50 1190 0101

Утвержден

РУСБ.10152-02-УД

Подп. и дата	
Инв. № дубл	
Взам. инв. №	
Подп. и дата	
Инв. № подл	

ОПЕРАЦИОННАЯ СИСТЕМА СПЕЦИАЛЬНОГО НАЗНАЧЕНИЯ «ASTRA LINUX SPECIAL EDITION»

Руководство администратора. Часть 1.

Бюллетень № 2022-0114SE47

РУСБ.10152-02 95 01-1

Листов 360

2022

Литера O_1

4.7.1

АННОТАЦИЯ

Настоящий документ является первой частью руководства администратора программного изделия РУСБ.10152-02«Операционная система специального назначения «Astra Linux Special Edition» (далее по тексту — ОС).

Документ предназначен для администраторов системы и сети. Администраторы безопасности должны руководствоваться документом РУСБ.10152-02 97 01-1 «Операционная система специального назначения «Astra Linux Special Edition». Руководство по КСЗ. Часть 1».

Руководство администратора состоит из двух частей:

- РУСБ.10152-02 95 01-1 «Операционная система специального назначения «Astra Linux Special Edition». Руководство администратора. Часть 1»;
- РУСБ.10152-02 95 01-2 «Операционная система специального назначения «Astra Linux Special Edition». Руководство администратора. Часть 2».

Перед установкой и настройкой ОС необходимо провести ее контроль, предусмотренный формуляром при первичном закреплении экземпляра ОС за ответственным лицом.

В первой части руководства приведено назначение, установка и настройка ОС. Рассмотрены системные компоненты, службы и команды, базовые сетевые службы, средства организации ЕПП, защищенная графическая подсистема, управление программными пакетами, резервное копирование и восстановление данных, система печати, защищенная СУБД, защищенные комплексы программ гипертекстовой обработки данных и электронной почты, средства контроля целостности, централизованного протоколирования и разграничения доступа к подключаемым устройствам. Приведен список сообщений для администратора.

Требования к обеспечению безопасности среды функционирования, а также настройка параметров, необходимых для безопасной эксплуатации ОС, приведены в документе РУСБ.10152-02 97 01-1 и выполняются администратором безопасности.

Дополнительная информация о настройке компонентов и управлении программными пакетами, а также варианты реализации отдельных решений с использованием ОС приведены на официальном сайте `https://wiki.astralinux.ru`.

Во второй части руководства приведено описание работы с защищенной СУБД.

СОДЕРЖАНИЕ

[1)] Для процессоров, поддерживающих технологию виртуализации.

[1)] Для процессоров, поддерживающих технологию виртуализации.

РУСБ.10152-02 95 01-2 «Операционная система специального назначения «Astra Linux Special Edition». Руководство администратора. Часть 2»

1. АДМИНИСТРИРОВАНИЕ ОС

Административное управление в ОС отделено от общего доступа пользователей.

Большинство операций по настройке и администрированию ОС требуют привилегий суперпользователя (`root`), например:

- монтирование и размонтирование ФС;
- изменение корневого каталога процесса командой `chroot`;
- создание файлов устройств;
- установка системных часов;
- изменение принадлежности файлов;
- задание host-имени системы;
- конфигурирование сетевых интерфейсов.

ВНИМАНИЕ! После установки ОС интерактивный вход в систему суперпользователя по умолчанию заблокирован. Для администрирования системы при установке операционной системы создается пользователь, входящий я группу `astra-admin` и имеющий максимальный уровень целостности. Пользователям, входящим группу `astra-admin`, через механизм `sudo` предоставляются права для выполнения действий по настройке ОС, требующих привилегий `root`. Далее по тексту такой пользователь именуется администратором. Описание механизма `sudo` приведено в 1.1.2.

ВНИМАНИЕ! Действия по администрированию ОС при включенном режиме мандатного контроля целостности (МКЦ) необходимо выполнять в привилегированном режиме на высоком уровне целостности.

1.1. Получение прав суперпользователя

Существует несколько способов получения прав суперпользователя:

- вход в систему от имени учетной записи `root` (по умолчанию заблокирован);
- использование команды `su` (по умолчанию заблокирован);
- использование команды `sudo` (рекомендуется).

1.1.1. su

Команда `su` используется пользователем для запуска команд от имени другого пользователя. В том числе могут быть запущены команды от имени учетной записи `root`.

При запуске команды `su` без параметров подразумевается, что пользователь хочет запустить командный интерпретатор `shell` от имени учетной записи `root`. При этом система просит ввести пароль от учетной записи `root`. При вводе правильного пароля запускаемый интерпретатор команд получает права и привилегии суперпользователя, ко-

торые сохраняются до завершения его работы. Для получения прав суперпользователя пользователю не требуется завершать свою сессию и вновь входить в систему.

С помощью команды su, вводимой с параметром -c, пользователь может исполнять отдельные команды от имени учетной записи root без запуска командного интерпретатора shell. Преимущество такого способа состоит в том, что пользователь получает права и привилегии суперпользователя на строго ограниченное время, а именно, на время исполнения заданной команды. Например, при необходимости поменять атрибуты файла от имени учетной записи root ввести команду:

```
su -c 'chmod 0777 /tmp/test.txt'
```

В этом случае (после ввода пароля учетной записи root) команда chmod получит права и привилегии суперпользователя, но по ее завершении пользователь останется в своей сессии и не будет обладать правами и привилегиями суперпользователя.

Кроме выполнения команд от имени учетной записи root, команда su позволяет выполнять команды от имени любого другого пользователя. Для этого необходимо знать пароль этого пользователя. Если пользователь вошел в систему под именем root и выполняет команду su, то знание пароля пользователя не требуется — в данном случае все команды от имени любого пользователя исполняются свободно.

Недостаток команды su состоит в том, что она не регламентирует команды, разрешенные конкретному пользователю на запуск от имени учетной записи root. Таким образом, если у пользователя есть права на запуск команды su, то он может выполнить от имени учетной записи root любые команды. Поэтому ее запуск должен быть разрешен только доверенным пользователям. Также рекомендуется при вводе команды использовать полное путевое имя /bin/su, а не просто su.

Описание команды приведено в man su.

1.1.2. sudo

Команда sudo используется обычным пользователем для запуска команд от имени учетной записи root. Для работы команда sudo просматривает конфигурационный файл /etc/sudoers, который содержит список пользователей, имеющих полномочия на ее применение и перечень команд, которые они имеют право выполнять. В качестве аргументов команда sudo принимает командную строку, которую следует выполнить с правами суперпользователя. Если данному пользователю разрешено выполнять указанную им команду, то sudo просит пользователя ввести его собственный пароль. Таким образом, для каждого пользователя установлен набор команд, которые он может исполнять от имени учетной записи root, и нет необходимости передавать пользователям пароль учетной записи root.

Кроме выполнения указанной команды, `sudo` ведет файл регистрации выполненных команд, вызвавших их лиц, каталогов, из которых вызывались команды, и времени их вызова. Эта информация регистрируется с помощью системы `syslog`.

Для изменения файла `/etc/sudoers` администратору следует использовать специальную команду `visudo`.

Преимущество механизма `sudo` в том, что обычные пользователи могут выполнять определенные задачи от имени учетной записи `root`, не имея при этом неограниченных прав и привилегий.

Описание команды приведено в `man sudo`.

1.2. Механизмы разделения полномочий

К механизмам разделения полномочий между системными администраторами ОС могут быть отнесены:

- механизм привилегий;
- механизм повышения полномочий на время выполнения команды (программы);
- механизм установки ACL на файлы.

Описание механизмов разделения полномочий приведено в документе РУСБ.10152-02 97 01-1 «Операционная система специального назначения «Astra Linux Special Edition». Руководство по КСЗ. Часть 1».

1.2.1. Механизм привилегий

Механизм привилегий ОС предназначен для передачи отдельным пользователям прав выполнения определенных, строго оговоренных, административных действий. Обычный пользователь системы не имеет дополнительных привилегий.

Привилегии наследуются процессами от своих «родителей» и не могут быть переданы сторонним процессам. Процессы, запущенные от имени суперпользователя, независимо от наличия у них привилегий, имеют возможность осуществлять все привилегированные действия.

Распределение (первоначальная настройка) привилегий выполняется только суперпользователем с максимальным уровнем целостности, установленным в ОС.

1.2.2. Механизм повышения полномочий

Механизм повышения полномочий позволяет повысить полномочия пользователя на время выполнения определенной программы. Настройка механизма может быть выполнена только суперпользователем с максимальным уровнем целостности, установленным в ОС.

1.2.3. Механизм установки ACL на файлы

Механизм установки ACL на файлы облегчает задачу распределения полномочий, позволяя предоставлять доступ только к тем файловым объектам, к которым он необходим в соответствии с ролью пользователя. Настройку механизмов ACL выполняет администратор с максимальным уровнем целостности, установленным в ОС.

2. УСТАНОВКА И НАСТРОЙКА ОС

2.1. Установка ОС

DVD-диск с дистрибутивом ОС (инсталляционный образ системы) содержит все необходимые файлы для выполнения ее полной или частичной установки на жесткий диск целевого компьютера, имеющего устройство чтения DVD-дисков. ОС можно также установить с USB-накопителя.

Подробное описание последовательности действий при установке ОС с DVD-диска и с USB-накопителя приведены в руководстве по установке, размещенном в каталоге `/install-doc` на DVD-диске с дистрибутивом.

2.2. Режимы работы ОС

При установке ОС необходимо выбрать уровень защищенности, на котором будет функционировать ОС. Доступно три уровня:

1) «Базовый»;
2) «Усиленный»;
3) «Максимальный».

Выбранному уровню защищенности соответствует определенный перечень функций безопасности ОС. На каждом уровне защищенности доступны к использованию функции безопасности предыдущего уровня.

ВНИМАНИЕ! Функции безопасности, недоступные на выбранном уровне защищенности, не могут быть включены в процессе установки и функционирования ОС.

Для уровня защищенности «Базовый» доступны функции безопасности:

1) «Запрет вывода меню загрузчика» — при выборе данного пункта будет запрещен вывод меню загрузчика GRUB 2. В процессе загрузки будет загружаться ядро ОС, выбранное по умолчанию. По умолчанию пункт не выбран;

2) «Запрет трассировки ptrace» — при выборе данного пункта будет выключена возможность трассировки и отладки выполнения программного кода. По умолчанию пункт выбран;

3) «Запрос пароля для команды sudo» — при выборе данного пункта будет включено требование ввода пароля при использовании механизма `sudo`. По умолчанию пункт выбран;

4) «Запрет установки бита исполнения» — при выборе данного пункта будет включен режим запрета установки бита исполнения, обеспечивающий предотвращение несанкционированного запуска исполняемых файлов и сценариев для командной оболочки. По умолчанию пункт не выбран;

5) «Запрет исполнения скриптов пользователя» — при выборе данного пункта будет блокировано интерактивное использование пользователем интерпретаторов. По умолчанию пункт не выбран;

6) «Запрет исполнения макросов пользователя» — при выборе данного пункта будет блокировано исполнение макросов в стандартных приложениях. По умолчанию пункт не выбран;

7) «Запрет консоли» — при выборе данного пункта будет блокирован консольный вход в систему для пользователя и запуск консоли из графического интерфейса сессии пользователя. По умолчанию пункт не выбран;

8) «Системные ограничения ulimits» — при выборе данного пункта будут включены системные ограничения, установленные в файле `/etc/security/limits.conf`. По умолчанию пункт не выбран;

9) «Запрет автонастройки сети» — при выборе данного пункта будет выключена автоматическая настройка сети в процессе установки ОС, сеть необходимо будет настроить вручную в соответствии с рекомендациями настоящего руководства. По умолчанию пункт не выбран;

10) «Местное время для системных часов» — при выборе данного пункта будет включен режим интерпретации показаний аппаратных (RTC) часов. По умолчанию пункт не выбран.

Для уровня защищенности «Усиленный» доступны функции безопасности уровня «Базовый», а также следующие функции:

1) «Мандатный контроль целостности» — при выборе данного пункта будет включен механизм мандатного контроля целостности. По умолчанию пункт выбран;

2) «Замкнутая программная среда» — при выборе данного пункта будет включен механизм, обеспечивающий проверку неизменности и подлинности загружаемых исполняемых файлов формата ELF. По умолчанию пункт не выбран;

3) «Очистка освобождаемой внешней памяти» — при выборе данного пункта будет включен режим очистки блоков ФС непосредственно при их освобождении, а также режим очистки разделов страничного обмена. По умолчанию пункт не выбран.

Для уровня защищенности «Максимальный» доступны функции безопасности уровней «Базовый» и «Усиленный», а также функция:

1) «Мандатное управление доступом» — при выборе данного пункта будет включен механизм мандатного управления доступом. По умолчанию пункт выбран.

Соответствие уровней защищенности и доступных функций безопасности приведено в таблице 1.

Таблица 1

Функция безопасности	Уровень защищенности		
	«Базовый»	«Усиленный»	«Максимальный»
Замкнутая программная среда	Не доступна	Доступна (по умолчанию выключена)	Доступна (по умолчанию выключена)
Очистка освобождаемой внешней памяти	Не доступна	Доступна (по умолчанию выключена)	Доступна (по умолчанию выключена)
Мандатный контроль целостности	Не доступна	Доступна (по умолчанию включена)	Доступна (по умолчанию включена)
Мандатное управление доступом	Не доступна	Не доступна	Доступна (по умолчанию включена)

Описание функций безопасности ОС и порядок их использования приведено в документе РУСБ.10152-02 97 01-1. Подробнее о настройке системных часов приведено в 6.7.4.4.

2.3. Первичная настройка ОС

В пунктах 2.3.1–2.3.3 приведено описание настроек ОС для соответствующего уровня защищенности в случае, если при установке ОС были выбраны предложенные по умолчанию значения.

2.3.1. Уровень защищенности «Базовый» («Орел»)

Уровень защищенности «Базовый» рекомендуется применять для обработки общедоступной информации.

После установки ОС готова к использованию без дополнительных настроек.

На данном уровне защищенности для разграничения доступа применяется механизм дискреционного управления доступом. По умолчанию выключены режим отладки ptrace и возможность использовать механизм `sudo` без ввода пароля.

Дополнительно для защиты информации могут использоваться доступные системные ограничения, а также функции безопасности, ограничивающие действия пользователей.

Настройка функций безопасности выполняется в соответствии с документом РУСБ.10152-02 97 01-1.

2.3.2. Уровень защищенности «Усиленный» («Воронеж»)

Уровень защищенности «Усиленный» рекомендуется для обработки информации ограниченного доступа, не содержащей сведения, составляющие государственную тайну.

После установки ОС мандатный контроль целостности (МКЦ) ОС и файловой системы включаются автоматически. При включенном режиме МКЦ администрирование и настройка ОС должны выполняться только администратором на высоком уровне целостности.

На данном уровне защищенности для разграничения доступа применяется механизм дискреционного управления доступом. По умолчанию выключены режим отладки ptrace и возможность использовать механизм `sudo` без ввода пароля.

Дополнительно для защиты информации могут использоваться доступные системные ограничения, а также функции безопасности, ограничивающие действия пользователей.

Также на данном уровне защищенности для защиты информации доступны механизмы очистки памяти и организация замкнутой программной среды.

Настройка средств защиты информации и функций безопасности выполняется в соответствии с документом РУСБ.10152-02 97 01-1.

2.3.3. Уровень защищенности «Максимальный» («Смоленск»)

Уровень защищенности «Максимальный» рекомендуется для обработки информации ограниченного доступа, содержащей сведения, составляющие государственную тайну.

После установки ОС режим МКЦ ОС и файловой системы включаются автоматически. При включенном режиме МКЦ администрирование и настройка ОС должны выполняться только администратором на высоком уровне целостности.

На данном уровне защищенности для разграничения доступа по умолчанию применяются механизмы мандатного управления доступом и дискреционного управления доступом. После установки ОС требуется определить режим работы КСЗ и выполнить генерацию КСЗ в соответствии с РУСБ.10152-02 97 01-1.

По умолчанию выключены режим отладки ptrace и возможность использовать механизм `sudo` без ввода пароля.

Дополнительно для защиты информации могут использоваться доступные системные ограничения, а также функции безопасности, ограничивающие действия пользователей.

Также на данном уровне защищенности для защиты информации доступны механизмы очистки памяти и организация замкнутой программной среды.

Настройка средств защиты информации и функций безопасности выполняется в соответствии с документом РУСБ.10152-02 97 01-1.

2.4. Обновление ОС

Для установки обновлений ОС используется инструмент командной строки `astra-update`.

Общий синтаксис команды:

```
astra-update [действие] [параметр] [источник][[источник]..]
```

При запуске команды может быть выбрано только одно действие. Список основных действий `astra-update` приведен в таблице 2.

Таблица 2

Действие	Описание
`-c`	Проверить, можно ли устанавливать обновление. Изменения в систему не вносятся. Является действием по умолчанию — выполняется, если в команде действие не указано
`-a`	Установить обновление автоматически в интерактивном режиме (с выводом запросов пользователю), выполняя автоматическое выключение и включение функций безопасности. Представляет собой последовательное выполнение действий `-d`, `-i` и `-e`
`-A`	Установить обновление полностью автоматически (без вывода сообщений и запросов пользователю), выполняя автоматическое выключение и включение функций безопасности. Представляет собой последовательное выполнение действий `-d`, `-I` и `-e`. Данный режим предназначен для массовой автоматической установки обновлений на удаленных компьютерах, в том числе для использования в сценариях puppet/ansible. Устройства чтения оптических дисков, добавленные с помощью команды `sudo apt-cdrom add`, не будут использованы в процессе неинтерактивной установки, т.к. могут потребовать действий пользователя
`-d`	Отключить функции безопасности, мешающие обновлению. Состояние функций безопасности при этом будет сохранено в файле `/etc/parsec/update-saveconf`
`-I`	Установить обновление в неинтерактивном режиме (без вывода сообщений и запросов пользователю) и не выполняя выключение и включение функций безопасности
`-i`	Установить обновление в интерактивном режиме (с выводом запросов пользователю) и не выполняя выключение и включение функций безопасности
`-e`	Включить функции безопасности, которые были выключены перед обновлением действием `-d`. Состояние функций безопасности будет восстановлено из файла `/etc/parsec/update-saveconf`. Если файл не существует, то никакие изменения в систему внесены не будут

Список параметров `astra-update` приведен в таблице 3.

Таблица 3

Параметр	Описание
`-k`	Сохранить источники для последующего использования (файлы `iso`-образов будут скопированы на диск и указаны в `/etc/fstab`, сетевые репозитории будут добавлены в файл `/etc/apt/sources.list`)
`-g`	Проверить контрольные суммы файла `iso`-образа по алгоритму ГОСТ
`-m`	Проверить контрольные суммы файла `iso`-образа по алгоритму MD5

Окончание таблицы 3

Параметр	Описание
`-r`	Установка обновления из репозиториев, перечисленных в файле `/etc/apt/sources.list` (без внесения изменений в сам файл)
`-n`	Только имитировать установку обновления, без внесения изменений в систему

В качестве источника может быть указан файл `iso`-образа или сетевой репозиторий. Может быть указано несколько источников, разделенных пробелом.

ВНИМАНИЕ! Инсталляционный образ системы всегда должен присутствовать в `/etc/apt/sources.list` или быть указан в качестве источника при выполнении команды `astra-update`.

Для установки обновлений также может использоваться графическая утилита `fly-astra-update`. Описание утилиты приведено в электронной справке.

3. СИСТЕМНЫЕ КОМПОНЕНТЫ

3.1. Управление устройствами

3.1.1. Типы устройств

В ОС существует два типа устройств:

1) блочные устройства с произвольным доступом — данные, записанные в такие устройства, могут быть прочитаны оттуда, куда записаны (например, жесткие диски);

2) символьные устройства с последовательным или произвольным доступом — данные, записанные в такие устройства, не могут быть прочитаны (например, последовательные порты).

Каждое поддерживаемое устройство представляется в ФС файлом устройства. При выполнении операций чтения или записи с файлом устройства происходит обмен данными с устройством, на которое указывает этот файл. Данный способ доступа к устройствам позволяет не использовать специальные программы (а также специальные методы программирования, такие как работа с прерываниями).

Так как устройства отображаются как файлы в ФС (в каталоге `/dev`), их можно просмотреть с помощью команды `ls`. После выполнения команды с параметром `-l`:

```
ls -l
```

на экран монитора выводится список файлов с указанием в первой колонке типа файла и прав доступа к нему. Например, для просмотра файла, соответствующего звуковому устройству, используется следующая команда:

```
ls -l /dev/dsp
crw-rw---T+ 1 root audio 14, 3 Июл 1 13:05 /dev/dsp
```

Первый символ `c` в первой колонке указывает на тип файла — в данном случае символьное устройство. Для обычных файлов используется символ «`-`» (дефис), для каталогов — `d`, для блочных устройств — `b` (описание команды приведено в `man ls`).

Наличие файлов устройств не означает, что данные устройства установлены в системе. Например, наличие файла `/dev/sda` не означает, что на компьютере установлен жесткий диск SCSI. Это предусмотрено для облегчения установки программ и нового оборудования, т. к. исключает необходимость поиска нужных параметры и создания файлов для новых устройств.

3.1.2. Жесткие диски

При администрировании дисков могут возникнуть вопросы, связанные с разделением жесткого диска на разделы, созданием и монтированием ФС, форматированием диска и др.

Одна из причин разделения жесткого диска — это хранение разных ОС на одном жестком диске. Другая причина — хранение пользовательских и системных файлов в раз-

ных дисковых разделах, что упрощает резервное копирование и восстановление, а также повышает защищенность системных файлов от повреждений.

Для использования диска или раздела необходимо создание на нем ФС.

Для штатного доступа к данным, находящимся в ФС, необходимо выполнить монтирование ФС. Монтирование выполняется с целью формирования единой структуры каталогов, обеспечения буферизации дисков и работы с виртуальной памятью.

Монтирование может выполняться как автоматически, так и вручную. ФС, монтируемые вручную, должны быть размонтированы вручную.

Центральный процессор и жесткий диск обмениваются информацией через дисковый контроллер. Это упрощает схему обращения и работы с диском, т. к. контроллеры для разных типов дисков могут быть построены с использованием единого интерфейса для связи с компьютером.

Каждый жесткий диск представлен отдельным файлом устройства в каталоге `/dev`: `/dev/hda` и `/dev/hdb` для первого и второго диска, подключенного по IDE шине, и `/dev/sda`, `/dev/sdb` и т. д. для дисков, использующих SCSI или SATA-интерфейс.

3.1.3. Разделы жесткого диска

Весь жесткий диск может быть разделен на несколько дисковых разделов, причем каждый раздел представлен так, как если бы это был отдельный диск. Разделение используется, например, при работе с двумя ОС на одном жестком диске. При этом каждая ОС использует для работы отдельный дисковый раздел и не взаимодействует с другими. Таким образом, две различные системы могут быть установлены на одном жестком диске.

Главная загрузочная запись MBR (Master Boot Record) диска содержит место для четырех основных разделов, пронумерованных от 1 до 4. Если необходимо добавить еще разделы на диск, то следует преобразовать основной раздел в дополнительный (extended). Далее дополнительный раздел разделяется на один или несколько логических разделов с номерами от 5 до 15.

3.1.3.1. Расширенные и логические разделы

При установке ОС область страничного обмена чаще всего размещается в основном отдельном дисковом разделе.

Схема, использующая расширенные разделы, позволяет разбивать основной раздел на подразделы. Основной раздел, разбитый таким образом, называется «расширенным разделом», а подразделы называются «логическими разделами». Они функционируют так же, как и основные разделы, различие состоит в схеме их создания.

3.1.3.2. Разбиение жесткого диска

При установке ОС разбиение жесткого диска (дисков) осуществляется средствами программы-установщика. При работе с ОС для разбиения жесткого диска на разделы используется программа `fdisk`.

Каждый раздел должен содержать четное количество секторов, т. к. в ОС используются блоки размером в 1 КБ, т. е. два сектора. Нечетное количество секторов приведет к тому, что последний из них будет не использован. Это ни на что не влияет, но при запуске `fdisk` будет выдано предупреждение.

При изменении размера раздела рекомендуется сначала сделать резервную копию всей необходимой информации, удалить раздел, создать новый раздел, а затем восстановить всю сохраненную информацию в новом разделе.

Описание программы приведено в `man fdisk`.

3.1.3.3. Файлы устройств и разделы

Каждому основному и расширенному разделу соответствует отдельный файл устройства. Существует соглашение для имен подобных файлов, которое заключается в добавлении номера раздела к имени файла самого диска. Разделы с 1 по 4 являются основными (вне зависимости от того, сколько существует основных разделов), а разделы с 5 по 15 — логическими (вне зависимости от того, к какому основному разделу они относятся). Например, `/dev/hda1` соответствует первому основному разделу первого IDE-диска, а `/dev/sdb7` — третьему логическому разделу второго диска с интерфейсом SCSI или SATA.

3.1.4. Форматирование

Форматирование — это процесс записи специальных отметок на магнитную поверхность, которые используются для разделения дорожек и секторов. Диск не может использоваться до тех пор, пока он не будет отформатирован.

Для IDE- и некоторых SCSI-дисков форматирование производится при их изготовлении и, обычно, не требуется повторения этой процедуры.

3.1.5. Программная организация дисковых разделов в RAID и тома LVM

В ядро ОС встроена программная реализация технологии RAID (уровни: `RAID 0`, `RAID 1`, `RAID 5` и их сочетания). Команда `mdadm` предоставляет административный интерфейс пользователя для создания и управления массивами.

После создания массива его устройство, например, `/dev/md0`, используется точно также, как `/dev/hda1` или `/dev/sdb7`.

LVM, с точки зрения ядра системы, использует унифицированные механизмы VFS и не нуждается в специальных конфигурациях ядра. В ОС обеспечивается полнофункциональ-

ное управление томами LVM, которое осуществляется стеком команд управления (около 30 программ).

LVM обеспечивает более высокий уровень абстракции, чем традиционные диски и разделы Linux. Это позволяет добиться большей гибкости при выделении пространства для хранения данных. Логические тома можно легко перемещать с одного физического устройства на другое, а их размер изменять. Физические устройства можно относительно просто добавлять и удалять. Томам, управляемым посредством LVM, можно назначать любые текстовые названия, например, `database` или `home`, а не служебные `sda` или `hda`, как у устройств.

3.2. Управление ФС

Файловая система — это методы и структуры данных, которые используются ОС для хранения файлов на диске или его разделе.

Перед тем, как раздел или диск могут быть использованы для хранения информации (файлов), он должен быть инициализирован, а требуемые данные перенесены на этот диск. Этот процесс называется «созданием ФС».

В ОС рекомендована к применению и используется по умолчанию ФС типа ext4, обеспечивающая поддержку длинных имен, символических связей, хранение мандатных атрибутов, возможность представления имен файлов русскими буквами. Дополнительно могут использоваться ФС ISO9660, FAT (MS-DOS), NTFS и др.

Все данные ОС состоят из множества файлов (программы, библиотеки, системные и пользовательские файлы) и все они располагаются в ФС. Структура ФС имеет вид «перевернутого дерева», верхнюю вершину которого называют корнем («/» — корневой каталог).

В зависимости от выбора, сделанного в процессе установки ОС, каталоги могут относиться к различным ФС.

После установки ОС файловая система может состоять, например, из следующих каталогов:

- `root`:
    - `/bin` — находятся выполняемые программы (точнее, их двоичные файлы). Они необходимы для работы системы. Многие команды ОС являются программами из этого каталога;
    - `/dev` — расположены особые файлы, называемые «файлами устройств» (device files). С их помощью осуществляется доступ ко всем физическим устройствам, установленным в системе;

- /boot — содержит необходимую информацию для загрузки системы (ядро (ядра), образ initrd, файлы загрузчика);
- /root — домашний каталог суперпользователя;
- /tmp — используется для хранения временных файлов, создаваемых программами в процессе своей работы. Работая с программами, создающими много больших временных файлов, лучше иметь отдельную ФС, чем простой каталог корневой ФС;
- /etc — содержит конфигурационные файлы ОС. Здесь находится файл паролей passwd, а также список ФС, монтируемых при начальной загрузке fstab. В этом же каталоге хранятся сценарии загрузки (startup scripts), список узлов (hosts) с их IP-адресами и множество других данных о конфигурации;
- /lib — содержатся разделяемые библиотеки, используемые многими программами во время своей работы. Применяя разделяемые библиотеки, хранящиеся в общедоступном месте, можно уменьшить размер программ за счет повторного использования одного и того же кода;
- /proc — является псевдофайловой системой и используется для чтения из памяти информации о системе;
- /sbin — хранятся системные двоичные файлы (большинство из них используется для нужд системного администрирования);

- /usr — хранятся различные программы и данные, не подлежащие изменению. Каталог /usr и его подкаталоги необходимы для функционирования ОС, т. к. содержат наиболее важные программы. Данный каталог почти всегда является отдельной ФС;
- /var — содержатся изменяемые файлы (такие как log-файлы и др.);
- /home — состоит из личных каталогов пользователей. Общепринято иметь здесь отдельную ФС, чтобы обеспечить пользователям достаточное пространство для размещения своих файлов. Если пользователей в системе много, возможно, придется разделить этот каталог на несколько ФС. Тогда, например, можно создать подкаталоги /home/staff и /home/admin для персонала и администрации, соответственно, установить каждый как самостоятельную ФС и уже в них создавать рабочие каталоги пользователей.

В личных каталогах каждого пользователя наряду с другими файлами имеются несколько конфигурационных файлов, которые для практических целей являются скрытыми. Они модифицируются редко. Файл становится скрытым, если поставить точку в начале имени файла. Увидеть скрытые файлы можно введя команду:

```
ls -a
```

3.2.1. Установка

ФС устанавливается, т. е. инициализируется, при помощи команды `mkfs`. Команда запускает требуемую программу в зависимости от типа устанавливаемой системы. Тип ФС указывается при помощи параметра `-t fstype` (описание команды приведено в `man mkfs`).

3.2.2. Монтирование

Перед началом работы с ФС она должна быть смонтирована. При этом ОС выполняет действия, обеспечивающие функционирование монтируемой ФС. Так как все файлы в ОС принадлежат одной структуре каталогов, то эта операция обеспечивает работу с ФС как с каталогом, называемым точкой монтирования.

Для монтирования ФС к дереву каталогов ОС необходимо убедиться, что каталог (точка монтирования), к которому следует примонтировать ФС, действительно существует.

Если использовать для точки монтирования непустой каталог, то его содержимое станет недоступно до размонтирования. Поэтому рекомендуется иметь специально созданные каталоги для монтирования разделов/устройств. Обычно они располагаются в `/mnt` и `/media`.

Предположим, что требуется монтировать файл `ISO9660` к точке монтирования `/mnt`. Каталог `/mnt` должен уже существовать, иначе монтирование завершится неудачно. После монтирования к каталогу в нем появятся все файлы и подкаталоги ФС. В противном случае каталог `/mnt` будет пустым.

Для того чтобы узнать, какой ФС принадлежит текущий каталог, следует воспользоваться командой:

```
df -h
```

В выводе команды будет отображена ФС и объем свободного пространства.

3.2.2.1. mount

В ОС для монтирования ФС используется команда `mount`. Синтаксис команды:

```
mount <device> <mountpoint>
```

где `<device>` — физическое устройство, которое необходимо примонтировать;

`<mountpoint>` — имя точки монтирования.

По умолчанию в целях обеспечения безопасности информации использовать команду `mount` может только суперпользователь.

Кроме параметров, указанных выше, команда `mount` может иметь в командной строке параметры, приведенные в таблице 4.

Таблица 4

Параметр	Описание
`-f`	Имитирует монтирование ФС. Выполняются все действия, кроме системного вызова для настоящего монтирования
`-v`	Подробный отчет. Предоставляет дополнительную информацию о своих действиях
`-w`	Подключает ФС с доступом для чтения и записи
`-r`	Подключает ФС с доступом только для чтения
`-n`	Выполняет монтирование без записи в файл `/etc/mtab`
`-t type`	Указывает тип монтируемой ФС
`-a`	Подключить все ФС, перечисленные в `/etc/fstab`
`-o list_of_options`	Применить список параметров к монтируемой ФС. Параметры в списке перечислены через запятую. За полным списком возможных параметров следует обратиться к руководству `man`

Если необходимый параметр не указан, `mount` попытается определить ее по файлу `/etc/fstab`.

Распространенные формы команды `mount`:

1) `mount /dev/hdb3 /mnt` — монтирует раздел жесткого диска `/dev/hdb3` к каталогу `/mnt`;

2) `mount -vat nfs` — монтирует все ФС NFS, перечисленные в файле `/etc/fstab`.

Если правильно примонтировать ФС не удается, то воспользоваться командой:

```
mount -vf device mountpoint
```

для получения отчета о результатах выполнения команды `mount`. В данном случае команда выполняет все действия, кроме монтирования, и выводится подробный отчет о каждом шаге выполнения команды.

Описание команды приведено в `man mount`.

3.2.2.2. fstab

Если список используемых ФС изменяется редко, то для удобства можно указать ОС монтировать ФС при загрузке и размонтировать при завершении работы. ФС для монтирования перечисляются в специальном конфигурационном файле `/etc/fstab` по одной в строке. Поля в строках разделяются пробелами или символами табуляции. В таблице 5 приведены поля файла `/etc/fstab`.

Таблица 5

Поле	Описание
ФС	Подключаемое блочное устройство или удаленная ФС

Окончание таблицы 5

Поле	Описание
Точка монтирования	Каталог монтирования ФС. Чтобы сделать систему невидимой в дереве каталогов (например, для файлов подкачки), используется слово `none`
Тип	Указывает тип монтируемой ФС
Опции монтирования	Список разделенных запятыми параметров для монтируемой ФС. Должен содержать, по крайней мере, тип монтирования. Более подробную информацию см. в руководстве `man` команды `mount`
Периодичность резервного копирования	Указывает, как часто следует выполнять резервное копирование с помощью команды `dump`. Если в поле стоит значение 0, то `dump` считает, что ФС не нуждается в резервном копировании
Номер прохода	Задает порядок проверки целостности ФС при загрузке с помощью команды `fsck`. Для корневой ФС следует указывать значение 1, для остальных — 2. Если значение не указано, целостность ФС при загрузке проверяться не будет

Рекомендуется монтировать ФС во время загрузки через `/etc/fstab` вместо команды `mount`. Далее приведен пример файла `fstab`.

Пример

```
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point>   <type>  <options>       <dump>  <pass>
# / was on /dev/sda11 during installation
UUID=a50cefb7-a198-4240-b198-581200027898 / ext4  usrquota,errors=remount-ro,secdel=2 0 1
# /home was on /dev/sda10 during installation
UUID=c94bba8d-95d4-467b-b3e0-2cd7f92c3355 /home  ext4  usrquota,secdelrnd  0  2
# swap was on /dev/sda5 during installation
UUID=ce71b251-2405-4eed-8130-5f92a56b67ac none            swap    sw       0  0
/dev/sr0        /media/cdrom0   udf,iso9660 user,noauto     0       0
```

Комментарии в файле начинаются с символа #.

Слово defaults в поле options указывает, что при подключении ФС следует применить набор параметров по умолчанию, а именно — ФС следует подключить с разрешенным доступом для чтения и записи; она должна рассматриваться как отдельное блочное устройство; весь файловый ввод-вывод должен выполняться асинхронно; разрешено выполнение программных файлов; ФС может подключаться с помощью команды:

```
mount -a
```

биты UID и GID файлов в этой ФС интерпретируются; обычным пользователям не разрешено подключать эту ФС.

Раздел подкачки /dev/sda3 используется ядром ОС для организации виртуальной памяти. Он должен присутствовать в файле /etc/fstab для информирования системы о его местонахождении. Чтобы он не отображался в дереве каталогов, точка подключения указана как none. Кроме того, разделы подкачки подключаются с параметром sw.

Псевдофайловая система /proc указывает на информационное пространство процессов в памяти. Соответствующий физический раздел для нее отсутствует.

ФС VFAT также можно подключать автоматически. Раздел /dev/sdb1 — это первый раздел второго жесткого диска SCSI. Он подключается как раздел VFAT, где vfat указывается в качестве типа ФС и /win — в качестве точки подключения.

Для получения полной информации о допустимых в файле /etc/fstab параметрах см. руководство man для fstab.

3.2.3. Размонтирование

Для размонтирования ФС используется команда umount. Размонтирование может понадобиться для проверки и восстановления ФС с помощью команды fsck. Удаленные ФС размонтируются в случае неполадок в сети.

Команда umount имеет следующий синтаксис:

```
umount <device>
umount <mountpoint>
umount -a
umount -t <fstype>
```

где <device> — физическое устройство, которое необходимо размонтировать;

<mountpoint> — имя точки монтирования;

параметр -a — размонтирует все ФС;

параметр -t — размонтирует только ФС указанного типа;

<fstype> — тип ФС.

Команда umount не размонтирует ФС, если они используются в текущий момент. Например, если какую-либо ФС смонтировать в /mnt и выполнить команды:

```
cd /mnt
umount /mnt
```

то появится сообщение об ошибке, т. к. ФС занята. Перед размонтированием `/mnt` необходимо перейти в каталог другой ФС.

Для принудительного размонтирования устройства, независимо от его использования, можно воспользоваться параметром `-f` команды `umount`:

```
umount -f /cdrom
```

Для размонтирования и извлечения из устройств сменных носителей информации используется команда `eject`.

Служебная программа `fuser` отображает сведения о процессах, использующих ФС. Например:

```
fuser -v  точка_монтирования
```

Для завершения всех процессов, использующих ФС, можно воспользоваться командой:

```
fuser -km точка_монтирования
```

Описание команд приведено в `man umount` и `man fuser`.

3.3. Управление пользователями

3.3.1. Работа с пользователями

Управление пользователями заключается в добавлении и удалении пользователей, а также в определении их привилегий и предусматривает:

- добавление имен пользователей для возможности их работы в системе;
- создание или изменение паролей пользователей;
- определение их привилегий;
- создание и назначение рабочих каталогов;
- определение групп пользователей;
- удаление имен пользователей.

Каждый пользователь должен иметь уникальное регистрационное имя, дающее возможность идентифицировать пользователя и избежать ситуации, когда один пользователь может стереть файлы другого. Кроме того, каждый пользователь должен иметь свой пароль для входа в систему.

3.3.1.1. Добавление

При добавлении пользователя в файл `/etc/passwd` вносится учетная запись в форме:

```
login_name: encrypted_password: user_ ID: group_ ID: user_ information:
login_directory: login_shell
```

В данной записи поля разделены двоеточиями, а значения этих полей приведены в таблице 6.

Таблица 6

Поле	Назначение
login_name	Регистрационное имя пользователя
encrypted_password	Указатель на теневой файл паролей (shadow)
user_ID	Уникальный номер, используемый ОС для идентификации пользователя. Для локальных пользователей не должен превышать 2499
group_ID	Уникальный номер или имя, используемые для идентификации первичной группы пользователя. Если пользователь является членом нескольких групп, он может (если это разрешено системным администратором) в процессе работы менять группу
user_information	Описание пользователя, например, его имя и должность
login_directory	Рабочий каталог пользователя (в котором он оказывается после входа в систему)
login_shell	Оболочка, используемая пользователем, после входа в систему (например, /bin/bash)

Также описание файла /etc/passwd приведено в man 5 passwd.

Для добавления пользователя применяется команда adduser с именем добавляемого пользователя в качестве параметра, например:

```
adduser User1
```

Команда adduser добавляет пользователя, создает домашний каталог, создает почтовый ящик, а также копирует файлы, имена которых начинаются с точки, из каталога /etc/skel в рабочий каталог пользователя. Каталог /etc/skel должен содержать все файлы-шаблоны, которые имеет каждый пользователь. Обычно это персональные конфигурационные файлы, такие как .profile, .cshrc и .login, для настройки оболочки. Команда adduser представляет собой файл сценария bash, находящийся в каталоге /usr/sbin. Можно добавить запрос дополнительной информации о пользователе. Чтобы это сделать, необходимо воспользоваться командой chfn для изменения стандартных записей о пользователе.

Описание команд приведено в man adduser и man chfn.

ВНИМАНИЕ! Для обеспечения нормальной работы пользователя с сетевыми службами в системе должна быть явно задана его классификационная метка (диапазон уровней конфиденциальности и категорий конфиденциальности) при помощи утилиты usermac или fly-admin-smc, даже если ему недоступны уровни и категории выше 0.

3.3.1.2. Установка пароля

Для установки пароля пользователя предназначена команда `passwd`. Необходимо определить пароли для каждого пользователя. Войдя в систему, пользователь сможет сам изменить свой пароль. Для установки пароля пользователя выполнить следующее:

1) ввести команду и регистрационное имя пользователя, например:

```
passwd User1
```

и нажать клавишу **<Enter>**;

2) после появления приглашения:

```
New password:
```

ввести пароль (он не будет отображаться на экране монитора);

3) после появления сообщения повторить ввод пароля еще раз, ввести его снова.

Пароль будет зашифрован и внесен в файл `/etc/shadow`. При выборе пароля необходимо учесть следующие правила: пароль должен иметь не менее шести символов (предпочтительно — восемь символов) и желательно, чтобы пароль содержал как прописные, так и строчные буквы, знаки препинания и цифры.

ВНИМАНИЕ! Пароль рекомендуется создавать способом, максимально затрудняющем его подбор. Наиболее безопасный пароль состоит из случайной (псевдослучайной) последовательности букв, знаков препинания, специальных символов и цифр.

Необходимо периодически изменять пароль.

После выполнения всех действий запись в файле будет выглядеть примерно так:

```
anna:x:123:121:Anna_M.:/home/anna:/bin/bash
```

Второе поле записи содержит пароль в зашифрованном виде.

Описание команды приведено в `man passwd`.

Примечание. Если пользователь забыл свой пароль, то администратор системы не может напомнить его пользователю, т. к. в явном виде пароль нигде не хранится. Поэтому действия по восстановлению доступа пользователя в систему сводятся к замене администратором пароля пользователя на новый пароль с помощью команды:

```
passwd user_name
```

3.3.1.3. Удаление

Есть несколько степеней удаления пользователя:

- лишение пользователя возможности входа в систему;
- удаление учетной записи;
- удаление пользователя и всех его файлов.

Лишение пользователя возможности входа в систему полезно в случае его длительного перерыва в работе.

На время отсутствия пользователя можно заблокировать его запись с помощью команды:

```
usermod -L user_name
```

При этом все пользовательские файлы и каталоги остаются нетронутыми, но войти в систему под его именем становится невозможно.

Для разблокировки записи необходимо выполнить команду:

```
usermod -U user_name
```

Одним из вариантов лишения пользователя возможности входа в систему может быть смена имени пользователя. При этом вход под старым именем становится невозможным. Для этого необходимо выполнить команду:

```
usermod -l new_user_name  old_user_name
```

Примечание. Имена домашнего каталога и почтового ящика при таком изменении имени пользователя не меняются. Эти параметры должны быть изменены вручную.

Удаление учетной записи пользователя производится либо путем непосредственного редактирования файла `/etc/passwd`, либо с помощью команды:

```
deluser user_name
```

По умолчанию учетная запись удаляется без удаления домашнего каталога и файлов системы, принадлежащих удаляемому пользователю. Для удаления домашнего каталога может использоваться дополнительный параметр `--remove-home`, а для поиска и удаления всех файлов системы, принадлежащих удаляемому пользователю, — параметр `--remove-all-files`.

Также удаление пользователя, его домашнего каталога и файлов системы могут быть выполнены вручную с помощью следующих команд:

1) для полного удаления пользователя и всех его файлов из системы выполнить команду:

```
find / -user user_name -exec rm -r {} \;
```

2) для удаления рабочего каталога пользователя выполнить команду:

```
rmdir user_home_dir
```

3) удалить запись о пользователе из файла `/etc/passwd`;

4) для удаления файлов, не принадлежащих ни одному пользователю в системе, выполнить команду:

```
find / -nouser -exec rm -r {} \;
```

Описание команд приведено в `man usermod`, `man deluser` и `man find`.

3.3.1.4. Неудачный вход в систему

Команда `faillog` показывает содержимое журнала неудачных попыток (файл `/var/log/faillog`) входа в систему. Также она может быть использована для управления счетчиком неудачных попыток и их ограничением. При запуске `faillog` без параметров

выводятся записи `faillog` только тех пользователей, у которых имеется хотя бы одна неудачная попытка входа.

Предельное число попыток входа для каждой учетной записи равно 10. Для сброса неудачных попыток входа необходимо пользоваться параметром `-r`.

Описание команды, а также файла `/var/log/faillog` приведено в `man faillog` и `man 5 faillog`.

3.3.2. Работа с группами

Каждый пользователь является членом группы. Различным группам можно назначить различные возможности и привилегии.

Информация о группах содержится в файле `/etc/group` в следующем формате:

```
Admin :: 21: user1, user2, user3
```

где `Admin` — имя группы, `21` — идентификатор, `user1`, `user2`, `user3` — члены группы. Пользователь может состоять в нескольких группах и переходить из одной в другую в процессе работы.

Описание файла `/etc/group` приведено в `man 5 group`.

3.3.2.1. Добавление

Добавление группы производится с помощью команды:

```
addgroup users
```

Данная команда добавляет группу `users`.

Также новую группу можно создать путем непосредственного редактирования файла `/etc/group` и ввода необходимой информации о группе.

ВНИМАНИЕ! Каждой группе присваивается уникальный идентификационный номер и ОС при работе учитывает номер группы, а не имя. Поэтому, если присвоить двум группам одинаковый номер, ОС будет воспринимать две группы как одну и ту же.

Описание команды приведено в `man addgroup`.

3.3.2.2. Удаление

Удаление группы производится с помощью команды:

```
delgroup users
```

Данная команда удаляет группу `users`.

Также удаление группы производится путем удаления записи о ней в файле `/etc/group`.

Описание команды приведено в `man delgroup`.

3.3.3. Рабочие каталоги пользователей

Рабочие каталоги пользователей на одном компьютере следует размещать в отдельном каталоге верхнего уровня (по умолчанию — `/home`). Если пользователей много,

то оптимально разделить их домашние каталоги по группам (подразделениям), например, `/home/hr` (отдел персонала) `/home/admins`, `/home/buhg` и т. д.).

Таким образом, рабочие каталоги будут логически сгруппированы, что в дальнейшем облегчит администрирование системы.

3.4. Перезагрузка и останов

Перезагрузка необходима в следующих случаях:

1) при подключении нового устройства или если работающее устройство «зависает» и его невозможно сбросить;
2) при модификации файла конфигурации, который используется только при начальной загрузке, т. к. для того чтобы изменения вступили в силу, необходимо загрузить систему заново;
3) если система «не отвечает» и невозможно зарегистрироваться и определить причину ошибки.

Перезагрузку можно выполнить несколькими способами:

1) дать команду `shutdown`;
2) использовать команду `reboot`;
3) использовать команду `init 6`.

Выключение системы предполагает корректное выключение системы, позволяющее избежать потерь информации и сбоев ФС.

Выключение системы можно выполнить несколькими способами:

1) выключить питание;
2) дать команду `shutdown`;
3) использовать команду `halt`;
4) использовать команду `init 0`.

Работая с ОС, следует соблюдать аккуратность при выходе из системы. Нельзя просто выключить компьютер, т. к. ОС хранит информацию ФС в оперативной памяти и при отключении питания информация может быть потеряна, а ФС повреждена.

Выключение питания может привести не только к потере данных и повреждению системных файлов, есть риск повредить жесткий диск, если он относится к числу тех, на которых перед отключением питания необходимо установить в соответствующее положение защитный переключатель либо провести парковку головок.

3.4.1. shutdown

Команда `shutdown` — самый безопасный и наиболее корректный способ инициирования останова, перезагрузки или возврата в однопользовательский режим.

Можно дать указание shutdown делать паузу перед остановом системы. Во время ожидания она посылает зарегистрированным пользователям через постепенно укорачивающиеся промежутки времени сообщения, предупреждая их о приближающемся останове. По умолчанию в сообщениях говорится о том, что система заканчивает работу, и указывается время, оставшееся до останова. При желании администратор может добавить собственное короткое сообщение, в котором содержится информация о том, почему система останавливается, и сколько примерно времени потребуется ожидать, прежде чем пользователи вновь смогут войти в систему.

Команда shutdown позволяет указать, что конкретно должен сделать компьютер: остановиться, перейти в однопользовательский режим или перезагрузиться. Иногда можно также указать, следует ли перед перезагрузкой проверить диски с помощью команды fsck.

Синтаксис команды:

```
shutdown [flags] time [warning-message]
```

где [warning-message] — сообщение, посылаемое всем пользователям, в настоящий момент зарегистрированным в системе, а time — время выполнения отключения системы. Значение может быть также задано в формате +m, где m — количество минут ожидания до остановки системы. Значение +0 может быть заменено словом now.

В таблице 7 перечислены основные параметры команды shutdown.

Таблица 7

Параметр	Назначение
-k	Послать предупреждение без реального завершения работы системы
-r	Перезагрузка компьютера после завершения работы
-h	Отключение компьютера после завершения работы
-n	Не синхронизировать диски. Этот параметр следует использовать крайне осторожно, т. к. могут быть потеряны или повреждены данные
-f	«Быстрая» перезагрузка. Создается файл /etc/fastboot, при наличии которого во время загрузки ОС не запускается программа fsck
-c	Отказаться от уже запущенного процесса завершения работы. Параметр time при этом не может быть использован

Описание команды приведено в man shutdown.

Команда shutdown посылает всем пользователям предупреждающее сообщение, затем ожидает определенное в командной строке время и посылает всем процессам сигнал SIGTERM. Затем вызывается команда halt или reboot — в зависимости от параметров командной строки.

3.4.2. halt и reboot

Команда `halt` выполняет все основные операции, необходимые для останова системы. Для вызова этой команды можно в командной строке указать:

```
shutdown -h
```

или непосредственно `halt`, которая регистрирует останов, уничтожает несущественные процессы, осуществляет системный вызов `sync`, дожидается завершения операций записи ФС, а затем прекращает работу ядра.

При указании `halt -n` вызов `sync` подавляется. Эта команда используется после исправления корневого раздела программой `fsck` для того, чтобы ядро не могло затереть исправления старыми версиями суперблока. Команда `halt -q` инициирует почти немедленный останов, без синхронизации, уничтожения процессов и записи в файлы регистрации. Этот флаг используется редко.

Команда `reboot` почти идентична команде `halt`. Различие заключается в том, что компьютер перезагружается с нуля, а не останавливается. Команда `reboot` вызывается командой:

```
shutdown -r
```

Описание команд приведено в `man halt` и `man reboot`.

4. СИСТЕМНЫЕ СЛУЖБЫ, СОСТОЯНИЯ И КОМАНДЫ

4.1. Системные службы

Службы — это специальные программы, выполняющие различные служебные функции. Обычно службы запускаются автоматически при наступлении определенного события (например, при загрузке ОС) и выполняются в фоновом режиме. В среде ОС для управления службами, точками монтирования и т. п. применяется системный менеджер systemd. Менеджер systemd обеспечивает параллельный запуск служб в процессе загрузки ОС, использует сокеты и активацию D-Bus для запускаемых служб, предлагает запуск демонов по необходимости, отслеживает запуск служб, поддерживает мгновенные снимки и восстановление состояния системы, монтирование и точки монтирования, а также внедряет основанную на зависимостях логику контроля процессов сложных транзакций.

Менеджер systemd оперирует специально оформленными файлами конфигурации — юнитами (unit). Каждый юнит отвечает за конкретную служб (`*.service`), точку монтирования (`*.mount`), устройство (`*.device`), файл подкачки (`*.swap`), сокет (`*.socket`) и т. д.

Отличительной особенностью systemd является использование контрольных групп Linux, обеспечивающих иерархическую структуризацию служб: любая запущенная служба помещается в отдельную контрольную группу с уникальным идентификатором. Когда служба запускает другую зависимую службу, то она автоматически включается в группу с тем же идентификатором. При этом непривилегированные службы не могут изменить свое положение в иерархии. При штатном завершении работы службы будут завершены и все зависимые от нее службы.

Описание использования менеджера systemd для управления доступом приведено в РУСБ.10152-02 97 01-1.

4.1.1. Общие сведения

Существует два механизма управления службами: systemV (сценарии, не являющиеся юнитами, в каталогах `/etc/init.d`, `/etc/rc{0-6,S}.d`) — устаревший, но сохраненный для обеспечения совместимости, и systemd (юниты в каталогах `/etc/systemd/system`, `/run/systemd/system`, `/lib/systemd/system`, а также в пользовательских каталогах) — современный механизм.

Таким образом, администраторам ОС доступны два инструмента для управления службами:

1) `/usr/sbin/service` (команда `service`) — устаревший инструмент, работающий только с службами, сценарии управления которых находятся в каталоге `/etc/init.d`;

2) /bin/systemctl (команда systemctl) — современный инструмент для управления всеми службами.

Оба эти инструмента обеспечивают интерфейс пользователя с юнитами (сценариями). Юниты (сценарии) в свою очередь обеспечивают интерфейс управления службами, предоставляя пользователю параметры для запуска, остановки, перезапуска, запроса состояния, а также для других действий со службой.

Сценарии systemV могут иметь произвольный набор параметров управления, поэтому предусмотрена возможность проверить доступные параметры с помощью команды service. Например, для службы syslog команда и результат ее работы будут выглядеть так:

```
/usr/sbin/service syslog
[info] Usage: /etc/init.d/syslog {start|stop|status|restart|reload|force-reload}.
```

Юниты systemd имеют фиксированный набор параметров, оформленных в виде параметров команды systemctl (start, stop, reload, restart и т.д.). Размещаются юниты в одном из каталогов:

- /usr/lib/systemd/system/ — юниты из установленных пакетов;
- /run/systemd/system/ — юниты, созданные в режиме рантайм. Данные юниты имеют приоритет выше, чем юниты из установленных пакетов;
- /etc/systemd/system/ — юниты, созданные и управляемые администратором. Данные юниты имеют приоритет выше, чем юниты, созданные в режиме рантайм.

Команда service выводит информацию только о службах, сценарии которых находятся в каталоге /etc/init.d. Проверить текущее состояние служб можно с помощью параметра --status-all команды service:

```
usr/sbin/service --status-all
[ + ]  acpi-support
[ + ]  acpid
[ - ]  anacron
...
```

Для получения полной информации, отслеживания и контроля состояния юнитов и менеджера systemd используется утилита командной строки systemctl:

```
systemctl -t service -a
UNIT                    LOAD      ACTIVE    SUB     DESCRIPTION
acpi-support.service    loaded    active    exited  LSB: Start some power...
? apache2.service       masked    inactive  dead    apache2.service
? apparmor.service      not-found inactive  dead    apparmor.service
assistant.service       loaded    active    running Assistant remote control...
...
```

Для просмотра списка установленных юнитов выполнить команду:

```
systemctl list-unit-files
```

Для просмотра списка запущенных юнитов выполнить команду:

```
systemctl list-units
```

или для просмотра списка запущенных юнитов определенного типа использовать данную команду с параметром `-t <тип_юнита>`:

```
systemctl list-units -t service
```

Основные параметры для использования с инструментом командной строки `systemctl` приведены в таблице 8.

Таблица 8

Параметр	Описание
`systemctl start <юнит>`	Незамедлительно запустить юнит
`systemctl stop <юнит>`	Незамедлительно остановить юнит
`systemctl restart <юнит>`	Перезапустить юнит
`try-restart <юнит>`	Перезапустить (не запускать неработающие) юниты
`systemctl reload <юнит>`	Перезагрузить настройки юнита
`systemctl status`	Вывести общую информацию о состоянии системы и список юнитов, которым соответствуют запущенные процессы. При запуске команды с именем юнита будет выведена информация о статусе данного юнита
`systemctl cat <юнит>`	Показать содержимое юнита
`systemctl is-enabled <юнит>`	Проверить включение юнита в автозапуск при загрузке системы
`systemctl enable <юнит>`	Добавить юнит в автозапуск при загрузке системы
`systemctl disable <юнит>`	Удалить юнит из автозапуска при загрузке системы
`systemctl mask <юнит>`	Маскировать юнит для исключения возможности его запуска
`systemctl unmask <юнит>`	Снять маску юнита
`systemctl help <юнит>`	Показать страницу руководства `man` юнита (при наличии поддержки данной функции для указанного юнита)
`systemctl daemon-reload`	Перезагрузить systemd для поиска новых или измененных юнитов
`systemctl --failed`	Показать список юнитов, которые не были запущены из-за ошибки
`isolate <юнит или цель>`	Если указано имя юнита, то запускает этот юнит и все его зависимости, остановив все остальные службы. Если указано имя целевого состояния выполнения, то переводит систему в указанное состояние выполнения (имя состояния указывается без расширения `.target`)

4.1.2. Конфигурационные файлы systemd

При использовании менеджера systemd возможно как корректировать существующие юниты, так и создавать новые.

Юнит представляет собой `ini`-подобный файл, имя которого состоит из имени юнита и суффикса, определяющего тип юнита. В общем случае юнит-файл включает секции `[Unit]` и `[Install]`, а также дополнительные секции, соответствующие конкретному типу юнита.

Секция `[Unit]` содержит описание юнита, а также информацию о зависимостях при запуске юнита:

- `Description=` — описание юнита;
- `Wants=` — зависимость требования запуска. Требование исходного юнита запустить юнит, указанный в параметре. При этом результат запуска юнита, указанного в параметре, не влияет на запуск исходного юнита. При отсутствии параметров `After=` и `Before=` юниты будут запущены одновременно;
- `Requires=` — зависимость требования запуска. Требование исходного юнита запустить юнит, указанный в параметре. При этом ошибка запуска юнита, приведенного в параметре, приведет к ошибке запуска исходного юнита. При отсутствии параметров `After=` и `Before=` юниты будут запущены одновременно;
- `After=` — зависимость порядка запуска. Дополнительный, но не обязательный параметр к параметрам `Wants=` и `Requires=`, указывающий на необходимость запуска исходного юнита только после запуска юнита, указанного в параметре. При этом если данный параметр используется с параметром `Wants=`, то исходный юнит будет запущен вне зависимости от результата запуска юнита, указанного в параметре;
- `Before=` — аналогичен параметру `After=`, только определяет запуск исходного юнита до запуска юнита, указанного в параметре.

Секция `[Install]` содержит информацию об установке юнита. Используется командами `systemctl enable <юнит>` и `systemctl disable <юнит>`. Может содержать следующие параметры:

- `Alias=` — список альтернативных имен юнита, разделенных пробелом. Имена должны иметь тот же суффикс, что и имя файла юнита. При использовании команды `systemctl enable` будут созданы символические ссылки из перечисленных имен на данный юнит.

ВНИМАНИЕ! Не все типы юнитов могут иметь альтернативные имена. Для типов `*.mount`, `*.slice`, `*.swap` и `*.automount` данный параметр не поддерживается;

- WantedBy= — указывает на целевое состояние (см. 4.2), при котором запускается данный юнит. При использовании команды systemctl enable будет добавлена символическая ссылка в <имя_состояния>.target;
- Also= — определяет список юнитов, которые также будут добавлены в автозапуск или удалены из автозапуска вместе с данным юнитом.

Секция [Service] в юните службы содержит следующие параметры:

1) Type= — определяет тип запуска службы:
   а) simple — используется по умолчанию. Служба будет запущена незамедлительно. Процесс при этом не должен разветвляться. Не рекомендуется использовать данный тип, если другие службы зависят от очередности при запуске данной службы. Исключение — активация сокета;
   б) forking — служба запускается однократно и процесс разветвляется с завершением родительского процесса. Рекомендуется использовать данный тип для запуска классических демонов. Потребуется также определить PIDFile, чтобы менеджер systemd мог отслеживать основной процесс;
   в) oneshot — используется для скриптов, которые завершаются после выполнения одного задания;
   г) notify — аналогичен типу simple, но дополнительно демон отправит менеджеру systemd сигнал о своей готовности;
   д) dbus — служба находится в состоянии готовности, когда определенное BusName появляется в системной шине DBus;
   е) idle — менеджер systemd отложит выполнение службы до момента отправки всех заданий;
2) PIDFile= — расположение pid-файла;
3) WorkingDirectory= — рабочий каталог приложения;
4) User= — пользователь, от имени которого будет запущена служба;
5) Group= — группа, от имени которой будет запущена служба;
6) OOMScoreAdjust= — приоритет завершения процесса при нехватке памяти, где 1000 — максимальное значение, означающее полный запрет на завершение процесса;
7) ExecStop= — указывает на скрипт, который должен быть выполнен перед остановкой службы;
8) ExecStart — указывает на команду, которая должна быть выполнена после запуска службы;
9) RemainAfterExit — предписывает systemd считать процесс активным после его завершения.

Секция `[Socket]` в юните сокета определяет следующие параметры для управления сокетом:

- `ExecStart=` — правило запуска;
- `ExecReload=` — правило перезапуска;
- `KillMode=` — правило завершения;
- `Restart=` — правило перезапуска при возникновении ошибки.

4.2. Системные (целевые) состояния

В systemd уровни запуска файлов реализованы в виде сгруппированных юнитов, представляющих целевое состояние (цель). Файлы, определяющие целевые состояния, хранятся в каталоге `/lib/systemd/system/` и имеют расширение имени `.target`. Для совместимости в ОС сохранено понятие «уровней выполнения». В стандартно установленной системе предусмотрено наличие шести системных уровней выполнения, каждому из которых соответствует целевое состояние.

Одна из целей назначается в качестве состояния по умолчанию, в которое переходит система после включения. В стандартно установленной ОС состоянием по умолчанию является `graphical.target` (уровень выполнения 5) — многопользовательский режим с графической оболочкой. Уровням выполнения 2, 3 и 4 соответствует цель `multi-user.target` (многопользовательский режим без графической оболочки), а целям `poweroff.target` (уровень выполнения 0) и `reboot.target` (уровень выполнения 6) соответствуют выключение и перезагрузка системы соответственно.

Проверить список соответствия состояний и уровней выполнения можно командой:

```
ls -la /lib/systemd/system/runlevel*
```

```
lrwxrwxrwx 1 ... /lib/systemd/system/runlevel0.target -> poweroff.target
lrwxrwxrwx 1 ... /lib/systemd/system/runlevel1.target -> rescue.target
lrwxrwxrwx 1 ... /lib/systemd/system/runlevel2.target -> multi-user.target
lrwxrwxrwx 1 ... /lib/systemd/system/runlevel3.target -> multi-user.target
lrwxrwxrwx 1 ... /lib/systemd/system/runlevel4.target -> multi-user.target
lrwxrwxrwx 1 ... /lib/systemd/system/runlevel5.target -> graphical.target
lrwxrwxrwx 1 ... /lib/systemd/system/runlevel6.target -> reboot.target
```

Каждая цель имеет собственное имя вида `<имя_состояния>.target` и предназначена для конкретных задач. Одновременно могут быть активны несколько целей. Цели могут наследовать все службы других целей, добавляя к ним свои. В systemd также имеются цели, имитирующие общие уровни выполнения SystemVinit, поэтому для переключения между целевыми юнитами можно использовать команду:

```
telinit RUNLEVEL
```

Для определения доступных целевых состояний используется команда:

```
systemctl list-unit-files --type=target
```

Для определения активных целевых состояний используется команда:

```
systemctl list-units --type=target
```

Для перехода в целевое состояние используется команда:

```
systemctl isolate <имя_состояния>.target
```

Данная команда изменят только текущий уровень выполнения и ее действие не повлияет на последующие загрузки системы.

