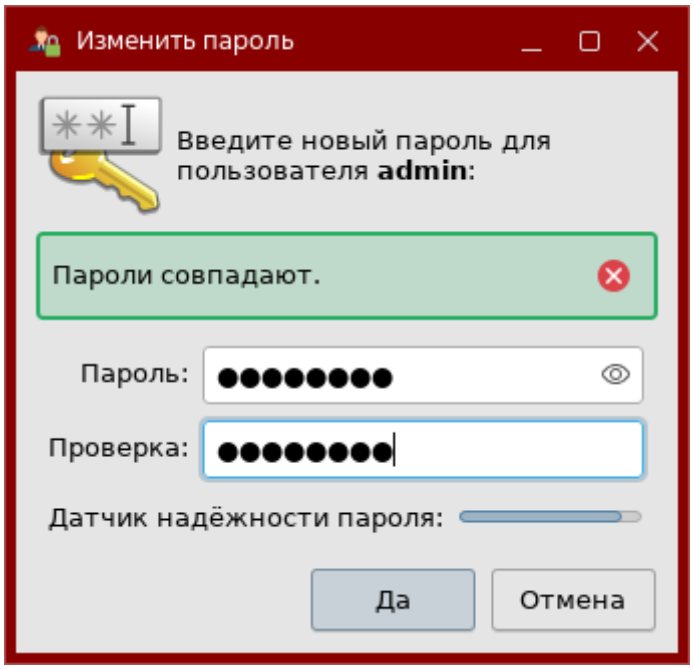


рис. 2.117 – Интерфейс приложения fly-passwd для смены пароля

Для выполнения расширенных задач администрирования вам нужно будет воспользоваться утилитами kadmin или kadmin.local. Но их работу лучше рассматривать в контексте этих задач.

5.3.3 Аутентификация по протоколу NTLM

Для интеграции с Active Directory служба каталога FreeIPA использует компоненты Samba, в которых реализована в том числе NTLM-аутентификация.

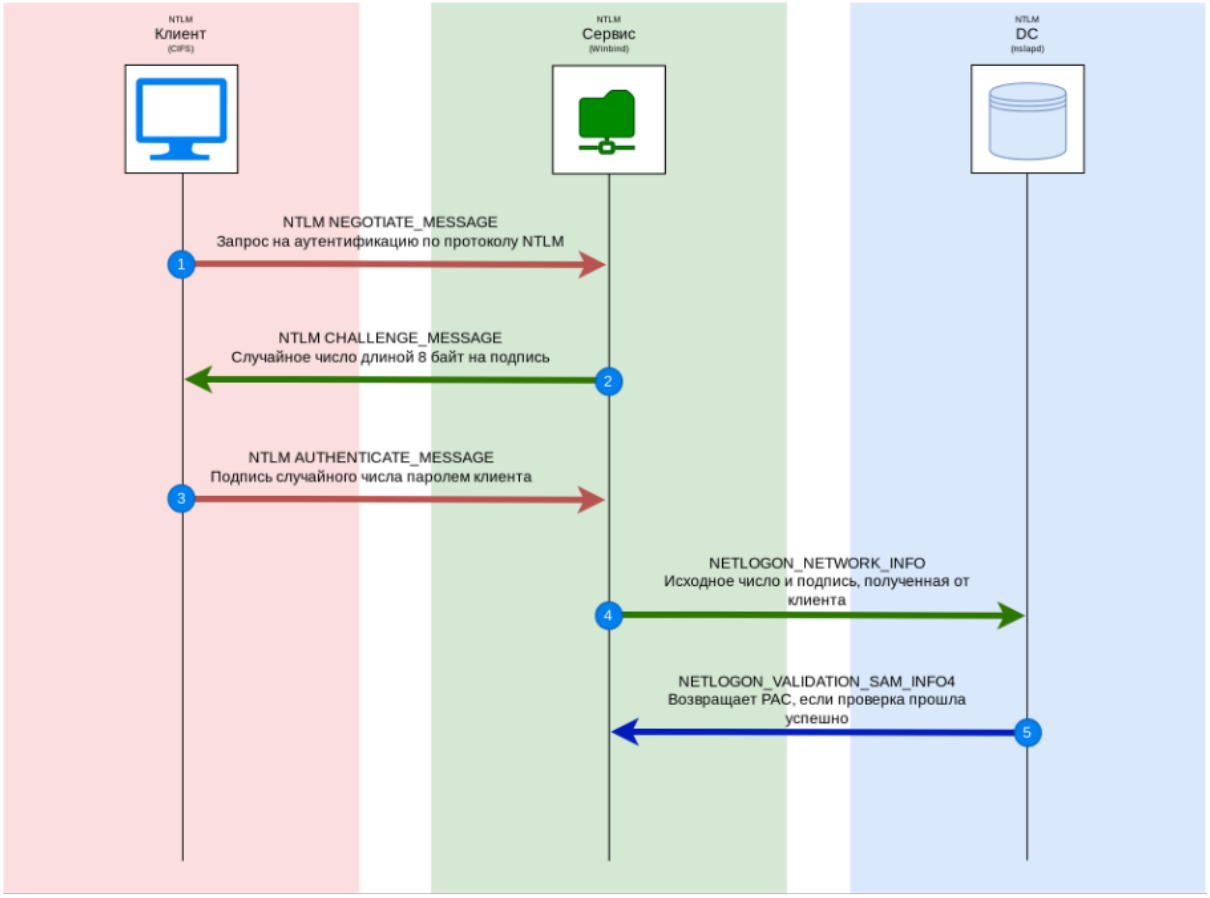


рис. 2.118 – Поток данных при NTLM-аутентификации

На рисунке рис. 2.118. представлена диаграмма аутентификации по протоколу NTLM в упрощенном виде.

Шаг #1 Клиент обращается к Сервису с запросом на аутентификацию.

Шаг #2 Сервер отправляет в ответ случайное число длиной 8 байт (челлендж), которое Клиенту нужно будет подписать паролем, чтобы подтвердить свою аутентичность.

Шаг #3 Клиент подписывает челлендж (шифрует своим паролем) и отправляет Сервису в качестве подтверждения того, что ему известен пароль и он действительно является тем, за кого себя выдает.

Шаг #4 Сервис не может выполнить проверку подписи самостоятельно, т.к. не располагает хешем пользовательского пароля, поэтому пересылает на контроллер домена исходное значение челленджа и полученную подпись.

Шаг #5 Контроллер домена проверяет подпись, используя пароль пользователя из LDAP-каталога, и в случае успеха возвращает PAC-сертификат, в котором содержатся идентификаторы безопасности всех групп, участником которых является текущий пользователь.

Для возможности NTLM-аутентификации у пользователей в каталоге FreeIPA есть атрибут `ipaNTHash`, который обновляется плагином `ipa_pwd_extop` каждый раз при смене пароля. Служба Samba через плагин `ipadb` извлекает значение `ipaNTHash` из LDAP-каталога и выполняет все необходимые проверки аутентичности.

Для проверки работы NTLM-аутентификации на контроллере можно воспользоваться утилитой `ntlm_auth`. За обработку запросов NTLM-аутентификации отвечает **Winbind**.

```
root@dc-1:~# ntlm_auth --debuglevel=10 --domain=ALD --username=admin --password='Pa$
↪$w0rd'
NT_STATUS_OK: The operation completed successfully. (0x0)
```

Текущие настройки службы Samba можно получить командой `testparm`. Роль ROLE_DOMAIN_PDC означает, что параметр `security` равен `USER`, т.е. Samba работает в режиме NT4-контроллера.

```
root@dc-1:~# testparm
Load smb config files from /etc/samba/smb.conf
lpcfg_do_global_parameter: WARNING: The "domain logons" option is deprecated
Loaded services file OK.
Weak crypto is allowed
Server role: ROLE_DOMAIN_PDC

Press enter to see a dump of your service definitions
...
```

Обращаем ваше внимание, что использовать NTLM для интеграции приложений в настоящее время считается даже менее безопасным, чем LDAP-коннекторы, потому что протокол NTLM подвержен MITM-атакам, таким как NTLM Relay, которые позволяют злоумышленникам получить авторизованный доступ к сервису, даже если им неизвестен пароль пользователя. Более того, Relay злоумышленников может направить Клиенту собственное значение челленджа, для которого у них построены радужные таблицы, что существенно упростит взлом пароля методом перебора.

5.4 Безопасный обмен данными с применением SSL/TLS

Доменный компьютер взаимодействует с подсистемами ALD Pro по разным протоколам, некоторые из которых используют шифрование:

- Доступ к каталогу по протоколам LDAP+StartTLS (389/TCP) и LDAPS (636/TCP) со стороны службы sssd и с помощью утилит из пакета ldap-utils.
- Доступ к веб-интерфейсам контроллера домена через браузер и REST API через утилиту ipa

по протоколу HTTPS (443/TCP).

- Доступ к удаленному рабочему столу доменного компьютера с портала управления ALD Pro по протоколу VNC через HTTPS (6080/TCP - 608x/TCP).
- Доступ к принтерам по протоколу IPP (631/TCP), актуально для принтеров, зарегистрированных в подсистеме печати.
- Доступ к PostgreSQL (5432/TCP), актуально для подсистемы репозиториев при репликации базы данных с информацией о загруженных пакетах с главного сервера на подчиненные.

Обращаем внимание, что расширение Kerberos-аутентификации PKINIT использует асимметричное шифрование с применением сертификатов X.509, но шифрование в этом случае используется только для взаимной аутентификации клиента и сервера, а защищенного SSL-соединения между ними не устанавливается.

5.4.1 Механизм защиты данных по протоколу SSL

Протокол SSL (*англ. Secure Sockets Layer, уровень защищенных сокетов*) в его текущей реализации TLS (*англ. Transport Layer Security, безопасность транспортного уровня*) обеспечивает быстрый и безопасный обмен данными между клиентом и сервером за счет сочетания симметричных и асимметричных алгоритмов шифрования: медленные асимметричные алгоритмы используются только на фазе рукопожатия для того, чтобы договориться о параметрах шифрования и безопасно обменяться сессионным ключом, а непосредственно обмен данными происходит уже с применением быстрых симметричных алгоритмов.

Разница между этими алгоритмами заключается в том, что симметричные алгоритмы позволяют расшифровать данные тем же ключом, с помощью которого они были зашифрованы, просто выполняя те же действия в обратном порядке, а в асимметричных алгоритмах используется пара взаимосвязанных ключей, поэтому данные, зашифрованные одним из ключей, могут быть расшифрованы только вторым ключом из той же пары.

Суть работы асимметричных алгоритмов можно продемонстрировать на примере из жизни, см. рис. 2.119. Допустим, что для установления безопасного канала связи нужно обменяться со второй стороной общим кодовым словом. Для этого через службу доставки ей пересылается открытый сейф с самозакрывающимся механизмом, оставляя ключ у первой стороны. Вторая сторона должна будет поместить в этот сейф записку с кодовым словом, защелкнуть замок и переслать сейф обратно. Не взламывая замок, открыть этот сейф можно будет только на первой стороне, т. к. ключ есть только у нее.

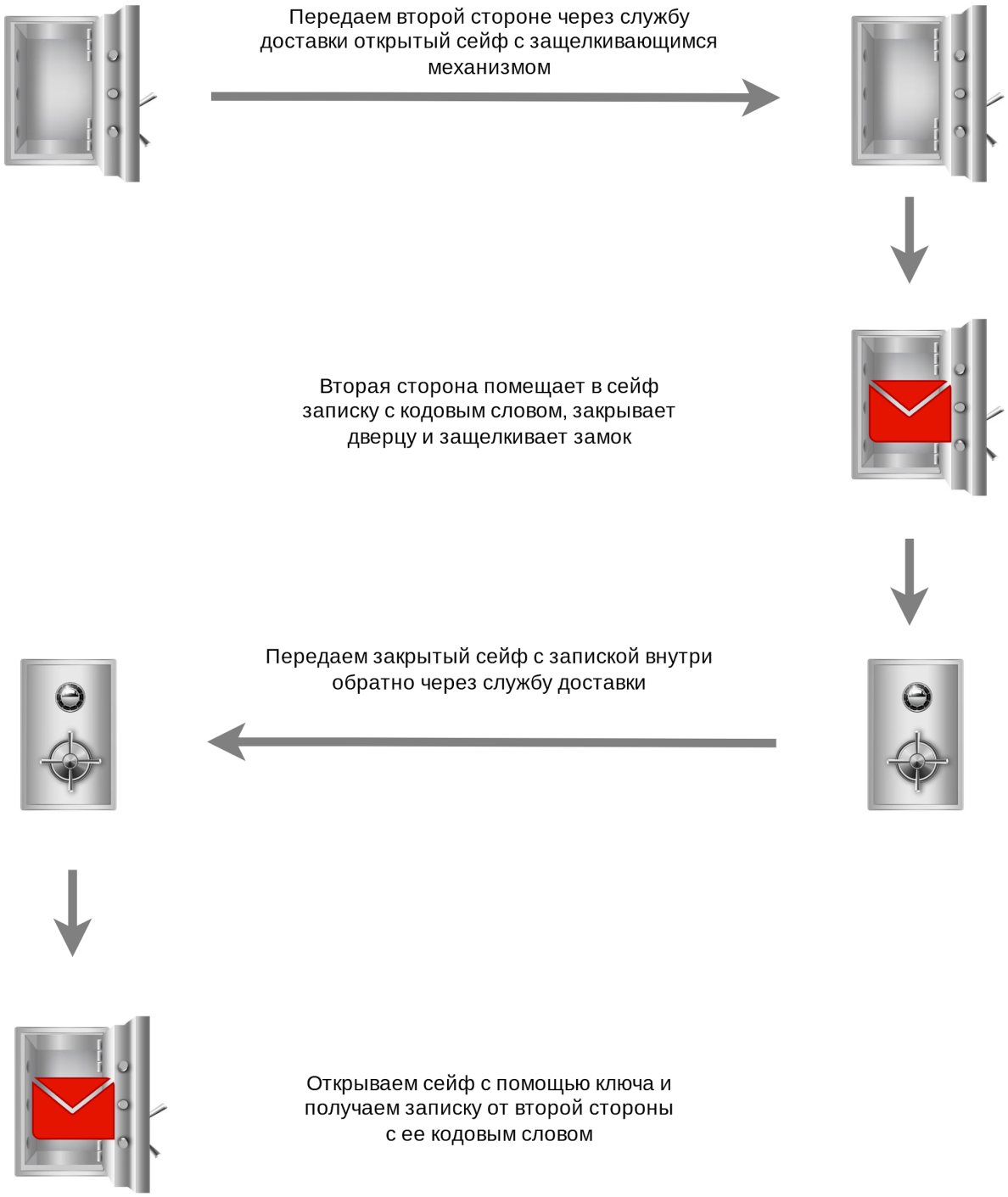


рис. 2.119 – Иллюстрация идеи асимметричного шифрования на примере использования сейфа с самозакрывающимся механизмом

Изначально протокол SSL разрабатывался для HTTP, но его универсальный TCP-подобный интерфейс позволил создать расширения и других протоколов прикладного уровня, например, LDAPS, FTPS, SMTPS, IMAPS и др.

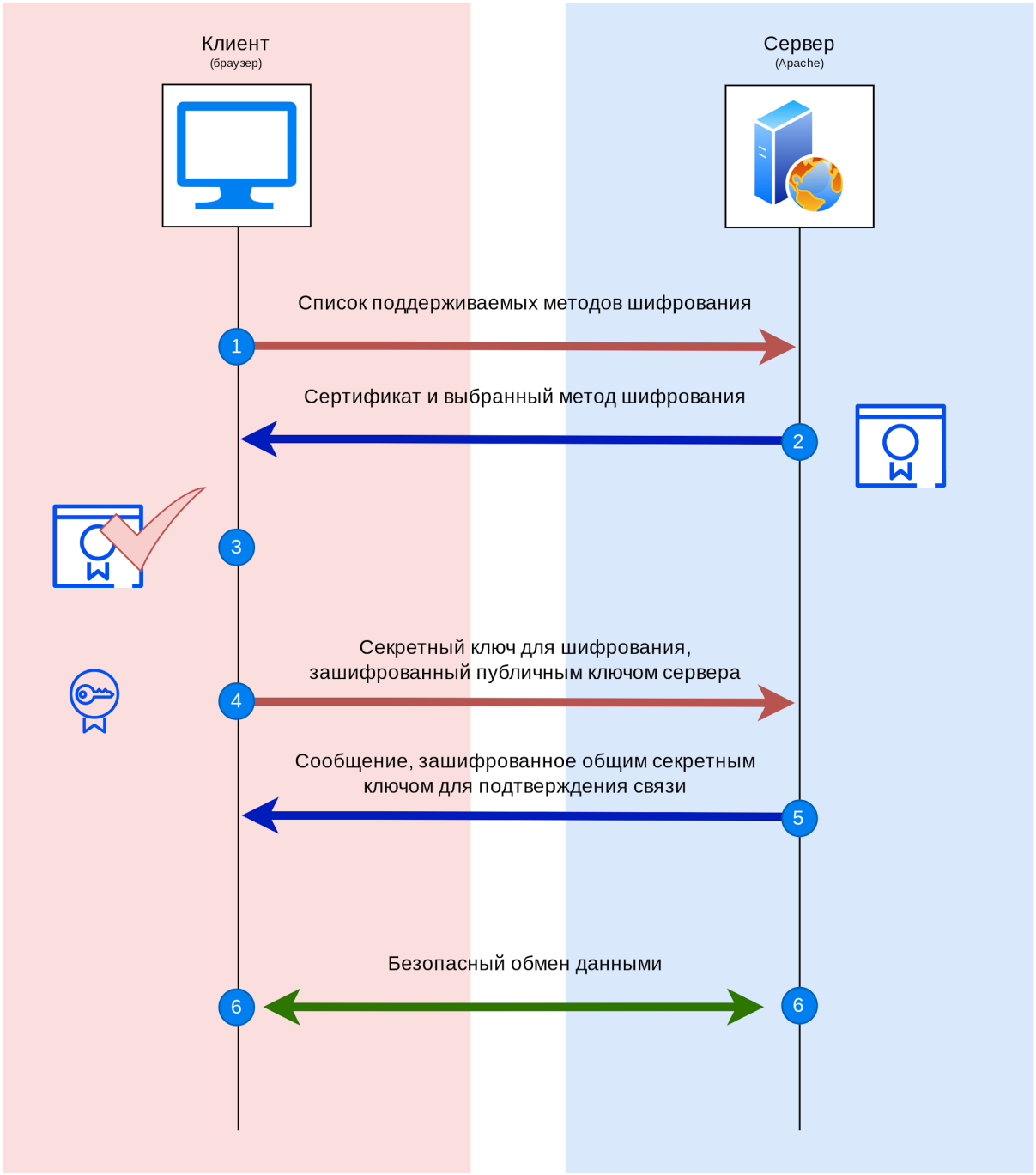


рис. 2.120 – Установление безопасного HTTPS-соединения

В качестве примера рассмотрим поток данных при открытии веб-страницы по протоколу HTTPS, см. рис. 2.120:

Шаг #1 Когда пользователь переходит по https-ссылке, браузер выполняет подключение к веб-серверу на 443/TCP порт и передает список поддерживаемых методов шифрования для установления безопасного соединения.

Шаг #2 Веб-сервер выбирает метод шифрования и передает клиенту свой SSL-сертификат, который

содержит публичный ключ и цифровую подпись для этого ключа. Вернее, он передает не один сертификат, а всю цепочку до корневого сертификата включительно.

Шаг #3 Клиент проверяет валидность цифровой подписи сертификата по своей базе доверенных корневых сертификатов, используя алгоритмы асимметричного шифрования. Клиент также проверяет подлинность сервера, сопоставляя имя, по которому обращается к нему, с тем именем, которое указано внутри сертификата. Если сертификат сайта не пройдет проверку, то пользователь получит сообщение о небезопасности соединения ввиду возможности атак посредника (*Man in the middle, MITM*).

Шаг #4 Клиент генерирует случайный ключ и шифрует его асимметричным алгоритмом с помощью публичного ключа сервера для безопасной передачи данных по сети.

Шаг #5 Сервер извлекает общий ключ из сообщения, расшифровывая его своим закрытым ключом, и передает Клиенту в ответ сообщение, зашифрованное теперь уже общим ключом, подтверждая возможность начала безопасного обмена данными.

Шаг #6 Клиент и сервер начинают обмен данными по защищенному каналу связи.

5.4.2 Доступ к каталогу по протоколам LDAP+StartTLS и LDAPS

Служба sssd получает данные из каталога по протоколу LDAP (389/TCP) с использованием расширения StartTLS, которое позволяет инициировать шифрованный обмен данными внутри уже установленного TCP-соединения вместо открытия нового соединения на отдельном порту.

Администратор может проверить, что порт 389/TCP действительно защищен шифрованием, и получить цепочку сертификатов с помощью утилиты openssl. Первый сертификат выписан на конкретное полное имя FQDN контроллера домена, а второй соответствует корневому сертификату домена.

```
$ echo Q | openssl s_client -starttls ldap -showcerts dc-1.ald.company.lan:389
CONNECTED(00000003)
depth=1 CN = CA Signing Certificate
verify return:1
depth=0 CN = dc-1.ald.company.lan
verify return:1
---
Certificate chain
 0 s:CN = dc-1.ald.company.lan
   i:CN = CA Signing Certificate
 1 s:CN = CA Signing Certificate
   i:CN = CA Signing Certificate
---
Server certificate
-----BEGIN CERTIFICATE-----
aMIIDxjCCAq6gAwIBAgIUZ4JathCvRrV6v+Aoi4kUNScViBswDQYJKoZIhvcNAQEL
```

(continues on next page)

(продолжение с предыдущей страницы)

```
aBQAwITEfMB0GA1UEAwwWQ0EgU2lnbmluZyBDZXJ0aWZpY2F0ZTAeFw0yMjExMTYx
NzM4MjVaFw0zMjExMTMxNzM4MjVaMCExHzAdBgNVBAMMFmRjLTEuYWxkLmNvbXBh
...
-----BEGIN CERTIFICATE-----
MIIDIzCCAgugAwIBAgIUULQd6OzgJ/ik7wJvV0gd6kUIFKAwDQYJKoZIhvcNAQEL
BQAwITEfMB0GA1UEAwwWQ0EgU2lnbmluZyBDZXJ0aWZpY2F0ZTAeFw0yMjExMTYx
NzM4MjVaFw00MjExMTExNzM4MjVaMCExHzAdBgNVBAMMFkNBIFNpZ25pbmcgQ2Vy
...
```

Доверие к этой цепочке проверяется с помощью корневого сертификата из файла `/etc/ipa/ca.crt`, на который указывает параметр `ldap_tls_cacert` из секции `domain` конфигурационного файла `/etc/sssd/sssd.conf`. Данный сертификат загружается на компьютер автоматически при вводе компьютера в домен. Как можно заметить по первым трём строкам, содержимое этого файла соответствует последнему сертификату из цепочки, полученной от сервера.

```
admin@dc-1:~$ cat /etc/sssd/sssd.conf
[domain/ald.company.lan]
...
ldap_tls_cacert = /etc/ipa/ca.crt
...

admin@dc-1:~$ cat /etc/ipa/ca.crt
-----BEGIN CERTIFICATE-----
MIIDIzCCAgugAwIBAgIUULQd6OzgJ/ik7wJvV0gd6kUIFKAwDQYJKoZIhvcNAQEL
BQAwITEfMB0GA1UEAwwWQ0EgU2lnbmluZyBDZXJ0aWZpY2F0ZTAeFw0yMjExMTYx
NzM4MjVaFw00MjExMTExNzM4MjVaMCExHzAdBgNVBAMMFkNBIFNpZ25pbmcgQ2Vy
...
```

Проверить содержимое сертификата можно с помощью все той же утилиты `openssl`. Как можно заметить, по умолчанию корневой сертификат выписан на 10 лет, после чего его нужно будет продлевать.

```
admin@dc-1:~$ openssl x509 -in /etc/ipa/ca.crt -text -noout
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            50:b4:1d:e8:ec:e0:27:f8:a4:ef:02:6f:57:48:1d:ea:45:08:14:a0
        Signature Algorithm: sha256WithRSAEncryption
        Issuer: CN = CA Signing Certificate
        Validity
            Not Before: Nov 16 17:38:25 2022 GMT
            Not After : Nov 11 17:38:25 2042 GMT
...
```

Для упрощения отладки работы компьютеров в домене на них автоматически устанавливается и настраивается пакет ldap-utils, в состав которого входят такие утилиты, как `ldapsearch`, `ldapadd` и др. Безопасность соединения проверяется с использованием корневого сертификата из файла `/etc/ssl/certs/ca-certificates.crt`, на который указывает параметр `TLS_CACERT` из конфигурационного файла `/etc/ldap/ldap.conf`.

```
admin@dc-1:~$ cat /etc/ldap/ldap.conf
...
TLS_CACERT /etc/ssl/certs/ca-certificates.crt
URI ldaps://dc-1.ald.company.lan
BASE dc=ald,dc=company,dc=lan
SASL_MECH GSSAPI
...
```

Файл ca-certificates.crt является связкой сертификатов, в которой содержится более 100 корневых сертификатов удостоверяющих центров, которым доверяет операционная система Astra Linux. Корневой сертификат домена, который отображается в файле /etc/ipa/ca.crt, обычно находится в самом конце этого файла. Чтобы просмотреть информацию по всем сертификатам из файла, можно использовать следующую команду:

```
openssl crl2pkcs7 -nocrl -certfile /etc/ssl/certs/ca-certificates.crt \
  | openssl pkcs7 -print_certs -text -noout | less
```

5.5 Учетная запись компьютера в домене

Для того чтобы служба sssd могла выполнять авторизованные LDAP-запросы к каталогу и проверять аутентичность пользователей по протоколу Kerberos V5, у каждого компьютера в домене есть своя собственная учетная запись.

Учетные записи компьютеров хранятся в cn=computers,cn=accounts,dc=ald,dc=company,dc=lan, имя LDAP-записи включает значение атрибута fqdn, например, fqdn=pc-1.ald.company.lan, а для аутентификации по протоколу Kerberos используется имя принципала в формате host/pc-1.ald.company.lan@ALD.COMPANY.LAN.

Билеты службы sssd кэшируются в файле /var/lib/sss/db/ccache_ALD.COMPANY.LAN, прочитать который можно с помощью утилиты klist, используя параметр -c.

```
root@pc-1:~# klist -c /var/lib/sss/db/ccache_ALD.COMPANY.LAN
Ticket cache: FILE:/var/lib/sss/db/ccache_ALD.COMPANY.LAN
Default principal: host/dc-1.ald.company.lan@ALD.COMPANY.LAN

Valid starting      Expires             Service principal
02.04.2025 16:32:19 03.04.2025 16:32:19  krbtgt/ALD.COMPANY.LAN@ALD.COMPANY.LAN
02.04.2025 16:32:19 03.04.2025 16:32:19  ldap/dc-1.ald.company.lan@ALD.COMPANY.LAN
```

Пароль учетной записи хоста хранится в файле /etc/krb5.keytab, поэтому пользователь может легко проходить аутентификацию и выполнять запросы от имени хоста. Проверка валидности keytab-файла является одной из рекомендаций по устранению неисправностей, если в журналах sssd появится информация о том, что служба не смогла пройти аутентификацию в домене. Такие ситуации могут возникать, например, если компьютер был удален из домена.

Содержимое keytab-файла можно посмотреть с помощью утилиты `klist`, используя параметр `-k`. Если добавить параметр `-e`, то можно будет увидеть, что для одной версии пароля в файле содержится сразу два ключа, полученные с помощью разных алгоритмов хеширования:

```
root@pc-1:~# klist -ke /etc/krb5.keytab
Keytab name: FILE:/etc/krb5.keytab
KVNO Principal
---- --------------------------------------------------------------------------
   1 host/pc-1.ald.company.lan@ALD.COMPANY.LAN (aes256-cts-hmac-sha1-96)
   1 host/pc-1.ald.company.lan@ALD.COMPANY.LAN (aes128-cts-hmac-sha1-96)
```

Как уже было сказано ранее, для того чтобы пройти аутентификацию от имени хоста, достаточно использовать команду `sudo kinit -k`. После выполнения команды `ldapsearch` можно увидеть, что в связке ключей появился билет на доступ к службе LDAP для учетной записи хоста:

```
admin@pc-1:~$ sudo kinit -k
admin@pc-1:~$ ldapsearch
...
admin@pc-1:~$ klist
Ticket cache: KEYRING:persistent:491200000:krb_ccache_HXKunix
Default principal: host/pc-1.ald.company.lan@ALD.COMPANY.LAN

Valid starting       Expires              Service principal
02.04.2025 16:03:39  03.04.2025 16:03:32  ldap/dc-1.ald.company.lan@ALD.COMPANY.LAN
02.04.2025 16:03:32  03.04.2025 16:03:32  krbtgt/ALD.COMPANY.lan@ALD.COMPANY.LAN
```

Для обеспечения безопасности пароли хешируются с использованием устойчивых к взлому алгоритмов, а перед хешированием к тексту пароля добавляется еще и значение соли, что полностью исключает возможность перебора хешей по радужным таблицам. Но даже такие меры безопасности не защищают от всех видов атак, поэтому пароли хостов рекомендуется время от времени менять, например, компания Microsoft предлагает делать это не реже одного раза в 30 дней.

Чтобы установить хосту новый пароль и выгрузить его в keytab-файл, можно воспользоваться утилитой `ipa-getkeytab`:

```
sudo ipa-getkeytab -p host/pc-1.ald.company.lan \
  -k /etc/krb5.keytab -s dc-1.ald.company.lan \
  -D uid=admin,cn=users,cn=accounts,dc=ald,dc=company,dc=lan -W -P
```

Рассмотрим пример использования команды. Обратите внимание, что после смены пароля команда `klist` в колонке KVNO показывает новую (вторую) версию ключей.

```
admin@pc-1:~$ sudo rm -rf /etc/krb5.keytab

admin@pc-1:~$ sudo ipa-getkeytab -p host/pc-1.ald.company.lan \
> -k /etc/krb5.keytab -s dc-1.ald.company.lan \
> -D uid=admin,cn=users,cn=accounts,dc=ald,dc=company,dc=lan -W -P
Введите пароль LDAP: ********
Новый пароль учётной записи: ********
Проверка пароля учётной записи: ********
```

(continues on next page)

(продолжение с предыдущей страницы)

```
Таблица ключей успешно получена и сохранена в: /etc/krb5.keytab

admin@pc-1:~$ sudo klist -ke /etc/krb5.keytab
Keytab name: FILE:/etc/krb5.keytab
KVNO Principal
---- --------------------------------------------------------------------------
   2 host/pc-1.ald.company.lan@ALD.COMPANY.LAN (aes256-cts-hmac-sha1-96)
   2 host/pc-1.ald.company.lan@ALD.COMPANY.LAN (aes128-cts-hmac-sha1-96)
```

Параметры вызова утилиты `ipa-getkeytab`:

- Ключ `-p` — задает имя учетной записи, для которой нужно сменить пароль.
- Ключ `-k` — указывает имя keytab-файла, в который требуется добавить ключи.
- Ключ `-s` — определяет адрес сервера (контроллера домена), через который должен быть выполнен запрос на изменение пароля.
- Ключ `-D` — задает отличительное имя (DN) привилегированной учетной записи, от которой можно выполнить изменения в каталоге. Если ключ `-D` не будет указан, то утилита `ipa-getkeytab` воспользуется учетными данными текущего Kerberos-пользователя.
- Ключ `-W` — указывает, что пароль привилегированной учетной записи будет предоставлен в интерактивном режиме.

  Для того, чтобы задать пароль прямо в командной строке, можно использовать ключ `-w`, но делать так крайне не рекомендуется, поскольку значение этого параметра будет видно на экране, в выводе команды ps и попадет в историю команд.
- Ключ `-P` — определяет, что пароль хоста будет введен вручную. Если этот параметр не указывать, пароль будет сгенерирован автоматически.

Учитывая, что без указания привилегированного пользователя в явном виде утилита `ipa-getkeytab` использует учетные данные текущего Kerberos-пользователя, мы можем сменить пароль компьютера, используя учетные данные самого компьютера из его текущего keytab-файла:

```
admin@pc-1:~$ sudo kinit -kt /etc/krb5.keytab
admin@pc-1:~$ sudo ipa-getkeytab -p host/pc-1.ald.company.lan -k /etc/krb5.keytab
admin@pc-1:~$ sudo klist -ke /etc/krb5.keytab
Keytab name: FILE:/etc/krb5.keytab
KVNO Principal

---- --------------------------------------------------------------------------
   2 host/pc-1.ald.company.lan@ALD.COMPANY.LAN (aes256-cts-hmac-sha1-96)
   2 host/pc-1.ald.company.lan@ALD.COMPANY.LAN (aes128-cts-hmac-sha1-96)
   3 host/pc-1.ald.company.lan@ALD.COMPANY.LAN (aes256-cts-hmac-sha1-96)
   3 host/pc-1.ald.company.lan@ALD.COMPANY.LAN (aes128-cts-hmac-sha1-96)
```

Регулярную смену пароля для учетной записи компьютера можно организовать с помощью дополнительного параметра групповой политики `maximum_password_age`.

5.6 Работа компьютера в автономном режиме

Для обеспечения комфортной работы пользователей в условиях нестабильного подключения к локальной сети по Wi-Fi или через VPN служба sssd может предоставлять часть сервисов автономно без подключения к серверу. Переход в автономный режим происходит автоматически в следующих случаях:

- пропало подключение к сети;
- нет возможности преобразовать полное имя сервера FQDN в IP-адрес;
- не удается подключиться к серверу.

Работая автономно, Монитор службы sssd предпринимает попытки возврата в режим онлайн по следующим триггерам:

- получено уведомление об изменении настроек сети, например, о подключении кабеля или изменении таблицы маршрутизации (см. информацию по библиотеке `man netlink`);
- получено уведомление об изменении конфигурационного файла `/etc/resolv.conf` (см. информацию по подсистеме `man inotify`);
- по расписанию каждые 30 секунд.

В ходе отладки администратор может переключать режимы вручную, отправляя процессу sssd сигналы SIGUSR1 (перейти в offline) и SIGUSR2 (перейти в online), но в большинстве случаев достаточно включить/выключить всю службу целиком.

```
root@pc-1:~# sssctl domain-status ald.company.lan | grep status
Online status: Online
root@pc-1:~# kill -SIGUSR1 $(pidof sssd)
root@pc-1:~# sssctl domain-status ald.company.lan | grep status
Online status: Offline
root@pc-1:~# kill -SIGUSR2 $(pidof sssd)
root@pc-1:~# sssctl domain-status ald.company.lan | grep status
Online status: Online
```

5.6.1 Автономный вход по кэшу пароля

Когда пользователь в первый раз входит в компьютер, его аутентификация возможна только через контроллер домена, но при получении успешного статуса PAM_SUCCESS служба sssd сохранит SHA512 хеш в локальной базе, что гарантирует возможность автономной аутентификации этого

пользователя даже в условиях отсутствия связи с сервером. Извлечь хеш пользовательского пароля из ldb-файла и проверить его валидность можно с помощью утилит ldbsearch и openssl.

```
sudo ldbsearch -H /var/lib/sss/db/cache_ald.company.lan.ldb \
  -b name=admin@ald.company.lan,cn=users,cn=ald.company.lan,cn=sysdb cachedPassword
```

Результат выполнения команды:

```
admin@dc-1:~$ sudo ldbsearch -H /var/lib/sss/db/cache_ald.company.lan.ldb \
>   -b name=admin@ald.company.lan,cn=users,cn=ald.company.lan,cn=sysdb cachedPassword
Can't load /etc/samba/smb.conf - run testparm to debug it
asq: Unable to register control with rootdse!
# record 1
dn: name=admin@ald.company.lan,cn=users,cn=ald.company.lan,cn=sysdb
cachedPassword: $6$0auaUM4DmDNk0v9M$ZymggD9uTU1kRjgCVt5b2uWM3f/UHKXxC7u1DpgE87
t.WYDZl97x98PhRP/Ty.YjiSKzdGwSBlnHqX9HItV77.

# returned 1 records
# 1 entries
# 0 referrals
```

Проверить актуальность этого хэша можно с помощью утилиты openssl:

```
admin@pc-1:~$ openssl passwd -6 -salt '0auaUM4DmDNk0v9M' 'Pa$$w0rd'
$6$0auaUM4DmDNk0v9M$ZymggD9uTU1kRjgCVt5b2uWM3f/UHKXxC7u1DpgE87t.WYDZl97x98PhRP/Ty.
↪YjiSKzdGwSBlnHqX9HItV77.
```

Функция хранения кэшей включена по умолчанию, за это отвечает параметр cache_credentials из файла sssd.conf. Срок хранения паролей не ограничен, так как параметр offline_credentials_expiration не задан.

Для первого входа в систему компьютер должен иметь доступ к контроллеру домена, что легко обеспечить из офиса, подключив устройство кабелем к локальной сети, но трудно выполнимо, если устройство нужно передать сотруднику для удаленной работы через службу доставки. В этом случае может оказаться, что офисная сеть будет ему недоступна до тех пор, пока он не войдет в систему и не подключится через домашний Wi-Fi.

Для решения этой проблемы в составе sssd-tools есть утилита sss_seed, которая позволяет имитировать первый вход, принудительно устанавливая ldb-кэш пароля:

```
root@pc-1:~# echo 'Pa$$w0rd' > /tmp/ssd-pwd.txt
root@pc-1:~# sss_seed --domain ALD.COMPANY.LAN \
--username ivan.kuznetsov --password-file /tmp/ssd-pwd.txt
Temporary password added to cache entry for admin@ald.company.lan
```

Имя пользователя должно быть известно в домене, иначе команда не сможет получить его идентификатор и другие сведения. Если на момент выполнения команды пользователь уже входил в это устройство, то утилита переопределит кэш его пароля.

При получении ноутбука пользователю нужно будет войти в систему, подключиться к корпоратив-

ной сети по VPN и сменить пароль, используя утилиту `fly-passwd`, о которой мы уже говорили ранее. Преимущество этой утилиты заключается в том, что она позволяет не только изменить пароль в домене, но и обновить кэш службы sssd, чтобы в дальнейшем пользователь мог входить в операционную систему компьютера, используя свой новый пароль.

5.6.2 Автоматическая Kerberos-аутентификация при переходе в онлайн режим

Хеш SHA512 отлично подходит для долгосрочного хранения секретов на диске, но с его помощью не получится пройти Kerberos-аутентификацию в домене, поэтому при входе пользователя в систему в автономном режиме служба sssd помещает текст пользовательского пароля в связку ключей ядра Linux, чтобы иметь возможность выписать TGT-билет, когда связь с сервером будет восстановлена. За возможность автоматической Kerberos-аутентификации пользователя при переключении в онлайн режим отвечает параметр `krb5_store_password_if_offline`, который по умолчанию равен `True`.

Вы должны понимать, что этот механизм подойдет не всем организациям, поскольку, обладая привилегиями суперпользователя, вы теоретически можете извлечь это значение из связки ключей, например, с помощью утилиты `keyctl`. Следующие действия нужно выполнять в контексте процесса службы sssd, для чего к процессу нужно будет подключиться, например, отладчиком **gdb** (*англ. GNU Debugger*). Для возможности использования отладчика потребуется включить функцию `ptrace`, которая в Astra Linux по умолчанию отключена по соображениям безопасности.

```
admin@dc-1:~$ sudo apt install gdb
admin@dc-1:~$ sudo astra-ptrace-lock disable
admin@dc-1:~$ sudo reboot
```

После перезагрузки выполните вход в систему тем же пользователем и просмотрите: 1. номер pid службы sssd для подключения к процессу Монитора 2. номер текущего окна терминала для перенаправления стандартного вывода.

Сетевая связанность компьютера с контроллером домена в этот момент должна быть заблокирована, т. к. после получения TGT-билета пароль больше не требуется и будет удален из связки ключей.

```
admin@dc-1:~$ sudo pidof sssd
487

admin@dc-1:~$ tty
/dev/pts/0
```

Теперь необходимо подключиться к процессу, чтобы извлечь содержимое ключа.

```
sudo gdb -p 487
```

```
GNU gdb (AstraLinuxSE 8.2.1-2) 8.2.1
...
(gdb) call system("keyctl show > /dev/pts/0")
[Detaching after fork from child process 7829]
Session Keyring
  84684334 --alswrv      0     0  keyring: _ses
 835659210 --alswrv      0     0   _ user: admin@ald.company.lan
$1 = 0
(gdb) call system("keyctl print 835659210 > /dev/pts/0")
[Detaching after fork from child process 7834]
Pa$$w0rd
$2 = 0
```

Важно обратить внимание, что технология автоматической Kerberos-аутентификации при переходе в онлайн режим является условно безопасной по следующим причинам:

- пароль хранится в оперативной памяти и не сохраняется на диске при выключении компьютера;
- пароль хранится в системе ограниченное время и автоматически удаляется при появлении связи с контроллером домена;
- извлечь пароль может только суперпользователь при включенной функции `ptrace`.

5.7 Динамическое обновление DNS-записей в домене

Компьютеры в домене взаимодействуют между собой по доменным именам, поэтому в каталоге важно иметь актуальную информацию по IP-адресам, которые закреплены за этими хостами. С серверами обычно не возникает проблем, т. к. им назначаются статические адреса, а вот интерфейсы рабочих станций настраиваются динамически, поэтому администраторам нужно обеспечить такое же динамическое обновление этой информации в каталоге.

Обновление DNS-записей возможно как по инициативе DHCP-сервера (*Remote Name Daemon Control, RNDC*), так и по инициативе рабочей станции (*GSS-TSIG*). У каждого из подходов есть свои сильные и слабые стороны, но второй способ видится наиболее простым и безопасным, так как в этом случае компьютеры отправляют запросы от своего имени с подтверждением аутентификации по протоколу Kerberos.

Для настройки динамического обновления DNS-записей в конфигурационном файле `/etc/sssd/sssd.conf` на стороне рабочей станции нужно добавить следующие параметры:

```
[domain/ad.example.com]
...
dyndns_update = true
dyndns_refresh_interval = 43200
dyndns_update_ptr = true
```

(continues on next page)

(продолжение с предыдущей страницы)

```
dyndns_ttl = 3600
...
```

После внесения указанных изменений можно перезапустить службу sssd и проверить обновление записей в каталоге при изменении IP-адреса на хосте.

Централизованное управление параметрами динамического обновления DNS можно реализовать с помощью дополнительного параметра групповой политики dyndns.

5.8 Отладка sssd службы и Kerberos

Служба sssd предоставляет две основные функции: получение информации о пользователях и их аутентификацию. Каждая из этих функций использует свои программные интерфейсы, поэтому должна рассматриваться отдельно. Однако успешная аутентификация возможна, только если информация о пользователе получена успешно, поэтому если что-то не работает, то в первую очередь нужно убедиться, что возможно получить хотя бы информацию о пользователе с помощью команд getent и id.

```
getent passwd $USER
```

Результат выполнения:

```
admin:*:491200000:491200000:Administrator:/home/admin:/bin/bash
```

Выполним команду id:

```
id $USER
```

Результат выполнения:

```
uid=491200000(admin) gid=491200000(admins) группы=491200000(admins),1001(astra-admin),
↪491200003(lpadmin),1005611117(ald trust admin)
```

5.8.1 Журналы отладки sssd

Служба sssd состоит из множества компонентов, и для настройки каждого из них в конфигурационном файле предназначена отдельная секция (например, за настройки монитора отвечает секция [sssd], а бэкенд настраивается в секции [domain/...]), поэтому для включения отладки конкретного компонента в его секции следует задать параметр «debug_level=N», где N - это число от 1 до 10.

```
[domain/ald.company.lan]
debug_level = 10
```

(continues on next page)

(продолжение с предыдущей страницы)

```
...
[sssd]
debug_level = 10
...
[nss]
debug_level = 10
...
[pam]
debug_level = 10
...
```

Уровни отладки 1-3 регистрируют сбои, а уровни 8-10 обеспечивают излишнюю детализацию, которая затрудняет быстрый анализ журналов, поэтому для решения большинства проблем начать лучше с шестого уровня.

Уровень отладки можно менять не только через конфигурационный файл, но и «на лету» без перезагрузки службы sssd, отправляя ей команды через D-Bus с помощью утилиты `sssctl` из пакета **sssd-tools**.

Очистим логи, чтобы было удобно смотреть отладочную информацию:

```
admin@dc-1:~$ sudo sssctl logs-remove
```

Изменим уровень отладки и просмотрим в лог-файле изменение `grep 'Debug level changed'`:

```
admin@dc-1:~$ sudo sssctl debug-level 6
admin@dc-1:~$ sudo cat /var/log/sssd/sssd.log | grep 'Debug level changed'
```

В результате выполнения мы видим строку интересующего нас события:

```
113 ... [sssd] [server_common_rotate_logs] (0x0010): Debug level changed to 0x07f0
```

Повысим уровень логирования до 8:

```
sudo sssctl debug-level 8
sudo cat -n /var/log/sssd/sssd.log | grep 'Debug level changed'
```

Таким образом вы можете отметить начало и конец отладки:

```
113 ...[sssd] [server_common_rotate_logs] (0x0010): Debug level changed to 0x07f0
157 ...[sssd] [server_common_rotate_logs] (0x0010): Debug level changed to 0x37f0
```

Вывести на экран диапазон строк можно с помощью утилиты sed:

```
sudo cat -n /var/log/sssd/sssd.log | sed '113,157!d'
```

В операционной системе Astra Linux журналы отладки хранятся в папке `/var/log/sssd`, по одному файлу журнала на каждый компонент. Ответчики пишут в файлы с именами `sssd_<имя_сервиса>`, например, Ответчик NSS пишет в файл `/var/log/sssd/sssd_nss.log`. Сооб-

щения от Бэкенда пишутся в файл с именем `sssd_<имя_домена>.log`. Есть еще журналы вспомогательных процессов, таких как `ldap_child.log` и `krb5_child.log`.

Прежде чем погружаться в изучение журналов и файлов конфигурации, крайне желательно внимательно ознакомиться с описанием архитектуры службы sssd.

Примечание: Во многих случаях полезно воспроизвести ошибку после очистки кэша, чтобы исключить его влияние, но помните, что в локальной базе кэшируются также и учетные данные, поэтому не удаляйте кэш, если вы работаете в автономном режиме. А также обратите внимание, что приведенные примеры сообщений отладки могут меняться для последующих релизов.

5.8.2 Общие рекомендации отладки sssd

Если команды `getent` или `id` не выводят вообще никакой информации о пользователе или группе:

- Проверьте работу службы разрешения имен командой `ping dc-1`, и указаны ли в `/etc/resolv.conf` правильные DNS-серверы.
- Проверьте, работает ли служба sssd с помощью команды `systemctl status sssd`.
- Проверьте конфигурацию `/etc/nsswitch.conf`, что модуль `sss` указан для всех нужных баз данных: `passwd`, `group`, `shadow`, `services`, `netgroup`, `sudoers`, `automount`.
- Проверьте поведение команды `id` на контроллере домена ALD Pro.
- Проверьте, доходит ли запрос до Ответчика. Для этого в разделе *[nss]* установите `debug_level = 6` и перезапустите службу sssd.

Приведем пример успешного запроса информации по пользователю:

```
[sssd[nss]] [get_client_cred] (0x4000): Client creds: euid[10327] egid[10327]␣
↪pid[18144].
[sssd[nss]] [setup_client_idle_timer] (0x4000): Idle timer re-set for client␣
↪[0x13c9960][22]
[sssd[nss]] [accept_fd_handler] (0x0400): Client connected!
[sssd[nss]] [sss_cmd_get_version] (0x0200): Received client version [1].
[sssd[nss]] [sss_cmd_get_version] (0x0200): Offered version [1].
[sssd[nss]] [nss_getby_name] (0x0400): Input name: admin
```

Если команда достигает Ответчика NSS, проверьте что она передается Поставщику данных (Data Provider). Неудачный запрос может выглядеть так:

```
[sssd[nss]] [cache_req_search_dp] (0x0400): CR #3: Looking up [admin@ipa.test] in␣
↪data provider
```

(continues on next page)

(продолжение с предыдущей страницы)

```
[sssd[nss]] [sss_dp_issue_request] (0x0400): Issuing request for␣
↪[0x41e51c:1:admin@ipa.test@ipa.test]
[sssd[nss]] [sss_dp_get_account_msg] (0x0400): Creating request for
[ipa.test][0x1][BE_REQ_USER][name=admin@ipa.test:-]
[sssd[nss]] [sss_dp_internal_get_send] (0x0400): Entering request␣
↪[0x41e51c:1:admin@ipa.test@ipa.test]
[sssd[nss]] [sss_dp_req_destructor] (0x0400): Deleting request:
[0x41e51c:1:admin@win.trust.test@win.trust.test]
[sssd[nss]] [sss_dp_get_reply] (0x0010): The Data Provider returned an error
[org.freedesktop.sssd.Error.DataProvider.Offline]
[sssd[nss]] [cache_req_common_dp_recv] (0x0040): CR #3: Data Provider Error: 3, 5,␣
↪Failed to get reply
from Data Provider
[sssd[nss]] [cache_req_common_dp_recv] (0x0400): CR #3: Due to an error we will␣
↪return cached data
```

Успешно обработанный запрос, напротив, будет выглядеть так:

```
[sssd[nss]] [cache_req_search_dp] (0x0400): CR #3: Looking up [admin@ipa.test] in␣
↪data provider
[sssd[nss]] [sss_dp_issue_request] (0x0400): Issuing request for␣
↪[0x41e51c:1:admin@ipa.test@ipa.test]
[sssd[nss]] [sss_dp_get_account_msg] (0x0400): Creating request for
[ipa.test][0x1][BE_REQ_USER][name=admin@ipa.test:-]
[sssd[nss]] [sss_dp_get_reply] (0x1000): Got reply from Data Provider - DP error␣
↪code: 0 errno: 0 error
message: Success
[sssd[nss]] [cache_req_search_cache] (0x0400): CR #3: Looking up [admin@ipa.test] in␣
↪cache
```

Если запрос к Поставщику данных успешно завершен, но вы по-прежнему не видите никаких результатов, следует переходить к журналам Бэкенда.

5.8.3 Отладка Бэкенда sssd

Бэкенд выполняет несколько различных операций, поэтому бывает трудно сразу понять, в чем заключается проблема. На самом высоком уровне бэкенд выполняет следующие шаги в указанном порядке:

- Бэкенд получает запрос от клиентской библиотеки и решает, к какому серверу следует подключиться для его обработки. Этот шаг может включать в себя поиск по SRV-записям.
- Бэкенд устанавливает соединение с сервером и проходит аутентификацию, используя учетные данные из keytab-файла хоста, если это требуется.
- Как только соединение установлено, бэкенд отправляет запрос на поиск информации, поэтому вы должны увидеть в запросе критерии фильтрации, базовую запись и запрашиваемые атрибуты.

- После того, как поиск завершится, соответствующие записи будут сохранены в кэше. Код состояния возвращается собеседнику.

  Установите `debug_level=6` в разделе `[domain/]`, перезапустите службу и выполните еще раз неудачный поиск информации о пользователе. При отладке работы Бэкенда в первую очередь убедитесь, что бэкенд находится в режиме онлайн. Сделать это можно с помощью утилиты `sssctl`:

```
sudo sssctl domain-status ald.company.lan
```

Результат выполнения:

```
Online status: Online

Active servers:
IPA: dc-1.ald.company.lan

Discovered IPA servers:
- dc-1.ald.company.lan
```

Проверьте, можно ли установить соединение с теми же параметрами безопасности, которые использует sssd:

```
admin@dc-1:~$ kinit -k && klist
Ticket cache: KEYRING:persistent:491200000:krb_ccache_pMxi5Bu
Default principal: host/dc-1.ald.company.lan@ALD.COMPANY.LAN

Valid starting       Expires              Service principal
02.04.2025 23:15:25  03.04.2025 23:15:25  krbtgt/ALD.COMPANY.LAN@ALD.COMPANY.LAN
```

Проверим запрос в LDAP, где Ключи `-H` и `-ZZ` делают запрос ближе к тому, как служба sssd взаимодействует с каталогом:

```
ldapsearch -ZZ -H ldap://dc-1.ald.company.lan \
  -b uid=admin,cn=users,cn=accounts,dc=ald,dc=company,dc=lan cn
```

Результат выполнения:

```
...
# admin, users, accounts, ald.company.lan
dn: uid=admin,cn=users,cn=accounts,dc=ald,dc=company,dc=lan
cn: Administrator
```

Если вы установите для Бэкенда десятый уровень отладки, то информация о трассировке Kerberos-аутентификации будет отражаться в файле `ldap_child.log`, но проще ее получить напрямую от утилиты `kinit`, если определить значение переменной окружения `KRB5_TRACE`:

```
set +o history
env KRB5_TRACE=/dev/stdout kinit <<< 'Pa$$w0rd'
set -o history
```

Результат выполнения:

```
[23001] 1696515238.426608: Getting initial credentials for admin@ALD.COMPANY.LAN
[23001] 1696515238.426610: Sending unauthenticated request
[23001] 1696515238.426611: Sending request (191 bytes) to ALD.COMPANY.LAN
[23001] 1696515238.426612: Initiating TCP connection to stream 10.0.1.11:88
[23001] 1696515238.426613: Sending TCP request to stream 10.0.1.11:88
[23001] 1696515238.426614: Received answer (514 bytes) from stream 10.0.1.11:88
[23001] 1696515238.426615: Terminating TCP connection to stream 10.0.1.11:88
[23001] 1696515238.426616: Response was from master KDC
...
[23001] 1696515238.426651: Storing admin@ALD.COMPANY.LAN ->
krb5_ccache_conf_data/pa_type/krbtgt\/ALD.COMPANY.LAN\@ALD.COMPANY.LAN@X-CACHECONF:␣
↪in KEYRING:persistent:491200000:krb_ccache_kFlzpn6``
```

Проверьте, присутствуют ли на сервере все атрибуты, необходимые для службы sssd, правильно ли указана базовая запись для поиска, особенно для доверенных поддоменов. По возможности попробуйте выполнить такой же запрос вручную с помощью утилиты `ldapsearch`.

5.8.4 Устранение неисправностей аутентификации, изменения пароля и контроля доступа

Если информация о пользователе может быть успешно получена, но аутентификация не проходит, то в первую очередь нужно смотреть журнал `/var/log/auth.log` на предмет сообщений от **pam_sss**.

Аутентификация начинается в PAM-стеке на фазе auth и выполняется с помощью провайдера аутентификации (auth_provider) службы sssd. Однако не все запросы на аутентификацию проходят через sssd, например, аутентификация по SSH происходит непосредственно в SSHD, а sssd взаимодействует с PAM-стеком только на фазе **account**.

Контроль доступа начинается в PAM-стеке на фазе account и выполняется с помощью провайдера доступа (access_provider) службы sssd.

И, наконец, смена пароля начинается на стороне PAM-стека на фазе **password** и выполняется с помощью провайдера смены пароля (chpass_provider) службы sssd.

Аутентификация через PAM-стек происходит по следующему шаблону:

- Приложение с поддержкой PAM-аутентификации начинает диалог. В зависимости от настроек PAM-стека в диалог может быть вовлечен модуль pam_sss. Для отладки процесса аутентификации сначала проверьте журнал `/var/log/auth.log` на предмет того, есть ли вообще обращения к pam_sss. Если вы не видите упоминания pam_sss, скорее всего, ваш PAM-стек настроен неправильно. Если вы видите, что есть обращения к pam_sss, включите отладку для Ответчика PAM.

- Ответчик PAM-службы sssd получает запрос на аутентификацию и в большинстве случаев пересылает его Бэкенду. Обратите внимание, что в отличие от запросов на идентификацию,

запросы на аутентификацию и контроль доступа обычно не кэшируется и всегда завершаются обращением к серверу. В некоторых случаях это может приводить к задержкам, но это достаточно важно, потому что дополнительные группы в GNU/Linux устанавливаются только в момент входа в систему.

Журналы Ответчика PAM должны показывать запросы, которые пришли от PAM-стека и были перенаправлены на бэкенд. Если вы видите запрос на аутентификацию к Ответчику PAM, но получаете ошибку от бэкенда, проверьте журналы бэкенда. Пример ошибки может выглядеть как:

```
[sssd[pam]] [sss_dp_issue_request] (0x0400): Issuing request for␣
↪[0x411d44:3:admin@ipa.example.com]
[sssd[pam]] [sss_dp_get_account_msg] (0x0400): Creating request for [ipa.example.
↪com][3][1][name=admin]
[sssd[pam]] [sss_dp_internal_get_send] (0x0400): Entering request␣
↪[0x411d44:3:admin@ipa.example.com]
[sssd[pam]] [sss_dp_get_reply] (0x1000): Got reply from Data Provider - DP error␣
↪code: 1 errno: 11 error
message: Offline
[sssd[pam]] [pam_check_user_dp_callback] (0x0040): Unable to get information from␣
↪Data Provider Error: 1,
11, Offline
```

- Бэкенд обрабатывает запрос. Это может включать эквивалент выполнения команды `kinit` в процессе `krb5_child`, аутентификацию по LDAP или проверку по списку контроля доступа. После того как запрос к бэкенду завершится, результат отправляется обратно собеседнику PAM.

  Посмотрите также файл `krb5_child.log`. Если установить уровень отладки `debug_level = 10`, то в этот файл будет включаться также низкоуровневая информация для трассировки работы протокола Kerberos. Вы также можете имитировать аутентификацию вручную с помощью утилиты `kinit`.

- Ответчик PAM получает результат и пересылает его обратно в модуль `pam_sss`. Сообщения об ошибке или статусе отображаются в `/var/log/auth.log`.

5.9 Расположение значимых файлов

В следующей таблице представлена сводная информация по наиболее полезным файлам для отладки компонентов системы.

табл. 2.7 – Расположение файлов журналов

Компонент системы	Конфигурация	Файл журнала
sssd		
SSSD	/etc/sssd/sssd.conf	/var/log/sssd/
PAM	/etc/pam.d/system-auth	/var/log/sssd/sssd_pam.log
NSS	/etc/nsswitch.conf	/var/log/sssd/sssd_nss.log
krb5_child	/etc/sssd/sssd.conf	/var/log/sssd/krb5_child.log
ldap_child	/etc/sssd/sssd.conf	/var/log/sssd/ldap_child.log
Data Provider (Backend)	/etc/sssd/sssd.conf	/var/log/sssd/sssd_<domainFQDN>.log
DNS		
OS	/etc/resolv.conf	Журнала нет, но в качестве альтернативы можно использовать команду `sudo tcpdump port 53`
BIND9	/etc/bind/named.conf	journalctl -u bind9-pkcs11.service
SAMBA		
Samba	/etc/samba/smb.conf	/var/log/samba/*.log
Winbind	/etc/security/pam_winbind.conf	/var/log/samba/log.*
Kerberos		
Kerberos	/etc/krb5.conf	/var/log/sssd/krb5_child.log
SaltStack		
Salt-master	/etc/salt/	/var/log/salt/master
Salt-minion	/etc/salt/	/var/log/salt/minion

5.10 Практика и тестирование

5.10.1 Практическая работа: Модуль 5. Работа Linux-компьютера в домене

См. также *Требования, правила и цели выполнения практической работы*

5.10.1.1 Практические задания

Задание 1.

1. Выясните время хранения записей в быстром кэше sssd.
2. Сравните время выполнения команды `id` для доменного пользователя до и после очистки кэша.

3. Выясните, какой поставщик данных пользуется сервисом sssd.
4. Выясните, где и в какой последовательности будет производиться поиск данных о правилах SUDO для доменного пользователя.

Задание 2 Отладка SSSD

1. Проверьте команду `getent passwd $USER`.
2. Проверьте команду `id`.
3. Остановите службу sssd.
4. Проверьте команду `id ppetrov`.
5. Переключитесь на другой терминал tty4 горячим сочетанием клавиш Ctrl + Alt + F4.
6. Выполните вход от локального пользователя `localadmin`.
7. Запустите службу sssd.
8. Проверьте команду `id ppetrov`.
9. Измените `debug level` на 6.
10. Измените `debug level` на 8.
11. Проверьте лог между изменением уровней с 6 на 8.

Задание 3 KRB5_TRACE

1. Проверьте `kinit` через `KRB5_TRACE` верным паролем.
2. Проверьте `kinit` через `KRB5_TRACE` ошибочным паролем.
3. Сделайте `kdestroy` и проверьте `klist` через `KRB5_TRACE`.
4. Сделайте вход `kinit` и проверьте `klist` через `KRB5_TRACE`.

5.10.1.2 Практическая работа (с ответами)

Задание 1.

1. Выясните время хранения записей в быстром кэше sssd.

```
admin@dc-1:~$ sudo grep memcache_timeout /etc/sssd/sssd.conf
```

2. Сравните время выполнения команды `id` для доменного пользователя до и после очистки кэша.

```
admin@dc-1:~$ time id admin
...
real    0m0,003s
user    0m0,002s
sys     0m0,000s

admin@dc-1:~$ sudo -i
root@dc-1:~# rm -rf /var/lib/sss/db/*; systemctl restart sssd
root@dc-1:~# time id admin
...
real    0m0,172s
user    0m0,002s
sys     0m0,000s
```

3. Выясните, какой поставщик данных пользуется сервисом sssd.

```
admin@dc-1:~$ sudo grep id_provider /etc/sssd/sssd.conf
id_provider = ipa
hostid_provider = ipa
```

4. Выясните, где и в какой последовательности будет производиться поиск данных о правилах SUDO для доменного пользователя.

```
admin@dc-1:~$ sudo grep sudoers /etc/nsswitch.conf
sudoers: files sss
admin@dc-1:~$ man nsswitch.conf
```

Задание 2.

1. Проверьте команду `getent passwd $USER`.

```
admin@dc-1:~$ getent passwd $USER
admin:*:491200000:491200000:Administrator:/home/admin:/bin/bash
```

2. Проверьте команду `id`.

```
admin@dc-1:~$ id
uid=491200000(admin) gid=491200000(admins) группы=491200000(admins),113(lpadmin),
↪1001(astra-admin),1005611117(ald trust admin)
```

3. Остановите sssd.

```
admin@dc-1:~$ systemctl stop sssd
```

4. Проверьте команду `id ppetrov`.

```
admin@dc-1:~$ id ppetrov
id: «ppetrov»: такого пользователя нет
```

5. Переключитесь на другой терминал tty4 (Ctrl + Alt + F4).

6. Выполните вход от локального пользователя `localadmin`.

7. Запустите службу sssd.

```
localadmin@dc-1:~$ sudo systemctl start sssd
```

8. Переключитесь обратно на терминал tty7 (Ctrl + Alt + F7)

9. Проверьте команду `id ppetrov`.

```
admin@dc-1:~$ id ppetrov
uid=491200010(ppetrov) gid=491200010(ppetrov) группы=491200010(ppetrov)
```

10. Измените `debug level` на 6.

    Очистите логи, чтоб удобно было смотреть отладочную информацию:

```
admin@dc-1:~$ sudo apt install sssd-tools
admin@dc-1:~$ sudo sssctl logs-remove
Truncating log files...
```

    Измените уровень отладки на 6 и просмотрите в лог файле изменение grep „Debug level changed“

```
admin@dc-1:~$ sudo sssctl debug-level 6
admin@dc-1:~$ sudo cat /var/log/sssd/sssd.log | grep 'Debug level changed'
... [sssd] [server_common_rotate_logs] (0x0010): Debug level changed to␣
↪0x07f0
```

11. Измените `debug level` на 8.

```
sudo sssctl debug-level 8
sudo grep 'Debug level changed' /var/log/sssd/sssd.log
```

    Результат выполнения команды:

```
113 ... [sssd] [server_common_rotate_logs] (0x0010): Debug level changed to␣
↪0x07f0
157 ... [sssd] [server_common_rotate_logs] (0x0010): Debug level changed to␣
↪0x37f0
```

    В фильтре лога мы можем видеть два события: начало и конец. Это нужно для удобного просмотра лога, чтобы знать, где искать интервал отладочной информации от и до:

12. Проверьте лог между изменением уровней с 6 на 8.

Выберете строки между кодами смены уровня отладки **0x07f0** и **0x37f0** можно с помощью regex:

```
sudo cat -n /var/log/sssd/sssd.log | sed '113,157!d'
```

```
113 ... [sssd] [sbus_issue_request_done] (0x0400): org.freedesktop.DBus.
↪GetConnectionUnixUser: Success
114 ... [sssd] [sbus_issue_request_done] (0x0400): org.freedesktop.DBus.
↪GetConnectionUnixUser: Success
115 ... [sssd] [sbus_issue_request_done] (0x0400): org.freedesktop.DBus.
↪GetConnectionUnixUser: Success
116 ... [sssd] [sbus_issue_request_done] (0x0400): org.freedesktop.DBus.
↪GetConnectionUnixUser: Success
117 ... [sssd] [sbus_issue_request_done] (0x0400): org.freedesktop.DBus.
↪GetConnectionUnixUser: Success
118 ... [sssd] [service_signal_done] (0x0010): Unable to signal service [2]: Нет␣
↪такого файла или каталога
119 ... [sssd] [sbus_issue_request_done] (0x0400): org.freedesktop.DBus.
↪GetConnectionUnixUser: Success
120 ... [sssd] [monitor_hup] (0x0020): Received SIGHUP.
121 ... [sssd] [te_server_hup] (0x0020): Received SIGHUP. Rotating logfiles.
...
157 ... [sssd] [server_common_rotate_logs] (0x0010): Debug level changed to admin
```

Задание 3.

1. Проверьте команду `kinit` через `KRB5_TRACE` верным паролем.

```
admin@dc-1:~$ env KRB5_TRACE=/dev/stdout kinit <<< 'Pa$$w0rd'
```

2. Проверьте `kinit` через `KRB5_TRACE` ошибочным паролем.

```
admin@dc-1:~$ env KRB5_TRACE=/dev/stdout kinit <<< 'Pa$$w0rd'
```

3. Сделайте `kdestroy` и проверьте `klist` через `KRB5_TRACE`.

```
admin@dc-1:~$ env KRB5_TRACE=/dev/stdout kdestroy
[31918] 1728039141.300859: Destroying ccache KEYRING:persistent:491200000:krb_
↪ccache_Rcw54Ot
admin@dc-1:~$ env KRB5_TRACE=/dev/stdout kinit <<< 'Pa$$w0rd'
[31948] 1728039158.966301: Getting initial credentials for admin@ALD.COMPANY.LAN
[31948] 1728039158.966303: Sending unauthenticated request
[31948] 1728039158.966304: Sending request (191 bytes) to ALD.COMPANY.LAN
[31948] 1728039158.966305: Initiating TCP connection to stream 10.0.1.11:88
[31948] 1728039158.966306: Sending TCP request to stream 10.0.1.11:88
[31948] 1728039158.966307: Received answer (514 bytes) from stream 10.0.1.11:88
[31948] 1728039158.966308: Terminating TCP connection to stream 10.0.1.11:88
[31948] 1728039158.966309: Response was from master KDC
[31948] 1728039158.966310: Received error from KDC: -1765328359/Additional pre-
↪authentication required
[31948] 1728039158.966313: Preauthenticating using KDC method data
[31948] 1728039158.966314: Processing preauth types: PA-PK-AS-REQ (16), PA-FX-
↪FAST (136), PA-ETYPE-INFO2 (19), PA-PKINIT-KX (147), PA-SPAKE (151), PA-ENC-
```

(continues on next page)

(продолжение с предыдущей страницы)

```
↪TIMESTAMP (2), PA_AS_FRESHNESS (150), PA-FX-COOKIE (133)
[31948] 1728039158.966315: Selected etype info: etype aes256-cts, salt ")AHB␣
↪ko0*Jz9hM7\", params ""
[31948] 1728039158.966316: Received cookie: MIT1\x00\x00\x00\x0157`\x80lf\xe8\
↪xado\xa2\x02\x89\xbe\xd7\x9f\xaf\x888G\xcdg\xaep\x81\x9a\xbb0m\xe2\xb464\x94\
↪xa7\xdc\xd4\x8a\x8a5\xf2n\xacK \xad\\x04#c+`\xaf_\xc8\xf0\xaa\xa2m\xa6\xec\x1d\
↪x16\xdd\x92\x0b\xf0LS,\x99]\xc4\x88\xb8X\xd5\xc6j\xea\x00\xed\x9e\xdcr.$\x84\
↪x16\xa2\x9c\xfa\x8b\x01\x93\x97\x86o\xc3\xb8\x89\xb6\xb2\x19>\xc28^\x82\xb5\
↪xee\x8c\xec\x17$\xe2\x9c\x97B\xec\xc7\x0a\x8f]\xa7*z\xbcoV\xc2
[31948] 1728039158.966317: PKINIT client has no configured identity; giving up
[31948] 1728039158.966318: Preauth module pkinit (147) (info) returned: 0/Success
[31948] 1728039158.966319: PKINIT client received freshness token from KDC
[31948] 1728039158.966320: Preauth module pkinit (150) (info) returned: 0/Success
[31948] 1728039158.966321: PKINIT client has no configured identity; giving up
[31948] 1728039158.966322: Preauth module pkinit (16) (real) returned: 22/
↪Недопустимый аргумент
[31948] 1728039158.966323: SPAKE challenge received with group 1, pubkey␣
↪5C87A100D3781D25AF2ED15C16AE46D9C2E064F24E719428B8D8F741EB2A5CBF
Password for admin@ALD.COMPANY.LAN:
[31948] 1728039158.966324: SPAKE key generated with pubkey␣
↪DE79ED1E4621F5E37676736ECDCC8D5943AD04966E7B701C1FD86F84EBEEC73D
[31948] 1728039158.966325: SPAKE algorithm result:␣
↪3817D848CD21F0851824F647159070BC8D0CC7DF42B90C30884A083094287F09
[31948] 1728039158.966326: SPAKE final transcript hash:␣
↪AA9210425A6D38B424115A89F88708162A29A75B005F5251788E14AD2E0E2C27
[31948] 1728039158.966327: Sending SPAKE response
[31948] 1728039158.966328: Preauth module spake (151) (real) returned: 0/Success
[31948] 1728039158.966329: Produced preauth for next request: PA-FX-COOKIE (133),
↪ PA-SPAKE (151)
[31948] 1728039158.966330: Sending request (450 bytes) to ALD.COMPANY.LAN
[31948] 1728039158.966331: Initiating TCP connection to stream 10.0.1.11:88
[31948] 1728039158.966332: Sending TCP request to stream 10.0.1.11:88
[31948] 1728039159.007419: Received answer (1575 bytes) from stream 10.0.1.11:88
[31948] 1728039159.007420: Terminating TCP connection to stream 10.0.1.11:88
[31948] 1728039159.007421: Response was from master KDC
[31948] 1728039159.007422: AS key determined by preauth: aes256-cts/5567
[31948] 1728039159.007423: Decrypted AS reply; session key is: aes256-cts/22FA
[31948] 1728039159.007424: FAST negotiation: available
[31948] 1728039159.007425: Initializing KEYRING:persistent:491200000:krb_ccache_
↪Rcw54Ot with default princ admin@ALD.COMPANY.LAN
[31948] 1728039159.007426: Storing admin@ALD.COMPANY.LAN -> krbtgt/ALD.COMPANY.
↪LAN@ALD.COMPANY.LAN in KEYRING:persistent:491200000:krb_ccache_Rcw54Ot
[31948] 1728039159.007427: Storing config in KEYRING:persistent:491200000:krb_
↪ccache_Rcw54Ot for krbtgt/ALD.COMPANY.LAN@ALD.COMPANY.LAN: fast_avail: yes
[31948] 1728039159.007428: Storing admin@ALD.COMPANY.LAN -> krb5_ccache_conf_
↪data/fast_avail/krbtgt\/ALD.COMPANY.LAN\@ALD.COMPANY.LAN@X-CACHECONF: in␣
↪KEYRING:persistent:491200000:krb_ccache_Rcw54Ot
[31948] 1728039159.007429: Storing config in KEYRING:persistent:491200000:krb_
↪ccache_Rcw54Ot for krbtgt/ALD.COMPANY.LAN@ALD.COMPANY.LAN: pa_type: 151
[31948] 1728039159.007430: Storing admin@ALD.COMPANY.LAN -> krb5_ccache_conf_
↪data/pa_type/krbtgt\/ALD.COMPANY.LAN\@ALD.COMPANY.LAN@X-CACHECONF: in␣
↪KEYRING:persistent:491200000:krb_ccache_Rcw54Ot
```

4. Сделайте вход `kinit` и проверьте `klist` через `KRB5_TRACE`.

```
admin@dc-1:~$ env KRB5_TRACE=/dev/stdout kinit <<< 'Pa$$w0rd'
...
admin@dc-1:~$ env KRB5_TRACE=/dev/stdout klist
Ticket cache: KEYRING:persistent:491200000:krb_ccache_Rcw54Ot
Default principal: admin@ALD.COMPANY.LAN

Valid starting       Expires              Service principal
02.04.2025 13:52:38  02.04.2025 13:52:38  krbtgt/ALD.COMPANY.LAN@ALD.COMPANY.LAN
```

5.10.2 Тест: Модуль 5. Работа Linux-компьютера в домене

Пожалуйста, ответьте на несколько вопросов, чтобы закрепить полученные знания. Правильные ответы для самопроверки вы сможете найти в конце списка. Обратите внимание, что у некоторых вопросов может быть несколько верных вариантов ответов.

5.10.2.1 Вопросы

Вопрос 1. Какая библиотека sssd отвечает за работу компьютера в домене ALD Pro?

a. libsss_ipa.so

b. libsss_files.so

c. libsss_ldap.so

d. libsss_ad.so

Вопрос 2. Какой ключ служит для указания адреса LDAP-сервера при вызове ldapsearch?

a. Ключ `-W`

b. Ключ `-ZZ`

c. Ключ `-o`

d. Ключ `-H`

Вопрос 3. Как расшифровывается аббревиатура KDC?

a. Kodak Digital Camera

b. Key Distribution Center

c. Knowledge Development Center

d. Kurt Donald Cobain

Вопрос 4. Аутентификация по протоколу Kerberos будет успешной, если время сервера и клиента расходятся менее, чем на:

a. 5 минут

b. 10 минут

c. 15 минут

d. 30 минут

Вопрос 5. Кому клиентское приложение передает подписанный челлендж при выполнении аутентификации по протоколу NTLM:

a. Контроллеру домена

b. Целевому сервису

Вопрос 6. Какие команды позволяют проверить работу sssd?

a. Команда `passwd`

b. Команды `getent passwd $USER` и `id`

c. Команда `ipactl status`

d. Команда `kinit`

Вопрос 7. Какая команда позволяет изменить уровень отладки для службы sssd?

a. Команда `sssdctl status`

b. Команда `systemctl status sssd`

c. Команда `sudo sssctl logs-remove`

d. Команда `sudo sssctl debug-level 6`

Ответы на вопросы

1. a
2. d
3. b
4. a

5. b

6. b

7. d

5.11 Заключение

В этом модуле мы рассмотрели архитектуру службы sssd и подробно разобрали протокол Kerberos, поэтому у нас теперь есть целостное представление о том, как именно работает Linux-компьютер в составе домена, что поможет в отладке любых нештатных ситуаций.

Следующий модуль будет уже про групповые политики, и вы узнаете, как с помощью ALD Pro можно экономить силы и время, если вам требуется настроить сотни и тысячи рабочих станций.

5.12 Дополнительные источники информации

- Статья на Хабре Windows-компьютеры в составе домена ALD Pro.
- Рекомендуем ознакомиться также с видеороликами нашего хорошего товарища на канале «ИТ-Проповедник». Учитывайте, что ролики записаны для версии 2.2.1.
  - Служба sssd в ALD Pro, отладка службы sssd и Kerberos
  - Виды аутентификации в ALD Pro - LDAP, Kerberos, NTLM

5.13 Обратная связь

Если остались вопросы, то их всегда можно задать в специальной теме.

Модуль 6. Управление политиками ALD Pro

6.1 Введение

В этом модуле мы обсудим, что такое групповые политики и как они позволяют экономить время при выполнении задач администрирования большой ИТ-инфраструктуры.

Мы сравним техническую реализацию политик MS AD с политиками FreeIPA и ALD Pro, выполним настройку и отладку политик с использованием параметров, доступных из коробки, а также посмотрим, как создаются дополнительные параметры, с помощью которых можно организовать централизованное управление настройками любых корпоративных приложений.

После изучения материалов этого модуля вы больше никогда не будете «администрировать ногами» то, что можно настроить централизованно через групповые политики.

6.2 Концепция групповых политик

Как известно, уникальные продукты никому не нужны, а у всех востребованных решений быстро появляются аналоги, поэтому, создавая политики, разработчики FreeIPA и ALD Pro ориентировались в первую очередь на передовой опыт Microsoft.

Для новых администраторов слово «политика» может показаться странным, т.к. обычно оно ассоциируется с чем-то серьезным про менеджмент на уровне государства, а в английском языке это слово означает просто «правила», например, «Правила поведения учащихся» у них называются «Школьной политикой» (*англ. School policies*).

Таким образом, «политики» в Windows — это правила, в соответствии с которыми отдельный компьютер выполняет настройку рабочей среды, а «групповыми» они называются потому, что мы можем группировать эти правила в так называемые объекты. Согласитесь, если бы технология называлась «сгруппированные правила», то было бы на порядок проще, хотя и не так красиво.

В домене Active Directory объекты групповой политики можно назначить на домены, сайты и структурные подразделения, настраивая типовым образом окружение для сотен и даже тысяч пользователей и компьютеров сразу. Это позволяет многократно снижать расходы на администрирование ИТ-инфраструктуры, чем и объясняется популярность групповых политик.

В настоящий момент сложно себе представить предприятие, где было бы более тысячи компьютеров и не применялись бы групповые политики. Поэтому навыки управления групповыми политиками являются обязательным минимумом для администратора домена уровня Senior.

6.3 Политики паролей

Пароль представляет собой набор символов, который должен быть известен только самому пользователю и проверяющей стороне, поэтому, если пользователь может предъявить доказательство того, что пароль ему известен, это является подтверждением аутентичности пользователя, что он именно тот, за кого себя выдает.

Пароли являются самым простым, но при этом не самым безопасным способом аутентификации,

так как их можно подобрать или перехватить, а если не устанавливать дополнительных ограничений по длине и сложности, то каждый двадцатый сотрудник в организации окажется счастливчиком, который угадает комбинацию из Топ-100, типа, 123456, password или qwerty.

Совокупность требований к паролям называют также политиками паролей, и чем строже эти требования, тем сложнее злоумышленнику подобрать пароль и воспользоваться результатами успешной атаки, но вместе с тем сложнее и пользователям работать в таком домене. Поэтому для разных категорий пользователей следует устанавливать разные требования, обеспечивающие баланс между удобством и безопасностью. И в этом вопросе главное — достичь компромисса с ИБ, поскольку, как известно, если жена хочет на море, а муж — в горы, то все едут загорать, но главе семейства разрешают взять с собой лыжи.

Изменить пароль можно по протоколу KPASSWD (464/TCP), через расширенную операцию LDAP+StartTLS (389/TCP) и просто записью нового значения в атрибут userPassword, опять же, по протоколу LDAP. Однако вне зависимости от того, какой способ будет выбран, на сервер передается именно сам пароль, а не его хеш. Это позволяет проверить длину пароля, сложность и другие параметры, после чего сгенерировать и сохранить совместимые хеши.

6.3.1 Политики паролей Windows

Раньше в домене Active Directory можно было задать только одну политику паролей через объект групповой политики Default Domain Policy. Да, вы можете создать несколько объектов и в каждом из них определить параметры из секции Password Policy, но ко всем пользователям в домене все равно будут применяться только те требования, которые определены в Default Domain Policy («Костыль?» — спросите вы. Не, ну в продуктах Microsoft такого не может быть!).

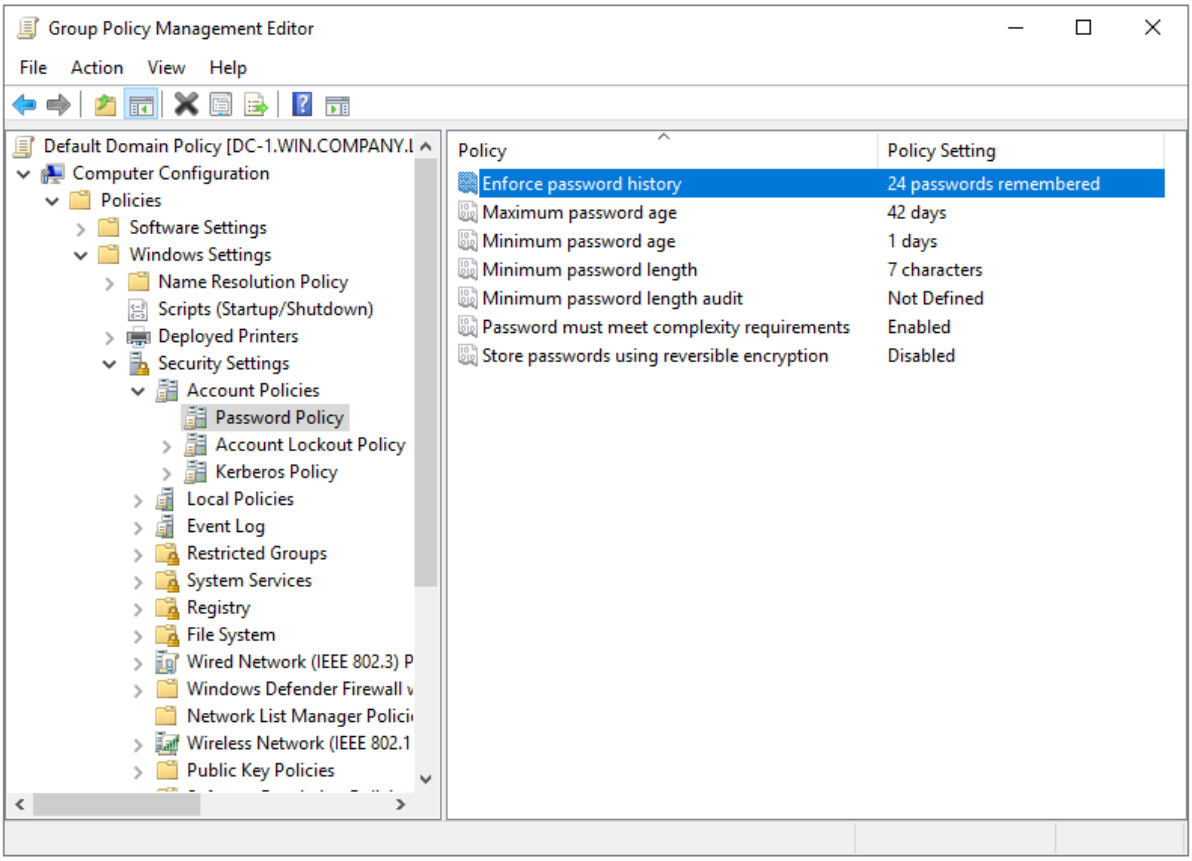


рис. 2.121 – Политика паролей по умолчанию (Default Domain Policy)

Начиная с Windows Server 2008, в MS AD появились так называемые детальные политики паролей (Fine-Grained Password Policy, FGPPs), которые настраиваются в Центре администрирования Active Directory (Active Directory Administrative Center) путем создания контейнера настроек пароля (Password Settings Container). Каждый контейнер может быть назначен конкретным пользователям и группам, переопределяя их требования к паролям.

Следует обратить внимание также на атрибут Precedence, который определяет приоритет данной политики паролей. Если на пользователя AD действуют несколько политик, то к нему будет применена политика с меньшим значением приоритета.

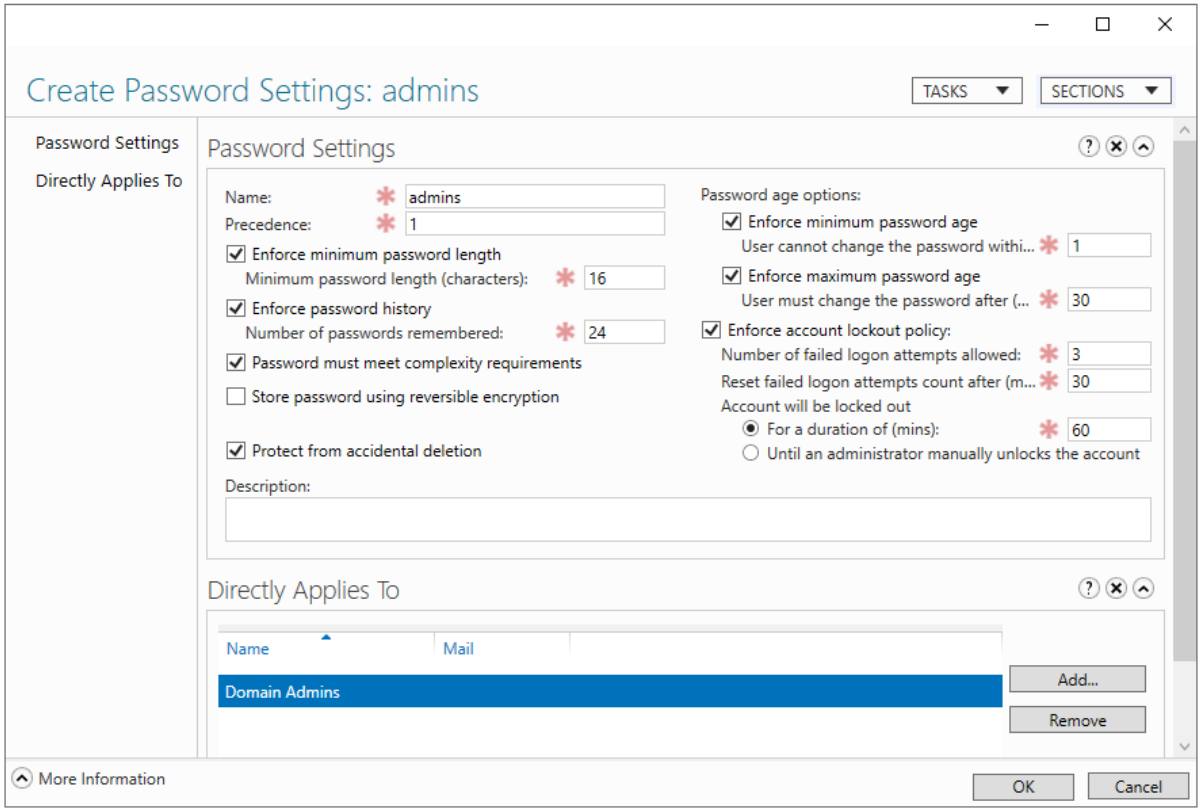


рис. 2.122 – Детальная политика паролей (Fine-Grained Password Policy, FGPPs)

6.3.2 Политики паролей FreeIPA

В службе каталога FreeIPA политики паролей устроены очень похожим образом: есть глобальная политика по умолчанию и можно создавать дополнительные политики для отдельных групп пользователей. Учитывая то, что пользователь может входить сразу в несколько групп, алгоритм проверки выглядит следующим образом:

- Из контейнера «cn=cosTemplates,cn=accounts» отбираются шаблоны, под действие которых попадает текущий пользователь в соответствии с его участием в группах.

  В этих записях содержится минимальное количество информации, а расширенные наборы параметров представлены в контейнере «cn=kerberos» и связаны с шаблонами ссылками через значение атрибута *krbPwdPolicyReference*.

- Если на пользователя не распространяется действие ни одной политики, ему будет назначена глобальная политика (global_policy) по умолчанию, см. рис. 2.123

- Если пользователь попадает под действие сразу нескольких политик, то параметры политик не суммируются, а выбирается одна из них, приоритет которой будет иметь наименьшее значение, см. табл. 2.8, что аналогично действию параметра Precedence в политиках паролей Windows.

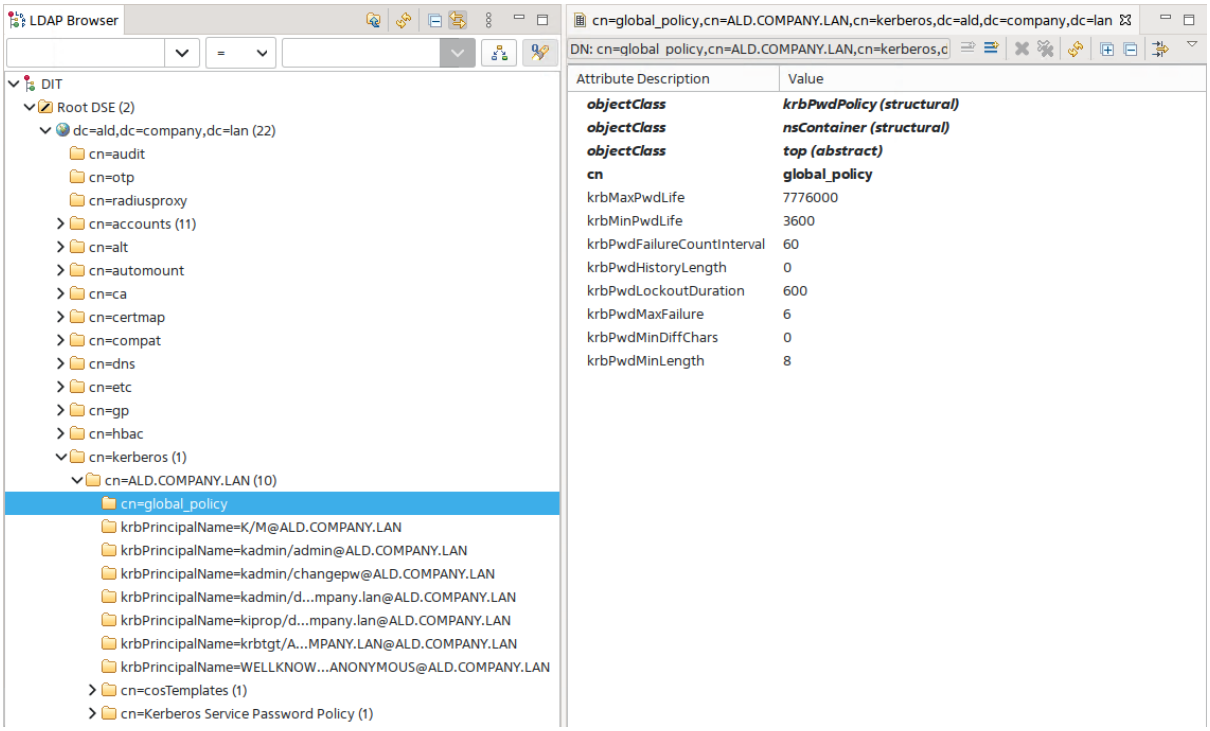


рис. 2.123 – Запись глобальной политики паролей в LDAP-каталоге

табл. 2.8 – Выбор политики в зависимости от приоритета

Параметр	Политика для группы A(приоритет 0)	Политика для группы B(приоритет 1)	Результат (используются параметры для группы A, приоритет 0)
Максимальный срок действия пароля	60 дней	90 дней	60 дней
Минимальная длина пароля	10 символов	0 символов (без ограничений)	10 символов

Проверки паролей ограничены возможностями MIT Kerberos, поэтому они поддерживают тот же самый набор параметров (см. табл. 2.9).

табл. 2.9 – Параметры политик паролей

Параметр глобальной политики	Значение по умолчанию
Максимальный срок действия задает период в количестве секунд, в течение которого система не будет требовать смены пароля	krbMaxPwdLife = 7776000 Пароль активен 90 дней, после чего пользователю будет предложено сменить его.
Минимальный срок действия задает период в секундах, в течение которого система будет запрещать повторную смену пароля	krbMinPwdLife = 3600 После смены пароля пользователь должен подождать 1 час перед повторной сменой.
Размер журнала определяет количество предыдущих паролей, которые нельзя использовать повторно	krbPwdHistoryLength = 0 Запрет на повторное использование паролей не налагается.
Классы символов – это параметр, который определяет требование по сложности пароля, указывая сколько разных классов символов должно быть использовано в пароле: цифры; буквы нижнего регистра; буквы верхнего регистра; символы UTF-8; все остальные символы, не вошедшие ни в одну из предыдущих групп, например, ! « # $ % и т.д. При использовании одного и того же символа более двух раз подряд сложность пароля будет занижена на один балл. Например, если сложность пароля «Secret1pwd» будет оценена в три балла (большие буквы + маленькие буквы + цифры), то для пароля «Secret111pwd» сложность снизится до двух баллов (минус один балл за повторы символа «1»).	krbPwdMinDiffChars = 0 Требований по сложности пароля по умолчанию нет. Интересный факт, что при подсчете повторов последний символ в пароле не учитывается, поэтому если повторяющиеся символы будут идти в конце строки, то нужно, чтобы их было не меньше четырех «Secretpwd1111», иначе штраф налагаться не будет.
Минимальная длина задает минимально допустимое количество символов в пароле.	krbPwdMinLength = 8 Пользователь не может использовать пароль короче 8 символов.
Максимальное количество ошибок определяет, сколько раз пользователь может неправильно ввести пароль, прежде чем его учетная запись будет временно заблокирована. Блокировка выполняется только на текущем контроллере, на другие сервера эта информация не передается.	krbPwdMaxFailure = 6 Пользователь будет временно заблокирован после 7 неверно введенных паролей подряд. Срок блокировки указан в параметре krbPwdLockoutDuration.
Интервал сброса ошибок задает период в секундах, по истечении которого счетчик неудачных попыток входа будет сброшен.	krbPwdFailureCountInterval = 60 Если пользователь выждет 1 минуту после 6 неудачных попыток введения пароля, то счетчик неудачных попыток обнулится.
Длительность блокировки задает период в секундах, в тече-	krbPwdLockoutDuration = 600

Следствием интеграции с MIT Kerberos является возможность использования таких атрибутов, как *krbPrincipalExpiration* и *krbLastSuccessfulAuth*. Например, с помощью команды `ipa user-mod` вы можете установить срок действия учетной записи, после которого пользователю станет недоступна Kerberos-аутентификация:

```
admin@dc-1:~$ ipa user-mod alexander.kuznetsov \
--principal-expiration='20330725115110Z'
```

Где срок действия учетной записи задается в формате временной метки GeneralizedTime:

- **2033** – год;
- **07** – месяц;
- **25** – день месяца;
- **115110** – часы, минуты, секунды;
- **Z** – часовой пояс по нулевому (Zero) меридиану.

Примечание: Стоит заметить, что по соображениям повышения производительности работы домена фиксация даты последнего входа по умолчанию отключена и этот атрибут исключен из репликации.

Однако, если у вас достаточно вычислительных ресурсов и хорошие каналы связи, то вы можете включить эти функции изменением конфигурации сервера. Для этого потребуется посмотреть текущие настройки и установить новое значение параметра *ipaconfigstring*, исключив из него значение «KDC:Disable Last Success»:

```
admin@dc-1:~$ ipa config-show | grep паролей
  Возможности подключаемого модуля паролей: AllowNThash, KDC:Disable Last Success
admin@dc-1:~$ ipa config-mod --ipaconfigstring='AllowNThash'
  Максимальная длина имени пользователя: 32
  Максимальная длина имени хоста: 64
  Основа домашних каталогов: /home
  Оболочка по умолчанию: /bin/bash
  Группа пользователей по умолчанию: ipausers
  Почтовый домен по умолчанию: ald.company.lan
  Ограничение времени поиска: 5
  Ограничение размера поиска: 500
  Поля поиска пользователей: uid,givenname,sn,telephonenumber,ou,title
  Поля поиска групп: cn,description
  Включить режим миграции: FALSE
  Основа субъекта сертификата: O=ALD.COMPANY.LAN
  Уведомление об окончании действия пароля (в днях): 4
  Возможности подключаемого модуля паролей: AllowNThash
  Порядок применения списка пользователей SELinux: guest_u:s0$xguest_u:s0$user_u:s0
↪$staff_u:s0-s0:c0.c1023$sysadm_u:s0-s0:c0.c1023$unconfined_u:s0-s0:c0.c1023
```

(continues on next page)

(продолжение с предыдущей страницы)

```
  Пользователь SELinux по умолчанию: unconfined_u:s0-s0:c0.c1023
  Типы PAC по умолчанию: MS-PAC, nfs:NONE
  Главные IPA-серверы: dc-1.ald.company.lan
  Главный IPA-сервер с поддержкой PKINIT: dc-1.ald.company.lan
  DNS-серверы IPA: dc-1.ald.company.lan
admin@dc-1:~$ sudo ipactl restart
[sudo] пароль для admin:
Restarting Directory Service
Restarting krb5kdc Service
Restarting kadmin Service
Restarting named Service
Restarting httpd Service
Restarting ipa-custodia Service
Restarting smb Service
Restarting winbind Service
Restarting ipa-otpd Service
Restarting ipa-dnskeysyncd Service
ipa: INFO: The ipactl command was successful
admin@dc-1:~$ kinit admin
Password for admin@ALD.COMPANY.LAN:
admin@dc-1:~$ ipa user-show admin --all --raw | grep krbLastSuccessful
  krbLastSuccessfulAuth: 20241009095204Z
```

6.3.3 Настройка политики паролей

Требования глобальной политики довольно лояльны, поэтому для группы администраторов их целесообразно сделать строже. Рассмотрим два способа создания политики: через портал управления и через командную строку.

Откройте страницу *Групповые политики → Политики паролей* и нажмите кнопку *+ Новая политика паролей*. В поле «Наименование группы пользователей» выберите admins, установите наименьший «Приоритет», равный 0, и нажмите кнопку *Сохранить*. Далее вам станет доступна страница управления политикой, где вы можете задать необходимые настройки, как на нижеприведенной иллюстрации.

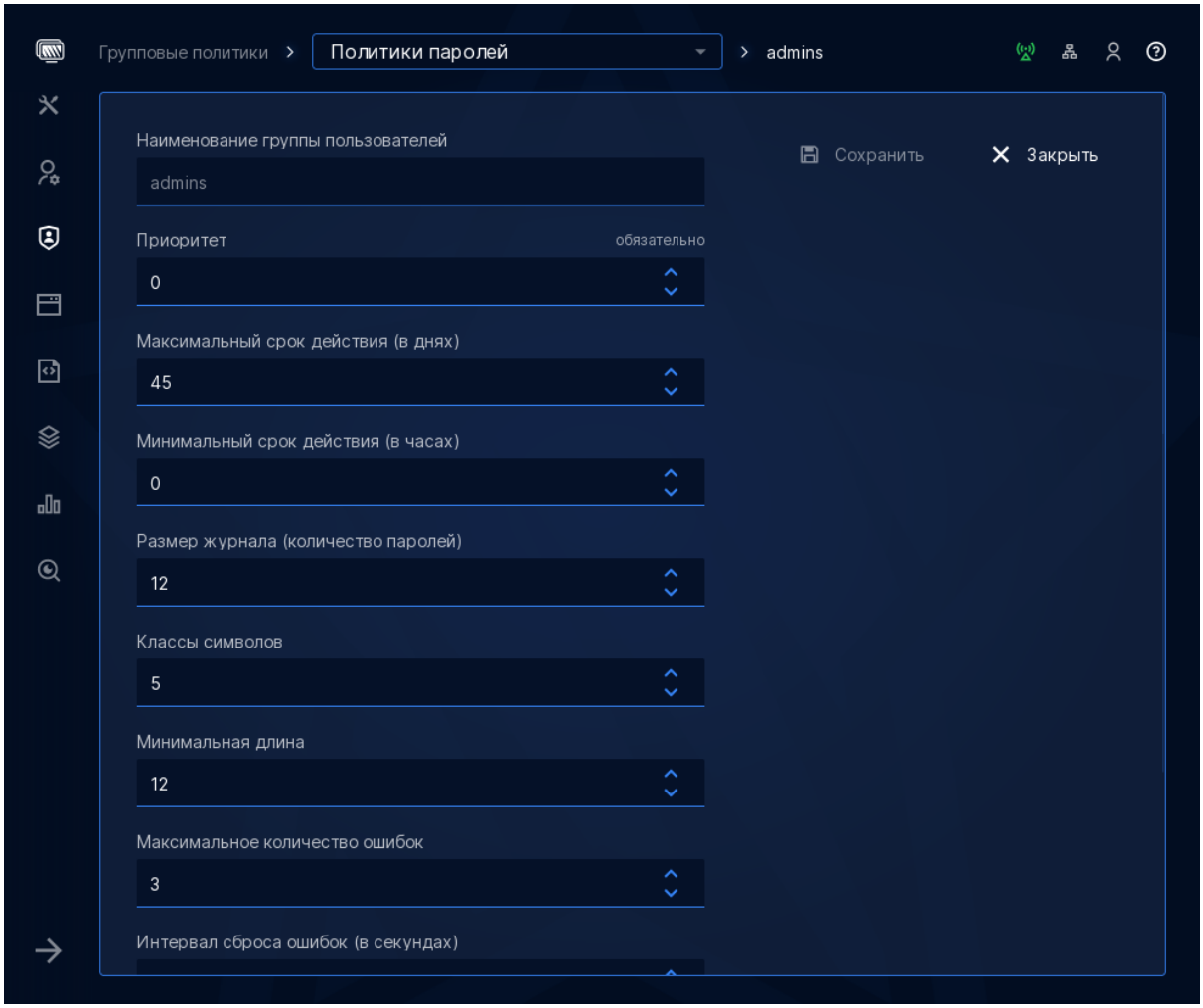


рис. 2.124 – Создание новой политики паролей

Политику паролей можно создать из командной строки с помощью команды `ipa pwpolicy-add`:

```
admin@dc-1:~$ ipa pwpolicy-add admins \
--priority=0 --maxlife=45 --minlife=0 --history=12 \
--minclasses=5 --minlength=12 --maxfail=3 \
--failinterval=120 --lockouttime=1200
  Группа: admins
  Максимальный срок действия (в днях): 45
  Минимальный срок действия (в часах): 0
  Размер журнала : 12
  Классы символов: 5
  Минимальная длина: 12
  Приоритет: 0
  Максимальное количество ошибок: 3
  Интервал сброса ошибок: 120
  Длительность блокировки: 1200
```

Где:

- Ключ `--maxlife=<число>` — максимальный срок действия в днях;

- Ключ `--minlife=<число>` — минимальный срок действия в часах;
- Ключ `--history=<число>` — размер журнала;
- Ключ `--minclasses=<число>` —классы символов;
- Ключ `--minlength=<число>` — минимальная длина;
- Ключ `--priority=<число>` — приоритет политики;
- Ключ `--maxfail=<число>` — максимальное количество ошибок;
- Ключ `--failinterval=<число>` — интервал сброса ошибок в секундах;
- Ключ `--lockouttime=<число>` — длительность блокировки в секундах.

Изменить параметры уже существующей политики можно командой `ipa pwpolicy-mod`. Увеличим интервал сброса ошибок для политики группы admins с 120 до 3600 секунд, но уменьшим длительность блокировки с 1200 до 300 секунд:

```
admin@dc-1:~$ ipa pwpolicy-mod admins --failinterval=3600 --lockouttime=300
  Группа: admins
  Максимальный срок действия (в днях): 45
  Минимальный срок действия (в часах): 0
  Размер журнала : 12
  Классы символов: 5
  Минимальная длина: 12
  Приоритет: 0
  Максимальное количество ошибок: 3
  Интервал сброса ошибок: 3600
  Длительность блокировки: 300
```

Обратите внимание, что изменение максимального срока действия пароля сразу же ни на что не повлияет, т.к. проверка выполняется по значению пользовательского атрибута *krbPasswordExpiration*, которое устанавливается в момент смены пароля. Текущее значение этого атрибута можно уточнить командой `ipa user-show`:

```
admin@dc-1:~$ ipa user-show admin --raw --all | grep krbPasswordExpiration
krbPasswordExpiration: 20240414095400Z
```

Таким образом, чтобы изменения в части срока действия паролей вступили в силу, пользователь должен хотя бы один раз сменить свой пароль, но при необходимости вы можете установить значение *krbPasswordExpiration* вручную с помощью команды `ipa user-mod`. Это удобно также в тех случаях, когда вам нужно сбросить пользователю пароль, но вы не хотите, чтобы система сразу же требовала смены пароля.

```
admin@dc-1:~$ ipa user-mod admin --password-expiration 20240301010101Z
----------------------------
Изменён пользователь "admin"
```

(continues on next page)

(продолжение с предыдущей страницы)

```
---------------------------
Имя учётной записи пользователя: admin
Фамилия: Administrator
Домашний каталог: /home/admin
Оболочка входа: /bin/bash
Псевдоним учётной записи: admin@ALD.COMPANY.LAN, root@ALD.COMPANY.LAN
Окончание действия пароля пользователя: 20240701010101Z
UID: 1902000000
ID группы: 1902000000
Учётная запись отключена: False
Link to department: ou=ald.company.lan,cn=orgunits,cn=accounts,dc=ald,dc=company,
↪dc=lan
Link to head department: ald.company.lan
Пароль: True
Участник групп: trust admins, ald trust admin, admins, lpadmin
Роли: ALDPRO - Main Administrator
Доступные ключи Kerberos: True
```

6.3.4 Группы с автоматическим участием

Область применения HBAC-правил можно задать с помощью групп пользователей и групп хостов, но в некоторых случаях может быть удобнее использовать для этого структурные подразделения. В этом случае вы можете воспользоваться вспомогательными группами и правилами автоучастия (*от англ. automember*).

Привязка объектов к структурным подразделениям в ALD Pro осуществляется с помощью атрибута rbtadp, в котором хранится ссылка на целевое подразделение в формате полного уникального имени записи (*от англ. Distinguished name, DN*). Например, если у вас в корне домена есть структурное подразделение «Московский офис», то значение атрибута **rbtadp** всех дочерних объектов будет «ou=Московский офис,ou=ald.company.lan,cn=orgunits,cn=accounts,dc=ald,dc=company,dc=lan».

Если вам потребуется ограничить выборку только прямыми наследниками, вы можете поставить символ подстановки «^» в начало строки, что позволит вам исключить объекты из подразделений, расположенных ниже по иерархии организационной структуры.

Создадим группу пользователей и правило автоучастия из веб-интерфейса:

1. Создайте группу пользователей. Для этого на странице *Пользователи и компьютеры → Группы пользователей* нажмите кнопку *+ Новая группа*, введите название группы «moscow-office-employees», выберите подразделение «Московский офис» и нажмите кнопку *Сохранить* .

2. Создайте правило автоучастия. Для этого потребуется воспользоваться интерфейсом FreeIPA https://dc-1.ald.company.lan/ipa/ui. Откройте страницу *Идентификация → Автоучастник → Правила группы пользователей*, нажмите кнопку *Добавить*, см. рис. 2.126. В диалоговом

окне добавления правила выберите группу «moscow-office-employees» из списка и нажмите кнопку *Добавить и изменить*.

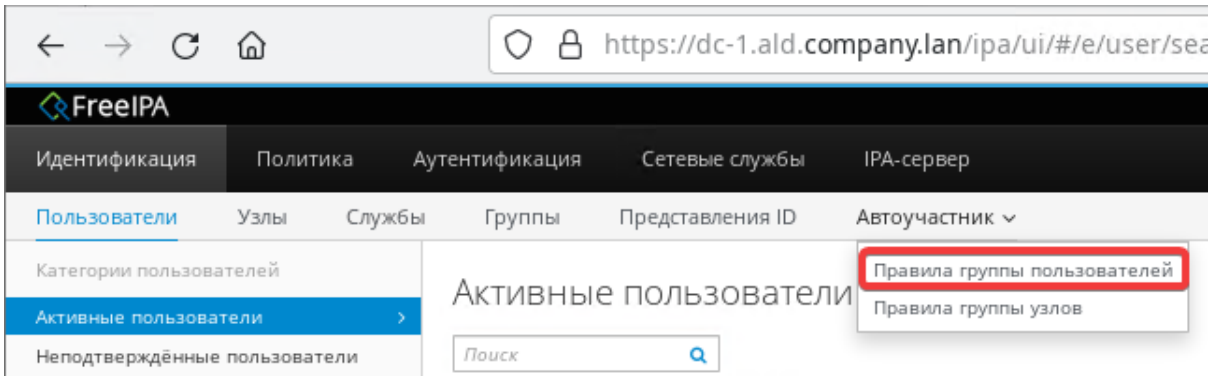


рис. 2.125 – Переход к списку правил автоучастия для групп пользователей

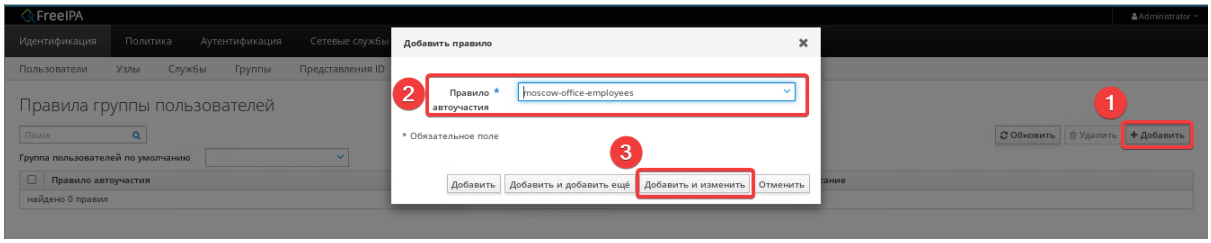


рис. 2.126 – Создание правила автоучастия для группы moscow-office-employees

1. На странице редактирования добавьте критерий отбора по атрибуту **rbtadp** и значению «**ou=Московский офис,ou=ald.company.lan,cn=orgunits,cn=accounts,dc=ald,dc=company,dc=lan**», см. рис. 2.127. Не забудьте нажать кнопку «Сохранить».

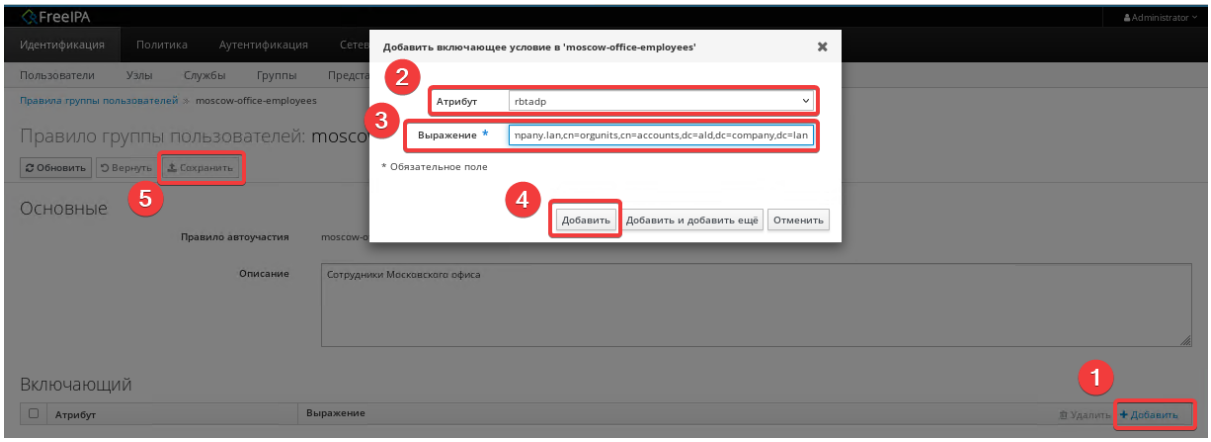


рис. 2.127 – Редактирование правила автоучастия для группы moscow-office-employees

Пересчет правил автоучастия происходит не мгновенно, но вы можете ускорить применение этих правил с помощью команды `ipa automember-rebuild`:

```
admin@dc-1:~$ ipa automember-rebuild --type=group
--------------------------------------------------------------------
Automember rebuild task finished. Processed (5) entries in 0 seconds
--------------------------------------------------------------------
```

```
admin@dc-1:~$ ipa group-show moscow-office-employees | grep "Пользователи"
Пользователи-участники: asidorov, ppetrov, iivanov, vsidorova
```

Если пользователи так и не появятся в списке участников группы, то необходимо проверить корректность фильтра. Обзор записей выполняется по полному вхождению, поэтому никаких лишних пробелов в середине строки быть не должно.

Тоже самое можно сделать из командной строки:

1. Создать группу пользователей moscow-office-employees в подразделении «Московский офис»:

```
admin@dc-1:~$ ipa group-add moscow-office-employees
admin@dc-1:~$ ipa group-mod moscow-office-employees --setattr="rbtadp=ou=Московский␣
↪офис,ou=ald.company.lan,cn=orgunits,cn=accounts,dc=ald,dc=company,dc=lan"
admin@dc-1:~$ ipa group-mod moscow-office-employees --desc='Группа, в которой будут␣
↪все сотрудники московского офиса'
```

2. Создать правило автоучастия и определить критерий автоучастия:

```
admin@dc-1:~$ ipa automember-add moscow-office-employees --type=group
admin@dc-1:~$ ipa automember-mod moscow-office-employees --type=group --desc='Правило␣
↪автоучастия для наполнения группы сотрудников московского офиса'
admin@dc-1:~$ ipa automember-add-condition moscow-office-employees \
--type=group --key=rbtadp --inclusive-
regex='ou=Московский офис,ou=ald.company.lan,cn=orgunits,cn=accounts,dc=ald,
↪dc=company,dc=lan'
```

3. Принудительно обновить состав автоучастников (запускать команду нужно только 1 раз, в дальнейшем правила будут срабатывать автоматически изменении подразделения объекта):

```
admin@dc-1:~$ ipa automember-rebuild --type=group
```

6.4 Политики доступа к узлу (HBAC-правила)

Только представьте себе: новому сотруднику с задатками пентестера выдают ноутбук, блокнот, кружку и доменную учетную запись, а он вместо того, чтобы пойти за кофе, начинает сразу же ломиться на серверы по всем доступным протоколам. И если системный администратор не подумает об этом сценарии заранее, то пользователь прорвется на сервер, а далее останется уповать только на правильные настройки доступа к объектам файловой системы.

Чтобы предотвратить подобного рода атаки, можно ограничить доступ к серверам по определенным протоколам из пользовательского сегмента сети, но лучше полностью запретить обычным

пользователям интерактивный вход на серверы.

6.4.1 Политика локального входа Windows

В домене Active Directory управлять правами локального входа в систему можно с помощью двух параметров групповых политик, которые находятся в разделе *Computer Configuration → Policies → Security Settings* в подразделе *Local Policies → User Rights Assignment* (*Конфигурация пользователя → Политики → Параметры безопасности* и *Локальные политики → Назначение прав пользователя*):

- **Allow log on locally** (Локальный вход в систему) — определяет список пользователей и групп, которым разрешен локальный вход в систему.
- **Deny log on locally** (Запретить локальный вход) — определяет список пользователей и групп, которым запрещен локальный вход в систему. Если пользователь попадает под действие двух параметров сразу, то запрещающий параметр имеет больше приоритет, поэтому доступ будет заблокирован.

На контроллеры домена AD интерактивно могут заходить только пользователи шести административных групп, что определено параметрами объекта групповой политики Default Domain Controllers Policy, см. рис. 2.128.

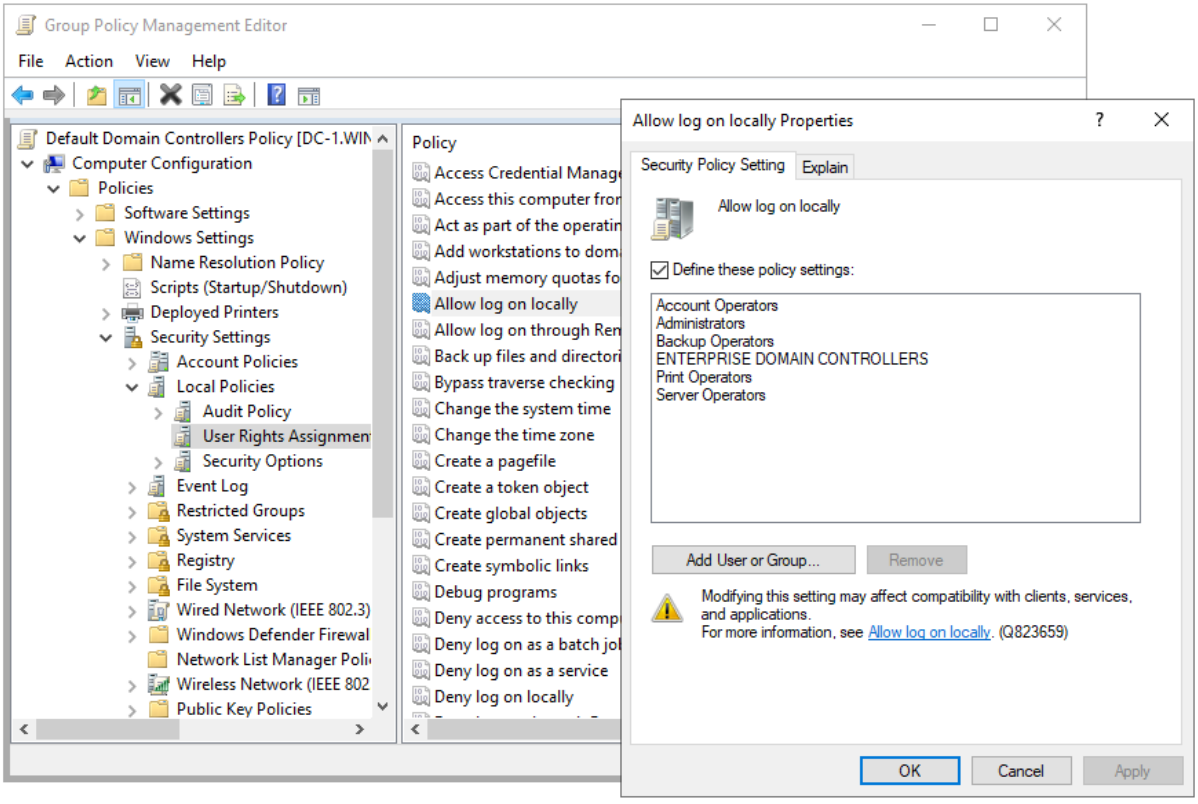


рис. 2.128 – Настройки параметра Allow log on locally в объекте групповой политики Default Domain Controllers Policy

6.4.2 Политики доступа к узлу FreeIPA

Для ограничения доступа к компьютерам в домене FreeIPA есть мощные правила управления доступом к хостам (*от англ. host-based access control, HBAC*), которые значительно превосходят по функциональности политики локального входа Windows. Они создают дополнительный слой авторизации, позволяя разрешить определенным пользователям использовать указанные службы на конкретных хостах.

Что такое «пользователи» и «хосты», обычно вопросов не вызывает, но понятие «служб» требует дополнительных пояснений. В контексте правил HBAC службами являются любые приложения, которые используют PAM-стек, и при этом не важно, являются ли эти приложения просто исполняемыми файлами или работают как демоны в фоновом режиме.

Как мы уже знаем, библиотека подключаемых модулей аутентификации (*от англ. Pluggable Authentication Modules, PAM*) обеспечивает унифицированный программный интерфейс для абстрагирования приложений (таких как `login`, `sshd` или `sudo`) от выполнения стандартных задач аутентификации.

Перечень необходимых модулей аутентификации для каждого приложения задается индивидуально с помощью конфигурационных файлов из директории `/etc/pam.d`, поэтому настройки могут быть изменены в любой момент, что не потребует повторной компиляции приложения. За обработку правил HBAC, например, отвечает модуль `pam_sss.so`, который является одной из клиентских библиотек службы sssd, обеспечивающей работу хоста в домене. Механизм работы правил HBAC отражен на рисунке рис. 2.129.

При подключении пользователя по `ssh` служба sshd обращается к библиотеке PAM, чтобы создать контекст безопасности для выполнения команд от имени этого пользователя. При вызове функции «pam_start» служба передает библиотеке свой идентификатор (PAM service name), который представляет собой простую строку, например, «sshd». Идентификатор службы обычно совпадает с именем исполняемого файла, но это не является обязательным условием, а некоторые приложения могут использовать даже несколько идентификаторов, например, утилита `sudo` может применять дополнительный идентификатор *sudo-i*. Идентификатор службы определяет имя файла из каталога `pam.d`, откуда библиотека PAM будет брать настройки стека модулей (например, для службы sshd настройки будут из файла `/etc/pam.d/sshd`).

Для вступления в силу изменений в настройках HBAC-правил вы можете воспользоваться на целевой машине командами `sss_cache -E` или `sssctl cache-remove`, входящие в пакет `sssd-tools`, чтобы очистить локальный кэш службы sssd. Пакет `sssd-tools` по умолчанию не установлен.

И, наверно, стоит заметить, что в разных дистрибутивах Linux приложения могут использовать разные идентификаторы, например, *ssh* и *sshd*, поэтому список служб нужно будет расширять в соответствии с тем, какие операционные системы с каким программным обеспечением используются

в инфраструктуре вашего предприятия.

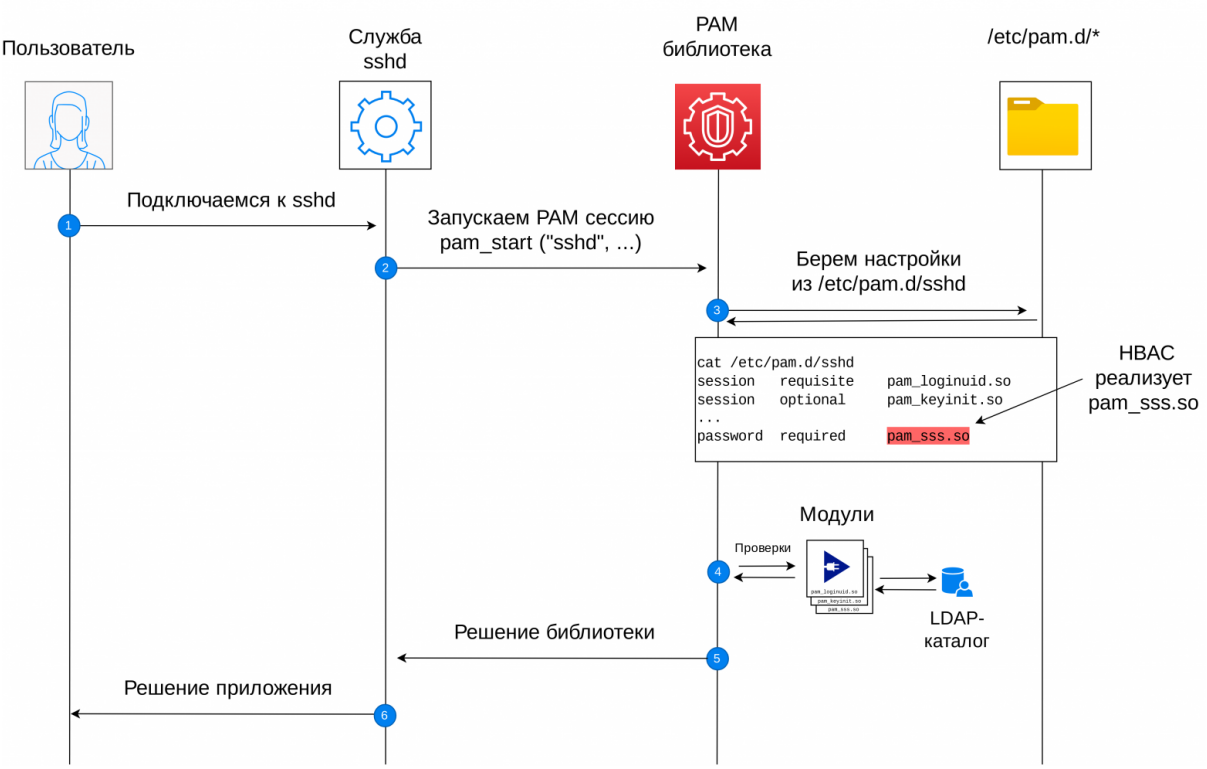


рис. 2.129 – Механизм работы HBAC-правил

Еще один важный момент. В силу особенностей FreeIPA в правилах HBAC не получится использовать следующие группы:

- группу пользователей **ipausers**, т.к. у нее нет POSIX-идентификатора, и поэтому на целевых хостах служба SSSD не отображает участие пользователей в этой группе;
- группу хостов **ipaservers**, т.к. у нее нет класса **mepManagedEntry** и соответствующих зависимостей.

Самый простой способ обойти указанные проблемы – это создать вспомогательные группы posix-ipausers/posix-ipaservers и сделать группы ipausers/ipaservers их участниками. В этом случае можно использовать вспомогательные группы в политиках без ограничений.

6.4.3 Настройка политик доступа к узлу

6.4.3.1 Ограничиваем доступ к серверам

Как мы уже отметили, правила HBAC являются «разрешающими», т. е. «по умолчанию» доступ к службам на доменных компьютерах запрещен, и его нужно открывать с помощью правил. Однако при развертывании домена автоматически создается правило *allow_all*, которое разрешает доступ

«всех ко всему», поэтому для управления авторизацией на уровне HBAC нам нужно сначала ограничить область применения этого правила, например, только группой администраторов.

Внести указанные настройки можно через веб-портал управления на странице *Групповые политики → Политики доступа к узлу → allow_all → Пользователи*, см. рис. 2.130, или через командную строку:

```
admin@dc-1:~$ kinit admin
admin@dc-1:~$ ipa hbacrule-show allow_all
admin@dc-1:~$ ipa hbacrule-mod allow_all --usercat=''
admin@dc-1:~$ ipa hbacrule-add-user allow_all --group admins
admin@dc-1:~$ ipa hbacrule-mod allow_all --desc='Разрешает администраторам доступ к␣
↪любому хосту в домене'
```

Где:

- **hbacrule-mod** — команда, с помощью которой можно модифицировать настройки существующей группы;
  - **allow_all** — имя правила, которые мы хотим модифицировать;
  - **usercat** (*от англ. user category*) — ключ, который позволяет изменить категорию для области применения в части пользователей. Может принимать значение „all“ или пустую строку „“;
- **hbacrule-add-user** — команда, с помощью которой можно расширить область применения правила в части пользователей;
  - **allow_all** — имя правила, которое мы хотим модифицировать;
  - **group** — ключ, который позволяет добавить группу пользователей в область применения HBAC-правила;
  - **admins** — имя группы, которая будет добавлена в область применения правила;
- **hbacrule-show** — команда, с помощью которой можно получить информацию о существующем HBAC-правиле;
  - **allow_all** — имя правила, по которому мы хотим получить информацию;

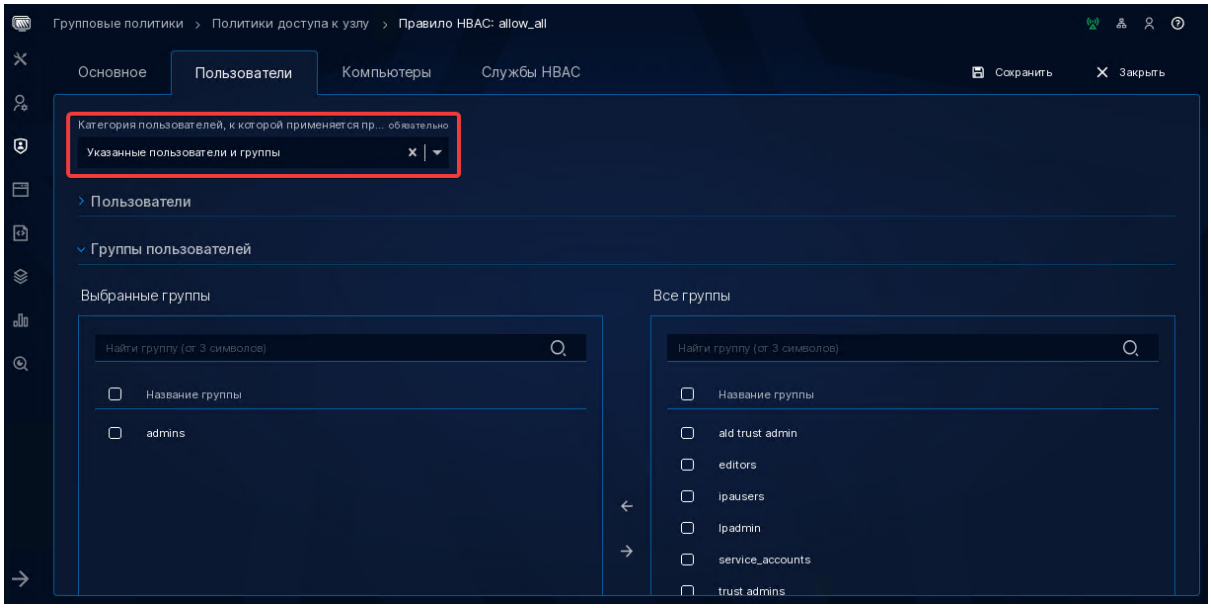


рис. 2.130 – Настройка HBAC-правила allow_all.