Для просмотра целевого состояния по умолчанию, которое systemd использует сразу после загрузки системы, используется команда:

```
systemctl get-default
```

Для просмотра дерева зависимостей юнитов от цели выполнить команду:

```
systemctl list-dependencies <имя_состояния>.target
```

Для проверки заданного по умолчанию состояния системы выполнить команду:

```
systemctl get-default
graphical.target
```

Для проверки соответствующего уровня выполнения выполнить команду:

```
sudo runlevel
N 5
```

Для изменения состояния системы, заданного по умолчанию, выполнить команду:

```
sudo systemctl set-default multi-user.target
Created symlink /etc/systemd/system/default.target ->
/lib/systemd/system/multi-user.target.
```

После изменения состояния, заданного по умолчанию, система будет переведена в него после перезагрузки. Для принудительного перевода системы в нужное состояние без перезагрузки используется команда `systemctl` с параметром `isolate` и именем целевого состояния (имя указывается без расширения `.target`):

```
sudo systemctl isolate multi-user
```

или команда `init`:

```
sudo init 3
```

Обе команды переведут систему в состояние `multi-user` (многопользовательский режим без графической оболочки), что соответствует третьему уровню выполнения. При этом будут запущены/остановлены все службы, указанные в соответствующем описании состояния.

Для обеспечения совместимости с более ранними реализациями помимо запуска/остановки юнитов, определенных в файлах `.target`, при переводе системы в другое состояние исполнения systemd проверяет все файлы управления службами, имеющиеся в соответствующем целевому уровню выполнения каталоге `/etc/rc{0-6}.d/`, и запускает/останавливает соответствующие этим файлам собственные юниты или, если

соответствующий юнит не обнаружен, автоматически генерирует юнит из файла управления и выполняет его.

Подробное описание данных команд и служб приведено на страницах руководства man.

4.3. Системные команды

Основные системные команды ОС приведены в таблице 9.

Таблица 9

Команда	Назначение
addgroup	Создание новой учетной записи группы
adduser	Создание новой учетной записи пользователя
ar	Создание и работа с библиотечными архивами
at	Формирование или удаление отложенного задания
awk	Язык обработки строковых шаблонов
bc	Строковый калькулятор
chfn	Управление информацией учетной записи пользователя (имя, описание)
chsh	Управление выбором командного интерпретатора (по умолчанию — для учетной записи)
cut	Разбивка файла на секции, задаваемые контекстными разделителями
delgroup	Удаление учетной записи группы
deluser	Удаление учетной записи пользователя и соответствующих файлов окружения
df	Вывод отчета об использовании дискового пространства
dmesg	Вывод содержимого системного буфера сообщений
du	Вычисление количества использованного пространства элементов ФС
echo	Вывод содержимого аргументов на стандартный вывод
egrep	Поиск в файлах содержимого согласно регулярных выражений
fgrep	Поиск в файлах содержимого согласно фиксированных шаблонов
file	Определение типа файла
find	Поиск файла по различным признакам в иерархии каталогов
gettext	Получение строки интернационализации из каталогов перевода
grep	Вывод строки, содержащей шаблон поиска
groupadd	Создание новой учетной записи группы
groupdel	Удаление учетной записи группы
groupmod	Изменение учетной записи группы
groups	Вывод списка групп
gunzip	Распаковка файла
gzip	Упаковка файла

Продолжение таблицы 9

Команда	Назначение
hostname	Вывод и задание имени хоста
install	Копирование файла с установкой атрибутов
ipcrm	Удаление ресурса IPC
ipcs	Вывод характеристик ресурса IPC
kill	Прекращение выполнения процесса
killall	Удаление процессов по имени
lpr	Система печати
ls	Вывод содержимого каталога
lsb_release	Вывод информации о дистрибутиве
mknod	Создание файла специального типа
mktemp	Генерация уникального имени файла
more	Постраничный вывод содержимого файла
mount	Монтирование ФС
msgfmt	Создание объектного файла сообщений из файла сообщений
newgrp	Смена идентификатора группы
nice	Изменение приоритета процесса перед его запуском
nohup	Работа процесса после выхода из системы
od	Вывод содержимого файла в восьмеричном и других видах
passwd	Смена пароля учетной записи
patch	Применение файла описания изменений к оригинальному файлу
pidof	Вывод идентификатора процесса по его имени
ps	Вывод информации о процессах
renice	Изменение уровня приоритета процесса
sed	Строковый редактор
sendmail	Транспорт системы электронных сообщений
sh	Командный интерпретатор
shutdown	Команда останова системы
su	Изменение идентификатора запускаемого процесса
sync	Сброс системных буферов на носители
tar	Файловый архиватор
umount	Размонтирование ФС
useradd	Создание новой учетной записи пользователя или обновление существующей
userdel	Удаление учетной записи пользователя и соответствующих файлов окружения
usermod	Модификация информации об учетной записи пользователя
w	Список пользователей, работающих в настоящий момент в системе, и ресурсов, с которым осуществляется работа

Окончание таблицы 9

Команда	Назначение
who	Вывод списка пользователей системы

Описание команд приведено на страницах руководства man.

4.3.1. Планирование запуска команд

4.3.1.1. at

Для запуска одной или более команд в заранее определенное время используется команда at. В ней можно определить время и/или дату запуска той или иной команды. Команда at требует двух (или большего числа) параметров. Как минимум, следует указать время запуска и какая команда должна быть запущена.

Команды для запуска с помощью команды at вводятся как список в строках, следующих за ней. Ввод каждой строки завершается нажатием клавиши **<Enter>**. По окончании ввода всей команды нажать клавиши **<Ctrl+D>** для ее завершения.

Примеры:

1. Запустить команды lpr /usr/sales/reports/. и echo "Files printed" в 8:00

```
at 8:00
lpr /usr/sales/reports/.
echo "Files printed"
```

После ввода всей команды отобразится следующая запись:

```
job 756603300.a at Tue Jul 8 08:00:00 2014
```

означающая, что указанные команды будут запущены в 8:00, идентификатор задания 756603300.a (может понадобится, если необходимо отменить задание командой at -d)

В результате выполнения команды в 8:00 будут распечатаны все файлы каталога /usr/sales/reports, и пользователю будет выведено сообщение на экран монитора.

2. Для запуска всех команд, перечисленных в файле getdone, в 17:30 следует воспользоваться одной из двух форм команды at:

```
at 17:30 < getdone
```

или

```
at 10:30 -f getdone
```

Обе приведенные команды эквивалентны. Разница заключается в том, что в первой команде используется механизм перенаправления потоков ввода-вывода, во второй команде — дисковый файл.

Кроме времени в команде at может быть определена дата.

Пример

```
at 10:00 Jul 14
lp /usr/sales/reports/
echo "Files printed"
```

Задания, определяемые администратором системы, помещаются в очередь, которую ОС периодически просматривает. Администратору необязательно находиться в системе для того, чтобы at отработала задания. В данном случае команда работает в фоновом режиме.

Для просмотра очереди заданий ввести:

```
at -l
```

Если предыдущие примеры были запущены, то будет выведено:

```
job 756603300.a at Sat Jul 8 08:00:00 2014 job 756604200.a at Sat Jul 14
  17:00:00 2014
```

Администратор системы видит только свои задания по команде at.

Для удаления задания из очереди следует запустить at с параметром -d и номером удаляемого задания:

```
at -d 756604200.a
```

В таблице 10 показаны варианты использования команды at.

Таблица 10

Формат команды	Назначение
at hh:mm	Выполнить задание во время hh:mm в 24-часовом формате
at hh:mm месяц день год	Выполнить задание во время hh:mm в 24-часовом формате в соответствующий день
at -l	Вывести список заданий в очереди; псевдоним команды — atq
at now+count time-units	Выполнить задание через определенное время, которое задано параметром count в соответствующих единицах — неделях, днях, часах или минутах
at -d job_ID	Удалить задание с идентификатором job_ID из очереди; псевдоним команды — atrm

Администратор системы может применять все эти команды. Для других пользователей права доступа к команде at определяются файлами /etc/at.allow и /etc/at.deny. Если существует файл /etc/at.allow, то применять команду at могут только перечисленные в нем пользователи. Если же такого файла нет, проверяется наличие файла /etc/at.deny, в котором отражено, кому запрещено пользоваться командой at. Если ни одного файла нет, значит, команда at доступна только суперпользователю.

Подробное описание команды приведено в man at.

4.3.1.2. cron

Для регулярного запуска команд в ОС существует команда `cron`. Администратор системы определяет для каждой программы время и дату запуска в минутах, часах, днях месяца, месяцах года и днях недели.

Команда `cron` запускается один раз при загрузке системы. Отдельные пользователи не должны иметь к ней непосредственного доступа. Кроме того, запуск `cron` никогда не осуществляется вручную путем ввода имени программы в командной строке, а только из сценария загрузки ОС.

При запуске `cron` проверяет очередь заданий команды `at` и задания пользователей в файлах `crontab`. Если команд для запуска нет, `cron` «засыпает» на одну минуту и затем вновь приступает к поискам команды, которую следует запустить в этот момент. Большую часть времени команда `cron` проводит в «спящем» состоянии, и для ее работы используется минимум системных ресурсов.

Чтобы определить список заданий для `cron` используется команда `crontab`. Для каждого пользователя с помощью данной команды создается файл `crontab` со списком заданий, находящийся в каталоге `/var/spool/cron/crontabs` и имеющий то же имя, что и имя пользователя.

Примечание. Пользователи, которым разрешено устанавливать задания командой `cron`, перечислены в файле `/etc/cron.allow`. Файл заданий для команды `cron` можно создать с помощью обычного текстового редактора, но при этом нельзя просто заменить им существующий файл задания в каталоге `/var/spool/cron/crontabs`. Для передачи `cron` сведений о новых заданиях обязательно должна использоваться команда `crontab`.

Каждая строка в файле `crontab` содержит шаблон времени и команду. Можно создать любое количество команд для `cron`. Команда выполняется тогда, когда текущее время соответствует приведенному шаблону. Шаблон состоит из пяти частей, разделенных пробелами или символами табуляции.

Синтаксис команд в файле `crontab`:

```
минуты часы день_месяца месяц_года день_недели задание
```

Первые пять полей представляют шаблон времени и обязательно должны присутствовать в файле. Для того чтобы `cron` игнорировала то или иное поле шаблона времени, следует поставить в ней символ * (звездочка).

Примечание. С точки зрения программы символ * означает скорее не «игнорировать поле», а «любое корректное значение», т. е. соответствие чему угодно.

Например, шаблон

```
02 00 01 * *
```

говорит о том, что команда должна быть запущена в две минуты пополуночи (поле часов нулевое) каждого первого числа любого (первая звездочка) месяца, каким бы днем недели оно не было (вторая звездочка).

В таблице 11 приведены допустимые значения полей записей `crontab`.

Таблица 11

Поле	Диапазон
минуты	00–59
часы	00–23 (полночь — 00)
день_месяца	01–31
день_года	01–12
день_недели	01–07 (понедельник — 01, воскресенье — 07)

Пример

Запись команды в файле `crontab`, выполняющая сортировку и отправку пользователю `pav` файла `/usr/sales/weekly` каждый понедельник в 7:30

```
30 07 * * 01 sort /usr/sales/weekly | mail -s"Weekly Sales" pav
```

Поле команд может содержать все, что может быть в команде, вводимой в командной строке оболочки. В нужное время `cron` для выполнения команд запустит стандартную оболочку (`bash`) и передаст ей команду для выполнения.

Для того чтобы определить несколько значений в поле используется запятая в качестве разделяющего символа. Например, если программа `chkquotes` должна выполняться в 9, 11, 14 и 16 часов по понедельникам, вторникам и четвергам 10 марта и 10 сентября, то запись выглядит так:

```
. 09,11,14,16 10 03,09 01,02,04 chkquotes
```

Параметры команды `crontab` приведены в таблице 12.

Таблица 12

Параметр	Описание
`-e`	Позволяет редактировать компоненты файла (при этом вызывается редактор, определенный в переменной `EDITOR` оболочки)
`-r`	Удаляет текущий файл `crontab` из каталога
`-l`	Используется для вывода списка текущих заданий `cron`

Команда `crontab` работает с файлом согласно регистрационному имени.

За корректное использование команды `cron` ответственность несут как администратор системы, так и пользователи, например, использование программы не должно вызвать перегрузку системы.

Подробное описание команд и файла `crontab` приведено в `man cron`, `man crontab` и `man 5 crontab`.

4.3.2. Администрирование многопользовательской и многозадачной среды

4.3.2.1. who

Для получения списка пользователей, работающих в ОС, используется команда `who`:

```
who
root console May 19 07:00
```

Результатом выполнения команды является список, содержащий идентификаторы активных пользователей, терминалы и время входа в систему.

Команда `who` имеет несколько параметров, однако далее рассмотрены только два из них:

1) `-u` — перечисляет пользователей с указанием времени бездействия (точка `.` означает, что пользователь активно работал в последнюю минуту, `old` — что последний раз он нажимал клавиши более суток назад);

2) `-H` — заставляет команду выводить подробную информацию о пользователях; при этом выводит строку заголовка таблицы пользователей, столбцы которой показаны в таблице 13.

Таблица 13

Поле	Описание
NAME	Имена пользователей
LINE	Использованные линии и терминалы
TIME	Время, прошедшее после регистрации пользователя в системе
IDLE	Время, прошедшее со времени последней активной работы пользователя
PID	Идентификатор процесса входной оболочки пользователя
COMMENT	Комментарий

С помощью параметров `-u` и `-H` можно увидеть:

```
who -uH
NAME LINE    TIME           IDLE    PID     COMMENT
root console Dec 12 08:00   .       10340
```

В список включен идентификатор процесса оболочки пользователя.

Подробное описание команды приведено в `man who`.

4.3.2.2. ps

Для получения информации о состоянии запущенных процессов используется команда `ps`. Команда выводит следующую информацию о процессах:

- выполненные процессы;

- процессы, вызвавшие проблемы в системе;
- как долго выполняется процесс;
- какие системные ресурсы затребовал процесс;
- идентификатор процесса (который будет необходим, например, для прекращения работы процесса с помощью команды `kill`) и т. д.

Данная информация полезна как для пользователя, так и для системного администратора. Запущенная без параметров командной строки `ps` выдает список процессов, порожденных администратором.

Наиболее распространенное применение команды `ps` — отслеживание работы фоновых и других процессов в системе. Поскольку в большинстве случаев фоновые процессы не взаимодействуют с экраном и с клавиатурой, команда `ps` остается основным средством наблюдения за ними.

В таблице 14 приведены четыре основных поля информации для каждого процесса, выводимые командой `ps`.

Таблица 14

Поле	Описание
`PID`	Идентификатор процесса
`TTY`	Терминал, с которого был запущен процесс
`TIME`	Время работы процесса
`COMMAND`	Имя выполненной команды

Подробное описание команды приведено в `man ps`.

4.3.2.3. nohup

Обычно дочерний процесс завершается после завершения родительского. Таким образом, если запущен фоновый процесс, он будет завершен при выходе из системы. Для того чтобы процесс продолжал выполняться после выхода из системы, применяется команда `nohup`, указанная в начале командной строки:

```
nohup sort sales.dat &
```

Команда `nohup` заставляет ОС игнорировать выход из нее и продолжать выполнение процесса в фоновом режиме, пока он не закончится. Таким образом, будет запущен процесс, который будет выполняться длительное время, не требуя контроля со стороны администратора системы.

Подробное описание команды приведено в `man nohup`.

4.3.2.4. nice

Команда `nice` позволяет запустить другую команду с предопределенным приоритетом выполнения, предоставляя администратору системы возможность определять приоритет

при выполнении своих задач. При обычном запуске все задачи имеют один и тот же приоритет, и ОС равномерно распределяет между ними процессорное время. С помощью команды nice можно понизить приоритет какой-либо «неспешной» задачи, предоставив другим задачам больше процессорного времени. Повысить приоритет той или иной задачи имеет право только суперпользователь.

Синтаксис команды nice:

```
nice -[<число>] command
```

Уровень приоритета определяется параметром <число>, при этом большее его значение означает назначение меньшего приоритета команде. Значение по умолчанию равно 10, параметр <число> представляет собой число, на которое значение должно быть уменьшено. Например, если запущен процесс сортировки:

```
sort sales.dat > sales.srt &
```

и ему следует дать преимущество над другим процессом, например, процессом печати, необходимо запустить второй процесс с уменьшенным приоритетом:

```
nice -5 lp mail_list &
```

Для того чтобы назначить процессу печати самый низкий возможный приоритет, ввести:

```
nice -10 lp mail_list &
```

Примечание. В случае команды nice тире означает знак параметра.

Только суперпользователь может повысить приоритет процесса, применяя для этого отрицательное значение параметра. Максимально возможный приоритет 20, присвоить его процессу суперпользователь может с помощью команды:

```
nice --10 job &
```

Наличие & в примере достаточно условно, можно изменять приоритеты как фоновых процессов, так и процессов переднего плана.

Подробное описание команды приведено в man nice.

4.3.2.5. renice

Команда renice позволяет изменить приоритет работающего процесса. Формат этой команды подобен формату команды nice:

```
renice -number PID
```

Для изменения приоритета работающего процесса необходимо знать его идентификатор, получить который можно с помощью команды ps, например:

```
ps -e : grep name
```

где name — имя интересующего процесса.

Команда grep отфильтрует только те записи, в которых будет встречаться имя нужной команды, и можно будет узнать идентификатор ее процесса. Если необходимо

изменить приоритет всех процессов пользователя или группы пользователей, в команде renice используется идентификатор пользователя или группы.

Пример

Использование команды renice для процесса пользователя pav

```
ps –ef : grep $LOGNAME
pav 11805 11804 0 Dec 22 ttysb 0:01 sort sales.dat > sales srt
pav 19955 19938 4 16:13:02 ttypo 0:00 grep pav
pav 19938    1 0 16:11:04 ttypo 0–00 bash
pav 19940 19938 42 16:13:02 ttypo 0:33 find . –name core –exec nn {};
```

Чтобы понизить приоритет процесса find с идентификатором 19940, ввести:

```
renice –5 19940
```

В случае команды renice работают те же правила, что и в случае команды nice, а именно:

- ее можно использовать только со своими процессами;
- суперпользователь может применить ее к любому процессу;
- только суперпользователь может повысить приоритет процесса.

Подробное описание команды приведено в man renice.

4.3.2.6. kill

Иногда необходимо прекратить выполнение процесса, не дожидаясь его нормального завершения. Может потребоваться в следующих случаях:

1) процесс использует слишком много времени процессора и ресурсов компьютера;
2) процесс работает слишком долго, не давая ожидаемых результатов;
3) процесс производит слишком большой вывод информации на экран или в файл;
4) процесс привел к блокировке терминала или другой сессии;
5) из-за ошибки оператора или программы используются не те файлы или параметры командной строки;
6) дальнейшее выполнение процесса бесполезно.

Если процесс работает не в фоновом режиме, нажатие клавиш **<Ctrl+C>** должно прервать его выполнение. Фоновый процесс прервать возможно только с помощью команды kill, посылающей процессу сигнал, требующий его завершения.

Используются две формы команды:

```
kill PID(s)
kill –signal PID(s)
```

Для завершения процесса с идентификатором 127 ввести:

```
kill 127
```

Для завершения процессов с идентификаторами 115, 225 и 325 ввести:

```
kill 115 225 325
```

С помощью параметра `-signal` можно, например, дать указание процессу перечитать конфигурационные файлы без прекращения работы. Список доступных сигналов можно получить с помощью команды:

```
kill -l
```

При успешном завершении процесса сообщение не выводится. Сообщение появится при попытке завершения процесса без наличия соответствующих прав доступа или при попытке завершить несуществующий процесс.

Завершение родительского процесса иногда приводит к завершению дочерних, однако для полной уверенности в завершении всех процессов, связанных с данным, следует указывать их в команде `kill`.

Если терминал оказался заблокированным, можно войти в систему с другого терминала:

```
ps -ef: grep $LOGNAME
```

и завершить работу оболочки на заблокированном терминале.

При выполнении команды `kill` процессу посылается соответствующий сигнал. Программы ОС могут посылать и принимать более 20 сигналов, каждый из которых имеет свой номер. Например, при выходе администратора ОС посылает всем его процессам сигнал 1, который заставляет все процессы (кроме запущенных с помощью `nohup`) прекратить работу. Кроме того, существуют программы, написанные таким образом, что будут игнорировать посылаемые им сигналы, включая сигнал 15, который возникает при запуске команды `kill` без указания конкретного сигнала.

Однако сигнал 9 не может быть проигнорирован — процесс будет завершен. Таким образом, если команда:

```
kill PID
```

не смогла завершить процесс (он виден при использовании команды `ps`), необходимо воспользоваться командой:

```
kill -9 PID
```

Команда:

```
kill -9
```

прекращает процесс, не давая возможности, например, корректно закрыть файлы, что может привести к потере данных. Использовать эту возможность следует только в случае крайней необходимости.

Для завершения всех фоновых процессов ввести:

```
kill 0
```

Преимущественное право контроля над процессом принадлежит владельцу. Права владельца могут отменяться только суперпользователем.

Ядро назначает каждому процессу четыре идентификатора: реальный и эффективный UID, реальный и эффективный GID. Реальные ID используются для учета использования системных ресурсов, а эффективные — для определения прав доступа. Как правило, реальные и эффективные ID совпадают. Владелец процесса может посылать в процесс сигналы, а также понижать приоритет процесса.

Процесс, приступающий к выполнению другого программного файла, осуществляет один из системных вызовов семейства `exec`. Когда такое случается, эффективные UID и GID процесса могут быть установлены равными UID и GID файла, содержащего образ новой программы, если у этого файла установлены биты смены идентификатора пользователя и идентификатора группы. Системный вызов `exec` — это механизм, с помощью которого такие команды, как `passwd`, временно получают права суперпользователя (команде `passwd` они нужны для того, чтобы изменить `/etc/passwd`).

Подробное описание команды приведено в `man kill`.

5. УПРАВЛЕНИЕ ПРОГРАММНЫМИ ПАКЕТАМИ

В ОС используются программные пакеты (далее по тексту — пакеты) в формате «.deb». Для управления пакетами в режиме командной строки (или в эмуляторе терминала в графическом режиме) предназначены набор команд нижнего уровня dpkg и комплекс программ высокого уровня apt, apt-cache и aptitude. В графическом режиме управлять пакетами можно с помощью программы synaptic (универсальная графическая оболочка для apt).

По умолчанию обычный пользователь не имеет права использовать эти инструменты. Для всех операций с пакетами (за исключением некоторых случаев получения информации о пакетах) необходимы права суперпользователя, которые администратор может получить через механизм sudo.

Примечание. Права доступа к исполняемым файлам позволяют всем пользователям запускать их на выполнение, но удалять или модифицировать такие файлы может только суперпользователь. Обычно приложения устанавливаются в каталог с правами чтения всеми пользователями, но без права записи в него.

Средства управления пакетами обеспечивают возможность автоматизированной установки обновлений ОС.

5.1. Набор команд dpkg

Набор команд dpkg предназначен, в основном, для операций с пакетами на локальном уровне. С помощью команды dpkg и других команд этого набора можно устанавливать и удалять пакеты, собирать их из исходных текстов, получать информацию о конкретном пакете и об установленных в системе пакетах:

```
dpkg -i <полный_путь>/<полное_имя_пакета>
```

Если пакет (например, iptables_1.4.21-2_amd64.deb), который необходимо установить, расположен, например, в домашнем каталоге пользователя /home/user1, следует выполнить следующую команду:

```
dpkg -i /home/user1/iptables_1.4.21-2_amd64.deb
```

В случае, если неудовлетворенные зависимости пакета отсутствуют, он будет установлен. В случае нарушения зависимостей dpkg выдаст сообщение об ошибке, в котором будут перечислены все необходимые пакеты, которые следует установить, чтобы разрешить обязательные зависимости.

Для удаления ненужного пакета, но сохранения всех его файлов настройки, следует выполнить команду:

```
dpkg -r <значимая_часть_имени_пакета>
```

Для пакета iptables_1.4.21-2_amd64.deb команда будет выглядеть следующим образом:

```
dpkg -r iptables
```

Для удаления пакета и очистки системы от всех его компонентов (в случае, если данный пакет не связан зависимостями с другими установленными пакетами) следует выполнить команду:

```
dpkg -P <значимая_часть_имени_пакета>
```

Если же удаляемый пакет зависит от других пакетов, последует сообщение об ошибке с перечнем зависимостей.

Следует отметить, что использование полного имени пакета регулируется для всех команд семейства dpkg простым правилом: для любых действий с уже установленным пакетом в командной строке применяется значимая часть имени, а во всех остальных случаях — полное имя.

Подробное описание команды приведено в man dpkg.

5.2. Комплекс программ apt

Комплекс программ apt предназначен, в основном, для управления всеми операциями с пакетами (в т. ч. автоматическим разрешением зависимостей) при наличии доступа к сетевым или локальным архивам (источникам) пакетов.

5.2.1. Настройка доступа к архивам пакетов

Информация о сетевых и локальных архивах пакетов для комплекса программ apt содержится в файле /etc/apt/sources.list. В файле находится список источников пакетов, который используется программами для определения местоположения архивов. Список источников разрабатывается для поддержки любого количества активных источников и различных видов этих источников. Источники перечисляются по одному в строке в порядке убывания их приоритета.

Описание файла /etc/apt/sources.list приведено в man sources.list.

Пример

Файл sources.list

```
deb cdrom:[OS Astra Linux 1.3.39 smolensk - amd64 DVD]/ smolensk contrib
main non-free
deb ftp://192.168.32.1/astra/unstable/smolensk/mounted-iso-main smolensk
main contrib non-free
deb ftp://192.168.32.1/astra/unstable/smolensk/mounted-iso-devel smolensk devel
contrib non-free
```

При установке ОС с дистрибутива строка `deb cdrom...` автоматически записывается в файл `sources.list`.

Включить данную строку в список источников также можно при помощи команды:

```
apt-cdrom add
```

DVD-диск с дистрибутивом ОС при этом должен находиться в устройстве чтения DVD-дисков (монтировать его не обязательно).

Строки, соответствующие источникам остальных типов, вносятся в файл при помощи любого редактора.

5.2.2. Установка и удаление пакетов

После установки ОС создается локальная БД о всех пакетах, которые находились на DVD-диске с дистрибутивом и архив установленных пакетов. Эта информация может выводиться в различной форме при помощи команды `apt-cache`. Например, команда:

```
apt-cache show iptables
```

выведет всю информацию, содержащуюся в описании пакета `iptables`.

Обновить содержимое локальной БД можно при помощи команды:

```
apt install update
```

Эту операцию необходимо выполнять при каждом изменении как списка источников пакетов, так и содержимого этих источников.

Полное обновление всех установленных в системе пакетов производится при помощи команды:

```
apt install dist-upgrade
```

Установка отдельного пакета (если он отсутствовал в системе) производится при помощи команды:

```
apt install <значимая_часть_имени_пакета>
```

При этом будут исследованы и разрешены все обязательные зависимости и, при необходимости, установлены необходимые дополнительные пакеты.

Удаление пакета (с сохранением его файлов настройки) производится при помощи команды:

```
apt remove <значимая_часть_имени_пакета>
```

Если при этом необходимо полностью очистить систему от всех компонент удаляемого пакета, то применяется команда:

```
apt remove --purge <значимая_часть_имени_пакета>
```

Описание команд приведено в `man apt-cache` и `man apt`.

6. БАЗОВЫЕ СЕТЕВЫЕ СЛУЖБЫ

6.1. Сеть TCP/IP

6.1.1. Пакеты и сегментация

Данные передаются по сети в форме сетевых пакетов, каждый из которых состоит из заголовка и полезной нагрузки. Заголовок содержит сведения о том, откуда прибыл пакет и куда он направляется. Заголовок, кроме того, может включать контрольную сумму, информацию, характерную для конкретного протокола, и другие инструкции по обработке. Полезная нагрузка — это данные, подлежащие пересылке.

6.1.2. Адресация пакетов

Сетевые пакеты могут достичь пункта назначения только при наличии правильного сетевого адреса. Протокол TCP/IP использует сочетание нескольких схем сетевой адресации.

Самый нижний уровень адресации задается сетевыми аппаратными средствами.

На следующем, более высоком, уровне используется адресация Интернет (которую чаще называют «IP-адресацией»). Каждому включенному в сеть устройству присваивается один четырехбайтовый IP-адрес (в соответствии с протоколом IPv4). IP-адреса глобально уникальны и не зависят от аппаратных средств.

IP-адреса идентифицируют компьютер, но не обеспечивают адресацию отдельных процессов и служб. Протоколы TCP и UDP расширяют IP-адреса, используя порты. Порт в данном случае представляет собой двухбайтовое число, добавляемое к IP-адресу и указывающее конкретного адресата той или иной сетевой службы. Все стандартные UNIX-службы связываются с известными портами, которые определены в файле `/etc/services`. Для того чтобы предотвратить попытки нежелательных процессов замаскироваться под эти службы, установлено, что порты с номерами до 1024 могут использоваться только суперпользователем. Описание файла `/etc/services` приведено в `man services`.

6.1.3. Маршрутизация

6.1.3.1. Таблица

Маршрутизация — это процесс направления пакета по ряду сетей, находящихся между источником и адресатом.

Данные маршрутизации хранятся в таблице маршрутизации. Каждый элемент этой таблицы содержит несколько параметров, включая поле надежности, которое расставляет маршруты по приоритетам, если таблица содержит противоречивую информацию. Для направления пакета по конкретному адресу подбирается наиболее подходящий маршрут.

Если нет ни такого маршрута, ни маршрута по умолчанию, то отправителю возвращается ошибка: «network unreachable» (сеть недоступна).

Таблицу маршрутизации компьютера можно вывести на экран монитора с помощью команды `route`.

6.1.3.2. Организация подсетей

Организация подсетей задается маской подсети, в которой биты сети включены, а биты компьютера выключены. Маска подсети задается во время начальной загрузки, когда конфигурируется сетевой интерфейс командой `ifconfig`. Ядро, как правило, использует сам класс IP-адресов для того, чтобы выяснить, какие биты относятся к сетевой части адреса; если задать маску явно, то эта функция просто отменяется.

При организации подсетей необходимо учесть, что если вычислительная сеть имеет более одного соединения с сетью Интернет, то другие сети должны уметь отличать подсети сети пользователя, чтобы определить в какой маршрутизатор следует послать пакет.

6.1.4. Создание сети TCP/IP

Процесс создания сети TCP/IP состоит из следующих этапов:

- планирование сети;
- назначение IP-адресов;
- настройка сетевых интерфейсов;
- настройка статических маршрутов.

6.1.4.1. Планирование сети

Планирование сети включает:

- определение сегментов сети;
- определение технических и программных средств, с помощью которых сегменты объединяются в сеть;
- определение серверов и рабочих станций, которые будут установлены в каждом сегменте;
- определение типа среды (витая пара и др.).

6.1.4.2. Назначение IP-адресов

Адреса назначают сетевым интерфейсам, а не компьютерам. Если у компьютера есть несколько интерфейсов, у него будет несколько сетевых адресов.

Назначая компьютеру IP-адрес, следует указать соответствие между этим адресом и именем компьютера в файле `/etc/hosts`. Это соответствие позволит обращаться к компьютерам по их именам.

6.1.4.3. Настройка сетевых интерфейсов

Команда `ifconfig` используется для включения и выключения сетевого интерфейса, задания IP-адреса, широковещательного адреса и связанной с ним маски подсети, а также

для установки других параметров. Она обычно выполняется во время первоначальной настройки, но может применяться и для внесения изменений в дальнейшем.

В большинстве случаев команда `ifconfig` имеет следующий формат:

```
ifconfig интерфейс [семейство] <адрес> up <параметр> ...
```

Пример

```
ifconfig eth0 128.138.240.1 up netmask 255.255.255.0 broadcast 128.138.240.255
```

Здесь интерфейс обозначает аппаратный интерфейс, к которому применяется команда. Как правило, это двух-трехсимвольное имя устройства, за которым следует число. Примеры распространенных имен `eth1`, `lo0`, `ppp0` образуются из имени драйвера устройства, используемого для управления им. Для того чтобы выяснить, какие интерфейсы имеются в системе, можно воспользоваться командой:

```
netstat-i
```

Ключевое слово `up` включает интерфейс, а ключевое слово `down` выключает его.

Описание команды приведено в `man ifconfig`.

6.1.4.4. Настройка статических маршрутов

Команда `route` определяет статические маршруты — явно заданные элементы таблицы маршрутизации, которые обычно не меняются даже в тех случаях, когда запускается серверный процесс маршрутизации.

Маршрутизация выполняется на уровне IP. Когда поступает пакет, предназначенный для другого компьютера, IP-адрес пункта назначения пакета сравнивается с маршрутами, указанными в таблице маршрутизации ядра. Если номер сети пункта назначения совпадает с номером сети какого-либо маршрута, то пакет направляется по IP-адресу следующего шлюза, связанного с данным маршрутом.

Существующие маршруты можно вывести на экран командой `route`.

Описание команды приведено в `man route`.

6.1.5. Проверка и отладка сети

6.1.5.1. ping

Команда `ping` служит для проверки соединений в сетях на основе TCP/IP.

Она работает в бесконечном цикле, если не задан параметр `-c`, определяющий количество пакетов, после передачи которого команда завершает свое выполнение. Чтобы прекратить работу команды `ping`, необходимо нажать **<Ctrl+C>**.

Описание команды приведено в `man ping`.

6.1.5.2. netstat

Команда `netstat` выдает информацию о состоянии, относящуюся к сетям:

- проверка состояния сетевых соединений;
- анализ информации о конфигурации интерфейсов;

- изучение таблицы маршрутизации;
- получение статистических данных о различных сетевых протоколах.

Команда `netstat` без параметров выдает информацию о состоянии активных портов TCP и UDP. Неактивные серверы, ожидающие установления соединения, как правило, не показываются (их можно просмотреть командой `netstat -a`).

Основные параметры команды `netstat`:

- `-i` — показывает состояние сетевых интерфейсов;
- `-r` — выдает таблицу маршрутизации ядра;
- `-s` — выдает содержимое счетчиков, разбросанных по сетевым программам.

Описание команды приведено в `man netstat`.

6.1.5.3. arp

Команда `arp` обращается к таблице ядра, в которой задано соответствие IP-адресов аппаратным адресам. В среде Ethernet таблицы ведутся с помощью протокола ARP и не требуют администрирования.

Команда `arp -a` распечатывает содержимое таблицы соответствий.

Описание команды приведено в `man arp`.

6.2. Служба FTP

В ОС передача файлов обеспечивается с помощью интерактивной команды `lftp`, вызываемой на клиентской стороне, и сервера `vsftpd`, который запускается на компьютере, выполняющем функцию сервера службы FTP. Обе команды реализуют протокол передачи файлов FTP. Для копирования файлов клиенту обычно (хотя существует и вариант анонимного доступа) необходимо знание имени и пароля пользователя, которому принадлежат файлы на сервере службы FTP.

6.2.1. Клиентская часть

Клиентская часть может быть установлена командой:

```
apt install lftp
```

Вызов команды `lftp` осуществляется командой:

```
lftp имя_сервера
```

Интерактивный доступ к серверу службы FTP обеспечивается следующими основными внутренними командами `lftp`:

- `open`, `user`, `close` — связь с удаленным компьютером;
- `lcd`, `dir`, `mkdir`, `lpwd` — работа с каталогами в FTP-сервере;
- `get`, `put`, `ftpcopy` — получение и передача файлов;
- `ascii`, `binary`, `status` — установка параметров передачи.

Выход из команды `lftp` осуществляется по команде `exit`.

Описание команды приведено в man lftp.

6.2.2. Сервер vsftpd

В ОС программный пакет vsftpd устанавливается командой:

```
apt install vsftpd
```

После установки следует обратить внимание на файлы документации в каталоге /usr/share/doc/vsftpd, где каталог EXAMPLE содержит различные примеры конфигурационного файла сервера vsftpd.conf. В руководстве man подробно описаны все возможности программы.

Команда располагается в каталоге /usr/sbin/vsftpd.

6.2.2.1. Конфигурационный файл

После установки сервера vsftpd он запускается автоматически и сразу готов к работе с параметрами по умолчанию. Если для работы сервера необходимы другие значения параметров, следует отредактировать конфигурационный файл /etc/vsftpd.conf.

В файле vsftpd.conf представлены три вида параметров:

- BOOLEAN — параметры, которые могут содержать значения YES и NO;
- NUMERIC — параметры, содержащие различные цифровые значения (например, время в секундах или номер порта соединения);
- STRING — параметры, содержащие текстовую строку (например, путь к каталогу на диске).

Следует заметить, что некоторые параметры могут явно отсутствовать в конфигурационном файле. Это означает, что для них используется значение, заданное по умолчанию и обозначаемое как Default: в руководстве man.

Не все параметры следует указывать напрямую, иначе конфигурационный файл может достичь очень больших размеров. В большинстве случаев достаточно записать в файл несколько строк, а для остальных настроек использовать значения по умолчанию.

Многие настройки зависят от других параметров. Если параметры, от которых они зависят, выключены, то и данные настройки будут выключены. Некоторые параметры являются взаимоисключающими и, следовательно, не будут работать в паре с такими включенными параметрами.

Описание службы vsftpd и файла vsftpd.conf приведено на страницах руководства man.

6.3. Служба DHCP

На компьютере, выполняющем роль сервера динамической конфигурации сети, должна быть установлена служба DHCP-сервера.

DHCP-сервер представлен пакетом `isc-dhcp-server` и графической утилитой `fly-admin-dhcp` для его быстрой настройки.

Для установки DHCP-сервера от имени администратора с использованием механизма `sudo` выполнит команду:

```
sudo apt install isc-dhcp-server
```

Установка графической утилиты `fly-admin-dhcp` выполняется командной:

```
sudo apt install fly-admin-dhcp
```

При установке `fly-admin-dhcp` также автоматически будет установлен пакет `isc-dhcp-server`.

Запуск службы DHCP-сервера осуществляется с помощью команды:

```
systemctl start isc-dhcp-server
```

или автоматически путем включения в список служб, запускаемых при старте системы.

Настройки службы DHCP-сервера задаются в файлах `/etc/default/isc-dhcp-server` и `/etc/dhcp/dhcpd.conf`.

В файле `/etc/default/isc-dhcp-server` для параметров `INTERFACES` указываются протоколы и сетевые интерфейсы, с которыми будет работать служба, например:

```
INTERFACESv4="eth0"
#INTERFACESv6=" "
```

При необходимости возможно указать несколько сетевых интерфейсов, разделенных пробелом.

В файле `/etc/dhcp/dhcpd.conf` указывается топология сети и параметры выдаваемой через DHCP-сервер информации.

ВНИМАНИЕ! Для запуска службы DHCP-сервера указанному в файле `/etc/default/isc-dhcp-server` сетевому интерфейсу должен быть присвоен IP-адрес и данный IP-адрес должен быть назначен вручную в файле `/etc/dhcp/dhcpd.conf`.

Сервер динамически назначает IP-адреса DHCP-клиентам обеих подсетей и осуществляет поддержку нескольких клиентов BOOTP. Первые несколько активных строк файла определяют ряд параметров и режимов, действующих для всех обслуживаемых сервером подсетей и клиентов. Конструкция каждой строки есть реализация шаблона «параметр — значение». «Параметр» может быть общим или стоять перед ключевым словом `option`. Параметры, следующие за словом `option`, — это ключи настройки. Они также состоят из имени ключа и его значения.

Кроме общих параметров, существуют т. н. «операторы топологии сети» или «объявления».

Описание некоторых параметров настройки DHCP-сервера, содержащихся в файле `/etc/dhcp/dhcpd.conf`, приведено в таблице 15.

Таблица 15

Параметр	Описание
max-lease-time	Определяет максимально допустимое время аренды. Независимо от длительности аренды, фигурирующей в запросе клиента, этот срок не может превышать значение, заданное данным параметром
get-lease-hostnames	Предписывает dhcpd предоставлять каждому клиенту наряду с динамическим адресом имя узла. Имя узла должно быть получено от DNS. Данный параметр — логический. При значении FALSE назначается адрес, но не имя узла. Значение TRUE используется только в сетях с небольшим количеством хостов, которым выделяются имена, т. к. поиск имен в DNS замедляет запуск демона
hardware type address	Параметр определяет аппаратный адрес клиента. Значение type может быть ethernet или token-ring. address должен быть соответствующим устройству физическим адресом. Параметр должен быть связан с оператором host. Он необходим для распознавания клиента BOOTP
filename file	Указывает файл загрузки для бездисковых клиентов. file — это ASCII-строка, заключенная в кавычки
range [dynamic-bootp]	Данный параметр указывает диапазон адресов. После него через пробел указывается нижний адрес диапазона и опционально верхний адрес. Если верхний адрес не указан, занимается весь теоретически возможный диапазон от нижнего адреса. Этот параметр всегда связан с оператором subnet. Все адреса должны принадлежать этой подсети. Флаг dynamic-bootp указывает, что адреса могут автоматически назначаться клиентам BOOTP также, как и клиентам DHCP. Если оператор subnet не содержит параметра range, для такой подсети динамическое распределение адресов не действует
server-name name	Имя сервера DHCP, передаваемое клиенту. name — это ASCII-строка, заключенная в кавычки
next-server name	Имя узла или адрес сервера, с которого следует получить загрузочный файл
fixed-address	Назначает узлу один или несколько фиксированных адресов. Действителен только в сочетании с параметром host. Если указано несколько адресов, выбирается адрес, корректный для данной сети, из которой выполняет загрузку клиент. Если такого адреса нет, никакие параметры не передаются
dynamic-bootp-lease-cuttof date	Устанавливает дату завершения действия адресов, назначенных клиентам BOOTP. Клиенты BOOTP не обладают способностью обновлять аренду и не знают, что срок аренды может истечь. Этот параметр меняет поведение сервера и используется только в особых случаях

Окончание таблицы 15

Параметр	Описание
`dynamic-bootp-lease-length`	Длительность аренды в секундах для адресов, автоматически назначаемых клиентам BOOTP. Данный параметр используется в особых ситуациях, когда клиенты используют образ загрузки BOOTP PROM. В ходе загрузки клиент действует в качестве клиента BOOTP, а после загрузки работает с протоколом DHCP и умеет обновлять аренду
`use-host-decl-names`	Предписывает передавать имя узла, указанное в операторе `host`, клиенту в качестве его имени. Логический параметр, может иметь значения `TRUE` или `FALSE`
`server-identifier hostname`	Определяет значение, передаваемое в качестве идентификатора сервера. По умолчанию передается первый IP-адрес сетевого интерфейса
`authoritative not authoritative`	Указывает, является ли сервер DHCP компетентным. `not authoritative` используется, когда в компетенцию сервера не входит распределение адресов клиентам
`use-lease-addr-for-default-route`	Логический параметр (`TRUE` или `FALSE`). Предписывает передавать клиенту арендованный адрес в качестве маршрута по умолчанию. Параметр используется только тогда, когда локальный маршрутизатор является сервером-посредником ARP. Оператор настройки `routers` имеет более высокий приоритет
`always-replay-rfc1048`	Логический параметр. Предписывает посылать клиенту BOOTP ответы в соответствии с RFC 1048
`allow keyword deny keyword`	Определяет необходимость отвечать на запросы различных типов. Ключевое слово `keyword` указывает тип разрешенных и запрещенных запросов. Существуют следующие ключевые слова: – `unknown-clients` — определяет возможность динамического назначения адресов неизвестным клиентам; – `bootp` — определяет необходимость отвечать на запросы BOOTP (по умолчанию обслуживаются); – `booting` — используется внутри объявления `host` для указания необходимости отвечать тому или иному клиенту. По умолчанию сервер отвечает всем клиентам

Каждый из операторов топологии может многократно встречаться в файле настройки. Операторы определяют иерархическую структуру. Операторы топологии, встречающиеся в файле `/etc/dhcp/dhcpd.conf`, приведены в таблице 16.

Таблица 16

Оператор	Описание
`group {[parameters] [options]}`	Группирует операторы `shared-network`, `subnet`, `host` и другие операторы `group`. Позволяет применять наборы параметров ко всем элементам группы

Окончание таблицы 16

Оператор	Описание
`shared-network name {[parameters] [options]}`	Используется только в случае, когда несколько подсетей находятся в одном физическом сегменте. В большинстве случаев различные подсети находятся в различных физических сетях. В качестве имени `name` может использоваться любое описательное имя. Оно используется только в отладочных сообщениях. Параметры, связанные с общей сетью, объявляются внутри фигурных скобок и действуют на все подсети общей сети. Каждый оператор `shared-network` содержит не менее двух операторов `subnet`, в противном случае нет необходимости использовать группирование

Общеупотребительные параметры, следующие за ключевым словом `option` в файле `/etc/dhcp/dhcpd.conf`, приведены в таблице 17.

Таблица 17

Параметр	Описание
`subnet-mask`	Определяет маску подсети в формате десятичной записи через точку. Если `subnet-mask` отсутствует, `dhcpd` использует маску подсети из оператора `subnet`
`time-offset`	Указывает разницу данного часового пояса с временем UTC в секундах
`routers`	Перечисляет адреса доступных клиентам маршрутизаторов в порядке предпочтения
`domain-name-servers`	Перечисляет адреса доступных клиентам серверов DNS в порядке предпочтения
`lpr-servers`	Перечисляет адреса доступных клиентам серверов печати LPR в порядке предпочтения
`host-name`	Указывает имя узла для клиента
`domain-name`	Определяет имя домена
`interface-mtu`	Определяет значение MTU для клиента в байтах. Минимально допустимое значение — 68
`broadcast-address`	Определяет широковещательный адрес для подсети клиента
`static-routes destination gateway`	Перечисляет доступные клиенту статические маршруты. Маршрут по умолчанию не может быть указан таким способом. Для его указания используется параметр `routers`
`trailer-encapsulation`	Определяет, следует ли клиенту выполнять инкапсуляцию завершителей (оптимизация, основанная на изменении порядка данных). Значение 0 означает, что инкапсуляцию выполнять не следует, 1 имеет противоположный смысл
`nis-domain string`	Строка символов, определяющая имя домена NIS
`dhcp-client-identifier string`	Используется в операторе `host` для определения идентификатора клиента DHCP. `dhcpd` может использовать данное значение для идентификации клиента вместо аппаратного адреса

ВНИМАНИЕ! Для корректной работы DHCP-сервера требуется в файле `/etc/dhcp/dhcpd.conf` раскомментировать параметр `authoritative`.

После завершения настроек следует перезапустить службу DHCP-сервера с помощью команды:

```
sudo systemctl restart isc-dhcp-server
```

Описание службы DHCP-сервера и файла `/etc/dhcp/dhcpd.conf` приведено на страницах руководств `man dhcpd` и `man dhcpd.conf`.

6.4. Служба NFS

Служба NFS обеспечивает общий доступ к файлам и каталогам систем *nix-систем (в т.ч. Linux), что позволяет использовать ФС удаленных компьютеров.

В ОС используется реализация NFS, работающий на уровне ядра и представленная пакетом `nfs-kernel-server`.

Доступ к ФС удаленных компьютеров обеспечивается с помощью программ на сторонах сервера и клиента.

При работе с сетевой ФС любые операции над файлами, производимые на локальном компьютере, передаются через сеть на удаленный компьютер.

6.4.1. Установка и настройка сервера

Для установки сервера выполнить от имени администратора команды:

```
apt update
apt install nfs-kernel-server
```

Для нормального запуска и возобновления работы службы сервера NFS требуется после установки пакета и перезагрузки компьютера внести изменения в UNIT-файл `/etc/systemd/system/multi-user.target.wants/nfs-server.service`, добавив следующие строки в секцию `unit`:

```
[Unit]
Requires=rpcbind.service
After=rpcbind.service
```

Затем перезапустить службу, выполнив команды:

```
systemctl daemon-reload
systemctl restart nfs-kernel-server
```

На стороне сервера существуют следующие программы, используемые для обеспечения службы NFS:

- `rpc.idmapd` — перенаправляет обращения, сделанные с других компьютеров к службам NFS;
- `rpc.nfsd` — переводит запросы к службе NFS в действительные запросы к локальной ФС;

- rpc.svcgssd — поддерживает создание защищенного соединения;

- rpc.statd — поддерживает восстановление соединения при перезагрузке сервера;

- rpc.mountd — запрашивается для монтирования и размонтирования ФС.

Описание программ приведено на страницах руководства man.

Запросы монтирования поступают от клиентских компьютеров к серверу монтирования mountd, который проверяет правильность клиентского запроса на монтирование и разрешает серверу службы NFS (nfsd) обслуживать запросы клиента, выполнившего монтирование. Клиенту разрешается выполнять различные операции с экспортированной ФС в пределах своих полномочий. Для получения хорошего качества обслуживания клиентов рекомендуется на сервере службы NFS одновременно запускать несколько копий процесса nfsd.

На стороне сервера выполняется экспортирование ФС. Это означает, что определенные поддеревья, задаваемые каталогами, объявляются доступными для клиентских компьютеров. Информация об экспортированных ФС заносится в файл /etc/exports, в котором указывается, какие каталоги доступны для определенных клиентских компьютеров, а также какими правами доступа обладают клиентские компьютеры при выполнении операций на сервере. В конфигурационный файл /etc/exports информация заносится строкой вида:

```
<общий_каталог> <IP-адрес_клиента>(<параметр>)
```

Параметр определяет правила монтирования общего ресурса для клиента. Если параметров несколько, то они указываются через запятую. Перечень параметров и их описание приведены в таблице 18.

Таблица 18

Параметр	Описание
rw	Предоставляет права на чтение и запись
ro	Предоставляет права только на чтение
no_root_squash	По умолчанию в общих ресурсах NFS пользователь root становится обычным пользователем (nfsnobody). Таким образом, владельцем всех файлов, созданных root, становится nfsnobody, что предотвращает загрузку на сервер программ с установленным битом setuid. Если указан параметр no_root_squash, то удаленные пользователи root могут изменить любой файл в разделяемой файловой системе и внести вредоносный код для других пользователей. В целях безопасности рекомендуется этот параметр не использовать
nohide	Служба NFS автоматически не показывает нелокальные ресурсы (например, примонтированные с помощью mount --bind). Данный параметр включает отображение таких ресурсов

Окончание таблицы 18

Параметр	Описание
`sync`	Синхронный режим доступа. Указывает, что сервер должен отвечать на запросы только после записи на диск изменений, выполненных этими запросами
`async`	Асинхронный режим доступа. Указывает серверу не ждать записи информации на диск и давать ответ на запрос сразу. Использование этого режима повышает производительность, но понижает надежность, т.к. в случае обрыва соединения или отказа оборудования возможна потеря данных
`noaccess`	Запрещает доступ к указанному каталогу. Применяется, если доступ к определенному каталогу выдан всем пользователям сети, но при этом необходимо ограничить доступ для отдельных пользователей
`all_squash`	Подразумевает, что все подключения будут выполняться от анонимного пользователя
`subtree_check`	Выполняет контроль поддерева — позволяет экспортировать не весь раздел, а лишь его часть. При этом сервер NFS выполняет дополнительную проверку обращений клиентов для проверки, что они предпринимают попытку доступа к файлам, находящимся в соответствующих подкаталогах. Параметр `subtree_check` включен по умолчанию
`no_subtree_check`	Отменяет контроль поддерева. Не рекомендуется использовать данный параметр, т.к. может быть нарушена безопасность системы. Параметр может применяться в том случае, если экспортируемый каталог совпадает с разделом диска
`anonuid=1000`	Привязывает анонимного пользователя к локальному UID
`anongid=1000`	Привязывает анонимную группу пользователя к локальной группе GID

Пример

Описание в конфигурационном файле `/etc/exports` экспорта разделяемого каталога `/nfsshare`

```
/srv/nfsshare 192.168.1.20/255.255.255.0(rw,nohide,all_squash,anonuid=1000,
    anongid=1000,no_subtree_check)
```

ВНИМАНИЕ! Следует обратить внимание на использование пробелов между IP-адресом/именем клиента и правами его доступа в файле `/etc/exports`. Добавление пробела влечет изменение трактовки прав доступа. Например, строка:

```
/tmp/nfs/ master.astralinux.ru(rw)
```

предоставляет ресурсу `master.astralinux.ru` права на доступ и чтение, в то время как строка:

```
/tmp/nfs/ master.astralinux.ru (rw)
```

предоставляет ресурсу `master.astralinux.ru` права только на чтение, а всем остальным — на чтение и запись.

После внесения изменений в файл /etc/exports необходимо выполнить команду:

```
exportfs -ra
```

6.4.2. Установка и настройка клиента

Для установки клиента выполнить на компьютере от имени администратора команды:

```
apt update
apt install nfs-common
```

После установки пакета nfs-common на клиенте возможно примонтировать разделяемые ресурсы. Список доступных ресурсов можно проверить, выполнив команду:

```
showmount -e <IP-адрес_сервера>
```

Для монтирования разделяемого ресурса на клиенте выполнить команду:

```
mount <IP-адрес_сервера>:<общий_каталог> <каталог_монтирования>
```

где <IP-адрес_сервера> — имя сервера NFS;

<общий_каталог> — экспортированный каталог сервера NFS;

<каталог_монтирования> — каталог монтирования на клиенте.

На стороне клиента для поддержки службы NFS используется модифицированная команда mount (если указывается ФС NFS4, то автоматически вызывается команда mount.nfs4). Команда модифицирована таким образом, чтобы она могла понимать запись:

```
<IP-адрес_сервера>:<общий_каталог>
```

Для удаленных ФС, которые являются частью постоянной конфигурации клиента и должны автоматически монтироваться во время начальной загрузки клиента, должны присутствовать соответствующие строки в файле /etc/fstab клиента, например:

```
192.168.1.10:/srv/nfsshare/    /mnt/share nfs rw,sync,hard,intr 0 0
```

Кроме того, для поддержки защищенных соединений на клиентской стороне должна запускаться команда rpc.gssd.

6.5. Служба DNS

Система доменных имен DNS (Domain Name System) представляет собой иерархическую распределенную систему для получения информации о компьютерах, службах и ресурсах, входящих в глобальную или приватную компьютерную сеть. Чаще всего используется для получения IP-адреса по имени компьютера или устройства, получения информации о маршрутизации почты и т.п.

Основой DNS является представление об иерархической структуре доменного имени и зонах. Распределенная база данных DNS поддерживается с помощью иерархии DNS-серверов, взаимодействующих по определенному протоколу. Каждый сервер, отвечающий за имя, может делегировать ответственность за дальнейшую часть домена другому серверу, что позволяет возложить ответственность за актуальность информации на серверы различных организаций (людей), отвечающих только за «свою» часть доменного имени.

Основными важными понятиями DNS являются:

- домен (область) — именованная ветвь или поддерево в дереве имен. Структура доменного имени отражает порядок следования узлов в иерархии; доменное имя читается справа налево от младших доменов к доменам высшего уровня (в порядке повышения значимости);
- полное имя домена (FQDN) — полностью определенное имя домена. Включает в себя имена всех родительских доменов иерархии DNS;
- зона — часть дерева доменных имен (включая ресурсные записи), размещаемая как единое целое на некотором сервере доменных имен;
- DNS-запрос — запрос от клиента (или сервера) серверу для получения информации.

Служба доменных имен `named` предназначена для генерации ответов на DNS-запросы. Существуют два типа DNS-запросов:

- прямой — запрос на преобразование имени компьютера в IP-адрес;
- обратный — запрос на преобразование IP-адреса в имя компьютера.

6.5.1. Установка DNS-сервера

В ОС используется DNS-сервер BIND9. Для установки службы DNS-сервера выполнить в терминале команду:

```
apt install bind9
```

При установке пакета `bind9` будет автоматически уставновлен пакет инструментов командной строки `bind9utils`, включающий:

- `named-checkconf` — инструмент проверки синтаксиса файлов конфигурации;
- `named-checkzone` — инструмент проверки файлов зон DNS;
- `rndc` — инструмент управления службой DNS.

Дополнительно также рекомендуется установить пакет инструментов командной строки для работы с DNS `dnsutils`, выполнив команду:

```
apt install dnsutils
```

В составе пакета `dnsutils` будут установлены следующие инструменты:

- `dig` — инструмент для опроса DNS-серверов и проверки их ответа;
- `nslookup` — инструмент для проверки преобразования имен в IP-адреса (разрешение имен);
- `nsupdate` — инструмент для динамического обновления записей DNS.

ВНИМАНИЕ! При установке службы DNS-сервера будут автоматически созданы учетная запись пользователя `bind` и группа `bind`. Соответственно, служба будет работать от имени `bind:bind`.

6.5.2. Настройка сервера службы доменных имен named

Конфигурационные параметры службы named хранятся в файлах каталога /etc/bind/, перечень конфигурационных файлов приведен в таблице 19.

Таблица 19 – Конфигурационные файлы службы доменных имен named

Файл	Описание
/etc/bind/named.conf	Основной конфигурационный файл. Содержит значения конфигурационных параметров для всего сервера и ссылки на другие конфигурационные файлы
/etc/bind/named.conf.options	Конфигурационный файл основных параметров сервера, основным из которых является параметр directory, содержащий каталог конфигурационных файлов зон. Значение по умолчанию /var/cache/bind
/etc/bind/named.conf.local	Конфигурационный файл описания локальных зон сервера. Для каждой зоны указываются пути к конфигурационным файлам для прямого и обратного разыменования (как правило, в указанном ранее каталоге /var/cache/bind)
/etc/bind/named.conf.default-zones	Конфигурационный файл зон по умолчанию. В частности, в этом файле содержатся ссылки на автоматически созданные файлы конфигурации /etc/bind/db.local и /etc/bind/127.db зоны localhost. В большинстве случаев не требует правки

Настройка сервера доменных имен является сложной задачей. Перед использованием DNS следует ознакомиться с существующей документацией, файлами помощи и страницами руководства man службы named, конфигурационного файла named.conf и сопутствующих утилит.

Далее приведен типовой пример настройки службы доменных имен named, обслуживающей одну доменную зону. Пример достаточен для демонстрации функционирующего домена ЕПП ОС.

Примечание. Обновление конфигурации сервера может выполняться без перезапуска самой службы доменных имен named вызовом:

```
rndc reload
```

Пример

Настройка сервера DNS домена my.dom подсети 192.168.1.

В конфигурационный файл /etc/bind/named.conf.local необходимо добавить следующие строки:

```
zone "my.dom" {
        type master;
        file "/var/cache/bind/db.my.dom";
```

```
};

zone "1.168.192.in-addr.arpa" {
        type master;
        file "/var/cache/bind/db.192.168.1";
};
```

Примечание. Имена конфигурационных файлов следует выбирать так, чтобы было понятно для какой конфигурации они используются. В приведенном примере имя конфигурационного файла для зоны обратного просмотра может быть, например, `/var/cache/bind/1.168.192.in-addr.arpa.zone` или `/var/cache/bind/db.my.dom.inv`.

Конфигурационный файл `/var/cache/bind/db.my.dom` содержит информацию зоны прямого просмотра:

```
;
; BIND data file for my.dom zone
;
$TTL    604800
@       IN      SOA     my.dom. root.my.dom. (
                         2014031301     ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      server.my.dom.
@       IN      A       192.168.1.100
@       IN      MX      1       server.my.dom.

server          IN      A       192.168.1.100
client1         IN      A       192.168.1.101
client2         IN      A       192.168.1.102
client3         IN      A       192.168.1.103

ns      IN      CNAME   server
;gw CNAMEs
ftp     IN      CNAME   server
repo    IN      CNAME   server
ntp     IN      CNAME   server
```

```
_https._tcp  IN SRV     10 10 443 server.my.com.

client1      IN TXT    "MAKS"
```

Конфигурационный файл /var/cache/bind/db.192.168.1 содержит информацию зоны обратного просмотра:

```
;
; BIND reverse data file for my.dom zone
;
$TTL    86400
@       IN      SOA my.dom. root.my.dom. (
                         2014031301     ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                          86400 )       ; Negative Cache TTL
;
@       IN      NS      server.my.dom.

100     IN      PTR     server.my.dom.
101     IN      PTR     client1.my.dom.
102     IN      PTR     client2.my.dom.
103     IN      PTR     client3.my.dom.
```

Описание зон может содержать следующие основные типы записей:

- NS — имя DNS сервера;
- A — связь имени с IP-адресом;
- CNAME — связь псевдонима с другим именем (возможно псевдонимом);
- PTR — обратная связь IP-адреса с именем;
- SRV — запись о сетевой службе;
- TXT — текстовая запись.

ВНИМАНИЕ! Перевод строки в конце конфигурационных файлов зон обязателен. В большинстве применений необходимо указание точки в конце имен компьютеров для предотвращения вывода корневого суффикса имени вида «1.168.192.in-addr.arpa».

6.5.3. Настройка клиентов для работы со службой доменных имен

Для работы со службой доменных имен на компьютерах необходимо наличие конфигурационного файла `/etc/resolv.conf`, содержащего информацию о доменах и именах серверов DNS, например:

```
domain my.dom
search my.dom
nameserver 192.168.1.100
```

Также может быть рассмотрена установка системы поддержки работы со службой доменных имен, содержащейся в пакете `resolvconf`.

ВНИМАНИЕ! Для взаимодействия DNS-сервера с клиентами, функционирующими в разных мандатных контекстах, требуется дополнительная настройка механизма `privsock`. Описание настройки сетевых служб для работы с использованием механизма `privsock` приведено в документе РУСБ.10152-02 97 01-1.

6.6. Настройка SSH

SSH — это клиент-серверная система для организации защищенных туннелей между двумя и более компьютерами. В туннелях защищаются все передаваемые данные, в т. ч. пароли.

В поставляемую в составе дистрибутива версию пакета `ssh` встроены алгоритмы защитного преобразования ГОСТ `grasshopper-ctr` (в соответствии с ГОСТ Р 34.13-2015) и имитовставки `hmac-gost2012-256-etm` (на основе ГОСТ Р 34.11-2012). Эти алгоритмы используются по умолчанию, их использование не требует специальной настройки.

При этом в список алгоритмов защитного преобразования (параметр конфигурации `Ciphers`) и выработки имитовставки (параметр конфигурации `MACs`), допустимых к использованию, по умолчанию включены следующие алгоритмы защитного преобразования (перечислены в порядке убывания приоритетов применения):

```
grasshopper-ctr,aes128-ctr,aes192-ctr,aes256-ctr,arcfour256,arcfour128,
aes128-cbc,3des-cbc
```

и алгоритмы выработки имитовставки (перечислены в порядке убывания приоритетов применения):

```
hmac-gost2012-256-etm,hmac-md5,hmac-sha1,umac-64@openssh.com,hmac-ripemd160
```

В конфигурационных файлах клиента (файл `/etc/ssh/ssh_config`) и сервера (файл `/etc/ssh/sshd_config`) имеются закомментированные строчки `Ciphers` и `MACs`), справочно отражающие список алгоритмов, принятых по умолчанию. Если требуется изменить набор допустимых алгоритмов или приоритеты их применения, следует раскомментировать данную строчку и указать нужные алгоритмы в порядке приоритета их выполнения.

Например, для приоритетного выбора более простых, а значит, более быстрых алгоритмов можно использовать следующие параметры конфигурации:

```
Ciphers aes128-ctr,aes192-ctr,aes256-ctr,arcfour256,arcfour128,aes128-cbc
MACs hmac-md5,hmac-sha1,umac-64@openssh.com,hmac-ripemd160
```

Проверить списки поддерживаемых алгоритмов можно следующими командами:

```
# список алгоритмов защитного преобразования:
ssh -Q cipher
# список алгоритмов выработки имитовставки:
ssh -Q mac
```

Дополнительная информация по применению ssh доступна на официальном сайте https://wiki.astralinux.ru.

6.6.1. Служба ssh

Служба ssh (синоним sshd) может быть установлена при установке ОС. При этом служба будет запущена автоматически после завершения установки и перезагрузки, что обеспечит удаленный доступ к установленной ОС для выполнения дальнейших настроек.

При необходимости служба может быть установлена отдельно:

```
apt install ssh
```

Проверить состояние службы:

```
systemctl status ssh
```

Служба берет свои конфигурации сначала из командной строки, затем из файла /etc/ssh/sshd_config. Синтаксис:

```
sshd [-deiqtD46] [-b bits] [-f config_file] [-g login_grace_time]
[-h host_key_file] [-k key_gen_time] [-o option] [-p port] [-u len]
```

Параметры, которые могут присутствовать в файле /etc/ssh/sshd_config, описаны в таблице 20. Пустые строки, а также строки, начинающиеся с #, игнорируются. Названия параметров не чувствительны к регистру символов.

Таблица 20

Параметр	Описание
AllowGroups	Задает список групп, разделенный пробелами, которые будут допущены в систему
DenyGroups	Действие, противоположное действию параметра AllowGroups: записанные в данный параметр группы не будут допущены в систему
AllowUsers	Задает разделенный пробелами список пользователей, которые получат доступ в систему. По умолчанию доступ разрешен всем пользователям

Продолжение таблицы 20

Параметр	Описание
DenyUsers	Действие, противоположное действию параметра AllowUsers: записанные в данный параметр пользователи не получат доступ в систему
AFSTokenPassing	Указывает на то, может ли маркер AFS пересылаться на сервер. Значение по умолчанию yes
AllowTCPForwarding	Указывает на то, разрешены ли запросы на переадресацию портов. Значение по умолчанию yes
Banner	Отображает полный путь к файлу сообщения, выводимого перед аутентификацией пользователя
ChallengeResponseAuthentication	Указывает на то, разрешена ли аутентификация по методу «оклик — ответ». Значение по умолчанию yes
Ciphers	Задает разделенный запятыми список методов защиты соединения, разрешенных для использования
CheckMail	Указывает на то, должна ли служба sshd проверять почту в интерактивных сеансах регистрации. Значение по умолчанию no
ClientAliveInterval	Задает интервал ожидания в секундах, по истечении которого клиенту посылается запрос на ввод данных
ClientAliveCountMax	Задает число напоминающих запросов, посылаемых клиенту. Если по достижении указанного предела от клиента не поступит данных, сеанс завершается и сервер прекращает работу. Значение по умолчанию 3
HostKey	Полный путь к файлу, содержащему личный ключ компьютера. Значение по умолчанию /etc/ssh/ssh_host_key
GatewayPorts	Указывает на то, могут ли удаленные компьютеры подключаться к портам, для которых клиент запросил переадресацию. Значение по умолчанию no
HostbasedAuthentication	Указывает на то, разрешена ли аутентификация пользователей с проверкой файлов .rhosts и /etc/hosts.equiv и открытого ключа компьютера. Значение по умолчанию no
IgnoreRhosts	Указывает на то, игнорируются ли файлы $HOME/.rhosts и $HOME/.shosts. Значение по умолчанию yes
IgnoreUserKnownHosts	Указывает на то, игнорируется ли файл $HOME/.ssh/known_hosts в режимах аутентификации RhostsRSAAuthentication и HostbasedAuthentication. Значение по умолчанию no

Продолжение таблицы 20

Параметр	Описание
KeepAlive	Если установлено значение yes (по умолчанию), демон sshd будет периодически проверять наличие связи с клиентом. В случае неуспешного завершения проверки соединение разрывается. Для выключения данного механизма задать значение параметра no в файле конфигурации сервера и клиента
KerberosAuthentication	Указывает на то, разрешена ли аутентификация с использованием Kerberos. Значение по умолчанию no
KerberosOrLocalPasswd	Указывает на то, должна ли использоваться локальная парольная аутентификация в случае неуспешной аутентификации на основе Kerberos
KerberosTgtPassing	Указывает на то, может ли структура TGT системы Kerberos пересылаться на сервер. Значение по умолчанию no)
KerberosTicketCleanup	Указывает на то, должен ли при выходе пользователя удаляться кэш-файл его пропуска Kerberos
ListenAddress	Задает интерфейс, к которому подключается служба sshd. Значение по умолчанию 0.0.0.0, т. е. любой интерфейс
LoginGraceTime	Задает интервал времени в секундах, в течение которого должна произойти аутентификация пользователя. Если процесс аутентификации не успевает завершиться вовремя, сервер разрывает соединение и завершает работу. Значение по умолчанию 600 с
LogLevel	Задает степень подробности журнальных сообщений. Возможные значения: QUIET, FATAL, ERROR, INFO (по умолчанию), VERBOSE, DEBAG (не рекомендуется)
MACs	Задает разделенный запятыми список доступных алгоритмов MAC (код аутентификации сообщений), используемых для обеспечения целостности данных
MaxStartups	Задает максимальное число одновременных неаутентифицированных соединений с демоном sshd
PAMAuthenticationViaKbdInt	Указывает на то, разрешена ли парольная аутентификация с использованием PAM. Значение по умолчанию no)
PasswordAuthentication	Если установлено значение yes (по умолчанию) и ни один механизм беспарольной аутентификации не приносит положительного результата, тогда пользователю выдается приглашение на ввод пароля, который проверяется самим демоном sshd. Если значение параметра no, парольная аутентификация запрещена
PermitEmptyPasswords	Если установлено значение yes, пользователи, не имеющие пароля, могут быть аутентифицированы службой sshd. Если установлено значение no (по умолчанию), пустые пароли запрещены

Окончание таблицы 20

Параметр	Описание
PermitRootLogin	Указывает на то, может ли пользователь root войти в систему с помощью команды ssh. Возможные значения: no (по умолчанию), without-password, forced-command-only и yes
PidFile	Задает путь к файлу, содержащему идентификатор главного процесса. Значение по умолчанию /var/run/sshd.pid
Port	Задает номер порта, к которому подключается sshd. Значение по умолчанию 22
PrintLastLog	Указывает на то, должна ли служба sshd отображать сообщение о времени последнего доступа. Значение по умолчанию yes
PrintMotd	Указывает на то, следует ли после регистрации в системе отображать содержимое файла /etc/motd. Значение по умолчанию yes
Protocol	Задает разделенный запятыми список версий протокола, поддерживаемых службой sshd
PubKeyAuthentication	Указывает на то, разрешена ли аутентификация с использованием открытого ключа. Значение по умолчанию yes
ReverseMappingCheck	Указывает на то, должен ли выполняться обратный поиск имен. Значение по умолчанию no
StrictModes	Если равен yes (по умолчанию), sshd будет запрещать доступ любому пользователю, чей начальный каталог и/или файл .rhosts принадлежат другому пользователю либо открыты для записи
Subsystem	Предназначается для конфигурирования внешней подсистемы. Аргументами является имя подсистемы и команда, выполняемая при поступлении запроса к подсистеме
SyslogFacility	Задает название средства, от имени которого регистрируются события в системе Syslog. Возможны значения: DAEMON, USER, AUTH (по умолчанию), LOCAL0-7
UseLogin	Указывает на то, должна ли применяться команда login для организации интерактивных сеансов регистрации. Значение по умолчанию no
X11Forwarding	Указывает на то, разрешена ли переадресация запросов к системе X Window. Значение по умолчанию no
X11DisplayOffset	Задает номер первого дисплея (сервера) системы X Window, доступного демону sshd для переадресации запросов. Значение по умолчанию 10
XAuthLocation	Задает путь к команде xauth. Значение по умолчанию /usr/X11R6/bin/xauth

6.6.2. Клиент ssh

Клиентом является команда ssh. Синтаксис командной строки:

```
ssh [-afgknqstvxACNTX1246] [-b bind_address] [-c cipher_spec] [-e escape_char]
[-i identity_file] [-login_name] [-m mac_spec] [-o option] [-p port]
[-F configfile] [-L port:host:hostport] [-R port:host:hostport]
[-D port] hostname | user@hostname [command]
```

Подробно со значениями флагов можно ознакомиться в руководстве man. В простом варианте инициировать соединение с сервером sshd можно командой:

```
ssh 10.1.1.170
```

где 10.1.1.170 — IP-адрес компьютера с запущенной службой sshd. При этом sshd будет считать, что пользователь, запрашивающий соединение, имеет такое же имя, под каким он аутентифицирован на компьютере-клиенте. Теоретически клиент ssh может заходить на сервер sshd под любым именем, используя флаг:

```
-l <имя_клиента>
```

Однако сервер будет согласовывать ключ сеанса (например, при беспарольной аутентификации по открытому ключу пользователя), проверяя открытые ключи в домашнем каталоге пользователя именно с этим именем на компьютере-клиенте. Если же используется парольная аутентификация, на компьютере-сервере должна существовать учетная запись с таким именем. Использовать беспарольную аутентификацию по открытым ключам компьютера настоятельно не рекомендуется, т. к. при этом способе в системе должны существовать потенциально опасные файлы: /etc/hosts.equiv, /etc/shosts.equiv, $HOME/.rhosts, $HOME/.shosts.

Команда ssh берет свои конфигурационные установки сначала из командной строки, затем из пользовательского файла $HOME/.ssh/config и из общесистемного файла /etc/ssh/ssh_config. Если идентичные параметры заданы по-разному, выбирается самое первое значение.

В таблице 21 описаны параметры, которые могут присутствовать в файле $HOME/.ssh/config или /etc/ssh/ssh_config. Пустые строки и комментарии игнорируются.

Таблица 21

Параметр	Описание
CheckHostIP	Указывает на то, должна ли команда ssh проверять IP-адреса в файле known_hosts. Значение по умолчанию yes
Ciphers	Задает разделенный запятыми список методов защиты сеанса, разрешенных для использования. По умолчанию aes122b-cbc, 3des-cbc, blowfish-cbc, cast128-cbc, arcfour, aes192-cbc, aes256-cbc

Продолжение таблицы 21

Параметр	Описание
Compression	Указывает на то, должны ли данные сжиматься с помощью команды gzip. Значение по умолчанию no. Эта установка может быть переопределена с помощью параметра командной строки -C
ConnectionAttempts	Задает число неудачных попыток подключения (одна в секунду), после чего произойдет завершение работы. Значение по умолчанию 4
EscapeChar	Задает escape-символ, используемый для отмены специального назначения следующего символа в сеансах с псевдотерминалом. Значение по умолчанию ~. Значение none запрещает использование escape-символа
ForwardAgent	Указывает на то, будет ли запрос к команде ssh-agent переадресован на удаленный сервер. Значение по умолчанию no
ForwardX11	Указывает на то, будут ли запросы к системе X Window автоматически переадресовываться через SSH-туннель с одновременной установкой переменной среды DISPLAY. Значение по умолчанию no
GatewayPorts	Указывает на то, могут ли удаленные компьютеры подключаться к локальным портам, для которых включен режим переадресации. Значение по умолчанию no
GlobalKnownHostsFile	Задает файл, в котором хранится глобальная база ключей компьютера. Значение по умолчанию /etc/ssh/ssh_known_hosts
HostbasedAuthentication	Указывает на то, разрешена ли аутентификация пользователей с проверкой файлов .rhosts, /etc/hosts.equiv и открытого ключа компьютера. Этот параметр рекомендуется установить в значение no
HostKeyAlgorithm	Задает алгоритмы получения ключей компьютеров в порядке приоритета. Значение по умолчанию ssh-rsa, ssh-dss
HostKeyAlias	Задает псевдоним, который должен использоваться при поиске и сохранении ключей компьютера
HostName	Задает имя или IP-адрес компьютера, на котором следует регистрироваться. По умолчанию выбирается имя, указанное в командной строке
IdentityFile	Задает файл, содержащей личный ключ пользователя. Значение по умолчанию $HOME/.ssh/identity. Вместо имени начального каталога пользователя может стоять символ ~. Разрешается иметь несколько таких файлов. Все они будут проверены в указанном порядке
KeepAlive	Если равен yes (по умолчанию), команда ssh будет периодически проверять наличие связи с сервером. В случае неуспешного завершения проверки (в т. ч. из-за временных проблем с маршрутизацией) соединение разрывается. Чтобы выключить этот механизм, следует задать данный параметр, равным no, в файлах /etc/ssh/sshd_config и /etc/ssh/ssh_config либо в файле $HOME/.ssh/config

Продолжение таблицы 21

Параметр	Описание
KerberosAuthentication	Указывает на то, разрешена ли аутентификация с применением Kerberos
KerberosTgtPassing	Указывает на то, будет ли структура TGT системы Kerberos пересылаться на сервер
LocalForward	Требует значения в формате порт:узел:удаленный_порт. Указывает на то, что запросы к соответствующему локальному порту перенаправляются на заданный порт удаленного узла
LogLevel	Задает степень подробности журнальных сообщений команды ssh. Возможные значения: QUIET, FATAL, ERROR, INFO (по умолчанию), VERBOSE, DEBAG
MACs	Задает разделенный запятыми список доступных алгоритмов аутентификации сообщений для обеспечения целостности данных. Стандартный выбор: hmac-md5, hmac-sha1, hmac-ripemd160@openssh.com, hmac-sha1-96, hmac-md5-96
NumberOfPasswordPrompts	Задает число допустимых попыток ввода пароля. Значение по умолчанию 3
PasswordAuthentication	Если равен yes (по умолчанию), то в случае необходимости команда ssh пытается провести парольную аутентификацию
Port	Задает номер порта сервера. Значение по умолчанию 22
PreferredAuthentications	Задает порядок применения методов аутентификации. Значение по умолчанию: publickey, password, keyboard-interactive
Protocol	Задает в порядке приоритета версии протокола SSH
ProxyCommand	Задает команду, которую следует использовать вместо ssh для подключения к серверу. Эта команда выполняется интерпретатором /bin/sh. Спецификация %p соответствует номеру порта, а %h — имени удаленного узла
PubkeyAuthentication	Указывает на то, разрешена ли аутентификация с использованием открытого ключа. Значение по умолчанию yes
RemoteForward	Требует значения в формате удаленный_порт:узел:порт. Указывает на то, что запросы к соответствующему удаленному порту перенаправляются на заданный порт заданного узла. Переадресация запросов к привилегированным портам разрешена только после получения прав суперпользователя на удаленной системе. Эта установка может быть переопределена с помощью параметра командной строки -R
StrictHostKeyCheking	Если равен yes, команда не будет автоматически добавлять ключи компьютера в файл $HOME/.ssh/known_hosts и откажется устанавливать соединение с компьютерами, ключи которых изменились. Если равен no, команда будет добавлять непроверенные ключи сервера в указанные файлы. Если равен ask (по умолчанию), команда будет спрашивать пользователя о том, следует ли добавлять открытый ключ сервера в указанные файлы

Окончание таблицы 21

Параметр	Описание
UsePrivilegedPort	Указывает на то, можно ли использовать привилегированный порт для установления исходящих соединений. Значение по умолчанию no
User	Задает пользователя, от имени которого следует регистрироваться в удаленной системе. Эта установка может быть переопределена с помощью параметра командной строки -l
UserKnownHostsFile	Задает файл, который используется для автоматического обновления открытых ключей
XAuthLocation	Задает путь к команде xauth. Значение по умолчанию /usr/X11R6/bin/xauth

Клиентские конфигурационные файлы бывают глобальными, на уровне системы (/etc/ssh/ssh_config), и локальными, на уровне пользователя ($HOME/.ssh/config). Следовательно, пользователь может полностью контролировать конфигурацию клиентской части SSH.

Конфигурационные файлы разбиты на разделы, установки которых относятся к отдельному компьютеру, группе компьютеров или ко всем компьютерам. Установки разных разделов могут перекрывать друг друга.

6.7. Службы точного времени

ОС предоставляет возможность выбора и настройки следующих служб точного времени:

1) использующие протокол синхронизации времени NTP:

   а) служба сетевого времени ntp — обеспечивает работу ОС в режиме как сервера точного времени, так и клиента. Представлена пакетом ntp, включающим исполняемый файл /usr/sbin/ntpd (демон ntpd), и пакетом ntpdate, включающим инструменты для работы с ntp;

   б) служба timesyncd — альтернатива службы ntp, но имеет меньше возможностей. Не может выполнять функции сервера точного времени;

   в) служба времени chronyd — альтернативная служба точного времени, рекомендованная к применению вместо службы ntp. Входит в пакет chrony;

2) сервис времени высокой точности PTP (Precision Time Protocol) — пакет linuxptp.

При настройке служб времени используются термины для обозначения времени, приведенные в таблице 22.

Таблица 22

Термин	Описание	Пример
Universal time, UTC	UTC (Coordinated Universal Time) — всемирное координированное время. Не зависит от местоположения компьютера, используется в качестве системного времени: времени в ядре ОС, для отметок времени записи журналов и для синхронизации времени службами времени	Universal time: Ср 2019-02-20 07:51:49 UTC
Time Zone	Временная зона. Определяет временное смещение и параметры сезонного (зимнего/летнего) времени.	Time zone: Europe/Moscow (MSK, +0300)
Local time	Локальное время (местное время). Получается из всемирного координированного времени добавлением временного смещения. Используется в основном для взаимодействия с пользователями системы	Local time: Ср 2019-02-20 10:51:49 MSK
RTC time	Аппаратное время, установленное в аппаратных часах компьютера (Real Time Clock, RTC, также CMOS или BIOS time). Используется для первоначальной установки времени при загрузке ОС. Аппаратные часы могут быть настроены как на всемирное координированное, так и на местное время. При установке системного времени на основании показаний аппаратных часов ОС принимает решение о том, какое именно время (UTC или местное) показывают аппаратные часы, на основании собственных внутренних настроек (см. `man timedatectl`)	RTC time: Ср 2019-02-20 07:51:49

6.7.1. Служба сетевого времени ntp

Сервер единого сетевого времени предназначен для синхронизации времени компьютеров. Синхронизация выполняется по протоколу NTP. Алгоритм коррекции временной шкалы включает внесение задержек, коррекцию частоты часов и ряд механизмов, позволяющих достичь точности порядка нескольких миллисекунд даже после длительных периодов потери связи с синхронизирующими источниками. Для надежной защиты передаваемого сигнала может использоваться аутентификация при помощи криптографических ключей. Целостность данных обеспечивается с помощью IP- и UDP-контрольных сумм.

В зависимости от точности синхронизации часов серверы относятся к различным уровням точности (stratum level). К уровню 1 относятся серверы, синхронизирующиеся по собственным сверхточным атомным часам или по радиочасам, к уровню 2 относятся серверы, синхронизирующиеся с серверами уровня 1, уровень 3 синхронизируется с уровнем 2 и т.д. до уровня 16, к которому относятся серверы с недостоверными (несинхронизированными) показаниями часов.

6.7.1.1. Режимы работы

Существует четыре режима работы сервера единого сетевого времени. Каждый режим определяет способ взаимодействия рабочих станций в сети синхронизации:

1) режим клиент-сервер — клиент посылает запрос серверу (нескольким серверам). Сервер обрабатывает запрос и посылает ответ. Такой режим работы обеспечивает синхронизацию времени на клиенте со временем на сервере, но время на сервере со временем на клиенте при этом не синхронизируется. В этом режиме в роли клиента могут выступать как клиентские (пользовательские) компьютеры, так и серверы. Обычно используется данный режим;

2) симметричный режим — обеспечивает высокую надежность синхронизации, т. к. при выходе из строя одного из источников времени система автоматически перенконфигурируется таким образом, чтобы исключить его из сети синхронизации. Может быть активным или пассивным:

- в активном режиме каждый компьютер в сети периодически посылает сообщения другому компьютеру вне зависимости от его доступности и уровня. Сообщения содержат предложение синхронизировать собственное время и время компьютера – получателя сообщения. Адреса компьютеров для отправки сообщений известны;
- в пассивном режиме адрес компьютера для отправки сообщения заранее не известен. Взаимодействие в этом режиме начинается с момента получения сообщения от компьютера с неизвестным адресом, работающего в активном режиме, и сохраняется до тех пор, пока компьютер достижим и функционирует на уровне ниже или равном уровню данного компьютера. Пассивный режим обычно используется первичными или вторичными серверами;

3) широковещательный режим — один или более серверов времени рассылают широковещательные сообщения, клиенты определяют время исходя из предположения, что задержка составляет несколько миллисекунд. Сервер при этом не принимает ответных сообщений. Такой режим используется в быстрых локальных сетях с большим числом рабочих станций и без необходимости в высокой точности;

4) межсетевой режим — аналогичен широковещательному, но сообщения передаются не в рамках одной подсети, ограниченной локальным широковещательным адресом, а распространяются и в другие сети. Для работы службы единого времени в межсетевом режиме выделен групповой IP-адрес (`224.0.1.1`), который используется как для серверов, так и для клиентов. Межсетевой режим используется в сетях, разделенных на подсети с помощью маршрутизаторов и мостов, которые не способны ретранслировать широковещательные сообщения.

Системы, реализующие службу единого времени, могут выступать в следующих ролях:

1) серверы — предоставляют службу времени другим системам;

2) равноправные узлы — серверы одинакового уровня, временно используемые для синхронизации при потере связи с более высокоуровневым сервером;

3) опросные клиенты — регулярно опрашивают сервера и синхронизируют системные часы по наиболее точному источнику времени;

4) вещательные клиенты — пассивно принимают вещательные пакеты от серверов на ЛВС. Создают меньший сетевой трафик, чем опросные клиенты, но обеспечивают меньшую точность.

6.7.1.2. Установка и базовая настройка сервера времени

Для установки и запуска службы `ntp` необходимо:

1) при стандартной установке ОС служба `ntp` устанавливается по умолчанию. Если требуется установить службу `ntp` отдельно, то используется команда:

```
apt install ntp
```

2) проверить показания аппаратных часов командой:

```
date
```

при необходимости настроить время аппаратных часов вручную, используя команду `timedatectl set-time`:

```
timedatectl set-time "2020-12-31 23:59:59"
```

или использовать графическую утилиту `fly-admin-date`. Отклонение показаний часов от реального времени не должно превышать 1000 секунд;

3) настроить сервер времени путем редактирования конфигурационного файла службы `/etc/ntp.conf` (описание конфигурационного файла приведено в 6.7.1.3):

а) если предполагается использовать имеющиеся службы времени или аппаратные часы точного времени, то настроить синхронизацию с ними в соответствии с инструкциями производителя;

б) при работе в сети с доступом в интернет рекомендуется исключить строки с параметром `pool`:

```
pool 0.debian.pool.ntp.org iburst
pool 1.debian.pool.ntp.org iburst
pool 2.debian.pool.ntp.org iburst
pool 3.debian.pool.ntp.org iburst
```

и указать собственные серверы времени для синхронизации, используя параметр `server`, например российские серверы ВНИИФТРИ:

```
server ntp4.vniiftri.ru
server ntp1.niiftri.irkutsk.ru
```

```
server vniiftri.khv.ru
```

в) возможно, для предотвращения отключения службы при потере связи, указать в качестве источника синхронизации собственные часы компьютера:

```
server 127.127.1.0
fudge 127.127.1.0 stratum 10
```

ВНИМАНИЕ! Указание в качестве источника синхронизации времени собственные часы компьютера может привести к выдаче неверных показаний времени. В случае возможной ситуации потери связи предпочтительно использовать службу времени `chronyd`;

г) изменить пункт:

```
# Clients from this (example!) subnet have unlimited access,
      but only if
# cryptographically authenticated.
```

д) запретить всем клиентам удаленно изменять настройки сервера и разрешить выполнять настройки при локальном подключении:

```
restrict -4 default kod notrap nomodify nopeer noquery limited
restrict -6 default kod notrap nomodify nopeer noquery limited
restrict 127.0.0.1
restrict ::1
```

В состав ОС входит графическая утилита `fly-admin-ntp`, позволяющая произвести большинство настроек службы `ntp` в графическом режиме (см. электронную справку);

4) разрешить автоматический запуск службы `ntp` после перезагрузки и запустить службу:

```
systemctl enable --now ntp
```

6.7.1.3. Конфигурационный файл ntp.conf

В конфигурационном файле приведены конфигурационные команды, состоящие из ключевого слова и следующих за ним параметров, разделенных пробелами. Команда должна занимать строго одну строку. Параметрами могут быть имена и адреса хостов (в форме IP-адресов и доменных имен), целые и дробные числа, текстовые строки. Необязательные параметры заключены в квадратные скобки «[ ]», альтернативные — отделены символом «|». Нотация вида «[...]» означает, что стоящий перед ней необязательный параметр может повторяться несколько раз.

Конфигурационный файл `/etc/ntp.conf` имеет следующие основные настройки:

```
pool <адрес> [iburst]
server <адрес> [key <имя_ключа> | autokey] [version <номер_версии>] [prefer]
      [minpoll <интервал_времени>] [maxpoll <интервал_времени>]
```

```
peer <адрес> [key <имя_ключа> | autokey] [version <номер_версии>] [prefer]
      [minpoll <интервал_времени>] [maxpoll <интервал_времени>]
broadcast <адрес> [key <имя_ключа> | autokey] [version <номер_версии>]
      [minpoll <интервал_времени>] [ttl <время>]
manycastclient <адрес> [key <имя_ключа> | autokey] [version <номер_версии>]
      [minpoll <интервал_времени>] [maxpoll<интервал_времени>] [ttl <время>]
```

Описание команд конфигурационного файла приведено в таблице 23.

Таблица 23

Команда	Описание
`pool`	Используется в конфигурации, устанавливаемой по умолчанию. Рекомендуется заменить этот параметр на параметр `server` (см. ниже). Позволяет получить от указанного сервера случайный набор серверов, с которыми далее будет выполняться односторонняя синхронизация (локальное время может быть синхронизировано с удаленным сервером, но удаленный сервер не может синхронизировать свое время с локальным). Необязательный параметр `iburst` указывает, что в случае недоступности указанного сервера следует выполнить 8 попыток соединения, после чего исключить сервер из дальнейших опросов
`server`	Позволяет установить постоянное соединение (организовать постоянную ассоциацию) клиента с указанным удаленным сервером. При этом локальное время может быть синхронизировано с удаленным сервером, но удаленный сервер не может синхронизировать свое время с локальным
`peer`	Устанавливается постоянное соединение (ассоциация) в симметрично-активном режиме с указанным удаленном сервером (`peer` — симметричным). В данном режиме локальные часы могут быть синхронизированы с удаленным симметричным сервером или удаленный сервер может синхронизироваться с локальными часами
`broadcast`	Организуется постоянная широковещательная ассоциация
`manycastclient`	Организуется межсетевой режим синхронизации с указанным групповым адресом
`vmanycast`	Указывает, что локальный сервер должен работать в клиентском режиме с удаленными серверами, которые обнаруживаются в процессе работы при помощи широковещательных/межсетевых сообщений

Описание параметров команд приведено в таблице 24.

Таблица 24

Параметр	Описание
`autokey`	Все отсылаемые сообщения включают поля аутентификации, защищенные в автоматическом режиме

Окончание таблицы 24

Параметр	Описание
`key <имя_ключа>`	Все отправляемые и принимаемые пакеты включают поля аутентификации, защищенные при помощи криптографического ключа с заданным идентификатором, значения которого составляют от 1 до 65534. По умолчанию поля аутентификации не используются
`minpoll <интервал_времени>`, `maxpoll <интервал_времени>`	Минимальный и максимальный интервалы опроса в секундах. Значение параметра равно степени в которую необходимо возвести двойку для получение требуемого значения интервала времени. Изменяется в пределах от 4 (16 с) до 17 (36,4 ч). По умолчанию `minpoll 6` (64 с), `maxpoll 10` (1024 с). Действительны только для команд `server` и `peer`
`noselect`	Указывает, что сервер используется только в демонстративных целях
`prefer`	Отмечает, что сервер является предпочтительным
`ttl <время>`	Указывает время жизни пакета. Используется только в широковещательном и межсетевом режимах
`version <номер_версии>`	Указывает версию протокола отправляемых пакетов. Может принимать значение от 1 до 4. Значение по умолчанию 4. Действителен для команд `server`, `peer` и `broadcast`

Полное описание настроек конфигурационного файла доступно в `man ntp.conf`.

6.7.1.4. Настройка аутентификации

Настройка аутентификации осуществляется с помощью конфигурационного файла `/etc/ntp.conf` с использованием дополнительных параметров команд `peer`, `server`, `broadcast` и `multicast`:

- `autokey [logsec]` — указывает интервалы в секундах между генерациями нового ключа;
- `controlkey key` — указывает идентификатор ключа для использования командой `ntpq`;
- `keys keyfile` — указывает местонахождение файла, хранящего ключи и их идентификаторы, используемые командами `ntpd`, `ntpq` и `ntpdc`. Данная команда эквивалентна использованию параметра `-k` командной строки;
- `keysdir <путь>` — указывает путь к каталогу, хранящему ключи. Значение по умолчанию `/usr/local/etc/`;
- `trustedkey key [...]` — указывает идентификаторы ключей, которые являются доверенными для аутентификации с симметричным ключом.

Для создания ключей используется команда `ntp-keygen`. Для запуска необходимо иметь права суперпользователя. При запуске она генерирует новые ключи и записывает их в соответствующие файлы.

6.7.1.5. ntpd

Демон `ntpd` обеспечивает работу службы `ntp` в фоновом режиме. Запускается при запуске службы `ntp` или исполняемым файлом `/usr/sbin/ntpd`. Описание демона `ntpd` доступно в `man ntpd`.

6.7.1.6. ntpq

Команда `ntpq` применяется для контроля состояния службы `ntp`. Команда получает состояние службы с помощью стандартных запросов и выводит сводку. Также может получать и выводить список серверов.

Может быть запущена как в интерактивном режиме, так и с использованием командной строки.

Команда имеет следующий синтаксис:

```
ntpq [-ip] [-c команда] [хост] [...]
```

Параметры командной строки приведены в таблице 25.

Таблица 25

Параметр	Описание
`-4`	Форсирование разрешения доменных имен в пространство имен протокола IP версии 4
`-6`	Использование пространства имен протокола IP версии 6
`-d`	Отладочный режим
`-i`	Форсирование интерактивного режима. Команды принимаются со стандартного выхода
`-p`	Вывод всех известных соседних серверов

Вывод команды `ntpq -p` при нормально работающей службе:

```
remote           refid          st t when poll reach   delay   offset  jitter
==============================================================================
0.ru.pool.ntp.o .POOL.          16 p    -   64     0    0.000    0.000   0.000
1.ru.pool.ntp.o .POOL.          16 p    -   64     0    0.000    0.000   0.000
2.ru.pool.ntp.o .POOL.          16 p    -   64     0    0.000    0.000   0.000
3.ru.pool.ntp.o .POOL.          16 p    -   64     0    0.000    0.000   0.000
127.127.1.0     .LOCL.          10 l 1101   64     0    0.000    0.000   0.000
+185.209.85.222  195.91.239.8    2 u   20   64   377   10.631    0.690   0.355
*195.91.239.8    .PPS.           1 u   19   64   377    1.256    0.081   0.065
+192.36.143.130  .PPS.           1 u   18   64   377   19.755    0.129   0.330
-37.193.156.169  80.242.83.227   2 u   12   64   377   44.877   -0.832   2.427
-95.165.138.248  89.109.251.24   2 u    7   64   377    3.118    0.241   0.140
```

В результатах вывода команды `ntpq -p`:

1) первый символ в строке:

   а) «*» — выбранный для синхронизации сервер времени;

б) «+» — новый сервер времени, доступный для синхронизации;

в) «o» — PPS-источник (источник секундных импульсов);

г) «пробел» — неработающий источник;

д) «–» — не рекомендованный для синхронизации сервер;

е) «x» — не доступный для синхронизации сервер;

2) `remote` — адрес опрошенного сервера времени;

3) `refid` — источник сигналов времени, с которым синхронизируется опрошенный сервер. Может быть другой сервер или аппаратные часы (PPS);

4) `st` — уровень (stratum) сервера;

5) `t` — тип сервера (`u` — unucast, `m` — multicast, `l` — local, `p` — pool и т.д.);

6) `when` —- время, прошедшее с последней синхронизации (последнего ответа сервера), в секундах, если не указано иное;

7) `poll` — интервал опроса (двоичный логарифм периода опроса в секундах);

8) `reach` — восьмеричное значение доступности. Отражает доступность сервера при последних восьми опросах, при 100% доступности проходит значения 0, 1, 3, 7, 17, 37, 77, 177, 377 и далее остатся равным 377;

9) `delay` — задержка ответа (время между отправкой запроса и получением ответа);

10) `offset` — смещение времени относительно локального сервера;

11) `jitter` — дисперсия (разброс) времени прохождения пакетов при обмене с сервером.

Интерактивные команды

Интерактивная команда состоит из командного слова и следующих за ним параметров (возможно использование от 0 до 4 параметров). Результат выполнения команды направляется на стандартный вывод (`stdout`) и может быть записан в файл, используя `> <имя_файла>`. Список интерактивных команд приведен в таблице 26.

Таблица 26

Команда	Описание
`? [<командное_слово>]` `help1 [<командное_слово>]`	Если задан параметр `?`, будет выдана информация о возможном использовании данной команды

`addvars <имя_переменной>`
`[ = <значение>] [...] rmvars`
`<имя_переменной> [...]`
`clearvars`
	Данные, передаваемые протоколом NTP, содержат ряд сущностей вида `<имя_переменной>=<значение>`. Команда `ntpq` поддерживает внутренний список, в котором данные встраиваются в контрольные сообщения. Команда `addvars` добавляет переменные в список, `rmvars` удаляет переменные из списка, `clearvars` полностью очищает список

`cooked`	Позволяет преобразовать вывод переменных и их значения в удобный для пользователя вид

Окончание таблицы 26

Команда	Описание
`debug more | less | off`	Позволяет включить/выключить внутреннюю команду запросов
`delay <миллисекунды>`	Указывает временный интервал для добавления к временной отметке (`timestamp`), которая включается в запросы, требующие аутентификации. Это используется для возможности изменения настроек сервера
`host <имя_хоста>`	Устанавливает имя хоста, к которому будут отсылаться последующие запросы
`hostnames [yes | no]`	Если указывается `yes`, доменные имена хостов выводятся на терминал. Иначе выводятся на терминал численные адреса. Значение по умолчанию `yes`
`keyid <идентификатор_ключа>`	Позволяет указать номер ключа для использования его в запросах, требующих аутентификацию
`ntpversion 1 | 2 | 3 | 4`	Устанавливает номер версии NTP. По умолчанию используется протокол версии 6
`passwd`	Запрашивает пароль, который будет использоваться в запросах, требующих аутентификации
`quit`	Выход из интерактивного режима `ntpq`
`raw`	Заставляет выводить результаты запросов команды, как будто они пришли от удаленного сервера
`timeout <миллисекунды>`	Устанавливает временной интервал запросов серверам. Значение по умолчанию 5000 мс

Команды контрольных сообщений

Каждая ассоциация (постоянное соединение), известная серверу единого времени, имеет 16-битный целочисленный идентификатор. Ассоциация с идентификатором 0 — определяет системные переменные, чьи имена лежат вне локального пространства имен. Команды контрольных сообщений приведены в таблице 27.

Таблица 27

Команда	Описание
`associations`	Получение и вывод списка идентификаторов ассоциаций и текущее состояние соседних серверов. Список выводится в виде колонок
`cv [assocID] [variable_name [ = value [...] ][...]`	Запрос на переменные серверных часов. На данный запрос отвечают серверы, имеющие внешние источники синхронизации времени
`lassociations`	Получает и выводит список идентификаторов ассоциаций и соседних серверов (`peer`), с которыми общается сервер
`lpassociations`	Выводит сведения о всех ассоциациях из кэшированного списка

Окончание таблицы 27

Команда	Описание
`peers`	Получение текущего списка соседних серверов (`peer`)

6.7.1.7. ntpdate

Команда `ntpdate` применяется для проверки работы сервера времени и коррекции показаний времени.

Должна быть запущена с правами `root`. Возможен запуск как из командной строки (вручную), так и из стартового скрипта, выполняемого при загрузке ОС. Возможно выполнение `ntpdate` по расписанию из сценария `cron` для периодической коррекции времени.

Для установки инструмента выполнить команду:

```
sudo apt install ntpdate
```

Команда имеет следующий синтаксис:

```
ntpdate [ -параметры] [ server ]
```

Основные параметры командной строки приведены в таблице 28.

Таблица 28

Параметр	Описание
`-a <ключ>`	Разрешение аутентификации и указание ключа для использования. По умолчанию аутентификация отключена
`-d`	Проверка доступности сервера времени запросом времени с подробной диагностикой без коррекции показаний локальных часов
`-q`	Проверка доступности сервера времени запросом времени без коррекции показаний локальных часов
`-u`	Предписывает использовать для запроса времени IP-порт, отличный от 123. По умолчанию `ntpdate` использует тот же IP-порт (123) что и демон `ntpd`, и, если служба `ntp` запущена, то `ntpdate` при запуске выдаст ошибку, что порт занят. Также IP-порт 123 может быть закрыт для обеспечения безопасности
`-b`	Принудительное пошаговая коррекция времени с помощью вызова функции `settimeofday()`. Параметр следует использовать при вызове из файла запуска во время начальной загрузки

Например, для осуществление периодической коррекции времени выполнить команду:

```
ntpdate -ubv 0.ru.pool.ntp.org
```

6.7.1.8. ntptrace

Команда `ntptrace` позволяет, проходя по цепочке серверов, определить источник точного времени. Следует учитывать, что серверы из цепочки не обязаны отвечать на запросы данной команды.

Команда `ntptrace` имеет следующий синтаксис:

```
ntptrace [ -<параметр> [<значение>] ]... [<сервер>]
```

Если на вход команде не поступает никаких аргументов, то началом поиска будет локальный сервер.

Параметры командной строки приведены в таблице 29.

Таблица 29

Параметр	Описание
`-n`	В результатах запроса вместо доменных имен серверов выдаются их IP-адреса. Данный параметр удобен, когда в сети отсутствует DNS
`-m`	Установка максимального количества серверов в цепочке. Значение по умолчанию 99
`-r`	Установка единичного удаленного сервера

6.7.1.9. Методы синхронизации системных часов

Система единого времени предусматривает два механизма для синхронизации системных часов с другими узлами в сети.

Команда `ntpdate`, выполняемая с параметром `-b`, опрашивает, как минимум, один сервер единого времени, затем синхронизирует системные часы с наиболее точным. Выполняется только при запуске ОС до запуска приложений.

После первоначальной синхронизации системных часов командой `ntpdate` во время загрузки ОС служба `ntp` постоянно работает в фоновом режиме, периодически опрашивая серверы службы единого времени, заданные в `/etc/ntp.conf`, и по мере необходимости корректирует системные часы. Данные незначительные корректировки во времени должны быть незаметными для приложений. Отклонение времени клиента от временем сервера записывается в файл погрешности `driftfile`. С течением времени `ntp` постепенно снижает это отклонение и, соответственно, снижается частота опроса серверов.

6.7.1.10. Синхронизация времени в виртуальной среде[1)]

Синхронность показаний времени на взаимодействующих компьютерах — одно из важнейших требований корректной работы в домене. Нарушение синхронности времени может вести к невозможности входа в домен, ошибкам репликации данных, потере информации.

В случае если на компьютере приняты настройки времени по умолчанию, то время, полученное от сервера времени, может отличается от показаний внутренних часов компьютера более, чем на тысячу секунд, что считается фатальной ошибкой. Такое расхождение времени автоматически не корректируется и требует ручной коррекции. Системы виртуализации способны самостоятельно синхронизировать время виртуальных машин, но возможна ситуация, когда в восстановленном из длительного «сна» образе виртуальной машины время автоматически синхронизироваться не будет из-за слишком большого расхождения.

[1)] Для процессоров, поддерживающих технологию виртуализации.

Для восстановления автоматической синхронизации времени следует остановить службу синхронизации времени, провести принудительную синхронизацию и повторно запустить службу:

```
service ntp stop
ntpdate -bv <IP_адрес_сервера_времени>
service ntp start
```

6.7.2. Служба timesyncd

Служба `timesyncd` не представлена отдельными пакетами, устанавливается автоматически при установке ОС, при этом автоматический запуск службы при загрузке ОС отключен.

Предназначена для использования в роли клиента и не может выступать сервером точного времени.

Поддерживает только упрощенный протокол передачи времени.

6.7.2.1. Настройка

Для использования `timesyncd` необходимо полностью удалить службы `ntp` и `chronyd` (если они были установлены), разрешить автоматический запуск и запустить службу `timesyncd`:

```
apt purge ntp chrony
systemctl enable systemd-timesyncd
systemctl start systemd-timesyncd
```

ВНИМАНИЕ! `timesyncd` немедленно завершает свою работу без сообщений об ошибке, если обнаружит на компьютере:

- установленную службу `ntp` (даже незапущенную);
- установленную службу `chronyd` (даже незапущенную);
- для виртуальных машин — установленные гостевые дополнения Oracle Virtual Box.

Запись о завершении работы `timesyncd` будет внесена в системный журнал `/var/log/syslog`.

Состояние службы можно проверить командой:

```
systemctl status systemd-timesyncd
```

или командой:

```
timedatectl status
```

Пример

Вывод команды `timedatectl status`

```
Local time: Ср 2018-12-26 11:08:12 MSK
Universal time: Ср 2018-12-26 08:08:12 UTC
RTC time: Ср 2018-12-26 08:08:12
```

```
Time zone: Europe/Moscow (MSK, +0300)
Network time on: yes
NTP synchronized: yes
RTC in local TZ: no
```

Автоматический запуск службы отключается командой:

```
timedatectl set-ntp false
```

6.7.2.2. Выбор серверов времени

Служба `timesyncd` по умолчанию настроена для работы с набором российских серверов времени (см. файл `/etc/systemd/timesyncd.conf`):

- ntp1.vniiftri.ru
- 0.ru.pool.ntp.org
- 1.ru.pool.ntp.org
- 2.ru.pool.ntp.org
- 3.ru.pool.ntp.org

Дополнительно служба `timesyncd` получает имена серверов времени от службы `systemd-networkd`, если в конфигурационных файлах этой службы (каталоги `/lib/systemd/network/`, `/run/systemd/network/`, `/etc/systemd/network/` или файл `/lib/`) указаны серверы единого времени, привязанные к сетевым интерфейсам.

Более подробная информация о службе `systemd-networkd` приведена в `man systemd.network`.

Дополнительные и резервные серверы могут быть указаны в собственных конфигурационных файлах службы `timesyncd`:

1) `/etc/systemd/timesyncd.conf`
2) `/etc/systemd/timesyncd.conf.d/*.conf`;
3) `/run/systemd/timesyncd.conf.d/*.conf`;
4) `/usr/lib/systemd/timesyncd.conf.d/*.conf`.

Основные параметры в конфигурационном файле:

1) `NTP=` — разделенный пробелами основной список имен серверов единого времени. Объединяется со списком полученных от службы `systemd-networkd`. По умолчанию список пустой и используются резервные серверы, указанные в параметре `FallbackNTP=`;

2) `FallbackNTP=` — разделенный пробелами список имен резервных серверов единого времени;

3) `timesyncd` — перебирает по очереди все серверы из основного списка и, если не удалось связаться ни с одним из серверов, обращается к серверам из резервного списка.

В стандартном конфигурационном файле значения параметров, принятые по умолчанию, указаны в виде комментариев.

Автоматический запуск службы отключается командой:

```
timedatectl set-ntp false
```

6.7.3. Служба chronyd

Служба точного времени рекомендованная к применению вместо службы `ntp`. Служба `chronyd` имеет ряд преимуществ перед `ntp`, в частности:

- не прекращает работу, обнаружив слишком большое отклонение времени, а пытается выполнить коррекцию времени;
- быстрее выполняет синхронизацию;
- работает, если порт 123 закрыт для исходящих запросов.

6.7.3.1. Установка

При установке ОС пакет `chrony` по умолчанию не устанавливается и может быть установлен следующей командой (при этом будет удален устанавливаемый по умолчанию пакет `ntp`):

```
apt install chrony
```

ВНИМАНИЕ! При установке контроллера домена FreeIPA пакет `chrony` будет установлен автоматически, при этом автоматически будет удален пакет `ntp`.

6.7.3.2. Настройка

В режиме клиента служба `chronyd` может запускаться с настройками по умолчанию без конфигурационного файла.

Для дополнительной настройки, а также для настройки работы службы в режиме сервера необходимо создать конфигурационный файл с именем `/etc/chrony/chrony.conf`. Примеры конфигурационных файлов находятся в каталоге `/etc/chrony`.

Для работы службы `chronyd` как сервер времени (т.е. отвечала другим клиентам на запросы), необходимо в конфигурационном файле добавить строку с разрешениями. Например, разрешить всем:

```
allow
```

Затем перезапустить службу:

```
systemctl restart chronyd
```

Более подробную информацию о конфигурационном файле см. в `man chrony.conf`.

6.7.4. Служба времени высокой точности PTP

Служба времени высокой точности PTP включает следующие службы:

- `ptp4l` — служба протокола времени высокой точности, реализующая работу по протоколу времени высокой точности PTP в соответствии со стандартом

IEEE 1588. Точность протокола зависит от способа установки отметок времени (time stamping) в пакетах IEEE 1588. При программном методе установки отметок времени обеспечивается точность 1-100 микросекунд, на точность влияют прерывания, загрузка процессора и иные факторы. Аппаратная поддержка обеспечивает точность до единиц микросекунд;

- phc2sys — служба синхронизации часов;

- timemaster — служба координации, обеспечивающая совместную работу службы времени ntp и службы времени высокой точности ptp.

6.7.4.1. Проверка оборудования

Служба времени высокой точности ориентирована на использование аппаратных средств точного времени, в частности, аппаратных возможностей сетевых карт (аппаратные отметки времени).

Службу времени высокой точности можно настроить и использовать без сетевых карт, поддерживающих аппаратные возможности, но это повлечет снижение точности. Настройка использования сетевых карт без аппаратной поддержки отметок времени приведена в 6.7.4.3.

Проверить, поддерживает ли сетевая карта аппаратные отметки времени, можно из командной строки с помощью команды ethtool. Для этого в системе необходимо установить пакет ethtool, если он не был установлен ранее, командой:

```
apt install ethtool
```

затем выполнить проверку:

```
ethtool -T eth0
```

6.7.4.2. Установка

Служба времени высокой точности PTP устанавливается из пакета linuxptp командой:

```
apt install linuxptp
```

6.7.4.3. Настройка службы

Настройка службы timemaster

Настройка службы timemaster осуществляется с помощью конфигурационного файла /etc/linuxptp/timemaster.conf.

Подробно параметры настройки описаны в man timemaster.

Включение службы домена точного времени ptp_domain

Для включения службы ptp4l необходимо раскомментировать в конфигурационном файле /etc/linuxptp/timemaster.conf секцию [ptp_domain 0].

Пример

Настройки домена точного времени, использующего интерфейс eth0

```
[ptp_domain 0]
```

```
interfaces eth0
delay 10e-6
```

Домен точного времени обслуживается службой ptp4l. Настройка службы выполняется в соответствии с 6.7.4.3.

Включение и настройка службы ntp

Для включения службы ntp в конфигурационном файле /etc/linuxptp/timemaster.conf в секции [timemaster] необходимо указать демон ntpd вместо cronyd:

```
[timemaster]
ntp_program ntpd
```

После внесения этих изменений служба timemaster сможет запускать демон ntpd под своим контролем, самостоятельный запуск демона ntpd следует отключить:

```
systemctl mask ntp
```

Настройка автоматического запуска timemaster

Необходимо разрешить автоматический запуск службы timemaster при старте ОС:

```
systemctl enable timemaster
```

Настройка службы ptp4l

Настройка службы tp4l осуществляется с помощью конфигурационного файла /etc/linuxptp/ptp4l.conf.

При использовании сетевых карт без аппаратной поддержки отметок времени, в конфигурационном файле /etc/linuxptp/ptp4l.conf необходимо заменить аппаратную поддержку time_stamping hardware на программную time_stamping software.

Подробное описание настроек конфигурационного файла приведено в man pdp4l.

Настройка службы phc2sys

Служба phc2sys не требует настройки. Если в системе установлена сетевая карта, поддерживающая аппаратные отметки времени, которую необходимо синхронизировать с системными часами RTC, phc2sys запускается автоматически с нужными параметрами. При работе с сетевыми картами, не поддерживающими аппаратные отметки времени, служба phc2sys не запускается.

Запуск всех служб

После завершения настройки запуск всех служб осуществляется командой:

```
systemctl start timemaster
```

Служба timemaster запустит все остальные службы.

Пример

Результат вывода команды запроса статуса работы службы при штатном функционировании и наличии аппаратной поддержки

```
systemctl status timemaster
? timemaster.service - Synchronize system clock to NTP and PTP time sources
Loaded: loaded (/lib/systemd/system/timemaster.service; disabled;
      vendor preset: enabled)
Active: active (running) since Mon 2019-04-22 15:51:02 MSK; 2s ago
Docs: man:timemaster
Main PID: 2508 (timemaster)
Tasks: 5 (limit: 4608)
CGroup: /system.slice/timemaster.service
  2508 /usr/sbin/timemaster -f /etc/linuxptp/timemaster.conf
  2509 /usr/sbin/ntpd -u ntp:ntp -g -n -c /var/run/timemaster/ntp.conf
  2510 /usr/sbin/ptp4l -l 5 -f /var/run/timemaster/ptp4l.0.conf -H -i eth0
  2511 /usr/sbin/phc2sys -l 5 -a -r -R 1.00 -z /var/run/timemaster/
          ptp4l.0.socket -n 0 -E ntpshm -M 0
```

6.7.4.4. Настройка режима интерпретации показаний аппаратных часов

Чтобы исключить проблемы с коррекцией времени и сменой сезонного локального времени, рекомендуется настраивать аппаратные часы на всемирное координированное время (UTC). По умолчанию ОС настроена так, чтобы показания аппаратных часов трактовались как время UTC.

Режим интерпретации показаний аппаратных часов может быть включен при установке ОС. После установки ОС режим интерпретации показаний аппаратных часов включается с помощью графической утилиты `fly-admin-date` путем установки во вкладке «Дата и время» флага «Системные часы установлены на UTC».

Проверка показаний системного, локального и аппаратного времени выполняется командой:

```
timedatectl
```

Если ОС настроена так, что показания аппаратных часов трактуются как локальное время, при выполнении команды `timadatectl` будет выдано соответствующее предупреждение.

Настройка аппаратных часов на время UTC с одновременной синхронизацией с системным временем выполняется командой:

```
timedatectl set-local-rtc 0
```

Для настройки с одновременной синхронизацией системного времени по показаниям часов RTC следует использовать параметр `--adjust-system-clock`.

Настройка аппаратных часов на локальное время выполняется командой:

```
timedatectl set-local-rtc 1
```

6.8. Программный коммутатор Open vSwitch

Open vSwitch (OVS) — программный многоуровневый коммутатор, обеспечивающий агрегацию портов, обнаружение петель, зеркалирование портов, сбор статистики о трафике на NetFlow-коллектор, изоляцию сети с помощью сетей VLAN путем тегирования портов, а также фильтрацию базовой сети и централизованное управления программными коммутаторами с помощью протокола OpenFlow.

Архитектура OVS состоит из трех основных компонентов: базы данных, программного коммутатора и управляющего контроллера. На каждом из физических узлов вместе с гипервизором располагаются собственные БД и коммутатор. БД обеспечивает хранение всей конфигурации своего узла: настройки интерфейсов, портов, различные правила и прочее. Коммутатор передает пакеты. Распределенность OVS достигается с помощью контроллера.

Коммутация пакетов происходит на уровне ядра, также поддерживается коммутация в пользовательском пространстве. При коммутации в пользовательском пространстве снижается производительность по причине частых переключений между режимами ядра и пользователя.

ВНИМАНИЕ! Данное программное средство из состава ОС не поддерживает классификационные метки и может использоваться только на минимальном уровне конфиденциальности.

6.8.1. Установка

На каждом узле, предназначенном для включения в сеть, должен быть установлен пакет `openvswitch-switch` программного коммутатора OVS . Для установки пакета используется команда:

```
apt-get openvswitch-switch
```

6.8.2. Особенности конфигурирования

Конфигурация всех Open vSwitch коммутаторов, портов, настройки поддерживаемых протоколов хранятся в собственной базе данных OVS (OVSDB). В стандартной конфигурации в OVSDB существуют следующие таблицы:

- Open_vSwitch — Схема
- Bridge
- Port
- Interface
- Flow_Table — конфигурация OpenFlow

- QoS
- Mirror
- Controller — параметры подключения к контроллеру OpenFlow
- Manager — конфигурация OVSDB
- NetFlow
- SSL
- sFlow
- IPFIX
- Flow_Sample_Collector_Set

Изначально почти все таблицы пусты, так как конфигурация отсутствует. Утилита `ovs-vsctl` используется для внесения изменений в OVSDB. Команда для внесения изменений в БД имеет следующий синтаксис:

```
ovs-vsctl <команда> <таблица> <запись> <ключ=значение>
```

Для создания программного коммутатора используется команда вида:

```
ovs-vsctl add-br <имя_коммутатора>
```

Для добавления порта коммутатора используется команда:

```
ovs-vsctl add-port <имя_коммутатора> <имя_порта>
```

Просмотреть записи, присутствующие в таблице, описывающей порты, можно с помощью команды:

```
ovs-vsctl list port
```

Для вывода списка портов, включенных в конкретный VLAN, необходимо выполнить команду:

```
ovs-vsctl find port tag=10
```

Конфигурация OVS заданная командами `ovs-vsctl` автоматически применяется и сохраняется в файле `/etc/network/interfaces`.

6.9. Сетевая защищенная файловая система

6.9.1. Назначение и возможности

Для организации защищенных файловых серверов предназначена сетевая защищенная ФС (СЗФС), в основу которой положена CIFS, работающая по протоколу SMB/CIFS. Протокол СЗФС содержит в себе сообщения, которые передают информацию о мандатном контексте (метке безопасности и дополнительных мандатных атрибутах управления доступом) субъекта доступа. Подробное описание мандатого контекста приведено в документе РУСБ.10152-02 97 01-1.

Условием корректного функционирования СЗФС является использование механизма ЕПП, обеспечивающее в рамках данной ЛВС однозначное соответствие между логическим именем пользователя и его идентификатором (а также именем группы и ее идентификато-

ром) на всех компьютерах (рабочих станциях и серверах), на которых данный пользователь может работать. Для корректной работы СЗФС необходима синхронизация UID/GID в системах клиента и сервера, т. к. информация о пользователях и группах передается в сеть в численных значениях.

СЗФС предоставляет следующие базовые возможности:

- разделение файловой системы ОС «Astra Linux Special Edition» операционной системой типа Windows и наоборот;
- совместное использование принтеров, подключенных к ОС «Astra Linux Special Edition», операционной системой типа Windows и наоборот.

6.9.2. Состав

Основой СЗФС является клиент-серверная архитектура.

Сервер представляет собой расширенный сервер Samba и выполняет следующие задачи:

1) управление разделяемыми ресурсами;
2) контроль доступа к разделяемым ресурсам. При подключении клиента сервер устанавливает метку безопасности процесса, обслуживающего запросы клиента, в соответствии с меткой безопасности этого клиента. Этим обеспечивается мандатный контроль доступа к разделяемым файлам на стороне сервера.

Клиент представляет собой сетевую ФС в составе системы управления файлами ядра ОС и реализует интерфейс между виртуальной ФС ядра и сервером СЗФС. Клиент СЗФС выполняет следующие задачи:

1) отображение каталогов и файлов смонтированного сетевого ресурса;
2) передача на сервер дополнительной информации о классификационной метке пользователя (процесса), работающего с разделяемым ресурсом.

С точки зрения пользователя, СЗФС выглядит как стандартная ФС, поддерживающая все механизмы защиты ОС и позволяющая работать с удаленной ФС с помощью стандартных команд.

В состав СЗФС входят следующие компоненты:

- `smbd` — служба сервера, которая обеспечивает работу службы печати и разделения файлов для клиентов операционной системы типа Windows. Конфигурационные параметры службы `smbd` описываются в файле `smb.conf`;
- `nmbd` — служба сервера, которая обеспечивает работу службы имен NetBIOS, а также может использоваться для запроса других служб имен;
- `smbclient` — служба, которую реализует клиент, используемый для доступа к другим серверам и для печати на принтерах, подключенных к серверам;

- testparm — команда, позволяющая протестировать конфигурационный файл smb.conf;

- smbstatus — команда, выводящая информацию о том, кто в настоящее время пользуется сервером Samba.

6.9.3. Настройка

СЗФС устанавливается в процессе установки ОС.

Основная настройка СЗФС в ОС осуществляется путем редактирования конфигурационного файла /etc/samba/smb.conf.

Файл /etc/samba/smb.conf состоит из основных именованных разделов [global], [homes] и [printers], возможно добавление пользовательских разделов. Внутри каждого раздела находится ряд параметров вида <имя_параметра> = <значение>.

В разделе [global] описаны параметры, управляющие сервером Samba в целом, а также находятся значения параметров по умолчанию для других разделов.

Примеры:

1. Фрагмент конфигурационного файла, определяющий рабочую группу WORKGR1, к которой относится компьютер, а также описывающий саму систему.

```
[global];
;workgroup = NT-Domain-Name или Workgroup-Name
workgroup = WORKGR1
;comment эквивалентен полю описания NT (Description field)
comment = Сервер СЗФС
```

2. Фрагмент конфигурационного файла, описывающий тип системы печати, доступной на сервере администратора, а также местонахождение конфигурационного файла принтера. Последняя строка говорит о том, что все принтеры, определенные в файле printcap, должны быть доступны в сети.

```
;printing = BSD или SYSV или A1X (и т.д.)
printing = bsd
printcap name = /etc/printcap
load printers = yes
```

3. Фрагмент конфигурационного файла, определяющий поддержку сервером гостевого входа. Следующие два параметра определяют работу с журнальными файлами. Параметр m сообщает службе Samba, что для каждого клиента ведется свой файл, а последняя строка говорит о том, что максимальный размер создаваемого журнального файла — 50 КБ.

```
;Раскомментируйте это поле, если вам нужен гостевой вход
```

```
;guest = pcguest
log file = /var/log/samba-log.%m
max log size = 50
```

Раздел [homes] позволяет подключаться к рабочим каталогам пользователей без их явного описания. При запросе клиентом определенной службы ищется соответствующее ей описание в файле и, если такового нет, просматривается раздел [homes]. Если этот раздел существует, просматривается файл паролей для поиска рабочего каталога пользователя, сделавшего запрос, и, найдя его, он становится доступным по сети. Основные параметры раздела [homes]:

1) comment — значение параметра выводится для клиента при запросе о доступных ресурсах;

2) browseable — определяет, как выводить ресурс в списке просмотра;

3) read only — определяет, может ли пользователь создавать и изменять файлы в своем рабочем каталоге при подключении по сети;

4) create mask — определяет права доступа для вновь создаваемых файлов в рабочем каталоге пользователя.

Пример

```
[homes]
comment = Home Directories
browseable = no
case sensitive = yes
read only = yes
create mask = 0700
directory mask = 0700
ea support = yes
```

Раздел [printers] используется для предоставления доступа к принтерам, определенным в файле /etc/. В разделе [printers] описываются параметры управления печатью при отсутствии иного явного описания. Параметры comment, browseable, read only, create mask аналогичны параметрам раздела [homes], остальные параметры:

1) path — определяет местонахождение файла спулера при печати через SMB;

2) printable — определяет, может ли использоваться данный ресурс для печати;

3) guest ok — определяет, может ли воспользоваться принтером гостевой пользователь.

Пример

```
[printers]
```

```
comment = All Printers
browseable = no
path = /var/spool/samba
printable = no
guest ok = no
read only = yes
create mask = 0700
```

После настройки параметров сервера по умолчанию можно создать разделяемые каталоги, доступ к которым могут получать определенные пользователи, группы пользователей или все пользователи.

Пример

Создание разделяемого каталога с доступом только для одного пользователя. Для этого необходимо создать отдельный раздел файла `smb.conf` и заполнить его необходимой информацией (обычно это пользователь, каталог и конфигурационная информация)

```
[User1]
comment = User1’ s remote source code directory
path = /usr/locaI/src
valid users = victor
browseable = yes
public = no
writeable = yes
create mode = 0700
```

В данном разделе создается разделяемый каталог с именем `User1`. На локальном сервере его путь — `/usr/local/src`, `browseable = yes`, поэтому ресурс будет виден в списках ресурсов сети, но т. к. `public = no`, получить доступ к нему сможет только пользователь `victor`. Предоставить доступ другим пользователям можно, поместив их в запись `valid users`.

По умолчанию сервер Samba поддерживает подключение по протоколу SMB всех версий, а клиент при подключении начинает процедуру согласования протокола подключения со старшей версии. Для принудительного определения диапазона возможных протоколов используются параметры конфигурационного файла `/etc/samba/smb.conf`, приведенные в таблице 30.

Таблица 30

Имя параметра	Синоним параметра	Значение по умолчанию	Описание
`server min protocol`	`min protocol`	`NT1`	Минимальная версия протокола сервера
`server max protocol`	`max protocol`, `protocol`	`SMB3_11`	Максимальная версия протокола сервера
`client min protocol`		`NT1`	Минимальная версия протокола клиента
`client max protocol`		`SMB3_11`	Максимальная версия протокола клиента
`client ipc min protocol`		`NT1` (значение параметра `client min protocol`)	Минимальная версия протокола клиента для межпроцессного взаимодействия
`client ipc max protocol`		`SMB3_11`	Максимальная версия протокола клиента для межпроцессного взаимодействия

При этом допустимые значения параметров, указанных в таблице 30, в зависимости от версии протокола приведены в таблице 31.

Таблица 31

Версия протокола	Значение	Примечание
SMB v1	`NT1`	

SMB v2	`SMB2`
	`SMB2_02`
	`SMB2_10`
	`SMB2_22`
	`SMB2_24`
		`SMB2 = SMB2_10`

SMB v3	`SMB3`
	`SMB3_00`
	`SMB3_02`
	`SMB3_10`
	`SMB3_11`
		`SMB3 = SMB3_11`

В зависимости от реализации клиент Samba может принудительно требовать от сервера версию протокола. Обычно версия протокола задается одним из параметров подключения и имеет собственную нотацию. Способы конфигурирования протокола в зависимости от типа клиента, а также допустимые значения приведены в таблице 32.

Таблица 32

Утилита	Конфигурирование	Допустимые значения	Значение по умолчанию
mount.cifs	Применение параметра монтирования vers=	1.0 2.0 2.1 3.0 3.02 3.1.1 3.11	3.11
smbclient	Использование параметра -m | --max-protocol с инструментом командной строки	NT1 SMB2 SMB3	Определяется параметрами в /etc/samba/smb.conf
Клиент ALD	Значение параметра CLIENT_SMB_VERSION в файле /etc/ald/ald.conf	1.0 2.0 2.1 3.0 3.02 3.1.1 3.11	3.11

После редактирования конфигурационного файла /etc/smb.conf необходимо протестировать его корректность при помощи команды testparm, которая проверяет наличие в файле внутренних противоречий и несоответствий.

Примечание. Выполнение testparm не подтверждает, что все службы и ресурсы, описанные в конфигурационном файле, доступны и будут корректно работать.

Команда testparm имеет следующий синтаксис:

```
testparm [configfile [hostname hostip]]
```

Параметр configfile определяет местоположение конфигурационного файла (если это не файл /etc/smb.conf). Параметр hostname hostip указывает команде testparm проверить доступ к службам со стороны узла, определяемого параметром.

Если ошибки не будут обнаружены, на экране появится сообщение вида:

```
it testparm
Load smb config files from /etc/smb.conf
Processing section "[homes]"
Processing section "[printers]"
Loaded services file OK.
Press enter to see a dump of your service definitions
```

При нажатии клавиши **<Enter>** testparm протестирует каждый раздел, определенный в конфигурационном файле.

В случае обнаружения ошибок о них будет предоставлена полная информация.

6.9.4. Графическая утилита настройки СЗФС

В состав ОС входит графическая утилита `fly-admin-samba`, которая позволяет настроить пользовательский доступ к ресурсам СЗФС. Установка утилиты выполняется командой:

```
apt install fly-admin-samba
```

Описание использования утилиты приведено в электронной справке.

6.9.5. Запуск сервера

Сервер запускается либо из инициализирующих сценариев, либо из `inetd` в качестве системной службы.

Если сервер запускается из сценариев инициализации, то можно воспользоваться для запуска и остановки работы сервера следующей командой:

```
systemctl {start|stop} smbd
```

Доступ пользователей ОС к ресурсам сервера осуществляется с помощью монтирования СЗФС. Другой возможностью является использование графической утилиты `fly-admin-samba` (см. электронную справку).

Инструмент командной строки `smbclient` позволяет получить информацию о разделяемых ресурсах или перенести файлы. Например, для запроса списка доступных ресурсов на удаленном сервере `win.netwhart.com` используется команда:

```
smbclient -L -I win.netwhart.com
```

где `-L` — указывает, что требуется вывести список разделяемых ресурсов;

`-I` — указывает, что указанное далее имя следует рассматривать как имя DNS, а не NetBIOS.