Примечание: Если из-за неправильной настройки HBAC-правил доступ к хостам будет все же заблокирован, вы сможете подключиться к порталу управления с любого другого компьютера, который не находится в домене, чтобы исправить ошибку. Доступ к порталу управления не регулируется через механизм HBAC-правил.

Если ограничить область действия правила **allow_all** группой администраторов, то для остальных сотрудников компании нужно будет создать правило **allow_computers**, которое предоставит им право входа на обычные компьютеры в домене.

Сделаем все из командной строки:

```
admin@dc-1:~$ ipa hostgroup-add computers # создание группы компьютеров
admin@dc-1:~$ ipa hostgroup-mod computers --desc='Группа, в которой будут все обычные␣
↪компьютеры домена' # описание
admin@dc-1:~$ ipa hostgroup-add-member computers --hosts pc-1 # включение в группу␣
↪хоста pc-1
admin@dc-1:~$ ipa hbacrule-del allow_computers # удаление созданного ранее HBAC-
↪правила
admin@dc-1:~$ ipa hbacrule-add allow_computers # создание HBAC-правила
admin@dc-1:~$ ipa hbacrule-mod allow_computers --desc='Разрешает всем пользователям␣
↪доступ к обычным компьютерам в домене' # описание
admin@dc-1:~$ ipa hbacrule-mod allow_computers --usercat=all # установить область␣
↪применения на всех пользователей
admin@dc-1:~$ ipa hbacrule-mod allow_computers --servicecat=all # установить область␣
↪применения на все службы
admin@dc-1:~$ ipa hbacrule-add-host allow_computers --hostgroup computers # назначить␣
↪правило на группу хостов
```

Где:

- Команда `ipa hbacrule-add` — создает HBAC-правила;
- Команда `ipa hbacrule-mod` — модифицирует правила, ключ `--desc` позволяет задать описание;
- Команда `ipa hbacrule-add-host` — команда позволяет определить область применения правила;
- Ключ `--hostgroups` — позволяет указать группу хостов.

Убедимся, что пользователь *iivanov* в данный момент имеет право входить на компьютер *pc-1*.

1) Войдем пользователем *iivanov* на компьютер *pc-1* в режиме рабочего стола.
2) После успешного входа сменим сессию на пользователя admin и проверим журнал `/var/log/auth.log`:

```
admin@pc-1:~$ sudo grep -i iivanov /var/log/auth.log | tail
. . .
Feb 2 12:12:55 pc-1 fly-dm[29087]: :0[29087]: pam_unix(fly-dm:auth): authentication␣
↪failure;
logname= uid=0 euid=0 tty=/dev/tty7 ruser= rhost= user=iivanov@ald.company.lan
Feb 2 12:12:55 pc-1 fly-dm[29087]: :0[29087]: pam_sss(fly-dm:auth): authentication␣
↪success;
logname= uid=0 euid=0 tty=/dev/tty7 ruser= rhost= user=iivanov@ald.company.lan
Feb 2 12:12:55 pc-1 fly-dm[29087]: :0[29087]: pam_sss(fly-dm:account): Access denied␣
↪for user
iivanov@ald.company.lan: 6 (Доступ запрещен)
. . .
Feb 2 12:43:12 pc-1 fly-dm[29087]: :0[29087]: pam_unix(fly-dm:auth): authentication␣
↪failure;
logname= uid=0 euid=0 tty=/dev/tty7 ruser= rhost= user=iivanov@ald.company.lan
Feb 2 12:43:12 pc-1 fly-dm[29087]: :0[29087]: pam_sss(fly-dm:auth): authentication␣
↪success;
logname= uid=0 euid=0 tty=/dev/tty7 ruser= rhost= user=iivanov@ald.company.lan
Feb 2 12:43:12 pc-1 fly-dm[29087]: :0[29087]: pam_kiosk2(fly-dm:session): No
iivanov@ald.company.lan profile found, trying common profile
Feb 2 12:43:12 pc-1 fly-dm[29087]: :0[29087]: pam_unix(fly-dm:session): session␣
↪opened for user
iivanov@ald.company.lan by (uid=0)
. . .
```

Из журнала видно, что при попытке входа пользователем **iivanov** до создания правила **computers_allow** эта учётная запись не прошла проверку в модуле pam_sss подсистемы аутентификации pam. Более того, по журналу мы видим, что доступ требовался службе с идентификатором fly-dm.

6.4.3.2 Гранулированный доступ к службам

Для тонкой настройки HBAC-правил необходимо тщательно анализировать типовые сценарии работы пользователей в части используемых служб. Например, для подключения по RDP потребуются fly-wm и xrdp-sesman, см. табл. 2.10:

табл. 2.10 – Службы, необходимые для типовых сценариев работы

Сценарий работы пользователя	К каким идентификаторам служб следует предоставить доступ
Локальный вход в систему	*fly-dm*
Удаленный доступ к менеджеру окон по протоколу RDP	*fly-wm*, *xrdp-sesman*
Удаленное администрирование по протоколу SSH	*sshd*, *sudo*

Чтобы понять, к какой службе требуется предоставить доступ, проще всего выполнить необходимое действие на целевой системе и посмотреть, какие сообщения об ошибках появятся в журнале `/var/log/auth.log`. На приведенном ниже примере видно, что пользователю не хватает прав на запуск `sshd`:

```
root@pc-1:~# tail -f /var/log/auth.log
...
Mar 15 15:25:22 client 4 sshd[30424]: pam_sss(sshd:account): Access denied
for user alexander.kuznetsov: 6 (Permission denied)
...
```

После развертывания домена в системе уже есть список наиболее распространенных служб, но какие-то службы нам все равно потребуется добавить вручную. Давайте создадим «xrdp-sesman». Сделать это можно через веб-интерфейс на странице *Групповые политики → Политики доступа к узлу → Службы HBAC*, см. рис. 2.131, или из командной строки:

```
admin@dc-1:~$ kinit admin
admin@dc-1:~$ ipa hbacsvc-add 'xrdp-sesman' --desc='Доступ по RDP'
```

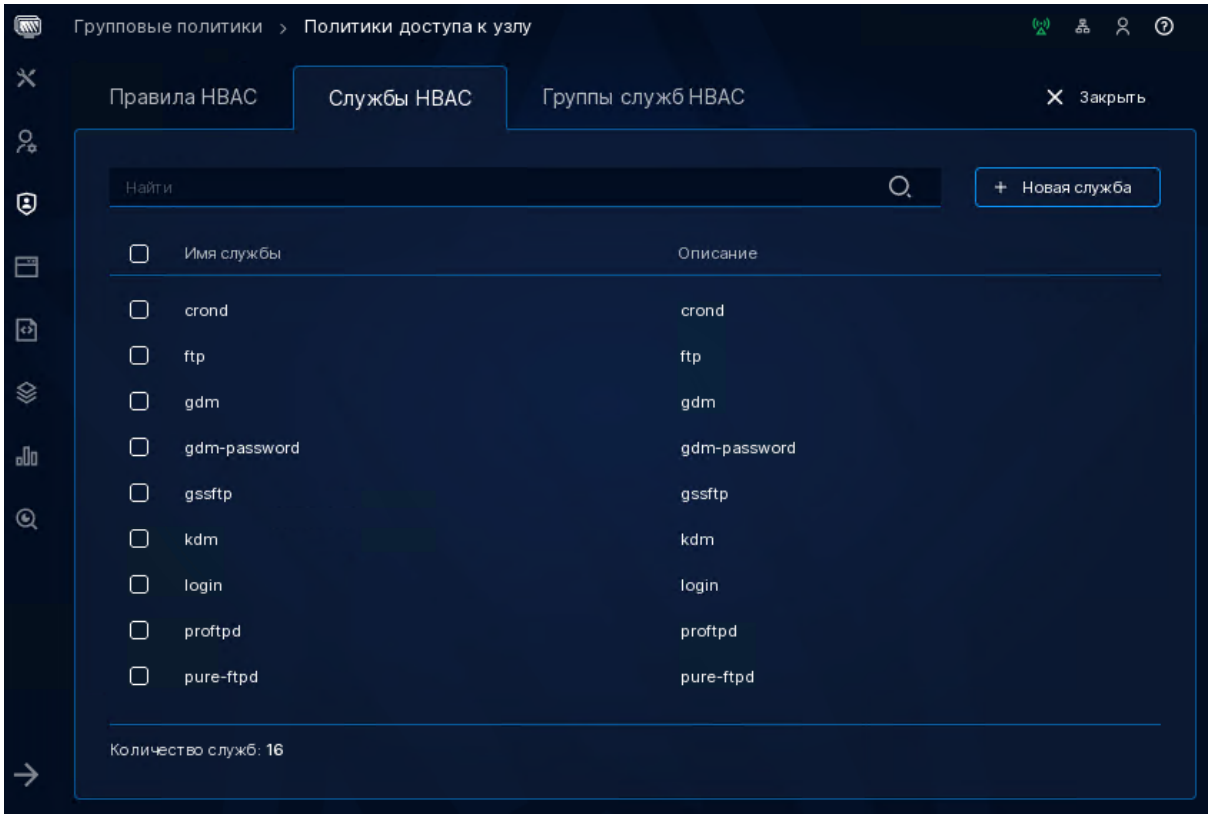


рис. 2.131 – Добавление служб HBAC в интерфейсе портала управления ALD Pro

Допустим, нам нужно предоставить возможность разработчикам из группы dev-users удаленно подключаться к своим компьютерам из группы dev-computers через SSH/RDP и выполнять отдельные команды из-под sudo. Для этого откройте страницу *Групповые политики → Политики доступа к узлу → Правила HBAC* и нажмите кнопку *Новое правило*. Введите имя правила «allow_developers» и нажмите кнопку *Сохранить*. Настройте область применения правила: группа пользователей «dev-users», группа хостов «dev-computers», службы fly-dm, fly-wm, xrdp-sesman, sshd и sudo. Тоже самое можно сделать из командной строки:

```
admin@dc-1:~$ kinit admin
admin@dc-1:~$ ipa hbacrule-add allow_developers --desc='Доступ для разработчиков'
admin@dc-1:~$ ipa hbacrule-add-user allow_developers --groups dev-users
admin@dc-1:~$ ipa hbacrule-add-host allow_developers --hostgroups dev-computers
admin@dc-1:~$ ipa hbacrule-add-service allow_developers --hbacsvcs fly-dm
admin@dc-1:~$ ipa hbacrule-add-service allow_developers --hbacsvcs fly-wm
admin@dc-1:~$ ipa hbacrule-add-service allow_developers --hbacsvcs xrdp-sesman
admin@dc-1:~$ ipa hbacrule-add-service allow_developers --hbacsvcs sshd
admin@dc-1:~$ ipa hbacrule-add-service allow_developers --hbacsvcs sudo
```

6.4.3.3 Проверка HBAC-правил

Когда у вас в домене два-три правила, их довольно легко проверить напрямую, подключаясь к целевым хостам под тестовыми учетными записями. Но в реальной инфраструктуре потребуется

управлять десятками правил, и упростить их отладку поможет команда `ipa hbactest`, которая работает как RSOP в Windows, моделируя применение правил по тому же алгоритму, который будет использовать служба SSSD.

Вы можете выполнить команду на любом контроллере домена и для любого сочетания пользователь-хост-сервис, чтобы получить ответ о том, есть ли в домене правило, которое соответствует этим критериям. Например, давайте проверим, что *iivanov* имеет доступ к службе sudo на хосте pc-1:

```
admin@dc-1:~$ ipa hbactest --user=iivanov --host=pc-1 --service=sudo
-------------------------
Доступ предоставлен: True
-------------------------
Соответствующие правила: allow_computers
Несоответствующие правила: allow_all
Несоответствующие правила: allow_computers_fly
Несоответствующие правила: allow_systemd-user
```

Выполним проверку конкретного правила allow_computers_fly с помощью ключа `--rules`, сможет ли пользователь *ppetrov* выполнить вход на компьютер pc-1, для чего ему нужна служба fly-dm:

```
admin@dc-1:~$ ipa hbactest --user=ppetrov --host=pc-1 --service=fly-dm --rules allow_
↪computers_fly
-------------------------
Доступ предоставлен: True
-------------------------
 Соответствующие правила: allow_computers_fly
```

6.5 Политики повышения привилегий (SUDO-правила)

Хорошо известно, что безопасники плохого не посоветуют, но и хорошего не разрешат, но для установки программного обеспечения и выполнения других задач администрирования сотрудникам все же необходимы привилегии суперпользователя.

Для этих целей можно, конечно, использовать привилегированную учетную запись, но из соображений безопасности рекомендуют все же работать из-под обычной учетной записи и повышать привилегии только при выполнении отдельных команд. Еще более востребован указанный подход, когда часть административных прав нужно делегировать обычным пользователям, например, чтобы разрешить им перезапуск служб.

6.5.1 Запуск от имени другого пользователя в Windows

В операционной системе Windows запустить приложение от имени другого пользователя проще всего, если кликнуть по файлу правой кнопкой мыши, удерживая Shift, и выбрать в контекстном меню *Run as different user*. Того же эффекта можно достичь утилитой runas.exe и командлетом

Start-Process с параметром -Verb RunAs. Но во всех этих случаях мы получаем функциональность, аналогичную возможностям утилиты su, поэтому нам потребуется передать пользователю пароль от привилегированной учетной записи.

Совсем недавно в Microsoft объявили о релизе утилиты sudo для Windows, которая в отличие от runas поддерживает запуск программ из-под администратора с применением механизма UAC (User Account Control) для верификации запроса, т. е. без запроса пароля. Превратится ли она в продвинутый способ выполнения команд с повышенными привилегиями — будем посмотреть.

6.5.2 Политики повышения привилегий FreeIPA

Возможности Linux по повышению привилегий значительно шире. Кроме утилиты su и управляющих стики-битов с незапамятных времен существовала утилита sudo, которая имеет очень богатые настройки и позволяет логировать неудачные попытки аутентификации.

Например, вызовом следующей команды обычный пользователь может установить приложение htop, если ему разрешено запускать утилиту apt через sudo:

```
iivanov@dc-1:~$ sudo apt install htop
```

Правила SUDO можно определить не только в локальном файле /etc/sudoers, но и централизованно через службу каталога, создавая дополнительный слой авторизации, с помощью которого можно разрешить определенным пользователям на конкретных хостах выполнять отдельные команды с повышенными привилегиями.

В то время, как правила HBAC реализованы на уровне PAM-стека, правила SUDO проверяются непосредственно внутри утилиты sudo. Но обратите внимание, что утилита sudo в начале своей работы запускает PAM-контекст, поэтому, для того чтобы пользователь смог воспользоваться своими SUDO-правилами, ему нужно предоставить доступ в том числе к HBAC-службе sudo.

Мы уже рассматривали подробно утилиту sudo, ее правила и лучшие практики их использования в *Модуль 10. Система аутентификации PAM и управление правами SUDO* курса *Часть 1. Astra Linux* . Поэтому не будем подробно останавливаться на этом и сразу перейдем к особенностям работы правил SUDO в домене.

Список источников правил утилита sudo получает через библиотеку службы имен (Name Service Switch, NSS), настройки которой находятся в файле /etc/nsswitch.conf. Как мы знаем, в операционных системах Linux через этот механизм настраиваются источники для получения информации о пользователях, группах, DNS-записях и многом другом. Основные вызовы NSS реализованы в библиотеке *libc*, а та уже, в свою очередь, выполняет обращение к необходимым бэкендам, как показано на рис. рис. 2.132.

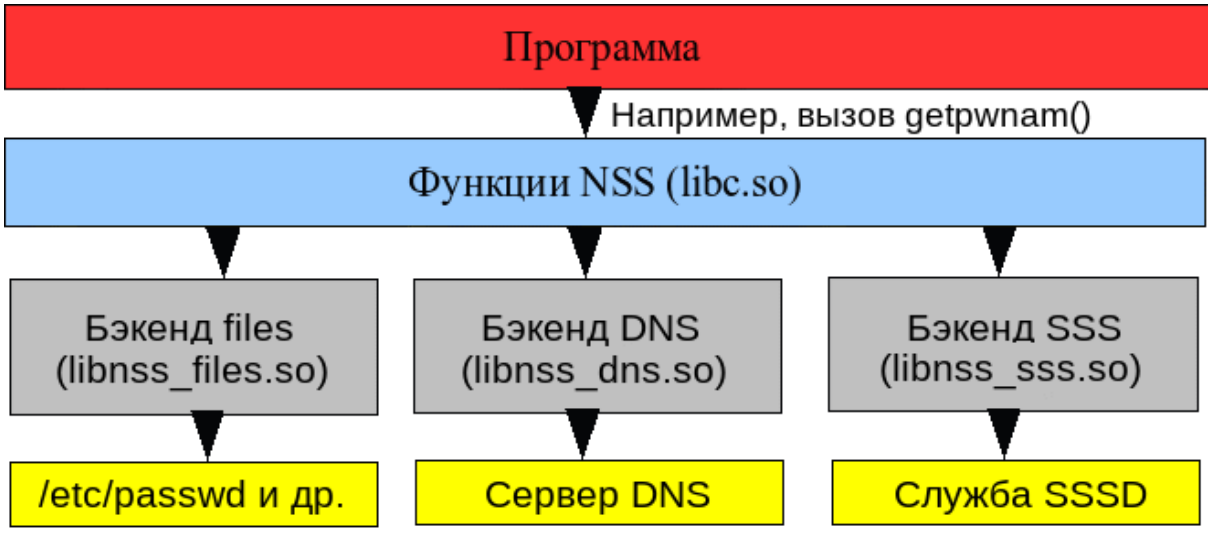


рис. 2.132 – Архитектура диспетчера службы имен (Name Service Switch, NSS)

После установки freeipa-client в файле `/etc/nsswitch.conf` можно найти строку с настройкой базы данных `/etc/sudoers`. По умолчанию правила сначала берутся из локального файла, а затем через модуль sss, который отвечает за взаимодействие с LDAP-каталогом через службу SSSD.

```
admin@dc-1:~$ cat /etc/nsswitch.conf | grep sudoers
sudoers: files sss
```

Поддержка каталогов появилась в sudo с выходом модуля ldap для nss в 2004 году. Источником правил для модуля служили записи из DN «ou=**sudoers**,{basedn}=имя,{basedn}=домена,{basedn}=организации».

Модуль sudo работал с примитивной схемой хранения данных, которая повторяла синтаксис локального файла `sudoers`, игнорируя доступную в каталоге нормализацию данных, например, в части пользователей, групп и хостов. Поэтому при реализации поддержки правил SUDO разработчики FreeIPA создали новую схему, лишенную указанных недостатков. Информация о правилах SUDO во FreeIPA хранится в DN «cn=**sudorules**,cn=sudo,{basedn}=имя,{basedn}=домена,{basedn}=организации», и служба SSSD берет данные напрямую из этой ветки каталога.

Однако для обеспечения совместимости со старым модулем ldap служба каталога FreeIPA с помощью плагина Compat автоматически конвертирует настройки правил в старый формат, как показано на рисунке рис. 2.133. Например, если для правила в cn=sudorules установить ipaEnabledFlag=FALSE, то соответствующая запись в ou=sudoers будет автоматически удалена, но стоит вернуть атрибуту значение TRUE, и запись будет создана повторно.

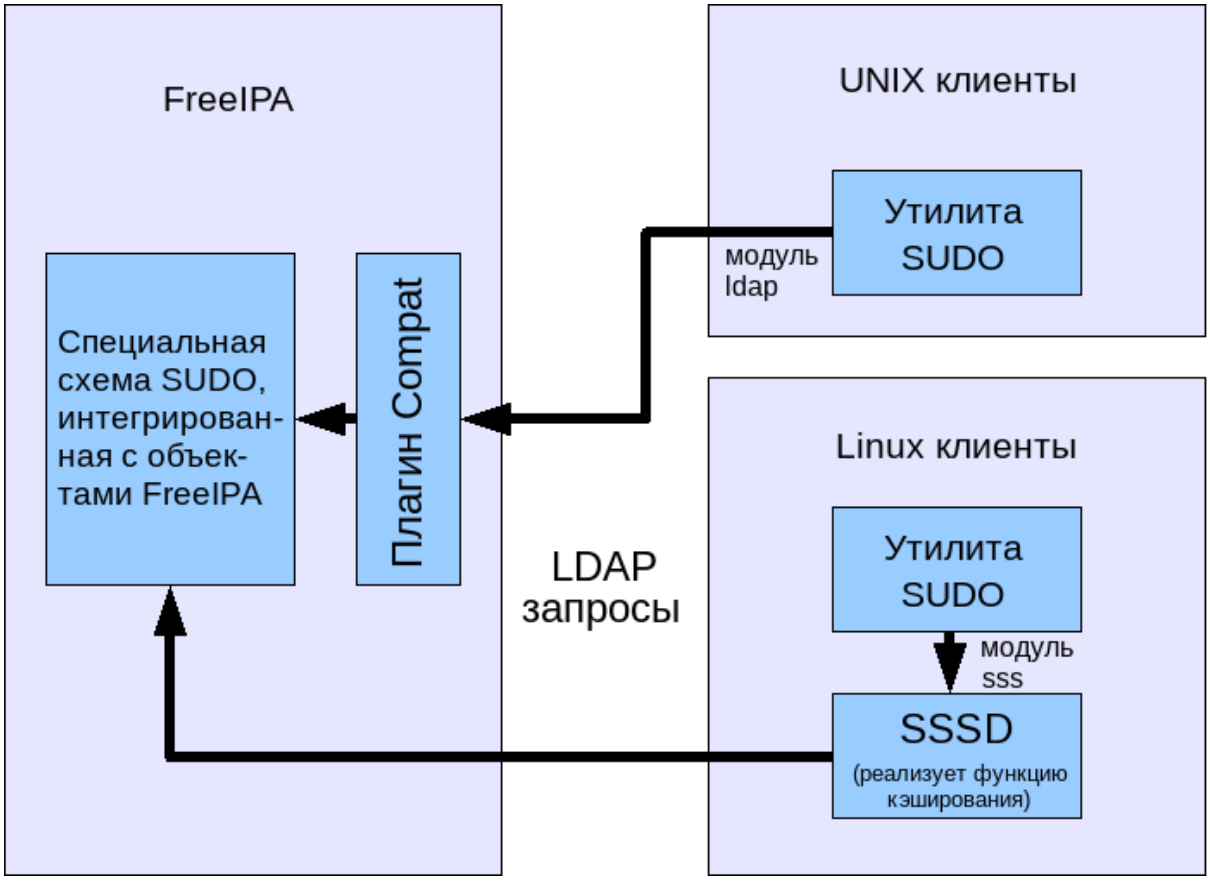


рис. 2.133 – Два источника правил SUDO для разных клиентов

Служба SSSD кэширует правила SUDO, что дает пользователям возможность повышать свои привилегии, даже если они не подключены к домену прямо сейчас. По умолчанию время кэширования составляет 5400 секунд, и для немедленного применения правил на клиентской машине можно выполнить удаление кэша, например, командой `sudo sssctl cache-remove`.

6.5.3 Настройка политики повышения привилегий

6.5.3.1 Создание правил SUDO

Давайте предоставим пользователям право на остановку и перезапуск служб с помощью утилиты systemctl на компьютерах, где они смогут выполнить интерактивный вход в систему.

Учитывая, что пользователи и хосты уже есть в домене, настройку правил следует начать с создания команды. Сделать это можно, например, через портал управления ALD Pro на странице *Групповые политики → Политики повышения привилегий → Команды Sudo*, см. рис. 2.134, или из командной строки:

```
admin@dc-1:~$ kinit admin
admin@dc-1:~$ ipa sudocmd-add '/usr/bin/systemctl' --desc='Запуск systemctl'
```

Где:

- Команда `ipa sudocmd-add` – создает в системе новую команду SUDO.
  - Параметр */usr/bin/systemctl* – полный путь к утилите. Название команды может содержать разрешенные параметры вызова.
  - Ключ `--desc` – позволяет задать описание команды.

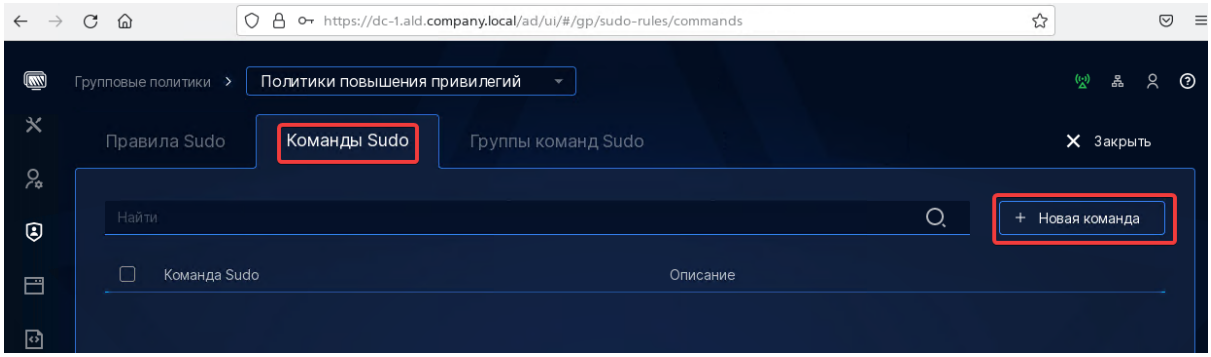


рис. 2.134 – Создание новой команды SUDO

Внимание: В названии команды можно использовать символы «a-z, A-Z, 0-9, -_./~» и пробелы (если только не в начале и конце строки).

Таким образом вы можете определить не только путь к утилите, но и параметры, с которыми вы разрешаете ее запускать. Например, можно разрешить только перезапуск службы sssd, если указать название команды как `/usr/bin/systemctl restart sssd.service`. В тоже время интерфейс ALD Pro версии 2.4.0 не позволяет использовать символ «=» в названии правил SUDO. Если вам вдруг понадобятся такие команды, вы всегда сможете их создать через командную строку или веб-интерфейс FreeIPA.

После того, как команда готова, нужно создать правило SUDO и назначить с помощью этого правила команду на пользователей и компьютеры из веб-интерфейса, см. рис. 2.135, или из командной строки:

```
admin@dc-1:~$ ipa sudorule-add 'systemctl'
admin@dc-1:~$ ipa sudorule-mod 'systemctl' --usercat=all
admin@dc-1:~$ ipa sudorule-mod 'systemctl' --hostcat=all
admin@dc-1:~$ ipa sudorule-add-allow-command 'systemctl' --sudocmds='/usr/bin/
↪systemctl'
```

Где:

- Команда `ipa sudorule-add` — создает правило SUDO;

- Команда `ipa sudorule-mod` — изменяет правило SUDO;
  - Ключ `--usercat` — задает категорию пользователей, которой разрешено использование данного правила.
  - Ключ `--hostcat` — задает категорию хостов, на которых разрешено использование данного правила.
- Команда `ipa sudorule-add-allow-command` — добавляет команду в правило SUDO.

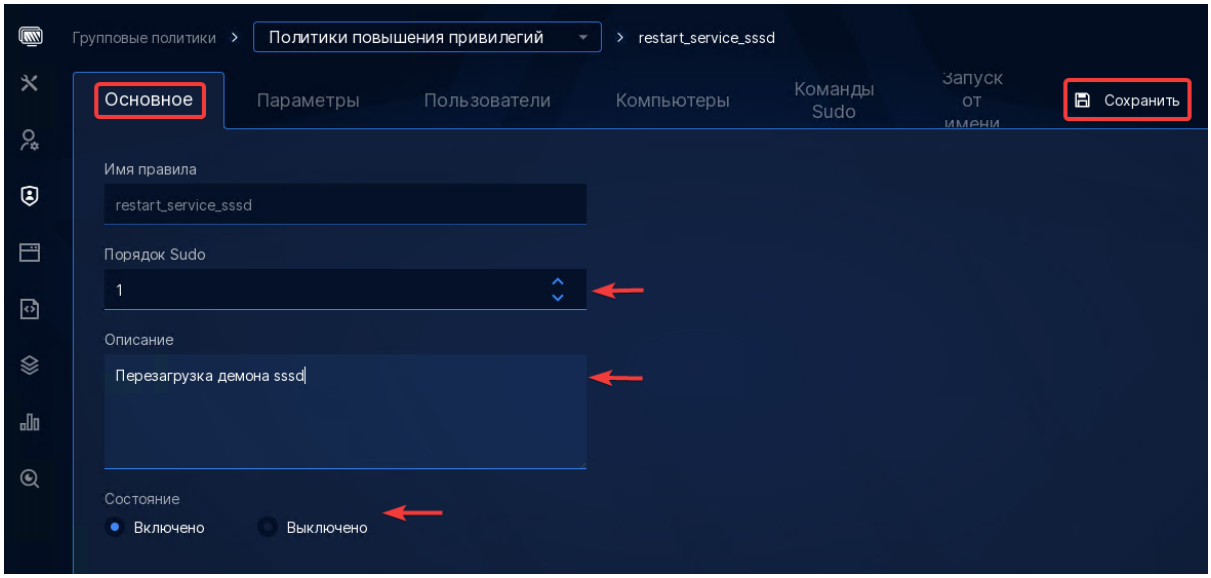


рис. 2.135 – Создание правила SUDO

Пройдем по основным параметрам правила:

- **Раздел Основные**:
  - **Порядок sudo** (необязательный параметр) – целое число, которое определяет очередность выполнения правил. Чем больше значение, тем позже обрабатывается правило, а значит оно может переопределить те правила, которые стоят перед ним.

    Если список команд содержит несколько значений, они обрабатываются в указанном порядке.

    Если у правила одновременно заданы и разрешающие, и запрещающие команды, то сначала обрабатываются разрешающие.
  - **Описание** – необязательный комментарий к правилу.
  - **Состояние** – переключатель определяет, включено правило или нет.

Обязательно сохранить изменения до перехода к следующей вкладке, иначе изменения будут

утеряны.

- **Вкладка Параметры**:

  С помощью параметров можно изменить поведение утилиты для ее тонкой настройки. Ниже приведено несколько наиболее востребованных параметров:

  - **authenticate** – с помощью этого флага можно обязать пользователей вводить пароль при выполнении команды через sudo. Параметр включен по умолчанию, и для отмены требования по вводу пароля его следует отключить, для чего нужно поставить восклицательный знак перед названием параметра «!authenticate».

  - **passwd_tries** – задает количество попыток ввода пароля, прежде чем sudo завершит работу и зарегистрирует ошибку. Задается в виде переменной, по умолчанию `passwd_tries=3`.

  - **timestamp_timeout** – задает количество минут, которое должно пройти перед тем, как sudo повторно запросит пароль. Если установить таймаут равным 0, то утилита будет запрашивать пароль всегда, если установить отрицательное значение, таймаут будет отключен и введенный ранее пароль будет храниться бессрочно. По умолчанию таймаут составляет 15 минут.

  Информацию по остальным параметрам можно найти в `man sudoers`. Значения по умолчанию, с которыми утилита `sudo` была скомпилирована, можно узнать, вызвав команду sudo с ключом `-V` под суперпользователем, например, `sudo sudo -V`.

Внимание: Обратите внимание, что интерфейсе ALD Pro во вкладке «Параметры» не позволяет использовать символ «=». Если вам понадобится установить параметр, например, **passwd_tries=2**, вы всегда сможете его создать через веб-интерфейс FreeIPA.

Пользователи, компьютеры и команды назначаются так же, как для правил HBAC. На вкладке «Запуск от имени» вы можете указать пользователей и группы, от имени которых пользователь сможет запускать указанные команды, используя ключи `-u` и `-g`. Для того чтобы разрешить действовать от суперпользователя, следует оставить эту вкладку незаполненной.

Внимание: Обратите внимание, что в интерфейсе ALD Pro в версиях 2.2.0 и 2.3.0 в настройках правила не получится задать значение «любая команда». Если вы захотите не ограничивать пользователя списком команд, воспользуйтесь командой `ipa sudorule-mod 'Имя_правила' --cmdcat=all` или веб-интерфейсом FreeIPA.

6.5.3.2 Проверка правил SUDO

Результирующий набор правил SUDO, который фактически применяется для конкретного пользователя на конкретном хосту, можно получить вызовом на целевой машине команды sudo с ключами `-l` (строчная L) и `-U`:

```
admin@dc-1:~$ sudo -l -U admin
Matching Defaults entries for admin on dc-1:
    env_reset, mail_badpass,
secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin,

secure_path=/usr/lib/parsec/bin\:/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\
↪:/sbin\:/bin

User admin may run the following commands on dc-1:
    (ALL : ALL) ALL
    (root) ALL
```

В отладке правил вам также поможет журнал утилиты sudo, для включения которого потребуется создать файл `/etc/sudo.conf` (по умолчанию файла не существует) со следующим содержимым:

```
Debug sudo /var/log/sudo_debug.log all@debug
Debug sudoers.so /var/log/sudo_debug.log all@debug
```

В файле *sudo_ debug.log* будет представлена информация о пользователе и среде окружения в момент запуска команды sudo:

```
sudo[22259] settings: debug_flags=all@debug
sudo[22259] settings: run_shell=true
sudo[22259] settings: progname=sudo
sudo[22259] settings: network_addrs=192.0.2.1/255.255.255.0
fe80::250:56ff:feb9:7d6/ffff:ffff:ffff:ffff::
sudo[22259] user_info: user=user_name
sudo[22259] user_info: pid=22259
sudo[22259] user_info: ppid=22172
sudo[22259] user_info: pgid=22259
sudo[22259] user_info: tcpgid=22259
sudo[22259] user_info: sid=22172
sudo[22259] user_info: uid=10000
sudo[22259] user_info: euid=0
sudo[22259] user_info: gid=554801393
sudo[22259] user_info: egid=554801393
sudo[22259] user_info:
groups=498,6004,6005,7001,106501,554800513,554801107,554801108,554801393,554801503,
↪554802131,554802244,554807670
sudo[22259] user_info: cwd=/
sudo[22259] user_info: tty=/dev/pts/1
sudo[22259] user_info: host=client
sudo[22259] user_info: lines=31
sudo[22259] user_info: cols=237
```

С помощью этой информации можно получить ответы на ряд вопросов:

- Какой источник информации использовался для извлечения правил SUDO:

```
sudo[22259] <- sudo_parseln @ ./fileops.c:178 := sudoers: files sss
```

- Со следующей строки включается в работу плагин SSSD:

```
sudo[22259] <- sudo_sss_open @ ./sssd.c:305 := 0
```

- Как много правил было получено от службы SSSD:

```
sudo[22259] Received 3 rule(s)
```

- Подошли эти правила или нет:

```
sudo[22259] sssd/ldap sudoHost 'ALL' ... MATCH!
sudo[22259] <- user_in_group @ ./pwutil.c:1010 := false
```

Вы можете также включить повышенный уровень логирования в настройках `/etc/sssd/sssd.conf` и перезапустить службу.

```
[domain/domain_name]
debug_level = 8
...
[sudo]
debug_level = 8
```

При использовании утилиты sudo будет создан файл журнала `/var/log/sssd/sssd_domain_name.log`, с помощью которого можно будет получить информацию по ряду вопросов:

- Как много правил было получено от службы SSSD:

```
[sdap_sudo_refresh_load_done] (0x0400): Received 4-rules rules
```

- Какие правила служба SSSD загрузила с сервера:

```
[sssd[be[LDAP.PB]]] [sysdb_save_sudorule] (0x0400): Adding sudo rule demo-name
```

- Находились ли подошедшие правила в кэше:

```
[sdap_sudo_refresh_load_done] (0x0400): Sudoers is successfully stored in cache
```

- Какой фильтр был использован для загрузки правил с сервера:

```
[sdap_get_generic_ext_step] (0x0400): calling ldap_search_ext with
[(&(objectClass=sudoRole)(|(!(sudoHost=*))(sudoHost=ALL)(sudoHost=client.example.
↪com)(sudoHost=client)(sudoHost=192.0.2.1)(sudoHost=192.0.2.0/
↪24)(sudoHost=2620:52:0:224e:21a:4aff:fe23:1394)(sudoHost=2620:52:0:224e::/
↪64)(sudoHost=fe80::21a:4aff:fe23:1394)(sudoHost=fe80::/
↪64)(sudoHost=+*)(|(sudoHost=*\\*)(sudoHost=*?*)(sudoHost=*\
↪2A*)(sudoHost=*[*]*))))][dc=example,dc=com]
```

Используйте этот фильтр, чтобы выполнить поиск в базе LDAP-каталога напрямую:

```
admin@dc-1:~$ ldapsearch -x -D "cn=Directory Manager" -W \
 -H ldap://dc-1.ald.company.lan -b dc=ald,dc=company,dc=lan '(objectClass=sudoRole)'
```

Собеседник (Responder) службы SSSD регистрирует свои события в файле журнала `/var/log/sssd/sssd_sudo.log`, с помощью которого можно ответить на следующие вопросы:

- Как много правил было получено от службы SSSD:

```
[sssd[sudo]] [sudosrv_get_sudorules_from_cache] (0x0400): Returning 4-rules rules for
[user@idm.example.com]
```

- Какой фильтр был применен при поиске кэша SSSD:

```
[sudosrv_get_sudorules_query_cache] (0x0200): Searching sysdb with
[(&(objectClass=sudoRule)(|(sudoUser=ALL)(sudoUser=user)(sudoUser=#10001)(sudoUser=
↪%group
-1)(sudoUser=%user)(sudoUser=+\*)))]
```

Для поиска извлечения правил из кэша используйте команду `ldbsearch` из состава пакета ldb-tools:

```
admin@dc-1:~$ ldbsearch -H /var/lib/sss/db/cache_ald.company.lan.ldb -b cn=sysdb '(&
↪(objectClass=sudoRule))'
```

6.6 Групповые политики

6.6.1 Групповые политики Windows

Первые политики появились в Windows еще в 95-й версии, и создать их можно было с помощью редактора Poledit. Данная операционная система по умолчанию не предоставляла безопасного окружения (пользователи могли входить в систему без паролей, менять многие системные настройки и вообще ни в чем себе не отказывать), поэтому для устранения указанной проблемы и был создан этот дополнительный инструмент.

В дальнейшем технологию политик переняли в NT4, и окончательно она была переосмыслена в Windows 2000 для Active Directory. С тех пор технология поддерживается всеми операционными системами и за прошедшую четверть века нисколько не потеряла своей актуальности.

6.6.1.1 Локальные политики

Политики бывают локальными и доменными. Локальные политики появились еще в NT 4.0 и представляют собой просто папки с файлами, внутри которых описаны значения параметров:

- `Windows\System32\GroupPolicy\Machine` — настройки компьютера;

- `Windows\System32\GroupPolicy\User` — настройки пользователя;
- `Windows\System32\GroupPolicyUsers` — настройки для конкретных пользователей и групп пользователей, если используются множественные локальные групповые политики (*англ. Multiple Local Group Policy Objects, MLGPO*). Данный механизм появился с Windows Vista.

Параметры хранятся в файлах Registry.pol (*от англ. policy*), максимальный размер этого файла с Windows 2012 увеличен до 100МБ, но это с большим запасом, т.к. до этого хватало и 64 Кб. Файлы Registry.pol содержат бинарные данные, но так как в них много строк unicode, в них можно «заглянуть» даже простым блокнотом.

Первые 4 байта (PReg) определяют сигнатуру, а следующие за ними 4 байта определяют версию. Версия увеличивается на единицу при каждом изменении GPO, и это значение необходимо для сверки данных, находящихся в LDAP-каталоге, локальном кэше и общей сетевой папке. После заголовка идет тело документа со значениями параметров, которые представляют собой просто разделы реестра Windows, которые необходимо доставить и установить на клиенте, поэтому файл называется реестром (*англ. Registry*). Например, это может быть параметр со ссылкой на файл для установки обоев рабочего стола.

Для построения пользовательского интерфейса редактор групповых политик использует так называемые административные шаблоны, которые представляют собой специальные XML-файлы `*.admx`. Если на компьютере установлены инструменты администрирования групповых политик, то эти файлы будут находиться в папке `C:\Windows\PolicyDefinitions\`, но для удобства централизованного доступа к административным шаблонам метафайлы можно скопировать в каталог `SYSVOL` по адресу: `\win.company.lan\SYSVOL\win.company.lan\Policies\PolicyDefinitions`. В этом случае интерфейс будет открываться чуть дольше, но зато одинаково на всех компьютерах.

Разработчики стороннего программного обеспечения могут создавать свои собственные административные шаблоны групповых политик для использования штатного механизма централизованного конфигурирования через изменение параметров реестра Windows. Например, так поступают разработчики MS Office, Adobe Reader, Google Chrome и др.

За применение групповых политик на серверах и рабочих станциях, начиная с Windows Vista, отвечает отдельная служба «Клиент групповых политик» (*от англ.Group Policy Service Client, GPSVC*), ранее эту задачу возлагали на WinLogon. Если клиентская служба групповой политики будет остановлена, то настройки применяться не будут, поэтому в целях безопасности права доступа по умолчанию настроены таким образом, что даже администратору запрещено ее останавливать, см. рис. 2.136. Текущие права доступа вы можете посмотреть командой `sc.exe sdshow gpsvc`.

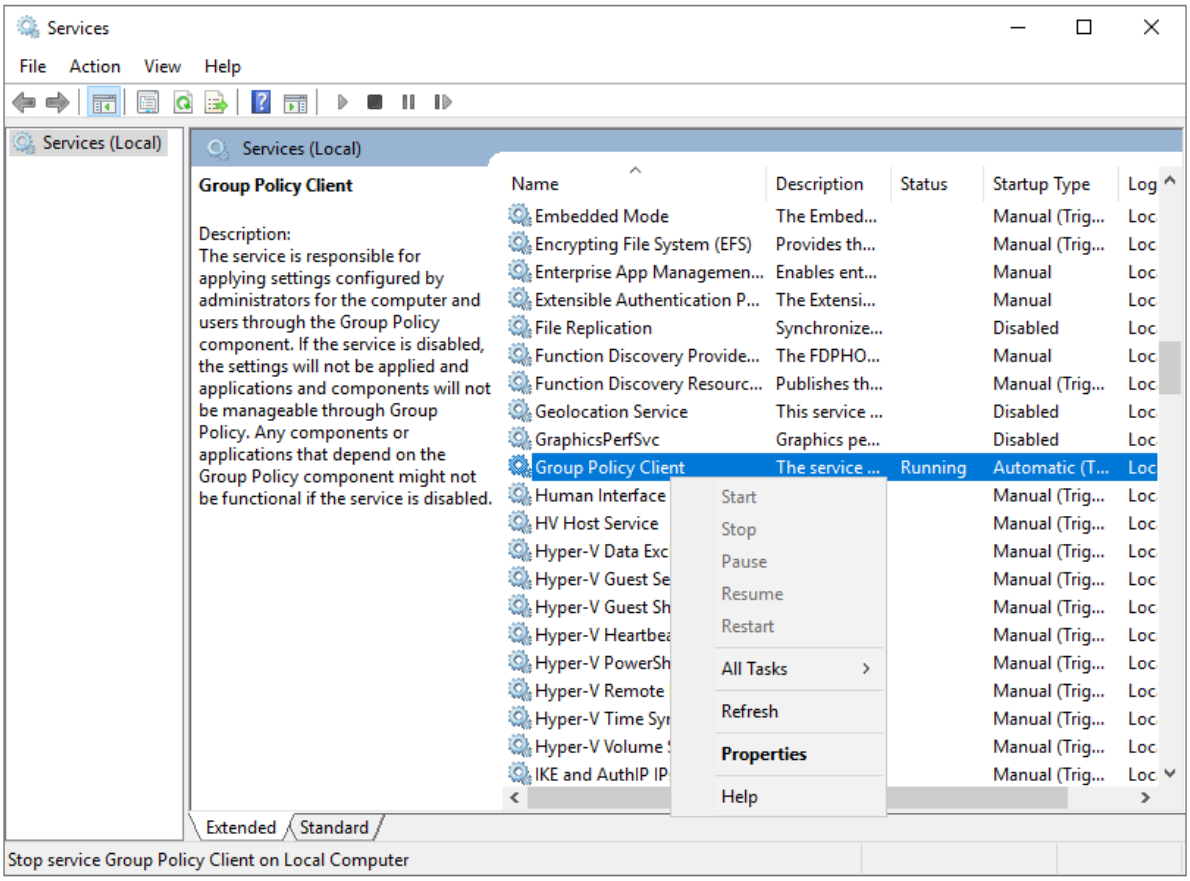


рис. 2.136 – Действия по управлению службой Group Policy Client, доступные администратору

Таким образом, локальная политика включает следующие компоненты: шаблон локальной групповой политики со значением параметров и административные шаблоны для их редактирования. Взаимосвязь компонентов показана на рис. рис. 2.137.

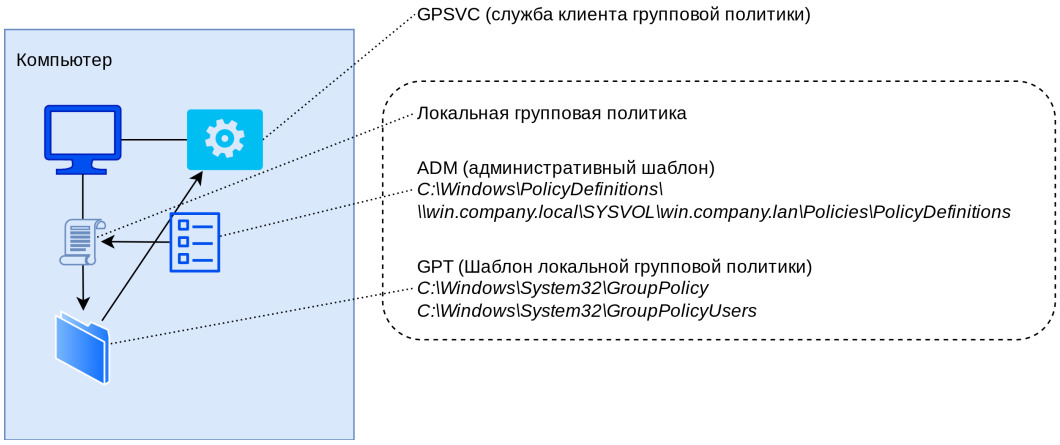


рис. 2.137 – Компоненты локальной политики Windows

6.6.1.2 Доменные групповые политики

Групповые политики появились в Windows 2000 с выходом Active Directory и представляют собой множество правил для настройки рабочего окружения, которые группируются в так называемые **объекты групповой политики** (*от англ. Group Policy Object, GPO*), см. рис. 2.138.

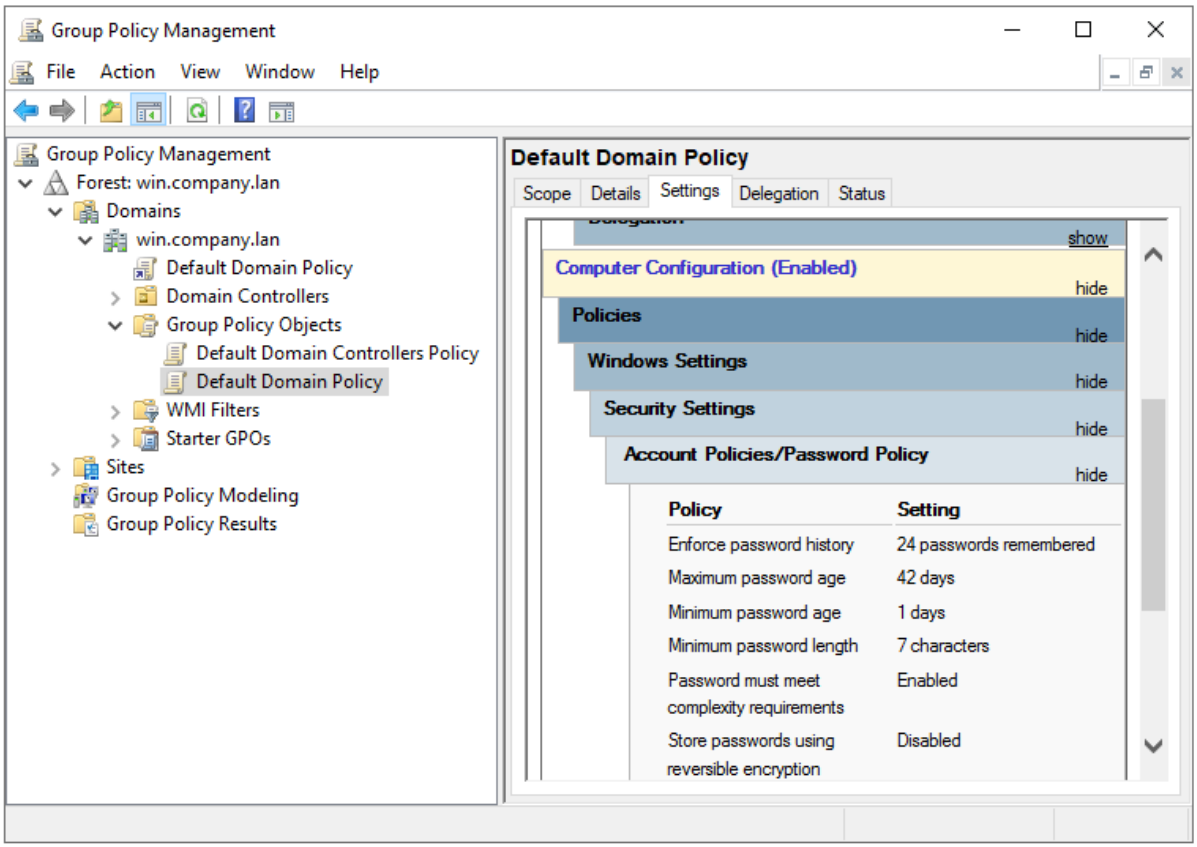


рис. 2.138 – Объект групповой политики Default Domain Policy

Как можно заметить, параметры представлены в двух секциях: настройки компьютера (*англ. Computer Configuration*) и настройки пользователя (*англ. User Configuration*), см. рис. 2.139.

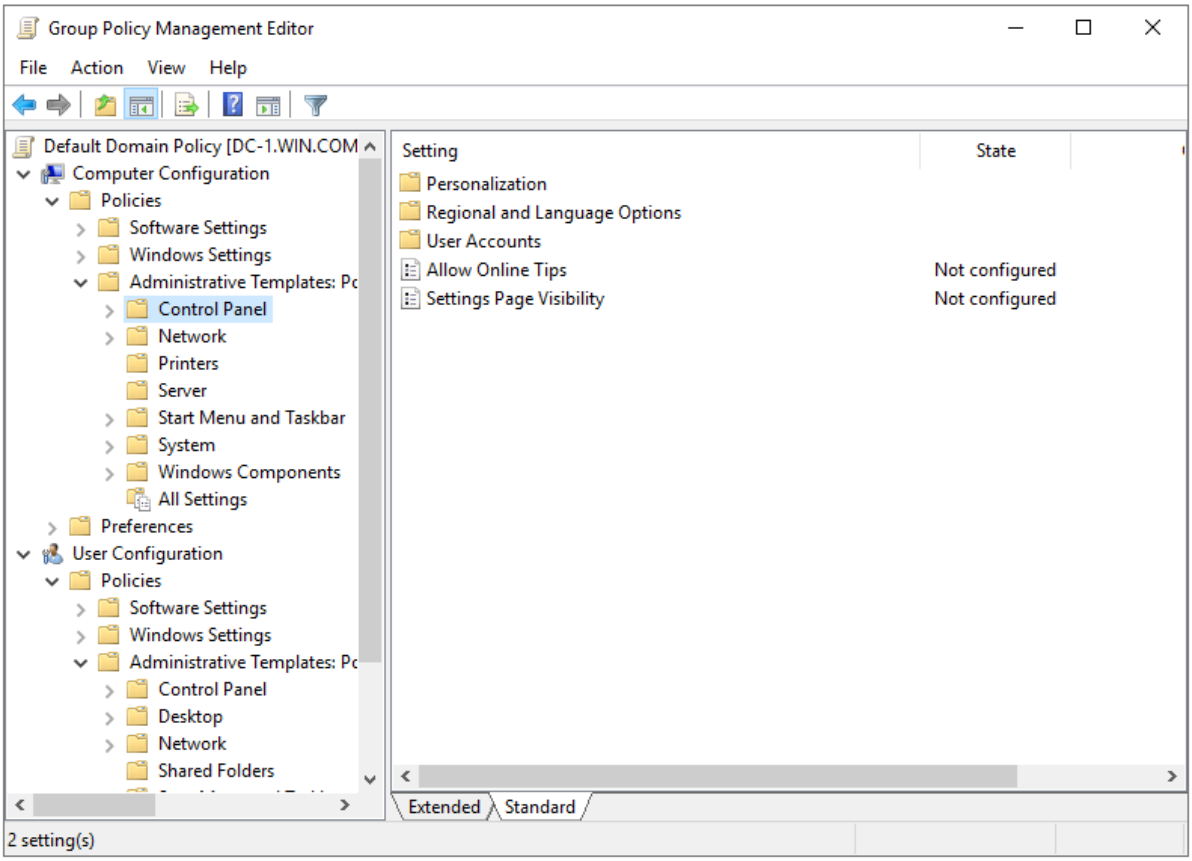


рис. 2.139 – Параметры пользователя и компьютера

Для локальных политик такое деление может показаться условным, поскольку параметры из обеих секций применяются в любом случае, и нужно только не забывать, что параметры компьютера могут переопределить значения аналогичных параметров из секции пользователя.

В случае доменных групповых политик вы должны понимать, что сотрудник может сесть за компьютер из другого структурного подразделения, и тогда пользователь и компьютер будут попадать в область действия разных GPO. Компьютер сначала отберет все GPO, в область действия которых попадает текущий пользователь, и применит параметры из секции пользователя. Затем он отберет все GPO, в область действия которых попадает текущий компьютер, и применит параметры из секции компьютера.

Каждый объект групповой политики включает в себя два компонента:

- **Контейнер групповой политики** (Group Policy Container, GPC) — это запись в LDAP-каталоге, которая определяет имя объекта, версию (порядковый номер изменений), ссылку на шаблон групповой политики gPCFileSysPath на диске и т.д.

  Например, контейнер GPO «Default Domain Policy» в домене win.company.lan можно найти в записи, см. рис. 2.140: `CN={31B2F340-...FB984F9},CN=Policies,CN=System,DC=win,DC=company,DC=local`

- **Шаблон групповой политики** (Group Policy Templates, GPT) — это папка на сетевом диске, в которой хранятся значения параметров по аналогии с папкой GroupPolicy локальных политик. Например, шаблон GPO «Default Domain Policy» в домене win.company.lan можно найти в `\win.company.local\SYSVOL\win.company.local\Policies{31B2F340-...FB984F9}`, см. рис. 2.140.

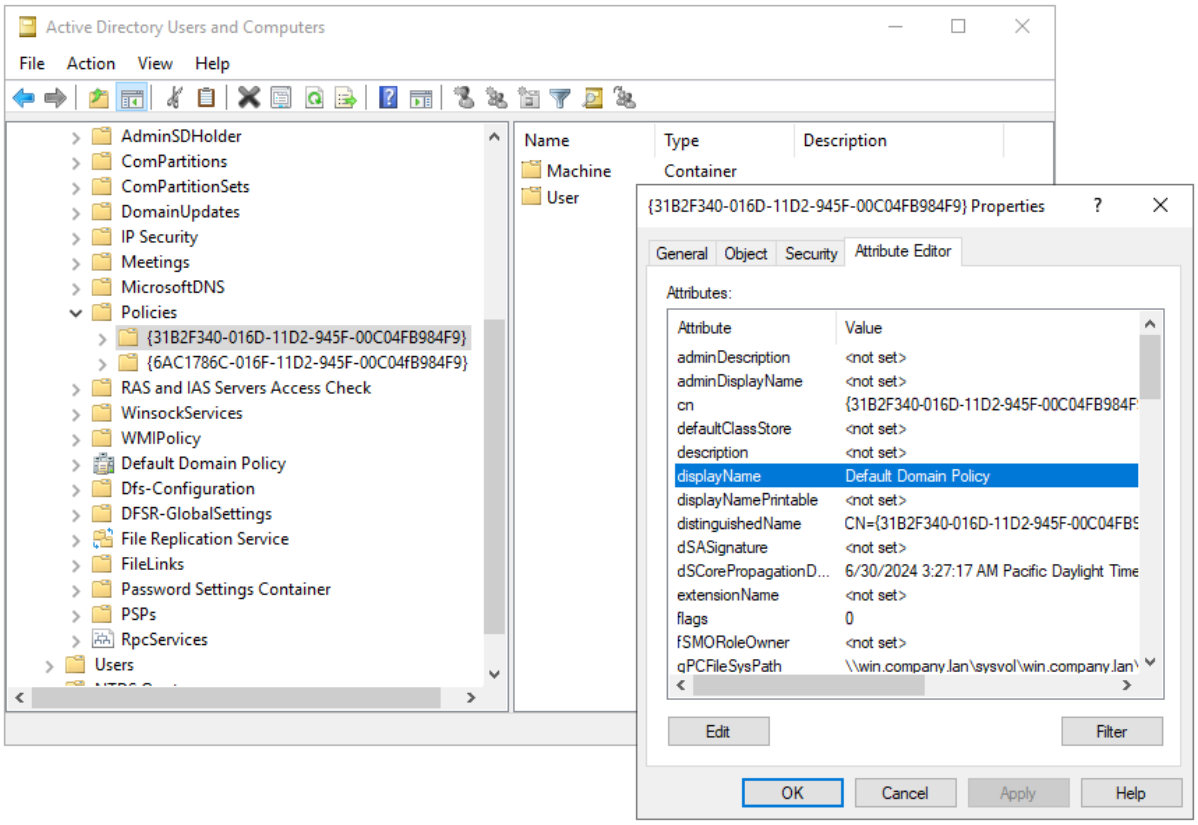


рис. 2.140 – Контейнер групповой политики Default Domain Policy

---

Примечание:

- Default Domain Policy: `{31B2F340-016D-11D2-945F-00C04FB984F9}`
- Default Domain Controller Policy: `{6AC1786C-016F-11D2-945F-00C04FB984F9}`

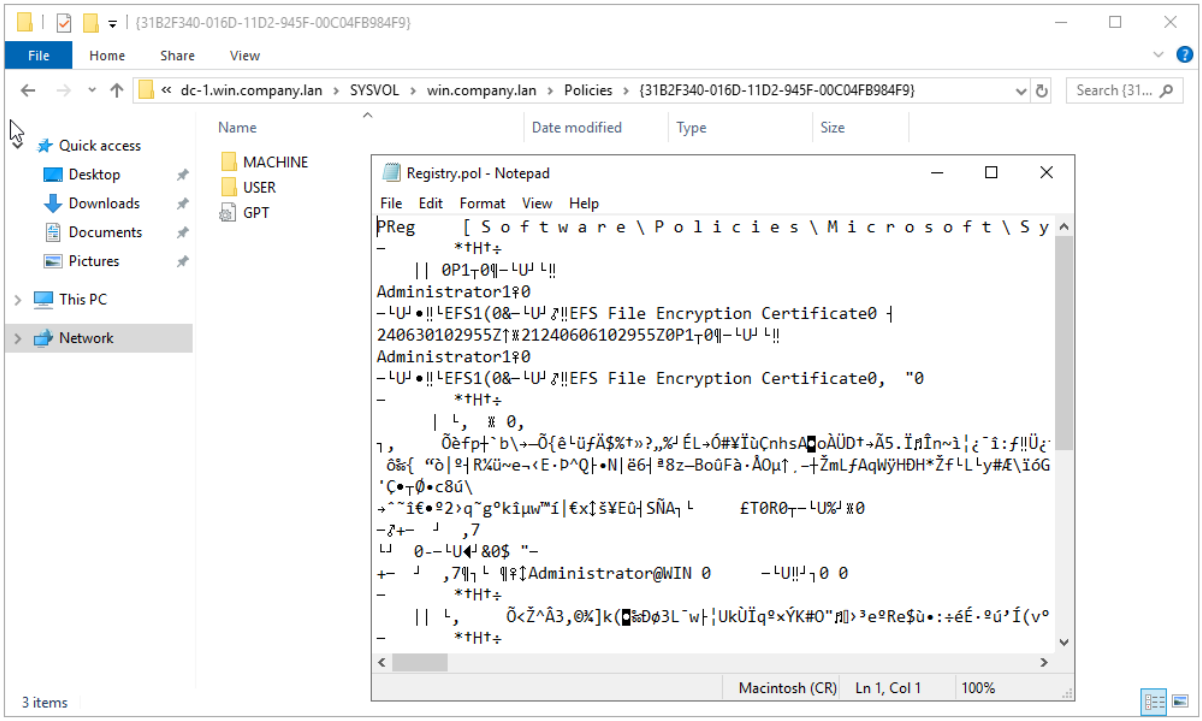


рис. 2.141 – Шаблон групповой политики Default Domain Policy

При назначении объекта групповой политики на структурное подразделение в записи LDAP-каталога, соответствующей этому подразделению, создается атрибут gPLink, который является ссылкой на контейнер GPO, см. рис. 2.142.

Например, в свойствах домена win.company.lan есть gPLink со значением: [LDAP://CN={31B2F340-..B984F9},CN=Policies,CN=System,DC=win,DC=company,DC=lan;0]

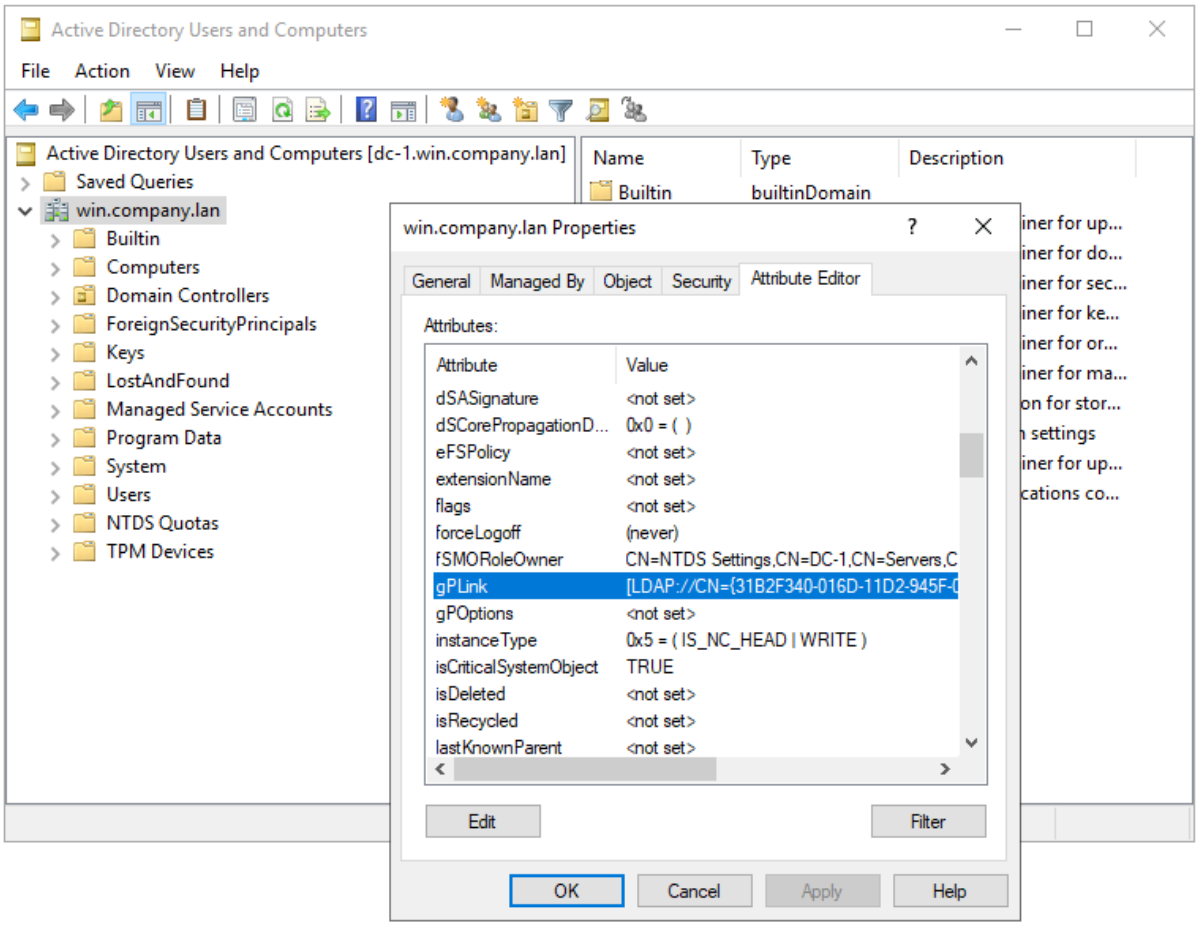


рис. 2.142 – Ссылка на контейнер объекта групповой политики Default Domain Policy

Таким образом, доменная групповая политика включает следующие компоненты: контейнер групповой политики, шаблон групповой политики, и административные шаблоны для редактирования параметров. Их взаимосвязь показана на рис. 2.143.

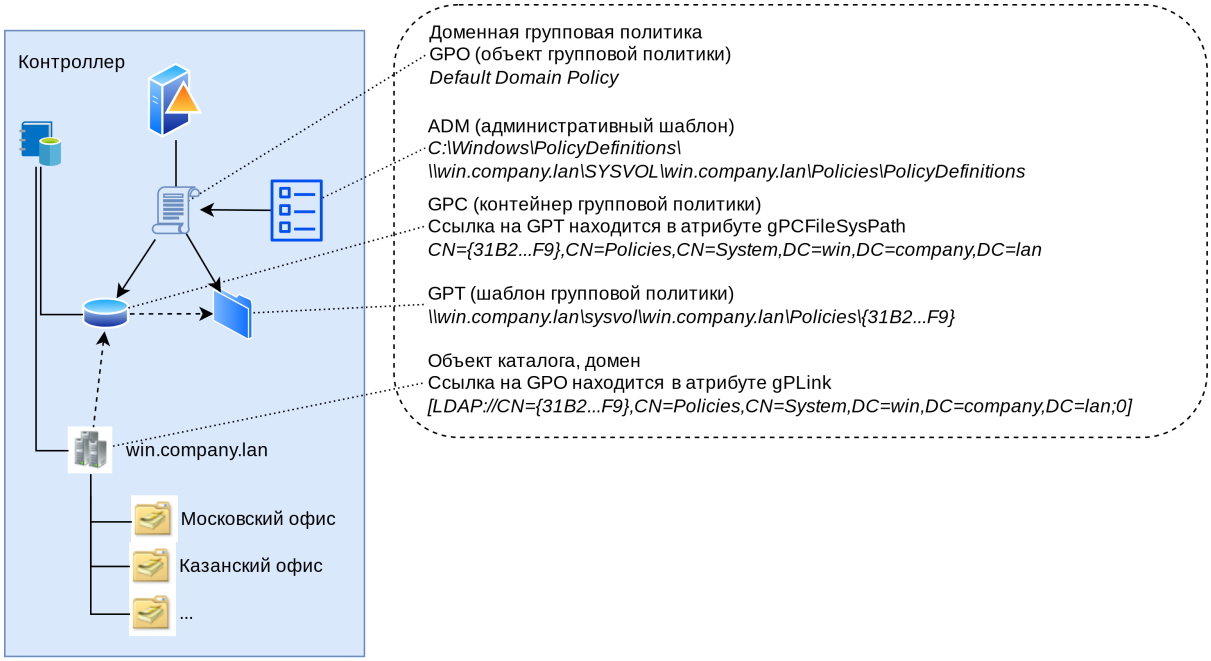


рис. 2.143 – Компоненты доменной групповой политики Active Directory

Область действия групповых политик, назначенных на домен, сайт или структурное подразделение можно сузить с помощью WMI-фильтров (Windows Management Instrumentation), которые позволяют задать дополнительные условия отбора целевых объектов на SQL-подобном языке WQL (WMI Query Language). Обычно эта технология используется в ситуациях, когда пользователи и компьютеры находятся в общем списке, а не в выделенном подразделении. Этот механизм крайне удобен, если нужно применить политику в зависимости от версии ОС, ее сетевых настроек, наличия определенного установленного ПО и других условий.

Кроме WMI-фильтров область действия политик можно сужать еще и через права доступа. Загрузка параметров, назначенных на компьютер, осуществляется из-под учетной записи компьютера, а загрузка параметров, назначенных на пользователя, соответственно, из-под учетной записи пользователя. Поэтому, если у компьютера/пользователя не будет соответствующих прав на чтение GPO, то служба GPSVC не сможет получить параметры объекта, и они не будут применены в системе. Права доступа назначаются через делегирование с помощью списков доступа (*англ. ACL, Access Control List*).

Когда мы изменяем права доступа на вкладке делегирования или через фильтры безопасности (*англ. security filter*), консоль управления выставляет соответствующие ACL в LDAP-каталоге и на сетевом диске. Набор привилегий в каталоге и на диске разный, но это не мешает достигнуть необходимого результата. Администратор может делегировать не только права на чтение, но и права на редактирование GPO, что дает гибкость в вопросах администрирования.

В классической клиент-серверной архитектуре доставка изменений до клиента может быть организована тремя способами:

- **Метод опроса** (*англ. pull*) — это когда клиент периодически опрашивает сервер о наличии каких-либо изменений. Эта модель основана на классическом подходе запрос/ответ (*англ. request/response*).

- **Метод проталкивания** (*англ. push*) — это когда клиент и сервер меняются местами, и уже сервер связывается с клиентом, чтобы передать ему какие-либо изменения.

  В этом случае сервер должен знать IP-адрес клиента и иметь к нему доступ по сети, что невозможно при размещении клиентов за NAT-шлюзом.

- **Метод длинных опросов** (*англ. long poll*) — это когда клиент подключается к серверу, после чего они продолжают удерживать соединение, чтобы у сервера была возможность оперативно передавать клиенту новые сообщения.

  Эта модель основана на веб-сокетах (*англ. WebSocket*), и ее используют, например, мессенджеры, такие как Telegram.

Учитывая, что в домене могут быть сотни серверов, а рабочие станции могут длительное время находиться в выключенном состоянии или вообще эксплуатироваться вне офиса, метод опроса видится наиболее экономичным способом доставки изменений до клиента. Поэтому в MS AD ответственность за извлечение информации о групповых политиках из домена возлагается на клиента, а точнее на его службу GPSVC, которая выполняет процедуру опроса с периодичностью один раз в 90 минут.

Однако во избежание большой нагрузки на контроллеры домена в начале рабочего дня, когда все сотрудники приходят в офис и включают свои рабочие станции, первое применение параметров групповой политики сразу после загрузки компьютера выполняется из кэша, а далее служба выбирает себе случайным образом момент времени для обновления параметров.

Информацию об участии пользователя/компьютера в организационной структуре, релевантные объекты групповой политики и их версии компьютер извлекает по LDAP-протоколу. Шаблоны групповой политики, в которых содержатся значения параметров, компьютер извлекает по SMB. Для отладки групповых политик администратор может отправить команду на принудительное обновление параметров групповой политики конкретному компьютеру через консоль GPMC или командлет Invoke-GPUpdate.

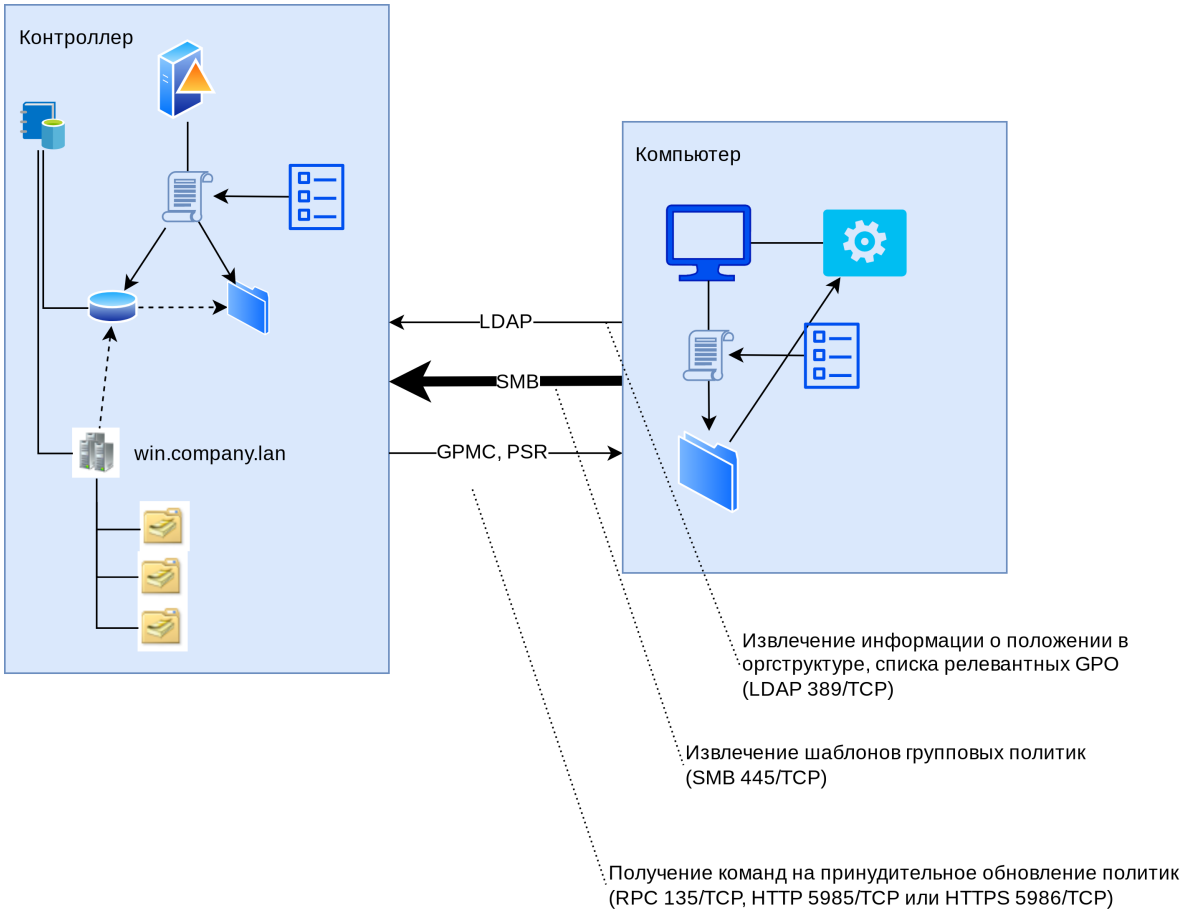


рис. 2.144 – Протоколы, используемые технологией групповых политик для доставки информации до клиента

6.6.1.3 Механизм репликации

Домен AD представляет собой распределенную систему, работу которой могут обеспечивать десятки или даже сотни контроллеров домена. Для согласования изменений между серверами применяются две технологии репликации:

- **Репликация LDAP-каталога** – используется для согласования изменений в контейнерах групповых политик. В этом случае применяются протоколы RPC (135/TCP) и SMTP (25/TCP).
- **Репликация папки SYSVOL** – используется для согласования изменений в шаблонах групповых политик. В этом случае применяется протокол DFS-R (5722/TCP).

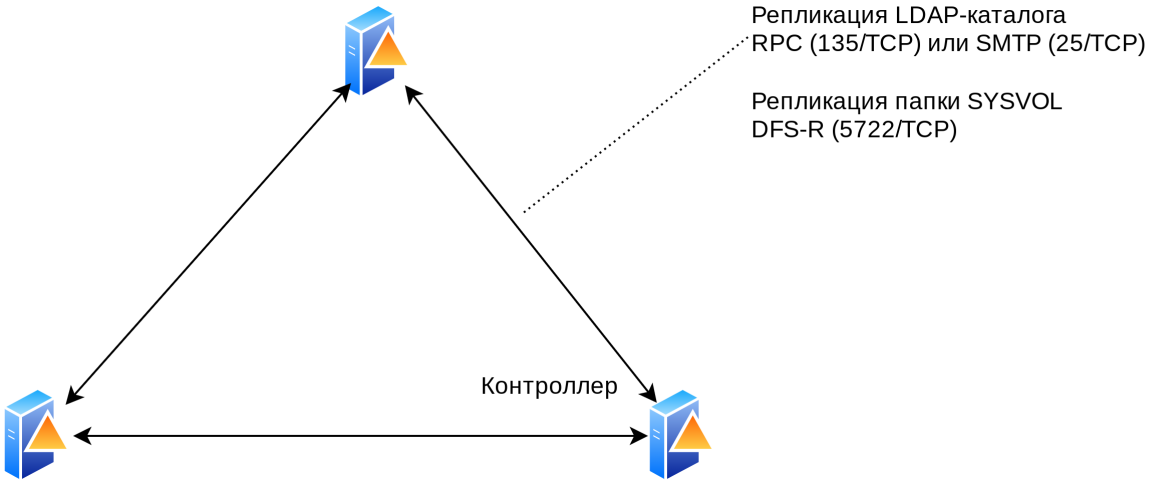


рис. 2.145 – Протоколы, используемые для репликации групповых политик между контроллерами

Вопросы репликации между контроллерами важны для сопровождения ИТ-инфраструктуры, но в контексте групповых политик полезнее рассмотреть более подробно способы доставки групповых политик до клиента.

6.6.1.4 Механизмы наследования и суммирования параметров

Порядок суммирования параметров групповых политик иногда обозначают аббревиатурой LSDOU, так как сначала применяются локальные политики (L), затем политики сайта (S), домена (D) и в завершение политики организационных подразделений (OU). Таким образом, параметры, назначенные на организационные подразделения, всегда превалируют над параметрами локальных политик.

При назначении GPO на структурное подразделение в область его действия попадают пользователи и компьютеры всех нижестоящих подразделений. На схеме рис. 2.146 приведен пример, в соответствии с которым целевой компьютер попадает в область действия всех GPO, начиная с ГПО-1 и заканчивая ГПО-8.

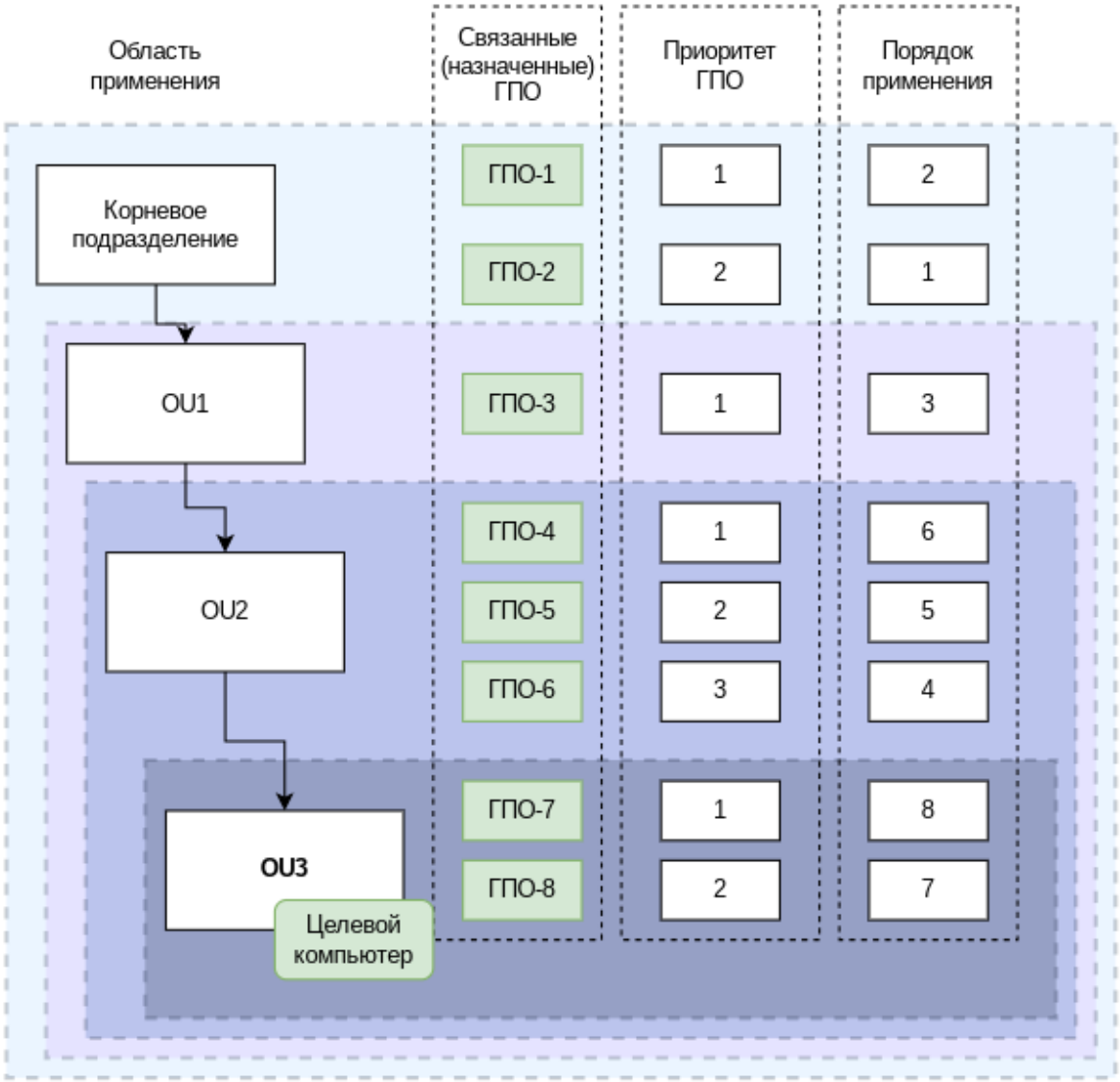


рис. 2.146 – Порядок наследования суммирования групповых политик

При этом, чем ближе GPO по иерархии к пользователю/компьютеру, тем позже будут применяться параметры этого объекта, поэтому они смогут переопределить ранее установленные значения. На приведенном ранее примере параметр ГПО-8 сможет переопределить значения, установленные параметром из ГПО-1.

Однако стандартный порядок наследования можно изменить, если поставить в настройках структурного подразделения флаг «Блокировать наследование» (*англ. Block Inheritance*). Это позволит отменить наследование GPO, назначенных на вышестоящие родительские подразделения. Например, на рисунке рис. 2.147 показан пример, в соответствии с которым целевой компьютер не попадает в область действия ГПО-1 и ГПО-2, т. к. на OU1 установлен запрет наследования.

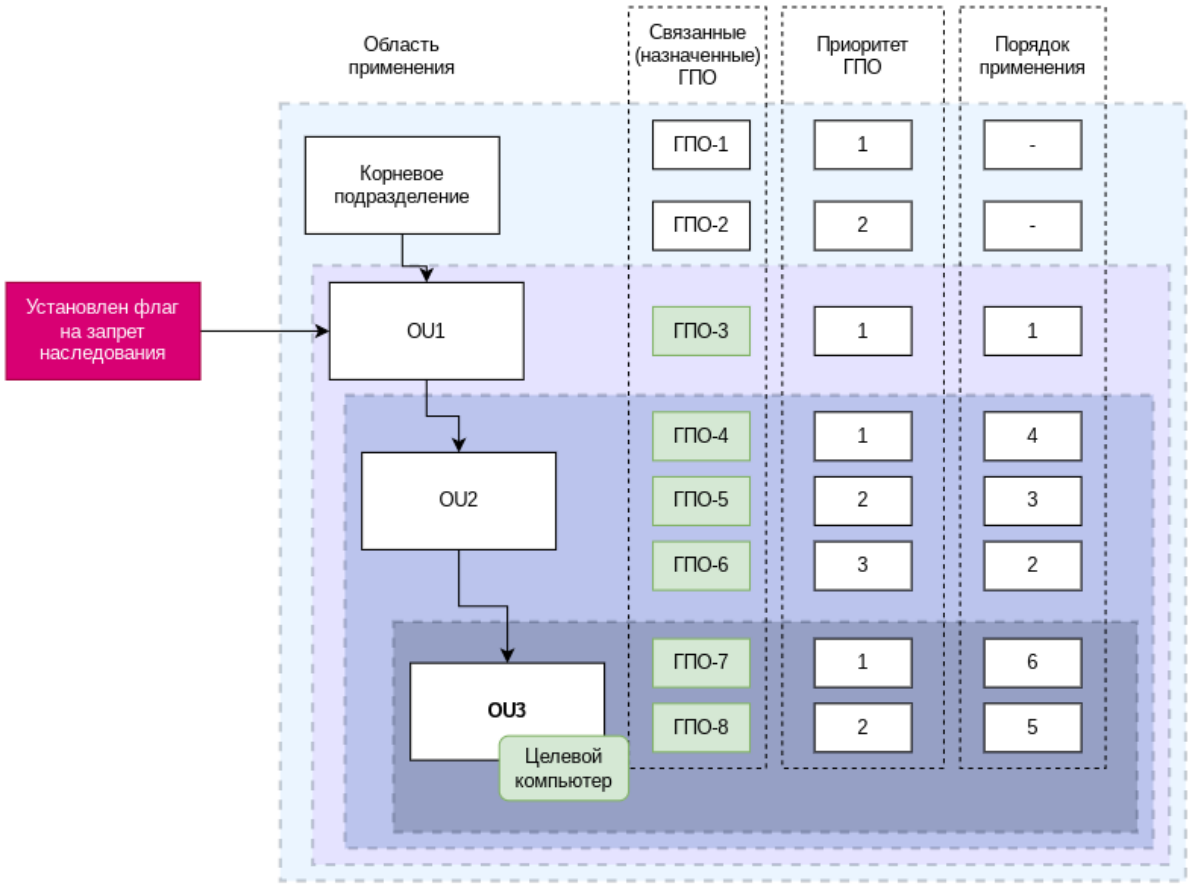


рис. 2.147 – Иллюстрация работы блокировки наследования GPO для структурного подразделения

Функция блокировки наследования удобна для отладки, и когда в рамках организационной структуры есть категории пользователей/компьютеров, на которые нужно назначить принципиально иные настройки. Например, в рамках московского офиса у вас может быть открытое пространство с совместно используемыми компьютерами, для которых проще будет задать настройки заново, чем переопределять общие настройки, назначенные в рамках офиса.

Обойти блокировку наследования можно с помощью флажка «Наследовать принудительно» (*англ. enforced*), который можно установить при назначении GPO на структурное подразделение. Более того, параметры объектов, отмеченных флажком Enforced, суммируются в отдельном цикле после суммирования параметров обычных GPO, поэтому они могут переопределить ранее установленные значения. Например, на рисунке :*aldpro_ mod6_ image33* показан пример, в соответствии с которым целевой компьютер попадает в область действия ГПО-1, несмотря на то, что для OU1 установлен запрет наследования.

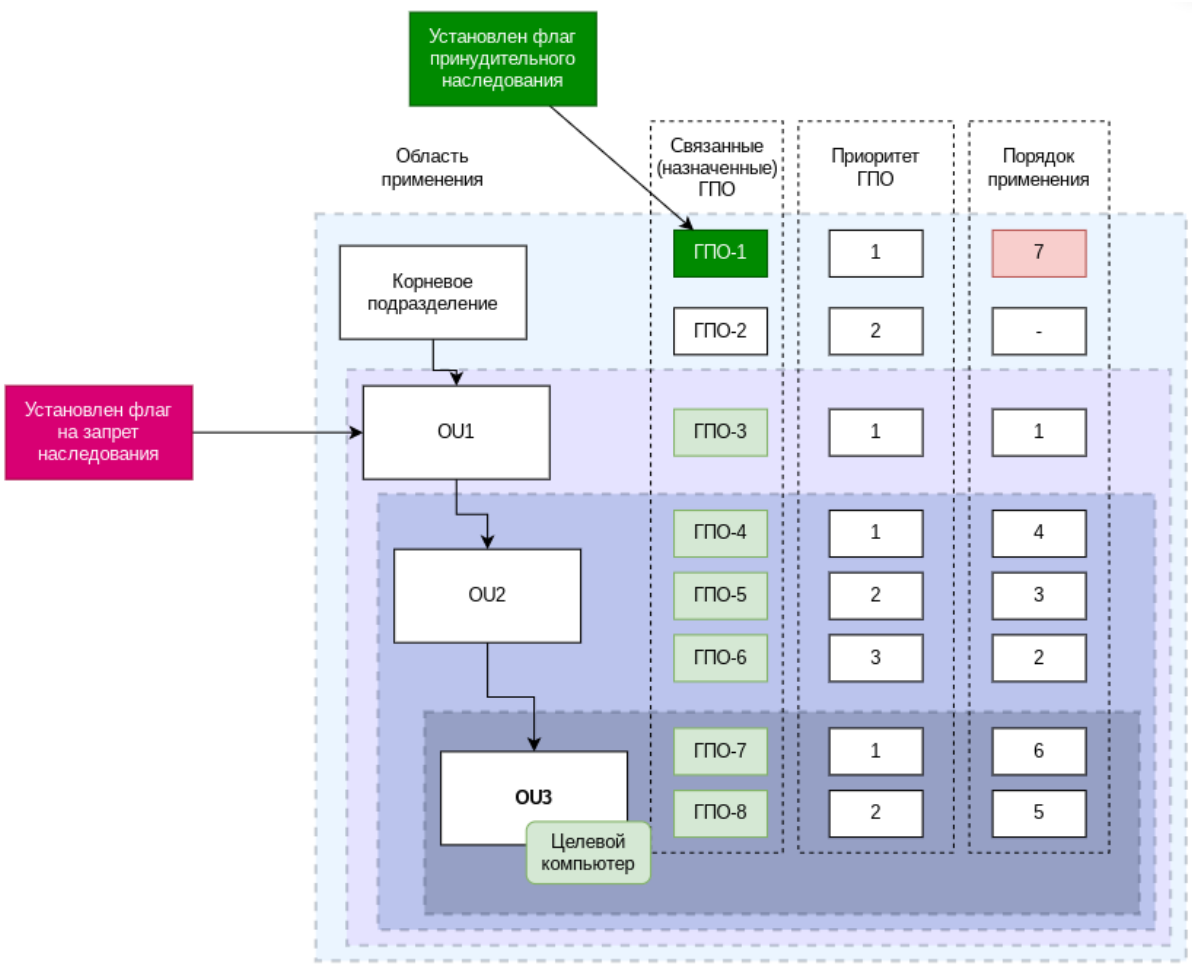


рис. 2.148 – Иллюстрация работы флага принудительного наследования

6.6.1.5 Отладка групповых политик

Служба GPSVC применяет параметры групповых политик, но не отчитывается контроллерам о результатах выполнения, поэтому администратор не располагает сводной информацией о фактическом применении параметров на всех хостах в домене.

Более того, параметры на пользователя имеют значение только в контексте конкретного входа в операционную систему, а доставлять на конкретный компьютер все параметры, определенные для всех категорий пользователей, не только бесполезно, но и еще небезопасно.

Для настройки и отладки групповых политик в домене администратор располагает следующими инструментами:

- Мастер моделирования групповой политики в GPMC — использует алгоритм службы GPSVC, чтобы спрогнозировать, какие параметры будут применены для конкретного пользователя на конкретном компьютере в заданных условиях (*Resultant Set of Policy, RSOP*). Отчет выводится в формате HTML-документа.

- Утилита `gpresult.exe` — показывает фактический результат применения параметров групповой политики на конкретном компьютере. Результаты можно вывести в формате HTML-отчета.

- Утилита `gpupdate` — позволяет форсировать применение параметров групповой политики, чтобы не дожидаться следующей итерации. Довольно часто используется с ключом /force, который позволяет повторно применить все параметры, даже если они не изменились с последнего применения.

Обычно сценарий настройки нового объекта выглядит следующим образом:

- Администратор создает новый объект групповой политики и назначает его на выделенное подразделение, в котором обычно находится один тестовый компьютер или виртуальная машина.

- Для проверки настроек администратор многократно выполняет в целевой системе команду `gpupdate /force` и сверяет полученный результат со своими ожиданиями.

- После настройки объекта групповой политики администратор назначает его на целевое структурное подразделение.

Такие инструменты, как `RSOP` и `GPResult`, используются обычно для отладки, когда нужно понять, почему та или иная настройка фактически не применилась к конкретной категории пользователей или компьютеров.

6.6.2 Групповые политики ALD Pro

Разработчики службы каталога FreeIPA решили сосредоточиться в большей степени на задачах аутентификации и авторизации, поэтому в части политик предлагают ограниченный набор параметров, имеющих прямое отношение к вопросам безопасности.

Реализованы политики FreeIPA очень хорошо, но их ни разу недостаточно. Поэтому команда ALD Pro выполнила интеграцию продукта с системой конфигурирования SaltStack, что позволило обеспечить централизованное управление настройками любых операционных систем и приложений в стиле групповых политик Microsoft.

6.6.2.1 Локальные политики

В операционной системе ALSE есть очень удобный инструмент для управления локальной политикой безопасности `fly-admin-smc`, с помощью которого можно управлять локальными пользователями и группами, настроить основные параметры безопасности, мандатного доступа, аудита, отчуждаемых носителей, см. рис. 2.149.

Вместе с тем, следует понимать, что это приложение не ведет какую-то собственную базу настроек, а считывает информацию непосредственно из конфигурационных файлов системы, поэтому при изменении параметров с помощью сторонних систем конфигурирования у вас не будет возможности вернуть значения по умолчанию.

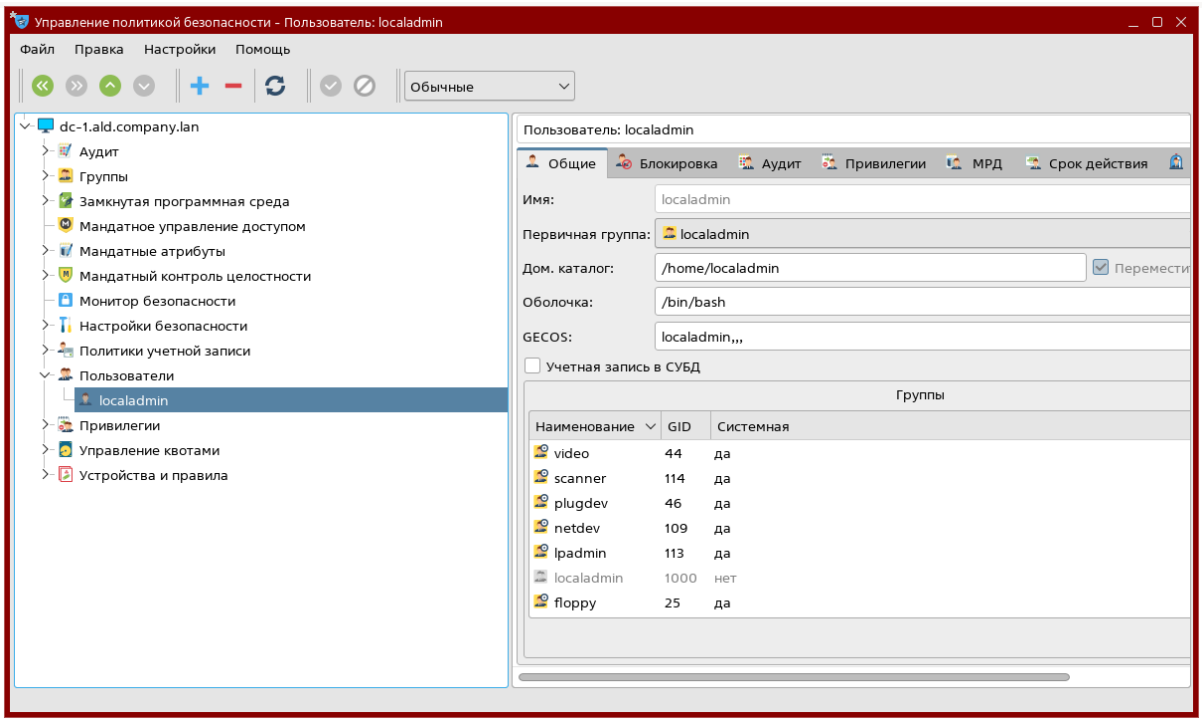


рис. 2.149 – Управление локальной политикой безопасности ALSE

Система ALD Pro не предлагает какой-либо реализации локальных политик и не резервирует изменяемые конфигурационные файлы, поэтому для возможности вернуться к исходным значениям вы можете включить версионирование всей папки `/etc/` с помощью обычного Git или более специализированного приложения `etckeeper`.

6.6.2.2 Доменные групповые политики

Групповые политики ALD Pro во многом похожи на групповые политики AD. Перечислим основные особенности:

- Параметры группируются в объекты групповой политики (GPO). В рамках каждого GPO есть две секции параметров — одна для формирования окружения компьютеров и вторая для окружения пользователей, см. рис. 2.150.
- Значения параметров (шаблоны GPO) хранятся в LDAP-каталоге и извлекаются по LDAP-протоколу, то есть обычные файлы и SMB-протокол не используются.
- За извлечение параметров из LDAP-каталога, их суммирование и применение отвечает служба

aldpro-salt-minion (до версии ALD Pro 2.4.0 salt-minion-standalone). Для взаимодействия с каталогом служба использует учетную запись компьютера из файла `/etc/krb5.keytab`.

- Для применения параметров на рабочих станциях автономная служба aldpro-salt-minion использует Salt-скрипты, тексты которых хранятся также в LDAP-каталоге.
- Объекты групповой политики назначаются на организационные подразделения, которые являются собственной доработкой продукта ALD Pro, т.к. в службе каталога FreeIPA этой возможности изначально нет.
- Назначить GPO на сайт в настоящий момент невозможно. Для назначения GPO на домен используется корневое подразделение организационной структуры.

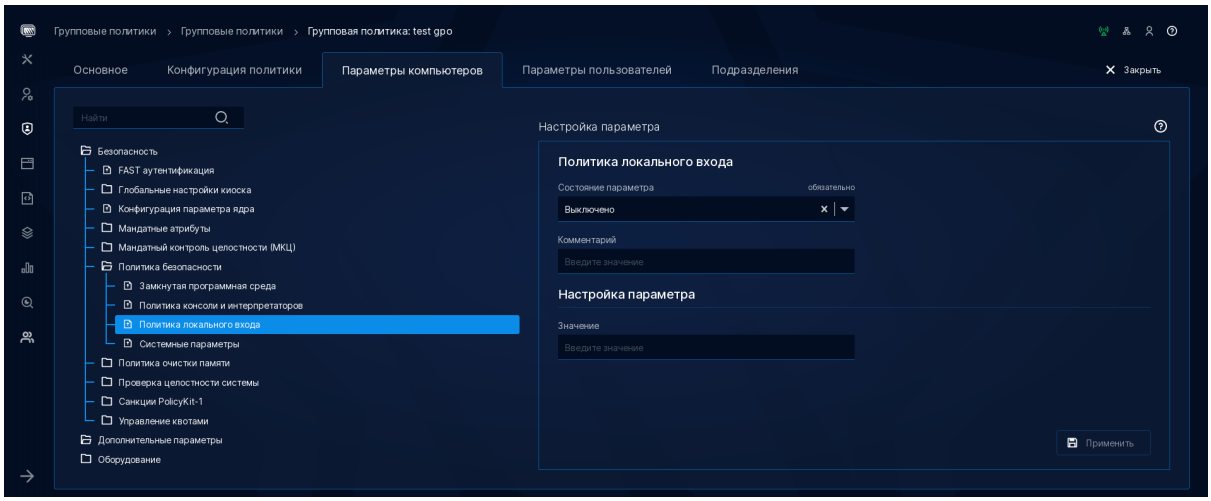


рис. 2.150 – Параметры компьютеров объекта групповой политики ALD Pro

В основе механизма групповых политик ALD Pro лежит Salt — система управления конфигурациями и удаленного выполнения операций с открытым исходным кодом, написанная на языке Python. Система работает по модели «издатель — подписчик», см. рис. 2.151, поэтому базовыми компонентами архитектуры являются:

- **Мастер** (издатель) — это сервер, выполняющий централизованное управление рабочими станциями.
- **Миньоны** (подписчики) — это рабочие станции или серверы, выступающие в качестве клиентов по отношению к Мастеру и принимающие от Мастера команды на удаленное конфигурирование.
- **Шина данных** — это система для передачи сообщений на базе ZeroMQ.

  Шина работает на Мастере и реализует модель длинных опросов (англ. long poll). Миньоны устанавливают с Шиной постоянное соединение по порту 4505/TCP (порт издателя, publisher) для получения заданий на конфигурирование от Мастера.

Кроме постоянного соединения Миньоны периодически подключаются к Шине по порту 4506/TCP (порт сервера запросов, request server) для загрузки необходимых файлов и отправки отчетов о выполнении заданий.

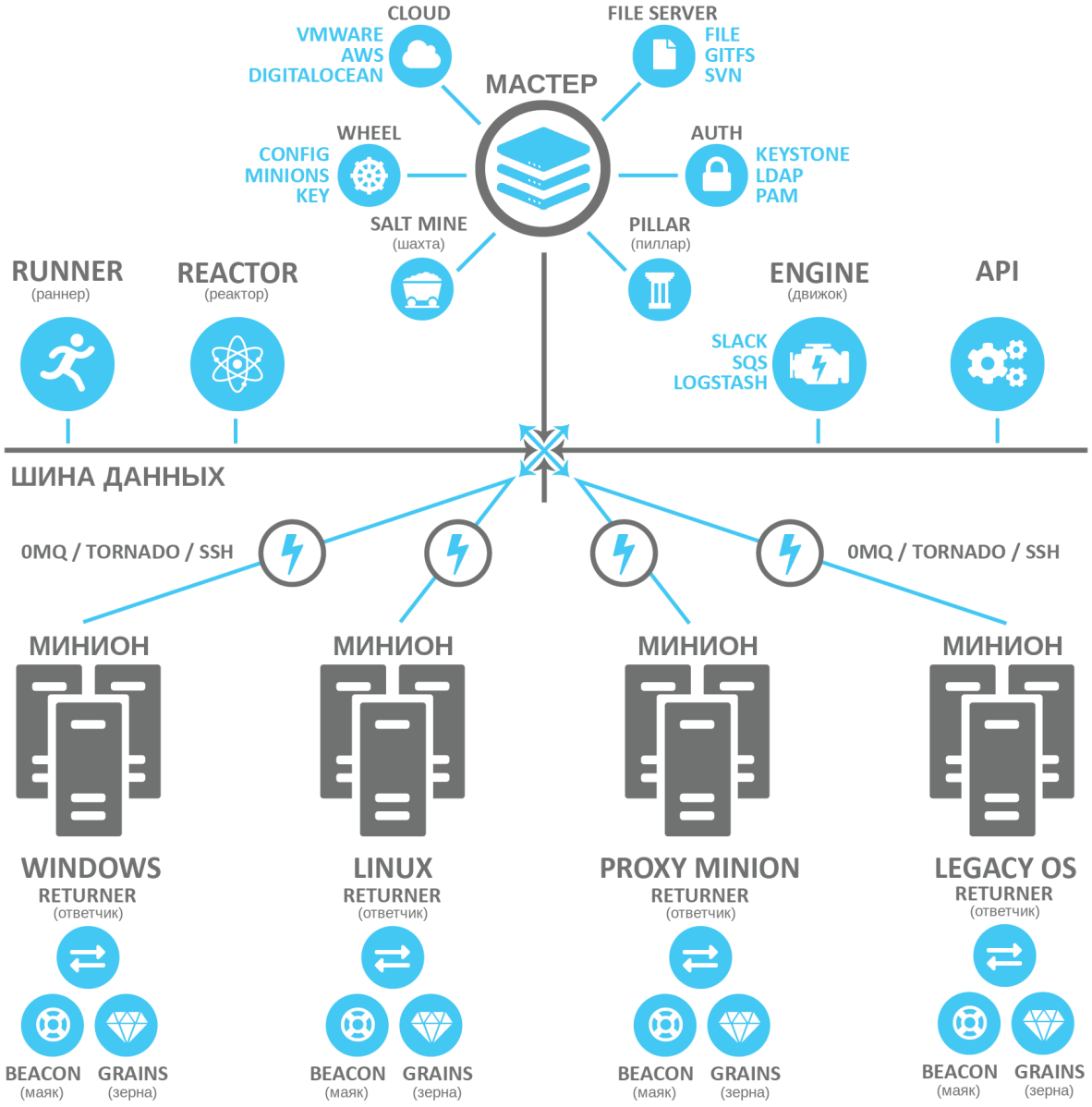


рис. 2.151 – Классическая архитектура Salt

6.6.2.3 Работа групповых политик в ALD Pro до версии 2.2.0

В продукте ALD Pro до версии 2.2.0 групповые политики использовали классическую архитектуру Salt: служба Salt-Minion рабочей станции подключалась к Salt-Master на контроллере домена, забирала справочник Pillar с результирующим набором параметров, загружала все необходимые salt-скрипты и применяла указанные параметры.

Описанная архитектура отлично подходит для управления инфраструктурой на сотни виртуаль-

ных машин с высокой сетевой связанностью, что подтверждается успешным использованием Salt в облаке LinkedIn (самое крупное внедрение продукта). Успешно применяют Salt и российские облачные провайдеры. Однако при использовании классического подхода Salt по управлению тысячами рабочих станций мы сталкиваемся с рядом сложностей:

- Суммирование параметров групповых политик на стороне сервера для построения результирующего набора параметров требует слишком много ресурсов.
- Аутентификация по ключу, которую использует Salt для установления соединения, требует ощутимо больше ресурсов. Это очень заметно в первые минуты после перезапуска службы Salt-Master.
- Удержание нескольких тысяч сетевых соединений по модели long poll требует слишком много ресурсов.

6.6.2.4 Работа групповых политик в ALD Pro с версии 2.2.0

Для того чтобы достичь показателя в 10к рабочих станций на один контроллер домена, в части групповых политик мы отказались от использования Мастера и перешли к модели Standalone Minion. Весь программный код, отвечающий за наследование и суммирование параметров групповых политик, выполняется на стороне клиента, а к контроллеру он ходит только за обновлением данных, см. рис. 2.152.

6.6.2.5 Работа групповых политик в ALD Pro с версии 2.4.0

Кардинальных изменений работы ГП в версии 2.4.0 не произошло. Перечислим некоторые значимые моменты:

- Появилась возможности влиять на порядок наследования через установку таких флажков, как «Блокировать наследование» (англ. Block Inheritance) и «Наследовать принудительно» (англ. Enforced).
- Упрошен синтаксис команд cli для форсирования применения политик и получения настроек результирующей политики (pillar).
- Переработаны скрипты коробочных параметров, повышена стабильность их работы.
- При изменении доп. параметра групповой политики, новые настройки применяются к клиентам автоматически (версия всех политик в которых используется такой доп. параметр инкриментируется).

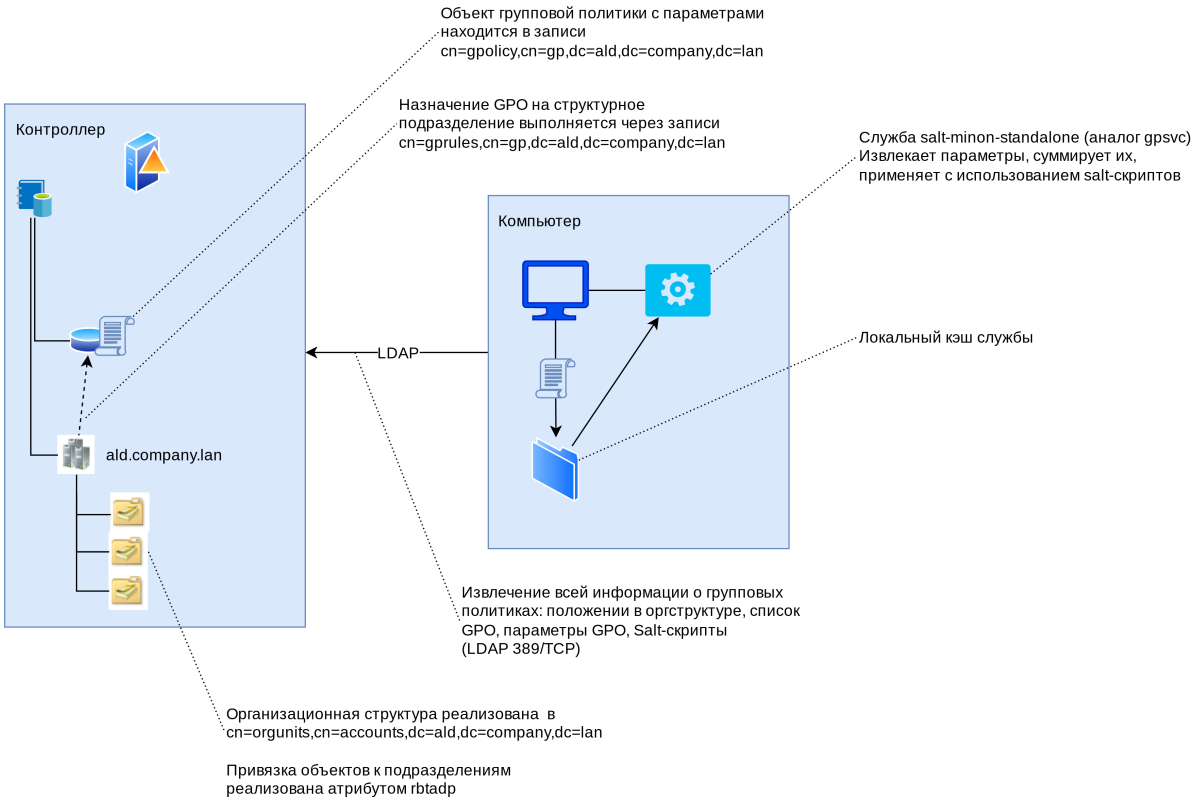


рис. 2.152 – Компоненты доменной групповой политики ALD Pro

Скрипты групповых политик находятся в каталоге `/srv/aldpro-salt/roots/states/policies/` (до версии 2.4 в каталоге /srv/salt/standalone/roots/states/policies/). В случае коробочных параметров они доставляются через установку и обновление deb-пакетов программного продукта, а в случае дополнительных параметров загружаются через LDAP.

```
admin@dc-1:~$ sudo tree -L 2 /srv/aldpro-salt/roots/states/policies/
/srv/aldpro-salt/roots/states/policies/
├── audit-policies
│   ├── files
│   └── init.sls
├── host-policies
│   ├── init.sls
│   ├── rbta_ldap_auth_fast
│   ├── rbta_ldap_crontab
│   ├── rbta_ldap_date_time_h
│   ...
│   ├── rbta_ldap_printers
│   ├── rbta_ldap_quotas
│   ├── rbta_ldap_security_policy
│   └── rbta_ldap_system_alternatives
├── sw-policies
│   ├── init.sls
│   └── README.md
├── update-policy.sls
└── user-policies
    ├── init.sls
    ├── rbta_ldap_autostart
```

(continues on next page)

(продолжение с предыдущей страницы)

```
   ├── rbta_ldap_date_time_u
...
   ├── rbta_ldap_power_management
   ├── rbta_ldap_quick_launch
   ├── rbta_ldap_start_menu
   └── rbta_ldap_windows_settings
```

Механизм групповых политик ALD Pro заимствует из MS AD также несколько очевидных способов повышения производительности:

- Служба aldpro-salt-minion не обращается к контроллеру сразу после включения, а выбирает случайным образом момент времени в окне продолжительностью в один час, чтобы не создавать пиковую нагрузку в начале рабочего дня.
- Параметры групповых политик извлекаются из LDAP-каталога только в том случае, если версия в локальном кэше не совпадает с версией GPO из каталога.

6.6.2.6 Механизм репликации

Информация по групповым политикам ALD Pro находится в LDAP-каталоге, поэтому реплицируется между контроллерами домена в штатном порядке в рамках установленной топологии.

Мы отказались от использования файлов, т.к. производительность современных серверов позволяет передавать всю необходимую информацию по LDAP-протоколу. Довольно долгое время файлы Registry.pol были не больше 64Кб, а записи размером до 100Кб хранить в каталоге считалось допустимым даже 20 лет назад.

6.6.2.7 Механизмы наследования и суммирования параметров

В части наследования и суммирования параметров система ALD Pro заимствует подходы, которые применяются в MS AD:

- Простые параметры суммируются так же, как обычные параметры групповых политик AD, поэтому, чем ближе будет GPO по иерархии к пользователю/компьютеру, тем позже будут применяться параметры этого объекта, поэтому они смогут переопределить ранее установленные значения.
- Составные параметры суммируются так же, как Предпочтения групповых политик (Group Policy Preferences), поэтому salt-скрипт через pillar получает таблицу атрибутов, где каждая строка соответствует набору атрибутов данного параметра из отдельно взятого объекта групповой политики. Скрипт должен будет самостоятельно удалить дубликаты строк, руководствуясь бизнес-логикой параметра групповой политики.

- С версии 2.4 появилась возможность влиять на порядок наследования через установку таких флажков, как «Блокировать наследование» (англ. Block Inheritance) и «Наследовать принудительно» (англ. Enforced).

Однако есть некоторые отличия:

- В соответствии с порядком наследования LSDOU в системе ALD Pro в настоящий момент можно назначать GPO только на домен и организационные подразделения, причем под доменом подразумевается корневое подразделение в иерархии. Локальных политик нет, и назначить GPO на сайт пока не представляется возможным.
- Групповые политики ALD Pro не поддерживают технологии WMI-фильтров и не позволяют назначать права доступа на отдельные GPO, поэтому в настоящий момент сузить область действия GPO не представляется возможным.

6.6.2.8 Настройка групповой политики

Давайте с помощью групповых политик запретим локальным пользователям вход в систему.

В домене MS AD для управления паролями локальных пользователей используется специальное решение LAPS (*Local Administrator Password Solution*), а для ALD Pro его планируется разработать в 2025 году, поэтому, пока оно недоступно, заблокировать локальным пользователям вход будет намного проще, чем следить за их паролями вручную.

1. Перейдем на страницу *Групповые политики → Групповые политики* и создадим новый объект групповой политики, нажав соответствующую кнопку.

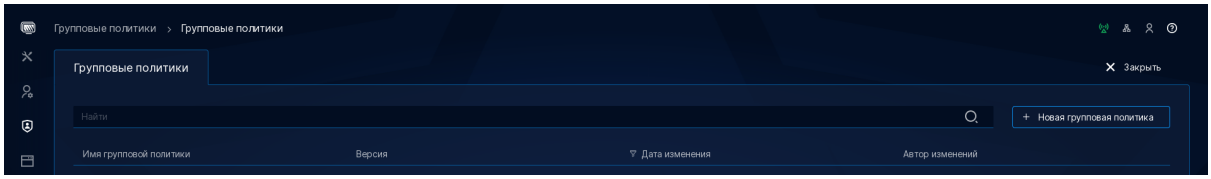


рис. 2.153 – Страница «Групповые политики»

2. До первого сохранения объекта нам доступна только вкладка *Основное*, где требуется задать имя объекта и описание. Введем следующие значения:

- Имя: Блокировка локальных учетных записей
- Описание: Политика запрещает локальным пользователям вход в систему
- И нажмем кнопку *Сохранить*

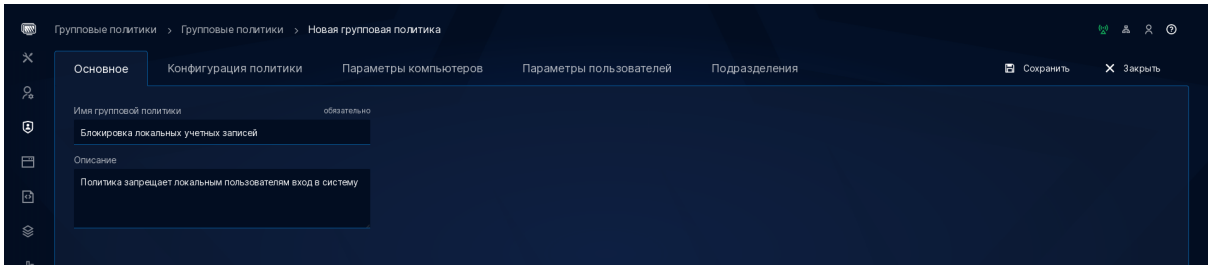


рис. 2.154 – Создание новой групповой политики

Теперь нам доступны для редактирования остальные вкладки:

- **Конфигурация политики** – отображает список параметров, которые задействованы в этом объекте групповой политики. Вы можете удалить параметр или быстро перейти к его редактированию.
- **Параметры компьютеров** – содержит параметры для настройки окружения компьютера.
- **Параметры пользователей** – параметры для настройки окружения пользователя.
- **Подразделения** – содержит список подразделений, на которые назначен объект групповой политики. На этой вкладке вы можете удалить назначение или назначить объект на дополнительные подразделения.

3. На вкладке *Параметры компьютеров* включим параметр из списка *Безопасность → Политика безопасности → Политика локального входа*, установим значение «no», *Состояние параметра* установим во «Включено» и нажмем кнопку *Применить*

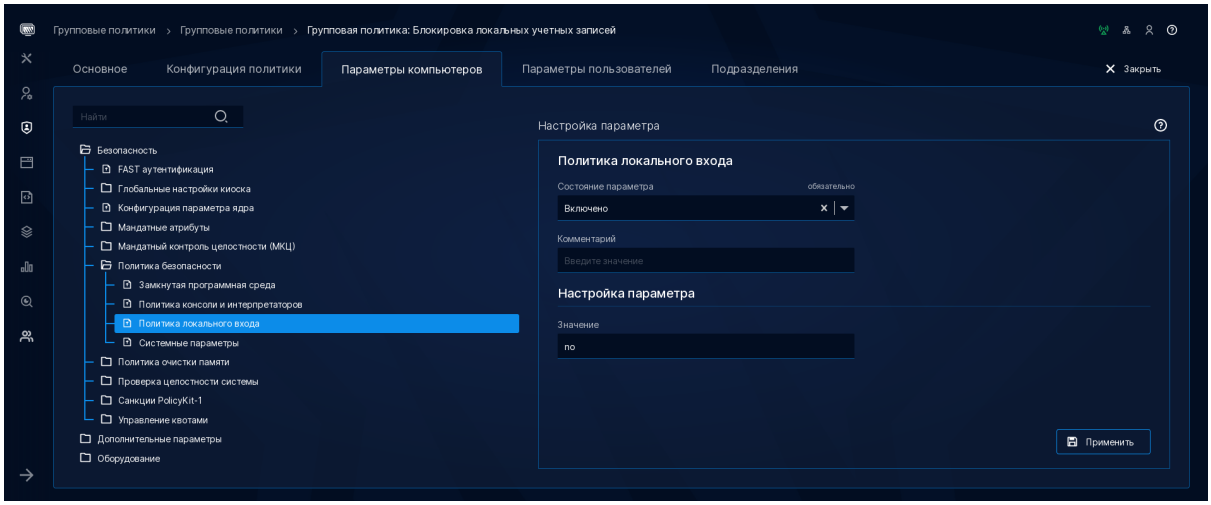


рис. 2.155 – Настройка параметров компьютеров групповой политики

Примечание: Подробную справку о параметре групповой политики можно получить, если клик-

нуть по значку с символом вопроса, который расположен в правом верхнем углу карточки. Как вы можете заметить, значение «no» означает, что вход будет запрещен для всех категорий локальных пользователей, в том числе и по ssh.

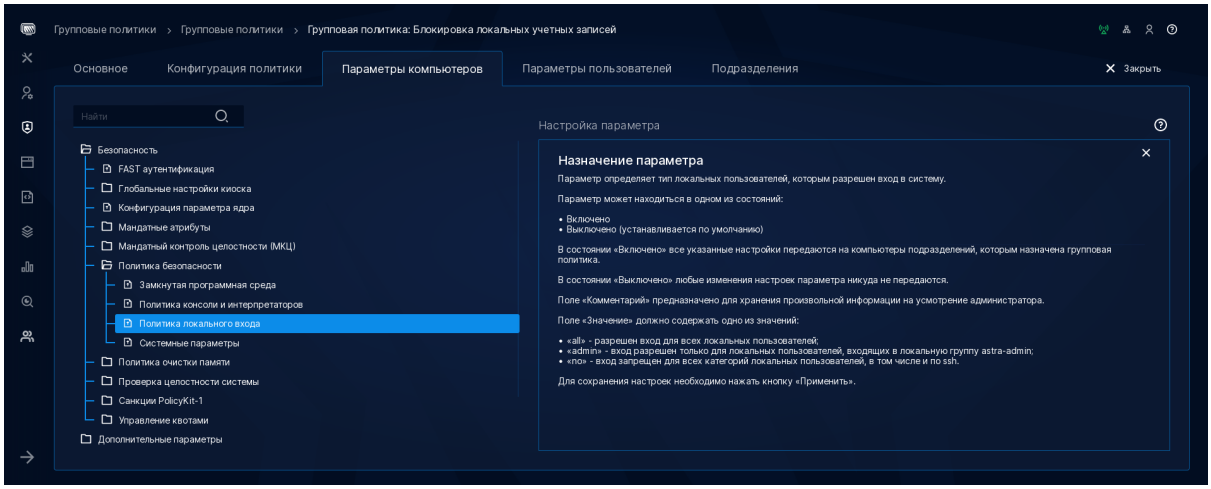


рис. 2.156 – Справка настроек параметров компьютеров групповой политики

Примечание: Чтобы закрыть справку и вернуться к форме редактирования, сделайте клик по значку в форме крестика, который расположен в правом верхнем углу карточки.

---

4. Перейдем на вкладку *Подразделения* и создадим новое назначение кнопкой *+ Добавить подразделение*.

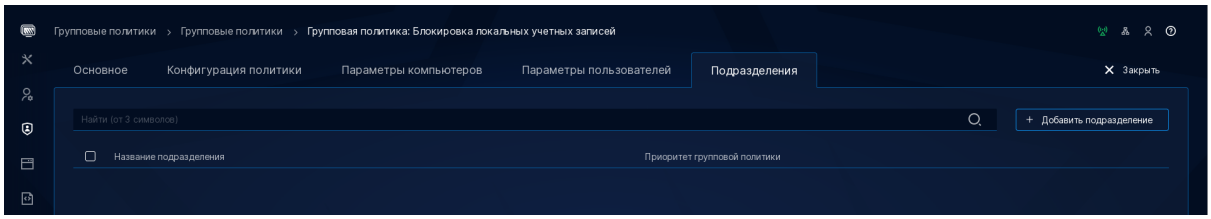


рис. 2.157 – Назначение групповой политики на подразделение

Так как мы хотим применить объект групповой политики на весь домен, то выберем корневое подразделение «ald.company.lan» и нажмем кнопку *Сохранить*.

Обратите внимание на поле «Приоритет групповой политики». Это значение позволяет определить порядок применения параметров, если на одно и тоже организационное подразделение назначено несколько объектов групповой политики.

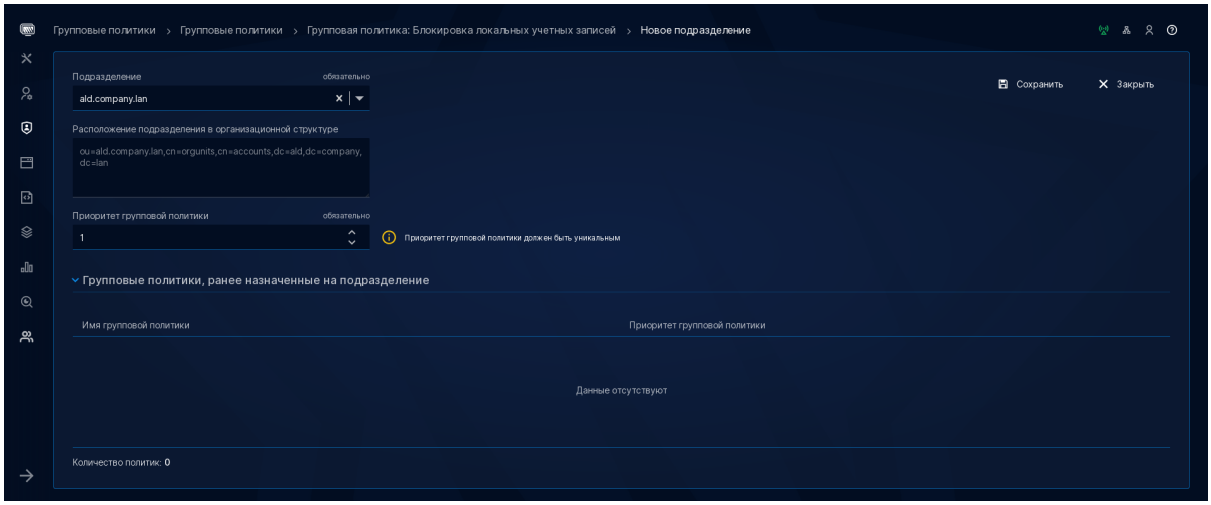


рис. 2.158 – Выбор подразделения для групповой политики

5. Файлы, которые используются для применения этого параметра, находятся в каталоге `/srv/aldpro-salt/roots/states/policies/host-policies/rbta_ldap_security_policy`.

   Скрипт `init.sls` заменяет содержимое файла `/etc/parsec/parsec.conf` шаблоном из файла `local_login_policy/parsec.conf.j2`, подставляя значение переменной *login_policy*.

   Чтобы не ждать обновления параметров, забежим немного вперед и выполним на компьютере следующие команды:

```
admin@dc-1:~$ sudo aldpro-gpupdate --gp
admin@dc-1:~$ cat /etc/parsec/parsec.conf
```

6. Если мы теперь попытаемся выполнить вход в систему локальным администратором, то получим ошибку от системы Parsec.

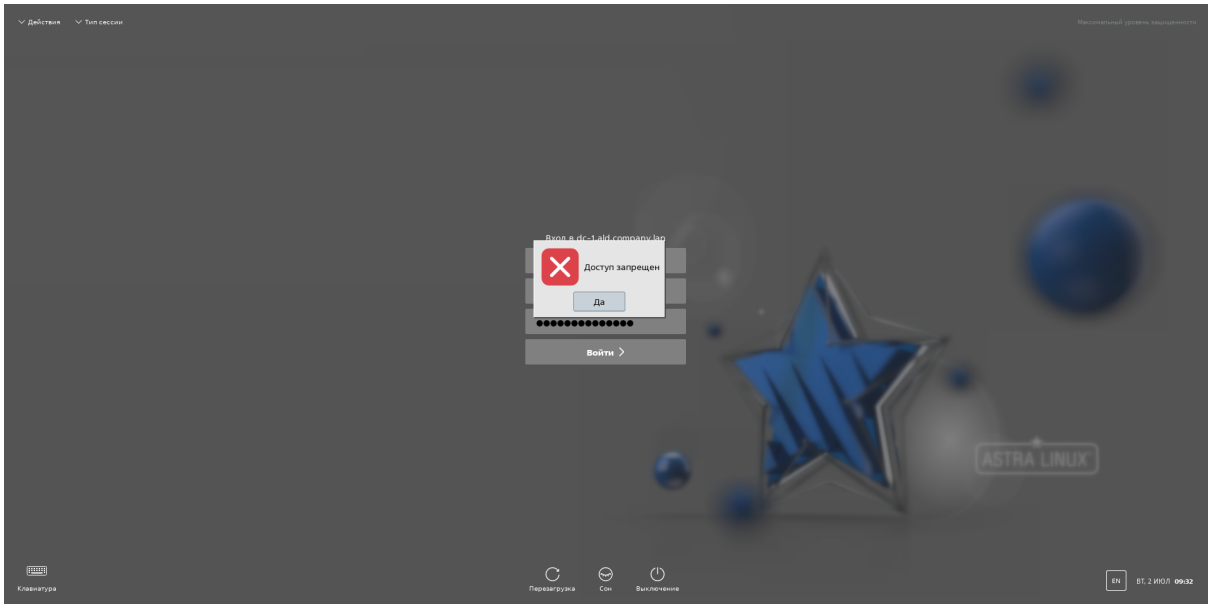


рис. 2.159 – Предупреждение при входе в систему

Если вы захотите вернуть как было, то нужно установить параметр all, обновить параметры политики и перезагрузить хост.

6.6.2.9 Механизм определения пользовательских сессий

Предлагаем немного подробнее рассмотреть реализацию механизма определения активных пользовательских сессий в ALD Pro при применении политики параметров пользователя. Обратите внимание, что параметры пользователя применяются в цикле индивидуально по отношению к каждой учетной записи, с помощью которой был выполнен интерактивный вход в операционную систему. Для операционных систем 1.8_x86-64 список сессий извлекается через интерфейс dbus (org.freedesktop.login1), а для всех остальных ОС с помощью классической утилиты users (см. модули aldpro_user.py и dbus.py).

Система не учитывает вход из-под sudo su, но не ограничивается только графическими сессиями, поэтому, если вам потребуется обогатить данные pillar параметрами другого пользователя, вы можете выполнить вход с помощью простой утилиты login:

```
admin@pc-1:~$ sudo login testuser
```

Давайте рассмотрим механизм определения сессий по шагам:

1. Пользователь подключается к рабочей станции: графическое подключения Fly или консольное подключение Login.

2. Запускаются модули PAM в зависимости от выбранного метода подключения: fly-dm (/etc/pam.d/fly-dm) или login (/etc/pam.d/login).

3. Выше упомянутые PAM модули вызывают PAM модуль Session (/etc/pam.d/common-session). В нем содержится опция запуска внешнего скрипта salt-masterless-login:

```
session optional pam_exec.so /usr/sbin/salt-masterless-login
```

PAM модуль передает переменную PAM_USER данному Python скрипту, который отделяет доменного пользователя от локального, сравнивая его с /etc/passwd. В случае, если пользователь доменный, то будет запущена следующая команда:

```
aldpro-salt-call \
--local \
-m \
/srv/aldpro-salt/roots/_modules/ \
state.single \
module.run \
aldpro_user.store_domain_user \
-c \
/srv/aldpro-salt/config \
queue=True
```

Примечание: В ванильной версии Salt для выполнения salt-функций на миньоне предназначена утилита salt-call. В продукте ALD Pro мы упаковываем salt-клиент в отдельное python окружение, поэтому для взаимодействия с нашей автономной службой aldpro-salt-minion вам следует использовать утилиту **aldpro-salt-call**.

---

4. Команда, запущенная в предыдущем шаге вызывает функцию **store_domain_user** из модуля **/srv/aldpro-salt/roots/_modules/aldpro_user.py**, по окончанию работы которой, у нас формируется файл со списком пользовательских сессий **/opt/rbta/aldpro-salt/minion/gp-user_sessions.json**

```
cat /opt/rbta/aldpro-salt/minion/gp-user_sessions.json
{"admin": {"last_active": "2025-03-19 14:33:15.275652"}}
```

Файл **gp-user_sessions.json** регулярно очищается с помощью функции **purge_inactive_sessions**. Она удаляет все сессии, которые были неактивны последние 20 дней. Запускается раз в день. Посмотреть следующий запуск можно:

```
aldpro-salt-call schedule.show_next_fire_time purge_inactive_
↪sessions
```

5. На последнем этапе запускается модуль **gp_sum.py** (/srv/aldpro-salt/roots/_modules/gp_sum.py). Функция **build_gp_pillars** из модуля **gp_sum.py** с помощью **dbus** определяет все текущие сессии, далее сравнивает их со списком **gp-user_sessions.json** и в случае совпадения строит **pillar** для этого пользователя.

На рис. 2.160 представлены описанные шаги в виде схемы для удобства восприятия.

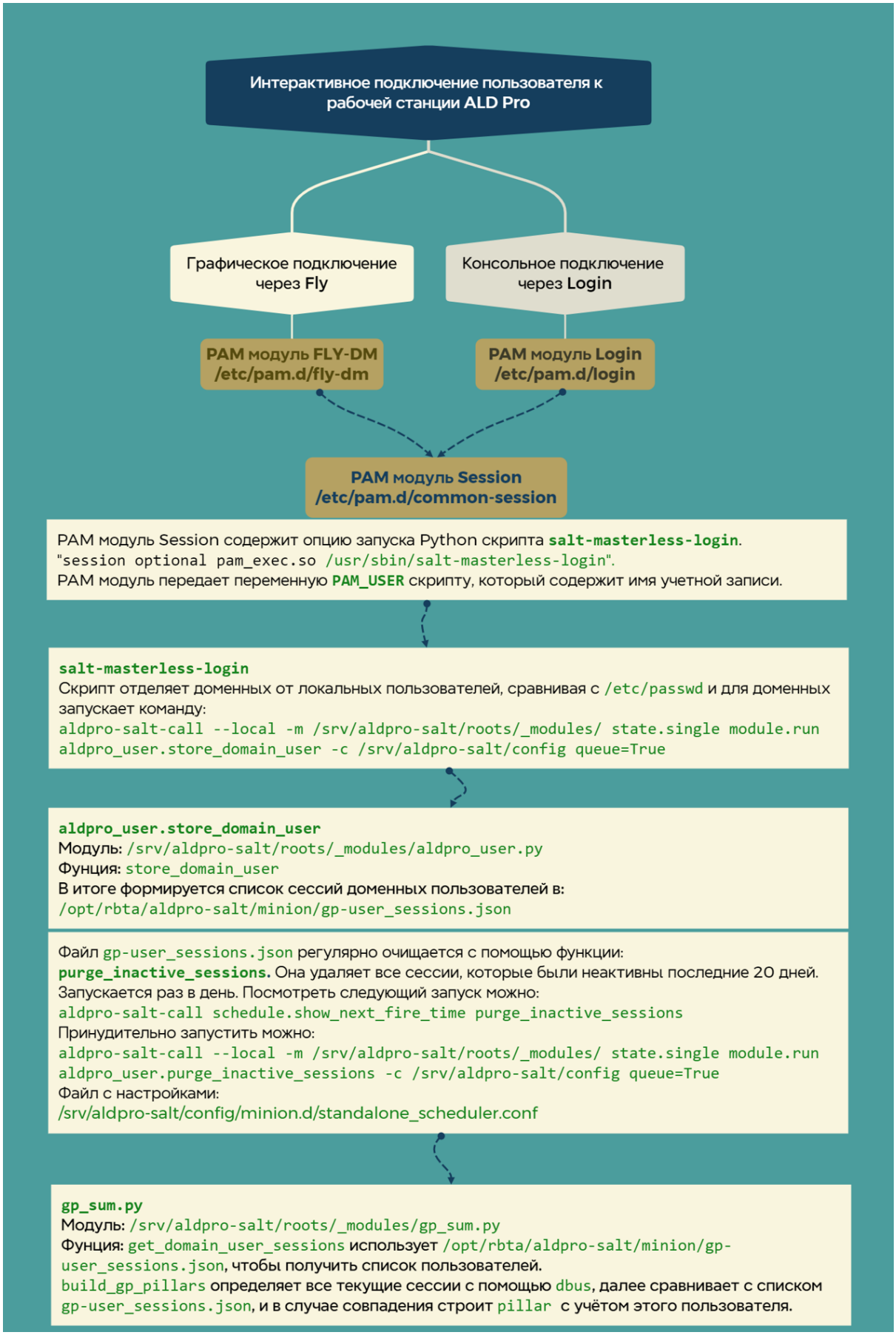


рис. 2.160 – Механизм определения пользовательских сессий

6.6.2.10 Отладка групповых политик

Служба aldpro-salt-minion применяет параметры групповых политик по расписанию. Сразу после загрузки компьютера служба запускает отложенное задание, таймаут которого складывается из двух значений:

- Постоянная часть в 30 минут — определяет минимальный интервал между последующими обновлениями параметров групповых политик.
- Случайный разброс до 50 минут — позволяет снизить пиковую нагрузку в начале дня, когда все сотрудники разом включают свои компьютеры.

Таким образом, компьютер должен гарантированно обновить параметры групповых политик в течение 80 минут после загрузки. Приблизительное время очередного обновления по расписанию вы можете узнать командой:

```
admin@pc-1:~$ sudo aldpro-gpupdate --gp_timer
Таймер заданий ГП
local:
   ----------
   next_fire_time:
       2025-03-24T14:43:09
   result:
       True
```

Вывод этой команды учитывает только постоянную составляющую таймера *next_fire_time*, поэтому возможна ситуация, когда указанное время уже истекло, но применение политик еще не состоялось. Фактически команда будет запущена в момент времени splay, значение которого можно узнать с помощью функции schedule.job_status:

```
admin@pc-1:~$ sudo aldpro-salt-call schedule.job_status build_and_run_gp
local:
 ----------
 _last_run:
     2025-03-24T14:13:09
 _next_fire_time:
     2025-03-24T14:43:09
 _next_scheduled_fire_time:
     2025-03-24T11:06:27
 _seconds:
     1800
 _splay:
     2025-03-24T14:53:30
 args:
 enabled:
     True
 function:
     gp_sum.build_and_run_gp
 jid_include:
     True
 kwargs:
```

(continues on next page)

(продолжение с предыдущей страницы)

```
    ----------
maxrunning:
    1
minutes:
    30
name:
    build_and_run_gp
return_job:
    False
run:
    True
run_on_start:
    False
splay:
    ----------
    end:
        3000
    start:
        300
```

Для выполнения отладки новых объектов групповых политик вы можете запускать принудительное обновление параметров командой:

```
admin@pc-1:~$ sudo aldpro-gpupdate --gp --verbose
```

Где:

- –gp – форсированное применение групповых политик с очисткой кэша и формированием нового набора параметров результирующей политики (pillar).
- –verbose – детальный вывод результата выполняемой команды.

Фактически «под капотом» при этом происходит вызов команды:

```
admin@pc-1:~$ aldpro-salt-call gp_sum.build_and_run_gp force=True verbose=True
```

Вызов этой команды дает тот же результат, что и `gpupdate /force` в Windows.

После форсированного применения всех параметров на рабочей станции будет сформирован *pillar* и загружены все необходимые sls-скрипты, поэтому у вас появится возможность форсировать применение отдельного параметра без полной очистки кэша. Форсированное применение отдельных параметров существенно ускоряет написание скриптов дополнительных параметров, т.к. позволяет вносить изменения не через LDAP каталог, а напрямую в локальные копии *init.sls* из соответствующих папок в директории */srv/aldpro-salt/roots/states/policies/host-policies/*

Для отладки конкретного параметра компьютера используйте функцию *state.apply*, которой нужно передать имя скрипта из каталога `/srv/aldpro-salt/roots/states/policies/host-policies`:

```
$ sudo aldpro-salt-call state.apply <имя_скрипта> force=True verbose=True
```

Например, чтобы принудительно обновить параметр «Настройки сервиса сетевого времени Chrony», за который отвечает salt-скрипт из каталога `rbta_ldap_date_time_h`, требуется выполнить следующую команду:

```
$ sudo aldpro-salt-call state.apply rbta_ldap_date_time_h force=True verbose=True
...
[INFO    ] Completed state [/etc/chrony/chrony.conf] at time 07:59:15.835203
(duration_in_ms=1.932)
...
```

И не забывайте про ключ `--log-level`, с помощью которого можно выбрать желаемый уровень детализации: all, garbage, trace, debug, profile, info, warning, error, critical, quiet (по умолчанию warning).

```
$ sudo aldpro-salt-call --log-level=debug state.apply rbta_ldap_date_time_h
```

Чтобы форсировать применение параметра пользователя, вам потребуется добавить в *pillar* дополнительный ключ *user* с идентификатором того пользователя, для которого вы хотите применить этот параметр. Напомним, что каталог с политиками пользователей находится здесь `/srv/aldpro-salt/roots/states/policies/user-policies`. Например, если параметр с идентификатором *rbta_ldap_env_vars_u* нужно применить для пользователя *admin*, то команда будет выглядеть следующим образом:

```
sudo aldpro-salt-call state.apply rbta_ldap_env_vars_u pillar="{'user':'admin'}"
```

Примечание: Команда выше применит параметры указанному пользователю только в том случае, если pillar сформирован с участием пользователя. Например, вы создали групповую политику, назначили на нужное подразделение, но при этом служба *aldpro-salt-minion* его еще не применила, то команда не отработает, так как pillar не содержит информацию для пользователя. Для ускорения процесса обновления pillar можно форсировать применение политик с помощью команды *aldpro-gpupdate*.

Для вывода параметров результирующей политики компьютера применяется команда:

```
admin@pc-1:~$ sudo aldpro-gpupdate --gp_host_pillar
Просмотр pillar ГПО компьютера
local:
   ----------
```

Так как эту команду мы запустили на компьютере pc-1.ald.company.lan, то под капотом была вызвана функция pillar.get следующим образом:

```
aldpro-salt-call pillar.get aldpro-hosts:pc-1.ald.company.lan
```

Точно такая же команда есть для просмотра данных pillar для конкретного пользователя:

```
aldpro-gpupdate --gp_user_pillar admin
```

Под капотом эта команда вызывает функцию pillar.get следующим образом:

```
aldpro-salt-call pillar.get aldpro-users:admin
```

6.6.3 Разработка дополнительных параметров

6.6.3.1 Создание дополнительного параметра

Для того чтобы в домене можно было организовать централизованное управление корпоративными приложениями, в системе ALD Pro реализована возможность создания дополнительных параметров групповых политик, которые решают ту же задачу, что и административные шаблоны Windows.

Давайте создадим дополнительный параметр групповых политик, с помощью которого можно будет централизованно управлять источниками программного обеспечения для пакетного менеджера apt.

Чтобы создать дополнительный параметр, нужно перейти на страницу *Управление доменом* → *Доп. параметры групповых политик*. Вы можете выбрать вкладку *Параметры компьютеров* или *Параметры пользователей* в зависимости от того, хотите ли вы создать параметр для настройки окружения пользователя или компьютера. Будьте внимательны, т.к. перенести параметр из одного раздела в другой после его создания не получится.

На вкладке *Параметры компьютеров* кликнем по узлу *Дополнительные параметры* и нажмем кнопку *+ Новый параметр*

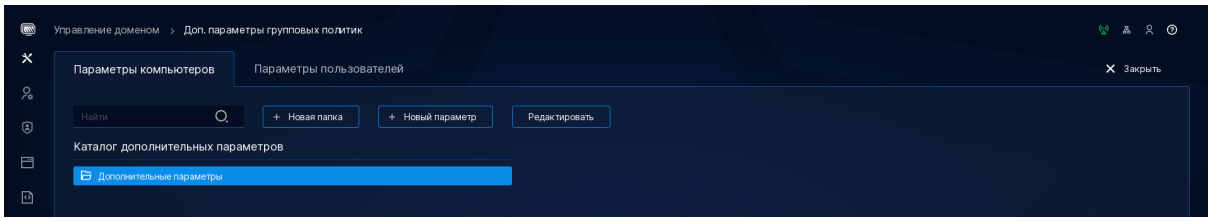


рис. 2.161 – Доп. параметры групповых политик

До первого сохранения параметра нам будет доступна только вкладка *Основное*. Заполним ее параметры:

- Название параметра: **Источники программного обеспечения**

  Это имя параметра, под которым вы его хотите видеть при редактировании объектов груп-

повых политик на портале управления.

- Уникальный идентификатор: **repo_source_list**

  а полный его идентификатор *rbta_ldap_custom_gp_host_repo_source_list*, который будет использоваться в качестве имени папки на диске и в salt-скриптах.

- Тип каталога: **Параметр компьютерной групповой политики**

  Определяет, относится ли данный параметр к настройкам компьютера или пользователя. Параметр недоступен для редактирования, его значение определяется автоматически в зависимости от того, с какой страницы вы перешли к созданию дополнительного параметра.

- Тип параметра: **Составной параметр**

  Он позволяет указать способ хранения атрибутов:

  - **Простой параметр** — задает плоский список атрибутов. Например, для настройки межсетевого экрана может потребоваться задать уровень детализации журнала, и эта настройка подразумевает одно конкретное значение, поэтому она может быть реализована атрибутом «простого» параметра.

  - **Составной параметр** — задает таблицу атрибутов, где в каждой строке представлен типовой набор данных. Если продолжать пример с межсетевым экраном, то для настройки фильтрации может потребоваться разрешить входящие ssh-соединения, запретить исходящие smtp и т.п., поэтому такие настройки нужно реализовывать атрибутами «составного» параметра.

- Папка параметра: **Дополнительные параметры**

  Указывает раздел каталога, в котором будет представлен этот параметр.

- Назначение параметра (нужно вставить весь текст целиком):

```
Позволяет централизованно настраивать источники программного
обеспечения пакетного менеджера apt путем изменения
содержания файла /etc/apt/sources.list.d/repo_source.list

Атрибут "Источник" определяет строку в формате source.list:

deb <адрес_репозитория> <код_дистрибутива> <компонент1> <компонентN>

Где:

- deb - указывает на то, что репозиторий соответствует репозиторию бинарных файлов с␣
↪предварительно скомпилированными пакетами. Для репозиториев с исходными кодами␣
↪используют «deb-src».

- адрес репозитория - задает источник пакетов, у интернет-репозиториев адрес␣
```

(continues on next page)

(продолжение с предыдущей страницы)

```
↪начинается с «http(s)://», адреса локальных репозиториев начинаются с «file:/». При␣
↪добавлении репозитория с диска командой apt-cdrom add в файле появится строка␣
↪«cdrom:[]/».

- код дистрибутива - дополняет адрес, уточняя необходимый релиз продукта, так как в␣
↪одном репозитории могут находиться пакеты сразу для нескольких релизов.

- компонент - это группа пакетов, объединенная по условиям использования.

Условия использования компонентов следующие:

 - non-free - группа содержит пакеты, которые не соответствуют принципам свободного␣
↪ПО, имеют патенты или другие юридические ограничения;

 - contrib - группа содержит пакеты, которые сами по себе соответствуют принципам␣
↪свободного ПО, но зависят от пакетов из группы «non-free», т.е. не могут без них␣
↪работать;

 - main - группа содержит пакеты свободного ПО, которые не зависят от пакетов из␣
↪групп «contrib» и «non-free».
```

- Обязательно нажмите кнопку *Сохранить*.

На вкладке атрибутов нам нужно добавить один элемент:

- Название атрибута: «**Источник**» Название, под которым вы хотите видеть этот атрибут на карточке параметра.
- Уникальный идентификатор: «**repo_source_item**» Идентификатор атрибута, под которым значение атрибута будет доступно в пилларе для его использования из salt-скрипта.
- Описание: «**Строка источника для apt**» Краткие комментарии, которые помогут упростить поддержку этого атрибута в будущем, когда потребуется внести какие-либо изменения. Данная информация недоступна при настройке групповой политики, поэтому, если вы хотите дать подсказку администратору о возможных значениях этого атрибута, внесите эту информацию в описание параметра.

Осталось только написать скрипт параметра, с помощью которого его значения будут применяться на рабочих станциях.

Скрипты Salt похожи на PHP, в них текст перемежается императивными инструкциями языка программирования, по результату выполнения которых получается итоговый документ. За императивную часть отвечают инструкции шаблонизатора Jinja2, а декларативная часть задается по правилам Salt в YAML-подобном синтаксисе.

Далее мы подробно ознакомимся с синтаксисом Jinja, а пока просто перенесем текст скрипта:

```
{% set id = 'rbta_ldap_custom_gp_host_repo_source_list' %}
```

(continues on next page)

(продолжение с предыдущей страницы)

```
{% set node = salt['grains.get']('nodename') %}
{% set gpo = salt['pillar.get']('aldpro-hosts:' + node + ':' + id) %}
{% set filename = '/etc/apt/sources.list.d/repo_source.list' %}
{% set lines = [] %}

{% if gpo %}
    {%- for item in gpo %}
        {%- if item.repo_source_item.lower() != 'none' %}
            {%- do lines.append(item.repo_source_item) %}
        {%- endif %}
    {%- endfor %}
{% endif %}

{{ id }}:
    {%- if lines|length == 0 %}
        file.absent:
            - name: {{ filename }}
    {%- else %}
        file.managed:
            - name: {{ filename }}
            - user: root
            - group: root
            - mode: 644
            - contents:
            {%- for line in lines %}
                - {{ line }}
            {%- endfor %}
    {%- endif %}
```

Данный скрипт создает файл `/etc/apt/sources.list.d/repo_source.list`, настраивает права доступа и вносит в него указанные источники программного обеспечения.

Теперь вы можете создать объект групповой политики и с помощью этого параметра централизованно настраивать источники программного обеспечения.

6.6.3.2 Синтаксис скриптов

Императивные инструкции шаблонизатора Jinja

Скрипт «Hello world»

Создайте файл test.sls с простейшим Jinja-скриптом, чтобы проверить, как это работает:

```
$ mkdir ~/test_salt && cd ~/test_salt
$ echo '{% do salt.log.info("Hello world!") %}' > hello.sls
```

Где:

- `{% ... %}` — синтаксис для вставки инструкции языка шаблонизатора Jinja;

- `do` — оператор, который вызывает указанную функцию;
- `salt.log.info` — имя вызываемой функции;
- `("Hello world!")` — параметр, передаваемый в функцию.

Выполните его с помощью утилиты aldpro-salt-call:

```
sudo aldpro-salt-call --local --file-root=. state.sls hello
```

Где:

- **aldpro-salt-call** — утилита, которая используется для локального запуска функций Salt на Миньоне без участия Мастера;
- ключ `--local` — исключает обращение к мастеру для получения настроек;
- ключ `--file-root` — устанавливает текущую директорию «.» в качестве рабочего каталога;
- параметр `state.sls` — указывает, что нужно вызывать функцию sls из модуля state;
- параметр `hello` — задает имя скрипта (без расширения).

Результат вывода должен быть таким:

```
admin@dc-1:~/test_salt$ sudo aldpro-salt-call --local --file-root=. state.sls hello
[INFO    ] Loading fresh modules for state activity
[INFO    ] Fetching file from saltenv 'base', \*\* done \*\* 'hello.sls'
[INFO    ]  Hello world!
local:

Summary for local
------------
Succeeded: 0
Failed: 0
------------
Total states run:      0
Total run time:    0.000 ms
```

Максимально подробную информацию о выполнении команды можно получить в режиме отладки. Для этого нужно добавить ключ `--log-level=debug`:

```
sudo aldpro-salt-call --log-level=debug --local --file-root=. state.sls hello
```

Чтобы вызвать salt-функцию из jinja-инструкций, нужно воспользоваться оператором **do** и обращаться к модулю **log** через встроенный объект salt. Для выполнения salt-скрипта из текстовой строки воспользуемся функцией **template_str** из модуля **state**:

```
admin@dc-1:~$ sudo aldpro-salt-call state.template_str '{% do salt.log.info("Привет,␣
↪мир!") %}'
...
```

(continues on next page)

(продолжение с предыдущей страницы)

```
[INFO    ] Привет, мир!
...
```

Разметка Jinja

Инструкции языка Jinja внедряются с помощью фигурных скобок, см. табл. 2.11:

табл. 2.11 – Выбор политики в зависимости от приоритета

Jinja	Аналог PHP	Комментарий
{% ... %}	<? ... ?>	Синтаксис для вставки инструкции языка шаблонизатора Jinja
{% print(...) %}	<? echo (...) ?>	Пример инструкции для вывода в документ по месту вызова значения выражения
{{ ... }}	<?= ... ?>	Краткий синтаксис для вывода в документ по месту вызова значения выражения
{# ... #}	<?/* ... */?>	Синтаксис для вставки комментариев средствами языка Jinja2

Для повышения читабельности кода управляющие структуры обычно оформляют отступами, но yaml-документы крайне чувствительны к пробелам, поэтому данный инструмент форматирования оказывается недоступен без применения специальных операторов подавления отступов, см. табл. 2.12. Чтобы убрать лишние пробелы и пустые строки вам нужно всего лишь поставить знак дефиса слева, справа или с обоих концов управляющего блока:

табл. 2.12 – Специальные операторы подавления отступов

Jinja	Комментарий
{%-	Удаляет пробелы и пустые строки слева
-%}	Удаляет перенос текущей строки и переносит ее на предыдущую
{%- и -%}	Удаляет пробелы и пустые строки слева, в том числе перенос текущей строки, поэтому текст окажется в конце предыдущей строки

Работа с переменными

Переменные на языке Jinja задаются оператором **set**. Вам доступен широкий перечень типов данных: булевы, числовые, текстовые, списки, кортежи, словари:

```
{% set boolean_val = True %}
{% set int_val = 777 %}
```

(continues on next page)

(продолжение с предыдущей страницы)

```
{% set string_val = 'hello' %}
{% set list_val = ['h', 'e', 'l', 'l', 'o'] %}
{% set tuple_val = ('h', 'e', 'l', 'l', 'o') %}
{% set dict_val = {'b': True, 'i': 777, 's': 'hello'} %}
```

При работе с числовыми переменными доступны обычные математические операторы:

```
{% set a = 5 %}
{% set b = 10 %}
{% set a = a + b %}
{% set b = a - b %}
{% set a = a - b %}
```

Конкатенация строк возможна как специальным оператором «~», который предварительно конвертирует все значения в строки, так и обычным оператором сложения «+»:

```
{% set a = 5 %}
{% set b = 10 %}
{% set c = a ~ b %}
{% set d = 'World' %}
{% set e = 'Hello, ' + d %}
```

Для выполнения более сложных преобразований над переменными вам доступно множество функций, которые можно применять как методы через точку или в стиле конвейеров bash через символ вертикальной черты, за что их также называют фильтрами:

```
{% set a = 'Hello'.upper() %}
{% set b = 'world' | upper %}
```

Работа со строками

Строки являются объектами типа **str**, поэтому вам доступны все методы этого класса docs.python.org

Более того, вы можете работать со строками, как со списками, обращаясь к символам по их индексам:

```
admin@dc-1:~$ sudo aldpro-salt-call state.template_str '{% do salt.log.info("Привет,␣
↪большой мир!"[0] ) %}'
...
[INFO    ] П
...
```

В квадратных скобках можно указывать три значения: индекс первого символа (включительно), индекс последнего символа (невключительно) и шаг, который позволяет включать в подстроку не все символы подряд, как по умолчанию (когда шаг=1), а, например, только каждый второй (если шаг=2). Индекс можно задавать с конца строки, в этом случае нужно использовать отрицательные

значения. Шаг может тоже иметь отрицательное значение, тогда символы в строке будут идти в обратном порядке. В качестве индекса можно подставлять результаты поиска подстроки с помощью функции find().

Приведем более сложные примеры работы со строками:

```
# Переменной "s" присвоена ссылка на строку текста
{% set s = "Привет, большой мир!" %}
# Переменной "f" присвоено значение "П"
{% set f = s[0] %}
# Переменной "f" присвоено значение "большой" с 8-го по 16-й символы
{% set f = s[8:16] %}
# Переменной "f" присвоено значение ", большой ", что соответствует фрагменту между␣
↪указанными словами
{% set f = s[s.find("Привет")+6:s.find("мир")] %}
# Переменной "f" присвоено значение "Привет" с 0-го по 6-й символы
{% set f = s[:6] %}
# Переменной "f" присвоено значение "!", которое соответствует первому символу с␣
↪конца
{% set f = s[-1] %}
# Переменной "f" присвоено значение исходной строки с обратнымпорядком символов
{% set f = s[::-1] %}
```

Работа со списками

Для работы с массивами предназначен тип данных list. Списки можно определить с помощью квадратных скобок, причем нумерация элементов будет начинаться с нуля. В работе со списками можно использовать те же приемы, что и в работе со строками:

```
# Присваиваем переменной "a" ссылку на список из трех элементов
{% set a = ["раз", "два", "три"] %}
# Присваиваем переменной "s" значение первого элемента с индексом 0, т.е. "раз"
{% set s = a[0] %}
# Присваиваем переменной "a2" ссылку на список из первых двух элементов исходного␣
↪списка, т.е. ["раз", "два"]
{% set a2 = a[0:2] %}
# Присваиваем переменной "a2" ссылку на список из трех элементов исходного списка в␣
↪обратном порядке
{% set a2 = a[::-1] %}
```

Чтобы узнать количество элементов в списке, можно воспользоваться фильтрами count или length. Добавить элемент в конец списка можно с помощью метода append(). Если нужно добавить элемент в конкретную позицию списка, используйте метод insert(). Удалить элемент по индексу можно методом pop(), а по значению - методом remove(). В программном коде Python можно заменять значение элемента напрямую по индексу (например, c[1]=»two»), но в Jinja это не работает, поэтому приходится использовать связку методов pop() и insert().

```
# Присваиваем переменной "a" ссылку на список из трех элементов
{% set a = ["раз", "два", "три"] %}
# Присваиваем переменной "c" значение, равное количеству элементов в списке "a"
```

(continues on next page)

(продолжение с предыдущей страницы)

```
{% set c = a | count }
# Добавляем в список "a" еще один элемент
{% do a.append("четыре") %}
# Добавляем в список "a" еще два элемента
{% do a.extend(["пять", "я иду искать"]) %}
# Присваиваем переменной "b" ссылку на список из двух элементов
{% set b = ["кто не спрятался", "я не виноват"] %}
# Присваиваем переменной "c" ссылку на новый список, состоящий из элементов массивов
↪"a" и "b"
{% set c = a + b %}
# Удаляем из списка "c" элемент с индексом 1, под которым сейчас находится значение
↪"два"
{% do c.pop(1) %}
# Вставляем в список "c" новый элемент "two" под индексом 1
{% do c.insert(1, "two") %}
# Удаляем из списка "c" элемент с значением "я иду искать"
{% do c.remove("я иду искать") %}
```

Полный перечень методов см. docs.python.org

Есть так же еще один способ замены элемента с помощью связки фильтров map и replace, но данный способ следует использовать с осторожностью, понимая его ограничения.

```
{% set a = ["раз", "два", "три"] %}
{% set a = a | map("replace", "три", "иду искать") | list %}
```

Работа со словарями

Для работы с ассоциативными массивами, у которых есть ключ и значение, предназначен тип данных dict. Словари можно определить с помощью фигурных скобок, доступ к объектам словаря выполняется не по индексу, а по имени ключа, которое можно указывать в квадратных скобках, через точку как свойство объекта, в качестве параметра безопасной функции get.

```
# Переменной "user" присвоена ссылка на словарь
{% set user = {"login": "localadmin", "uid": 1000, "gid": 1000} %}
# Переменной "login" присвоено значение ключа "login" из словаря "user", т.е.
↪"localadmin"
{% set login = user["login"] %}
# Переменной "uid" присвоено значение ключа "uid" из словаря "user", т.е. 1000
{% set uid = user.uid %}
# Безопаснее всего использовать метод get, который позволяет определить так же␣
↪значение по умолчанию
{% set gid = user.get("uid", -1) %}
```

Узнать количество элементов в списке, можно с помощью фильтров count или length. Чтобы получить список всех ключей, можно воспользоваться методом keys(), чтобы получить список значений - методом values().

```
# Переменной "user" присвоена ссылка на словарь
```

(continues on next page)

(продолжение с предыдущей страницы)

```
{% set user = {"login": "admin", "uid": 1000, "gid": 1000} %}
# Присваиваем переменной "c"  значение, равное количеству элементов в словаре "user"
{% set c = user | count %}
# Присваиваем переменной "k" ссылку на объект dict_keys, который содержит список всех␣
↪ключей словаря
{% set k = user.keys() %}
# Присваиваем переменной "v" ссылку на объект dict_values, который содержит список␣
↪всех значений словаря
{% set v = user.values() %}
# Присваиваем переменной "i" ссылку на объект dict_items, который содержит список␣
↪кортежей из ключей словаря
{% set i = user.items() %}
# Обновляем в словаре "a" значение ключа "uid", присваиваем ему значение 500
{% do a.update({"uid":500}) %}
```

Ветвления и циклы

Если выполнение набора команд должно зависеть от некоторого условия, то ветвление можно задать с помощью операторов if/elif/else/endif:

```
{% if ram >= 2048 %}
  {% set ram_requirement_success = True %}
{% else %}
  {% set ram_requirement_success = False %}
{% endif %}
```

Для упрощения кода условные конструкции записывать в одну jinja-инструкцию (inline if). В этом случае программный код примет вид:

```
{% set ram_requirement_success = True if ram >= 2048 else False %}
```

Переменным можно так же присваивать результаты логических операций напрямую. Тот же самый код можно записать следующим образом:

```
{% set ram_requirement_success = (ram >= 2048) %}
```

Циклы for являются аналогом конструкций типа foreach других языков программирования и предназначены для выполнения заданного набора инструкций применительно к каждому элементу из списка:

```
{% set users = ['ivanov', 'petrov', 'kuznetsov'] %}
{% for user in users %}
  {% do salt.log.info(user.upper()) %}
{% endfor %}
```

Если вам нужно будет обработать данные словаря, воспользуйтесь методом items(), чтобы получить список его элементов:

```
{% set users = {'u1':'ivanov', 'u2':'petrov', 'u3':'kuznetsov'} %}
{% for id, user in users.items() %}
  {% do salt.log.info(user.upper()) %}
{% endfor %}
```

Так как циклы Jinja работают только со списками, то для эмуляции обычных циклов со счетчиком вам нужно воспользоваться функцией range([min], max). Если передать этой функции один параметр, то будет сгенерирован список с указанным количеством элементов, нумерация которых будет начинаться с нуля. Если передать два числа, то нумерация элементов будет в указанном диапазоне.

```
{% do salt.log.info('Обратный отсчет') %}
{% for i in range(0, 10) %}
  {% do salt.log.info(10 - i) %}
{% endfor %}
```

При работе с циклами важно помнить, что переменные, объявленные с помощью *{% set %}*, существуют только в рамках текущей интерации цикла. При переходе к следующей итерации они пересоздаются заново и не сохраняют изменения.

```
{% set boolean_val = False %}

{% for item in ['a', 'b', 'c'] %}
  {% if item == 'b' %}
    {% set boolean_val = True %}
  {% endif %}
{% endfor %}

{% do salt.log.info(boolean_val) %}
```

В примере выше boolean_val был изменён в итерации на *True*, при выходе из цикла он сбросился в *False*, нам это покажет модуль *log*.

Для решения проблемы нужно использовать объект *namespace()*, который создает глобальный контейнер. Значение в *namespace()* не сбрасывается при переходе к новой итерации. В примере ниже значение *ns.boolean_val* будет равен *True*.

```
{% set ns = namespace(boolean_val=False) %}

{% for item in ['a', 'b', 'c'] %}
  {% if item == 'b' %}
    {% set ns.boolean_val = True %}
  {% endif %}
{% endfor %}

{% do salt.log.info(ns.boolean_val) %}
```

Вывод значений переменных на экран

Работая с salt-скриптами у вас нет возможности использовать отладчик, поэтому для получения дополнительной информации о работе своего программного кода используйте функцию info из метода log, чтобы вывести на экран значения отдельных переменных:

```
{% set net = "10.0.1.0/24" %}
{% set net_parts = net.split("/") %}

# Покажет список ['10.0.1.0', '24']
{% do salt.log.info(net_parts) %}
{% set ip = net_parts[0] %}

# Покажет значение '10.0.1.0'
{% do salt.log.info(ip) %}
{% set mask = net_parts[1] %}

# Покажет значение '24'
{% do salt.log.info(mask) %}
```

Быстрая проверка работы salt-функций

Работу функций salt можно проверить напрямую с помощью утилиты aldpro-salt-call. Например, если вы хотите понять, будет ли функция get_str из модуля random использовать другие классы символов, если ей задать только параметр lowercase, вы можете выполнить следующую команду:

```
admin@dc-1:~$ sudo aldpro-salt-call random.get_str length=20 lowercase=True
local:
    Z8Wta..yO|1Hq<>zf4$;
```

Работу программного кода jinja можно проверить с помощью функции template_str из модуля state. В следующем примере показано, как можно быстро проверить работу функции pop и append для работы со списками. Для вывода отладочной информации на экран используйте оператор do, с помощью которого можно вызвать функцию info из модуля log.

```
admin@dc-1:~$ sudo aldpro-salt-call state.template_str '{% set myList = ["one","two",
↪"three"] %}{% set temp = myList.pop(1) %}{% set temp = myList.append("two") %}{% do␣
↪salt.log.info(myList) %}'
[INFO    ] Loading fresh modules for state activity
[INFO    ] ['one', 'three', 'two']
local:
Summary for local----------
Succeeded: 0
Failed:
0----------
Total states run:
0
Total run time:  0.000 ms
```

Информация о целевой системе (grains)

Разрабатывая скрипты Jinja вы можете опираться на информацию о целевом хосте, на котором этот скрипт будет применен. Данная информация доступна в словаре grains (зерна), тут вы найдете информацию об операционной системе, памяти, дисках, настройках сетевых интерфейсов и многом другом. Например, используя ключ nodename, можно получить имя хоста:

```
{% set node = salt['grains.get']('nodename') %}
```

Актуальное содержание словаря grains для конкретного хоста можно посмотреть утилитой aldpro-salt-call с помощью команды `sudo aldpro-salt-call grains.items`:

```
admin@dc-1:~$ sudo aldpro-salt-call grains.items
local:
    ----------
    aldpro_dc_list:
        - dc-1.ald.company.lan
    aldpro_machine_type:
        dc
    astra_version:
        ----------
        distr:
            smolensk
        version:
            1.7.6
    basedn:
        dc=ald,dc=company,dc=lan
    biosreleasedate:
        12/01/2006
    biosversion:
        VirtualBox
    chasis_type:
        ----------
        chasis:
            Other
        is_notebook:
            False
 ...
```

Выполним запрос *grains.items* на pc-1:

```
admin@dc-1:~$ sudo aldpro-salt-call grains.items
local:
   ----------
   . . .
   dns
       ----------
       domain
       ip4_nameservers
           - 10.0.1.11
       ip6_nameservers
       nameservers
           - 10.0.1.11
       options
```

(continues on next page)

(продолжение с предыдущей страницы)

```
      search
           - ald.company.lan
      sortlist
  domain
      ald.company.lan
  efi
      **False**
  efi-secure-boot:
      **False**
  fqdn
      pc-1.ald.company.lan
  fqdn_ip4
       - 10.0.1.51
. . .
  mem_total
      **1978**
  nodename:
      pc-1.ald.company.lan
  num_cpus:
      **2**
  num_gpus
      **1**
  os:
      AstraLinux
  os_family
      Debian
. . .
```

Параметры групповых политик в pillar

Скрипт получает значения параметров групповых политик через справочник *pillar*. Служба Salt-Minion-Standalone извлекает данные из LDAP-каталога с учетом наследования, выполняет суммирование и предоставляет результирующий словарь.

Содержание словаря pillar можно посмотреть с миньона утилитой `aldpro-salt-call` (данная команда является условным аналогом GPResult в MS AD) с помощью команды

```
sudo aldpro-salt-call pillar.items
```

Содержимое справочника определено в файле `/srv/aldpro-salt/roots/pillar/top.sls`

```
base:
  '*':
    - fonts
    - policy_pillar_path
    - aldpro_audit
    - aldpro_gpo
    - aldpro_po
    - aldpro_user
```

Поясним ключи этого yaml-документа:

- На первом уровне находятся ключи, которые определяют среду окружения. В файле описана только одна среда base, которая является средой окружения по умолчанию.
- На втором уровне находятся ключи, которые таргетируют миньоны. В файле задана только одна цель с помощью символа подстановки «звездочка», под которую попадают миньоны с любым именем.
- На третьем уровне указаны включаемые sls-файлы, в которых содержатся непосредственно кэшированные значения параметров.

Механизм кэширования данных pillar довольно сложный, поэтому есть сразу несколько файлов, в которых находятся данные параметров компьютера:

- `/opt/rbta/aldpro-salt/minion/gp-`*host,user*`_settings.json` - параметры компьютера/пользователя из всех объектов групповых политик, которые назначены на текущий компьютер и всех пользователей, которые выполнили интерактивный вход в операционную систему этого компьютера
- `/opt/rbta/aldpro-salt/minion/gp-`*host,user*`_pillar.json` - результат суммирования параметров компьютера/пользователей в формате json
- `/srv/aldpro-salt/roots/pillar/aldpro_`*gpo,user*`.sls` - экспорт результатов суммирования параметров компьютера/пользователей в соответствующие sls-файлы словаря pillar

Если вы захотите изменить параметры компьютера на лету без внесения изменений в LDAP-каталог, это можно сделать в файле aldpro_{gpo,user}.sls. Вместе с тем, учитывайте, что эти изменения будут автоматически удалены при очередном применении групповых политик с обновлением кэша.

Если же вам нужно проверить, будет ли применена конкретная политика на хосте, вы можете применить состояние с параметром test=True. Если по результатам выполнения команды вы увидите сообщение «No changes needed to be made», значит система уже получила pillar с заданной ГП:

```
  # aldpro-salt-call state.apply rbta_ldap_env_vars_h test=True
...
        ID: /etc/bash.bashrc
  Function: file.blockreplace
    Result: True
   Comment: No changes needed to be made
...
```

Более подробную информацию о модулях pillar можно найти в документации по проекту: docs.saltproject.io

Декларативные описания Salt

Декларативная часть скрипта описывает в YAML-формате желаемую конфигурацию компьютера. В структуре документа указано, какие методы нужно вызывать для конфигурирования системы и с какими параметрами. В качестве примера создадим файл `software.sls` со скриптом, который описывает состояние, после применения которого в системе должен стать доступен диспетчер задач `htop`.

Создадим файл `vim software.sls` со следующим содержимым:

```
software_htop: # Имя состояния
  pkg.installed: # Вызов метода installed модуля salt.states.pkg
    - pkgs: # Аргументы метода installed
      - htop: # Значение аргумента pkgs
```

Скрипт можно выполнить с помощью команды `aldpro-salt-call`:

```
sudo aldpro-salt-call --local --file-root=. state.sls software
```

Рассмотрим параметры команды подробнее:

- –file-root=. — устанавливает рабочую директорию, в которой будет выполняться поиск *.sls файлов
- state.sls — указывает, что следует использовать функцию sls из модуля state, которая выполняет sls-скрипт
- software — указывает имя sls-файла без расширения

По результатам выполнения команды видно, что установка прошла успешно и htop готов к работе:

```
...
[INFO    ] Loading fresh modules for state activity
[INFO ] Completed state [software_htop] at time 11:39:24.930751 (duration_in_ms=3955.
↪279)
local:
----------
          ID: software_htop
    Function: pkg.installed
      Result: True
     Comment: The following packages were installed/updated: htop
     Started: 11:39:20.975472
    Duration: 3955.279 ms
     Changes:
              ----------
              htop:
                  ----------
                  new:
                      2.2.0-1
                  old:
```

(continues on next page)

(продолжение с предыдущей страницы)

```
Summary for local
------------
Succeeded: 1 (changed=1)
Failed:    0
------------
Total states run:     1
Total run time: 3.955 s
```

Язык YAML, также как и JSON, является языком разметки, который позволяет описывать иерархические структуры данных. Первые три строки в приведенном примере задают пару ключ-значение, между которыми стоит знак двоеточия. Ключи могут состоять из одного и более слов, причем заключать их в кавычки необязательно. В качестве значений поддерживаются как скалярные типы данных (int, float, boolean, string), так и вложенные словари, что позволяет создавать древовидные структуры данных.

Второй и последующий уровни дерева должны обозначаться отступами с помощью двух и более пробелов, использовать символы табуляции вместо пробелов недопустимо. Если требуется прокомментировать какой-то фрагмент документа, то слева от комментария нужно поставить символ решетки.

На первом уровне структуры данных должен быть указан ключ с именем состояния, в нашем случае это *software_ htop*. Имя выбирается произвольно, но в одном документе эти имена не должны повторяться. В предыдущих релизах ALD Pro идентификаторы состояний не должны были повторяться в пределах всех скриптов групповых политик. В актуальных версиях продукта данное ограничение снято, но мы все равно рекомендуем в идентификаторах состояний в качестве префикса использовать значение идентификатора параметра групповой политики.

На втором уровне структуры данных указывается функция, которую нужно вызвать. В нашем случае это функция *installed* из модуля *pkg* (искать по имени *salt.states.pkg*), который является модулем состояния, т. е. его методы приводят систему к заданному состоянию, описывая желаемый конечный результат, а не то, какие действия нужно выполнить. Существуют также модули исполнения, которые выполняют в системе конкретные действия, ранее мы уже показывали пример с использованием метода *run* модуля *cmd* для выполнения bash-команды.

В следующих строках определено, что функции нужно передать аргумент pkgs с значением htop. Обратите внимание, что аргумент и его значение являются элементами списков, поэтому в начале этих строк стоит символ дефиса.

6.6.4 Рекомендации к salt-скриптам

Для применения групповых политик у каждого параметра есть свой собственный скрипт, который представляет собой программный код на языке шаблонизатора Jinja. После рендеринга шабло-

на должен получаться yaml-документ, описывающий некоторую формулу Salt. Каждая формула может описывать один и более состояний Salt (стейтов), а каждое состояние, в свою очередь, определяет вызов одной и более функций из модулей Salt, выполнение которых, собственно, и вносит изменения на целевом хосте. В этом документе приведен ряд рекомендаций, которые помогут сделать ваши скрипты значительно лучше. Для пояснения рекомендаций мы приводим много примеров из проекта aldpro-group-policy-importer, поэтому настоятельно рекомендуем ознакомиться с файлами из этого проекта и использовать их в качестве дополнительного материала.

Ранее служба Salt-Minion выполняла скрипты дополнительных параметров всегда, т.е. вне зависимости от того, назначен этот параметр или нет. Именно поэтому в скрипте дополнительного параметра repo_source_list у нас есть проверка:

```
{% set id = 'rbta_ldap_custom_gp_host_repo_source_list' %}
{% set node = salt['grains.get']('nodename') %}
{% set gpo = salt['pillar.get']('aldpro-hosts:' + node + ':' + id) %}
...
{% if gpo %}
...
{% endif %}
```

С версии 2.2.0 архитектура ALD Pro в части работы групповых политик изменилась. Теперь скрипты выполняются только в том случае, если на компьютер или пользователя назначен соответствующий параметр, поэтому делать эту проверку уже необязательно, и ее оставляют для обеспечения со старыми версиями.

6.6.4.1 Идентификаторы состояний должны включать идентификатор параметра групповой политики

В предыдущих версиях продукта скрипты групповых политик объединялись в одно общее дерево состояний, поэтому разработчикам нужно было вручную обеспечивать уникальность всех используемых идентификаторов состояний, иначе могла возникать ошибка «found conflicting ID „имя_идентификатора“». Указанную ошибку было довольно трудно отловить, т.к. содержимое yaml-документов может определяться динамически с помощью управляющих конструкций jinja в зависимости от параметров pillar, которые определяет пользователь при настройке параметров групповых политик. Данная проблема решалась путем добавления к идентификаторам состояний дополнительного префикса. В роли такого префикса выступал идентификатор самого параметра, уникальность которого гарантировалось уже самой системой по умолчанию. Начиная с версии 2.4.0, каждый параметр групповой политики выполняется по отдельности, поэтому скрипты могут содержать повторяющиеся идентификаторы, но этого рекомендуется все равно избегать. Например, если идентификатор параметра называется «rbta_ldap_custom_gp_host_sec_ssh», то у состояния, которое обеспечивает перезапуск службы, идентификатор может иметь значение «rbta_ldap_custom_gp_host_sec_ssh_restart_ssh», а чтобы уменьшить вероятность опечаток,

значение идентификатора рекомендовалось формировать через подстановку значения переменной.

Пример скрипта

В следующем примере вы можете увидеть, что в первой строке идентификатор параметра записывается в переменную id, значение которой далее используется в строках 15 и 26 для формирования уникальных идентификаторов состояний.

```
{% set id = 'rbta_ldap_custom_gp_host_sec_ssh' %}
{% set node = salt['grains.get']('nodename') %}
{% set gpo = salt['pillar.get']('aldpro-hosts:' + node + ':' + id) %}
{% if gpo %}

   {%- macro getvalid(value, available) -%}
       {{- value if value!='' and value.lower() in available|lower else available[0]␣
↪-}}
   {%- endmacro %}

   {%- set permit_root_login = getvalid(gpo['permit_root_login'], ['no','yes',
↪'without-password','forced-commands-only']) %}
   {%- set client_alive_interval = gpo['client_alive_interval']|int(15) %}
   {%- set client_alive_count_max = gpo['client_alive_count_max']|int(1) %}
   {%- set protocol = gpo['protocol']|int(2) %}

{{ id }}:
       ini.options_present:
           - name: '/etc/ssh/sshd_config'
           - separator: ' '
           - strict: False
           - sections:
               PermitRootLogin: '{{ permit_root_login }}'
               ClientAliveInterval: '{{ client_alive_interval }}'
               ClientAliveCountMax: '{{ client_alive_count_max }}'
               Protocol: '{{ protocol }}'

{{ id }}_restart_ssh:
           service.running:
               - name: ssh
               - restart: True
               - watch:
                  - file: /etc/ssh/sshd_config

{% endif %}
```

6.6.4.2 После шаблонизации в формуле должно оставаться как минимум одно состояние

Управляющие конструкции Jinja позволяют динамически управлять содержанием salt формул, но эти конструкции следует использовать таким образом, чтобы после работы шаблонизатора в salt-формуле оставалось как минимум одно состояние, иначе это сильно затруднит отладку групповых политик. Даже если jinja-код определит, что никаких изменений вносить не требуется, он должен оставить в формуле хотя бы одно состояние, которое будет информировать системного администра-

тора о результатах применения данного параметра групповой политики. Для того чтобы описать такое состояние вы можете использовать одну из функций модуля test, например, *show_ notification*, *nop*, *succeed_ without_ changes*, *fail_ without_ changes*, *succeed_ with_ changes* или *fail_ with_ changes*.

Пример скрипта

В следующем примере вы можете увидеть, что в строках 39-43 описано состояние с использованием функции test.fail_without_changes, которое останется в salt-формуле, если скрипт будет запущен на операционной системе, которая не поддерживается.

```
{% set id = 'rbta_ldap_custom_gp_host_repo_source_list' %}
{% set node = salt['grains.get']('nodename') %}
{% set gpo = salt['pillar.get']('aldpro-hosts:' + node + ':' + id, default=[]) %}

{% for item in gpo %}
{% do item.update({'os_type': item.get('os_type', '').lower() }) %}
{% endfor %}

{% set os_type = salt['grains.get']('os') | lower %}

{% set lines = gpo | selectattr('os_type', '==', os_type)
                   | selectattr('repo_source_item', 'string')
                   | map(attribute='repo_source_item')
                   | map('trim')
                   | select('!=', '')
                   | unique
                   | list
%}

{{ id }}:
   {%- if os_type in ['astra', 'debian', 'alt'] %}
       file.managed:
           - name: /etc/apt/sources.list.d/repo_source.list
           - contents: |
                   {%- for line in lines %}
                   {{ line }}
                   {%- endfor %}
   {%- elif os_type in ['red os', 'redhat'] %}
       file.managed:
           - name: /etc/apt/sources.list.d/repo_source.list
           - contents: |
                   {%- for line in lines %}
                       {%- set sublines = line.split('|') %}
                       [customs_{{ loop.index }}]
                       {%- for subline in sublines %}
                       {{ subline }}
                       {%- endfor %}
                   {% endfor %}
   {% else %}
       test.fail_without_changes:
           - name: "ОС не поддерживается!"
           - failhard: False
   {% endif %}
```

6.6.4.3 Скрипт нужно выполнять только если параметр групповой политики назначен на пользователя/компьютер

В предыдущих версиях продукта все скрипты групповых политик выполнялись вне зависимости от того, назначены они на пользователя/компьютер или нет. По этой причине от разработчиков дополнительных параметров требовалось, чтобы в скрипте выполнялась проверка на наличие соответствующих ключей в данных pillar, что обычно делали следующим образом:

Пример скрипта

В следующем примере используется ветвление {% if gpo %}, которое позволяет выполнить скрипт только в том случае, если соответствующий параметр назначен на компьютер.

```
{% set id = 'rbta_ldap_custom_gp_host_dyndns' %}
{% set node = salt['grains.get']('nodename') %}
{% set gpo = salt['pillar.get']('aldpro-hosts:' + node + ':' + id) %}
{% if gpo %}

{% macro getvalid(value, available) -%}
{{- value if value!='' and value.lower() in available|lower else available[0] -}}
{%- endmacro %}

{% set domain = salt['grains.get']('domain') %}
{% set dyndns_update = getvalid(gpo['dyndns_update'], ['true', 'false'])|lower %}
{% set dyndns_update_ptr = getvalid(gpo['dyndns_update_ptr'], ['true','false
↪'])|lower %}
{% set dyndns_refresh_interval = gpo['dyndns_refresh_interval'] if gpo['dyndns_
↪refresh_interval'] != '' and gpo['dyndns_refresh_interval'] | int != 0 else '43200'
↪%}
{% set dyndns_ttl = gpo['dyndns_ttl'] if gpo['dyndns_ttl'] != '' and gpo['dyndns_ttl
↪']|int != 0 else '3600' %}

{{id}}_dyndns_update:
  ini.options_present:
    - name: /etc/sssd/sssd.conf
    - separator: '='
    - sections:
        domain/{{domain}}:
          dyndns_update: {{dyndns_update}}

{{id}}_dyndns_update_ptr:
  ini.options_present:
    - name: /etc/sssd/sssd.conf
    - separator: '='
     - sections:
        domain/{{domain}}:
          dyndns_update_ptr: {{dyndns_update_ptr}}

{{id}}_dyndns_refresh_interval:
  ini.options_present:
    - name: /etc/sssd/sssd.conf
    - separator: '='
```

(continues on next page)

(продолжение с предыдущей страницы)

```
    - sections:
        domain/{{domain}}:
          dyndns_refresh_interval: {{dyndns_refresh_interval}}

{{id}}_dyndns_ttl:
  ini.options_present:
    - name: /etc/sssd/sssd.conf
    - separator: '='
    - sections:
        domain/{{domain}}:
          dyndns_ttl: {{dyndns_ttl}}

{% endif %}
```

В последних релизах аналогичная проверка уже встроена в систему, поэтому вне зависимости от настроек групповых политик всегда будут выполняться только несколько избранных скриптов коробочных параметров. Например, всегда будет отрабатывать скрипт chrony, который по умолчанию выполняет настройку службы синхронизации времени в иерархии домена. Не смотря на то, что данная проверка теперь не является обязательной, ее рекомендуется оставлять для обеспечения обратной совместимости.

6.6.4.4 Скрипт должен гарантировать идемпотентность

Групповые политики применяются на компьютере каждый час, поэтому повторное выполнение скриптов не должно приводить к каким-либо ошибкам, а итоговый результат должен быть таким же, как и при первом запуске. Данное свойство называется идемпотентностью, и оно гарантируется при использовании функций из модулей состояния.

Пример скрипта

В следующем примере в строках 26-29 продемонстрировано использование функции из file.append из модуля состояния, которая не просто добавляет строку в файл каждый раз при ее вызове, а гарантирует, что эта строка будет добавлена, если ее еще нет в файле.

```
{% set id = 'rbta_ldap_custom_gp_host_sec_print' %}
{% set node = salt['grains.get']('nodename') %}
{% set gpo = salt['pillar.get']('aldpro-hosts:' + node + ':' + id) %}
{% if gpo %}

   {%- macro getvalid(value, available) -%}
       {{- value if value!='' and value.lower() in available|lower else available[0]␣
↪-}}
   {%- endmacro %}

   {%- set deny_print = getvalid(gpo['deny_print'], ['true', 'false'])| lower %}
```

(continues on next page)

(продолжение с предыдущей страницы)

```
   {%- set filename = '/etc/cups/cupsd.conf' %}

   {%- if deny_print=='true' %}

{{ id }}_AllowAll:
           file.blockreplace:
               - name: {{ filename }}
               - marker_start: '<Location />'
               - marker_end: '</Location>'
               - append_if_not_found: True
               - content: |
                   Order allow,deny
                   Allow from 127.0.0.1

{{ id }}_DefaultShared:
           file.append:
               - name: {{ filename }}
               - text: 'DefaultShared No'

   {%- else %}

{{ id }}_AllowAll:
           file.blockreplace:
               - name: {{ filename }}
               - marker_start: '<Location />'
               - marker_end: '</Location>'
               - append_if_not_found: True
               - content: |
                   Order allow,deny
                   Allow all

{{ id }}_DefaultShared:
           file.line:
               - name: {{ filename }}
               - mode: delete
               - match: 'DefaultShared'

   {%- endif %}


{% endif %}
```

6.6.4.5 Ресурсоемкие действия не следует выполнять повторно, если в них нет необходимости

Некоторые действия по конфигурированию целевой системы могут расходовать значительные вычислительные ресурсы, поэтому их выполнения следует избегать, если в этом нет необходимости и система уже находится в целевом состоянии. Например, если в системе уже включен мандатный контроль целостности, то при повторном выполнении скрипта включать его еще раз не требуется. Именно такую логику реализуют функции из модулей состояния, а при использовании функций выполнения, таких как cmd.run, для достижения похожего результата следует использовать такие директивы, как unless и onlyif.

Пример скрипта

В следующем примере в строке 14 продемонстрировано использования директивы unless, которая позволяет избежать повторной записи атрибута nsosversion в LDAP каталог, если его значение не изменилось.

```
{% set id = 'rbta_ldap_custom_gp_host_update_nsosversion' %}
{% set node = salt['grains.get']('nodename') %}
{% set gpo = salt['pillar.get']('aldpro-hosts:' + node + ':' + id) %}
{% if gpo %}
{%- set host = salt['grains.get']('fqdn') %}
{%- set grains_key = gpo['grains_key'] if grains.get(gpo['grains_key']) else 'lsb_
↪distrib_description' %}
{%- set nsosversion = salt['grains.get'](grains_key) %}
{{ id }}:
   cmd.run:
       - name: ipa host-mod '{{ host }}' --setattr=nsosversion='{{ nsosversion }}'
       - unless: ipa host-show '{{ host }}' --raw --all | grep 'nsosversion' | grep '
↪{{ nsosversion }}'
{% endif %}
```

6.6.4.6 Связанные состояния нужно объединять в цепочки

В тех случаях, когда какие-то состояния нужно применять в строго определенной последовательности, их нужно объединять в цепочки с помощью директивы require, т.к. очередность состояний в sls-файле не дает полной гарантии, что их функции будут выполнены в том же порядке. После парсинга yaml-документов формируется низкоуровневая структура данных (low data), которая описывает вызов функций, и к этой структуре применяются различные оптимизации, которые повлиять на порядок выполнения функций.

Дополнительные возможности по формированию цепочек состояний могут предоставить такие директивы, как *onchanges*, *watch*, *listen*, *prereq*, *onfail* и *use*, которые позволяют выполнять функцию не просто после указанной зависимости, но еще и только в том случае, если эта зависимая функция возвращает определенное состояние. Кроме уже обозначенных возможностей есть еще директива order, с помощью которой можно также повлиять на последовательность выполнения состояний, но она существенно уступает по функциональным возможностям, т.к. позволяет поставить выполнение функции только в начало (order: 1) или конец (order:last) очереди, а назначать конкретные значения очередности выполнения крайне не рекомендуется.

В следующем примере в строках 26-31 продемонстрировано использование директивы watch, которая гарантирует, что служба ssh будет перезапущена только после выполнения тех состояний, которые затрагивают изменение файла /etc/ssh/sshd_config, и только в том случае, если файл будет действительно изменен.

```
{% set id = 'rbta_ldap_custom_gp_host_sec_ssh' %}
```

(continues on next page)

(продолжение с предыдущей страницы)

```
{% set node = salt['grains.get']('nodename') %}
{% set gpo = salt['pillar.get']('aldpro-hosts:' + node + ':' + id) %}
{% if gpo %}

   {%- macro getvalid(value, available) -%}
       {{- value if value!='' and value.lower() in available|lower else available[0]␣
↪-}}
   {%- endmacro %}

   {%- set permit_root_login = getvalid(gpo['permit_root_login'], ['no', 'yes',
↪'without-password','forced-commands-only']) %}
   {%- set client_alive_interval = gpo['client_alive_interval']|int(15) %}
   {%- set client_alive_count_max = gpo['client_alive_count_max']|int(1) %}
   {%- set protocol = gpo['protocol']|int(2) %}

{{ id }}:
       ini.options_present:
           - name: '/etc/ssh/sshd_config'
           - separator: ' '
           - strict: False
           - sections:
               PermitRootLogin: '{{ permit_root_login }}'
               ClientAliveInterval: '{{ client_alive_interval }}'
               ClientAliveCountMax: '{{ client_alive_count_max }}'
               Protocol: '{{ protocol }}'

{{ id }}_restart_ssh:
           service.running:
               - name: ssh
               - restart: True
               - watch:
                  - file: /etc/ssh/sshd_config

{% endif %}
```

6.6.4.7 Используйте возможности Salt на максимум

Штатные модули Salt предоставляют множество функций, с помощью которых можно в несколько строк и с гарантированно высоким качеством настроить практически все, что угодно, поэтому прежде чем начинать работу по созданию собственных модулей, расширяющих возможности Salt, следует тщательным образом изучить те возможности, которые доступны из коробки. По той же причине следует критически анализировать уже написанные скрипты на предмет того, можно ли их дополнительно улучшить.

Резюмируя, можно сказать, что разработка собственных моделей состояния оправдано только в самых редких случаях, когда эти функции будут задействованы как минимум в 3+ скриптах и предполагают реализацию очень сложной логики.

Пример скрипта

Мы уже приводили пример с параметром rbta_ldap_custom_gp_host_sec_ssh, но его скрипт отлично подойдет в том числе и для иллюстрации этого тезиса. В строках 15-24 продемонстрирована возможность настройки обычного конфигурационного файла с помощью модуля ini, когда в файле нет секций, а в качестве разделителя между ключом и значением используется символ пробела.

```
{% set id = 'rbta_ldap_custom_gp_host_sec_ssh' %}
{% set node = salt['grains.get']('nodename') %}
{% set gpo = salt['pillar.get']('aldpro-hosts:' + node + ':' + id) %}
{% if gpo %}

   {%- macro getvalid(value, available) -%}
       {{- value if value!='' and value.lower() in available|lower else available[0]␣
↪-}}
   {%- endmacro %}

   {%- set permit_root_login = getvalid(gpo['permit_root_login'], ['no', 'yes',
↪'without-password','forced-commands-only']) %}
   {%- set client_alive_interval = gpo['client_alive_interval']|int(15) %}
   {%- set client_alive_count_max = gpo['client_alive_count_max']|int(1) %}
   {%- set protocol = gpo['protocol']|int(2) %}

{{ id }}:
       ini.options_present:
           - name: '/etc/ssh/sshd_config'
           - separator: ' '
           - strict: False
           - sections:
               PermitRootLogin: '{{ permit_root_login }}'
               ClientAliveInterval: '{{ client_alive_interval }}'
               ClientAliveCountMax: '{{ client_alive_count_max }}'
               Protocol: '{{ protocol }}'

{{ id }}_restart_ssh:
           service.running:
               - name: ssh
               - restart: True
               - watch:
                  - file: /etc/ssh/sshd_config

{% endif %}
```

6.6.4.8 Избегайте повторов

В скриптах групповых политик следует избегать следующих антипаттернов:

- дублирования управляющих конструкций jinja;
- впечатывания одних и тех же констант вручную;
- повторного описания состояний с минимальными отличиями.

Если выбор будет между тем, чтобы повторить управляющую конструкцию jinja или описание состояния, то более желательно избежать дублирования управляющих конструкций, чтобы упростить чтение кода.

6.6.4.9 Проверяйте данные pillar

Данные pillar, поступающие от пользователей, должны быть проверены тщательным образом, чтобы исключать ошибки ввода или даже злонамеренные инъекции. Например:

- Используйте метод get для безопасного извлечения параметров, чтобы исключить появление ошибок в тех случаях, когда параметр не определен. Используйте возможности метода по определению значения по умолчанию *{%- set param = gpo.get(„param“, „default value“) -%}*
- Если метод get использовать невозможно, задействуйте фильтр default для установки значений по умолчанию. Например, *{%- set param = gpo[„param“] | default(„installed“) -%}*
- Целочисленные значения:
  - Приводите целочисленные атрибуты к целым числам с помощью фильтра int. Например, {%- set param = gpo[„param“] | int(45) -%}
  - Ограничивайте целочисленные атрибуты допустимым диапазоном значений «сверху» с помощью фильтра min. Например, {%- set param = [100, param] | min -%}
  - Ограничивайте целочисленные атрибуты допустимым диапазоном значений «снизу» с помощью фильтра max. Например, {%- set param = [0, param] | max -%}
- Строковые значения
  - Удаляйте лишние пробелы в начале и конце строки с помощью фильтра trim. Например, {%- set param = gpo[„param“] | trim -%}
  - Приводите строку к требуемому регистру символов с помощью фильтров lower и upper. Например, {%- set param = gpo[„param“] | lower -%}

Для более сложной обработки входных данных используйте макросы, например:

```
{%- macro getvalid(value, available) -%}
   {{- value if value!='' and value.lower() in available|lower else available[0] -}}
{%- endmacro %}
...
{%- set pkg_state = getvalid(item.pkg_state, ['installed', 'removed']) %}
```

6.6.4.10 Проверяйте неявные преобразования

Язык шаблонизации jinja предоставляет большие возможности, но вместе с тем несет и большие угрозы, т.к. требует тщательной проверки используемых синтаксических конструкций, в первую очередь в отношении неявного преобразования данных. Например, когда вы выполняете ветвление по значению переменной, в которой должен быть предположительно список элементов, вы должны понимать, что этот список неявно преобразуется к переменной Boolean и от того, каким образом выполняется это преобразование, зависит работа вашего алгоритма.

Примечание: Пустые списки интерпретируются в Jinja как False, а списки, в которых есть какие-то значения, как True

---

```
{% if items %}
{# ветка, которая будет выполнена, когда в списке items есть элементы #}
{% else %}
{# когда список пуст #}
{% endif %}
```

6.6.4.11 Импорт дополнительных параметров

В разделе дополнительных источников информации приведена ссылка на архив с набором дополнительных параметров групповых политик из личного кабинета клиента.

В архиве находится в том числе скрипт `policy.py`, с помощью которого дополнительные параметры можно автоматически импортировать в домен. Скрипт написан на языке Python 3 и взаимодействует с контроллером домена через REST API.

Перед выполнением импорта следует определить параметры подключения к серверу через переменные окружения. Пароль можно передавать открытым текстом, но скрипт поддерживает в том числе и Kerberos-аутентификацию.

```
export ALDPRO_API_SERVER_URL='dc-1.ald.company.lan'
export ALDPRO_API_CA_PATH='/etc/ipa/ca.crt'
export ALDPRO_API_LOGIN='admin'
export ALDPRO_API_PASSWORD='Pa$$w0rd'
```

Для импорта всех параметров сразу можно использовать скрипт import.sh, который по цепочке вызовет остальные sh-скрипты, создающие объекты в каталоге.

```
admin@dc-1:~$ sh import.sh
```

Пример создания папки из скрипта `import_folder_common.sh`:

```
python3 policy.py folder add \
--host \
--parent-folder 'Дополнительные параметры' \
--folder-name 'Общие параметры'
```

Пример создания параметра `import_parameter_common_firefox_policy.sh`:

```
python3 policy.py parameter add \
--host \
--id 'rbta_ldap_custom_gp_host_firefox_policy' \
--name 'Политика браузера Firefox' \
--parent-folder 'Общие параметры' \
--description 'Позволяет централизованно настраивать параметры браузера Firefox путем␣
↪изменения содержания файла /usr/lib/firefox/distribution/policies.json

Атрибут "Блокировать доступ к странице addons" устанавливает значение параметра␣
↪BlockAboutAddons
- true - доступ заблокирован
- false - доступ предоставлен

Если значение не задано, то параметр будет отсутствовать в файле политики, по␣
↪умолчанию доступ предоставляется.

Атрибут "Блокировать доступ к странице config" устанавливает значение параметра␣
↪BlockAboutConfig
- true - доступ заблокирован
- false - доступ предоставлен

Атрибут "Список доменов для Kerberos-аутентификации" устанавливает значение параметра␣
↪Authentication.SPNEGO, элементы списка разделяются запятыми, например:
- ald.company.lan, win.company.lan

Атрибут "Список доверенных сертификатов" устанавливает значение параметра␣
↪Certificates.Install. Требуется указывать полный путь к файлу на локальном диске␣
↪целевой машины, элементы списка разделяются запятыми, например:
- /etc/ipa/ca.crt, /etc/ssl/certs/ca-certificates.crt

Атрибут "Адрес домашней страницы" устанавливает значение параметра Homepage.URL,␣
↪требуется указать полный URL адрес страницы, например
- https://dc-1.ald.company.lan

Если адрес домашней страницы задан, то дополнительно будут установлены параметры␣
↪Homepage.Locked=true и Homepage.StartPage=homepage-locked' \
--attr 'Блокировать доступ к странице addons':block_about_addons:'' \
--attr 'Блокировать доступ к странице config':block_about_config:'' \
--attr 'Список доменов для Kerberos-аутентификации':authentication_spnego:'' \
--attr 'Список доверенных сертификатов':trusted_certificates:'' \
--attr 'Адрес домашней страницы':homepage_url:'' \
--script 'import_parameter_common_firefox_policy.sls' \
--script-comment 'Версия 1.0'
```

6.7 Управление и настройка ПО с помощью политик ALD Pro

Система ALD Pro позволяет централизовано устанавливать и настраивать программное обеспечение через «Политики ПО» и «Групповые политики». Если первые обеспечивают более удобное управление и простоту настройки, то вторые большую гибкость и сложность.

- «Групповые политики» используются для приведения и поддержания в заданном состоянии настроек на доменных компьютерах. Групповые политики не предусматривают возможность установки и настройки ПО «из коробки». Однако, создавая дополнительные параметры групповых политик, мы ограничены только тем, какие модули и методы Salt мы можем использовать.

- «Политики ПО» позволяют устанавливать программы только из deb-пакетов доступных в заранее подготовленных репозиториях. Настройка программы тут может выполняется через создание файлов из шаблонов с параметрами. Политики ПО позволяют устанавливать, обновлять и удалять (если ПО было установлено через политику) программы, создавать конфигурационные файлы. Однако, политики ПО не применимы для выполнения более сложных действий при конфигурировании ПО, например, запуск службы, создание символических ссылок, запуск программ. Для этих целей могут быть использованы дополнительные параметры групповых политик.

С одной стороны, управление ПО через «Групповые политики» отрывает почти безграничные возможности по установке ПО. Можно устанавливать программы хоть из пакетов, хоть из исходных кодов, хоть используя AppImage или устанавливать программы, работающие через Wine. Так же мы можем использовать возможности Salt для создания и изменения конфигурационных файлов приложения.

С другой стороны, создание дополнительных параметров групповых политик не тривиальная задача для среднестатистического администратора. Тут необходимо базовое знание Salt и умение писать bash скрипты, императивные шаблоны Jinja2 и декларативные состояния Salt. Это увеличивает порог входа для использования этого подхода. Однако этот порог может быть понижен, если будет подготовлена толковая инструкция по настройке такого дополнительного параметра ГП.

Кроме того, в ALD Pro разработали специальный инструмент командной строки policy.py – утилиту на Python, которая, используя специальные входные файлы определенного формата, автоматически создает в домене ALD Pro дополнительные параметры групповых политик. Администратору остается только создать групповую политику с необходимом дополнительным параметром, корректно задать значения для этого параметра и назначить её на целевые АРМ.

Минусом установки ПО как через «Политики ПО», так и через «Групповые политики» является отсутствии обратной связи со стороны целевого компьютера. Администратор может централизовано назначить установку, обновление, удаление или настройку ПО через интерфейс ALD Pro, но

не сможет централизовано собрать данные о том, где настройки применились корректно, а где не применились или применились с ошибками.

Продемонстрируем в данном модуле возможности настройки ПО через групповые политики и дополнительные параметры.

В качестве примера «сложного» ПО, требующего дополнительных шагов для корректной работы, установим и настроим клиент RuBackup. Причем сделаем это с помощью дополнительных параметров групповых политик.

Для установки ПО через политики ПО, в качестве источников пакетов необходимо использовать дополнительную подсистему ALD Pro – сервер репозитория ПО, которая рассмотрена в *Модуль 11. Дополнительные подсистемы и интеграции* в разделе *Подсистема «Репозиторий программного обеспечения»*, где мы с вами детальнее рассмотрим работу политики ПО, установим сервер репозитрия ПО и разольем Yandex браузер на клиентские рабочие станции. Установка пакетов через групповые политики может осуществляться из любого источника.

6.7.1 Управление через «Групповые политики»

6.7.1.1 Создание доп. параметров групповой политики для управления и настройки ПО

Для того чтобы в домене можно было организовать централизованное управление корпоративными приложениями, в системе ALD Pro реализована возможность создания дополнительных параметров групповых политик, которые решают ту же задачу, что и административные шаблоны Windows.

Создадим дополнительный параметр групповых политик, с помощью которого можно будет централизованно управлять источниками программного обеспечения для пакетного менеджера apt.

Создание доп. параметров групповых политик доступно в одноименном разделе портала управления *Управление доменом → Доп. параметры групповых политик*. Вы можете выбрать вкладку *Параметры компьютеров* или *Параметры пользователей* в зависимости от того, хотите ли вы создать параметр для настройки окружения пользователя или компьютера. Будьте внимательны, т. к. перенести параметр из одного раздела в другой после его создания не получится.

1. Так как установка агента RuBackup это настройка общая для всего хоста, на вкладке *Параметры компьютеров* кликнем по узлу *Дополнительные параметры* и нажмем кнопку *+ Новый параметр*.

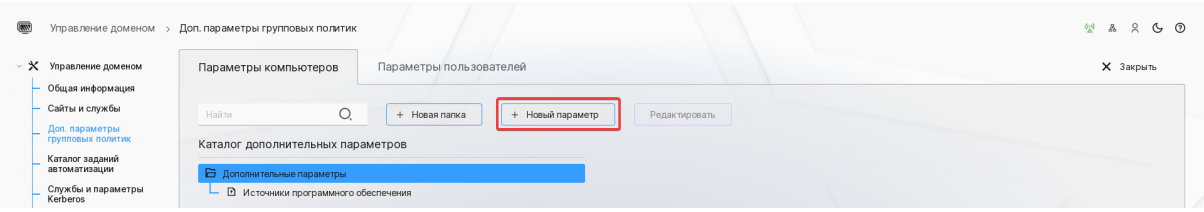


рис. 2.162 – Создание нового дополнительного параметра

2. До первого сохранения параметра нам будет доступна только вкладка *Основное*. Заполним ее параметры:

- Название параметра: *Управление клиентом RuBackup*.

  Это имя параметра, под которым вы его хотите видеть при редактировании объектов групповых политик на портале управления.

- Уникальный идентификатор: *manage_rubackup_client*.

  Полное значение идентификатора будет «rbta_ldap_custom_gp_host_manage_rubackup_client». Оно будет использоваться в качестве имени папки на диске и в salt-скриптах.

- Тип каталога: *Параметр компьютерной групповой политики*.

  Определяет, относится ли данный параметр к настройкам компьютера или пользователя. Параметр недоступен для редактирования, его значение определяется автоматически в зависимости от того, с какой страницы вы перешли к созданию дополнительного параметра.

- Тип параметра: *Простой параметр*.

Позволяет указать способ хранения атрибутов:

  - *Простой параметр* — задает плоский список атрибутов. Например, для настройки межсетевого экрана может потребоваться задать уровень детализации журнала, и эта настройка подразумевает одно конкретное значение, поэтому она может быть реализована атрибутом «простого» параметра.

  - *Составной параметр* — задает таблицу атрибутов, где в каждой строке представлен типовой набор данных. Если продолжать пример с межсетевым экраном, то для настройки фильтрации может потребоваться разрешить входящие ssh-соединения, запретить исходящие smtp и т.п., поэтому такие настройки нужно реализовывать атрибутами «составного» параметра.

- Папка параметра: *Дополнительные параметры*.

  Указывает раздел каталога, в котором будет представлен этот параметр.

- Назначение параметра:

```
Позволяет централизованно управлять клиентом RuBackup: устанавливать, настраивать и␣
↪удалять.
Атрибут "Требуемое действие" определяет тип требуемого действия над клиентом RuBackup␣
↪и
может принимать значения:
install - для установки
remove - для удаления
Атрибут "Основной сервер RuBackup" определяет имя или ip адрес мастер-сервера␣
↪RuBackup.
Используется в случае установки или перенастройки клиента.
```

- *Требования к ОС* оставьте по умолчанию.

  Тут задается информация о том, с какими версиями ОС совместим данный параметр.

Обязательно нажмите кнопку *Сохранить*.

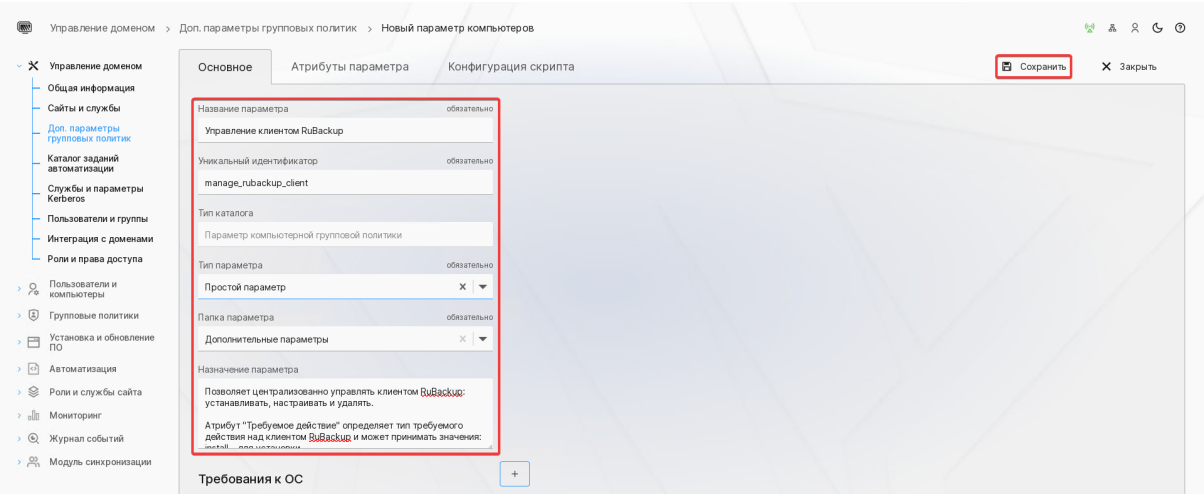


рис. 2.163 – Вкладка «Основное» дополнительного параметра

3. На вкладке «Атрибуты параметра» нам нужно добавить два элемента:

a. «Требуемое действие»

- *Название атрибута*: Требуемое действие.

  Название, под которым вы хотите видеть этот атрибут на карточке параметра.

- *Уникальный идентификатор*: action.

  Идентификатор атрибута, под которым значение атрибута будет доступно в пилларе для его использования из salt-скрипта.

- *Описание*: Требуемое действие: install или remove.

  Краткие комментарии, которые помогут упростить поддержку этого атрибута в будущем, ко-

гда потребуется внести какие-либо изменения. Данная информация недоступна при настройке групповой политики, поэтому, если вы хотите дать подсказку администратору о возможных значениях этого атрибута, внесите эту информацию в описание параметра.

- *Обязательный*: отметьте галочкой.

  Атрибуты, отмеченные как *Обязательный*, должны быть заполнены при применении параметра.

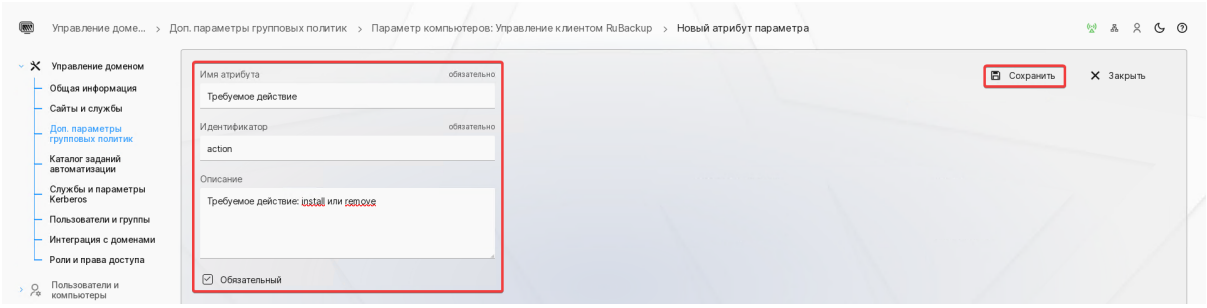


рис. 2.164 – Вкладка «Атрибуты параметра» дополнительного параметра.

b. «Основной сервер RuBackup».

- *Название атрибута*: «Основной сервер RuBackup».
- *Уникальный идентификатор*: «rubackup_master».
- *Описание*: «Имя или ip адрес мастер-сервера RuBackup».
- *Обязательный*: оставьте пустым.

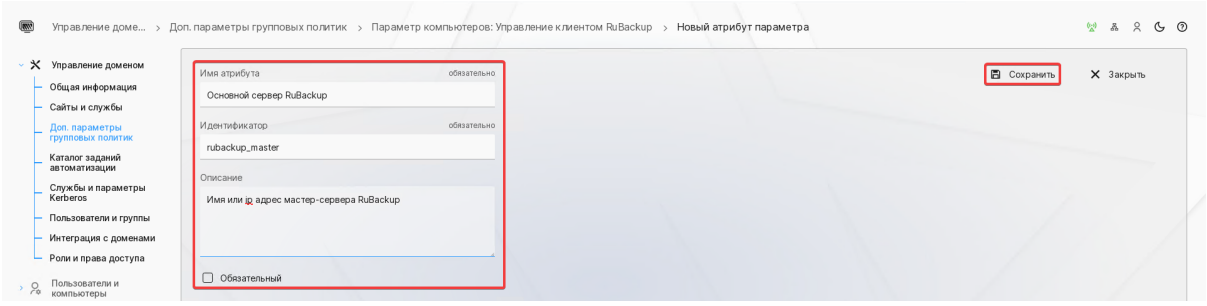


рис. 2.165 – Вкладка «Атрибуты параметра» дополнительного параметра

4. На вкладке *Конфигурация скрипта* необходимо внести текст Salt-скрипта и заполнить обязательное поле комментарий, по которому можно будет идентифицировать его версии.

Нажмите кнопку *Редактировать файл* и перенесите текст скрипта, приведённый ниже:

```
# Определение переменных из параметров задания автоматизации
{% set id = 'rbta_ldap_custom_gp_host_manage_rubackup_client' %}
{% set node = salt['grains.get']('nodename') %}
```

(continues on next page)

(продолжение с предыдущей страницы)

```
{% set gpo = salt['pillar.get']('aldpro-hosts:' + node + ':' + id) %}

{% if gpo %}
  {% set action = gpo['action'] %}
  {% set rubackup_master = gpo['rubackup_master'] %}

  # Установка
  {% if action.lower() == 'install' %}
    {% do salt.log.info("Установка клиента RuBackup") %}

    {{ id }}_install_rubackup_client:
      pkg.installed:
        - name: rubackup-common
        - name: rubackup-client

    {{ id }}_root_profile_configure:
      file.append:
        - name: /root/.bashrc
        - text:
          - "PATH=$PATH:/opt/rubackup/bin"
          - "LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/opt/rubackup/lib"
          - "export PATH"
          - "export LD_LIBRARY_PATH"

    {{ id }}_configure_rubackup_client:
      {% if not salt[ 'file.file_exists' ]('/opt/rubackup/keys/master-key') %}
        cmd.run:
          - name: '/opt/rubackup/bin/rb_init --agree-with-eula --primary-fqdn "{{␣
↪rubackup_master }}" --client-iface eth0 --remote-replica --centr-recovery --local-
↪backup-dir /rubackup-tmp --used-ip-version ipv4 --node client --monitoring'
          - require:
            - pkg: rubackup-client
      {% else %}
        file.keyvalue:
          - name: /opt/rubackup/etc/config.file
          - key: who-is-primary-server
          - value: {{ rubackup_master }}
          - separator: ' '
          - ignore_if_missing: True
      {% endif %}

    {{ id }}_start_rubackup_client_service:
      service.running:
        - name: rubackup_client
        - enable: true
        - require:
          - pkg: rubackup-client

  # Удаление
  {% elif action.lower() == 'remove' %}
    {% do salt.log.info("Удаление клиента RuBackup") %}

    {{ id }}_stop_rubackup_client:
      service.dead:
        - name: rubackup_client
        - enable: false
```

(continues on next page)

(продолжение с предыдущей страницы)

```
    {{ id }}_remove_rubackup_client:
      pkg.purged:
        - name: rubackup-client
        - name: rubackup-common

    {{ id }}_housekeeping_config_files1:
      file.line:
        - name: /root/.bashrc
        - mode: delete
        - match: 'PATH=\$PATH:/opt/rubackup/bin'
        - quiet: True

    {{ id }}_housekeeping_config_files2:
      file.line:
        - name: /root/.bashrc
        - mode: delete
        - match: 'LD_LIBRARY_PATH=\$LD_LIBRARY_PATH:/opt/rubackup/lib'
        - quiet: True

    {{ id }}_housekeeping_config_files3:
      file.line:
        - name: /root/.bashrc
        - mode: delete
        - match: 'export PATH'
        - quiet: True

    {{ id }}_housekeeping_config_files4:
      file.line:
        - name: /root/.bashrc
        - mode: delete
        - match: 'export LD_LIBRARY_PATH'
        - quiet: True

    {{ id }}_housekeeping_config_files5:
      file.absent:
        - name: /opt/rubackup/
  {% else %}
    {% do salt.log.info("Ошибка! Выбрано некорректное действие {{ action }}") %}
  {% endif %}
{% endif %}
```

Данный скрипт, в зависимости от типа выбранного действия, установит и настроит или удалит клиент RuBackup.

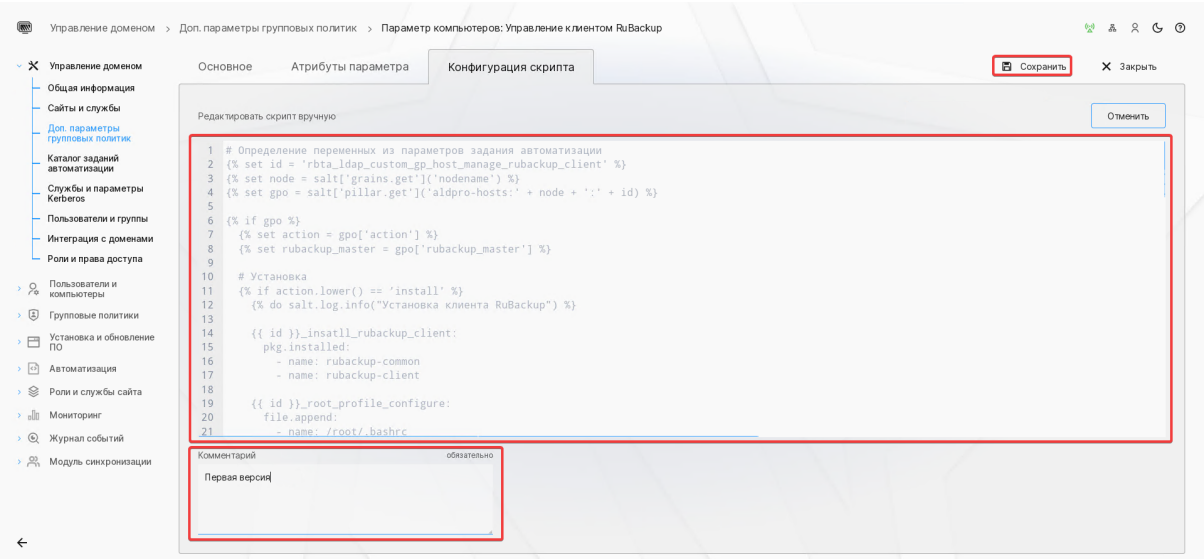


рис. 2.166 – Вкладка «Конфигурация скрипта» дополнительного параметра

6.7.1.2 Создание, настройка и применение групповой политики.

Для применения доп. параметра ГП на доменных компьютерах необходимо создать и настроить групповую политику с этим параметром.

1. Для создания групповой политики перейдите в раздел «Групповые политики» и нажмите соответствующую кнопку *Групповые политики → Групповые политики → + Новая групповая политика* .

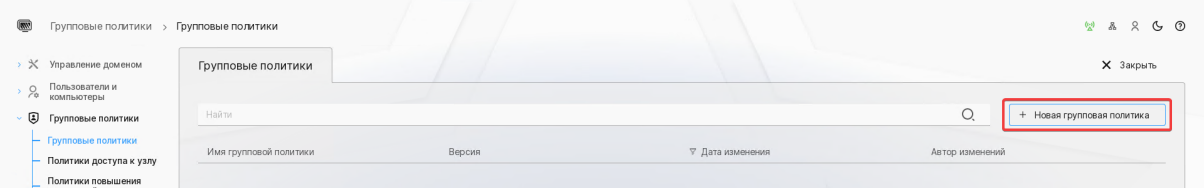


рис. 2.167 – Создание новой групповой политики

2. До первого сохранения объекта нам доступна только вкладка «Основное», где требуется задать имя объекта и описание. Введем следующие значения:

- *Имя* : Управление клиентом RuBackup
- *Описание* : Устанавливает, перенастраивает или удаляет клиент RuBackup

И нажмем кнопку *Сохранить* .

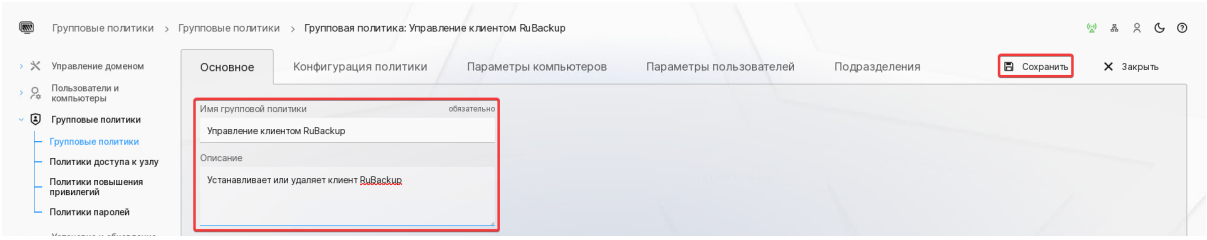


рис. 2.168 – Групповая политика вкладка «Основное»

Теперь нам доступны для редактирования остальные вкладки:

- *Конфигурация политики* – отображает список параметров, которые задействованы в этом объекте групповой политики. Вы можете удалить параметр или быстро перейти к его редактированию.
- *Параметры компьютеров* – содержит параметры для настройки окружения компьютера.
- *Параметры пользователей* – параметры для настройки окружения пользователя.
- *Подразделения* – содержит список подразделений, на которые назначен объект групповой политики. На этой вкладке вы можете удалить назначение или назначить объект на дополнительные подразделения.

3. Включим параметр *Управление клиентом RuBackup* (*Параметры компьютеров* → *Дополнительные параметры* → *Управление клиентом RuBackup*) и установим значение поля:

- *Требуемое действие* : install
- *Основной сервер RuBackup* : имя или ip адрес вашего сервера RuBackup

и нажмем кнопку *Применить* .

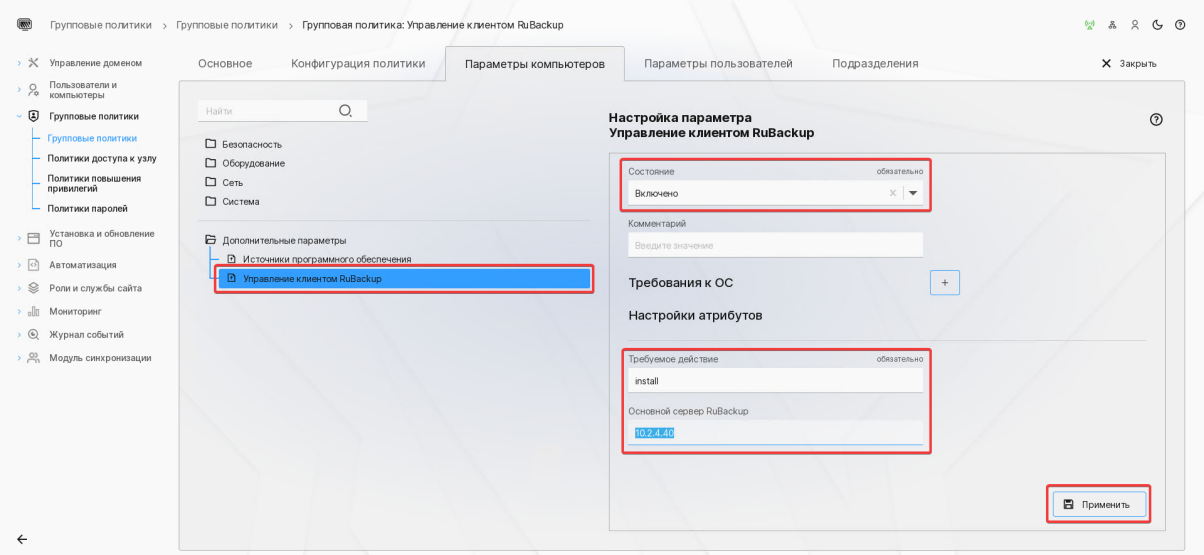


рис. 2.169 – Групповая политика вкладка «Параметры компьютеров»

Подробную справку о параметре групповой политики можно получить, если кликнуть по значку с символом вопроса, который расположен в правом верхнем углу карточки. Как вы можете заметить, это тот самый текст, который мы вводили в поле *Назначение параметра* при создании доп. параметра ГП.

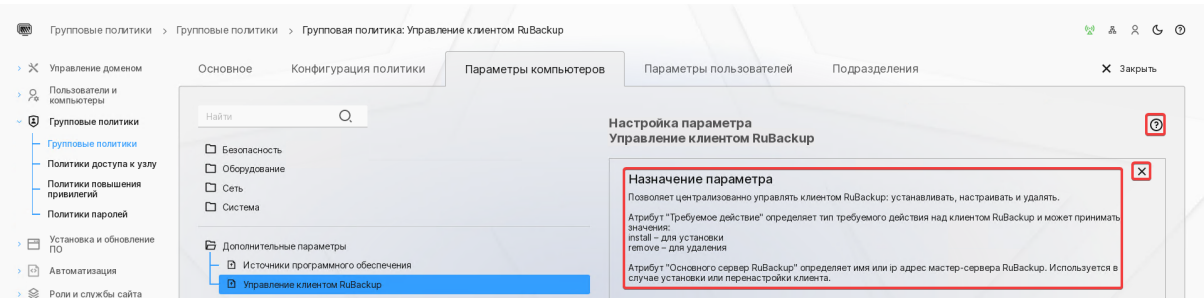


рис. 2.170 – Вызов справки по групповой политке

Чтобы закрыть справку и вернуться к форме редактирования, сделайте клик по значку в форме крестика, который расположен в правом верхнем углу карточки.

4. Перейдем на вкладку *Подразделения* и добавим новое назначение кнопкой *+ Добавить подразделение*.

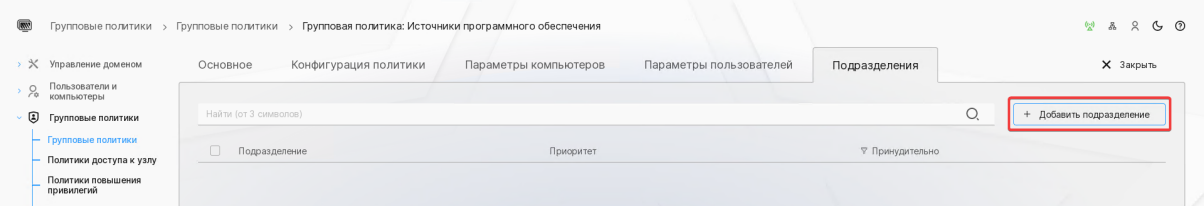


рис. 2.171 – Групповая политика вкладка «Подразделения»

4.1. Так как мы хотим применить объект групповой политики на клиентские компьютеры, то выберем подразделение, например, *Клиенты* и нажмем кнопку *Сохранить*.

Обратите внимание на поле *Приоритет групповой политики*. Это значение позволяет определить порядок применения параметров, если на одно и то же организационное подразделение назначено несколько объектов групповой политики.

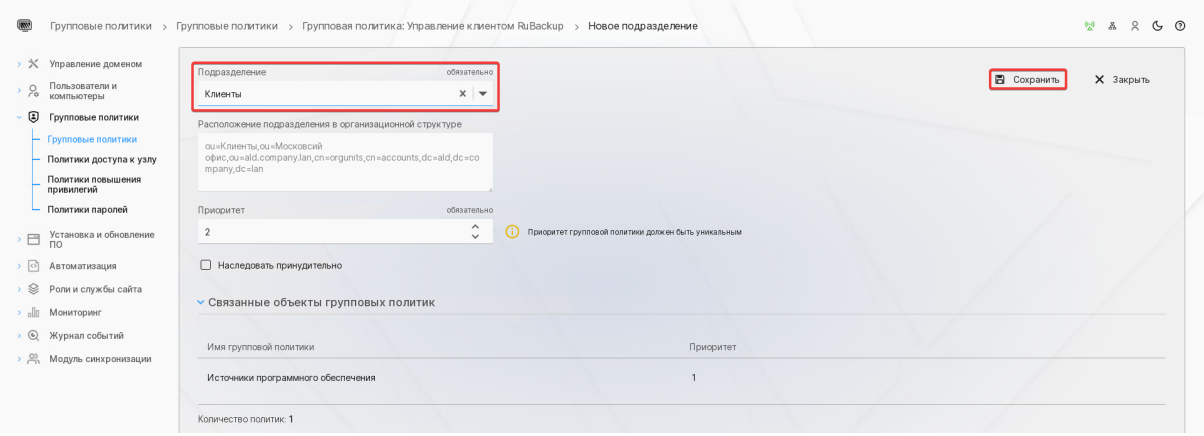


рис. 2.172 – Групповая политика выбор подразделения

5. В карточке компьютера из подразделения *Клиенты* можно убедиться, что параметр ГП на него назначен и посмотреть значения его атрибутов.

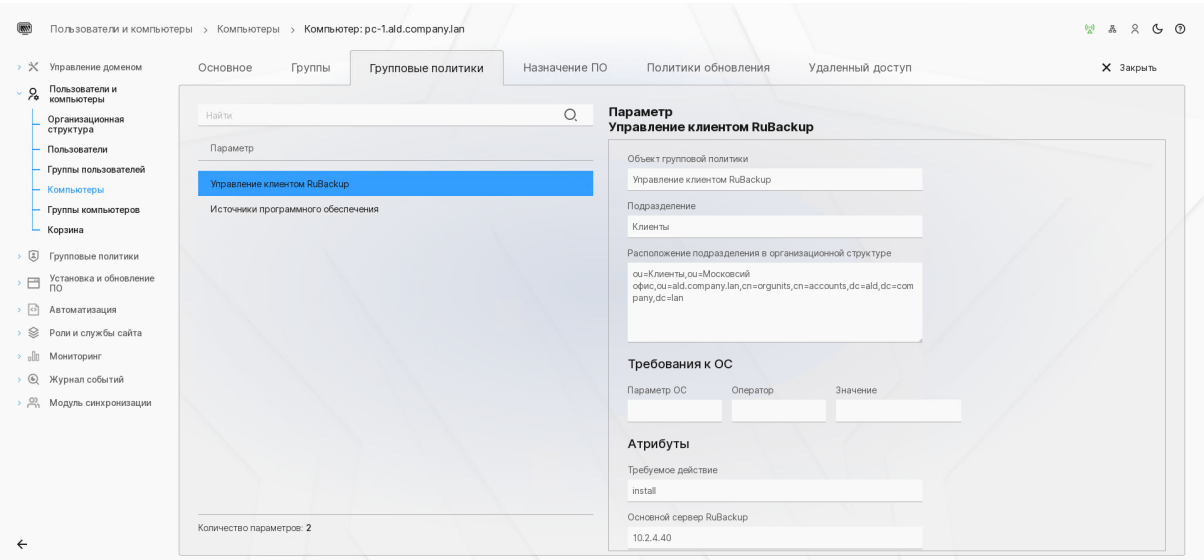


рис. 2.173 – Карточка компьютера

6. Полный путь до файла скрипта, который используются для применения этого параметра:

```
/srv/aldpro-salt/roots/states/policies/host-policies/rbta_ldap_custom_gp_host_manage_
↪rubackup_client/init.sls
```

Чтобы не ждать обновления параметров выполним на компьютере pc-1.ald.company.lan команду:

```
sudo aldpro-gpupdate --gp
```

```
systemctl status rubackup_client
```

```
admin@pc-1:~$ sudo aldpro-gpupdate --gp
Форсированное применение групповых политик
[INFO    ] Loading fresh modules for state activity
. . .
[INFO    ] Running state [run_rbta_ldap_custom_gp_host_manage_rubackup_client] at␣
↪time 12:38:17.600170
[INFO    ] Executing state salt.state for [run_rbta_ldap_custom_gp_host_manage_
↪rubackup_client]
[INFO    ] Loading fresh modules for state activity
[INFO    ] Установка клиента RuBackup
[INFO    ] Running state [rubackup-client] at time 12:38:22.952570
[INFO    ] Executing state pkg.installed for [rubackup-client]
[INFO    ] Executing command dpkg-query in directory '/root'
[INFO    ] Executing command apt-cache in directory '/root'
[INFO    ] Executing command apt-get in directory '/root'
[INFO    ] Executing command dpkg in directory '/root'
[INFO    ] Executing command systemd-run in directory '/root'
[INFO    ] Executing command dpkg-query in directory '/root'
[INFO    ] Made the following changes:
'rubackup-client' changed from 'absent' to '2.3.0.12-1'
'rubackup-common' changed from 'absent' to '2.3.0.12-1'
. . .
id__': 'rbta_ldap_custom_gp_host_manage_rubackup_client_configure_rubackup_client'}}}}
[INFO    ] Completed state [run_rbta_ldap_custom_gp_host_manage_rubackup_client] at␣
↪time 12:40:12.964257 (duration_in_ms=115364.087)
. . .
admin@pc-1:~$ systemctl status rubackup_client
o rubackup_client.service - RuBackup client
  Loaded: loaded (/etc/systemd/system/rubackup_client.service; enabled; vendor␣
↪preset: enabled)
  Active: active (running) since Sat 2024-12-28 12:40:10 MSK; 2min 7s ago
Main PID: 1897226 (rubackup_client)
   Tasks: 5 (limit: 2250)
  Memory: 46.4M
   CPU: 23.512s
  CGroup: /system.slice/rubackup_client.service
          └─1897226 /opt/rubackup/bin/rubackup_client start
```

Для удаления клиента RuBackup достаточно изменить значение атрибута *Требуемое действие* на *remove*

6.8 LAPS - Управление паролями локальных администраторов

При установке операционной системы создается учетная запись как минимум одного локального администратора, с помощью которой на этом хосте можно получить права суперпользователя, поэтому жизненный цикл локальных учетных записей на компьютерах в домене требует повышенного внимания со стороны системных администраторов. Существуют разные подходы к решению этой

задачи, например, в службе каталога ALD Pro уже был параметр групповой политики, с помощью которого можно было централизованно запретить локальным пользователям вход в операционную систему компьютеров ALSE, но теперь стало доступно еще и специализированное решение по управлению паролями локальных администраторов - Local Admins Password Solution или кратко, LAPS.

6.8.1 Механизм работы LAPS

Локальная учетная запись администратора может потребоваться в тех случаях, когда войти в операционную систему компьютера с помощью учетной записи из домена окажется невозможно. Вместе с тем, администрировать локальные учетные записи вручную представляется крайне сложной задачей, поэтому решение LAPS становится важным компонентом для обеспечения безопасности ИТ-инфраструктуры.

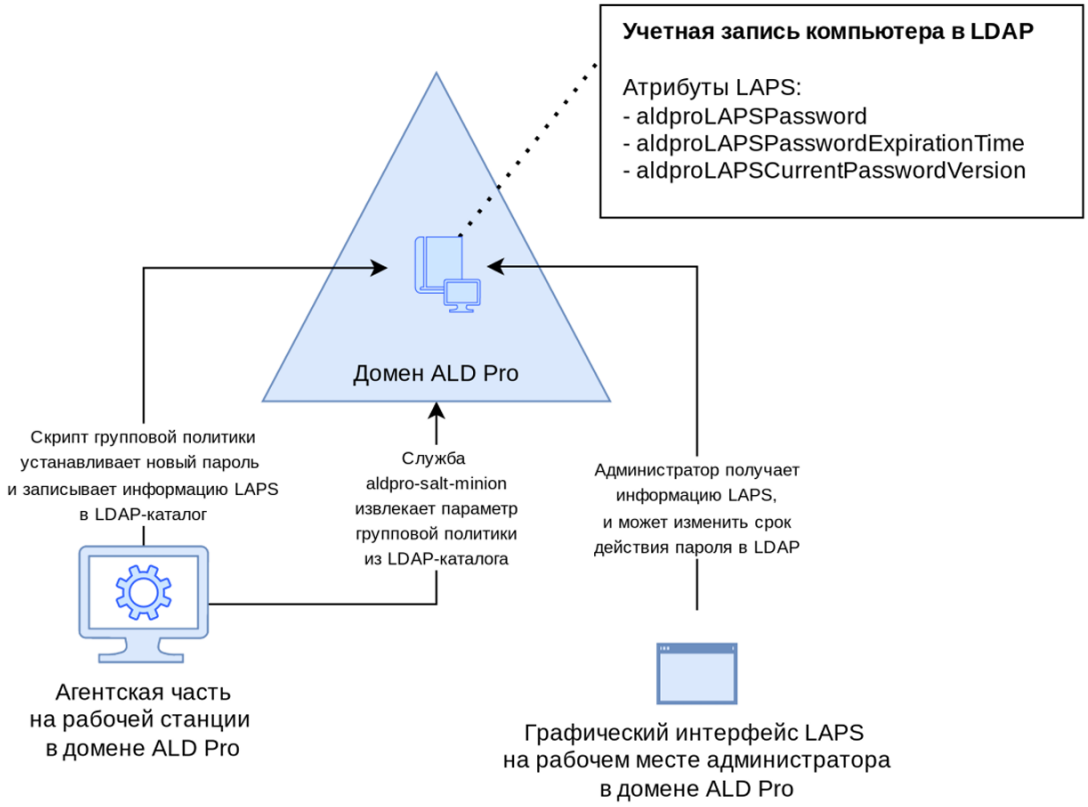


рис. 2.174 – Архитектура LAPS

Решение LAPS и состоит из трех основных компонентов:

1. **Каталог LDAP** - хранит информацию о пароле локального администратора в защищенных атрибутах учетной записи компьютера, доступ к которым имеют только участники группы admins и сами компьютеры. Решение LAPS использует следующие атри-

буты:

- Атрибут **aldproLAPSPassword** – хранит json-объект с информацией о пароле локального администратора, причем компьютер имеет право только на запись этого атрибута, а чтение ему недоступно. Объект json содержит следующие ключи:
  - **name** - имя локального администратора, которому был установлен пароль на данном компьютере;
  - **pass** - пароль, закодированный в base64;
  - **time** - время, когда этот пароль был установлен в системе в GMT;
  - **error** - необязательный ключ с текстом ошибки, которая возникла при изменении пароля.
- Атрибут **aldproLAPSPasswordExpirationTime** – хранит срок действия пароля, по истечению которого пароль должен быть обновлен.
- Атрибут **aldproLAPSCurrentPasswordVersion** – хранит случайное число длиной 16 символов, определяющее версию пароля. Это значение позволяет определить, что операционная система целевого хоста была восстановлена из резервной копии и нужно изменить пароль еще раз во внеочередном порядке, чтобы в каталоге было актуальное значение пароля.

2. **Параметр групповой политики** - определяет набор настроек и содержит salt скрипт, реализующий логику LAPS. При каждом применении групповых политик скрипт параметра выполняет на компьютере следующие действия:
   a. Извлекает из учетной записи компьютера в домене значения атрибутов **aldproLAPSPasswordExpirationTime** и **aldproLAPSCurrentPasswordVersion** с информацией о пароле локального администратора, который был сохранен в LDAP-каталоге.
   b. Извлекает из локального файла /etc/shadow-aldpro-laps.ldif значения атрибутов **admin_account_name**, **password_length**, **password_complexity** и **current_password_version** с информацией о пароле локального администратора, который был изменен в операционной системе компьютера.
   c. Проверяет список условий, по любому из которых нужно обновлять пароль:
      - Если параметр групповой политики применяется в первый раз, и ранее пароль локального администратора еще не был установлен.
      - Если в настройках групповой политики изменилось имя локального поль-

зователя, длина или сложность пароля.

- Если срок действия пароля истек или больше допустимого значения, определенного текущими настройками политики.

- Если не совпадает версия пароля в LDAP-каталоге и в локальном файле (операционная система компьютера или LDAP-каталог были восстановлены из резервной копии).

d. Изменяет пароль локальному пользователю и снимает с него блокировку, если это требуется.

e. Сохраняет информацию о новом пароле в LDAP-каталог и в локальный файл.

f. Блокирует остальных локальных пользователей системы, если это требуется.

3. **Графический интерфейс** - представляет из себя оконное приложение, которое позволяет найти информацию LAPS для конкретного компьютера по его полному доменному имени. Администратор может увидеть текущий пароль локального администратора, а также изменить срок действия этого пароля, если ему потребуется форсировать обновление пароля.

6.8.2 Установка решения

Решение распространяется в виде deb-пакета, который можно установить на любой компьютер в домене, будь то контроллер или рабочее место администратора:

```
sudo apt install ./aldpro-laps_1.0-1_amd64.deb
```

Для возможности использования решения после установки deb-пакета нужно будет расширить схему каталога и импортировать в домен дополнительный параметр групповой политики.

6.8.2.1 Импорт переменных окружения

Сначала откройте окно терминала и определите несколько переменных среды окружения в соответствии с параметрами вашего домена. Чтобы эти команды не сохранились в истории, добавьте символ пробела в начало каждой строки:

```
export SUFFIX='dc=ald,dc=company,dc=lan'
export ALDPRO_API_SERVER='dc-1.ald.company.lan'
export ALDPRO_API_CA_PATH='/etc/ipa/ca.crt'
export ALDPRO_API_LOGIN='admin'
export ALDPRO_API_PASSWORD='Pa$$w0rd'
```

6.8.2.2 Расширение схемы каталога

Теперь внесите изменения в схему каталога, для этого выполните следующую команду:

```
cat /opt/rbta/aldpro/aldpro-laps/schema/update.ldif | envsubst | ldapmodify -c -H␣
↪ldaps://ALDPRO_API_SERVER -D 'cn=Directory Manager' -w $ALDPRO_API_PASSWORD
```

- утилита **cat** считывает содержимое файла update.ldif и передает его в стандартный поток вывода;
- утилита **envsubst** подставляет значения переменных из среды окружения (в файле используется переменная SUFFIX);
- утилита **ldapmodify** вносит изменения в LDAP-каталог.

Проверить наличие изменений в схеме каталога можно командой:

```
ldapsearch -o ldif-wrap=no -H ldaps://$ALDPRO_API_SERVER -D 'cn=Directory Manager' -w
↪$ALDPRO_API_PASSWORD -b 'cn=schema' + | grep LAPS
```

```
objectClasses: ( 1.3.6.1.4.1.52616.100.2.4.3000 NAME 'aldpro-laps' DESC 'Local␣
↪administrator password object class' SUP top AUXILIARY MAY ( aldproLAPSPassword $␣
↪aldproLAPSPasswordExpirationTime $ aldproLAPSCurrentPasswordVersion ) X-ORIGIN (
↪'ALD Pro' 'user defined' ) )
attributeTypes: ( 1.3.6.1.4.1.52616.100.2.3.3000 NAME 'aldproLAPSPassword' DESC
↪'Local administrator name, password and time in json format' SYNTAX    1.3.6.1.4.1.
↪1466.115.121.1.15 SINGLE-VALUE X-ORIGIN ( 'ALD Pro' 'user defined' ) )
attributeTypes: ( 1.3.6.1.4.1.52616.100.2.3.3002 NAME
↪'aldproLAPSCurrentPasswordVersion' DESC 'Local administrator password version'␣
↪SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 SINGLE-VALUE X-ORIGIN ( 'ALD Pro' 'user defined
↪' ) )
attributeTypes: ( 1.3.6.1.4.1.52616.100.2.3.3001 NAME
↪'aldproLAPSPasswordExpirationTime' DESC 'Local administrator password expiration␣
↪time' SYNTAX 1.3.6.1.4.1.1466.115.121.1.24 SINGLE-VALUE X-ORIGIN ( 'ALD Pro' 'user␣
↪defined' ) )
```

Изменения в схему вносятся через указанный в параметре **ALDPRO_API_SERVER** контроллер домена и реплицируются на все остальные серверы автоматически. Команду можно запускать на любом компьютере в домене неограниченное количество раз. При повторном выполнении команды вы будете просто получать предупреждение **«Type or value exists»** о том, что указанные в файле инструкции доступа aci уже находятся в каталоге.

6.8.2.3 Импорт дополнительного параметра групповой политики

Следующим действием импортируйте дополнительный параметр групповой политики следующей командой:

```
cd /opt/rbta/aldpro/aldpro-laps/policy/
sh import.sh
```

Скрипт создаст в дополнительных параметрах папку LAPS и загрузит в нее параметр «Пароли локальных администраторов».

6.8.3 Настройка политики LAPS в домене

Теперь вы можете создать объект групповой политики в домене и настроить его с использованием нового параметра.

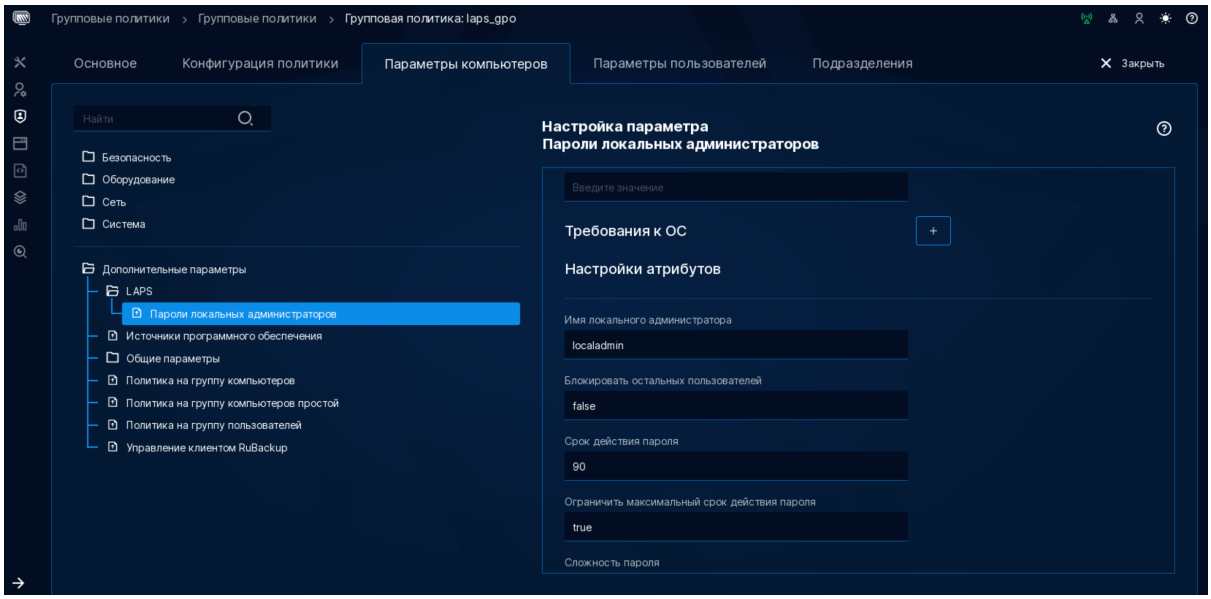


рис. 2.175 – Настройка параметра групповой политики LAPS

Параметр позволяет централизованно управлять паролями локальных администраторов путем внесения изменений в файл /etc/shadow с помощью salt-функции user.present. На сервере в LDAP-каталоге изменения сохраняются с помощью команды ipa host-mod из под учетной записи компьютера.

Атрибут **«Имя локального администратора»** определяет логин или числовой UID того пользователя, паролем которого нужно управлять. Для корректной работы параметра групповой политики значение является обязательным.

Атрибут **«Блокировать остальных»** позволяет указать, нужно ли блокировать вход по паролю для остальных локальных пользователей системы.

Допустимые значения:

- **Lock** - параметр блокирует вход по паролю для остальных локальных пользователей системы, в файле /etc/shadow к паролям пользователей будет добавляться восклицательный знак. В качестве синонима Lock может использоваться значение True.

- **Unlock** - параметр снимает блокировку входа по паролю со всех локальных пользователей системы, которым назначен пароль. Если у пользователя нет пароля (в качестве пароля записан восклицательный знак), то блокировка сниматься не будет, так как вход без пароля является небезопасным.

- **Ignore** - параметр не затрагивает остальных пользователей системы. Это поведение определено по умолчанию, поэтому в качестве синонима можно использовать False, пустую строку или любое другое значение.

Атрибут **«Срок действия пароля»** позволяет задать период, по истечению которого пароль локального администратора на компьютере будет обновлен. Значение представляет собой целое число в диапазоне от 1 до 365, указывающее количество дней. Если значение не определено, по умолчанию устанавливается срок действия пароля в 30 дней.

Атрибут **«Ограничить максимальный срок действия пароля»** позволяет запретить установку значений, превышающих текущие настройки параметра групповой политики.

Допустимые значения:

- **True** - пароль локального администратора будет автоматически обновлен, если в момент применения параметра групповой политики срок действия пароля в LDAP каталоге окажется больше того значения, которое должно быть установлено в соответствии с текущими настройками параметра групповой политики. Значение **True** используется по умолчанию.

- **False** - администраторы вправе устанавливать вручную любой срок действия пароля LAPS, превышающий текущую дату и время, и это не приведет к внеочередному обновлению пароля.

Например, если настройками параметра групповой политики определен срок действия пароля в 30 дней, а на календаре уже 1-го августа, то при включенном ограничении администратор будет не вправе установить вручную срок действия пароля на 3-е сентября или более поздние даты, т.к. в этом случае пароль будет автоматически обновлен.

Атрибут **«Сложность пароля»** позволяет задать желаемый уровень сложности нового пароля для локального администратора. Допустимые значения:

- 1 - прописные символы;

- 2 - прописные символы + строчные символы;

- 3 - прописные символы + строчные символы + цифры (этот уровень используется по умолчанию);

- 4 - прописные символы + строчные символы + цифры + специальные символы;
- 5 - предыдущий набор символов за вычетом непечатных знаков, например, I, O, Q, l, o, 0, 1, и др.

Атрибут **«Длина пароля»** позволяет задать желаемую длину нового пароля для локального администратора. Значение должно представлять собой целое число в диапазоне от 8 до 64. Если значение не определено, по умолчанию пароли генерируются длиной в 14 символов.

Для принудительного применения параметров групповых политик на целевых хостах вы можете использовать следующую команду:

```
sudo aldpro-gpupdate --gp
```

6.8.4 Просмотр информации LAPS в графическом интерфейсе

Графическая утилита LAPS предназначена для управления паролями локальных администраторов Linux-компьютеров, которые подключены к домену ALD Pro. Авторизованный пользователь может узнать учетные данные локального администратора, чтобы использовать их для устранения проблем в работе операционной системы в условиях недоступности доменных сервисов. Для работы с утилитой пользователь должен быть участником группы admins. После установки deb-пакета запустить приложение можно из меню (*Пуск → ALD Pro → LAPS*).

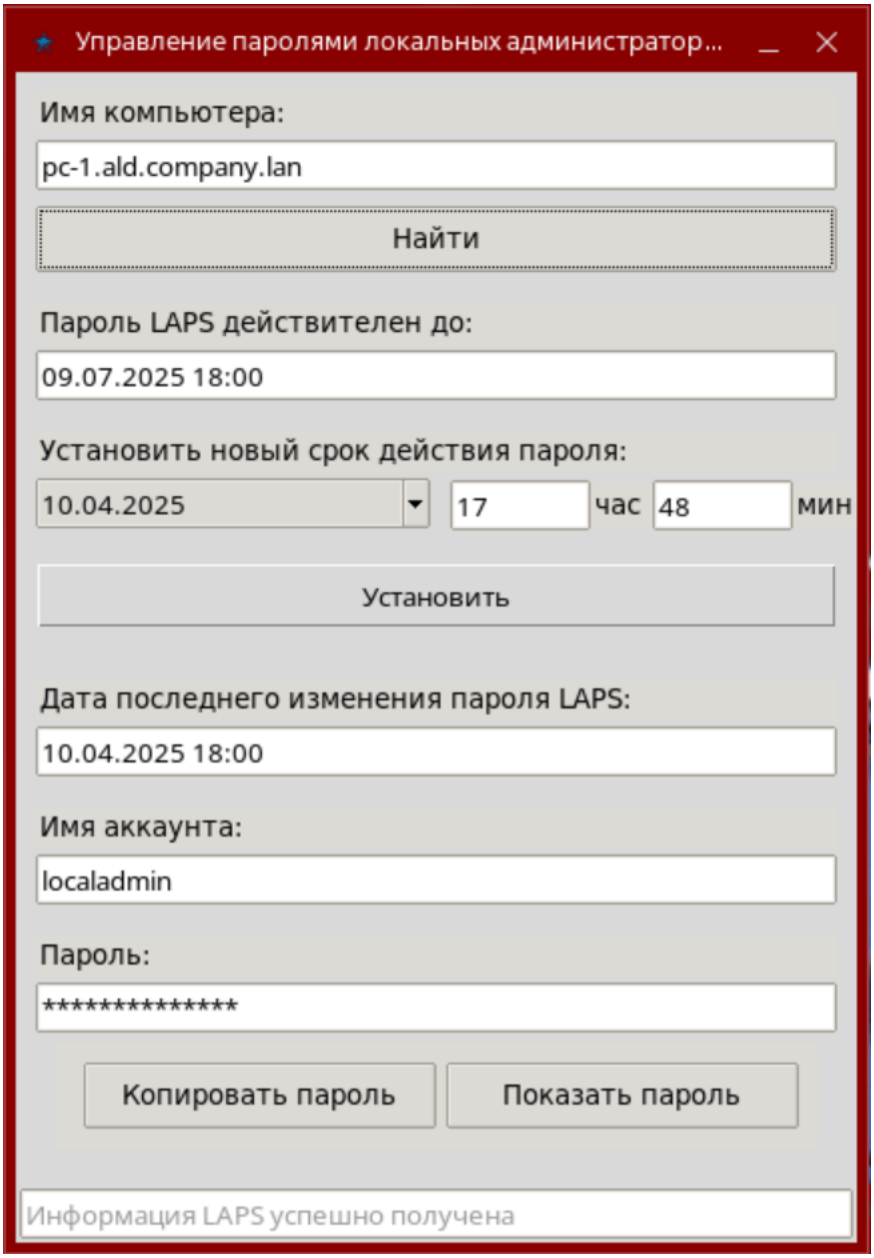


рис. 2.176 – Графический интерфейс администратора LAPS

В основном окне приложения доступны следующие элементы интерфейса:

1. Поле **«Имя компьютера»** предназначено ввода полного имени компьютера, по которому будет выполнен поиск записи в LDAP-каталоге.

2. Кнопка **«Найти»** запускает поиск учетной записи компьютера в LDAP-каталоге.

   Для выбора контроллера домена приложение использует результаты автоматического обнаружения сервисов от службы SSSD из файла /var/lib/sss/pubconf/kdcinfo.DOMAIN. Аутентификация выполняется по протоколу Kerberos V5 с использованием билетов из связки ключей Linux. Если в

связке ключей не окажется действительного TGT-билета пользователя, то по нажатию кнопки “Найти” откроется окно для ввода пароля от доменной учетной записи, из-под которой был выполнен вход в систему.

В случае неуспешного поиска в строке состояния будет выведена информация об ошибке, а кнопки «Установить», «Копировать пароль» и «Показать пароль» будут неактивны. При успешном поиске в соответствующих полях отобразится найденная информация, как указано на рис. 2.176.

3. Поле **«Пароль LAPS действителен до»** отображает текущее значение срока действия пароля локального администратора для выбранного компьютера.

4. Далее на форме представлены элементы управления для установки нового срока действия пароля. Вы можете как уменьшить это значение, так и увеличить его. Пароль будет автоматически обновлен при применении параметра групповой политики в следующих двух случаях:

- Если срок действия будет установлен меньше текущей даты и времени.
- Если срок действия будет больше того значения, которое должно быть установлено в соответствии с текущими настройками параметра групповой политики, и включено ограничение максимального срок действия пароля.

5. Кнопка **«Установить»** сохраняет новое значение срока действия пароля.

6. Поле **«Дата последнего изменения пароля LAPS»** отображает дату и время, когда пароль был установлен групповой политикой последний раз. Значение отображается в том часовом поясе, который установлен на рабочем месте администратора.

7. Поле **«Имя аккаунта»** отображает имя локального администратора, для которого был установлен пароль в системе.

8. Поле **«Пароль»** отображает текст пароля, который был установлен локальному администратору в операционной системе компьютера. По умолчанию пароль скрыт символами звездочки. Если при применении параметра групповой политики произошла ошибка, то в поле «Пароль» будет отображаться текст с уточнением причины этой ошибки.

9. Кнопка **«Копировать пароль»** позволяет скопировать скрытый пароль локального администратора в буфер обмена.

10. Кнопка **«Показать пароль»** - позволяет отключить отображение пароля на экране открытым текстом.

11. Строка состояния выводит информацию о работе приложения для упрощения отладки.

6.9 Практика и тестирование

6.9.1 Практическая работа: Модуль 6. Управление политиками ALD Pro

См. также *Требования, правила и цели выполнения практической работы*

6.9.1.1 Практические задания

Подготовка

Для корректного прохождения практических заданий необходимо выключить мандатный контроль целостности (МКЦ) на компьютере *pc-1*.

Для этого войдите на компьютер под учётной записью admin, выполните команду `sudo astra-mic-control disable` и перезагрузите машину:

```
admin@pc-1:~$ sudo astra-mic-control disable
admin@pc-1:~$ sudo reboot
```

Задание 1.

1. Создайте группу пользователей *moscow-office-employee-it* и правило автоучастия для автоматического включения в эту группу учетных записей из подразделения «Отдел ИТ». **Используйте портал управления.**

2. Создайте группу компьютеров *moscow-office-computers* и правило автоучастия для автоматического включения в эту группу учетных записей из подразделения «Московский офис». **Используйте утилиту ipa.**

3. Убедитесь, что в группы были добавлены соответствующие учетные записи.

Задание 2.

1. Удалите HBAC-правило *allow_ computers*.

2. Создайте HBAC-правило *allow_ moscow_ computers_ administration* и обеспечьте доступ ко всем службам для группы *moscow-office-employee-it* на компьютерах из группы *moscow-office-computers*. **Используйте портал управления.**

3. Используя HBAC-правило *allow_moscow_computers_sudo*, обеспечьте доступ к службе *sudo* для пользователя **ppetrov** на компьютерах из группы *moscow-office-computers*. **Используйте утилиту ipa.**

4. Убедитесь, что созданные HBAC-правила работают корректно.

Задание 3.

1. Создайте правило sudo *allow_all_moscow_computers*, позволяющее участникам группы *moscow-office-employee-it* выполнять любые команды с привилегиями суперпользователя на компьютерах из группы *moscow-office-computers*. **Используйте портал управления.**

2. Создайте правило sudo *allow_gpupdate_gp_moscow_computers*, позволяющее участникам группы *moscow-office-employees* выполнять команду `aldpro-salt-call state.apply gpupdate.gp -c /srv/aldpro-salt/config/` с привилегиями суперпользователя на компьютерах из группы *moscow-office-computers*. Используйте утилиту ipa.

3. Убедитесь, что созданные правила SUDO работают корректно.

Задание 4.

1. Создайте групповую политику «Переменная location», которая создает переменную окружения пользователя *LOCATION* и присваивает ей значение *Moscow*. Политика должна применяться к подразделению «Московский офис».

2. Создайте групповую политику «Ограничение прав на выключение компьютера», ограничивающую возможность как локального, так и удаленного выключения компьютера только от имени суперпользователя. Политика должна применяться к подразделению «Московский офис».

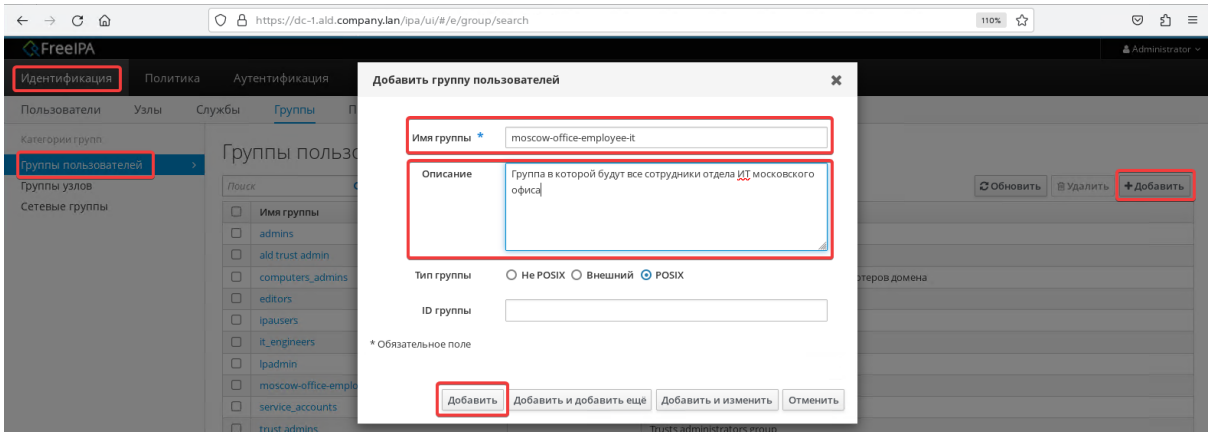
3. Убедитесь, что созданные групповые политики работают корректно.