Для пересылки файла необходимо сначала подключиться к серверу путем выполнения команды:

```
smbclient '\\WORKGR1\PUBLIC' -I win.netwhart.com -U tackett
```

где `\\WORKGR1\PUBLIC` — определяет удаленную службу на другом компьютере (обычно это каталог ФС или принтер);

`-U` — позволяет определить имя пользователя для подключения к ресурсу (при этом, если необходимо, СЗФС запросит соответствующий пароль).

После подключения появится приглашение:

```
Smb:\
```

где `\` — текущий рабочий каталог.

Используя инструмент командной строки `smbclient` можно указать команды для передачи файлов и работы с ними. Дополнительно описание параметров инструмента приведено в руководстве `man smbclient`.

6.9.6. Правила конвертации меток целостности

В ОС используется метка целостности, которая может принимать значение 256 и более.

Для штатной работы СЗФС Samba из состава ОС с СЗФС Samba других систем, в которых максимальное значение метки целостности составляет 255, реализована совместимость меток целостности. При передаче из ОС файла с меткой целостности, значение которой составляет 256 или более, в систему с максимальным значением метки целостности равным 255, метка целостности передаваемого файла будет преобразована в максимальное значение 255, т.е. будет выполнено понижение целостности при передаче файла.

Подробное описание метки целостности ОС приведено в документе РУСБ.10152-02 97 01-1.

6.10. Средство создания защищенных каналов

Для создания защищенных каналов типа точка-точка или сервер-клиент между компьютерами сети используется свободная реализация технологии виртуальной частной сети (VPN) с открытым исходным кодом OpenVPN. Данная технология позволяет устанавливать соединения между компьютерами, находящимися за NAT и сетевым экраном, без необходимости изменения их настроек.

Для обеспечения безопасности управляющего канала и потока данных OpenVPN использует библиотеку `OpenSSL` (устанавливается автоматически при установке ОС). При этом OpenVPN использует алгоритмы защитного преобразования, которые запрашивает и получает от `OpenSSL`.

Поставляемый в составе дистрибутива вариант OpenVPN поддерживает работу с динамически подключаемой библиотекой `OpenSSL` алгоритмов защитного преобразования в соответствии с требованиями ГОСТ (пакет библиотеки алгоритмов ГОСТ `libgost-astra`).

Дополнительная информация по применению OpenVPN и библиотеки алгоритмов ГОСТ `libgost-astra` доступна на сайте `wiki.astralinux.ru` по ссылке `https://wiki.astralinux.ru/display/doc/OpenVPN`.

6.10.1. Установка

Установка программного продукта OpenVPN выполняется либо из графического менеджера пакетов Synaptic, либо из терминала.

Для установки OpenVPN из терминала необходимо:

1) на компьютере, предназначенном на роль сервера OpenVPN, и на клиентских компьютерах установить пакет `openvpn`:

```
apt-get install openvpn
```

2) на компьютере, предназначенном на роль сервера, для управления службой openvpn установить графическую утилиту fly-admin-openvpn-server или инструмент командной строки astra-openvpn-server, выполнив соответствующую команду:

```
apt-get install fly-admin-openvpn-server
apt-get install astra-openvpn-server
```

Примечания:

1. При установке графической утилиты автоматически будет установлен инструмент командной строки astra-openvpn-server.

2. При установке инструмента командной строки astra-openvpn-server будет автоматически установлен и настроен пакет алгоритмов защитного преобразования ГОСТ libgost-astra;

6.10.2. Управление с помощью инструмента командной строки

6.10.2.1. Параметры инструмента командной строки

Команды, используемые с инструментом командной строки astra-openvpn-server, приведены в таблице 33.

Таблица 33

Параметр	Описание
Информационные команды	
-h, --help	Вывод справки
-v, --version	Вывод версии
--show-ciphers	Вывод списка поддерживаемых ключей
Управление выводом	
-s	Не выводить сообщения и предупреждения. Может быть указана в любом месте. Отменяет вывод комментариев о ходе выполнения, предупреждений, сообщений об ошибках
Управление сервером	
start	Запустить службу openvpn. При выполнении этой команды без указания дополнительных параметров служба будет запущена со стандартной конфигурацией из файла /etc/openvpn/server.conf. Если файл конфигурации, ключи и сертификаты сервера не существуют, то они будут созданы с параметрами по умолчанию. С данной командой дополнительно могут быть заданы параметры сервера, указаны файлы для аутентификации и параметры аутентификации
stop	Остановить службу. После выполнения данной команды другие команды не выполняются
status	Проверить службу. После выполнения данной команды другие команды не выполняются

Продолжение таблицы 33

Параметр	Описание
`rebuild-server-certs`	Остановить службу, удалить все сертификаты сервера и клиентов, повторно сгенерировать все сертификаты сервера и запустить сервер. Имена файлов сертификатов сервера берутся из файла конфигурации сервера. Если файл конфигурации отсутствует, то остальные действия не выполняются. После выполнения данной команды другие команды не выполняются
Параметры сервера	
`server <IP-адрес> <маска>`	IP-адрес и маска создаваемой сети VPN (по умолчанию IP=10.8.0.0 и MASK=255.255.255.0), например: `astra-openvpn-server server "10.0.0.8 255.255.255.0"`
`port <порт>`	Порт (по умолчанию 1194)
`cipher <метод>`	Метод защитного преобразования данных (по умолчанию `grasshopper-cbc`). Поддерживаются следующие методы защитного преобразования: - `grasshopper-cbc` — алгоритм «Кузнечик» ГОСТ Р 34.13-2015; - `AES-256-GCM` — рекомендован для применения в системах общего назначения; - `AES-256-CBC` — допустим для применения в системах общего назначения; - `AES-128-CBC` — используется для совместимости со старыми системами, к применению не рекомендуется
Указание файлов для аутентификации	
`cert <имя_файла>.crt`	Файл сертификата пользователя
`ca <имя_файла>.crt`	Файл сертификата удостоверяющего центра
`key <имя_файла>.key`	Личный ключ
`dh <имя_файла>.pem`	Файл Диффи-Хеллмана
`tls-auth <имя_файла>.key`	Файл аутентификации TLS
Параметры аутентификации	
`EASYRSA_REQ_COUNTRY`	Название страны
`EASYRSA_REQ_PROVINCE`	Название области
`EASYRSA_REQ_CITY`	Название города
`EASYRSA_REQ_ORG`	Название организации
`EASYRSA_REQ_EMAIL`	Адрес электронной почты
`EASYRSA_REQ_OU`	Название подразделения организации
`EASYRSA_REQ_CN`	Имя пользователя
Генерация и отзыв ключей клиентов	
`client <имя_клиента>`	Создать ключи и сертификаты для указанного клиента

Окончание таблицы 33

Параметр	Описание
`revoke <имя_клиента>`	Отозвать сертификат указанного клиента
Параметры индивидуальной настройки сервера	
`get <параметр>`	Прочитать значение параметра из файла конфигурации `/etc/openvpn/server.conf`. Если файл конфигурации не существует, то он будет создан с параметрами по умолчанию
`del <параметр>`	Удалить значение параметра из файла конфигурации `/etc/openvpn/server.conf`. Если файл конфигурации не существует, то он будет создан с параметрами по умолчанию, после чего указанный параметра будет удалн
`set <параметр> <значение>`	Записать значение параметра в файл конфигурации `/etc/openvpn/server.conf`. Если файл конфигурации не существует, то он будет создан с параметрами по умолчанию, после чего в файл будет записано указанное значение

Примечания:

1. Если в командной строке заданы информационные команды, то будет выполнена первая из них. Дальнейшее выполнение сценария будет прекращено.
2. Команды управления сервером несовместимы с командами генерации и отзыва ключей для клиентов.
3. Полный список параметров индивидуальной настройки сервера доступен в документации на OpenVPN.

6.10.2.2. Запуск службы

Для запуска службы `openvpn` из терминала ввести команду:

```
astra-openvpn-server start
```

При запуске службы будут созданы следующие стандартные файлы и каталоги:

- файл конфигурации службы `openvpn`:

`/etc/openvpn/server.conf`

- локальный удостоверяющий центр, размещается в каталоге:

`/etc/openvpn/openvpn-certificates`

- сертификат открытого ключа удостоверяющего центра:

`/etc/openvpn/keys/ca.crt`

- сертификат открытого ключа:

`/etc/openvpn/keys/server.crt`

- закрытый ключ сервера:

`/etc/openvpn/keys/server.key`

- файл параметров Диффи-Хеллмана для авторизации пользователей:

`/etc/openvpn/keys/dh2048.pem`

- файл дополнительной аутентификации TLS:

`/etc/openvpn/keys/ta.key`

- дополнительно, при выполнении отзыва сертификатов, будет создан стандартный файл списка отзыва сертификатов:

`/etc/openvpn/keys/crl.pem`

Также при первом запуске службы будут выполнены настройки межсетевого экрана и другие настройки ОС для работы `openvpn` как стандартной системной службы с автоматическим запуском при включении компьютера.

Запуск команды `astra-openvpn-server start` с указанием файлов для аутентификации (см. таблицу 33) позволяет при создании файла конфигурации и запуске службы `openvpn` задать расположение ранее установленных файлов ключей и сертификатов.

ВНИМАНИЕ! Чтобы избежать запроса пароля при автоматическом запуске службы `openvpn` необходимо файлы создавать без применения защитного преобразования.

Пример

Запуск сервера с указанием ранее установленных файлов ключей и сертификатов

```
astra-openvpn-server start cert /root/secrets/server.crt
    ca /root/secrets/ca.crt key /root/secrets/server.key
    dh /root/secrets/dh2048.pem tls-auth /root/secrets/ta.key
```

Указание файлов для аутентификации несовместимо с указанием параметров идентификации (см. таблицу 33).

ВНИМАНИЕ! В случае если указан хотя бы один файл для аутентификации, то все файлы будут проверены на существование. При отсутствии одного из файлов сценарий будет завершн с ошибкой без выполнения каких-либо действий. Проверка файлов на корректность не выполняется.

ВНИМАНИЕ! Если заданы файлы для аутентификации, то создание собственного удостоверяющего центра не выполняется.

6.10.2.3. Генерация сертификатов и ключей

При использовании собственного удостоверяющего центра создание ключей и сертификатов для клиентов осуществляется на сервере OpenVPN с помощью инструмента командной строки `astra-openvpn-server`. Для создания клиентского комплекта файлов используется команда `client`:

```
astra-openvpn-server client <имя_клиента>
```

При генерации могут быть заданы параметры аутентификации (см. таблицу 33).

Команда генерации ключей клиента несовместима с параметрами сервера и командами управления сервером (см. таблицу 33).

При выполнении данной команды для указанного клиента будет создан новый файл закрытого ключа <имя_клиента>.key и файл сертификата открытого ключа <имя_клиента>.crt, подписанный удостоверяющим центром.

Для удобства последующей передачи файлов ключей клиенту, созданные файлы будут скопированы в каталог /etc/openvpn/clients-keys/<имя_клиента>. Дополнительно в каталог будут скопированы и другие, необходимые для работы клиента, файлы: файл сертификата удостоверяющего центра (по умолчанию ca.crt) и файл дополнительной аутентификации TLS (ta.key).

Дополнительно при создании пользовательских ключей могут быть указаны такие параметры аутентификации, как страна, город, организация и др. (см. таблицу 33). В таблице 33 приведены значения параметров аутентификации, используемые по умолчанию при генерации сертификатов.

ВНИМАНИЕ! Если задан любой из параметров аутентификации, то будет произведена автоматическая генерация сертификатов.

Пример

Задание дополнительных параметров аутентификации при выполнении команды создания сертификатов для клиента

```
astra-openvpn-server client ivanov \
EASYRSA_REQ_COUNTRY RU \
EASYRSA_REQ_PROVINCE MO \
EASYRSA_REQ_CITY MOSCOW \
EASYRSA_REQ_ORG COMPANY \
EASYRSA_REQ_EMAIL ivanov@company.ru
```

ВНИМАНИЕ! Клиентские ключи генерируются без применения защитных преобразований, чтобы избежать ввода пароля при подключении клиента к серверу.

Параметры аутентификации несовместимы с указанием файлов для аутентификации (см. таблицу 33).

6.10.2.4. Отзыв сертификатов

Отзыв сертификатов применяется для запрета подключений клиента даже в тех случаях, когда в распоряжении клиента имеются копии всех сертификатов и ключей.

Для отзыва сертификата используется команда revoke инструмента командной строки astra-openvpn-server:

```
astra-openvpn-server revoke <имя_клиента>
```

Команда отзыва ключей клиента несовместима с параметрами сервера и командами управления сервером (см. таблицу 33).

При выполнении данной команды:

- сертификат клиента в базе данных удостоверяющего центра будет помечен как «отозванный»;
- будет создан (или обновлн ранее созданный) список отозванных сертификатов;
- новый список отозванных сертификатов будет скопирован в каталог `/etc/openvpn/keys`, сервер OpenVPN будет автоматически перезапущен для применения обновлений.

6.10.2.5. Замена сертификатов

Полная замена сертификатов сервера выполняется с помощью инструмента командной строки `astra-openvpn-server`:

```
astra-openvpn-server rebuild-server-certs
```

При выполнении данной команды:

- останавливается служба `openvpn`;
- удаляются все файлы удостоверяющего центра;
- удаляются все копии сертификатов сервера и клиентов;
- создается новый удостоверяющий центр;
- создаются новые сертификаты сервера;
- повторно запускается сервер.

Имена файлов сертификатов сервера берутся из файла конфигурации сервера. Если файл конфигурации отсутствует, то никакие действия не выполняются. После выполнения данной команды другие команды не выполняются.

6.10.2.6. Настройка клиента

На компьютер клиента должны быть перенесены файлы ключей и сертификатов, созданные на сервере, либо с помощью отчуждаемого носителя информации, либо путем передачи по защищенному соединению (например, `ssh`).

Для настройки компьютера клиента следует установить программное обеспечение OpenVPN. Установка выполняется либо из графического менеджера пакетов Synaptic, либо из терминала командой:

```
apt-get install openvpn
```

После установки программного обеспечения OpenVPN следует выполнить следующие действия:

1) создать файл конфигурации клиента. В качестве исходного файла возможно использовать входящий в комплект установки OpenVPN стандартный шаблон файла конфигурации, предоставляемый разработчиками OpenVPN. Шаблон файла конфигурации расположен в `/usr/share/doc/openvpn/examples/sample-config-files/client.conf`.

Шаблон файла следует скопировать в каталог /etc/openvpn/client, выполнив команду:

```
cp /usr/share/doc/openvpn/examples/sample-config-files/client.conf
     /etc/openvpn/client
```

2) в скопированном файле конфигурации внести следующие исправления:

а) для параметра remote указать в качестве его значения IP-адрес сервера OpenVPN. Если был изменен порт, то также указать данное значение вместо стандартного;

б) в строках:

```
;user nobody
;group nogroup
```

удалить начальные символы «;»:

```
user nobody
group nogroup
```

в) для параметров ca, cert и key указать расположение соответствующих файлов сертификатов и ключа для авторизации, например:

```
ca /etc/openvpn/keys/ca.crt
cert /etc/openvpn/keys/home-pc.crt
key /etc/openvpn/keys/home-pc.key
```

г) для параметра tls-auth указать расположение файла дополнительной аутентификации TLS, например:

```
tls-auth /etc/openvpn/keys/ta.key
```

д) для параметра cipher указать метод защитного преобразования данных, используемый службой. Используемый метод защитного преобразования можно узнать на сервере OpenVPN с помощью инструмента командной строки astra-openvpn-server (команда astra-openvpn-server get cipher), либо с помощью графического инструмента fly-admin-openvpn-server. Защитному преобразованию в соответствии с алгоритмами «Кузнечик» по ГОСТ Р 34.12-2015 соответствует значение grasshopper-cbc;

е) сохранить исправленный файл.

Для проверки работы клиента OpenVPN из командной строки использовать команду:

```
/usr/sbin/openvpn --config /etc/openvpn/client/client.conf
```

где client.conf — конфигурационный файл клиента.

Для запуска клиента OpenVPN в качестве службы выполнить команду:

```
systemctl start openvpn-client@<имя_файла_конфигурации>
```

где <имя_файла_конфигурации> — имя файла конфигурации без расширения, расположенного в каталоге /etc/openvpn/client.

6.10.3. Управление службой с помощью графической утилиты

После установки графической утилиты `fly-admin-openvpn-server` для ее запуска перейти меню «Пуск — Панель управления — Прочее».

В графической утилите доступны:

- вкладка «Настройки» — в ней располагаются элементы управления для настройки сервера OpenVPN. По умолчанию доступны базовые настройки, расширенные настройки становятся доступны после нажатия кнопки **[Показать расширенные настройки]**. Описание настроек приведено в 6.10.3.2;
- вкладка «Клиентские сертификаты» — в ней располагаются элементы управления клиентскими сертификатами. Описание управления сертификатами приведено в 6.10.3.3;
- кнопки **[Запустить]** и **[Остановить]** — служат для управления службой `openvpn`.

6.10.3.1. Управление службой

Для запуска службы `openvpn` используя графическую утилиту необходимо:

1) запустить графическую утилиту согласно 6.10.3. При первом запуске графической утилиты будет создана конфигурация службы `openvpn` по умолчанию и будут выпущены сертификаты сервера;
2) при необходимости отредактировать конфигурацию и сертификаты;
3) нажать кнопку **[Запустить]**.

ВНИМАНИЕ! Графическая утилита при ее запуске не производит автоматический запуск службы `openvpn`.

При запуске службы будут созданы следующие стандартные файлы и каталоги:

- файл конфигурации службы `openvpn`:

`/etc/openvpn/server.conf`

- локальный удостоверяющий центр, размещается в каталоге:

`/etc/openvpn/openvpn-certificates`

- сертификат открытого ключа удостоверяющего центра:

`/etc/openvpn/keys/ca.crt`

- сертификат открытого ключа:

`/etc/openvpn/keys/server.crt`

- закрытый ключ сервера:

`/etc/openvpn/keys/server.key`

- файл параметров Диффи-Хеллмана для авторизации пользователей:

`/etc/openvpn/keys/dh2048.pem`

- файл дополнительной аутентификации TLS:

`/etc/openvpn/keys/ta.key`

- дополнительно, при выполнении отзыва сертификатов, будет создан стандартный файл списка отзыва сертификатов:

`/etc/openvpn/keys/crl.pem`

В случае, если на компьютере установлены и настроены библиотеки, поддерживающие метод защитного преобразования по алгоритму ГОСТ Р 34.12-2015 («Кузнечик»), для защиты канала данных будет выбран данный метод. В противном случае будет выбран метод защитного преобразования `AES-256-GCM`.

Также при первом запуске службы будут выполнены настройки межсетевого экрана и другие настройки ОС для работы `openvpn` как стандартной системной службы с автоматическим запуском при включении компьютера.

Для остановки службы `openvpn` используя графическую утилиту необходимо нажать кнопку **[Остановить]**.

6.10.3.2. Настройка службы

Настройка службы выполняется во вкладке «Настройки» графической утилиты.

Базовые настройки включают:

1) «IP-адрес» — позволяет задать IP-адрес создаваемой сети VPN. По умолчанию установлено значение `10.8.0.0`;

2) «Маска» — позволяет задать маску создаваемой сети VPN. По умолчанию установлено значение `255.255.255.0`;

3) «Порт» — сетевой порт сервера, который будут использовать клиенты для подключения. По умолчанию установлено значение `1194` . Поддерживаются номера свободных портов от 1 до 65535;

4) «Метод защитного преобразования» — по умолчанию установлено значение `grasshopper-cbc` («Кузнечик»). Поддерживаются следующие методы:

а) `grasshopper-cbc` — алгоритм «Кузнечик» ГОСТ Р 34.13-2015;

б) `AES-256-GCM` — рекомендован для применения в системах общего назначения;

в) `AES-256-CBC` — допустим для применения в системах общего назначения;

г) `AES-128-CBC` — используется для совместимости со старыми системами, к применению не рекомендуется.

Расширенные настройки позволяют задать расположение ранее предустановленных файлов ключей и сертификатов внешнего удостоверяющего центра, а также заново выпустить сертификаты локального удостоверяющего центра.

Для указания расположения ранее предустановленных файлов ключей и сертификатов внешнего удостоверяющего центра используются следующие поля:

- «Сертификат пользователя» — сертификат открытого ключа;

- «Сертификат ЦС» — сертификат открытого ключа удостоверяющего центра;
- «Личный ключ» — закрытый ключ сервера;
- «Файл Диффи-Хеллмана» — файл параметров Диффи-Хеллмана;
- «Файл аутентификации TLS» — файл дополнительной аутентификации TLS.

Проверка файлов на корректность не проводится.

Кнопка **[Сбросить сертификаты]** предназначена для удаления всех сертификатов локального удостоверяющего центра и повторного выпуска сертификатов сервера. После выполнения этого действия сертификаты клиентов станут недействительными, и клиенты потеряют возможность подключения к серверу OpenVPN. При выполнении данного действия:

- останавливается служба `openvpn`;
- удаляются все файлы удостоверяющего центра;
- удаляются все копии сертификатов сервера и клиентов;
- создается новый удостоверяющий центр;
- создаются новые сертификаты сервера;
- повторно запускается сервер.

6.10.3.3. Управление сертификатами

Управление сертификатами выполняется во вкладке «Клиентские сертификаты» графической утилиты.

В данной вкладке расположены таблица с данными о клиентских сертификатах и кнопки управления:

1) **[Создать сертификат]** — создание ключа и сертификата пользователя. При нажатии на кнопку будет открыто диалоговое окно с полями:

а) «Имя пользователя» — имя сертификата. Имя сертификата должно быть уникальным, не может быть пустым и не может содержать пробелы;

б) «Страна» — двухбуквенный код страны. Если поле пустое, то по умолчанию будет установлено значение «RU»;

в) «Область» — название области. Если поле пустое, то по умолчанию будет установлено значение «MO»;

г) «Город» — название города. Если поле пустое, то по умолчанию будет установлено значение «Moscow»;

д) «Организация» — название организации. Если поле пустое, то по умолчанию будет установлено значение «none»;

е) «Email» — адрес электронной почты. Если поле пустое, то по умолчанию будет установлено значение «none»;

ж) «Отдел» — название подразделения организации. Если поле пустое, то по умолчанию будет установлено значение «none»;

з) «Имя» — имя пользователя. Если поле пустое, то по умолчанию будет установлено значение «noname»;

При нажатии на кнопку **[Да]** будет создан новый файл закрытого ключа `<имя_клиента>.key` и файл сертификата открытого ключа `<имя_клиента>.crt`, подписанный удостоверяющим центром.

Для удобства последующей передачи файлов ключей клиенту, созданные файлы будут скопированы в каталог `/etc/openvpn/clients-keys/<имя_клиента>`. Дополнительно в каталог будут скопированы и другие, необходимые для работы клиента, файлы: файл сертификата удостоверяющего центра (по умолчанию `ca.crt`) и файл дополнительной аутентификации TLS (`ta.key`).

ВНИМАНИЕ! Клиентские ключи генерируются без применения защитных преобразований, чтобы избежать ввода пароля при подключении клиента к серверу;

2) **[Удалить сертификат]** — отзыв клиентских сертификатов. Отзыв сертификатов применяется для запрета подключений клиента даже в тех случаях, когда в распоряжении клиента имеются копии всех сертификатов и ключей. Для отзыва сертификата выбрать в таблице клиентов строку с отзываемым сертификатом и нажать данную кнопку. При нажатии на данную кнопку будут выполнены следующие действия:

- сертификат клиента в базе данных удостоверяющего центра будет помечен как «отозванный»;
- будет создан (или обновлн ранее созданный) список отозванных сертификатов;
- новый список отозванных сертификатов будет скопирован в каталог `/etc/openvpn/keys`, и сервер OpenVPN будет автоматически перезапущен для применения обновлений;

3) **[Открыть каталог сертификатов]** — открытие каталога `/etc/openvpn/keys` в файловом менеджере.

6.10.3.4. Настройка клиента

Настройка сетевых подключений клиентских компьютеров осуществляется с помощью графической утилиты `network-manager-openvpn`. Установка утилиты выполняется командой:

```
apt-get install network-manager-openvpn network-manager-openvpn-gno
```

Для настройки клиентского подключения нажать левой кнопкой мыши на значок сетевых соединений в области уведомлений панели задач и в раскрывшемся меню выбрать «Соединения VPN — Добавить VPN соединение».

Для создания нового соединение в открывшемся окне из выпадающего списка выбрать «OpenVPN» и нажать **[Создать]**.

В появившейся экранной форме необходимо:

1) в поле «Шлюз» указать IP-адрес ранее запущенного сервера OpenVPN;

2) в поле «Тип» оставить значение по умолчанию «Сертификат TLS»;

3) в поле «Сертификат CA» указать путь к скопированному файлу сертификата удостоверяющего центра `ca.crt` (6.10.2.3);

4) в поле «Сертификат пользователя» указать путь к скопированному файлу сертификата открытого ключа пользователя `<имя_клиента>.crt` (6.10.2.3);

5) в поле «Приватный ключ Пользователя» указать путь к файлу закрытого ключа `<имя_клиента>.key` (6.10.2.3);

6) нажать кнопку **[Дополнительно]**;

7) в открывшейся экранной форме перейти во вкладку «Аутентификация TLS»;

8) отметить пункт «Использовать дополнительную аутентификацию TLS», указать ранее скопированный на компьютер пользователя файл дополнительной аутентификации TLS и обязательно выбрать направление ключа «1».

Все остальные настройки можно оставить заданными по умолчанию. После нажатия кнопки **[ОК]** созданное VPN-соединение будет сохранено.

Для включения сохраннного соединения нужно повторно нажать левой кнопкой мыши на значок сетевых подключений в области уведомлений панели задач, в раскрывшемся меню выбрать «Соединения VPN» и отметить включаемое соединение.

Для экспорта параметров созданного клиентского соединения с целью их повторного использования на других клиентах выполнить следующие действия:

1) нажать левой кнопкой мыши на значок сетевых соединений в области уведомлений панели задач и в раскрывшемся меню выбрать «Соединения VPN — Configure VPN»;

2) из появившегося списка соединений выбрать нужное соединение, нажать кнопку **[Изменить]**, затем нажать **[Export]**;

3) указать файл, в который сохранить параметры соединения.

При создании соединения VPN используя ранее сохраненные параметры соединения необходимо:

1) нажать левой кнопкой мыши на значок сетевых соединений в области уведомлений панели задач и в раскрывшемся меню выбрать «Соединения VPN — Добавить VPN соединение»;

2) в открывшемся окне из выпадающего списка выбрать «Импортировать сохраненную конфигурацию VPN» и нажать **[Создать]**;

3) указать путь к файлу с параметрами соединения.

6.10.4. Диагностика работы службы и клиента

В процессе работы службы и клиента OpenVPN информация о событиях записывается в системный журнал сервера или клиента, соответственно.

Для просмотра системного журнала полностью используется команда:

```
journalctl
```

Для просмотра последних событий и вывода новых событий по мере их появления используется команда:

```
journalctl -f
```

Для вывода только новых сообщений от службы `openvpn` по мере их добавления в журнал используется команда:

```
tail -f /var/log/syslog | grep ovpn-server
```

При каждом подключении клиента в журнал сервера записывается информация о параметрах подключения, в том числе о выбранном методе защитного преобразования передаваемых данных для входящего и исходящего каналов.

Для проверки установленного метода защитного преобразования используется команда:

```
grep  "Data Channel: Cipher" /var/log/syslog
```

6.10.5. Использование инструмента XCA для создания собственного удостоверяющего центра

6.10.5.1. Установка инструмента XCA

Для безопасного и эффективного управления файлами ключей и сертификатов рекомендуется использовать графический инструмент создания и управления удостоверяющим центром XCA.

Инструмент XCA применяется для создания простейшего удостоверяющего центра (Certification Authority, CA) и инфраструктуры открытых ключей (Public Key Infrastructure, PKI).

Инструмент XCA входит в состав ОС. Установка выполняется либо из графического менеджера пакетов Synaptic, либо из терминала командой:

```
apt install xca
```

После установки инструмент XCA доступен для запуска из меню «Пуск — Утилиты — Цифровые сертификаты XCA» (при использовании классического меню «Пуск»). По умолчанию инструмент XCA запускается на языке операционной системы. Выбор языка возможно изменить вручную через меню «Файл — Язык».

После первого запуска инструмента XCA необходимо создать новую БД. Для этого:

1) выбрать в меню пункт «Файл — Новая база данных»;
2) указать название и путь размещения БД;
3) нажать **[Сохранить]**.

Перед созданием БД будет запрошена установка пароля для доступа к БД. При нажатии **[Да]** без установки пароля БД будет создана без пароля.

ВНИМАНИЕ! Утеря БД может привести к компрометации или полной неработоспособности систем, использующих выданные центром сертификаты. Рекомендуется разворачивать удостоверяющий центр на отдельном физическом компьютере, не подключенном к сети, передачу сертификатов осуществлять с помощью съемных носителей информации и принять все возможные меры для ограничения доступа к БД.

6.10.5.2. Подготовка шаблонов

Перед созданием сертификатов для упрощения дальнейшей работы рекомендуется заполнить и сохранить типовые значения полей, которые будут применяться в дальнейшем при создании сертификатов. Для этой цели в инструменте XCA предусмотрен механизм шаблонов.

Для создания нового шаблона перейти во вкладку «Шаблоны» и нажать кнопку **[Новый шаблон]**. Из появившегося списка выбрать типовой шаблон. Новый шаблон будет создан как копия выбранного предустановленного шаблона. В инструменте XCA предусмотрено три предустановленных шаблона:

- [default] CA — предустановленный шаблон сертификата удостоверяющего центра (УЦ);
- [default] HTTPS_client — предустановленный шаблон сертификата клиента;
- [default] HTTPS_server — предустановленный шаблон сертификата сервера.

Предустановленные шаблоны ориентированы на службу HTTPS, поэтому рекомендуется создать на их основе свои шаблоны, полностью настроенные на службу OpenVPN. Для всех шаблонов во вкладке «Субъект» следует заполнить следующие поля:

- «Внутреннее имя» — любое имя;
- «countryName» — двухбуквенный код страны;
- «stateOrProvinceName» — двухбуквенный код региона;
- «localityName» — название города;
- «organizationName» — название организации;
- «organizationalUnitName» — название структурной единицы внутри организации;
- «commonName» — общедоступное имя;
- «emailAddress» — адрес электронной почты.

При заполнении информационных полей шаблона не рекомендуется использовать кириллицу. Все поля являются необязательными, однако, в шаблоне, как минимум, обязательно должно быть заполнено либо поле «Внутреннее имя», либо поле «commonName».

Дополнительно необходимо внести следующие изменения в предустановленные шаблоны:

1) для шаблона сертификата УЦ — во вкладке «Расширения» проверить корректность данных:

а) тип сертификата «Центр сертификации»;

б) наличие флага «Critical»;

в) наличие флага «Subject Key Identifier». При необходимости уточнить даты и срок действия сертификата.

После корректировки шаблона сохранить его, нажав кнопку **[Да]**;

2) для шаблона сертификата сервера — во вкладке «Расширения» проверить корректность данных:

а) тип сертификата «Конечный субъект»;

б) наличие флага «Critical»;

в) наличие флага «Subject Key Identifier». При необходимости уточнить даты и срок действия сертификата.

Во вкладке «Область применения ключа»:

а) в левом поле «X509v3 Key Usage» должны быть выбраны пункты «Digital Signature» и «Key Encipherment»;

б) в левом поле «X509v3 Key Usage» снять выбор с пункта «Non Repudiation»;

в) в правом поле «X509v3 Extended Key Usage» должен быть выбран пункт «TLS Web Server Authentication».

Во вкладке «Netscape» в поле «Netscape Cert Type» снять выбор с пункта «SSL Server».

После корректировки шаблона сохранить его, нажав кнопку **[Да]**;

3) для шаблона сертификата клиента — во вкладке «Расширения» проверить корректность данных:

а) тип сертификата «Конечный субъект»;

б) наличие флага «Critical»;

в) наличие флага «Subject Key Identifier». При необходимости уточнить даты и срок действия сертификата.

Во вкладке «Область применения ключа»:

а) в левом поле «X509v3 Key Usage» снять выбор с пунктов «Data Encipherment» и «Key Encipherment»;

б) в левом поле «X509v3 Key Usage» должен быть выбран пункт «Key Agreement»;

в) в правом поле «X509v3 Extended Key Usage» должен быть выбран пункт «TLS Web Client Authentication».

Во вкладке «Netscape» в поле «Netscape Cert Type» снять выбор с пунктов «SSL Client» и «S/MIME».

После корректировки шаблона сохранить его, нажав кнопку **[Да]**.

6.10.5.3. Типовая схема применения инструмента XCA

Типовая упрощенная схема применения инструмента XCA включает в себя следующие действия:

1) создание корневого сертификата УЦ;
2) создание закрытого ключа и сертификата открытого ключа сервера;
3) экспорт для использования сервером:
   а) сертификата УЦ в соответствии с 6.10.5.7;
   б) закрытого ключа сервера в соответствии с 6.10.5.8;
   в) сертификата открытого ключа сервера в соответствии с 6.10.5.8;
   г) файла параметров Диффи-Хеллмана в соответствии с 6.10.5.8;
   д) файла параметров дополнительной аутентификации протокола TLS в соответствии с 6.10.5.8;
4) создание закрытого ключа и сертификата открытого ключа клиента;
5) экспорт для использования клиентом:
   а) сертификата УЦ в соответствии с 6.10.5.7;
   б) закрытого ключа клиента в соответствии с 6.10.5.9;
   в) сертификата открытого ключа клиента в соответствии с 6.10.5.9;
   г) файла параметров дополнительной аутентификации протокола TLS в соответствии с 6.10.5.9;
6) повторная генерация сертификатов по мере истечения их срока действия.

Пункты 4) и 5) перечисления выполняются для каждого нового подключаемого клиента. Пункт 6) повторяется для удостоверяющего центра, сервера и клиентов по мере истечения срока действия их сертификатов.

Процедура экспорта подразумевает копирование необходимых данных в файлы и перенос соответствующих файлов на компьютеры сервера и клиентов с использованием процедур, предотвращающих несанкционированный доступ к передаваемой информации (сменные носители, защищенные каналы связи и др.).

6.10.5.4. Создание корневого сертификата удостоверяющего центра

Корневой сертификат может быть получен из внешнего УЦ или создан как самозаверенный собственный корневой сертификат.

Для создания самоподписанного корневого сертификата необходимо запустить инструмент XCA и выполнить следующие действия:

1) перейти во вкладку «Сертификаты», нажать **[Новый сертификат]**;

2) в открывшемся окне будет установлен флаг «Создать самозаверенный сертификат» и в поле «Шаблон для нового сертификата» выбрать предустановленный шаблон «[default] CA». Если был создан собственный шаблон, то выбрать его, затем нажать кнопку **[Применить всё]**;

3) перейти во вкладку «Субъект». Если был применен собственный шаблон, то поля формы будут автоматически заполнены данными из выбранного шаблона (кроме поля «Внутреннее имя», которое должно быть указано индивидуально для каждого сертификата). Заполнить следующие незаполненные поля:

   а) «Внутреннее имя» — указать имя сертификата, например, «rootCA»;

   б) «commonName» — указать то же имя — «rootCA»;

   в) нажать кнопку **[Сгенерировать новый ключ]**.

   Будет предложено создать новый закрытый ключ с заданным именем. Проверить параметры ключа: «Тип Ключа: RSA», «Длинна ключа: 2048 bit». Нажать кнопку **[Создать]**, затем нажать **[Да]**;

4) перейти во вкладку «Расширения»:

   а) убедиться, что в поле «Тип» выбран «Центр Сертификации»;

   б) проверить установку флагов «Critical» и «Subject Key Identifier»;

   в) определить период действия сертификата, заполнив поля «Период действия» и «Срок действия»;

5) перейти во вкладку «Область применения ключа», убедиться, что в левом поле «X509v3 Key Usage» выбраны пункты:

   а) «Certificate Sign»;

   б) «CRL Sign»;

6) перейти во вкладку «Netscape», убедиться, что в поле «Netscape Cert Type» выбраны пункты:

   а) «SSL CA»;

   б) «S/MIME CA»;

   в) «Object signing CA»;

7) после проверок нажать **[Да]** для создания сертификата.

После выполнения данных действий в списке сертификатов появится корневой сертификат, который в дальнейшем будет использовать для подписания других сертификатов.

6.10.5.5. Создание сертификата сервера

Для создания сертификата сервера выполнить следующие действия:

1) перейти во вкладку «Сертификаты», нажать **[Новый сертификат]**;
2) в открывшемся окне во вкладке «Первоисточник»:
   а) установить флаг «Использовать этот сертификат для подписи» (флаг «Создать самозаверенный сертификат» будет снят автоматически) и в соответствующем выпадающем списке выбрать созданный согласно 6.10.5.4 корневой сертификат;
   б) в поле «Шаблон для нового сертификата» выбрать предустановленный шаблон «[default] HTTPS_server». Если был создан собственный шаблон, то выбрать его, затем нажать кнопку **[Применить всё]**;
3) перейти во вкладку «Субъект». Если был применен собственный шаблон, то поля формы будут автоматически заполнены данными из выбранного шаблона (кроме поля «Внутреннее имя», которое должно быть указано индивидуально для каждого сертификата). Заполнить следующие незаполненные поля:
   а) «Внутреннее имя» — указать имя сертификата;
   б) «commonName» — указать то же имя;
   в) нажать кнопку **[Сгенерировать новый ключ]**.
   Будет предложено создать новый закрытый ключ с заданным именем. Проверить параметры ключа: «Тип Ключа: RSA», «Длинна ключа: 2048 bit». Нажать кнопку **[Создать]**, затем нажать **[Да]**;
4) перейти во вкладку «Расширения»:
   а) убедиться, что в поле «Тип» выбран «Конечный субъект»;
   б) проверить установку флагов «Critical» и «Subject Key Identifier»;
   в) определить период действия сертификата, заполнив поля «Период действия» и «Срок действия»;
5) перейти во вкладку «Область применения ключа», убедиться, что:
   а) в левом поле «X509v3 Key Usage» выбраны пункты «Digital Signature» и «Key Encipherment»;
   б) в правом поле «X509v3 Extended Key Usage» выбран пункт «TLS Web Server Authentication»;
6) нажать **[Да]** для создания сертификата.

После создания сертификата сервера он отобразится в общем списке сертификатов. Инструмент XCA представляет список сертификатов в виде дерева, корнем которого является корневой сертификат удостоверяющего центра.

6.10.5.6. Создание сертификата клиента

Для создания сертификата клиента выполнить следующие действия:

1) перейти во вкладку «Сертификаты», нажать **[Новый сертификат]**;
2) в открывшемся окне во вкладке «Первоисточник»:
   а) установить флаг «Использовать этот сертификат для подписи» (флаг «Создать самозаверенный сертификат» будет снят автоматически) и в соответствующем выпадающем списке выбрать созданный согласно 6.10.5.4 корневой сертификат;
   б) в поле «Шаблон для нового сертификата» выбрать предустановленный шаблон «[default] HTTPS_client». Если был создан собственный шаблон, то выбрать его, затем нажать кнопку **[Применить всё]**;
3) перейти во вкладку «Субъект». Если был применен собственный шаблон, то поля формы будут автоматически заполнены данными из выбранного шаблона (кроме поля «Внутреннее имя», которое должно быть указано индивидуально для каждого сертификата). Заполнить следующие незаполненные поля:
   а) «Внутреннее имя» — указать имя сертификата;
   б) «commonName» — указать то же имя;
   в) нажать кнопку **[Сгенерировать новый ключ]**.
   Будет предложено создать новый закрытый ключ с заданным именем. Проверить параметры ключа: «Тип Ключа: RSA», «Длинна ключа: 2048 bit». Нажать кнопку **[Создать]**, затем нажать **[Да]**;
4) перейти во вкладку «Расширения»:
   а) убедиться, что в поле «Тип» выбран «Конечный субъект»;
   б) проверить установку флагов «Critical» и «Subject Key Identifier»;
   в) определить период действия сертификата, заполнив поля «Период действия» и «Срок действия»;
5) перейти во вкладку «Область применения ключа», убедиться, что:
   а) в левом поле «X509v3 Key Usage» выбран пункт «Key Agreement»;
   б) в правом поле «X509v3 Extended Key Usage» выбран пункт «TLS Web Client Authentication»;
6) нажать **[Да]** для создания сертификата.

После создания сертификата клиента он отобразится в общем списке сертификатов.

6.10.5.7. Экспорт корневого сертификата удостоверяющего центра

Для работы серверов и клиентов нужен только сертификат УЦ. Закрытый корневой сертификат УЦ не должен передаваться в другие системы, однако, его копии следует хранить в системах резервного копирования и восстановления.

Для экспорта корневого сертификата:

- в основном окне программы перейти во вкладку «Сертификаты»;
- в списке выбрать корневой сертификат и нажать кнопку **[Экспорт]**;
- в открывшейся окне указать имя файла контейнера сертификата, место сохранения и выбрать формат экспорта «PEM (`*.crt`)»;
- нажать кнопку **[Да]**.

6.10.5.8. Экспорт файлов сертификатов и ключей сервера

Для экспорта сертификата сервера необходимо:

1) в основном окне программы перейти во вкладку «Сертификаты»;
2) в списке выбрать сертификат сервера и нажать кнопку **[Экспорт]**;
3) в открывшейся окне указать место сохранения и выбрать формат экспорта «PEM (`*.crt`)»;
4) нажать кнопку **[Да]**.

Для экспорта закрытого ключа сервера необходимо:

1) в основном окне программы перейти во вкладку «Закрытые ключи»;
2) в списке выбрать закрытый ключ сервера и нажать кнопку **[Экспорт]**;
3) в открывшейся окне выбрать формат экспорта «Закрытый ключ PEM (`*.pem`)»;
4) нажать кнопку **[Да]**.

Закрытый ключ сервера экспортируется в открытом виде без применения защитного преобразования данных.

Закрытый ключ сервера должен находиться на сервере и не должен передаваться клиентам.

Для создания файла с параметрами Диффи-Хеллмана необходимо:

1) в основном окне программы выбрать в меню пункт «Extra — Создать DH параметр»;
2) в открывшейся окне указать значение «2048 (2048 бит)»;
3) нажать кнопку **[Да]**.

Примечание. Генерация занимает много времени, об активности программы свидетельствует индикатор в правом нижнем углу окна программы;

4) в открывшейся окне указать место для сохранения полученного файла;
5) нажать кнопку **[Да]** для сохранения.

Создание файл дополнительной аутентификации протокола TLS в инструменте XCA не предусмотрено. Данный файл должен быть создан отдельно средствами OpenVPN при помощи команды:

```
openvpn --genkey --secret <имя_файла>
```

6.10.5.9. Экспорт файлов сертификатов и ключей клиента

Для экспорта сертификата клиента необходимо:

1) в основном окне программы перейти во вкладку «Сертификаты»;

2) в списке выбрать сертификат клиента и нажать кнопку **[Экспорт]**;

3) в открывшейся окне указать место сохранения и выбрать формат экспорта «PEM (`*.crt`)»;

4) нажать кнопку **[Да]**.

Для экспорта закрытого ключа клиента необходимо:

1) в основном окне программы перейти во вкладку «Закрытые ключи»;

2) в списке выбрать закрытый ключ клиента и нажать кнопку **[Экспорт]**;

3) в открывшейся окне выбрать формат экспорта «Закрытый ключ PEM (`*.pem`)»;

4) нажать кнопку **[Да]**.

Закрытый ключ клиента экспортируется в открытом виде без применения защитного преобразования данных.

6.10.5.10. Отзыв сертификатов. Списки отзыва сертификатов

Для отзыва сертификата необходимо:

1) в основном окне программы перейти во вкладку «Сертификаты»;

2) найти в списке отзываемый сертификат, нажать правой кнопкой мыши и в раскрывшемся меню выбрать «Отозвать».

Аналогичным способом можно отменить отзыв сертификата, выбрав пункт «Вернуть».

Списки отозванных сертификатов привязываются к корневому сертификату УЦ, подписавшего эти сертификаты.

Для просмотра списка отозванных сертификатов, относящихся к корневому сертификату, необходимо:

1) в основном окне программы перейти во вкладку «Сертификаты»;

2) в списке выбрать корневой сертификат, нажать правой кнопкой мыши и в раскрывшемся меню выбрать «ЦС — Управление отзывами».

Откроется список отозванных сертификатов.

Для создания списка отозванных сертификатов в формате, пригодном для экспорта в другие системы, необходимо:

1) в основном окне программы перейти во вкладку «Сертификаты»;

2) в списке выбрать корневой сертификат, нажать правой кнопкой мыши и в раскрывшемся меню выбрать «ЦС — Сгенерировать CRL»;

3) в открывшемся окне, при необходимости, уточнить параметры списка;

4) нажать кнопку **[Да]**.

Созданные списки отзыва можно просмотреть во вкладке «Списки отзыва сертификатов». Из этой же вкладки списки отозванных сертификатов можно экспортировать, нажав кнопку **[Экспорт]**, формат экспорта «PEM (`*.pem`)».

6.11. Средство удаленного администрирования Ansible

Ansible является программным решением для настройки и централизованного управления конфигурациями удаленных машин, в том числе одновременно группой машин. Для работы Ansible используется существующая инфраструктура SSH.

В Ansible для применения конфигурации на удаленной машине используется режим push mode, который заключается в распространении конфигурации с управляющей машины на удаленную.

6.11.1. Состав

В состав Ansible входят модули, обеспечивающие развртывание, контроль и управление компонентами удаленных машин. Перечень основных модулей приведен в таблице 34.

Таблица 34

Модуль	Описание
shell	Позволяет запускать shell-команды на удаленном узле, например: `ansible -i step-02/hosts -m shell -a 'uname -a' host0.example.org`
copy	Позволяет копировать файл из управляющей машины на удаленный узел: `ansible -i step-02/hosts -m copy -a 'src=<исходный_каталог> dest=<каталог_назначения>' host0.example.org`
setup	Предназначен для сбора фактических данных с узлов: `ansible -i step-02/hosts -m setup host0.example.org`

6.11.2. Установка и настройка Ansible

На управляющей машине должен быть установлен Python версии 2.6 или выше. На управляемых машинах должен быть установлен Python версии 2.4 или выше.

Дополнительно для работы Ansible необходимы следующие Python-модули на управляющей машине:

- python-yaml;
- paramiko;
- python-jinja2.

Установка модулей осуществляется путем выполнения команды:

```
apt-get install python-yaml python-jinja2 python-paramiko python-crypto
```

Для установки Ansible выполнить команду:

```
apt-get install ansible
```

Перечень машин, которыми нужно управлять, задается двумя способами:

- в текстовом файле (по умолчанию используется ini-файл) в каталоге /etc/ansible/hosts;
- с помощью скрипта, получающего перечень машин из сторонних программных продуктов, например, от Zabbix.

Кроме списка управляемых машин в ini-файле может указываться дополнительная информация: номера портов для подключения по SSH, способ подключения, пароль для подключения, имя пользователя, объединения групп и т.п.

Примеры:

1. Конфигурационный ini-файл, в квадратных скобках указаны имена групп управляемых машин

```
[dbservers]
nude1.example.ru
nude2.example.ru

[webservers]
srv1.example.ru ansible_ssh_port=8877 ansible_ssh_host=192.168.1.1
srv2.example.ru
srv[3:20].example.ru
```

2. Конфигурационный YAML-файл

```
all:
  hosts:
    mail.example.ru:
  children:
    webservers:
      hosts:
        srv1.example.ru:
        jumper:
              ansible_port: 8877
              ansible_host: 192.168.1.1
        srv2.example.ru:
    dbservers:
      hosts:
        nude1.example.ru:
        nude2.example.ru:
```

В дополнение к конфигурационному файлу при определении и управлении группами удаленных машин используется переменные параметры. Переменные параметры могут

быть объединены в группы. Данные о переменных предпочтительно хранить в отдельных YAML-файлах в соответствующих каталогах:

- `/etc/ansible/group_vars/<имя_группы>` — для переменных группы машин ;
- `/etc/ansible/host_vars/<имя_машины>` — для переменных отдельных машин.

6.11.3. Сценарии Ansible

Ansible позволяет использовать сценарии, предназначенные для выполнения на управляемых машинах. Сценарии пишутся на языке YAML.

Для выполнения сценария используется команда `ansible-playbook` со следующим синтаксисом:

```
ansible-playbook <имя_файла_сценария.yml> ... [другие параметры]
```

Описание основных параметров сценариев приведено в таблице 35.

Таблица 35

Параметр	Описание
`hosts`	Указываются управляемые узлы или группы узлов, к которым нужно применить изменения
`tasks`	Описывается состояние, в которое необходимо привести управляемый узел, альтернативой могут быть роли
`gather_facts`	Указывает собирать или нет информацию об узлах перед выполнением задач. Значение по умолчанию — «Да»
`vars`	Указываются переменные, которые будут использованы при выполнении сценария
`connection`	Используется для указания метода соединения с узлами: pure ssh, paramiko, fireball, chroot, jail, local, accelerate
`sudo`	После установления соединения выполнять задачу с привилегиями другого пользователя. Значение по умолчанию — `root`
`sudo_user`	В сочетании с параметром `sudo` можно указать пользователя, с привилегиями которого будет выполнена задача
`vars_prompt`	Перед выполнением сценария Ansible в интерактивном режиме может уточнить указанные в этом разделе параметры
`remote_user` (`user`)	Имя пользователя для авторизации на удалнном узле

7. СРЕДСТВА ОБЕСПЕЧЕНИЯ ОТКАЗОУСТОЙЧИВОСТИ И ВЫСОКОЙ ДОСТУПНОСТИ

7.1. Pacemaker и Corosync

В состав ОС входит набор программного обеспечения Pacemaker и Corosync, используемого для построения кластерных систем высокой доступности. Основные особенности Pacemaker и Corosync:

- обнаружение и восстановление после сбоев узлов и служб;
- независимость от подсистемы хранения — не требуется общее хранилище;
- независимость от типов ресурсов — все что может быть заскриптовано, может быть кластеризовано;
- поддержка кластеров любого размера;
- поддержка кворумных и ресурсозависимых кластеров;
- поддержка избыточной конфигурации;
- автоматическая репликация конфигурации, может быть обновлена с любого узла кластера;
- возможность задания порядка запуска ресурсов независимо от того, на каком узле они находятся;
- поддержка ресурсов, запускаемых на множестве узлов, — клонов;
- поддержка ресурсов с мульти-режимами работы (master/slave, primary/secondary).

С точки зрения кластера все используемые сущности — службы, точки монтирования, тома и разделы — это ресурсы, поэтому в данном руководстве под словом «ресурс» понимается все, что находится под управлением кластера.

7.1.1. Установка

Для установки Pacemaker и Corosync необходимо выполнить следующее:

1) на каждом сервере отказоустойчивого кластера установить пакет:

```
sudo apt-get install pacemaker pcs
```

2) на каждом сервере разрешить автозапуск Corosync. Для этого в конфигурационном файле `/etc/default/corosync` указать параметр:

```
START=yes
```

3) на каждом сервере следует произвести запуск необходимых служб `hacluster`:

```
sudo systemctl start corosync
sudo systemctl start pacemaker
sudo systemctl restart pacemaker
```

7.1.2. Пример настройки кластера

Настройка Pacemaker и Corosync на примере двух серверов с ОС: `server-1` и `server-2`. Оба сервера должны видеть друг друга по имени, для этого должен быть настро-

ен DNS или в файле `/etc/hosts` содержаться соответствующие записи. Для настройки необходимо выполнить следующий порядок действий:

1) на каждом сервере настроить синхронизацию времени по сети (служба `ntp`);
2) на каждом сервере удалить возможно сохранившуюся предыдущую конфигурацию кластера:

```
pcs cluster destroy
```

3) на каждом сервере установить одинаковый пароль пользователю `hacluster`:

```
passwd hacluster
```

4) на первом (главном) сервере настроить авторизацию, создать и запустить кластер:

```
pcs cluster auth server-1 server-2 -u hacluster
pcs cluster setup --force --start --name CLUSTERNAME server-1 server-2
```

5) на обоих серверах перезапустить службу:

```
systemctl restart pcsd
```

6) на первом сервере разрешить автозапуск кластера:

```
pcs cluster enable --all
```

7) для текущего кластера, состоящего из двух узлов, выставить базовые настройки, выполнив команды:

```
pcs property set stonith-enabled=false
pcs property set symmetric-cluster=false
pcs property set no-quorum-policy=ignore
```

Для управления кластером Pacemaker используются утилиты `pcs` и `crm`. Например, проверка статуса кластера выполняется с помощью команды `pcs status`, просмотр текущей конфигурации — с помощью команды `crm configure show`. Результат проверки статуса кластера из примера:

```
============
Cluster name: CLUSTERNAME
Last updated: Wed Oct 25 12:11:08 2017
Last change: Wed Oct 25 12:11:06 2017
Stack: corosync
Current DC: server-1 (1) - partition with quorum
Version: 1.1.12-561c4cf
2 Nodes configured
0 Resources configured
============

Online: [ server-1 server-2 ]
```

```
Full list of resources:

PCSD Status:
server-1: Online
server-2: Online
```

Настройка кластера завершена. Управление кластером осуществляется как из консоли, так и через веб-интерфейс:

```
https://server-1:2224/
```

7.2. Keepalived

Keepalived используется в качестве управляющего ПО для организации мониторинга и обеспечения высокой доступности узлов и служб.

Демон Keepalived обеспечивает автоматический переход на резервный ресурс в режиме ожидания в случае возникновения ошибки или сбоя основного ресурса.

Для обеспечения автоматического перехода используется протокол VRRP (Virtual Redundancy Routing Protocol). Данный протокол позволяет использовать виртуальный IP-адрес VIP (virtual IP), который является плавающим (расшаренным) между узлами.

7.2.1. Установка

Пакет Keepalived необходимо установить на каждом узле, доступность которых требуется обеспечить, и на каждом резервном узле. Для установки выполнить следующую команду:

```
apt-get install keepalived -y
```

7.2.2. Пример настройки

Настройка Keepalived на примере двух серверов с ОС: `server-1` (основной) и `server-2` (резервный). На серверах должен быть настроен режим репликации для обеспечения горячего резервирования. Также на обоих серверах должно быть два сетевых интерфейса. Одному из сетевых интерфейсов основного сервера присвоить VIP.

На каждом сервере в конфигурационный файл `/etc/sysctl.conf` добавить строку:

```
net.ipv4.ip_forward = 1
net.ipv4.ip_nonlocal_bind = 1
```

и выполнить для проверки команду:

```
sysctl -p
```

На основном сервере откорректировать конфигурационный файл Keepalived `/etc/keepalived/keepalived.conf`, указав необходимые значения для основных параметров:

- `interface` — интерфейс подключения;

- state — статус сервера, для основного указывается значение MASTER;
- virtual_router_id — идентификатор виртуального маршрутизатора (должен быть одинаковым для обоих серверов);
- priority — приоритет основного сервера. Должен быть больше, чем резервного;
- auth_type — значение PASS задает парольную аутентификацию для серверов;
- auth_pass — общий пароль для всех узлов кластера;
- virtual_ipaddress — виртуальный IP-адрес.

Пример

Конфигурационный файл /etc/keepalived/keepalived.conf основного сервера

```
global_defs {
   notification_email {
       username@domain.ru
   }
   notification_email_from servers@domain.ru
       smtp_server 1.1.1.1
       smtp_connect_timeout 30
       router_id main
}

vrrp_instance server-1 {
   interface eth0
   state MASTER
   virtual_router_id 200
   priority 100
   advert_int 1
   authentication {
       auth_type PASS
       auth_pass password
        }

   virtual_ipaddress {
       10.1.9.190/32 dev eth0
        }

}
```

Для применения настроек и запуска демона Keepalived выполнить команду:

```
systemctl start keepalived
```

Далее необходимо откорректировать конфигурационный файл Keepalived `/etc/keepalived/keepalived.conf` резервного сервера, указав необходимые значения для основных параметров:

- `interface` — интерфейс подключения;
- `state` — статус сервера, для резервного указывается значение `BACKUP`;
- `virtual_router_id` — идентификатор виртуального маршрутизатора (должен быть одинаковым для обоих серверов);
- `priority` — приоритет резервного сервера. Должен быть меньше, чем основного;
- `auth_type` — значение `PASS` задает парольную аутентификацию для серверов;
- `auth_pass` — общий пароль для всех узлов кластера;
- `virtual_ipaddress` — виртуальный IP-адрес.

Keepalived

Пример

Конфигурационный файл `/etc/keepalived/keepalived.conf` резервного сервера

```
global_defs {
   notification_email {
       username@domain.ru
   }
   notification_email_from servers@domain.ru
       smtp_server 1.1.1.1
       smtp_connect_timeout 30
       router_id reserve
}

vrrp_instance server-2 {
   interface eth0
   state BACKUP
   virtual_router_id 200
   priority 50
   advert_int 1
   authentication {
       auth_type PASS
       auth_pass password
 }
```

```
    virtual_ipaddress {
        10.4.1.190/32 dev eth0
  }
}
```

Для применения настроек и запуска демона Keepalived выполнить команду:

```
systemctl start keepalived
```

7.3. Распределенная файловая система Ceph

Распределенные файловые системы используются в высокоскоростных вычислениях и фокусируются на высокой доступности, производительности и масштабируемости. ОС поддерживает распределенную файловую систему Ceph.

Ceph — распределенная объектная сеть хранения, обеспечивающая файловый и блочный интерфейсы доступа. Может использоваться на системах, состоящих как из нескольких серверов, так и из тысяч узлов. Встроенные механизмы продублированной репликации данных обеспечивают высокую отказоустойчивость системы. При добавлении или удалении новых узлов массив данных автоматически балансируется с учетом изменений. В Ceph обработка данных и метаданных разделена на различные группы узлов в кластере.

Кластер хранения данных состоит из нескольких различных демонов программного обеспечения. Каждый из этих демонов отделен от других и отвечает за определенную функцию Ceph. Схема на рис. 1 определяет функции каждого компонента Ceph.

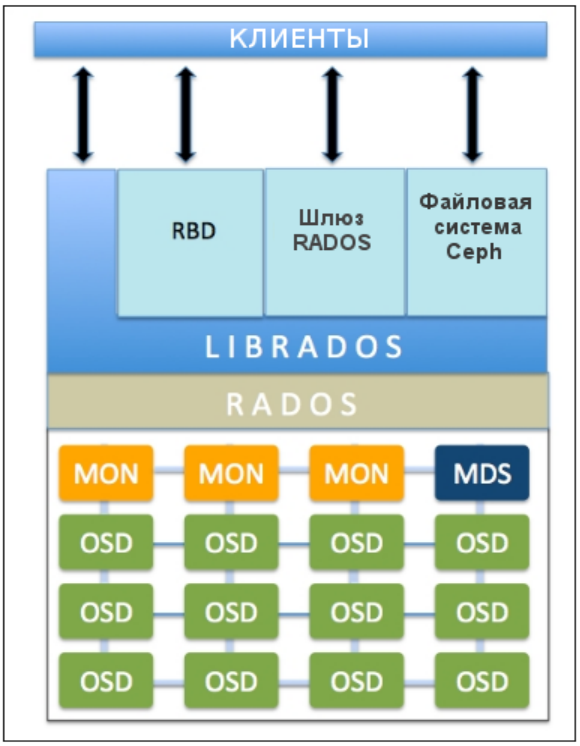

Рис. 1

Безотказное автономное распределенное хранилище объектов (RADOS) является основой хранения данных кластера Ceph. Все в Ceph хранится в виде объектов, а хранилище объектов RADOS отвечает за хранение этих объектов независимо от их типа данных. Слой RADOS гарантирует, что данные всегда остаются в согласованном состоянии и надежны. Для согласованности данных он выполняет репликацию данных, обнаружение отказов и восстановление данных, а также миграцию данных и изменение баланса в узлах кластера.

Когда приложение выполняет операцию записи на кластер Ceph, данные сохраняются в виде объектов в устройстве хранения объектов (OSD) Ceph. Это единственная составляющая кластера Ceph, в которой хранятся фактические данные пользователя, и эти же данные получает клиент, когда выполняет операцию чтения. Как правило, один OSD демон связан с одним физическим диском кластера.

Монитор (MON) Ceph отслеживает состояние всего кластера путем хранения карты состояния кластера, которая включает в себя карты OSD, MON, PG и CRUSH. Все узлы кластера сообщают узлам монитора и делают общедоступной информацию обо всех изменениях в своих состояниях. Монитор поддерживает отдельную карту информации для каждого компонента. Монитор не хранит фактические данные.

Кроме того, в кластере должна быть запущена служба мониторинга и управления (Ceph Manager — MGR), которая отвечает за отслеживание метрик времени выполнения и текущего состояния кластера Ceph, включая использование хранилища, текущие метрики производительности и нагрузку на систему. Служба MGR предоставляет интерфейс взаимодействия для внешних систем управления и мониторинга.

Библиотека librados обеспечивает доступ к RADOS с поддержкой языков программирования PHP, Ruby, Python, C и C++. Она предоставляет собственный интерфейс для кластера хранения данных Ceph, RADOS и является основанием для других служб, таких как RBD, RGW, а также интерфейса POSIX для CephFS. librados API поддерживает прямой доступ к RADOS и позволяет создать свой собственный интерфейс к хранилищу кластера Ceph.

Блочное устройство Ceph (Ceph Block Device, известное также как RADOS block device (RBD)) предоставляет блочное хранилище, которое может отображаться, форматироваться и монтироваться как любой другой диск в сервере. Блочное устройство Ceph обладает функциональностью корпоративных хранилищ, такой как: динамичное выделение, моментальные снимки.

Сервер метаданных (MDS) Ceph отслеживает метаданные файловой иерархии и сохраняет их только для CephFS. Блочное устройство Ceph и шлюз RADOS не требуют метаданных, следовательно, они не нуждаются в демоне Ceph MDS. MDS не предоставляет данные непосредственно клиентам, тем самым устраняя единую точку отказа в системе.

Файловая система Ceph (CephFS) предлагает POSIX-совместимую распределенную файловую систему любого размера. CephFS опирается на CephFS MDS, т.е. метаданные для хранения иерархии.

7.3.1. Развертывание Ceph с помощью средства ceph-deploy

Средство `ceph-deploy` обеспечивает быстрый способ развертывания Ceph без тонкой настройки, используя ssh, sudo и Python.

Далее описан возможный вариант настройки распределенного хранилища на базе Ceph на примере кластера из трех узлов `astra-ceph1`, `astra-ceph2`, `astra-ceph3` и административной рабочей станции `astra-ceph-admin`. На узлах кластера будут развернуты службы MON и OSD. Кроме того, на узле `astra-ceph1` будет запущена служба мониторинга и управления.

ВНИМАНИЕ! Данная конфигурация предназначена только для ознакомления и тестирования Ceph. При развертывании рабочей системы на объекте не рекомендуется размещать MON и OSD на одном узле.

В составе каждого из узлов кластера имеются два жестких диска: на дисках sda установлена ОС, на дисках sdb будут инициализированы OSD.

На узлах `astra-ceph1`, `astra-ceph2` и `astra-ceph3` установлен фиксированный IP-адрес. В качестве сервера DNS указан IP-адрес административной рабочей станции `astra-ceph-admin`.

На узлах `astra-ceph1`, `astra-ceph2` и `astra-ceph3` настроена служба синхронизации времени в соответствии с 6.7.1. В качестве сервера единого сетевого времени выступает административная рабочая станция `astra-ceph-admin`.

При развертывании с помощью средства `ceph-deploy` администрирование кластера осуществляется с административной рабочей станции `astra-ceph-admin`. Для удобства развертывания в файле `/etc/hosts` указаны короткие имена узлов кластера:

```
10.0.0.1 astra-ceph-admin
10.0.0.171 astra-ceph1
10.0.0.172 astra-ceph2
10.0.0.173 astra-ceph3
```

ВНИМАНИЕ! На узлах `astra-ceph1, astra-ceph2, astra-ceph3` и административной рабочей станции `astra-ceph-admin` должны быть установлены служба SSH и пакет Python версии 2.x.