6.9.1.2 Практическая работа (с ответами)

Задание 1.

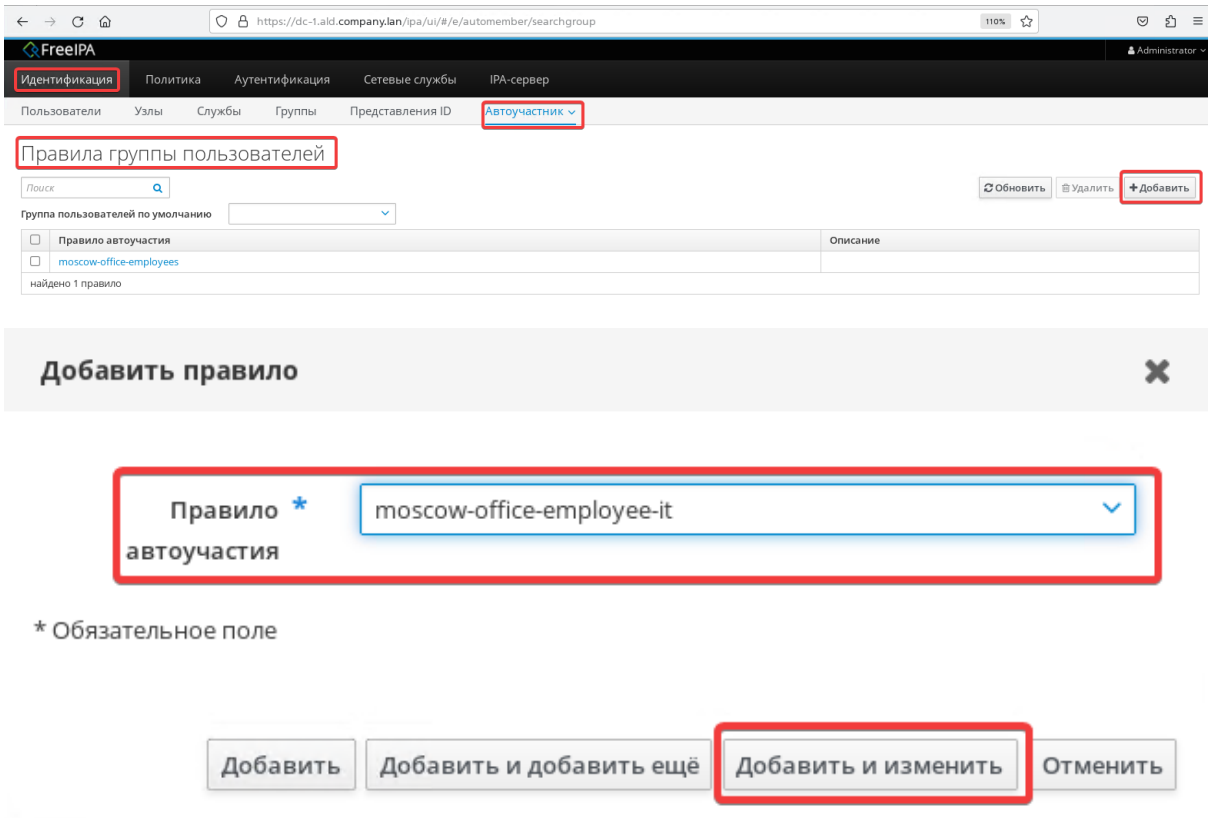
1. Создайте группу пользователей *moscow-office-employee-it* и правило автоучастия для автоматического включения в эту группу учетных записей из подразделения «Отдел ИТ». **Используйте портал управления.**

- В интерфейсе FreeIPA создадим группу пользователей *moscow-office-employee-it*.

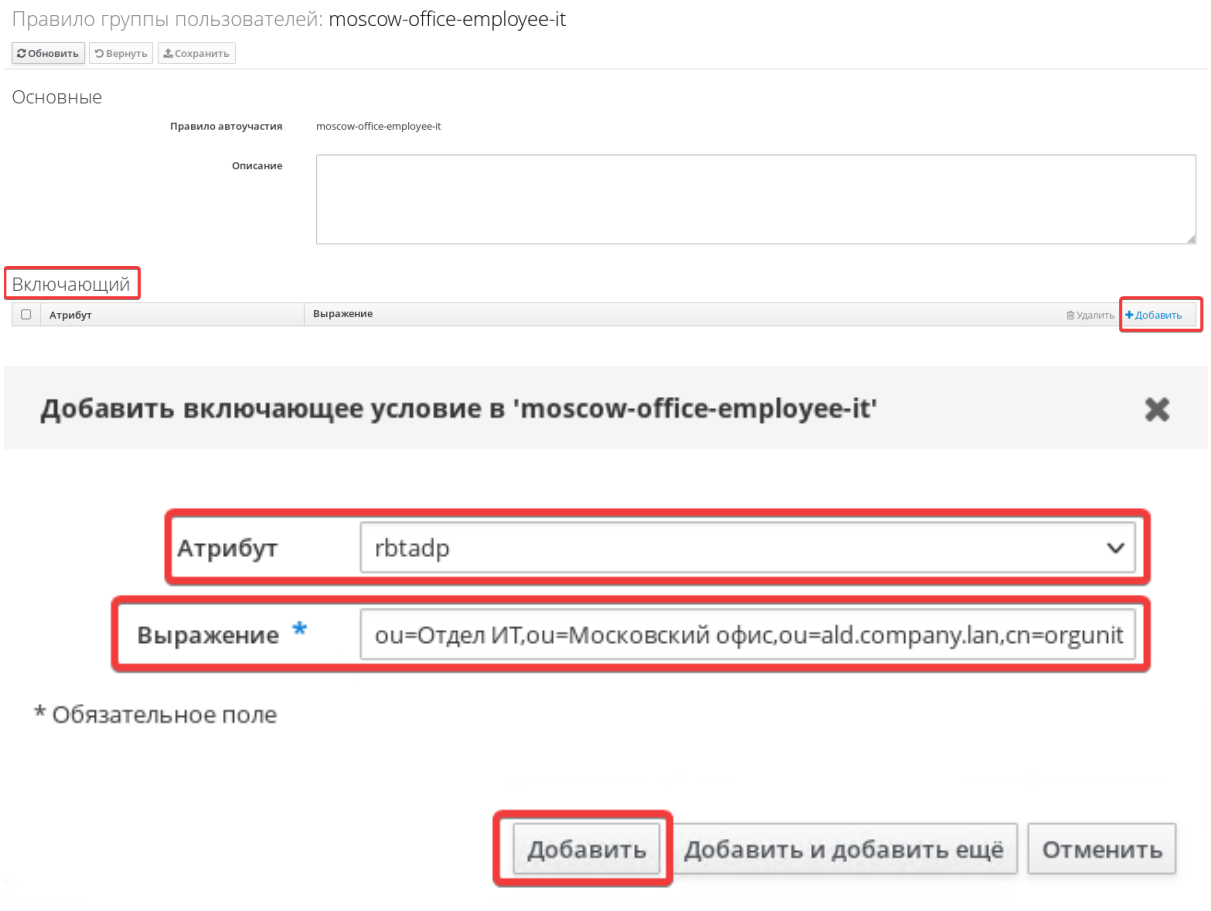


По желанию можно добавить описание «Группа, в которой будут все сотрудники отдела ИТ московского офиса».

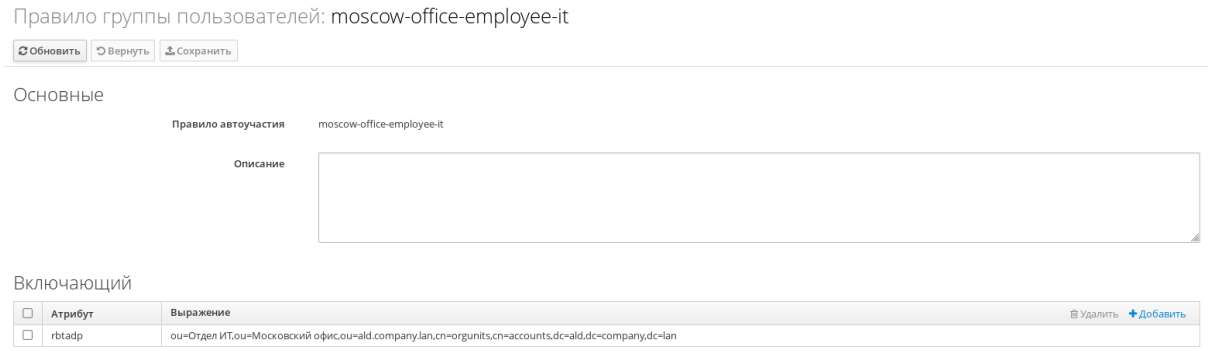
- В интерфейсе FreeIPA создадим правило автоучастия для группы пользователей *moscow-office-employee-it*.



- Добавим в правило включающий атрибут **rbtadp** с выражением «ou=Отдел ИТ,ou=Московский офис,ou=ald.company.lan, cn=orgunits,cn=accounts,dc=ald,dc=company,dc=lan».



- Результат должен быть таким:



2. Создайте группу компьютеров *moscow-office-computers* и правило автоучастия для автоматического включения в эту группу учетных записей из подразделения «Московский офис». **Используйте утилиту ipa.**

- Создадим группу:

```
admin@dc-1:~$ ipa hostgroup-add moscow-office-computers
admin@dc-1:~$ ipa hostgroup-mod moscow-office-computers \
--setattr="rbtadp=ou=Московский офис,ou=ald.company.lan,cn=orgunits,cn=accounts,
↪dc=ald,dc=company,dc=lan"
admin@dc-1:~$ ipa hostgroup-mod moscow-office-computers --desc='Группа, в которой␣
↪будут все компьютеры московского офиса'
```

- Создадим правило автоучастия:

```
admin@dc-1:~$ ipa automember-add moscow-office-computers --type=hostgroup
admin@dc-1:~$ ipa automember-mod moscow-office-computers --type=hostgroup --desc=
↪'Правило автоучастия для наполнения группы компьютеров московского офиса'
admin@dc-1:~$ ipa automember-add-condition moscow-office-computers --type=hostgroup --
↪key=rbtadp --inclusive-regex='ou=Московский офис,ou=ald.company.lan,cn=orgunits,
↪cn=accounts,dc=ald,dc=company,dc=lan'
```

3. Убедитесь, что в группы были добавлены соответствующие учетные записи.

- Принудительно обновим состав автоучастников групп пользователей и хостов:

```
admin@dc-1:~$ ipa automember-rebuild --type=group
admin@dc-1:~$ ipa automember-rebuild --type=hostgroup
```

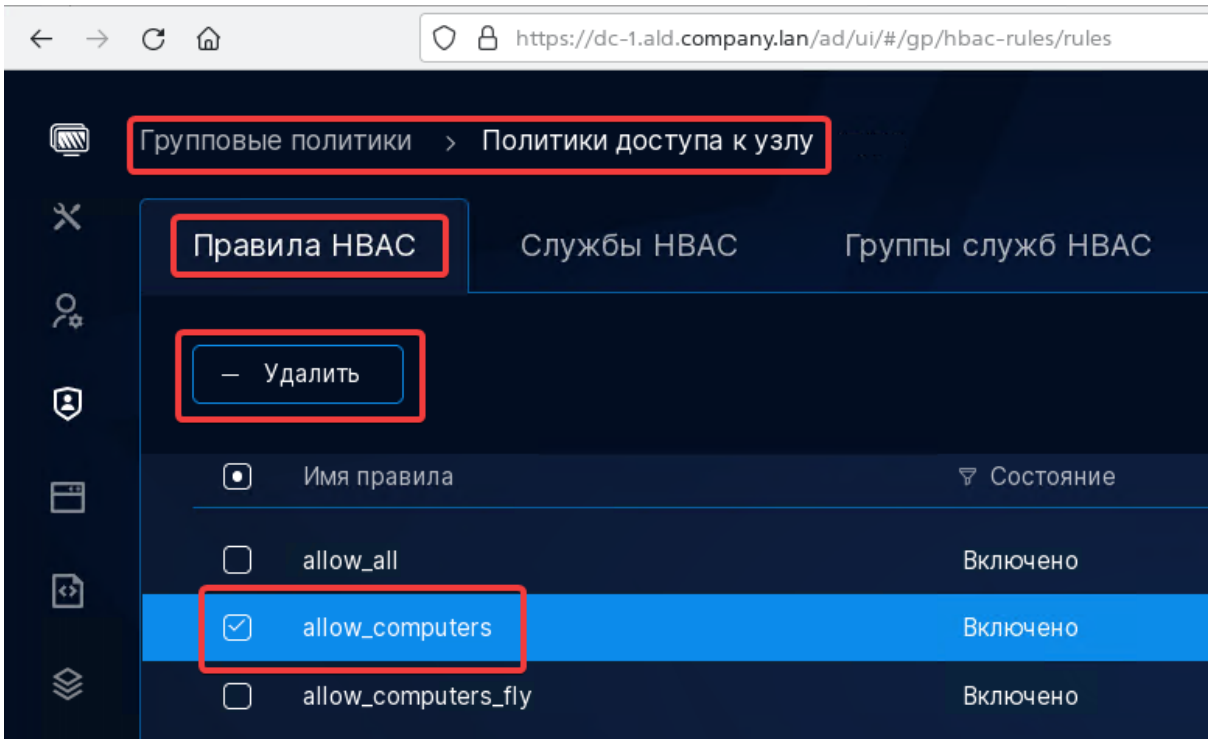
- Проверим состав групп *moscow-office-employee-it* и *moscow-office-computers*:

```
admin@dc-1:~$ ipa group-show Moscow-office-employee-it | grep участники
 Пользователи-участники: iivanov
```

```
admin@dc-1:~$ ipa hostgroup-show moscow-office-computers | grep участники
 Узлы-участники: pc-1.ald.company.lan
```

Задание 2.

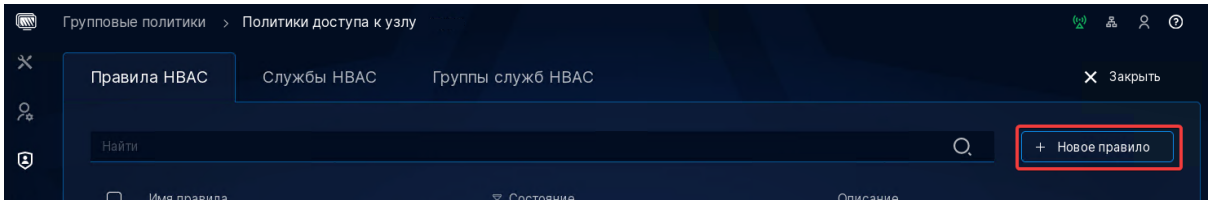
1. Удалите HBAC-правило *allow_ computers*. Перейдем в раздел портала управления «Политики доступа к узлу» и удалим правило.

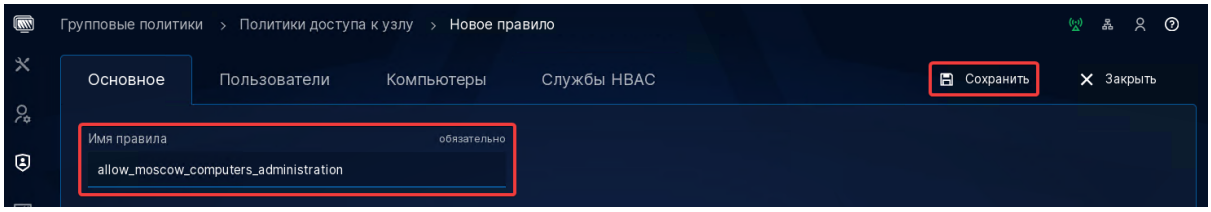


2. Создайте HBAC-правило *allow_ moscow_ computers_ administration*, обеспечьте доступ ко всем службам для группы *moscow-office-employee-it* на компьютерах в группе *moscow-office-computers*.

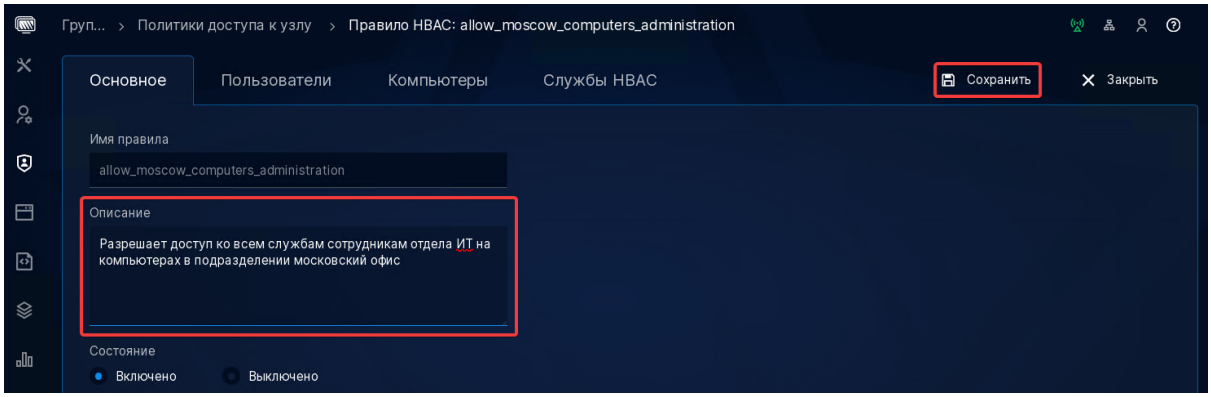
Используйте портал управления.

- Создадим новое правило и назовем его *allow_ moscow_ computers_ administration*.

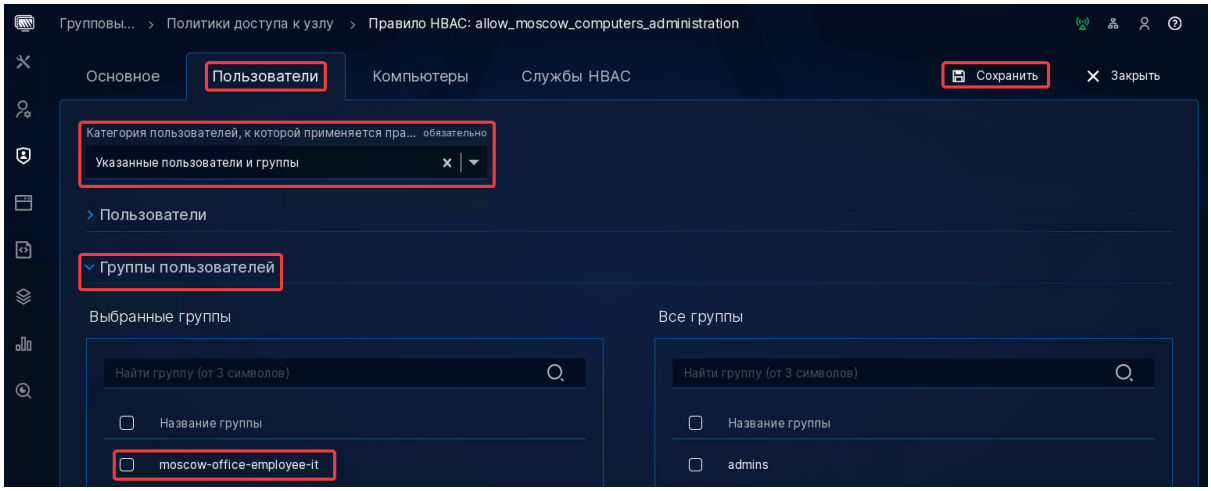




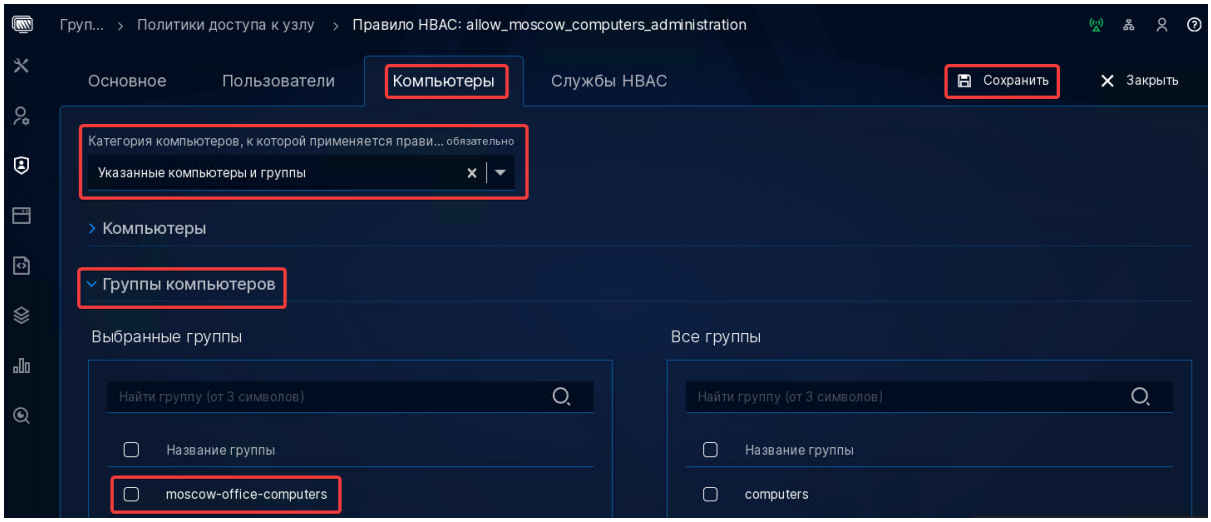
По желанию можно заполнить поле *Описание* «Разрешает доступ ко всем службам сотрудникам отдела ИТ на компьютерах в подразделении московский офис».



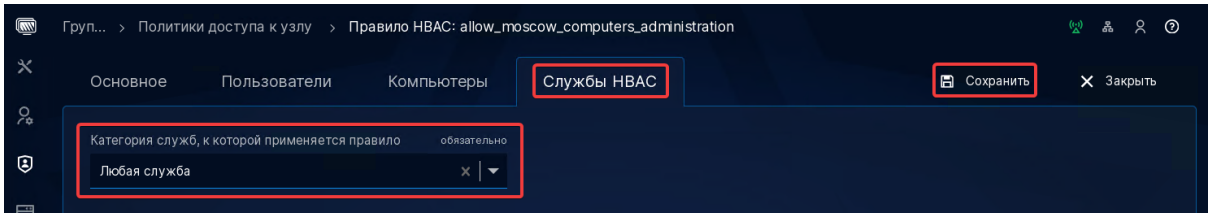
- Применим правило к группе пользователей *moscow-office-employee-it*.



- Применим правило к группе компьютеров *moscow-office-computers*.



- Применим правило к любой службе.



3. Используя HBAC-правило *allow_moscow_computers_sudo*, обеспечьте доступ к службе *sudo* для пользователя **ppetrov** на компьютерах в группе *moscow-office-computers*.

Используйте утилиту ipa.

```
admin@dc-1:~$ ipa hbacrule-add allow_moscow_computers_sudo # создание политики HBAC
admin@dc-1:~$ ipa hbacrule-mod allow_moscow_computers_sudo --desc='Разрешает доступ к␣
↪службе sudo пользователю ppetrov на компьютерах группы moscow_office_computers' #␣
↪описание
admin@dc-1:~$ ipa hbacrule-add-user allow_moscow_computers_sudo --users=ppetrov #␣
↪установить область применения на пользователя ppetrov
```

(continues on next page)

(продолжение с предыдущей страницы)

```
admin@dc-1:~$ ipa hbacrule-add-host allow_moscow_computers_sudo --hostgroups moscow-
↪office-computers # установить область применения на группу хостов moscow-office-
↪computers
admin@dc-1:~$ ipa hbacrule-add-service allow_moscow_computers_sudo --hbacsvcs=sudo #␣
↪установить область применения на службу sudo
admin@dc-1:~$ ipa hbacrule-show allow_moscow_computers_sudo
```

4. Убедитесь, что созданные HBAC-правила работают корректно.

```
admin@dc-1:~$ ipa hbactest --user=iivanov --host=pc-1 --service=login
-------------------------
Доступ предоставлен: True
-------------------------
 Соответствующие правила: allow_moscow_computers_administration
 Несоответствующие правила: allow_all
 Несоответствующие правила: allow_computers_fly
 Несоответствующие правила: allow_moscow_computers_sudo
 Несоответствующие правила: allow_systemd-user
```

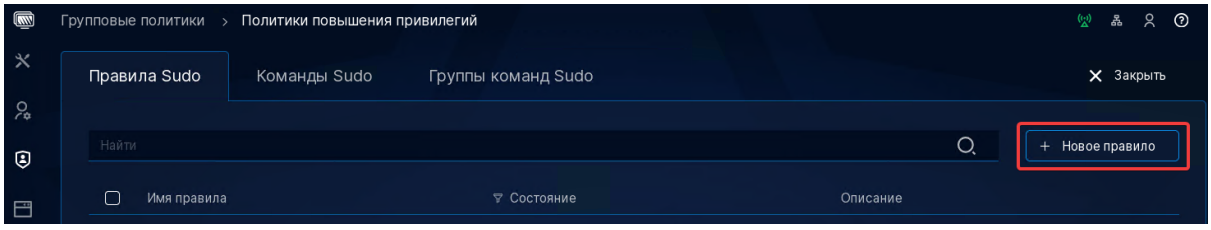
```
admin@dc-1:~$ ipa hbactest --user=ppetrov --host=pc-1 --service=sudo
-------------------------
Доступ предоставлен: True
-------------------------
 Соответствующие правила: allow_moscow_computers_sudo
 Несоответствующие правила: allow_all
 Несоответствующие правила: allow_computers_fly
 Несоответствующие правила: allow_moscow_computers_administration
 Несоответствующие правила: allow_systemd-user
```

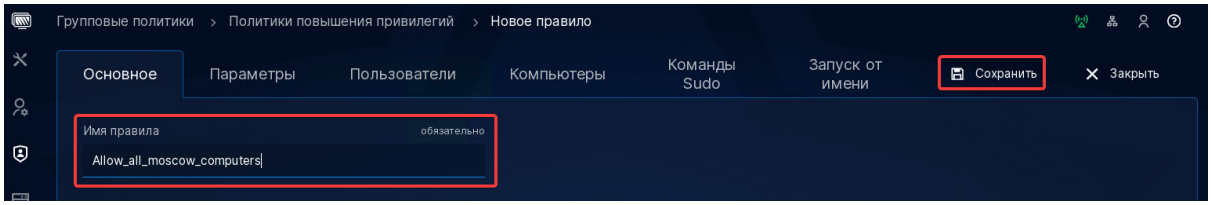
Задание 3.

1. Создайте правило sudo *Allow_all_moscow_computers*, позволяющее участникам группы *moscow-office-employee-it* выполнять любые команды с привилегиями локального суперпользователя на компьютерах группы *moscow-office-computers*.

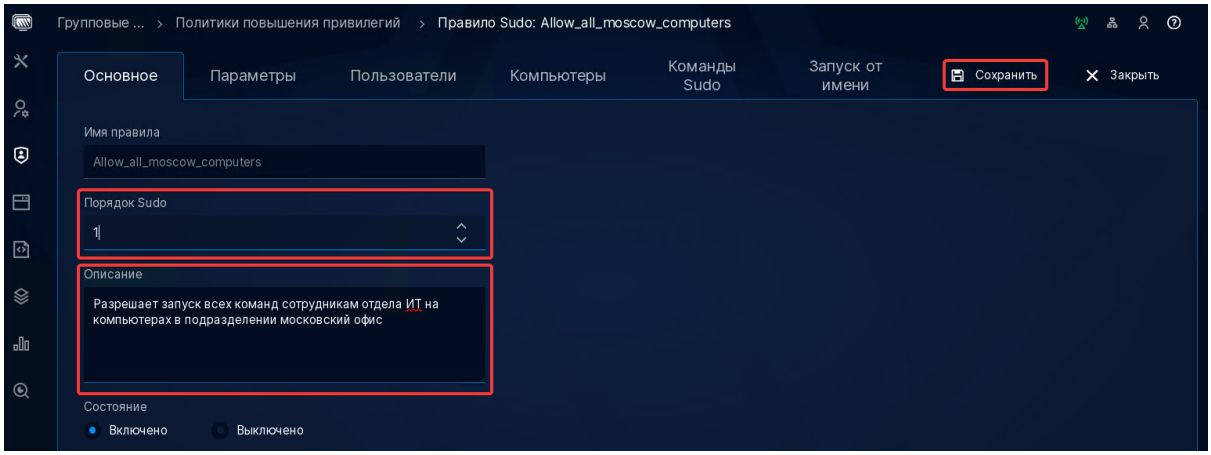
Используйте портал управления.

- Создадим новое правило sudo *Allow_all_moscow_computers* с порядком «1».

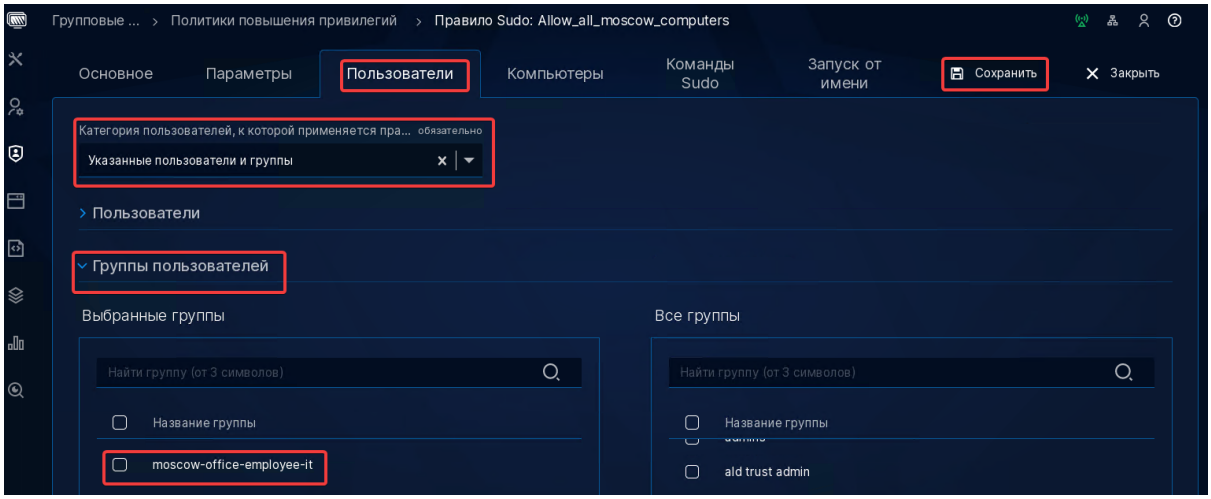




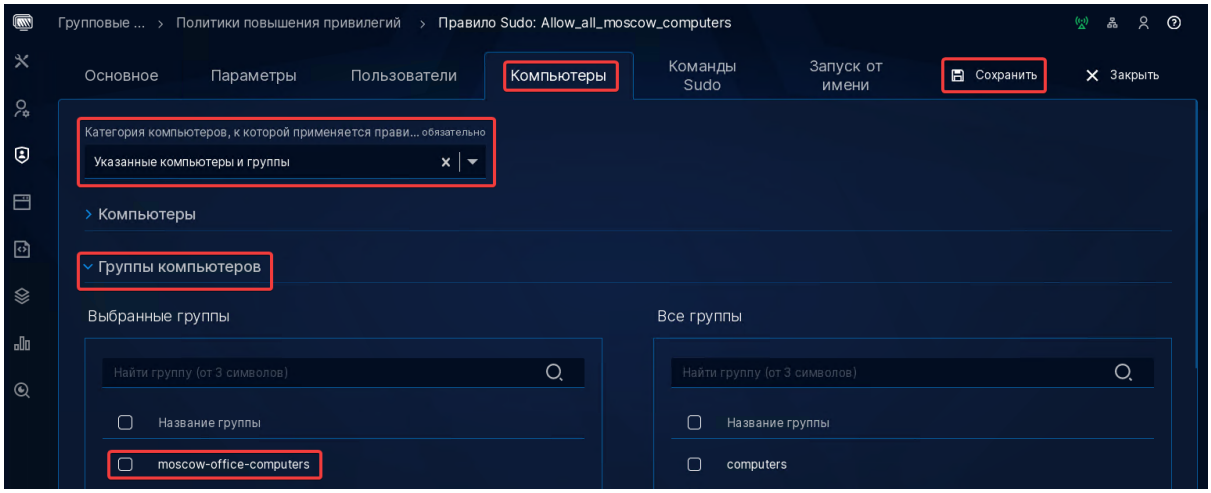
По желанию можно заполнить поле *Описание* «Разрешает запуск всех команд сотрудникам отдела ИТ на компьютерах в подразделении московский офис».



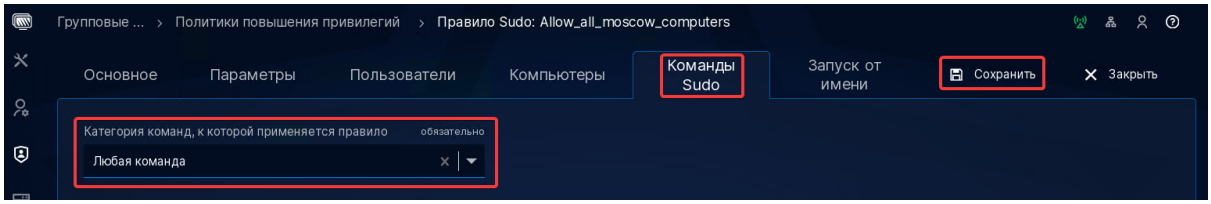
- Применим правило к группе пользователей *moscow-office-employee-it*.



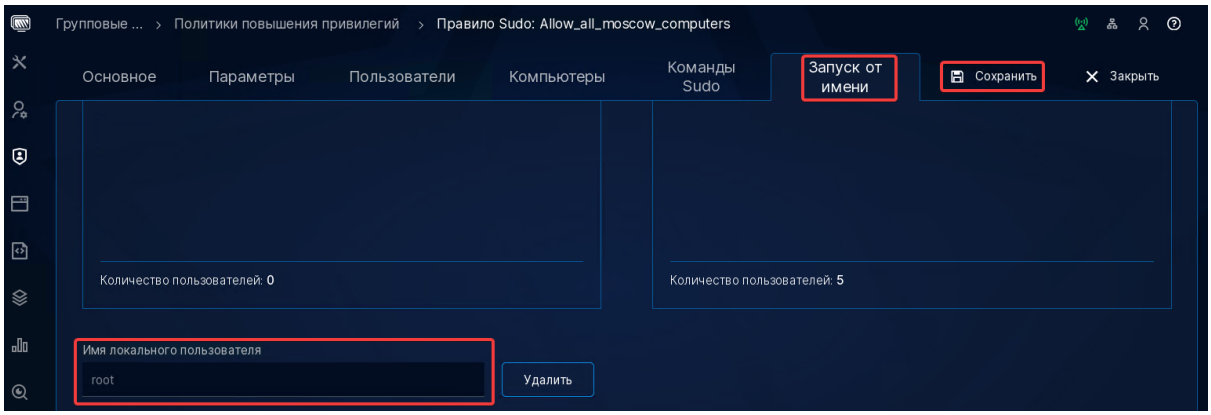
- Применим правило к группе компьютеров *moscow-office-computers*.



- Разрешим запуск любой команды.



- Разрешим запуск от имени локального суперпользователя.



2. Создайте правило sudo *Allow_gpupdate_gp_moscow_computers*, позволяющее участникам группы *moscow-office-employees* выполнять команду `aldpro-salt-call state.apply gpupdate.gp -c /srv/aldpro-salt/config/` с повышенными привилегиями на компьютерах группы *moscow-office-computers*.

Используйте утилиту ipa.

- Создадим команду sudo:

```
$ ipa sudocmd-add 'aldpro-salt-call state.apply gpupdate.gp -c /srv/aldpro-salt/
↪config/' --desc='gpupdate.gp'
```

- Создадим правило sudo:

```
$ ipa sudorule-add 'Allow_gpupdate_gp_moscow_computers' # Создадим правило sudo
$ ipa sudorule-mod 'Allow_gpupdate_gp_moscow_computers' --desc "Разрешает gpupdate.gp␣
↪сотрудникам московского офиса на компьютерах в подразделении московский офис" #␣
↪Описание
$ ipa sudorule-add-user 'Allow_gpupdate_gp_moscow_computers' --groups moscow-office-
↪employees # Применим правило к группе пользователей moscow-office-employees
$ ipa sudorule-add-host 'Allow_gpupdate_gp_moscow_computers' --hostgroups moscow-
↪office-computers # Применим правило к группе компьютеров moscow-office-computers
$ ipa sudorule-add-allow-command 'Allow_gpupdate_gp_moscow_computers' --sudocmds=
↪"aldpro-salt-call
state.apply gpupdate.gp -c /srv/aldpro-salt/config/" # Применим правило к команде
↪"aldpro-salt-call
state.apply gpupdate.gp -c /srv/aldpro-salt/config/"
$ ipa sudorule-show 'Allow_gpupdate_gp_moscow_computers' # Вывести правило и его␣
↪параметры
```

1. Убедимся, что созданные правила SUDO работают корректно:

- Войдем на pc-1 под учетной записью admin.

- Очистим кэш и перезапустим службу sssd.

```
admin@pc-1:~$ sudo rm -rf /var/lib/sss/db/*
admin@pc-1:~$ sudo systemctl restart sssd
```

- Убедимся, что к пользователю iivanov применяется политика *Allow_ all_ moscow_ computers*:

```
admin@pc-1:~$ sudo -l -U iivanov
Matching Defaults entries for iivanov on pc-1:
   env_reset, mail_badpass,
   secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin, !
↪lecture,
   secure_path=/usr/lib/parsec/bin\:/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/
↪bin\:/sbin\
   :/b
   in

User iivanov may run the following commands on pc-1:
   (root) salt-call state.apply gpupdate.gp -c /srv/salt/standalone/config
   (root) ALL
```

- Как мы можем убедиться, к пользователю применилась политика *Allow_ gpupdate_ gp_ moscow_ computers* и *Allow_ all_ moscow_ computers*.

- Как видим, пользователь iivanov может выполнять любую команду на pc-1 от имени суперпользователя.

- Убедимся, что пользователь iivanov может использовать службу login. Выполним в терминале:

```
admin@pc-1:~$ sudo login iivanov
```

- Убедимся, что пользователь iivanov может выполнить команду с правами суперпользователя:

```
iivanov@pc-1:~$ sudo ls
Пароль:
Desktop  Desktops  SystemWallpapers  Видео  Документы  Загрузки
Изображения  Музыка  Общедоступные  Шаблоны
```

- Убедимся, что пользователь ppetrov не может использовать службу login. Выполним в терминале:

```
iivanov@pc-1:~$ sudo login ppetrov
Пароль:
Доступ запрещен
```

- Убедимся, что к пользователю ppetrov применяется политика *Allow_ gpupdate_ gp_ moscow_ computers*

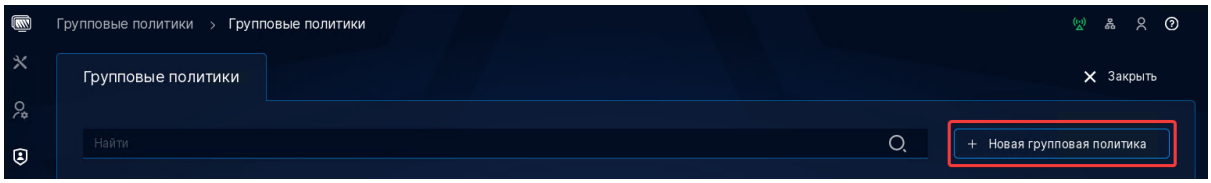
```
iivanov@pc-1:~$ sudo -l -U ppetrov
Matching Defaults entries for ppetrov on pc-1:
   env_reset, mail_badpass,
secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin, !
↪lecture,
  secure_path=/usr/lib/parsec/bin\:/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/
↪bin\:/sbin\:
/bin

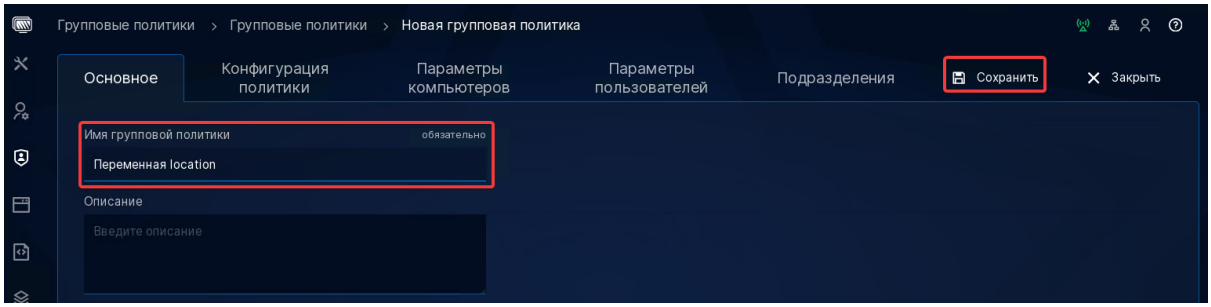
User ppetrov may run the following commands on pc-1:
  (root) salt-call state.apply gpupdate.gp -c /srv/salt/standalone/config
```

Задание 4.

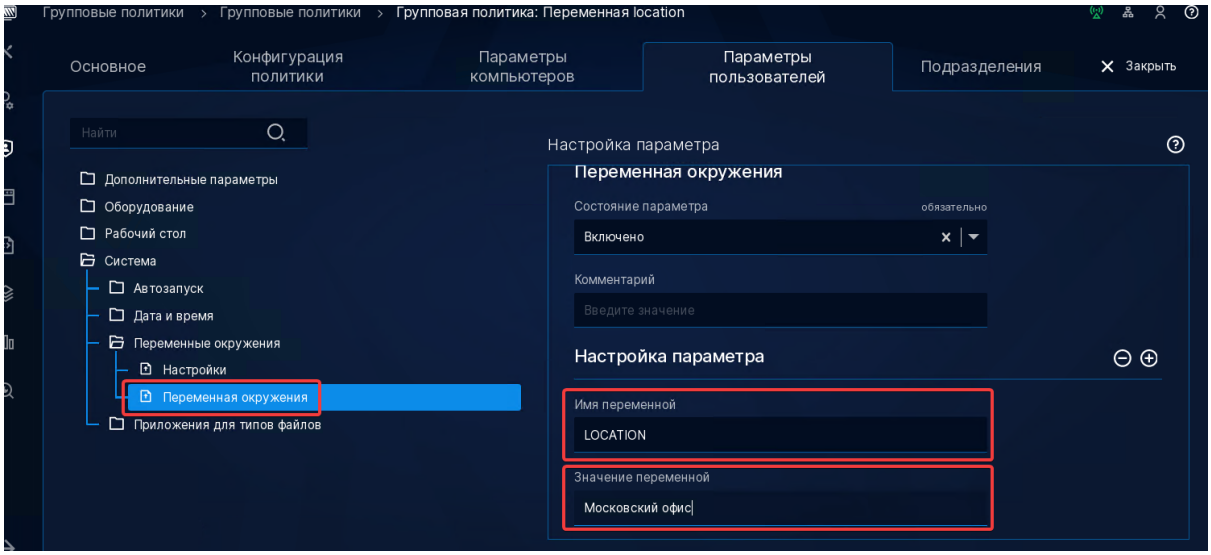
1. Создайте групповую политику «Переменная location», которая создает переменную окружения пользователя *LOCATION* и присваивает ей значение *Moscow*. Политика должна применяться к подразделению «Московский офис».

- На портале управления перейдем в раздел «Групповые политики» и создадим новую групповую политику «Переменная location».

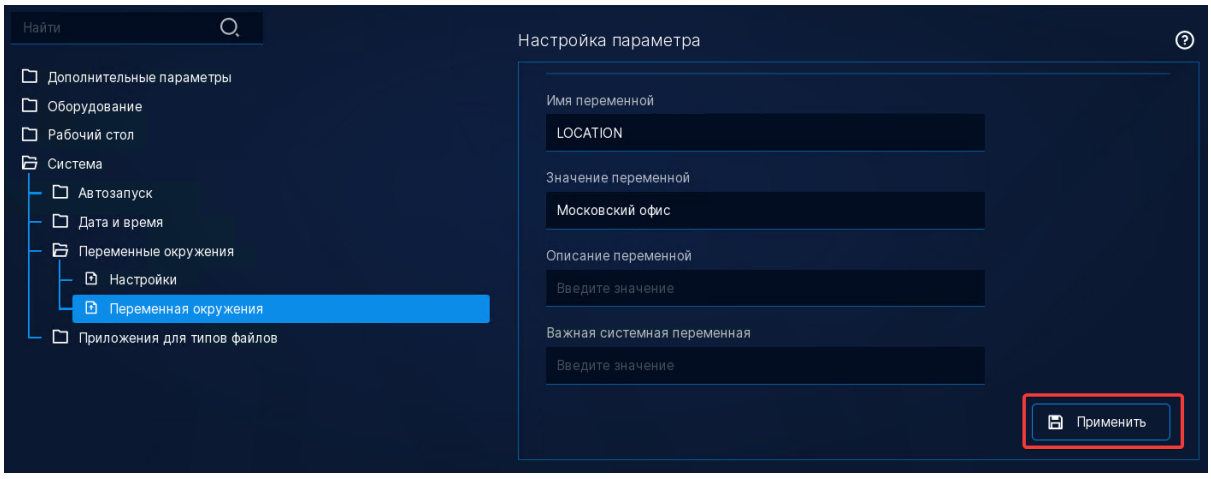




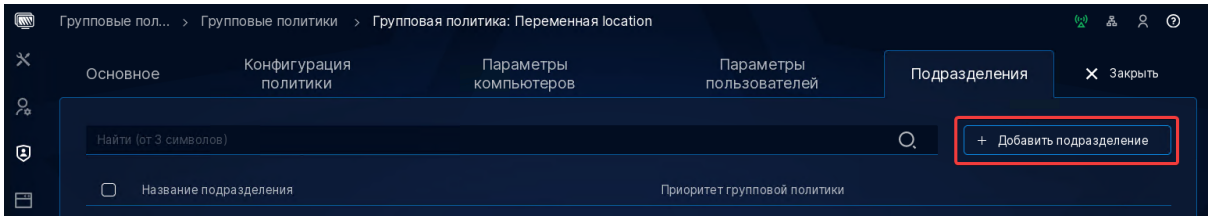
- Добавим параметр пользователя «Переменная окружения»:
  - Имя: LOCATION
  - Значение: Московский офис

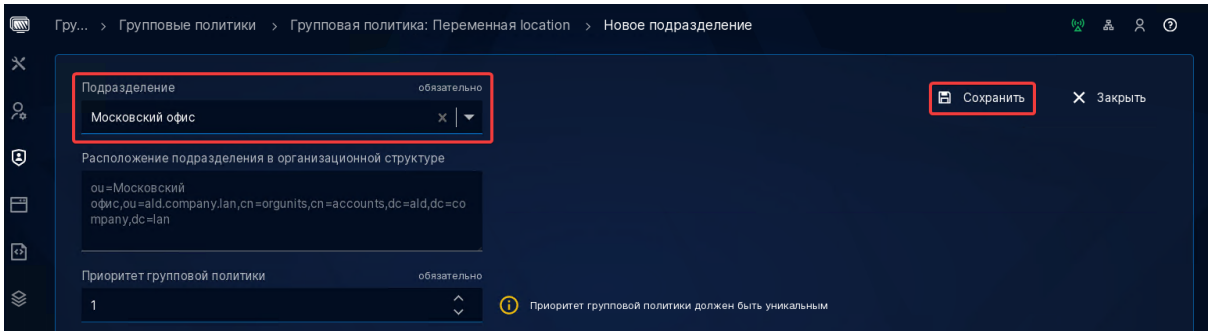


Не забудьте нажать кнопку *Применить*.



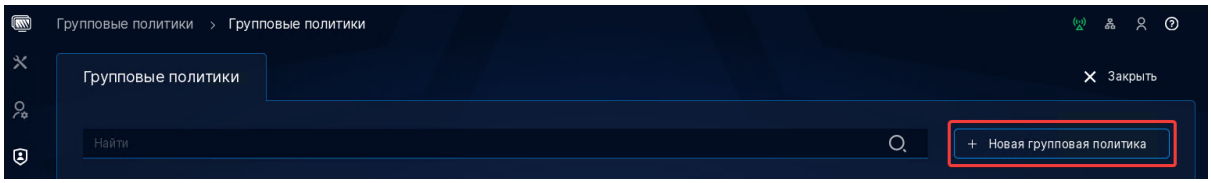
- Примените политику на подразделение «Московский офис».

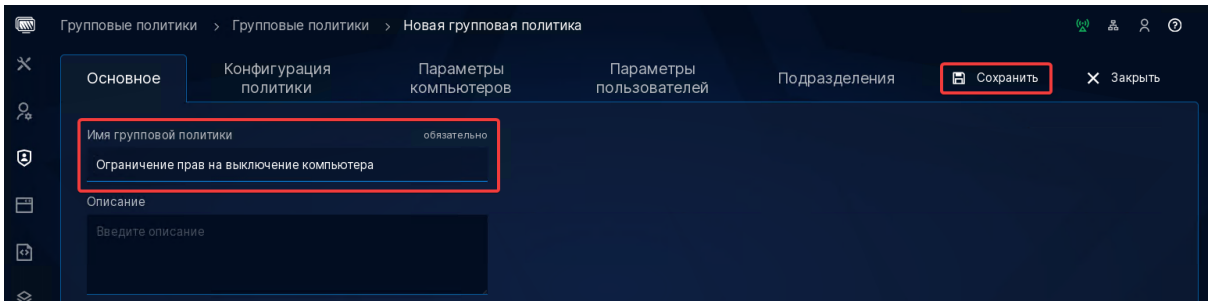




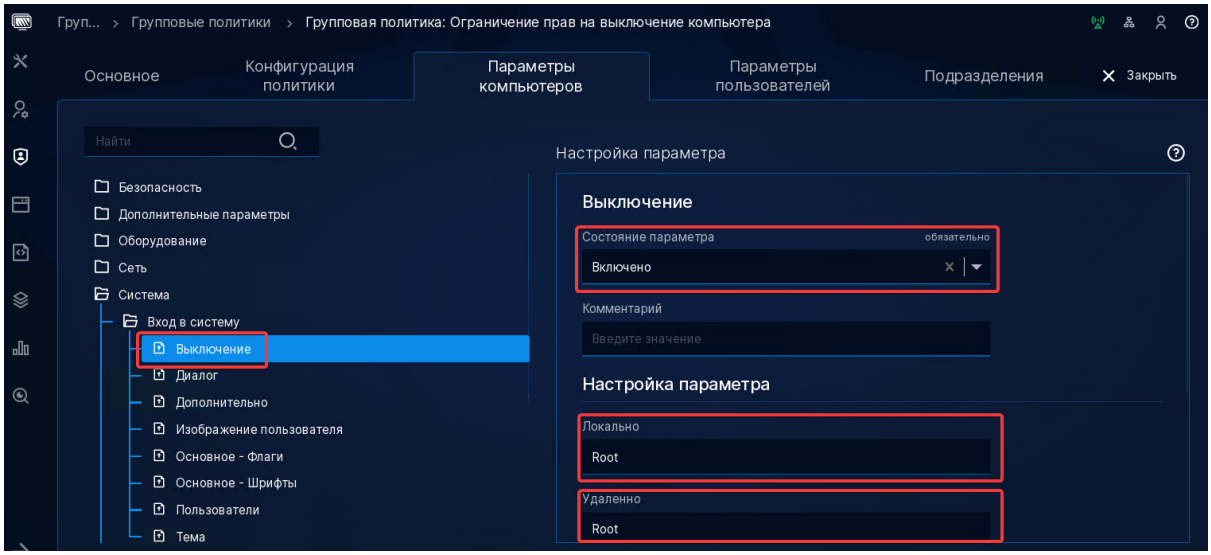
2. Создайте групповую политику «Ограничение прав на выключение компьютера», ограничивающую возможность как локального, так и удаленного выключения компьютера только от имени суперпользователя. Политика должна применяться к подразделению «Московский офис».

- На портале управления перейдем в раздел «Групповые политики» и создадим новую групповую политику «Ограничение прав на выключение компьютера».

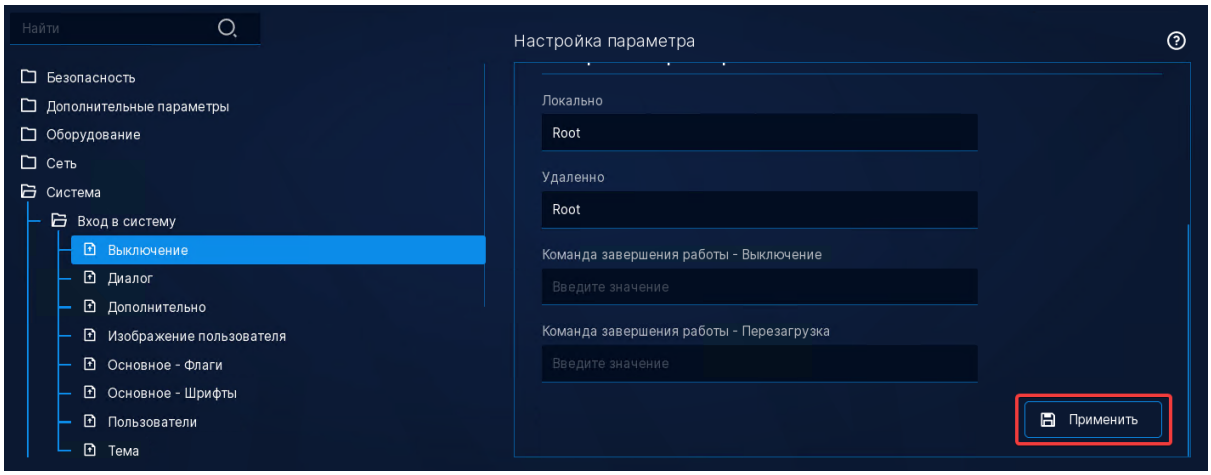




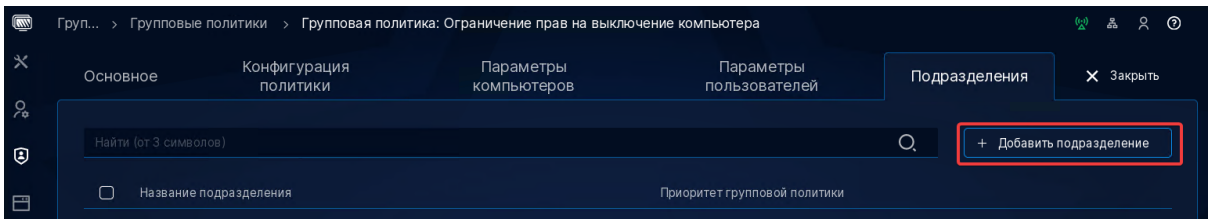
- Добавим параметр компьютера *Выключение*:
  - Локально: Root
  - Удаленно: Root

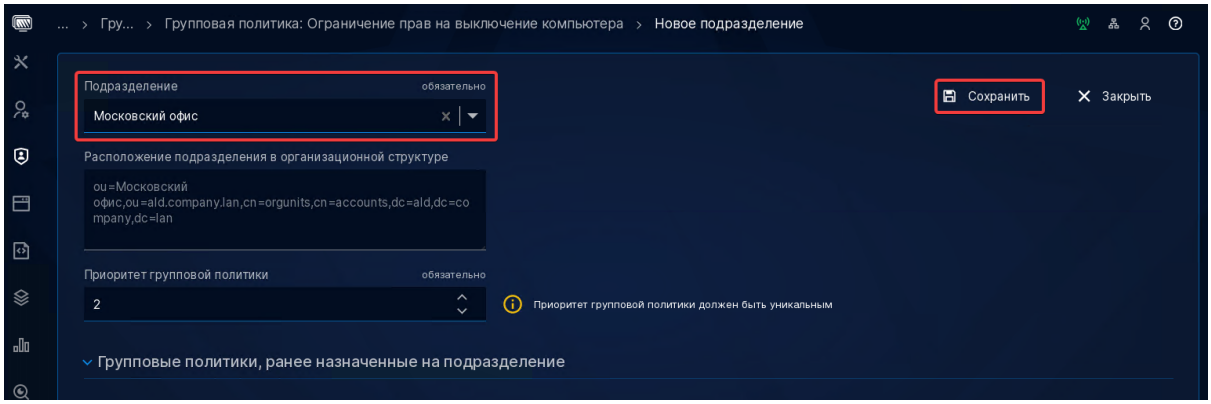


Не забудьте нажать кнопку *Применить*.



- Применим политику на подразделение «Московский офис».





3. Убедитесь, что созданные групповые политики работают корректно.

- Войдем на pc-1 под учетной записью admin.
- Выполним принудительное применение политик с переформированием pillar от имени пользователя admin:

```
$ sudo aldpro-gpupdate --gp --verbose
```

- Выполним вход в систему под пользователем iivanov.
- Проверим содержимое файлов с pillar:

```
iivanov@pc-1:~$ sudo cat /opt/rbta/aldpro-salt/minion/gp-user_pillar.json
[sudo] пароль для iivanov:
{"aldpro-users": {"iivanov": {"rbta_ldap_env_vars_u": {"rbta_ldap_env_vars_u__
↪variables": [{"rbta_ldap_env_vars_u__variables__name": "LOCATION", "rbta_ldap_env_
↪vars_u__variables__value": "\u041c\u043e\u0441\u043a\u043e\u0432\u0441\u043a\u0438\
↪u0439 \u043e\u0444\u0438\u0441", "rbta_ldap_env_vars_u__variables__description": "",
↪ "rbta_ldap_env_vars_u__variables__important": ""}]}}}, "aldpro-hosts": {}}iivanov
```

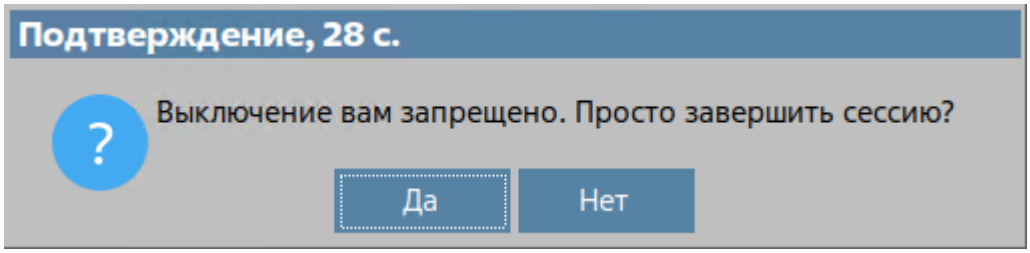
sudo cat /opt/rbta/aldpro/standalone/gp-host_pillar.json

```
iivanov@pc-1:~$ sudo cat /opt/rbta/aldpro-salt/minion/gp-host_pillar.json
{"aldpro-users": {}, "aldpro-hosts": {"pc-1.ald.company.lan": {"rbta_ldap_security_
↪policy": {"rbta_ldap_security_policy__local_login_policy": {"rbta_ldap_security_
↪policy__local_login_policy__policy": "all"}}, "rbta_ldap_login": {"rbta_ldap_login__
↪shutdown": {"rbta_ldap_login__shutdown__local": "root", "rbta_ldap_login__shutdown__
↪remote": "root"}}}}}
```

- Как видим, наши политики применились. Проверим на практике.
- Проверим переменную среды LOCATION:

```
iivanov@pc-1:~$ env | grep LOCATION
LOCATION=Московский офис
```

- Попробуем выключить компьютер. Будет выведено сообщение о запрете.



- Попробуем перезагрузить компьютер. Будет выведено то же сообщение о запрете.
- Попробуем перезагрузить компьютер через командную строку от имени суперпользователя:

```
$ sudo reboot
```

- Как видим, с правами суперпользователя мы смогли перезагрузить компьютер.

6.9.2 Тест: Модуль 6. Управление политиками ALD Pro

Пожалуйста, ответьте на несколько вопросов, чтобы закрепить полученные знания. Правильные ответы для самопроверки вы сможете найти в конце списка. Обратите внимание, что у некоторых вопросов может быть несколько верных вариантов ответов.

6.9.2.1 Вопросы

Вопрос 1. Какого типа хеш хранит в себе атрибут krbPrincipalKey в домене ALD Pro?

a. AES

b. MD4

c. PBKDF2_SHA256

d. SHA1

Вопрос 2. В каком контейнере хранятся **политики и параметры политик** паролей в домене ald.company.lan?

a. cn=kerberos,dc=ald,dc=company,dc=lan

b. cn=kerberos,cn=accounts,dc=ald,dc=company,dc=lan

c. cn=pwpolicies,cn=accounts,dc=ald,dc=company,dc=lan

d. cn=pwpolicies,dc=ald,dc=company,dc=lan

e. cn=cosTemplates,cn=accounts,dc=ald,dc=company,dc=lan

f. cn=pwpolicies,cn=kerberos,dc=ald,dc=company,dc=lan

Вопрос 3. Для чего используются HBAC-правила?

a. Назначение прав на объекты каталога

b. Назначение прав на объекты файловых систем

c. Разрешение использования служб на хостах

d. Запрещение использования служб на хостах

Вопрос 4. С помощью какого механизма работают HBAC-правила?

a. Служба sshd

b. Служба sssd

c. Правила SUDO

d. PAM-стек и модуль pam_sss.so

Вопрос 5. В каком контейнере хранятся правила SUDO в домене ald.company.lan?

a. cn=sudoers,dc=ald,dc=company,dc=lan

b. cn=sudorules,dc=ald,dc=company,dc=lan

c. cn=sudorules,cn=sudo,dc=ald dc=company,dc=lan

d. cn=sudoers,cn=sudo,dc=ald dc=company,dc=lan

Вопрос 6. Каким способом настраивается на хостах использование правил SUDO, хранящихся в LDAP каталоге?

a. Через директиву @include в файле /etc/sudoers

b. Через настройку базы данных sudoers в файле /etc/nsswitch.conf

c. Через настройку базы данных sudoers в файле /etc/sssd/sssd.conf

d. Через директиву @include в файле /etc/sssd/sssd.conf

Вопрос 7. Что такое Standalone Миньон?

a. Центральный демон Salt, который может отдавать команды слушающим Миньонам.

b. Рабочая станция, на которой запущен демон Salt Minion, который может прослушивать команды от Мастера (Salt Master) и выполнять полученные задачи.

c. Рабочие станции или серверы, на которых работает демон Salt Minion, обладающий широкой функциональностью, работающий автономно от мастера. Используется для запуска заданий в системе без подключения к мастеру.

d. Вспомогательная служба Salt, работающая как прокси-сервер.

Вопрос 8. Как часто выполняется загрузка и применение групповых политик по умолчанию?

a. От 30 до 80 минут

b. От 30 до 50 минут

c. От 20 до 60 минут

d. От 20 до 50 минут

Вопрос 9. Какая команда используется для немедленной загрузки и применения политик (аналог `gpupdate /force`)?

a. Команда `sudo aldpro-gpupdate --gp`

b. Команда `sudo salt-call state.apply gpupdate.gp -c /srv/salt/standalone/config/`

c. Команда `sudo salt-call state.apply gpupdate.gp –force`

d. Команда `sudo salt-call state.apply gpupdateforce.gp -c /srv/salt/standalone/config/`

Вопрос 10. Какие из утверждений верные?

a. Если на один компьютер назначено несколько политик, использующих один и тот же параметр, то значения простых параметров будут переопределяться, а значения составных параметров – суммироваться.

b. Если на один компьютер назначено несколько политик, использующих один и тот же параметр, то значения простых и составных параметров будут суммироваться системой.

c. Если на одно подразделение назначено несколько политик, их атрибуты обрабатываются в порядке указанных приоритетов.

d. У корневого подразделения всегда наивысший приоритет.

e. У корневого подразделения всегда наименьший приоритет.

f. Скрипты групповых политик выполняются Минионом только один раз с результирующим содержанием словаря pillar вне зависимости от того, сколько раз соответствующий параметр был использован в групповых политиках, назначенных на этот компьютер или пользователя.

g. Скрипты групповых политик выполняются Минионом сколько раз, сколько соответствующий параметр был использован в групповых политиках, назначенных на этот компьютер или пользователя.

Ответы на вопросы

1. a
2. b, e
3. c
4. d
5. b, d
6. b
7. c
8. a
9. a
10. a, c, e, f

6.10 Заключение

В этом модуле мы прошли тему политик, поэтому можно сказать, что обязательный минимум уже выполнен. Далее мы перейдем к установке и настройке дополнительных подсистем, которые позволят вам быстро закрыть типовые потребности, которые будут возникать на большинстве предприятий.

6.11 Дополнительные источники информации

- Основы Salt, Крейг Себеник и Томас Хэтч
- Дополнительные параметры групповых политик из личного кабинета
- Формат даты GeneralizedTime

- Рекомендуем ознакомиться также с видеороликами нашего хорошего товарища на канале «ИТ-Проповедник». Учитывайте, что ролики записаны для версии 2.2.1, поэтому часть информации потеряла актуальность, но в целом они могут быть крайне полезны.
    - Политики паролей в ALD Pro 2.2.1

6.12 Обратная связь

Если остались вопросы, то их всегда можно задать в специальной теме.

Модуль 7. Дополнительные подсистемы ALD Pro

7.1 Введение

Из этого модуля вы узнаете о том, как с помощью штатных подсистем ALD Pro закрыть базовые потребности предприятий по обеспечению общего доступа к файлам. Мы рассмотрим не только процесс установки и настройки подсистемы через портал управления, но и разберем, как работают лежащие в ее основе компоненты, поэтому материал будет полезен даже в том случае, если вам потребуется настроить все врукопашную.

7.2 Процедура развертывания подсистем

Давайте сравним процесс установки ролей Windows и подсистем ALD Pro.

7.2.1 Установка ролей в Windows Server

Операционная система Windows из соображений безопасности по умолчанию устанавливается в минимальной комплектации, что позволяет администраторам настраивать сервера, которые будут выполнять только строго возложенные на них функции без чего бы то ни было лишнего. Процедура установки дополнительного программного обеспечения из состава операционной системы называется добавлением роли или компонентов (Add Roles and Features).

Серверные роли определяют основную функцию сервера в инфраструктуре предприятия, какие услуги он предоставляет пользователям или другим серверам в сети, например, контроллер домена, файловый сервер, принт-сервер, веб-сервер и т.д. Компоненты системы — это дополнительные приложения, которые устанавливаются вместе с ролями для расширения функциональности. Например, для роли веб-сервера можно дополнительно включить службу FTP или функцию публикации WebDAV.

Установка выполняется из Диспетчера сервера (англ. Server Manager) с помощью Мастера добавления ролей и компонентов (англ. Add Roles and Features Wizard), см. рис. 2.177.

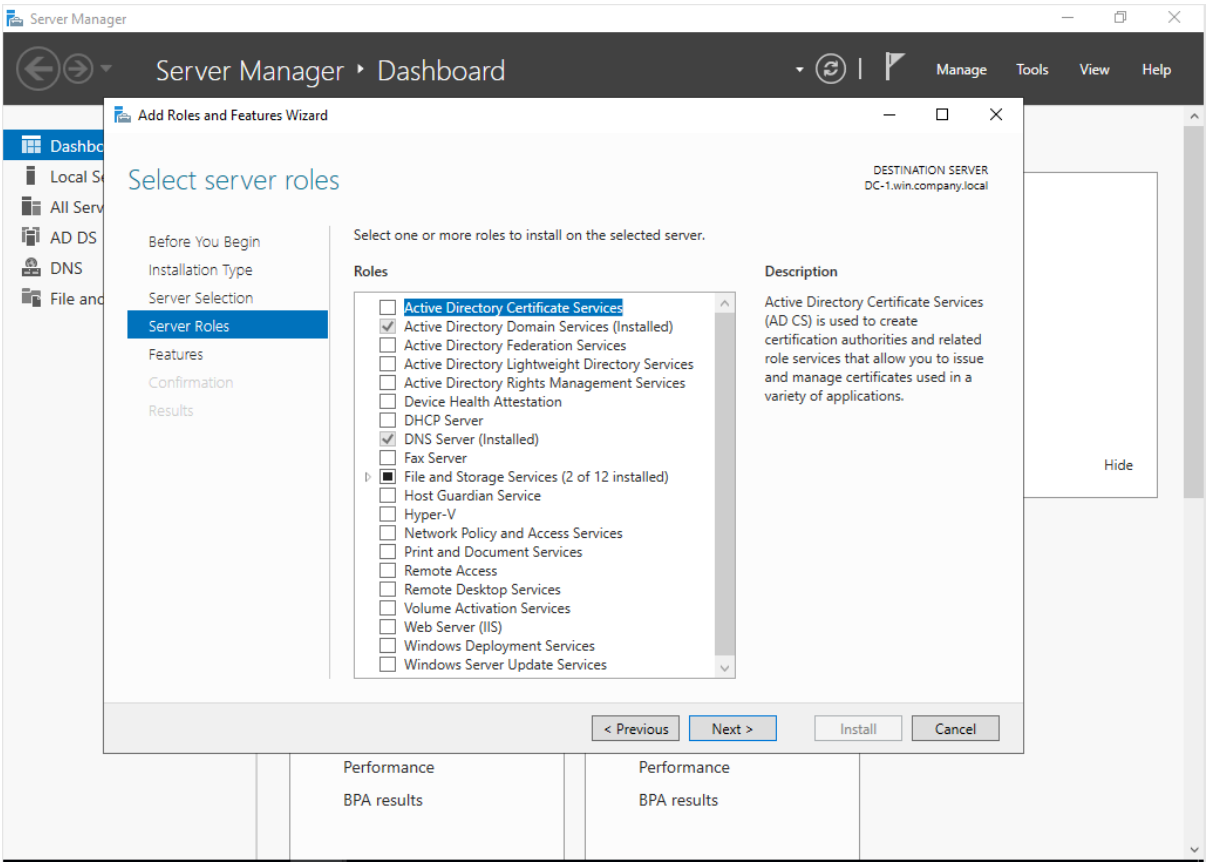


рис. 2.177 – Установка ролей из диспетчера сервера

Запустить приложение можно как непосредственно с того сервера, которым планируется управлять, так и с любого другого рабочего места. В этом случае Мастер установки будет выполнять PowerShell-командлеты удаленно по протоколу Windows Remote Management (WinRM), который является частной реализацией SOAP-протокола WS-Management, см. рис. 2.178. По умолчанию протокол WinRM должен быть включен на всех серверах Windows.

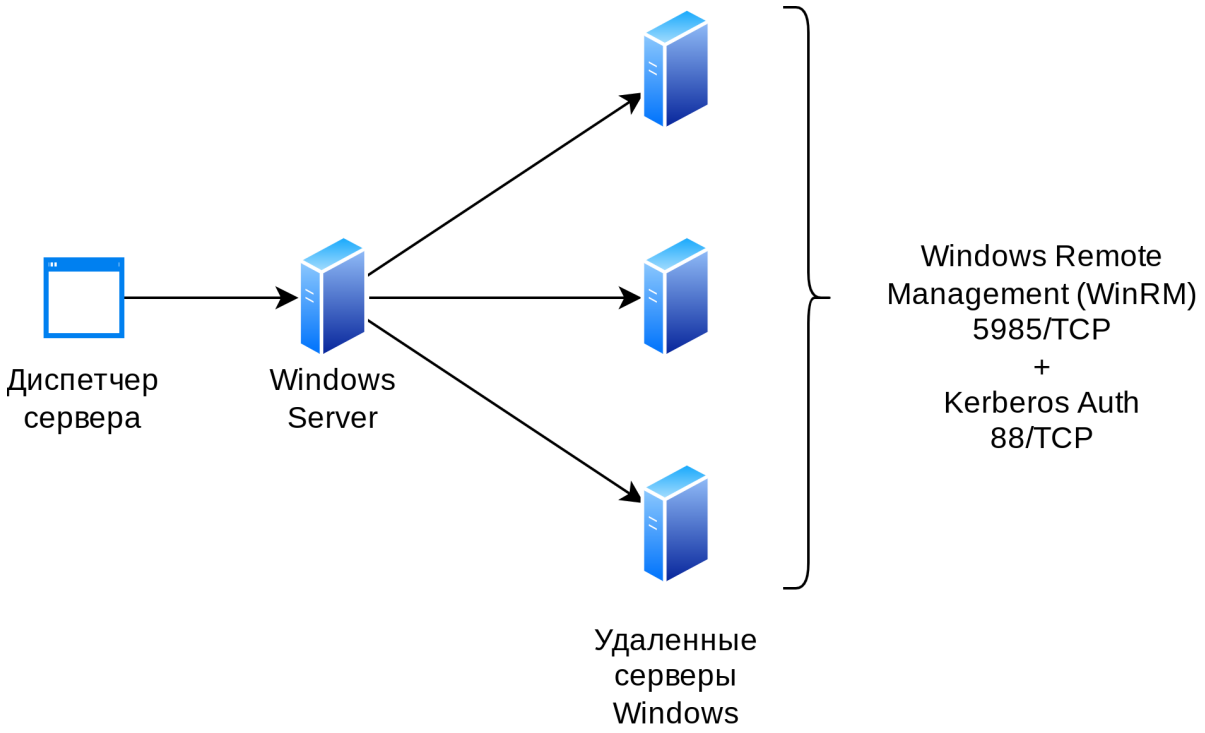


рис. 2.178 – Удаленная установка ролей через WinRM

Удаленная установка ролей и компонентов выполняется в интерактивном режиме, как по SSH, в связи с чем требуется постоянный доступ по сети к управляемым узлам. Когда мастер установки запускает командлет на удаленном хосте, PowerShell автоматически сериализует передаваемые объекты в XML и обратно, поэтому процесс ничем не отличается от того, как если бы Диспетчер был бы запущен локально.

Для настройки ролей используются специализированные оснастки, такие как Центр управления Active Directory (англ. Active Directory Administrative Center), Диспетчер DNS (англ. DNS Manager) и др. Удаленное управление серверами в этих оснастках выполняется аналогично через WinRM.

7.2.2 Установка подсистем в ALD Pro

Подсистемы ALD Pro, как роли Windows Server, определяют основную функцию сервера. Их можно установить на серверы ALSE 1.7.x с уровнем защищенности «Максимальный» («Смоленск»), но если в Windows объединение нескольких ролей на одном сервере считается просто нежелательным, то в ALD Pro это недопустимо.

Как мы уже отмечали, в Linux очень высок уровень повторного использования кода, и разные приложения довольно часто используют одни и те же компоненты, поэтому во избежание проблем с зависимостями пакетов каждая подсистема должна устанавливаться исключительно в отдельной операционной системе. Для этого вместе с лицензией на контроллер ALD Pro поставляется восемь лицензий на ALSE, одна из которых нужна для установки контроллера, а семь оставшихся вы

можете использовать для установки любых подсистем.

Процедура установки подсистемы предполагает два шага: сначала сервер вводится в домен, а затем на него через портал управления назначается необходимая роль. До вресии ALDPro 2.4.0 повышение роли сервера, передача команд на установку и обновление подсистем выполнялась через шины данных ZeroMQ от SaltStack. Это удобный механизм централизованного управления, который отлично подходит для оркестрации приложений в датацентрах с хорошей сетевой связанностью. Однако, в ИТ-инфраструктуре крупных предприятий с высокой географической распределенностью его производительность в ряде случаев падает:

- По мере роста в домене количества рабочих станций нагрузка на серверы со стороны SaltStack становилась слишком высокой. Это вызывало проблемы с аутентификацией пользователей: они не могли войти в домен.
- После перезагрузки salt-master все клиенты (миньоны) одновременно пытались подключиться к нему, что создавало высокую нагрузку на контроллер домена. В результате к контроллеру подключалась лишь часть клиентов, а остальные прекращали попытки по истечении таймаута.

На основе опыта и обратной связи от пользователей в версии ALDPro 2.4.0 процесс установки, обновления, удаления и конфигурирования подсистем был изменен. Разворачивание, обновление, управление подсистемами и репликация КД теперь работает по pull-модели.

7.2.2.1 Общая схема

Учитывая, что репликация LDAP работает не мгновенно и обеспечивает целостность данных только в конечном итоге, в версии ALDPro 2.4.0 перешли от «заданий автоматизации» к концепции настройки через «целевое состояние». В каталог будет записана метаинформация, описывающая желаемое конечное состояние, а не конкретные шаги по его достижению.

Мета-информация хранится в атрибутах учетной записи хоста, т. к. она имеет прямое отношение к конкретному хосту. Такое решение позволяет задействовать ролевую модель для управления правами доступа на конфигурирование подсистем — если у пользователя нет прав на изменение атрибутов учетной записи хоста, то он не сможет управлять этой подсистемой.

Пример потока данных при установке на PC_32 реплики контроллера домена отображен на рис. 2.179:

В панели управления контроллера домена DC_1 администратор как обычно назначает хосту PC_32 роль контроллера домена в сайте головного офиса hq. Логин/пароль привилегированной учетной записи для продвижения сервера до контроллера домена не требуется.

Бэкенд портала управления контроллера DC_1 создает для компьютера индивидуальную пользовательскую учетную запись с ограниченным сроком действия (например, в один день), назначает на нее необходимые роли и записывает данные этой учетной записи в защищенный атрибут компьютера rbta_subsystem_credentions. Метаинформацию о целевом состоянии хоста бэкенд вписывает в атрибут rbta_subsystem_metainfo.

Контроллер DC_1 передает изменения в учетной записи компьютера на контроллер DC_2 в порядке штатной репликации.

Хост PC_32 извлекает метаинформацию из LDAP каталога контроллера DC_2 в рамках процедуры применения групповых политик каждые полчаса и начинает установку подсистемы в соответствии с указанным целевым состоянием. Выполнение процедуры может быть форсировано по требованию администратора путем выполнения команды *aldpro-gpupdate –pm*.

Хост PC_32 после установки подсистемы записывает в LDAP каталог контроллера DC_2 результаты установки путем изменения метаинформации в атрибуте своей учетной записи.

Контроллер DC_2 передает информацию на контроллер DC_1 в порядке штатной репликации

Бэкенд портала управления контроллера DC_1 вычитывает информацию из каталога через бэкенд при обновлении страницы.

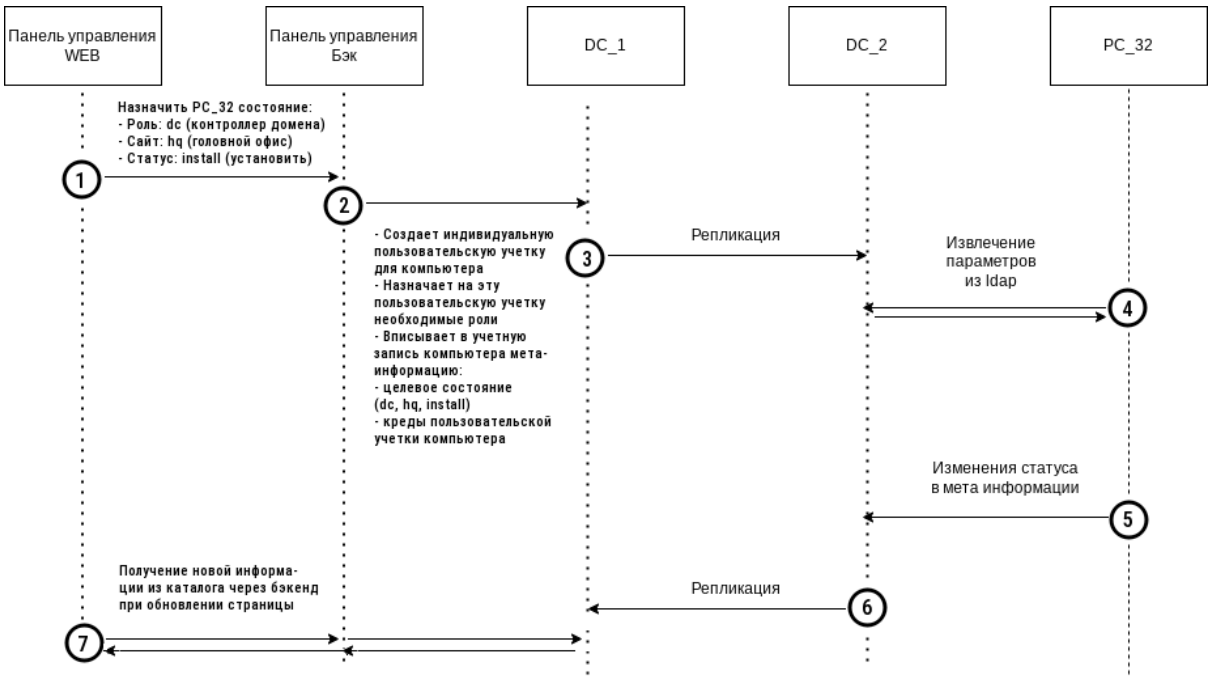


рис. 2.179 – Поток данных при установке реплики контроллера

7.2.2.2 Жизненный цикл подсистем

Учитывая сложность изменений, которые нужно вносить на целевом хосте при установке, обновлении и настройке подсистем, далеко не все изменения принципиально возможны и должны быть разрешены. Например, не следует разрешать установку двух подсистем на один хост или установку контроллера домена после удаления роли файлового сервера. Поэтому для управления подсистемами определен конечный набор состояний и допустимых переходов между ними, например, см. рис. рис. 2.180

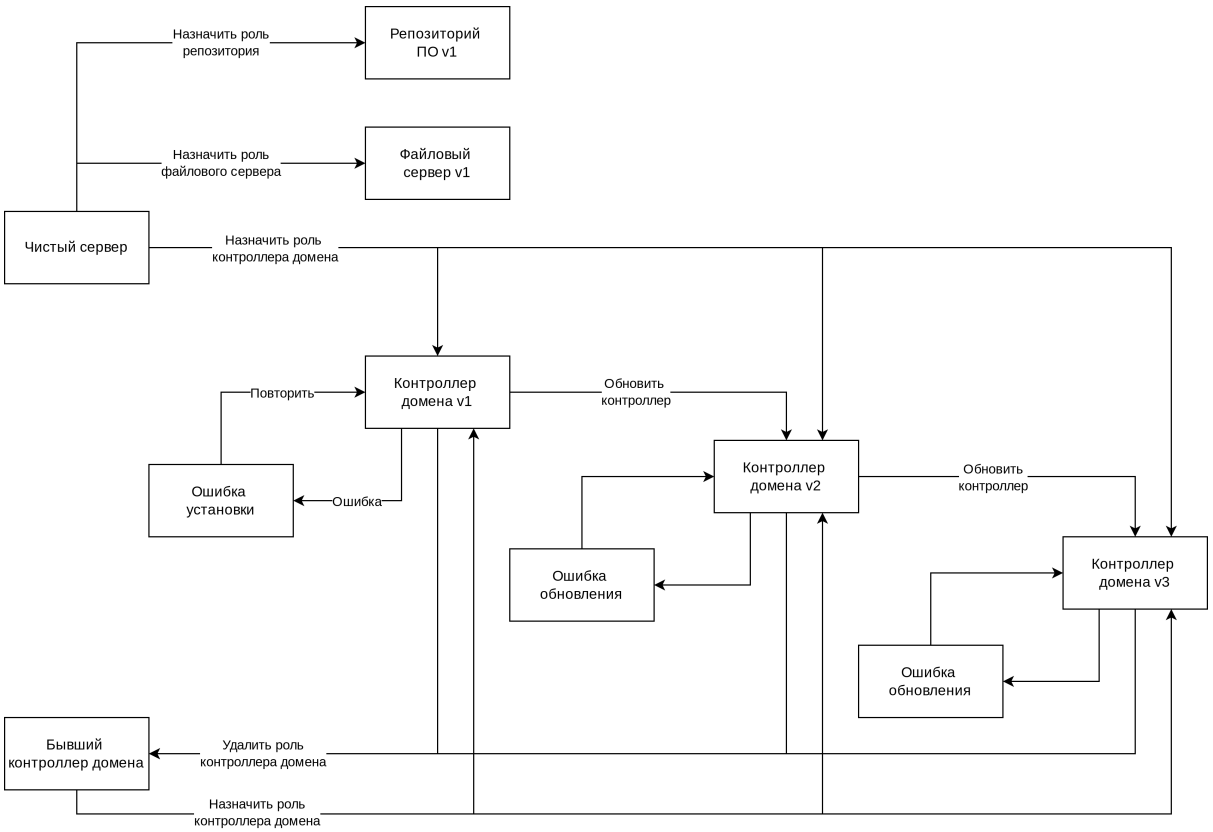


рис. 2.180 – Жизненный цикл подсистем

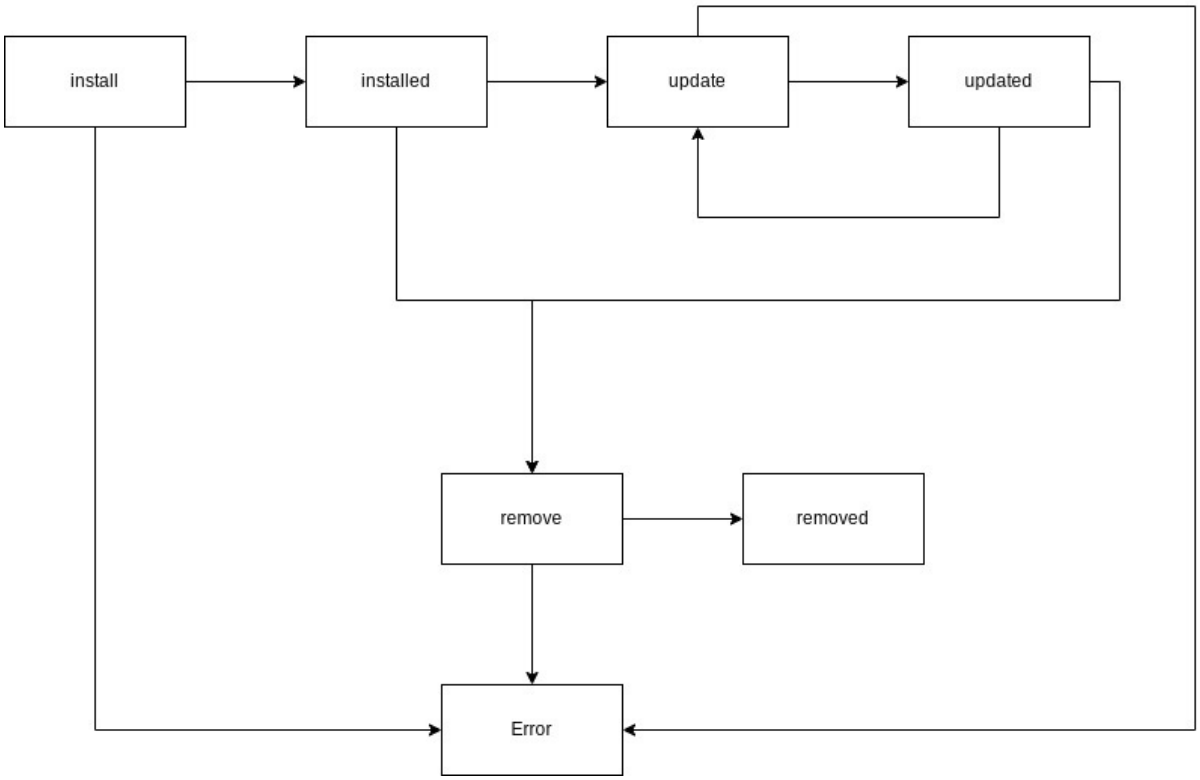


рис. 2.181 – Машина состояний подсистем

Контроль допустимых переходов должен выполняться как на портале управления так и на самой подсистеме. Интерфейс портала управления должен исключать возможность внесения конфликтующих изменений, например, если на целевой хост уже назначена роль файлового сервера, такой хост должен быть недоступен для продвижения до контроллера домена.

Однако, механизмов контроля на портале управления недостаточно, т. к. контроллер домена может не обладать актуальной информацией о состоянии системы, поэтому конечную проверку непротиворечивости состояний выполняет сам целевой хост. Получив новое состояние хост должен сам принять решение, является ли такое состояние допустимым или нет. Если состояние допустимо, то хост начинает переход в новое состояние. Если перейти в указанное состояние невозможно, то хост записывает в каталог текущее состояние. Журнал с информацией о принятых и отклоненных состояниях храниться также в LDAP каталоге, чтобы администратор с портала управления на любом контроллере домена смог понять, почему его предыдущая команда была отклонена.

7.3 Подсистема «Общий доступ к файлам»

Подсистема общего доступа к файлам ALD Pro реализована на базе программного обеспечения Samba, которое позволяет серверу Linux предоставлять доступ к своим ресурсам по протоколу общего доступа к сетевым файлам CIFS (Common Internet File System) аналогично тому, как это делает сервер Windows.

Протокол CIFS является диалектом протокола SMB (Server Message Block) от компании Microsoft, и оба названия могут использоваться как синонимы. Следует понимать, что этот протокол позволяет выполнять любые удаленные вызовы (Remote Protocol Calls, RPC), поэтому через него в Windows работает не только общий доступ к файлам и принтерам, но и такие функции, как NETLOGON и LSA. Этой особенностью протокола объясняется, почему в Samba были реализованы в том числе функции контроллеров домена NT4 и Active Directory.

В настоящее время Samba представляет собой комбайн со множеством функций и настроек, некоторые из которых даже несовместимы друг с другом, поэтому для правильной настройки этой службы нужно знать ее архитектуру и особенности каждого из режимов работы.

7.3.1 Архитектура подсистемы

Перечислим основные компоненты Samba на файловом сервере:

- **smdb** – служба, реализующая доступ к ресурсам сервера по SMB на порту 445/TCP.
- **nmdb** – служба, реализующая NetBIOS. Используется для разрешения NetBIOS имен в IP-адреса.
- **winbind** – клиентская часть, которая позволяет хосту быть участником домена Active Directory. Служба используется, например, для автоматического обнаружения контроллеров домена, преобразования идентификаторов, NTLM-аутентификации. В настоящий момент служба считается устаревшей, и часть ее функций передана SSSD.
- **sssd** – клиентская часть, которая позволяет хосту быть участником домена. На серверах с Samba служба SSSD берет на себя часть функций winbind, иногда ограничивая возможности Samba.

  Например, разработчики SSSD по соображениям безопасности принципиально против реализации в своем продукте поддержки протокола NTLM, поэтому, если файловый сервер Samba устанавливается совместно со службой SSSD, то проксирование аутентификационных запросов по протоколу NTLM будет невозможно.
- **ipsam passdb** – модуль passdb позволяет Samba работать в режиме автономного сервера (standalone), извлекая информацию о пользователях и паролях из локальной базы данных.

  Реализация модуля ipasam позволяет вместо локальной базы использовать данные из LDAP-каталога. При подключении к каталогу модуль выполняет аутентификацию по протоколу Kerberos, а для шифрования трафика использует GSSAPI SASL, поэтому допустимо обращение к каталогу не только по unix-сокету, но и по сети.

После успешной установки подсистемы, администратор может приступить к настройке сетевых па-

пок на файловом сервере. Информация хранится и передается через LDAP каталог. В этот каталог включаются только те атрибуты, которые можно изменять через интерфейс портала управления.

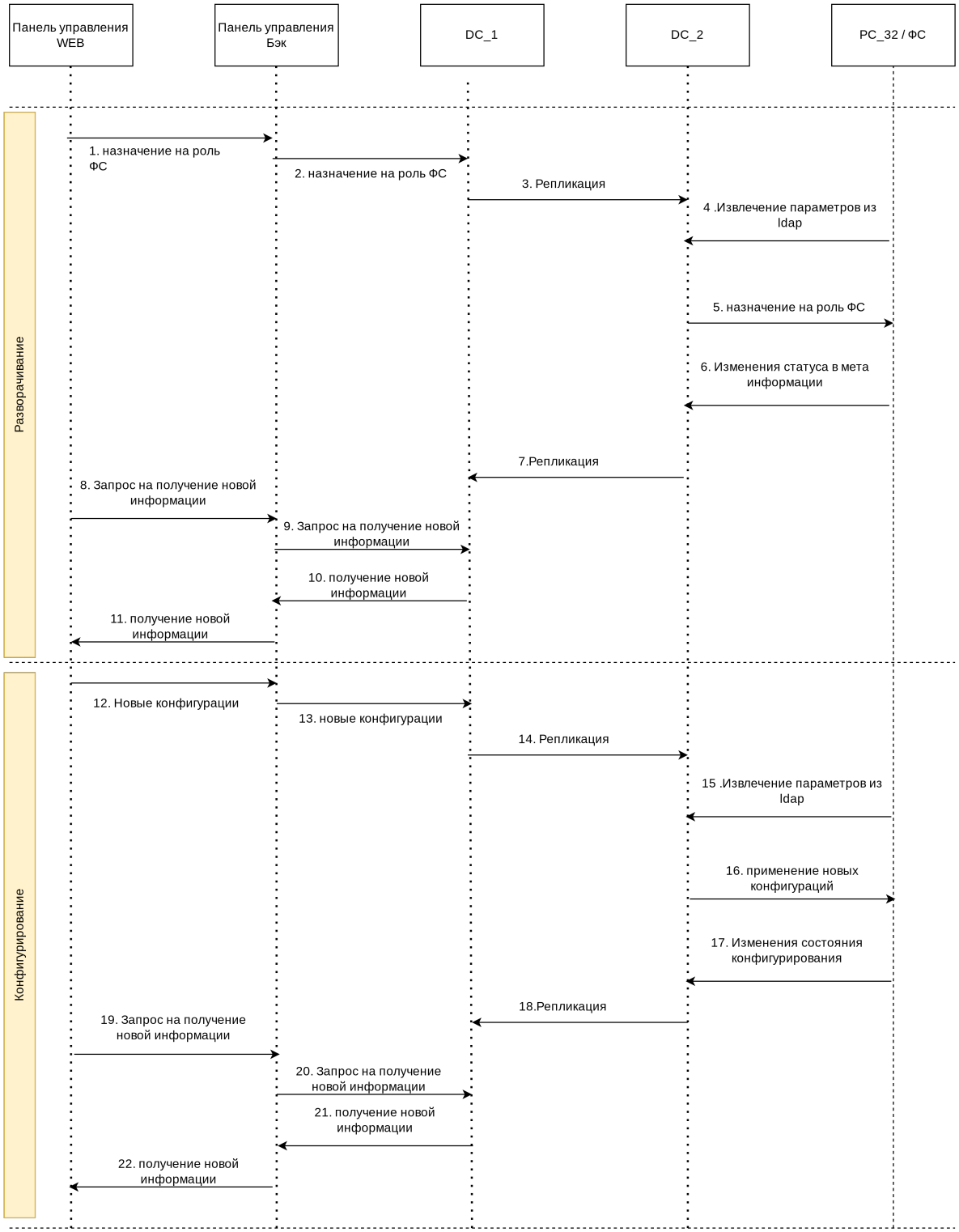


рис. 2.182 – Поток данных при установке файлового сервера

7.3.2 Установка подсистемы

Диаграмма состояний файлового сервера соответствует диаграмме состояний на рисунке рис. 2.181, за исключением того, что после удаления файлового сервера, мы не можем использовать машину повторно в качестве любой подсистемы, не переустанавливая ОС.

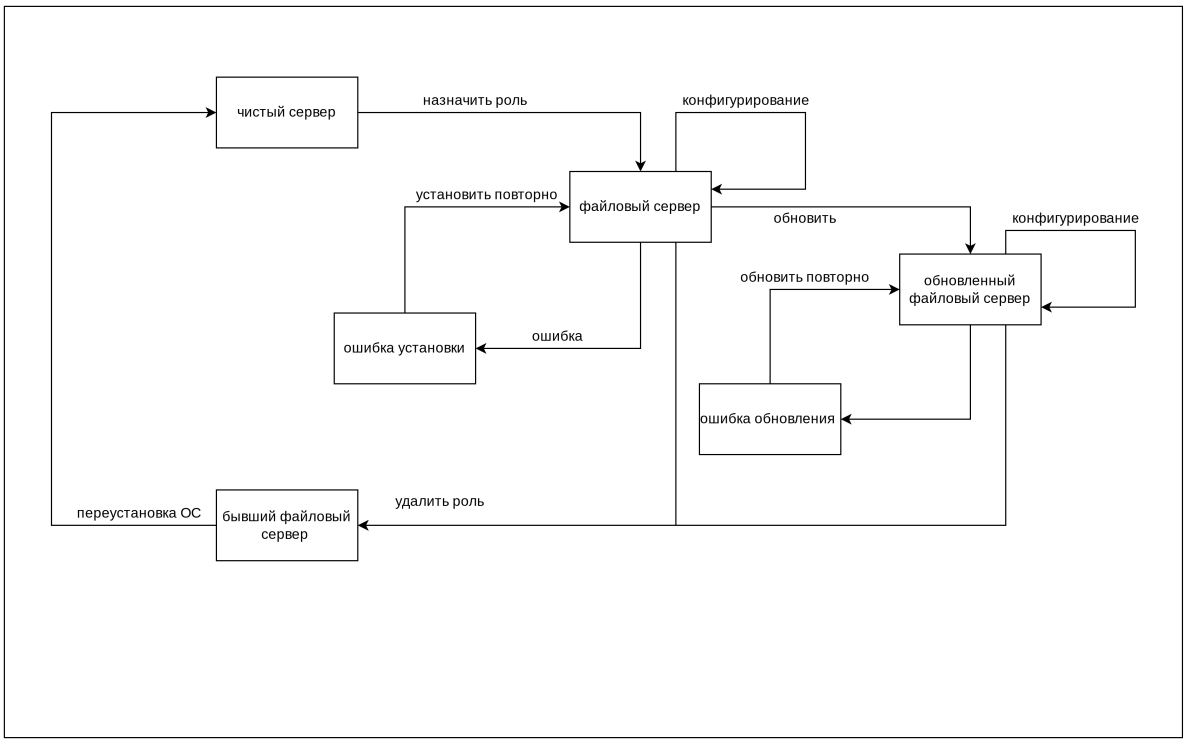


рис. 2.183 – Жизненный цикл файлового сервера

Ранее мы уже создавали виртуальные машины для установки подсистем, но если вы этого не сделали, давайте напомним основные моменты:

1. Клонируйте машину **Стенд ALD Pro — file-1.ald.company.lan** из снимка, см. Не забывайте сгенерировать новый MAC-адрес для сетевого интерфейса, см. рис. 2.184.

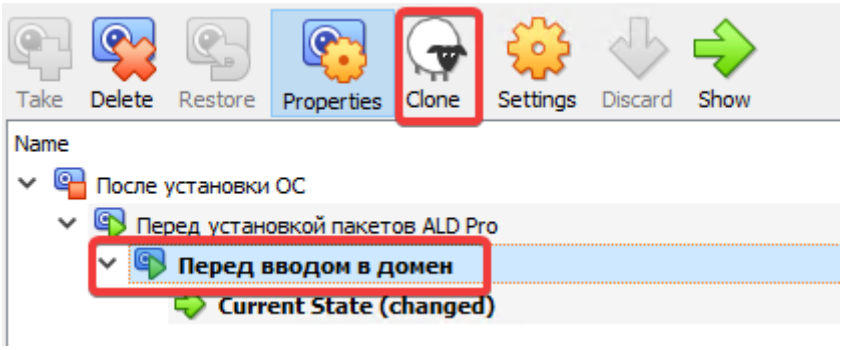


рис. 2.184 – Клонирование виртуальной машины

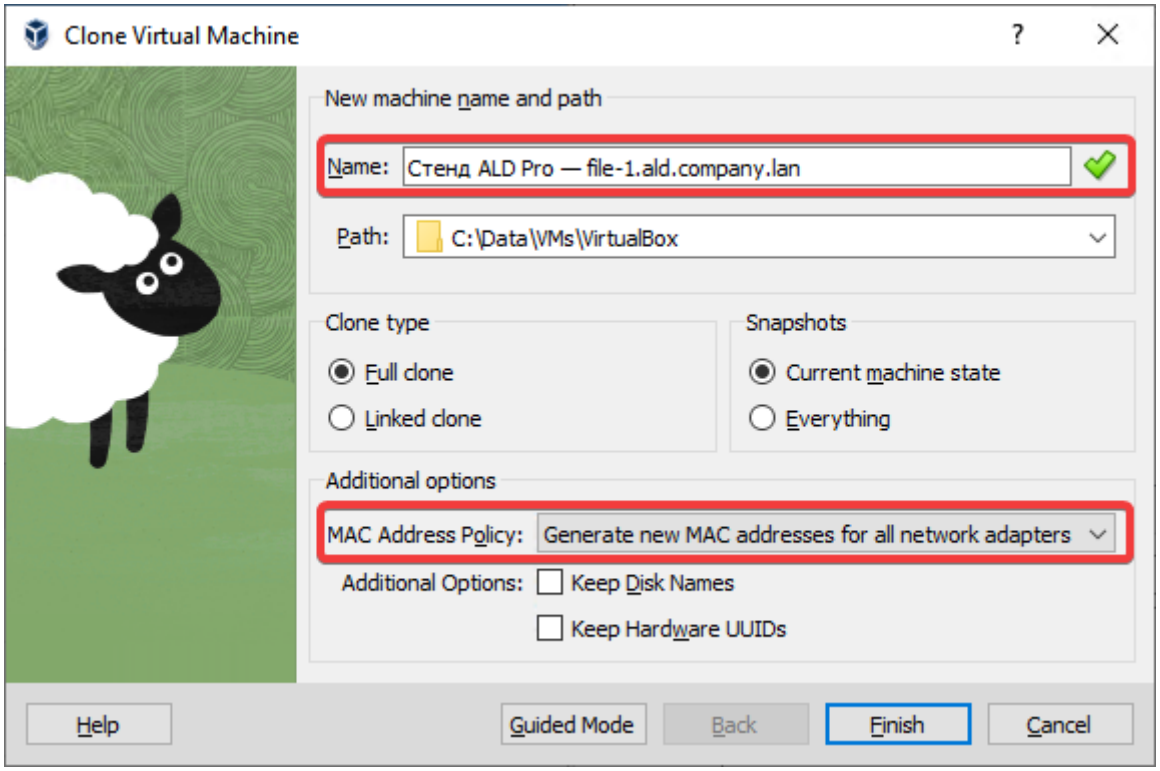


рис. 2.185 – Генерация нового MAC-адреса

2. Назначьте сетевому интерфейсу адрес 10.0.1.26/24 через NetworkManager (или в файле `/etc/network/interfaces`) и примените изменения.

3. Введите компьютер в домен:

```
$ set +o history
$ sudo /opt/rbta/aldpro/client/bin/aldpro-client-installer --domain ald.company.lan --
↪account admin -- password 'Pa$$w0rd' --host file-1 --gui --force
$ set -o history
```

4. Перезагрузите машину:

```
$ sudo reboot
```

Теперь в портале управления на странице *Роли и службы сайта → Общий доступ к файлам* на вкладке *Перечень серверов* назначьте эту роль новому серверу *file-1*. Для этого нажмите на кнопку *+ Новый сервер* и заполните карточку.

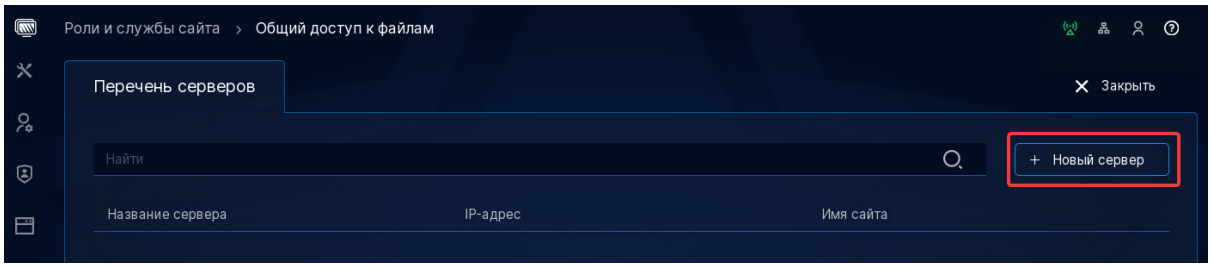


рис. 2.186 – Добавление нового сервера общего доступа к файлам

Выберите сервер **file-1.ald.company.lan** из списка и укажите сайт «Головной офис». Чтобы применить изменения, нажмите на кнопку *Сохранить*.

Привязка к сайту будет использоваться в будущих релизах для настройки гранулярного доступа. Планируется реализовать возможность делегирования прав на администрирование серверов в соответствии с их положением в иерархии структурных подразделений.

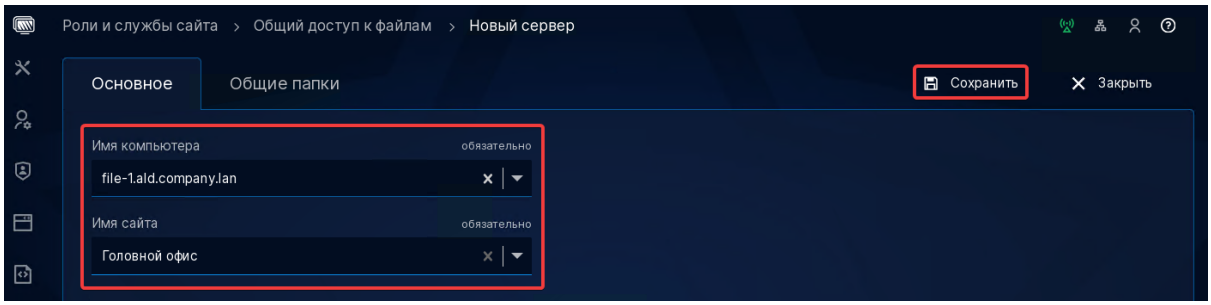


рис. 2.187 – Назначение роли общего доступа к файлам на сервер file-1

Проверить выполнение скрипта автоматизации можно на странице *Роли и службы сайта > Общий доступ к файлам > Сервер: file.ald.company.lan*, вкладка «Основное».

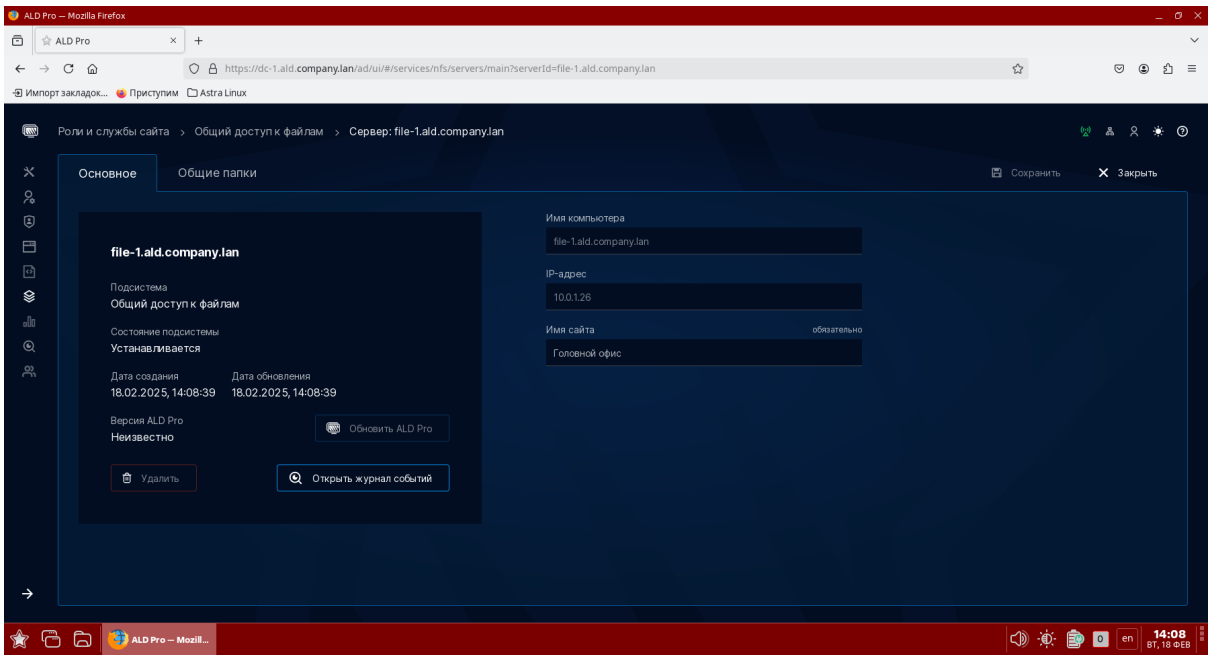


рис. 2.188 – Журнал заданий автоматизации

Установка начнется в течении 30 минут. В ALD Pro версии 2.4.0 после применения политики обновления необходимо выполнить команду:

```
sudo aldpro-gpupdate --pm
```

В версии 2.2.0 мы перешли на автономную службу salt-minion. Она обеспечила высокую произво-

дительность механизма групповых политик. В версии 2.4.0 мы закрепили этот успех и перевели на этот механизм управление подсистемами. Вместе с тем у этого решения есть недостаток - невозможность форсировать применение политик и настроек по инициативе сервера. Работа над улучшением ведется, но пока приходится ждать или форсировать применение изменений вручную.

Следующая команда собирает pillar на основе переменных окружения из ldap и запускает установку/обновление/удаление;

```
sudo aldpro-roles --iud --action [install, update, remove]
```

Чтобы немедленно применить новые параметры заданные через портал управления, требуется выполнить команду:

```
sudo aldpro-roles --subsystem_settings smb
```

Эта команда собирает pillar переменных окружения из ldap и применяет параметры для указанной подсистемы.

Внимание: Командой aldpro-roles нельзя одновременно установить подсистему и обновить для нее настройки из LDAP-каталога. Следующая команда вызовет ошибку: `aldpro-roles --iud --action install --subsystem_settings smb`.

В случае успешной установки в портале управления в разделе «*Роли и службы сайта → Общий доступ к файлам*» на вкладке *Перечень серверов* появится новый сервер:

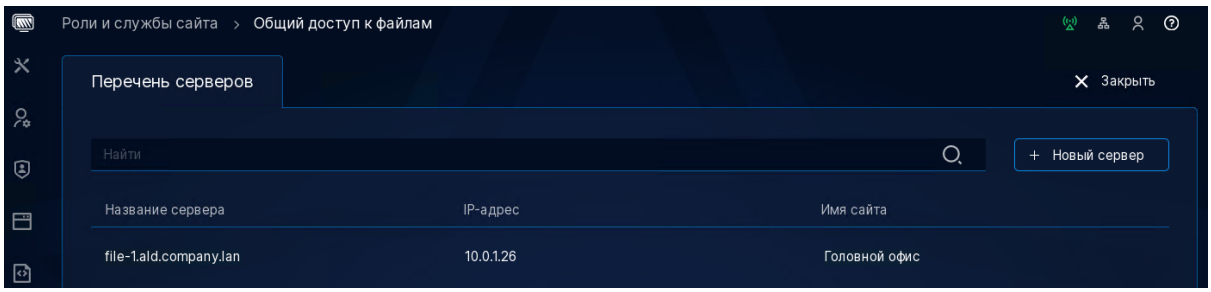


рис. 2.189 – Файловый сервер, готовый к настройке

7.3.3 Работа с сервером общего доступа к файлам

7.3.3.1 Просмотр общих папок

На странице *Роли и службы сайта → Общий доступ к файлам* откройте карточку сервера и перейдите на вкладку *Общие папки*.

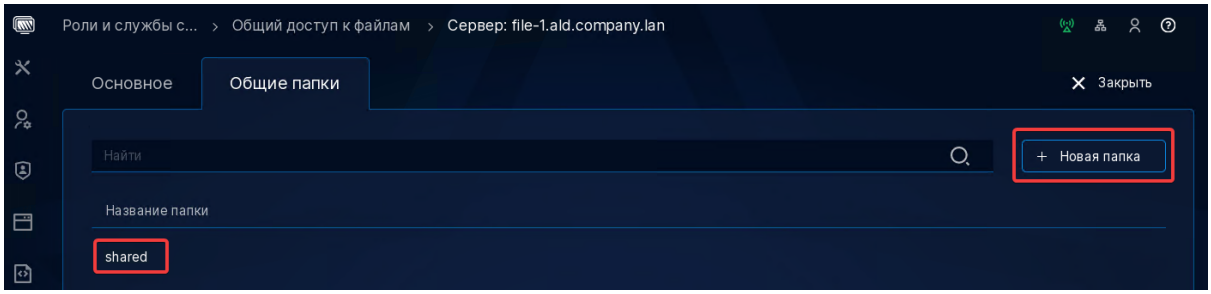


рис. 2.190 – Общие папки файлового сервера

Если после установки подсистемы на сервере создать общую папку `shared`, она будет доступна по протоколу smb по адресу `smb://file-1.ald.company.lan/shared`. Вы можете открыть эту папку с любого компьютера в домене с аутентификацией по Kerberos, см. рис. 2.191. Если у вас не сработает Kerberos-аутентификация, проверьте, что пользователю назначены права доступа на эту папку и в связке ключей у него есть действительный TGT-билет.



рис. 2.191 – Доступ к папке shared

7.3.3.2 Управление общей папкой

Откройте общую папку для того, чтобы перейти к ее настройкам. Перед вами будет три вкладки:

- **Основное** – на этой вкладке вы можете изменить название папки, а также:
  - выбрать тип виртуальной файловой системы,
  - указать будет ли эта папка доступна для просмотра в браузерах сетевых ресурсов (Browsable = yes - папка будет доступна для просмотра).
  - можно ли создавать правила доступа к папке на основе пользователей и групп (Access based share enum = yes - можно)
  - установить права доступа для новых файлов и каталогов
  - включить или отключить наследование прав доступа от родительской папки

Внизу справа расположена кнопка «Удалить папку» для выполнения соответствующего действия.

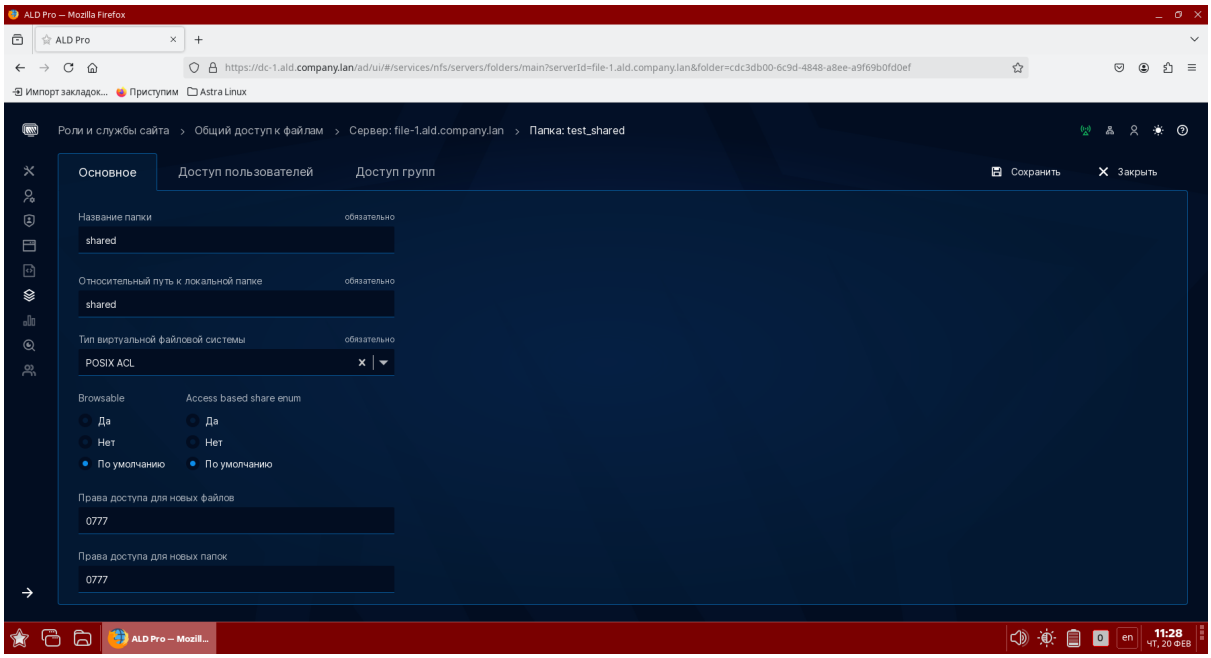


рис. 2.192 – Основные настройки общей папки

- **Доступ пользователей** – на вкладке представлен список пользователей, которым предоставлен доступ к папке на уровне SMB-подключения, с указанием типа предоставляемого права допуска (чтение/полный доступ). Для изменения прав доступа необходимо нажать на кнопку «Редактировать доступ». По умолчанию список пуст.

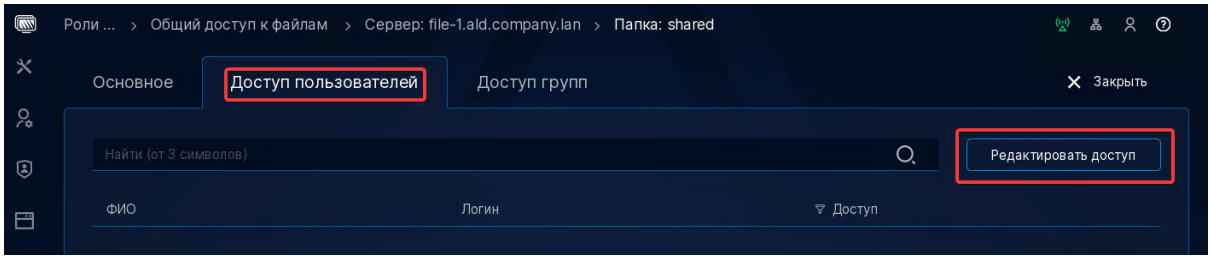


рис. 2.193 – Права доступа для пользователей

- **Доступ групп** – на вкладке представлен список групп, которым предоставлен доступ на уровне SMB-подключения.

Так как по умолчанию списки прав доступа для пользователей и групп пустые, то никто не имеет прав на общую папку. В этом можно убедиться, попробовав создать файл. Перейдем в окно менеджера файлов на **pc-1**, где мы открывали эту папку и попробуем создать файл.

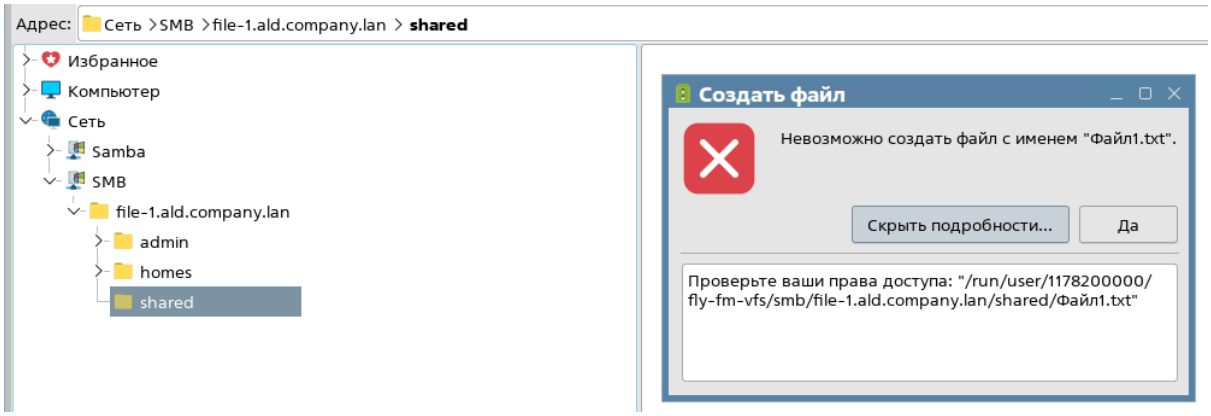


рис. 2.194 – Ошибка при создании файлов в случае отсутствия прав доступа

Назначение прав на общую папку

Предоставим полные права пользователю admin на общую папку. Для этого в карточке общей папки на вкладке «Доступ пользователей» кликнем на кнопку «редактировать доступ».

На карточке «Редактирование доступа пользователей» можно выбрать:

- **Чтение** – доступ на просмотр.
- **Полный доступ** – доступ на просмотр, добавление, изменение и удаление.

Выберем вариант «Полный доступ» и добавим пользователя «Administrator». Для применения не забудем нажать кнопку «Сохранить».

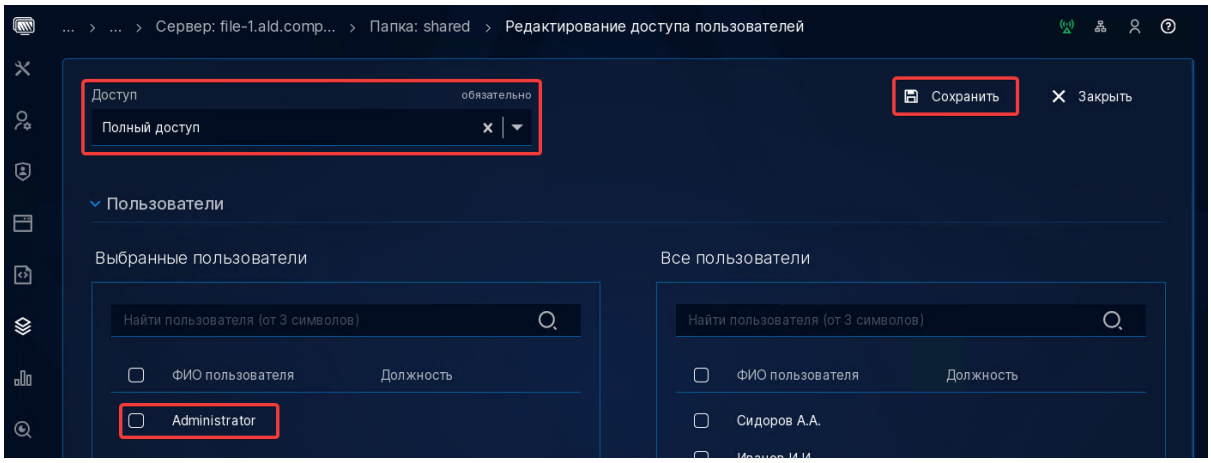


рис. 2.195 – Назначение прав доступа на пользователя

После назначения прав доступа вы можете убедиться, что никаких ошибок при создании файлов больше не возникает.

Попробуйте отредактировать этот файл в текстовом редакторе Kate, после чего откройте сетевую

папку из-под учетной записи другого пользователя, например, iivanov, и убедитесь, что файл ему недоступен. Предоставим права пользователю iivanov на чтение общей папки shared через портал управления, используя группу «moscow-office-employee-it».

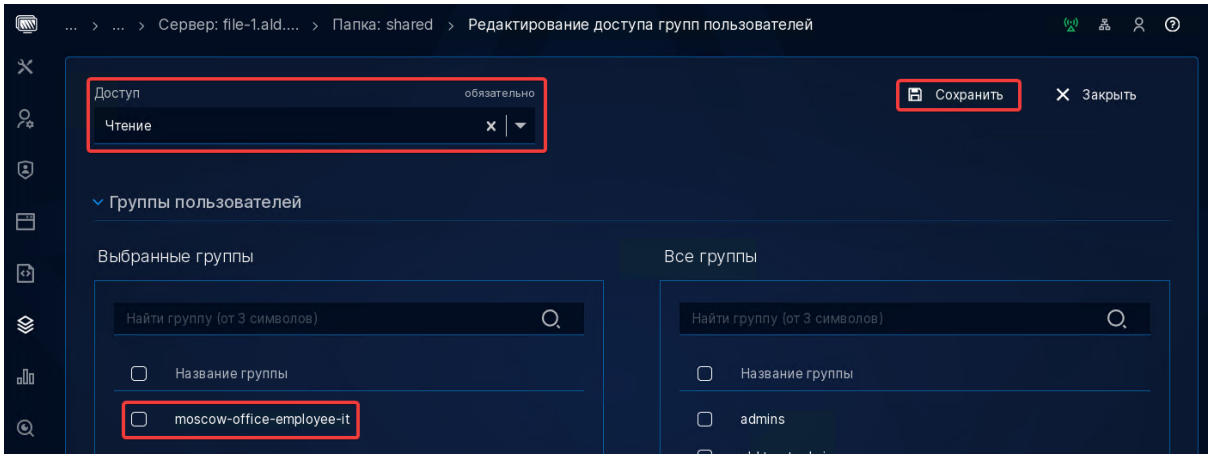


рис. 2.196 – Назначение прав доступа на группу

Обновите окно менеджера файлов и убедитесь, что пользователь iivanov получил право на просмотр файлов из общей папки, но у него нет прав на запись.

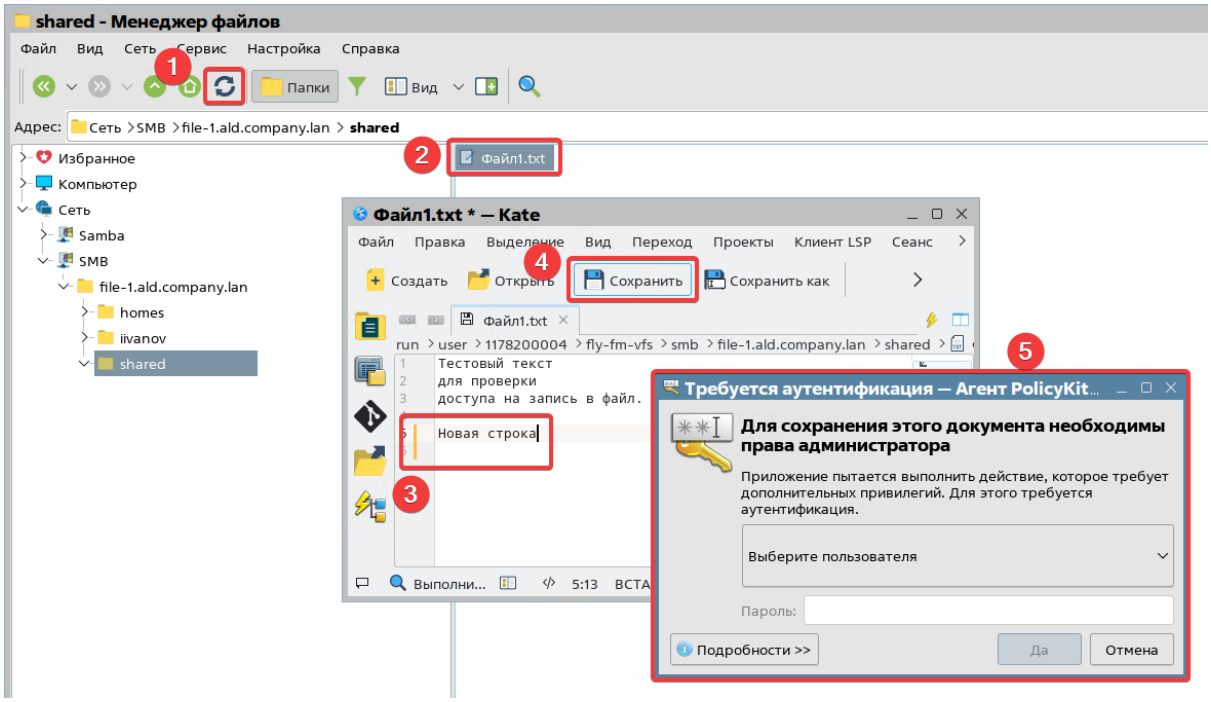


рис. 2.197 – Проверка прав доступа на файлы

Создание общей папки

Для создания общей папки на карточке сервера общего доступа к файлам перейдем на вкладку «Общие папки» и кликнем по кнопке «+ Новая папка».

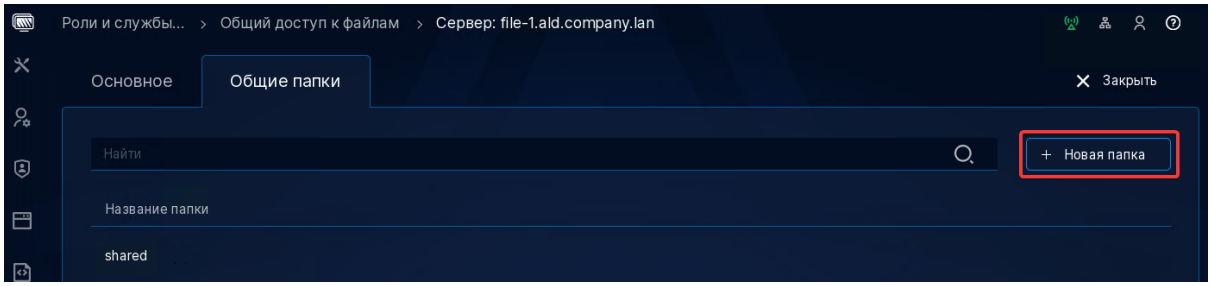


рис. 2.198 – Создание новой общей папки

После создания новой общей папки вам нужно будет настроить права доступа на вкладках «Доступ пользователей» и «Доступ групп».

Если планируется, что общая папка будет занимать очень много места, вы можете разместить ее физически на другом диске и примонтировать через fstab. Физически она находится на сервере по адресу `/opt/samba_shares/test_share`.

7.3.3.3 Конфигурационный файл Samba-сервера

До вресии 2.4.0 все настройки при создании и изменении общих папок, выполняемых через портал управления сохраняются в конфигурационном файле Samba-сервера `/etc/samba/smb.conf`. Посмотрим его содержимое.

```
[global]
workgroup = ALDPRO
realm = ald.company.lan
log level = 7
domain master = No
domain logons = Yes
dedicated keytab file = FILE:/etc/samba/samba.keytab
kerberos method = dedicated keytab
log file = /var/log/samba/log.%m
passdb backend = ipasam:"ldap://dc-1.ald.company.lan:389 ldap://dc-2.ald.company.
↪lan:389"
ldap group suffix = cn=groups,cn=accounts
ldap machine suffix = cn=computers,cn=accounts
ldap user suffix = cn=users,cn=accounts
ldap ssl = off
ldap suffix = dc=ald,dc=company,dc=lan
ldap admin dn = cn=Directory Manager
max log size = 100000
disable spoolss = Yes
security = user
create krb5 conf = No
rpc_daemon:lsasd = fork
rpc_daemon:epmd = fork
rpc_server:tcpip = yes
rpc_server:netlogon = external
rpc_server:samr = external
rpc_server:lsasd = external
```

(continues on next page)

(продолжение с предыдущей страницы)

```
rpc_server:lsass = external
rpc_server:lsarpc = external
rpc_server:epmapper = external
ldapsam:trusted = yes
idmap config \* : backend = tdb
restrict anonymous = 2
usershare allow guests = no

[homes]
browsable = yes
writable = yes
create mask = 0600
directory mask = 0700
valid users = %S
read only = No
guest ok = no

[shared]
path = /home/share
writable = yes
browseable = yes
valid users = admin, @moscow-office-employee-it
admin users = admin
read list = @moscow-office-employee-it

[test_share]
path = /opt/samba_shares/test_share
browseable = yes
valid users = @admin
```

В файле на файловом сервере ALD Pro присутствуют следующие разделы:

- **[global]** – параметры этого раздела применяются к серверу в целом.
- **[homes]** – в этом разделе задаются параметры для сетевых папок пользователей, которые будут «создаваться на лету» при обращении пользователей к файловому серверу. В этой секции должны быть определены следующие параметры:
  - «**valid users = %S**» — шаблон %S автоматически заменяется именем удаленного пользователя в тот момент, когда он пытается получить доступ к своему общему ресурсу.
  - Путь к каталогу отсутствует, по умолчанию он указывает на `/home/%S`.

    Следует понимать, что папку пользователя на диске в каталоге /home/ нужно создать заранее, иначе пользователь будет видеть ее при подключении по SMB, но не сможет создавать в ней файлы.
- **[shared]** – раздел с параметрами общей папки, которая в версии до 2.4.0 создавалась по умолчанию. Её путь указывает на `/home/share`.
- **[test_share]** – раздел с параметрами общей папки, которую создали вручную.

Начиная с версии 2.4.0 часть настроек перенесена в LDAP, а *passdb backend* больше не используется.

Теперь конфигурация Samba генерируется по шаблону:

```
[global]
workgroup = ALD
realm = ALD.COMPANY.LAN
dedicated keytab file = FILE:/etc/samba/samba.keytab
kerberos method = dedicated keytab
idmap config ALD : 1457400000-1457599999
idmap config ALD : backend = sss
idmap config * : range = 0-0
log file = /var/log/samba/log.%m
include = registry
include = /etc/samba/share.conf

[homes]
browsable = yes
writable = yes
create mask = 0600
directory mask = 0700
valid users = %S
read only = No
guest ok = no
```

Здесь:

- **ALD** - это значение до первой точке в имени домена в верхнем регистре.
- **1457400000-1457599999** - это диапазон id. Он расчитывается по формуле из атрибутов записи cn=ALD.COMPANY.LAN_id_range,cn=ranges,cn=etc,dc=ald,dc=company,dc=lan:
  - первое число - `ipaBaseID`
  - второе число - `ipaBaseID + ipaIDRangeSize - 1`
- **include = /etc/samba/share.conf** - подключает конфигурационный файл, в который пишется конфигурация общих папок.

Если в секции `[global]` конфигурационного файла smb.conf задан параметр `include = registry`, то Samba будет включать настройки из базы `/var/lib/samba/registry.tdb`, которая имеет структуру, аналогичную реестру Windows. На файловых серверах реестр не используется, но на контроллерах домена это хранилище хранит большую часть настроек. Вы можете задействовать реестр на файловых серверах с той же целью.

Для управления реестром нужно запускать утилиту `samba-regedit`, параметры службы Samba хранятся в ветке реестра `/HKEY_LOCAL_MACHINE/SOFTWARE/Samba/smbconf/`.

Результирующие настройки можно увидеть с помощью утилиты testparm:

```
admin@dc-1:~$ sudo testparm
```

(continues on next page)

(продолжение с предыдущей страницы)

```
Load smb config files from /etc/samba/smb.conf
lpcfg_do_global_parameter: WARNING: The "domain logons" option is deprecated
Loaded services file OK.
Weak crypto is allowed
idmap range not specified for domain '*'
ERROR: Invalid idmap range for domain *!

Server role: ROLE_DOMAIN_PDC

Press enter to see a dump of your service definitions

# Global parameters
[global]
        create krb5 conf = No
        dedicated keytab file = FILE:/etc/samba/samba.keytab
        disable spoolss = Yes
        domain logons = Yes
        domain master = No
        kerberos method = dedicated keytab
        ldap admin dn = cn=Directory Manager
        ldap group suffix = cn=groups,cn=accounts
        ldap machine suffix = cn=computers,cn=accounts

...
```

7.3.4 Аутентификация на файловом сервере Samba

7.3.4.1 Аутентификация по протоколу Kerberos

При установке подсистемы общего доступа к файлам Kerberos-аутентификация работает из коробки. Скрипты автоматизации присоединяют службу Samba к домену, выполняя следующие действия:

- Создают запись «krbprincipalname=cifs/file-1.ald.company.lan@ALD.COMPANY.LAN» в контейнере «cn=services,cn=accounts,dc=ald, dc=company,dc=lan».
- Выгружают ключи этой учетной записи в файл /etc/samba/samba.keytab.

Указанная учетная запись используется службой Samba для проверки сервисных билетов Kerberos и получения информации из службы каталога по LDAP-протоколу. Содержимое keytab-файла можно посмотреть с помощью команды `klist`:

```
admin@dc-1:~$ sudo klist -ket /etc/samba/samba.keytab
Keytab name: FILE:/etc/samba/samba.keytab
KVNO Timestamp           Principal
---- ------------------- ------------------------------------------------------
   1 10.03.2024 16:38:27 cifs/file-1.ald.company.lan@ALD.COMPANY.LAN (aes256-cts-hmac-
↪sha1-96)
   1 10.03.2024 16:38:27 cifs/file-1.ald.company.lan@ALD.COMPANY.LAN (aes128-cts-hmac-
↪sha1-96)
```

Требование использовать указанный keytab-файл задано в параметрах «kerberos method» и «dedicated keytab» конфигурационного файла smb.conf:

```
admin@dc-1:~$ sudo cat /etc/samba/smb.conf
```

```
[global]
...
kerberos method = dedicated keytab
dedicated keytab file = /etc/samba/samba.keytab
...
```

Кроме «dedicated keytab» служба Samba может работать еще в двух режимах:

- **system keytab**, когда используется системный keytab-файл /etc/krb5.keytab.
- **secrets only**, когда используется учетная запись из локальной базы secrets.tdb. Если kerberos method не задан, то служба работает в режиме secrets only по умолчанию.

7.3.4.2 Аутентификация по протоколу NTLM (по паролю)

Если подключаться к файловому серверу по IP-адресу или с компьютера, который не является участником домена, то служба smbd предложит выполнить аутентификацию по протоколу NTLMv2, и в сетевом трафике можно будет увидеть пакеты NTLMSSP.

Как вы помните, протокол аутентификации NTLM предполагает, что есть три участника: Клиент, Сервер и Контроллер домена. Сервер при получении запроса на аутентификацию должен сгенерировать «challenge» сообщение размером 8 байт и попросить Клиента зашифровать его своим паролем. Полученное от Клиента сообщение вместе с исходным числом Сервер должен передать для проверки Контроллеру, который проверяет аутентичность Клиента и сообщает о своем решении Серверу, см. рис. 2.199.

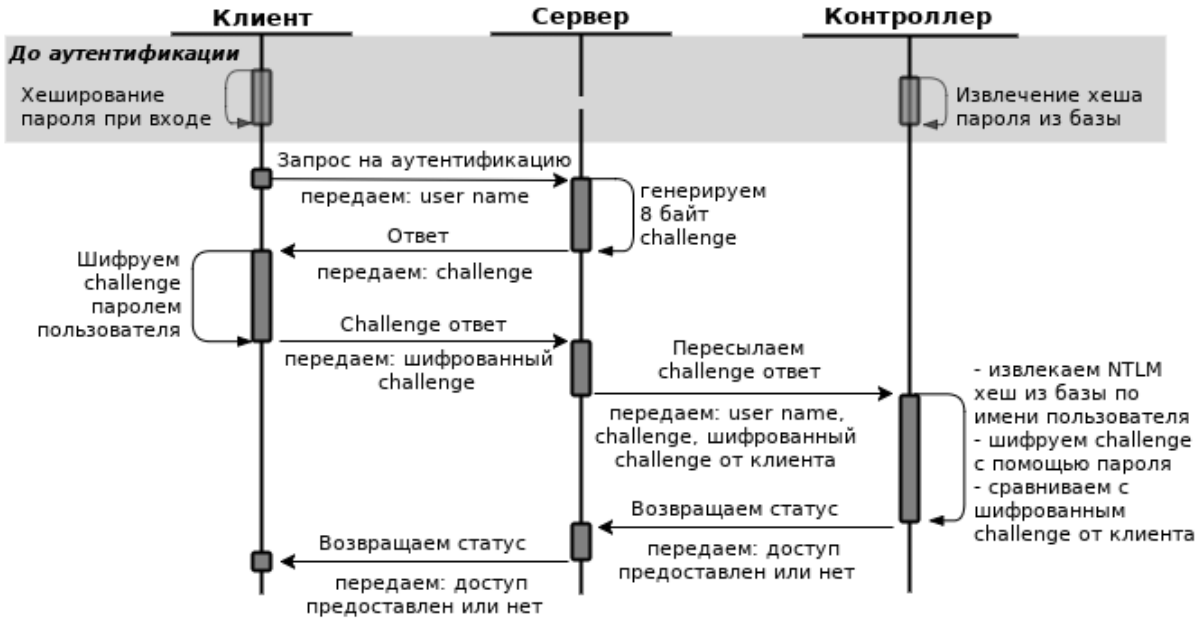


рис. 2.199 – Поток данных при NTLM-аутентификации

Таким образом, в вопросах аутентификации пользователей файловый сервер должен действовать как участник домена, у которого нет прямого доступа к учетным данным. Именно так и работают режимы «security = domain» и «security = ads». Однако на файловом сервере ALD Pro устанавливается библиотека libwbclient от SSSD, которая не поддерживает перенаправление NTLM-запросов, поэтому данный механизм работать не будет.

Однако на файловом сервере ALD Pro служба smbd по умолчанию работает в режиме BDC NT4 контроллера домена (параметр «security» равен «user») и чисто технически может самостоятельно выполнять аутентификацию пользователей по паролю без проксирования запросов, извлекая ipaNTHash из LDAP-каталога с помощью модуля ipasam, см. параметр passdb backend в файле smb.conf:

```
security = user
passdb backend = ipasam:ldap://dc-1.ald.company.lan:389
```

Но из соображений безопасности учетная запись файлового сервера по умолчанию не имеет доступа к атрибуту ipaNThash, то есть модуль ipasam на файловом сервере участвует только в преобразовании идентификаторов. Если вам нужна NTLM-аутентификация и вы готовы администрировать файловый сервер теми же сотрудниками, кто отвечает за домен, то вам нужно назначить учетной записи сервера роль «ALDPRO - CIFS server»:

```
admin@dc-1:~$ ipa role-add-member "ALDPRO - CIFS server" --services=cifs/file-1.ald.
↪company.lan
...
  Имя роли: ALDPRO - CIFS server
  Описание: Служебная роль для системных сервисов.
```

(continues on next page)

(продолжение с предыдущей страницы)

```
  Privileges: CIFS server privilege
  Службы-участники: cifs/file-1.ald.company.lan@ALD.COMPANY.LAN
----------------------------------
Количество добавленных участников 1
----------------------------------
```

После назначения роли «ALDPRO - CIFS server» файловый сервер сможет извлекать NTLM-хеш и выполнять проверку аутентичности пользователей по протоколу NTLM. Подключение к LDAP-каталогу модуль ipasam выполняет с использованием безопасной Kerberos-аутентификации, а трафик дополнительно шифруется механизмом SASL GSS-API Integrity.

И обратите внимание, что при выполнении NTLM-аутентификации файловый менеджер fly-fm предложит сохранить пароль в связке ключей gnome-keyring, которую не нужно путать со связкой ключей в ядре Linux, где лежат Kerberos-ключи.

Благодаря gnome-keyring пользователь может входить на файловый сервер по паролю без его повторного входа даже после перезагрузки машины. Физически пароль будет находиться в файле ~/.local/share/keyrings/login.keyring, поэтому для очистки кэша вы можете просто удалить этот файл или дополнительно установить удобный диспетчер паролей seahorse, см. рис. 2.200.

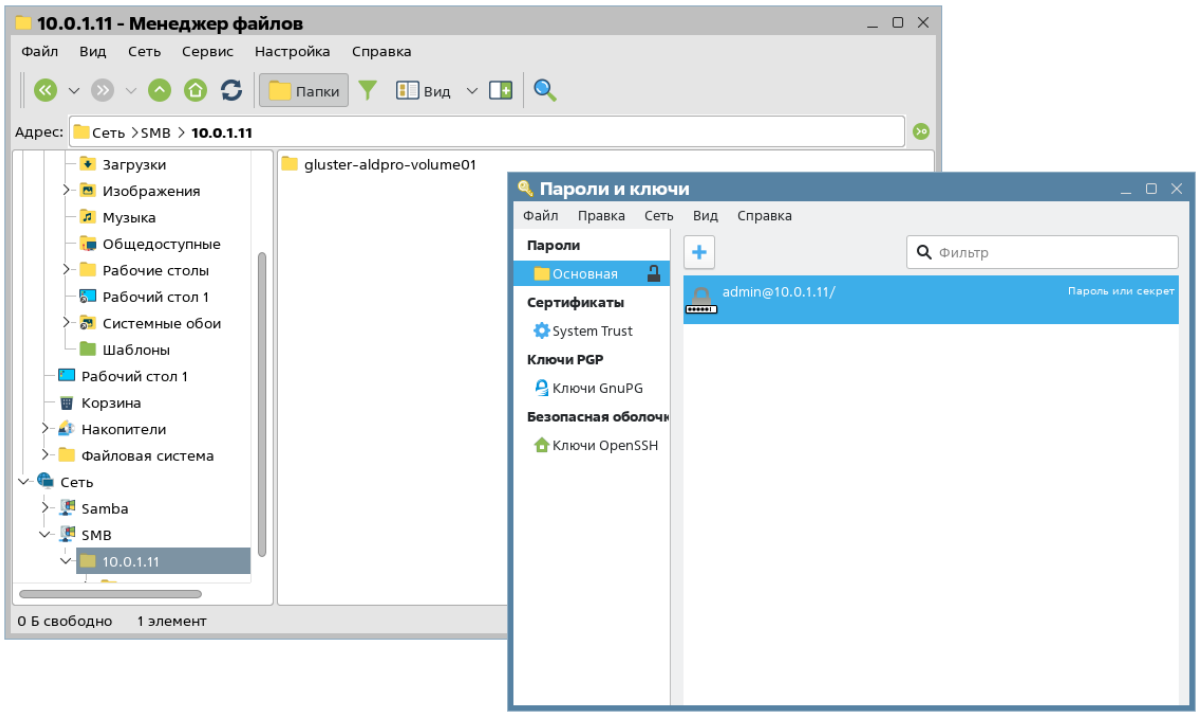


рис. 2.200 – Просмотр NTLM-пароля в связке ключей Seahorse

7.3.5 Авторизация на файловом сервере Samba

Авторизация – это предоставление субъектам информационной системы доступа к ресурсам на основе разрешений. Такую модель называют дискреционной или избирательной (от англ. discretionary access control, DAC).

Для того чтобы одних пользователей можно было отличить от других, им назначаются уникальные имена и идентификаторы безопасности. Для авторизации по протоколу SMB используются идентификаторы SID в стиле Windows, а внутри Linux-систем мы работаем с POSIX-идентификаторами.

Мы уже знаем два существенных отличия между этими идентификаторами, которые оказывают значительное влияние на работу с общими файловыми ресурсами:

1. В идентификаторе Windows S-1-5-21-1491017894-2377586105-2170988794-500 последнее значение 500 идентифицирует субъект относительно домена, а предшествующие ему три числа 1491017894-2377586105-2170988794 предназначены для идентификации домена и совпадают для всех пользователей и групп. В POSIX мы имеем дело с обычными целыми числами от 0 до 2^32 (4 294 967 296), т.е. нет уникальной доменной части, поэтому для идентификации доменов приходится выделять в этом множестве отдельные диапазоны.

2. В системе Windows пользователи и группы находятся в одном пространстве идентификаторов SID, а в модели POSIX – это два разных непересекающихся множества: идентификаторы пользователей называют UID (User Identifier), а идентификаторы групп – GID (Group Identifier), поэтому пользователь root с идентификатором 0 и одноименная группа root с идентификатором 0 — это совершенно разные субъекты безопасности, что значительно затрудняет преобразование идентификаторов для обеспечения совместимости.

Извлечение авторизационной информации об участии пользователей в группах и преобразование идентификаторов могут выполнять следующие компоненты:

- ipasam.so – модуль бэкенда базы данных, который обеспечивает работу Samba в режиме контроллера с извлечением данных из LDAP-каталога. Конкретно этот модуль был разработан специально для FreeIPA и учитывает особенности ее схемы данных.

- cifs_idmap_sss.so – модуль для преобразования идентификаторов от sssd, который подключается, если в smb.conf для опции idmap config используется backend = sss.

- libwbclient.so – модуль службы winbind, который должен использоваться для преобразования идентификаторов сторонних доменов. На компьютерах, где установлен freeipa-client, он заменен на модуль от sssd, в котором нет поддержки перенаправления NTLM-запросов на контроллер.

- libnss_sss.so – модуль для NSS от SSSD, используется при обращении к функциям glibc

операционной системой для извлечения информации о пользователях и группах, когда в nsswitch.conf источником базы указан sss.

7.3.5.1 Авторизация при подключении по Kerberos

Служба smbd имеет доступ к файлу /etc/samba/samba.keytab, что позволяет ей расшифровывать Kerberos-билеты и извлекать из них PAC-сертификат с информацией об участии пользователя в группах. В процессе авторизации участвуют службы sssd и winbind.

До версии 2.4.0 в журналах winbind вы могли видеть, что PAC-сертификат успешно раскодирован, но на этапе преобразования SID-идентификаторов в POSIX возникали ошибки. Поэтому в конечном итоге информацию об участии пользователя в группах служба smbd брала через SSSD.

Такие ошибки возникают, потому что файловый сервер по умолчанию ставится как BDC-контроллер, поэтому SID хоста совпадает с SID домена, а в нашем случае нужно, чтобы SID у хоста был другим. Назначить новое значение можно, например, следующей командой:

```
net setlocalsid "S-1-5-21-$(shuf -i 1000000000-4294967295 -n 1)-$(shuf -i 1000000000-
↪4294967295 -n 1)-
$(shuf -i 1000000000-4294967295 -n 1)"
```

После этого можно будет отключить passdb backend, установить режим ADS и перезагрузить службы сервера:

```
cat /etc/samba/smb.conf
...
#passdb backend = ...
security = ADS
...
```

С версии 2.4.0 указанные настройки применяются по умолчанию, и авторизация работает через PAC-сертификат.

7.3.5.2 Авторизация при подключении по NTLM

В случае NTLM-аутентификации служба winbind не требуется, информацию об участии пользователей в группах служба smbd получает через SSSD, см. параметр ipa_server в файле /etc/sssd/sssd.conf.

7.3.6 Особенности настройки прав доступа к файлам в Samba

В случае файлового сервера Samba ресурсами являются общие файлы, в роли субъектов выступают пользователи и группы, а разрешения, так же как для файлового сервера Windows, могут быть

заданы на двух уровнях:

- Сначала применяются разрешения на уровне SMB-подключения, которые задаются на вкладке «Share Permissions».
- Затем дополнительно накладываются разрешения, установленные списками доступа файловой системы, которые задаются на вкладке «Security».

  Если на уровне SMB-подключения пользователю предоставлены права только на чтение, то, имея даже «полный доступ» на уровне файловой системы, он все равно не сможет редактировать файлы в этой общей папке.

7.3.6.1 Разрешения на уровне SMB-подключения

Интерфейс Windows для настройки SMB-разрешений отображает шесть флажков, и может показаться, что их допустимо устанавливать в любом сочетании, см. рис. 2.201. Однако, если вы кликните по флажку «Allow Full Control», то увидите, что флажки «Allow Change» и «Allow Read» будут включены автоматически, то есть мы имеем дело фактически с одним выпадающим списком из трех значений, каждое из которых вбирает в себя разрешения предыдущих уровней.

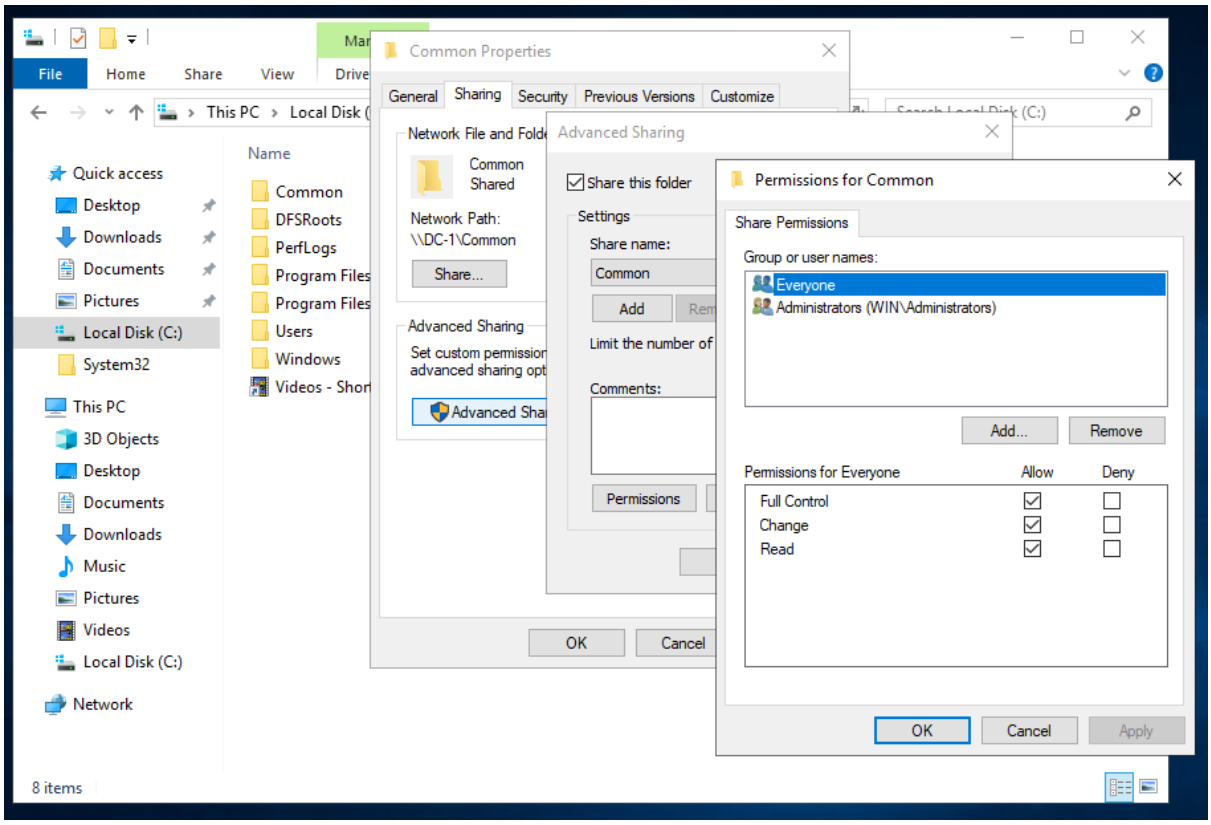


рис. 2.201 – Настройка SMB-разрешений в Windows

С флажками Deny еще интереснее: они точно так же, как флажки Allow, представляют собой один

выпадающий список с тремя значениями, но при этом пользователь полностью потеряет право на подключение к общему ресурсу вне зависимости от того, какой из флажков вы установите: «Deny Read», «Deny Change» или «Deny Full Control».

В системе Windows разрешения SMB предоставляют следующие права:

Разрешение	Описание
Read (Чтение)	Дает право: • видеть имена файлов и папок; • читать содержимое и атрибуты файлов; • иметь доступ к вложенным папкам общего ресурса; • запускать исполняемые файлы.
Change (Изменение)	Дополняет предыдущее разрешение следующими правами: • создавать и удалять файлы и вложенные папки; • изменять содержимое и атрибуты файлов и папок.
Full Control (Полный доступ)	Дополняет предыдущее разрешение правами: • становиться владельцами файлов и папок; • изменять списки доступа ACL файловой системы. Разрешение Full Control имеет смысл только в том случае, если общая папка расположена на томе NTFS, который поддерживает списки доступа.

В службе Samba для настройки SMB-разрешений предназначены две группы параметров:

- одна группа определяет права на подключение;
- вторая отвечает за уровень разрешений в рамках установленного подключения.

Права на подключение к общему ресурсу задаются двумя параметрами «valid users» и «invalid users» по алгоритму, который представлен на рисунке рис. 2.202.

табл. 2.13 – Права на подключение к общему ресурсу

Параметр	Описание
valid users	Позволяет ограничить доступ к общей папке указанному списку пользователей и групп пользователей. Если параметр не задан, то по умолчанию доступ предоставляется всем пользователям. Служба Samba может работать с разными источниками групп, порядок обращения к которым задается специальными символами в начале имени группы: • символ «+» указывает, что это UNIX группа; • символ «&» указывает, что это NIS группа; • символы «+&» указывают, что сначала нужно выполнить поиск по UNIX базе, а затем обратиться к базе NIS; • символы «&+» меняют предыдущий порядок на противоположный; • символ «@» является краткой записью «&+», то есть указывает, что сначала нужно выполнить поиск по NIS базе, а затем обратиться к UNIX базе. Технология сетевых групп NIS уже устарела, но служба каталога FreeIPA ее поддерживает, и при установке клиента freeipa в конфигурационном файле `/etc/nsswitch.conf` задается значение «nis sss» для базы netgroup. Поэтому, не смотря на то, что в интерфейсе ALD Pro управление сетевыми группами не представлено, на файловых серверах мы все же используем символ @ в начале имен групп для обеспечения обратной совместимости.
invalid users	Позволяет запретить доступ к общей папке указанному списку пользователей и групп пользователей. Если пользователь входит сразу в оба списка, и в «valid users» и в «invalid users», то доступ к общей папке этому пользователю будет запрещен. По умолчанию параметр не задан.

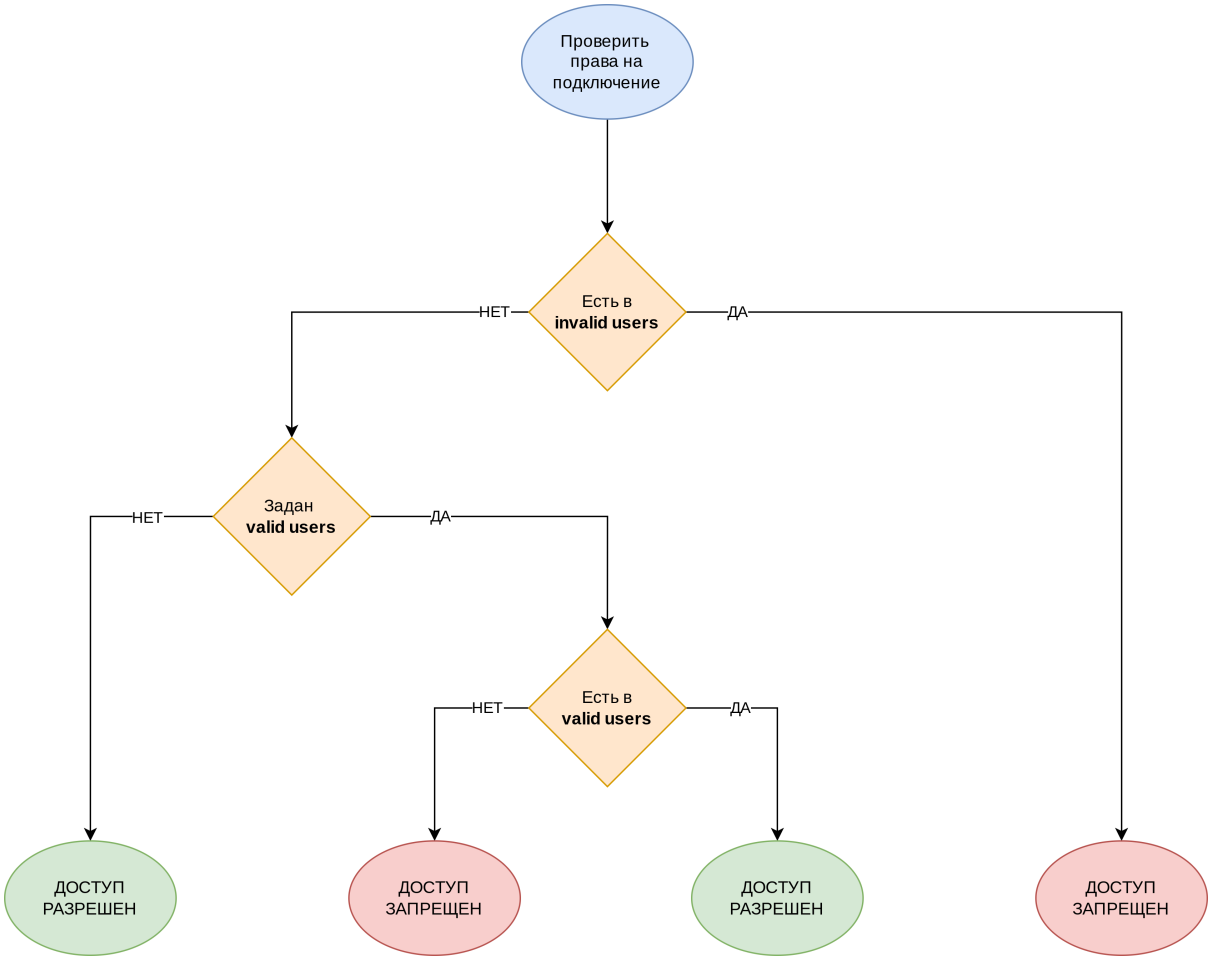


рис. 2.202 – Алгоритм определения прав на подключение к общему ресурсу

Если подключение разрешено, то доступ может быть либо на чтение, либо на запись. Права доступа определяются параметрами «read list», «write list», «read only» и «writable» по алгоритму, который представлен на рисунке рис. 2.203.

табл. 2.14 – Параметры прав доступа

Параметр	Описание
read list	Определяет список пользователей и групп пользователей, доступ которых будет ограничен правами на чтение. Параметр «read list» имеет смысл, если пользователям по умолчанию предоставлены права на запись с помощью параметров «writable = YES» или «read only = NO». Параметр «read list» не действует, если пользователя явно включили в «write list».
write list	Определяет список пользователей и групп пользователей, которым будет предоставлен доступ «на изменение», т.е. «чтение» + «запись». Если пользователь через разные группы входит сразу и в «read list» и в «write list», то этому пользователю будут предоставлены права на изменение.
read only	Определяет уровень доступа «по умолчанию» для пользователей, которые не включены в списки «read list» и «wrtie list». Если параметр не задан, предполагается, что «read only = yes», поэтому пользователи будут ограничены правами на «чтение».
writable или writeable	Параметры *writable* и *writeable* являются синонимами, допустимо использовать любое написание. Оба параметра являются инверсией параметру read only, поэтому не имеет разницы, что вы укажете «read only = yes» или «writable = no». Если параметры заданы несколько раз, то силу имеет последнее из указанных значений.

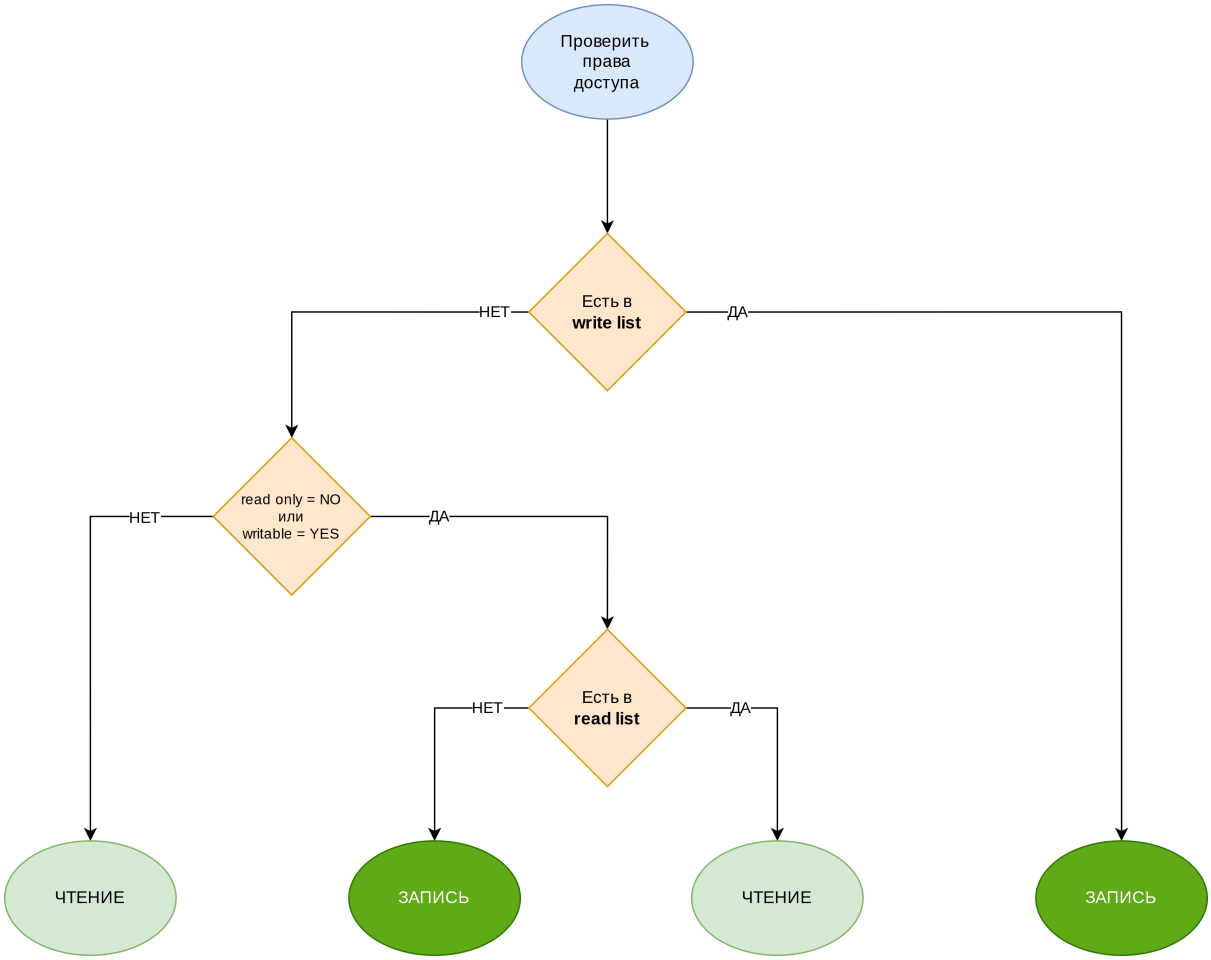


рис. 2.203 – Алгоритм определения прав доступа к общему ресурсу

Учитывая особенность алгоритма, для разграничения прав доступа используют одну из следующих стратегий:

1. Разрешают всем пользователям «чтение» и далее расширяют кому-то из них права до уровня «изменение» с помощью «write list».

```
writable = no
write list = editor_user_1 editor_user_2
```

2. Разрешают всем пользователям «запись» по умолчанию и далее урезают права для определенных пользователей с помощью «read list» (так сделано в ALD Pro).

```
writable = yes
read list = reader_user_1 reader_user_2
```

Схемы предоставления прав доступа Windows и Samba немного отличаются, поэтому можно привести только частичное соответствие:

табл. 2.15 – Соответствия прав доступа Windows и Samba

Права Windows	Права Samba
Allow Read	Права доступа «Чтение». Субъект нужно включить в списки: • «valid users» • «read list»
Allow Write	Нет полного соответствия, т.к. добавление пользователя в «write list» дает ему не только права на запись, но и права на изменение прав доступа к объектам, если он является их владельцем.
Allow Full Control	«Изменение и назначение прав». Этот уровень предполагает, что у пользователя есть не только права на запись, но и права на изменение прав доступа к объектам. Его можно назвать «Полный доступ в соответствии с ACL файловой системы». Субъект нужно включить в списки: • «valid users» • «write list» Для информации. Изменение прав доступа будет возможно только из приложения `smbclient` при подключении по протоколу NT1 и через файловый менеджер Windows. В файловом менеджере fly права доступа можно менять только при локальном обращении к файлам, а если подключать папку через SMB, то эта функция будет недоступна.
Нет аналога	Административный доступ, сейчас в интерфейсе ALD Pro называется «полный доступ». Субъект нужно включить в списки: • «valid users» • «write list» • «admin users» В этом случае пользователь будет действовать в общей папке от имени root в обход ACL файловой системы. Для информации. В настоящий момент ALD Pro не добавляет таких пользователей в список «write list», поэтому пользователь будет ограничен правами на чтение, если он одновременно является участником группы, которой дан такой уровень доступа. Добавьте пользователя в список «write list» вручную.
Deny Read, Deny Write или Deny Full Control	Доступ запрещен. Субъект нужно включить в список «invalid users».

7.3.6.2 Разрешения на уровне списков доступа файловой системы

Служба Samba работает от имени root, поэтому имеет доступ ко всем объектам файловой системы, но при обработке запросов от пользователей самостоятельно выполняет проверку прав, имитируя одну из доступных моделей безопасности. Вы можете выбрать любой из представленных режимов работы в зависимости от поставленной задачи:

- **Из коробки** Samba максимально приближается к тому, как Linux работает с моделями RWX и UGO. Несовместимые атрибуты DOS, с помощью которых в Windows файл отмечается как «архивный», «скрытый» или «системный», можно отбрасывать (что снижает совместимость с Windows), сопоставлять со стандартными битами Execute (что снижает совместимость с Linux) или хранить в расширенных атрибутах файлов, если это поддерживается файловой системой.

- Плагин **acl_xattr** позволяет значительно приблизиться к имитации Windows ACL, сохраняя часть информации в расширенных атрибутах, но продолжая активно использовать POSIX ACL, что дает возможность сохранить пользователям Linux прямой доступ к файлам.

- Плагин **nfs4acl_xattr** переводит файловый сервер в режим максимальной совместимости с Windows ACL, сохраняя всю необходимую информацию в расширенных атрибутах файлов, но это исключает возможность работы с файлами напрямую.

Для подключения плагина в настройках сетевой папки нужно задать параметр vfs objects, например:

```
[test_share]
...
vfs objects = acl_xattr
```

Модель безопасности Linux

С моделью безопасности Linux, включающей как права «пользователь – группа – остальные», так и расширенную модель, использующую списки прав доступа POSIX ACL, мы подробно познакомились в курсе «Администратор рабочих мест под управлением ОС Astra Linux», поэтому мы не будем здесь подробно рассматривать этот аспект.

Вспомним только как работает алгоритм определения прав доступа к объекту файловой системы Linux с учетом ACL, см. рис. 2.204:

- Если пользователь является владельцем, он получает права пользователя, и процедура завершается.

- Если пользователь не является владельцем, но включен в ACL, то он получает права, ука-

занные в списке доступа с учетом маски, и процедура завершается. Маска — это специальное правило POSIX ACL, которое определяет максимальные права доступа.

- Если пользователь не является владельцем и не включен в список ACL, то проверка переходит на группы. Если пользователь является участником группы-владельца или одной из групп ACL, то он получает сумму этих прав с учетом маски, и процедура завершается.
- Если пользователь не попал ни в одну из категорий, то он получает права, предназначенные для всех остальных пользователей.

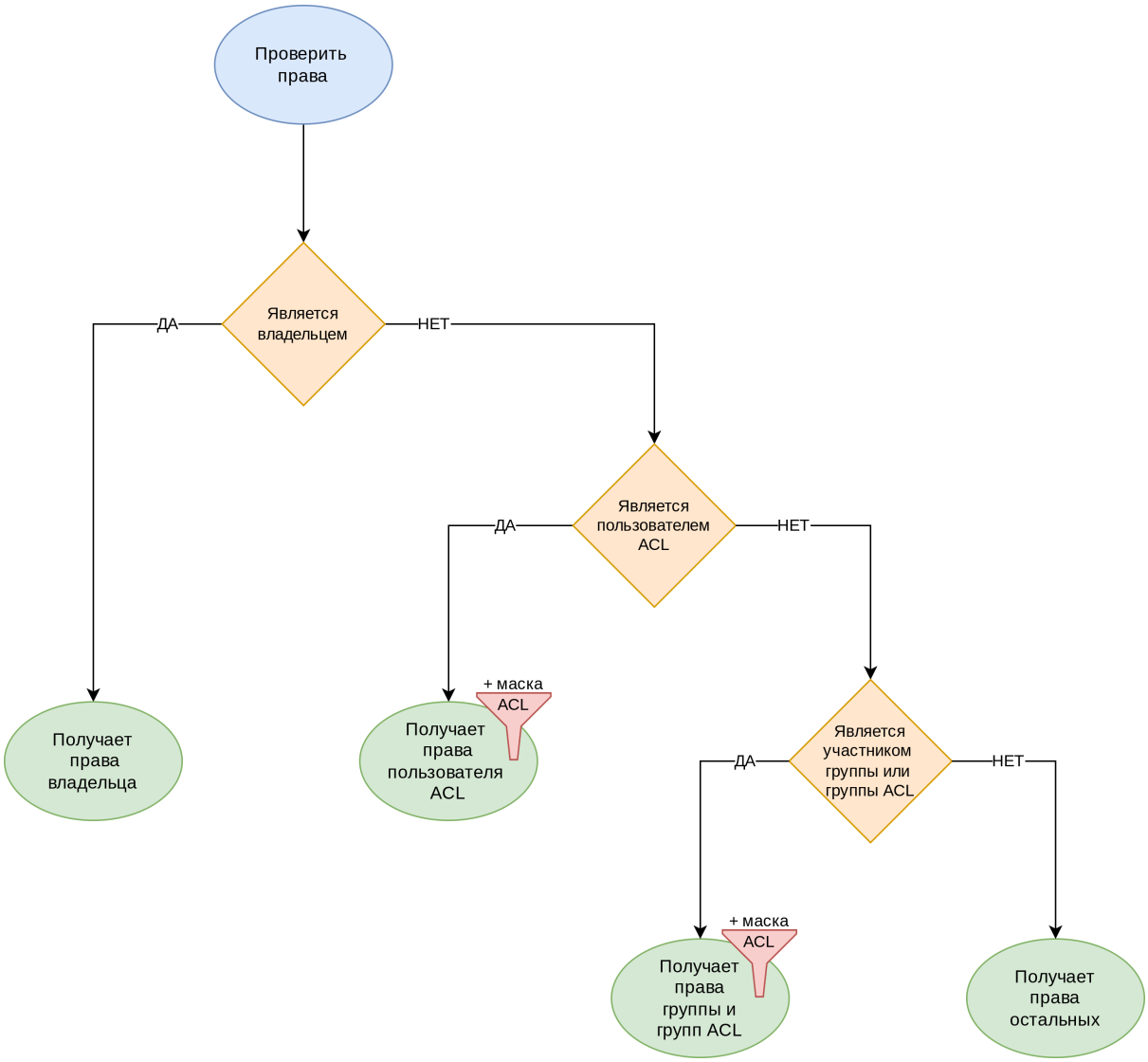


рис. 2.204 – Алгоритм определения прав доступа к объекту файловой системы Linux с учетом ACL

Специфика работы с правами доступа при подключении SMB-ресурса

Общий ресурс можно подключить, например, средствами FUSE в пространстве пользователя. Если требуется временный доступ, то в адресной строке можно просто ввести:

`smb://file-1.ald.company.lan/test_share`

Где:

- **smb** — указывает на протокол SMB
- **file-1.ald.company.lan** — fqdn файлового сервера
- **test_share** — имя общего ресурса для подключения

Если требуется постоянный доступ к ресурсу, то в меню *Сеть → Новое сетевое место...* можно создать именованное подключение к сетевому ресурсу. В появившемся окне нужно задать имя ресурса и его адрес, как было указано выше.

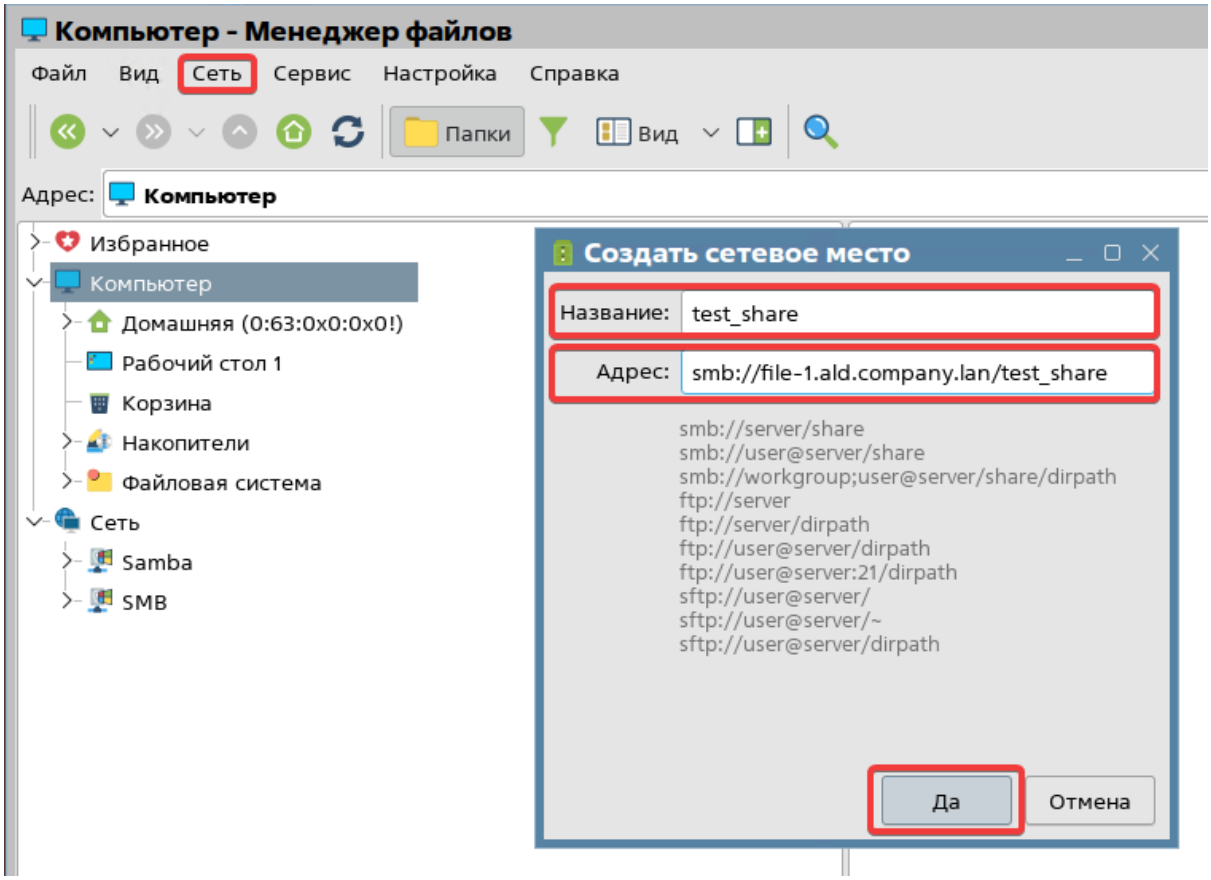


рис. 2.205 – Создание сетевого подключения

Еще один способ – это выполнить монтирование сетевого ресурса из-под привилегированной учетной записи с помощью библиотеки cifs-utils.

```
admin@dc-1:~$ sudo apt install cifs-utils
admin@dc-1:~$ sudo mount.cifs //file-1.ald.company.lan/test-share /home/admin/test-
↪share/ -o sec=krb5i,cruid=$USER,uid=$UID,gid=$GROUPS,file_mode=0770,dir_mode=0770
```

Где:

- `sudo` – необходимо, чтобы команда была выполнена с повышенными привилегиями, но в переменных $USER, $UID и $GORUPS были данные из переменных окружения текущего пользователя.
- `//file-1.ald.company.lan/test-share` — имя к сетевому ресурсу.
- `/home/admin/test-share` — точка монтирования, этот каталог должен существовать на момент подключения сетевого ресурса.
- `-o` — ключ для определения параметров (options) подключения.
- `sec=krb5i` — определяет использование Kerberos-аутентификации с проверкой целостности пакетов.
- `cruid=$USER` — определяет владельца учетных данных Kerberos (credentials uid).
- `uid=$UID` — определяет, какой пользователь должен быть указан владельцем для объектов общего сетевого ресурса.
- `gid=$GROUPS` — определяет, какая группа должна быть указана владельцем для объектов общего сетевого ресурса.
- `file_mode=0770` — определяет, какие права доступа должны быть выставлены для файлов общего сетевого ресурса.
- `dir_mode=0770` — определяет, какие права доступа должны быть выставлены для каталогов общего сетевого ресурса.

После перезагрузки компьютера монтирование не сохранится. Если требуется постоянный доступ к ресурсу, монтирование можно выполнить с помощью fstab.

Вне зависимости от способа подключения общего ресурса работа с ним будет существенно отличаться от работы с файлами в проводнике Windows или при локальном обращении к файлам, т.к. вы будете видеть не действительные права доступа и владельцев, как они заданы на файловом сервере, а те значения, под которыми этот общий ресурс монтирован на текущем хосте.

При использовании FUSE ресурс монтируется в пользовательском пространстве, поэтому владельцем будет текущий пользователь и его первичная группа, на каталоги будут права «u=rwx,g=rx,o=rx», а на файлы «u=rwx,g=r,o=r». Именно поэтому пользователь сможет запустить любое приложение из общего ресурса, если на сервере у него есть права на чтение этого файла.

При монтировании ресурса с помощью cifs-utils параметры будут определяться значениями опций «uid», «gid», «file_mode» и «dir_mode». Если при подключении к ресурсу использовать Kerberos-аутентификацию, то настоятельно рекомендуется предоставлять права к файлам только владельцу

этих учетных данных. Вы также можете выбрать, нужно ли ставить флаг execute и разрешать запуск приложений из монтируемой папки, или это будет излишне.

Модель безопасности из коробки

Из коробки Samba реализует проверку прав доступа к объектам файловой системы в стиле Linux.

Файловый менеджер fly-fm позволяет работать с расширенными ACL, но при добавлении новых пользователей и групп в списке отображаются только локальные пользователи файлового сервера. Чтобы добавить в список доменных пользователей, в настройках sssd.conf нужно включить опцию «enumerate = true»:

```
admin@dc-1:~$ sudo cat /etc/sssd/sssd.conf
...
[domain/ald.company.lan]
enumerate = true
...
```

Однако опция enumerate подойдет, только когда в домене несколько сотен пользователей. В больших инсталляциях она вызовет деградацию в работе службы SSSD, поэтому в этом случае для добавления субъектов рекомендуется использовать утилиту setfacl из командной строки. В ближайших релизах появится утилита, которая позволит добавлять пользователей и группы в ACL из графического интерфейса.

Рассмотрим еще несколько важных настроек. Если вы хотите предоставлять пользователям доступ напрямую, а не только по сети, то обязательно нужно отключить сопоставление DOS-атрибута «Archive» на флаг Execute владельца файла, иначе при создании файлов они будут иметь права «rwx r– r–». На файловых серверах ALD Pro атрибуты DOS сохраняются в расширенных атрибутах, поэтому сопоставление не требуется и только мешает.

```
[test_share]
...
map archive = no
```

Если общий ресурс располагается на диске, который смонтирован без поддержки расширенных атрибутов, то Samba может хранить DOS-атрибуты «Read-only», «Hidden», «Archive» и «System» в обычных битах доступа Linux, как показано на рисунке рис. 2.206.

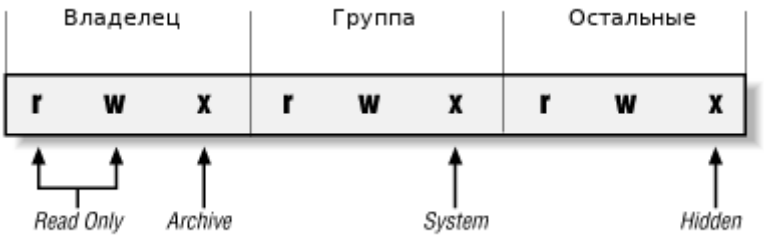


рис. 2.206 – Сопоставление DOS-атрибутов на разрешения Linux

Для того чтобы при создании новых файлов на сервере назначать им права доступа по умолчанию, можно воспользоваться параметрами «create mask» и «directory mask» или параметрами «force create mode» и «force directory mode» соответственно. Параметры «force» предпочтительнее, так как позволяют задавать в том числе флаг «Execute».

```
[test_share]
...
force create mode = 0666
force directory mode = 0777
```

Фиксированные права доступа выставляются на общую папку в целом и не могут быть переопределены для вложенных каталогов, поэтому в некоторых случаях удобнее будет использовать механизм наследования разрешений, который можно включить параметром «inherit permissions».

```
[test_share]
...
inherit permissions = yes
```

Если для параметра «inherit permissions» установлено значение «yes», то параметры «create mask», «directory mask», «force create mode» и «force directory mode» игнорируются. Новые каталоги наследуют те же разрешения, что и родительские каталоги. Новые файлы наследуют от родителя только биты чтения и записи, а биты Execute устанавливаться не будут из соображений безопасности, что подходит для обычных рабочих каталогов.

Модель безопасности ACL XATTR

Плагин acl_xattr имитирует работу Windows ACL, используя расширенные атрибуты в связке с базовыми возможностями POSIX ACL.

```
[test_share]
...
vfs objects = acl_xattr
```

Функции Windows ACL будут немного ограничены, но при этом останется частичная совместимость с Linux, поэтому с некоторыми оговорками пользователям можно давать прямой доступ к файлам

при выполнении интерактивного входа в графический интерфейс или подключении к серверу по протоколам SSH, XRDP и др.

Администратор сможет управлять правами с помощью файлового менеджера Windows при подключении по SMB и с помощью файлового менеджера `fly-fm` при прямом обращении к файлам с сервера (не через SMB).

Модель безопасности NFS4 ACLs

Плагин nfs4acl_xattr максимально полно имитирует работу Windows ACL, сохраняя списки доступа NFS4 в виде бинарных объектов в расширенных атрибутах файлов (EAs/xattrs). Такой подход позволяет добиться максимальной совместимости с Windows, но исключает возможность предоставления пользователям прямого доступа к файлам, используя стандартные механизмы проверки прав доступа средствами Linux. Настраивать права доступа к таким общим папкам можно будет только из файлового менеджера Windows.

7.4 Практика и тестирование

7.4.1 Практическая работа: Модуль 7. Дополнительные подсистемы ALD Pro

См. также *Требования, правила и цели выполнения практической работы*

7.4.1.1 Практические задания

1. Установите подсистему общего доступа к файлам.
2. Выдайте разрешение на чтение общей папки «shared» группе «moscow-office-employee-it».
3. Создайте общую папку «test_share».
4. Измените разрешения на папку «shared» так, чтобы пользователь iivanov мог редактировать ее содержимое, а остальные пользователи в подразделении «Московский офис» могли только читать. Убедитесь, что все работает, как ожидалось.
5. Выдайте разрешение на общую папку «test_share» таким образом, чтобы
   a. все пользователи в подразделении «Московский офис» могли только читать содержимое папки;
   b. пользователь ppetrov мог бы его редактировать, но не имел бы на неё прав суперпользователя;

c. участники группы «moscow-office-employee-it» имели бы административные права на общую папку и её содержимое.

7.4.2 Тест: Модуль 7. Дополнительные подсистемы ALD Pro

Пожалуйста, ответьте на несколько вопросов, чтобы закрепить полученные знания. Правильные ответы для самопроверки вы сможете найти в конце списка. Обратите внимание, что у некоторых вопросов может быть несколько верных вариантов ответов.

7.4.2.1 Вопросы

Вопрос 1. Какими правами на общие папки можно управлять с портала управления?

a. NTFS ACL

b. POSIX ACL

c. SMB

d. UGO (Владелец – Группа – Все остальные)

Вопрос 2. Какие SMB-права получит пользователь, если на общей папке настроены опции:

- «writable = yes»
- Пользователя нет в «invalid users»
- «valid users» не заданы
- пользователь состоит в списке «read list»?

a. У пользователя не будет никаких прав

b. У пользователя будут права на чтение

c. У пользователя будут права на запись

d. У пользователя будут полные права

Вопрос 3. Где хранится конфигурация файлового сервера ALD Pro?

a. «cn=smb,cn=file,cn=aldpro,cn=etc,{basedn}»

b. «cn=smb,cn=samba,cn=aldpro,cn=etc,{basedn}»

c. /etc/samba/smb.conf

d. /opt/rbta/file/smb.conf

Вопрос 4. Как можно отследить процесс развертывания подсистемы ALD Pro?

a. `sudo tail -f /var/log/apt/term.log /var/log/salt/minion`

b. `sudo tail -f /var/log/aldpro/subsystems/install`

c. `sudo journalctl -all`

d. `sudo journalctl -f`

Ответы на вопросы

1. c
2. b
3. c
4. a, d

7.5 Заключение

В этом модуле мы познакомились с подсистемой общего доступа к файлам, узнали, как выполнять ее установку и базовую настройку. Далее мы продолжим знакомство с подсистемами, а еще впереди у нас установка резервного контроллера!

7.6 Дополнительные источники информации

- The Official Samba 3.2.x HOWTO and Reference Guide, Jelmer R. Vernooij, John H. Terpstra, and Gerald (Jerry) Carter (крайней версией Samba является 4, но в ней добавили только функции Active Directory, а код 3-й версии, который отвечал за общий доступ к файлам, был взят без существенных изменений, поэтому материалы предлагаемой книги все еще сохраняют свою актуальность).

7.7 Обратная связь

Если остались вопросы, то их всегда можно задать в специальной теме.

Модуль 8. Управление топологией в домене

8.1 Введение

Из этого модуля вы узнаете о том, как обеспечить отказоустойчивость и горизонтальное масштабирование системы за счет создания кластера из нескольких контроллеров домена. Мы рассмотрим также особенности алгоритма репликации и способы решения конфликтов, которые могут возникать при редактировании записей на нескольких контроллерах сразу в условиях нарушения сетевой связанности.

8.2 Повышение роли сервера до контроллера домена

Для обеспечения надежности и отказоустойчивости рекомендуется устанавливать не менее двух контроллеров домена.

С помощью увеличения числа контроллеров добиваются так же горизонтального масштабирования, а в крупных инфраструктурах дополнительные контроллеры могут устанавливаться в различных сегментах сети, чтобы ускорить обработку запросов от ближайших к ним пользователей — эта технология в Active Directory называется сайтами, а в FreeIPA ее называют локациями, что более точно отражает суть решаемой задачи.

Процедуру, в ходе которого сервер делают контроллером домена, называют повышением роли сервера или продвижением сервера до контроллера домена (*англ. promoting*). В системе ALD Pro эта операция выполняется в два шага:

- Установка ОС и ввод машины в домен.
- Дополнительные настройки сервера и повышение роли до контроллера домена через портал управления.

Не будем подробно останавливаться на вводе хоста в домен, отметим только, что к серверу предъявляются те же требования, как и к основному контроллеру домена:

- Сервер должен быть установлен в режиме «Смоленск», имя хоста dc-2
- Служба NetworkManager отключена, настройка сетевого интерфейса выполнена с использованием файлов `/etc/network/interfaces` и `/etc/resolv.conf`
- Настройки сетевого интерфейса в `/etc/network/interfaces`
  - **IP-адрес**: *10.0.1.12*
  - **Маска**: *255.255.255.0*

- **Шлюз**: *10.0.1.1*

- В настройках `/etc/resolv.conf` в качестве DNS-сервера указан IP-адрес первого контроллера домена, в нашем случае это **10.0.1.11** и DNS-суффикс **ald.company.lan** (параметр search)

- В файле `/etc/hosts` добавлена статичная запись «10.0.1.12 dc-2.ald.company.lan dc-2», а строка с адресом 127.0.1.1 закомментирована.

8.2.1 Дополнительные настройки

8.2.1.1 Файл resolv.conf

На момент установки резервного контроллера разрешение имен должно выполняться через первый контроллер домена, поэтому содержимое файла `/etc/resolv.conf` должно быть следующим:

```
search ald.company.lan
nameserver 10.0.1.11
```

Примечание: До версии 2.4.0 перед установкой резервного контроллера домена нужно было заблокировать файл resolv.conf от изменений с помощью команды chattr +i. С версии 2.4.0 дополнительных действий с файлом выполнять не требуется, а после установки резервного контроллера домена в качестве nameserver в этом файле будет автоматически установлен IP-адрес 127.0.0.1.

8.2.1.2 Параметр gather_job_timeout

Примечание: С версии ALD Pro 2.4.0 данный параметр потерял свою актуальность в связи с переходом на standalone Salt Minion. При использовании более старых версий рекомендуется ознакомиться с информацией ниже.

Если контроллер домена установлен на медленных дисках или испытывает высокую нагрузку в момент установки подсистем, Salt Master может не успевать вычитывать результаты заданий из базы данных в установленные таймауты и в журнале автоматизации будет появляться ошибка «Run failed on the minions».

Эта ошибка не является критичной, т.к. подсистема фактически установлена и корректно работает, но чтобы ее избежать в конфигурационном файле /etc/salt/master.d/master.conf рекомендуется увеличить значение gather_job_timeout до 100-300 секунд.

8.2.1.3 Протокол IPv6

Так же на момент создания реплики один из компонентов FreeIPA проверяет в системе функционирование протокола IPv6. Отсутствие протокола приводит к фатальной ошибке и невозможности создания резервного контроллера. Для прохождения проверки необходимо, чтобы при старте операционной системы был активен модуль ядра, отвечающий за этот протокол. По умолчанию протокол IPv6 включён. Чтобы убедиться, что модуль ядра не выключен, необходимо в файле `/etc/default/grub` проверить отсутствие параметра «ipv6.disable=1», а при наличии убрать его, обновить загрузчик командой `sudo update-grub` и перезагрузить компьютер.

Примечание: При этом, работа протокола и наличие ip адресов протокола ipv6 на сетевых интерфейсах не требуется. На всех сетевых интерфейсах протокол может быть выключен через параметр sysctl `net.ipv6.conf.all.disable_ipv6 = 1`.

Подробнее про включение/отключение протокола IPv6 можно прочитать на нашей wiki.

8.3 Продвижение сервера до контроллера домена с портала управления

Продвижение резервного контроля осуществляется с портала управления на странице *Управление доменом → Сайты и службы → Контроллеры домена*. Нажмите кнопку *Новый контроллер домена*, выберите из списка сервер dc-2.ald.company.lan, при необходимости назначьте роль глобального каталога и роль PKIProxy, остальные параметры заполните, как показано на рисунке рис. 2.207.

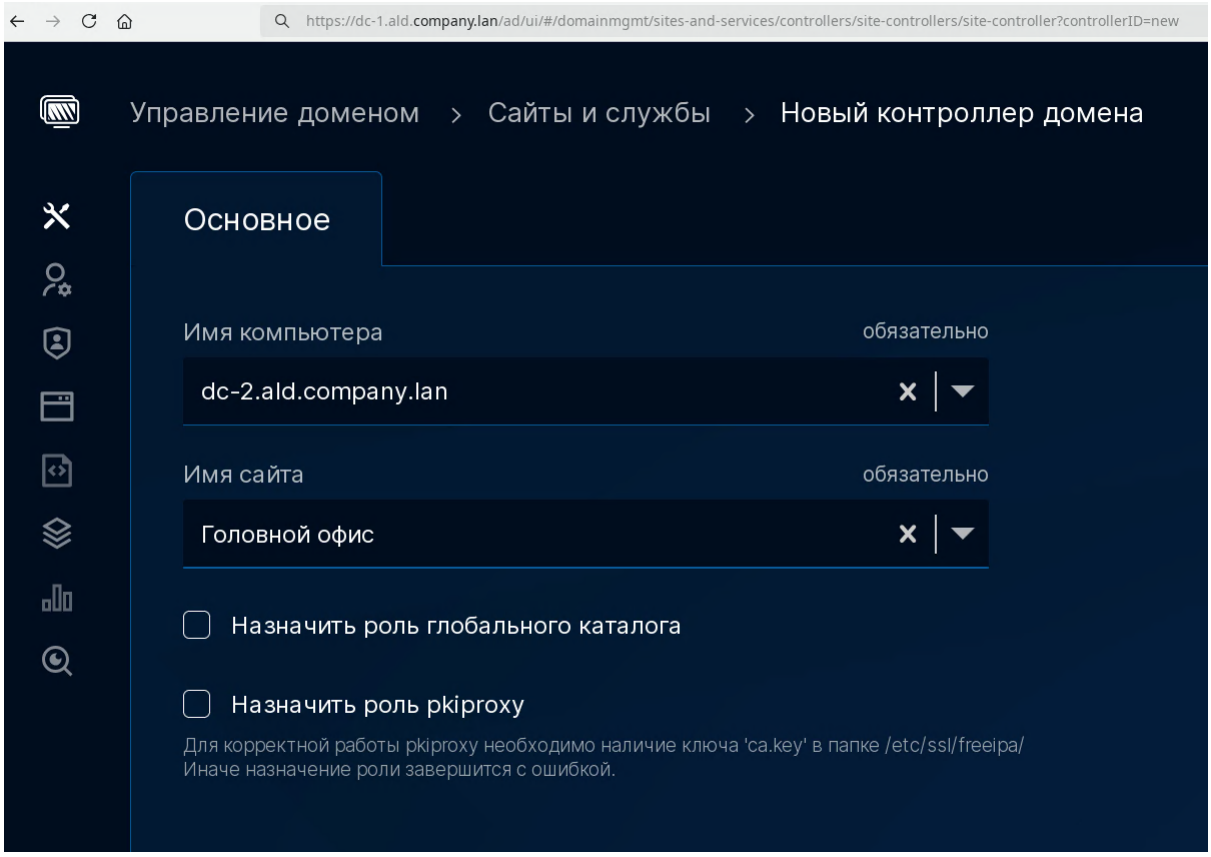


рис. 2.207 – Добавление нового контроллера домена

При добавлении нового контроллера домена есть возможность задать следующие параметры:

- **Имя контроллера** - выберите целевой хост из списка.
- **Имя сайта** - определите сайт, который нужно назначить контроллеру домена.
- **Назначить роль глобального контроллера** - установите флажок, чтобы автоматически установить на сервере модуль глобального каталога.
- **Назначить роль pkiproxy** - установите флажок, чтобы будущий контроллер мог обрабатывать запросы на выпуск сертификатов. Для успешного назначения этой роли на целевом хосте нужно заранее создать файл /etc/ssl/freeipa/ca.key и поместить в него закрытый ключ от доменного сертификата.

Примечание: До версии 2.4.0 для повышения роли сервера до контроллера домена требовалось вводить учетные данные администратора домена. Причем, для установки всех необходимых компонентов требовалось вручную добавить пользователя в группу **ald trust admin**.

Установка роли будет выполнена автоматически через полчаса, но вы можете форсировать уста-

новку подсистемы следующей командой:

```
sudo aldpro-roles --iud
```

Если установка подсистемы завершится ошибкой по какой-либо причине, например, из-за временной недоступности интернет-репозиториев, то повторную установку можно будет запустить следующей командой:

```
sudo aldpro-roles --iud --action install
```

Статус выполнения задания по установке резервного контроллера можно посмотреть в разделе *Управление доменом → Сайты и службы → Контроллеры домена → dc-2.ald.company.lan* , в строке состояние подсистемы, как показано на рисунке рис. 2.208:

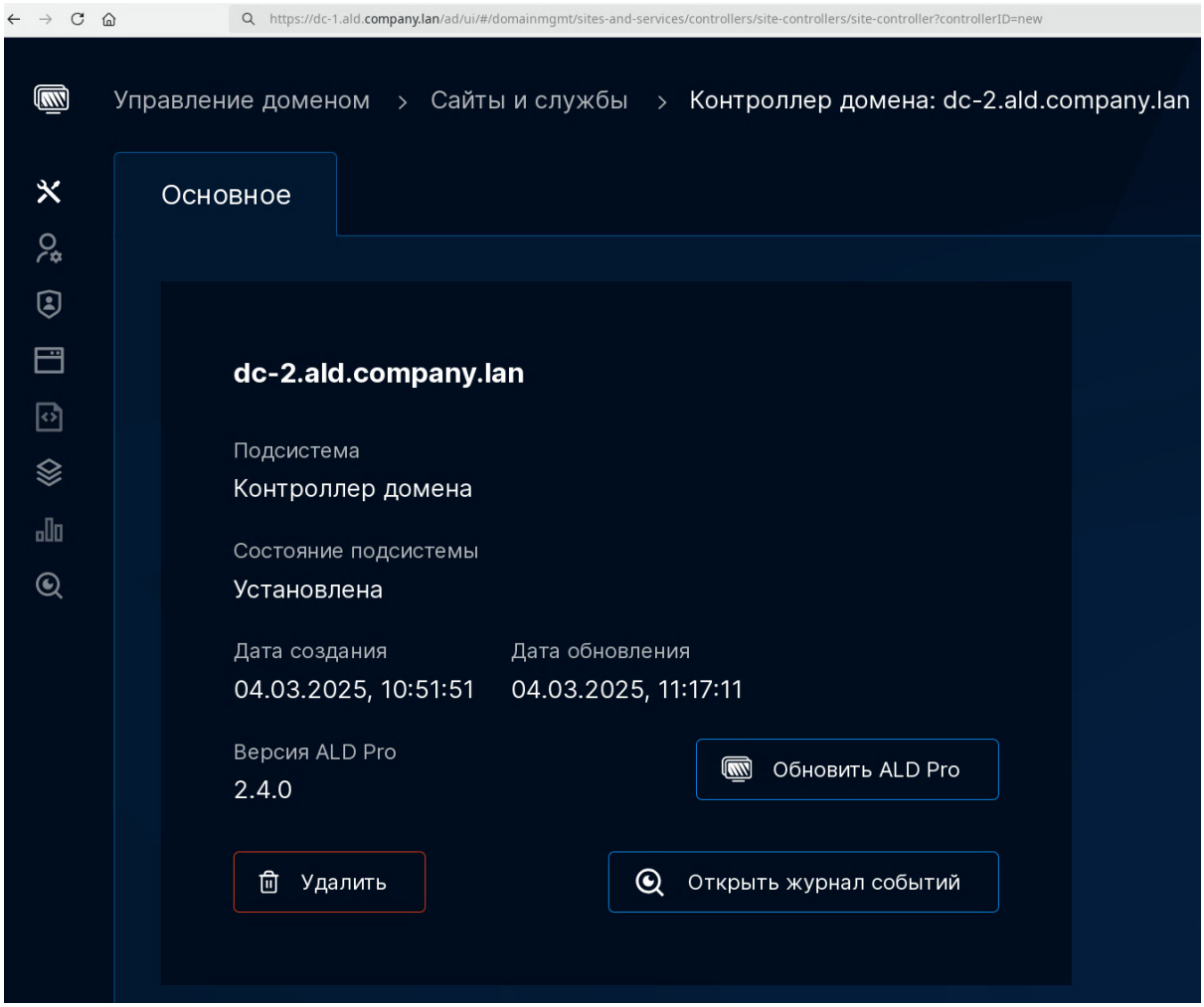


рис. 2.208 – Статус выполнения задания по установке резервного контроллера

Дополнительную информацию о событиях связанных с установкой резервного контроллера можно получить нажав кнопку **Открыть журнал событий**.

Считывание заданий происходит автоматически каждые 30 минут, что определяется на-

стройками параметра {«minutes»: 30} задания run_orch_subsystem в конфигурационном файле /srv/aldpro-salt/config/minion.d/standalone_scheduler.conf. В течении 30 минут хост **dc-2.ald.company.lan** получит информацию из LDAP о продвижении до резервного контроллера и начнется процесс установки и настройки, который займет около 25 минут. Дополнительную информацию о процессе установки вы можете получить на сервере **dc-2** в журналах /var/log/aldpro-salt/minion и /var/log/freeipa-replica-install.log или воспользоваться утилитой journalctl с параметром -f, которая позволяет получить информацию из **journald** в режиме реального времени:

```
sudo journalctl -f
```

После установки резервного контроллера домена начнется репликация данных, которая займет некоторое время. Информацию о завершении репликации можно будет увидеть в файле /var/log/dirsrv/slapd-ALD-COMPANY-LAN/errors.

```
sudo cat /var/log/dirsrv/slapd-ALD-COMPANY-LAN/errors | grep 'import userRoot'

[25/Nov/2024:09:40:09.407239382 +0300] - INFO - bdb_import_monitor_threads - import␣
↪userRoot: Workers finished; cleaning up...
[25/Nov/2024:09:40:09.641808546 +0300] - INFO - bdb_import_monitor_threads - import␣
↪userRoot: Workers cleaned up.
[25/Nov/2024:09:40:09.668458995 +0300] - INFO - bdb_public_bdb_import_main - import␣
↪userRoot: Indexing complete. Post-processing...
[25/Nov/2024:09:40:09.684394481 +0300] - INFO - bdb_public_bdb_import_main - import␣
↪userRoot: Generating numsubordinates (this may take several minutes to complete)...
[25/Nov/2024:09:40:09.756181641 +0300] - INFO - bdb_public_bdb_import_main - import␣
↪userRoot: Generating numSubordinates complete.
[25/Nov/2024:09:40:09.770676960 +0300] - INFO - bdb_get_nonleaf_ids - import␣
↪userRoot: Gathering ancestorid non-leaf IDs...
[25/Nov/2024:09:40:09.784493153 +0300] - INFO - bdb_get_nonleaf_ids - import␣
↪userRoot: Finished gathering ancestorid non-leaf IDs.
[25/Nov/2024:09:40:09.799225688 +0300] - INFO - ldbm_get_nonleaf_ids - import␣
↪userRoot: Starting sort of ancestorid non-leaf IDs...
[25/Nov/2024:09:40:09.815333717 +0300] - INFO - ldbm_get_nonleaf_ids - import␣
↪userRoot: Finished sort of ancestorid non-leaf IDs.
[25/Nov/2024:09:40:09.846484631 +0300] - INFO - bdb_ancestorid_new_idl_create_index -␣
↪import userRoot: Creating ancestorid index (new idl)...
[25/Nov/2024:09:40:09.893268580 +0300] - INFO - bdb_ancestorid_new_idl_create_index -␣
↪import userRoot: Created ancestorid index (new idl).
[25/Nov/2024:09:40:09.907374703 +0300] - INFO - bdb_public_bdb_import_main - import␣
↪userRoot: Flushing caches...
[25/Nov/2024:09:40:09.921178776 +0300] - INFO - bdb_public_bdb_import_main - import␣
↪userRoot: Closing files...
[25/Nov/2024:09:40:10.723980460 +0300] - INFO - bdb_public_bdb_import_main - import␣
↪userRoot: Import complete. Processed 3616 entries in 7 seconds. (516.57 entries/sec)
```

Далее отключите DNSSEC и разрешите рекурсивную обработку и кэширование запросов в /etc/bind/ipa-options-ext.conf.

Теперь, когда у нас есть установленная реплика, мы можем перейти к вопросам репликации.

8.4 Механизм репликации

Создавая службу каталога, компания Microsoft исходила из того, что предприятиям требуется надежное централизованное решение для управления аутентификацией в географически распределенных инфраструктурах, которые не отличаются постоянной сетевой связанностью между всеми компонентами сети.

Первый домен от Microsoft появился в 90-е годы в Windows NT и характеризовался наличием одного ведущего сервера PDC (Primary Domain Controller) и нескольких резервных реплик BDC (Backup Domain Controller). Для преобразования имен в домене использовался протокол NetBIOS, а для аутентификации — протокол NTLM.

Для своего времени это было инновационное решение, но очень скоро в мире UNIX появились более продвинутые технологии LDAP, DNS и Kerberos, которые лучше соответствовали потребностям предприятий, поэтому в Windows 2000 компания Microsoft выпустила переосмысленную версию домена под названием Active Directory.

Сильной стороной нового домена на базе LDAP-каталога стала возможность создавать кластеры из нескольких ведущих мастеров, которые позволяли изменять данные даже в условиях разделения сети. Подобный набор требований сильно выбивается из общих представлениях о кластерах, поэтому целесообразно будет немного затронуть вопросы теории распределенных систем.

Согласно теореме CAP, создавая распределенные системы, разработчики могут обеспечить соответствие только двум требованиям из трех:

- Согласованность данных (*англ. consistency*) — во всех вычислительных узлах в один момент времени данные не противоречат друг другу.
- Доступность (*англ. availability*) — запрос к системе не зависает и обрабатывается в установленные таймауты. Если запрос не может быть обработан, система так и должна об этом сказать, не заставляя пользователя ожидать бесконечно долго.
- Устойчивость к разделению (*англ. partition tolerance*) — при потере связи между частями системы обслуживание пользователя не прекращается.

Очевидно, что кластеры классических СУБД являются CA-системами, которые гарантируют согласованность данных, а вот служба каталога является AP-системой, которая фокусируется на устойчивости к разделению. Это не означает, что в домене нет согласованности данных, просто эта согласованность обеспечивается только «в конечном итоге».

При потере связи между московским и казанским офисами базовые услуги аутентификации будут предоставляться в каждом филиале независимо друг от друга. Администратор московского офиса сможет создать учетную запись, а пользователь сможет выполнить вход в систему, смену пароля и

подключение к местному файловому серверу. Но если в этот момент телепортировать сотрудника в казанский офис, то у него работать ничего не будет, т.к. изменения о пользователе в эту часть домена еще не доехали.

Процедуру в ходе которой содержимое каталога синхронизируется между серверами называют репликацией, и основная сложность этого алгоритма заключается в автоматическом решении конфликтов, которые неминуемо возникают при одновременном редактировании данных на нескольких серверах при отсутствии связи между ними.

8.4.1 Репликация в Windows

В отличие от модели репликации с единственным мастером, которая используется в системе Microsoft Windows NT, в домене Active Directory используется модель репликации с несколькими мастерами, поэтому изменения могут быть внесены на любом сервере и не всегда контроллеры в домене имеют идентичную информацию. Если же перестать вносить изменения в каталог, то через некоторое время контроллеры придут в согласованное состояние.

В ходе репликации контроллеры домена могут передавать друг другу изменения, выполненные на других узлах кластера, поэтому полная связанность «все со всеми» не требуется. Топология домена MS AD настраивается автоматически с помощью службы проверки согласованности знаний (англ. Knowledge Consistency Checker, KCC). При этом существуют значительные отличия в работе репликации внутри сайта и между сайтами.

При репликации данных внутри сайта MS AD стремится к тому, чтобы изменения как можно быстрее синхронизировались между контроллерами домена:

- Топология внутри сайта выстраивается автоматически службой KCC, которая объединяет узлы в кольца. Алгоритм стремится к тому, чтобы число прыжков ретрансляции (hop) было не больше трех. Когда количество контроллеров домена в сайте становится больше семи, создаются дополнительные связи для уменьшения числа ретрансляций.

- Репликация работает в режиме Pull, т.е. контроллер вытягивает изменения с другого сервера с помощью процедур удаленного вызова (англ. Remote Procedure Call, RPC), но извлечение данных начинается при получении соответствующего уведомления от передающей стороны, что гарантирует быстрое распространение изменений в сети.

  После изменения данных контроллер ожидает 15 секунд и запускает цикл репликации, уведомляя соседа о наличии изменений. После завершения сессии репликации с одним партнером контроллер домена уведомляет о наличии изменений следующего соседа. Небольшое ожидание помогает оптимизировать репликацию, если в каталог вносится сразу несколько изменений. Значение задержки может быть изменено

через реестр.

- Трафик репликации внутри сайта не сжимается, поскольку узлы обычно связаны высокоскоростными соединениями и целесообразнее загрузить каналы связи, чем создавать излишнюю нагрузку на центральный процессор.

Основная цель репликации между сайтами состоит в том, чтобы уменьшить нагрузку на сеть:

- Для межсайтовой репликации KCC автоматически выбирает специальные сервера-плацдармы (bridgehead server), помимо этого, администратор домена может вручную указать контроллеры домена, которые будут выполнять роль сервера-плацдарма.

- Межсайтовая репликация происходит по расписанию, т. е. сервера-плацдармы не уведомляют контроллеры из других сайтов о наличии изменений. Периодичность репликации может быть настроена вручную. Например, если пропускная способность сети крайне низкая, то репликацию можно выполнять в нерабочие часы.

- Для оптимизации нагрузки на сеть траффик межсайтовой репликации дополнительно сжимается. В зависимости от пропускной способностью и надежности сети могут использоваться протоколы IP или SMTP (протокол электронной почты).

Для репликации данных контроллеры используют журналы изменений, в которых записи идентифицируются номером изменений (англ. update sequence number, USN) и отметкой времени (timestamp). Номер USN представляет из себя число, которое увеличивается на единицу при каждом изменении базы Active Directory, например, если ранее значение highestCommittedUSN было 49459, а теперь 49482, то это означает, что за прошедшее время с помощью этого контроллера в базу данных было внесено 23 изменения.

В ходе сессии репликации сервер запрашивает у партнера таблицу highestCommittedUSN по всем контроллерам в домене, сопоставляет с известными ему значениями и запрашивает разницу. Таблица, в которой хранится информация о highestCommittedUSN, называется вектором обновления, или up-to-date vector. Посмотреть ее можно с помощью утилиты repadmin.

```
C:\> repadmin /showutdvec dc1 dc=win,dc=company,dc=lan
Default-First-Site-NameDC2 @ USN 49482 @ Time 2024-06-21 09:39:49
Default-First-Site-NameDC1 @ USN 30651 @ Time 2024-06-21 09:38:24
```

Одной из самых сложных задач в репликации с несколькими ведущими серверами является разрешение конфликтов, которые неизбежно возникают, если вносить изменения на нескольких контроллерах сразу. Рассмотрим несколько случаев:

- **Изменение атрибута существующей записи**

  В этом случае AD руководствуется версией и датой изменений. Более приоритет-

ным считается изменение, у которого выше номер версии. Если номер версии будет совпадать, то AD сочтет более приоритетным то изменение, которое было сделано последним по метке времени.

- **Создание записей с одинаковым отличительным именем DN (distinguished name)**

  В этом случае более приоритетным будет запись, у которой больше значение GUID. Вторая запись будет переименована в Name#CNF:userGUID, где символ # указывает на то, что эта запись является дубликатом. Если вам потребуется задействовать дублирующую запись, ее потребуется переименовать вручную.

- **Создание записи в удаленном контейнере**

  Если на одном контроллере удалить пустое организационное подразделение «Прочие сотрудники», а на другом контроллере в тоже самое время создать в этом подразделении нового сотрудника, то запись будет перемещена в контейнер с именем LostAndFound. Получить доступ к этому контроллеру можно через оснастку Пользователи и компьютеры (Users and Computers) в расширенном режиме (Advanced).

8.4.2 Репликация в ALD Pro

После установки второго контроллера домена вы можете посмотреть информацию о топологии на странице *Управление доменом → Сайты и службы → Соглашения о репликации*. Репликация ALD Pro (FreeIPA) во многом работает также как в MS AD, но есть и ряд отличий.

За работу репликации в 389 Directory Server отвечает плагин **Multimaster Replication Plugin**. Алгоритм его работы основан на использовании **журнала изменений**, который хранится в отдельной базе данных. Для работы механизма репликации требуются заранее установленные соглашения о репликации, за управление которыми отвечает плагин **IPA Topology Configuration**.

Журнал изменений (начиная с версии ALSE 1.7.4) храниться в файле `/var/lib/dirsrv/slapd-<Имя_Инстанса>/db/userRoot/replication_changelog.db`, например `/var/lib/dirsrv/slapd-ALD-COMPANY-LAN/db/userRoot/replication_changelog.db`. Настройки этого журнала можно найти в записи с DN *cn=changelog,cn=userRoot,cn=ldbm database,cn=plugins,cn=config*.

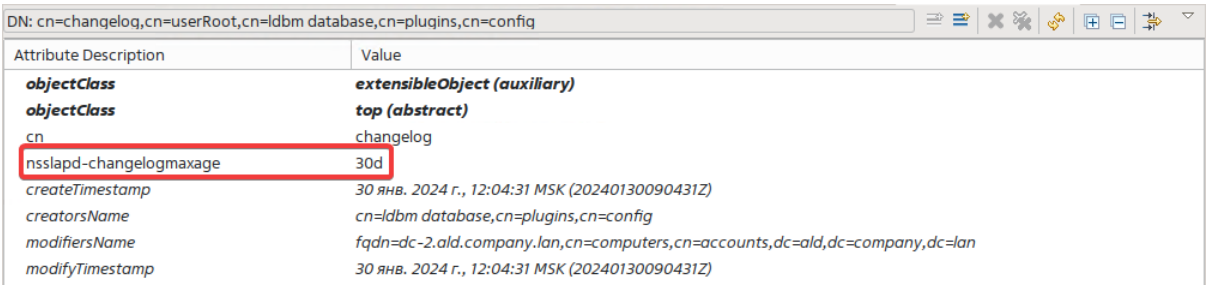


рис. 2.209 – Срок хранения записей в журнале

По умолчанию, данные в журнале изменений хранятся в течении 30 дней. Это позволяет корректно выполнить репликацию, если с момента нарушения связи до восстановления репликации прошло не более 30 дней. В противном случае потребуется повторная **полная инициализация контроллера** домена с полной загрузкой копии каталога от одного из серверов в домене.

Период в 30 дней считается оптимальным с точки зрения соотношения размера журнала и «безопасного» периода невозможности выполнить репликацию. Однако, при необходимости, этот период может быть изменен. Для этого, в записи с DN *«cn=changelog,cn=userRoot,cn=ldbm database,cn=plugins,cn=config»* можно задать два атрибута:

- nsslapd-changelogmaxage — максимальный возраст записей в журнале изменений (по умолчанию 30 дней)
- nsslapd-changelogmaxentries — максимальное количество записей в журнале (по умолчанию значение не задано)

В старых версиях 389-ds-base (для ALSE 1.7.3 и ниже) журнал и настройки хранились в каталоге `/var/lib/dirsrv/slapd-<Имя_Инстанса>/cldb` и в записи с DN *cn=changelog5,cn=config* соответственно.

Примечание: В службе каталога 389 Directory Server существует так же дополнительный механизм репликации, артефакты которого могут сбивать вас с толку. Этот механизм реализован в соответствии с RFC 4533 и за его работу отвечают плагины Content Synchronization (frontend) и Retro Changelog Plugin (backend). Он предназначен для интеграции с внешними приложениями, например, почтой или корпоративным порталом, которым нужно поддерживать актуальную информацию о пользователях, группах и других объектах каталога.

8.4.2.1 Поставщики и потребители

Репликация работает по обычному LDAP-протоколу с шифрованием (порт 389 + StartTLS). Сервер, передающий изменения, называется **Поставщиком**, а сервер, принимающий изменения, называется **Потребителем**. По итогам репликации на Потребителе формируется полная копия данных каталога, поэтому его также называют **Репликой**.

В домене FreeIPA, как и в домене MS AD, каждая реплика является **Мастером**, то есть доступна на запись и с ее помощью можно вносить изменения в каталог. Такую репликацию, когда один и тот же сервер в разные моменты времени выступает то в роли Поставщика, то в роли Потребителя, называют репликацией с несколькими **Ведущими** серверами.

В отличии от MS AD репликация в домене ALD Pro (FreeIPA) всегда инициируется Поставщиком, а не Потребителем (т.е. работает по модели Push). В сеансе репликации Поставщик подключается к Потребителю, выполняя аутентификацию по протоколу Kerberos с использованием учетной записи ldap/dc-1.ald.company.lan. Затем Поставщик запрашивает у Потребителя сводную информацию о состоянии его репликации (так называемую таблицу RUV, Replica Update Vector), чтобы определить, есть ли у него для Потребителя актуальные изменения, и запускает передачу данных.

8.4.2.2 Соглашение о репликации

В домене ALD Pro нет службы KCC как в MS AD, поэтому топологию нужно настраивать вручную через создание сегментов и соглашений о репликации.

Сегмент топологии (*англ. Topology segment*) определяет связь между двумя узлами и реплицируется между контроллерами домена, а **соглашение о репликации** хранится только на Поставщике и содержит настройки для подключения к Потребителю. Таким образом на каждый сегмент топологии приходится по два соглашения.

Примечание: Обратите внимание, что в интерфейсе ALD Pro есть небольшое упрощение в терминологии, поэтому то, что в ALD Pro называют «соглашениями о репликации», в FreeIPA на самом деле является «сегментами».

Если у сервера более одного соглашения, то сервер устанавливает исходящие сессии репликации по каждому из этих соглашений по очереди. Для повышения отказоустойчивости рекомендуется создавать не менее двух исходящих соглашений о репликации, так как при надежности одного соединения в 90% два соединения дадут две девятки 90% + 10%*90% = 99%.

Создавать более четырех соглашений целесообразно только для уменьшения количества ретрансляций, но нужно следить за тем, чтобы все сервера находились в рабочем состоянии. Если у кон-

троллера будет много соглашений с неработающими репликами, то выполняя цикл репликации, сервер будет терять много времени на ожидание ответов от этих серверов. Поэтому инструменты FreeIPA предупреждают о возможных проблемах, если увидят, что на сервере более четырех соглашений, но это не является техническим ограничением.

Примечание: Продуктовая команда проводила успешное тестирование топологии «звезда», в которой у центрального контроллера было более 150 соглашений о репликации. Проблемы возникали только в том случае, когда сервер терял сетевую связанность с десятками реплик сразу.

Соглашение настраивается на Поставщике, хранится в его файле dse.ldif и доступно через LDAP в ветке с DN «*cn=mapping tree,cn=config*». Соглашение содержит следующую информацию:

- Адрес сервера Потребителя, на который нужно передавать данные (nsDS5ReplicaHost и nsDSReplicaPort)
- Способ передачи данных (nsDS5ReplicaTransportInfo, по умолчанию LDAP)
- Способ аутентификации (nsDS5ReplicaBindMethod=SASL/GSSAPI)
- Суффикс базы данных для репликации (nsDS5ReplicaRoot, по умолчанию dc=ald,dc=company,dc=local)
- Атрибуты, которые должны или не должны реплицироваться (nsDS5ReplicatedAttributeList, nsDS5ReplicatedAttributeListTotal, nsds5ReplicaStripAttrs)

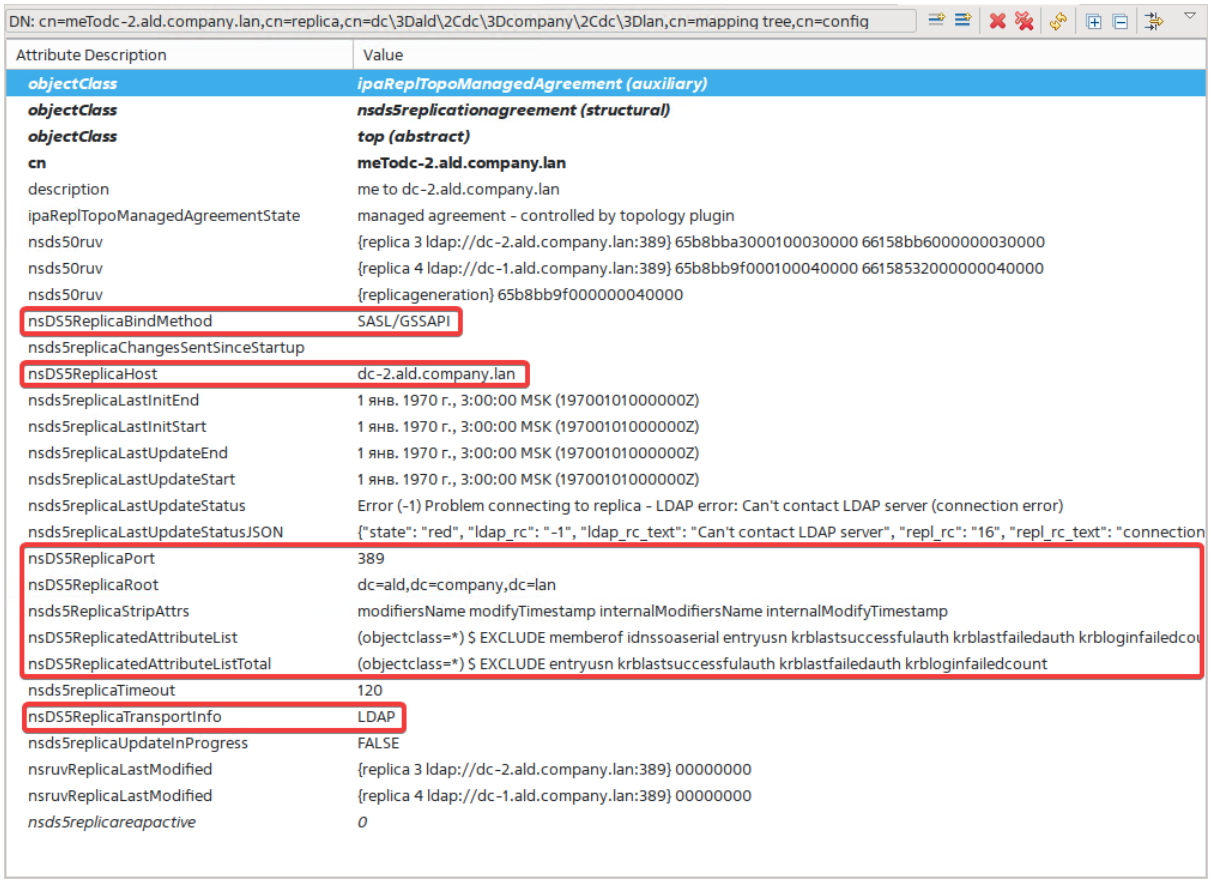


рис. 2.210 – Параметры соглашения о репликации

Кроме обязательных параметров есть много дополнительных, которые позволяют, например, установить расписание репликации (nsDS5ReplicaUpdateSchedule).

8.4.2.3 Журнал изменений

Как мы уже узнали выше, для обеспечения работы механизма репликации сервер ведет журнал изменений. Когда мы создаем в каталоге запись или меняем значение атрибута, сервер не только вносит изменения в базу данных каталога, но и записывает информацию об этой операции в журнал изменений.

Прямого доступа к журналу через LDAP у нас нет, но с помощью утилиты `dbscan` можно прочитать файл изменений:

```
admin@dc-1:~$ sudo dbscan -f /var/lib/dirsrv/slapd-ALD-COMPANY-LAN/db/userRoot/
↪replication_changelog.db | head -n 15
dbid: 66158532000000040000
   encrypted: no
   replgen: 1712686215 Tue Apr 9 21:10:15 2024
   csn: 66158532000000040000
   uniqueid: 88e30e06-bf4e11ee-9d29cb43-d00727f4
   dn: cn=repl keep alive 4,dc=ald,dc=company,dc=lan
```

(continues on next page)

(продолжение с предыдущей страницы)

```
   operation: modify
         keepalivetimestamp: 20240409181015Z
         modifiersname: cn=Multisupplier Replication Plugin,cn=plugins,cn=config
         modifytimestamp: 20240409181015Z
         entryusn: 1394188

dbid: 66158534000000040000
   encrypted: no
```

Размер журнала будет зависеть от количества изменений в каталоге и от параметров в DN «*cn=changelog,cn=userRoot,cn=ldbm database,cn=plugins,cn=config*», которые определяют максимальное количество записей в журнале и их максимальный возраст:

- **nsslapd-changelogmaxage** — максимальный возраст записей в журнале изменений (по умолчанию 30 дней)
- **nsslapd-changelogmaxentries** — максимальное количество записей в журнале (по умолчанию значение не задано)

  Очевидно, что если контроллер домена будет выключен более 30 дней, то в домене не останется данных для возможности инкрементального восстановления его целостности и потребуется полная инициализация данных.

8.4.2.4 Порядковый номер изменения (CSN)

Для эффективного решения конфликтов репликации каждое изменение идентифицируется порядковым номером изменения CSN (Change Sequence Number или Change State Number), которое представляет из себя 20-разрядное число в шестнадцатеричной системе счисления, например, 64a0238f 0008 0003 0000, где:

1) **Часть A** (метка времени) — разряды 1-8 (64a0238f) — это часть идентификатора в формате временной метки в unix time_t (количество секунд с начала эпохи), которая обычно соответствует моменту времени, когда изменение было сделано на Ведущей реплике.

   Однако, идентификаторы должны идти подряд, поэтому, если время на сервере станет меньше, чем CSN последнего изменения, то генератор CSN продолжит использовать последнее значение, увеличивая его на единицу при исчерпании емкости следующих четырех разрядов числа «B».

2) **Часть B** (порядковый номер) — разряды 9-12 (0008) — это часть идентификатора, которая определяет порядковый номер изменения в пределах значения числа «A». Позволяет идентифицировать до 65 536 изменений (16^4), поэтому, как только указанный диапазон будет исчерпан, значение числа «A» должно быть увеличено на единицу, а нумерация числа «B»

начнется с начала.

3) **Часть C** (идентификатор реплики) — разряды 13-16 (0003) — это часть идентификатора, соответствующая номеру Ведущего сервера, через который было внесено данное изменение в каталог.

   Уникальный номер Ведущего сервера должен быть в диапазоне до 65 534. Последнее значение 65 535 используется для обозначения «read-only» реплик, которые называются Хабами, но не путайте их с RODC в терминологии MS AD, т.к. в них не заложена какая-то особая логика работы с хешами пользовательских паролей.

   Как вы можете заметить, на уровне протокола заложены практически неограниченные возможности по масштабированию. Продуктовая команда провела успешное тестирование работы домена на 400 контроллеров и планирует довести тесты до значений в несколько тысяч.

4) **Часть D** (резерв) — разряды 17-20 (0000) — это часть идентификатора, зарезервированная для возможности расширения функциональности протокола репликации в будущем.

8.4.2.5 Алгоритм репликации

Репликация использует **журнал изменений** и порядковые номера **CSN**. Упрощенно алгоритм выглядит следующим образом:

1) В сеансе репликации **Поставщик** запрашивает у **Потребителя**, какой номер последнего изменения по Ведущему серверу «X» ему известен (**CSN Max**).

2) Далее Поставщик сравнивает полученное значение **CSN Max** с своим собственным значением **CSN Max** по этому Ведущему серверу «X».

3) Если окажется, что собственное значение у **Поставщика** больше, чем значение, полученное от **Потребителя**, значит у **Поставщика** есть **более свежие** обновления по указанному Ведущему серверу «X». В этом случае Поставщик передаст Потребителю новые данные.

В действительности для повышения производительности 389 Directory Server использует так называемые вектора обновления реплик (Replica Update Vector, RUV), которые позволяют уменьшить нагрузку на диск и центральный процессор за счет сокращения количества обращений к журналу изменений.

Каждый вектор содержит четыре значения:

- **Replica Id** — идентификатор Ведущего сервера, который является источником прямых изменений.
- **URL** — адрес Ведущего сервера для обращения по LDAP.

- **Min CSN** — номер первого изменения от данного Ведущего сервера в журнале изменений текущего сервера.
- **Max CSN** — номер последнего изменения от данного Ведущего сервера в журнале изменений текущего сервера.

С учетом RUV алгоритм репликации выглядит следующим образом:

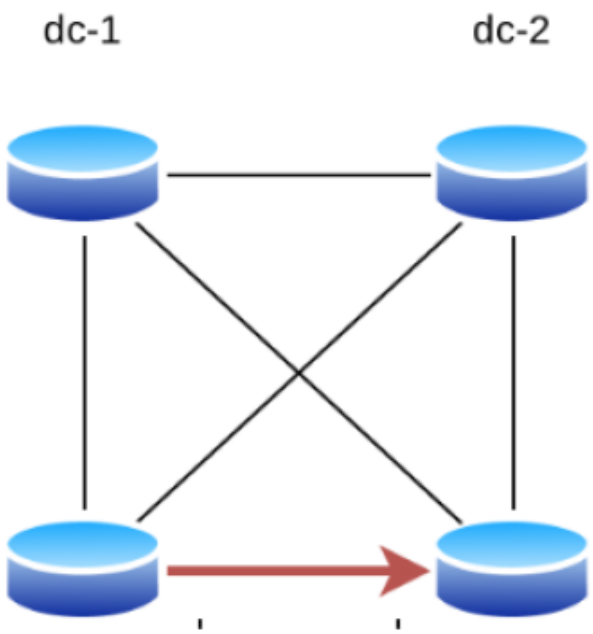
- В начале сеанса репликации Поставщик берет свою таблицу RUV и загружает копию таблицы RUV Потребителя. Значения Max CSN берутся из этих таблиц вместо прямых обращений к журналам изменений.
- После завершения репликации по конкретному Ведущему серверу Потребитель сохраняет в свою таблицу RUV (а конкретнее в строку по этому Ведущему серверу) крайнее значение CSN, которое стало ему известно от Поставщика.
- Новое значение CSN в RUV Потребителя будет использовано в качестве стартовой точки в следующем сеансе репликации.

Актуальные значения векторов хранятся в памяти сервера, но их можно посмотреть с помощью утилиты `dsconf`, подкоманда `replication get-ruv`:

```
admin@dc-1:~$ sudo dsconf -D 'cn=Directory Manager' -w AstraLinux ALD-COMPANY-LAN␣
↪replication get-ruv --suffix dc=ald,dc=company,dc=lan
RUV:        {replica 4 ldap://dc-1.ald.company.lan:389} 65b8bb9f000100040000␣
↪6643fc28000000040000
Replica ID: 4
LDAP URL:   ldap://dc-1.ald.company.lan:389
Min CSN:    2024-01-30 09:04:31 1 0 (65b8bb9f000100040000)
Max CSN:    2024-05-15 00:04:56 (6643fc28000000040000)

RUV:        {replica 3 ldap://dc-2.ald.company.lan:389} 65b8bba3000100030000␣
↪6616b4be000000030000
Replica ID: 3
LDAP URL:   ldap://dc-2.ald.company.lan:389
Min CSN:    2024-01-30 09:04:35 1 0 (65b8bba3000100030000)
Max CSN:    2024-04-10 15:48:14 (6616b4be000000030000
```

Рассмотрим работу алгоритма репликации на примере соглашения dc3 — dc4, когда dc-3 выступает в роли Поставщика, а dc-4 в роли Потребителя. Для наглядности будем обозначать номера изменений Max CSN с помощью одноразрядных целых чисел.



	dc-3		dc-4	
	Max CSN	?	Max CSN	Решение
dc-1	3	>	1	передать
dc-2	4	<	5	нет
dc-3	7	=	7	нет
dc-4	3	<	5	нет

рис. 2.211 – Пример репликации с использованием значения MAX CSN

Поставщик dc-3 подключается к Потребителю dc-4 и построчно сверяет свою таблицу RUV с таблицей RUV, полученной от Потребителя.

- По строке #1 видно, что «3 > 1», а значит, у Поставщика dc-3 есть два изменения, о которых

еще не знает Потребитель dc-4, поэтому принимается решение о том, чтобы их передать. После получения новых изменений Потребитель в своей таблице RUV для реплики dc-1 обновит CSN Max.

- По строке #2 видно, что «4 < 5», т.е. у Поставщика dc-3 даже более старые данные, чем у Потребителя dc-4, но репликации не происходит, так как в рамках текущей сессии репликации сервер dc-4 выступает в роли Потребителя.
- По строке #3 видно, что «7 = 7», т.е. оба сервера обладают одинаковыми данными.
- По строке #4 ситуация аналогична строке #2.

Примечание: Если вы выключите все контроллеры домена и сделаете снимки всех виртуальных машин, то вы сможете восстановить домен целиком, используя эти резервные копии.

Однако, в случае восстановления из резервной копии отдельного контроллера он не получит от соседей тех изменений, которые были созданы непосредственно с его участием, поэтому потребуется его полная инициализация.

8.4.2.6 Конфликты репликации

Конфликты репликации в домене ALD Pro (FreeIPA) решаются примерно так же, как в MS AD:

- **Изменение атрибута существующей записи**

  В этом случае FreeIPA руководствуется значением CSN — приоритет отдается тому изменению, у которого номер больше, поэтому ручное вмешательство не требуется.

- **Создание записей с одинаковым отличительным именем DN**

  Если создать две записи с одинаковым DN на двух контроллерах домена сразу, то механизм разрешения конфликтов оставит ту из них, что была создана раньше, а вторая запись будет переименована и скрыта. В имя скрываемой записи будет добавлен уникальный идентификатор nsuniqueid, являющийся операционным атрибутом.

  Например, если создать пользователя figaro на двух контроллерах сразу, то у нас будет сразу две записи:

  1. uid=figaro,cn=users,cn=accounts,dc=ald,dc=company,dc=lan
  2. nsuniqueid=7341f021-20b331ee-a3ef96aea91d3705+uid=figaro,cn=users,cn=accounts,dc=ald,dc=company,dc=

  Обработка конфликта будет происходить на каждом контроллере домена индивидуально, а

операция переименования записи в nsuniqueid+uid не отразится в журнале изменений. К переименованной записи будет добавлен так же класс объекта ldapsubentry и операционный атрибут nsds5ReplConflict.

- **Создание записи в удаленном контейнере**

  В системе ALD Pro структурные подразделения реализованы через ссылки в атрибуте rbtadp, поэтому при создании пользователя в удаленном подразделении мы получим просто неработающую ссылку, которую можно будет легко исправить в карточке объекта. Однако, подобные ситуации могут возникать, например, при редактировании объектов групповых политик.

  Если на одном контроллере домена удалить запись, а на другом в тот же момент создать ее потомка, то механизм разрешения конфликтов автоматически создаст запись с классом glue взамен удалённой родительской записи. Обработка конфликта будет происходить на каждом контроллере домена индивидуально в момент его появления, и эти изменения не отразятся в журнале изменений changelog.

  Запись с классом glue будет иметь атрибуты «objectClass: glue», «objectClass: extensibleobject» и «nsds5ReplConflict: deletedEntryHasChildren». В зависимости от ситуации удалённая родительская запись может быть создана одним из следующих способов:

  1. Созданием новой записи с классом glue и минимальным набором атрибутов.
  2. Восстановлением удалённой записи из надгробия (англ. tombstone) и превращением её в запись с классом glue.
  3. Превращением еще не удаленной записи в запись с классом glue.

8.4.2.7 Оптимизация межсайтовой репликации

В домене ALD Pro (FreeIPA) в отличие от MS AD внутрисайтовая и межсайтовая репликации по умолчанию ничем не отличаются. Однако в соглашениях о репликации есть несколько атрибутов, с помощью которых вручную можно добиться некоторых оптимизаций.

Параметр nsds5ReplicaEnabled

Параметр позволяет отключить передачу изменений по соглашению о репликации, не удаляя соглашение полностью. Он удобен для отладки, например, с его помощью можно легко создать условия для возникновения конфликтов репликации.

Параметр	Описание
DN запись	cn=ReplicationAgreementName,cn=replica,cn=suffixDN,cn=mapping tree,cn=config
Допустимое значение	on | off
Значение по умолчанию	on
Синтаксис	строка

Параметр nsDS5ReplicaUpdateSchedule

Параметр позволяет задать расписание, по которому соглашение о репликации будет считаться активным. Для удаленных офисов может быть целесообразным разрешить репликацию только в ночные часы, чтобы разгрузить спутниковые каналы связи.

Параметр	Описание
DN запись	cn=ReplicationAgreementName,cn=replica,cn=suffixDN,cn=mapp tree,cn=config
Допустимое значение	**Расписание задается строкой в формате XXXX-YYYY 0123456, где** • *XXXX* — время начала, • *YYYY* — время окончания в формате HHMM, • числа *0123456* — дни недели, начиная с воскресенья.
Значение по умолчанию	Если параметр не задан, предполагается значение 0000-2359 0123456, т.е. соглашение активно все время
Синтаксис	строка
Пример	nsDS5ReplicaUpdateSchedule: 0000-2359 0123456

Параметр nsslapd-connection-buffer

Параметр устанавливает поведение буферизации соединения. Возможные значения:

1) 0 — отключить буферизацию. Только одиночные блоки данных (Protocol Data Unit, PDU)

считываются за раз.

2) 1 — обычный фиксированный размер LDAP_SOCKET_IO_BUFFER_SIZE в 512 байт.

3) 2 — адаптируемый размер буфера.

Значение #2 обеспечивает более высокую производительность, если клиент отправляет большой объем данных, что лучше соответствует межсайтовому характеру репликации.

Параметр	Описание
DN запись	cn=config
Допустимые значения	0 | 1 | 2
Значение по умолчанию	1
Синтаксис	integer
Пример	nsslapd-connection-buffer: 1

Параметры nsDS5ReplicaBusyWaitTime и nsDS5ReplicaSessionPauseTime

Параметр **BusyWaitTime** устанавливает задержку в секундах, которую Поставщик должен выждать, прежде чем предпринимать повторную попытку соединения с Потребителем после того, как тот вернул ответ «занято». Значение по умолчанию 3 сек. Параметр **SessionPauseTime** устанавливает задержку в секундах, которую Поставщик должен выждать между сеансами обновления. Значение по умолчанию равно 0 сек.

Для межсайтовой репликации целесообразно устанавливать длительные интервалы ожидания, чтобы каждый сайт получал гарантированную возможность для выполнения процедуры репликации.

Параметр nsDS5ReplicaBusyWaitTime	Описание
DN запись	cn= ReplicationAgreementName, cn=replica,cn=suffixDN,cn=mapping tree,cn=config
Допустимые значения	Любое допустимое целое число
Значение по умолчанию	3
Синтаксис	Integer
Пример	nsDS5ReplicaBusyWaitTime: 3

Параметр nsDS5ReplicaSessionPauseTime	Описание
DN входа	cn=ReplicationAgreementName, cn=replica,cn=suffixDN,cn=mapping tree,cn=config
Допустимые значения	Любое допустимое целое число
Значение по умолчанию	0
Синтаксис	Целое число
Пример	nsDS5ReplicaSessionPauseTime: 0

Параметры nsds5ReplicaFlowControlWindow и nsds5ReplicaFlowControlPause

Параметр **FlowControlWindow** устанавливает максимальное количество изменений, которые могут быть отправлены Поставщиком без получения подтверждения от Потребителя. После достижения лимита поставщик приостанавливает передачу данных по соглашению о репликации на время, указанное в параметре **FlowControlPause**.

Параметры позволяют адаптировать соглашение о репликации под пропускную способность сети.

Параметр nsds5ReplicaFlowControlPause	Описание
DN запись	cn=ReplicationAgreementName, cn=replica,cn=suffixDN,cn=mapping tree,cn=config
Допустимые значения	от 0 до максимальной длины 64 бита
Значение по умолчанию	2000
Синтаксис	Integer
Пример	nsds5ReplicaFlowControlPause: 2000

Параметр nsds5ReplicaFlowControlWindow	Описание
DN запись	cn=ReplicationAgreementName, cn=replica,cn=suffixDN,cn=mapping tree,cn=config
Допустимые значения	от 0 до максимальной длины 64 бита
Значение по умолчанию	1000
Синтаксис	Целое число
Пример	nsds5ReplicaFlowControlWindow: 1000

Для межсайтовой репликации количество отправляемых изменений должно быть как можно боль-

ше, но при этом нужно добиться того, чтобы Поставщик не отправлял изменения быстрее, чем Потребитель их обрабатывает, иначе в журнале ошибок Поставщика будут регистрироваться следующие сообщения:

```
Общее управление потоком обновлений дает потребителю время (2000 мс) перед отправкой␣
↪дополнительных записей [отправлено msgid: xxx, rcv: yyy])
Если полное обновление не удается, вы можете попытаться увеличить␣
↪nsds5ReplicaFlowControlPause и/или уменьшить nsds5ReplicaFlowControlWindow в␣
↪конфигурации соглашения реплики.
```

8.5 FSMO роли в MS AD

В домене MS AD все контроллеры равны, но некоторые равнее других. Для уменьшения вероятности конфликтов и оптимизации работы репликации в MS AD есть пять ролей хозяина операций (flexible single-master operator, FSMO):

1. Новый домен в лес доменов MS AD может добавить только хозяин именования доменов (Domain Naming Master).

   В домене FreeIPA нет леса, но есть похожая операция в момент установления доверительных отношений с лесом MS AD: служба каталога извлекает информацию обо всех поддоменах доверенного леса и назначает каждому из них диапазон идентификаторов. Вероятность того, что администраторы будут устанавливать доверительные отношения с несколькими лесами параллельно, крайне мала, поэтому дополнительная роль не требуется.

   Кстати, в ALD Pro сделан первый шаг на пути реализации лесов и с версии 2.3.0 вы уже сможете установить доверительные отношения между доменами ALD Pro. Это позволит создать дополнительные периметры безопасности, т. к. хеши паролей не покидают периметр домена, и снизить требования к аппаратным характеристикам контроллеров.

2. Схему каталога в MS AD может изменить только хозяин схемы (Schema Master).

   В службе каталога FreeIPA нет выделенной роли, изменить схему можно с любого контроллера в домене. Более того, вы можете не только добавлять, но и удалять атрибуты и классы объектов, хотя эти изменения относятся к небезопасным и их рекомендуется избегать.

   Вам следует учитывать, что изменения схемы распространяются в домене не мгновенно, и некоторое время сервера могут работать с разными ревизиями схемы. Репликация схемы выполняется в начале каждой сессии репликации.

3. Изменить фантомные объекты, являющиеся ссылками на объекты из других доменов леса в MS AD может только хозяин инфраструктуры (Infrastructure Master).

   В службе каталога FreeIPA есть внешние группы, которые позволяют добавлять субъекты из

доверенного домена MS AD, но никакого автоматического обновления информации по этим ссылкам не выполняется, поэтому дополнительная роль не требуется.

4. Диапазоном свободных идентификаторов в домене MS AD распоряжается только хозяин RID (Relative ID Master или RID Master). Этот хозяин выдает другим контроллерам свободные идентификаторы по 500 штук. Контроллеры запрашивают следующую пачку, когда количество свободных идентификаторов становится меньше 100.

   В службе каталога FreeIPA нет хозяина RID, им может стать любой контроллер домена. Если контроллеру потребуется идентификатор, то он заберет у соседа не 500 идентификаторов, а сразу половину всего оставшегося диапазона. За работу этого механизма отвечает плагин динамического назначения номеров DNA. Посмотреть DNA диапазоны, назначенные серверам, можно командой ipa-replica-manage dnarange-show.

5. И самая нагруженная роль в MS AD — это эмулятор PDC (Primary Domain Controller Emulator, PDC Emulator). Без этого контроллера все продолжает работать, но не так хорошо, как вместе с ним. Он отвечает за следующие операции:

   1. Является источником времени для контроллеров в домене.

      В домене ALD Pro эту функцию выполняют корневые NTP-сервера.

   2. В приоритетном порядке получает изменения паролей, чтобы успешно перепроверять аутентификацию пользователей, если пароль оказался неверным. Это ускорить распространение паролей в сети, уменьшая вероятность блокировки пользователей сразу после смены пароля.

      Учитывая, что в условиях недоступности эмулятора PDC проблем с паролями в домене не возникает, эта оптимизация не считается критически важной и в FreeIPA ее нет.

   3. Редактирование групповых политик по умолчанию выполняется через эмулятор PDC. Если эмулятор будет недоступен, то оснастка попросит указать явно контроллер, через который нужно вносить изменения.

      В FreeIPA и ALD Pro указанный подход не используется. Конфликты репликации объектов GPO без каких-либо проблем решаются в штатном порядке.

   4. Редактирование пространства имен DFS по умолчанию выполняется через эмулятор PDC, но конфликты при редактировании этих данных могли бы решаться и в штатном порядке.

8.6 Работа механизмов автоматического обнаружения сервисов

В домене ответственность за балансировку всегда возлагается на клиента. Механизмы AD и FreeIPA (SSSD) очень похожи, поэтому разберем подробно только одну сторону вопроса.

8.6.1 Автоматическое обнаружение при вводе хоста в домен утилитой ipa-client-install

Ввод машины в домен осуществляется с помощью утилиты aldpro-client-installer, которая по цепочке вызывает astra-freeipa-client и ipa-client-install. Передача параметров выполняется так, как представлено в таблице 1.

табл. 2.16 – Передача параметров при вводе машины в домен

aldpro-client-installer →	astra-freeipa-client →	ipa-client-install
получает: --domain ald.company.lan --account admin --password 'AstraLinux' --host pc-1 --gui --force	**получает:** -d ald.company.lan -u admin -p AstraLinux -y --par "--hostname=pc-1.ald.c	**получает:** 'domain_name': None, 'principal': 'admin', 'host_name': 'pc-1.ald.company.lan', 'force': False, rce-join" 'force_join': True, 'mkhomedir': True, и др., при этом пароль не логируется
вызывает **astra-freeipa-client** -d "{domain_name}" -u "{login}" -p "{password}" -y --par "--hostname={host}.{doma	**вызывает** **ipa-client-install** -U -p $username -w $userpass $par **перед этим устанавливает** hostnamectl set-hostname $compname.${domain,,}	

Механизм автоматического обнаружения сервисов предусмотрен только в скрипте ipa-client-install и работает для определения следующих параметров:

- **Realm/DNS Domain**: ALD.COMPANY.LAN/ald.company.lan

  Область Керберос и fqdn домена, отличаются обычно только регистром символов. Можно сделать так, чтобы эти значения были разными, но такие настройки сложны и не имеют значительных преимуществ, поэтому не рекомендуются.

- **IPA Server**: dc-1.ald.company.lan

FQDN имя контроллера, через который должен быть выполнен ввод машины в домен. Обычно не указывается явно и определяется автоматически.

- **Client hostname**: pc-1.ald.company.lan

  FQDN имя, которое должно быть у компьютера после ввода в домен, DNS зона должна совпадать с fqdn домена. Если значения не будут совпадать, то работа хоста будет нарушена. Разработчики специально не сделали такую проверку, т.к. это дает гибкость в некоторых сценариях развертывания.

Если домен не указан явно через параметр `--domain`, то скрипт будет пытаться получить это значение из следующих источников:

- из параметра `--host`;
- из текущего имени компьютера `hostname` (используется имя, а не fqdn, т. е. это то значение, которое возвращает команда `hostname`, а не `hostname -f`);
- из файла `resolv.conf`, параметры search и domain.

Проверка имени домена выполняется путем извлечения SRV записей для протокола ldap по tcp. Для каждого fqdn скрипт проверяет все компоненты домена вверх по иерархии, поэтому, к примеру, если в переменной hostname было значение «pc-1.ald.company.lan», то скрипт установки проверит следующие SRV-записи:

- _ldap._tcp.ald.company.lan
- _ldap._tcp.company.lan
- _ldap._tcp.lan

Обратите внимание, что утилита astra-freeipa-client нарушает логику передачи параметров, т.к. вызывает утилиту ipa-client-install с значением {„domain_name“: None}, но перед этим она устанавливает в системе имя хоста «pc-1.ald.company.lan», поэтому ввод машины в домен отрабатывает корректно.

Сервер IPA можно указать явно через параметр server, но в том случае параметр domain также должен быть задан в явном виде. Если параметр server не будет указан, то сервер будет найден через поиск SRV-записей _ldap._tcp.DOMAIN.

При использовании опции server в файлы sssd.conf и krb5.conf будет внесен фиксированный список серверов. Параметр можно указывать несколько раз подряд, чтобы задать для переменной ipa_server список значений.

8.6.2 Автоматическое обнаружение сервисов в SSSD

Для обеспечения надежной работы компьютера в домене в службе SSSD реализован механизм автоматического обнаружения контроллеров домена. В момент загрузки служба SSSD берет параметр domains и обращается к соответствующей секции файла /etc/sssd/sssd.conf для настройки бэкенда:

```
[sssd]
domains = ald.company.lan

[domain/ald.company.lan]
id_provider = ipa
ipa_server = _srv_, dc-1.ald.company.lan
ipa_domain = ald.company.lan
ipa_hostname = pc-1.ald.company.lan
auth_provider = ipa
chpass_provider = ipa
access_provider = ipa
cache_credentials = True
ldap_tls_cacert = /etc/ipa/ca.crt
krb5_store_password_if_offline = True
```

Рассмотрим ключевые параметры бэкенда:

- **ipa_provider** — указывает, что в качестве поставщика идентификационных данных будет выступать сервер IPA
- **ipa_server** — задает список FQDN-имен и IP-адресов контроллеров домена в порядке, в каком нужно к ним подключаться. Может использоваться также служебное имя «_srv_», соответствующее механизму автоматического обнаружения (service discovery) через поиск SRV-записей «_ldap._tcp.DOMAIN», как указано в RFC 2782.
- **ipa_backup_server** — с помощью этого параметра можно указать список резервных серверов, которые будут выбираться только в случае недоступности основных. Этот параметр не предполагает возможности использования механизма автообнаружения.
- **ipa_domain** — указывает имя домена IPA. Это необязательный параметр, если он не указан, то используется доменное имя из названия конфигурации.
- **ipa_hostname** — требуется устанавливать на машинах, где имя хоста не соответствует полному имени компьютера, которое используется в домене IPA для идентификации этого хоста.

Записи SRV, создаваемые FreeIPA для своих сервисов, существенно отличаются от тех, которые можно создать вручную из интерфейса. Этим записям в каталоге назначен класс idnsTemplateObject и задано дополнительное свойство «idnsTemplateAttribute;cnamerecord», которое определяет шаблон подстановки для поддержки технологии сайтов (локаций).

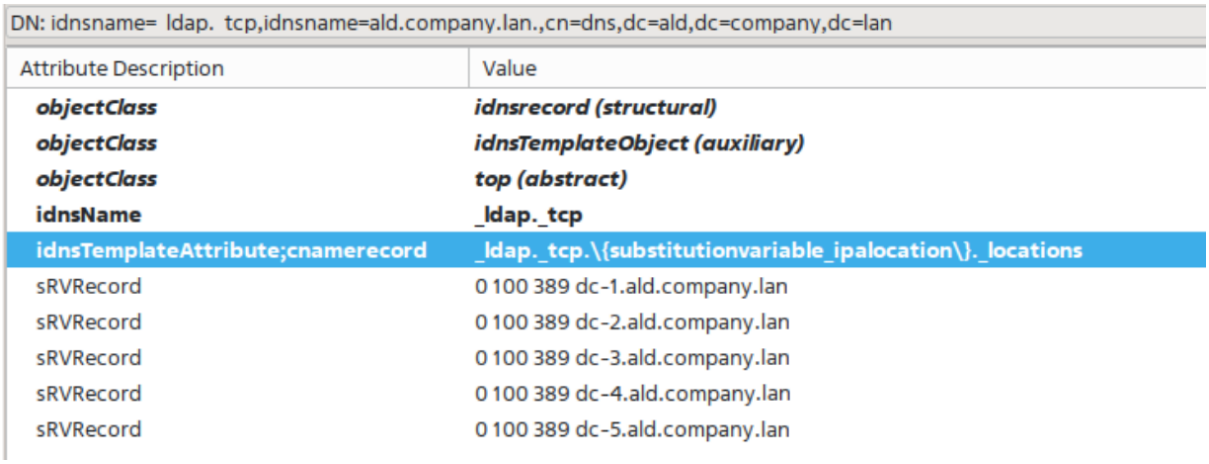


рис. 2.212 – Записи шаблонов, реализующие поддержку сайтов

Для привязки серверов к локациям у дочерних записей DN «cn=servers,cn=dns,dc=ald,dc=company,dc=lan» задано значение атрибута «idnsSubstitutionVariable;ipalocation», соответствующее имени сайта, в котором находится этот сервер.

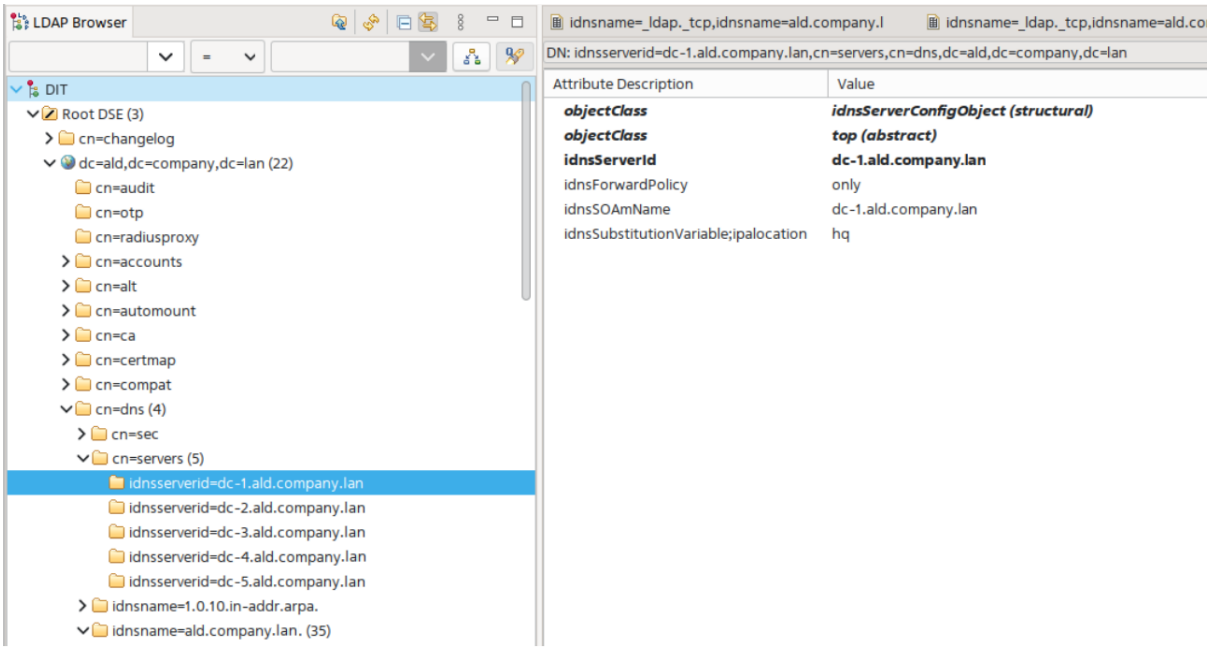


рис. 2.213 – Запись контроллера с информацией о принадлежности к сайту

Например, в домене из пяти контроллеров при обращении к DNS серверу 10.0.1.11 (dc-1.ald.company.lan) служба BIND9 выполняет подстановку и возвращает результат для _ldap._tcp.hq_locations.ald.company.lan, что соответствует сайту hq (англ. head quarters — головной офис). В список будут включены все контроллеры домена, но для серверов из сайта hq приоритет будет равен 0, а у всех остальных 50, поэтому служба SSSD в первую очередь будет обращаться к контроллерам из «своего» сайта, к которому компьютер косвенно привязан через

обслуживающий его DNS-сервер.

```
$ nslookup -q=SRV _ldap._tcp.ald.company.lan
Server:         127.0.0.1
Address:        127.0.0.1#53

_ldap._tcp.ald.company.lan    canonical name = _ldap._tcp.hq._locations.ald.company.
↪lan.
_ldap._tcp.hq._locations.ald.company.lan      service = 50 100 389 dc-4.ald.company.
↪lan.
_ldap._tcp.hq._locations.ald.company.lan      service = 0 100 389 dc-1.ald.company.
↪lan.
_ldap._tcp.hq._locations.ald.company.lan      service = 0 100 389 dc-2.ald.company.
↪lan.
_ldap._tcp.hq._locations.ald.company.lan      service = 50 100 389 dc-3.ald.company.
↪lan.
_ldap._tcp.hq._locations.ald.company.lan      service = 50 100 389 dc-5.ald.company.
↪lan.
```

После того, как список серверов определен, SSSD начинает искать доступный контроллер. Сначала SSSD выполняет поиск по серверам с приоритетом 0 из того же сайта, и только в случае неудачи переключается на остальные. Если компьютер работает в домене ALD Pro (FreeIPA), то проверка выполняется по одному серверу, а если провайдером выступает AD, то сервера проверяются группами по 5 штук, но доступность контроллера в любом случае определяется возможностью совершения анонимных LDAP-запросов.

```
SRCH base="" scope=0 filter="(objectClass=*)" attrs="* altServer namingContexts␣
↪supportedControl supportedExtension supportedFeature
```

Если служба SSSD сможет получить TGT/TGS билеты для доступа к LDAP, используя /etc/krb5.keytab файл хоста, то она выполнит еще несколько авторизованных запросов для получения дополнительной информации о диапазонах идентификаторов, участии хоста в группах и др. настройках.

```
...
SRCH base="fqdn=pc-1.ald.company.lan,cn=computers,cn=accounts,dc=ald,dc=company,dc=lan
↪" scope=0 filter="(objectClass=*)" attrs="objectClass cn memberOf ipaUniqueID"
...
```

Адрес сервера, с которым происходил обмен, будет сохранен как активный сервер, а FQDN остальных контроллеров, полученных через SRV-запрос, служба сохранит, как обнаруженные сервера (Discovered IPA servers).