Перед началом развертывания на всех узлах кластера и административной рабочей станции astra-ceph-admin необходимо выполнить следующие предварительные действия:

1) создать учетную запись администратора, например ceph-adm, выполнив команду:

```
sudo adduser ceph-adm
```

В результате появится диалог, в котором необходимо задать пароль администратора и ввести дополнительную информацию. По окончании диалога необходимо ответить «y» («Да»);

2) предоставить созданной учетной записи привилегии sudo без запроса пароля, последовательно выполнив следующие команды:

```
echo "ceph-adm ALL = (root) NOPASSWD:ALL"|sudo tee /etc/sudoers.d/ceph-adm
sudo chmod 0440 /etc/sudoers.d/ceph-adm
```

3) задать высокий уровень целостности для учетной записи ceph-adm:

```
sudo pdpl-user -i 63 ceph-adm
```

4) запустить службу SSH, выполнив команду:

```
sudo systemctl enable --now ssh
```

ВНИМАНИЕ! Во время развертывания Ceph на всех узлах кластера и административной рабочей станции astra-ceph-admin должен быть доступен диск с дистрибутивом ОС.

Развертывание Ceph с помощью средства ceph-deploy выполняется на административной рабочей станции astra-ceph-admin от имени учетной записи ceph-adm.

ВНИМАНИЕ! Недопустимо использование средства ceph-deploy от имени суперпользователя через механизм sudo.

Для развертывания Ceph необходимо выполнить следующую последовательность действий на административной рабочей станции astra-ceph-admin:

1) настроить беспарольный ssh-доступ на все узлы кластера, выполнив последовательность команд:

```
ssh-keygen
for N in $(seq 1 3); do ssh-copy-id ceph-adm@astra-ceph$N; done
```

В ходе выполнения команд для каждого узла кластера необходимо ответить «yes» («Да») и ввести пароль учетной записи ceph-adm;

2) скопировать ssh-ключ на административную рабочую станцию astra-ceph-admin для беспарольного доступа к ней:

```
ssh-copy-id ceph-adm@astra-ceph-admin
```

3) установить средство ceph-deploy:

```
sudo apt install ceph-deploy
```

4) установить Ceph на узлах кластера:

```
ceph-deploy --username ceph-adm install --mon --osd astra-ceph1
   astra-ceph2 astra-ceph3
```

Параметры `--mon` и `--osd` определяют компоненты Ceph, необходимые для установки. В противном случае будут установлены все компоненты Ceph;

5) перезагрузить узлы кластера:

```
for N in $(seq 1 3); do ssh ceph-adm@astra-ceph$N sudo reboot; done
```

6) установить на узле `astra-ceph1` службу MGR:

```
ceph-deploy --username ceph-adm install --mgr astra-ceph1
```

7) создать новый кластер Ceph, при этом указать в команде узлы кластера, на которых в дальнейшем будут инициализированы первоначальные мониторы:

```
ceph-deploy --username ceph-adm new astra-ceph1 astra-ceph2 astra-ceph3
```

После выполнения команды будут созданы конфигурационный файл (по умолчанию `ceph.conf`) и `keyring`-файл мониторов;

8) инициализировать мониторы на ранее указанных узлах кластера, выполнив команду:

```
ceph-deploy --username ceph-adm mon create-initial
```

9) инициализировать службу мониторинга и управления на узле `astra-ceph1`, используя команду:

```
ceph-deploy --username ceph-adm mgr create astra-ceph1
```

10) создать OSD на дисках sdb узлов кластера `astra-ceph1`, `astra-ceph2`, `astra-ceph3` и добить их в кластер, используя команды:

```
ceph-deploy --username ceph-adm osd create --data /dev/sdb astra-ceph1
ceph-deploy --username ceph-adm osd create --data /dev/sdb astra-ceph2
ceph-deploy --username ceph-adm osd create --data /dev/sdb astra-ceph3
```

11) установить основные компоненты Ceph на `astra-ceph-admin`, используя команду:

```
ceph-deploy --username ceph-adm install --cli astra-ceph-admin
```

12) скопировать конфигурационный файл и `keyring`-файл пользователя `admin` (ключевой файл администратора распределенной файловой системы Ceph, создается автоматически при установке) на `astra-ceph-admin`, используя команду:

```
ceph-deploy admin astra-ceph-admin
```

После завершения развертывания кластера Ceph проверить его состояние можно командой:

```
sudo ceph -s
```

В случае корректной работы кластера параметр `health` принимает значение `HEALTH_OK`.

Пример

Вывод команды `ceph -s` для приведенного варианта развертывания

```
cluster:
id:     03ff5b8a-a453-4da3-8296-2d473649bcc4
health: HEALTH_OK

services:
mon: 3 daemons, quorum astra-ceph1,astra-ceph2,astra-ceph3 (age 3h)
mgr: astra-ceph1(active, since 3h)
osd: 3 osds: 3 up (since 3h), 3 in (since 25h)

data:
pools:   0 pools, 0 pgs
objects: 0 objects, 0 bytes
usage:   3.0 GiB used, 45 GiB / 48 GiB avail
pgs:
```

В приведенном примере общий объем хранилища равен 48 ГБ (три диска по 16 ГБ), из них 3 ГБ заняты под служебные нужды Ceph. Необходимо учитывать, что доступное пространство будет расходоваться в зависимости от заданного фактора репликации (уровня избыточности данных). По умолчанию фактор репликации равен 3. Это значит, что каждый объект хранится в трех экземплярах на разных дисках. Таким образом, в наличии имеется 15 ГБ свободного места для использования в кластере ceph. Это место делится поровну между всеми пулами.

7.3.2. Использование кластера Ceph

Ceph представляет для клиента различные варианты доступа к данным:

- файловая система `cephfs`;
- блочное устройство `rbd`;
- объектное хранилище с доступом через s3 совместимое api.

Ниже представлен пример настройки работы с хранилищем в виде файловой системы `cephfs`. Основное преимущество `cephfs` в том, что возможно монтировать один и тот же каталог с данными на чтение и запись множеству клиентов. Для того, чтобы клиенты могли подключать Ceph как файловую систему, необходимо в кластере инициализировать хотя бы один сервер метаданных (MDS)

Для организации доступа к файловой системе cephfs необходимо выполнить следующую последовательность действий на административной рабочей станции astra-ceph-admin:

1) установить и активировать службу MDS на узле astra-ceph1:

```
ceph-deploy --username ceph-adm install --mds astra-ceph1
ceph-deploy --username ceph-adm mds create astra-ceph1
```

2) создать в кластере пулы для данных cephfs_data и метаданных cephfs_metadata, указав для каждого пула размер плейсмент-группы (PG) равный 64:

```
sudo ceph osd pool create cephfs_data 64
sudo ceph osd pool create cephfs_metadata 64
```

Значением размера PG должно быть число, являющееся степенью 2 (64, 128, 256...). При этом необходимо соблюсти баланс между количеством групп на OSD и их размером. Чем больше PG на одной OSD, тем больше понадобится памяти для хранения информации об их расположении. А чем больше размер самой PG, тем больше данных будет перемещаться при балансировке. Примерная формула расчета PG такая: `Total PGs = (Number OSD * 100) / max_replication_count`. Более подробная формула есть на официальном сайте - https://ceph.com/pgcalc/.

3) создать файловую систему cephfs:

```
sudo ceph fs new testcephfs cephfs_metadata cephfs_data
```

Для проверки доступа к файловой системе cephfs необходимо выполнить следующую последовательность действий на административной рабочей станции astra-ceph-admin:

1) получить ключа администратора, для этого выполнить команду:

```
cat ceph.client.admin.keyring
```

и скопировать в буфер значение параметра key. Пример вывода после выполнения команды:

```
key = AQBBVX1gl2oJJBAAh40D+1Kphz/0QA/Gbkz1sw==
caps mds = "allow *"
caps mgr = "allow *"
caps mon = "allow *"
caps osd = "allow *"
```

2) создать файл с токеном администратора (полученном на предыдущем шаге):

```
echo "AQBBVX1gl2oJJBAAh40D+1Kphz/0QA/Gbkz1sw==">admin.secret
```

3) создать локальный каталог, в который будет монтироваться файловая система cephfs:

```
sudo mkdir /mnt/cephfs
```

4) смонтировать в локальный каталог файловую систему cephfs:

```
sudo mount -t ceph 10.0.0.171:6789:/ /mnt/cephfs -o
  name=admin,secretfile=admin.secret
```

где 10.0.0.171 — адрес одного из мониторов. Их надо указывать все три, но в данном случае, пул подключается временно только для того, чтобы создать в нем каталог. Достаточно и одного монитора.

5) проверить результат монтирования, выполнив команду:

```
df -h | grep cephfs
```

Пример вывода после выполнения команды:

```
10.0.0.171:6789:/    15G            0   15G            0% /mnt/cephfs
```

Для того чтобы начать пользоваться файловой системой cephfs необходимо выполнить следующую последовательность действий:

1) на административной рабочей станции astra-ceph-admin создать в cephfs каталог data1, который будет монтироваться к другому серверу:

```
sudo mkdir /mnt/cephfs/data1
```

2) создать пользователя client.data1 для доступа к каталогу data1, для этого выполнить команду:

```
sudo ceph auth get-or-create client.data1 mon 'allow r' mds 'allow r,
   allow rw path=/data1' osd 'allow rw pool=cephfs_data'
```

и скопировать значение параметра key. Пример вывода после выполнения команды:

```
[client.data1]
  key = AQDh335g/MDeKBAAOxnXO/H4W7g2snPOpq+lCA==
```

Значение ключа доступа также можно посмотреть с помощью команды:

```
sudo ceph auth get-key client.data1
```

3) на любом другом компьютере локальной сети, поддерживающим работу с cephfs, смонтировать каталог data1, указав все 3 монитора:

```
sudo mount -t ceph 10.0.0.171,10.0.0.172,10.0.0.173:/ /mnt -o
  name=data1,secret='AQDh335g/MDeKBAAOxnXO/H4W7g2snPOpq+lCA=='
```

В приведенном примере 10.0.0.171,10.0.0.172,10.0.0.173 — IP-адреса мониторов кластера, значение параметра secret — токен пользователя, полученный на предыдущем шаге.

4) на компьютере локальной сети проверить результат монтирования, выполнив команду:

```
df -h | grep mnt
```

Пример вывода после выполнения команды:

```
10.0.0.171,10.0.0.172,10.0.0.173:/    15G     0   15G      0% /mnt
```

Таким образом, каталог `data1` на файловой системе `cephfs` подключен. При создании какого-либо файла в каталоге `data1`, этот же файл будет виден на компьютере локальной сети, к которому подключен этот же каталог.

7.4. Средство эффективного масштабирования HAProxy

Для эффективного масштабирования используется программное средство HAProxy. HAProxy обеспечивает высокую доступность, отказоустойчивость и распределение нагрузки для TCP- и HTTP-приложений посредством распределения входящих запросов на несколько обслуживающих серверов.

HAProxy предоставляет следующие возможности:

- периодическая проверка доступности обслуживающих серверов, на которые перенаправляются запросы пользователей;
- несколько алгоритмов определения доступности сервера: tcp-check, http-check, mysql-check;
- распределение HTTP/HTTPS/TCP-запросов между доступными серверами;
- возможность закрепления определенных клиентов за конкретными обслуживающими серверам (stick-tables);
- поддержка IPv6 и UNIX sockets, HTTP/1.1 сжатия (deflate, gzip, libslz), SSL, полная поддержка постоянного HTTP-соединения;
- поддержка переменных блоков и Lua-скриптов в конфигурации сервера;
- web-интерфейс с актуальным состоянием и статистикой работы программы.

7.4.1. Установка

На основном сервере, который будет принимать запросы и распределять их, необходимо установить пакет HAProxy:

```
apt install haproxy
```

7.4.2. Настройка

Настройка выполняется в конфигурационном файле `/etc/haproxy/haproxy.cfg`, включающем следующие разделы:

- `global` — определяет общую конфигурацию для всего HAProxy;
- `defaults` — является обязательным и определяет настройки по-умолчанию для остальных разделов;
- `frontend` — используется для описания набора интерфейсов для принятия соединений от клиентов, а также правил распределения нагрузки;
- `backend` — используется для описания набора серверов, к которым будет выполняться подключение переадресованных входящих соединений, а также определения алгоритма распределения нагрузки;

- listen — объединенный раздел для описания frontend и backend. Используется для описания прокси-сервера в одном разделе, как правило, только для TCP-трафика.

В таблице 36 представлены основные примеры значений параметров конфигурационного файла и их описание.

Таблица 36 – Параметры конфигурационного файла /etc/haproxy/haproxy.cfg

Раздел	Параметр	Описание
global	log <address> <facility> [max level [min level]] Например, log 127.0.0.1 local0 notice	Добавляет сервер системного журнала. <facility> — должен быть одним из 24 стандартных типов регистрации событий: kern user mail daemon auth syslog lpr news uucp cron auth2 ftp ntp audit alert cron2 local0 local1 local2 local3 local4 local5 local6 local7
	maxconn <number> Например, maxconn 10000	Устанавливает максимальное число одновременных подключений для каждого процесса haproxy
	nbproc <number> Например, nbproc 2	Задает количество процессов haproxy. По умолчанию создается только один процесс haproxy
	daemon	Устанавливает процессу haproxy режим работы «daemon»
	user	Пользователь, от имени которого работает процесс haproxy
	group	Группа, от имени которой работает процесс haproxy
	chroot /var/lib/haproxy	Устанавливает окружение процесса haproxy
defaults	log global	Включает в регистрацию событий информацию о трафике
	mode http	Режим работы HAProxy. Возможны два режима: - http — выполняется анализ Layer 7, подходит для распределения http-трафика; - tcp — распределение любого трафика
	option dontlognull	Отключает регистрацию пустых подключений
	retries 3	Количество попыток определить состояние обслуживающего сервера после сбоя подключения
	option redispatch	Распределяет запросы после сбоя подключения к одному из обслуживающих серверов
	option httpclose	Закрывает пассивные соединения
	option forwardfor	Включает X-Forwarded-For для передачи IP-адреса клиента обслуживающему серверу
frontend	frontend http	Задает имя frontend
	bind *:80	Задает IP-адрес и порт для прослушивания запросов

Продолжение таблицы 36

Раздел	Параметр	Описание
backend	backend sitecluster	Задает имя обслуживающего сервера
	balance (roundrobin/leastconn/ static-rr/uri/source)	Настройка алгоритма распределения. Поддерживаются следующие алгоритмы: - Round Robin — направляет новые подключения к следующему серверу в циклическом списке, который видоизменяется при помощи веса сервера, на основании которого идет распределение запросов. Вес сервера можно изменить «на лету». Параметр включается при помощи команды balance roundrobin; - Least Connected — направляет новые подключения к серверу с наименьшим числом соединений. Параметр включается при помощи команды balance leastconn; - Static Round Robin — направляет новые подключения к следующему серверу в циклическом списке, который видоизменяется при помощи веса сервера, на основании которого идет распределение запросов. В отличие от стандартной реализации Round Robin, в данном алгоритме нельзя изменить вес сервера «на лету». Изменение веса сервера требует перезагрузки HAProxy. Параметр включается при помощи команды balance static-rr; - Source — выбирает сервер исходя из хеша, построенного на основе IP-адреса пользователя. Таким образом, пользователь всегда обращается к одному и тому же серверу
	server srv-1.3.my.com 21.86.21.20:80 cookie site113ha check inter 2000 fall 3 minconn 30 maxconn 70 weight 100	Описание обслуживающего сервера, где: - srv-1.3.my.com — имя сервера; - 21.86.21.20:80 — IP-адрес: порт; - cookie site113ha — задание cookie, необходимого для правильного распределения сессий клиентов; - check inter 2000 fall 3 — проверка доступности сервера каждые 2 с, при наличии трех ошибок считать сервер недоступным; - minconn 30 maxconn 70 — организация очереди запросов, ограничение не более 70 одновременно обрабатывающихся запросов; - weight 100 — вес сервера, возможные значения от 1 до 100
	stats enable	Включает статистику
	fullconn 200	Задает максимальное значение одновременных подключений

Окончание таблицы 36

Раздел	Параметр	Описание
listen	`listen stats-srv-3.my.com *:8180`	Описывает IP-адрес и порт доступа к статистике
	`stats uri /stats`	URL доступа к статистике
	`stats realm Haproxy Statistics`	Заголовок (title) страницы статистики
	`stats show-legends`	Отображает в статистике дополнительную информацию о параметрах
	`stats refresh 5s`	Указывает интервал автоматического обновления страницы статистики
	`stats auth test:test`	Устанавливает логин и пароль доступа к странице статистики

Пример

Конфигурационный файл для распределения нагрузки сервера Apache

```
global
     log /dev/log local0
     log /dev/log local1 notice
     maxconn 40000
     chroot /var/lib/haproxy
     stats socket /run/haproxy/admin.sock mode 660 level admin
     stats timeout 30s
     user haproxy
     group haproxy
     daemon        # Размещение сертификатов SSL
     ca-base /etc/ssl/certs
     crt-base /etc/ssl/private    # Алгоритмы защитного преобразования,
     # применяемые для SSL-подключений
     # Подробнее см. по ссылке:
     # https://hynek.me/articles/hardening-your-web-servers-ssl-ciphers/
     ssl-default-bind-ciphers ECDH+AESGCM:DH+AESGCM:ECDH+AES256:DH+AES256:
          ECDH+AES128:DH+AES:ECDH+3DES:DH+3DES:RSA+AESGCM:RSA+AES:RSA+3DES:
          !aNULL:!MD5:!DSS
     ssl-default-bind-options no-sslv3

defaults
     log global
     mode http
```

```
        option httplog
        option dontlognull
        retries 3
        option redispatch
        maxconn 2000
        timeout connect 5000
        timeout client 50000
        timeout server 50000
        errorfile 400 /etc/haproxy/errors/400.http
        errorfile 403 /etc/haproxy/errors/403.http
        errorfile 408 /etc/haproxy/errors/408.http
        errorfile 500 /etc/haproxy/errors/500.http
        errorfile 502 /etc/haproxy/errors/502.http
        errorfile 503 /etc/haproxy/errors/503.http
        errorfile 504 /etc/haproxy/errors/504.http

frontend localnodes
        bind *:80
        mode http
        default_backend nodes

backend nodes
        mode http
        balance roundrobin
        server webserver1 192.168.13.150:80 cookie serv1 check
        server webserver2 192.168.13.151:80 cookie serv2 check
```

8. СРЕДСТВА ОРГАНИЗАЦИИ ЕПП

8.1. Архитектура ЕПП

Единое пространство пользователей представляет собой средства организации работы пользователя в сети компьютеров, работающих под управлением ОС. В основу положен доменный принцип построения сети, подразумевающий объединение в одну сеть логически связанных компьютеров, например, принадлежащих одной организации. При этом пользователь получает возможность работы с сетевыми ресурсами сети и взаимодействия с другими пользователями.

Организация ЕПП обеспечивает:

- сквозную аутентификацию в сети;
- централизацию хранения информации об окружении пользователей;
- централизацию хранения настроек системы защиты информации на сервере.

Сетевая аутентификация и централизация хранения информации об окружении пользователя основана на использовании двух основных механизмов: NSS, описание которого приведено в 8.1.1, и PAM, описание которого приведено в 8.1.2.

В качестве источника данных для базовых системных служб на базе механизмов NSS и PAM используется служба каталогов LDAP в соответствии с 8.1.3.

Сквозная доверенная аутентификация реализуется технологией Kerberos в соответствии с 8.1.4.

Централизация хранения информации об окружении пользователей подразумевает и централизованное хранение домашних каталогов пользователей. Для этого используется СЗФС CIFS в соответствии с 6.9.

При создании ЕПП в качестве основной службы рекомендуется использовать службу FreeIPA, описанную в 8.3. Для обеспечения совместимости с другими системами в состав входит служба ALD, описание которой приведено в 8.2.

8.1.1. Механизм NSS

Механизм NSS предоставляет всем программам и службам, функционирующим на локальном компьютере, системную информацию через соответствующие программные вызовы. Он обращается к конфигурационному файлу `/etc/nsswitch.conf`, в котором указаны источники данных для каждой из системных служб. Краткое описание системных служб приведено в таблице 37.

Таблица 37

Служба	Источник данных по умолчанию	Описание
passwd	/etc/passwd	Окружение пользователя (домашний каталог, идентификатор пользователя и пр.)
shadow	/etc/shadow	Пароли пользователей
group	/etc/group	Принадлежность пользователей группам
hosts	/etc/hosts	Соответствие имен хостов адресам
services	/etc/services	Характеристики сетевых служб (порт, тип транспортного протокола)

Каждая из базовых системных служб поддерживает ряд библиотечных программных вызовов, таких как getpwent, getspent, getgrent, getservent. При выполнении данных программных вызовов производится поиск в конфигурационном файле /etc/nsswitch.conf источника данных соответствующей службы (например, passwd для получения домашнего каталога пользователя). По умолчанию в качестве источника данных системных служб используются соответствующие конфигурационные файлы в каталоге /etc (источник files). NSS при получении имени источника данных из конфигурационного файла /etc/nsswitch.conf осуществляет поиск программной разделяемой библиотеки в каталоге /lib с именем libnss_<имя_источника_данных>-<версия_библиотеки>.so, где в качестве имени источника данных выступает строка, полученная из /etc/nsswitch.conf. Например, при вызове getpwent, при условии, что в /etc/nsswitch.conf находится строка:

```
passwd : files
```

будет вызвана соответствующая функция из библиотеки /lib/libnss_files.so.

8.1.2. Механизм PAM

Механизм PAM (Pluggable Authentication Modules — подключаемые модули аутентификации) позволяет интегрировать различные низкоуровневые методы аутентификации и предоставить единые механизмы для использования прикладных программ в процессе аутентификации. Механизм состоит из набора разделяемых библиотек и конфигурационных файлов — сценариев процедур аутентификации.

В каталоге /etc/pam.d расположены конфигурационные файлы PAM для соответствующих служб, в т. ч. и для login (авторизованный вход в систему). В конфигурационном файле службы дана информация по проведению аутентификации.

Модули PAM вызываются при выполнении следующих функций:

1) auth — аутентификация;
2) account — получение привилегий доступа;
3) password — управление паролями;

4) session — сопровождение сессий.

Для выполнения каждой функции может быть перечислено несколько модулей PAM, которые будут вызываться последовательно, образуя стек PAM для данной задачи. Каждый вызываемый модуль возвращает в стек результат своей работы: успешный (PAM_SUCCESS), неуспешный (PAM_AUTH_ERR), игнорирующий (PAM_IGNORE) или иной. Для каждого вызова может быть указан набор управляющих флагов в виде соответствия кода возврата и того, как результат работы модуля скажется на обработке всей служебной задачи, например, ignore, ok, die. Для управления аутентификацией используются следующие флаги:

- requisite — немедленное прекращение дальнейшего выполнения служебной задачи с общим неуспешным результатом в случае неуспешного результата выполнения данного модуля;
- required — требование удачного выполнения этого модуля одновременно с выполнением всех остальных, перечисленных в данной служебной задаче;
- sufficient — в случае позитивных результатов выполнения данного модуля и всех предыдущих с флагом required в стеке задачи немедленно прекращается дальнейшее выполнение служебной задачи в целом с общим позитивным результатом. Если же модуль вернул негативный результат, то его значение игнорируется;
- optional — выполнение данного модуля никак не сказывается на результате всей задачи, но играет дополнительную информационную роль.

8.1.3. Служба каталогов LDAP

Служба каталогов LDAP — общее название клиент-серверной технологии доступа к службе каталогов X.500 с помощью протокола LDAP. Служба каталогов X.500 является средством иерархического представления информационных ресурсов, принадлежащих некоторой отдельно взятой организации, и информации об этих ресурсах. При этом служба каталогов обеспечивает централизованное управление как самими ресурсами, так и информацией о них, а также позволяет контролировать их использование третьими лицами. Каждый ресурс может принадлежать одному или более классам. Каждый класс показывает, что ресурс является определенным типом сущности, и имеет определенный набор свойств. Совокупности классов могут объединяться в схемы, которые описывают типы ресурсов, применяемые в отдельно взятой предметной области.

Информация, хранящаяся в каталоге, называется «информационной базой каталога» (DIB). Пользователь каталога, который может быть как человеком, так и компьютером, получает доступ к каталогу посредством клиента. Клиент от имени пользователя каталога взаимодействует с одним или более серверами. Сервер хранит фрагмент DIB.

DIB содержит два типа информации:

- пользовательская — информация, предоставляемая пользователям и, быть может, изменяемая ими;
- административная и функциональная — информация, используемая для администрирования и/или функционирования каталога.

Множество записей, представленных в DIB, организовано иерархически в структуру дерева, известную как «информационное дерево каталога» (DIT). При этом запись в каталоге LDAP состоит из одного или нескольких атрибутов, обладает уникальным именем (DN — Distinguished Name) и может состоять только из тех атрибутов, которые определены в описании класса записи. В схеме определено, какие атрибуты являются для данного класса обязательными, а какие — необязательными. Каждый атрибут, хранящийся в каталоге LDAP, имеет определенный синтаксис (например, тип данных), который накладывает ограничения на структуру и формат его значений. Сравнение значений не является частью определения синтаксиса, а задается отдельно определяемыми правилами соответствия. Правила соответствия специфицируют аргумент, значение утверждения, которое также имеет определенный синтаксис.

Предполагается, что информация каталога достаточно статична, т.е. чаще читается, чем модифицируется. Примером подобного каталога является специализированная БД, например, телефонная книга, база данных службы DNS.

Службы каталогов LDAP могут быть использованы в качестве источника данных для базовых системных служб на базе механизмов NSS и PAM.

В результате вся служебная информация пользователей сети может располагаться на выделенном сервере в распределенной гетерогенной сетевой среде. Добавление новых сетевых пользователей в этом случае производится централизованно на сервере службы каталогов.

Благодаря предоставлению информации LDAP в иерархической древовидной форме разграничение доступа в рамках службы каталогов LDAP может быть основано на введении доменов. В качестве домена в данном случае будет выступать поддерево службы каталогов LDAP.

8.1.4. Доверенная аутентификация Kerberos

Kerberos является протоколом, обеспечивающим централизованную аутентификацию пользователей и применяющим техническое маскирование данных для противодействия различным видам атак.

Основным компонентом системы Kerberos является центр распределения ключей (KDC). Программы, настроенные на взаимодействие с Kerberos, называются «керберизованными приложениями». KDC отвечает за аутентификацию в некоторой области Kerberos.

В процессе работы система Kerberos выдает билеты (tickets) на использование различных служб.

Сервером Kerberos называется компьютер, на котором выполняется серверная программа Kerberos, или сама программа KDC. Клиент Kerberos — это компьютер или программа, которые получают билет от сервера Kerberos. Обычно действия системы Kerberos инициирует пользователь, отправляющий запрос на получение услуг от некоторого сервера приложения (например, сервера почты). Kerberos предоставляет билеты принципалам, в роли которых выступают пользователи или серверные программы. Для описания принципала применяется идентификатор, состоящий из трех компонентов: основы (`primary`), экземпляра (`instance`) и области (`realm`). Данный идентификатор имеет вид:

`основа/экземпляр@область`

Система Kerberos выполняет следующие задачи:

1) обеспечение аутентификации в сети. Для предотвращения НСД к службам сервер должен иметь возможность идентифицировать пользователей. Кроме того, в некоторых средах важно, чтобы клиент мог идентифицировать серверы. Это исключит работу пользователей с фальшивыми серверами, созданными для незаконного сбора конфиденциальной информации;

2) защиту паролей. Открытость паролей, используемых в ряде сетевых служб, создает угрозу безопасности системы, т. к. они могут быть перехвачены и использованы для незаконного доступа к системе. Для решения данной проблемы используется техническое маскирование билетов Kerberos.

Технология Kerberos представляет собой механизм аутентификации пользователей и служб, основным достоинством которой является повышенная защищенность при использовании в сети, которая достигается механизмом защищенного обмена билетами между пользователями, службами и сервером учетных записей Kerberos. При данном механизме пароли пользователей по сети не передаются, что обеспечивает повышенную защищенность от сетевых атак. С помощью механизма открытых и закрытых ключей, а также синхронизации часов клиентских компьютеров с сервером Kerberos обеспечивается уникальность билетов и их защищенность от подделки.

В ОС используется реализация MIT Kerberos;

3) обеспечение однократной регистрации в сети. Система Kerberos дает возможность пользователю работать с сетевыми службами, пройдя лишь единожды аутентификацию на своем компьютере. При этом для обмена с приложениями дополнительно вводить пароль не требуется.

Локальные системы учетных записей пользователей и система ЕПП существуют в ОС параллельно. Различие между ними проводится с помощью разграничения

диапазонов UID (значения UID меньшие, чем 2500, относятся к локальным пользователям, а большие или равные 2500 — к пользователям ЕПП).

ВНИМАНИЕ! Обязательным требованием для функционирования аутентификации по Kerberos является синхронизация времени на клиенте и сервере. Синхронизация может быть обеспечена использованием сервера NTP (см. 6.7).

8.1.5. Централизация хранения атрибутов СЗИ в распределенной сетевой среде

В среде ОС пользователю поставлен в соответствие ряд атрибутов, связанных с механизмами СЗИ ОС, например:

- привилегии администрирования, вхождение в группы;
- разрешенные параметры входа (список разрешенных компьютеров домена);
- политики паролей и учетных записей;
- мандатные атрибуты (диапазон доступных уровней и категорий конфиденциальности, разрешенные уровни целостности, привилегии);
- параметры регистрации событий (маски регистрируемых успешных и неуспешных событий).

Часть из атрибутов характерна только для ЕПП, другая — является отражением общих атрибутов СЗИ ОС. Доступ к мандатным атрибутам пользователей осуществляется с использованием программной библиотеки `parsec`. Данная библиотека получает из соответствующего конфигурационного файла информацию об источнике данных для мандатных СЗИ системы. По умолчанию используются локальные текстовые файлы. Концепция ЕПП подразумевает хранение системной информации о пользователе (в т. ч. и его мандатные атрибуты) централизованно. В этом случае вся информация хранится в службе каталогов LDAP.

8.2. Служба Astra Linux Directory

Служба ALD представляет собой систему управления ЕПП.

Она является надстройкой над технологиями LDAP, Kerberos 5, CIFS и обеспечивает автоматическую настройку всех необходимых файлов конфигурации служб, реализующих перечисленные технологии, а так же предоставляет интерфейс управления и администрирования.

Настройка окружения пользователя при входе в систему обеспечивается PAM-модулем ALD, который выполняет следующие функции:

- получение параметров окружения пользователя с сервера домена;
- проверка возможности входа пользователя на данный компьютер по списку разрешенных пользователю компьютеров;

- проверка возможности использования пользователем типа ФС его домашнего каталога;
- настройка параметров окружения пользователя;
- монтирование домашнего каталога пользователя;
- включение доменного пользователя в заданные локальные группы.

Перечисленные параметры и ограничения входа пользователя задаются с помощью соответствующих команд утилиты администрирования `ald-admin` и параметрами конфигурационного файла `/etc/ald/ald.conf` в соответствии с 8.2.3.

В состав ОС входит графическая утилита `fly-admin-smc`, которая позволяет администратору произвести управление ЕПП в графическом режиме (см. электронную справку).

8.2.1. Состав

Все необходимые компоненты службы ALD входят в состав пакетов, приведенных в таблице 38.

Таблица 38

Наименование	Описание
`ald-client`	Клиентская часть ALD. Содержит утилиту конфигурирования клиентского компьютера `ald-client`, PAM-модуль ALD, службу обработки заданий ALD `aldd` и утилиту автоматического обновления пользовательских билетов `ald-renew-tickets`. Пакет должен устанавливаться на все клиентские компьютеры, входящие в домен
`ald-admin`	Пакет администрирования ALD. Содержит утилиту администрирования ALD `ald-admin`. Пакет должен устанавливаться на компьютеры, с которых будет осуществляться администрирование ALD. При установке данного пакета также устанавливается клиентская часть
`ald-client-fs`	Расширение для организации файл-сервера ALD. Содержит необходимые подгружаемые модули для конфигурирования файл-сервера ALD и расширение команд `ald-client` и `ald-client-fs`. Пакет может устанавливаться на клиентские компьютеры, выступающие в роли файл-сервера
`ald-server-dc`	Серверная часть ALD. Содержит утилиту конфигурирования сервера `ald-init`. Пакет должен устанавливаться на сервер домена. При установке данного пакета также устанавливается средство администрирования `ald-admin` и клиентская часть
`ald-server`	Метапакет для установки полного сервера ALD. Пакет должен устанавливаться на сервер домена. При установке данного пакета устанавливается пакет сервера домена ALD `ald-server-dc`

Служба ALD обладает расширяемой архитектурой, состоящей из ядра, отвечающего за основной функционал системы, ряда интерфейсов (LDAP, Kerberos, Config, RPC) и модулей расширения, предназначенных для расширения командного интерфейса утилит и настройки необходимых служб и подсистем, что позволяет расширять функциональность

ALD, устанавливая дополнительные пакеты. Наименование пакета расширения отражает его назначение:

- `ald-client-...` — расширение, необходимое клиентской части ALD;
- `ald-admin-...` — расширение утилиты администрирования ALD;
- `ald-server-...` — расширение, необходимое для организации хранения атрибутов на сервере ALD.

Реализованы следующие расширения для поддержки централизации хранения атрибутов СЗИ в распределенной сетевой среде:

- `ald-client-parsec` — конфигурирование подсистемы хранения атрибутов СЗИ;
- `ald-admin-parsec` — расширение команд утилиты администрирования `ald-admin`;
- `ald-server-parsec` — расширение функциональности сервера ALD для хранения атрибутов СЗИ (необходимые схемы и правила LDAP).

ВНИМАНИЕ! Без установки пакетов расширения совместно с соответствующими основными пакетами невозможна централизация хранения атрибутов СЗИ в распределенной сетевой среде, что может привести к невозможности входа пользователей в систему.

Для снижения нагрузки на сервер ALD и повышения производительности служба обработки заданий ALD `aldd` выполняет кэширование редко изменяемых данных ALD в локальном кэше. Расширения ALD могут обрабатывать события службы кэширования для выполнения необходимых операций для обновления локального кэша.

ВНИМАНИЕ! Измененная на сервере информация может попасть в локальный кэш с задержкой.

Описание пакетов и возможностей указанных утилит приведено в руководстве `man`, список статей руководства `man` приведен в таблице 39.

Таблица 39

Наименование	Описание
`ald 7`	ALD
`ald-client 8`	Клиентская часть ALD и команды утилиты управления клиентом ALD `ald-client`
`ald-admin 1`	Команды утилиты администрирования ALD `ald-admin`
`ald-init 8`	Команды утилиты управления сервером домена `ald-init`
`aldd 8`	Служба обработки заданий ALD `aldd`
`pam_ald 8`	PAM-модуль ALD
`ald-renew-tickets 1`	Утилита автоматического обновления пользовательских билетов `ald-renew-tickets`
`ald.conf 5`	Формат конфигурационного файла `ald.conf`

Окончание таблицы 39

Наименование	Описание
`ald-client-fs 8`	Расширение для организации файл-сервера ALD
`ald-parsec-cfg 7`	Расширение конфигурирования подсистемы хранения атрибутов СЗИ
`ald-parsec-aud 7` `ald-admin-parsec-aud 1`	Расширение централизации настроек расширенного аудита
`ald-parsec-devac 7` `ald-admin-parsec-devac 1`	Расширение для подсистемы контроля доступа к подключаемым носителям

`ald-parsec-mac 7`
`ald-admin-parsec-mac 1`
`pam_ald_mac 8`
	Расширение централизации хранения атрибутов СЗИ

8.2.2. Установка

Установка службы ALD может осуществляться как при начальной установке ОС путем выбора соответствующих пунктов в программе установки, так и в ручном режиме уже в работающей системе.

ВНИМАНИЕ! В случае установки сервера ALD в ручном режиме возможно получения следующей ошибки установки:

```
insserv: Service nfs-common has to be enabled to start service nfs-kernel-server
insserv: exiting now!
update-rc.d: error: insserv rejected the script header
```

Данная ошибка вызвана тем, что в соответствии с политикой ОС по минимизации сетевых уязвимостей, большинство сетевых служб по умолчанию выключены. Для успешной установки сервера ALD необходимо вручную включить необходимую службу:

```
systemctl enable nfs-common
```

ВНИМАНИЕ! Для создания ЕПП ALD, в которое должны быть интегрированы клиенты, поддерживающие режимы мандатного управления доступом и/или мандатного контроля целостности, необходимо использовать сервер ALD с включенными соответствующими режимами. После установки сервера ALD изменение его режимов работы мандатного управления доступом и мандатного контроля целостности не поддерживается.

Для настройки автоматического запуска служб также можно использовать графическую утилиту `systemdgenie`.

ВНИМАНИЕ! Без установки пакетов расширения совместно с соответствующими основными пакетами невозможна централизация хранения атрибутов СЗИ в распределенной сетевой среде, что может привести к невозможности входа пользователей в систему.

Для облегчения установки службы ALD на конкретный компьютер предназначены метапакеты, обеспечивающие установку всех необходимых пакетов в зависимости от назначения данного компьютера:

- `ald-client-common` — установка клиентской части ALD;
- `ald-admin-common` — установка утилиты администрирования БД ALD;
- `ald-server-common` — установка сервера домена ALD.

При раздельной установке расширений ALD на сервере необходимо после установки выполнить операции инициализации расширений командой:

```
ald-init install-ext
```

которая произведет необходимые настройки и изменения существующей БД ALD. При инициализации БД ALD при установленных пакетах расширения данные действия осуществляются автоматически.

8.2.3. Настройка

Настройка всех компонентов ALD осуществляется автоматически утилитами конфигурирования. Для нормального функционирования ALD необходимо выполнение следующих условий:

1) разрешение имен должно быть настроено таким образом, чтобы имя системы разрешалось, в первую очередь, как полное имя (например, `myserver.example.ru`). Утилита `hostname` должна возвращать короткое имя компьютера, например, `myserver`.

Пример

Файл `/etc/hosts` (разрешение имен может быть настроено и с помощью сервера DNS согласно 6.5)

```
127.0.0.1       localhost
192.168.1.1     myserver.example.ru myserver
```

2) должна быть выполнена синхронизация времени в ОС серверов и клиентов ALD для аутентификации по Kerberos. Например, с использованием сервера NTP согласно 6.7.

Настройки сервера и клиентов ALD содержатся в файле `/etc/ald/ald.conf`. Формат файла:

```
ИМЯ_ПАРАМЕТРА=значение # Комментарий
```

Описание параметров конфигурационного файла `/etc/ald/ald.conf` приведено в таблице 40.

Таблица 40

Параметр	Описание
VERSION	Для ОС должно быть установлено значение «1.7». Значение «1.6», «1.5» или «1.4» может быть установлено для совместимости
DOMAIN	Имя домена. Должно быть задано в формате .example.ru для сервера ALD. Если данный параметр меняется, то необходимо заново инициализировать сервер командой: ald-init init Можно также воспользоваться командами резервного копирования и восстановления для переименования домена.
SERVER	Полное имя серверного компьютера ALD. Например: my-server.example.ru
CLIENT_SMB_VERSION	Версия протокола SMB клиента, которую поддерживает сервер. Возможные значения 1.0, 2.1, 3.11
MINIMUM_UID	Минимальный номер глобального пользователя. Пользователи с номером меньше данного считаются локальными и аутентифицируются через локальные файлы /etc/passwd и /etc/shadow. Примечание. Для нормальной работы домена не рекомендуется пересечение по номерам локальных и глобальных пользователей и групп. Не рекомендуется задавать MINIMUM_UID меньше 1000
DEFAULT_LOGIN_SHELL	Командная оболочка, которая устанавливается при создании нового пользователя. Применяется при администрировании с данного компьютера. По умолчанию используется /bin/bash
DEFAULT_LOCAL_GROUPS	Перечень локальных групп, членство в которых устанавливается при создании нового пользователя. Применяется при администрировании с данного компьютера. При входе в домен пользователю будет добавляться членство в указанных группах для использования тех или иных возможностей компьютера, например, воспроизведение звука
ALLOWED_LOCAL_GROUPS	Перечень разрешенных локальных групп, членство в которых устанавливается при входе пользователя. Применяется для данного компьютера. При входе в домен пользователю будет добавляться членство в установленных для него локальных группах в пределах разрешенных на данном компьютере

Продолжение таблицы 40

Параметр	Описание
TICKET_MAX_LIFE=10h	Максимальное время жизни билета Kerberos (если его не обновлять). Формат параметра: NNd (дни), или NNh (часы), или NNm (минуты). При входе в домен пользователь получает билет. При выходе из домена билет уничтожается. Если билет не обновлять, то после истечения срока действия билета пользователь потеряет доступ к своему домашнему каталогу. Чтобы восстановить доступ, ему придется выполнить команду kinit или зайти в систему заново. Чтобы доступ не был потерян, билет следует периодически обновлять (до истечения срока действия). Настроить автоматическое обновление можно с помощью утилиты ald-renew-ticket. Для удобства можно настроить данный параметр на большое количество времени, например, 30d. Но это менее безопасно
TICKET_MAX_RENEWABLE_LIFE=7d	Максимальное обновляемое время жизни билета Kerberos. Формат параметра: NNd (дни), или NNh (часы), или NNm (минуты). По истечении данного срока билет не может быть обновлен. Данный параметр должен быть больше, чем параметр TICKET_MAX_LIFE. Примечание. Для клиентских компьютеров параметры TICKET_MAX_LIFE и TICKET_MAX_RENEWABLE_LIFE определяются как наименьшие значения этих параметров, заданных в файлах ald.conf на сервере и на клиентском компьютере
NETWORK_FS_TYPE	Определяет, какая сетевая ФС будет использоваться для глобальных пользовательских домашних каталогов. Возможные значения: – none — сетевая ФС не используется. Работает только аутентификация глобальных пользователей. Используются локальные домашние каталоги пользователей (следующие параметры, относящиеся к сетевой ФС, игнорируются); – cifs — используется Samba/CIFS
SERVER_EXPORT_DIR	Только для сервера. Задает абсолютный путь к каталогу на сервере, где будет располагаться хранилище домашних каталогов. Данный каталог будет экспортирован по Samba/CIFS
CLIENT_MOUNT_DIR	Задает абсолютный путь к точке монтирования хранилища домашних каталогов на клиентских компьютерах
SERVER_FS_KRB_MODES	Только для сервера. Задает режимы экспорта сервера Samba/CIFS (перечисленные через запятую). Возможные режимы: - krb5 — только Kerberos-аутентификация; - krb5i — (integrity) аутентификация и проверка целостности (подпись) пакетов. Должен быть указан хотя бы один режим

Окончание таблицы 40

Параметр	Описание
CLIENT_FS_KRB_MODE	Задает Kerberos-режим монтирования на клиентском компьютере. Должен быть указан один из режимов: krb5 или krb5i
SERVER_POLLING_PERIOD	Только для сервера. Задает период (в секундах) опроса заданий службой aldd. По умолчанию составляет 60 с
SERVER_PROPAGATE_PERIOD	Только для сервера. Задает период (в секундах) репликации БД ALD на резервные сервера. По умолчанию составляет 600 с
UTF8_GECOS	Только для сервера. Задает признак модификации схемы LDAP для возможности использования кириллицы в поле описания GECOS пользователя. По умолчанию установлен равным 1
USE_RPC	Разрешает администрирование с помощью RPC интерфейса. По умолчанию установлен равным 1
RPC_PORT	Порт RPC интерфейса. По умолчанию установлен равным 17302
RPC_RESTRICTED	Список запрещенных к исполнению RPC команд
SERVER_ON	Отображает состояние сервера ALD (устаревшее). Возможные значения 0 и 1. Если SERVER_ON=0, то: - домашние каталоги не экспортируются; - разрешение имен по LDAP выключается в nsswitch.conf; - все принципалы Kerberos деактивируются (allow_tickets=0); - службы LDAP, Samba, Kerberos, останавливаются; - служба sssd перезапускается. В настоящее время состояние ALD может быть получено командой status утилит ald-client, ald-init и ald-admin
CLIENT_ON	Отображает состояние клиентской части ALD (устаревшее). Возможные значения 0 и 1. Если CLIENT_ON=0, то: - домашние каталоги не монтируются; - разрешение имен по LDAP выключается в nsswitch.conf; - служба sssd перезапускается. В настоящее время состояние ALD может быть получено командой status утилит ald-client, ald-init и ald-admin

По завершении первичной настройки конфигурационного файла сервера для инициализации домена необходимо выполнить команду:

```
ald-init init
```

Подробнее информацию о создании домена приведена в 8.2.6.1.

Для ввода нового компьютера в домен после первичной настройки конфигурационного файла на клиенте необходимо выполнить команду:

```
ald-client start
```

Примечание. Для удобства ввод нового компьютера в домен может быть выполнен командой `ald-client join <имя сервера домена>-`. В этом случае конфигурационный файл будет настроен автоматически.

В случае изменения конфигурационного файла `/etc/ald/ald.conf` необходимо выполнить команду `commit-config` для того, чтобы изменения вступили в силу:

```
ald-init commit-config
```

на сервере и

```
ald-client commit-config
```

на клиентах.

Пример

Файл `/etc/ald/ald.conf`

```
VERSION=1.7
DOMAIN=.example.ru
SERVER=my-server.example.ru
MINIMUM_UID=2500
DEFAULT_LOGIN_SHELL=/bin/bash
DEFAULT_LOCAL_GROUPS=users,audio,video,scanner
ALLOWED_LOCAL_GROUPS=users,audio,video,scanner
TICKET_MAX_LIFE=10h
TICKET_MAX_RENEWABLE_LIFE=7d
NETWORK_FS_TYPE=cifs
SERVER_EXPORT_DIR=/ald_export_home
CLIENT_MOUNT_DIR=/ald_home
SERVER_FS_KRB_MODES=krb5,krb5i
CLIENT_FS_KRB_MODE=krb5i
SERVER_POLLING_PERIOD=60
SERVER_PROPAGATE_PERIOD=600
UTF8_GECOS=1
SERVER_ON=1
CLIENT_ON=1
```

8.2.4. Шаблоны конфигурационных файлов

Служба ALD в процессе своей работы осуществляет конфигурирование необходимых сетевых служб (Samba, Kerberos, LDAP и т.п.) с помощью их конфигурационных файлов. Для

удобства существуют шаблоны модифицируемых службой ALD конфигурационных файлов, расположенные в каталоге `/etc/ald/config-templates`.

ВНИМАНИЕ! При установке, инициализации, удалении или запуске/остановке службы ALD основные конфигурационные файлы различных служб могут быть перезаписаны на основе шаблонов, что может повлечь потерю внесенных вручную изменений.

Примечание. При необходимости дополнительной настройки служб внесение изменений должно осуществляться не только в основные конфигурационные файлы, но и в их шаблоны.

Перечень шаблонов конфигурационных файлов приведен в таблице 41.

Таблица 41

Имя шаблона	Служба	Описание/ размещение конфигурационного файла
`ald-pam-profile`	`pam-auth-update`	Шаблон PAM
`ald*.ldif`	OpenLDAP	LDAP схемы ALD
`base-init.ldif`	OpenLDAP	LDAP скрипт начальной инициализации БД ALD
`exim-mail.ldif`	OpenLDAP	LDAP схема для Exim
`idmapd.conf`	NFS	`/etc/idmapd.conf`
`kadm5.acl`	Kerberos admin server	`/etc/krb5kdc/kadm5.acl`
`kdc.conf`	Kerberos KDC	`/etc/krb5kdc/kdc.conf`
`kpropd.conf`	Kerberos	`/etc/krb5kdc/kpropd.conf`
`krb5.conf`	Kerberos клиенты	`/etc/krb5.conf`
`ldap.conf`	LDAP клиенты	`/etc/ldap/ldap.conf`
`sssd.conf`	SSSD	`/etc/sssd/sssd.conf`
`sasl2_slapd.conf`	OpenLDAP	Описание для SASL2
`slapd.d17.ldif`	OpenLDAP	LDAP скрипт начальной инициализации LDAP сервера
`smb.conf`	Samba	`/etc/samba/smb.conf`

ВНИМАНИЕ! При ручной правке шаблонов конфигурационных файлов не рекомендуется удалять или менять строки, изначально содержащиеся в шаблоне или содержащие параметризованные значения.

ВНИМАНИЕ! При переустановке ALD или выполнении команд ALD `install-config` шаблоны в `/etc/ald/config-templates` будут перезаписаны из `/usr/lib/ald/config-templates`.

8.2.4.1. Конфигурационные файлы LDAP

К конфигурационным файлам LDAP относятся схемы LDAP и скрипты инициализации сервера LDAP и БД ALD.

Примечание. Скрипты инициализации используются только в процессе создания БД ALD.

При необходимости регистрации дополнительных LDAP схем, необходимо поместить требуемую схему в каталог `/etc/ldap/schema` и добавить ее включение в шаблон `slapd.d17.ldif` по аналогии с остальными.

При необходимости дополнительного начального заполнения БД ALD возможна правка шаблона `base-init.ldif`.

8.2.4.2. Конфигурационные файлы Kerberos

К конфигурационным файлам Kerberos относятся специальные конфигурационные файлы служб сервера Kerberos и конфигурационный файл `/etc/krb5.conf`, содержащий основные настройки домена.

Важной характеристикой является алгоритм защиты аутентификационной информации (`supported_enctypes` в `/etc/krb5kdc/kdc.conf` и `default_tgs_enctypes`, `default_tkt_enctypes`, `permitted_enctypes` в `/etc/krb5.conf`).

Список используемых алгоритмов защиты аутентификационной информации приведен в таблице 42.

Таблица 42

Тип алгоритма	Назначение
`gost-cts`	Отечественные алгоритмы по ГОСТ 28147-89 и ГОСТ Р 34.11-2012, применяются по умолчанию в ALD
`aes256-cts`	Применяется по умолчанию в Kerberos
`des-cbc-crc`	Слабый и устаревший алгоритм, применяется для поддержки NFS, не рекомендуется к использованию
`rc4-hmac`	Применяется для поддержки работы клиентов Samba, так как являлся основным в Windows

В случае отсутствия необходимости использования NFS или утилит Samba (`smbclient`) — типы алгоритмов `des-cbc-crc` и `rc4-hmac` могут не указываться.

Примечание. Для работы с NFS также необходима установка параметра `allow_weak_crypto` в файле `/etc/krb5.conf`, что снижает надежность аутентификации.

ВНИМАНИЕ! Использование NFS не рекомендуется.

8.2.4.3. Конфигурационные файлы Samba

Конфигурационный файл `/etc/smb.conf` содержит описание глобальных настроек и разделяемых ресурсов.

Средства Samba используются в рамках ALD только для централизованного хранения домашних каталогов пользователей. Существует возможность использования других

сетевых разделяемых файловых ресурсов путем описания их в конфигурационном файле `smb.conf` согласно руководству `man` на `smb.conf`.

ВНИМАНИЕ! Возможности по созданию разделяемых ресурсов для сетевой печати не используются, так как не обеспечивают необходимой защиты выводимой информации.

Существует возможность работы с разделяемыми ресурсами с помощью стандартных утилит Samba (`net`, `smbclient`), в том числе с пользовательскими разделяемыми ресурсами (`usershare`). Для этого необходима поддержка сервером Kerberos типа алгоритма `rc4-hmac` (см. 8.2.4.2).

Примечание. В случае необходимости предоставления доступа к разделяемым файловым ресурсам пользователям другого домена (см. 8.2.6.8) следует установить значение параметра `allow trusted domains = yes`.

8.2.4.4. Распространение конфигурационных файлов в домене

Существует возможность распространения конфигурационных файлов в домене. Для этого предназначены команды вида `ald-admin doc-*` (описание команд приведено в руководстве `man ald-admin`).

С помощью команды `ald-admin doc-add` подготовленный конфигурационный файл передается на сервер, где сохраняется в каталоге `/var/lib/ald/documents`. В команде с помощью параметров `--location` и `--file` указываются путь целевого размещения файла на компьютерах домена и путь к загружаемому файлу соответственно.

Службы обработки заданий `aldd` компьютеров сети выполняют обновление указанного конфигурационного файла по указанному при создании пути (должен быть доступен на запись). При этом проверяется время модификации файла. Если время модификации целевого файла новее, перезапись доменной версией не производится.

ВНИМАНИЕ! Механизм должен использоваться с особой осторожностью, поскольку выполняет перезапись локальных конфигурационных файлов версиями с сервера. При этом создаются резервные копии предыдущих версий.

8.2.5. Сценарии сессии пользователя

Astra Linux Directory содержит средства выполнения дополнительных действий при создании новой сессии пользователя или ее завершении в случае работы пользователя в ЕПП.

Для этих целей PAM модуль ALD при создании и завершении сессии пользователя ЕПП исполняет следующие сценарии:

- `/etc/ald/ald.session` — скрипт, исполняющий от имени суперпользователя дополнительные скрипты из каталога `/etc/ald/ald.session.d` во время создания сессии пользователя после монтирования домашнего каталога;

- /etc/ald/ald.reset — скрипт, исполняющий от имени суперпользователя дополнительные скрипты из каталога /etc/ald/ald.reset.d во время завершения сессии пользователя до размонтирования домашнего каталога.

Примечание. Могут существовать и другие каталоги дополнительных скриптов, например, /etc/ald/ald.mac.session.d и /etc/ald/ald.mac.reset.d, для дополнительных этапов работы сессии пользователя.

Рассматриваемый механизм удобен для организации выполнения дополнительных действий при создании и завершении сессии пользователя. Например, одним из обязательных условий работы с домашними каталогами на сетевых ФС является обеспечение корректного их размонтирования. Помешать этому могут процессы, запущенные и не завершившиеся во время работы сессии пользователя и удерживающие открытые файлы в домашнем каталоге.

В случае возникновения подобной ситуации следует определить такие процессы с помощью утилит fuser или lsof, в качестве аргументов которым передается путь к домашнему каталогу пользователя вида /ald_home/имя_пользователя и путь к точке монтирования вида /run/ald.mounts/имя_пользователя, например:

```
fuser /ald_home/user1
fuser /run/ald.mounts/user1
lsof /ald_home/user1
lsof /run/ald.mounts/user1
```

После этого необходимо завершить определенные таким образом процессы. Данная последовательность действий должна быть оформлена в виде скрипта, расположенного в каталоге /etc/ald/ald.reset.d, что позволит обеспечить его выполнение во время завершения сессии пользователя.

Примечание. Настоящий скрипт может быть более интеллектуальным для учета различных свойств процессов или причин их появления.

ВНИМАНИЕ! Поскольку действия выполняются от имени суперпользователя, к разработке подобных сценариев необходимо подходить с особой осторожностью.

8.2.6. Администрирование домена

С помощью утилит администрирования ALD существует возможность выполнения следующих административных действий:

- создание нового домена;
- резервирование/восстановление конфигурации домена;
- контроль целостности конфигурации домена;
- добавление/удаление компьютеров в домен;
- управление учетными записями пользователей домена;

- управление учетными записями сетевых служб домена;
- управление атрибутами СЗИ.

Примечание. Расширения ALD могут изменять состав административных действий и команд утилит администрирования.

Утилиты администрирования могут быть запущены в пакетном режиме для массового выполнения операций. При этом, как правило, используется параметр `--force`.

Примечание. При использовании параметра `--force` необходимые для выполнения пароли администратора и пользователей должны быть переданы утилите с помощью файла паролей.

Операции по администрированию должны выполняться пользователями, обладающими определенными административными полномочиями. В зависимости от назначенных привилегий пользователей ALD можно разделить на следующие группы по полномочиям:

- главный администратор домена `admin/admin` — обладает всеми полномочиями по управлению доменом;
- администраторы домена — пользователи с привилегией `admin`. Обладают полномочиями по управлению конфигурацией домена и учетными записями;
- ограниченные администраторы домена — пользователи с привилегиями `hosts-add` или `all-hosts-add`. Обладают полномочиями по добавлению компьютеров в домен;
- пользователи утилит администрирования — пользователи с привилегией `adm-user`. Обладают полномочиями по запуску утилит администрирования (используется пакетами расширения для детализации полномочий управления);
- обычные пользователи.

ВНИМАНИЕ! Расширения ALD могут привносить свое деление полномочий. Например, пакет `ald-admin-parsec` содержит набор команд управления мандатными атрибутами. При этом предусмотрена соответствующая группа администраторов МАС. Для возможности управления мандатными атрибутами конкретным пользователем ему должна быть предоставлена привилегия `adm-user` и он должен быть добавлен в группу командой `macadmin-add`.

8.2.6.1. Управление конфигурацией домена

Создание нового домена, а так же его резервирование/восстановление осуществляются с помощью утилиты управления сервером домена `ald-init`.

Перед созданием домена на контроллере домена должны быть установлены все требуемые пакеты серверных расширений, в этом случае конфигурация нового домена будет автоматически создана с их поддержкой. Также корректным образом должны быть настроены система разрешения имен и конфигурационный файл `ald.conf` (см. 8.2.3).

В случае указания необходимости сервера ЕПП при начальной установке ОС с диска конфигурационный файл `ald.conf`, как правило, уже содержит корректные значения домена и имени сервера.

Создание или пересоздание домена осуществляется командой `init` утилиты управления сервером домена `ald-init`.

При необходимости может выполняться сохранение резервной копии конфигурации домена командами с префиксом `backup` утилиты управления сервером домена `ald-init`. Восстановление ранее сохраненных резервных копий осуществляется соответствующими командами с префиксом `restore-backup` утилиты управления сервером домена `ald-init` (см. 8.2.6.7).

При появлении в процессе работы сообщений об ошибках или некорректной работе механизмов ЕПП следует воспользоваться командой `test-integrity` утилиты администрирования `ald-admin` для проверки целостности конфигурации домена, что включает в себя проверку состояния и согласованности всех сущностей ALD. В ходе проверки может быть локализована причина ошибок и сбоев, что облегчит их устранение (см. 8.2.7).

8.2.6.2. Использование RPC интефейса

Штатным режимом работы ALD является управление доменом с помощью службы обработки заданий ALD `aldd` сервера с помощью RPC интерфейса.

Утилита `ald-admin` по умолчанию работает в интерактивном режиме с запросом пароля администратора. Также пароли администратора и пользователей могут быть переданы с помощью файла паролей.

Для доменных пользователей возможно выполнение утилиты с использованием существующей аутентификационной информации пользователя (при наличии у него привилегий администрирования домена). При этом указывается параметр командной строки `-c`.

ВНИМАНИЕ! Для корректной работы RPC интерфейса билеты Kerberos пользователей должны обладать свойством `forwardable`. Для домена ALD свойство `forwardable` используется по умолчанию. При получении билетов утилитой `kinit` следует использовать параметр `-f`. В противном случае выдается ошибка вида: «Ошибка подготовки сообщения KRB-CRED».

Существует ряд специальных RPC команд, применяемых к любому компьютеру домена ALD (в качестве аргумента команды может указываться имя компьютера):

- `rpc-status` — получение информации о роли компьютера в домене;
- `rpc-statistics` — получение статистической информации о RPC сервере `aldd` указанного компьютера;

- `rpc-execute` — выполнение указанной команды на удаленном компьютере (команда выполняется от имени инициировавшего запрос пользователя домена).

Описание команд может быть получено с помощью встроенной команды помощи `help`.

Примечание. Список RPC команд сервера может быть получен с помощью команды `rpc-statistics` с параметром `--commands`.

ВНИМАНИЕ! Существует возможность запрета исполнения выбранных RPC указанием параметра `RPC_RESTRICTED` в конфигурационном файле `/etc/ald.conf` конкретного компьютера.

8.2.6.3. Управление учетными записями

В ЕПП различаются учетные записи пользователей домена, учетные записи компьютеров домена и учетные записи сетевых служб, работающих в среде ЕПП.

Учетная запись пользователя домена содержит всю необходимую информацию о пользователе ЕПП и включает в себя: соответствующего принципала Kerberos, политику паролей, свойства, необходимые для входа пользователя в систему, настройки подключения домашнего каталога, привилегии пользователя ЕПП и его атрибуты СЗИ.

Привилегии ALD и указанные ограничения могут быть установлены для учетной записи с помощью команды `user-ald-cap` утилиты администрирования `ald-admin`.

Также учетная запись пользователя может содержать ограничения по входу в домен. В качестве ограничений используется список компьютеров, на которых он может осуществлять вход, и признак временной блокировки.

ВНИМАНИЕ! После создания новой учетной записи список разрешенных для входа компьютеров пуст: пользователь не имеет права входа в систему. Список компьютеров, с которых ему будет разрешен вход, должен быть явно указан после создания учетной записи.

Учетная запись пользователя может обладать административными привилегиями или входить в группы администраторов, заданные расширениями ALD.

ВНИМАНИЕ! Удаление учетной записи пользователя может быть выполнено только администратором, обладающим доступом ко всем его атрибутам (входящим во все необходимые группы администраторов).

ВНИМАНИЕ! Существует некоторое время для распространения информации о создании или удаления пользователя. Это связано с механизмами кеширования NSS. При пересоздании пользователя с тем же именем могут возникать ошибки (например, входа в систему, монтирования домашнего каталога и т.п.) из-за выдачи на удаленных системах устаревшего идентификатора пользователя.

Учетная запись компьютера домена представляет собой набор принципалов Kerberos для функционирования компьютера в домене.

Ввод нового компьютера в домен осуществляется с помощью запущенной на нем утилиты `ald-client` командой `commit-config` (возможно с параметрами). При этом пользователь должен обладать полномочиями по добавлению компьютера в домен.

Примечание. Для удобства ввод нового компьютера в домен может быть выполнен командой `ald-client join <имя сервера домена>-`. В этом случае конфигурационный файл будет настроен автоматически. Также автоматически будет создана учетная запись компьютера в домене.

С помощью утилиты `ald-admin` учетной записи компьютера может быть добавлено описание или она может быть удалена.

Учетная запись службы домена представляет собой принципала Kerberos для функционирования службы в домене.

ВНИМАНИЕ! Каждая служба, поддерживающая сквозную аутентификацию Kerberos, должна обладать принципалом Kerberos, т. е. быть зарегистрированной в домене. После регистрации в домене набор ключей службы должен быть выгружен в файл, указанный в ее конфигурации.

В ALD для предоставления службам определенных полномочий по получению информации из домена используется объединение служб в группы служб. Например, для получения мандатных атрибутов пользователей служба должна входить в группу служб `mac`.

Для облегчения конфигурирования сетевых служб, работающих в среде ЕПП, предусмотрены команды управления учетными записями служб утилиты `ald-admin` и команды выгрузки ключей утилиты `ald-client`.

Указанные команды имеют префиксы `service-` и `svc-`.

ВНИМАНИЕ! В случае добавления компьютера в домен, пересоздания домена или принципалов служб может потребоваться удаление файлов типа `krb5.keytab`, содержащих выгруженные ранее ключи.

Детальное описание команд приведено в руководстве `man`. Настройка некоторых сетевых служб приведена в 8.5.

8.2.6.4. Ограничения по выборке данных из LDAP

Существуют ограничения по получению данных от службы каталогов LDAP. По умолчанию разрешается получать не более 500 записей.

ВНИМАНИЕ! Возможны нарушения работы ЕПП в случае превышения числа пользователей или компьютеров этого значения.

Для гибкого управления ограничениями предусмотрены команды утилиты `ald-admin`: `ldap-limits` для просмотра и `ldap-setlimit` для установки.

Службы каталогов LDAP поддерживают ограничения для различных пользователей или групп пользователей по размеру и времени выполнения выборки. При этом существуют

мягкие ограничения, применяемые по умолчанию, которые могут быть превышены заданием параметров выборки в прикладном ПО, и жесткие, которые не могут быть превышены.