Информацию о результатах автоматического обнаружения можно узнать с помощью утилиты sssctl из состава пакета sssd-tools. Данный инструмент администрирования взаимодействует со службой SSSD через Ответчик IFP по протоколу DBus.

```
$ sssctl domain-list
ald.company.lan
```

(continues on next page)

(продолжение с предыдущей страницы)

```
$ sssctl domain-status ald.company.lan
Online status: Online

Active servers:
IPA: dc-1.ald.company.lan

Discovered IPA servers:
- dc-1.ald.company.lan
- dc-3.ald.company.lan
- dc-2.ald.company.lan
- dc-4.ald.company.lan
- dc-5.ald.company.lan
- dc-1.ald.company.lan
```

8.6.3 Автоматическое обнаружение сервисов в библиотеке libkrb5

8.6.3.1 На обычном компьютере в домене

Для администрирования компьютеров в домене системные администраторы часто используют утилиты из пакета krb5-user, такие как kinit, klist и др., которые обращаются к функциям библиотеки libkrb5.

```
rood@pc-1:~# ldd $(which kinit)
...
        libkrb5.so.3 => /lib/x86_64-linux-gnu/libkrb5.so.3 (0x0000721d7c1b5000)
...
```

Настройки этой библиотеки задаются в файле /etc/krb5.conf, но при установке sssd в папку плагинов libkrb5 копируется файл sssd_krb5_locator_plugin.so, поэтому, если служба sssd включена, то при выполнении kinit учитываются настройки из /etc/sssd/sssd.conf.

```
rood@pc-1:~# dpkg -S /usr/lib/x86_64-linux-gnu/krb5/plugins/libkrb5/sssd_krb5_locator_
↪plugin.so
sssd-common: /usr/lib/x86_64-linux-gnu/krb5/plugins/libkrb5/sssd_krb5_locator_plugin.
↪so
```

На доменном компьютере в файле sssd.conf в секции [domain/ald.company.lan] в параметре ipa_server первым значением по умолчанию идет «_srv_», поэтому служба sssd выполняет поиск сервера kdc через обращение к SRV записям _kerberos._tcp.ald.company.lan

```
[domain/ald.company.lan]
...
ipa_server = _srv_, dc-1.ald.company.lan
...
```

После автоматического обнаружения сервера провайдер аутентификации (auth-provider) службы sssd записывает IP-адрес контроллера и FQDN-имена еще двух запасных серверов в файл kdcinfo

и функции Kerberos будут использовать эти данные для подключения.

```
 rood@pc-1:~# sssctl domain-status ald.company.lan
 Online status: Online

 Active servers:
 IPA: dc-1.ald.company.lan

 Discovered IPA servers:
 - dc-5.ald.company.lan
 - dc-1.ald.company.lan
 - dc-4.ald.company.lan
 - dc-3.ald.company.lan
 - dc-2.ald.company.lan
 - dc-1.ald.company.lan

# cat /var/lib/sss/pubconf/kdcinfo.ALD.COMPANY.LAN
10.0.1.11
dc-4.ald.company.lan
dc-3.ald.company.lan
```

Например, при выполнении kinit запрос сначала будет направляться к серверу с IP-адресом 10.0.1.11, в случае его недоступности к серверу dc-4.ald.company.lan, далее к dc-3.ald.company.lan и только в случае недоступности всех трех серверов запрос завершится ошибкой.

Если в параметре ipa_server убрать значение «_srv_» и оставить только FQDN контроллера, через который этот хост был введен в домен, то после перезапуска sssd все Kerberos-запросы пойдут только через указанный сервер.

Если службу sssd выключить, то библиотека libkrb5 будет выполнять поиск контроллера штатными механизмом. По умолчанию в файле /etc/krb5.conf, в секции [libdefaults] установлен параметр dns_lookup_kdc = true, поэтому поиск сервера будет выполняться через обращение к SRV-записям, но отличие будет состоять в том, что libkrb5 не поддерживает кэширования, поэтому автоматическое обнаружение будет срабатывать каждый раз при обращении к функциям библиотеки.

```
rood@pc-1:~# cat /etc/krb5.conf
[libdefaults]
...
  dns_lookup_kdc = true
...
```

Файл конфигурации krb5.conf позволяет отключать автоматическое обнаружение, тогда с помощью параметра kdc можно будет указать конкретный сервер

```
rood@pc-1:~# cat /etc/krb5.conf
[libdefaults]
...
  dns_lookup_kdc = false
...
[realms]
  ALD.COMPANY.LAN = {
```

(continues on next page)

(продолжение с предыдущей страницы)

```
    kdc = dc-1.ald.company.lan
...
  }
```

8.6.3.2 На контроллере домена

На контроллерах домена kinit ведет себя иначе, чем на доменных компьютерах, т.к. кроме службы sssd устанавливается еще и служба winbind, у которой есть собственный плагин winbind_krb5_locator.so.

```
admin@dc-1:~$ ls -l /usr/lib/x86_64-linux-gnu/krb5/plugins/libkrb5/
итого 36
-rw-r--r-- 1 root root 19248 ноя 22 2021 sssd_krb5_locator_plugin.so
-rw-r--r-- 1 root root 14840 июн 6 2022 winbind_krb5_locator.so

admin@dc-1:~$ dpkg -S /usr/lib/x86_64-linux-gnu/krb5/plugins/libkrb5/winbind_krb5_
↪locator.so
winbind: /usr/lib/x86_64-linux-gnu/krb5/plugins/libkrb5/winbind_krb5_locator.so
```

Активный контроллер домена, предлагаемый службой winbind можно узнать через команду dsgetdcname утилиты wbinfo:

```
root@dc-3:~# wbinfo --dsgetdcname ALD
dc-2.ald.company.lan
\\10.0.1.12
1
7afac3a6-d0e1-4bbf-85ab-04710eeee49e
ald.company.lan
ald.company.lan
0xe00003fd
Default-First-Site-Name
Default-First-Site-Name

root@dc-3:~# env KRB5_TRACE=/dev/stdout kinit admin <<< AstraLinux | grep 10.0.1
[23264] 1690099753.374402: Initiating TCP connection to stream 10.0.1.12:88
[23264] 1690099753.374403: Sending TCP request to stream 10.0.1.12:88
[23264] 1690099753.374404: Received answer (542 bytes) from stream 10.0.1.12:88
[23264] 1690099753.374405: Terminating TCP connection to stream 10.0.1.12:88
[23264] 1690099753.374432: Initiating TCP connection to stream 10.0.1.12:88
[23264] 1690099753.374433: Sending TCP request to stream 10.0.1.12:88
[23264] 1690099753.374434: Received answer (1587 bytes) from stream 10.0.1.12:88
[23264] 1690099753.374435: Terminating TCP connection to stream 10.0.1.12:88
```

В поставке Samba идет также утилита net, которая тоже выдает информацию о KDC, но ее результаты расходятся с тем, какой сервер фактически используют утилиты kinit.

```
root@dc-1:~# net ads info
Failed to get server's current time!
LDAP server: 10.0.1.13
LDAP server name: dc-3.ald.company.lan
```

(continues on next page)

(продолжение с предыдущей страницы)

```
Realm: ALD.COMPANY.LAN
Bind Path: dc=ALD,dc=COMPANY,dc=LAN
LDAP port: 389
Server time: Чт, 01 янв 1970 03:00:00 MSK
KDC server: 10.0.1.12
Server time offset: 0
Last machine account password change: Чт, 01 янв 1970 03:00:00 MSK
```

8.6.4 Автоматическое обнаружение сервисов в клиенте IPA

Для автоматизации управления доменом системные администраторы используют утилиту ipa, которая обращается к одному из контроллеров домена через REST API, используя прозрачную Kerberos-аутентификацию. После ее использования в связке ключей можно увидеть появление сервисного билета на доступ к службе HTTP конкретного контроллера.

```
admin@pc-1:~$ kinit admin
Password for admin@ALD.COMPANY.LAN:
admin@pc-1:~$ klist
Ticket cache: KEYRING:persistent:491200000:krb_ccache_mkw306I
Default principal: admin@ALD.COMPANY.LAN

Valid starting       Expires              Service principal
02.04.2025 23:04:53  03.04.2025 23:04:50  krbtgt/ALD.COMPANY.LAN@ALD.COMPANY.LAN

admin@pc-1:~$ ipa user-show admin
  Имя учётной записи пользователя: admin
  Фамилия: Administrator
  Домашний каталог: /home/admin
  Оболочка входа: /bin/bash
  Псевдоним учётной записи: admin@ALD.COMPANY.LAN, root@ALD.COMPANY.LAN
  UID: 491200000
  ID группы: 491200000
  Учётная запись отключена: False
  Link to department: ou=ald.company.lan,cn=orgunits,cn=accounts,dc=ald,dc=company,
↪dc=lan
  Пароль: True
  Участник групп: admins, trust admins, lpadmin
  Роли: ALDPRO - Main Administrator
  Доступные ключи Kerberos: True

admin@pc-1:~$ klist
Ticket cache: KEYRING:persistent:491200000:krb_ccache_mkw306I
Default principal: admin@ALD.COMPANY.LAN

Valid starting       Expires              Service principal
02.04.2025 23:04:59  03.04.2025 23:04:50  HTTP/dc-1.ald.company.lan@ALD.COMPANY.LAN
02.04.2025 23:04:53  03.04.2025 23:04:50  krbtgt/ALD.COMPANY.LAN@ALD.COMPANY.LAN
```

Утилита ipa определяет сервер для подключения в следующем порядке:

1. Сначала используется значение директивы xmlrpc_uri из файла /etc/ipa/default.conf

2. Если сервер, указанный в default.conf, окажется недоступен, утилита ipa через запрос SRV записей _ldap._tcp.DOMAIN получит список всех доступных серверов и будет перебирать их случайным образом.

На контроллерах домена в файле /etc/ipa/default.conf указан также параметр ldap_uri, который определяет адрес для подключения бэкенда REST API к LDAP-каталогу. По умолчанию предполагается, что бэкенд подключается только к той службе каталога, которая работает на том же сервере, поэтому подключение выполняется через UNIX-сокет по ldapi (LDAP Over Interprocess Communication facilities, IPC).

```
rood@dc-1:~# cat /etc/ipa/default.conf
...
ldap_uri = ldapi://%2Frun%2Fslapd-ALD-COMPANY-LAN.socket
...
```

8.6.5 Автоматическое обнаружение сервисов клиентом LDAP

Для работы компьютера в домене утилиты OpenLDAP (ldapsearch, ldapmodify и др.) не требуются, но пакет ldap-utils устанавливается и настраивается автоматически для упрощения отладки возникающих проблем. За настройку LDAP-клиента отвечает скрипт ipaclient/install/client.py, который устанавливает в файле /etc/ldap/ldap.conf следующие параметры по умолчанию:

- **TLS_CACERT /etc/ssl/certs/ca-certificates.crt**

  Определяет путь к файлу с цепочкой SSL-сертификатов для проверки безопасности подключения по LDAPS и при использовании команды StartTLS.

- **URI ldaps://dc-1.ald.company.lan**

  Определяет протокол подключения LDAPS и путь к LDAP-серверу по умолчанию. Значение в файле соответствует адресу контроллера домена, через который был выполнен ввод машины в домен. Это значение не обновляется динамически, поэтому при недоступности данного сервера утилиты перестанут работать без указания другого сервера в явном виде через параметр `-H`.

- **BASE dc=ald,dc=company,dc=lan**

  Определяет DN записи, которая будет использоваться в качестве базы поиска по умолчанию.

- **SASL_MECH GSSAPI**

  Определяет механизм аутентификации, по умолчанию используется GSSAPI, т. е. Kerberos-аутентификация через связку ключей Linux.

Для обеспечения отказоустойчивости скриптов автоматизации вы можете использовать, например, результат автоматического обнаружения сервисов из файла kdcinfo:

```
ldapsearch -H ldaps://$(tail -n 1 /var/lib/sss/pubconf/kdcinfo.ALD.COMPANY.LAN)
```

8.6.6 Автоматическое обнаружение сервисов ALD Pro

Автоматическое определение сервисов для подключения агентов Salt Minion и Zabbix выполняет скрипт /opt/rbta/aldpro/client/bin/aldpro-service-discovery.py. Агент групповых политик Salt Minion Standalone выполняет самостоятельно.

Для определения списка серверов скрипт aldpro-service-discovery.py запрашивает SRV-записи _ldap._tcp.DOMAIN, в качестве FQDN домена используя значение параметра domain секции global из файла /etc/ipa/default.conf. В предыдущих реализациях в список вносились только контроллеры из того же сайта, с версии 1.4.0 скрипт вносит в начало списка контроллеры из того же сайта, а в конец добавляет все остальные.

Агент Zabbix отвечает за передачу данных на сервер мониторинга. Он может работать как в активном, так и в пассивном режимах. Настройки подключения к серверу находятся в файле /etc/zabbix/zabbix_agentd.conf.d/00-servers.conf Для определения сервера мониторинга скрипт aldpro-service-discovery.py выполняет запрос к REST API /discover/api/ds/monitoring/servers.

Агент Syslog-NG отвечает за передачу событий в подсистему журналирования. Он не предусматривает поддержки автоматического обнаружения сервисов и конфигурируется через SaltStack, за что отвечает скрипт syslogng_install.sls из папки /etc/salt/states/aldpro/automations/subsystems/projects/.

8.7 Отладка репликации

8.7.1 Проверка состояния репликации

8.7.1.1 Получение сводной информации о соглашениях (dsconf)

Утилита dsconf показывает информацию по соглашениям о репликации текущего сервера, запускать нужно от имени **root**:

```
localadmin@dc-2:~$ sudo dsconf -j ALD-COMPANY-LAN replication status --suffix dc=ald,
↪dc=company,dc=lan
{
   "type": "list",
   "items":  [
       {
            "agmt-name": [
```

(continues on next page)

(продолжение с предыдущей страницы)


```
                "meTodc-1.ald.company.lan"
            ],
            "replica": [
                "dc-1.ald.company.lan:389"
            ],
            "replica-enabled": [
                "on"
            ],
            "update-in-progress": [
                "FALSE"
            ],
            "last-update-start": [
                "20231121133318Z"
            ],
            "last-update-end": [
                "20231121133318Z"
            ],
            "number-changes-sent": [
                "3:3/19 "
            ],
            "number-changes-skipped": [
                "unavailable"
            ],
            "last-update-status": [
                "Error (0) Replica acquired successfully: Incremental update succeeded
↪"
            ],
            "last-init-start": [
                "19700101000000Z"
            ],
            "last-init-end": [
                "19700101000000Z"
            ],
            "last-init-status": [
                "unavailable"
            ],
            "reap-active": [
                "0"
            ],
            "replication-status": [
                "Not in Synchronization: supplier (655ca929000300030000) consumer␣
↪(Unavailable) State (green) Reason (error (0) replica acquired successfully:␣
↪incremental update succeeded)"
            ],
            "replication-lag-time": [
                "unavailable"
            ]
        }
    ]
}
```


Приведенную информацию можно представить в формате сводной таблицы, но потребуется установить пакет **jq** и использовать комплексную команду форматирования:

```
localadmin@dc-2:~$ sudo apt install jq -y;
localadmin@dc-2:~$ sudo -i;
localadmin@dc-2:~$ (printf 'SUFFIX \tAGREEMENT \tSTATE \tTIME-SINCE \tLDAP-STATUS \
↪tREPL-STATUS \n'; dsconf -j ALD-COMPANY-LAN replication list | jq '.items[]' -r |␣
↪xargs -P8 -i -- dsconf -j ALD-COMPANY-LAN repl-agmt list --suffix={} | jq '.items[].
↪attrs | (.nsds5replicalastupdatestatusjson[0] | fromjson) as $status | [.
↪nsds5replicaroot[0], .cn[0], $status.state, $status.date, $status.ldap_rc_text,
↪$status.repl_rc_text] | @tsv' -r | sort ) | column -s$'\t' -t;exit
```

```
localadmin@dc-2:~$ sudo -i;
root@dc-2:~# (printf 'SUFFIX \tAGREEMENT \tSTATE \tTIME-SINCE \tLDAP-STATUS \tREPL-STATUS \n'; dsconf -j ALD-COM
PANY-LAN replication list | jq '.items[]' -r | xargs -P8 -i -- dsconf -j ALD-COMPANY-LAN repl-agmt list --suffix
={} | jq '.items[].attrs | (.nsds5replicalastupdatestatusjson[0] | fromjson) as $status | [.nsds5replicaroot[0],
 .cn[0], $status.state, $status.date, $status.ldap_rc_text, $status.repl_rc_text] | @tsv' -r | sort ) | column -
s$'\t' -t;exit
SUFFIX                  AGREEMENT                STATE   TIME-SINCE           LDAP-STATUS   REPL-STATUS
dc=ald,dc=company,dc=lan  meTodc-1.ald.company.lan  green   2023-11-21T13:43:19Z  Success       replica acquired
Выход
localadmin@dc-2:~$
```

рис. 2.214 – Информация о соглашении в табличном виде.

8.7.1.2 Сравнение данных между двумя контроллерами (ds-replcheck)

Для проверки состояния репликации между двумя конкретными серверами RedHat предлагает использовать утилиту ds-replcheck. Секция «Supplier RUV» показывает данные, известные о Мастере (Поставщике) на Реплике (Потребителе), а секция «Replica RUV» наоборот показывает, данные о Реплике (Потребителе), известные на Мастере (Поставщике).

```
ds-replcheck online -D "cn=Directory Manager" -m ldap://dc-1.ald.company.lan:389 -r␣
↪ldap://dc-2.ald.company.lan:389 -b dc=ald,dc=company,dc=lan
```

```
localadmin@dc-2:~$ ds-replcheck online -D "cn=Directory Manager" -m ldap://dc-1.ald.
↪company.lan:389 -r ldap://dc-2.ald.company.lan:389 -b dc=ald,dc=company,dc=lan
Enter password:
================================================================================
Replication Synchronization Report (Tue Nov 21 16:55:30 2023)
================================================================================


Database RUV's
======================================================

Supplier RUV:
 {replica 3 ldap://dc-2.ald.company.lan:389} 655ca76b000000030000 655ca929000300030000
 {replica 4 ldap://dc-1.ald.company.lan:389} 655ca759000000040000 655cb587000000040000
 {replicageneration} 655ca758000000040000

Replica RUV:
 {replica 3 ldap://dc-2.ald.company.lan:389} 655ca76b000000030000 655caa96000000030000
 {replica 4 ldap://dc-1.ald.company.lan:389} 655ca759000000040000 655cb587000000040000
 {replicageneration} 655ca758000000040000

 Replication State: Supplier and Replica are in perfect synchronization
```

(continues on next page)

(продолжение с предыдущей страницы)

```
Entry Counts
=======================================================

Supplier: 2318
Replica: 2318

Tombstones
=======================================================

Supplier: 8
Replica: 8

Entry Inconsistencies
=======================================================

uid=admin,cn=users,cn=accounts,dc=ald,dc=company,dc=lan
-------------------------------------------------------
- Attribute 'krblastfailedauth' is different:
     Supplier:
       - Origin value: b'20231121123000Z'

     Replica:
       - Value:      20231121130311Z
       - State Info: krbLastFailedAuth;adcsn-655caa8f000000030000;vucsn-
↪655caa8f000000030000:
20231121130311Z
       - Date: Tue Nov 21 16:03:11 2023

krbprincipalname=ldap/dc- 1.ald.company.lan@ALD.COMPANY.LAN,cn=services,cn=accounts,
↪dc=ald,dc=company,dc=lan
-------------------------------------------------------------------------------------
↪-----------------------
- Replica missing attribute: "krbloginfailedcount"

idnsname=ald.company.lan.,cn=dns,dc=ald,dc=company,dc=lan
---------------------------------------------------------
- Attribute 'idnssoaserial' is different:
     Supplier:
       - Value:      1700571291
       - State Info: idnsSOAserial;adcsn-655ca896000100040000;vucsn-
↪655ca896000100040000:
1700571291
       - Date: Tue Nov 21 15:54:46 2023
     Replica:
       - Value:      1700571405
       - State Info: idnsSOAserial;adcsn-655ca90d000000030000;vucsn-
↪655ca90d000000030000:
1700571405
       - Date: Tue Nov 21 15:56:45 2023


idnsname=1.0.10.in-addr.arpa.,cn=dns,dc=ald,dc=company,dc=lan
-------------------------------------------------------------
- Attribute 'idnssoaserial' is different:
     Supplier:
       - Value:      1700571207
```

(continues on next page)

(продолжение с предыдущей страницы)

```
        - State Info: idnsSOAserial;adcsn-655ca846000900040000;vucsn-
↪655ca846000900040000:
1700571207
        - Date: Tue Nov 21 15:53:26 2023

     Replica:
        - Value:      1700571406
        - State Info: idnsSOAserial;adcsn-655ca90e000000030000;vucsn-
↪655ca90e000000030000:
1700571406
        - Date: Tue Nov 21 15:56:46 2023


krbprincipalname=DNS/dc- 1.ald.company.lan@ALD.COMPANY.LAN,cn=services,cn=accounts,
↪dc=ald,dc=company,dc=lan
--------------------------------------------------------------------------------
↪----------------------
- Replica missing attribute: "krbloginfailedcount"
krbprincipalname=ipa-dnskeysyncd/dc-1.ald.company.lan@ALD.COMPANY.LAN,cn=services,
↪cn=accounts,dc=ald,dc=company, dc=lan
--------------------------------------------------------------------------------
↪---------------------------------

- Replica missing attribute: "krbloginfailedcount"

krbprincipalname=DNS/dc- 2.ald.company.lan@ALD.COMPANY.LAN,cn=services,cn=accounts,
↪dc=ald,dc=company,dc=lan
--------------------------------------------------------------------------------
↪----------------------
- Supplier missing attribute: "krbloginfailedcount"

- Replica's State Info: krbLoginFailedCount;vucsn-655ca847000700030000: 0
- Date: Tue Nov 21 15:53:27 2023


krbprincipalname=ipa-dnskeysyncd/dc -2.ald.company.lan@ALD.COMPANY.LAN,cn=services,
↪cn=accounts,dc=ald,dc=company, dc=lan
--------------------------------------------------------------------------------
↪-----------------------------------
- Supplier missing attribute: "krbloginfailedcount"

- Replica's State Info:  krbLoginFailedCount;vucsn-655ca84e000100030000: 0
- Date: Tue Nov 21 15:53:34 2023


Result
============================================================

There are replication differences between Supplier and Replica
```

8.7.1.3 Сводная статистика по состоянию репликации (checkipaconsistency)

Внимание: В данном разделе представлена информация о том, как можно проверить целостность

домена с использованием скриптов `checkipaconsistency` из публичного pip-репозитория, но установка модулей из открытых источников без дополнительной проверки кода обычно запрещена политиками безопасности.

Для успешной установки нам потребуется выполнить вход в сессию **root** пользователя:

```
sudo -i
```

Теперь выполним установку пакета python3-pip, который также установит зависимости:

```
apt install python3-pip
```

Проверим работоспособность **pip** и установим утилиту **cipa**:

```
pip3 --version
pip3 install --user checkipaconsistency
```

Добавим ссылку на запуск по сокращенному имени **cipa**:

```
ln -s /root/.local/bin/cipa /usr/bin/cipa
```

Проверим репликацию. Чтобы пароль не сохранился в истории, вы можете отключить ее или добавить пробел в начало строки.

```
root@dc-2:~# set +o history
root@dc-2:~# cipa -d ald.company.lan -W 'AstraLinux'
Initial config saved to /root/.config/checkipaconsistency - PLEASE EDIT IT!
+--------------------+---------+---------+-------+
| FreeIPA servers:   | dc-2    | dc-1    | STATE |
+--------------------+---------+---------+-------+
| Active Users       | 1       | 1       | OK    |
| Stage Users        | 0       | 0       | OK    |
| Preserved Users    | 0       | 0       | OK    |
| Hosts              | 3       | 3       | OK    |
| Services           | 9       | 9       | OK    |
| User Groups        | 6       | 6       | OK    |
| Host Groups        | 1       | 1       | OK    |
| Netgroups          | 0       | 0       | OK    |
| HBAC Rules         | 2       | 2       | OK    |
| SUDO Rules         | 0       | 0       | OK    |
| DNS Zones          | 2       | 2       | OK    |
| Certificates       | 0       | 0       | OK    |
| LDAP Conflicts     | 0       | 0       | OK    |
| Ghost Replicas     | 0       | 0       | OK    |
| Anonymous BIND     | ROOTDSE | ROOTDSE | OK    |
| Microsoft ADTrust  | False   | True    | FAIL  |
| Replication Status | dc-1 0  | dc-2 0  | OK    |
+--------------------+---------+---------+-------+
root@dc-2:~# set -o history
```

8.7.2 Просмотр количества изменений по соглашению (ldapsearch)

Довольно полезным видится атрибут nsds5replicaChangesSentSinceStartup, который может показать, сколько изменений прошло через конкретное соглашение о репликации:

```
admin@dc-1:~$ ldapsearch -LLL -b 'cn=mapping tree,cn=config'
↪'(objectClass=nsDS5ReplicationAgreement)' nsds5replicaChangesSentSinceStartup |␣
↪perl -MMIME::Base64 -Mutf8 -pe 's/^([-a-zA-Z0-9;]+):(:\s+)(\S+)$/$1.$2.&decode_
↪base64($3)/e'
. . .
nsds5replicaChangesSentSinceStartup: 4:255/43681
```

8.7.3 Просмотр журнала изменений (dbscan)

В ходе расследования инцидентов вам может потребоваться заглянуть в журнал изменений, сделать это можно с помощью утилиты dbscan:

```
admin@dc-1:~$ sudo dbscan -f /var/lib/dirsrv/slapd-ALD-COMPANY-LAN/db/userRoot/
↪replication_changelog.db | head -n 15
dbid: 66158532000000040000
       encrypted: no
       replgen: 1712686215 Tue Apr 9 21:10:15 2024
       csn: 66158532000000040000
       uniqueid: 88e30e06-bf4e11ee-9d29cb43-d00727f4
       dn: cn=repl keep alive 4,dc=ald,dc=company,dc=lan
       operation: modify
               keepalivetimestamp: 20240409181015Z
               modifiersname: cn=Multisupplier Replication Plugin,cn=plugins,cn=config
               modifytimestamp: 20240409181015Z
               entryusn: 1394188

dbid: 66158534000000040000
       encrypted: no
```

8.7.4 Просмотр таблицы RUV (dsconf)

Посмотреть таблицу RUV можно с помощью команды `replication get-ruv` утилиты dsconf:

```
sudo dsconf -D 'cn=Directory Manager' -w AstraLinux ALD-COMPANY-LAN replication get-
↪ruv --suffix dc=ald,dc=company,dc=lan
```

```
 admin@dc-1:~$ sudo dsconf -D 'cn=Directory Manager' -w AstraLinux ALD-COMPANY-LAN␣
↪replication get-ruv --suffix dc=ald,dc=company,dc=lan
 RUV:        {replica 4 ldap://dc-1.ald.company.lan:389} 65b8bb9f000100040000␣
↪6643fc28000000040000
 Replica ID: 4
 LDAP URL:   ldap://dc-1.ald.company.lan:389
 Min CSN:    2024-01-30 09:04:31 1 0 (65b8bb9f000100040000)
 Max CSN:    2024-05-15 00:04:56 (6643fc28000000040000)
```

(continues on next page)

(продолжение с предыдущей страницы)

```
RUV:        {replica 3 ldap://dc-2.ald.company.lan:389} 65b8bba3000100030000␣
↪6616b4be000000030000
Replica ID: 3
LDAP URL:   ldap://dc-2.ald.company.lan:389
Min CSN:    2024-01-30 09:04:35 1 0 (65b8bba3000100030000)
Max CSN:    2024-04-10 15:48:14 (6616b4be000000030000
```

8.7.5 Просмотр конфликтов репликации (dsconf)

Конфликты репликации можно посмотреть через dsconf или прямыми запросами в LDAP:

```
sudo dsconf -D "cn=Directory Manager" ldap://dc-1.ald.company.lan repl-conflict list␣
↪dc=ald,dc=company,dc=lan
```

```
admin@dc-1:~$ sudo dsconf -D "cn=Directory Manager" ldap://dc-1.ald.company.lan repl-
↪conflict list dc=ald,dc=company,dc=lan
Enter password for cn=Directory Manager on ldap://dc-1.ald.company.lan:
There were no conflict entries found under the suffix
```

8.7.6 Повторная инициализация контроллера

Если восстановить целостность информации контроллера домена инкрементальным путем окажется невозможно, то нужно будет выполнить повторную инициализацию. Для этого вам необходимо с портала управления выполнить следующие действия:

1. Открыть страницу «Управление доменом > Сайты и службы > Соглашения о репликации».
2. Открыть соглашение о репликации, в котором участвует контроллер, восстановленный из резервной копии.
3. Нажать на кнопку «Переинициализировать контроллер домена» рядом с тем контроллером, который был восстановлен из резервной копии.

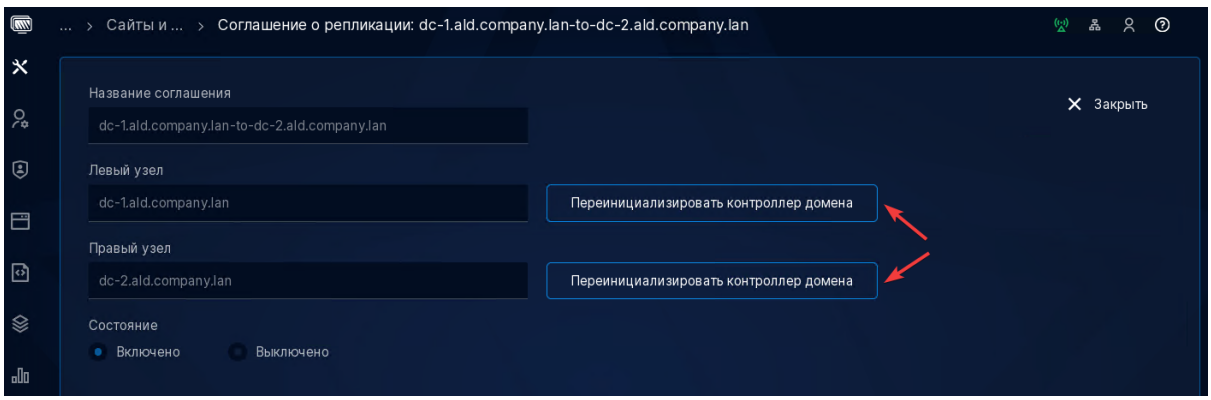


рис. 2.215 – Переинициализация контроллера домена

То же самое можно сделать и из командной строки. Для переинициализации заходим на поврежденную реплику и запускаем команду реинициализации:

```
admin@dc-1:~$ sudo ipa-replica-manage re-initialize --from dc-2.ald.company.lan
```

Где:

- ключ `re-initialize` — это подкоманда реинициализации реплики;
- ключ `--from` - указывает реплику, с которой нужно забрать исходные данные, например, dc-2.ald.company.lan.

8.7.7 Механика работы плагина топологии и односторонние соглашения о репликации

В домене FreeIPA все реплики являются «ведущими», поэтому при создании связи между двумя контроллерами репликация должна проходить как в прямом, так и в обратном направлении. Но время от времени при продвижении серверов до контроллеров домена возникают ошибки, которые приводят к тому, что устанавливаются «односторонние соглашения репликации».

Как вы уже знаете, в домене FreeIPA есть сегменты и соглашения. Напомним, что сегменты являются реплицируемой информацией и находятся в корневом суффиксе домена «cn=domain,cn=topology,cn=ipa,cn=etc,dc=ald,dc=company,dc=lan», а соглашения о репликации находятся в «cn=replica,cn=dc\3Dald\2Cdc\3Dcompany\2Cdc\3Dlan,cn=mapping tree,cn=config» и не реплицируются между контроллерами.

При создании связи между двумя серверами A и B должно быть создано два соглашения: на сервере A создается соглашение для передачи изменений серверу B и наоборот.

У соглашений о репликации есть атрибут ipaReplTopoManagedAgreementState, указывающий на то, управляются ли они плагином топологии или нет. Если соглашение управляется плагином топологии, то плагин топологии автоматически удалит это соглашение, если будет удален соответствующий сегмент топологии.

Проблема «односторонних соглашений репликации» возникает по причине «гонок» между двумя взаимообратными процессами:

1. первый процесс отвечает за создание соглашений о репликации для новой реплики;
2. второй процесс отвечает за удаление соглашений при удалении сегментов топологии.

Проблема решается увеличением таймаута nsslapd-topo-plugin-startup-delay, например, до 40 секунд, и будет применена в одном из следующих релизов. Но, если вам потребуется исправить «односторонние» соглашения на работающем стенде, то нужно выполнить следующие команды:

```
$ kinit admin
$ ldapsearch -LLL -o ldif-wrap=no -b "cn=domain,cn=topology,cn=ipa,cn=etc,dc=ald,
↪dc=company,dc=lan" "(ipaReplTopoSegmentDirection=left-right)" > left-right.ldif
$ cat left-right.ldif | sed 's/-to-/-from-/' | sed 's/left-right/right-left/' > right-
↪left.ldif
$ ldapadd -f right-left.ldif
```

Как видите, сначала мы выгружаем односторонние Сегменты в файл left-right.ldif. Затем делаем две подстановки «-to-» на «-from-» и «left-right» на «right-left», сохраняя результаты в новом файле. И на последнем шаге загружаем полученные записи в каталог. После чего останется только перезапустить службу каталога на проблемных контроллерах.

Допустим, у нас было одностороннее соглашение в направлении dc-1 => dc-2. В момент загрузки контроллеров будут выполнены следующие действия:

1. Плагин топологии dc-2 увидит rigth-left Сегмент в обратном направлении без соответствующего Соглашения о репликации и создаст его.

2. После создания Сегмента плагин топологии dc-2 добавит к Соглашению о репликации атрибут ipaReplTopoManagedAgreementState, чтобы это Соглашение можно было удалить через управление топологией.

3. Плагин топологии dc-1 обнаружит, что в домене есть два взаимообратных «Left-Right» и «Right-Left» Сегмента, поэтому Сегмент dc-2 => dc-1 он удалит, а для Сегмента dc-1 => dc-2 изменит направление с «Left-Right» на «Both».

8.8 Практика и тестирование

8.8.1 Практическая работа: Модуль 8. Управление топологией в домене

См. также *Требования, правила и цели выполнения практической работы*

8.8.1.1 Практические задания

Развернуть резервный контроллер домена **Стенд ALD Pro — dc-2.ald.company.lan**:

1. Убедитесь, что уровень защищенности максимальный, а целостность текущего пользователя 63.

2. Отключите службу NetworkManager.

3. Настройте сетевой интерфейс и разрешение имен через файлы /etc/network/interfaces и /etc/resolvconf:

- IP-адрес: 10.0.1.12/24
- Шлюз: 10.0.1.1
- Сервер DNS: 10.0.1.11
- Поисковый домен: ald.company.lan

4. Убедитесь, что имя сайта dl.astralinux.ru и имя контроллера домена dc-1 и dc-1.ald.company.lan разрешаются и пингуются.
5. Настройте имя машины и файл /etc/hosts в соответствии с рекомендациями.
6. Убедитесь, что в системе заданы корректные репозитории.
7. Убедитесь, что приоритеты пакетов настроены корректно.
8. Установите пакеты ALD Pro для клиентской машины.
9. Выполните ввод машины в домен со следующими параметрами (не забудьте отключить сохранение истории на время ввода команды):
   - Имя домена: ald.company.lan
   - Имя машины: dc-2
   - Имя учетной записи: admin
   - Пароль: AstraLinux
   - `--gui`
   - `--force`
10. Перезагрузите компьютер и войдите под доменной записью **admin**.
11. Выполните повышение роли сервера до контроллера домена через портал управления ALD Pro на странице *Управление доменом → Сайты и службы → Контроллеры домена* .
12. Проверьте состояние доменных служб.
13. Проверьте файл `/etc/resolv.conf`, убедитесь, что там есть запись для DNS-сервера 127.0.0.1.
14. Отключите DNSSEC.
15. Разрешите выполнение рекурсивных запросов и кэширование.
16. Войдите в портал управления dc-2, используя Kerberos-аутентификацию.

17. Проверьте соглашение о репликации на странице *Управление доменом → Сайты и службы → Соглашения о репликации* .

18. Проверьте статус репликации утилитами dsconf и скриптами проекта checkipaconsistency.

8.8.1.2 Практическая работа (с ответами)

Развернуть резервный контроллер домена **Стенд ALD Pro — dc-2.ald.company.lan**:

1. Убедитесь, что уровень защищенности максимальный, а целостность текущего пользователя 63:

```
cat /etc/astra/build_version
sudo astra-modeswitch getname
sudo pdp-id
```

2. Отключите службу NetworkManager:

```
sudo systemctl stop NetworkManager
sudo systemctl disable NetworkManager
sudo systemctl mask NetworkManager
sudo systemctl status NetworkManager
```

3. Настройте сетевой интерфейс и разрешение имен через файлы /etc/network/interfaces и /etc/resolvconf:

   - IP-адрес: 10.0.1.12/24
   - Шлюз: 10.0.1.1
   - Сервер DNS: 10.0.1.11
   - Поисковый домен: ald.company.lan

```
sudo vim /etc/network/interfaces
```

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
  address 10.0.1.12
  netmask 255.255.255.0
  gateway 10.0.1.1
```

```
sudo vim /etc/resolv.conf
```

```
nameserver 10.0.1.11
search ald.company.lan
```

```
sudo ip addr flush dev eth0
sudo systemctl restart networking
```

4. Убедитесь, что имя сайта dl.astralinux.ru и имя контроллера домена dc-1 и dc-1.ald.company.lan разрешаются и пингуются:

```
ping -c 2 dc-1
ping -c 2 dc-1.ald.company.lan
ping -c 2 dl.astralinux.ru
```

5. Настройте имя машины и файл /etc/hosts в соответствии с рекомендациями:

```
sudo vim /etc/hosts
```

```
127.0.0.1 localhost.localdomain localhost
10.0.1.12 dc-2.ald.company.lan dc-2
```

6. Убедитесь, что в системе заданы корректные репозитории в соответствующих файлах:

- Для ОС:

```
sudo vim /etc/apt/sources.list
```

```
deb http://dl.astralinux.ru/astra/frozen/1.7_x86-64/1.7.6/repository-base 1.7_x86-64␣
↪main contrib non-free
deb http://dl.astralinux.ru/astra/frozen/1.7_x86-64/1.7.6/repository-extended 1.7_x86-
↪64 main contrib non-free
```

- Для ALD Pro:

```
sudo vim /etc/apt/sources.list.d/aldpro.list
```

```
deb https://dl.astralinux.ru/aldpro/frozen/01/2.4.0 1.7_x86-64 main base
```

7. Убедитесь, что приоритеты пакетов настроены корректно:

```
Package: *
Pin: release n=1.7_x86-64
Pin-Priority: 900
```

8. Установите пакеты ALD Pro для клиентской машины:

```
sudo DEBIAN_FRONTEND=noninteractive apt-get install -y -q aldpro-client
```

9. Выполните ввод машины в домен со следующими параметрами (не забудьте отключить сохранение истории на время ввода команды):

   - Имя домена: ald.company.lan
   - Имя машины: dc-2

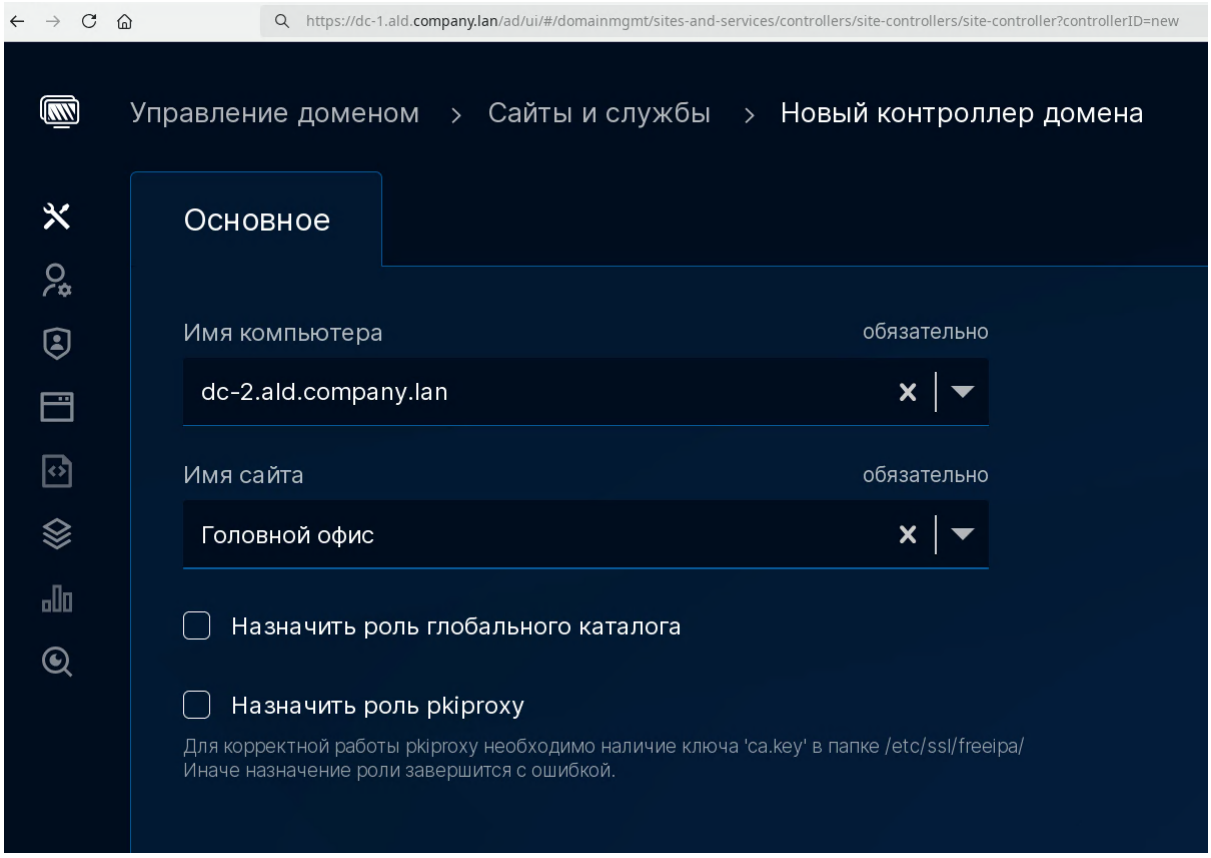
- Имя учетной записи: admin
- Пароль: AstraLinux
- `--gui`
- `--force`

```
set +o history
sudo /opt/rbta/aldpro/client/bin/aldpro-client-installer --domain ald.company.lan --
↪account admin --password 'AstraLinux' --host dc-2 --gui --force
set -o history
```

10. Перезагрузите компьютер и войдите под доменной записью **admin**:

```
sudo reboot
```

11. Выполните повышение роли сервера до контроллера домена через портал управления ALD Pro на странице *Управление доменом → Сайты и службы → Контроллеры домена*. Нажмите кнопку *Новый контроллер домена*, выберите из выпадающего списка *Имя компьютера* сервер dc-2.ald.company.lan, а из *Имя сайта* Головной офис.



12. Проверьте состояние доменных служб:

```
sudo ipactl status
```

13. Проверьте файл /etc/resolv.conf:

```
admin@dc-2:~$ cat /etc/resolv.conf ; echo
# auto-generated by IPA installer
search ald.company.lan
nameserver 127.0.0.1
```

14. Отключите DNSSEC:

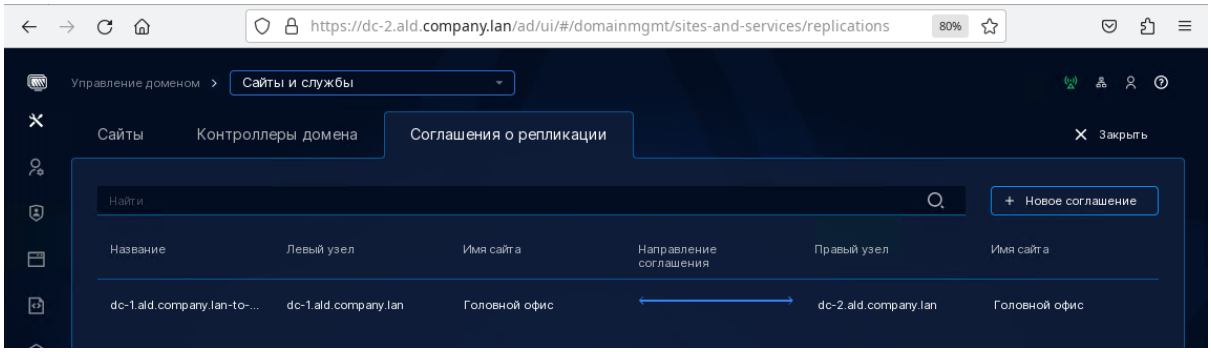
```
sudo sed -i 's/dnssec-validation yes/dnssec-validation no/g' /etc/bind/ipa-options-
↪ext.conf
```

15. Разрешите выполнение рекурсивных запросов и кэширование:

```
sudo tee -a /etc/bind/ipa-options-ext.conf > /dev/null <<EOT
allow-recursion { any; };
allow-query-cache { any; };
EOT
sudo systemctl restart bind9-pkcs11.service
```

16. Войдите в портал управления dc-2, используя Kerberos-аутентификацию.

17. Проверьте соглашение о репликации на странице *Управление доменом → Сайты и службы → Соглашения о репликации*.



18. Проверьте состояние репликации утилитой dsconf:

```
sudo dsconf -j ALD-COMPANY-LAN replication status --suffix dc=ald,dc=company,dc=lan

sudo apt install jq -y;
sudo -i;
(printf 'SUFFIX \tAGREEMENT \tSTATE \tTIME-SINCE \tLDAP-STATUS \tREPL-STATUS \n';␣
↪dsconf -j ALD-COMPANY-LAN replication list | jq '.items[]' -r | xargs -P8 -i --␣
↪dsconf -j ALD-COMPANY-LAN repl-agmt list --suffix={} | jq '.items[].attrs | (.
↪nsds5replicalastupdatestatusjson[0] | fromjson) as $status | [.nsds5replicaroot[0],␣
↪.cn[0], $status.state, $status.date, $status.ldap_rc_text, $status.repl_rc_text] |␣
↪@tsv' -r | sort ) | column -s$'\t' -t``;exit

ds-replcheck online -D "cn=Directory Manager" -m ldap://dc-1.ald.company.lan:389 -r␣
↪ldap://dc-2.ald.company.lan:389 -b dc=ald,dc=company,dc=lan
```

Проверьте состояние репликации скриптами проекта checkipaconsistency:

```
sudo -i
apt install python3-pip
pip3 --version && pip3 install --user checkipaconsistency
ln -s /root/.local/bin/cipa /usr/bin/cipa
set +o history
cipa -d ald.company.lan -W AstraLinux
set -o history
```

8.8.2 Тест: Модуль 8. Управление топологией в домене

Пожалуйста, ответьте на несколько вопросов, чтобы закрепить полученные знания. Правильные ответы для самопроверки вы сможете найти в конце списка. Обратите внимание, что у некоторых вопросов может быть несколько верных вариантов ответов.

8.8.2.1 Вопросы

Вопрос 1. Какой уровень защищенности должен быть на сервере, роль которого планируется повысить до контроллера домена ALD Pro?

a. Базовый «Орел»

b. Усиленный «Воронеж»

c. Максимальный «Смоленск»

d. Не имеет значения

Вопрос 2. Какую ветку репозитория необходимо использовать на сервере, роль которого планируется повысить до контроллера домена ALD Pro?

a. stable

b. frozen

c. latest

d. fixed

Вопрос 3. Какой пакет будет установлен на сервере в ходе повышения роли до контроллера домена?

a. aldpro

b. aldpro-mp

c. aldpro-dc

d. aldprodc

Вопрос 4. Кто инициирует процесс репликации в ALD Pro?

a. Поставщик

b. Потребитель

c. Могут оба

d. cron

Вопрос 5. Найдите корректные этапы процесса репликации в ALD Pro?

a. В сеансе репликации поставщик связывается с потребителем

b. Контроллер домена по таймеру связывается со своим партером по репликации

c. Поставщик запрашивает у потребителя сводную таблицу о состоянии репликации RUV

d. Потребитель определяет, есть ли более свежие данные у поставщика

e. Потребитель инициирует передачу данных

Вопрос 6. Расположите корректные этапы процесса репликации в ALD Pro в правильном порядке?

a. Поставщик инициирует передачу данных

b. Поставщик запрашивает у потребителя сводную таблицу о состоянии репликации RUV

c. Поставщик определяет, есть ли более свежие данные для потребителя

d. В сеансе репликации поставщик связывается с потребителем

Ответы на вопросы

1. c
2. b
3. b
4. a
5. a, c
6. D > B > C > A

8.9 Заключение

В этом модуле мы разобрались с вопросами репликации, что крайне важно для сопровождения доменов, так как даже на малых предприятиях обычно устанавливают как минимум два контроллера. Далее мы затронем вопросы интеграции с MS AD, без чего не обойдется, пожалуй, ни один проект импортозамещения.

8.10 Дополнительные источники информации

- Теорема CAP
- Рекомендуем ознакомиться также с видеороликами нашего хорошего товарища на канале «ИТ-Проповедник». Учитывайте, что ролики записаны для версии 2.2.1, поэтому часть информации потеряла актуальность, но в целом они могут быть крайне полезны.
    - Установка резервного контроллера домена ALD PRO 2.2.1

8.11 Обратная связь

Если остались вопросы, то их всегда можно задать в специальной теме.

Модуль 9. Миграция и гибридное развертывание с MS AD

9.1 Введение

Из этого модуля вы узнаете о том, как подступиться к проектам импортозамещения и чем вам помогут доверительные отношения Kerberos. Мы рассмотрим, зачем в ALD Pro нужен модуль глобального каталога и как он работает. В конечном итоге, чем больше инструментов гибридного развертывания вы задействуете, тем проще вам будет выполнить миграцию инфраструктуры.

9.2 Стратегии миграции

9.2.1 Возможности гибридного развертывания

В настоящее время многие компании задумываются над миграцией в инфраструктуру Linux и вместо Microsoft AD DS планируют использовать службу каталога с открытым исходным кодом FreeIPA или более функциональные решения на ее основе, такие как ALD Pro. Но сколь бы ни были амбициозны планы по миграции, перевести сразу все сервисы и все рабочие места на Linux может оказаться непосильной задачей даже для очень сильных команд, потому что в один момент нужно сразу очень много ресурсов:

- Нужны квалифицированные администраторы, чтобы перенастроить информационные системы для работы в Linux-домене. Не всегда есть готовые решения, особенно для корпоративных приложений, поэтому требуются дополнительные исследования и длительное тестирование.

- Нужны разработчики, чтобы внести изменения в пользовательские данные (как минимум изменить идентификаторы безопасности) или даже в программный код. Причем по некоторым корпоративным системам могут быть утеряны компетенции и отсутствовать документация разработчика, что потребует выполнения обратного проектирования.

- Нужны инженеры поддержки, чтобы обучить пользователей работе с новыми клиентскими приложениями в новой операционной системе и ответить на все их вопросы.

Учитывая нехватку кадров, при планировании импортозамещения следует максимально полно использовать все доступные механизмы гибридного развертывания для организации поэтапного перехода:

- Доверительные отношения. Позволяют пользователям одного домена получать доступ к ресурсам другого домена.

  Между доменами MS Active Directory и ALD Pro (FreeIPA) возможно установление одно- и двусторонних транзитивных доверительных отношений лес к лесу и одно- и двусторонних нетранзитивных доверительных отношений домен к домену (external trusts).

  Для полноценной поддержки двусторонних отношений на стороне ALD Pro реализована функция глобального каталога, чего нет в ванильной версии FreeIPA, — это позволяет на стороне Windows при назначении прав доступа в стандартных оснастках Windows выполнять поиск по объектам доверенного леса ALD Pro.

- Синхронизация данных каталогов. Позволяет обеспечить доступ пользователей к серверным приложениям по логину/паролю вне зависимости от того, к какому каталогу подключена эта система напрямую. В продукте ALD Pro реализован модуль синхронизации, который позволяет поддерживать целостность информации двух доменов, включая пароли. Захват паролей

на стороне AD выполняется с помощью фильтра паролей, передача данных выполняется с использованием асимметричного шифрования.

- Прямая работа Linux машин в домене MS AD. Позволяет пользователям AD входить в Linux машины и выполнять прозрачную аутентификацию при обращении к «керберизированным» сервисам в домене. Реализация сценария возможна с помощью старой клиентской части Winbind или улучшенной альтернативы SSSD. Служба SSSD поддерживает и более экзотические сценарии работы, например, когда машина находится в двух доменах сразу.

- Прямая работа Windows машин в домене ALD Pro (FreeIPA). Позволяет пользователям ALD Pro входить на рабочие станции Windows и выполнять прозрачную аутентификацию при обращении к «керберизированным» сервисам в домене. Реализация сценария возможна в силу нативной поддержки Kerberos-доменов со стороны Windows и расширенных возможностей FreeIPA, которые дает интеграция с Samba AD.

- Реализация единого входа для веб-приложений через обмен цифровыми удостоверениями по протоколам OAuth, OpenID, SAML.

Задачи гибридного развертывания следует рассматривать в следующих трех аспектах:

- серверные приложения,

- рабочие места клиентов и

- типы учетных записей пользователей.

Серверные приложения по сложности миграции можно классифицировать на несколько групп:

- **Непереносимые:**

  - Приложения, которые работают только с AD DS, например, MS Exchange. Потребуют замены на аналоги (RuPost, CommuniGate или VK WorkMail).

  - Приложения, которые интегрированы с каталогом через NTLM, могут потребовать существенных доработок, хотя на контроллерах домена для работы в доверительных отношениях интегрирован компонент Samba AD, который предоставляет этот интерфейс.

- **Переносимые:**

  - Приложения, которые интегрированы с каталогом через Kerberos, например, 1С, и, соответственно, вся линейка этих учетных систем.

  - Приложения, которые интегрированы через LDAP, например, Bitrix24.

  - Web-приложения, которые интегрированы через обмен цифровыми удостоверениями по протоколам OAuth, OpenID, SAML.

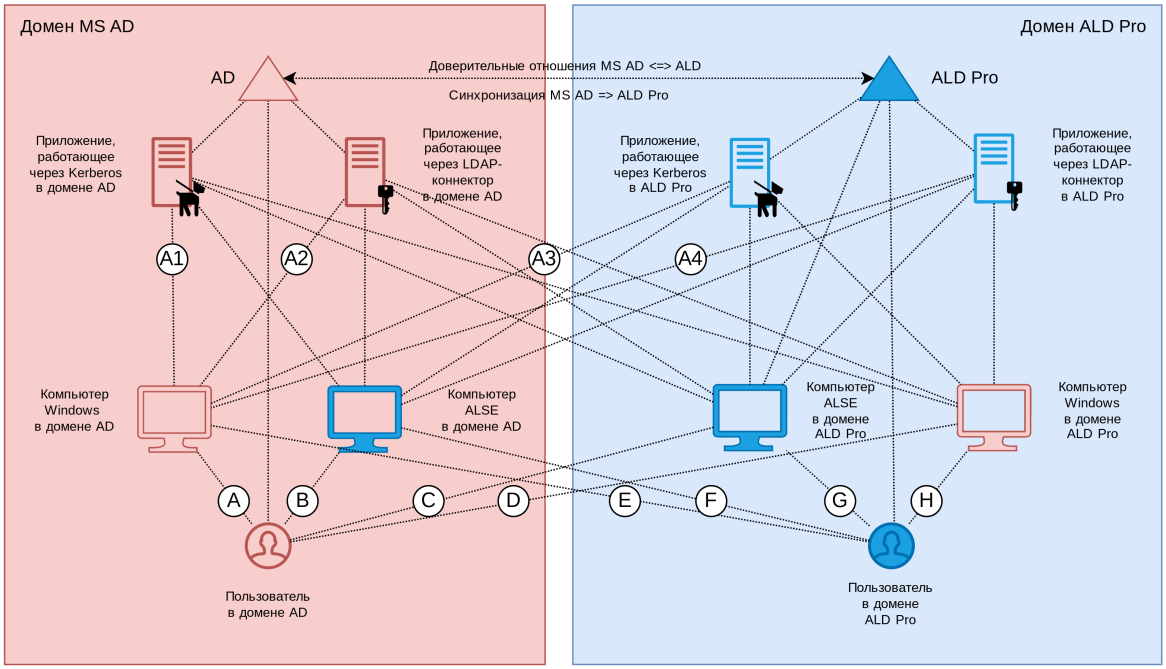


рис. 2.216 – Основные способы подключения к информационным системам

В качестве рабочих мест следует рассмотреть следующие варианты:

- MS Windows в домене MS AD — это исходные данные для начала миграции.
- Astra Linux в домене MS AD через SSSD или Winbind — промежуточный сценарий, который позволит начать замену клиентских приложений.
- MS Windows в домене ALD Pro — промежуточный этап, который поможет в тех случаях, когда невозможно заменить или переписать какие-то специфичные клиентские приложения. Вместе с тем не забывайте, что в некоторых случаях может помочь среда для запуска Windows-приложений Wine.
- Astra Linux в домене ALD Pro через SSSD — это то, к чему рекомендуется прийти.

Учетные записи могут быть:

- Из домена ALD Pro (FreeIPA);
- Из домена MS AD (в т.ч. Samba AD DC 4.7+);
- Локальные (не рекомендуется).

Таким образом, мы получаем множество способов входа в компьютер и подключения к серверным приложениям, причем для каждого приложения и каждой группы пользователей следует прорабатывать свой индивидуальный план миграции, см. рис. 2.216.

9.2.2 Типовые сценарии развертывания

9.2.2.1 Сценарий «Сервисный домен»

В домен ALD Pro вводятся только Linux-серверы, обеспечивающие работу корпоративных информационных систем, пользователи и сами корпоративные приложения остаются в домене Active Directory.

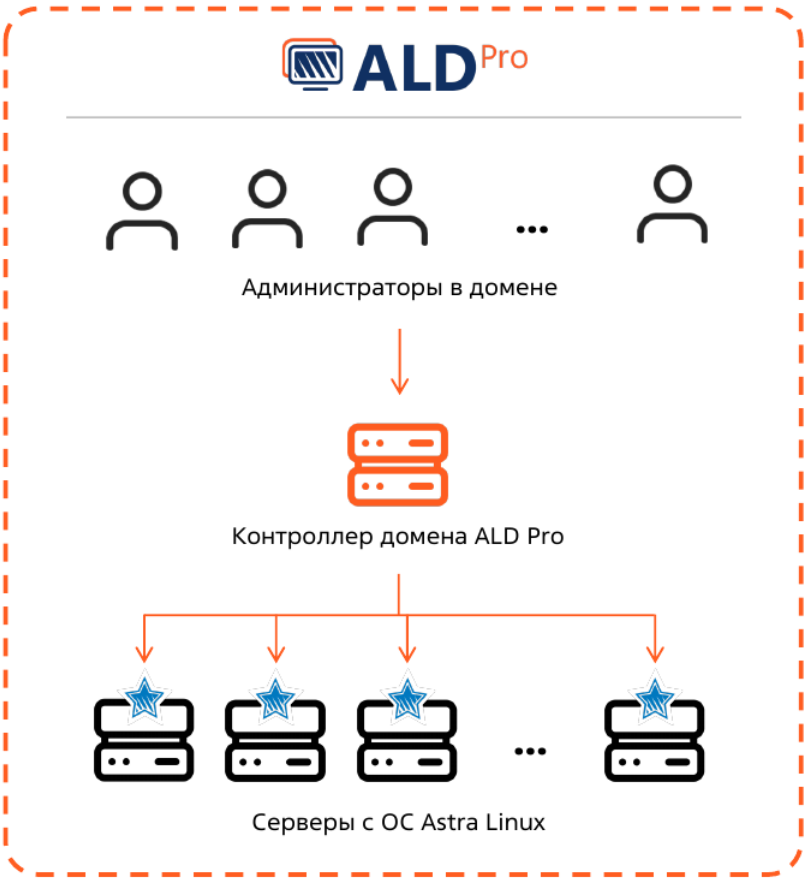


рис. 2.217 – Архитектура сервисного домена

Такой сценарий развертывания упрощает администрирование серверной группировки, предоставляя следующие возможности:

- Централизованное управление учетными записями администраторов системы, включая их SSH-ключи.
- Автоматизация настройки ОС Astra Linux на серверах через механизм групповых политик ALD Pro (через Salt-скрипты).
- Контроль доступа к серверам через правила HBAC и SUDO.

9.2.2.2 Сценарий «Ресурсный домен»

В домен ALD Pro переносятся не только серверы, но и «керберизированные» сетевые приложения (ресурсы), такие как файловые серверы, корпоративные веб-приложения и т.д. Пользователи и рабочие места сотрудников остаются в домене AD, доступ к ресурсам осуществляется через механизм доверительных отношений.

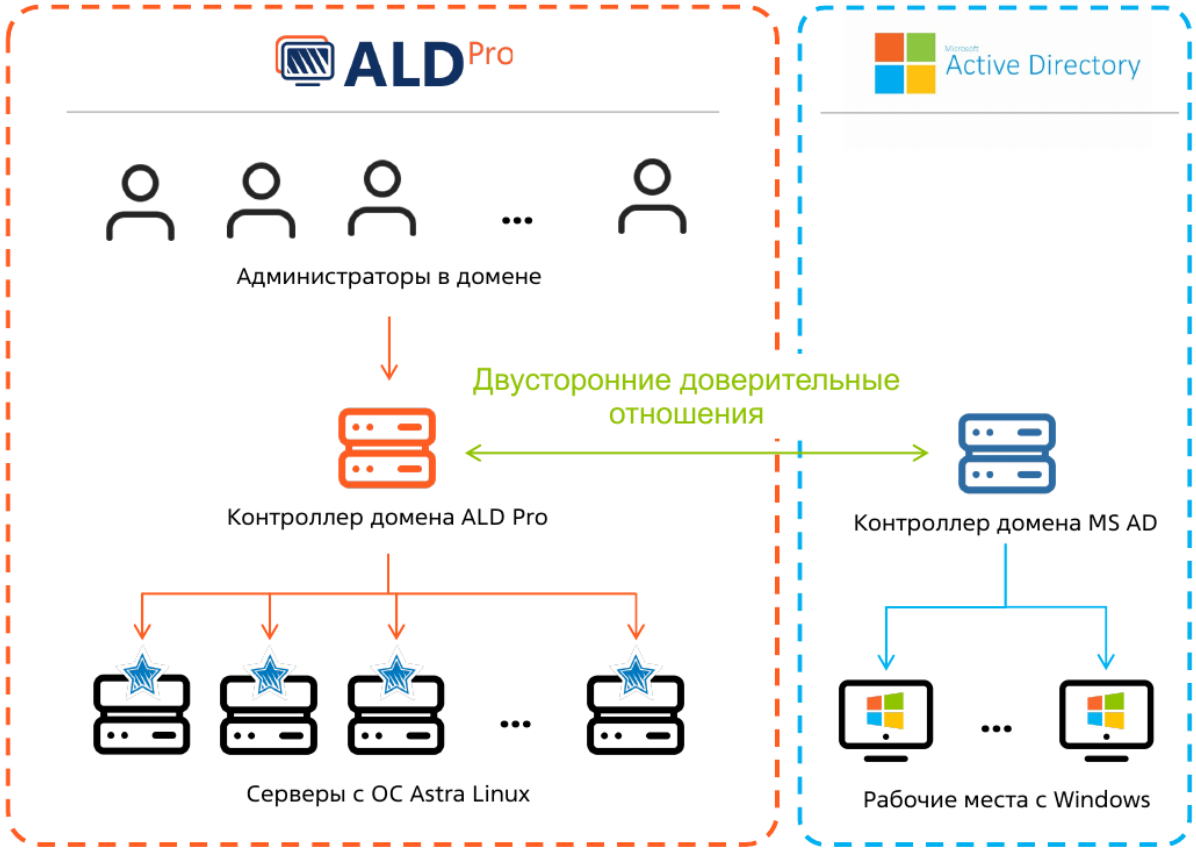


рис. 2.218 – Архитектура ресурсного домена

Такой сценарий развертывания потребует участия разработчиков и квалифицированных администраторов, но станет значительным шагом в направлении импортозамещения и предоставит следующие возможности:

- Прозрачная аутентификация пользователей при обращении к ресурсам в Linux-домене через механизм доверительных отношений.
- Возможность постепенного переноса корпоративных приложений в Linux-инфраструктуру по мере их готовности.
- Возможность переноса учетных записей пользователей в Linux-домен, как только все нужные приложения окажутся в новой инфраструктуре.

9.2.2.3 Сценарий «Управление рабочими местами»

Сценарий предполагает, что в домен ALD Pro вводятся только рабочие места, и это позволяет обеспечить централизованное управление инфраструктурой Linux-компьютеров, а пользователи и ресурсы остаются в домене MS AD. Вход пользователей в компьютер осуществляется через механизм доверительных отношений.

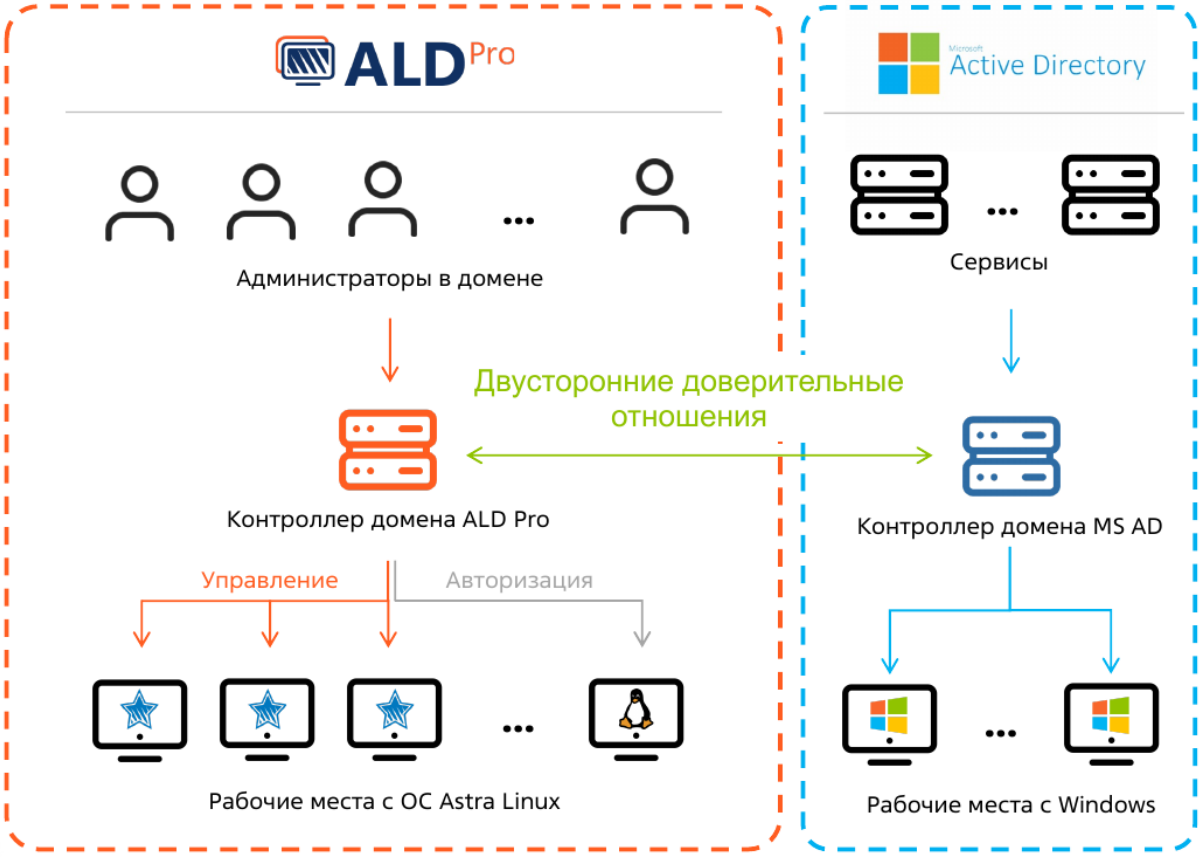


рис. 2.219 – Архитектура для управления рабочими местами

Этот сценарий позволит быстро заместить клиентскую инфраструктуру и отлично подходит организациям, у которых большинство корпоративных приложений не имеют толстых клиентов и работают как веб-приложения из браузера. Такой сценарий развертывания предоставит следующие возможности по управлению рабочими местами:

- Автоматизация настройки рабочих станций через механизм групповых политик.
- Контроль доступа к рабочим станциям через правила HBAC и SUDO.
- Установка ОС на рабочие станции пользователей по сети.

Служба каталога FreeIPA поддерживает транзитивные двусторонние доверительные отношения с лесом Active Directory, извлекая информацию обо всех поддоменах. Первый и второй уровни работают по умолчанию, для поддержки третьего и последующих уровней требуется внесение до-

полнительных настроек.

Права доступа и политики в домене ALD Pro можно будет назначать как на отдельных пользователей доверенных доменов, так и на группы, используя механизм внешних групп (External Groups).

Для интеграции приложений потребуется минимум усилий, т.к. Керберос-билеты пользователям будут выдаваться центрами распространения ключей MS AD на тех же принципалов, с теми же PAC-сертификатами, содержащими полный набор идентификаторов безопасности, в т.ч. по участию в универсальных группах.

При этом централизованное управление инфраструктурой рабочих станций через привычный механизм групповых политик позволит удержать затраты по администрированию на прежнем уровне. Из коробки сотрудникам будут доступны сотни параметров и более 800 атрибутов для конфигурирования рабочих станций Astra Linux, а для управления корпоративными приложениями возможно будет разработать дополнительные параметры.

Если в компании используются толстые клиенты, разработанные под операционную систему Windows, в некоторых случаях их получится запускать из-под Linux, используя среду для запуска Windows-приложений Wine, с пробросом Kerberos-ключей через GSSAPI и связку ключей ядра Linux. Wine - это не самый стабильный продукт, но в случае бизнес-приложений, которые не используют аппаратное ускорение и другие продвинутые API, этот способ может вполне сработать.

9.2.2.4 Сценарий «Полное развертывание инфраструктуры»

В данном сценарии пользователи и сервисы работают в домене ALD Pro, интеграция с Active Directory не используется. Этот сценарий не является примером гибридного развертывания, его следует считать результатом полного импортозамещения, когда все пользователи и сервисы переведены на Linux и интеграция с инфраструктурой Windows больше не требуется.

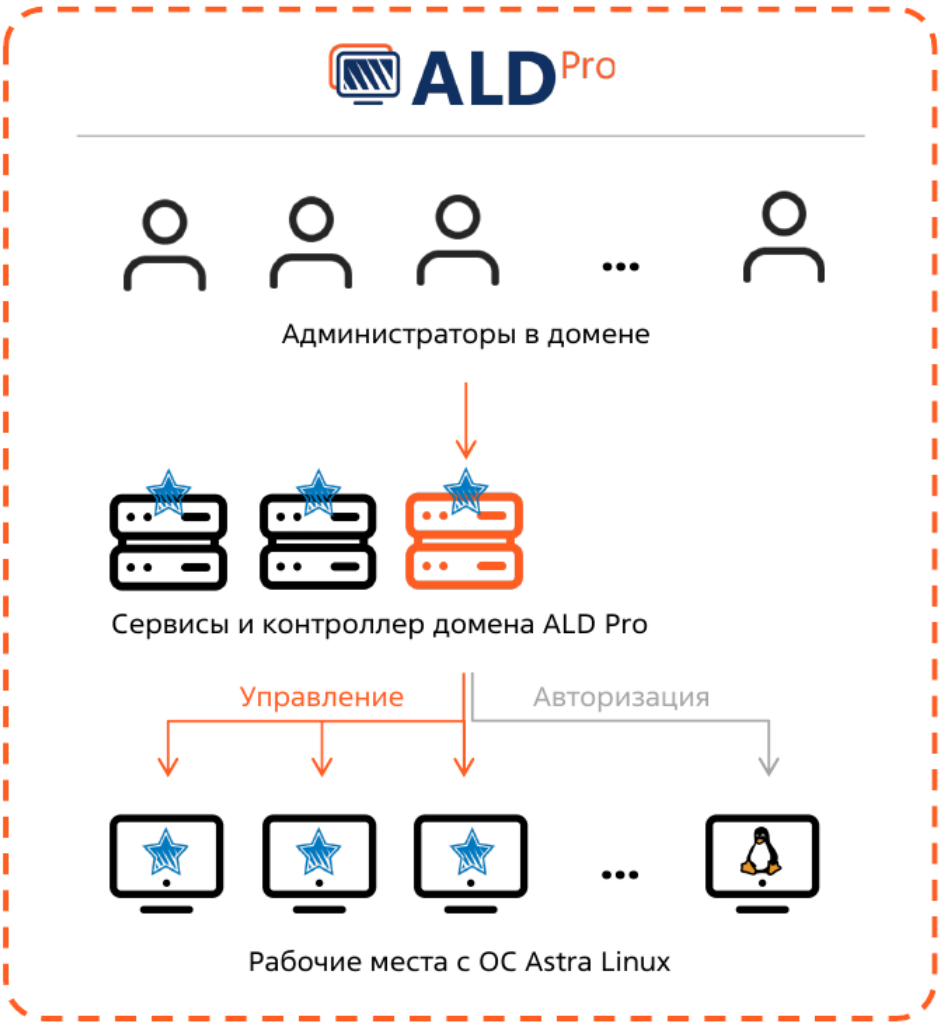


рис. 2.220 – Автономное развертывание

9.3 Доверительные отношения

9.3.1 Механизм работы доверительных отношений

Доверительные отношения дают возможность пользователям одного домена выполнять прозрачную аутентификацию по протоколу Kerberos V5 при обращении к ресурсам другого домена. Например, вы можете разрешить пользователям из домена AD DS входить в операционную систему компьютеров под управлением Astra Linux из домена ALD Pro или, наоборот, дать возможность пользователям ALD Pro подключаться с компьютеров Astra Linux к общим файловым ресурсам, находящимся в домене AD DS, см. рис. 2.221.

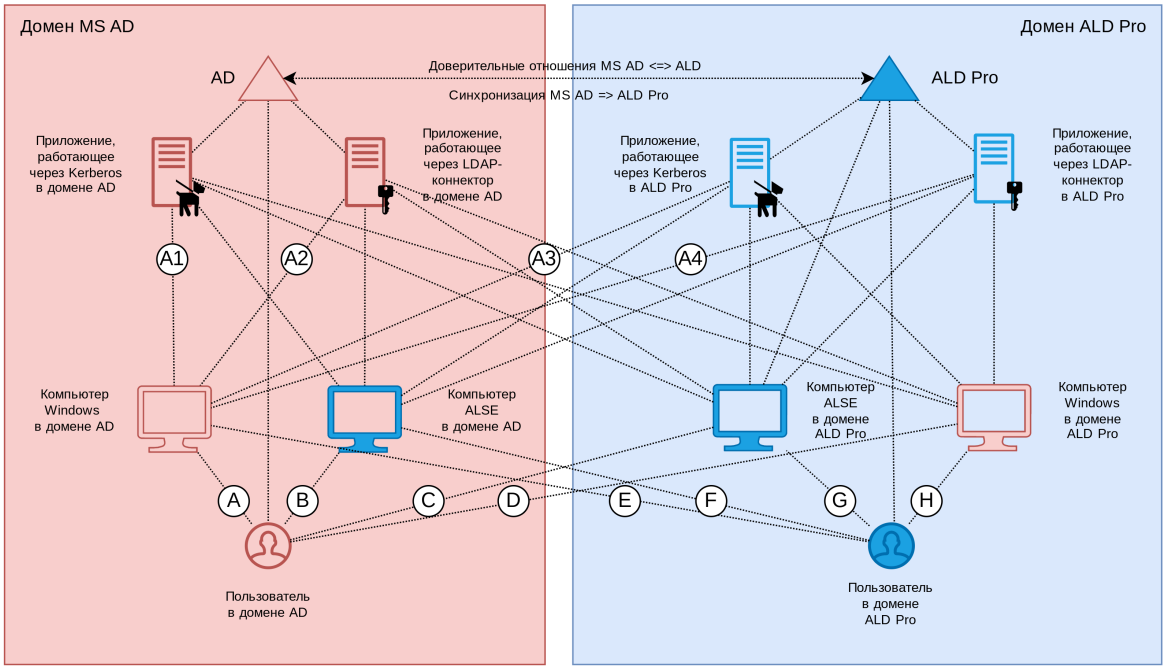


рис. 2.221 – Схема доверительных отношений ALD Pro — AD DS

Система ALD Pro построена на базе службы каталога FreeIPA, в которой поддержка доверительных отношений с AD DS появилась с версии 3.0 путем интеграции с пакетом программ Samba. Доверительные отношения могут быть установлены как между лесами, где FreeIPA выступает в роли леса с одним доменом (Forest Trust), так и между конкретными доменами (External Trust), что в определенных случаях может быть предпочтительнее, например, чтобы сократить количество рекурсивных запросов на получение cross-realm билетов.

Механизм доверительных отношений реализован в протоколе аутентификации Kerberos V5, и понятие доверия является его фундаментом. Например, когда вы выполняете первый вход в операционную систему, у службы SSSD нет прямого доступа к вашим учетным данным, поэтому она не может выполнить проверку аутентичности самостоятельно и вынуждена доверять сервисному билету, выданному контроллером домена. Основанием для доверия является тот факт, что билет зашифрован паролем учетной записи компьютера, который известен только контроллерам домена и самому компьютеру. На стороне контроллера пароль хранится в LDAP-каталоге в атрибуте krbPrincipalKey учетной записи хоста, а на стороне компьютера в специальном файле на диске, см. `klist -k /etc/krb5.keytab`.

Доверие между доменами работает похожим образом: в момент установления отношений в каждом из доменов создается специальная учетная запись доверенного домена с общим ключом или так называемый объект доверенного домена (trusted domain object, TDO). С помощью этого ключа «доверенный домен» (англ. trusted domain) шифрует кросс-доменные билеты пользователей, а «доверяющий домен» (англ. trusting domain) эти билеты расшифровывает и проверяет. Для дове-

ряющего домена такое отношение считается исходящим (outgoing), а для доверенного домена оно считается входящим (incoming), т.е. направление доверия и направление доступа противоположны друг другу, см. рис. 2.222.



рис. 2.222 – Направление доверия и доступа

Если между двумя доменами установлено только одно доверительное отношение, то такое отношение называется односторонним. Если же между доменами установлено два доверительных отношения и домены доверяют друг другу, то такие отношения называются двусторонними.

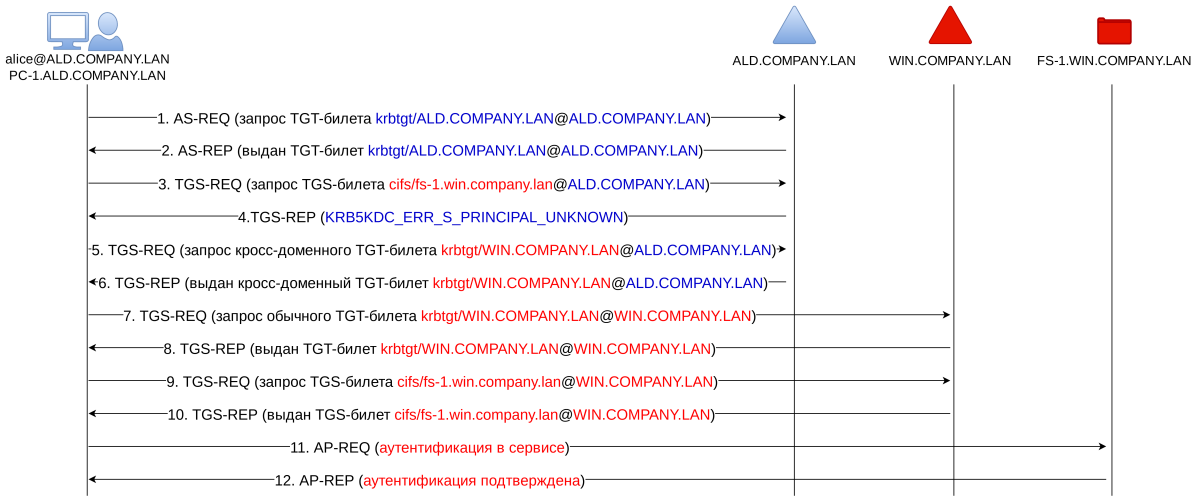


рис. 2.223 – Поток данных при выполнении Kerberos-аутентификации в доверительных отношениях

Рассмотрим, как работает аутентификация с использованием доверительных отношений, когда пользователю доверенного домена ALD.COMPANY.LAN нужно пройти аутентификацию в службе из доверяющего домена WIN.COMPANY.LAN, см. рис. 2.223:

1. Пользователь ALD запрашивает TGT-билет krbtgt/ALD.COMPANY.LAN@ALD.COMPANY.LAN у контроллера из своего родного домена ALD для выполнения последующей аутентификации на контроллерах ALD.

2. Контроллер ALD располагает учетными данными пользователя из своего домена, поэтому может подтвердить аутентичность запроса и выдать ему TGT-билет. Если пользователь сможет

расшифровать ответ, он подтвердит тем самым свою аутентичность.

3. Пользователь ALD, используя TGT-билет krbtgt/ALD.COMPANY.LAN@ALD.COMPANY.LAN, обращается к контроллеру ALD за билетом cifs/fs-1.win.company.lan@ALD.COMPANY.LAN для последующей аутентификации в службе cifs на файловом сервере fs-1 из доверяющего домена WIN.

4. Контроллер ALD располагает мастер-ключом домена ALD, поэтому может успешно проверить аутентичность пользователя, но у него нет учетных данных службы из доверяющего домена WIN, поэтому контроллер возвращает ошибку KRB5KDC_ERR_S_PRINCIPAL_UNKNOWN.

5. Пользователь ALD, используя TGT-билет krbtgt/ALD.COMPANY.LAN@ALD.COMPANY.LAN, обращается к своему контроллеру ALD за кросс-доменным TGT-билетом krbtgt/WIN.COMPANY.LAN@ALD.COMPANY.LAN для последующей аутентификации на контроллерах домена WIN.

6. Контроллер ALD располагает мастер-ключом домена ALD, поэтому может успешно проверить аутентичность пользователя. Контроллер ALD располагает также паролем доверительных отношений с доменом WIN, поэтому успешно выдает пользователю кросс-доменный TGT-билет.

   Кросс-доменный билет является вспомогательным, поэтому не сохраняется в связке ключей, и вы не сможете увидеть его командой `klist`, если только не запросите его целевым образом с помощью команды `kvno krbtgt/WIN.COMPANY.LAN@ALD.COMPANY.LAN`.

   Далее шаги 5 и 6 могут повториться рекурсивно несколько раз по количеству доменов, которые будут стоять на пути между клиентом и сервисом. Если домен A доверяет домену B, а домен B доверяет домену C, то пользователи из домена C могут получить доступ к сервисам из домена A транзитивно через домен B. По умолчанию доверительные отношения между ALD Pro и AD DS устанавливаются транзитивными.

7. Пользователь ALD, используя TGT-билет krbtgt/WIN.COMPANY.LAN@ALD.COMPANY.LAN, обращается к контроллеру WIN за полноценным TGT-билетом krbtgt/WIN.COMPANY.LAN@WIN.COMPANY.LAN.

8. Контроллер WIN располагает паролем доверительных отношений, поэтому успешно проверяет запрос, пересчитывает PAC-сертификат с учетом правил фильтрации и выдает пользователю полноценный TGT-билет из своего домена.

9. Пользователь ALD, используя TGT-билет krbtgt/WIN.COMPANY.LAN@WIN.COMPANY.LAN, обращается к контроллеру WIN за билетом cifs/fs-1.win.company.lan@WIN.COMPANY.LAN для аутентификации в службе cifs на файловом сервере fs-1 из этого домена.

10. Контроллер WIN располагает мастер-ключом домена WIN, поэтому может успешно проверить аутентичность пользователя. Контроллер WIN располагает также учетными данными сервиса из своего домена, поэтому успешно выдает пользователю TGS-билет cifs/fs-1.win.company.lan@WIN.COMPANY.LAN.

11. Пользователь ALD, используя TGS-билет cifs/fs-1.win.company.lan@WIN.COMPANY.LAN, обращается к службе cifs на файловом сервере fs-1 из домена WIN.

12. Файловый сервер располагает паролем службы cifs, поэтому может проверить и подтвердить аутентичность пользователя.

Примечание: В настройках службы SSSD можно задать опцию AD Site, с помощью которой можно настроить географическую балансировку между лесами доменов ALD Pro и AD DS. Опцию следует задать как на клиенте, так и на сервере ALD Pro. В этом случае при выполнении автоматического обнаружения сервисов клиент будет в первую очередь обращаться к SRV-записям указанного сайта.

Протокол Kerberos V5 регламентирует не только порядок аутентификации, но и позволяет передавать внутри билетов авторизационную информацию (поле AuthorizationData). Разработчики Active Directory разработали свой собственный стандарт MS-KILE, в соответствии с которым в этом поле можно передавать сертификат атрибута привилегий (Privilege Attribute Certificate, PAC), который содержит идентификаторы безопасности всех групп, участником которых является пользователь.

Сертификат MS-PAC формируется при выдаче TGT-билета и перекладывается из него во все сервисные билеты, поэтому для того чтобы изменения по участию пользователя в группах вступили в силу, ему нужно обновить свой TGT-билет, т.е. выйти из операционной системы и зайти в нее еще раз.

Для того чтобы пользователь из ALD Pro мог проходить на стороне MS AD не только аутентификацию, но и авторизацию, т. е. получать права доступа в соответствии со своим участием в группах, в службе каталога FreeIPA есть плагин ipa-kdb, который отвечает за формирование сертификата атрибута привилегий по стандарту MS-PAC.

При формировании сертификата используются значения атрибута ipaNTSecurityIdentifier, в котором хранится SID в формате MS Windows. Это значение рассчитывается в момент создания субъекта безопасности в соответствии с настройками текущего диапазона идентификаторов, за что отвечает плагин sidgen.

Посмотреть SID любого объекта можно, например, с помощью команды `sudo wbinfo -n 'ALD\\admin'`, где ALD — это NetBIOS имя домена ald.company.lan, admin — это имя субъекта:

```
admin@dc-1:~$ sudo wbinfo -n 'ALD\\admin'
S-1-5-21-1724891028-2898148248-1736958143-500 SID_USER (1)
```

Как мы видим, последний компонент идентификатора RID для пользователя admin из домена ALD Pro равен 500, что по соглашению MS соответствует идентификатору учетной записи Administrator. Можно добавить, что идентификатор группы admins равен 512, что соответствует группе Domain Admins. Относительные идентификаторы обычных пользователей и групп начинаются с 1000.

При желании вы можете исследовать PAC-сертификат Kerberos-билетов пользователей FreeIPA с помощью команды `net ads kerberos pac` и убедиться, что все необходимые идентификаторы там присутствуют. Но вы не сможете назначить права доступа пользователям FreeIPA, т.к. стандартные оснастки Windows выполняют поиск объектов доверенного леса путем обращения к глобальному каталогу на порт 3268/TCP.

Именно по этой причине довольно часто можно встретить утверждение, что FreeIPA не поддерживает двусторонние доверительные отношения, т.к. с одной стороны они есть, но пользоваться ими невозможно. Для решения этой проблемы в системе ALD Pro был реализован модуль глобального каталога, см. рис. 2.224:

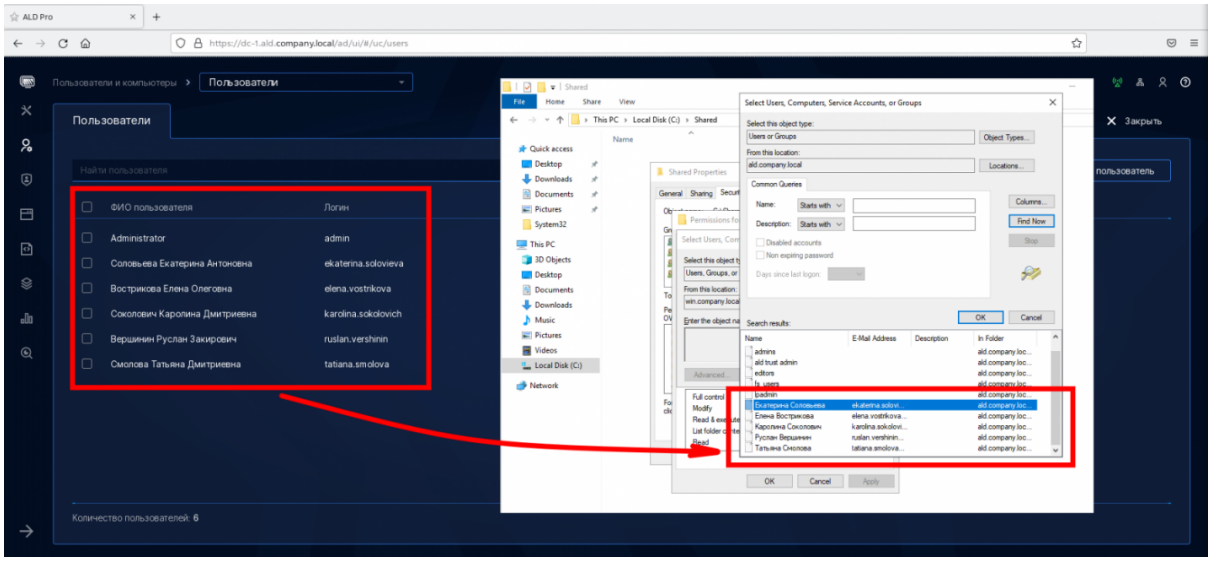


рис. 2.224 – Иллюстрация работы модуля глобального каталога в ALD Pro

Пользователям ванильных версий FreeIPA можно только предложить пару обходных решений. Во-первых, на объекты файловой системы доступ можно назначить напрямую по идентификатору безопасности субъекта с помощью утилиты ICACLS (Integrity Control Access Control List).

```
c:\> ICACLS "C:\\Common" /grant:r "\*S-1-5-21-896088827-1417987318-1335504985-500 " /T
```

Кстати, в стандартном окне свойств файлового менеджера такие пользователи будут отображаться

под своими обычными именами, т. к. процедура разрешения имен работает через RPC-вызов по протоколу SMB (445/TCP) без участия глобального каталога.

Во-вторых, можно воспользоваться более универсальным способом и назначать права через группы безопасности с областью действия «Локальный домен» (Domain Local). Добавить пользователей доверенного домена FreeIPA в локальную доменную группу можно следующим образом на powershell:

```
#SID Из леса ALD_Pro(Пользователь или группа)
$AldSid = 'S-1-5-21-1784717832-1844364183-3442789864-1013'

#Domain Local Group из леса Active Directory
$DomainLocalGroupDN = 'CN=Contoso-Group-DL,CN=Users,DC=contoso,DC=dom'

#Создание нового Directory Entry
$group = New-Object DirectoryServices.DirectoryEntry("LDAP://$($DomainLocalGroupDN)")

#Добавление AldSid в DomainLocal группу
[void]$group.member.Add("<SID=$AldSid>")
$group.CommitChanges()
$group.Close()
```

В обратном направлении работа с авторизационной информацией поставлена немного иначе. В билетах Kerberos, конечно же, есть PAC-сертификат, но Linux-приложения запрашивают эту информацию через вызовы к системным библиотекам, которые обрабатываются службой SSSD. Например, при вызове утилиты `id` никакие Kerberos-билеты пользователя не требуются:

```
admin@dc-1:~$ id Administrator@WIN.COMPANY.LAN
uid=1633600500(administrator@win.company.lan) gid=1633600500(administrator@win.
↪company.lan)
группы=1633600500(administrator@win.company.lan),1633600520(group policy creator
owners@win.company.lan),1633600519(enterprise admins@win.company.lan),
↪1633600512(domain
admins@win.company.lan),1633600518(schema admins@win.company.lan),1633600513(domain
users@win.company.lan),491200006(fs_users)
```

Для получения авторизационной информации на пользователя Administrator из домена WIN.COMPANY.LAN служба SSSD делает не обычный поисковый LDAP-запрос, а отправляет расширенную операцию, которую на стороне сервера обрабатывает плагин exdom. Учитывая, что в LDAP-каталоге нужной информации нет, плагин обращается к службе SSSD на сервере, которая обрабатывает этот запрос по-разному в зависимости от того, какая роль назначена серверу:

- На **контроллерах доверия** служба SSSD напрямую обращается к контроллерам AD DS по протоколу LDAP для преобразования идентификаторов и извлечения информации об участии пользователей доверенного домена в группах.
- На **агентах доверия** служба SSSD проксирует такие запросы на один из контроллеров доверия ALD Pro, опять же, путем выполнения расширенной LDAP-операции.

Примечание: В домене ALD Pro всем контроллерам назначается роль контроллеров доверия, поэтому на них устанавливается пакет программ Samba, и вы с любого из них можете установить доверительные отношения с доменом AD DS.

Если при повышении роли сервера пользователь, выполняющий эту операцию, не входил в состав участников группы «ald trust admin», то продвижение завершится с ошибкой и контроллер будет обладать только ролью агента доверия. Для того чтобы завершить установку и назначить серверу роль контроллера доверия, нужно включить пользователя в указанную группу и запустить повторно скрипт `ipa-aldtrust-install`

Осталось только разобраться с идентификаторами. Учитывая, что в среде Linux используются не SID, а POSIX-идентификаторы, на стороне ALD Pro служба SSSD выполняет сопоставление идентификаторов одним из двух способов:

1. Если в AD DS установлен компонент Identity Management for UNIX (что маловероятно), то у объектов будут заданы атрибуты uidnumber/gidnumber, и вы сможете их использовать. Для этого в момент установления доверительных отношений следует указать параметр *ipa-ad-trust-posix*.

2. Если компонента Identity Management for UNIX нет (что вероятно), то параметр *ipa-ad-trust-posix* задавать не нужно, и по умолчанию идентификаторы будут рассчитываться математически. Для этого на стороне FreeIPA каждому доверенному домену из леса AD DS будет назначен диапазон POSIX-идентификаторов. Размер этого диапазона по умолчанию составляет 200 тысяч идентификаторов, поэтому для крупных доменов AD DS его нужно будет расширить вручную.

Вы можете назначать права доступа непосредственно на эти идентификаторы или задействовать внешние группы (External groups), которые являются аналогом Domain Local групп AD DS, т.к. позволяют добавлять субъекты из доверенного домена. Единственно, учтите, что у внешних групп нет POSIX-идентификаторов, поэтому для предоставления доступа к файловым ресурсам вам нужно будет включить их в состав обычных POSIX-групп.

Например, в следующем примере показано, как доменных администраторов AD DS можно включить в группу доменных администраторов ALD Pro:

```
admin@dc-1:~$ ipa group-add 'ad_admins_external' --external --desc='Группа␣
↪администраторов AD DS'
admin@dc-1:~$ ipa -n group-add-member 'ad_admins_external' --external 'WIN.COMPANY.
↪LAN\\Domain Admins'
admin@dc-1:~$ ipa group-add-member admins --groups='ad_admins_external'
admin@dc-1:~$ id Administrator@WIN.COMPANY.LAN
... 491200000(admins) ...
```

9.3.2 Настройка доверительных отношений

9.3.2.1 Взаимное перенаправление DNS-зон

Для работы доверительных отношений нужно, чтобы из домена ald.company.lan разрешались доменные имена зоны win.company.lan и наоборот. Для этого нам нужно сделать взаимное перенаправление DNS-зон.

Для создания перенаправителя в домене win.company.lan выполните на контроллере домена AD DS следующую команду PowerShell:

```
PS> Add-DnsServerConditionalForwarderZone -Name "ald.company.lan" -ReplicationScope
↪"Forest" -MasterServers [IP адрес ALDPro контроллера]
```

Проверьте возможность разрешения имен из зоны ald.company.lan:

```
admin@dc-1:~$ ping dc-1.ald.company.lan
```

Из интерфейса DNS-перенаправитель можно создать в оснастке DNS Manager, см.:numref:*mod9_figure_image15*:

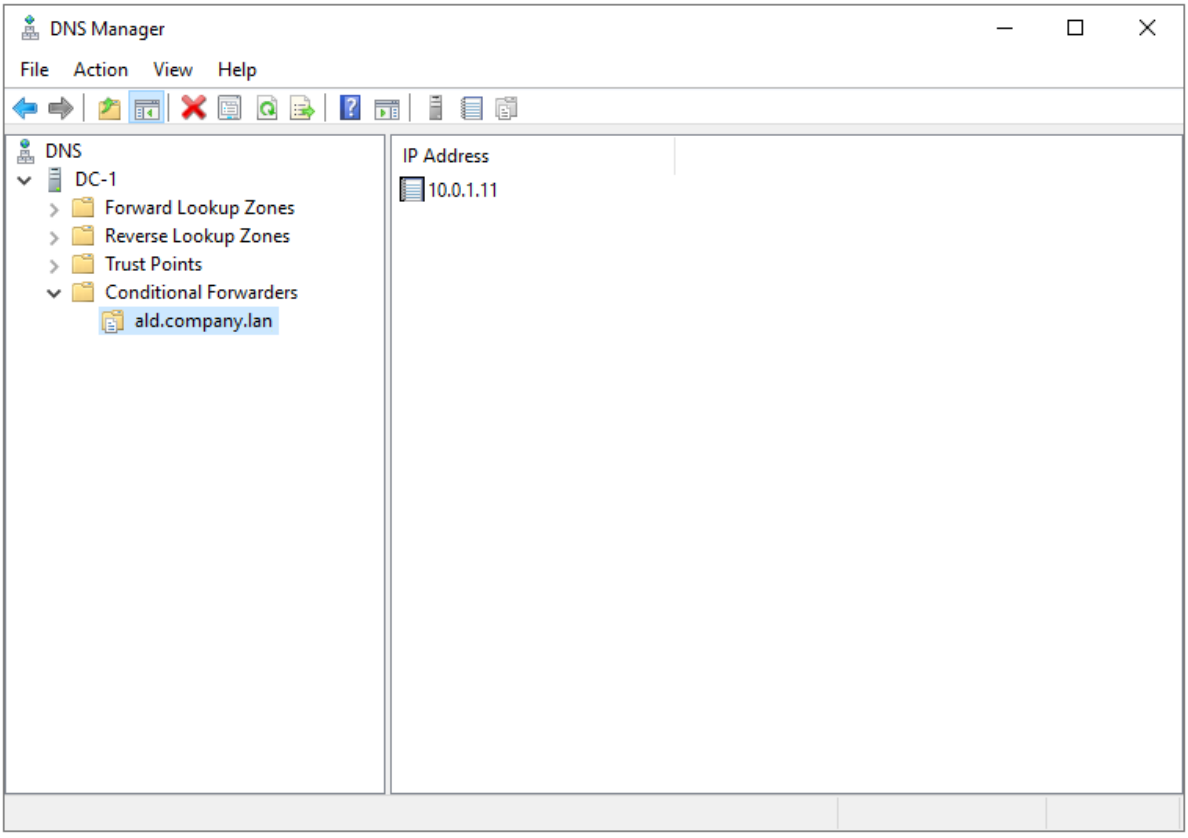


рис. 2.225 – Создание перенаправителя в DNS Manager Windows

Для создания перенаправителя в домене ald.company.lan выполните на контроллере домена

ALD Pro следующую команду bash:

```
admin@dc-1:~$ ipa dnsforwardzone-add win.company.lan --forwarder=[IP адрес␣
↪контроллера MS AD] --forward-policy=only
```

Проверьте возможность разрешения имен из зоны win.company.lan:

```
admin@dc-1:~$ ping dc-1.win.company.lan
```

Из интерфейса DNS-перенаправитель можно создать на странице *Роли и службы сайта* → *Служба разрешения имен Перенаправление запросов*, см. рис. 2.226:

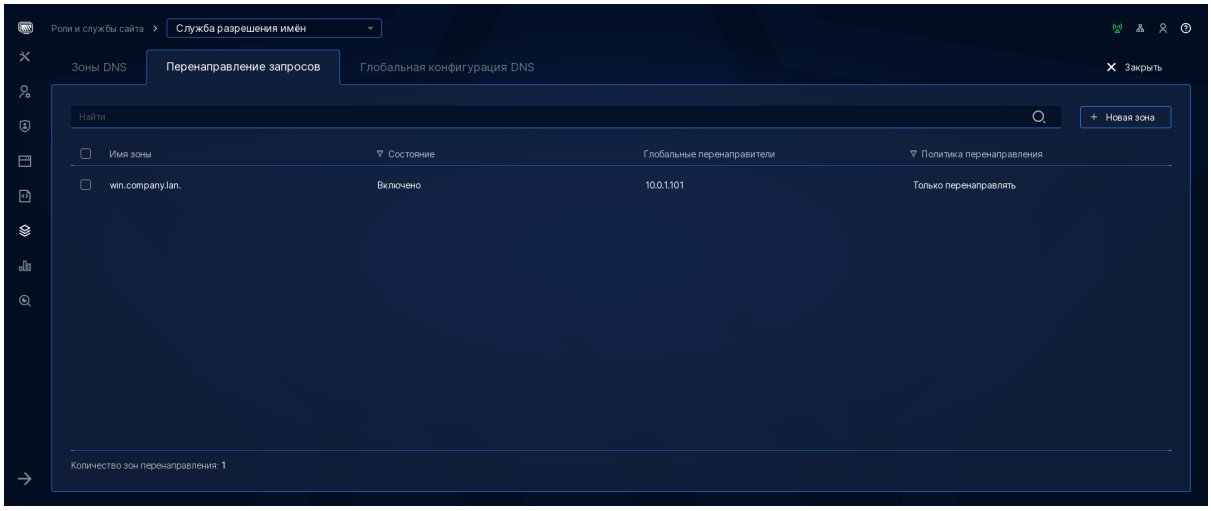


рис. 2.226 – Создание перенаправителя в портале управления ALD Pro

9.3.2.2 Настройка DNS-доверительных сетей (необязательно)

Если вы не хотите разрешать полный доступ к DNS на стороне ALD Pro, вам нужно будет настроить доверенные сети на каждом контроллере ALD Pro (FreeIPA). В файле /etc/bind/ipa-options-ext.conf нужно внести trusted_network в список разрешенных сетей:

```
allow-recursion { trusted_network; };
allow-query-cache { trusted_network; }

listen-on-v6 { any; };

dnssec-validation no;
```

В файле /etc/bind/ipa-ext.conf перечислите доверенные сети:

```
acl "trusted_network" {
  localnets;
  localhost;
  192.168.88.0/24;
  172.19.3.0/24;
  172.19.4.0/24;
};
```

9.3.2.3 Проверка настройки времени

Для работы протокола Kerberos требуется, чтобы время на контроллерах домена ALD Pro и MS AD отличалось не более, чем на 5 минут. Виртуальные машины по умолчанию берут время с хостовой машины, поэтому при запуске стенда на одной физической машине проблем обычно не возникает, а в реальной инфраструктуре синхронизация времени является прямой задачей администратора.

Примечание: Протокол Kerberos использует время в формате UTC, т.е. без временных зон, поэтому, если в Windows временные зоны работают некорректно из-за отсутствия актуального обновления системы, то вы должны использовать неправильною зону, но ни в коем случае не сбивать фактическое значение времени по UTC на эмуляторе PDC.

На стороне ALD Pro используйте команды:

```
admin@dc-1:~$ systemctl status chrony.service
admin@dc-1:~$ chronyc sources -v
admin@dc-1:~$ chronyc tracking
```

На стороне MS AD используйте в терминале PowerShell команды:

```
c:\> w32tm.exe /query /configuration
c:\> w32tm.exe /query /status
(Get-Date).ToUniversalTime()
```

9.3.2.4 Создание траста между лесами (Forest Trust)

Во всех инструкциях создание траста начинают с выполнения команды `ipa-adtrust-install`, которая подготавливает домен FreeIPA к работе с доверительными отношениями, в частности добавляет атрибут для хранения windows security identifier (SID). В случае ALD Pro это не требуется. Попробуйте команду `ipa trustconfig-show`, чтобы убедиться в наличии идентификатора безопасности SID у домена:

```
admin@dc-1:~$ ipa trustconfig-show
 Домен: ald.company.lan
 Идентификатор безопасности: S-1-5-21-1307086432-2724870100-1147875473
 Имя NetBIOS: ALD
 GUID домена: 7a522ccc-a0d7-4eac-a46a-46b12c93fa0e
 Резервная основная группа: Default SMB Group
 Агенты отношений доверия AD IPA: dc-1.ald.company.lan
 Контролёры отношений доверия AD IPA: dc-1.ald.company.lan
```

Перед созданием доверительных отношений нужно получить TGT-билет для учетной записи администратора домена admin:

```
admin@dc-1:~$ kinit admin
```

Также перед созданием траста, нужно разрешить анонимный доступ к Samba-серверу.

```
admin@dc-1:~$ sudo net conf setparm global "restrict anonymous" "0"
admin@dc-1:~$ sudo aldproctl restart -i
```

Включение анонимного доступа требуется не только для создания доверительных отношений, но и для настройки прав доступа к файлам в Samba. Дело в том, что при настройке прав для пользователей из домена ALD Pro, файловый сервер Windows сделает анонимный запрос к Samba серверу и при запрете анонимных запросов будет не возможно предоставить доступ для этих пользователей.

Траст можно добавить на стороне ALD Pro из командной строки.

```
admin@dc-1:~$ ipa -d -v trust-add --type=ad win.company.lan --admin administrator --
↪password --two-way=true
----------------------------------------------------------------------------
Добавлено отношение доверия Active Directory для области (realm) "win.company.lan"
----------------------------------------------------------------------------
Имя области (realm): win.company.lan
Имя домена NetBIOS: WIN
Идентификатор безопасности домена: S-1-5-21-292663598-2229028806-4201728183
Направление отношения доверия: Двустороннее отношение доверия
Тип отношения доверия: Домен Active Directory
Состояние отношения доверия: Установлено и проверено
```

Где:

- `win.company.lan` — задает FQDN имя домена AD DS, с которым требуется установить доверительные отношения.
- Ключ `-d` — активируют режим отладки.
- Ключ `-v` — активирует расширенный режим вывода информации.
- Ключ `--admin` — задает имя привилегированной учетной записи из домена AD.
- Ключ `--password` — указывает, что у пользователя нужно запросить пароль для учетной записи, которая указана ключом `--admin`.

  При установлении отношений рекомендуется указывать пароль, так как в этом случае все настройки будут выполнены автоматически. Но, если указанные учетные данные вам недоступны, вы можете попросить администратора домена AD DS создать траст на своей стороне и сообщить вам пароль доверительных отношений (trust password). В этом случае при установлении отношений нужно будет использовать ключ `--trust-secret`.
- Ключ `--two-way` — указывает на то, что доверительные отношения должны быть двусторонними.

Через графический интерфейс создать доверительные отношения можно на странице *Управление доменом → Интеграция с MS AD → Новое подключение*, см. рис. 2.227:

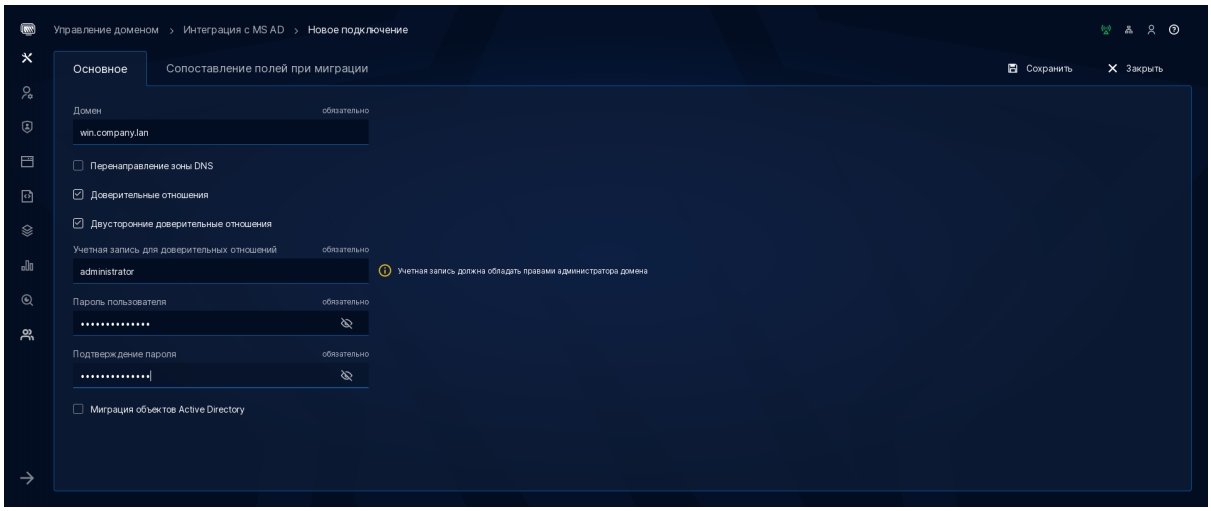


рис. 2.227 – Создание доверительных отношений из графического интерфейса ALD Pro

Если в этот момент отслеживать сетевой траффик, то можно будет увидеть, что аутентификация в домене AD DS выполняется по Kerberos, а установление траста выполняется по транспортному протоколу SMB2 путем вызова удаленных команд RPC (да-да, вы уже знаете, что SMB-протокол — это не только файлы и принтеры). В тоже время билет на доступ к службе krbtgt в Windows домене не сохраняется в кэше sssd, поэтому команда `klist` после установления траста не покажет вам дополнительных билетов.

После создания доверия можно посмотреть информацию о созданном трасте командами `ipa trust-find` и `ipa trust-show`:

```
admin@dc-1:~$ ipa trust-find
---------------------------
найдено 1 отношение доверия
---------------------------
  Имя области (realm): win.company.lan
  Имя домена NetBIOS: WIN
  Идентификатор безопасности домена: S-1-5-21-292663598-2229028806-4201728183
  Тип отношения доверия: Домен Active Directory
---------------------------------
Количество возвращённых записей 1
---------------------------------

admin@dc-1:~$ ipa trust-show win.company.lan
  Имя области (realm): win.company.lan
  Имя домена NetBIOS: WIN
  Идентификатор безопасности домена: S-1-5-21-292663598-2229028806-4201728183
  Направление отношения доверия: Двустороннее отношение доверия
  Тип отношения доверия: Домен Active Directory
```

И не забудьте, что для больших доменов AD DS нужно расширить диапазон идентифика-

торов, который по умолчанию составляет 200 тысяч. Для этого нужно воспользоваться командой `ipa idrange-mod` утилиты `ipa`. Точное имя диапазона можно посмотреть командой `ipa idrange-find`.

```
admin@dc-1:~$ ipa idrange-find
-------------------
найдено 2 диапазона
-------------------
  Имя диапазона: ALD.COMPANY.LAN_id_range
  Первый идентификатор POSIX диапазона: 491200000
  Количество идентификаторов в диапазоне: 200000
  Первый RID соответствующего диапазона RID: 1000
  Первый RID вторичного диапазона RID: 100000000
  Тип диапазона: local domain range

  Имя диапазона: WIN.COMPANY.LAN_id_range
  Первый идентификатор POSIX диапазона: 1633600000
  Количество идентификаторов в диапазоне: 200000
  Первый RID соответствующего диапазона RID: 0
  SID доверенного домена: S-1-5-21-3250248711-3862619044-3424259349
  Тип диапазона: Active Directory domain range

----------------------------------
Количество возвращённых записей 2
----------------------------------
admin@dc-1:~$ ipa idrange-mod --range-size=1000000 WIN.COMPANY.LAN_id_range
ipa: WARNING: Для применения изменений конфигурации необходимо перезапустить службу␣
↪sssd.service
на IPA-сервере WIN.COMPANY.LAN_id_range.
-------------------------------------------------------------
Изменён диапазон идентификаторов "WIN.COMPANY.LAN_id_range"
-------------------------------------------------------------
  Имя диапазона: WIN.COMPANY.LAN_id_range
  Первый идентификатор POSIX диапазона: 1633600000
  Количество идентификаторов в диапазоне: 1000000
  Первый RID соответствующего диапазона RID: 0
  SID доверенного домена: S-1-5-21-3250248711-3862619044-3424259349
  Тип диапазона: Active Directory domain range
```

Если со временем в лесу Active Directory появятся дополнительные поддомены, нужно будет выполнить вручную команду `trust-fetch-domains` утилиты `ipa`, чтобы для этих доменов были созданы диапазоны идентификаторов:

```
admin@dc-1:~$ ipa trust-fetch-domains "WIN.COMPANY.LAN"
```

9.3.2.5 Создание внешнего траста между доменами (External Trust)

Доверительные отношения можно создать не только «лес к лесу», но и «домен к домену». Такие отношения называются внешними (External) и являются нетранзитивными. Создать такие отношения можно из командной строки, используя дополнительный ключ `--external`.

```
admin@dc-1:~$ ipa -d -v trust-add --type=ad win.company.lan --admin administrator --
↪password --two-way=true
--external=true
----------------------------------------------------------------------------
Добавлено отношение доверия Active Directory для области (realm) "win.company.lan"
----------------------------------------------------------------------------
  Имя области (realm): win.company.lan
  Имя домена NetBIOS: WIN
  Идентификатор безопасности домена: S-1-5-21-292663598-2229028806-4201728183
  Направление отношения доверия: Двустороннее отношение доверия
  Тип отношения доверия: Нетранзитивное внешнее отношение доверия с доменом в другом␣
↪лесу Active
Directory
  Состояние отношения доверия: Установлено и проверено
```

9.3.2.6 Удаление траста

Удалить траст можно из командной строки и через веб-интерфейс ALD Pro:

```
admin@dc-1:~$ ipa trust-del win.company.lan
```

Обратите внимание, что на стороне Windows доверительные отношения нужно будет удалить вручную через оснастку Active Directory Domains and Trusts. Также не будут автоматически удалены диапазоны идентификаторов, что позволит вам повторно создать доверительные отношения с этим лесом/доменом в будущем, сохранив настройки сопоставления идентификаторов.

9.3.3 Проверка работоспособности доверительных отношений

В первую очередь ознакомьтесь со следующими ограничениями:

- *Политика AllowNBToInternet (KB5028166)*
- *Работа Samba RPC на серверах ALSE 1.7.6.UU2*
- *Оптимизация для больших инфраструктур*

Рассмотрим несколько диагностических действий, которые вы можете использовать для отладки работы доверительных отношений.

Посмотреть настройки доверия можно командой `ipa trustconfig-show`:

```
admin@dc-1:~$ ipa trustconfig-show
Домен: ald.company.lan
Идентификатор безопасности: S-1-5-21-1307086432-2724870100-1147875473
Имя NetBIOS: ALD
GUID домена: 7a522ccc-a0d7-4eac-a46a-46b12c93fa0e
Резервная основная группа: Default SMB Group
```

(continues on next page)

(продолжение с предыдущей страницы)

```
Агенты отношений доверия AD IPA: dc-1.ald.company.lan dc-XX.ald.company.lan
Контролёры отношений доверия AD IPA: dc-1.ald.company.lan dc-XX.ald.company.lan
```

Посмотреть все доверительные отношения можно командой `ipa trust-find`:

```
admin@dc-1:~$ ipa trust-find
--------------------------
найдено 1 отношение доверия
--------------------------
  Имя области (realm): win.company.lan
  Имя домена NetBIOS: WIN
  Идентификатор безопасности домена: S-1-5-21-3250248711-3862619044-3424259349
  Тип отношения доверия: Домен Active Directory
----------------------------------
Количество возвращённых записей 1
----------------------------------
```

Детальную информацию о конкретном трасте покажет команда `ipa trust-show <REALM>`:

```
admin@dc-1:~$ ipa trust-show win.company.lan
  Имя области (realm): win.company.lan
  Имя домена NetBIOS: WIN
  Идентификатор безопасности домена: S-1-5-21-3250248711-3862619044-3424259349
  Направление отношения доверия: Двустороннее отношение доверия
  Тип отношения доверия: Домен Active Directory
```

Проверить возможность извлечения авторизационной информации из доверенного домена можно с помощью команды `id`. Предварительно целесообразно очистить кэш службы SSSD как на клиенте, так и на сервере:

```
admin@dc-1:~$ rm -rf /var/lib/sss/db/\* /var/lib/sss/mc/\* && systemctl restart sssd
admin@dc-1:~$ id 'win\\administrator'
admin@dc-1:~$ id administrator@win.company.lan
```

Пример результата:

```
admin@dc-1:~$ id administrator@win.company.lan
uid=1131400500(administrator@win.company.lan) gid=1131400500(administrator@win.
↪company.lan)
группы=1131400500(administrator@win.company.lan),1131400520(group policy creator
owners@win.company.lan),1131400513(domain users@win.company.lan),1131400519(enterprise
admins@win.company.lan),1131400518(schema admins@win.company.lan),1131400512(domain␣
↪admi
ns@win.company.lan)
```

Проверять работу аутентификации в доверительных отношениях, выполняя `kinit` на машине ALD Pro пользователем из доверенного домена AD DS, на самом деле совершенно бесполезно. Доверительные отношения в этом случае не используются. Для корректной работы этого механизма вполне достаточно, чтобы было настроено перенаправление DNS-зон. Вы можете включить расширенную трассировку, чтобы увидеть, что запрос уходит сразу на IP-адрес контроллера домена

WIN.

```
admin@dc-1:~$ KRB5_TRACE=/dev/stderr kinit administrator@win.company.lan
[30804] 1698829046.855690: Resolving unique ccache of type KEYRING
[30804] 1698829046.855691: Getting initial credentials for administrator@win.company.
↪lan
[30804] 1698829046.855693: Sending unauthenticated request
[30804] 1698829046.855694: Sending request (195 bytes) to win.company.lan
[30804] 1698829046.855695: Initiating TCP connection to stream 192.168.88.85:88
[30804] 1698829046.855696: Sending TCP request to stream 192.168.88.85:88
[30804] 1698829046.855697: Received answer (198 bytes) from stream 192.168.88.85:88
[30804] 1698829046.855698: Terminating TCP connection to stream 192.168.88.85:88
[30804] 1698829046.855699: Response was from master KDC
[30804] 1698829046.855700: Received error from KDC: -1765328359/Additional pre-
↪authentication
required
[30804] 1698829046.855703: Preauthenticating using KDC method data
[30804] 1698829046.855704: Processing preauth types: PA-PK-AS-REQ (16), PA-PK-AS-REP_
↪OLD (15),
PA-ETYPE-INFO2 (19), PA-ENC-TIMESTAMP (2)
[30804] 1698829046.855705: Selected etype info: etype aes256-cts, salt "WIN-
F8I3I8FPJ70Administrator", params ""
[30804] 1698829046.855706: PKINIT client has no configured identity; giving up
[30804] 1698829046.855707: PKINIT client has no configured identity; giving up
[30804] 1698829046.855708: Preauth module pkinit (16) (real) returned: 22/
↪Недопустимый аргумент
Password for administrator@win.company.lan:
```

В проверке доверительных отношений очень помогает утилита `kvno`. С ее помощью вы можете запросить билет на любой сервис доверенного домена:

```
admin@dc-1:~$ kinit admin
Password for admin@ALD.COMPANY.LAN:

admin@dc-1:~$ kvno cifs/dc-1.win.company.lan@WIN.COMPANY.LAN
cifs/dc-1.win.company.lan@WIN.COMPANY.LAN: kvno = 5

admin@dc-1:~$ klist
Ticket cache: KEYRING:persistent:491200000:krb_ccache_bkldI6V
Default principal: admin@ALD.COMPANY.LAN

Valid starting       Expires              Service principal
02.04.2025 13:58:37  03.04.2025 23:58:37  cifs/dc-1.win.company.lan@WIN.COMPANY.LAN
02.04.2025 13:58:37  03.04.2025 23:58:37  krbtgt/WIN.COMPANY.LAN@WIN.COMPANY.LAN
20.04.2025 13:58:37  03.04.2025 13:58:32  krbtgt/ALD.COMPANY.LAN@ALD.COMPANY.LAN
```

Точно также можно получить даже кросс-доменный билет, который обычно не сохраняется в связке ключей:

```
admin@dc-1:~$ kvno krbtgt/WIN.COMPANY.LAN@ALD.COMPANY.LAN
krbtgt/WIN.COMPANY.LAN@ALD.COMPANY.LAN: kvno = 1
```

9.4 Глобальный каталог

9.4.1 Глобальный каталог AD

В лесу Active Directory может быть много поддоменов, и, чтобы ускорить поиск нужной информации, разработчики Microsoft придумали такую роль, как глобальный каталог (*англ. Global Catalog*).

Контроллер домена, которому назначена роль глобального каталога, с помощью штатной процедуры репликации извлекает из всех поддоменов краткую информацию об объектах (*Partial Attribute Set, PAT*) и предоставляет ее приложениям по запросу. При желании вы можете расширить набор реплицируемых атрибутов. Для этого в редакторе схемы для нужного атрибута достаточно установить флажок «Копировать этот атрибут в глобальный каталог» (*Replicate this attribute to the Global Catalog*).

Глобальный каталог предоставляет данные по протоколу LDAP на портах 3268/tcp (LDAP) и 3269/tcp (LDAPS). В Active Directory первый контроллер автоматически получает роль глобального каталога, но она может быть назначена не всем контроллерам в домене. Автоматическое обнаружение серверов, которым назначена роль глобального каталога, выполняется по SRV-записям вида _ gc._tcp.<DNSDomainName>.

9.4.2 Глобальный каталог ALD Pro

В службе каталога FreeIPA нет глобального каталога, что становится большой проблемой при использовании системы в гибридных сценариях развертывания, поэтому в ALD Pro эта функция была реализована в виде дополнительного модуля.

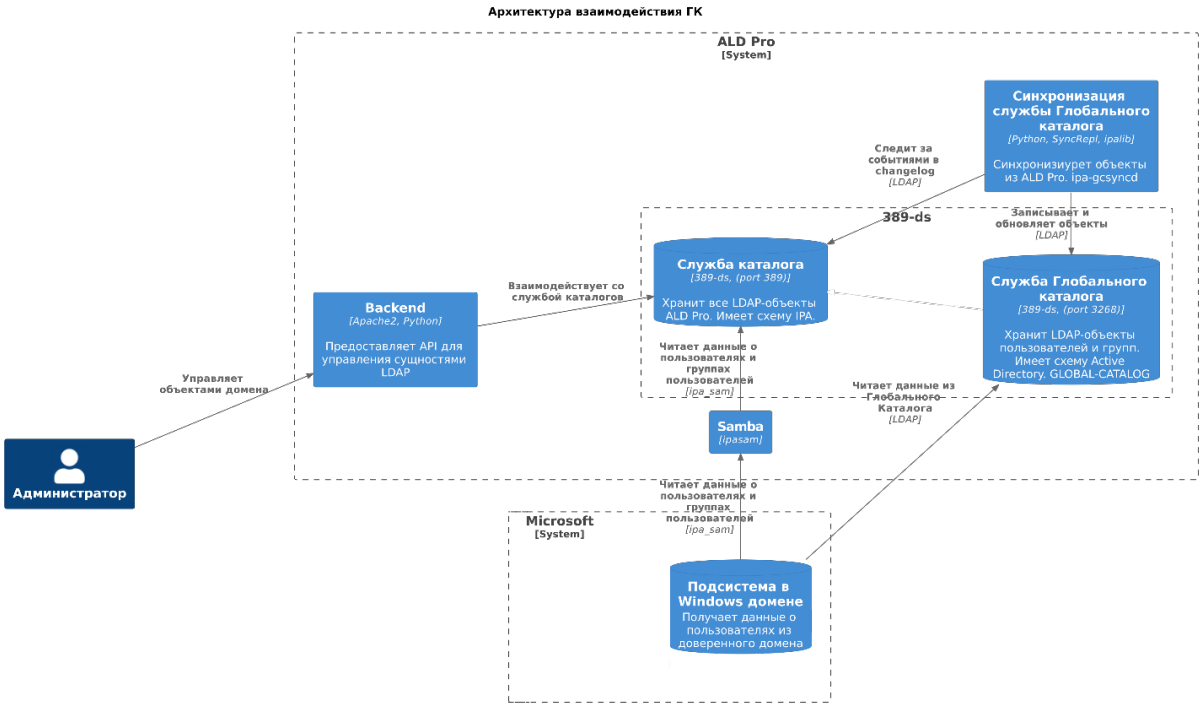


рис. 2.228 – Архитектура глобального каталога ALD Pro

Модуль глобального каталога устанавливается на всех контроллерах в домене, и за его работу отвечают службы `dirsrv@GLOBAL-CATALOG` и `ipa-gcsyncd`, см. рис. 2.228. Первая служба обеспечивает работу дополнительного LDAP-бэкенда на порту 3268/TCP, а вторая синхронизирует информацию глобального каталога с основным каталогом, выполняя необходимую трансформацию данных.

Синхронизация работает на основе механизма репликации syncrepl. Служба `ipa-gcsyncd` извлекает из основного LDAP-каталога информацию о пользователях и группах, модифицирует ее с учетом схемы AD DS, которую ожидают увидеть клиенты Windows, и сохраняет в базу глобального каталога. Модификация данных выполняется по следующим шаблонам.

Шаблон модификации учетных записей пользователей

```
{% macro first_val(attr) -%}
   {{ entry[attr][0] }}
{%- endmacro %}
dn: cn={{ first_val('uid') }},cn=users,{{ suffix }}
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: ad-top
objectClass: ad-organizationalPerson
objectClass: user
objectClass: securityPrincipal
objectClass: posixAccount
objectClass: inetUser
```

(continues on next page)

(продолжение с предыдущей страницы)

```
objectClass: gcobject
cn: {{ first_val('uid') }}
sn: {{ first_val('sn') }}
{%- if entry['givenname'] %}
givenName: {{ first_val('givenname') }}
{%- endif %}
instanceType: 4
{%- if entry['displayname'] %}
displayName: {{ entry.single_value['displayname'] }}
{%- endif %}
name: {{ first_val('cn') }}
objectGUID:: {{ guid }}
userAccountControl: 66048
primaryGroupID: 513
objectSid:: {{ sid }}
sAMAccountName: {{ pkey }}
sAMAccountType: 805306368
{%- if entry['krbcanonicalname'] %}
userPrincipalName: {{ entry.single_value['krbcanonicalname'] }}
{%- else %}
userPrincipalName: {{ first_val('krbprincipalname') }}
{%- endif %}
objectCategory: CN=Person,CN=Schema,CN=Configuration,{{ suffix }}
{%- if entry['mail'] %}
mail: {{ first_val('mail') }}
{%- endif %}
uidnumber: {{ first_val('uidnumber') }}
gidnumber: {{ first_val('gidnumber') }}
{%- for uid in entry['uid'] %}
uid: {{ uid }}
{%- endfor %}
homeDirectory: {{ first_val('homedirectory') }}
{%- for group in entry['memberof'] %}
memberof: {{ group }}
{%- endfor %}
nTSecurityDescriptor:
D:(A;;RPWPCRCCDCLCLORCWOWDSDDTSW;;;DA)(A;;RPWPCRCCDCLCLORCWOWDSDDTSW;;;SY)(A;;
↪RPLCLORC;;;AU)(OA;;WP;736e4812-af31-11d2-b7df-00805f48caeb;bf967ab8-0de6-11d0-a285-
↪00aa003049e2;CO)(A;;SD;;;CO)
gcuuid: {{ entryuuid }}
```

Шаблон модификации групп

```
{% macro first_val(attr) -%}
   {{ entry[attr][0] }}
{%- endmacro %}
dn: cn={{ pkey }},cn=users,{{ suffix }}
objectClass: top
objectClass: ad-top
objectClass: group
objectClass: securityprincipal
objectClass: nsmemberof
objectClass: gcobject
objectClass: posixGroup
cn: {{ pkey }}
```

(continues on next page)

(продолжение с предыдущей страницы)

```
instanceType: 4
name: {{ pkey }}
objectGUID:: {{ guid }}
objectSid:: {{ sid }}
sAMAccountName:  {{ pkey }}
sAMAccountType: 268435456
objectCategory: CN=Group,CN=Schema,CN=Configuration,{{ suffix }}
ntsecuritydescriptor: D:(A;;RPWPCRCCDCLCLORCWOWDSDDTSW;;;DA)(A;;
↪RPWPCRCCDCLCLORCWOWDSDDTSW;;;SY)(A;;RPLCLORC;;;AU)(A;;RPWPCRCCDCLCLORCWOWDSDDTSW;;;
↪AO)(A;;RPLCLORC;;;PS)(OA;;CR;ab721a55-1e2f-11d0-9819-0aa0040529b;;AU)
{%- for member in entry['member'] %}
member: {{ member }}
{%- endfor %}
{%- for group in entry['memberof'] %}
memberof: {{ group }}
{%- endfor %}
gidnumber: {{ gidnumber }}
groupType: {{ groupType }}
gcuuid: {{ entryuuid }}
```

9.4.3 Установка глобального каталога ALD Pro

Вы можете поставить модуль глобального каталога при установке первого контроллера домена. Для этого вам необходимо дополнительно установить пакет aldpro-gc и выполнять продвижение с ключом `--setup_gc`:

```
sudo DEBIAN_FRONTEND=noninteractive apt-get install -q -y aldpro-mp aldpro-gc
sudo aldpro-server-install -d ald.company.lan -n dc-1 -p 'Pa$$w0rd' --ip 10.0.1.11 --
↪no-reboot --setup_gc
```

В дальнейшем вы можете установить модуль, используя ключ `--update`. Для этого потребуется, последовательно выполнить команду, на каждом контроллере, где планируется установка глобального каталога:

```
sudo aldpro-server-install -d ald.company.lan -n [Имя ALDPro контроллера] -p 'Pa$$w0rd
↪' --ip [IP адрес ALDPro контроллера] --update --setup_gc
```

9.4.4 Проверка глобального каталога ALD Pro

В первую очередь ознакомьтесь со следующим ограничением:

- *Проверка целостности глобального каталога*

После установки глобального каталога вы можете убедиться, что модуль был установлен корректно с помощью следующей команды:

```
sudo aldproctl status
```

До установки ГК:

```
admin@dc-1:~$ sudo aldproctl status
.....Сервисы ALD Pro.....
Сервис aldpro-mp-services: ЗАПУЩЕН
.....Сервисы FreeIPA.....
Сервис Directory Service: ЗАПУЩЕН
Сервис krb5kdc: ЗАПУЩЕН
Сервис kadmin: ЗАПУЩЕН
Сервис named: ЗАПУЩЕН
Сервис httpd: ЗАПУЩЕН
Сервис ipa-custodia: ЗАПУЩЕН
Сервис smb: ЗАПУЩЕН
Сервис winbind: ЗАПУЩЕН
Сервис ipa-otpd: ЗАПУЩЕН
Сервис ipa-dnskeysyncd: ЗАПУЩЕН
.....Другие сервисы.....
Сервис celery: ЗАПУЩЕН
Сервис celerybeat: ЗАПУЩЕН
```

После установки ГК:

```
admin@dc-1:~$ sudo aldproctl status
.....Сервисы ALD Pro.....
Сервис aldpro-mp-services: ЗАПУЩЕН
Сервис globalcatalog: ЗАПУЩЕН
Сервис ipa-gcsyncd: ЗАПУЩЕН
.....Сервисы FreeIPA.....
Сервис Directory Service: ЗАПУЩЕН
Сервис krb5kdc: ЗАПУЩЕН
Сервис kadmin: ЗАПУЩЕН
Сервис named: ЗАПУЩЕН
Сервис httpd: ЗАПУЩЕН
Сервис ipa-custodia: ЗАПУЩЕН
Сервис smb: ЗАПУЩЕН
Сервис winbind: ЗАПУЩЕН
Сервис ipa-otpd: ЗАПУЩЕН
Сервис ipa-dnskeysyncd: ЗАПУЩЕН
.....Другие сервисы.....
Сервис celery: ЗАПУЩЕН
Сервис celerybeat: ЗАПУЩЕН
```

Далее вы обнаружите systemd-службу `ipa-gcsyncd.service`, которая отвечает за синхронизацию глобального каталога.

```
admin@dc-1:~$ systemctl status ipa-gcsyncd.service
o ipa-gcsyncd.service - Служба синхронизации Глобального каталога
   Loaded: loaded (/lib/systemd/system/ipa-gcsyncd.service; disabled; vendor preset:␣
↪enabled)
   Active: active (running) since Fri 2024-03-29 17:09:05 MSK; 35min ago
 Main PID: 2253 (gcsyncd.py)
    Tasks: 1 (limit: 4915)
   Memory: 87.5M
      CPU: 2.978s
   CGroup: /system.slice/ipa-gcsyncd.service
           └─2253 /usr/bin/python3 /opt/gc/exec/gcsyncd.py
```

(continues on next page)

(продолжение с предыдущей страницы)

```
map 29 17:09:05 dc-1.aldpro.mycompany.lan systemd[1]: Started Служба синхронизации␣
↪Глобального
каталога.
map 29 17:09:12 dc-1.aldpro.mycompany.lan gcsyncd.py[2253]: ipa: INFO: LDAP bind...
map 29 17:09:12 dc-1.aldpro.mycompany.lan gcsyncd.py[2253]: ipa: INFO: Commencing␣
↪sync process
map 29 17:09:12 dc-1.aldpro.mycompany.lan gcsyncd.py[2253]: ipa: INFO: Initial LDAP␣
↪dump is done,
now synchronizing
```

Вы также обнаружите после установки ГК, что в DNS создались дополнительные SRV-записи для автоматического обнаружения сервиса со стороны Windows-клиентов.

```
_ldap._tcp.Default-First-Site-Name._sites.gc._msdcs
_ldap._tcp.gc._msdcs
_gc._tcp.Default-First-Site-Name._sites
_gc._tcp
```

Найти их можно с помощью утилиты `ipa`, отфильтровав по слову «gc» с помощью утилиты `grep`:

```
admin@dc-1:~$ ipa dnsrecord-find ald.company.lan | grep gc
  Имя записи: \_ldap._tcp.Default-First-Site-Name._sites.gc._msdcs
  Имя записи: \_ldap._tcp.gc._msdcs
  Имя записи: \_gc._tcp.Default-First-Site-Name._sites
  Имя записи: \_gc._tcp
```

Потом можно посмотреть каждую по отдельности:

```
admin@dc-1:~$ sudo ipa dnsrecord-find ald.company.lan --name=_gc._tcp
...
  Имя записи: \_gc._tcp
  SRV record: 0 100 3268 dc-1.ald.company.lan.
----------------------------------
Количество возвращённых записей 1
----------------------------------
```

Просмотреть записи глобального каталога можно следующим запросом:

```
root@dc-1:~# ldapsearch -QLLL -H ldap://dc-1.ald.company.lan:3268 cn
dn: dc=ald,dc=company,dc=lan

dn: cn=users,dc=ald,dc=company,dc=lan
cn: users

dn: cn=computers,dc=ald,dc=company,dc=lan
cn: computers

dn: cn=configuration,dc=ald,dc=company,dc=lan
cn: configuration

dn: cn=schema,cn=configuration,dc=ald,dc=company,dc=lan
cn: schema
```

(continues on next page)

(продолжение с предыдущей страницы)

```
dn: uid=read-only-principal,cn=configuration,dc=ald,dc=company,dc=lan

dn: cn=ForeignSecurityPrincipals,dc=ald,dc=company,dc=lan
cn: ForeignSecurityPrincipals

dn: cn=admin,cn=users,dc=ald,dc=company,dc=lan
cn: admin

dn: cn=admins,cn=users,dc=ald,dc=company,dc=lan
cn: admins

dn: cn=editors,cn=users,dc=ald,dc=company,dc=lan
cn: editors
```

Для проверки целостности записей глобального каталога есть служба aldpro-gc-inspector. Запустить процесс проверки целостности можно командой `sudo systemctl start aldpro-gc-inspector`, а посмотреть статус проверки командой `sudo systemctl status aldpro-gc-inspector`. Суммарное количество ошибок можно найти в файле `/var/run/aldpro/gc`.

9.5 Присоединение Windows-машины к домену ALD Pro

Конечная цель проектов импортозамещения в ИТ — полный отказ от операционной системы Windows. Но, как говорится, гладко было на бумаге, да забыли про овраги. Может так оказаться, что быстро заменить какие-то клиентские корпоративные приложения, написанные под эту операционную систему, не получится. В этом случае вам может пригодиться возможность присоединения Windows-компьютеров к домену ALD Pro.

Присоединение Windows-клиентов к домену FreeIPA не находится в фокусе внимания команды разработчиков этой службы каталога, так как проект ставит своей целью не заменить Active Directory для Windows-компьютеров, а создать аналогичное решение для компьютеров под управлением операционной системы Linux. Однако компания Microsoft еще с версии Windows 2000 поддерживает возможность интеграции своих ОС с областями Kerberos, которые совместимы с RFC 2136, а центр распределения ключей FreeIPA работает как раз на базе эталонной реализации MIT KDC. То есть, ничто не препятствует тому, чтобы вводить Windows-компьютеры напрямую в домен FreeIPA.

Более того, разработчики FreeIPA реализовали возможность интеграции своего домена с Active Directory на уровне доверительных отношений за счет компонента Samba AD, поэтому контроллеры домена, помимо Kerberos, поддерживают такие протоколы, как NetBIOS, SMB, Netlogon (MS-NRPC) и NTLM. В базе данных DNS есть SRV-записи по стандартам Microsoft, а Kerberos-билеты содержат не только аутентификационные данные, но и сертификат PAC, включающий SID'ы пользователя и всех его групп.

Все эти особенности открывают дополнительные возможности по работе Windows-компьютеров в составе домена FreeIPA. Далее мы расскажем, как правильно задействовать эти отличительные черты, чтобы получить доступ к следующим функциям:

- вход в ОС Windows с помощью учетных записей из домена ALD Pro;
- прозрачная аутентификация при обращении пользователя к керберизированным ресурсам, например, файловым серверам;
- получение уведомлений об истечении срока действия пароля и возможность сменить пароль штатными функциями ОС Windows;
- предоставление доменным пользователям прав доступа к локальным ресурсам Windows-компьютера с учетом их участия в доменных группах;
- синхронизация времени рабочей станции с временем на контроллерах домена;
- динамическое изменение DNS-записей в домене при изменении IP-адреса на сетевом интерфейсе Windows-компьютера.

Полная инструкция занимает два десятка страниц и подробно рассказывает о том, как настроить Kerberos-аутентификацию, синхронизацию времени, динамическое обновление DNS-записей, домен для входа по умолчанию и многие другие аспекты. Вы можете ознакомиться с этой информацией в нашей статье на Хабре, поэтому укажем только, что команда ALD Pro разработала для вас специальную графическую утилиту `aldpro-join.exe`, которая позволяет выполнить все настройки в пару кликов, см. рис. 2.229:

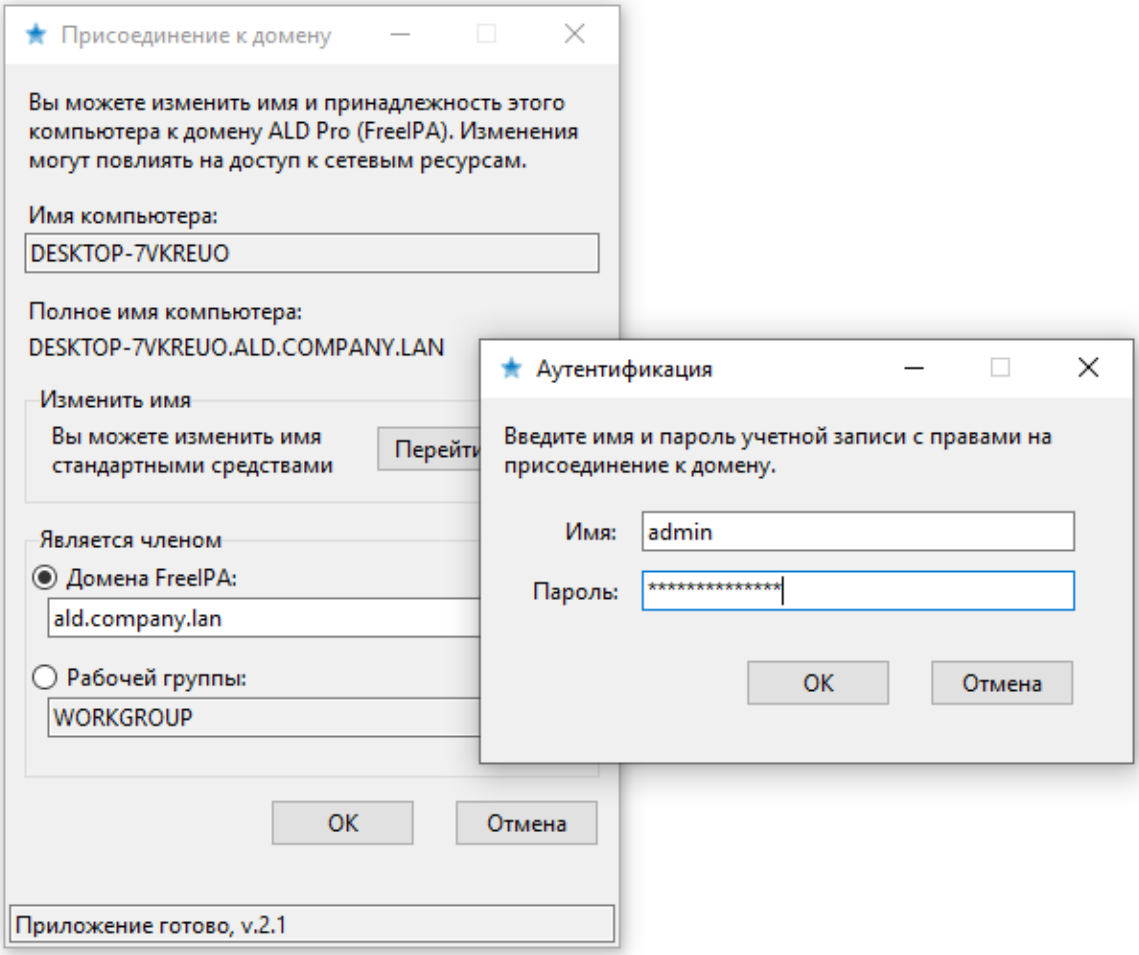


рис. 2.229 – Присоединение компьютера к домену с помощью утилиты aldpro-join.exe

Приложение написано на языке Python и доступно в публичном репозитории продуктовой команды на GitFlic как в виде исходных кодов, так и в виде скомпилированного с помощью Pyinstaller исполняемого exe-файла.

9.6 Практика и тестирование

9.6.1 Практическая работа: Модуль 9. Миграция и гибридное развертывание с MS AD

См. также *Требования, правила и цели выполнения практической работы*

9.6.1.1 Практические задания

Задание 1. Настройка стенда для доверительных отношений

Параметры стенда

В качестве практического занятия мы с вами установим доверительные отношения на уровне леса (Forest Trust) между доменами MS AD и ALD Pro.

Инфраструктура ALD Pro у вас уже развернута. Развернем инфраструктуру Microsoft:

1. Разверните контроллер MS AD на Windows Server 2019:

- Имя контроллера: windc-1.win.company.lan
- Имя домена (Realm): win.company.lan
- Статичный IP адрес: 10.0.1.101
- Маска сети: 255.255.255.0
- Шлюз: 10.0.1.1

2. Создайте winpc-1 на Windows Server 2019 и подключите его в домен win.company.lan:

- Имя хоста: winpc-1.win.company.lan
- Имя домена (Realm): win.company.lan
- Статичный IP адрес: 10.0.1.102
- Маска сети: 255.255.255.0
- Шлюз: 10.0.1.1

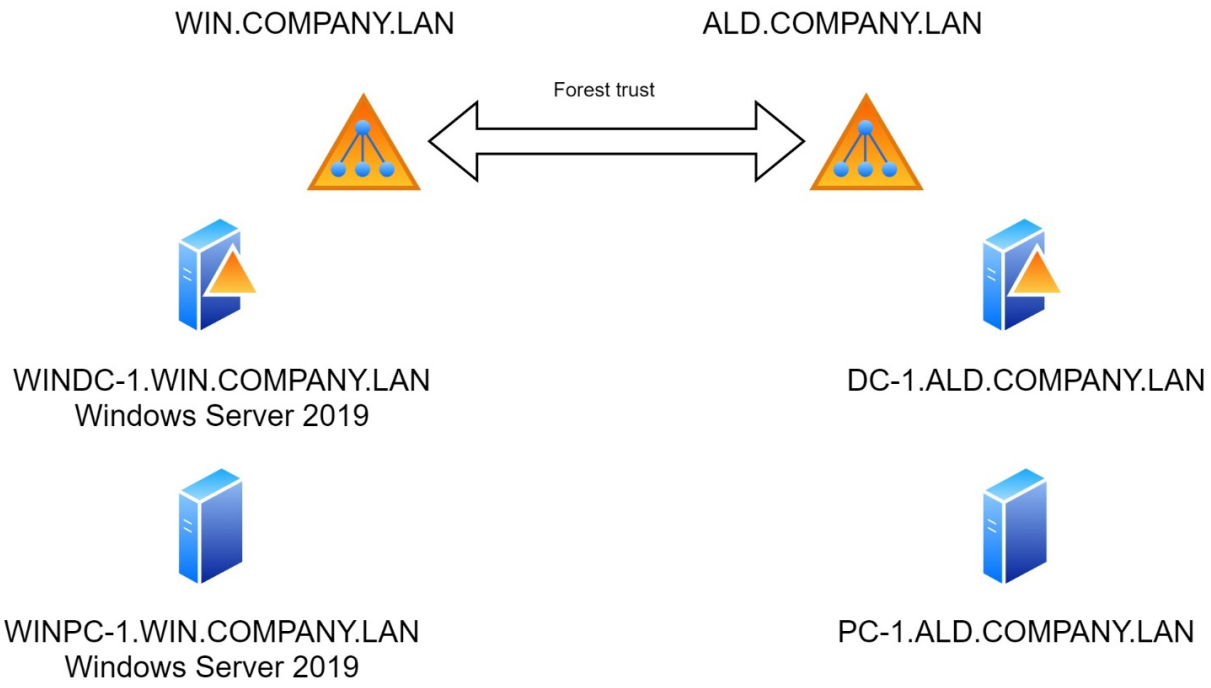


рис. 2.230 – Стенд доверительных отношений

Проверка доступности портов

Следующим пунктом идет проверка портов. Этот пункт в рамках данной практики мы тоже опускаем, так как оба домена мы разворачиваем в одном сегменте сети. Фаерволы операционных систем по умолчанию не блокируют стандартные порты.

Взаимное перенаправление DNS-зон

Для работы доверительных отношений с компьютеров в домене ald.company.lan должны разрешаться имена компьютеров домена win.company.lan и наоборот. Для этого нам нужно сделать взаимное перенаправление DNS-зон.

Для создания перенаправителя (ConditionalForwarder) для зоны ald.company.lan в домене win.company.lan запустите PowerShell на контроллере домена win.company.lan (Active Directory) и выполните следующую команду:

```
Add-DnsServerConditionalForwarderZone -Name "ald.company.lan" -ReplicationScope
↪"Forest" -MasterServers 10.0.1.11
```

где

- ald.company.lan - DNS-имя перенаправляемой зоны;
- 10.0.1.11 - это IP адрес контроллера ALD Pro.

Проверка доступности домена ald.company.lan:

```
ping dc-1.ald.company.lan

Resolve-DnsName _ldap._tcp.ald.company.lan -Type SRV

Resolve-DnsName _kerberos._tcp.ald.company.lan -Type SRV

Resolve-DnsName _ldap._tcp.dc._msdcs.ald.company.lan -Type SRV

Resolve-DnsName _kerberos._tcp.dc._msdcs.ald.company.lan -Type SRV
```

Для создания перенаправителя (ConditionalForwarder) для зоны win.company.lan в домене ald.company.lan запустите окно терминала на контроллере домена ald.company.lan (ALD Pro) и выполните следующие команды:

```
admin@dc-1:~$ kinit admin

admin@dc-1:~$ ipa dnsforwardzone-add win.company.lan --forwarder=10.0.1.101 --forward-
↪policy=only
```

где:

- win.company.lan - DNS-имя перенаправляемой зоны;

- 10.0.1.101 - это IP-адрес контроллера MS AD.

Проверка доступности домена win.company.lan:

```
admin@dc-1:~$ ping windc-1.win.company.lan

admin@dc-1:~$ dig SRV _ldap._tcp.win.company.lan

admin@dc-1:~$ dig SRV _kerberos._tcp.win.company.lan
```

Настройка DNS-доверительных сетей (необязательно)

Следующим пунктом идет проверка портов. Этот пункт в рамках данной практики мы тоже опускаем, так как оба домена мы разворачиваем в одном сегменте сети.

Проверка настройки времени

Основной задачей проверки является убедиться, что время на контроллерах домена ALD Pro и MS AD синхронизированы.

Проверьте сначала на стороне ALD Pro. Начните с проверки статуса службы:

```
root@dc-1:~# systemctl status chrony.service
```

Проверьте источник времени:

```
root@dc-1:~# chronyc sources -v
```

Проверьте расхождение времени:

```
root@dc-1:~# chronyc tracking
```

Конфигурация сервиса времени находится в `/etc/chrony/chrony.conf`.

Далее проверьте настройки времени на стороне MS AD. Так как контроллер MS AD будет поднят с нуля и настройка синхронизации времени будет выставлена по умолчанию, то ниже приведем пример настройки с внешним NTP-источником.

```
w32tm /config /reliable:yes /syncfromflags:manual /manualpeerlist:"ntp.msk-ix.ru" /
↪update

net stop w32time

net start w32time

w32tm /resync

w32tm /monitor /computers:"ntp.msk-ix.ru"

w32tm /stripchart /computer:ntp.msk-ix.ru
```

Задание 2. Установка глобального каталога

Для установки глобального каталога в домене ALD Pro необходимо выполнить следующую команду на первом контроллере домена dc-1:

```
admin@dc-1:~$ sudo aldpro-server-install -d ald.company.lan -n dc-1 -p 'AstraLinux' --
↪ip 10.0.1.11 --update --setup_gc
```

После установки вы можете убедиться, что ГК был установлен, выполнив команду `ipactl status`:

- До установки ГК:

```
root@dc-1:~# ipactl status
Directory Service: RUNNING
krb5kdc Service: RUNNING
kadmin Service: RUNNING
named Service: RUNNING
httpd Service: RUNNING
ipa-custodia Service: RUNNING
smb Service: RUNNING
winbind Service: RUNNING
ipa-otpd Service: RUNNING
ipa-dnskeysyncd Service: RUNNING
ipa: INFO: The ipactl command was successful
```

- После установки ГК:

```
root@dc-1:~# ipactl status
Directory Service: RUNNING
krb5kdc Service: RUNNING
kadmin Service: RUNNING
named Service: RUNNING
httpd Service: RUNNING
ipa-custodia Service: RUNNING
smb Service: RUNNING
winbind Service: RUNNING
ipa-otpd Service: RUNNING
ipa-dnskeysyncd Service: RUNNING
dirsrv@GLOBAL-CATALOG Service: RUNNING
ipa-gcsyncd Service: RUNNING
ipa: INFO: The ipactl command was successful
```

Далее вы обнаружите systemd службу `ipa-gcsyncd.service`, которая отвечает за синхронизацию глобального каталога.

```
root@dc-1:~# systemctl status ipa-gcsyncd
o ipa-gcsyncd.service - Служба синхронизации Глобального каталога
  Loaded: loaded (/lib/systemd/system/ipa-gcsyncd.service; disabled; vendor preset:␣
↪enabled)
  Active: active (running) since Fri 2024-03-29 17:09:05 MSK; 35min ago
Main PID: 2253 (gcsyncd.py)
 Tasks: 1 (limit: 4915)
 Memory: 87.5M
  CPU: 2.978s
```

(continues on next page)

(продолжение с предыдущей страницы)

```
CGroup: /system.slice/ipa-gcsyncd.service
   └─2253 /usr/bin/python3 /opt/gc/exec/gcsyncd.py

мар 29 17:09:05 dc-1.aldpro.mycompany.lan systemd[1]: Started Служба синхронизации
Глобального каталога.
мар 29 17:09:12 dc-1.aldpro.mycompany.lan gcsyncd.py[2253]: ipa: INFO: LDAP bind...
мар 29 17:09:12 dc-1.aldpro.mycompany.lan gcsyncd.py[2253]: ipa: INFO: Commencing sync
process
мар 29 17:09:12 dc-1.aldpro.mycompany.lan gcsyncd.py[2253]: ipa: INFO: Initial LDAP␣
↪dump is
done, now synchronizing
```

После установки глобального каталога вы также обнаружите, что в DNS создались дополнительные SRV-записи для обнаружения этого сервиса клиентами. Найти их можно с помощью утилиты ipa, отфильтровав по слову «gc»:

```
root@dc-1:~# ipa dnsrecord-find ald.company.lan | grep gc

  Имя записи: _ldap._tcp.Default-First-Site-Name._sites.gc._msdcs
  Имя записи: _ldap._tcp.gc._msdcs
  Имя записи: _gc._tcp.Default-First-Site-Name._sites
  Имя записи: _gc._tcp
```

Потом можно посмотреть каждую по отдельности:

```
root@dc-1:~# ipa dnsrecord-find ald.company.lan --name=_gc._tcp
  Имя записи: _gc._tcp
  SRV record: 0 100 3268 dc-1.ald.company.lan.
----------------------------------
Количество возвращённых записей 1
----------------------------------
```

Задание 3. Создание двустороннего траста (Forest Trust)

Во всех инструкциях создание траста начинают с выполнения команды `ipa-adtrust-install`, которая подготавливает домен FreeIPA к работе с доверительными отношениями, в частности добавляет атрибут для хранения windows security identifier (SID). В случае ALD Pro это не требуется. Попробуйте команду `ipa trustconfig-show`, чтобы убедиться в наличии атрибута SID (идентификатор безопасности):

```
root@dc-1:~# ipa trustconfig-show
Домен: ald.company.lan
Идентификатор безопасности: S-1-5-21-1307086432-2724870100-1147875473
Имя NetBIOS: ALD
GUID домена: 7a522ccc-a0d7-4eac-a46a-46b12c93fa0e
Резервная основная группа: Default SMB Group
Агенты отношений доверия AD IPA: dc-1.ald.company.lan
Контролёры отношений доверия AD IPA: dc-1.ald.company.lan
```

Перед созданием доверительных отношений нужно запросить Kerberos-билет для учетной записи admin:

```
root@dc-1:~# kinit admin
```

Траст можно добавить на стороне ALD Pro из командной строки. При появлении проблем рекомендуется использовать ключи -d и -v для получения дополнительной информации об ошибках:

```
root@dc-1:~# ipa -d -v trust-add --type=ad win.company.lan --admin administrator --
↪password --two-way=true
--------------------------------------------------------------------------------
Добавлено отношение доверия Active Directory для области (realm) "win.company.lan"
--------------------------------------------------------------------------------
 Имя области (realm): win.company.lan
 Имя домена NetBIOS: WIN
 Идентификатор безопасности домена: S-1-5-21-292663598-2229028806-4201728183
 Направление отношения доверия: Двустороннее отношение доверия
 Тип отношения доверия: Домен Active Directory
 Состояние отношения доверия: Установлено и проверено
```

Если в этот момент отслеживать сетевой трафик, то можно будет увидеть, что аутентификация в Windows-домене выполняется по Kerberos, а установление траста выполняется по транспортному протоколу SMB2 путем вызова удаленных команд RPC (все верно, SMB-протокол — это не только файлы и принтеры). В тоже время билет на доступ к службе krbtgt в Windows-домене не сохраняется в кэше sssd службы, поэтому команда `klist` после установления траста не покажет вам дополнительных билетов.

После создания доверия можно посмотреть информацию о созданном тресте командами `ipa trust-find` и `ipa trust-show`:

```
root@dc-1:~# ipa trust-find
---------------------------
найдено 1 отношение доверия
---------------------------
 Имя области (realm): win.mycompany.lan
 Имя домена NetBIOS: WIN
 Идентификатор безопасности домена: S-1-5-21-3328403581-165198287-4002972307
 Тип отношения доверия: Домен Active Directory
--------------------------------
Количество возвращённых записей 1
--------------------------------
```

```
root@dc-1:~# ipa trust-show win.company.lan
Имя области (realm): win.company.lan
Имя домена NetBIOS: WIN
Идентификатор безопасности домена: S-1-5-21-3328403581-165198287-4002972307
Направление отношения доверия: Двустороннее отношение доверия
Тип отношения доверия: Домен Active Directory
```

Можно посмотреть и диапазон POSIX-идентификаторов доверенного домена:

```
root@dc-1:~# ipa idrange-find
------------------
найдено 2 диапазона
------------------
Имя диапазона: ALD.COMPANY.LAN_id_range
Первый идентификатор POSIX диапазона: 536400000
Количество идентификаторов в диапазоне: 200000
Первый RID соответствующего диапазона RID: 1000
Первый RID вторичного диапазона RID: 100000000
Тип диапазона: local domain range

Имя диапазона: WIN.COMPANY.LAN_id_range
Первый идентификатор POSIX диапазона: 1801200000
Количество идентификаторов в диапазоне: 200000
Первый RID соответствующего диапазона RID: 0
SID доверенного домена: S-1-5-21-3328403581-165198287-4002972307
Тип диапазона: Active Directory domain range
--------------------------------
Количество возвращённых записей 2
--------------------------------
```

Задание 4. Проверка работоспособности доверительных отношений

Два основных типа запросов, которые может выполнять клиент - это поиск идентификационных данных (id) и аутентификация (login). Для пользователей ALD Pro (IPA) поиск идентификационных данных представляет собой обычный поиск по протоколу LDAP на ALD Pro (IPA)-сервере, а аутентификация выполняется с помощью внутреннего инструмента, подобного kinit, также подключающегося к ALD Pro (IPA)-серверу.

Доверенные пользователи AD (Trusted AD users) работают по-другому: поиск идентификационных данных (id) с клиентских компьютеров подключается к ALD Pro (IPA)-серверу с помощью расширенной операции LDAP. Плагин extdom сервера ALD Pro (IPA) запрашивает у экземпляра SSSD, запущенного на сервере, информацию о пользователе AD, экземпляр SSSD выполняет поиск на сервере Active Directory с использованием протокола LDAP и передает данные обратно клиенту ALD Pro (IPA). Для запросов аутентификации на основе пароля (login) клиенты ALD Pro (IPA) подключаются непосредственно к серверам AD с помощью Kerberos.

Имейте в виду, что, поскольку во время проверки подлинности должен быть задан правильный набор групп, запрос на проверку подлинности также должен выполнить поиск идентификатора, обычно операцию initgroups, перед проверкой подлинности.

1. Первым делом выполним аутентификацию и проверим, все ли контроллеры домена ALD Pro перечислены в контроллерах отношений доверия.

Выполнить аутентификацию:

```
root@dc-1:~# kinit admin
```

Показать, какие контроллеры домена могут обрабатывать Trust с Active Directory:

```
root@dc-1:~# ipa trustconfig-show
Домен: ald.company.lan
Идентификатор безопасности: S-1-5-21-3134460814-120682416-4104473534
Имя NetBIOS: ALD
GUID домена: 7a522ccc-a0d7-4eac-a46a-46b12c93fa0e
Резервная основная группа: Default SMB Group
Агенты отношений доверия AD IPA: dc-1.ald.company.lan dc-XX.ald.company.lan
Контролёры отношений доверия AD IPA: dc-1.ald.company.lan dc-XX.ald.company.lan
```

Показать созданный траст:

```
root@dc-1:~# ipa trust-find win.company.lan
```

Показать направление отношения доверия:

```
root@dc-1:~# ipa trust-show win.company.lan
```

Если контроллер ALD Pro не указан в списке, то требуется проверить наличие ошибок в логах на проблемном контроллере:

```
root@dc-1:~# grep -insRH --color=auto "ipa-ad" /var/log
/var/log/ipaserver-install.log:162:2023-05-11T14:56:28Z DEBUG The ipa-adtrust-install
command failed, exception: ScriptError: Unrecognized error during check of admin␣
↪rights: 'member_user'
```

В данном случае нужно вручную повторно запустить команду `ipa-adtrust-install` на проблемном контроллере.

2. Далее проверим топологию репликации суффикса.

```
root@dc-1:~# ipa topologysuffix-verify --help
Usage: ipa [global-options] topologysuffix-verify NAME [options]
Options:

 -h, --help show this help message and exit
```

Проверка топологии репликации для суффикса, которые выполняются:

a. проверка того, не отключена ли топология (иначе говоря, имеются ли пути репликации между всеми серверами).

b. проверка того, не превышено ли рекомендованное количество соглашений о репликации между серверами.

```
root@dc-1:~# ipa topologysuffix-verify domain
============================================================================
```

(continues on next page)

(продолжение с предыдущей страницы)

```
Топология репликации суффикса "domain" соответствует требованиям.
======================================================================
```

3. Проверим работоспособность траста с помощью команды id. Требуется выполнить на каждом ALD Pro контроллере домена. Очистка всего доступного кэша и перезапуск службы sssd:

```
root@dc-1:~# rm -f /var/lib/sss/db/* /var/lib/sss/mc/* && systemctl restart sssd
```

Получение информации о пользователе в домене win.company.lan:

```
root@dc-1:~# id 'win\administrator'
```

или так:

```
root@dc-1:~# id administrator@win.company.lan
uid=1131400500(administrator@win.company.lan)
gid=1131400500(administrator@win.company.lan)
группы=1131400500(administrator@win.company.lan),1131400520(group policy creator
owners@win.company.lan),1131400513(domain
users@win.company.lan),1131400519(enterprise
admins@win.company.lan),1131400518(schema
admins@win.company.lan),1131400512(domain admi
ns@win.company.lan)
```

4. Проверка работоспособности траста с помощью Kerberos

Выполним запрос Kerberos-билета из домена MS AD, включив логирование протокола Kerberos:

https://web.mit.edu/kerberos/krb5-1.12/doc/user/user_commands/kinit.html

```
root@dc-1:~# KRB5_TRACE=/dev/stderr kinit administrator@win.company.lan
[30804] 1698829046.855690: Resolving unique ccache of type KEYRING
[30804] 1698829046.855691: Getting initial credentials for administrator@win.company.
↪lan
[30804] 1698829046.855693: Sending unauthenticated request
[30804] 1698829046.855694: Sending request (195 bytes) to win.company.lan
[30804] 1698829046.855695: Initiating TCP connection to stream 10.0.1.101:88
[30804] 1698829046.855696: Sending TCP request to stream 10.0.1.101:88
[30804] 1698829046.855697: Received answer (198 bytes) from stream 10.0.1.101:88
[30804] 1698829046.855698: Terminating TCP connection to stream 10.0.1.101:88
[30804] 1698829046.855699: Response was from master KDC
[30804] 1698829046.855700: Received error from KDC: -1765328359/Additional pre-
↪authentication required
[30804] 1698829046.855703: Preauthenticating using KDC method data
[30804] 1698829046.855704: Processing preauth types: PA-PK-AS-REQ (16), PA-PK-AS-REP_
↪OLD (15), PA-ETYPE-INFO2 (19), PA-ENC-TIMESTAMP (2)
[30804] 1698829046.855705: Selected etype info: etype aes256-cts, salt "WIN-
↪F8I3I8FPJ70Administrator", params ""
[30804] 1698829046.855706: PKINIT client has no configured identity; giving up
[30804] 1698829046.855707: PKINIT client has no configured identity; giving up
[30804] 1698829046.855708: Preauth module pkinit (16) (real) returned: 22/
↪Недопустимый аргумент
Password for administrator@win.company.lan:
```

Далее выполним команду KLIST, чтобы посмотреть полученные билеты Kerberos в кэше:

```
root@dc-1:~# klist
Ticket cache: KEYRING:persistent:0:krb_ccache_6c40ahD
Default principal: Administrator@WIN.COMPANY.LAN

Valid starting       Expires              Service principal
24.10.2023 13:48:50  24.10.2023 23:48:50  krbtgt/WIN.COMPANY.LAN@WIN.COMPANY.LAN
    renew until 25.10.2023 13:48:44
```

Теперь попробуем получить служебный билет Kerberos для доступа к файловому серверу kvno:

```
root@dc-1:~# kvno -S CIFS windc-1.win.company.lan
CIFS/windc-1.win.company.lan@WIN.COMPANY.LAN: kvno = 3
```

```
root@dc-1:~# klist
Ticket cache: KEYRING:persistent:0:krb_ccache_6c40ahD
Default principal: Administrator@WIN.COMPANY.LAN

Valid starting       Expires              Service principal
24.10.2023 14:17:17  24.10.2023 23:48:50  CIFS/windc-1.win.company.lan@WIN.COMPANY.LAN
24.10.2023 13:48:50  24.10.2023 23:48:50  krbtgt/WIN.COMPANY.LAN@WIN.COMPANY.LAN
   renew until 25.10.2023 13:48:44
```

5. Дополнительные проверки:

Выполняем аутентификацию пользователя из домена AD DS, получаем TGT-билет от контроллера AD DS

```
root@dc-1:~# kinit administrator@win.company.lan
```

Получаем для пользователя Cross-realm TGT-билет у контроллера AD DS для сквозной аутентификации на контроллерах ALD Pro:

```
root@dc-1:~# kvno krbtgt/ALD.COMPANY.LAN@WIN.COMPANY.LAN
```

Используя Cross-realm TGT-билет, получаем у контроллера ALD Pro сервисный билет для аутентификации на хосте из домена ALD Pro с fqdn именем dc-1.ald.company.lan:

```
root@dc-1:~# kvno host/dc-1.ald.company.lan@ALD.COMPANY.LAN
```

Получаем для пользователя Cross-realm TGT-билет у контроллера ALD Pro для сквозной аутентификации на контроллерах Active Directory:

```
root@dc-1:~# kvno krbtgt/WIN.COMPANY.LAN@ALD.COMPANY.LAN
```

Используя Cross-realm TGT-билет, получаем у контроллера Active Directory сервисный билет для аутентификации на хосте из домена Active Directory:

```
root@dc-1:~# kvno host/windc-1.win.company.lan@WIN.COMPANY.LAN
```

Конечный результат

```
root@dc-1:~# klist
Ticket cache: KEYRING:persistent:0:krb_ccache_B4jH9CO
Default principal: Administrator@WIN.COMPANY.LAN

Valid starting       Expires              Service principal
24.10.2023 14:41:04  25.10.2023 00:22:09  host/windc-1.win.company.lan@WIN.COMPANY.LAN
24.10.2023 14:39:55  25.10.2023 00:22:09  krbtgt/WIN.COMPANY.LAN@ALD.COMPANY.LAN
24.10.2023 14:28:13  25.10.2023 00:22:09  host/dc-1.ald.company.lan@ALD.COMPANY.LAN
24.10.2023 14:25:15  25.10.2023 00:22:09  krbtgt/ALD.COMPANY.LAN@WIN.COMPANY.LAN
24.10.2023 14:22:09  25.10.2023 00:22:09  krbtgt/WIN.COMPANY.LAN@WIN.COMPANY.LAN
   renew until 25.10.2023 14:22:04
```

Задание 5. Дополнительное задание

1. Попробуйте выполнить вход на **pc-1.ald.company.lan**, под учетной записью Administrator@WIN.COMPANY.LAN.

Примечание: Если вы изменили HBAC-правило «allow_all», которое изначально разрешает всем подключаться к хостам ALD Pro (например, запретили вход всем, кроме группы admins), то необходимо добавить пользователя Administrator@WIN.COMPANY.LAN в external-группу, которую затем включить разрешающее правило. Если этого не сделать, то попытка зайти на хост ALD Pro вы будете получать ошибку «Доступ запрещен».

---

2. Попробуйте выполнить вход на **winpc-1.win.company.lan** под учетной записью admin@ALD.COMPANY.LAN.

3. Создайте общую папку на **winpc-1.win.company.lan**, дайте полный доступ пользователю admin из домена **ALD.COMPANY.LAN**. Зайдите на **pc-1.ald.company.lan** под учетной записью admin@ALD.COMPANY.LAN, подключите общую папку, созданную выше. Убедитесь в наличии доступа, создав файл и внеся в него изменения.

4. На **windc-1.win.company.lan** запустите оснастку **Active Directory Domains and Trusts** и убедитесь, что доверительное отношение с ald.company.lan создалось.

9.6.2 Тест: Модуль 9. Миграция и гибридное развертывание с MS AD

Пожалуйста, ответьте на несколько вопросов, чтобы закрепить полученные знания. Правильные ответы для самопроверки вы сможете найти в конце списка. Обратите внимание, что у некоторых вопросов может быть несколько верных вариантов ответов.

9.6.2.1 Вопросы

Вопрос 1. Каким ключом шифруются cross-realm билеты, с помощью которых пользователи доверенного домена проходят аутентификацию на контроллерах доверяющего домена?

a. Hash пароля пользователя

b. Ключ специальной учетной записи (TDO)

c. Hash пароля компьютера

d. Ключ не используется

Вопрос 2. К какому контроллеру домена обратится клиент, если пользователь доверенного домена ALD Pro хочет получить доступ к сервису из доверяющего домена MS AD? Предполагается, что TGT-билет в связке ключей уже есть.

a. ALD Pro

b. MS AD

c. Directory Service

d. Samba File Server

Вопрос 3. Какой командой вы сможете установить доверительное отношение на стороне ALD Pro?

a. Команда `ipa trust-create`

b. Команда `ipa trust-add`

c. Команда `ipa trust-new`

Вопрос 4. Какой командой можно посмотреть все существующие доверительные отношения на стороне ALD Pro?

a. Команда `ipa trust-find`

b. Команда `ipa trust-show`

c. Команда `ipa trustconfig-show`

Вопрос 5. После разворачивания среды ALD Pro вы вспомнили, что забыли установить глобальный каталог (ГК). Что можно предпринять?

a. Глобальный каталог устанавливается только при первом продвижении контроллера

b. Команда `aldpro-server-install` с ключами `--update --setup_gc`

c. Команда `aldpro-server-install` с ключом `--setup_gc`

Вопрос 6. В процессе настройки доступа к общей папке в MS AD вы обнаружили, что не видите пользователей из домена ALD Pro. Что можно проверить (выберите все подходящие варианты)?

a. Проверить службы глобального каталога на контроллере ALD Pro с помощью команды `ipactl status`

b. Проверить доступность портов 3268/3269 на ALD Pro

c. Проверить наличие SRV-записей для глобального каталога в DNS

d. Проверить доступность веб-сервера Apache

Ответы на вопросы

1. b
2. a
3. b
4. a
5. b
6. a, b, c

9.7 Заключение

В этом модуле мы рассмотрели один из наиболее действенных механизмов гибридного развертывания, без которого не обойдется ни один проект импортозамещения! Далее займемся резервным копированием.

9.8 Дополнительные источники информации

- Статья «Двусторонние трасты с AD DS и реализация глобального каталога в ALD Pro»
- Статья «Хотите присоединить Windows к домену ALD Pro (FreeIPA)? Спросите меня как»

9.9 Обратная связь

Если остались вопросы, то их всегда можно задать в специальной теме.

Модуль 10. Резервное копирование ALD Pro

10.1 Введение

Потеря данных — это страшный сон для владельца любого бизнеса, так как после такого события девять из десяти компаний обычно не выживают. Речь тут идет, конечно же, о потере клиентской базы или интеллектуальной собственности, но даже в менее критичных случаях ущерб может оказаться весьма значителен. Поэтому иногда шутят, что все системные администраторы делятся на три группы: тех, кто не делает бэкапы, кто делает и кто проверяет возможность восстановления системы из резервной копии.

Из этого модуля вы узнаете о том, как выполнять резервное копирование подсистем ALD Pro и в чем заключаются особенности их восстановления, связанные с хранением данных в распределенной базе каталога.

10.2 Резервное копирование дисков

Любое серверное приложение представляет собой набор файлов, в которых хранятся пользовательские данные, настройки приложения и программный код. Резервное копирование может быть нацелено на любую из этих категорий файлов или копировать диски целиком, как блочные устройства.

Важно понимать, что при выполнении резервного копирования нужно не только сделать копию данных, но и обеспечить ее согласованность:

- Если мы полностью выключим сервер и сделаем снимок виртуальной машины (или создадим копии всех его дисков), то у нас не возникнет проблем с запуском системы, см. рис. 2.231.
- Если же мы попытаемся копировать файлы или блочное устройство во время работы системы, то данные могут оказаться несогласованными и система не заработает или какие-то изменения окажутся утеряны.

  Современные гипервизоры позволяют создавать горячие снимки виртуальных машин, которые сохраняют не только состояние файлов, но и состояние оперативной памяти на момент создания снимка, что позволяет продолжить работу системы с того же места, но использовать такие копии все же не рекомендуется, т.к. они не всегда могут гарантировать полную

консистентность данных.

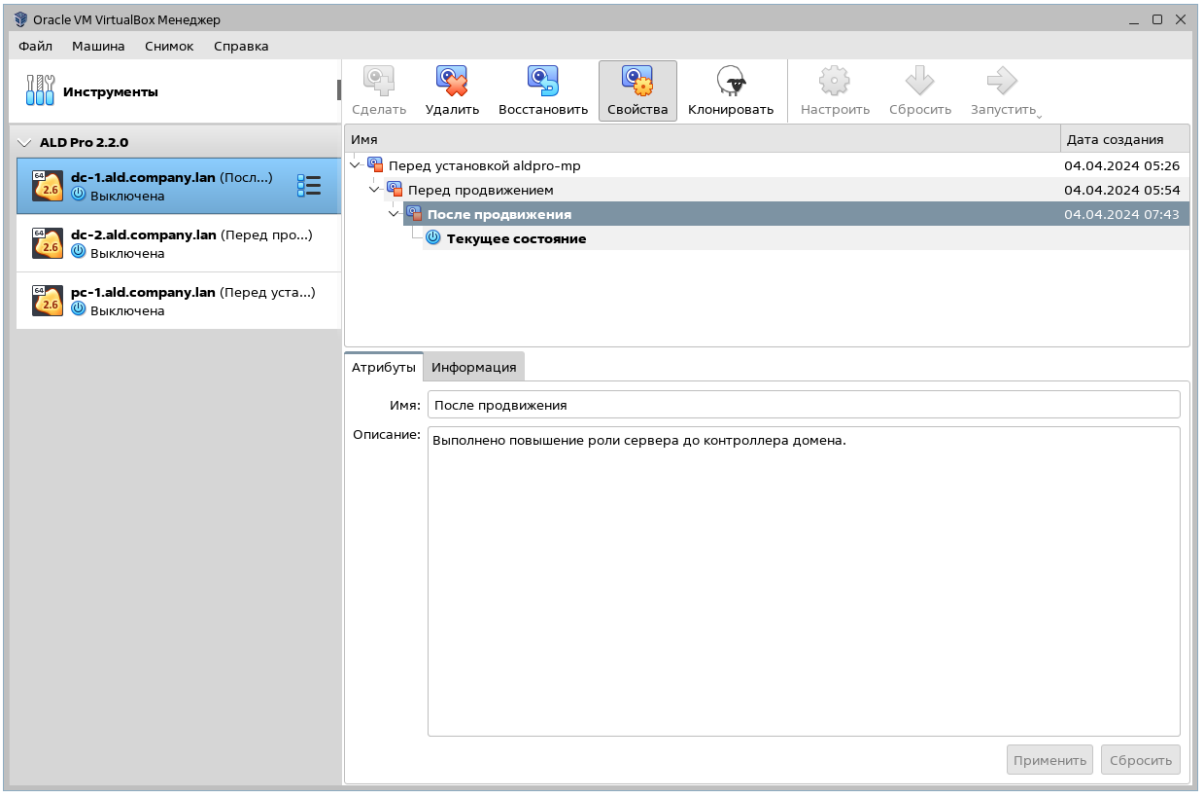


рис. 2.231 – Создание полного снимка виртуальной машины

Примечание: В случае выключения контроллера домена его клиенты автоматически переключатся на другие серверы, но если так делать постоянно, то лучше выделить так называемую скрытую реплику (англ. hidden replica), которая будет исключена из SRV-записей, чтобы клиенты не выбирали ее в качестве активного контроллера.

Для того чтобы обычный контроллер сделать скрытой репликой, нужно изменить статус командой `ipa server-state $(hostname) --state=hidden`. Чтобы вернуть контроллер в строй, нужно изменить статус на enabled. Текущее состояние домена покажет команда `ipa config-show`.

Однако, процедура резервного копирования и восстановления данных усложняется, когда мы имеем дело с кластером серверов, или когда информационная система хранит часть своих данных в таком кластере. Например, если в домене несколько контроллеров, то при восстановлении отдельно взятого сервера у нас нарушится репликация, и чем позднее мы выполним повторную инициализацию такого контроллера, тем больше изменений будет утеряно.

Потребность повторной инициализации вызвана тем, что при восстановлении контроллера домена из резервной копии соседние серверы, с которыми он связан соглашениями о репликации, не смогут

отдать ему те изменения, которые были внесены в домен непосредственно с его участием. Очень подробно механизм репликации был рассмотрен в *Модуль 8. Управление топологией в домене*.

На лабораторном стенде перед созданием снимков виртуальных машин вы без проблем можете выключить все контроллеры домена, чтобы получить консистентное состояние. Но в продуктивной среде такой сценарий невозможен, и отдельные снимки можно использовать в двух случаях:

- С помощью такой копии можно быстро восстановить операционную систему вместе с настройками приложений. Останется только выполнить инициализацию контроллера, и сервер будет полностью готов к работе.
- С помощью такой копии вы можете восстановить данные каталога, актуальные на момент создания копии. В этом случае после восстановления контроллера вам нужно будет выполнить инициализацию в противоположном направлении, затирая данные действующих контроллеров информацией из восстановленной копии.

Чтобы переинициализировать контроллер, выполните следующие действия, см. рис. 2.232:

- Откройте страницу *Управление доменом → Сайты и службы → Соглашения о репликации* .
- Выберите соглашение о репликации, в котором участвует сервер, восстановленный из резервной копии.
- Нажмите на кнопку *Переинициализировать контроллер домена* рядом с тем сервером, данные которого вы хотите затереть.

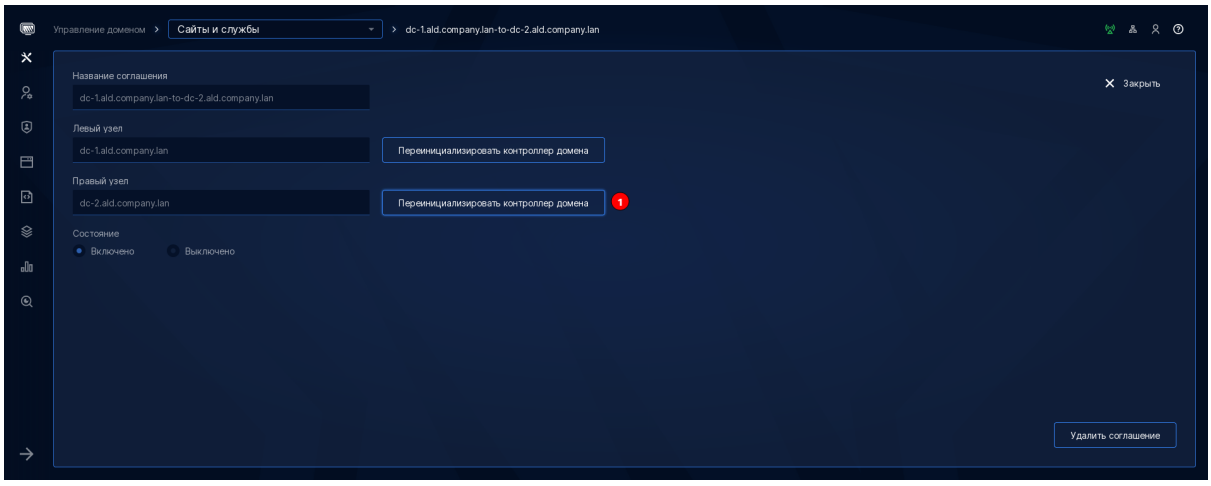


рис. 2.232 – Переинициализация контроллера домена

Из командной строки это можно сделать с помощью утилиты `ipa-replica-manage`:

```
admin@dc-2:~$ sudo ipa-replica-manage re-initialize --from dc-1.ald.company.lan
```

При восстановлении обычных подсистем (не контроллеров) из снимков нужно учитывать, что с версии 2.4.0 все подсистемы хранят свое целевое состояние в каталоге LDAP.

10.3 Резервное копирование данных

Учитывая, что полное выключение сервера приводит к временному прерыванию в обслуживании клиентов, в многие системы реализуют функции резервного копирования, которые гарантируют целостность данных при снятии копий «на горячую». Например, вы можете сделать дамп базы PostgreSQL или каталога 389DS. Мы рассмотрим несколько скриптов резервного копирования контроллера домена и отдельных подсистем.

10.3.1 Контроллеры домена

Чтобы выполнить резервное копирование контроллера домена перейдем на нем в режим суперпользоватля. Затем создадим в каталоге */root* скрипт *backup_ dc.sh*:

```
#!/usr/bin/env bash
now="$(date +'%Y-%m-%dT%H-%M-%S')"
app="$(echo $0 | cut -d '/' -f 2 )"
# функция логирования
function log() { logger -t "$app" "$1"; echo "$1"; }
# получим версию aldpro для указания бекапа
aldpro_version="$(cat /etc/apt/sources.list.d/aldpro.list | cut -d '/' -f 7 | cut -d
↪' ' -f 1)"
log "Начато создание резервной копии ${app} ${aldpro_version} ${HOSTNAME} ${now}"
# папка резервной копии ALD Pro
aldpro_backups="/var/backups/aldpro/"

# Название директории для резервной копии формируется
# из текущей даты, времени в формате ISO и версии ALD Pro и находится в папке $aldpro_
↪backups
BACKUP_PATH="${aldpro_backups}${now}-${aldpro_version}"
log "Создание временной директории для резервных копий $BACKUP_PATH"
mkdir -p $BACKUP_PATH

log "Создание резервной копии FreeIPA $(ipa --version)"
ipa-backup
log "Архивирование РК FreeIPA"
tar -zcvf $BACKUP_PATH/ipa.tar.gz /var/lib/ipa/backup
log "Очистка промежуточного бэкапа FreeIPA"
rm -rf /var/lib/ipa/backup/*

log "Остановка затрагиваемых бэкапом сервисов"
systemctl stop apache2 celery celerybeat rabbitmq-server postgresql aldpro-salt-minion

log "Архивирование БД PostgreSQL"
tar -zcvf $BACKUP_PATH/postgresql.tar.gz /var/lib/postgresql/
log "Архивирование RabbitMQ"
tar -zcvf $BACKUP_PATH/rabbitmq.tar.gz /var/lib/rabbitmq/mnesia/
log "Архивирование директории ipa-client"
```

(continues on next page)

(продолжение с предыдущей страницы)

```
tar -zcvf $BACKUP_PATH/ipa-client.tar.gz /var/lib/ipa-client/
log "Архивирование логов"
tar -zcvf $BACKUP_PATH/log.tar.gz --exclude=faillog --exclude=lastlog /var/log/
log "Архивирование директории etc"
tar -zcvf $BACKUP_PATH/etc.tar.gz /etc/
log "Архивирование директории rbta"
tar -zcvf $BACKUP_PATH/rbta.tar.gz /opt/rbta/

log "Запуск затрагиваемых бэкапом сервисов"
systemctl start apache2 celery celerybeat rabbitmq-server postgresql aldpro-salt-
↪minion

log "Копируем директорию резервной копии на другой носитель"
# Вместо /mnt/ вы можете настроить свою директорию (облако или сетевой диск)
cp -r $BACKUP_PATH /mnt/

log "Удаляем резервные копии старше 30 дней для ротации"
find "${aldpro_backups}" -type d -mtime +30 -exec rm -rf {} \;
log "Создание копии завершено!"
```

Работу скрипта можно отслеживать в системном журнале. Откроем новое окно терминала и выполним команду:

```
sudo journalctl -f | grep backup_dc
```

Поставим на скрипт бит исполнения запустим и запустим его:

```
chmod u+x ./backup_dc.sh
./backup_dc.sh
```

Просмотрим созданные файлы в директории резервных копий:

```
ls -lh  "/var/backups/aldpro/$(ls /var/backups/aldpro | sort | tail -1)"
```

Для восстановления Контроллера домена перейдем в режим суперпользователя и в каталоге */root* создадим файл *restore_ dc.sh*. Запишем в него следующие строки:

```
#!/bin/bash
help="Для запуска восстановления нужно указать папку аргументом\n ./restore_dc.sh␣
↪DATE-DIR"
aldpro_backups="/var/backups/aldpro/"
app="$(echo $0 | cut -d '/' -f 2 )"
# функция логирования
function log() { logger -t "$app" "$1"; echo "$1"; }
function err() { logger --priority user.error -t "$app" "$1"; echo -e "\033[0;31m$1\
↪033[0m";  }
# проверка наличия аргумента папки РК для восстановления
if [[ -z "$1" ]]; then
   err "Ошибка! Нет аргумента в виде папки!"
   echo -e "\n$help"
   echo -e "\nСписок доступных резервных копий:"
   ls "$aldpro_backups"
   exit 1
```

(continues on next page)

(продолжение с предыдущей страницы)

```
fi
RESTORE_PATH="${aldpro_backups}$1"
# Перейти в директорию с резервными копиями
cd $RESTORE_PATH
# Получаем ошибку если нет такой директории резервной копии
check_backup=$?
# Проверим дату бекапа в аргументе, иначе выведем список дат для удобства
if [[ "$check_backup" -gt 0 ]]; then
   err "Ошибка! Нет $RESTORE_PATH резервной копии!"
   echo "Выберите дату резервной копии:"
   ls "$aldpro_backups"
   exit "$check_backup"
fi
log "Начато восстановление $RESTORE_PATH резервной копии!"
set -e # для возможности отмены Ctrl+Z и если ошибка, то выход будет сразу
log "Разархивирование РК IPA ipa.tar.gz"
tar -C "/" -xvf ipa.tar.gz
# вычислим дату резервной копии из архива
ipa_date_time="$(tar tf ipa.tar.gz | grep 'ipa-full' | cut -d '/' -f 5 | grep -v '^$'␣
↪| sort | tail -1 | uniq)"
log "Восстановление IPA из РК /var/lib/ipa/backup/$ipa_date_time"
ipa-restore "/var/lib/ipa/backup/$ipa_date_time"
log "Остановка затрагиваемых восстановлением сервисов"
systemctl stop apache2 celery celerybeat rabbitmq-server postgresql aldpro-salt-minion
log "Восстановление БД Postgresql"
tar -C "/" -xvf postgresql.tar.gz
log "Восстановление RabbitMQ"
tar -C "/" -xvf rabbitmq.tar.gz
log "Восстановление логов"
tar -C "/" -xvf log.tar.gz
log "Восстановление директории etc"
tar -C "/" -xvf etc.tar.gz
log "Восстановление директории rbta"
tar -C "/" -xvf rbta.tar.gz
log "Восстановление ipa-client"
tar -C "/" -xvf ipa-client.tar.gz

log "Восстановление завершено. Перезагрузка!"
sleep 5 && reboot
```

Сделаем скрипт исполняемым и запустим его. В качестве аргумента скрипту следует передать дату резервной копии:

```
chmod u+x ./restore_dc.sh
./restore_dc.sh "ДАТА-ИЗ-СПИСКА"
```

Если такой даты нет, то будет выведен список всех имеющихся резервных копий. Эта информация содержится в переменной `$aldpro_backup`.

Во время восстановления утилита *ipa* потребует ввести пароль от учетной записи *Directory Manager*. Кроме того, надо будет подтвердить восстановление написав *yes*:

```
root@dc-1:~# ./restore_dc.sh 2025-04-03T13-07-30-2.4.0
```

(continues on next page)

(продолжение с предыдущей страницы)

```
var/lib/ipa/backup/
var/lib/ipa/backup/ipa-full-2025-04-03T13-07-30-2.4.0/
var/lib/ipa/backup/ipa-full-2025-04-03T13-07-30-2.4.0/header
var/lib/ipa/backup/ipa-full-2025-04-03T13-07-30-2.4.0/ipa-full.tar
/var/lib/ipa/backup/ipa-full-2025-04-03T13-07-30-2.4.0
Directory Manager (existing master) password:

Preparing restore from /var/lib/ipa/backup/ipa-full-2025-04-03T13-07-30-2.4.0 on dc-1.
↪ald.company.lan
Performing FULL restore from FULL backup
Temporary setting umask to 022
Restoring data will overwrite existing live data. Continue to restore? [no]:
```

Если в домене несколько контроллеров, то после восстановления сервера вы будете наблюдать ошибки репликации в журнале `/var/log/dirsrv/slapd-ALD-COMPANY-LAN/errors`, и вам нужно будет как можно скорее выполнить повторную инициализацию этого сервера (или, наоборот, использовать данные этого сервера для переинициализации остальных контроллеров в домене). Например, если мы восстановили *dc-2* и хотим забрать актуальные данные с *dc-1*, то нужно выполнить команду:

```
admin@dc-2:~$ sudo ipa-replica-manage re-initialize --from dc-1.ald.company.lan
Update in progress, 7 seconds elapsed
Update succeeded
```

Напомним, что для проверки состояния репликации между двумя конкретными серверами RedHat предлагает использовать утилиту `ds-replcheck`:

```
ds-replcheck online -D "cn=Directory Manager" -m ldap://dc-1.ald.company.lan:389 -r␣
↪ldap://dc-2.ald.company.lan:389 -b dc=ald,dc=company,dc=lan
```

Наконец, состояние репликации можно проверить с помощью утилиты *cipa* которую мы рассматривали в *Модуль 8. Управление топологией в домене*:

```
read -s -p "Enter password: " pswd && cipa -d ald.company.lan -W "$pswd"
```

Результат выполнения этой команды будет примерно таким:

```
+--------------------+---------+---------+-------+
| FreeIPA servers:   | dc-1    | dc-2    | STATE |
+--------------------+---------+---------+-------+
| Active Users       | 2       | 2       | OK    |
| Stage Users        | 0       | 0       | OK    |
| Preserved Users    | 1       | 1       | OK    |
| Hosts              | 4       | 4       | OK    |
| Services           | 10      | 10      | OK    |
| User Groups        | 6       | 6       | OK    |
| Host Groups        | 1       | 1       | OK    |
| Netgroups          | 0       | 0       | OK    |
| HBAC Rules         | 2       | 2       | OK    |
| SUDO Rules         | 0       | 0       | OK    |
```

(continues on next page)

(продолжение с предыдущей страницы)

```
| DNS Zones          | 2       | 2       | OK    |
| Certificates       | 0       | 0       | OK    |
| LDAP Conflicts     | 0       | 0       | OK    |
| Ghost Replicas     | 0       | 0       | OK    |
| Anonymous BIND     | ROOTDSE | ROOTDSE | OK    |
| Microsoft ADTrust  | True    | True    | OK    |
| Replication Status | dc-2 19 | dc-1 0  | FAIL  |
+--------------------+---------+---------+-------+
```

Здесь:

- **Replication Status** - указывает состояние ошибки (FAIL) или успеха (OK)
- **dc-2 19** - число различий с репликой (19)
- **dc-1 0** - число различий с репликой (0)

10.3.2 Подсистема общего доступа к файлам

Создадим файл *backup_files.sh* в домашней папке *root* с таким содержимым:

```
#!/usr/bin/env bash
now="$(date +'%Y-%m-%dT%H-%M-%S')"
app="$(echo $0 | cut -d '/' -f 2 )"
# функция логирования
function log() { logger -t "$app" "$1"; echo "$1"; }
# папка резервной копии ALD Pro
aldpro_backups="/var/backups/aldpro/"
aldpro_version="$(cat /etc/apt/sources.list.d/aldpro.list | cut -d '/' -f 7 | cut -d
↪' ' -f 1)"
log "Начато создание РК ${app} ${aldpro_version} ${HOSTNAME} ${now}"

# Название директории для резервной копии формируется
# из текущей даты, версии и находится в папке
BACKUP_PATH="${aldpro_backups}${now}-${aldpro_version}"
log "Создание временной директории для резервных копий $BACKUP_PATH"
mkdir -p $BACKUP_PATH
log "Остановка затрагиваемых бэкапом сервисов"
systemctl stop winbind smbd nmbd aldpro-salt-minion
#log "Архивирование логов"
#tar -zcvf "$BACKUP_PATH/log.tar.gz"--exclude=faillog --exclude=lastlog /var/log/
log "Архивирование директории etc"
tar -zcvf "$BACKUP_PATH/etc.tar.gz" /etc/
log "Архивирование директории ipa-client"
tar -zcvf "$BACKUP_PATH/ipa-client.tar.gz" /var/lib/ipa-client/
log "Архивирование директории /opt/samba_shares/"
tar -zcvf "$BACKUP_PATH/samba.tar.gz" /opt/samba_shares/
log "Запуск затрагиваемых бэкапом сервисов"
systemctl start winbind smbd nmbd aldpro-salt-minion

# Вместо /mnt/ вы можете настроить свою директорию (облако или сетевой диск)
log "Копируем директорию резервной копии на другой носитель"
```

(continues on next page)

(продолжение с предыдущей страницы)

```
cp -r $BACKUP_PATH /mnt/

log "Удаляем резервные копии старше 30 дней для ротации"
find "${aldpro_backups}" -type d -mtime +30 -exec rm -rf {} \;
log "Создание копии ${app} завершено!"
```

Для восстановления создадим файл restore_files.sh в домашней папке root с таким содержимым:

```
#!/bin/bash
set -e # для возможности отмены Ctrl+Z и если ошибка то выход будет сразу
app="$(echo $0 | cut -d '/' -f 2 )"
help="Для запуска восстановления нужно указать папку аргументом\n ./${app} DATE-DIR"
aldpro_backups="/var/backups/aldpro/"
# функция логирования
function log() { logger -t "$app" "$1"; echo "$1";  }
function err() { logger --priority user.error -t "$app" "$1"; echo -e "\033[0;31m$1\
↪033[0m";  }
# проверка наличия аргумента папки РК для восстановления
if [[ -z "$1" ]]; then
   err "Ошибка! Нет аргумента в виде папки!"
   echo -e "\n$help"
   echo -e "\nСписок доступных резервных копий:"
   ls "$aldpro_backups"
   exit 1
fi
# Переход в директорию с резервными копиями
RESTORE_PATH="${aldpro_backups}$1"
# Перейти в директорию с резервными копиями
cd $RESTORE_PATH
# Получаем ошибку если нет такой директории резервной копии
check_backup=$?
# Проверим дату бекапа в аргументе, иначе выведем список дат для удобства
if [[ "$check_backup" -gt 0 ]]; then
   err "Ошибка! Нет $RESTORE_PATH резервной копии!"
   echo "Выберите дату резервной копии:"
   ls "$aldpro_backups"
   exit "$check_backup"
fi

log "Начато восстановление $RESTORE_PATH резервной копии!"
log "Остановка затрагиваемых восстановлением сервисов"
systemctl stop winbind smbd nmbd aldpro-salt-minion
# log "Восстановление логов"
# tar -C "/" -xvf log.tar.gz
log "Восстановление директории etc"
tar -C "/" -xvf etc.tar.gz
log "Восстановление ipa-client"
tar -C "/" -xvf ipa-client.tar.gz
log "Восстановление общих директорий samba"
tar -C "/" -xvf samba.tar.gz

log "Восстановление завершено. Перезагрузка!"
sleep 5 && reboot
```

10.3.3 Подсистема печати

Для выполнения резервного копирования создадим файл *backup_ cups.sh* в домашней папке *root* с таким содержимым :

```
#!/usr/bin/env bash
now="$(date +'%Y-%m-%dT%H-%M-%S')"
app="$(echo $0 | cut -d '/' -f 2 )"
# функция логирования
function log() { logger -t "$app" "$1"; echo "$1"; }
# папка резервной копии ALD Pro
aldpro_backups="/var/backups/aldpro/"
aldpro_version="$(cat /etc/apt/sources.list.d/aldpro.list | cut -d '/' -f 7 | cut -d
↪' ' -f 1)"
log "Начато создание РК ${app} ${aldpro_version} ${HOSTNAME} ${now}"

# Название директории для резервной копии формируется
# из текущей даты, версии и находится в папке $aldpro_backups
BACKUP_PATH="${aldpro_backups}${now}-${aldpro_version}"

log "Создание временной директории для резервных копий $BACKUP_PATH"
mkdir -p $BACKUP_PATH

log "Остановка затрагиваемых бэкапом сервисов"
systemctl stop cups aldpro-salt-minion
log "Архивирование директории ipa-client"
tar -zcvf $BACKUP_PATH/ipa-client.tar.gz /var/lib/ipa-client/
# архивирование логов
tar -zcvf $BACKUP_PATH/log.tar.gz --exclude=faillog --exclude=lastlog /var/log/
log "Архивирование директории etc"
tar -zcvf $BACKUP_PATH/etc.tar.gz /etc/
log "Запуск затрагиваемых бэкапом сервисов"
systemctl start cups aldpro-salt-minion

# Вместо /mnt/ вы можете настроить свою директорию (облако или сетевой диск)
log "Копируем директорию резервной копии на другой носитель"
cp -r $BACKUP_PATH /mnt/

log "Удаляем резервные копии старше 30 дней для ротации"
find "${aldpro_backups}" -type d -mtime +30 -exec rm -rf {} \;
log "Создание копии ${app} завершено!"
```

Для восстановления подсистемы печати создадим файл *restore_ cups.sh* в домашней папке *root* с таким содержимым:

```
#!/bin/bash
app="$(echo $0 | cut -d '/' -f 2 )"
help="Для запуска восстановления нужно указать папку аргументом\n ./${app} DATE-DIR"
aldpro_backups="/var/backups/aldpro/"
# функция логирования
function log() { logger -t "$app" "$1"; echo "$1";  }
function err() { logger --priority user.error -t "$app" "$1"; echo -e "\033[0;31m$1\
↪033[0m";  }
# проверка наличия аргумента папки РК для восстановления
if [[ -z "$1" ]]; then
```

(continues on next page)

(продолжение с предыдущей страницы)

```
    err "Ошибка! Нет аргумента в виде папки!"
    echo -e "\n$help"
    echo -e "\nСписок доступных резервных копий:"
    ls "$aldpro_backups"
    exit 1
fi
# Переход в директорию с резервными копиями
RESTORE_PATH="${aldpro_backups}$1"
# Перейти в директорию с резервными копиями
cd $RESTORE_PATH
# Получаем ошибку если нет такой директории резервной копии
check_backup=$?
# Проверим дату бекапа в аргументе, иначе выведем список дат для удобства
if [[ "$check_backup" -gt 0 ]]; then
    err "Ошибка! Нет $RESTORE_PATH резервной копии!"
    echo "Выберите дату резервной копии:"
    ls "$aldpro_backups"
    exit "$check_backup"
fi

log "Начато восстановление $RESTORE_PATH резервной копии!"
set -e # для возможности отмены Ctrl+Z и если ошибка, то выход будет сразу
log "Остановка затрагиваемых восстановлением сервисов"
systemctl stop cups aldpro-salt-minion
# log "Восстановление логов"
tar -C "/" -xvf log.tar.gz
log "Восстановление директории etc"
tar -C "/" -xvf etc.tar.gz
log "Восстановление ipa-client"
tar -C "/" -xvf ipa-client.tar.gz

log "Восстановление завершено. Перезагрузка!"
sleep 5 && reboot
```

10.3.4 Подсистема репозиториев ПО

Для выполнения резервного копирования подсистемы репозиториев ПО создадим файл *backup_repo.sh* в домашней папке *root* с таким содержимым:

```
#!/usr/bin/env bash
now="$(date +'%Y-%m-%dT%H-%M-%S')"
app="$(echo $0 | cut -d '/' -f 2 )"
# функция логирования
function log() { logger -t "$app" "$1"; echo "$1"; }
# папка резервной копии ALD Pro
aldpro_backups="/var/backups/aldpro/"
aldpro_version="$(cat /etc/apt/sources.list.d/aldpro.list | cut -d '/' -f 7 | cut -d
↪' ' -f 1)"
log "Начато создание РК ${app} ${aldpro_version} ${HOSTNAME} ${now}"

# Название директории для резервной копии формируется
# из текущей даты, версии и находится в папке $aldpro_backups
BACKUP_PATH="${aldpro_backups}${now}-${aldpro_version}"
```

(continues on next page)

(продолжение с предыдущей страницы)

```
log "Создание временной директории для резервных копий $BACKUP_PATH"
mkdir -p $BACKUP_PATH

log "Остановка затрагиваемых бэкапом сервисов"
systemctl stop apache2 postgresql rabbitmq-server aldpro-salt-minion
log "Архивирование БД PostgreSQL"
tar -zcvf $BACKUP_PATH/postgresql.tar.gz /var/lib/postgresql/
log "Архивирование данных брокера очередей RabbitMQ"
tar -zcvf $BACKUP_PATH/rabbitmq.tar.gz /var/lib/rabbitmq/mnesia/
log "Архивирование директории ipa-client"
tar -zcvf $BACKUP_PATH/ipa-client.tar.gz /var/lib/ipa-client/
log "Архивирование директории repo/storage"
tar -zcvf $BACKUP_PATH/storage.tar.gz /opt/rbta/aldpro/repo/storage/
# архивирование логов
log "Архивирование логов"
tar -zcvf $BACKUP_PATH/log.tar.gz --exclude=faillog --exclude=lastlog /var/log/

log "Запуск бэкапируемых сервисов"
systemctl start apache2 postgresql rabbitmq-server aldpro-salt-minion

# Вместо /mnt/ вы можете настроить свою директорию (облако или сетевой диск)
log "Копируем директорию резервной копии на другой носитель"
cp -r $BACKUP_PATH /mnt/

log "Удаляем резервные копии старше 30 дней для ротации"
find "${aldpro_backups}" -type d -mtime +30 -exec rm -rf {} \;
log "Создание копии ${app} завершено!"
```

Для восстановления подсистемы репозитория создадим файл *restore_ repo.sh* в домашней папке *root* с таким содержимым:

```
#!/bin/bash
app="$(echo $0 | cut -d '/' -f 2 )"
help="Для запуска восстановления нужно указать папку аргументом\n ./${app} DATE-DIR"
aldpro_backups="/var/backups/aldpro/"
# функция логирования
function log() { logger -t "$app" "$1"; echo "$1";  }
function err() { logger --priority user.error -t "$app" "$1"; echo -e "\033[0;31m$1\
↪033[0m";  }
# проверка наличия аргумента папки РК для восстановления
if [[ -z "$1" ]]; then
   err "Ошибка! Нет аргумента в виде папки!"
   echo -e "\n$help"
   echo -e "\nСписок доступных резервных копий:"
   ls "$aldpro_backups"
   exit 1
fi
RESTORE_PATH="${aldpro_backups}$1"
# Перейти в директорию с резервными копиями
cd "$RESTORE_PATH"
# Получаем ошибку если нет такой директории резервной копии
check_backup=$?
# Проверим дату бекапа в аргументе, иначе выведем список дат для удобства
if [[ "$check_backup" -gt 0 ]]; then
```

(continues on next page)

(продолжение с предыдущей страницы)

```
    err "Ошибка! Нет $RESTORE_PATH резервной копии!"
    echo "Выберите дату резервной копии:"
    ls "$aldpro_backups"
    exit "$check_backup"
fi
log "Начато восстановление $RESTORE_PATH резервной копии!"
set -e # для возможности отмены Ctrl+Z и если ошибка, то выход будет сразу
log "Остановка затрагиваемых восстановлением сервисов"
systemctl stop apache2 postgresql rabbitmq-server aldpro-salt-minion
log "Восстановление БД PostgreSQL"
tar -C "/" -xvf postgresql.tar.gz
log "Восстановление RabbitMQ"
tar -C "/" -xvf rabbitmq.tar.gz
# # Восстановление логов
tar -C "/" -xvf log.tar.gz
log "Восстановление директории etc"
tar -C "/" -xvf etc.tar.gz
log "Восстановление ipa-client"
tar -C "/" -xvf ipa-client.tar.gz
log "Восстановление директории с репозиториями"
tar -C "/" -xvf storage.tar.gz

log "Восстановление завершено. Перезагрузка!"
sleep 5 && reboot
```

10.3.5 Подсистема DHCP

Для выполнения резервного копирования подсистемы DHCP создадим файл *backup_ dhcp.sh* в домашней папке *root* с таким содержимым:

```
#!/usr/bin/env bash
now="$(date +'%Y-%m-%dT%H-%M-%S')"
app="$(echo $0 | cut -d '/' -f 2 )"
# функция логирования
function log() { logger -t "$app" "$1"; echo "$1"; }
# папка резервной копии ALD Pro
aldpro_backups="/var/backups/aldpro/"
aldpro_version="$(cat /etc/apt/sources.list.d/aldpro.list | cut -d '/' -f 7 | cut -d
↪' ' -f 1)"
log "Начато создание РК ${app} ${aldpro_version} ${HOSTNAME} ${now}"

# Название директории для резервной копии формируется
# из текущей даты, версии и находится в папке $aldpro_backups
BACKUP_PATH="${aldpro_backups}${now}-${aldpro_version}"

log "Создание временной директории для резервных копий $BACKUP_PATH"
mkdir -p $BACKUP_PATH
log "Остановка затрагиваемых бэкапом сервисов"
systemctl stop isc-dhcp-server aldpro-salt-minion
# Архивирование директории ipa-client
tar -zcvf $BACKUP_PATH/ipa-client.tar.gz /var/lib/ipa-client/

# Архивирование директории etc
```

(continues on next page)

(продолжение с предыдущей страницы)

```
tar -zcvf $BACKUP_PATH/etc.tar.gz /etc/

log "Запуск затрагиваемых бэкапом сервисов"
systemctl start isc-dhcp-server aldpro-salt-minion

# Вместо /mnt/ вы можете настроить свою директорию (облако или сетевой диск)
log "Копируем директорию резервной копии на другой носитель"
cp -r $BACKUP_PATH /mnt/

log "Удаляем резервные копии старше 30 дней для ротации"
find "${aldpro_backups}" -type d -mtime +30 -exec rm -rf {} \;
log "Создание копии ${app} завершено!"
```

Для восстановления подсистемы DHCP создадим файл *restore_ dhcp.sh* в домашней папке *root* с таким содержимым:

```
#!/bin/bash
app="$(echo $0 | cut -d '/' -f 2 )"
help="Для запуска восстановления нужно указать папку аргументом\n ./${app} DATE-DIR"
aldpro_backups="/var/backups/aldpro/"
# функция логирования
function log() { logger -t "$app" "$1"; echo "$1";  }
function err() { logger --priority user.error -t "$app" "$1"; echo -e "\033[0;31m$1\
↪033[0m";  }
# проверка наличия аргумента папки РК для восстановления
if [[ -z "$1" ]]; then
   err "Ошибка! Нет аргумента в виде папки!"
   echo -e "\n$help"
   echo -e "\nСписок доступных резервных копий:"
   ls "$aldpro_backups"
   exit 1
fi
RESTORE_PATH="${aldpro_backups}$1"
# Перейти в директорию с резервными копиями
cd $RESTORE_PATH
# Получаем ошибку если нет такой директории резервной копии
check_backup=$?
# Проверим дату бекапа в аргументе, иначе выведем список дат для удобства
if [[ "$check_backup" -gt 0 ]]; then
   err "Ошибка! Нет $RESTORE_PATH резервной копии!"
   echo "Выберите дату резервной копии:"
   ls "$aldpro_backups"
   exit "$check_backup"
fi
log "Начато восстановление $RESTORE_PATH резервной копии!"
set -e # для возможности отмены Ctrl+Z и если ошибка, то выход будет сразу
log "Остановка затрагиваемых восстановлением сервисов"
systemctl stop isc-dhcp-server aldpro-salt-minion
log "Восстановление директории etc"
tar -C "/" -xvf etc.tar.gz
log "Восстановление  ipa-client"
tar -C "/" -xvf ipa-client.tar.gz

log "Восстановление завершено. Перезагрузка!"
sleep 5 && reboot
```

10.3.6 Подсистема PXE

Для выполнения резервного копирования подсистемы PXE создадим файл *backup_pxe.sh* в домашней папке *root* с таким содержимым:

```
#!/usr/bin/env bash
now="$(date +'%Y-%m-%dT%H-%M-%S')"
app="$(echo $0 | cut -d '/' -f 2 )"
# функция логирования
function log() { logger -t "$app" "$1"; echo "$1"; }
# папка резервной копии ALD Pro
aldpro_backups="/var/backups/aldpro/"
aldpro_version="$(cat /etc/apt/sources.list.d/aldpro.list | cut -d '/' -f 7 | cut -d
↪' ' -f 1)"
log "Начато создание РК ${app} ${aldpro_version} ${HOSTNAME} ${now}"

# Название директории для резервной копии формируется
# из текущей даты, версии и находится в папке $aldpro_backups
BACKUP_PATH="${aldpro_backups}${now}-${aldpro_version}"

log "Создание временной директории для резервных копий $BACKUP_PATH"
mkdir -p $BACKUP_PATH
log "Остановка затрагиваемых бэкапом сервисов"
systemctl stop apache2 tftpd-hpa postgresql aldpro-salt-minion
log "Архивирование PostgreSQL"
tar -zcvf $BACKUP_PATH/postgresql.tar.gz /var/lib/postgresql/
log "Архивирование логов"
tar -zcvf $BACKUP_PATH/log.tar.gz --exclude=faillog --exclude=lastlog /var/log/
log "Архивирование директории etc"
tar -zcvf $BACKUP_PATH/etc.tar.gz /etc/
log "Архивирование директории ipa-client"
tar -zcvf $BACKUP_PATH/ipa-client.tar.gz /var/lib/ipa-client/
log "Архивирование директории tftp"
tar -zcvf $BACKUP_PATH/tftp.tar.gz /var/www/tftp/
log "Запуск затрагиваемых бэкапом сервисов"
systemctl start apache2 tftpd-hpa postgresql aldpro-salt-minion

# Вместо /mnt/ вы можете настроить свою директорию (облако или сетевой диск)
log "Копируем директорию резервной копии на другой носитель"
cp -r $BACKUP_PATH /mnt/

log "Удаляем резервные копии старше 30 дней для ротации"
find "${aldpro_backups}" -type d -mtime +30 -exec rm -rf {} \;
log "Создание копии ${app} завершено!"
```

Для восстановления подсистемы PXE создадим файл *restore_pxe.sh* в домашней папке *root* с таким содержимым:

```
#!/bin/bash
app="$(echo $0 | cut -d '/' -f 2 )"
help="Для запуска восстановления нужно указать папку аргументом\n ./${app} DATE-DIR"
aldpro_backups="/var/backups/aldpro/"
# функция логирования
function log() { logger -t "$app" "$1"; echo "$1";  }
function err() { logger --priority user.error -t "$app" "$1"; echo -e "\033[0;31m$1\
```

(continues on next page)

(продолжение с предыдущей страницы)

```
↪033[0m";  }
# проверка наличия аргумента папки РК для восстановления
if [[ -z "$1" ]]; then
   err "Ошибка! Нет аргумента в виде папки!"
   echo -e "\n$help"
   echo -e "\nСписок доступных резервных копий:"
   ls "$aldpro_backups"
   exit 1
fi
RESTORE_PATH="${aldpro_backups}$1"
# Перейти в директорию с резервными копиями
cd $RESTORE_PATH
# Получаем ошибку если нет такой директории резервной копии
check_backup=$?
# Проверим дату бекапа в аргументе, иначе выведем список дат для удобства
if [[ "$check_backup" -gt 0 ]]; then
   err "Ошибка! Нет $RESTORE_PATH резервной копии!"
   echo "Выберите дату резервной копии:"
   ls "$aldpro_backups"
   exit "$check_backup"
fi
log "Начато восстановление $RESTORE_PATH резервной копии!"
set -e # для возможности отмены Ctrl+Z и если ошибка, то выход будет сразу
log "Остановка затрагиваемых восстановлением сервисов"
systemctl stop apache2 tftpd-hpa postgresql aldpro-salt-minion
log "Восстановление БД PostgreSQL"
tar -C "/" -xvf postgresql.tar.gz
log "Восстановление логов"
tar -C "/" -xvf log.tar.gz
log "Восстановление директории etc"
tar -C "/" -xvf etc.tar.gz
log "Восстановление ipa-client"
tar -C "/" -xvf ipa-client.tar.gz
log "Восстановление директории tftp"
tar -C "/" -xvf tftp.tar.gz

log "Восстановление завершено. Перезагрузка!"
sleep 5 && reboot
```

10.3.7 Подсистема мониторинга

Для выполнения резервного копирования подсистемы мониторинга создадим файл *backup_ mon.sh* в домашней папке *root* с таким содержимым:

```
#!/usr/bin/env bash
now="$(date +'%Y-%m-%dT%H-%M-%S')"
app="$(echo $0 | cut -d '/' -f 2 )"
# функция логирования
function log() { logger -t "$app" "$1"; echo "$1"; }
# папка резервной копии ALD Pro
aldpro_backups="/var/backups/aldpro/"
aldpro_version="$(cat /etc/apt/sources.list.d/aldpro.list | cut -d '/' -f 7 | cut -d
↪' ' -f 1)"
```

(continues on next page)

(продолжение с предыдущей страницы)

```
log "Начато создание РК ${app} ${aldpro_version} ${HOSTNAME} ${now}"

# Название директории для резервной копии формируется
# из текущей даты, версии и находится в папке
BACKUP_PATH="${aldpro_backups}${now}-${aldpro_version}"

log "Остановка затрагиваемых бэкапом сервисов"
systemctl stop apache2 zabbix-agent zabbix-server postgresql aldpro-salt-minion
log "Архивирование директории ipa-client"
tar -zcvf $BACKUP_PATH/ipa-client.tar.gz /var/lib/ipa-client/
log "Архивирование логов"
tar -zcvf $BACKUP_PATH/log.tar.gz --exclude=faillog --exclude=lastlog /var/log/
log "Архивирование директории etc"
tar -zcvf $BACKUP_PATH/etc.tar.gz /etc/
log "Архивирование zabbix"
tar -zcvf $BACKUP_PATH/zabbix.tar.gz /usr/share/zabbix/
log "Архивирование БД PostgreSQL"
tar -zcvf $BACKUP_PATH/postgresql.tar.gz /var/lib/postgresql/
# архивирование логов
log "Архивирование логов"
tar -zcvf $BACKUP_PATH/log.tar.gz --exclude=faillog --exclude=lastlog /var/log/

log "Запуск затрагиваемых бэкапом сервисов"
systemctl start apache2 zabbix-agent zabbix-server postgresql aldpro-salt-minion

# Вместо /mnt/ вы можете настроить свою директорию (облако или сетевой диск)
log "Копируем директорию резервной копии на другой носитель"
cp -r $BACKUP_PATH /mnt/

log "Удаляем резервные копии старше 30 дней для ротации"
find "${aldpro_backups}" -type d -mtime +30 -exec rm -rf {} \;
log "Создание копии ${app} завершено!"
```

Для восстановления подсистемы мониторинга создадим файл *restore_mon.sh* в домашней папке *root* с таким содержимым:

```
#!/bin/bash
app="$(echo $0 | cut -d '/' -f 2 )"
help="Для запуска восстановления нужно указать папку аргументом\n ./${app} DATE-DIR"
aldpro_backups="/var/backups/aldpro/"
# функция логирования
function log() { logger -t "$app" "$1"; echo "$1";  }
function err() { logger --priority user.error -t "$app" "$1"; echo -e "\033[0;31m$1\
↪033[0m";  }
# проверка наличия аргумента папки РК для восстановления
if [[ -z "$1" ]]; then
   err "Ошибка! Нет аргумента в виде папки!"
   echo -e "\n$help"
   echo -e "\nСписок доступных резервных копий:"
   ls "$aldpro_backups"
   exit 1
fi
RESTORE_PATH="${aldpro_backups}$1"
# Перейти в директорию с резервными копиями
cd $RESTORE_PATH
```

(continues on next page)

(продолжение с предыдущей страницы)

```
# Получаем ошибку если нет такой директории резервной копии
check_backup=$?
# Проверим дату бекапа в аргументе, иначе выведим список дат для удобства
if [[ "$check_backup" -gt 0 ]]; then
   err "Ошибка! Нет $RESTORE_PATH резервной копии!"
   echo "Выберите дату резервной копии:"
   ls "$aldpro_backups"
   exit "$check_backup"
fi
log "Начато восстановление $RESTORE_PATH резервной копии!"
set -e # для возможности отмены Ctrl+Z и если ошибка, то выход будет сразу
log "Остановка затрагиваемых бэкапом сервисов"
systemctl stop apache2 zabbix-agent zabbix-server postgresql aldpro-salt-minion
log "Восстановление БД PostgreSQL"
tar -C "/" -xvf postgresql.tar.gz
log "Восстановление ipa-client"
tar -C "/" -xvf ipa-client.tar.gz
log "Восстановление директории etc"
tar -C "/" -xvf etc.tar.gz
log "Восстановление директории zabbix"
tar -C "/" -xvf zabbix.tar.gz
log "Восстановление логов"
tar -C "/" -xvf log.tar.gz

log "Восстановление завершено. Перезагрузка!"
sleep 5 && reboot
```

10.3.8 Подсистема журналирования

Для выполнения резервного копирования подсистемы журналирования создадим файл *backup_ audit.sh* в домашней папке *root* с таким содержимым:

```
#!/usr/bin/env bash
now="$(date +'%Y-%m-%dT%H-%M-%S')"
app="$(echo $0 | cut -d '/' -f 2 )"
# функция логирования
function log() { logger -t "$app" "$1"; echo "$1"; }
# папка резервной копии ALD Pro
aldpro_backups="/var/backups/aldpro/"
aldpro_version="$(cat /etc/apt/sources.list.d/aldpro.list | cut -d '/' -f 7 | cut -d
↪' ' -f 1)"
log "Начато создание РК ${app} ${aldpro_version} ${HOSTNAME} ${now}"
# Название директории для резервной копии формируется
# из текущей даты, версии и находится в папке ${aldpro_backups}
BACKUP_PATH="${aldpro_backups}${now}-${aldpro_version}"
log "Создание временной директории для резервных копий $BACKUP_PATH"
mkdir -p $BACKUP_PATH
# Архивирование логов
tar --exclude='/var/log/lastlog' --exclude='/var/log/wtmp' -zcvf $BACKUP_PATH/log.tar.
↪gz /var/log/
# Архивирование директории etc
tar -zcvf $BACKUP_PATH/etc.tar.gz /etc/
# Архивирование директории ipa-client
```

(continues on next page)

(продолжение с предыдущей страницы)

```
tar -zcvf $BACKUP_PATH/ipa-client.tar.gz /var/lib/ipa-client/

# Вместо /mnt/ вы можете настроить свою директорию (облако или сетевой диск)
log "Копируем директорию резервной копии на другой носитель"
cp -r $BACKUP_PATH /mnt/

log "Удаляем резервные копии старше 30 дней для ротации"
find "${aldpro_backups}" -type d -mtime +30 -exec rm -rf {} \;
log "Создание копии ${app} завершено!"
```

Для восстановления создадим файл *restore_ audit.sh* в домашней папке *root* с таким содержимым:

```
#!/bin/bash
app="$(echo $0 | cut -d '/' -f 2 )"
help="Для запуска восстановления нужно указать папку аргументом\n ./${app} DATE-DIR"
aldpro_backups="/var/backups/aldpro/"
# функция логирования
function log() { logger -t "$app" "$1"; echo "$1";  }
function err() { logger --priority user.error -t "$app" "$1"; echo -e "\033[0;31m$1\
↪033[0m";  }
# проверка наличия аргумента папки РК для восстановления
if [[ -z "$1" ]]; then
   err "Ошибка! Нет аргумента в виде папки!"
   echo -e "\n$help"
   echo -e "\nСписок доступных резервных копий:"
   ls "$aldpro_backups"
   exit 1
fi
RESTORE_PATH="${aldpro_backups}$1"
# Перейти в директорию с резервными копиями
cd $RESTORE_PATH
# Получаем ошибку если нет такой директории резервной копии
check_backup=$?
# Проверим дату бекапа в аргументе, иначе выведем список дат для удобства
if [[ "$check_backup" -gt 0 ]]; then
   err "Ошибка! Нет $RESTORE_PATH резервной копии!"
   echo "Выберите дату резервной копии:"
   ls "$aldpro_backups"
   exit "$check_backup"
fi
log "Начато восстановление $RESTORE_PATH резервной копии!"
set -e # для возможности отмены Ctrl+Z и если ошибка, то выход будет сразу

log "Остановка затрагиваемых восстановлением сервисов"
systemctl stop syslog-ng aldpro-salt-minion
log "Восстановление логов"
tar -C "/" -xvf log.tar.gz
log "Восстановление директории etc"
tar -C "/" -xvf etc.tar.gz
log "Восстановление ipa-client"
tar -C "/" -xvf ipa-client.tar.gz

log "Восстановление завершено. Перезагрузка!"
sleep 5 && reboot
```

10.3.9 Настройка регулярного резервного копирования через CRON

Допустим, мы создали скрипт /root/backup_dc.sh для резервного копирования контроллера домена и хотим запускать его регулярно в нерабочие часы. Для этого мы можем настроить планировщик заданий cron.

Чтобы перейти к редактированию конфигурационного файла cron, выполните команду `sudo crontab -e`. Задания cron задаются в следующем синтаксисе:

```
минута час день_месяца месяц день_недели команда
```

Где:

- **минута** — значение от 0 до 59, которое указывает, на какой минуте в пределах часа должна быть выполнена указанная команда.

  Например, значение 5 означает, что задание требуется выполнить на 5-й минуте.

  Если указать символ «*», то задание будет выполняться каждую минуту. Если указать «*/5», то задание будет выполняться через каждые 5 минут.

- **час** — значение от 0 до 23, которое указывает, в каком часу в пределах суток должна быть выполнена команда. Например, значение 4 означает, что задание нужно выполнять в 4 часа ночи.

  Так же как с минутами, символ «*» означает, то задание будет выполняться каждый час, а если указать «*/4», то через каждые 4 часа.

- **день_месяца** — значение от 1 до 31, которое указывает, в какой день месяца нужно выполнять задание. Например, значение 3 означает, что задание нужно выполнять 3-го числа каждого месяца. Если указать «*», то задание не будет привязано к дням месяца.

- **месяц** — значение от 1 до 12, которое указывает, в каком месяце требуется выполнять задание. Например, значение 2 означает, что задание нужно выполнять в феврале. Если указать «*», то задание не будет привязано к месяцам года.

- **день_недели** — значение от 0 до 6, которое указывает, по каким дням недели нужно выполнять задание. Отсчет начинается с воскресенья (0) и заканчивается субботой (6). Например, значение 1 означает, что задание нужно выполнять по понедельникам.

  Вместо цифр можно использовать названия дней недели в формате SUN, MON и т.д. Если указать «*», то задание не будет привязано к дням недели.

- **команда** — задает команду, которая должна выполняться по указанному расписанию. Может содержать полный путь к файлу, параметры, инструкции перенаправления вывода. Если

используются пробелы, команду лучше ограничить кавычками.

Приведем несколько полных примеров:

- Например, если мы захотим запускать скрипт в час ночи, то нужно ввести строку:

```
0 1 * * * /root/backup_dc.sh
```

- Если нужно запускать скрипт в час ночи, но только по воскресениям:

```
0 1 * * sun /root/backup_dc.sh
```

- Если вы хотите запускать скрипт в час ночи, но только один раз в месяц 1-го числа:

```
0 1 1 * * /root/backup_dc.sh
```

10.4 Практика и тестирование

10.4.1 Практическая работа: Модуль 10. Резервное копирование ALD Pro

См. также *Требования, правила и цели выполнения практической работы*

10.4.1.1 Практические задания

Задание 1. Резервное копирование dc-1 через снимок виртуальной машины

1. Выключите dc-1.
2. Сделайте снимок виртуальной машины dc-1.
3. Загрузите виртуальную машину и выполните какие-нибудь действия на контроллере, например, создайте пользователя через портал управления.
4. Выключите машину dc-1 и восстановите ее из снимка.
5. Проверьте статус репликации.
6. Устраните проблемы с репликацией через переинициализацию контроллера.

Задание 2. Резервное копирование dc-1 через копирование данных

1. Зайдите на dc-1.
2. Запустите скрипт создания резервной копии.

3. Проверьте файлы резервной копии.

4. Выполните какие-нибудь действия на контроллере, например, создайте пользователя через портал управления.

5. Запустите скрипт восстановления контроллера из резервной копии.

6. Проверьте статус репликации.

7. Устраните проблемы с репликацией через переинициализацию контроллера.

Задание 3. Регулярные задачи cron

1. Сохраните скрипт резервного копирования контроллера на dc-1 в файл.

2. Создайте задачу в cron для ежедневного резервного копирования контроллера домена, которая должна запускаться в час ночи.

Задание 4 (опционально)

1. Сделайте резервное копирование остальных подсистем

10.4.2 Тест: Модуль 10. Резервное копирование ALD Pro

Пожалуйста, ответьте на несколько вопросов, чтобы закрепить полученные знания. Правильные ответы для самопроверки вы сможете найти в конце списка. Обратите внимание, что у некоторых вопросов может быть несколько верных вариантов ответов.

10.4.2.1 Вопросы

Вопрос 1. Можно ли восстанавливать контроллер из снимка виртуальной машины?

a. Нет, нельзя

b. Да, можно, если контроллер один

c. Да, можно, если сделать переинициализацию

Вопрос 2. Можно ли выполнять резервное копирование единственного контроллера?

a. Нет, резервное копирование предполагает остановку сервисов

b. Да, если настроить выполнение копирования в ночное время по расписанию

Вопрос 3. Какой командой можно переинициализировать контроллер?

a. Команда `ipa-replica-manage re-initialize --from dc-2`

b. Команда `ipa-replica-install`

c. Команда `ipa-replica-conncheck --replica dc-2`

d. Команда `ipa-restore`

Вопрос 4. Какой командой Red Hat предлагает смотреть статус реплики?

a. Команда `ipactl status`

b. Команда `ipa user admin`

c. Команда `ds-replcheck`

d. Команда `sudo cat ~/.dsrc`

Вопрос 5. В каком файле можно увидеть ошибки репликации после восстановления контроллера из резервной копии?

a. Файл `/var/log/dirsrv/slapd-ALD-COMPANY-LAN/audit`

b. Файл `/var/log/dirsrv/slapd-ALD-COMPANY-LAN/errors`

c. Файл `/var/log/dirsrv/slapd-ALD-COMPANY-LAN/access`

d. Файл `/var/log/dirsrv/slapd-ALD-COMPANY-LAN/security`

Ответы на вопросы

1. b, c
2. b
3. a
4. c
5. b

10.5 Заключение

Вот мы и подошли к завершению. Осталось только понять, как можно интегрировать дополнительные информационные системы с каталогом, чтобы извлечь максимум выгоды от внедрения каталога.

10.6 Дополнительные источники информации

- Отсутствуют

10.7 Обратная связь

Если остались вопросы, то их всегда можно задать в специальной теме.

Модуль 11. Дополнительные подсистемы и интеграции

11.1 Введение

Из этого модуля вы узнаете о том, как устроены подсистемы, обеспечивающие управление локальными репозиториями, общий доступ к принтерам, мониторинг и аудит. Мы также рассмотрим вопросы интеграции службы каталога со сторонними клиент-серверными приложениями.

11.2 Изменения в версии ALD Pro 2.4.0

Как изменилась схема развертывания и конфигурирования подсистем, мы подробно рассмотрели в *Модуль 7. Дополнительные подсистемы ALD Pro*. Напомним только, что теперь подсистемы управляются с помощью автономной службы salt-minion. У этого решения есть достоинство - требования к аппаратному обеспечению существенно снизились. Однако есть и недостаток - теперь нельзя форсировать установку подсистем и применение конфигурации по инициативе контроллера домена. Чтобы целевой компьютер принял назначенную ему роль или применил новую конфигурацию, требуется подождать в пределах 30 минут, либо форсировать эти действия непосредственно на самом компьютере.

11.2.1 Команды форсирования

Когда мы устанавливаем, обновляем или удаляем подсистемы с портала управления, команды на целевой машине выполняются по таймеру. Форсировать процесс можно, перейдя на соответствую-

щий компьютер и дав там команду aldpro-roles. Команда может быть отдана в таком виде:

```
sudo aldpro-roles --iud
```

В этом случае она соберет данные из LDAP и поместит их в файл `/etc/aldpro-salt/stack/deploy/subsystem.yml`. Затем выполнит установку, обновление либо удаление подсистемы согласно полученным данным.

Можно уточнить, какие именно действия следует выполнить:

```
admin@file-1:~$ sudo aldpro-roles --iud --action [install, update, remove]
```

В таком случае команда соберет данные из LDAP, поместит их в файл и выполнит действие, указанное параметром *–action*.

Файл `/etc/aldpro-salt/stack/deploy/subsystem.yml` имеет такой формат:

```
aldpro_deploy_data:
action:install
location: hq
service_name: log
site: {имя сайта}
state_created: {временная метка}
state_updated: {временная метка}
target_host: {имя машины}
```

В случае если с портала управления выполняется настройка подсистемы, новые настройки будут переданы на подсистему через определенный временной интервал. Немедленно применить новые параметры можно командой:

```
admin@file-1:~# aldpro-roles --subsystem_settings [имя подсистемы: 'cups', 'dhcp',
↪'smb', 'os', 'repo', 'audit']
```

Такая команда соберет конфигурацию из LDAP и применит к указанной подсистеме. Пример:

```
sudo aldpro-roles --subsystem_settings audit
```

Внимание: Нельзя одновременно установить подсистему и получить для нее параметры. Команда в такой форме *aldpro-roles –iud –action install –subsystem_ settings smb* вызовет ошибку!

11.3 Подсистема «Репозиторий программного обеспечения»

Система Astra Linux является дериватом Debian, поэтому приложения устанавливаются из deb-пакетов, которые распространяются через специальные APT-совместимые репозитории. В свое время идея Debian поставлять систему с набором заранее скомпилированных и четко согласованных

пакетов стала настоящей революцией в мире Linux.

До этого для установки дополнительного программного обеспечения пользователям нужно было обладать квалификацией на уровне системного программиста. Требовалось разобраться, из каких компонентов состоит приложение, каких версий должны быть эти компоненты, в каком порядке их нужно компилировать и устанавливать в системе. А теперь `apt install`, и можно пользоваться.

Как вы уже знаете, официальные интернет-репозитории операционной системы Astra Linux доступны по адресу https://download.astralinux.ru и включают в себя четыре основных раздела: *main*, *update*, *base*, *extended*. Репозитории *main* и *update* попадают под сертификацию, репозиторий *base* включает пакеты *main* и *update*, а также содержит некоторые дополнительные средства разработки. Репозиторий *extended* содержит дополнительное программное обеспечение, которое может функционировать в среде Astra Linux, но не дорабатывается для реализации функций безопасности.

В официальных репозиториях находятся тысячи пакетов, готовых к установке, но вам все равно могут потребоваться собственные корпоративные репозитории, например, для установки приложений сторонних разработчиков или эксплуатации системы в закрытом периметре. Для решения этой задачи в продукте ALD Pro есть специальная подсистема на базе **Reprepro** и **Apache**.

11.3.1 Репозитории пакетного менеджера APT

Мы уже рассматривали подробно устройство deb-пакетов и работу пакетного менеджера APT (см. *Модуль 15. Управление ПО в Astra Linux*), поэтому напомним только основные моменты.

Менеджер APT может извлекать пакеты по протоколу HTTP(S), для чего в файле `sources.list` требуется задать источник в определенном формате, см. рис. 2.233.



рис. 2.233 – Синтаксис строки репозитория debian

Назначение элементов синтаксиса:

- **deb** — указывает на то, что репозиторий соответствует репозиторию бинарных файлов с предварительно скомпилированными пакетами. Для репозиториев с исходными кодами используют «deb-src».
- **https://download.astralinux.ru/astra/stable/1.7_x86-64/repository-main/** — задает адрес репозитория. У интернет-репозиториев адрес начинается с `http(s)://`, адреса локаль-

ных репозиториев начинаются с `file:/`. При добавлении репозитория с диска командой `sudo apt-cdrom add` в файле появится строка `deb cdrom:[]`.

- **1.7_x86-64** — кодовое имя дистрибутива. В одном репозитории могут находиться пакеты сразу для нескольких релизов.

- **main contrib non-free** — это группы пакетов, которые объединяются по условиям использования:

  - **non-free** — группа содержит пакеты, которые не соответствуют принципам свободного ПО, имеют патенты или другие юридические ограничения;

  - **contrib** — группа содержит пакеты, которые сами по себе соответствуют принципам свободного ПО, но зависят от пакетов из группы *non-free* (т. е. не могут без них работать);

  - **main** — группа содержит пакеты свободного ПО, которые не зависят от пакетов из групп *contrib* и *non-free*.

Репозиторий представляет собой просто папку с файлами на веб-сервере, но пакеты свалены не в одну кучу, а распределены по каталогам и дополняются файлами с метаинформацией, что позволяет ускорить индексацию репозиториев на клиентских машинах. Для индексации пакетов менеджер загружает в первую очередь файл **Release** или **InRelease** по следующему URL-адресу: `<полный_путь>/dists/код_дистрибутива/` (*от англ. dists — это сокращение от distributions, т.е. дистрибутивы*).

Исследовать содержимое репозиториев Astra затруднительно, т. к. на серверах отключен просмотр содержимого папок, но вы можете обратиться к официальному репозиторию Debian. Например, файл InRelease для Debian 10 Buster находится по адресу: `https://deb.debian.org/debian/dists/buster/InRelease`.

Вот пример содержимого InRelease файла:

```
admin@dc-1:~$ curl https://deb.debian.org/debian/dists/buster/InRelease
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA256

Origin: Debian
Label: Debian
Suite: oldoldstable
Version: 10.13
Codename: buster
Changelogs: http://metadata.ftp-master.debian.org/changelogs/@CHANGEPATH@_changelog
Date: Sat, 10 Jun 2023 08:53:33 UTC
Acquire-By-Hash: yes
Architectures: amd64 arm64 armel armhf i386 mips mips64el mipsel ppc64el s390x
Components: main contrib non-free
Description: Debian 10.13 Released 10 September 2022
MD5Sum:
```

(continues on next page)

(продолжение с предыдущей страницы)

```
...
   bb284daaa0f0afdb56e4e95bd16e8bdc 10688592 main/binary-i386/Packages.gz
   f3d4bcfbd689fef5c5ffb9f8a9148eb0 7865552 main/binary-i386/Packages.xz
   588bad28a94db77998ce094cb82f2728 122 main/binary-i386/Release
...
```

В файле InRelease перечислены индексные файлы дистрибутива, которые пакетный менеджер выкачивает по указанным ссылкам. Например, для обработки ссылки `main/binary-i386/Packages.gz` менеджер apt загрузит файл https://deb.debian.org/debian/dists/buster/main/binary-i386/Packages.gz.

Индексные файлы пакетов содержат метаинформацию по каждому из пакетов. Например, пакет **freeipa-client** относится как раз к *main*, и мы найдем информацию о нем в файле `Packages.gz`:

```
...
Package: freeipa-client
Source: freeipa
Version: 4.7.2-3
Installed-Size: 298
Maintainer: Debian FreeIPA Team <pkg-freeipa-devel@alioth-lists.debian.net>
Architecture: i386
Replaces: freeipa-admintools (<< 4.6.3-2~)
Provides: freeipa-admintools
Depends: bind9utils, certmonger (>= 0.79.5-2), curl, dnsutils, freeipa-common (= 4.7.
↪2-3), krb5-user, libnss3-tools, libnss-sss, libpam-sss, libsasl2-modules-gssapi-mit,
↪ libsss-sudo, libxmlrpc-core-c3
(>= 1.16.33-3.1ubuntu5), oddjob-mkhomedir, python-dnspython, python-ipaclient (= 4.7.
↪2-3), python-gssapi, python-ldap, python-sss, sssd (>= 1.14.0), python:any,␣
↪libbasicobjects0 (>= 0.4.0), libc6 (>=
2.4), libcollection4 (>= 0.4.0), libcom-err2 (>= 1.43.9), libini-config5 (>= 0.4.0),␣
↪libk5crypto3 (>= 1.9+dfsg~beta1), libkrb5-3 (>= 1.13~alpha1+dfsg), libldap-2.4-2 (>
↪= 2.4.7), libnspr4 (>= 2:4.9-2~),
libnss3 (>= 2:3.13.4-2~), libpopt0 (>= 1.14), libref-array1 (>= 0.4.0), libsasl2-2 (>
↪= 2.1.27+dfsg), libssl1.1 (>= 1.1.0)
Recommends: chrony
Suggests: libpam-krb5
Breaks: freeipa-admintools (<< 4.6.3-2~)
Description: FreeIPA centralized identity framework -- client
Homepage: http://www.freeipa.org
Description-md5: 1aa0c0a3f974364e79585d72db619762
Section: net
Priority: optional
Filename: pool/main/f/freeipa/freeipa-client_4.7.2-3_i386.deb
Size: 111540
MD5sum: 6052e1e0e87c8f938b9a2795b5410cb0
SHA256: a1a75a1d0c1e653ae782348e0d61dc05c9117bc77e8651af4c491a94d7d7a877
...
```

Путь к deb-пакету относительно корня репозитория задается в параметре Filename, например, freeipa-client можно загрузить по ссылке https://deb.debian.org/debian/pool/main/f/freeipa/freeipa-client_4.7.2-3_i386.deb.

Вы можете создать все необходимые файлы и папки вручную, но это требует много времени и большой внимательности, поэтому для упрощения работы с репозиториями были разработаны вспомогательные инструменты, такие как `reprepro`.

11.3.2 Архитектура подсистемы репозиториев

Для управления корпоративными репозиториями в продукте ALD Pro используется приложение `reprepro`, которое ранее называлось *mirrorer*. Этот инструмент позволяет управлять базой данных пакетов и файловой структурой apt-репозиториев.

Для каждого репозитория создается каталог `/opt/rbta/aldpro/repo/storage/<repository_id>`, внутри которого находятся следующие подкаталоги:

- **conf** — содержит конфигурационные файлы reprepro. Приложение может использовать следующие три конфигурационных файла:
  - **distributions** — определяет список дистрибутивов репозитория и является обязательным.
  - **options** — определяет параметры репозитория и является опциональным, в ALD Pro не используется.
  - **updates** — определяет настройки для синхронизации с внешними репозиториями и является опциональным, в ALD Pro не используется.
- **db** — содержит базы данных приложения `reprepro` в формате Berkeley DB, которые значительно ускоряют работу с метаинформацией.

  При желании вы можете посмотреть содержимое файлов `*.db` с помощью утилиты `db5.3_dump` из пакета db5.3-util. Пример вывода содержимого:

```
root@repo-1:~# db5.3_dump
/opt/rbta/aldpro/repo/storage/68518a45-d915-499d-9b61-884df60fd316/70606a6d-2947-4eb1-
↪8959-c9e4bb01366a/db/packages.db
VERSION=3
format=print
database=latest|main|amd64
...
```

- **dists** — папка дистрибутивов репозитория. Тут вы найдете метафайлы Release и Packages, которые обновляются автоматически при добавлении/удалении пакетов.
- **pool** — в эту папку загружаются непосредственно файлы *.deb.

Управление репозиторием осуществляется через портал управления, который взаимодействует с сервером репозиториев через API по протоколу HTTPS. Например, для получения списка всех репозиториев фронтенд обращается к API бэкенда портала управления по адресу

`https://dc-1.ald.company.lan/ad/api/repo/repositories?limit=25&sortby=`, а бэкенд пересылает запрос на адрес `https://repo-1.ald.company.lan/repo/api/repositories?limit=25&offset=0&sortby`.

За работу API на стороне репозитория отвечает wsgi-приложение, файл которого расположен по адресу `/opt/rbta/aldpro/repo/core/wsgi.py`. Настройки Apache для этого приложения находятся в файле `/etc/apache2/conf-enabled/rbta-ad-repo.conf`. Программный код для добавления и удаления репозиториев вы найдете в файле `/opt/rbta/aldpro/repo/repository/tasks.py`.

Для ускорения работы подсистемы репозиториев бэкенд API сохраняет всю необходимую информацию в базе PostgreSQL, поэтому управлять репозиториями напрямую через утилиту `reprepro` недопустимо. Вы можете заглянуть в эту SQL базу, используя утилиту `pg_dump`:

```
admin@repo-1:~$ sudo -u postgres pg_dump repo | less
...
COPY public.repository_repositoryversiondeb (id,
repositoryversion_deb_description, repositoryversion_deb_origin,
repositoryversion_deb_label, repositoryversion_deb_version,
repositoryversion_deb_suite, repositoryversion_deb_codename,
repositoryversion_deb_components, repositoryversion_deb_architectures,
repositoryversion_deb_repository_version_id_id) FROM stdin;
1 yandex browser 1 latest main amd64 70606a6d-2947-4eb1-8959-c9e4bb01366a
...
```

Веб-сервер Apache используется не только для API, но и как сервер доставки файлов репозитория до рабочих станций. Дело в том, что утилита `reprepro` решает только узкоспециализированную задачу по управлению apt-совместимыми репозиториями, а для предоставления доступа к этим репозиториям по протоколам HTTP(S) и FTP нужно использовать дополнительно Apache, Nginx или любой другой сервер доставки контента.

После публикации репозитория бэкенд на стороне подсистемы обновляет ссылки в каталоге `/opt/rbta/aldpro/repo/storage/link_root_folder/`, который доступен через HTTP(S), что определено в ранее упомянутом конфигурационном файле веб-сервера `/etc/apache2/conf-enabled/rbta-ad-repo.conf`. Кстати, каталоги `conf` и `db` можно было бы и не показывать по сети, т.к. они не нужны пакетному менеджеру.

```
admin@repo-1:~$ sudo ls -l /opt/rbta/aldpro/repo/storage/link_root_folder/customdeb/
итого 16
lrwxrwxrwx 1 root www-data 108 июл 16 06:37 conf -> /opt/rbta/aldpro/repo/storage/
↪68518a45-d915-499d-9b61-884df60fd316/70606a6d-2947-4eb1-8959-c9e4bb01366a/conf
lrwxrwxrwx 1 root www-data 106 июл 16 06:37 db -> /opt/rbta/aldpro/repo/storage/
↪68518a45-d915-499d-9b61-884df60fd316/70606a6d-2947-4eb1-8959-c9e4bb01366a/db
lrwxrwxrwx 1 root www-data 109 июл 16 06:37 dists -> /opt/rbta/aldpro/repo/storage/
↪68518a45-d915-499d-9b61-884df60fd316/70606a6d-2947-4eb1-8959-c9e4bb01366a/dists
lrwxrwxrwx 1 root www-data 108 июл 16 06:37 pool -> /opt/rbta/aldpro/repo/storage/
↪68518a45-d915-499d-9b61-884df60fd316/70606a6d-2947-4eb1-8959-c9e4bb01366a/pool
```

```
admin@repo-1:~$ cat /etc/apache2/conf-enabled/rbta-ad-repo.conf
...
Alias /repos /opt/rbta/aldpro/repo/storage/link_root_folder/
```

(continues on next page)

(продолжение с предыдущей страницы)

```
<Directory /opt/rbta/aldpro/repo/storage/link_root_folder/ >
   SetHandler none
   AllowOverride None
   Satisfy Any
   Require all granted
   Options +Indexes
   IndexIgnore tmp
</Directory>
...
```

Обратите внимание, что при загрузке пакетов в новую версию репозитория предыдущая версия автоматически снимается с публикации, и ее папка удаляется из каталога `link_root_folder`.

11.3.3 Повышение роли сервера до подсистемы репозитория

Повысить роль сервера до подсистемы репозиториев программного обеспечения можно с портала управления ALD Pro на странице *Установка и обновление ПО → Репозитории ПО → Серверы репозиториев ПО*, см. рис. 2.234. Сайт является обязательным полем, но пока еще не используется в системе. В будущем эта привязка позволит делегировать права доступа на управление подсистемами.

Задание на установку подсистемы отправляется по цепочке Фронтенд > Бэкенд > LDAP > репликация > LDAP > По умолчанию сервер репозитория проверяет LDAP на предмет изменений каждые 30 минут, для ускорения этого процесса можно выполнить следующую команду:

```
admin@repo-1:~$ sudo aldpro-roles --iud
```

Отслеживать статус установки можно в карточке сервера репозитория ПО по пути *Установка и обновление ПО → Репозитории ПО → Серверы репозиториев ПО → Сервер: repo-1.ald.company.lan*. За установку подсистемы отвечает скрипт `install.sls` из каталога `/srv/salt/standalone/roots/states/aldpro/subsystems/repo`.



рис. 2.234 – Добавление сервера репозиториев

Первый сервер устанавливается как основной, а второй сервер репозиториев считается репликой, и его DNS-имя вносится в файл `/etc/lsyncd/lsyncd.conf.lua` в секцию *replicas*.

```
replicas = {
{%- for replica in replicas -%}
"repo@{{ replica }}",
{%- endfor -%}
}
```

Примечание: Если при перезапуске службы lsyncd одна из реплик недоступна, основной процесс lsyncd завершается с ошибкой, что приводит к отсутствию синхронизации репозиториев. Чтобы процесс не завершался, в конфигурационный файл в секции settings можно добавить параметр insist=true. Если вам требуется создать более одной реплики, то после повышения роли сервера нужно вручную добавить имена хостов в файл конфигурации lsyncd и перезапустить службу.

Служба lsyncd подключается к репликам по протоколу ssh и использует утилиту `rsync` для копирования файлов и папок. Ключ для подключения по ssh находится в `/var/lib/repo/.ssh/id_rsa` на основном сервере, соединение происходит под локальной учетной записью repo. Информация о загружаемых пакетах вносится в SQL базу основного сервера, с которого передается на реплики. Изменить данные на репликах невозможно, так как их базы работают только на чтение.

11.3.4 Создание репозитория из ISO-файла

Вы можете создать корпоративный репозиторий из ISO-файла. Для этого вам сначала нужно на странице *Установка и обновление ПО → Репозитории ПО* создать новый репозиторий, см. рис. 2.235.

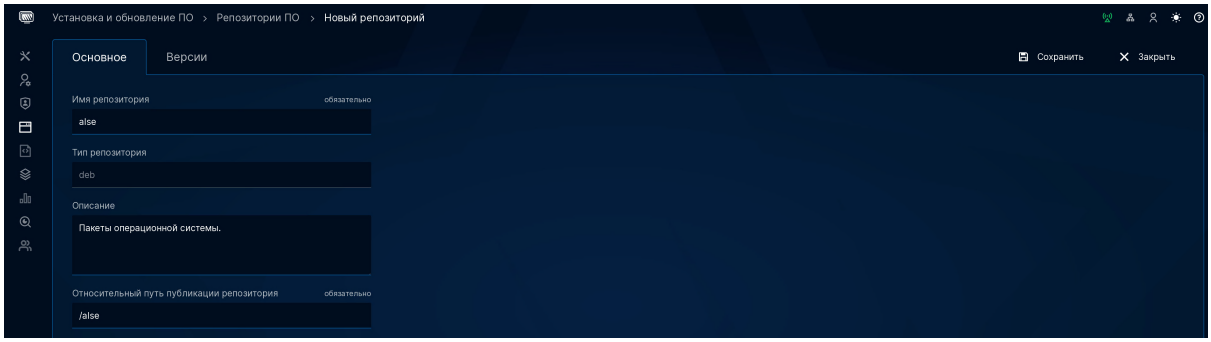


рис. 2.235 – Новый репозиторий с базового образа Astra Linux 1.7.6

Где:

- **Имя репозитория** = **alse** — текстовая строка, которая будет использоваться в качестве имени каталога на сервере.
- **Тип репозитория** = **deb** — поле не редактируется, в настоящий момент поддерживаются толь-

ко APT-совместимые репозитории.

- **Описание** — вспомогательная информация для сопровождения репозиториев.
- **Абсолютный путь публикации** = /**alse** — абсолютный путь к папке на веб-сервере, в которой будут находиться файлы репозитория после публикации.

После сохранения нового репозитория станут доступны дополнительные вкладки для его настройки. Перейдите на вкладку *Версии* и создайте новую версию из ISO-образа, см. рис. 2.236.

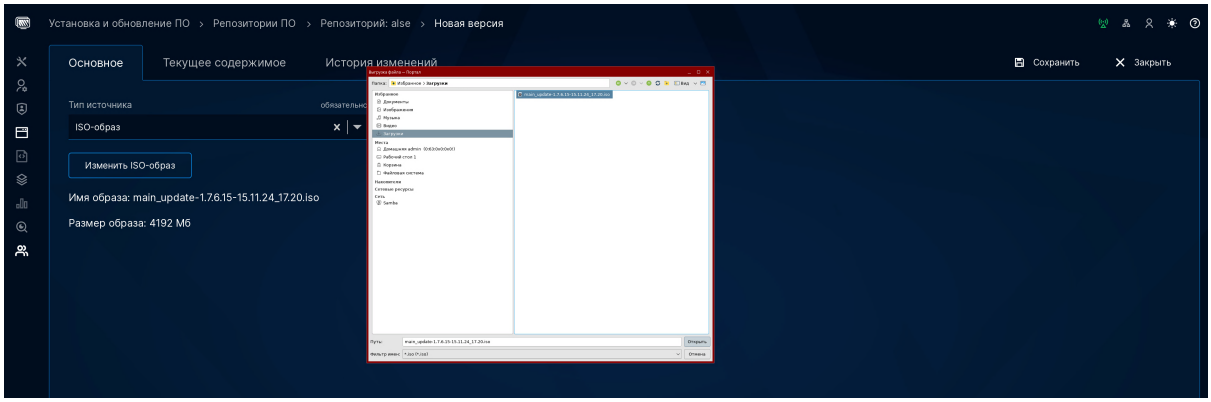


рис. 2.236 – Создание новой версии репозитория из ISO-файла

Если у вас вместо ISO-файла будет установочный компакт-диск, то вы можете создать из него необходимый файл с помощью утилиты dd следующей командой:

```
admin@dc-1:~$ dd if=/dev/sr0 of=al174main.iso bs=100M status=progress
```

При загрузке ISO-файла фронтенд портала управления передает данные на бэкенд, а тот в свою очередь пересылает данные сразу программному интерфейсу основного сервера репозиториев. Передача осуществляется по протоколу HTTPS, а после загрузки файла служба celery на основном сервере репозиториев выполняет в фоне следующие задачи:

- распаковывает файл в каталог storage;
- вносит информацию в базу PostgreSQL, чтобы она была доступна на портале управления на вкладке *Текущее содержимое* ;
- автоматически публикует версию репозитория, после чего она переходит из состояния «Обработка» в «Опубликована».

Чтобы добавить корпоративный репозиторий, создайте файл /etc/apt/sources.list.d/alse.list следующей командой:

```
admin@pc-1:~$ echo "deb [trusted=yes] http://repo-1.ald.company.lan/repos/alse/ 1.7_
↪x86-64 main contrib non-free" | sudo tee /etc/apt/sources.list.d/alse.list
```

Где:

- **[trusted=yes]** — указывает, что источник является доверенным и можно устанавливать пакеты без проверки цифровых подписей;
- **http://repo-1.ald.company.lan/repos/alse/** — задает путь к папке репозитория на веб-сервере, который мы указали ранее. Можно посмотреть во вкладке *Основное* выбранного репозитория.
- **1.7_x86-64** — кодовое имя дистрибутива, которое можно посмотреть в описании версии после загрузки ISO-образа.
- **main, contrib и non-free** — компоненты дистрибутива, которые можно посмотреть в описании версии после загрузки ISO-образа.

После добавления ссылки на корпоративный репозиторий обновите индекс пакетов и результат выполнения обновления кэша:

```
admin@pc-1:~$ sudo apt update
Игн:1 http://repo-1.ald.company.lan/repos/alse 1.7_x86-64 InRelease
Пол:2 http://repo-1.ald.company.lan/repos/alse 1.7_x86-64 Release [5 766 B]
Пол:3 http://repo-1.ald.company.lan/repos/alse 1.7_x86-64 Release.gpg [833B]
Сущ:4 http://dl.astralinux.ru/astra/frozen/1.7_x86-64/1.7.6/repository-base 1.7_x86-
↪64 InRelease
Пол:5 http://repo-1.ald.company.lan/repos/alse 1.7_x86-64/main amd64 Packages [1 310␣
↪kB]
Сущ:6 http://dl.astralinux.ru/astra/frozen/1.7_x86-64/1.7.6/repository-extended 1.7_
↪x86-64 InRelease
Пол:7 http://repo-1.ald.company.lan/repos/alse 1.7_x86-64/contrib amd64 Packages [2␣
↪155 B]
Пол:8 http://repo-1.ald.company.lan/repos/alse 1.7_x86-64/non-free amd64 Packages [55,
↪8 kB]
Сущ:9 https://dl.astralinux.ru/aldpro/frozen/01/2.4.0 1.7_x86-64 InRelease
Получено 1 374 kB за 0с (3 309 kB/s)
Чтение списков пакетов... Готово
Построение дерева зависимостей
Чтение информации о состоянии... Готово
```

Для дополнительной проверки вы можете отключить все публичные интернет-репозитории в файле `/etc/apt/sources.list`, обновить индекс пакетов еще раз, после чего убедиться в том, что установка утилиты `htop` проходит из корпоративного репозитория.