Команда установки ограничений имеет следующий синтаксис:

```
ald-admin ldap-setlimit <кому> <вид ограничения>
```

где видами ограничения могут быть:

- `size=число` — единое задание мягкого и жесткого ограничении по размеру выборки;
- `size.soft=число` — задание мягкого ограничения по размеру выборки;
- `size.hard=число` — задание жесткого ограничения по размеру выборки;
- `time=секунды` — единое задание мягкого и жесткого ограничении по времени выполнения выборки;
- `time.soft=секунды` — задание мягкого ограничения по времени выполнения выборки;
- `time.hard=секунды` — задание жесткого ограничения по времени выполнения выборки.

В качестве аргумента команды `<кому>` могут выступать следующие значения:

- `*` — все, включая анонимных и аутентифицированных пользователей;
- `anonymous` — анонимные пользователи;
- `users` — аутентифицированные пользователи;
- `self` — ассоциированный с целью пользователь;
- `dn...` — варианты синтаксиса DN;
- `group...` — варианты синтаксиса групп.

Примечание. Перед установкой ограничений LDAP рекомендуется ознакомиться с доступной документацией по работе служб каталогов LDAP.

Подробное описание команд работы с ограничениями LDAP приведены в руководстве `man ald-admin`.

8.2.6.5. Регистрация действий администратора и протоколирование

При работе компоненты ALD ведут журналы своей работы. В журналах фиксируются информация о выполняемых действиях и ошибках. При этом фиксируется дата и время возникновения события, тип события и имя исполняемого модуля с указанием идентификатора процесса.

Доступны следующие журналы работы:

- `~./ald/ald-admin.log`, `~./ald/ald-init.log`, `~./ald/ald-client.log` — журналы работы утилит `ald-admin`, `ald-init`, `ald-client` соответственно. Располагаются в домашнем каталоге пользователя, который запускал их на исполнение;

- /var/log/ald/aldd.log — журнал работы службы обработки заданий ALD aldd.

Способ вывода журналов, их размещение и детализация могут быть заданы для каждой из утилит или служб при их запуске с помощью следующих параметров командной строки:

Таблица 43

Параметр	Описание
--log-dest=способы	Задает способ журнализации, где аргумент принимается как набор разрядов: - 1 (0x1) — stderr; - 2 (0x2) — syslog; - 3 (0x4) — csvlog. По умолчанию для утилит используется stderr+csvlog, а для служб — syslog+csvlog
--log-file=путь	Задает путь к файлу журнала (в случае использования способа csvlog)
--log-level=уровень	Задает детализацию журнала: - 0 — ошибки; - 1 — предупреждения; - 2 — уведомления; - 3 — информация; - 4 — отладка

Регистрация действий администратора по управлению доменом осуществляется централизовано на сервере домена. При этом по умолчанию вывод информации осуществляется в следующие файлы:

- /var/log/ald/aldlog.log — журнал регистрации изменений шаблонов протоколирования;

- /var/log/ald/audit.log — журнал регистрации согласно настроенным шаблонам протоколирования.

Примечание. В случае необходимости может быть настроена переадресация журналов регистрации действий администратора в системный журнал syslog с помощью конфигурационного файла следующего вида, размещаемого в каталоге /etc/rsyslog.d/:

```
$ModLoad imfile
$InputFileName /var/log/ald/audit.log
$InputFileTag ald_audit
$InputFileStateFile stat_ald_audit
$InputFileSeverity notice
$InputFilePollInterval 1
$InputRunFileMonitor
```

Управление регистрацией действий администратора производится с помощью команд вида `'ald-admin logging-*'`, которые позволяют изменять путь к файлу регистрации событий (журнал регистрации изменений шаблонов протоколирования имеет фиксированное расположение), создавать или изменять шаблоны протоколирования.

Шаблон протоколирования состоит из имени, ldap-суффикса и режима протоколирования:

- `all` — регистрировать все события;
- `succ` — регистрировать успешные события;
- `fail` — регистрировать неуспешные события;
- `none` — не выполнять регистрацию событий.

ВНИМАНИЕ! Не рекомендуется без особой необходимости добавлять или изменять суффиксы шаблонов протоколирования.

8.2.6.6. Домашние каталоги и особенности монтирования сетевых ФС

Централизация хранения информации об окружении пользователей подразумевает и централизованное хранение домашних каталогов пользователей. Для этого используется СЗФС CIFS (см. 6.9).

Для хранения домашних каталогов, содержащих незащищенные данные, могут быть использованы и другие сетевые ФС (например, NFS4). ALD в настоящее время поддерживает автоматическое монтирование только СЗФС CIFS и NFS4. Также для хранения домашних каталогов пользователя может быть использована и локальная ФС компьютера.

Учетная запись пользователя ALD содержит информацию о типе ФС домашнего каталога пользователя и его расположении (сервер домашних каталогов для сетевых ФС и путь к каталогу для локальных ФС). По умолчанию в качестве типа ФС используется СЗФС CIFS, а в качестве расположения — контроллер домена.

Примечание. Особенность организации домашних каталогов пользователя включает в себя обеспечение возможности перехода пользователя к своим каталогам с другой классификационной меткой с помощью ссылки `mac`. Использование таких ссылок в samba по умолчанию запрещено. Для разрешения этой возможности используется глобальный параметр `allow insecure wide links` в шаблоне конфигурационного файла samba (см. 8.2.4.3).

Пакет расширения `ald-client-fs` позволяет на любом компьютере домена развернуть сервер домашних каталогов, который впоследствии можно будет указать как расположение домашних каталогов. Регистрация, запуск и останов сервера осуществляется с помощью расширения командного интерфейса утилиты управления клиентом `ald-client`.

ВНИМАНИЕ! Существует возможность изменения сервера расположения домашнего каталога пользователя. В этом случае домашний каталог пользователя должен быть

физически перемещен со старого сервера на новый, в противном случае пользователь не сможет войти в систему. Такая же ситуация может произойти при замене основного сервера резервным.

Монтирование домашних каталогов выполняется PAM-модулем ALD автоматически при входе пользователя. При этом могут проверяться ограничения на тип ФС домашнего каталога пользователя.

ВНИМАНИЕ! Существует некоторое время для распространения информации о создании или удаления пользователя. Это связано с механизмами кеширования NSS. При пересоздании пользователя с тем же именем могут возникать ошибки (например, ошибки входа в систему, монтирования домашнего каталога и т.п.) из-за выдачи на удаленных системах устаревшего идентификатора пользователя. При возникновении таких ошибок следует перезапустить на используемых компьютерах службу `sssd` и обеспечить корректные значения прав доступа к каталогу пользователя на сервере домашних каталогов.

ВНИМАНИЕ! Для корректной работы с монтированием домашних каталогов необходимо обеспечить освобождение точек монтирования при завершении сессии пользователя (см. 8.2.5).

Существует возможность на серверах домашних каталогов (файл-серверах) заводить общие папки, доступные для пользователей. Для конфигурирования файл-сервера следует руководствоваться документацией и справкой по используемой ФС. Монтирование таких каталогов может быть выполнено при помощи команды `mount` или редактированием конфигурационного файла `fstab`. Автоматическое монтирование может быть обеспечено PAM-модулем `pam_mount`.

Примечание. При необходимости работы с разделяемыми ресурсами с помощью стандартных утилит Samba (`net`, `smbclient`), в том числе с пользовательскими разделяемыми ресурсами (`usershare`), могут потребоваться дополнительные настройки (см. 8.2.4.3).

8.2.6.7. Создание резервных копий и восстановление

В целях уменьшения времени на восстановление работоспособности сервера в случае возникновения программно-аппаратных сбоев предусмотрено создание резервной копии баз данных сервера домена.

Резервирование/восстановление домена осуществляются с помощью утилиты управления сервером домена `ald-init`.

ВНИМАНИЕ! Программная конфигурация ALD сервера, на котором будет выполняться восстановление, должна точно соответствовать той, при которой выполнялось создание резервной копии. Должны быть установлены все требуемые пакеты серверных расширений ALD. Также корректным образом должна быть настроена система разрешения имен (см. 8.2.3).

Существует несколько вариантов создания резервной копии следующими командами утилиты управления сервером ALD `ald-init`:

- `backup` — создание физической копии контроллера домена: в этом случае в подкаталоге `.ald` домашнего каталога пользователя, выполняющего операцию, создаются два архива `ald-base.tar.gz` и `ald-keys.tar.gz`, содержащие архив фрагментов ФС сервера с информационными БД и БД ключевой информации соответственно. Данный вариант является единственным, при котором сохраняется ключевая информация и пароли пользователей;
- `backup-ldif` — создание логической копии LDAP БД контроллера домена: в этом случае в подкаталоге `.ald` домашнего каталога пользователя, выполняющего операцию, создается LDIF файл БД LDAP контроллера домена с именем по умолчанию вида `ald.<имя_домена>.ldif`;
- `backup-portable` — создание логической копии БД контроллера домена в переносимом текстовом формате: в этом случае в подкаталоге `.ald` домашнего каталога пользователя, выполняющего операцию, создается текстовый файл с именем по умолчанию вида `ald.{имя домена}.pbk.gz`.

Для восстановления перечисленных вариантов резервных копий используются команды утилиты управления сервером ALD `ald-init restore-backup`, `restore-backup-ldif` и `restore-backup-portable` соответственно. При этом пересоздаются базы данных LDAP и Kerberos.

ВНИМАНИЕ! При использовании вариантов создания логической копии командами `backup-ldif` и `backup-portable` ключевая информация и пароли пользователей не сохраняются. После восстановления требуется назначение новых паролей пользователей, повторный ввод рабочих станций в домен и пересоздание локальных файлов ключей всех зарегистрированных служб. При этом в процессе восстановления необходимо задать пароль по умолчанию для пользователей. Важно обеспечить введение такого пароля, который будет удовлетворять требованиям всех парольных политик домена. В противном случае восстановление не может быть выполнено.

Примечание. После выполнения восстановления служба заданий ALD `aldd` выполняет настройку привилегий пользователей и другие необходимые действия. При этом может выполнятся перезапуск различных служб сервера, в том числе и службы администрирования Kerberos. Следует дождаться завершения всех настроек перед выполнением других административных действий.

После выполнения восстановления следует воспользоваться командой `test-integrity` утилиты администрирования `ald-admin` для проверки целостно-

сти конфигурации домена, что включает в себя проверку состояния и согласованности всех сущностей ALD.

8.2.6.8. Доверительные отношения между доменами

В случае наличия нескольких доменов ALD поддерживается возможность обращения клиентов одного домена к ресурсам другого домена. Для этого между доменами должны быть установлены доверительные отношения.

В ALD используются симметричные доверительные отношения между доменами. В случае необходимости ограничения доступа клиентам чужого домена к той или иной службе, соответствующие настройки ограничения доступа должны быть выполнены средствами конфигурирования самой службы.

При работе с пользователями других доменов должны использоваться имена их учетных записей Kerberos вида `<имя_пользователя>@<REALM>`.

Для установки доверительных отношений между доменами необходимо в каждом из них произвести добавление другого домена командой `ald-admin trusted-add`. Детальное описание команд приведено в руководстве `man` для утилиты `ald-admin`.

ВНИМАНИЕ! Введение доверительных отношений требует изменения конфигурационных файлов на каждом компьютере домена. Изменения будут внесены после перезагрузки компьютеров. Для оперативного изменения конфигурации на отдельном компьютере без перезагрузки может быть использована команда `ald-client restart`.

ВНИМАНИЕ! Доверительные отношения не сохраняются при создании резервной копии домена командами `backup-ldif` и `backup-portable` и должны быть установлены заново после пересоздания домена (см. 8.2.6.7).

Примечание. В случае необходимости предоставления доступа к разделяемым файловым ресурсам пользователям могут требоваться дополнительные настройки (см. 8.2.4.3).

8.2.6.9. Создание резервного сервера ALD

Для обеспечения отказоустойчивости домена ALD реализована возможность создания резервного сервера ALD в режиме master-slave, а также функция репликации серверов ALD в режиме multimaster.

Под резервным сервером ALD в режиме master-slave подразумевается сервер, на который реплицируется конфигурация основного контроллера домена и который может заменить собой основной контроллер домена в случае необходимости (например в случае выхода из строя основного контроллера домена) без потери служебной информации: учетных записей пользователей, паролей, политик паролей и другой централизованной информации.

При использовании репликации серверов ALD в режиме multimaster каждый из серверов является основным контроллеров домена и изменения на любом из них реплицируются на остальные сервера.

Примечание. Резервный сервер ALD позволяет обращаться за информацией к службе каталогов LDAP и службе аутентификации Kerberos, что обеспечивает работу пользователей даже при сбое основного контроллера домена. Для этого резервный сервер должен быть указан в соответствующих конфигурационных файлах.

ВНИМАНИЕ! Репликация сервера ALD выполняется только для контроллера домена и не обеспечивает перенос домашних каталогов пользователей. Для сохранения домашних каталогов рекомендуется использовать выделенный сервер домашних каталогов (см. 8.2.6.6).

ВНИМАНИЕ! Расширения ALD могут привносить свое деление полномочий, что может потребовать дополнительных настроек для обеспечения полной репликации баз данных ALD.

Создание резервного сервера ALD и управление репликацией осуществляется утилитой управления сервером ALD `ald-init`.

ВНИМАНИЕ! Состав установленных пакетов ALD на основном и резервном серверах при использовании режима master-slave (или на всех серверах ALD в режиме `multimaster`) должен быть идентичным.

Создание резервного сервера ALD в режиме master-slave заключается в выполнении команды инициализации резервного сервера:

```
ald-init init --slave <имя_хоста>
```

Для выполнения функции резервирования в режиме master-slave используются различные механизмы репликации, в том числе и собственные механизмы репликации служб LDAP и Kerberos.

В ходе выполнения команды инициализации резервного сервера в режиме master-slave будет выведена информация об обнаруженном основном сервере домена, и произведены настройки резервного сервера. После выполнения инициализации на резервный сервер будет осуществляться репликация всей необходимой информации с основного сервера.

Примечание. Репликация в режиме master-slave выполняется от имени системной учетной записи службы обработки заданий ALD `aldd`, запущенной на резервном сервере. Указанная учетная запись входит в группу администраторов, что позволяет ей реплицировать данные домена.

При развернутом резервном сервере ALD в режиме master-slave администрирование домена выполняется только на основном контроллере домена.

ВНИМАНИЕ! Механизм создания резервных серверов ALD в режиме master-slave не является «горячим резервом». Замена основного контроллера домена резервным предполагает действия системного администратора по замене основного сервера резервным (см. 8.2.6.10).

В случае необходимости резервный сервер может быть переведен в оперативный режим работы командой `ald-init promote`.

Примечание. Репликация баз данных в режиме master-slave выполняется в определенные промежутки времени. Например, базы Kerberos по умолчанию обновляются раз в 10 минут, что задается параметром `SERVER_PROPAGATE_PERIOD` в конфигурационном файле `/etc/ald/ald.conf` основного сервера (см. 8.2.3).

Удаление экземпляра резервного сервера в режиме master-slave может быть выполнено командой `ald-init destroy`.

Создание сервера ALD в режиме multimaster выполняется командой инициализации сервера:

```
ald-init init --master <имя_хоста>
```

Для выполнения функции репликации в режиме multimaster используются механизмы репликации служб LDAP.

После настройки репликации серверов ALD в режиме multimaster администрирование домена можно выполнять с любого из серверов. Каждый сервер будет передавать информацию о внесенных изменениях в конфигурацию домена другим серверам, функционирующим в режиме multimaster.

8.2.6.10. Замена основного сервера резервным

В случае выхода из строя основного контроллера домена администратор должен произвести следующие действия по замене основного сервера домена резервным:

1) перевести один из резервных серверов в оперативный режим работы командой `ald-init promote`.

   ВНИМАНИЕ! При переводе резервного сервера в оперативный режим основной сервер принудительно исключается из домена во избежание конфликтов. После восстановления он может быть возвращен в домен в качестве резервного;

2) на всех клиентских машинах, включая сервер домашних каталогов (если есть), в конфигурационном файле `/etc/ald/ald.conf` в качестве параметра `SERVER` указать новый контроллер домена (бывший резервный сервер). После этого должна быть выполнена команда `ald-client commit-config`.

8.2.7. Проверка целостности конфигурации и устранение ошибок

В ALD встроены средства проверки целостности конфигурации домена, что включает в себя проверку состояния и согласованности всех сущностей ALD.

При возникновении в процессе работы сообщений об ошибках или некорректной работе механизмов ЕПП следует воспользоваться командой `test-integrity` утилиты администрирования `ald-admin`.

В ходе проверки может быть локализована причина ошибок и сбоев, что облегчит их устранение.

При выполнении команды производится проверка состояния и согласованности сущностей домена, при этом отображается текущая проверяемая группа сущностей и результат проверки (при указании параметра `--verbose` дополнительно выводиться текущая проверяемая сущность). В результате выполнения проверки могут быть выведены следующие диагностические сообщения:

```
Проверка целостности базы данных ALD сформировала диагностических сообщений: N
1: <диагностическое сообщение 1>
...
```

При обнаружении критичных ошибок команда завершается с выдачей сообщения об ошибке:

```
Проверка целостности базы данных ALD выявила ошибок: N.
При нормальном функционировании ALD таких ошибок возникать не должно.
Попробуйте удалить ошибочные сущности и создать их заново. Если это
не поможет, или если появятся новые ошибки - обратитесь к разработчикам.
```

Диагностические сообщения могут содержать рекомендации по устранению выявленного нарушения.

ВНИМАНИЕ! Рекомендуется использовать предлагаемый вариант устранения нарушения средствами ALD, так как для ручного устранения нарушений требуются глубокие знания технологий, механизмов функционирования и инструментов администрирования LDAP и Kerberos.

ВНИМАНИЕ! Перед критичными исправлениями, требующими пересоздания домена, рекомендуется, по возможности, сохранить резервную копию домена. После восстановления из резервной копии некоторые ошибки могут исчезнуть.

Часть ошибок может быть устранена автоматически. Для этого необходимо указание параметра `--fix` при вызове команды `ald-admin test-integrity`. Автоматически выполняются следующие действия:

- создание недостающих индексов и ограничений уникальности в LDAP;
- удаление несуществующих членов групп пользователей, компьютеров, служб и администраторов;
- пересоздание политик паролей по существующей информации (LDAP или Kerberos);
- синхронизация компьютеров по информации из Kerberos (`host-renew`);

- синхронизация параметров политик паролей из LDAP в Kerberos;
- настройка глубины истории заданий и их ротация;
- корректировка списка разрешенных компьютеров и групп компьютеров;
- отбор административных прав при любом нарушении свойств пользователей.

Список возможных ошибок и способов их устранения приведен в таблице 44.

ВНИМАНИЕ! При вызове команды `ald-admin test-integrity` с параметром `--fix` выполняются действия по исправлению сразу всех ошибок. Во избежание неверных исправлений следует учитывать характер автоматических действий, описанных в таблице 44.

Примечание. В зависимости от установленных расширений состав проверок и диагностических сообщений может отличаться. Подход к устранению ошибок, не приведенных в таблице, может быть выполнен по аналогии с описанными.

Таблица 44

Ошибки	Способы устранения
Ошибки общего вида	
Какая-либо сущность ALD не найдена или нарушен синтаксис имени сущности	Сущность может быть либо пересоздана заново, либо удалена командами утилиты `ald-admin`. Некоторые сущности могут быть созданы или удалены средствами администрирования LDAP и Kerberos
Нарушен синтаксис или значение свойств и параметров сущностей ALD	Сущность может быть модифицирована командами утилиты `ald-admin`. Некоторые сущности могут быть модифицированы средствами администрирования LDAP и Kerberos
Проверка конфигурации домена	
Имя домена отличается от значения в ald.conf	Исправление файла ald.conf, если имя домена верно
Версия домена отличается от значения в ald.conf	Исправление файла ald.conf, если не используется режим совместимости
Проверки LDAP	
Модуль LDAP не зарегистрирован	Неверно задан шаблон домена `slapd.16.ldif`. Необходимо указать загрузку указанного модуля и пересоздать домен. Существует возможность решения средствами администрирования LDAP, но в этом случае без модификации шаблона ошибка может повториться после пересоздания домена
Индекс LDAP не зарегистрирован. Ограничение уникальности LDAP не зарегистрировано	При указании параметра `--fix` автоматически создается
Проверка системных принципалов	

Продолжение таблицы 44

Ошибки	Способы устранения
Не найден системный принципал в БД Kerberos	Необходимо его создать с помощью команды kadmin(1) и сгенерировать для него ключ в файле ключей. Или проинициализировать сервер заново с помощью команд `ald-init init` или `restore-backup(-ldif)`
Проверка компьютеров	
`host\...` принципалы не найдены для следующих компьютеров...	Удалить и пересоздать с помощью команд `host-*` или создать с помощью команды kadmin(1) и сгенерировать для него ключ в файле ключей
Следующие компьютеры не найдены в LDAP, хотя их принципалы присутствуют в Kerberos:...	Обновить информацию в LDAP командой `host-renew` или удалить их из БД Kerberos с помощью команды kadmin(1). При указании параметра `--fix` выполняется команда `host-renew`
Проверка групп компьютеров	
Компьютер в группе компьютеров неверен или не найден в LDAP	Модифицировать состав группы компьютеров или ввести указанный компьютер в домен. При указании параметра `--fix` компьютер удаляется из группы
Проверка серверов ALD	
Компьютер для сервера ALD не найден	Критичная ошибка конфигурации. При необходимости следует пересоздать домен
Сервер с идентификатором уже существует	Изменить идентификатор одного из серверов путем модификации соответствующего файла `ald.conf`
Основной контролер домена ALD уже был найден	Критичная ошибка конфигурации. Выявить неверный сервер ALD. Если ошибка во флагах компьютера, следует исправить флаги с помощью `host-mod`, в противном случае вывести неверный сервер из домена
Проверка политик паролей	
Следующие политики паролей не найдены в LDAP/Kerberos (но присутствуют в Kerberos/LDAP):...	Удалить их и создать заново. При указании параметра `--fix` пересоздаются по оставшейся части информации
Политика паролей `default` не найдена в Kerberos	Создать вручную командой kadmin(1). При указании параметра `--fix` создается
Политика паролей не найдена в Kerberos/LDAP	Удалить ее и создать заново. При указании параметра `--fix` пересоздается по оставшейся части информации
Политика паролей в LDAP не совпадает с аналогичной в Kerberos	Установить параметры политики заново. При указании параметра `--fix` обновляется из LDAP
Проверка пользователей	

Продолжение таблицы 44

Ошибки	Способы устранения
Для принципала отсутствует соответствующий пользователь в БД LDAP	Если принципал не создан вручную для других целей, следует удалить его утилитой kadmin(1) и создать пользователя заново
Отсутствует принципал Kerberos для пользователя	Создать принципал вручную с помощью kadmin(1) или удалить и создать пользователя заново
Политика паролей пользователя в LDAP не совпадает с Kerberos	Установить политику пользователя заново. При указании параметра `--fix` пользователю назначается политика паролей из LDAP
Пользователь имеет UID, который меньше, чем MINIMUM_UID	Ошибка создания пользователя. Удалить пользователя и создать его заново с правильным UID или изменить с помощью команды `user-mod`.
Пользователь ссылается на несуществующую политику	Изменить неправильные параметры пользователя. При указании параметра `--fix` пользователю назначается политика паролей по умолчанию «`default`»
Пользователь ссылается на несуществующую группу	Изменить неправильные параметры пользователя. При указании параметра `--fix` пользователю назначается группа по умолчанию «`Domain Users`»
Неправильный синтаксис домашнего каталога пользователя. Неправильный синтаксис командной оболочки пользователя. Неправильный синтаксис GECOS пользователя.	Изменить неправильные параметры пользователя
Следующие компьютеры, указанные в списке привилегий пользователя, неверны или не найдены в БД LDAP	Добавить их в домен или изменить привилегии пользователя командой `user-ald-cap`. При указании параметра `--fix` компьютеры удаляются из списка привилегий пользователя
Следующие группы компьютеров, указанные в списке привилегий пользователя, неверны или не найдены в БД LDAP	Добавить их в домен или изменить привилегии пользователя командой `user-ald-cap`. При указании параметра `--fix` группы компьютеров удаляются из списка привилегий пользователя
Проверка групп	
Группа имеет GID, который меньше, чем MINIMUM_GID	Ошибка создания группы. Удалить группу и создать ее заново с правильным GID или изменить с помощью команды `group-mod`.
Группа содержит несуществующего пользователя	Изменить состав группы с помощью команды `group-mod`. При указании параметра `--fix` пользователь удаляется из группы
Проверка администраторов	
Группа администраторов не найдена	Критическая ошибка. Необходимо пересоздать домен

Окончание таблицы 44

Ошибки	Способы устранения
Следующие bind-DN не найдены в группе администраторов:...	Добавить с помощью команд `ald-admin` недостающих членов в группу или пересоздать домен. При указании параметра `--fix` добавляются автоматически
Служба присутствует в группе администраторов, но не найдена в базе данных	Модифицировать группу администраторов или создать указанную службу заново. При указании параметра `--fix` служба удаляется из группы администраторов
Пользователь присутствует в группе администраторов, но не обладает привилегией администратора. Пользователь обладает привилегией администратора, но не присутствует в группе администраторов	Установить правильные привилегии пользователя командой `user-ald-cap`. При указании параметра `--fix` пользователь удаляется из группы администраторов или лишается привилегий
Проверка служб	
Компьютер службы не найден в LDAP	Если принципал не создан вручную для других целей, следует удалить служб или ввести указанный компьютер в домен
Проверка групп служб	
Группа служб содержит службу с неверным именем. Группа службу содержит несуществующую службу	Изменить состав группы с помощью команды `sgroup-svc-rm`. При указании параметра `--fix` служба удаляется из группы
Проверка доменных документов	
Неверная версия документа. Неверный путь к файлу	Исправить свойства документа
Файл не существует	Удалить документ и создать заново
Проверка доверенных доменов	
Доверенный домен области не найден. Inbound/Outbound TGT принципал не найден	Удалить доверенный домен и создать заново
Не удалось разыменовать KDC домена	Проверить настройку системы разрешения имен и наличие связи с указанным сервером. При необходимости удалить доверенный домен
Проверка серверных заданий	
Параметр `task-history` должен быть числом от 2 до 2000	Установите корректное значение. При указании параметра `--fix` устанавливается значение по умолчанию 100
Количество завершенных заданий превышает параметр `task-history`	Удалить задания вручную. При указании параметра `--fix` выполняется ротация заданий

8.3. Служба FreeIPA

Служба FreeIPA предназначена для реализации централизованного управления сетевыми службами, идентификацией и аутентификацией, а также для установки доверительных отношений и обеспечения взаимодействия Linux-систем с доменом Active Directory (AD).

В FreeIPA используется системный демон SSSD (System Security Services Daemon), управляющий доступом к удаленным каталогам и механизмам аутентификации, входящим в состав FreeIPA.

FreeIPA основывается на технологиях LDAP и Kerberos и поддерживает миграцию учетных записей из LDAP и NIS. FreeIPA предоставляет следующий функционал:

- DNS сервер;
- сервер времени;
- управление доступом на основе политик.

Управление FreeIPA доступно как через терминал, так и через web-интерфейс.

FreeIPA позволяет создавать централизованные системы по управлению идентификацией пользователей, заданию политик доступа и аудита для сетей на основе ОС. В состав FreeIPA входят следующие компоненты:

- сервер 389 Directory Server — используется в качестве сервера LDAP;
- MIT Kerberos 5 — используется для аутентификации и единой точки входа;
- Apache и Python — используются для управления ПО, входящим в состав FreeIPA;
- BIND и DHCP — используются для управления службой DNS в сети.

В соответствии с моделью мандатного доступа служба FreeIPA реализует для зарегистрированных с помощью службы пользователей:

- задание уровней конфиденциальности;
- задание уровня целостности;
- задание PARSEC-привилегий.

8.3.1. Структура

Основу доменной структуры FreeIPA составляет домен IPA, в который может входить множество DNS доменов. Домен IPA воспринимается внешним доменом AD как отдельный лес доменов AD, при этом домен Primary DNS домена IPA выступает в роли корневого домена леса доменов FreeIPA.

Интеграция домена IPA с доменом AD возможна двумя способами:

- синхронизация учетных записей пользователей и их паролей (не рекомендуется);
- создание доверительных отношений между лесами доменов (рекомендуется).

Далее приводится описание только рекомендованного способа интеграции на основе доверительных отношений между доменом AD и доменом IPA.

В целях обеспечения отказоустойчивости FreeIPA поддерживает работу в режиме «ведущий–ведомый», при этом рекомендуется использовать две или три (но не более четырех) реплики FreeIPA.

8.3.2. Состав

Все необходимые компоненты службы FreeIPA входят в состав пакетов, приведенных в таблице 45.

Таблица 45

Наименование	Описание
`freeipa-admintools`	Пакет администрирования FreeIPA, содержит набор утилит по управлению сервером FreeIPA
`freeipa-client`	Клиентская часть FreeIPA. Пакет должен устанавливаться на все клиентские компьютеры, входящие в домен
`freeipa-server`	Серверная часть FreeIPA. Пакет должен устанавливаться на контроллере домена. При установке данного пакета также устанавливается средство администрирования `ipa` и клиентская часть
`freeipa-server-dns`	Пакет, предназначенный для установки или интеграции с DNS сервером
`freeipa-server-trust-ad`	Пакет для интеграции с Active Directory от Microsoft путем установки доверительных отношений
`fly-admin-freeipa-server`	Графическая утилита управления FreeIPA
`astra-freeipa-server`	Инструмент командной строки управления FreeIPA
`fly-admin-freeipa-client`	Графическая утилита управления FreeIPA с клиентского компьютера
`astra-freeipa-client`	Инструмент командной строки управления FreeIPA с клиентского компьютера

Служба FreeIPA состоит из ядра, отвечающего за основной функционал системы, ряда интерфейсов (LDAP, Kerberos, Config, RPC) и модулей расширения, предназначенных для расширения командного интерфейса утилит и настройки необходимых служб и подсистем, что позволяет повышать функциональность FreeIPA, устанавливая дополнительные пакеты.

В FreeIPA возможно использование следующих модулей расширения, при этом наименование пакета расширения отражает его назначение:

- `freeipa-client-...` — расширение, необходимое клиентской части FreeIPA;
- `freeipa-admintools-...` — расширение утилиты администрирования FreeIPA;

- freeipa-server-... — расширение, необходимое для организации хранения атрибутов на сервере FreeIPA.

Описание пакетов приведено в справочных страницах man, список которых приведен в таблице 46.

Таблица 46

Наименование	Описание
ipa	Администрирование домена IPA
default.conf	Образец конфигурационного файла default.conf
ipa-client-install	Настройка клиентской части FreeIPA
ipa-server-install	Настройка серверной части FreeIPA
ipa-server-upgrade	Обновление сервера FreeIPA
ipa-dns-install	Утилита добавления DNS как службы на серверной части FreeIPA
ipa-backup	Резервное копирования мастер-сервера FreeIPA
ipactl	Интерфейс управления серверной частью FreeIPA
ipa-advise	Предоставляет рекомендации по конфигурациям для различных вариантов использования
ipa-cacert-manage	Управление сертификатами CA на FreeIPA
ipa-certupdate	Обновление локальных БД сертификатов FreeIPA вместе с сертификатами от сервера
ipa-client-automount	Настройка автомонтирования и ФС NFS для FreeIPA
ipa-compat-manage	Включение и выключение модуля совместимости схемы
ipa-csreplica-manage	Управление репликой FreeIPA CS
ipa-getcert	Инструмент ipa-getcert выдает запросы службе certmonger от имени вызывающего пользователя
ipa-getkeytab	Получение keytab-файла. Keytab — это файл с одним или несколькими секретными ключами для принципала Kerberos. Keytab-файлы используются службами, например, sshd, при аутентификации Kerberos
ipa-join	Подключение хоста к области FreeIPA и получение keytab-файла для размещения службы хоста принципала Kerberos
ipa-kra-install	Установка KRA на серверной части FreeIPA
ipa-ldap-updater	Обновление настроек FreeIPA LDAP
ipa-managed-entries	Включения и выключение модулей схемы управляемых модулей ввода
ipa-nis-manage	Включение и выключение модуля прослушивателя NIS
ipa-otptoken-import	Импорт OTP-токенов из RFC 6030 XML файлов
ipa-replica-conncheck	Проверка сетевого подключения реплики и мастер-сервера перед установкой
ipa-replica-install	Создание реплики FreeIPA

Окончание таблицы 46

Наименование	Описание
ipa-replica-manage	Управление репликой FreeIPA
ipa-replica-prepare	Создание файла реплики FreeIPA
ipa-restore	Восстановление мастер-сервера FreeIPA
ipa-rmkeytab	Удаление принципала Kerberos из keytab-файла
ipa-server-certinstall	Установка новых SSL-сертификатов сервера
ipa-winsync-migrate	Полный переход от пользователей AD, созданных winsync, к обычным пользователям AD
ipa-upgradeconfig	Обновление конфигурации Apache FreeIPA

8.3.3. Установка и удаление

Программные компоненты FreeIPA входят в состав ОС и могут быть установлены с помощью стандартной графической утилиты для работы с пакетами Synaptic либо из терминала.

ВНИМАНИЕ! Без установки пакетов расширения совместно с соответствующими основными пакетами невозможна централизация хранения атрибутов СЗИ в распределенной сетевой среде, что может привести к невозможности входа пользователей в систему.

Для развертывания FreeIPA необходимо:

1) на компьютере, предназначенном на роль контроллера домена, установить следующие программные компоненты:

   а) astra-freeipa-server, для установки из терминала ввести команду:

   ```
   apt install astra-freeipa-server
   ```

   б) fly-admin-freeipa-server, для установки из терминала ввести команду:

   ```
   apt install fly-admin-freeipa-server
   ```

   При установке графической утилиты fly-admin-freeipa-server автоматически будет установлен инструмент командной строки astra-freeipa-server;

2) на клиентских компьютерах установить следующие программные компоненты:

   а) astra-freeipa-client, для установки из терминала ввести команду:

   ```
   apt install astra-freeipa-client
   ```

   б) fly-admin-freeipa-client, для установки из терминала ввести команду:

   ```
   apt install fly-admin-freeipa-client
   ```

   При установке графической утилиты fly-admin-freeipa-client автоматически будет установлен инструмент командной строки astra-freeipa-client.

При установке данных компонентов обеспечивается установка всех необходимых пакетов в зависимости от назначения компьютера.

ВНИМАНИЕ! Для создания ЕПП FreeIPA, в которое должны быть интегрированы клиенты, поддерживающие режимы мандатного управления доступом и/или мандатного контроля целостности, необходимо использовать сервер FreeIPA с включенными соответствующими режимами. После установки сервера FreeIPA изменение его режимов работы мандатного управления доступом и мандатного контроля целостности не поддерживается.

Для удаления контроллера домена с помощью инструмента командной строки `astra-freeipa-server` используется команда:

```
astra-freeipa-server -U
```

8.3.4. Настройка контроллера домена

При развертывании FreeIPA в качестве контроллера домена следует использовать отдельный компьютер с фиксированным IP-адресом, который в дальнейшем не должен изменяться.

ВНИМАНИЕ! Работа FreeIPA осуществляется только при выключенном режиме `AstraMode` web-сервера Apache2. Описание режима приведено в 10.2. Программы установки `astra-freeipa-server` и `fly-admin-freeipa-server` автоматически выключают данный режим.

Настройка всех компонентов FreeIPA осуществляется автоматически утилитами конфигурирования `astra-freeipa-server` и `fly-admin-freeipa-server`. Для нормального функционирования FreeIPA необходимо выполнение следующих условий:

1) использовать доменное имя второго уровня и ниже, например, `domain.net`, `testdomain.test.lan`;

2) разрешение имен должно быть настроено таким образом, чтобы имя системы разрешалось, в первую очередь, как полное имя, например, `myserver.example.ru`. Утилита `hostname` должна возвращать полное имя компьютера, например, `myserver.example.ru`.

Пример

```
/etc/hosts
127.0.0.1       localhost
192.168.1.1     myserver.example.ru myserver
```

Разрешение имен также может быть настроено с помощью сервера DNS в соответствии с 6.5;

3) должна быть выполнена синхронизация времени в ОС серверов и клиентов FreeIPA для аутентификации по Kerberos. Например, с использованием сервера NTP в соответствии с 6.7.

8.3.5. Запуск службы FreeIPA

8.3.5.1. Запуск с использованием графической утилиты

Для запуска службы FreeIPA на контроллере домена с помощью графической утилиты `fly-admin-freeipa-server` необходимо из терминала запустить графическую утилиту командой:

```
fly-admin-freeipa-server
```

и затем в открывшейся форме указать следующие данные:

- в поле «Домен» — имя домена;
- в поле «Имя компьютера» — имя компьютера, определяется автоматически;
- в поле «Пароль» — пароль администратора домена. Указанный пароль будет использоваться для входа в web-интерфейс FreeIPA и при работе с инструментом командной строки.

Далее запуск службы FreeIPA осуществляется нажатием кнопки **[Создать]**. После успешного запуска появится web-ссылка для перехода в web-интерфейс FreeIPA. Теперь можно войти в web-интерфейс и продолжить настройку через него. Порядок работы с FreeIPA используя web-интерфейс приведен в 8.3.13.

8.3.5.2. Запуск с использованием инструмента командной строки

Для запуска службы FreeIPA на контроллере домена с помощью инструмента командной строки `astra-freeipa-server` выполнить команду:

```
astra-freeipa-server -d <имя_домена> -n <имя_компьютера> -o
```

После выполнения команды будет определен адрес компьютера и будут выведены на экран все исходные данные.

Пример

```
compname= astraipa
domain= astradomain.ad
будет использован ip address = 192.168.32.97 или укажите ip адрес ключем -ip
продолжать ? (y\n)
```

Для подтверждения данных ввести `y` и нажать **<Enter>**. После подтверждения появится запрос на установку пароля администратора домена. Указанный пароль будет использоваться для входа в web-интерфейс FreeIPA и при работе с инструментом командной строки.

Параметры инструмента командной строки `astra-freeipa-server` приведены в таблице 47.

Таблица 47

Параметр	Описание
`-h, --help`	Вывести справку по командам
`-d`	Задать имя домена
`-n`	Задать имя компьютера
`-ip`	Задать IP-адрес web-интерфейса. Если адрес не задан, то инструмент пытается определить его автоматически
`-y`	Отключить запрос подтверждения после вывода заданных параметров запуска
`-i`	Вывести информацию о существующем домене
`-px`	Получить пароль администратора домена из `stdin`
`-p`	Получить пароль администратора домена из командной строки (небезопасно)
`-s`	Включить установку и запуск поддержки AD SMB
`-c`	Запретить изменять файл `/etc/hosts`
`-o`	Запретить проверку регистрации домена. Применяется при установке в изолированной сети
`-e`	Отключить установку и запуск собственной службы DNS
`-U`	Удалить все настройки
`-l`	Указать сертификат (имя компьютера и домена должны совпадать)
`-lp`	Указать пароль сертификата

После ввода пароля автоматически будет выполнен процесс инициализации входящих в FreeIPA подсистем, ход выполнения которого будет отображаться на экране. После успешного завершения инициализации на экран будут выведены сообщения о перезапуске системных служб, а также данные контроллера домена и ссылка для web-интерфейса.

Пример

```
Restarting Directory Service
Restarting krb5kdc Service
Restarting kadmin Service
Restarting named Service
Restarting ipa_memcached Service
Restarting httpd Service
Restarting ipa-custodia Service
Restarting ipa-otpd Service
Restarting ipa-dnskeysyncd Service
Starting ntpd Service
ipa: INFO: The ipactl command was successful
Существует настроенный домен
host = astraipa.astradomain.ad
```

```
basedn = dc=astradomain,dc=ad
domain = astradomain.ad
xmlrpc_uri = https://astraipa.astradomain.ad/ipa/xml
WEB: https://astraipa.astradomain.ad
```

После завершения работы мастера требуется убедиться в наличии открытых портов на сервере:

1) TCP Ports:
   - 80, 443: HTTP/HTTPS;
   - 389, 636: LDAP/LDAPS;
   - 88, 464: kerberos;
   - 53: bind;
2) UDP Ports:
   - 88, 464: kerberos;
   - 53: bind;
   - 123: ntp.

Настройки сервера FreeIPA содержатся в конфигурационном файле `/etc/ipa/default.conf`. Формат файла:

```
имя_параметра=значение # Комментарий
```

Описание параметров конфигурационного файла приведено в таблице 48.

Таблица 48

Параметр	Описание
`basedn <запись_DN>`	Задает базовую запись DN, используемую при выполнении операций LDAP. Запись должна быть в формате DN (например, `dc=example,dc=com`)
`context <контекст>`	Задает контекст, в котором выполняется IPA. Работа IPA определяется в зависимости от контекста. Текущие определенные контексты — `cli` и `server` (клиент и сервер). Кроме того, значение используется для загрузки файла `/etc/ipa/<контекст>.conf` для применения контекстной конфигурации. Например, если необходимо всегда выполнять клиентские запросы в подробном режиме, но при этом не использовать подробный режим на сервере, то следует добавить параметр `verbose` в `/etc/ipa/cli.conf`
`debug <boolean>`	При значении `True` предоставляет подробную информацию. В частности, значение `debug` устанавливается для глобального уровня log-журнала. Значение по умолчанию `False`
`domain <имя_домена>`	Домен сервера FreeIPA, например, `example.com`
`enable_ra <boolean>`	Значение `True` определяет, что будет использоваться удаленная служба удостоверяющего центра, например, когда служба Dogtag используется в качестве удостоверяющего центра. Эта настройка применяется исключительно в конфигурации сервера IPA

Продолжение таблицы 48

Параметр	Описание
`fallback <boolean>`	Значение `True` определяет, что клиент IPA должен выполнять возврат и обращаться к другим службам в случае сбоя первого подключения
`host <имя_хоста>`	Задает имя хоста локальной системы
`in_server <boolean>`	Определяет, будут ли запросы направляться на сервер IPA (`True`) или обрабатываться локально (`False`). Внутри IPA они используются подобно контексту. Та же самая IPA-конструкция используется IPA-инструментами командной строки и сервера. Этот параметр указывает конструкции, выполнить ли команду так, как если бы она была на сервере или переслать ее через XML-RPC на удаленный сервер
`in_tree <boolean>`	Используется при разработке. Параметр указывается при необходимости выполнить код в исходном дереве
`interactive <boolean>`	Определяет, следует ли запрашивать значения. Значение по умолчанию `True`
`ldap_uri <URI>`	Указывает URI сервера IPA LDAP для подключения. Схема URI может быть ldap или ldapi. По умолчанию используется ldapi, например, `ldapi://%2fvar%2frun%2fslapd-EXAMPLE-COM.socket`

`log_logger_XXX <регулярное_выражение, ...>`	Перечень регулярных выражений `regexp`, разделенных запятыми. Логированиям (loggers), соответствующим `regexp`, будет присвоен уровень XXX.
	Уровни логирования (logger levels) могут быть явно заданы для конкретных логирований в отличие от глобального уровня журналирования (global logging level). Конкретные логирования обозначаются списком регулярных выражений, привязанных к уровню. Если имя логирования соответствует регулярному выражению, то ему присваивается соответствующий уровень. Этот элемент конфигурации должен начинаться с `log_logger_level_`, а затем должен следовать символический или числовой уровень журнала (log level), например:
	`log_logger_level_debug = ipalib\.dn\..*`
	`log_logger_level_35 = ipalib\.plugins\.dogtag`
	В первой строке сказано, что любое логирование, относящееся к модулю `ipalib.dn`, будет иметь свой уровень, настроенный для отладки.
	Во второй строке сказано, что логирование `ipa.plugins.dogtag` будет настроена на уровень 35.
	Этот элемент конфигурации полезен, если требуется просмотреть вывод журнала только для одного или нескольких выбранных логирований. Включение флага глобальной отладки приведет к огромному количеству вывода. Настройка позволяет отключить глобальный флаг отладки и выборочно включить для конкретного логирования. Обычно логирования привязаны к классам и модулям.
	Примечание. Имена логирований (logger names) — список с разделяющей точкой, образующий путь в данном дереве логирования (logger tree). Символ точки также является метасимволом регулярного выражения (соответствует любому символу), поэтому, чтобы избежать точек в именах логирования, обычно требуется перед ними ставить обратную косую черту «\.».

Продолжение таблицы 48

Параметр	Описание
mode <режим_работы>	Определяет режим работы сервера. В настоящее время поддерживаемыми значениями являются эксплуатация (production) и разработка (development). При работе в режиме production некоторые самопроверки пропускаются для повышения производительности
mount_ipa <URI>	Задает точку монтирования для регистрации сервера разработки. По умолчанию /ipa/
prompt_all <boolean>	Определяет, должны ли для клиента IPA запрашиваться все параметры, в т.ч. необязательные значения. По умолчанию устанавливается False
ra_plugin <имя>	Задает имя назначенного для использования CA. Текущими параметрами являются dogtag и selfsign. Настройка на стороне сервера. Изменять значение не рекомендуется, т.к. назначенный CA настраивается только во время первоначальной установки
realm <realm>	Указывает область Kerberos
session_auth_duration <интервал_времени>	Задает допустимый интервал для времени кэширования учетных данных проверки подлинности в сеансе. По истечении срока действия учетные данные будут автоматически переопределены. Например, 2 hours, 1h:30m, 10 minutes, 5min, 30sec
session_duration_type <тип_вычисления>	Определяет способ вычисления срока действия сеанса. Возможные значения: - inactivity_timeout — срок действия увеличивается на значение session_auth_duration каждый раз, когда пользователь обращается к службе; - from_start сроком действия сеанса является начало сеанса пользователя плюс значение session_auth_duration
server <имя_сервера>	Задает имя сервера IPA
skip_version_check <boolean>	Пропустить проверки версии API клиента и сервера. Может привести к ошибкам/сбоям, когда новые клиенты обращаются к прежним серверам. Использовать с осторожностью
startup_timeout <время_ожидания>	Определяет время ожидания в секундах до начала запуска сервера. Значение по умолчанию 120 секунд
startup_traceback <boolean>	Если сервер IPA не запускается при заданном значении True, то сервер будет пытаться сгенерировать обратное python-отслеживание, чтобы облегчить определение причины сбоя
validate_api <boolean>	Используется внутри исходного пакета IPA для проверки неизменности API. Применяется для предотвращения регрессии. Если установлено значение True, то некоторые ошибки игнорируются, чтобы обеспечить загрузку инфраструктуры IPA, достаточной для проверки API, даже если дополнительные компоненты не установлены. Значение по умолчанию False
verbose <boolean>	При установке значения True предоставляет дополнительные сведения — устанавливает глобальный уровень журнала (global log level) на событие info

Окончание таблицы 48

Параметр	Описание
wait_for_dns <boolean>	Контролирует синхронность работы IPA команд dnsrecord-{add,mod,del}. Команды DNS будут повторять DNS-запросы указанное количество попыток до тех пор, пока DNS-сервер возвращает ответ up-to-date на запрос об измененных записях. Задержка между повторными попытками одна секунда. Команды DNS будут порождать исключение DNSDataMismatch, если ответ не совпадает с ожидаемым значением, даже после указанного числа попыток. DNS-запросы будут отправлены в очередь для разрешения решателем, который сконфигурирован в файле /etc/resolv.conf на сервере IPA. **ВНИМАНИЕ!** Не включать параметр в режиме production! Это может вызвать проблемы, если решатель (resolver) на сервере IPA использует кэширование сервера, а не локального сервера авторизации или, например, если DNS-ответы будут изменены шлюзом DNS64. Значение по умолчанию disable (выключено), параметр отсутствует
xmlrpc_uri <URI>	Задает URI сервера XML-RPC для клиента. Может использоваться IPA и используется некоторыми внешними средствами, такими как ipa-getcert. Например, https://ipa.example.com/ipa/xml
jsonrpc_uri <URI>	Задает URI сервера JSON для клиента. Используется IPA. Если параметр не задан, он наследуется от xmlrpc_uri. Например, https://ipa.example.com/ipa/json
rpc_protocol <URI>	Задает тип RPC-вызовов IPA makes: jsonrpc или xmlrpc. По умолчанию используется jsonrpc

Более подробное описание конфигурационного файла приведено в руководстве man.

Пример

Конфигурационный файл /etc/ipa/default.conf

```
[global]
host = server.example.ru
basedn = dc=example,dc=ru
realm = EXAMPLE.RU
domain = example.ru
xmlrpc_uri = https://server.example.ru/ipa/xml
ldap_uri = ldapi://%2fvar%2frun%2fslapd-EXAMPLE-RU.socket
enable_ra = False
ra_plugin = none
mode = production
```

Для дальнейшего конфигурирования и администрирования FreeIPA следует использовать web-интерфейс. Порядок работы с FreeIPA с использованием web-интерфейса приведен в 8.3.13.

8.3.5.3. Управление службами FreeIPA

Для проверки работы и управления службами FreeIPA используется команда `ipactl`:

- запуск служб FreeIPA:

```
ipactl start
```

- отображение текущего состояния всех служб FreeIPA:

```
ipactl status
```

- перезапуск служб FreeIPA:

```
ipactl restart
```

- остановка служб FreeIPA:

```
ipactl stop
```

Дополнительно с командой `ipactl` можно использовать параметр `-d` для выполнения команды в режиме отладки:

```
ipactl start -d
```

8.3.6. Ввод компьютера в домен

8.3.6.1. Настройка клиентского компьютера

Для ввода нового компьютера в домен необходимо:

1) наличие установленного пакета `astra-freeipa-client`;

2) клиентский компьютер и сервер FreeIPA должны видеть друг друга в сети. Для проверки можно использовать команду:

```
ping <ip-адрес>
```

3) клиентский компьютер не должен входить в другой домен (в частности, в домен ALD);

4) разрешение имен должно быть настроено таким образом, чтобы имя системы разрешалось, в первую очередь, как полное имя, например, `myclient.example.ru`;

5) утилита `hostname` должна возвращать полное имя компьютера, например, `myclient.example.ru`.

Пример

```
/etc/hosts
127.0.0.1 localhost
192.168.1.2 myclient.example.ru myclient
192.168.1.1 myserver.example.ru myserver
```

Разрешение имен также может быть настроено с помощью сервера DNS в соответствии с 6.5.

Далее необходимо настроить DNS-адрес сервера FreeIPA на клиентском компьютере одним из способов:

1) указать в конфигурационном файле `resolv.conf`;
2) указать в файле `interfaces`;
3) используя утилиту NetworkManager.

ВНИМАНИЕ! В некоторых случаях, если адрес сервера FreeIPA стоит в DNS не первым, клиентский компьютер может не находить домен.

Ввод компьютера в домен можно выполнить с помощью инструмента командной строки или графической утилиты.

8.3.6.2. Ввод компьютера в домен с использованием инструмента командной строки

Для ввода компьютера в домен с использованием инструмента командной строки `astra-freeipa-client` необходимо выполнить команду:

```
sudo astra-freeipa-client -d <контроллер_домена> -u admin -px
```

Для просмотра перечня дополнительных параметров для запуска с командой `astra-freeipa-client` выполнить:

```
astra-freeipa-client --help
```

8.3.6.3. Ввод компьютера в домен с использованием графической утилиты

Для ввода компьютера в домен с использованием графической утилиты `fly-admin-freeipa-client` следует запустить утилиту из командной строки или через меню «Пуск — Панель управления — Сеть — Настройка FreeIPA клиент Fly».

В открывшемся окне, приведенном на рис. 2, следует ввести:

1) в поле «Домен» — имя домена;
2) в поле «Логин» — имя администратора домена;
3) в поле «Пароль» — пароль администратора домена.

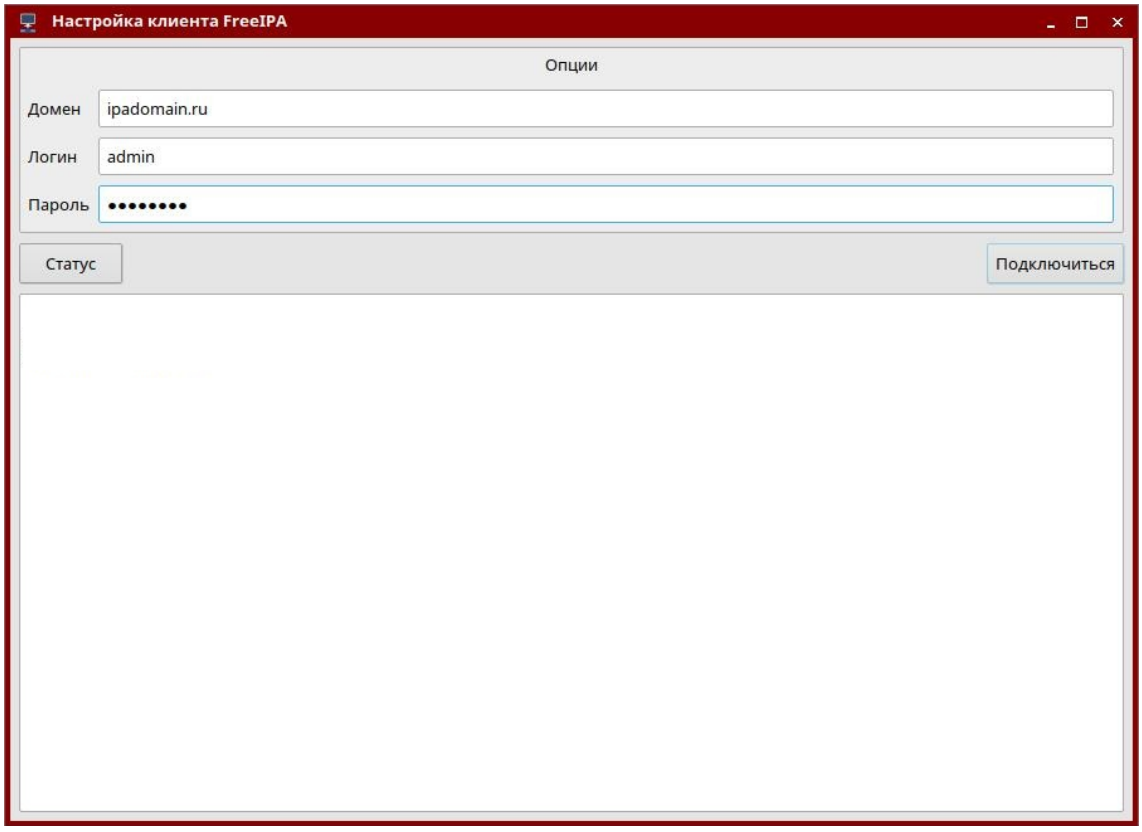

Рис. 2

После ввода данных следует нажать кнопку **[Подключиться]**.

8.3.6.4. Отображение списка доменных учетных записей в окне входа в ОС

По умолчанию список доменных учетных записей не отображается в окне входа в ОС, в том числе даже если выполнена соответствующая настройка в графической утилите `fly-admin-dm` — включено отображение списка пользователей и настроен диапазон отображаемых пользователей таким образом, чтобы системные идентификаторы пользователей (`uid`) домена FreeIPA попадали в него. Описание графической утилиты `fly-admin-dm` см. в электронной справке.

Для включения отображения списка доменных пользователей, дополнительно к настройкам с помощью графической утилиты `fly-admin-dm`, необходимо откорректировать конфигурационный файл `/etc/sssd/sssd.conf`, изменив в секции `[domain]` значение параметра `enumerate` на `TRUE` или добавив параметр, если он отсутствует:

```
[domain]
enumerate = True
```

При включении отображения списка доменных пользователей в окне входа в ОС рекомендуется ограничивать выводимый список путем задания соответствующего диапазона в графической утилите `fly-admin-dm`, т.к. вывод большого списка пользователей может снизить производительность.

8.3.7. Шаблоны конфигурационных файлов

Служба FreeIPA в процессе работы осуществляет конфигурирование сетевых служб (Samba, Kerberos, LDAP и т.п.) с помощью их конфигурационных файлов. Для удобства существуют шаблоны конфигурационных файлов, модифицируемых службой FreeIPA. Шаблоны расположены в каталогах `/usr/share/ipa` и `/usr/share/ipa/advise/legacy/`.

Перечень шаблонов конфигурационных файлов приведен в таблице 49.

Таблица 49

Имя шаблона	Служба	Описание, размещение конфигурационного файла
`*.ldif`	389-BASE	LDAP схемы
`default.conf`	IPA	`/etc/ipa/default.conf`
`ipa-httpd.conf.template`	IPA	`/etc/systemd/system/apache2.service.d/ipa.conf`
`ipa-kdc-proxy.conf.template`	IPA	`/etc/ipa/kdcproxy/ipa-kdc-proxy.conf`
`sssd.conf.template`	SSSD	`/etc/sssd/sssd.conf`
`ldap.conf`	LDAP клиенты	`/etc/ldap/ldap.conf`
`krb5.conf.template`	Kerberos клиенты	`/etc/krb5.conf`
`kdc.conf.template`	Kerberos KDC	`/etc/krb5kdc/kdc.conf`
`certmap.conf.template`	389-BASE	`/etc/dirsrv/config/certmap.conf`
`bind.named.conf.template`	BIND9	`/etc/bind/named.conf`
`custodia.conf.template`	IPA	`/etc/ipa/custodia/custodia.conf`
`smb.conf.template`	Samba	`/etc/samba/smb.conf`
`opendnssec_conf.template`	Opendnssec	`/etc/opendnssec/conf.xml`
`pam.conf.sssd.template`	SSSD	`/etc/pam.d/`
`mldap.conf`	PARSEC	`/etc/parsec/mldap.conf`
`mswitch.conf`	PARSEC	`/etc/parsec/mswitch.conf`
`krb.con.template`	IPA	`/usr/share/ipa/html`
`krbrealm.con.template`	IPA	`/usr/share/ipa/html`
`krb5.ini.template`	IPA	`/usr/share/ipa/html`

8.3.8. Администрирование домена

8.3.8.1. Создание резервной копии и восстановление

Поддерживается создание резервных копий двух типов: полная резервная копия всей системы и резервная копия только данных. Установка пароля на резервные копии не поддерживается.

Резервные копии хранятся в каталоге `/var/lib/ipa/backup`. Для полного резервного копирования и резервного копирования данных используются, соответственно, обозначения `ipa-full-YEAR-MM-DD-HH-MM-SS` и `ipa-data-YEAR-MM-DD-HH-MM-SS`, где `YEAR-MM-DD-HH-MM-SS` — год, месяц, день, час, минуты и секунды в часовом поясе GMT создания резервной копии, например, 2018-03-05-10-30-22.

В каталоге `/var/lib/ipa/backup` размещается файл, в котором приведена информация о резервных копиях: тип, система, даты резервного копирования, версия FreeIPA, версия резервного копирования и др.

ВНИМАНИЕ! Резервную копию невозможно восстановить на другом компьютере или на другой версии FreeIPA.

Резервное копирование выполняется с помощью команды `ipa-backup`. Дополнительно с командой возможно использовать параметры, приведенные в таблице 50.

Таблица 50

Параметр	Описание
`--data`	Резервное копирование только данных. По умолчанию выполняется резервное копирование всех FreeIPA-файлов и данных
`--logs`	Включить в резервную копию лог-файлы службы FreeIPA
`--online`	Выполнить резервное копирование без остановки сервера. Требуется использования параметра `--data`
`-v`, `--verbose`	Выводить сведения об отладке
`-d`, `--debug`	Используется с параметром `--verbose` для вывода более детальных сведений об отладке
`-q`, `--quiet`	Выводить только сведения о ошибках
`--log-file=FILE`	Выполнить журналировние в файл `FILE`

8.3.8.2. Создание резервного сервера FreeIPA

Новый сервер FreeIPA возможно настроить на выполнение роли резервного сервера (реплики). Созданная реплика будет являться точной копией исходного сервера FreeIPA и приравниваться к мастер-серверу. Изменения, внесенные в любой мастер-сервер, автоматически реплицируются на другие мастер-сервера.

Для добавления реплики в домен FreeIPA необходимо выполнить следующие действия:

1) на реплике назначить фиксированный IP-адрес, который впоследствии не должен изменяться, и зарегистрировать реплику в качестве клиента в домене FreeIPA в соответствии с 8.3.6;

2) на реплике установить программный компонент `astra-freeipa-server` в соответствии с 8.3.3;

3) на реплике запустить службу SSH, выполнив команду:

```
sudo systemctl enable --now ssh
```

4) на основном сервере домена с использованием инструмента `astra-freeipa-server-crt` выпустить сертификат для реплики с последующим переносом сертификата в домашний каталог администратора реплики:

```
astra-freeipa-server-crt --host <реплика> --export --push
    <администратор>@<IP-адрес> --pin <пароль> --48
```

где `<реплика>` — полное доменное имя реплики;

`<администратор>` — имя администратора реплики;

`<IP-адрес>` — IP-адрес реплики;

`<пароль>` — пароль к создаваемому контейнеру закрытого ключа и сертификата;

– `--48` — указание создать сертификат для FreeIPA версии 4.8.x (по умолчанию будут создаваться сертификаты для FreeIPA версии 4.6.x) .

Во время выпуска сертификата на все вопросы ответить «y» («Да»), и затем вести пароль администратора реплики;

5) на реплике из домашнего каталога администратора, в который ранее был скопирован контейнер закрытого ключа и сертификата, выполнить команду:

```
astra-freeipa-replica -a <реплика>.p12 --pin <пароль>
```

где `<реплика>` — полное доменное имя реплики (в таком формате задается имя файла контейнера закрытого ключа и сертификата);

`<пароль>` — пароль к созданному контейнеру закрытого ключа и сертификата.

В ходе выполнения команды необходимо вести пароль администратора домена, а затем на все вопросы ответить «y» («Да»).

В случае успешной активации реплика должна появиться на топологической схеме в web-интерфейсе FreeIPA («IPA-сервер — Топология — Topology Graph», см. рис. 3).

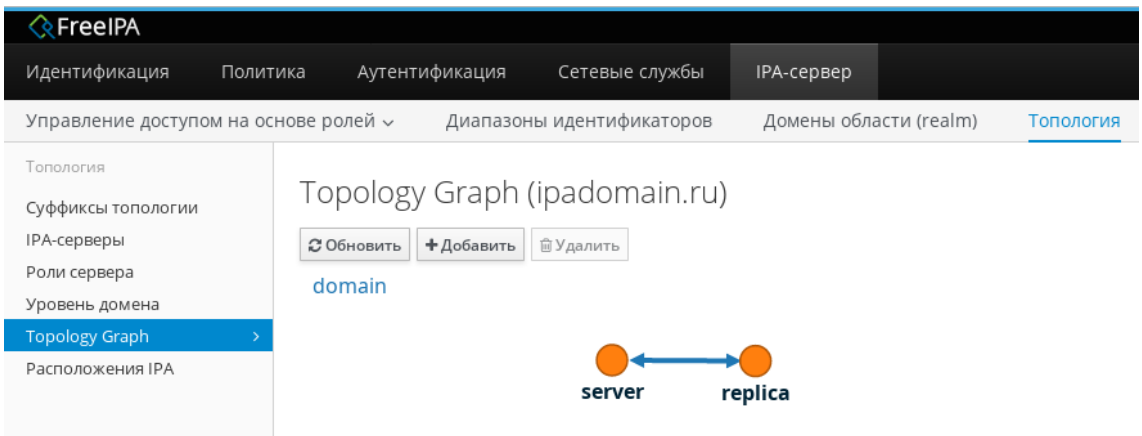

Рис. 3

8.3.9. Доверительные отношения между доменами

8.3.9.1. Общие сведения

Перед настройкой доверительных отношений контроллер домена AD должен быть настроен и работоспособен, а службы FreeIPA запущены в соответствии с 8.3.5.

ВНИМАНИЕ! Не удастся установить доверительные отношения с доменом AD, если имя области сервера FreeIPA не совпадает с его доменным именем.

Для создания доверительных отношений сервера FreeIPA с доменом AD служит пакет `freeipa-server-trust-ad`. Установка службы доверительных отношений выполняется с помощью инструмента командной строки `ipa-adtrust-install`.

В случае необходимости переустановки ранее удаленных объектов или поврежденных файлов конфигурации команду `ipa-adtrust-install` можно запустить несколько раз. Таким образом могут быть созданы новая конфигурация Samba (файл `smb.conf`) и конфигурация, на которой базируется регистрация. Некоторые элементы, например, конфигурация локального диапазона, не могут быть изменены в результате повторного запуска команды `ipa-adtrust-install`, т. к. в данном случае изменения могут затронуть и другие объекты.

К брандмауэру сервера FreeIPA дополнительно предъявляются требования разрешить домену FreeIPA и домену AD обмениваться информацией, т. е. при выполнении команды `ipa-adtrust-install` предполагается, что следующие порты открыты:

- 135/tcp EPMAP
- 138/tcp NetBIOS-DGM
- 139/tcp NetBIOS-SSN
- 445/tcp Microsoft-DS
- 1024/tcp
- 3268/tcp Microsoft-GC
- 138/udp NetBIOS-DGM
- 139/udp NetBIOS-SSN
- 389/udp LDAP

Дополнительно с командой `ipa-adtrust-install` возможно использовать параметры, приведенные в таблице 51.

Таблица 51

Параметр	Описание
`-d, --debug`	Выводить детальные сведения об отладке

Продолжение таблицы 51

Параметр	Описание
`--netbios-name=NETBIOS_NAME`	Задать имя NetBIOS для домена FreeIPA. Если не указано, то оно определяется на основе ведущей компоненты DNS-имени домена. Если запустить команду `ipa-adtrust-install` во второй раз с другим именем NetBIOS, то это имя изменится. **ВНИМАНИЕ!** Изменение имени NetBIOS может нарушить существующие доверительные отношения с другими доменами
`--add-sids`	Добавить SIDs для существующих пользователей и групп как активные на заключительных шагах запуска команды `ipa-adtrust-install`. Если в среде существует множество действующих пользователей и групп и несколько реплик, то выполнение данного действия может привести к высокой скорости репликации трафика и снижению производительности всех серверов FreeIPA в среде. Чтобы избежать этого рекомендуется генерацию SIDs запускать после выполнения команды `ipa-adtrust-install`, для этого загрузить отредактированную версию `ipa-sidgen-task-run.ldif` с помощью команды `ldapmodify` на сервере домена AD
`--add-agents`	Добавить мастер-сервер FreeIPA в список, что позволяет предоставлять информацию о пользователях доверенных лесов. Обычный мастер-сервер FreeIPA может предоставлять эту информацию клиентам SSSD. Мастер-серверы FreeIPA не добавляются в список автоматически, т. к. для этого требуется перезапуск службы LDAP на каждом из них. Компьютер, на котором выполнена команда `ipa-adtrust-install`, добавляется автоматически. **ВНИМАНИЕ!** Мастер-серверы FreeIPA, на которых команда `ipa-adtrust-install` не была запущена, могут работать с информацией о пользователях доверенных лесов только если они активированы путем выполнения команды `ipa-adtrust-install` на любом другом мастер-сервере FreeIPA
`-U`, `--unattended`	Удалить без подтверждения. Ввод данных пользователем не будет запрашиваться
`--rid-base=RID_BASE`	Задать первое значение RID локального домена. Первый Posix ID локального домена будет присвоен данному RID, второй будет присвоен RID+1 и т. д.

Окончание таблицы 51

Параметр	Описание
`--secondary-rid-base=SECONDARY_RID_BASE`	Задать начальное значение вторичного RID диапазона, которое используется только в том случае, если пользователь и группа используют один и тот же Posix ID
`-A, --admin-name=ADMIN_NAME`	Задать имя пользователя с правами администратора для данного сервера FreeIPA. По умолчанию `admin`
`-a, --admin-password=password`	Задать пароль для пользователя с правами администратора для данного сервера FreeIPA. Будет запрашиваться в интерактивном режиме если параметр `-U` не указан. Учетные данные администратора будут использованы для получения билета Kerberos перед настройкой поддержки доверительные отношения перекрестной области, а также в дальнейшем, чтобы убедиться, что билет содержит MS-PAC сведения, необходимые для фактического добавления отношений доверия с доменом AD при помощи команды `ipa trust-add -type=ad`

8.3.9.2. Предварительная настройка

Серверы домена AD и домена FreeIPA должны находиться в одной сети и на обоих серверах должна успешно выполняться команда:

```
ping <IP-адрес>
```

где `<IP-адрес>` — IP-адрес сервера домена AD при выполнении команды на сервере домена FreeIPA или IP-адрес сервера домена FreeIPA при выполнении команды на сервере домена AD.

8.3.9.3. Настройка синхронизация времени

При установке и инициализации FreeIPA конфигурация службы синхронизации времени настраивается автоматически для использования общедоступных серверов точного времени.

При развертывании FreeIPA в сети без доступа к общедоступным серверам точного времени необходимо исправить настройки службы синхронизации времени в файле `/etc/ntp.conf`, выполнив команды:

```
sudo sed -i -e "s/^\([[:space:]]*.*debian\.pool\.ntp\.org.*\)/#&/" /etc/ntp.conf
echo server <IP-адрес> | sudo tee -a /etc/ntp.conf > /dev/null
```

где `<IP-адрес>` — IP-адрес сервера времени

Затем выполнить процедуру перезапуска автоматической синхронизации времени командами:

```
sudo service ntp stop
```

```
sudo ntpdate -bv <IP-адрес>
sudo service ntp start
```

где <IP-адрес> — IP-адрес сервера времени

ВНИМАНИЕ! При использовании виртуальных машин процедура перезапуска автоматической синхронизации обязательно должна быть выполнена после каждого перезапуска и/или отката виртуальных машин.

8.3.9.4. Инициализация доверительных отношений

Для инициализации доверительных отношений необходимо на сервере домена FreeIPA выполнить следующие действия:

1) получить полномочия администратора домена и проверить работоспособность служб FreeIPA, выполнив команды:

```
kinit <администратор_домена_FreeIPA>
id <администратор_домена_FreeIPA>
getent passwd <администратор_домена_FreeIPA>
```

В результате выполнения команд не должны быть выявлены ошибки;

2) запустить службу доверительных отношений FreeIPA командой:

```
sudo ipa-adtrust-install
```

На все вопросы ответить «Да» («y») и затем вести пароль администратора домена FreeIPA. Проверить правильность автоматического определения имени домена и ответить «Да» («y»);

3) настроить и проверить перенаправление DNS. Добавление зоны перенаправления осуществляется командой:

```
ipa dnsforwardzone-add <домен_AD> --forwarder=WIN_IP ?forward-policy=only
```

Проверка успешного выполнения команды выполняется путем:

а) проверки доступности сервер домена AD:

```
ping -c 3 <сервер_домена_AD>.<домен_AD>
```

б) проверки доступности службы FreeIPA:

```
dig SRV _ldap._tcp.<домен_FreeIPA>
```

в) проверки доступности службы домена AD:

```
dig SRV _ldap._tcp.<домен_AD>
```

4) сохранить конфигурацию Samba, выполнив команду:

```
cp /etc/samba/smb.conf /etc/samba/smb.conf && sudo testparm | sudo tee
    /etc/samba/smb.conf > /dev/null
```

5) проверить работоспособность службы Samba командой:

```
smbclient -k -L <сервер_домена_FreeIPA>.<домен_FreeIPA>
```

6) установить доверительные отношения между доменами:

а) одностороннее доверительное отношение — одностороннее доверие к домену AD, при котором область FreeIPA доверяет лесу доменов AD, используя механизм доверительных отношений между деревьями доменов AD, но дерево доменов AD не доверяет области FreeIPA. Пользователи дерева доменов AD получают доступ к ресурсам области FreeIPA. Устанавливается командой:

```
ipa trust-add --type=ad <домен_AD> --admin <администратор_домена_AD>
     --password
```

б) двустороннее доверительное отношение устанавливается командой:

```
ipa trust-add --type=ad <домен_AD> --admin <администратор_домена_AD>
     --password --two-way=true
```

в) внешнее доверительное отношение — отношение доверия между доменами AD, находящимися в разных лесах доменов AD. Установление доверительных отношений между лесами доменов всегда требует установления доверительных отношений между корневыми доменами этих лесов, однако, внешнее доверительное отношение может быть установлено между любыми доменами в лесу. Применяется для установления доверительных отношений с конкретными доменами и не переходит границы доверенного домена. Устанавливается командой:

```
ipa trust-add --type=ad <домен_AD> --admin <администратор_домена_AD>
     --password --two-way=true --external
```

7) после установления доверительных отношений следует выполнить команду для получения списка доверенных доменов:

```
ipa trust-fetch-domains <домен_AD>
```

Домен должен быть найден при выполнении команды:

```
ipa trustdomain-find <домен_AD>
```

8) для работы пользователей домена AD в домене FreeIPA следует зарегистрировать данных пользователей, добавив соответствующие группы и пользователей в них:

```
ipa group-add --desc='ad domain external map' ad_admins_external
     --external
ipa group-add --desc='ad domain users' ad_admins
ipa group-add-member ad_admins_external --external '<домен_AD>\Domain
     Admins'
ipa group-add-member ad_admins --groups ad_admins_external
```

На запросы «member_user» и «member_group» нажать клавишу **<Enter>**.

9) для предоставления пользователям прав на доступ к разделяемым ресурсам требуется указать их идентификаторы безопасности.

Для получение идентификатора безопасности пользователей домена AD на сервере AD из оболочки CMD (но не из оболочки PowerShell) выполнить команду:

```
c:\> wmic useraccount get name,sid
```

Для получение идентификатора безопасности пользователей домена FreeIPA на сервере FreeIPA выполнить команду:

```
ipa group-show ad_admins_external --raw
```

Пример

Добавление разделяемого каталога `/share_dir`, доступного для пользователей домена AD под именем `share_name`:

```
sudo mkdir /share_dir
sudo net conf setparm 'share_name' 'comment' 'Trust test share'
sudo net conf setparm 'share_name' 'read only' 'no'
sudo net conf setparm 'share_name' 'valid users' "$d_admins_sid"
sudo net conf setparm 'share_name' 'path' '/share_dir'
```

Проверить, что ресурс добавлен, выполнив команду:

```
smbclient -k -L <сервер_домена_FreeIPA>.<домен_FreeIPA>
```

После добавления каталога при помощи Internet Explorer проверить, что ресурс доступен с сервера AD.

8.3.9.5. Проверка установки доверительных отношений

При успешной установке доверительных отношений пользователи домена AD должны получить возможность входа в систему с использованием своего имени и пароля:

- через терминал;
- через графический интерфейс;
- через SSH (если установлена соответствующая сетевая служба)

Также пользователям AD предоставляется возможность доступа к разделяемым ресурсам.

ВНИМАНИЕ! Для входа необходимо использовать полное имя пользователя с указанием домена, к которому пользователь относится, например, `Administrator@windomain.ad`, при это имя домена пишется строчными буквами, а имя пользователя с сохранением строчных и заглавных букв.

Проверка настройки DNS на сервере домена AD выполняется из командной строки. Для просмотра записей выполнить команду:

```
c:\>nslookup.exe
```

В выводе выполнения команды будут приведены записи о работе служб и служб домена:

1) записи, отвечающие за работу служб Kerberos через UDP и LDAP через TCP:

```
> set type=SRV
> _kerberos._udp.<домен_FreeIPA>.
```

```
_kerberos._udp.<домен_FreeIPA>.       SRV service location:
priority                = 0
weight                  = 100
port                    = 88
svr hostname   = <сервер_домена_FreeIPA>.<домен_FreeIPA>.
> _ldap._tcp.<домен_FreeIPA>.
_ldap._tcp.<домен_FreeIPA>       SRV service location:
priority                = 0
weight                  = 100
port                    = 389
svr hostname   = <сервер_домена_FreeIPA>.<домен_FreeIPA>.
```

2) записи, отвечающие за имя Kerberos realm домена FreeIPA:

```
> set type=TXT
_kerberos.<домен_FreeIPA>.
_kerberos.<домен_FreeIPA>.        Text =
        "<домен_FreeIPA>"
```

3) после выполнения команды `ipa-adtrust-install` должны появиться записи, отвечающие за работу служб MS DC Kerberos через UDP и LDAP через TCP:

```
> set type=SRV
> _kerberos._udp.dc._msdcs.<домен_FreeIPA>.
_kerberos._udp.dc._msdcs.<домен_FreeIPA>.        SRV service location:
priority                = 0
weight                  = 100
port                    = 88
svr hostname = <сервер_домена_FreeIPA>.<домен_FreeIPA>.
> _ldap._tcp.dc._msdcs.<домен_FreeIPA>.
_ldap._tcp.dc._msdcs.<домен_FreeIPA>.        SRV service location:
priority                = 0
weight                  = 100
port                    = 389
svr hostname = <сервер_домена_FreeIPA>.<домен_FreeIPA>.
```

Проверка наличия записей для работы служб AD на DNS-сервере AD выполняется из командной строки. Для просмотра записей выполнить команду:

```
c:\>nslookup.exe
```

Запись, отвечающая за работу служб Kerberos через UDP и LDAP через TCP:

```
> set type=SRV
> _kerberos._udp.dc._msdcs.<домен_AD>.
```

```
_kerberos._udp.dc._msdcs.<домен_AD>.  SRV service location:
priority = 0
weight = 100
port = 88
svr hostname = <сервер_домена_AD>.<домен_AD>.
> _ldap._tcp.dc._msdcs.<домен_AD>.
_ldap._tcp.dc._msdcs.<домен_AD>.  SRV service location:
priority = 0
weight = 100
port = 389
svr hostname = <сервер_домена_AD>.<домен_AD>.
```

Проверка настройки DNS на сервере домена FreeIPA и наличия записей для работы служб FreeIPA на DNS-сервере FreeIPA выполняется из командной строки.

Запись, отвечающая за работу служб Kerberos через UDP и LDAP через TCP:

```
# dig +short -t SRV _kerberos._udp.<домен_FreeIPA>.
0 100 88 <сервер_домена_FreeIPA>.<домен_FreeIPA>.
# dig +short -t SRV _ldap._tcp.<домен_FreeIPA>.
0 100 389 <сервер_домена_FreeIPA>.<домен_FreeIPA>.
```

Запись, отвечающая за имя Kerberos realm домена FreeIPA:

```
dig +short -t TXT _kerberos.<домен_FreeIPA>.
"<домен_FreeIPA>"
```

После выполнения команды `ipa-adtrust-install` должны появиться записи, отвечающие за работу служб MS DC Kerberos через UDP и LDAP через TCP:

```
# dig +short -t SRV _kerberos._udp.dc._msdcs.<домен_FreeIPA>.
0 100 88 <сервер_домена_FreeIPA>.<домен_FreeIPA>.

# dig +short -t SRV _ldap._tcp.dc._msdcs.<домен_FreeIPA>.
0 100 389 <сервер_домена_FreeIPA>.<домен_FreeIPA>.
```

Проверка наличия записей для работы служб AD на DNS-сервере FreeIPA выполняется из командной строки.

Записи, отвечающие за работу служб Kerberos через UDP и LDAP через TCP:

```
# dig +short -t SRV _kerberos._udp.dc._msdcs.<домен_AD>.
0 100 88 <сервер_домена_AD>.<домен_AD>.

# dig +short -t SRV _ldap._tcp.dc._msdcs.<домен_AD>.
0 100 389 <сервер_домена_AD>.<домен_AD>.
```

Если запись `_kerberos._udp.dc._msdcs.source-<домен_AD>.` недоступна, то необходимо проверить `_kerberos._tcp.dc._msdcs.source-<домен_AD>.`

8.3.10. Создание самоподписанного сертификата

8.3.10.1. Создание сертификата с помощью инструмента XCA

Установка и настройка инструмента XCA выполняется в соответствии с 6.10.5.1.

Для создания цепочки сертификатов необходимо запустить инструмент XCA и выполнить следующие действия:

1) создать корневой сертификат:

   а) во вкладке «Закрытые ключи» нажать кнопку **[Новый ключ]**. В открывшемся окне в поле «Внутреннее имя» указать имя «rootKey» и нажать **[Создать]**;

   б) перейти во вкладку «Сертификаты», нажать **[Новый сертификат]**;

   в) в открывшемся окне «Создать сертификат x509» перейти во вкладку «Субъект»:

      1) в поле «Внутреннее имя» указать имя сертификата «rootCA»;

      2) в поле «commonName» указать то же имя — «rootCA»;

      3) в блоке «Закрытый ключ» выбрать ранее созданный ключ «rootKey»;

   г) в окне «Создать сертификат x509» перейти во вкладку «Расширения»:

      1) в поле «Тип» выбрать «Центр Сертификации»;

      2) определить период действия сертификата, указав в блоке «Выбор периода» значение «10»;

      3) нажать кнопку **[Применить]**, затем нажать **[Да]**.

2) создать сертификат для сервера:

   а) в основном окне программы перейти во вкладку «Закрытые ключи» и нажать кнопку «Новый ключ»;

   б) в открывшемся окне в поле «Внутреннее имя» указать имя «serverKey» и нажать **[Создать]**;

   в) перейти во вкладку «Сертификаты», нажать **[Новый сертификат]**;

   г) в открывшемся окне «Создать сертификат x509» во вкладке «Первоисточник»:

      1) в блоке «Подписание» установить флаг «Использовать этот сертификат для подписи» и выбрать значение «rootCA» (имя корневого сертификата);

      2) в поле «Алгоритм подписи» указать «SHA 256»;

   д) в окне «Создать сертификат x509» перейти во вкладку «Субъект»:

      1) в поле «Внутреннее имя» указать FQDN сервера, для которого формируется сертификат, например, `dc01.example.ru`;

      2) в поле «commonName» также указать FQDN сервера, для которого формируется сертификат;

      3) в блоке «Закрытый ключ» выбрать ранее созданный ключ «serverKey»;

е) в окне «Создать сертификат x509» перейти во вкладку «Расширения»:

1) в поле «Тип» выбрать «Конечный субъект»;

2) определить период действия сертификата, указав в блоке «Выбор периода» значение «10»;

3) нажать кнопку **[Применить]**, затем нажать **[Да]**.

3) экспортировать сертификат сервера:

а) в основном окне программы перейти во вкладку «Сертификаты»;

б) выбрать требуемый сертификат сервера и нажать кнопку **[Экспорт]**;

в) в открывшемся окне указать имя файла контейнера сертификата и его расположение;

г) в блоке «Формат для экспорта» выбрать формат «PKCS12» и нажать кнопку **[Да]**;

д) задать пароль на экспортируемый контейнер и нажать кнопку **[Да]**.

На контроллере домена FreeIPA для указания контейнера с сертификатом выполнить команду `astra-freeipa-server` с параметрами `-l` и `-lp`:

```
astra-freeipa-server -l <путь_к_контейнеру> -lp <пароль_к_контейнеру>
```

Просмотреть перечень дополнительных параметров для запуска с командой `astra-freeipa-server` можно выполнив:

```
astra-freeipa-server --help
```

8.3.10.2. Создание сертификата с помощью инструмента командной строки

Инструмент командной строки `astra-freeipa-server-crt` автоматизирует выпуск сертификатов для серверов (реплик) FreeIPA и предназначен для автоматизации работы в системах, в которых не применяется DogTag, являющийся штатной системой управления сертификатами FreeIPA.

Установка инструмента командной строки `astra-freeipa-server-crt` выполняется автоматически при установке графической утилиты `fly-admin-freeipa-server` или инструмента командной строки `astra-freeipa-server` в соответствии с 8.3.3.

При инициализации домена FreeIPA в соответствии с 8.3.5 в каталоге `/etc/ssl/freeipa` первого контроллера домена автоматически создаются файлы, перечень которых приведен в таблице 52.

Таблица 52

Наименование, размещение файла	Описание
`/etc/ssl/freeipa/ca.key`	Закрытый ключ удостоверяющего центра
`/etc/ssl/freeipa/ca.crt`	Сертификат закрытого ключа удостоверяющего центра
`/etc/ssl/freeipa/server.key`	Закрытый ключ сервера
`/etc/ssl/freeipa/server.crt`	Сертификат закрытого ключа сервера

При первом запуске инструмента командной строки astra-freeipa-server-crt будет создан новый закрытый ключ сервера, который будет размещен в файле /etc/ssl/freeipa/<имя_компьютера>.<имя_домена>.key. Созданный закрытый ключ будет использоваться для выпуска и перевыпуска всех сертификатов.

ВНИМАНИЕ! Замена закрытых ключей посредством инструмента командной строки astra-freeipa-server-crt не поддерживается.

Кроме того, при запуске инструмента командной строки astra-freeipa-server-crt без указания параметров будет создан новый сертификат сервера. Выпущенный сертификат будет размещен в файле /etc/ssl/freeipa/<имя_компьютера>.<имя_домена>-<дата_время>.crt.

ВНИМАНИЕ! По умолчанию будут создаваться сертификаты для FreeIPA версии 4.6.x. Поэтому при запуске инструмента командной строки astra-freeipa-server-crt всегда необходимо указывать параметр --48 (создавать сертификаты для FreeIPA версии 4.8.x).

Параметры инструмента командной строки astra-freeipa-server приведены в таблице 53.

Таблица 53

Параметр	Описание
-h, --help	Вывести справку по инструменту командной строки
--certdir DIR	Задать имя каталога (DIR) для поиска ключа и сертификата удостоверяющего центра и для размещения создаваемых сертификатов. Если каталог не существует — он будет создан. Значение по умолчанию /etc/ssl/freeipa
--host FQDN	Указать полное доменное имя (FQDN) сервера, для которого выпускается сертификат. Если имя не задано — используется hostname текущего сервера
--cacrt FILE	Указать имя файла (FILE) с существующим сертификатом удостоверяющего центра. Значение по умолчанию /etc/ssl/freeipa/ca.crt
--cakey FILE	Указать имя файла (FILE) с существующим закрытым ключом удостоверяющего центра. Значение по умолчанию /etc/ssl/freeipa/ca.key
--sekey FILE	Указать имя файла (FILE) с закрытым ключом сервера. Если файл не существует — будет создан новый закрытый ключ. Если имя не задано — ключ будет размещен в файле с именем FQDN.key в каталоге для размещения сертификатов
--sekey_parm ALG	Указать через двоеточие алгоритм и длину закрытого ключа сервера (ALG). Значение по умолчанию rsa:2048
--secrt_days NUM	Указать в днях срок действия (NUM) выпускаемого сертификата. Значение по умолчанию 3650 дней

Окончание таблицы 53

Параметр	Описание
`--export`	Экспортировать сертификат в контейнер формата `pkcs12` для установки нового сервера (новой реплики) FreeIPA. Экспорт будет выполнен в файл с именем `FQDN-<дата_время>.p12` в каталоге для размещения сертификатов. Не требуется для обновления сертификата уже установленного сервера
`--pin PIN`	Пароль (`PIN`) для экспорта сертификата. Чтобы задать пустой пароль, указать пробел в кавычках `--pin " "`. Если пароль не задан — он будет запрошен в процессе выполнения команды
`--push ADMIN`	Попытаться скопировать через `ssh/scp` созданные файлы на сервер, указанный в параметре `--host`, и зарегистрировать их. В параметре `ADMIN` можно задать не только имя пользователя, но и адрес целевого сервера, например `admin@192.168.32.11`. Все действия будут выполняться от имени `ADMIN`. Все файлы будут копироваться в домашний каталог этого пользователя. Если выполнялся экспорт сертификата для нового сервера (новой реплики), то сертификат будет скопирован в файл с именем `FQDN.p12`. Если создавался новый сертификат для существующего сервера, то: - копия этого сертификата будет скопирована в файл с именем `FQDN.crt`; - будет сделана попытка зарегистрировать его в БД сертификатов `/etc/apache2/nssdb`. **ВНИМАНИЕ!** После регистрации в БД сертификатов нового сертификата службы FreeIPA должны быть перезапущены вручную
`--46`	Создать сертификаты для FreeIPA версии 4.6.x. Данный параметр используется по умолчанию
`--48`	Создать сертификаты для FreeIPA версии 4.8.x
`-y`	Выполнить действия без запроса подтверждения

Пример использования инструмента командной строки `astra-freeipa-server-crt` для создания реплики в домене FreeIPA представлен в 8.3.8.2.

В случае необходимости выпуска новых сертификатов, например при истечении срока действия, можно воспользоваться следующей командой:

```
astra-freeipa-server-ctr --host <имя_компьютера>.<имя_домена>
    --push <имя_локального_администратора>
```

8.3.11. Настройка web-сервера Apache2 для работы в домене FreeIPA

Настройка работы web-сервера Apache2 в домене FreeIPA осуществляется:

- для выполнения авторизации с использованием Kerberos;
- для обеспечения сквозной аутентификации приложений.

Для развертывания web-сервера Apache2 требуется установить пакет `apache2` на компьютере, предназначенной для выполнения роли web-сервера. Установка выполняется командой:

```
sudo apt install apache2
```

Для обеспечения совместной работы web-сервера Apache2 в домене FreeIPA требуется:

1) настроенный домен FreeIPA, например `ipadomain0.ru`, с настроенной службой разрешения имен (DNS);
2) отдельный компьютер для размещения web-сервера Apache2;
   а) web-сервер должен быть введен в домен FreeIPA в соответствии с 8.3.6;
   б) разрешение имен должно быть настроено таким образом, чтобы имя системы разрешалось как полное имя web-сервера (FQDN), например `web.ipadomain0.ru`;
   в) web-серверу должен быть назначен постоянный IP-адрес.

8.3.11.1. Настройка авторизации Kerberos

Для настройки авторизации Kerberos требуется дополнительно установить модуль авторизации Kerberos `libapache2-mod-auth-kerb`. Установка выполняется командой:

```
sudo apt install libapache2-mod-auth-kerb
```

Если в ОС установлен в соответствии с 10.3 модуль аутентификации через PAM web-сервера Apache2 `authnz_pam`, отключить его при помощи команды:

```
a2dismod authnz_pam
```

и активировать модуль web-сервера Apache2 `auth_kerb` при помощи команды:

```
a2enmod auth_kerb
```

Далее web-сервер необходимо зарегистрировать как доменную службу — либо с помощью web-интерфейса администратора FreeIPA, либо получив билет Kerberos администратора домена и выполнив команду `ipa service-add` (команду следует выполнить либо на контроллере домена, либо на web-сервере):

```
kinit admin
ipa service-add HTTP/web.ipadomain0.ru
```

где `web.ipadomain0.ru` — полное доменное имя компьютера, на котором будет развернута служба.

Затем на web-сервере выгрузить таблицу ключей для зарегистрированной службы:

```
sudo kinit admin
sudo ipa-getkeytab -p HTTP/web.ipadomain0.ru@IPADOMAIN0.RU
-k /etc/ipa/apache2.keytab
```

Параметр `-k` команды `ipa-getkeytab` задает имя файла, в который будет сохранена таблица ключей (`/etc/ipa/apache2.keytab`).

Для выгруженного файла с таблицей ключей задать права доступа, выполнив команды:

```
chown www-data /etc/ipa/apache2.keytab
chmod 600 /etc/ipa/apache2.keytab
```

Примечания:

1. Для получения ключей не требуется механизм `sudo` — достаточно билета Kerberos. Механизм `sudo` используется для записи таблицы ключей в каталог `/etc/ipa/`. При этом билет Kerberos также должен быть получен с помощью механизма `sudo`, так так полученный от имени обычного пользователя билет будет недействителен.
2. Команду получения таблицы ключей `ipa-getkeytab` можно выполнить на контроллере домена — в этом случае полученную таблицу ключей необходимо защитить от несанкционированного доступа и скопировать в соответствующий каталог на web-сервере.

Далее требуется на web-сервере создать конфигурационный файл аутентификации для областей, требующих авторизации, например файл `/etc/apache2/conf-available/kerberos-auth.conf` со следующими строками:

```
        <Directory /var/www/html/>
# тип авторизации
AuthType Kerberos
# Подсказка с информацией о ресурсе (выводится при запросе пароля)
AuthName "Astra Kerberos protected area"
# Имя области (realm) Kerberos
KrbAuthRealms IPADOMAIN0.RU
# Имя ранее зарегистрированной доменной службы
KrbServiceName HTTP/web.ipadomain0.ru
# Имя файла таблицы ключей
Krb5Keytab /etc/ipa/apache2.keytab
# Включение авторизации Kerberos по найденным билетам Kerberos
# Может быть отключено (off), тогда, если разрешено, будет
# запрашиваться имя и пароль (см.ниже)
KrbMethodNegotiate on
# Разрешение запрашивать имя и пароль для получения билета Kerberos
# Может быть отключено (off), тогда, если разрешено, будет
# использоваться имеющийся билет Kerberos (см.выше)
KrbMethodK5Passwd on
# Разрешение входа только авторизованным пользователям
require valid-user
# Сохранять аутентификационные данные для обеспечения сквозной
```

```
# аутентификации из сценариев с другими службами, например, сервером
# Postgresql
KrbSaveCredentials on
</Directory>
```

Созданный конфигурационный файл аутентификации необходимо указать в конфигурационных файлах виртуальных web-сайтов, размещаемых в каталоге `/etc/apache2/sites-available/`, с помощью директивы `Include`. Например, для конфигурационного файла `000-default.conf` виртуального web-сайта, устанавливаемого по умолчанию:

```
<VirtualHost *:80>
ServerAdmin webmaster@localhost
DocumentRoot /var/www/html

ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined

Include conf-available/kerberos-auth.conf
</VirtualHost>
```

Аналогично конфигурационный файл аутентификации `kerberos-auth.conf` можно включить в конфигурационный файл сайта `/etc/apache2/sites-enabled/default-ssl.conf`.

Для использования аутентификации Kerberos необходимо, чтобы web-браузер пользователя поддерживал метод аутентификации `negotiate`.

Для включения аутентификации `negotiate` в web-браузере Mozilla Firefox необходимо:

1) в адресной строке web-браузера ввести:

   `about:config`

2) указать серверы, для которых доступна аутентификация `negotiate`, задав для параметра `network.negotiate-auth.trusted-uris` маски соответствующих доменов;

3) если необходимо обеспечить сквозную аутентификацию из сценариев при работе с другими службами, например с сервером PostgreSQL, в web-браузере Mozilla Firefox для параметра `network.negotiate-auth.delegation-uris` следует задать маски доменов, которым можно передавать данные для сквозной аутентификации. В сценариях следует выставить переменную окружения `KRB5CCNAME`. Например, для языка PHP:

   ```
   putenv("KRB5CCNAME=".$_SERVER[?KRB5CCNAME?]);
   ```

8.3.11.2. Настройка защищенных соединений SSL с использованием сертификатов

При установке web-сервера Apache2 для защищенных соединений SSL по умолчанию используется предустановленный закрытый ключ `/etc/ssl/private/ssl-cert-snakeoil.key` и соответствующий ему сертификат `/etc/ssl/certs/ssl-cert-snakeoil.pem`. Данные ключ и сертификат следует заменить на ключ и сертификат, выданные удостоверяющим центром согласно 8.3.10.1.

Созданные удостоверяющим центром сертификат и ключ, например `apache.crt` и `apache.key`, следует сохранить в каталоге `/etc/ipa/`.

Расположение сертификатов необходимо указать в конфигурационных файлах web-сайтов, поддерживающих соединения SSL. Например, в конфигурационном файле `etc/apache2/sites-enabled/default-ssl.conf` web-сайта, устанавливаемого по умолчанию:

```
SSLCertificateFile /etc/ipa/apache.crt
SSLCertificateKeyFile /etc/ipa/apache.key
```

Для начала работы с использованием SSL необходимо:

1) загрузить модуль работы по протоколу SSL, выполнив команду:

```
sudo a2enmod ssl
```

2) включить web-сайт, для которого настраивается работа по протоколу SSL. Например, для включения устанавливаемого по умолчанию web-сайта `default-ssl` выполнить команду:

```
sudo a2ensite default-ssl
```

3) обновить конфигурацию web-сервера, выполнив команду:

```
sudo systemctl reload apache2
```

8.3.11.3. Настройка каталогов для работы с конфиденциальными данными

При необходимости возможно настроить каталоги для работы с конфиденциальными данными. Для этого следует:

1) на web-сервере назначить мандатные атрибуты каталогам с виртуальными серверами:

```
sudo pdpl-file 3:0:-1:CCNR /var/www/
sudo pdpl-file 3:0:-1:CCNR /var/www/html/
```

2) перезапустить web-сервер:

```
sudo systemctl restart apache2
```

8.3.12. Сквозная аутентификация в СУБД

Для работы СУБД PostgreSQL с FreeIPA необходимо выполнение следующих условий:

1) наличие в системах, на которых функционируют сервер и клиенты СУБД PostgreSQL, установленного пакета клиентской части FreeIPA `freeipa-client`;

2) разрешение имен должно быть настроено таким образом, чтобы имя системы разрешалось, в первую очередь, как полное имя (например, `postgres.example.ru`);

3) клиентская часть FreeIPA должна быть настроена на используемый FreeIPA домен (8.3.6).

Подробное описание работы с защищенной СУБД PostgreSQL приведено в документе РУСБ.10152-02 95 01-2.

Для обеспечения совместной работы сервера СУБД PostgreSQL с FreeIPA необходимо, чтобы сервер СУБД PostgreSQL функционировал как служба Kerberos. Выполнение данного условия требует наличия в БД Kerberos принципала для сервера СУБД PostgreSQL, имя которого задается в формате:

```
servicename/hostname@realm
```

где `servicename` — имя учетной записи пользователя, от которой осуществляется функционирование сервера СУБД PostgreSQL (по умолчанию `postgres`) и которое указывается в конфигурационном файле сервера PostgreSQL как значение параметра `krb_srvname`;

`hostname` — полное доменное имя системы, на которой функционирует сервер СУБД PostgreSQL;

`realm` — имя домена FreeIPA.

Для обеспечения совместной работы сервера СУБД PostgreSQL с FreeIPA необходимо:

1) создать в БД FreeIPA с помощью утилиты администрирования FreeIPA принципала, соответствующего устанавливаемому серверу PostgreSQL. Принципал создается с автоматически сгенерированным случайным ключом;

Пример

```
ipa service-add postgres/postgres.example.ru
```

2) создать файл ключа Kerberos для сервера СУБД PostgreSQL с помощью утилиты администрирования FreeIPA `ipa service-add`.

Пример

Создание файла ключа Kerberos на контроллере домена

```
ipa-getkeytab -s domain.example.ru -k /etc/apache2/keytab
     -p HTTP/apache2.example.ru
```

Полученный файл должен быть доступен серверу СУБД PostgreSQL по пути, указанному в конфигурационном параметре `krb_server_keyfile` (для приведенного примера путь `/etc/apache2/keytab`). Пользователю, от имени которого работает сервер СУБД PostgreSQL (по умолчанию `postgres`), должны быть предоставлены права на чтение данного файла;

3) назначить владельцем файла `krb5.keytab` пользователя `postgres`, выполнив команду:

```
chown postgres /etc/postgresql/x.x/main/krb5.keytab
```

4) задать в конфигурационном файле сервера СУБД PostgreSQL `/etc/postgreslq/x.x/main/postgresql.conf` следующие значения для параметров:

```
krb_server_keyfile = '/etc/postgresql/x.x/main/krb5.keytab'
krb_srvname = 'postgres'
```

5) указать для внешних соединений в конфигурационном файле сервера СУБД PostgreSQL `/etc/postgreslq/x.x/main/pg_hba.conf` метод аутентификации `gss`.

Пример

```
host all all 192.168.32.0/24 gss
```

8.3.13. Web-интерфейс FreeIPA

Использование web-интерфейса возможно после запуска FreeIPA согласно 8.3.5.1 или 8.3.5.2.

Для входа в web-интерфейс ввести в адресной строке браузера ссылку, предоставленную при запуске FreeIPA.

В случае если при первом входе в web-интерфейс появится сообщение о том, что соединение не защищено, следует добавить данный адрес в исключения.

Для входа в web-интерфейс используется имя учетной записи `admin` и пароль, заданный при запуске FreeIPA (см. 8.3.5.1 и 8.3.5.2).

8.3.13.1. Установка мандатных атрибутов (user mac)

Для установки мандатных атрибутов пользователя необходимо:

1) выбрать пользователя и перейти во вкладку «Параметры»;

2) используя раскрывающиеся списки «Min MAC», «Max MAC» и «Уровень целостности» задать мандатные атрибуты;

3) для установки мандатных атрибутов нажать **[Сохранить]**.

Поле «Мандатный атрибут» должно принять заданное значение в соответствии с рис. 4.

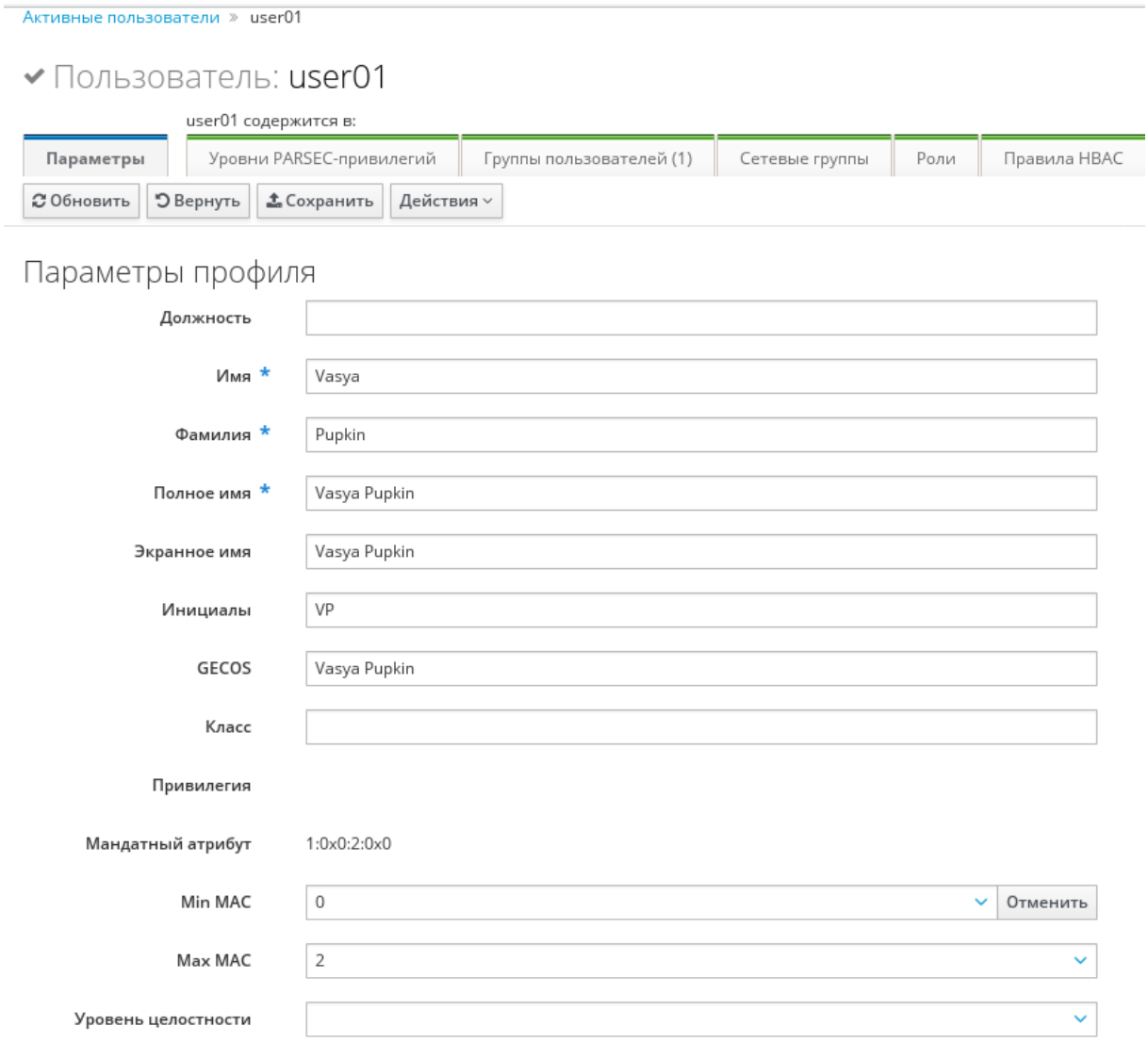

Рис. 4

8.3.13.2. Установка привилегий PARSEC (parsec cap)

Для установки привилегий PARSEC необходимо:

1) выбрать пользователя и перейти во вкладку «Уровни PARSEC-привилегий»;

2) нажать **[Добавить]**;

3) в открывшемся окне в блоке «Доступен» отметить требуемые привилегии;

4) переместить отмеченные привилегии в блок «Ожидаемый», нажав кнопку **[>]**, затем нажать **[Добавить]** (см. рис. 5).

Поле «Мандатный атрибут» должно принять заданное значение.

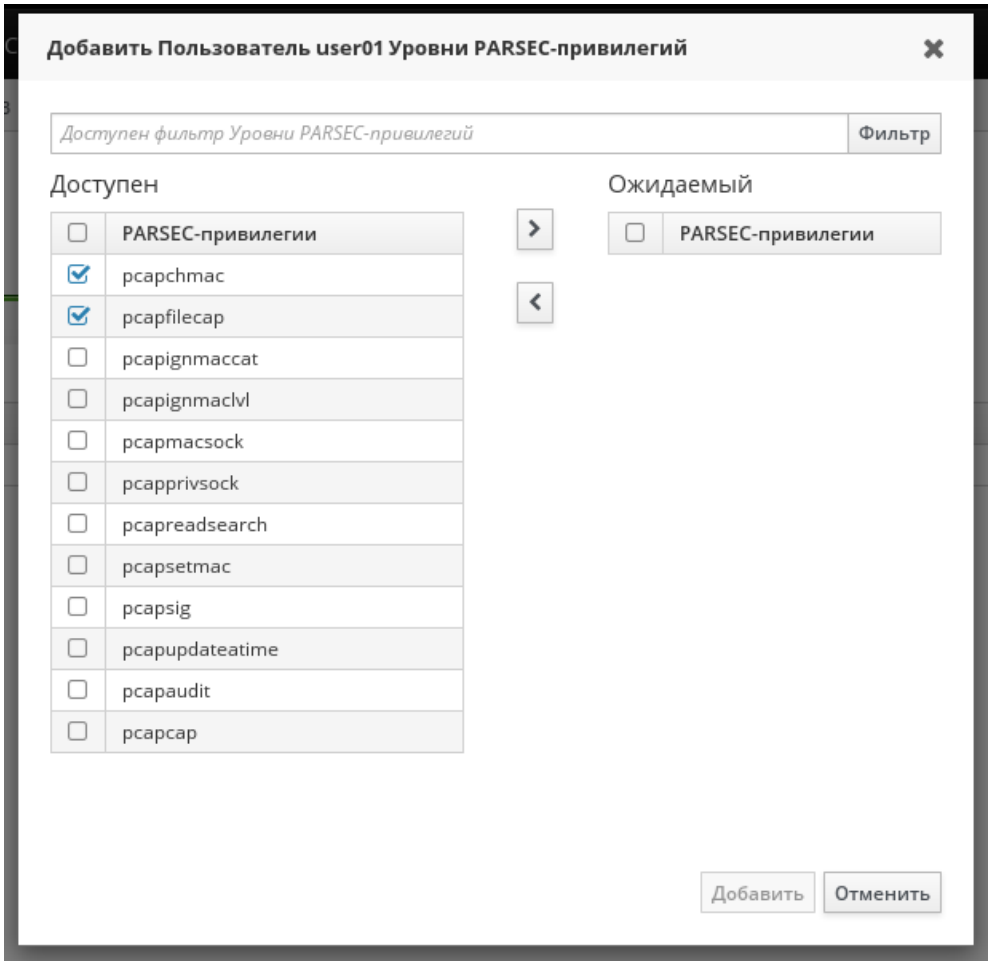

Рис. 5

Описание PARSEC-привилегий приведено в документе РУСБ.10152-02 97 01-1.

8.4. Samba

В состав ОС входит пакет программ Samba, предназначенный для решения задач совместимости со средой Microsoft Active Directory.

Samba позволяет ОС выступать как в роли контроллера домена AD, так и в роли клиента домена.

Возможности Samba:

- служба аутентификации на базе Kerberos;
- LDAP-совместимая служба каталогов с поддержкой репликации;
- поддержка групповых политик;
- поддержка доверительных отношений;
- DNS-сервер на базе BIND или собственной реализации.

В состав ОС входят консольные и графические средства, позволяющие инициализировать AD домен или подключиться к уже существующему.

Актуальные инструкции для разных сценариев применения приведены на официальном сайте `wiki.astralinux.ru`.

8.4.1. Настройка контроллера домена

В состав ОС входит инструмент командной строки `astra-sambadc`, включающий сценарии автоматизированной настройки и построения нового контроллера домена или включения в существующий домен в роли контроллера домена.

Для установки инструмента выполнить команду:

```
apt install astra-sambadc
```

При выполнении команды также будут установлены необходимые для работы домена AD пакеты `samba`, `winbind` и `ntp`.

Для создания нового домена в дополнение к инструменту `astra-sambadc` и автоматически устанавливаемым пакетам следует установить пакет `krb5-kdc`:

```
apt install krb5-kdc
```

Для создания нового домена используется команда:

```
astra-sambadc -d <имя_домена> -px
```

Данные, необходимые для создания домена и не указанные при выполнении команды, будут запрошены в интерактивном режиме.

Дополнительная информация по использованию команды доступна при выполнении команды с параметром `-h`:

```
astra-sambadc -h
```

Для настройки и построения нового контроллера домена или включения в существующий домен в роли контроллера домена в графическом режиме используется утилита `fly-admin-ad-server`.

Для установки графической утилиты выполнить команду:

```
apt install fly-admin-ad-server
```

Описание графической утилиты приведено в электронной справке.

8.4.2. Настройка участников домена

В состав ОС входит инструмент командной строки `astra-winbind`, включающий сценарии автоматизированной настройки компьютера для ввода в существующий домен.

ВНИМАНИЕ! Перед вводом компьютера в домен необходимо настроить на этом компьютере службу разрешения имен (DNS) так, чтобы в качестве сервера DNS использовался сервер DNS домена. Если этого не сделать, то контроллер домена не будет обнаружен.

Для ввода компьютера в домен используется команда:

```
astra-winbind -dc <имя_домена> -u <имя_администратора_домена> -px
```

Данные, необходимые для ввода в домен и не указанные при выполнении команды, будут запрошены в интерактивном режиме.

Дополнительная информация по использованию команды доступна при выполнении команды с параметром `-h`:

```
astra-winbind -h
```

Для ввода компьютера в существующий домен в графическом режиме используется утилита `fly-admin-ad-client`. Описание графической утилиты приведено в электронной справке.

Для проверки успешности присоединения к домену можно использовать команду:

```
net ads testjoin -k
```

8.5. Настройка сетевых служб

Ряд сетевых служб, таких как СУБД PostgreSQL, электронная почта, обработка гипертекстовых документов (web), система печати и др. для работы в ЕПП должны быть соответствующим образом настроены. Как правило, настройка заключается в обеспечении возможности использования этими службами сквозной аутентификации по Kerberos и получения необходимой информации из БД LDAP.

Примечание. При выполнении настройки сетевых служб потребуется использование учетной записи привилегированного пользователя через механизм `sudo`. При снятии блокировки на интерактивный вход в систему для суперпользователя `root` не рекомендуется осуществлять переключение в режим суперпользователя командой `su`. Необходимо использовать команду:

```
# su -
```

ВНИМАНИЕ! Для обеспечения нормальной работы пользователя с сетевыми службами в ЕПП должна быть явно задана его классификационная метка (диапазон уровней конфиденциальности и категорий конфиденциальности) с помощью соответствующих утилит, даже если ему не доступны уровни и категории выше 0.

Описание настройки следующих сетевых служб приведены в соответствующих подразделах:

- система обмена сообщениями электронной почты описана в 14.4;
- защищенный комплекс программ гипертекстовой обработки данных описан в 10.4;
- защищенный комплекс программ печати и маркировки документов описан в 12.3.2, а также в документе РУСБ.10152-02 95 01-2.

9. ВИРТУАЛИЗАЦИЯ СРЕДЫ ИСПОЛНЕНИЯ

9.1. Средства виртуализации[1)]

ОС поддерживает технологию виртуализации. Данная технология позволяет запускать множество виртуальных машин (называемых гостевыми) на одной физической машине (называемой хостовой машиной). При этом гостевые операционные системы, установленные на каждой из гостевых машин, могут целиком отличаться друг от друга и от операционной системы хостовой машины и являются полностью изолированными. Монитор виртуальных машин (гипервизор) обеспечивает параллельную работу гостевых операционных систем, их изоляцию, защиту, управление ресурсами и другие необходимые функции. Основными средствами, необходимыми для создания среды виртуализации, являются:

- сервер виртуализации libvirt;
- программа эмуляции аппаратного обеспечения QEMU.

9.1.1. Сервер виртуализации libvirt

Сервер виртуализации libvirt предоставляет средства создания, учета и управления виртуальными машинами. В эти задачи входит настройки конфигурации виртуальных машин и их запуск, управление файлами-образами дисковых носителей виртуальных машин, виртуальными сетевыми адаптерами и сетями и формирование контекста функционирования виртуальной машины в виде процесса ОС.

Пакет сервера виртуализации состоит из службы сервера виртуализации `libvirtd`, предоставляющей возможность удаленного управления по сети с использованием различных протоколов и способов аутентификации, клиентской библиотеки `libvirt0`, командной оболочки `virsh` и ряда других утилит командной строки. Графический интерфейс управления виртуализацией обеспечивается пакетом `virt-manager`.

ВНИМАНИЕ! Все конфигурационные файлы или файлы сервера виртуализации libvirt, содержащие ключевую информацию Kerberos или PKI, не должны быть доступны пользователям.

Сервер виртуализации использует следующие каталоги хостовой файловой системы (ФС):

1) `/etc/libvirt/` — каталог конфигурации сервера виртуализации libvirt:
   а) `qemu/` — каталог конфигурационных XML-файлов виртуальных машин QEMU:
      - `network/` — каталог конфигурационных XML-файлов виртуальных сетей;
      - `*.xml` — конфигурационные XML-файлы виртуальных машин QEMU;
   б) `storage/` — каталог конфигурационных файлов пулов файлов-образов;

[1)] Для процессоров, поддерживающих технологию виртуализации.

в) libvirt.conf — клиентский конфигурационный файл сервера виртуализации libvirt;

г) libvirtd.conf — конфигурационный файл службы сервера виртуализации libvirtd (см. 9.1.2);

д) qemu.conf — конфигурационный файл QEMU (см. 9.1.6);

2) /var/lib/libvirt/ — рабочий каталог сервера виртуализации libvirt:

а) images/ — каталог файлов-образов по умолчанию;

б) network/ — рабочий каталог виртуальных сетей;

в) qemu/ — рабочий каталог запущенных виртуальных машин QEMU:

- save/ — каталог сохраненных состояний виртуальных машин;
- snapshot — каталог снимков виртуальных машин;

г) runimages/ — каталог расположения копий файлов-образов виртуальных машин, запущенных в режиме запрета модификации файлов-образов;

3) /var/run/libvirt/ — каталог текущего рабочего состояния сервера виртуализации libvirt:

а) network/ — рабочий каталог запущенных виртуальных сетей;

б) qemu/ — каталог текущих конфигурационных xml-файлов запущенных виртуальных машин QEMU;

в) libvirt-sock — Unix-сокет для локальных соединений со службой сервера виртуализации libvirtd;

г) libvirt-sock-ro — Unix-сокет, доступный только для чтения, для локальных соединений со службой сервера виртуализации libvirtd.

9.1.2. Служба сервера виртуализации libvirtd

Служба сервера виртуализации libvirtd предоставляет возможность удаленного управления сервером виртуализации по сети с использованием различных протоколов и способов аутентификации. При этом поддерживается возможность решения всех задач по созданию и учету виртуальных машин, настройке их конфигурации и непосредственно запуска.

Доступ к службе сервера виртуализации возможен как с помощью локальных Unix-сокетов, так и по сети с помощью консольных или графических инструментов управления виртуальными машинами.

Основным конфигурационным файлом службы сервера виртуализации является /etc/libvirt/libvirtd.conf. Он содержит описание необходимых для работы службы настроек и параметров. Файл разбит на секции, описывающие параметры функционирования службы сервера виртуализации: интерфейсы взаимодействия и права доступа к ним,

способы и параметры аутентификации, политику разграничения доступа, состав выводимой в журнал информации и т.п. Наиболее важные параметры приведены в таблице 54.

Таблица 54 – Параметры конфигурационного файла `/etc/libvirt/libvirtd.conf`

Параметр	Описание
`listen_tls`	Принимать TLS-соединения с использованием сертификатов
`listen_tcp`	Принимать TCP-соединения **ВНИМАНИЕ!** Одной установки данного параметра недостаточно: необходимо указать параметр `-l` для параметра `libvirtd_opts` в конфигурационном файле `/etc/default/libvirtd`
`listen_addr`	Адрес сетевого интерфейса для приема соединений
`tls_port`	Порт для сетевых соединений TLS
`tcp_port`	Порт для сетевых соединений TCP
`auth_tcp`	Используемая для TCP-соединений аутентификация. Параметр должен содержать значение `"sasl"` (см. 9.1.3).
`access_drivers`	Применяемый драйвер доступа к серверу виртуализации. Параметр должен содержать значение `[ "parsec" ]`.
`admin_group`	Группа администраторов сервера виртуализации (по умолчанию `"libvirt-admin"`)

Примечание. Конфигурационные параметры TLS для доступа к серверу виртуализации libvirt рассматриваются в 9.1.7.

9.1.3. Конфигурационные файлы сервера виртуализации

При использовании механизмов SALS для доступа к серверу виртуализации libvirt или к рабочим столам виртуальных машин через систему VNC или по протоколу SPICE необходимо наличие соответствующих конфигурационных файлов с параметрами SASL в каталоге `/etc/sasl2`. Для сервера виртуализации требуется файл `libvirt.conf`, для QEMU (VNC и SPICE) — `qemu.conf`.

Описание основных параметров конфигурационного файла SASL `/etc/sasl2/libvirt.conf` приведено в таблице 55.

Таблица 55

Параметр	Описание
`mech_list`	Список механизмов SASL. При использовании в ЕПП ALD должен содержать только значение `gssapi`.
`keytab`	Путь к файлу ключевой информации Kerberos. Параметр необходим при использовании в ЕПП. Должен содержать корректные значения для файлов, содержащих ключевую информацию для VNC и SPICE
`sasldb_path`	Путь к базе данных SASL. При использовании в ЕПП не применяется и должен быть закомментирован.

ВНИМАНИЕ! Файлы ключевой информации Kerberos для VNC и SPICE должны быть доступны на чтение пользователям, запускающим виртуальные машины, и группе `libvirt-qemu`.

Для VNC и SPICE могут быть заданы другие пути расположения конфигурационного файла SASL. Описание конфигурационного файла `qemu.conf` приведено в 9.1.6.

9.1.4. Консольный интерфейс virsh

В состав пакетов сервера виртуализации libvirt входит консольный интерфейс управления виртуальными машинами `virsh`, позволяющий в консоли с помощью командной оболочки производить действия по управлению конфигурацией виртуальных машин.

Командная оболочка содержит набор команд по управлению виртуальными машинами, файлами-образами носителей, виртуальными интерфейсами и сетями и позволяет править конфигурационные файлы виртуальных машин.

Более подробно возможности консольного интерфейса управления виртуальными машинами `virsh` описаны в соответствующем руководстве `man`.

9.1.5. Графическая утилита virt-manager

Графическая утилита управления виртуальными машинами `virt-manager` предоставляет доступ к возможностям сервера виртуализации libvirt из графического интерфейса пользователя. Внешний вид окна утилиты приведен на рис. 6.

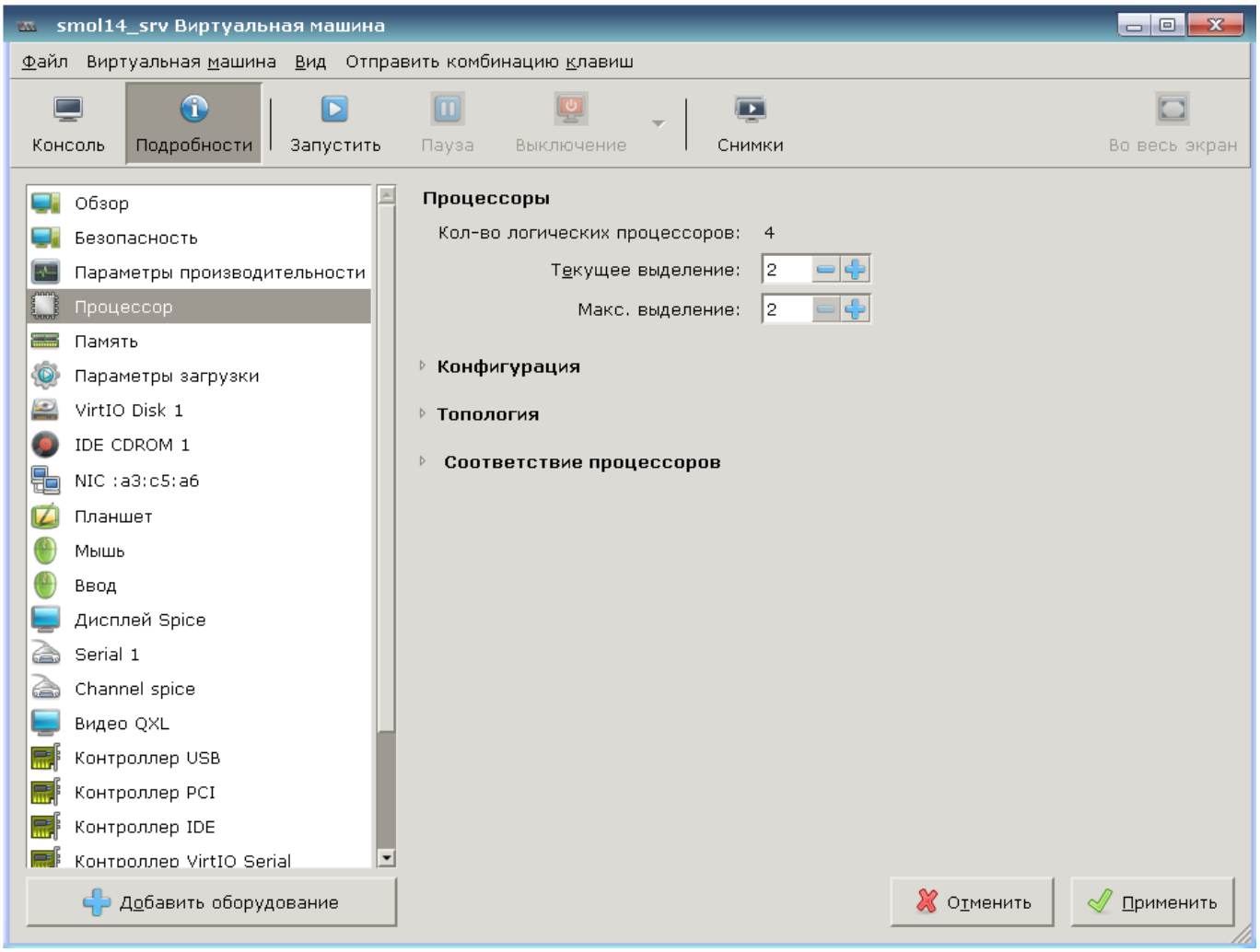

Рис. 6

Утилита позволяет выполнять действия по созданию виртуальных машин, управлению их конфигурацией и файлами-образов дисковых носителей. Также обеспечивает удаленный доступ к рабочему столу выбранной виртуальной машины по протоколам VNC и SPICE.

9.1.6. Средства эмуляции аппаратного обеспечения на основе QEMU

Средства эмуляции аппаратного обеспечения на основе QEMU реализуют программно-аппаратное окружение запускаемой виртуальной машины, включая заданную конфигурацию аппаратной платформы и набор эмулируемых устройств, доступных гостевой операционной системе. В случае совпадения гостевой аппаратной платформы и аппаратной платформы хостовой машины используются возможности аппаратной поддержки виртуализации средствами виртуализации KVM (Kernel-based Virtual Machine) для хостовых операционных систем семейства Linux.

Компонент состоит из пакетов, представляющих программу эмуляции аппаратного обеспечения QEMU для различных аппаратных платформ (в частности, аппаратных платформ x86 и x86-64) и необходимый набор утилит командной строки.

QEMU Guest Agent (гостевой агент QEMU) обеспечивает возможность взаимодействия с гостевой ОС. Для отсылки и получения команд данный агент использует последовательное соединение virtio. Он позволяет зафиксировать файловую систему до выполнения снимка, при этом в снимке не будет большой части записанных данных. Фиксация файловой системы возможна только с драйверами хранилищ Ceph и qcow2. Для использования агента необходимо установить пакет `qemu-guest-agent` на гостевой ОС.

ВНИМАНИЕ! Применение гостевого агента QEMU доступно только для виртуальных машин, запущенных из-под нулевого мандатного контекста.

Запущенная средствами QEMU/KVM виртуальная машина представляет собой отдельный процесс хостовой операционной системы.

Основным конфигурационным файлом QEMU является `/etc/libvirt/qemu.conf`. Он содержит описание параметров, необходимых для запуска и функционирования виртуальных машин: интерфейсов взаимодействия с рабочим столом виртуальных машин, способов и параметров аутентификации, политики управления безопасностью и изоляцией виртуальных машин, — а также значения по умолчанию некоторых параметров конфигурации виртуальных машин. Описание основных параметров конфигурационного файла `/etc/libvirt/qemu.conf` приведено в таблице 56.

Таблица 56

Параметр	Описание
`vnc_listen`	Адрес сетевого интерфейса для приема соединений VNC
`vnc_tls`	Использовать TLS для приема соединений VNC
`vnc_tls_x509_cert_dir`	Путь к сертификатам TLS при работе VNC
`vnc_password`	Пароль для соединений VNC
`vnc_sasl`	Использовать SASL для приема соединений VNC
`vnc_sasl_dir`	Каталог конфигурационного файла `qemu.conf`, содержащий SASL-параметры для приема соединений VNC (см. 9.1.3)
`spice_listen`	Адрес сетевого интерфейса для приема соединений по протоколу SPICE
`spice_tls`	Использовать TLS для приема соединений по протоколу SPICE
`spice_tls_x509_cert_dir`	Путь к сертификатам TLS при работе по протоколу SPICE
`spice_password`	Пароль для соединений по протоколу SPICE
`spice_sasl`	Использовать SASL для приема соединений по протоколу SPICE
`spice_sasl_dir`	Каталог конфигурационного файла `qemu.conf`, содержащий SASL-параметры для приема соединений по протоколу SPICE
`security_driver`	Применяемый драйвер безопасности. Параметр должен содержать значение `"parsec"`

Окончание таблицы 56

Параметр	Описание
`run_images_dir`	Каталог расположения копий файлов-образов виртуальных машин, запущенных в режиме запрета модификации файлов-образов

Примечание. Конфигурационные параметры TLS для доступа к рабочим столам виртуальных машин посредством VNC рассматриваются в 9.1.8.

ВНИМАНИЕ! При использовании в виртуальной машине SPICE-графики, в гостевой ОС должен быть установлен QXL-драйвер. В ОС драйвер устанавливается с пакетом `xserver-xorg-video-qxl`.

9.1.7. Идентификация и аутентификация при доступе к серверу виртуализации libvirt

Сервер виртуализации может использовать для идентификации и аутентификации клиентов следующие механизмы:

- локальная peer-cred аутентификация;
- удаленная SSH-аутентификация (строка соединения `qemu+ssh://...`);
- удаленная SASL-аутентификация, в том числе с поддержкой Kerberos (строка соединения `qemu+tcp://...`);
- удаленная TLS-аутентификация с использованием сертификатов (строка соединения `qemu+tls://...`).

Параметры для различных способов аутентификации задаются в конфигурационном файле `/etc/libvirt/libvirtd.conf`: параметры локальных UNIX сокетов (секция «UNIX socket access control»), разрешение приема сетевых соединений tcp и tls (параметры `listen_tls` и `listen_tcp`) и порты для их приема (параметры `tls_port` и `tcp_port`), расположение необходимых файлов при использовании сертификатов x509 (секция «TSL x509 certificate configuration»), варианты авторизации (параметры `auth_unix_ro`, `auth_unix_rw`, `auth_tcp`, `auth_tls`).