```
admin@pc-1:~$ sudo vi /etc/apt/sources.list
admin@pc-1:~$ sudo apt update
Игн:1 http://repo-1.ald.company.lan/repos/alse 1.7_x86-64 InRelease
Сущ:2 http://repo-1.ald.company.lan/repos/alse 1.7_x86-64 Release
Сущ:4 https://dl.astralinux.ru/aldpro/frozen/01/2.2.0 1.7_x86-64 InRelease
...
Чтение списков пакетов... Готово
Построение дерева зависимостей
Чтение информации о состоянии... Готово
```

(continues on next page)

(продолжение с предыдущей страницы)

```
Все пакеты имеют последние версии.

admin@pc-1:~$ sudo apt install htop
Чтение списков пакетов... Готово
Построение дерева зависимостей
Пол:1 http://repo-1.ald.company.lan/repos/alse 1.7_x86-64/main amd64 htop amd64 2.2.0-
↪1 [89,9 kB]
Получено 89,9 kB за 0с (0 B/s)
...
Распаковывается htop (2.2.0-1) ...
Настраивается пакет htop (2.2.0-1) ...
...
```

11.3.5 Создание репозитория вручную

Вы можете загружать вручную отдельные deb-пакеты сторонних разработчиков. Для примера создадим репозиторий с названием **yandex-browser**, в который загрузим последнюю версию браузера от компании Яндекс. На портале управления ALD Pro создайте новый репозиторий с названием «yandex-browser» и укажите абсолютный путь «/yandexbrowser», см. рис. 2.237

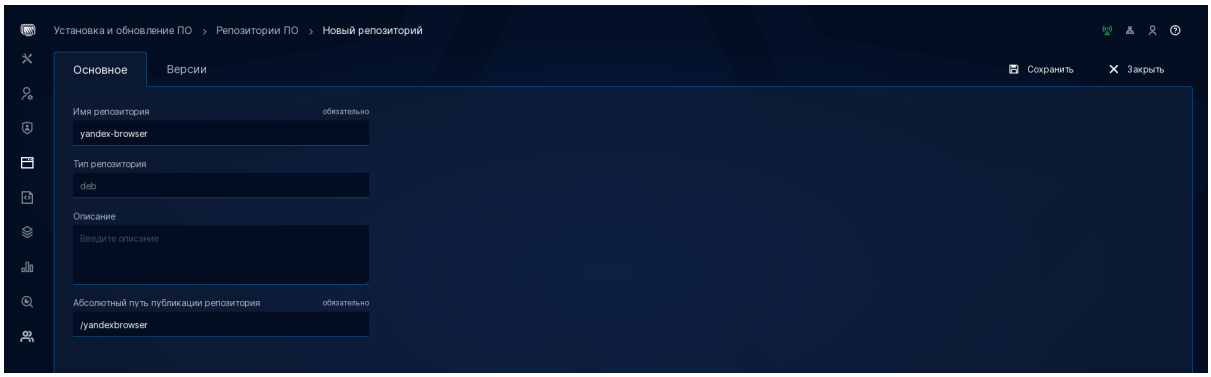


рис. 2.237 – Создание репозитория для Яндекс браузера

Создайте новую версию репозитория, см. рис. 2.238

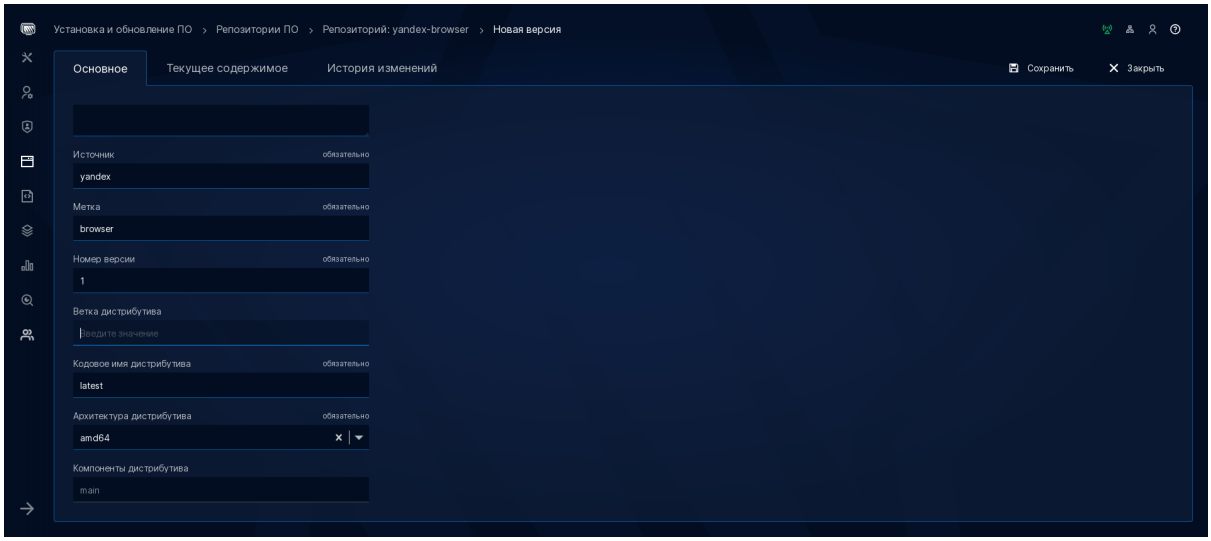


рис. 2.238 – Создание новой версии репозитория

Где:

- **Источник** = **yandex** — Справочное поле для описания источника пакета.
- **Метка** = **browser** — Справочное поле для описания вида программного пакета.
- **Номер версии** = **1** — Целое число для версионирования репозитория. При обновлении версии репозитория нужно использовать следующий порядковый номер.
- **Кодовое имя дистрибутива** = **latest** — Используется для возможности размещения в одном репозитории нескольких дистрибутивов, но подсистема ALD Pro позволяет работать только с одним дистрибутивом, поэтому не имеет практического значения, рекомендуется использовать значение latest.
- **Архитектура дистрибутива** = **amd64** - Используется для возможности размещения в одном репозитории пакетов для разных архитектур. Подсистема ALD Pro позволяет разместить только одну версию пакетов, поэтому не имеет практического значения, и можно использовать значение all.
- **Компоненты дистрибутива** = **main** — Используется для возможности распределения пакетов по категориям. В репозитории ALD Pro можно разместить пакеты только одной категории *main*, поле не редактируется.

Теперь вам станет доступна загрузка пакетов на вкладке *Текущее содержимое*. Скачайте наиболее свежую версию браузера следующими командами:

```
admin@dc-1:~$ cd ~/Загрузки
admin@dc-1:~/Загрузки$ wget https://download.yandex.ru/browser/astra-os/yandex-
↪browser.deb
```

(continues on next page)

(продолжение с предыдущей страницы)

```
admin@dc-1:~/Загрузки$ mv yandex-browser.deb "yandex-browser-$(dpkg -f yandex-browser.
↪deb Version).deb"
```

Загрузите пакет в репозиторий и обновите страницу, чтобы увидеть изменения, см. рис. 2.239. Пакет может не появиться в списке, если файл не будет принят утилитой reprepro.

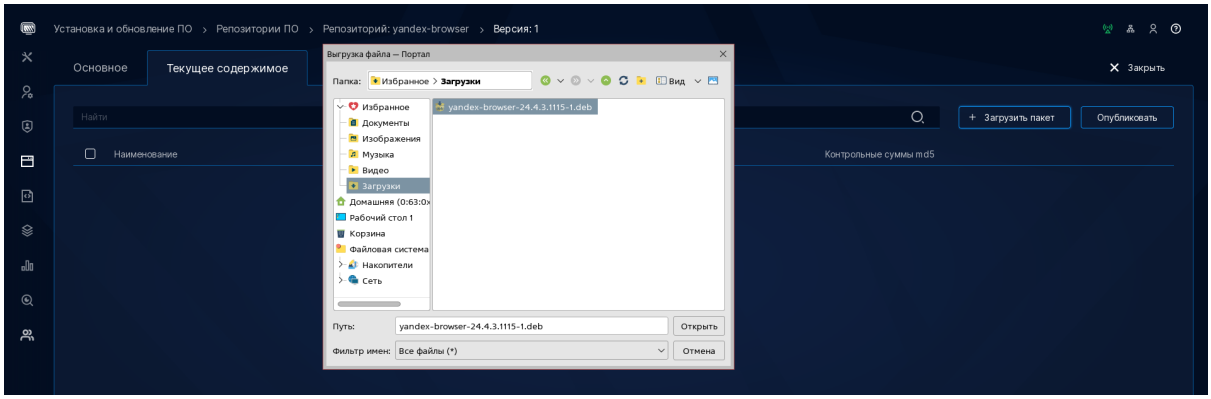


рис. 2.239 – Загрузка deb-пакета в репозиторий

Для того чтобы репозиторий стал доступен пользователям, его нужно опубликовать, нажав кнопку *Опубликовать* . Ссылку для подключения репозитория можно добавить командой:

```
echo "deb [trusted=yes] http://repo-1.ald.company.lan/repos/yandexbrowser/ latest main
↪" | sudo tee /etc/apt/sources.list.d/yandexbrowser.list
```

Если вы теперь выполните обновление индекса пакетов, то с помощью команды policy утилиты apt-cache сможете увидеть, что Яндекс браузер будет устанавливаться из *yandexbrowser* дистрибутива *latest*.

```
admin@pc-1:~$ sudo apt update
Игн:1 http://repo-1.ald.company.lan/repos/alse 1.7_x86-64 InRelease
Игн:2 http://repo-1.ald.company.lan/repos/yandexbrowser latest InRelease
Сущ:3 http://repo-1.ald.company.lan/repos/alse 1.7_x86-64 Release
Пол:4 http://repo-1.ald.company.lan/repos/yandexbrowser latest Release [865 B]
Игн:5 http://repo-1.ald.company.lan/repos/yandexbrowser latest Release.gpg
Пол:6 http://repo-1.ald.company.lan/repos/yandexbrowser latest/main amd64 Packages␣
↪[881 B]
Сущ:8 https://dl.astralinux.ru/aldpro/frozen/01/2.2.0 1.7_x86-64 InRelease
Получено 1 746 B за 0с (6 089 B/s)
Чтение списков пакетов... Готово
Построение дерева зависимостей
Чтение информации о состоянии... Готово
Все пакеты имеют последние версии.

admin@pc-1:~$ sudo apt-cache policy yandex-browser-stable
yandex-browser-stable:
   Установлен: (отсутствует)
   Кандидат: 24.4.3.1115-1
   Таблица версий:
```

(continues on next page)

(продолжение с предыдущей страницы)

```
     24.4.3.1115-1 500
        500 http://repo-1.ald.company.lan/repos/yandexbrowser latest/main amd64␣
↪Packages
```

11.3.6 Изменение опубликованной версии репозитория

После публикации репозитория вы не сможете внести никаких изменений в текущую версию, что довольно неудобно, т.к. при создании новой версии все пакеты придется загружать повторно. В будущих релизах продукта эта особенность будет изменена, но если вам прямо сейчас нужно сопровождать довольно большой корпоративный репозиторий, то можно воспользоваться следующим обходным решением.

Статус репозитория определяется значением столбца *repositoryversion_status* в таблице **repository_repositoryversion** базы *repo*. Если версия репозитория будет опубликована, то в этом столбце вы увидите значение **published**, а, чтобы снять репозиторий с публикации, вам достаточно вручную установить значение на **edit** и удалить ссылки из каталога `link_root_folder`. При повторной публикации версии репозитория ссылки будут созданы автоматически.

Следующей командой вы сможете изменить состояние версии репозитория «yandex-browser» с значения «published» на «edit»:

```
sudo -i -u postgres psql -d repo -c \
  "UPDATE repository_repositoryversion  \
  SET repositoryversion_status='edit' \
  WHERE repositoryversion_status='published' AND \
  repositoryversion_repository_id = \
  (SELECT repository_id FROM repository_repository WHERE \
  repository_name='yandex-browser');"
```

Результат выполнения команды:

```
UPDATE 1
```

Строка «UPDATE 1» свидетельствует о том, что изменение записи прошло успешно. Текущее состояние репозиториев вы можете получить следующей командой:

```
sudo -i -u postgres psql -x -d repo -c \
  "SELECT * FROM repository_repositoryversion \
  JOIN repository_repository ON \
  repositoryversion_repository_id = repository_id;"
```

Результат выполнения команды:

```
-[ RECORD 1 ]-------------------+-------------------------------------
repositoryversion_id            | 46304e0e-33c0-42c1-ba51-13f0445440fd
repositoryversion_version       | 1
```

(continues on next page)

(продолжение с предыдущей страницы)

```
repositoryversion_status        | published
repositoryversion_created_at    | 2024-07-17 05:33:04.868885+03
repositoryversion_source_type   | iso
repositoryversion_repository_id | 5e9c1aca-6655-4396-aea8-8baaeef1564c
repository_id                   | 5e9c1aca-6655-4396-aea8-8baaeef1564c
repository_name                 | alse
repository_type                 | deb
repository_source_type          | local
repository_created_at           | 2024-07-17 05:31:41.284323+03
repository_updated_at           | 2024-07-17 05:31:41.284342+03
repository_version              | 1
repository_description          | Пакеты операционной системы
repository_uri_relative_path    | /alse
repository_status               | ok
-[ RECORD 2 ]-------------------+-------------------------------------
repositoryversion_id            | 0bab4493-1beb-4a96-87bc-43b713b5e9a2
repositoryversion_version       | 1
repositoryversion_status        | edit
repositoryversion_created_at    | 2024-07-17 06:03:41.917093+03
repositoryversion_source_type   | file
repositoryversion_repository_id | 3db1adef-b447-49d7-a1e9-158f29730b98
repository_id                   | 3db1adef-b447-49d7-a1e9-158f29730b98
repository_name                 | yandex-browser
repository_type                 | deb
repository_source_type          | local
repository_created_at           | 2024-07-17 05:57:09.822516+03
repository_updated_at           | 2024-07-17 05:57:09.822532+03
repository_version              | 1
repository_description          |
repository_uri_relative_path    | /yandexbrowser
repository_status               | ok
...
```

После отмены публикации репозитория вам нужно также удалить ссылку в каталоге веб-сервера:

```
admin@repo-1:~$ sudo rm -rf /opt/rbta/aldpro/repo/storage/link_root_folder/
↪yandexbrowser
```

После выполнения указанных команд вы сможете продолжить редактирование репозитория через портал управления ALD Pro. По окончанию редактирования не забудьте снова опубликовать репозиторий, чтобы он стал доступен по протоколу HTTP(S) для рабочих станций.

11.3.7 Политики установки программного обеспечения

После публикации корпоративного репозитория вам становятся доступны политики установки программного обеспечения, которые работают через Salt. Давайте установим браузер Яндекс через групповые политики.

Для этого нам сначала нужно на странице *Установка и обновление ПО → Каталог ПО* создать раздел «Корпоративные приложения» и в нем новый набор программного обеспечения «Браузер

Яндекс», см. рис. 2.240

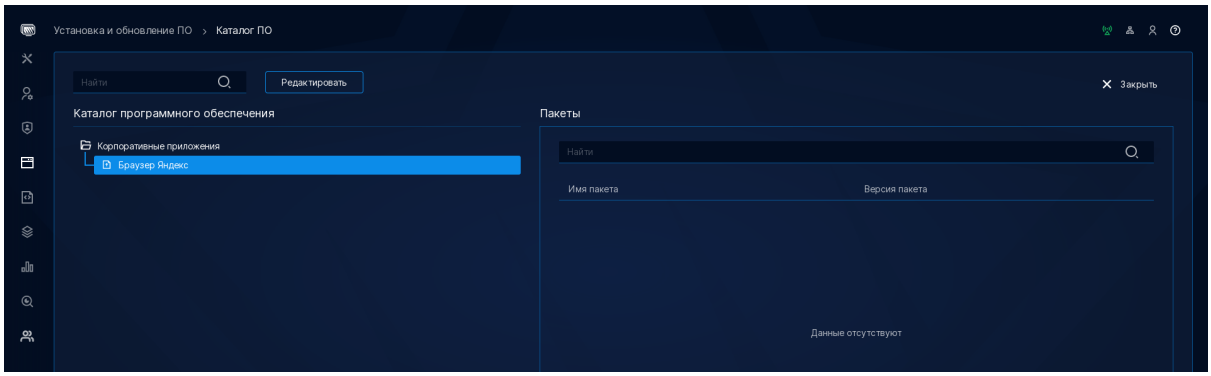


рис. 2.240 – Создание набора ПО

Откройте созданный набор и на вкладке *Пакеты* загрузите пакет браузера Яндекс. В списке будет ограниченное количество записей, поэтому вам нужно будет начать вводить название пакета «yandex. . . », чтобы сработала функция поиска и показала доступные значения, см. рис. 2.241.

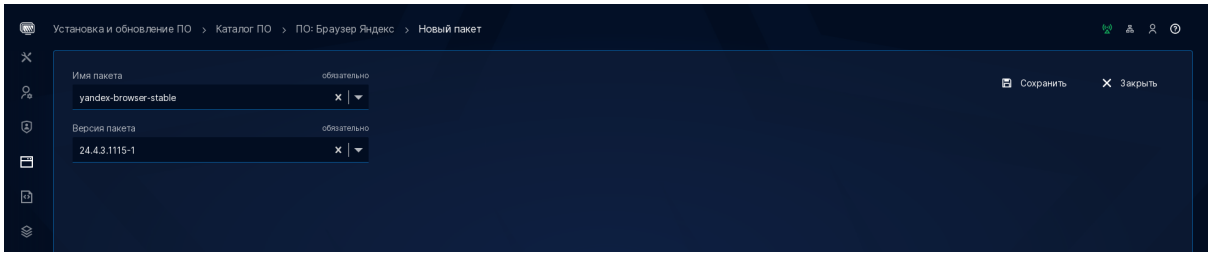


рис. 2.241 – Добавление пакета в набор ПО

Групповые политики установки программного обеспечения позволяют автоматически настраивать конфигурационные файлы этих приложений. Для этого в настройках набора программного обеспечения на вкладке *Параметры* нужно определить перечень допустимых атрибутов. Например, для Яндекс браузера мы можем установить домашнюю страницу параметром `{{ homepage_url }}` через политику HomepageLocation, см. рис. 2.242.

Примечание: Для русскоязычных доменов потребуется написать адрес закодированный в punycode, например: `моя-фирма.рф — xn----8sbxocjq4a1h.xn--p1ai`

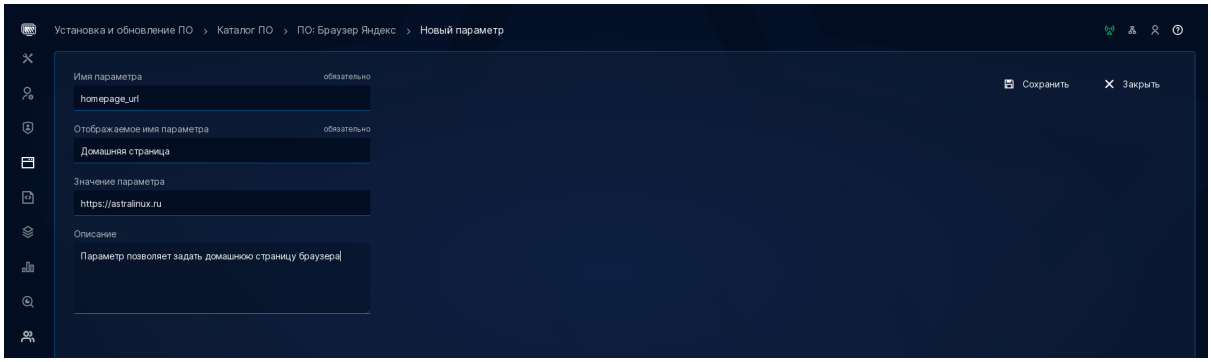


рис. 2.242 – Настройка параметра для набора ПО

Давайте создадим шаблон файла `/etc/opt/yandex/browser/policies/managed/managed_policies.json`.

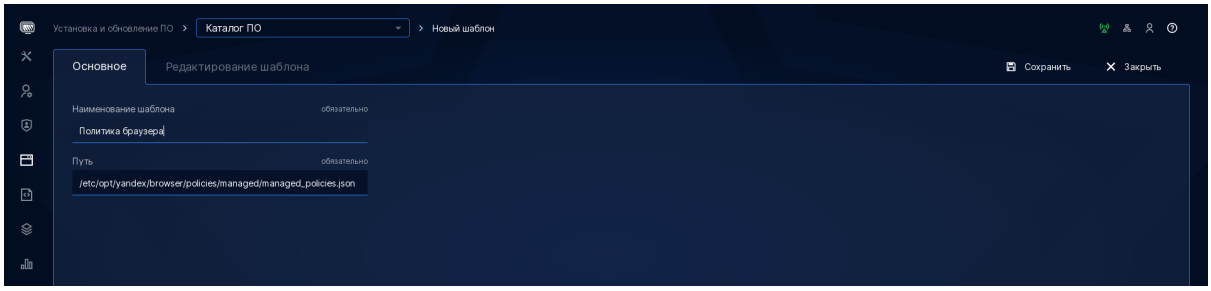


рис. 2.243 – Создание шаблона конфигурационного файла

На странице «Шаблоны конфигурации» нужно будет задать следующий salt-скрипт:

```
{%- set homepage_url = parameters.get('homepage_url', none) -%}
{
    "HomepageLocation": "{{ homepage_url }}"
}
```

Примечание: Обратите внимание, что при редактировании объекта групповой политики установки программного обеспечения вы сможете задать несколько наборов атрибутов, как будто эти параметры являются составными. В версии 2.0.0 так и было, поэтому для получения значения атрибута функцию нужно было вызывать так: `parameters[0].get(...)`. Начиная с версии 2.2.0, переменная parameters содержит один последний набор атрибутов, поэтому вы можете использовать метод `parameters.get` без уточнения индекса элемента.

Теперь вы сможете создать новый объект групповой политики на странице *Установка и обновление ПО → Политики ПО*, см. рис. 2.244

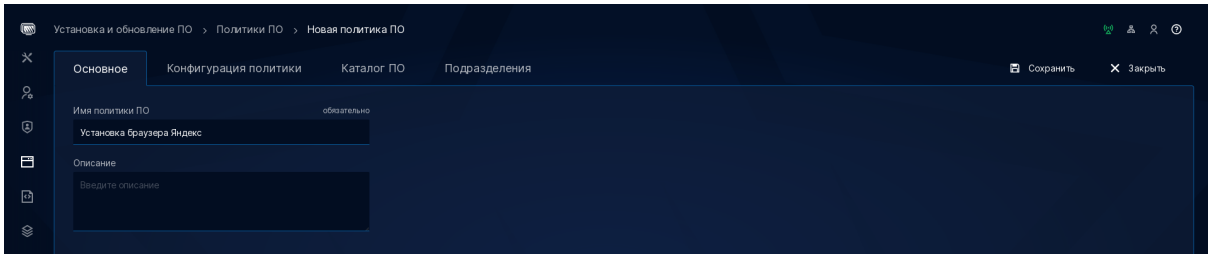


рис. 2.244 – Создание нового объекта групповой политики установки ПО

Чтобы добавить набор программного обеспечения в объект групповой политики, вам нужно на вкладке *Каталог ПО* выбрать его в дереве и нажать кнопку *Применить*. Удалить ранее добавленный набор вы сможете на вкладке *Конфигурация политики*.

Назначить объект групповой политики на компьютеры можно на вкладке *Подразделения*. Вы можете выбрать подразделение в целом или уточнить список компьютеров и групп компьютеров, которые будут попадать в область действия этого объекта групповой политики.

Осталось только обновить кэш и форсировать применение групповой политики установки ПО:

```
admin@pc-1:~$ sudo aldpro-gpupdate --swp
```

Не забудьте, что для установки свежей версии браузера вам нужно переключить систему на корпоративные репозитории **alse** и **yandex-browser**. Ранее в модуле про групповые политики мы уже приводили пример дополнительного параметра групповой политики «Источники программного обеспечения», с помощью которого вы можете централизованно настраивать источники программного обеспечения в домене.

11.3.8 Установка ПО через дополнительные параметры групповых политик

Политики установки программного обеспечения удобны тем, что позволяют установить приложения из deb-пакетов, загруженных в подсистему репозиториев, и настроить конфигурационные файлы. Однако бывает так, что требуется установить и настроить приложения, размещенные на внешних репозиториях. В этом случае вам помогут следующие два дополнительных параметра групповых политик, которые можно импортировать в систему.

Параметр „Источники программного обеспечения“ позволяет централизованно настраивать источники программного обеспечения для пакетных менеджеров apt и yum.

Файл import_parameter_common_source_list.sh

```
python3 policy.py parameter add \
--host \
--array \
--id 'rbta_ldap_custom_gp_host_repo_source_list' \
```

(continues on next page)

(продолжение с предыдущей страницы)

```
--name 'Источники программного обеспечения' \
--parent-folder 'Общие параметры' \
--description 'Параметр позволяет централизованно настраивать источники программного␣
↪обеспечения для пакетных менеджеров apt и yum, проверен для операционных систем␣
↪Astra Linux Special Edition, Ред ОС и Alt. В системах на базе APT параметры␣
↪подключения к репозиториям будут внесены в файл /etc/apt/sources.list.d/repo_source.
↪list, а в системах на базе yum - в файл /etc/yum.repos.d/repo_source.repo.

С помощью данного параметра вы можете определить репозитории для нескольких типов␣
↪операционных систем сразу и дополнительно сузить область применения параметра с␣
↪помощью фильтрации по требованиям к ОС.

В графическом интерфейсе Astra Linux и Alt настройка репозиториев может быть␣
↪выполнена с помощью приложений Synaptic, а в Ред ОС - с помощью Dnfdragora.

Параметр является составным и может определять несколько источников сразу. Все␣
↪указанные источники будут включены в итоговый файл за исключением полных дубликатов.

Атрибут "Тип ОС" указывает, к какой операционной системе относится конкретный␣
↪источник программного обеспечения. Значение атрибута представляет собой строку, и␣
↪может принимать следующие значения:
- Astra, Debian - для систем, использующих пакетный менеджер apt;
- Red OS, RedHat - для систем, использующих пакетный менеджер yum/dnf;
- ALT - для ALT Linux с учетом особенностей их пакетного менеджера.

Атрибут "Источник" позволяет задать строку, которая описывает внешний репозиторий␣
↪программного обеспечения.

Для операционной системы Astra Linux каждая строка переносится "как есть" и должна␣
↪иметь следующий вид:

deb <адрес_репозитория> <код_дистрибутива> <компонент1> <компонент2> <компонент3>

где:
- deb - обозначает репозиторий бинарных файлов с предварительно скомпилированными␣
↪пакетами, для репозиториев с исходными пакетами используется «deb-src»;
- адрес репозитория - URL или локальный путь к источнику пакетов (например, «http(s):/
↪/», «file:/» или «cdrom:[]/» при добавлении с диска);
- код дистрибутива - уточняет релиз или версию дистрибутива, поскольку один␣
↪репозиторий может содержать пакеты для нескольких версий;
- компоненты - логически объединённые группы пакетов, определённые условиями␣
↪использования.

Условия использования компонентов следующие:
- non-free - группа содержит пакеты, которые не соответствуют принципам свободного ПО,
↪ имеют патенты или другие юридические ограничения;
- contrib - группа содержит пакеты, которые сами по себе соответствуют принципам␣
↪свободного ПО, но зависят от пакетов из группы «non-free», т. е. не могут без них␣
↪работать;
- main - группа содержит пакеты свободного ПО, которые не зависят от пакетов из групп␣
↪«contrib» и «non-free».

Для операционной системы Ред ОС для каждой строки будет создана отдельная секция, в␣
↪которую будут перенесены указанные в строке параметры. Деление на подстроки␣
↪выполняется по символу вертикальной черты |. Например, строка, описывающая источник,
↪ может выглядеть следующим образом:
```

(continues on next page)

(продолжение с предыдущей страницы)

```
name=RedOS Cert - Base 7.3|baseurl=https://repo1.red-soft.ru/redos/7.3c/$basearch/
↪os|enabled=0|gpgcheck=1|gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-RED-SOFT

Для того, чтобы отключить источник, не удаляя соответствующую запись, для␣
↪операционных систем Astra Linux и Alt можно поставить символ решетки в начало␣
↪строки, а для Red OS установить параметр enable=0' \
--attr 'Тип ОС':os_type:'' \
--attr 'Источник':repo_source_item:'' \
--script 'import_parameter_common_source_list.sls' \
--script-comment 'Версия 2.0'
```

Файл import_parameter_common_source_list.sls

```
{% set id = 'rbta_ldap_custom_gp_host_repo_source_list' %}
{% set node = salt['grains.get']('nodename') %}
{% set gpo = salt['pillar.get']('aldpro-hosts:' + node + ':' + id, default=[]) %}

{% for item in gpo %}
{% do item.update({'os_type': item.get('os_type', '').lower() }) %}
{% endfor %}

{% set os_type = salt['grains.get']('os') | lower %}

{% set lines = gpo | selectattr('os_type', '==', os_type)
                   | selectattr('repo_source_item', 'string')
                   | map(attribute='repo_source_item')
                   | map('trim')
                   | select('!=', '')
                   | unique
                   | list
%}

{{ id }}:
   {%- if os_type in ['astra', 'debian', 'alt'] %}
      file.managed:
            - name: /etc/apt/sources.list.d/repo_source.list
            - contents: |
                  {%- for line in lines %}
                  {{ line }}
                  {%- endfor %}
   {%- elif os_type in ['red os', 'redhat'] %}
      file.managed:
            - name: /etc/yum.repos.d/repo_source.repo
            - contents: |
                  {%- for line in lines %}
                        {%- set sublines = line.split('|') %}
                        [customs_{{ loop.index }}]
                        {%- for subline in sublines %}
                        {{ subline }}
                        {%- endfor %}
                  {% endfor %}
   {% else %}
      test.fail_without_changes:
            - name: "ОС не поддерживается!"
```

(continues on next page)

(продолжение с предыдущей страницы)

```
            - failhard: False
   {% endif %}
```

Параметр „Установка пакетов“ позволяет централизованно устанавливать программного обеспечения, указывая просто наименование пакетов.

Файл import_parameter_common_packages.sh

```
python3 policy.py parameter add \
--host \
--array \
--id 'rbta_ldap_custom_gp_host_packages' \
--name 'Установка пакетов' \
--parent-folder 'Общие параметры' \
--description 'Позволяет централизованно управлять установкой программного␣
↪обеспечения с помощью пакетного менеджера.

Атрибут "Название пакета" определяет название пакета, который должен быть установлен/
↪удален.

Атрибут "Состояние" определяет, должен ли пакет быть установлен или удален.␣
↪Допустимые значения:
- installed - пакет должен быть установлен в системе (по умолчанию)
- removed - пакет должен быть удален' \
--attr 'Название пакета':pkg_name:'' \
--attr 'Состояние':pkg_state:'' \
--script 'import_parameter_common_packages.sls' \
--script-comment 'Версия 1.0'
```

Файл import_parameter_common_packages.sls

```
{% set id = 'rbta_ldap_custom_gp_host_packages' %}
{% set node = salt['grains.get']('nodename') %}
{% set gpo = salt['pillar.get']('aldpro-hosts:' + node + ':' + id) %}
{% if gpo %}

   {%- macro getvalid(value, available) -%}
      {{- value if value!='' and value.lower() in available|lower else available[0] -}
↪}
   {%- endmacro %}

   {%- for item in gpo %}
      {%- if item.pkg_name is not none %}

            {%- set pkg_name = item.pkg_name %}
            {%- set pkg_state = getvalid(item.pkg_state, ['installed', 'removed
↪'])|lower %}

{{ id }}_{{ pkg_name }}:
               {%- if pkg_state=='removed' %}
                  pkg.removed:
                        - name: {{ pkg_name }}
               {%- else %}
                  pkg.installed:
```

(continues on next page)

(продолжение с предыдущей страницы)

```
                        - name: {{ pkg_name }}
                        - refresh: True
                        - cache_valid_time: 300
                {%- endif %}

      {%- endif %}
   {%- endfor %}

{% endif %}
```

11.4 Подсистема «Служба печати»

В ходе реализации проектов импортозамещения на предприятиях крайне важно сохранить привычный уровень удобства в работе с наиболее востребованными сервисами, в том числе и такими как печать. Для этого в продукте ALD Pro реализована подсистема службы печати на базе программного обеспечения CUPS, которая позволяет обеспечить централизованный доступ к принтерам по протоколу IPP, что устраняет необходимость установки специализированных драйверов на рабочие станции.

11.4.1 Повышение роли сервера до подсистемы службы печати

Повысить роль сервера до подсистемы службы печати можно с портала управления ALD Pro на странице *Роли и службы сайта → Служба печати*, см. рис. 2.245. Сайт является обязательным полем, но пока еще не используется в системе. В будущем эта привязка позволит делегировать права доступа на управление подсистемами.

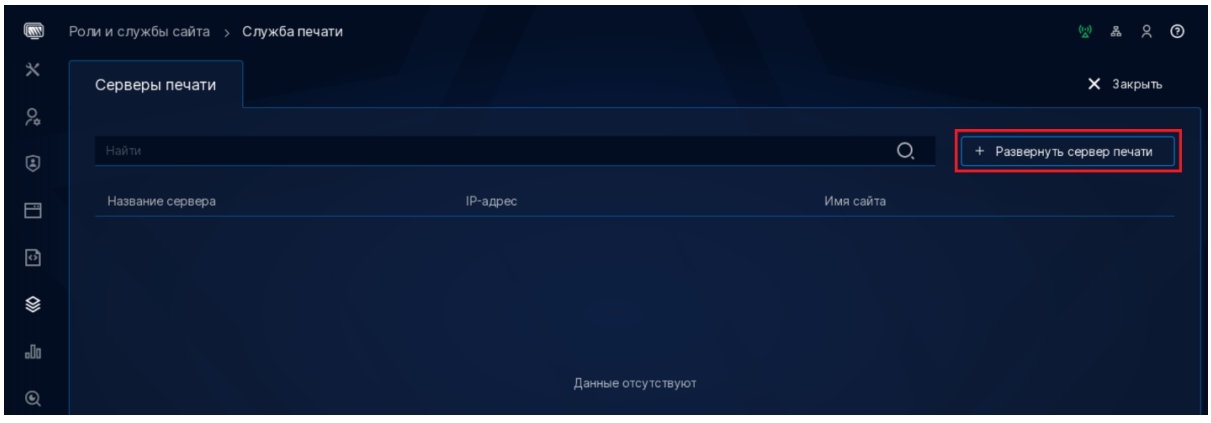


рис. 2.245 – Добавление сервера службы печати

Задание на установку подсистемы отправляется по цепочке Фронтенд > Бэкенд > LDAP > репликация > LDAP > По умолчанию сервер репозитория проверяет LDAP на предмет изменений каждые 30 минут, для ускорения этого процесса можно выполнить следующую команду:

```
admin@cups-1:~$ sudo aldpro-roles --iud
```

Отслеживать статус установки можно в карточке сервера репозитория ПО по пути *Роли и службы сайта → Служба печати → Сервер: cups-1.ald.company.lan*. За установку подсистемы отвечает скрипт install.sls из каталога /srv/salt/standalone/roots/states/aldpro/subsystems/cups.

11.4.2 Архитектура подсистемы службы печати

Мы уже рассматривали подробно устройство службы CUPS, см. *Модуль 16. Работа с подсистемой печати*, поэтому отметим только, что в случае подсистемы ALD Pro доступ к службе с рабочих мест сотрудников осуществляется по протоколу IPP, порт 631/TCP или 443/TCP (в случае SSL), см. рис. 2.246. Чисто технически CUPS поддерживает также подключение по протоколам LPD, Samba и NetTalk, но в ALD Pro мы используем только неавторизованный доступ через IPP.

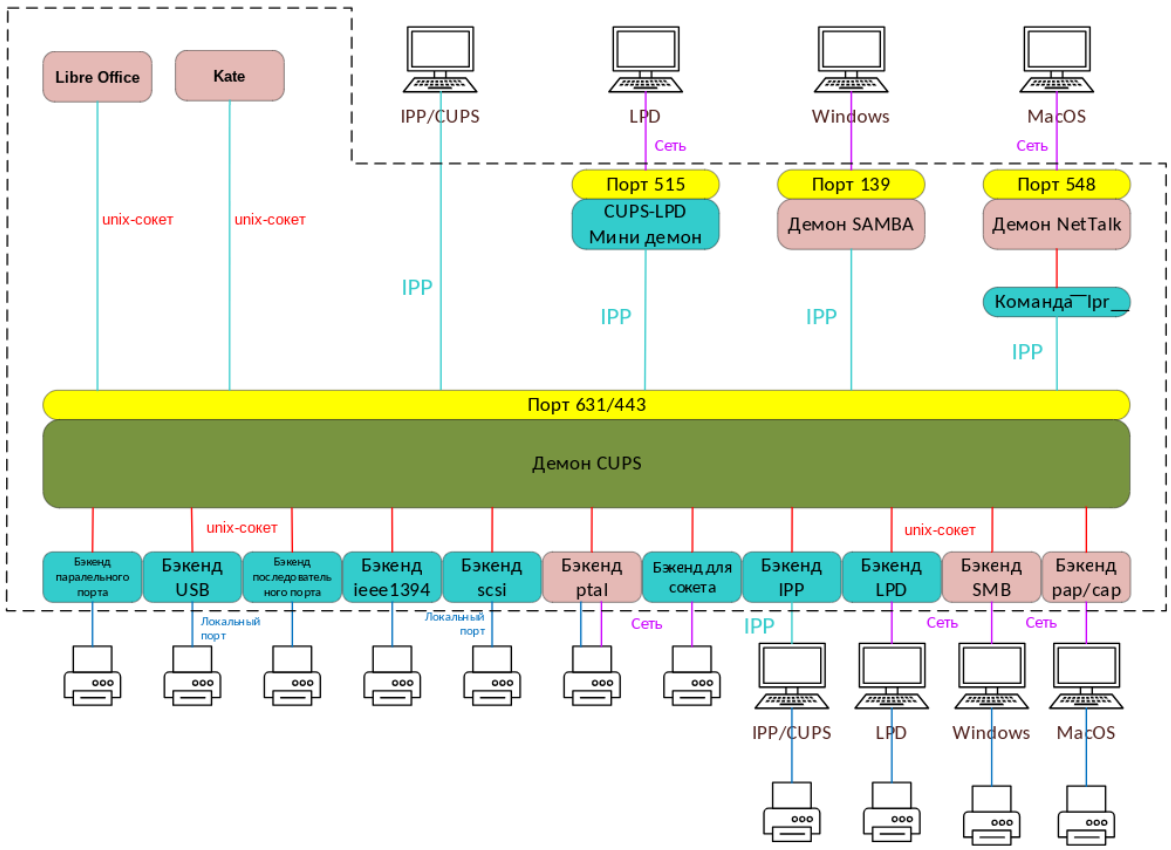


рис. 2.246 – Архитектура подсистемы службы печати

Администрировать службу CUPS в операционной системе Astra Linux могут участники локальной группы lpadmin с идентификатором 113, что задается параметром «Require user @SYSTEM» в разделе <Location /admin> конфигурационного файла /etc/cups/cupsd.conf.

```
admin@cups-1:~$ cat /etc/cups/cupsd.conf
. . .
<Location /admin>
   AuthType Default
   Order allow,deny
   Allow all
   Require user @SYSTEM
</Location>
. . .
```

Значение *@SYSTEM* говорит, что необходимо смотреть значение параметра «SystemGroup» в файле `/etc/cups/cups-files.conf`, где мы как раз и увидим нашу группу *lpadmin*.

```
cat /etc/cups/cups-files.conf
. . .
# Administrator user group, used to match @SYSTEM in cupsd.conf policy rules...
# This cannot contain the Group value for security reasons...
SystemGroup lpadmin
. .
```

Чтобы предоставить пользователю admin возможность администрировать принтеры, необходимо добавить его в группу lpadmin. Это можно сделать с помощью следующей команды:

```
admin@cups-1:~$ sudo usermod -aG lpadmin admin
```

После добавления в группу станет доступна панель администрирования CUPS по адресу **https://localhost:631/admin**. Панель запросит логин и пароль пользователя.

11.4.3 Добавление нового принтера

Чтобы добавить новый принтер вы должны подключить его USB-кабелем к серверу, на котором установлена подсистема службы печати, и установить необходимые драйверы. Вам в первую очередь нужно добиться возможности печати пробной страницы с этого сервера.

Большинство современных принтеров и МФУ имеют сетевой порт или позволяют подключить принтер к компьютерной сети через Wi-Fi. В случае сетевого доступа в настройках принтера рекомендуется установить ограничение по IP-адресу, чтобы сотрудники не пытались подключать принтер напрямую.

Добавить новый принтер можно через портал управления в разделе *Роли и службы сайта* → *Служба печати*. Для этого откройте страницу соответствующего сервера и на вкладке *Список принтеров* нажмите кнопку *+ Новый принтер*, см. рис. 2.247.

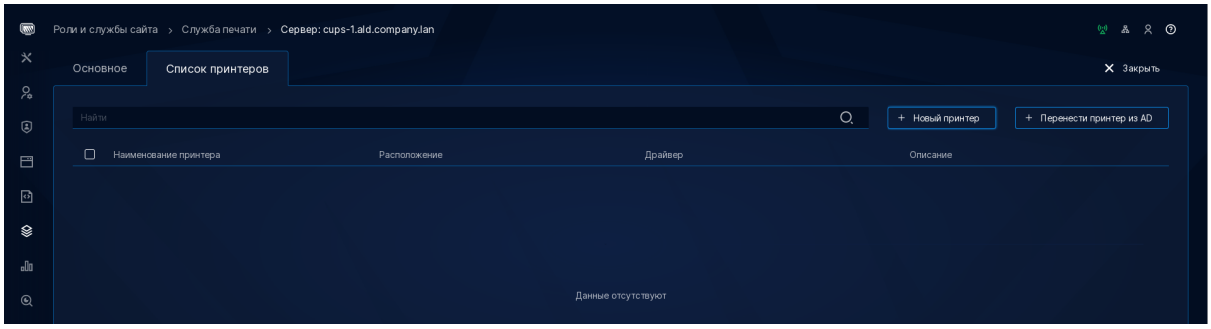


рис. 2.247 – Добавление нового принтера

В карточке нового принтера следует заполнить следующие поля, см. рис. 2.248.

- **Подключение** — укажите адрес физического принтера, например:
  - для PDF принтера: `cups-pdf:/`;
  - для USB принтера: `usb://<производитель>/<имя_устройства>`, например для Samsung ML-2850: `usb://Samsung/ML-2850%20Series`;
  - для сетевого принтера: `ipp://<ip_адрес>`.

В нашем случае в качестве примера рассмотрим установку PDF принтера на сервер отличный от cups-1 с использованием протокола ipp. Для этого потребуется установить пакет **printer-driver-cups-pdf** на целевой хост с помощью команды:

```
admin@printer-1:~$ sudo apt install printer-driver-cups-pdf
```

После установки пакета в поле подключение напишите: **ipp://printer-1.ald.company.lan/printers/PDF**.

- **Наименование принтера** — задайте имя принтера, под которым он также будет отображаться на пользовательском компьютере (на латинице и без пробелов), например: **PDF**.
- **Расположение** — укажите физическое месторасположение принтера, например, **кабинет 101, этаж 1**.
- **Драйвер** — укажите путь к файлу `*.ppd` на вашем компьютере с драйвером устройства печати, например, файл драйвера PDF принтера `/usr/share/ppd/cups-pdf/CUPS-PDF_noopt.ppd`. Драйвер будет передан на сервер службы печати для возможности работы сервера с устройством. А также драйвера принтеров можно поискать на сайте OpenPrinting .
- **Описание** — в этом поле вы можете указать дополнительную информацию об устройстве, например, модель принтера.

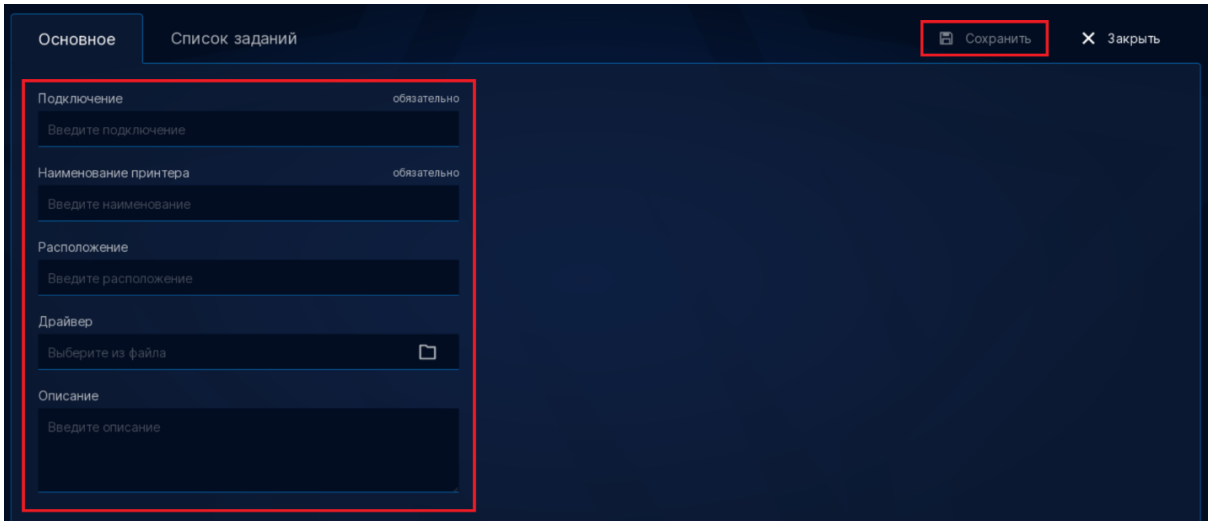


рис. 2.248 – Карточка нового принтера

Примечание: Если в системе настроена миграция из домена MS Active Directory, то система позволит вам перенести основную информацию о принтере, но лучше сделать все вручную, чтобы сразу настроить драйвер принтера.

На вкладке Список заданий вы сможете увидеть только активные задания на печать, а успешно завершенные задания отображаться не будут. По наличию записей в этом журнале можно понять, что с принтером что-то не так, например, он неисправен или недоступен.

Для проверки функционала воспользуемся виртуальным принтером «PDF», который был создан ранее после установки службы печати на сервере cups-1.ald.company.lan и установки пакета printer-driver-cups-pdf на сервер printer-1.ald.company.lan. Перед использованием принтера необходимо выполнить следующие действия:

1. Добавление принтера в систему

После создания принтера в LDAP создаётся задание на его добавление в систему (выполнение заданий происходит раз в 30 минут). Для ускорения процесса выполните команду:

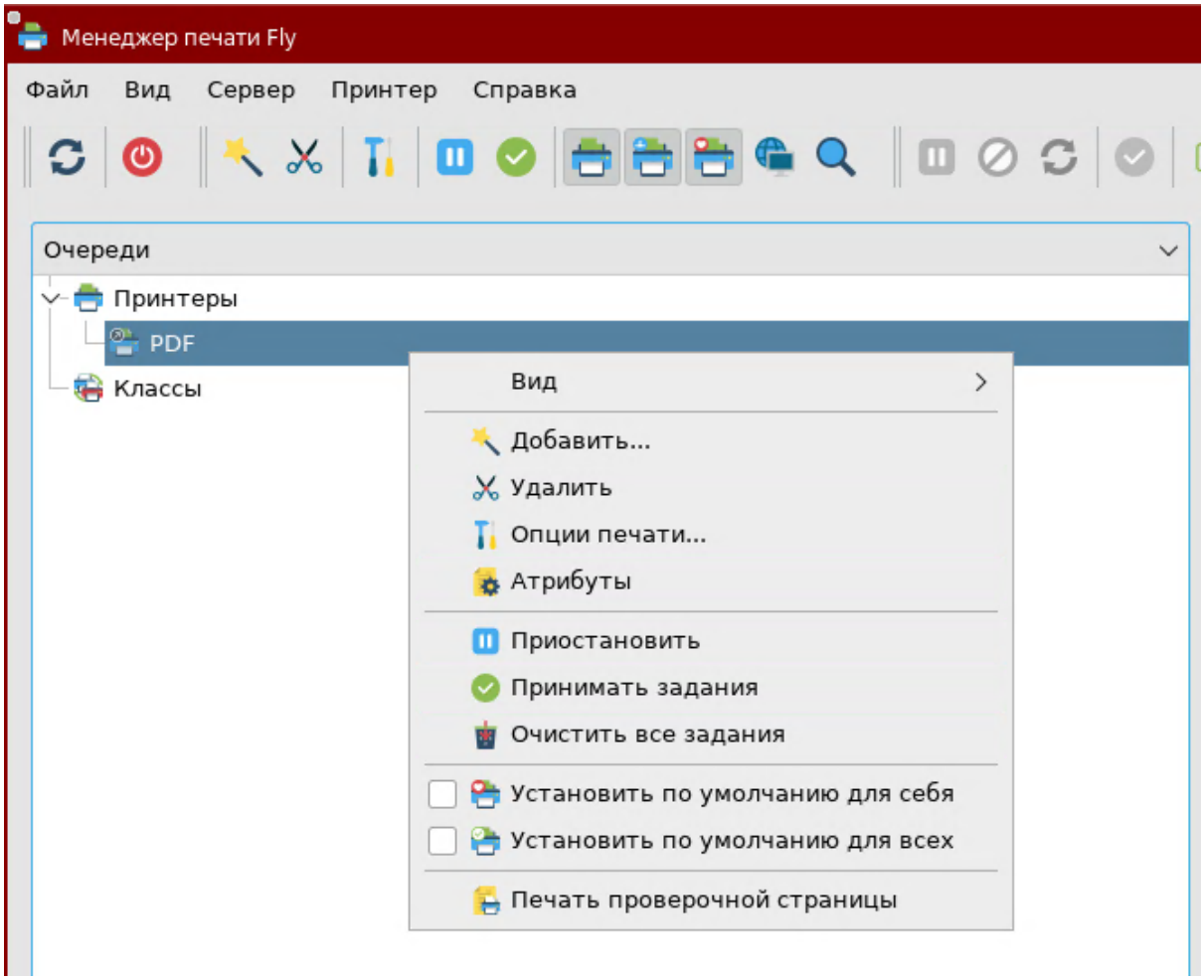
```
admin@cups-1:~$ sudo aldpro-roles --subsystem_settings cups
```

2. Активация принтера

Пользователем с правами администратора открыть оснастку *Панель управления → Оборудование → Принтеры* или выполнить команду

```
admin@cups-1:~$ sudo fly-admin-printer
```

В оснастке раскрыть поле Принтеры, выбрать созданный нами принтер и активировать его нажав кнопку Принимать задания.



3. Настройка общего доступа

На сервере printer-1.ald.company.lan с установленным PDF-принтером необходимо *Разрешить общий доступ*. Выполните команды:

```
admin@printer-1:~$ sudo lpadmin -p PDF -o printer-is-shared=true
admin@printer-1:~$ sudo cupsctl --share-printers
```

После выполнения этих действий принтер PDF появится в списке принтеров подсистемы службы печати на портале управления ALD Pro. Файлы отправленные на печать пользователем admin будут сохраняться на сервере **printer-1.ald.company.lan** по пути /home/admin/PDF/.

11.4.4 Установка принтера с помощью групповой политики

Чтобы централизованно добавить принтер на рабочие станции сотрудников, воспользуйтесь параметром групповой политики *Установить принтеры* из раздела «Оборудование». Для этого создайте объект групповой политики «Принтеры ИТ» на странице *Групповые политики* → *Групповые политики* и включите в нем требуемый параметр, см. рис. 2.249.

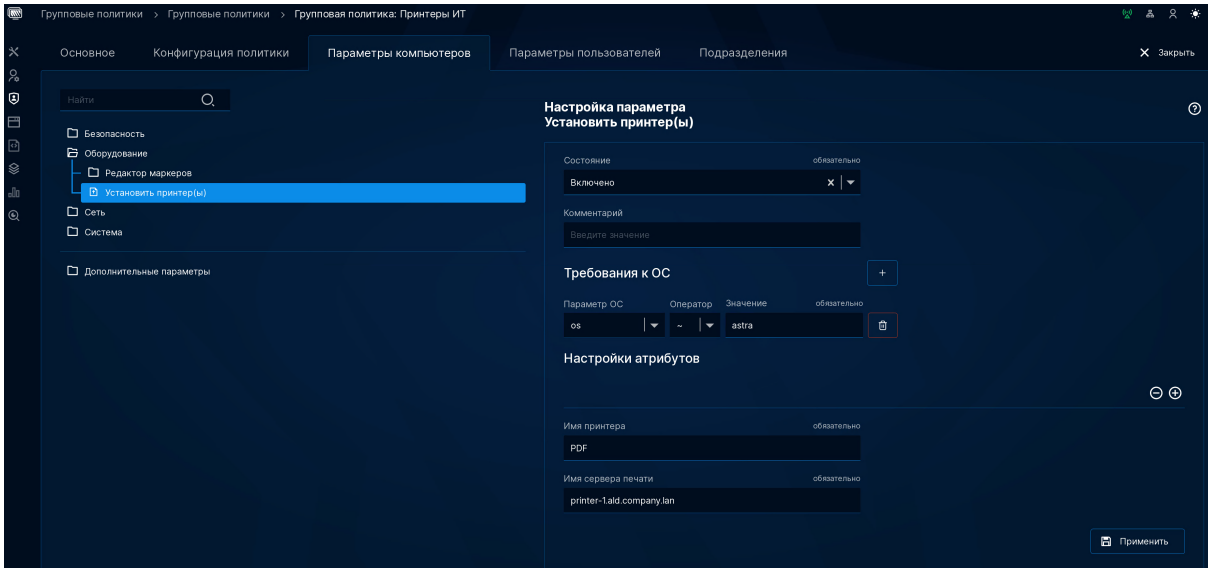


рис. 2.249 – Настройка параметра групповой политики для добавления принтера

Не забудьте назначить объект групповой политики на требуемое подразделение. Чтобы форсировать применение политики, выполните команды:

```
Total run time:   2.942 s
admin@pc-1:~$ sudo aldpro-gpupdate --gp
. . .
[INFO    ] Loading fresh modules for state activity
[INFO    ] Completed state [cups] at time 16:09:57.526321 (duration_in_ms=31798.943)
[INFO    ] Running state [lpadmin -p PDF -E -v ipp://printer-1.ald.company.lan/
↪printers/PDF -m raw] at time 16:09:57.534416
[INFO    ] Executing state cmd.run for [lpadmin -p PDF -E -v ipp://printer-1.ald.
↪company.lan/printers/PDF -m raw]
[INFO    ] Executing command 'lpadmin' in directory '/root'
[INFO    ] {'pid': 65757, 'retcode': 0, 'stdout': '', 'stderr': "lpadmin: Raw queues␣
↪are deprecated and will stop working in a future version of CUPS.\nlpadmin: Use the
↪'everywhere' model for shared printers."}
[INFO    ] Completed state [lpadmin -p PDF -E -v ipp://printer-1.ald.company.lan/
↪printers/PDF -m raw] at time 16:09:57.648331 (duration_in_ms=113.914)
. . .
```

После применения групповой политики в Менеджере печати на *pc-1* вы увидите, что появился общий принтер *PDF* (ipp://printer-1.ald.company.lan/printers/PDF), см. рис. 2.250.

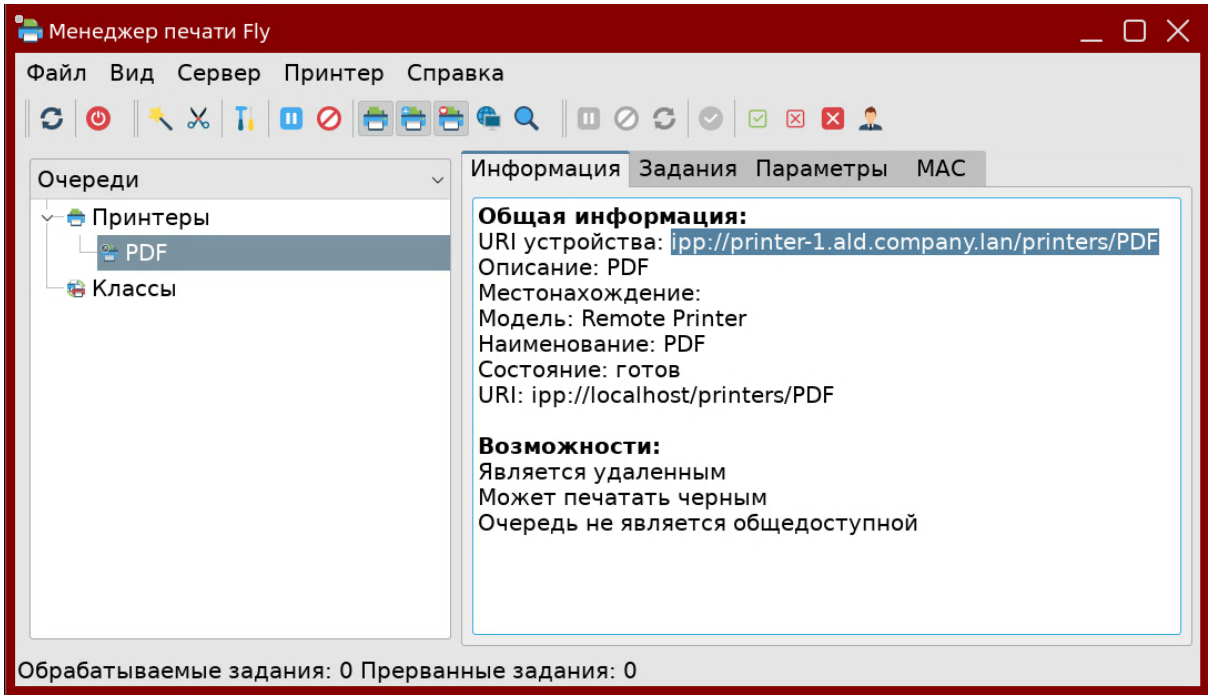


рис. 2.250 – Общий принтер, установленный через групповую политику

В окне «Печать» текстовых редакторов, например, LibreOffice, кроме принтера PDF вам будет доступен также принтер «PDF_cups_1», и это тот же самый принтер, но его настройки система получила автоматически через широковещательный запрос к серверу CUPS-1, так как по умолчанию включена служба avahi, а в параметрах сервера включена опция «Browsing».

По умолчанию печать на автоматически найденный принтер работать не будет, поэтому в конфигурационном файле /etc/cups/printers.conf рекомендуется установить опцию «Browsing No» и перезапустить службу:

```
admin@cups-1:~$ cat /etc/cups/printers.conf
...
<DefaultPrinter PDF>
Browsing No
. . .
</DefaultPrinter>
...
```

Примечание: Если вам потребуется задействовать автоматическое обнаружение принтеров, то вместо того, чтобы отключить Browsing, установите параметр «DNSSDHostName cups-1.ald.company.lan», и тогда клиенты начнут обращаться к принтеру по правильному имени вместо cups-1.local. Только не забывайте, что широковещательные запросы не смогут выйти за пределы L2 сети.

11.5 Подсистема «Мониторинг»

Для отслеживания состояния серверной группировки в продукте ALD Pro предназначена отдельная подсистема мониторинга на базе Zabbix с использованием инструментов построения графического представления данных Grafana.

Повышение роли сервера до подсистемы мониторинга выполняется на вкладке *Серверы мониторинга* страницы *Мониторинг → Журнал событий мониторинга*. Нажмите кнопку *+ Развернуть сервер мониторинга*, см. рис. 2.251.

Задание на установку подсистемы отправляется по цепочке Фронтенд > Бэкенд > LDAP > репликация > LDAP > По умолчанию сервер репозитория проверяет LDAP на предмет изменений каждые 30 минут, для ускорения этого процесса можно выполнить следующую команду:

```
admin@repo-1:~$ sudo aldpro-roles --iud
```

Отслеживать статус установки можно в карточке сервера репозитория ПО по пути *Мониторинг → Журнал событий мониторинга → Серверы мониторинга → Сервер: monitoring.ald.company.lan*. За установку подсистемы отвечает скрипт install.sls из каталога /srv/salt/standalone/roots/states/aldpro/subsystems/zabbix.

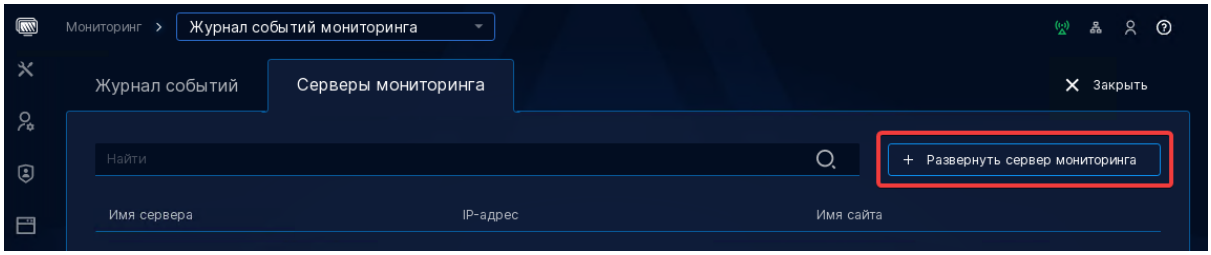


рис. 2.251 – Добавление сервера мониторинга

Подсистема мониторинга состоит из двух основных компонентов, см. рис. 2.252:

- **Zabbix** — свободная система мониторинга статусов разнообразных сервисов компьютерной сети, серверов и сетевого оборудования, написанная Алексеем Владышевым. Для хранения данных используется MySQL, PostgreSQL, SQLite или Oracle Database, веб-интерфейс написан на PHP.

- **Grafana** — свободная программная система визуализации данных, ориентированная на данные систем ИТ-мониторинга. Реализована как веб-приложение в стиле «приборных панелей» с диаграммами, графиками, таблицами, предупреждениями. Часто используется совместно с СУБД или системами мониторинга.

  Система подключается к многообразным источникам данных, поддерживает расширение с помощью системы плагинов. Позволяет конечным пользователям строить сложные панели

мониторинга с помощью интерактивных запросов.

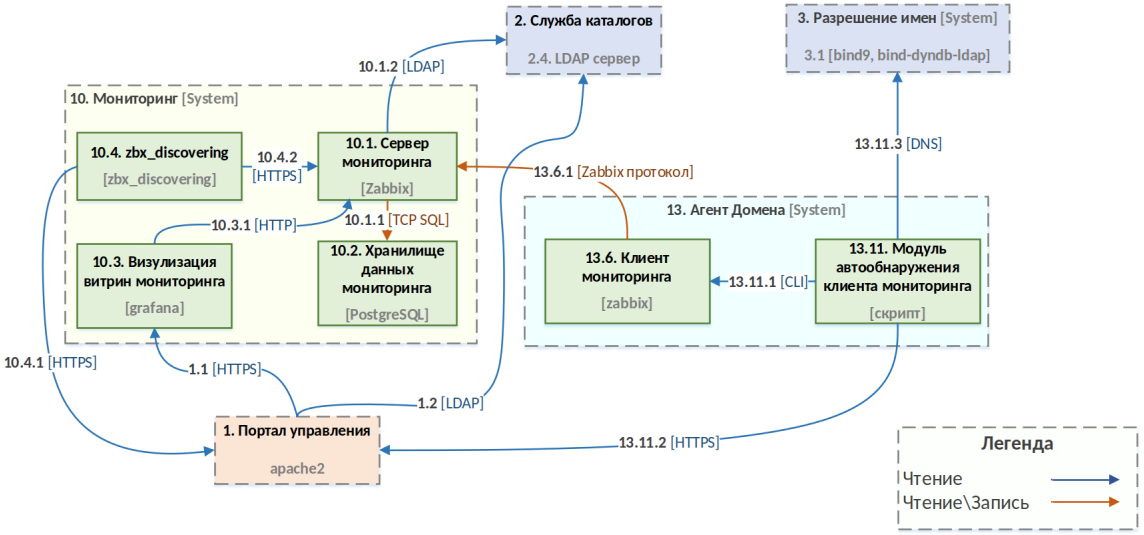


рис. 2.252 – Архитектура подсистемы «Мониторинг»

В таблице ниже достаточно подробно объясняется, по каким протоколам взаимодействуют компоненты системы между собой.

табл. 2.17 – Описание взаимодействия компонентов подсистемы «Мониторинг»

Номер	Инициатор	Адресат	Протокол	Передаваемая информация
13.11.1.	Агент домена Модуль автообнаружения	Агент домена - Клиент мониторинга	CLI	Модуль автообнаружения настраивает клиент мониторинга на адреса серверов мониторинга
13.11.2.	Агент домена - Модуль автообнаружения	Портал управления	HTTPS	Получает список серверов мониторинга
13.11.3.	Агент домена - Модуль автообнаружения	Разрешение имен	DNS	Чтение DNS записей с DNS сервера
13.6.1.	Агент домена - Клиент мониторинга	Мониторинг - Сервер мониторинга	Zabbix	Клиент мониторинга передает метрики рабочей станции на сервера мониторинга
10.1.1.	Мониторинг - Сервер мониторинга	Мониторинг - Хранилище данных мониторинга	TCP SQL	Запись информации о собранных метриках
10.1.2.	Мониторинг - Сервер мониторинга	Служба каталогов - LDAP сервер	LDAP	Получает список машин и развернутых на них подсистем
10.3.1.	Мониторинг - Визуализация витрин мониторинга	Мониторинг - Сервер мониторинга	HTTP	Получает метрики для отображения на витринах мониторинга
10.4.1.	Мониторинг - zbx_discovering	Портал управления	HTTPS	Получает список машин и развернутых на них подсистем
10.4.2.	Мониторинг - zbx_discovering	Портал управления	HTTPS	Передает список машин и развернутых на них подсистем
1.1	Портал управления	Мониторинг - Визуализация витрин мониторинга	HTTPS	Чтение дашбордов витрин мониторинга для визуализации собранных метрик в интерфейсе портала управления
1.2.	Портал управления	Служба каталогов - LDAP-сервер	LDAP	Происходит авторизация портала управления на сервере мониторинга через LDAP-сервер

После установки подсистемы информация о состоянии серверной группировки начинает отобра-

жаться на странице *Управление доменом → Общая информация*. Дополнительную информацию о состоянии системы вы сможете найти, например, на витринах мониторинга, см. рис. 2.253.

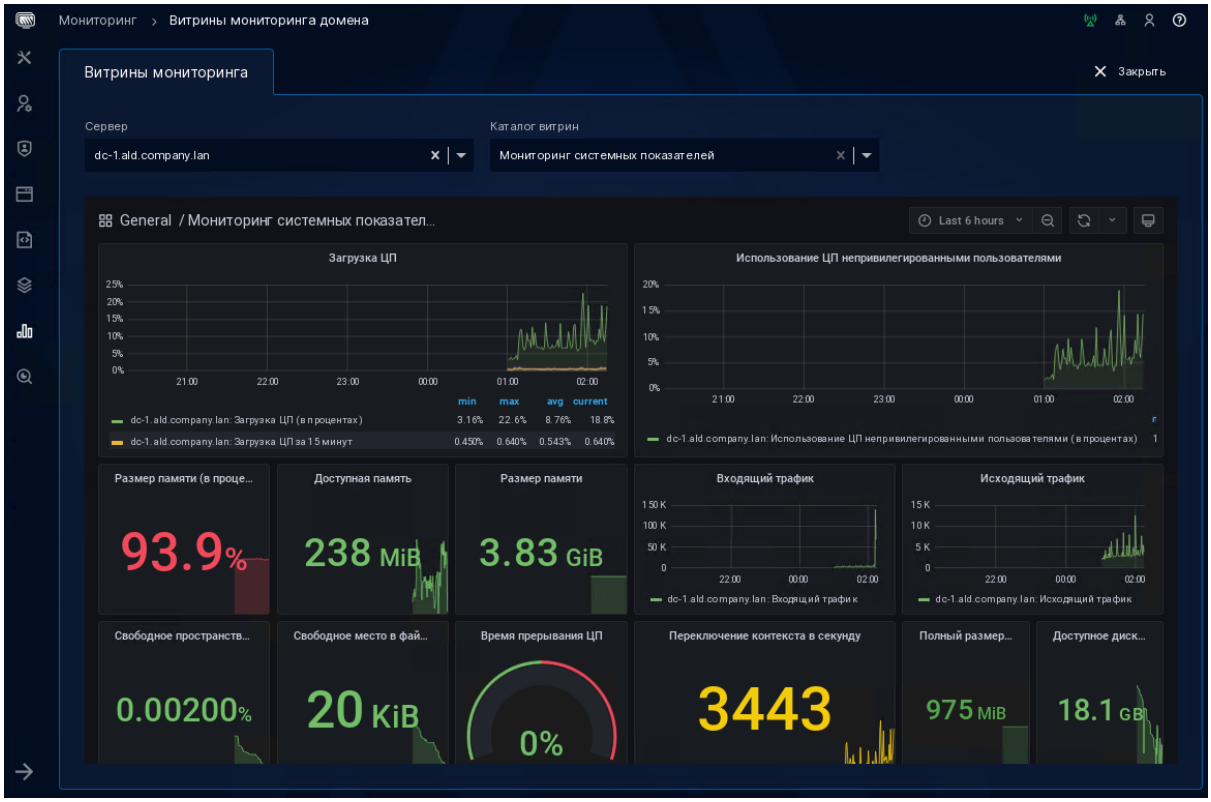


рис. 2.253 – Витрина мониторинга

Подключаясь к административным порталам Zabbix и Grafana, вы можете, например, добавлять новые параметры и изменять существующие витрины.

11.6 Подсистема «Аудит»

Подсистема аудита (журнала событий) предназначена для сбора событий с серверов и рабочих станций. Начиная с версии 2.0.0, подсистема реализована на базе продукта Syslog-ng.

Серверов аудита в домене может быть несколько. Повышение роли сервера до подсистемы аудита выполняется на странице *Журнал событий → Серверы журнала событий*, см. рис. 2.254.

Задание на установку подсистемы отправляется по цепочке Фронтенд > Бэкенд > LDAP > репликация > LDAP > По умолчанию сервер репозитория проверяет LDAP на предмет изменений каждые 30 минут, для ускорения этого процесса можно выполнить следующую команду:

```
admin@repo-1:~$ sudo aldpro-roles --iud
```

Отслеживать статус установки можно в карточке сервера репозитория ПО по пути *Журнал событий → Серверы журнала событий → Сервер:*

audit-1.ald.company.lan . За установку подсистемы отвечает скрипт install.sls из каталога /srv/salt/standalone/roots/states/aldpro/subsystems/audit .

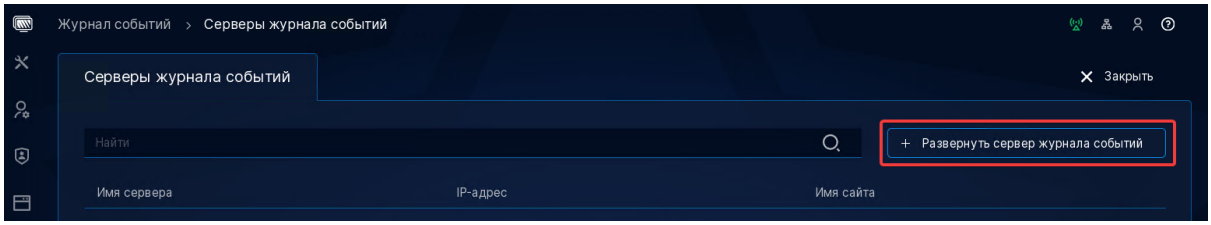


рис. 2.254 – Добавление сервера аудита

Для сбора событий с серверов и рабочих станций требуется настроить правила аудита на странице *Журнал событий → Настройка сбора журналов событий* , см. рис. 2.255.

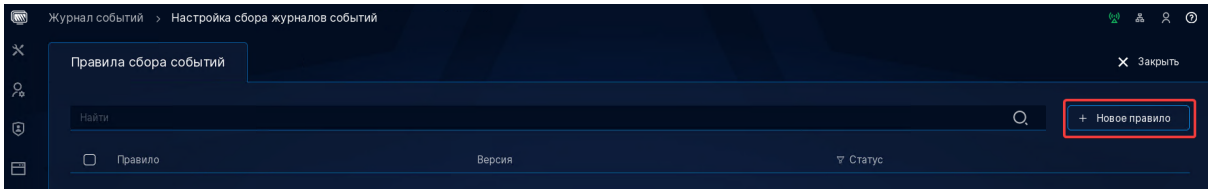


рис. 2.255 – Добавление правила аудита

В карточке нового правила нужно настроить следующие поля, см. рис. 2.256:

- **Имя правила** – введите любое значение для идентификации правила.
- **Тип логов** – выберите один из трех доступных типов событий:
  - **Логи авторизации Fly** — информация о входе в графический интерфейс системы.
  - **Логи удаленного подключения** — информация о подключениях по SSH.
  - **Логи состояния подключения к сети** — информация о подключениях к сети из журналов NetworkManager.
- **Сервер сбора логов** – укажите один из доступных серверов аудита, на который нужно отправлять события.
- **Описание** – дополнительная информация о правиле для удобства администрирования.
- **Статус** – позволяет отключить правило без его удаления.

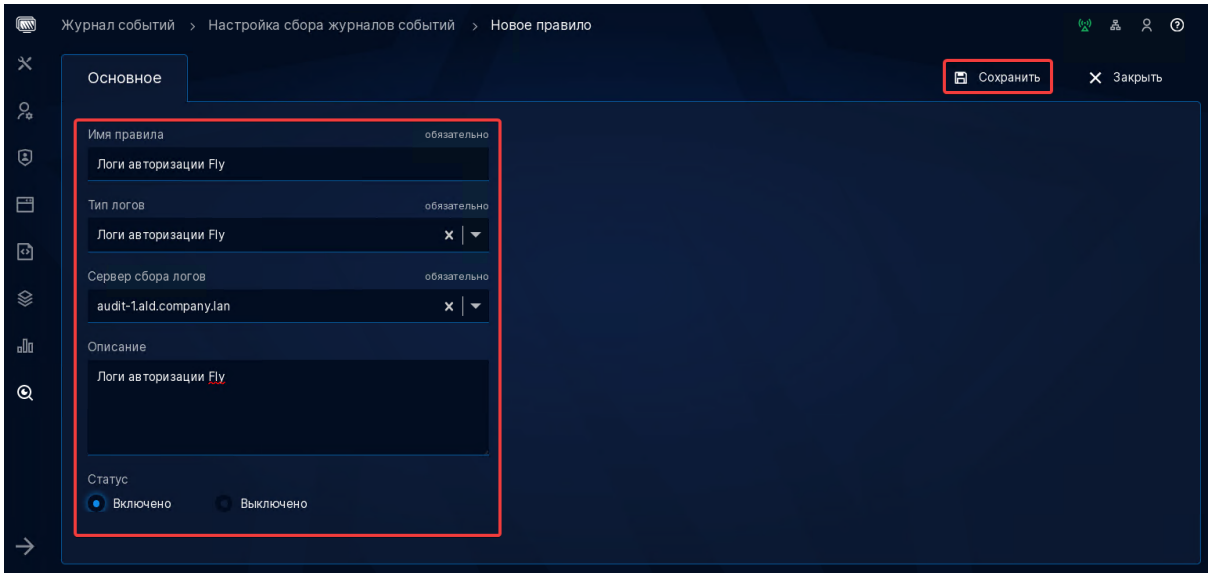


рис. 2.256 – Настройка правила аудита

Для принудительного обновления правил аудита на *pc-1* используйте команды:

```
admin@pc-1:~$ sudo aldpro-gpupdate --audit
```

Как вы можете заметить, перечень событий крайне небольшой, поэтому для интеграции с SIEM-системой MaxPatrol был разработан дополнительный параметр групповой политики, с помощью которого можно кастомизировать настройки syslog-ng на клиентах.

11.7 Интеграция с приложениями сторонних разработчиков

Для того чтобы извлечь максимальную выгоду от внедрения службы каталога, вам нужно интегрировать с ней как можно больше корпоративных приложений. Это позволит сократить количество паролей, которые нужно запоминать пользователям, а, может быть, даже упростит администрирование прав доступа.

Разработчики приложений должны позаботиться о возможности интеграции своих продуктов со службой каталога заранее, а в идеале предоставить подробные инструкции о том, как выполнить эту настройку. Однако иногда вы будете сталкиваться с тем, что не будет ни инструкций, ни достоверных фактов о возможности такой интеграции, поэтому крайне полезно понимать общие принципы:

- Во-первых, интеграция со службой каталога нужна как минимум **на уровне аутентификации**. Служба каталога поддерживает Kerberos, LDAP и NTLM, а если ее интегрировать с Keycloak, то станут доступны также протоколы OAuth2, OpenID и SAML.

  Многие приложения поддерживают аутентификацию через LDAP, но, как мы уже знаем, это не самый удачный способ, т.к. в этом случае через информационную систему будут проходить

учетные данные пользователей в открытом виде, и администрировать ее должны администраторы домена.

Наиболее удачным видится использование Kerberos-аутентификации, т.к. в этом случае серверное приложение получает только билет, зашифрованный ее собственным паролем. Это позволяет делегировать администрирование системы независимой команде системных администраторов, исключая возможность разглашения учетных данных сотрудников.

Некоторые устаревшие приложения будут поддерживать NTLM-аутентификацию. В этом случае через систему пароли не будут проходить в открытом виде, но данный протокол все-таки уже устарел и существенно уступает по безопасности даже LDAP, т.к. более подвержен MITM-атакам.

Протоколы OAuth2, OpenID и SAML сопоставимы по уровню безопасности Kerberos, но предназначены в первую очередь для аутентификации web-приложений.

- Во-вторых, приложения могут быть интегрированы **на уровне авторизации**, т.е. использовать информацию об участии пользователей в группах.

  Данная информация обычно извлекается из каталога по протоколу LDAP и опирается на значение атрибутов *memberof*. Наиболее сложный аспект этой интеграции заключается в сопоставлении субъектов безопасности. Приложения могут использовать полное отличительное имя (DN), логин (uid, sAMAccountName, kerberos principal name), идентификатор безопасности (SID, uidNumber, gidNumber).

  Служба каталога FreeIPA использует схему данных, которая отличается от схемы Active Directory, но для совместимости у пользователей и групп есть идентификаторы SID в стиле Windows, которые вы можете использовать в целях интеграции.

  Приложения, которые максимально интегрированы с Windows, извлекают авторизационную информацию об участии пользователя в группах в виде набора идентификаторов безопасности SID непосредственно из сертификата атрибута привилегий (PAC), который находится внутри Kerberos-билета.

  В службе каталога FreeIPA билеты пользователей содержат MS-PAC, и файловый сервер Samba, например, может получать авторизационную информацию из билетов. Такой способ позволяет немного снизить нагрузку на контроллеры домена, но в части безопасности он ничем не лучше LDAP.

  Протокол OAuth2 предоставляет механизм для делегирования доступа, позволяя приложениям действовать от имени пользователя без получения его учетных данных (обеспечивает получение согласия пользователя, выдачу токенов доступа, определение области доступа и обновление токенов). Однако OAuth2 сам по себе не выполняет проверку логина/пароля

(аутентификацию) и не определяет права доступа на основе участия в группах (авторизацию).

Протокол SAML позволяет передавать информацию как для проверки логина/пароля (аутентификация), так и для проверки ACL на основе групп (авторизация).

Протокол OpenID предназначен для аутентификации, то есть проверки логина и пароля. LDAP может использоваться как хранилище информации о пользователях, включая их учетные данные и принадлежность к группам. В сочетании с OpenID протокол LDAP может участвовать в процессах аутентификации (предоставляя данные для проверки логина/пароля) и авторизации (предоставляя информацию о членстве в группах для проверки ACL). Логика авторизации реализуется на уровне приложения.

Большинство клиент-серверных приложений ограничиваются только функциями аутентификации и авторизации, но есть и более сложные приложения. Например, IDM-системы, которые взаимодействуют с каталогом через активные коннекторы и могут создавать, изменять и блокировать учетные записи пользователей, реализуя сложный жизненный цикл с элементами документооборота. Для получения информации о возможностях интеграции обращайтесь в первую очередь к рабочей документации по соответствующему продукту.

Команда ALD Pro регулярно выпускает буклет интеграций в котором вы найдете все актуальные интреграции с нашим продуктом и узнаете о новых возможностях, которые они предоставляют.

11.8 Практика и тестирование

11.8.1 Практическая работа: Модуль 11. Дополнительные подсистемы и интеграции

См. также *Требования, правила и цели выполнения практической работы*

11.8.1.1 Практические задания

Задание 1.

1. Установите подсистему репозитория.
2. Создайте репозиторий и загрузите в него пакеты Astra Linux 1.7.6 из ISO-образа.
3. Создайте репозиторий и загрузите в него пакет браузера Яндекс вручную.
4. Установите на **pc-1** браузер Яндекс через групповую политику с настройкой домашней страницы.

Задание 2.

1. Установите подсистему службы печати.
2. Сделайте доступным виртуальный PDF-принтер.
3. Добавьте принтер на рабочую станцию через групповую политику.

Задание 3.

1. Установите подсистему мониторинга.
2. Изучите витрину мониторинга «Мониторинг подсистемы “Службы каталогов”» на **dc-1**.
3. Остановите службу apache2 на резервном контроллере домена **dc-2**.
4. Изучите витрину мониторинга «Мониторинг системных показателей» на **dc-1**.
5. Изучите вкладку «Журнал событий», дождитесь события чрезвычайной важности с **dc-2**.
6. Изучите витрину мониторинга «Мониторинг подсистемы Службы каталогов» на **dc-2**.
7. Изучите витрину мониторинга «Мониторинг системных показателей» на **dc-2**.
8. Запустите службу apache2 на резервном контроллере домена **dc-2**.
9. Изучите вкладку «Журнал событий», дождитесь, пока не пропадут события чрезвычайной важности с **dc-2**.

Задание 4.

1. Установите подсистему журналирования.
2. Создайте правило с типом «Логи авторизации Fly».
3. Создайте правило с типом «Логи состояния подключения к сети».
4. Принудительно примените новое правило на **pc-1**, используя команду.
5. Изучите файлы в /etc/syslog-ng/aldpro.
6. Отключите и включите поддержку сети в апплете «Network Manager» на **pc-1**.
7. Проверьте по логам на audit-1, что правило применяется корректно.

Задание 5. (опционально)

Выполните интеграцию службы каталога ALD Pro с приложениями сторонних разработчиков по инструкциям:

- Инструкция «Интеграция 1С:Предприятие со службой каталога ALD Pro»;
- Инструкция «Интеграция Bitrix со службой каталога ALD Pro»;
- Инструкция «Интеграция OpenVPN со службой каталога ALD Pro».

11.8.2 Тест: Модуль 11. Дополнительные подсистемы и интеграции

Пожалуйста, ответьте на несколько вопросов, чтобы закрепить полученные знания. Правильные ответы для самопроверки вы сможете найти в конце списка. Обратите внимание, что у некоторых вопросов может быть несколько верных вариантов ответов.

11.8.2.1 Вопросы

Вопрос 1. Какого вида бывают репозитории ПО в ALD Pro?

a. Репозиторий из ISO-образа

b. Репозиторий из deb-пакета

c. Репозиторий из tar-архива

d. Репозиторий из общей папки

Вопрос 2. В какой директории хранятся файлы репозиториев ПО в ALD Pro?

a. Директория `/opt/rbta/aldpro/repo/storage/`

b. Директория `/opt/rbta/aldpro/reprepro/repo/storage/`

c. Директория `/opt/rbta/aldpro/reprepro/storage/`

d. Директория `/opt/rbta/aldpro/reprepro/repo`

Вопрос 3. Что такое «Витрина мониторинга»?

a. Список доступных вариантов мониторинга на сервере Zabbix

b. Виджеты мониторинга в портале управления

c. Показатели производительности в интерфейсе Grafana

d. Список доступных серверов мониторинга в портале управления

Вопрос 4. Какие типы логов могут собираться подсистемой журналирования?

a. Логи подключения внешних устройств

b. Логи удаленного подключения

c. Логи состояния служб

d. Логи авторизации Fly

e. Логи состояния подключения к сети

f. Логи аутентификации Kerberos

Вопрос 5. Если учесть, что правила аудита настраиваются через механизм групповых политик, то через какой максимальный промежуток времени после создания правила соответствующие события начнут перенаправляться на сервер журналирования?

a. 30 минут

b. 50 минут

c. 60 минут

d. 80 минут

Ответы на вопросы

1. a, b
2. a
3. b
4. b, d, e
5. d

11.9 Заключение

Как вы могли заметить, служба каталога — это крайне непростая система, состоящая из множества компонентов, использующих разные технологии и протоколы, но мы постарались охватить все основные моменты, чтобы вы чувствовали себя уверенно при администрировании домена на предприятиях.

Мы уверены, что информация, представленная в этих учебных модулях, заложит надежный фундамент для вашего дальнейшего саморазвития. Но если вы столкнетесь с какой-то непонятной проблемой, обязательно обращайтесь в нашу техническую поддержку и сообщество ALD Профессионалов.



11.10 Дополнительные источники информации

- Debian-Paketmanagement, Axel Beckert, Frank Hofmann (немецкий)
- Настройка политик Яндекс Браузера для организаций
- Алфавитный указатель политик Яндекс Браузера
- Драйверы принтеров OpenPrinting
- Инструкция «Интеграция 1С:Предприятие со службой каталога ALD Pro»
- Инструкция «Интеграция Bitrix со службой каталога ALD Pro»
- Инструкция «Интеграция OpenVPN со службой каталога ALD Pro»

- Буклет интеграций ALD Pro

11.11 Обратная связь

Если остались вопросы, то их всегда можно задать в специальной теме.

Правила межсетевого экрана

В инфраструктуре крупных предприятий компьютерную сеть обычно нарезают на множество сегментов, взаимодействие между которыми ограничивают правилами межсетевого экрана для повышения безопасности в соответствии с концепцией нулевого доверия (Zero Trust). Чтобы правильно настроить сетевое оборудование, воспользуйтесь перечнем правил, который опубликован на вебстранице.

Каждая строка таблицы описывает параметры сетевого взаимодействия между двумя компонентами системы, например:

- **Нужно разрешать**: да
- **В обе стороны**: да
- **Источник**
  - **Подсистема**: контроллер домена
  - **IP-адрес**: IP адрес контроллера домена
- **Приемник**
  - **Подсистема**: контроллер доверенного домена AD DS
  - **Служба**: DNS
  - **IP-адрес**: IP адрес контроллера доверенного домена AD DS
  - **Порт**: 53
- **Протокол L3**: TCP/UDP
- **Протокол L4**: DNS
- **Подсистемы**: контроллер домена, контроллер доверенного домена AD DS
- **Назначение трафика**: Перенаправление DNS запросов для получения информации из дове-

ренной зоны. Большинство запросов будет обрабатываться с использованием протокола UDP, а переключение на TCP происходит, если ответы сервера будут превышать 512 байт на одно сообщение. По этому же порту происходит динамическое обновление DNS-записей со стороны клиентов по протоколу GSS-TSIG.

Дополнительные комментарии по параметрам:

- **Нужно разрешать** — указывает, нужно ли разрешать сетевое взаимодействие, описываемое в рамках правила, для нормальной работы продукта. Значение «нет» устанавливается в одном из следующих случаев:

  - Порт открывается одним из сопутствующих приложений системы, т. е. его функции не используются в продукте (например, Avahi). Учет таких правил помогает упростить проверку со стороны сотрудников информационной безопасности.

  - Порт используется для сетевого взаимодействия между компонентами, которые расположены в пределах одного хоста (например, PostgreSQL). Учет таких правил упрощает проверку и помогает лучше понять сетевое взаимодействие между компонентами системы.

  - Правило устарело и более не требуется (например, правила для GlusterFS). Удержание таких правил в таблице упрощает ретроспективный анализ правил, созданных ранее на сетевом оборудовании.

- **В обе стороны** — указывает, что источник и приемник должны взаимодействовать по указанному порту как в прямом так и обратном направлении, что позволяет исключить дублирование правил, например, при взаимодействии контроллеров ALD Pro с контроллерами AD DS. Может принимать значения «да/нет».

- **Источник** — группа колонок, описывающих компонент системы, который выступает инициатором сетевого взаимодействия.

- **Приемник** — группа колонок, описывающих компонент системы, который выступает принимающей стороной, о есть открывает серверный порт и ждет входящего сетевого соединения от Источника.

- **Подсистема** — наименование собственного компонента продукта или компонента стороннего вендора, который используется в гибридных сценариях развертывания. Например, контроллер домена, контроллер доверенного домена AD DS, доменный компьютер, компьютер доверенного домена AD DS, сервер репозитория и т.д.

  В предпоследней колонке «Подсистемы» указаны обе подсистемы, как со стороны Источника, так и Приемника, чтобы упростить поиск правил с помощью автоматических фильтров.

- **Служба** — наименование службы на принимающей стороне, которая непосредственно открывает порт и ожидает входящего сетевого соединения. Например, для контроллера домена это может быть 389 Directory Server (ns-slapd), KDC (krb5kdc), Bind9 (named-pkcs11), Apache (apache2) и т. д.

- **IP-адрес** — описание конкретного адреса или целого сегмента сети, в котором может находиться Источник или Приемник.

- **Порт** — номер сетевого порта, который прослушивает принимающая сторона.

- **Протокол L3** — наименование протокола транспортного уровня, который используется в рамках соединения, например, TCP или UDP.

- **Протокол L4** — протокол уровня приложений, используемый в рамках соединени. Например, HTTP, LDAP, Kerberos, DNS, DHCP, TFTP и т. п.

- **Назначение трафика** — описание функционального назначения трафика, какая информация передается между участниками сетевого взаимодействия, и для чего это нужно. Информация помогает в принятии решения об открытии конкретных портов и отладки возникающих проблем.

Известные ограничения

Краткое содержание

- *Оптимизация для больших инфраструктур*
- *Политика AllowNBToInternet (KB5028166)*
- *Работа Samba RPC на серверах ALSE 1.7.6.UU2*
- *Проверка целостности глобального каталога*

В этом разделе представлено описание известных ограничений и способов их решения.

Оптимизация для больших инфраструктур

При входе Windows-пользователя в систему компьютера ALSE из домена ALD Pro служба SSSD не только проверяет аутентичность пользователя с помощью Kerberos-билетов, но и запрашивает у своего контроллера домена авторизационную информацию об этом пользователе через расширенную операцию LDAP, что может приводить к появлению задержек в больших инфраструктурах, если пользователь является участником больших групп.

Для устранения данной проблемы на всех контроллерах домена ALD Pro (FreeIPA) следует настраивать опции ignore_group_members и subdomain_inherit следующим образом:

```
[domain/ald.company.lan]
ignore_group_members = true
subdomain_inherit = ignore_group_members
...
```

После изменения данного параметра следует очистить кэш SSSD на сервере командами:

```
sudo systemctl stop sssd
sudo sss_cache -E
```

(continues on next page)

(продолжение с предыдущей страницы)

```
sudo rm -rf /var/lib/sss/db/*
sudo rm -rf /var/lib/sss/mc/*
sudo systemctl start sssd
```

Политика AllowNBToInternet (KB5028166)

После установки обновления KB5028166 на Windows Server начинает зависать Kerberos-аутентификация на контроллерах домена и файловых серверах ALD Pro.

Причина заключается в том, что при установке обновления в ключе реестра HKEY_LOCAL_MACHINESYSTEMCurrentControlSetServicesNetBTParameters параметру AllowNBToInternet устанавливается значение 0, что блокирует запросы к контроллерам домена по протоколу NetBIOS из других сетей и препятствует возможности обнаружения активных контроллеров MS AD со стороны службы winbind на контроллерах домена и файловых серверах ALD Pro.

На контроллерах домена и файловых серверах ALD Pro библиотека Kerberos для обнаружения активного контроллера обращается сначала к плагину службы winbind, поэтому зависание этой службы приводит к тому, что на серверах перестает работать доменная аутентификация. Выключение службы winbind восстанавливает работу атуентификации, т.к. библиотека Kerberos в этом случае переключается на файл kdcinfo от службы sssd, но не решает проблему, т.к. эта служба необходима для нормальной работы системы в других сценариях.

Устранить проблему можно одним из следующих способов:

- Использовать продукт ALD Pro на серверах ALSE 1.7.6+ или установить патч, который можно получить через техническую поддержку.
- Включить параметр AllowNBToInternet=1 на контроллерах домена MS AD.
- Разместить контроллеры домена и файловые сервера ALD Pro в одной сети с контроллерами MS AD.

Работа Samba RPC на серверах ALSE 1.7.6.UU2

При установке продукта на ALSE 1.7.6 UU2 или обновлении операционной системы до этой версии на стороне MS AD замедляется преобразование идентификаторов безопасности в имена пользователей ALD Pro и пропадает возможность монтирования сетевых ресурсов.

Причина заключается в том, что в этой версии на стороне контроллеров домена ALD Pro не рабо-

тает RPC API, поэтому Windows Server множество раз получает отказ, прежде чем переключается на резервный механизм по протоколу SMB.

Устранить проблему можно установкой патча, который можно получить через техническую поддержку.

Проверка целостности глобального каталога

В версии 2.4.0 появился инструмент для проверки целостности глобального каталога, но в алгоритме проверки есть дефект, который может приводить к тому, что инспектор aldpro-gc-inspector будет сообщать о нарушении целостности даже в нормальном состоянии. Дефект будет устранен в последующих версиях продукта.

Часто задаваемые вопросы (FAQ)

Что такое программный комплекс ALD Pro?

Программный комплекс ALD Pro (Astra Linux Directory Professional) - это набор сетевых служб сервера Astra Linux для создания службы каталога и организации централизованного управления ИТ-инфраструктурой, что крайне важно для снижения дрейфа настроек и уменьшения расходов на техническое сопровождение при переходе на отечественные операционные системы. Продукт построен на хорошо известных компонентах с открытым исходным кодом, которые используют только открытые протоколы для обмена информацией.

Чем помогает продукт ALD Pro на предприятиях?

Продукт ALD Pro решает следующие основные задачи:

- **Централизованное управление учётными записями**. Хранение и управление учётными записями пользователей, компьютерами и сервисами в одном месте и, как следствие, снижение расходов на администрирование.
- **Единая точка сквозной аутентификации Kerberos V5**. Освобождает пользователей от необходимости повторного ввода пароля и его хранения в открытом виде в диспетчерах паролей.
- **Делегирование полномочий**. Реализована возможность распределения зон ответственности между администраторами.
- **Групповые политики**. Централизованное управления параметрами компьютеров и пользователей при помощи групповых политик на базе SaltStack.
- **Автоматическое развертывание сервисов**. Возможно выполнять автоматическое развёртывание следующих сервисов: репозитории deb-пакетов, файловый сервер SMB, север печати IPP, DHCP, установка ОС по сети, мониторинг.

Какие ключевые преимущества у продукта ALD Pro?

Преимущества перед решением Microsoft Active Directory:

- Исключает санкционные риски и возможность вендор-атак.
- Продукт включен в реестр отечественного ПО.
- У продукта выше совместимость с отечественными ОС на базе ядра Linux.

Преимущества перед службами каталога на базе Samba AD:

- Продукт основан на базе службы каталога FreeIPA, которая имеет более полную поддержку клиентской части SSSD и является лучшим решением для обеспечения работы Linux-компьютеров в составе домена. Если для SSSD бэкендом выступает Samba AD, то вам будут доступны только базовые функции аутентификации и авторизации.
- Продукт Samba AD не имеет перспектив дальнейшего развития, тем более для управления Linux-хостами.
- Более подробное сравнение этих служб каталога вы сможете найти в нашей статье.

Преимущества перед службами каталога на базе FreeIPA:

- Реализован удобный веб-портал с русскоязычным интерфейсом, который позволяет выполнять типовые заявки на обслуживание из современного графического интерфейса без необходимости обращения к командной строке и написания скриптов автоматизации. В портал управления встроен справочный центр, чтобы у пользователей под рукой всегда был доступ к рабочей документации.
- Реализована иерархия структурных подразделений и высокопроизводительный механизм групповых политик как в MS AD, позволяющий управлять десятками тысяч Linux-хостов из графического интерфейса в несколько кликов. На компьютерах в домене устанавливается автономная служба Salt Minion, которая как служба GPSVC от Microsoft извлекает параметры из каталога через LDAP, выполняет их суммирование в соответствии с положением пользователя/компьютера в иерархии структурных подразделений, применяет их с использованием скриптов одной из лучших систем конфигурирования SaltStack.
- Автоматизировано управление несколькими наиболее востребованными подсистемами: сервер репозиториев, файловый сервер, сервер печати, DHCP, PXE, аудит, мониторинг. Установка подсистем выполняется из веб-интерфейса в несколько кликов путем назначения серверу соответствующей роли. Настройка подсистем может быть осуществлена с любого контроллера домена вне зависимости от топологии сети, настройки будут доставлены до ближайшего сервера через LDAP-репликацию.

- Разработаны инструменты миграции и гибридного развертывания, такие как модуль синхронизации и глобальный каталог, чтобы помочь предприятиям в решении задач импортозамещения. Продуктовая команда ALD Pro первой в мире реализовала возможность установления доверительных отношений между доменами FreeIPA.

- Более 40+ готовых интеграций с продуктами сторонних вендоров, список которых постоянно расширяется.

- Создан открытый курс по продукту, который включает в себя более 30 учебных модулей, раскрывающих наиболее важные аспекты работы с системой. Планируется и дальше уделять значительные усилия развитию документации, чтобы максимально упростить освоение используемых технологий.

Какой технологический стек используется в ALD Pro?

В составе продукта используются следующие компоненты:

- **Контроллер домена**:

  - FreeIPA - служба каталога, которая обеспечивает интеграцию DNS-сервера (Bind9) и Kerberos KDC (MIT KDC) с LDAP-каталогом, предоставляет общее API, веб-интерфейс и инструменты командной строки.

  - 389 Directory Server - LDAP-каталог, который обеспечивает централизованное хранения данных о пользователях, хостах, сервисных учетных записях, и обеспечивает доступ к этой информации по открытому протоколу LDAP. Именно 389 Directory Server обеспечивает мультимастер-репликацию данных между всеми контроллерами в домене.

  - MIT KDC - центр распространения ключей Kerberos (Key Distribution Center), обеспечивает аутентификацию по протоколу Kerberos V5.

  - Bind9 - DNS-сервер, обеспечивает работу в домене службы доменных имен.

  - NTP Chrony - сервер времени, обеспечивает синхронизацию времени между всеми участниками в домене.

  - Компоненты ALD Pro:

    - Портал управления - веб-приложение, которое расширяет базовые возможности службы каталога FreeIPA, реализуя групповые политики и механизм управления всеми остальными подсистемами продукта. Указанное приложение разработано с применением фреймворка Django, в качестве встроенной базы данных используется PostgreSQL, фронтенд реализован на ReactJS, а в роли веб-сервера выступает

Apache.

- Модуль глобального каталога - набор программного обеспечения, который позволяет установить на контроллере домена еще один экземпляр LDAP-каталога с неполным набором атрибутов в схеме MS AD и сделать его доступным для Windows-клиентов по порту 3268/TCP. Глобальный каталог обеспечивает возможность полноценного использования двухсторонних доверительных отношений, позволяя администраторам MS AD настраивать доступ к ресурсам своего домена для пользователей и групп из домена ALD Pro с использованием стандартных оснасток Windows. В ванильной службе каталога FreeIPA нет модуля глобального каталога, поэтому поддержка двусторонних доверительных отношений без доработок от ALD Pro оказывается неполной.

- Модуль синхронизации - дает возможность поддерживать консистентность информации между двумя доменами MS AD и ALD Pro в части пользователей (включая их пароли), групп и структурных подразделений, что позволяет обеспечить сотрудникам доступ по логину/паролю к любой системе, вне зависимости от того, к какому домену она подключена. С помощью этого модуля можно реализовать плавную миграцию ИТ-инфраструктуры, в которой используется много информационных систем, подключенных к службе каталога через LDAP-коннекторы.

- **Клиентская часть**:

  - SSSD - клиентская служба, обеспечивающая работу компьютеров в составе домена.

  - Компоненты ALD Pro:

    - Salt Minion - автономная служба с дополнительными скриптами от продуктовой команды ALD Pro, обеспечивающая извлечение параметров групповых политик из LDAP-каталога и их применение на компьютерах в домене.

    - Агент Zabbix - служба подсистемы мониторинга Zabbix для сбора метрик (используется только на серверах подсистем).

    - Агент Syslog-NG - штатная служба операционной системы Astra Linux для сбора событий и передачи их в подсистему журналирования (на сервер аудита).

- **Подсистемы**:

  - Reprepro - служба, которая лежит в основе подсистемы репозиториев (сервера репозиториев). Предназначена для управления Debian-репозиториями, в которых можно хранить deb-пакеты для установки приложений.

  - Samba - служба, которая лежит в основе подсистемы общего доступа к файлам (файло-

вого сервера). Предоставляет общий доступ к файлам по протоколу SMB.

- CUPS - служба, которая лежит в основе подсистемы общего доступа к принтерам (принт-сервера). Предоставляет доступ к принтерам по протоколу IPP.
- ISC DHCP - служба, которая лежит в основе подсистемы динамической настройки хостов (DHCP-сервера).
- TFTP-HPA - служба, которая лежит в основе подсистемы установки ОС по сети (PXE-сервера).
- Syslog-NG - служба, которая лежит в основе подсистемы аудита (сервера журналирования).
- Zabbix - служба, которая лежит в основе подсистемы мониторинга.

Портал управления ALD Pro представляет собой web-приложение, созданное с использованием Django - свободного фреймворка на языке Python3 для разработки веб-приложений. В качестве базы данных используется PostgreSQL, фронтенд реализован с помощью ReactJS, а в роли веб-сервера выступает Apache.

Возможно ли заменить ALD Pro установкой отдельных компонентов?

Продукт ALD Pro представляет собой связующее программное обеспечение для объединения множества opensource-компонентов. Вы можете собрать систему на тех же компонентах без использования программного кода компании Астра, но в этом случае вам будут недоступны следующие функции:

- Распределение объектов каталога по иерархии организационных подразделений и делегирование полномочий по их управлению.
- Централизованная настройка рабочих станций через механизм групповых политик и возможность использования более 100 готовых параметров групповых политик, содержащих порядка 1000 настроек для конфигурирования хостов под управлением Astra Linux Special Edition.
- Автоматическое развертывание подсистем, предоставляющих наиболее востребованные ИТ-сервисы (сервер репозиториев, файловый сервер, принт-сервер, DHCP, установка ОС по сети, аудит, мониторинг).
- Удаленное управление рабочим столом пользователя для оказания технической поддержки.
- Глобальный каталог для полноценной поддержки двусторонних доверительных отношений.
- Синхронизация между двумя доменами MS AD и ALD Pro в части пользователей (включая их

пароли), групп и структурных подразделений, что позволяет обеспечить сотрудникам доступ по логину/паролю к любой системе, вне зависимости от того, к какому домену она подключена. Синхронизация дает возможность организовать плавную миграцию ИТ-инфраструктур, в которых используется много информационных систем, подключенных к службе каталога через LDAP-коннекторы.

- И многое другое.

Все подсистемы ALD Pro устанавливаются на один сервер?

Во избежание проблем с зависимостями deb-пакетов каждая подсистема ALD Pro устанавливается в отдельном экземпляре операционной системы, поэтому вместе с лицензией на контроллер домена в подарок идет 8 лицензий ALSE для развертывания одного контроллера и семи подсистем. Вы можете установить как по одному экземпляру каждой подсистемы, так и семь файловых серверов по своему усмотрению.

Для обеспечения отказоустойчивости системы рекомендуется приобретать не менее двух лицензий на установку контроллеров домена. В этом случае у вас будет возможность установить до 14 подсистем.

Сколько контроллеров может быть в домене ALD Pro?

Домен ALD Pro построен на базе службы каталога FreeIPA и поддерживает горизонтальное масштабирование. Вы можете создать любое количество контроллеров, и они все будут работать в режиме мульти-мастер, т.е. изменения можно будет вносить на любом их них. Инженеры продуктовой команды проводили успешное тестирование очень крупных доменов, в которых одновременно работало более 400 контроллеров.

База данных каталога автоматически реплицируется по протоколу LDAP, обеспечивая целостность «в конечном итоге», поэтому на всех серверах в домене будет содержаться примерно одна и та же информация. Для обеспечения отказоустойчивости на каждом контроллере рекомендуют создавать от двух до четырех соглашений о репликации, но это значение не является техническим ограничением. Инженеры продуктовой команды проводили успешное тестирование серверов, у которых было одновременно более 150 соглашений о репликации с другими контроллерами домена.

В случае географически распределенных инфраструктур вы можете объединить контроллеры в сайты, чтобы обслуживать клиентов только ближайшими к ним серверами. Количество сайтов не ограничено, вы можете создать любое число сайтов в соответствии с потребностями вашего предприятия.

Есть ли ограничение на количество пользователей/компьютеров в домене?

В программном комплексе ALD Pro сервер LDAP реализован на базе 389 Directory Server, который для хранения информации использует встраиваемую высокопроизводительную базу данных Berkeley DB (BDB). Максимально допустимый размер базы BDB зависит от размера страницы данных, но составляет не менее 2ТБ, что для службы каталога означает отсутствие каких-либо ограничений.

В то же время, чем больше объектов в базе, тем больше оперативной памяти требуется серверу, чтобы обрабатывать запросы на чтение без обращения к медленным дискам. В среднем нужно выделять 1 ГБ на каждые 20 тысяч объектов, поэтому для каталога на 2 млн. пользователей нужно порядка 100 ГБ ОЗУ. Инженеры продуктовой команды проводили успешное тестирование домена с каталогом, в котором было более 30 млн. объектов.

Примечание: Для экономии памяти при работе с большими каталогами рекомендуется отключать плагин `Schema Compatibility`, который строит виртуальную копию каталога для совместимости с устаревшими UNIX-клиентами.

Нагрузка по обслуживанию доменных компьютеров в части аутентификации и авторизации пользователей распределяется равномерно между всеми контроллерами в пределах сайта (рабочие станции выбирают активный контроллер случайным образом по результатам процедуры автоматического обнаружения, используя SRV-записи). Однако чем больше компьютеров нужно обслуживать конкретному контроллеру, тем больше серверу нужно ядер центрального процессора, в среднем по одному ядру на каждую тысячу обслуживаемых хостов.

Инженеры продуктовой команды проводили успешное тестирование контроллеров на серверах до 32 ядер включительно, и зафиксировали, что производительность растет линейно с коэффициентом 1.8, т.е. при увеличении числа ядер в 2 раза производительность в среднем возрастает на 80%. Мы рекомендуем эксплуатировать контроллеры, рассчитанные на 3-5 тысяч обслуживаемых хостов (8 ядер).

Если резюмировать, то домен ALD Pro на базе службы каталога FreeIPA не имеет ограничений по количеству пользователей, но для обеспечения высокой производительности системы вам требуется крайне тщательно подойти к вопросу планирования ресурсов, чтобы рассчитать необходимое количество контроллеров, их производительность и общую топологию домена.

Будет ли работать доменная аутентификация, если контроллер домена выйдет из строя?

Для аутентификации доменных пользователей клиентская служба SSSD выбирает три ближайших контроллера, выполняя автоматическое обнаружение сервисов по SRV-записям. Если активный контроллер окажется недоступен, то через минуту служба SSSD переключится на следующий сер-

вер по списку. Если все контроллеры домена окажутся недоступны, то служба SSSD перейдет в автономный режим и сможет выполнить аутентификацию доменного пользователя, используя информацию из кэша.

Можно ли войти в операционную систему ноутбука, который введен в домен, если я нахожусь вне офиса?

Для первого входа пользователя в систему компьютер должен иметь доступ к контроллеру домена, а в дальнейшем пользователь сможет входить без подключения к локальной сети офиса, т.к. в соответствии с настройкой `cache_credentials = True` в файле `/etc/sssd/sssd.conf` хеш пароля будет сохранен в кэше службы SSSD. Продолжительность кэширования не ограничена, т.к. в файле `sssd.conf` по умолчанию не определен параметр `offline_credentials_expiration`.

Если администратору потребуется передать сотруднику ноутбук через курьерскую службу, он может выполнить предварительную настройку кэша службы SSSD для конкретного пользователя с помощью команды `sss_seed`, чтобы сотрудник смог войти в операционную систему, не имея подключения к VPN.

Для удаленной смены пароля доменные пользователи могут использовать утилиту `fly-passwd`, которая позволяет не только изменить пароль в домене, но и обновить информацию в кэше службы SSSD.

Все вышеуказанное достоверно только для входа в компьютер. Получение Kerberos-билетов возможно только путем обращения к службе распространения ключей KDC на контроллере домена.

Где хранится каталог, какая используется база данных?

В качестве бэкенда служба каталога ALD Pro (FreeIPA) использует компонент 389 Directory Server, который известен так же под названиями Red Hat Directory Server, Fedora Directory Server и Netscape Directory Server.

Продукт 389 Directory Server является продолжением развития slapd от мичиганского университета, от которого также ответвился и OpenLDAP, но по мнению экспертов реализация 389ds превосходит другие форки по надежности и масштабируемости. В сети Интернет есть упоминания об инсталляциях на тысячи реплик.

Для работы с данными 389 Directory Server использует Sleepycat Berkeley DB (BDB) — высокопроизводительную встраиваемую систему управления нереляционными базами данных, которая хранит пары «ключ — значение» в виде бинарного дерева и поддерживает множество значений для одного ключа.

Текущая производительность BDB абсолютно сопоставима с производительностью базы MS AD, а в ближайшее время ожидается переход на бэкенд LMDB, который в большинстве операций показывает скорость выше в несколько раз.

Для прямого взаимодействия с базой администраторам доступны такие утилиты как `dsconf` и `dbscan`, но основное взаимодействие с каталогом осуществляется все же по открытому протоколу LDAP(S).

Где хранится организационная структура?

Организационная структура, как совокупность подразделений с их вложенностью, хранится в LDAP-записи каталога с DN `cn=orgunits,cn=accounts,dc=ald,dc=company,dc=lan`.

Для обеспечения обратной совместимости пользователи и компьютеры физически не перемещаются в этот контейнер, а их принадлежность к подразделениям определяется значением атрибута rbtadp (от RusBiTech Astra Division Path, OID 1.3.6.1.4.1.52616.100.2.3.1000).

Информация о структуре организации используется для назначения групповых политик и делегирования полномочий на управление объектами каталога.

Поддерживает ли ALD Pro протокол аутентификации Kerberos V5?

Служба каталога ALD Pro реализована на базе FreeIPA (389 Directory Server + MIT KDC + Samba + BIND9) и поддерживает аутентификацию по протоколам LDAP, Kerberos V5 и NTLM.

Для аутентификации по протоколу LDAP используется хеш пароля из атрибута userPassword, для Керберос-аутентификации ключи хранятся атрибуте krbPrincipalKey, а хеш пароля в формате NTLM находится в атрибуте ipaNTHash. Обновление всех этих атрибутов происходит автоматически при смене пароля по любому из протоколов, будь то LDAP или KPasswd.

Обратите внимание на то, что протокол NTLM считается устаревшим и предоставляется компонентом Samba на контроллерах домена исключительно для совместимости с доменом MS AD при работе в доверительных отношениях. Клиентская служба SSSD при обращении к домену использует только безопасную Kerberos-аутентификацию, которая дополнительно защищается технологией FAST (Kerberos Armoring).

Поддерживает ли ALD Pro работу с клиентскими ОС от других вендоров?

К службе каталога ALD Pro можно присоединять серверы и рабочие станции под управлением других операционных систем из семейств Linux и Windows, но функциональность будет отличаться.

Для компьютеров с операционными системами Linux сторонних вендоров, которые вводятся в домен с помощью службы SSSD, будут доступны только следующие базовые функции:

- аутентификация/авторизация доменных пользователей;
- прозрачная аутентификация пользователей при обращении к керберизированным сервисам (например, к файловому серверу);
- базовые политики безопасности (HBAC- и SUDO-правила, карты монтирования autofs).

Для компьютеров с операционной системой Windows, которые вводятся в домен с помощью утилиты aldpro-join.exe, будут доступны следующие функции:

- у пользователя в системе будет Kerberos-билет для прозрачной аутентификации в керберизированных сервисах;
- пользователь будет действовать в системе из-под своей доменной учетной записи, которая содержит как его собственный SID, так и идентификаторы всех его доменных групп.

Для компьютеров с операционной системой Astra Linux помимо базового функционала доступны расширенные функции:

- применение групповых политик ALD Pro и заданий автоматизации с использованием salt-скриптов;
- централизованная настройка аудита событий безопасности;
- возможность оказания технической поддержки путем подключения к рабочему столу пользователя по протоколу VNC.

В рамках дорожной карты развития продукта ALD Pro на 2025 год планируется реализация мультивенодрного подхода и выпуск клиентской части для двух операционных систем альтернативных вендоров (Ред ОС и Альт Линукс), что даст возможность централизованно управлять настройками этих систем с помощью механизма групповых политик ALD Pro.

Как лицензируется ALD Pro?

Для установки контроллера домена вам нужно приобрести лицензию на сервер ALD Pro, т.е. серверные лицензии приобретаются по количеству контроллеров в домене. Лицензия на операционную систему Astra Linux идет в комплекте с лицензией на сервер ALD Pro, поэтому приобретать ее дополнительно для установки контроллера домена не требуется.

Для присоединения сервера или рабочей станции к домену вам нужно приобрести клиентскую лицензию ALD Pro на управляемое устройство. Лицензию на управляемое устройство ALD Pro

можно приобрести как в составе бандла вместе с лицензией на операционную систему Astra Linux (с дополнительной скидкой), так и отдельно, например, при необходимости использования в составе домена операционных систем сторонних вендоров.

Для установки подсистем в комплекте с каждой лицензией на сервер ALD Pro идет 7 дополнительных лицензий на операционную систему Astra Linux и 7 лицензий на управляемые устройства ALD Pro. Вы можете по своему усмотрению установить как все системы по одной, так и семь файловых серверов сразу. Если указанного количества лицензий окажется недостаточно, вам нужно будет приобрести дополнительные лицензии на операционную систему Astra Linux и управляемые устройства ALD Pro по количеству дополнительных подсистем.

Решение ALD Pro приобретается отдельно от операционной системы?

Серверную лицензию ALD Pro можно приобрести только вместе с лицензией на операционную систему ALSE. Клиентскую лицензию ALD Pro на управляемое устройство можно приобрести как вместе с операционной системой ALSE в составе так называемого бандла (со скидкой), так и отдельно. Предприятия могут приобретать лицензии на управляемые устройства без лицензий на операционную систему, если они приобрели лицензии на операционные системы ранее или используют операционные системы альтернативных вендоров.

Словарь терминов

На этой странице представлены термины, которые используются в материалах курса.

389 Directory Server

LDAP-каталог с открытым исходным кодом, предназначенный для хранения иерархических данных в формате «ключ-значение».

ACI

Access Control Instructions (ACI) – это инструкция по контролю доступа 389-ds сервера. Уровень разрешений, предоставленный пользователю, зависит от предоставленной аутентификационной информации и других параметров.

ALD Pro

Служба каталога, которая позволит при переходе от Windows к Linux сохранить лучшие практики системного администрирования, доступные ранее при эксплуатации Active Directory.

ALSE

Операционная система специального назначения (*англ. Astra Linux Special Edition, ALSE*), предназначенная для применения в составе информационных (автоматизированных) систем в целях обработки и защиты информации любой категории доступа — общедоступной информации, а также информации, доступ к которой ограничен федеральными законами (информации ограниченного доступа).

Apache Directory Studio

Графическое приложение для удобного просмотра и редактирования записей LDAP-каталогов.

API

Application Programming Interface – программный интерфейс прикладного программирования для интеграции сторонних приложений.

Astra Automation

Платформа для автоматизации ИТ-операций на базе Ansible. Astra Automation подходит для

серверных, сетевых, облачных и прочих ИТ-инфраструктур любого масштаба, но наибольшую выгоду дает на крупных инфраструктурах, требующих сопровождения в условиях ограниченного штата сотрудников.

BIND

BIND 9 – открытая и наиболее распространённая реализация DNS-сервера, обеспечивающая выполнение преобразования DNS-имени в IP-адрес и наоборот. В составе ALD Pro (FreeIPA) служба BIND9 берет DNS-записи из LDAP-каталога.

boot menu

Встроенное в BIOS меню загрузки, предназначенное для выбора приоритета загрузки устройств.

Celery

Асинхронная очередь задач или очередь заданий с открытым исходным кодом, основанная на распределенной передаче сообщений.

CI/CD

Continuous integration/Continuous delivery(deployment) – Комбинация непрерывной интеграции и непрерывного развертывания программного обеспечения в процессе разработки.

CIFS

Название для первой версии SMB (*от англ. Common Internet File System*), а начиная со второй версии его называют SMB (Server Message Block).

CLI

Command Line Interface CLI – Интерфейс командной строки.

CSV

Текстовый формат, предназначенный для представления табличных данных.

CUPS

Common UNIX Printing System – Сервер печати для UNIX-подобных операционных систем.

CVE

Common Vulnerabilities and Exposures – База данных общеизвестных уязвимостей информационной безопасности. Каждой уязвимости присваивается идентификационный номер вида CVE-год-номер, описание и ряд общедоступных ссылок с описанием.

deb-пакет

Разновидность пакетов для установки программного обеспечения с использованием пакетного менеджера dpkg в операционных системах семейства Debian. Широко известны так же rpm-пакеты для операционных систем из семейства Red Hat.

DHCP

Динамическая настройка узла (*от англ. Dynamic Host Configuration Protocol*) – Сетевой протокол, позволяющий сетевым устройствам автоматически получать IP-адрес и другие параметры, необходимые для работы в сети TCP/IP. RFC 2131

Django

Свободный фреймворк для веб-приложений на языке Python, использующий шаблон проектирования MVC.

DNS

Система доменных имен (*от англ. Domain Name System*) – распределённая система для получения информации о доменах.

DNS-зона

Часть иерархического дерева доменных имен, переданная на обслуживание определенному DNS-серверу (или нескольким) с целью делегирования другому лицу ответственности за этот и все подчиненные домены. RFC 1035

DNS-клиент

Набор программных средств для работы с DNS. Сам DNS-сервер периодически также выступает в качестве клиента, когда обращается к другим DNS-серверам. RFC 1035

DNS-сервер

Хост, хранящий ресурсные записи и обрабатывающий DNS-запросы. DNS-сервер может самостоятельно разрешать адреса, относящиеся к зоне его ответственности или передавать запросы по зонам, которые он не обслуживает, вышестоящим серверам. RFC 1035

Fly

Графическая оболочка ОС Astra Linux.

FQDN

Fully Qualified Domain Name – Полное доменное имя, которое не имеет неоднозначностей в определении. Это текстовая запись с именем хоста и именем домена, включая домен верхнего уровня, в формате: [hostname].[domain].[tld].

FreeIPA

Служба каталога, предназначенная для централизованного управления сетевыми службами, идентификацией и аутентификацией, а также для установки доверительных отношений и обеспечения взаимодействия с доменом Active Directory (MS AD).

FTP

File Transfer Protocol — Протокол передачи файлов. RFC 959

FTP-сервер

Сервер, предназначенный для обмена файлами через Интернет или локальную компьютерную сеть по протоколу FTP. RFC 959

GID

Group Identifier – Уникальный идентификатор группы пользователей в POSIX-системах.

Git

Наиболее популярная система контроля версий для совместной разработки программного кода.

Grafana

Свободная программная система визуализации данных, ориентированная на данные систем ИТ-мониторинга.

GUI

Graphical User Interface – Графический интерфейс, который предоставляет пользователю удобный способ работы с операционной системой и приложениями.

HBAC

Host-based access control – Набор правил для настройки доступа пользователей или групп пользователей к определенным хостам с использованием указанных сервисов.

HTTP

Протокол передачи гипертекста (*от англ. HyperText Transfer Protocol*), по которому данные в интернете передаются между разными источниками, обычно между компьютерами и серверами.

HTTPS

Расширение протокола HTTP, которое предусматривает шифрование данных через SSL.

IP

Internet Protocol – Маршрутизируемый протокол сетевого уровня стека TCP/IP. RFC 791

IP-адрес

Уникальный адрес, идентифицирующий устройство в интернете или локальной сети, работающей по протоколу IP. RFC 791

ISO-образ

Образ оптического диска, содержащий файловую систему стандарта ISO 9660. Данный формат часто используется для распространения дистрибутивов операционных систем.

jid

Job Identity – Уникальный идентификатор задания Salt.

job

Задание, которое выполняется службой Salt-Minion.

Kerberos

Безопасный протокол трехсторонней аутентификации в компьютерной сети, где в качестве третьей доверенной стороны выступает Kerberos-сервер. Основанием доверия между доменом и пользователем является пароль пользователя, а между доменом и сервисом – пароль сервиса.

LDAP

LDAP-протокол – Протокол прикладного уровня для доступа к службе каталога. RFC 4511

LSDOU

Порядок суммирования параметров групповых политик от *англ. Local, Site, Domain, OU.* Сначала применяются локальные политики (L), затем политики сайта (S), домена (D) и в завершение политики организационных подразделений (OU). Таким образом, параметры, назначенные на организационные подразделения всегда превалируют над параметрами локальных политик.

MAC-адрес

Уникальный идентификатор сетевого устройства, присваиваемый каждому сетевому устройству. RFC 7769

Microsoft Active Directory

Служба каталога от компании Microsoft для операционных систем семейства Windows.

MS AD

Служба каталога Active Directory от компании Microsoft, предназначенная для централизованного управления ИТ-инфраструктурой, в состав которой входят преимущественно Windows-компьютеры.

NetBIOS

Network Basic Input/Output System — протокол был разработан фирмой Sytek Corporation по заказу IBM в 1983 году для работы в локальных сетях. Он включает в себя интерфейс сеансового уровня (*англ. NetBIOS interface*), в качестве транспортных протоколов использует TCP и UDP.

NetworkManager

Программа (network-manager) для управления сетевыми соединениями в Linux.

NTP

Протокол сетевого времени (*от англ. Network Time Protocol*), который широко используется для синхронизации системных часов между набором распределенных серверов времени и

клиентов. RFC 5905

OTP

Одноразовый пароль (*англ. One-Time Password*), который используется для повышения безопасности аутентификации пользователя.

PEM

Privacy Enhanced Mail – Наиболее распространенный формат для сертификатов X.509, CSR и криптографических ключей. Файл PEM представляет собой текстовый файл, содержащий один или несколько элементов в кодировке Base64 ASCII, каждый из которых имеет заголовки и нижние колонтитулы в виде обычного текста.

pool

Логическое объединение серверов управления или сервер-шлюзов, которые распределяют между собой рабочие нагрузки и принимают на себя рабочие нагрузки в случае сбоя одного из серверов.

POSIX идентификатор

Уникальное числовое значение, используемое для идентификации пользователей (UID) и групп пользователей (GID) в операционных системах, которые следуют стандартам POSIX.

PostgreSQL

Свободная объектно-реляционная система управления базами данных.

Postinstall скрипт

Скрипт, включающий в себя инструкции для автоматического выполнения после установки операционной системы, предназначен для автоматизации установки и настройки определенных параметров

Postman

Платформа для создания и использования API.

Preseed скрипт

Скрипт, содержащий заранее определенные ответы на вопросы, которые операционная система задает в процессе установки, такие как параметры настройки, выбор языка, разделов диска и другие конфигурационные опции

Ready for Astra Linux

Программа технологического партнёрства с производителями оборудования и сторонними разработчиками программного обеспечения.

В рамках программы проводятся совместные испытания ПО (оборудования) и ОС «Astra Linux Special Edition». Методика совместных испытаний опубликована на странице

– https://wiki.astralinux.ru/rfa

Redis

Резидентная система управления базами данных класса NoSQL с открытым исходным кодом, работающая со структурами данных типа «ключ — значение».

salt-master

Серверная служба системы Salt, к которой подключаются управляемые хосты для получения команд.

salt-minion

Клиентская служба системы Salt, которая подключается к серверу для получения от него команд.

SaltStack

Компания, которая разрабатывает технологию централизованного управления конфигурациями хостов под названием Salt. Компания находится в Salt Lake City, отсюда и название.

Samba

Пакет программ, которые позволяют обращаться к сетевым дискам и принтерам на различных операционных системах по протоколу SMB/CIFS.

SLA

Service Level Agreement – Соглашение об уровне предоставления услуги.

SMB

Server Message Block – Сетевой протокол для выполнения удаленных процедур (англ. *Remote Procedure Calls, RPC*). Через этот протокол реализуется не только общий доступ к файлам и принтерам, но и выполняется преобразование идентификаторов в домене, а так же работает аутентификация NETLOGON и др. функции.

SMTP

Simple Mail Transfer Protocol – Базовый протокол для передачи электронной почты в Интернете. RFC 5321

snapshot

Моментальная копия хранимых данных, файлов и каталогов файловой системы, базы данных или её части на определённый момент времени.

SPNEGO

Simple and Protected GSSAPI Negotiation Mechanism – Механизм GSSAPI для согласования выбора технологии безопасности в клиент-серверном программном обеспечении. Используется, когда клиентское приложение хочет аутентифицироваться на удалённом сервере.

SSH

Протокол для безопасного удаленного входа в систему и других сетевых служб в незащищенной сети (*англ. Secure Shell*). RFC 4254

SSH-ключи

Связанная пара из открытого и закрытого ключа для безопасной аутентификации по протоколу SSH с использованием асимметричных алгоритмов шифрования.

SSL-сертификат

Цифровой сертификат (открытый ключ и цифровая подпись), удостоверяющий подлинность сервера и позволяющий установить безопасное зашифрованное соединение Secure Sockets Layer.

SSO

Метод аутентификации (*англ. Single Sign-On*), который позволяет предоставить пользователю авторизованный доступ ко множеству систем, выполнив аутентификацию только один раз. В домене ALD Pro (FreeIPA) используется SSO на базе протокола Kerberos V5. В последние годы с развитием веб-технологий широкую популярность приобрели протоколы OAuth2, OIDC, SAML, поддержку которых можно реализовать путем интеграции домена с Web SSO, например, с Keycloak.

SSSD

System Security Services Daemon (SSSD) — это пакет приложений для управления аутентификацией и авторизацией в операционных системах на базе Linux. Основная цель службы SSSD заключается в том, чтобы хранить данные удаленной базы в локальном кэше и передавать их локальным приложениям.

Поддержание актуальности и достоверности кэша является сложной задачей, поэтому SSSD состоит из нескольких компонентов (процессов и библиотек), которые взаимодействуют друг с другом с помощью различных методов межпроцессного взаимодействия.

SUDO

Утилита для выполнения команд от имени суперпользователя (*от англ. substitute user and do*).

UID

User Identifier – Уникальный идентификатор пользователя в POSIX-системах.

Zabbix

Свободная система мониторинга состояния любых сетевых служб и оборудования.

ZeroMQ

Шина сообщений в Salt, которая позволяет создать распределенную асинхронную сеть обмена

сообщениями.

Дистрибутив

Форма распространения программного обеспечения, обычно содержащая программу-установщик (для выбора режимов и параметров установки) и набор файлов, содержащих отдельные части программного средства. ГОСТ Р 53623-2009

Домен

Логический административный контейнер для таких объектов, как пользователи и компьютеры.

Клиент-сервер

Архитектурное решение, в котором нагрузка распределена между поставщиками услуг, называемых серверами и заказчиками услуг, называемых клиентами.

Клиентская часть

Программное обеспечение, работающее на устройстве пользователя и взаимодействующее с серверной частью посредством запросов. Предназначена по большей части для задания параметров и визуализации данных.

Компоненты дистрибутива

Списки пакетов, входящих в состав дистрибутива/репозитория, содержатся в самом дистрибутиве/репозитории.

Контроллер домена

Сервер, выполняющий функцию централизованного управления доменом.

ЛВС

Локальная вычислительная сеть – сеть, охватывающая небольшую территорию и использующая ориентированные на эту территорию средства и методы передачи данных. ГОСТ 24402-88

МКЦ

Мандатный контроль целостности (*англ. MIC, Mandatory Integrity Control*) – Механизм безопасности, который реализует динамическое управление целостностью данных и ресурсов в информационной системе. Под целостностью понимается степень доверия к субъекту или содержанию объекта.

В рамках мандатного контроля целостности, каждому объекту в системе (например, файлу, процессу, ресурсу) присваивается уровень (метка) целостности.

МРД

Мандатное управление доступом (*англ. MAC, Mandatory Access Control*) – Подход к управлению безопасностью в информационных системах, при котором только администратор (а не

владелец объектов) определяет политику доступа к объектам.

При реализации политики мандатного управления доступом субъектам и объектам присваиваются классификационные метки, состоящие из уровня и категории конфиденциальности. Также сущностям могут быть присвоены дополнительные мандатные атрибуты управления доступом.

Организационная структура

Логические объекты, используемые для определения иерархической структуры компании.

ОС

Операционная система – совокупность системных программ, предназначенная для обеспечения определенного уровня эффективности системы обработки информации за счет автоматизированного управления ее работой и предоставляемого пользователю определенного набора услуг. ГОСТ 15971-90

Политика ПО

Набор правил, позволяющих устанавливать, настраивать и обновлять ПО на компьютерах.

Пользователь

Объект который получает доступ к службам и серверам в домене с помощью проверки подлинности Kerberos.

Программное обеспечение

Совокупность компьютерных программ и программных документов, необходимых для эксплуатации этих программ. ГОСТ Р 51904-2002

Изменения в курсе

Версия для релиза 2.4.0+ от 12.05.2025

Содержание второй части курса значительно переработано с учетом возможностей LTS-релиза 2.4.0 и нововведений крайней версии 2.5.0. Реализованы интерактивные тесты.

Версия для релиза 2.x.x от 24.02.2025

Содержание первой части курса актуализировано с учетом нововведений, которые стали доступны в операционной системе ALSE версии 1.7.6.

Версия для релиза 2.x.x от 11.11.2024

Добавили отзывы, исправили опечатки, внесли ряд других изменений.

Версия для релиза 2.x.x от 04.11.2024

Первое размещение материалов в публичном пространстве. До этого материалы курса распространялись только в виде отдельных pdf-файлов для проведения обучения в фокус-группах с целью апробации. На момент публикации материалов обучение по курсу уже прошли более 500 инженеров.

Все материалы в PDF-формате

Для того чтобы материалы курса были всегда под рукой, мы подготовили для вас специальную PDF-версию. Смеем заметить, что это не только позволит вам с пользой проводить время в дороге до офиса, но и поможет с настройкой домена в закрытых периметрах.

Скачать pdf-файл