По умолчанию используется следующее расположение файлов сертификатов на сервере для аутентификации к серверу виртуализации libvirt:

- `/etc/pki/CA/cacert.pem` — корневой сертификат;
- `/etc/pki/CA/crl.pem` — файл отозванных сертификатов;
- `/etc/pki/libvirt/servercert.pem` — сертификат открытого ключа сервера виртуализации libvirt;
- `/etc/pki/libvirt/private/serverkey.pem` — закрытый ключ сервера виртуализации libvirt.

Примечание. Файлы ключей сервера виртуализации libvirt должны быть доступны на чтение группе `libvirt-qemu`.

По умолчанию используется следующее расположение файлов сертификатов на клиенте для аутентификации к серверу виртуализации libvirt (в домашнем каталоге пользователя ~):

- `/etc/pki/CA/cacert.pem` — корневой сертификат;
- `~/.pki/libvirt/clientcert.pem` — сертификат открытого ключа клиента;
- `~/.pki/libvirt/clientkey.pem` — закрытый ключ клиента.

В случае SASL-аутентификации используется конфигурационный файл `/etc/sasl2/libvirt.conf`, в котором задаются параметры аутентификации SASL (например применяемые механизмы). Имя службы сервера виртуализации libvirt при использовании SASL-аутентификации регистрируется как `libvirt/<имя сервера>@<домен>`.

ВНИМАНИЕ! При указании механизма SASL `gssapi` следует в конфигурационном файле `/etc/default/libvirtd` указать с помощью соответствующей переменной окружения расположение файла ключей Kerberos сервера виртуализации, например:

`export KRB5_KTNAME=/etc/libvirt/libvirt.keytab`.

9.1.8. Идентификация и аутентификация при доступе к рабочему столу виртуальных машин

Параметры аутентификации при доступе к рабочему столу виртуальной машины задаются в конфигурационном файле `/etc/libvirt/qemu.conf` раздельно для VNC и SPICE. В данном конфигурационном файле указываются необходимые параметры аутентификации и пути к конфигурационным файлам SASL, например `/etc/sasl2/qemu.conf`. Имя служб VNC и SPICE при использовании SASL-аутентификации регистрируется как `vnc/<имя сервера>@<домен>` и `spice/<имя сервера>@<домен>`, соответственно.

По умолчанию используется следующее расположение файлов сертификатов на сервере для аутентификации к виртуальной машине посредством VNC:

- `/etc/pki/libvirt-vnc/ca-cert.pem` — корневой сертификат;
- `/etc/pki/libvirt-vnc/server-cert.pem` — сертификат открытого ключа сервера VNC QEMU;
- `/etc/pki/libvirt-vnc/server-key.pem` — закрытый ключ сервера VNC QEMU.

Примечание. Файлы ключей сервера VNC QEMU должны быть доступны на чтение группе `libvirt-qemu`.

По умолчанию используется следующее расположение файлов сертификатов на клиенте для аутентификации к серверу VNC QEMU (в домашнем каталоге пользователя ~):

- `/etc/pki/CA/cacert.pem` — корневой сертификат;

- ~/.pki/libvirt-vnc/clientcert.pem — сертификат открытого ключа клиента;
- ~/.pki/libvirt-vnc/private/clientkey.pem — закрытый ключ клиента.

9.2. Контейнеризация

В ОС реализован механизм контейнеризации, обеспечивающий режим виртуализации и изоляции ресурсов на уровне ядра операционной системы. Использование данного механизма позволяет запускать приложение и необходимый ему минимум системных библиотек в полностью стандартизованном контейнере, соединяющемся с хостовой машиной при помощи определенных интерфейсов. Контейнеры используют ядро операционной системы хостовой машины и, в отличие от полной виртуализации, не требуют эмуляции аппаратного обеспечения. Приложения, запущенные внутри разных контейнеров, изолированы и не могут влиять друг на друга.

В состав ОС входит программное обеспечение Docker для автоматизации развертывания и управления приложениями в средах с поддержкой контейнеризации.

9.2.1. Установка Docker

Установка Docker возможна либо через графический менеджер пакетов Synaptic, либо через терминал с помощью команды:

```
sudo apt install docker.io
```

После установки следует включить в группу docker пользователей, которые будут работать с Docker, командой:

```
sudo usermod -aG docker <имя_пользователя>
```

Текущего пользователя можно включить в группу командой:

```
sudo usermod -aG docker $USER
```

Для того, чтобы эта команда вступила в силу, следует выйти из текущей сессии пользователя и зайти повторно.

Включение в группу docker позволит пользователям в дальнейшем запускать Docker без использования sudo.

9.2.2. Работа с Docker

Полный список команд Docker доступен по команде:

```
docker help
```

Информацию об аргументах конкретной команды, например docker attach, можно получить следующими способами:

```
docker attach --help
man docker-attach
```

Далее описана работа с наиболее важными функциями Docker.

ВНИМАНИЕ! Приведенное описание подразумевает работу в привилегированном режиме. Данный режим не рекомендуется к применению в связи с потенциальной небезопасностью использования контейнеров в привилегированном режиме. Рекомендуется работать с Docker в режиме rootless, описанном в 9.2.2.6.

9.2.2.1. Создание образа Docker

Образ — это шаблон контейнера, включающий в себя:

1) базовую файловую систему;

2) слои — изменения в файловой системе, расположенные друг над другом в том порядке, в котором эти изменения были произведены;

3) параметры выполнения, используемые при запуске контейнера из данного образа.

Примечание. Из одного образа возможно запускать несколько контейнеров.

Каждый слой образа представляет собой инструкцию, выполняемую в базовой файловой системе при создании образа. В процессе работы контейнера изменения файловой системы образуют новый слой контейнера, а слои образа остаются неизменными.

Слои могут быть последовательно записаны в текстовом документе, который называется докерфайлом (Dockerfile).

Образ возможно создать тремя способами:

- из `chroot`-окружения;
- с помощью докерфайла.
- на основе контейнера.

Создание образа из chroot-окружения

Для создания собственных образов Docker из `chroot`-окружения необходимо установить пакет `debootstrap`. Это можно сделать либо с помощью графического менеджера пакетов Synaptic, либо из терминала, выполнив команду:

```
sudo apt -y install debootstrap docker.io
```

Для создания образа Docker необходимо:

1) собрать `chroot`-окружение;

2) настроить `chroot`-окружение;

3) конвертировать `chroot`-окружение в образ Docker.

Сборка `chroot`-окружения выполняется инструментом командной строки `debootstrap`.

Загрузка пакетов для сборки `chroot`-окружения может быть выполнена из репозиториев, доступных по сети, например:

```
sudo debootstrap --verbose
     --components=main,contrib,non-free smolensk chroot-smolensk17
     ftp://<сервер>/<путь_к_репозиторию>
```

где chroot-smolensk17 — каталог сборки окружения;

ftp://<сервер>/<путь_к_репозиторию> — расположение репозитория в сети.

Загрузка пакетов для сборки chroot-окружения также может быть выполнена из копии репозитория в локальной ФС, например:

```
debootstrap --verbose smolensk chroot-smolensk17
    file:///<путь_к_локальному_репозиторию>
```

где chroot-smolensk17 — каталог сборки окружения;

file:///<путь_к_локальному_репозиторию> — каталог локального репозитория.

Настройка chroot-окружения выполняется следующим образом:

1) при необходимости настроить для chroot-окружения разрешение имен в файле /etc/resolv.conf и список репозиториев в /etc/apt/sources.list (например скопировать настройки из одноименных файлов на хостовой машине);

2) перейти в chroot-окружение командой sudo chroot и обновить пакеты окружения:

```
sudo chroot chroot-smolensk17
apt update
apt dist-upgrade
exit
```

Для создания образа Docker следует добавить настроенное chroot-окружение в архив с помощью утилиты tar и конвертировать полученный архив в образ командой docker import.

Пример

Создать образ wiki/astralinux:smolensk17 из chroot-окружения:

```
sudo tar -C chroot-smolensk17 -cpf - . | sudo docker import -
    wiki/astralinux:smolensk17 --change "ENV PATH
    /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"  --change
    'CMD ["/bin/bash"]'
```

где -C chroot-smolensk17 — задать каталог chroot-smolensk17 в качестве рабочего каталога для архивирования;

-cpf - . — создать новый архив из рабочего каталога с сохранением разрешений, установленных на входящие в него каталоги и файлы, и передать архив в stdin;

docker import - — импортировать данные для создания образа из stdout;

-change "ENV PATH ..." — задать переменную окружения PATH;

-change 'CMD ["/bin/bash"]' — задать команду, которая будет автоматически выполнена в контейнере при запуске контейнера из данного образа.

Если все операции выполнены успешно, то созданный образ будет виден в списке образов, доступном по команде:

```
sudo docker images
```

В созданный образ можно войти, выполнив команду:

```
sudo docker run -it --rm wiki/astralinux:smolensk17
```

где `-i` — запустить контейнер в интерактивном режиме;

`-t` — выделить терминал для контейнера;

`--rm` — удалить контейнер после выхода из него.

Создание образа с использованием докерфайла

Команда `docker build` позволяет создавать новые образы на основе существующих, используя инструкции из докерфайла и контекст — совокупность каталогов и файлов в указанном месте. Местоположение контекста может быть задано как путь к каталогу в файловой системе либо как ссылка на репозиторий Git в сети. Докерфайл по умолчанию называется `Dockerfile` и располагается в корневом каталоге контекста. Произвольное расположение докерфайла задается параметром `-f`, например:

```
docker build -f <путь_к_докерфайлу> .
```

Перед выполнением инструкций в докерфайле проводится проверка инструкций на корректность. Если в инструкциях содержится ошибка (например неправильный синтаксис), при попытке собрать образ будет выведено сообщение об ошибке.

Пример

Создать образ с использованием докерфайла, содержащего несуществующую инструкцию `RUNCMD`:

```
docker build -t test/myimg .
```

В терминале будет выведено сообщение об ошибке:

```
Sending build context to Docker daemon 2.048 kB
Error response from daemon: Unknown instruction: RUNCMD
```

При создании нового образа слои инструкций выполняются строго в том порядке, в котором они записаны.

Чтобы собрать новый образ на основе существующего (например `wiki/astralinux:smolensk17`) следует воспользоваться командой `docker build`:

1) создать корневой каталог контекста сборки:

```
mkdir build-smolensk
```

2) создать в контексте сборки файл, например `data-to-import`, содержащий произвольный текст:

```
echo "Это импортированные данные" > build-smolensk/data-to-import
```

3) в файл build-smolensk/Dockerfile внести следующий текст:

```
# указание из какого образа выполнять сборку
FROM wiki/astralinux:smolensk17
# скопировать файл data-to-import из контекста сборки в образ
COPY /data-to-import /srv
# создать в образе пустой файл /srv/created-file
RUN touch /srv/created-file
# вывести на печать содержимое скопированного файла
RUN cat /srv/data-to-import
# вывести на печать рабочий каталог
RUN echo Current work directory is $(pwd)
```

4) выполнить сборку образа, например с меткой test:

```
docker build -t test build-smolensk/
```

Вывод в терминале будет иметь следующий вид:

```
Sending build context to Docker daemon   5.12kB
Step 1/5 : FROM wiki/astralinux:smolensk17
---> 60d0611fe56a
Step 2/5 : COPY /data-to-import /srv
---> 7a75a002d29f
Step 3/5 : RUN touch /srv/created-file
---> Running in 709bb54af8c3
Removing intermediate container 709bb54af8c3
---> b5fd28178901
Step 4/5 : RUN cat /srv/data-to-import
---> Running in 4c69f455cf2f
Это импортированные данные
Removing intermediate container 4c69f455cf2f
---> c8f8c7c3797a
Step 5/5 : RUN echo Current work directory is $(pwd)
---> Running in 27db5fcaaba5
Current work directory is /
Removing intermediate container 27db5fcaaba5
---> 14446097a09e
Successfully built 14446097a09e
Successfully tagged test:latest
```

5) проверить, что образ test присутствует в списке образов:

```
docker images
```

6) если образ был успешно создан, запустить контейнер из образа и проверить содержимое контейнера:

```
docker run --rm -it test
```

Вывод в терминале будет иметь следующий вид:

```
root@978a4cc9fbd8:/# ls -l /srv
total 4
-rw-r--r-- 1 root root  0 Jan 20 10:12 created-file
-rw-r--r-- 1 root root 51 Jan 20 10:11 data-to-import

root@978a4cc9fbd8:/# cat /srv/data-to-import
Это импортированные данные

root@978a4cc9fbd8:/# exit
```

Из вывода в терминале видно, что в контейнере присутствует файл `data-to-import`, скопированный из контекста сборки в образ, и пустой файл `created-file`, созданный в образе при сборке.

Подробное описание команды `docker build` и работы с докерфайлами приведено в `man docker-build` и `man dockerfile`, соответственно.

Создание образа из контейнера

При наличии сохраненного или активного контейнера (описание работы с контейнерами приведено в 9.2.2.3) данный контейнер возможно конвертировать в образ Docker следующей командой:

```
docker container commit <параметры> <имя_контейнера> <имя_образа>
```

Пример

Создать образ `test-image` из контейнера `test`:

```
docker container commit test test-image
```

Все изменения в контейнере относительно образа, из которого тот был запущен, а также команды, переданные в качестве параметров при создании нового образа, сформируют новый слой создаваемого образа.

Параметры данной команды описаны в `man docker-container-commit`.

9.2.2.2. Копирование образа

Образ, хранящийся на локальной машине, может быть скопирован (например на другую машину).

Пример

Для того чтобы скопировать образ `wiki/astralinux:smolensk17` на другую машину, необходимо:

1) выгрузить образ в архив:

```
docker save -o astralinux-smolensk17.bz2 wiki/astralinux:smolensk17
```

где `-o` — задает имя файла, в который будет выведен образ. Если этот параметр не указан, образ будет выведен в `stdout`;

2) скопировать полученный файл `astralinux-smolensk17.bz2` на целевую машину;

3) на целевой машине загрузить файл в локальный реестр образов:

```
docker load -i astralinux-smolensk17.bz2
```

где `-i` — указывает имя файла, из которого будет загружен образ. Если этот параметр не указан, образ будет загружен из `stdin`.

Эту процедуру возможно выполнить одной командой с копированием созданного архива через SSH (для этого на целевой машине должен быть настроен SSH):

```
docker save wiki/astralinux:smolensk17 | bzip2 | ssh user@host
'bunzip2 | docker load'
```

где `docker save wiki/astralinux:smolensk17` — выгрузить образ `wiki/astralinux:smolensk17` в `stdout`;

`bzip2` — программа сжатия данных;

`ssh user@host 'bunzip2 | docker load'` — подключиться через SSH к машине с именем `host` от имени пользователя `user` и запустить команды загрузки образа из стандартного ввода (`stdin`). Пользователь `user` на целевой машине должен иметь право работать с Docker без использования `sudo`.

9.2.2.3. Создание и работа с контейнерами

Для того, чтобы создать новый контейнер с заданным именем из образа, используется следующая команда:

```
docker run <параметры> <имя_образа>
```

Примеры:

1. Создать контейнер с именем `run-smolensk` из образа `smolensk`:

```
docker run --name run-smolensk --rm -it smolensk
```

где `run-smolensk` — имя контейнера. Если параметр не указан, присваивается случайное имя;

`-rm` — уничтожить контейнер после завершения его работы. Если параметр не указан, контейнер будет локально сохранен;

`-i` — запустить контейнер в интерактивном режиме. Если параметр не указан, контейнер запустится в фоновом режиме;

`-t` — создать терминал;

`smolensk` — имя образа, из которого создается контейнер.

2. Для создания нескольких контейнеров с произвольными именами из образа `smolensk` следует:

1) запустить контейнер из образа `smolensk`:

```
docker run smolensk
```

2) выполнить команду повторно;

3) вывести список контейнеров:

```
docker container ls -a
```

В выводе будут отображены два контейнера со случайными именами, созданные из образа `smolensk`:

```
CONTAINER ID IMAGE ... NAMES
b894e0b0b22d smolensk ... admiring_murdock
825a33f9c18c smolensk ... amazing_morse
```

Для запуска сохраненного контейнера следует использовать команду:

```
docker start <имя_контейнера>
```

Пример

Запустить контейнер `amazing_morse` в интерактивном режиме:

```
docker start -ai amazing_morse
```

К контейнеру, работающему в фоновом режиме, можно подключиться командой:

```
docker attach <имя_контейнера>
```

Пример

Подключиться к контейнеру `amazing_morse`, работающему в фоновом режиме:

```
docker attach amazing_morse
```

9.2.2.4. Запуск контейнеров на выделенном уровне МКЦ

С целью изоляции и ограничения среды исполнения потенциально опасного или вредоносного кода в ОС реализована возможность запуска контейнеров на низком уровне МКЦ. Описание функции приведено в документе РУСБ.10152-02 97 01-1.

9.2.2.5. Монтирование файловых ресурсов хостовой машины в контейнер

Docker поддерживает следующие типы монтирования файловых ресурсов хостовой машины в контейнер:

1) `bind` — монтирование файла или каталога, расположенного на хостовой машине, в контейнер;

2) `mount` — монтирование управляемых Docker изолированных томов для хранения данных;

3) `tmpfs` — монтирование временного файлового хранилища (`tmpfs`) в контейнер. Это позволяет контейнеру размещать временные ресурсы в памяти хостовой машины.

Параметры монтирования задаются при создании контейнеров и сохраняются в течение их работы.

ВНИМАНИЕ! Чтобы предотвратить нежелательные изменения в конфигурации хостовой машины, следует исключить монтирование файловых ресурсов, влияющих на конфигурацию хостовой машины, либо ограничить права доступа контейнера к файловым ресурсам правом на чтение.

Параметры монтирования могут быть заданы с использованием одного из двух флагов, определяющих формат, в котором будут заданы параметры и их значения:

1) с использованием флага `-v` — параметры монтирования задаются набором значений, разделенных двоеточием. Набор параметров зависит от типа монтирования. Например, тип монтирования `bind` будет иметь следующий вид:

```
docker run -v <монтируемый_ресурс>:<точка_монтирования>:
     <дополнительные_параметры> <имя_образа>
```

2) с использованием флага `--mount` — параметры монтирования задаются в виде `<параметр>=<значение>` и отделяются друг от друга запятыми:

```
docker run --mount type=<тип_монтирования>,source=<монтируемый_ресурс>,
     target=<точка_монтирования>,<дополнительные_параметры> <имя_образа>
```

bind

Тип монтирования `bind` монтирует каталог, расположенный на хостовой машине, в ФС контейнера. Содержимое каталога на хостовой машине и в точке монтирования в контейнере полностью идентично, а изменения в одном каталоге повторяются в другом.

ВНИМАНИЕ! Данный метод монтирования является устаревшим.

С использованием флага `-v` параметры типа монтирования `bind` задаются следующим образом:

```
docker run -v <монтируемый_ресурс>:<точка_монтирования>:
     <дополнительные_параметры> <имя_образа>
```

С использованием флага `--mount` параметры типа монтирования `bind` задаются следующим образом:

```
docker run --rm -it --mount type=<тип_монтирования>,
    source=<монтируемый_ресурс>,target=<точка_монтирования>,
    <дополнительные_параметры> <имя_образа>
```

Примеры:

1. Смонтировать рабочий каталог хостовой машины в каталог `/app` контейнера с параметром `read-only`, используя флаг `--mount`:

```
docker run --rm -it --mount type=bind,source="$(pwd)",target=/app,
    readonly smolensk
```

2. Смонтировать рабочий каталог хостовой машины в каталог `/app` контейнера с параметром `read-only`, используя флаг `-v`:

```
docker run --rm -it -v $(pwd):/app:ro smolensk
```

Тип монтирования `bind` позволяет настраивать распространение монтирования (bind propagation). В контексте контейнеризации распространение монтирования определяет, как события монтирования (монтирование и размонтирование ресурсов) в контейнере могут повлиять на ресурсы хостовой машины и/или других контейнеров, а события монтирования на хостовой машине — на ресурсы одного или нескольких контейнеров.

Для настройки распространения монтирования используется параметр `bind-propagation`. Параметр принимает следующие значения:

- `shared` — если при создании контейнера в него был смонтирован каталог с этим параметром, например каталог `/myfiles` на хостовой машине в `/contfiles` в контейнере, то другие ресурсы, смонтированные внутри `/contfiles`, также будут доступны в `/myfiles`. Аналогично ресурсы, смонтированные внутри `/myfiles`, будут доступны в `/contfiles`. Если при создании нескольких контейнеров в каждый из них был смонтирован с этим параметром один и тот же каталог, то смонтированные внутри него ресурсы также будут доступны в каждом из данных контейнеров;

  ВНИМАНИЕ! В режиме `shared` изменения в одной точке монтирования распространяются на все остальные точки монтирования, что может привести к нежелательным изменениям файловых объектов других контейнеров, и, как следствие, нарушению их работы;

- `rshared` — то же, что `shared`, но применяется рекурсивно;

- `slave` — если при создании контейнера в него был смонтирован каталог с этим параметром, например каталог `/myfiles` на хостовой машине в `/contfiles` в контейнере, то другие ресурсы, смонтированные внутри `/myfiles`, также будут

доступны в каталоге /contfiles, но при этом ресурсы, смонтированные внутри /contfiles, не будут доступны на хостовой машине;

- rslave — то же, что slave, но применяется рекурсивно;

- private — если при создании контейнера в него был смонтирован каталог с этим параметром, например каталог /myfiles на хостовой машине в /contfiles в контейнере, то другие ресурсы, смонтированные внутри /contfiles, не будут доступны на хостовой машине, а ресурсы, смонтированные внутри /myfiles, не будут доступны в контейнере;

- rprivate — то же, что private, но применяется рекурсивно. Используется по умолчанию.

Примеры:

1. Смонтировать подкаталог /target рабочего каталога хостовой машины в каталог /app контейнера с типом распространения монтирования rslave, используя флаг --mount:

```
docker run -d -it --mount type=bind,source="$(pwd)"/target,target=/app,
readonly, bind-propagation=rslave smolensk
```

2. Смонтировать подкаталог /target рабочего каталога хостовой машины в каталог /app контейнера с типом распространения монтирования shared, используя флаг -v:

```
docker run -d -it -v "$(pwd)"/target:/app:ro,shared smolensk
```

mount

Том Docker представляет собой файловую систему, расположенную на хостовой машине вне контейнера и находящуюся по управлением Docker. Тома хранятся в каталоге Docker на хостовой машине, например /var/lib/docker/volumes/.

Тома существуют независимо от жизненного цикла контейнера и могут быть многократно использованы разными контейнерами.

Управление томами описано в man docker-volume.

Для создания тома используется следующая команда:

```
docker volume create <имя_тома>
```

Пример

Создать том с именем my-vol:

```
docker volume create my-vol
```

С флагом -v параметры монтирования mount задаются следующим образом:

```
docker run -v <имя_тома>:<точка_монтирования> <имя_образа>
```

С использованием флага --mount команда будет иметь следующий вид:

```
docker run --mount src=<имя_тома>,dst=<точка_монтирования> <имя_образа>
```

Примеры:

1. Смонтировать том `my-vol` в каталог `/app` контейнера с использованием флага `-v`:

```
docker run --rm -it -v my-vol:/app smolensk
```

2. Смонтировать том `my-vol` в каталог `/app` контейнера с использованием флага `--mount`:

```
docker run --rm -it --mount src=my-vol,dst=/app smolensk
```

tmpfs

Тип монтирования `tmpfs` монтирует временное файловое хранилище (`tmpfs`) в ФС контейнера, что позволяет контейнеру хранить временные файловые ресурсы в памяти хостовой машины. Доступ к этим файловым ресурсам имеет только тот контейнер, в котором они были созданы. При остановке контейнера временные файловые ресурсы будут полностью удалены из ФС контейнера и памяти хостовой машины.

С использованием флага `--mount` параметры монтирования `tmpfs` задаются следующим образом:

```
docker run --mount type=tmpfs,destination=<точка_монтирования> <имя_образа>
```

С использованием флага `--tmpfs` команда будет иметь следующий вид:

```
docker run --tmpfs <точка_монтирования> <имя_образа>
```

Примеры:

1. Запустить контейнер из образа `smolensk` с монтированием `tmpfs` в каталог контейнера `/app`, используя флаг `--mount`:

```
docker run --rm -it --mount type=tmpfs,destination=/app smolensk
```

2. Запустить контейнер из образа `smolensk` с монтированием `tmpfs` в каталог контейнера `/app`, используя флаг `--tmpfs`:

```
docker run --rm -it --tmpfs /app smolensk
```

Примечания:

1. Синтаксис `--tmpfs` не поддерживает использование параметров монтирования.
2. Монтирование `tmpfs` не поддерживает флаг `-v`.

9.2.2.6. Работа с Docker в непривилегированном режиме

Для работы с Docker в непривилегированном (rootless) режиме используются утилиты `rootless-helper-astra` и `rootlessenv`, входящие в состав пакета `rootless-helper-astra`. Для установки режима rootless необходимо выполнить следующие шаги:

1) установить Docker для работы в привилегированном режиме, как описано в 9.2.1;
2) установить пакет `rootless-helper-astra`:

```
sudo apt install rootless-helper-astra
```

3) запустить службы rootless Docker для текущего пользователя:

```
sudo systemctl start rootless-docker@$USER
```

4) при необходимости для настройки автозапуска служб выполнить:

```
sudo systemctl enable rootless-docker@$USER
```

Запуск и настройка автозапуска служб могут быть выполнены для нескольких пользователей.

Чтобы запустить службы для нескольких пользователей, необходимо отдельно для каждого пользователя выполнить:

```
sudo systemctl start rootless-helper@<имя_пользователя>
```

Чтобы настроить автозапуск служб для нескольких пользователей, необходимо отдельно для каждого пользователя выполнить:

```
sudo systemctl enable rootless-helper@<имя_пользователя>
```

Инструмент `rootlessenv` настраивает окружение для работы в режиме rootless.

Чтобы запустить контейнер от имени текущего пользователя с использованием `rootlessenv`, следует выполнить:

```
rootlessenv docker run --rm -ti <имя_образа>
```

Для запуска контейнера от имени произвольного пользователя с `rootlessenv` выполнить:

```
sudo -u <имя_пользователя> rootlessenv docker run --rm -ti <имя_образа>
```

Аналогичным образом в режиме rootless выполняются другие команды Docker.

Работа с образами и контейнерами, созданными в режиме rootless, возможна только в режиме rootless.

Пример

Получить список контейнеров, созданных в режиме rootless:

1) запустить контейнер в режиме rootless:

```
rootlessenv docker run --rm -ti smolensk17
```

2) получить список созданных контейнеров в привилегированном режиме:

```
sudo docker container list
```

Созданного образа не будет в списке;

3) получить список созданных контейнеров в режиме rootless:

```
rootlessenv docker container list
```

В списке будет отображен созданный контейнер.

Работа с утилитами `rootless-helper-astra` и `rootlessenv` более подробно описана в `man rootless-helper-astra` и `man rootlessenv`, соответственно.

10. ЗАЩИЩЕННЫЙ КОМПЛЕКС ПРОГРАММ ГИПЕРТЕКСТОВОЙ ОБРАБОТКИ ДАННЫХ

Защищенный комплекс программ гипертекстовой обработки данных — это ПО, осуществляющее взаимодействие по HTTP-протоколу между сервером и браузерами: прием запросов, поиск указанных файлов и передача их содержимого, выполнение приложений на сервере и передача клиенту результатов их выполнения. Комплекс представлен web-сервером Apache2 и браузером Firefox.

ВНИМАНИЕ! Для обеспечения нормальной работы пользователя с сетевыми службами должна быть явно задана его классификационная метка (диапазон уровней конфиденциальности и категорий конфиденциальности) с помощью соответствующих утилит, даже если ему не доступны уровни и категории выше 0. Дополнительная информация приведена в документе РУСБ.10152-02 97 01-1.

10.1. Настройка сервера

После установки сервера Apache2 необходимо установить пакет `libapache2-mod-authnz-pam` для его настройки и подготовки к приему запросов на всех сетевых интерфейсах на 80 порту.

Если по каким-то причинам он не работоспособен, следует проверить минимально необходимые настройки сервера:

1) в файле `/etc/apache2/ports.conf` должны быть указаны параметры:

```
NameVirtualHost *:80
Listen 80
```

2) в каталоге `/etc/apache2/sites-available` должны находиться файлы с настройками виртуальных хостов и как минимум один из них должен быть разрешен к использованию командой:

```
a2ensite config_filename
```

ВНИМАНИЕ! В команде необходимо использовать только имя файла (без указания полного пути).

Минимальное содержимое таких файлов с конфигурациями виртуальных хостов выглядит следующим образом:

```
<VirtualHost *:80>
        ServerAdmin webmaster@localhost
        ServerName server.domain.name
        DocumentRoot /path/to/root/dir/
        <Directory /path/to/root/dir/>
                Options Indexes FollowSymLinks MultiViews
                AllowOverride None
        </Directory>
```

```
        ErrorLog /var/log/apache2/error.log
        LogLevel warn
        CustomLog /var/log/apache2/access.log combined
</VirtualHost>
```

В случае когда web-сервер должен предоставлять пользователям доступ к объектам файловой системы с различными мандатными атрибутами, на корневой каталог виртуального хоста (по умолчанию `/var/www/html`) и все его родительские каталоги должны быть установлены значения мандатных атрибутов не меньше максимальных атрибутов объектов, к которым будет разграничиваться доступ. Кроме того, на корневой каталог виртуального хоста (по умолчанию `/var/www/html`) должен быть установлен тип метки `ccnr`. Операция может быть выполнена с использованием утилиты `pdpl-file` от имени учетной записи администратора через механизм `sudo`. Дополнительная информация приведена в документе РУСБ.10152-02 97 01-1.

После окончания правки конфигурационных файлов необходимо перезапустить сервер командой:

```
systemctl restart apache2
```

10.2. Режим работы AstraMode

Сервер гипертекстовой обработки данных Apache2, входящий в состав ОС, в условиях применения мандатного управления доступом не допускает возможности анонимного использования ресурсов и требует обязательной настройки авторизации пользователей.

Для управления авторизацией пользователей и обеспечения работы средств разграничения доступа в сервере гипертекстовой обработки данных Apache2 используется параметр `AstraMode` в конфигурационном файле `/etc/apache2/apache2.conf`.

По умолчанию режим включен, а параметр `AstraMode` в конфигурационном файле отсутствует, что соответствует значению `AstraMode on`.

Если не требуется использование политик управления доступом пользователей, подключаемых к данному серверу, авторизацию возможно выключить, добавив строку `AstraMode off` в конфигурационном файле `/etc/apache2/apache2.conf`.

ВНИМАНИЕ! При выключенной авторизации пользователей Apache2 осуществляет все запросы к своим ресурсам посредством только одной системной учетной записи (по умолчанию `www-data`), которая в случае настроенной политики мандатного контроля целостности имеет уровень целостности `1`.

10.3. Настройка авторизации

Настройка сквозной аутентификации и авторизации для сервера и клиента, работающих в рамках ЕПП, описана в 10.4. Если не настроена аутентификация через Kerberos,

то для всех ресурсов должна использоваться аутентификация и авторизация через PAM, при этом будет использоваться пользовательская БД, прописанная в настройках ОС. Для выполнения аутентификации и авторизации через PAM должен быть установлен пакет `libapache2-mod-authnz-pam` и выполнена следующая команда от имени учетной записи администратора:

```
a2enmod authnz_pam
```

В конфигурационных файлах виртуальных хостов web-сервера Apache2 указать:

```
AuthType Basic
AuthName "PAM authentication"
AuthBasicProvider PAM
AuthPAMService apache2
Require valid-user
```

Логин и пароль пользователя будут передаваться от пользователя к серверу в открытом виде с использованием метода аутентификации Basic. Для корректного функционирования авторизации через PAM пользователю, от которого работает web-сервер (по умолчанию `www-data`), необходимо выдать права на чтение информации из БД пользователей и сведений о метках безопасности:

```
usermod -a -G shadow www-data
setfacl -d -m u:www-data:r /etc/parsec/macdb
setfacl -R -m u:www-data:r /etc/parsec/macdb
setfacl -m u:www-data:rx /etc/parsec/macdb
```

Если установлен модуль web-сервера Apache2 `auth_kerb` из пакета `libapache2-mod-auth-kerb` для аутентификации через Kerberos в соответствии с 10.4, выключить его использование при помощи команды:

```
a2dismod auth_kerb
```

Для передачи в http-заголовке текущего иерархического уровня конфиденциальности и текущих неиерархических категорий конфиденциальности пользователя может быть сконфигурирован модуль Apache2 `mod_headers`. Для этого необходимо:

1) в конфигурационном файле `/etc/apache2/apache2.conf` добавить строку:

```
Header set MyHeader "%m %c"
```

где `%m` — место подстановки текущего иерархического уровня конфиденциальности;
`%c` — текущих неиерархических категорий конфиденциальности;

2) включить модуль, выполнив команду:

```
a2enmod headers
```

3) перезапустить сервер Apache2:

```
systemctl restart apache2
```

Сервер для PAM-аутентификации использует сценарий PAM, содержащийся в конфигурационном файле `/etc/pam.d/apache2`. PAM-сценарий включает `common-auth` и

common-account. По умолчанию в ОС для фиксации числа неверных попыток входа пользователей применяется PAM-модуль pam_tally. Использование pam_tally в секции auth в файле /etc/pam.d/common-auth обеспечивает увеличение счетчика неверных попыток входа пользователя при начале процесса аутентификации. Для корректной работы данного механизма необходимо разрешить пользователю www-data запись в /var/log/faillog, выполнив команду:

```
setfacl -m u:www-data:rw /var/log/faillog
```

Выполнить перезапуск сервера:

```
systemctl restart apache2
```

10.4. Настройка для работы в ЕПП

10.4.1. Настройка для работы со службой FreeIPA

Для обеспечения работы web-сервера Apache2 со службой FreeIPA следует произвести настройку web-сервера и контроллера домена FreeIPA. Порядок действий по настройке описан в 8.3.11.

10.4.2. Настройка для работы со службой ALD

Для обеспечения совместной работы web-сервера Apache2 с ALD необходимо:

1) наличие в системе, на которой работает web-сервер, установленного пакета клиента ALD — ald-client;

2) разрешение имен должно быть настроено таким образом, чтобы имя системы разрешалось, в первую очередь, как полное имя (например, myserver.example.ru);

3) клиент ALD должен быть настроен на используемый ALD домен согласно 8.2.3;

4) в системе должен быть установлен модуль web-сервера Apache2 auth_kerb из пакета libapache2-mod-auth-kerb.

Наличие модуля web-сервера Apache2 auth_kerb предоставляет возможность организации совместной работы с ALD с использованием для аутентификации пользователей посредством Kerberos метода GSSAPI.

Для проведения операций по настройке ALD и администрированию Kerberos необходимо знание паролей администраторов ALD и Kerberos.

Для обеспечения возможности работы web-сервера Apache2 с ALD необходимо:

1) если установлен в соответствии с 10.3 модуль аутентификации через PAM web-сервера Apache2 authnz_pam, выключить его при помощи команды:

```
a2dismod authnz_pam
```

2) активировать модуль web-сервера Apache2 auth_kerb при помощи команды:

```
a2enmod auth_kerb
```

3) в конфигурационных файлах виртуальных хостов web-сервера Apache2 в секции <Directory>, для которой настраивается доступ пользователей ЕПП, указать:

```
AuthType Kerberos
KrbAuthRealms REALM
KrbServiceName HTTP/server.my_domain.org
Krb5Keytab /etc/apache2/keytab
KrbMethodNegotiate on
KrbMethodK5Passwd off
require valid-user
```

4) создать в БД ALD с помощью утилиты администрирования ALD принципала, соответствующего настраиваемому web-серверу Apache2. Принципал создается с автоматически сгенерированным случайным ключом:

```
ald-admin service-add HTTP/server.my_domain.org
```

5) ввести созданного принципала в группу служб mac, используя следующую команду:

```
ald-admin sgroup-svc-add HTTP/server.my_domain.org
--sgroup=mac
```

6) создать файл ключа Kerberos для web-сервера Apache2 с помощью утилиты администрирования ALD ald-client, используя следующую команду:

```
ald-client update-svc-keytab HTTP/server.my_domain.org
   --ktfile="/etc/apache2/keytab"
```

Полученный файл должен быть доступен web-серверу Apache2 по пути, указанному в конфигурационном параметре Krb5Keytab (в данном случае /etc/apache2/keytab). Права доступа к этому файлу должны позволять читать его пользователю, от имени которого работает web-сервер Apache2 (как правило, владельцем файла назначается пользователь www-data);

7) сменить полученного на предыдущем шаге владельца файла keytab на пользователя www-data, выполнив следующую команду:

```
chown www-data /etc/apache2/keytab
```

8) сделать файл /etc/apache2/keytab доступным на чтение для остальных пользователей:

```
chmod 644 /etc/apache2/keytab
```

9) перезапустить web-сервер Apache2, выполнив команду:

```
systemctl restart apache2
```

Браузер пользователя должен поддерживать аутентификацию negotiate. В последних версиях браузера Konqueror данная поддержка присутствует автоматически. В браузере Mozilla Firefox в настройках, доступных по адресу about:config, необходимо указать, для каких серверов доступна аутентификация negotiate. Для выполнения данной настройки

следует задать маски доменов или в общем случае http- и https-соединения в качестве значений параметра `network.negotiate-auth.trusted-uris`, вставив, например, значения `http://`, `https://`.

При необходимости обеспечения сквозной аутентификации из скриптов с другими службами, например, серверу postgresql, в конфигурационном файле виртуального хоста следует дополнительно указать:

```
KrbSaveCredentials on
```

В браузере Mozilla Firefox в настройках задать в качестве значений параметра `network.negotiate-auth.delegation-uris` маски доменов которым можно передавать данные для сквозной аутентификации. В запускаемых скриптах выставить переменную окружения `KRB5CCNAME`. Например, для PHP это будет выглядеть так:

```
putenv("KRB5CCNAME=".$_SERVER['KRB5CCNAME']);
```

11. ЗАЩИЩЕННАЯ ГРАФИЧЕСКАЯ ПОДСИСТЕМА

Защищенная графическая подсистема в составе ОС функционирует с использованием графического сервера Xorg.

По умолчанию в графическую подсистему встроена мандатная защита.

Для установки пакетов графической подсистемы следует в процессе работы программы установки ОС отметить в окне «Выбор программного обеспечения» строку «Рабочий стол Fly». В этом случае рабочий стол Fly установится с настройками по умолчанию, и в процессе загрузки установленной системы после окончания работы системного загрузчика произойдет переход к окну графического входа в систему: пакеты `fly-dm` (запуск серверной части системы) и `fly-qdm` (поддержка графического интерфейса). После завершения процедуры аутентификации на экране монитора появится графический рабочий стол.

11.1. Конфигурирование менеджера окон и рабочего стола в зависимости от типа сессии

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 «слабых» систем или удаленных терминалов можно создавать режим `fly-light` и т.д.

Для создания нового режима необходимо добавить файл (файлы) сессии с расширением `desktop` в `/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`, `fly-mobile` и т.д.). Данная переменная является именем ярлыка сессии из `/usr/share/fly-dm/sessions` (но без расширения `.desktop`), которая указывает на тип сессии. Например:

`DESKTOP_SESSION=fly` — десктопный

`DESKTOP_SESSION=fly-tablet` — планшетный

`DESKTOP_SESSION=fly-mobile` — мобильный

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/`:

`apprc`

```
apprc.fly-mini
apprc.fly-mobile
apprc.fly-tablet
en.fly-wmrc
en.fly-wmrc.fly-mini
en.fly-wmrc.fly-mobile
en.fly-wmrc.fly-tablet
en.miscrc
en.miscrc.fly-mini
en.miscrc.fly-mobile
en.miscrc.fly-tablet
keyshortcutrc
keyshortcutrc.fly-mini
keyshortcutrc.fly-mobile
keyshortcutrc.fly-tablet
ru_RU.UTF-8.fly-wmrc
ru_RU.UTF-8.fly-wmrc.fly-mini
ru_RU.UTF-8.fly-wmrc.fly-mobile
ru_RU.UTF-8.fly-wmrc.fly-tablet
ru_RU.UTF-8.miscrc
ru_RU.UTF-8.miscrc.fly-mini
ru_RU.UTF-8.miscrc.fly-mobile
ru_RU.UTF-8.miscrc.fly-tablet
sessrc
sessrc.fly-mini
sessrc.fly-mobile
sessrc.fly-tablet
theme/default.themerc
theme/default.themerc.fly-mini
theme/default.themerc.fly-tablet
theme/default.themerc.fly-mobile
```

Также есть конфигурационный файл `fly-wmrc.mini`, который служит для совместимости и включает все файлы с расширением `*.fly-mini`. Названия этих файлов определяют их назначение, а в комментариях в файлах приведены особенности использования.

Если использовались файлы типа:

```
~/.fly/*rc
~/.fly/theme/*rc
```

/usr/share/fly-wm/*rc

/usr/share/fly-wm/theme/*rc

то необходимо переделать формирование имени конфигурационного файла. Например, это сделано в утилитах fly-admin-theme, fly-admin-hotkeys, fly-admin-winprops и др.

В ярлыках в полях NotShowIn и OnlyShowIn можно использовать имена типов сессий (fly, fly-tablet, fly-mobile и т.д.). Функция FlyDesktopEntry::isDisplayable() из libflycore изменена с учетом нахождения в сессии какого-либо типа ($DESKTOP_SESSION), также в libflycore добавлены:

```
const char * flySessionName()
const char * flySessionConfigSuffix()
```

Используя имена типов сессий в NotShowIn и OnlyShowIn, можно скрывать/показывать определенные ярлыки из меню «Пуск», панели задач или автозапуска (в зависимости от текущего режима).

Если у какой-либо Qt-программы есть сохраняемые/восстанавливаемые параметры, «чувствительные» к типу сессии (планшет, десктоп и т.д.), то программа будет иметь такие параметры в отдельных экземплярах для каждого типа сессии, добавляя, например, суффиксы $DESKTOP_SESSION к именам параметров.

11.2. Рабочий стол как часть экрана

В файлах *themerc (прежде всего в ~/.fly/theme/current.themerc) можно задавать параметры FlyDesktopWidth и FlyDesktopHeight, которые определяют размер (в пикселях) рабочего стола на экране. Это может быть полезно, например, для:

- деления широкоформатного монитора на две части: с рабочим столом и свободной областью, куда можно перетаскивать окна;
- для задания области рабочего стола только на левом мониторе в двухмониторной конфигурации с Xinerama.

11.3. Удаленный вход по протоколу XDMCP

По умолчанию в системе удаленный вход по протоколу XDMCP запрещен. Чтобы его разрешить необходимо:

1) в файле /etc/X11/fly-dm/Xaccess заменить localhost на символ *;

2) в файле /etc/X11/fly-dm/fly-dmrc убедиться, что Enable=true:

```
...
[Xdmcp]
Enable=true
...
```

11.4. Автоматизация входа в систему

Для включения автоматизации входа пользователя в систему на разных разрешенных ему уровнях секретности с последующим легким переключением между такими входами необходимо в секции `[Service]` файла `/lib/systemd/system/fly-dm.service` задать переменную:

```
...
Environment=DM_LOGIN_AUTOMATION=value
...
```

Затем на рабочих столах пользователя создать, например, следующие ярлыки:

- ярлык для запуска или перехода в сессию с меткой `0:0:0x0:0x0`:

```
[Desktop Entry]
Name = session 0
Name[ru] = Сессия 0
Type = Application
NoDisplay = false
Exec = fly-dmctl maclogin user password 0:0:0x0:0x0
Icon = ledgreen
X-FLY-IconContext = Actions
Hidden = false
Terminal = false
StartupNotify = false
```

- ярлык для запуска или перехода в сессию с меткой `1:0:0x0:0x0`:

```
[Desktop Entry]
Name = session 1
Name[ru] = Сессия 1
Type = Application
NoDisplay = false
Exec = /usr/bin/fly-dmctl maclogin user password 1:0:0x0:0x0
Icon = ledyellow
X-FLY-IconContext = Actions
Hidden = false
Terminal = false
StartupNotify = false
```

- ярлык для запуска или перехода в сессию с меткой `2:0:0x0:0x0`:

```
[Desktop Entry]
Name = session 2
Name[ru] = Сессия 2
Type = Application
```

```
NoDisplay = false
Exec = fly-dmctl maclogin user password  2:0:0x0:0x0
Icon = ledred
X-FLY-IconContext = Actions
Hidden = false
Terminal = false
StartupNotify = false
```

С помощью ярлыков данного типа пользователь сможет максимально легко переключаться между сессиями с разными метками безопасности, предварительно разрешенными пользователю администратором системы.

11.5. Рабочий стол Fly

В состав рабочего стола Fly входит оконный менеджер и графические утилиты, которые могут быть использованы для администрирования ОС. Большинство утилит представляет собой графические оболочки соответствующих утилит командной строки.

Основные графические утилиты для настройки и администрирования системы приведены в таблице 57.

Таблица 57

Утилита	Описание
`fly-admin-autostart` «Автозапуск»	Установки приложений, запускаемых автоматически при загрузке рабочего стола
`fly-admin-dm` «Вход в систему»	Настройка графического входа в систему
`fly-admin-hotkeys` «Горячие клавиши Fly»	Запуск редактора горячих клавиш для настройки соответствия между сочетаниями клавиш и действиями
`fly-admin-date` «Дата и время»	Просмотр установленного времени, даты, часового пояса, календаря, изменение формата отображения времени на системных часах, даты и времени на всплывающем сообщении при наведении курсора мыши на системные часы в области уведомлений на панели задач
`fly-admin-time` «Синхронизация времени»	Синхронизация времени с сервером времени
`fly-admin-grub2` «Загрузчик GRUB2»	Графическая утилита настройки загрузчика ОС GRUB 2
`systemdgenie` «Инициализация системы»	Графическая утилита управления службой Systemd
«Менеджер пакетов Synaptic»	Графическая утилита установки пакетов
`fly-admin-mouse` «Мышь»	Настройка кнопок мыши и скорости перемещения курсора

Продолжение таблицы 57

`gufw` «Настройка межсетевого экрана»	Программа настройки межсетевого экрана UFW (Uncomplicated Firewall)
`fly-admin-screen` «Настройка монитора»	Настройка размера изображения, разрешения, частоты обновления и других параметров монитора
`fly-brightness` «Настройка яркости Fly»	Программа для настройки яркости в планшетном режиме
`fly-admin-reflex` «Обработка «горячего» подключения»	Настройка реакций при подключении устройств в процессе работы
`fly-orientation` «Ориентация экрана»	Настройка ориентации экрана
`fly-admin-theme` «Оформление Fly»	Настройка обоев, тем, шрифтов, экрана блокировки и других элементов рабочего стола
`fly-menuedit` «Панель быстрого запуска»	Добавление и удаление программ из панели быстрого запуска
`fly-menuedit` «Меню Пуск»	Добавление и удаление программ из меню «Пуск»
`fly-admin-center` «Панель управления»	Централизованный доступ к графическим утилитам настройки и администрирования системы
`fly-admin-winprops` «Параметры окон»	Настройка поведения и внешнего вида окон рабочего стола
`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 локальной политикой безопасности и управление ЕПП. Позволяет управлять: пользователями, группами и настройками и атрибутами (мандатным управлением доступом пользователя, параметрами протоколирования, привилегиями, политикой срока действия пароля, политикой блокировки); базами данных Parsec (аудитом, мандатными атрибутами и привилегиями); политикой создания пользователей; настройками безопасности (устанавливать параметры монтирования для очистки блоков памяти при их освобождении, настраивать очистку разделов страничного обмена при выключении системы); параметрами подключения внешних устройств (учитывать носители и управлять их принадлежностью, протоколированием и мандатными атрибутам
`fly-mimeapps` «Приложения для типов файлов»	Просмотр доступных приложений и установка приложения по умолчанию для типов файлов, установка команды для запуска обозревателя и для создания вложений почтового клиента. Примечание. Утилита запускается только с аргументом `-d`

Продолжение таблицы 57

`fly-admin-printer` «Принтеры»	Программа «Менеджер печати Fly» для настройки печати в графическом режиме
`fly-xkbmap` «Раскладка клавиатуры»	Настройка раскладок клавиатуры
`fly-admin-policykit-1` «Санкции PolicyKit-1»	Управление санкциями Policykit
`fly-admin-session` «Сессии Fly»	Настройки для сессий рабочего стола
`nm-connection-editor` «Сетевые соединения»	Настройки сетевых соединений (по умолчанию при загрузке системы выполняется автозапуск программы)
`fly-admin-network` «Параметры сети»	Управление автозапуском сетевых служб
`fly-admin-alternatives` «Системные альтернативы»	Управление системой альтернатив дистрибутивов, основанных на Debian
`fly-admin-kiosk` «Системный киоск»	Управление ограничением среды
`fly-start-panel` «Меню-панель «Пуск»	Запуск меню-панели «Пуск», адаптированной для планшетного и мобильного режимов
«Установка принтеров, факсов и сканеров HP»	Графическая утилита установки новых устройств HP
`fly-admin-fonts` «Шрифты»	Просмотр и импорт системных шрифтов
`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 РАМ
`ksystemlog` «Просмотр системных журналов Ksystemlog»	Просмотр журнала расширенной системы протоколирования
`system-config-audit` «Конфигурация аудита»	Управление аудитом и редактирование правил аудита
`fly-sosreport` «Центр системных отчетов»	Программа сбора данных о конфигурации системы и работе подсистем для последующей диагностики ошибок
`fly-run` «Запуск приложения»	Запуск программы или осуществление доступа к ресурсу из командной строки, в т.ч. от имени другого пользователя и/или с другими мандатными атрибутами
«Контроль целостности файлов»	Графическая утилита программы afick монитора изменений файлов системы
`fly-admin-device-manager` «Менеджер устройств»	Получение информации об устройствах, доступных в системе, а также для настройки некоторых из них
`fly-admin-usbip` «Сервис удаленных USB-накопителей»	Программа захвата и перенаправления физических USB-носителей на удаленные компьютеры для их использования по сети

Продолжение таблицы 57

`fly-admin-format` «Форматирование внешнего носителя»	Программа форматирования USB-носителей
`fly-admin-iso` «Запись ISO образа на USB носитель»	Программа записи iso-образа на USB-носитель
`fly-fm` «Менеджер файлов»	Просмотр папок рабочего стола и элементов ФС, выполнение основных функций управления файлами, монтирование и размонтирование ФС носителей доступных устройств хранения данных, обращение к сетевым Samba-ресурсам, работа с архивами, выполнение кодирующего преобразования
`qbat` «Монитор батарей QBat»	Программа QBat для мониторинга батарей электропитания
`fly-print-monitor` «Монитор печати»	Обзор и управление системой печати из области уведомлений на панели задач
`fly-find` «Поиск файлов»	Поиск файлов и каталогов
`fly-admin-int-check` «Проверка целостности системы»	Проверка целостности системы для рабочего стола Fly
`fly-admin-marker` «Редактор маркеров»	Настройка маркировки печати сопроводительной надписи документов
`fly-print-station` «Управление печатью документов»	Программа маркировки выводимых на печать документов
«Редактор разделов Gparted»	Создание, перераспределение или удаление системных разделов ОС
`ksysguard` «Системный монитор»	Отслеживание системных параметров
`fly-term` «Терминал Fly»	Эмулятор консольного режима
`mc` «Менеджер файлов MC»	Просмотр папок и элементов ФС, выполнение основных функций управления файлами, монтирование и размонтирование ФС носителей доступных устройств хранения данных, обращение к сетевым ресурсам, работа с архивами, выполнение кодирующего/раскодирующего преобразования
`ark` «Работа с архивами Ark»	Программа для работы с архивами файлов
`kgpg` «Управление ключами KGpg»	Программа управления ключами GPG
`fly-admin-ald-server` «Доменная политика безопасности»	Настройка службы контроллера домена ALD
`fly-admin-ald-client` «Настройка ALD клиента Fly»	Ввод клиентского компьютера в существующий домен ALD

Окончание таблицы 57

Утилита	Описание
`fly-admin-ad-client` «Настройка клиента Active Directory Fly»	Ввод клиентского компьютера в существующий 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 (System Security Services Daemon (SSSD))
`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»	Программа подготовки 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»	Смена пароля
`fly-scan` «Сканирование»	Установка сканера и сканирование с сохранением изображения (запускается с аргументом `--noautoselect`)
`fly-su` «Подмена пользователя»	Выполнение команды от имени другого пользователя
`fly-hexedit` «Двоичный редактор»	Редактор данных в двоичных файлах
`fly-jobviewer` «Очередь печати»	Просмотр и управление очередью заданий на печать
`fly-astra-update` «Установка обновлений»	Программа установки обновлений
`fly-admin-repo` «Редактор репозиториев»	Программа для создания и управления репозиториями

Не все из приведенных в таблице 57 утилит устанавливаются по умолчанию при установке ОС. Описание утилит доступно в электронной справке. Вызов электронной справки осуществляется с помощью ярлыка «Помощь», размещенного на рабочем столе, а также путем нажатия комбинации клавиш **<Alt+F1>** или путем нажатия клавиши **<F1>** в активном окне графической утилиты.

11.6. Блокировка экрана при бездействии

Блокировка экрана при неактивности задается в конфигурационных файлах типов сессий `*themerc*`, расположенных в каталоге пользователя `/home/<имя_пользователя>/.fly/theme/`, следующими параметрами:

```
ScreenSaverDelay=0/<время_неактивности_в_секундах>
LockerOnSleep=true/false
LockerOnDPMS=true/false
LockerOnLid=true/false
LockerOnSwitch=true/false
```

При этом имена актуальных для сессии пользователя конфигурационных файлов начинаются с `current`, а файлы, имена которых начинаются с `default`, используются для создания и восстановления файлов `current`.

При создании учетной записи пользователя и его первом входе конфигурационные файлы `default.themerc*` копируются из каталога `/usr/share/fly-wm/theme/` в каталог пользователя `/home/<имя_пользователя>/.fly/theme/`.

Пользователю доступно управление блокировкой экрана своей сессии при неактивности из графической утилиты `fly-admnin-theme` (см. электронную справку).

Администратору для управления блокировкой экрана пользователей, в т.ч. централизованного, доступен конфигурационный файл `/usr/share/fly-wm/theme.master/themerc`. В файле указываются строки:

```
[Variables]
ScreenSaverDelay=0/<время_неактивности_в_секундах>
LockerOnSleep=true/false
LockerOnDPMS=true/false
LockerOnLid=true/false
LockerOnSwitch=true/false
```

При входе пользователя в сессию после считывания параметров из конфигурационных файлов пользователя проверяется наличие файла `/usr/share/fly-wm/theme.master/themerc` с секцией `[Variables]`. При наличии файла из него считываются параметры, и считанные параметры переопределяют аналогичные параметры, считанные ранее из конфигурационных файлов пользователя.

В ОС выполняется мониторинг каталога `/usr/share/fly-wm/theme.master/` и файла `/usr/share/fly-wm/theme.master/themerc`. При создании/изменении файла `/usr/share/fly-wm/theme.master/themerc` срабатывает механизм мониторинга и параметры из файла считываются и применяются к текущим сессиям всех пользователей.

Каталог `/usr/share/fly-wm/theme.master/` может являться разделяемым ресурсом.

Пользователю не доступна возможность переопределить параметры, заданные в `/usr/share/fly-wm/theme.master/themerc`.

11.7. Мандатное управление доступом

Мандатная защита, встроенная в рабочий стол Fly и устанавливаемая по умолчанию вместе с ОС, позволяет администратору задавать отдельно для каждого пользователя разрешенный диапазон иерархических уровней конфиденциальности и неиерархических категорий конфиденциальности. Для этой цели следует использовать графическую утилиту `fly-admin-smc`.

После того как пользователь, для которого установлены возможные иерархические уровни конфиденциальности и неиерархические категории конфиденциальности, отличные от нуля, войдет в систему, ему будет предложено установить конкретный иерархический уровень конфиденциальности и конкретную неиерархическую категорию конфиденциальности для данной сессии в пределах разрешенных диапазонов. Выбранные значения этих параметров отображаются на цветном индикаторе с числом внутри, расположенном в области уведомлений на панели задач. Для получения информационного сообщения следует навести курсор на индикатор (рис. 7).

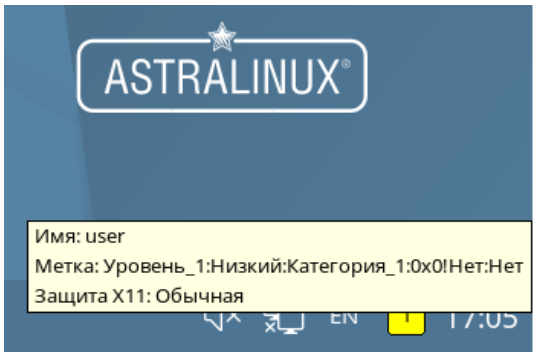

Рис. 7

12. ЗАЩИЩЕННЫЙ КОМПЛЕКС ПРОГРАММ ПЕЧАТИ И МАРКИРОВКИ ДОКУМЕНТОВ

Одной из основных служб, предоставляемых ОС, является служба печати, модифицированная для маркировки документов и позволяющая осуществлять печать документов в соответствии с требованиями, предъявляемыми к защищенным ОС.

Защищенный комплекс программ печати и маркировки документов CUPS обеспечивает:

- управление заданиями, выдаваемыми на печать;
- выполнение команд администратора печати;
- предоставление информации о состоянии принтеров локальным и удаленным программам;
- выдачу информационных сообщений пользователям;
- маркировку выводимых на печать документов.

12.1. Устройство системы печати

Состав защищенного комплекса программ печати и маркировки приведен в таблице 58.

Таблица 58

Название	Пакет	Описание
CUPS	`cups-daemon`	Сервер печати. Обрабатывает запросы от пользователя и выполняет запуск служебных программ
`fly-admin-printer`	`fly-admin-printer`	Графическая утилита для настройки принтеров и сервера печати CUPS, а также управления очередью печати. При установленном пакете `fly-admin-printer-mac` позволяет маркировать документы и настраивать метки безопасности принтера
`fly-print-monitor`	`fly-print-monitor`	Графическая утилита для отслеживания состояния принтеров и сервера печати
`fly-jobviewer`	`fly-jobviewer`	Графическая утилита для управления очередью печати. При установленном пакете `fly-admin-printer-mac` позволяет также маркировать документы

Окончание таблицы 58

Название	Пакет	Описание
psmarker	parsec-cups	Программа для маркировки документов в формате PostScript. Модифицирует исходный файл задания, добавляя в него маркеры. Запускается с помощью CUPS
fonarik	parsec-cups	Программа для создания файла маркировки на обратной стороне последнего листа. Запускается с помощью CUPS
markerdb	parsec-cups	Программа для записи журнала маркировки. Вызывается из CUPS после завершения задания маркировки. В процессе маркировки не участвует
pdfhelper	parsec-cups	Программа для определения размера и ориентации PDF-документов. Запускается с помощью CUPS перед маркировкой
markjob	parsec-cups-client	Инструмент для маркировки документов в консольном режиме
libfly-admin-printer-mac	libfly-admin-printer-mac2	Библиотека с функциями маркировки и просмотра журнала для графических клиентов. Упрощает взаимодействие с сервером CUPS при выполнении задач маркировки
fly-admin-printer-mac	fly-admin-printer-mac	Утилита, добавляющая функции маркировки в графические утилиты fly-admin-printer и fly-jobviewer
fly-print-station	fly-print-station	Графическая утилита для маркировки документов
fly-admin-marker	fly-admin-marker	Графическая утилита для редактирования шаблонов маркировки. Работает только локально вместе с сервером печати CUPS

Планировщик — это сервер, который управляет списком доступных принтеров и направляет задания на печать, используя подходящие фильтры и выходные буферы (backends).

Файлами конфигурации являются:

- файл конфигурации сервера;

- файлы определения принтеров и классов;
- типы MIME и файлы правил преобразования;
- файлы описания PostScript-принтеров (PPD).

Конфигурационный файл сервера похож на файл конфигурации web-сервера и определяет все свойства управления доступом.

Файлы описания принтеров и классов перечисляют доступные очереди печати и классы. Классы принтеров — наборы принтеров. Задания, посланные классу принтеров, направляются к первому доступному принтеру данного класса.

Очередь печати — механизм, который позволяет буферизовать и организовать задания, посылаемые на принтер. Необходимость организации такого механизма обуславливается тем, что принтер является медленно действующим устройством, и задания не могут быть распечатаны мгновенно. Очевидно, что в многопользовательской среде возникает конкуренция со стороны пользователей при доступе к принтерам, поэтому задания необходимо располагать в очереди. Для этого используется буферный каталог `/var/spool/cups/`.

Файлы типов MIME перечисляют поддерживаемые MIME-типы (`text/plain`, `application/postscript` и т. д.) и правила для автоматического обнаружения формата файла. Они используются сервером для определения поля `Content-Type` для GET- и HEAD-запросов и обработчиком запросов IPP, чтобы определить тип файла.

Правила преобразования MIME перечисляют доступные фильтры. Фильтры используются, когда задание направляется на печать, таким образом, приложение может послать файл удобного (для него) формата системе печати, которая затем преобразует документ в требуемый печатный формат. Каждый фильтр имеет относительную «стоимость», связанную с ним, и алгоритм фильтрования выбирает набор фильтров, который преобразует файл в требуемый формат с наименьшей общей «стоимостью».

Файлы PPD описывают возможности всех типов принтеров. Для каждого принтера имеется один PPD-файл. Файлы PPD для не-PostScript-принтеров определяют дополнительные фильтры посредством атрибута `cupsFilter` для поддержки драйверов принтеров.

В ОС стандартным языком описания страниц является язык PostScript. Большинство прикладных программ (редакторы, браузеры) генерируют программы печати на этом языке. Когда необходимо напечатать ASCII-текст, программа печати может быть ASCII-текстом. Имеется возможность управления размером шрифтов при печати ASCII-текста. Управляющая информация используется для контроля доступа пользователя к принтеру и аудита печати. Также имеется возможность печати изображений в форматах GIF, JPEG, PNG, TIFF и документов в формате PDF.

Фильтр — программа, которая читает из стандартного входного потока или из файла, если указано его имя. Все фильтры поддерживают общий набор параметров, включающий

имя принтера, идентификатор задания, имя пользователя, имя задания, число копий и параметры задания. Весь вывод направляется в стандартный выходной поток.

Фильтры предоставлены для многих форматов файлов и включают, в частности, фильтры файлов изображения и растровые фильтры PostScript, которые поддерживают принтеры, не относящиеся к типу PostScript. Иногда несколько фильтров запускаются параллельно для получения требуемого формата на выходе.

Программа `backend` — это специальный фильтр, который отправляет печатаемые данные устройству или через сетевое соединение. В состав системы печати включены фильтры для поддержки устройств, подключаемых с помощью параллельного и последовательного интерфейсов, а также шины USB.

Клиентские программы используются для управления заданиями и сервером печати.

Управление заданиями включает:

- формирование;
- передачу серверу печати;
- мониторинг и управление заданиями в очереди на печать.

Управление сервером включает:

- запуск/остановку сервера печати;
- запрещение/разрешение постановки заданий в очередь;
- запрещение/разрешение вывода заданий на принтер.

В общем случае вывод данных на принтер происходит следующим образом:

1) программа формирует запрос на печать задания к серверу печати;

2) сервер печати принимает подлежащие печати данные, формирует в буферном каталоге файлы с содержимым задания и файлы описания задания, при этом задание попадает в соответствующую очередь печати;

3) сервер печати просматривает очереди печати для незанятых принтеров, находит в них задания и запускает конвейер процессов, состоящий из фильтров и заканчивающийся выходным буфером, информация из которого поступает в принтер посредством драйверов ОС;

4) контроль и мониторинг процесса печати выполняется с помощью программ `lpq`, `lpc`, `lprm`, `lpstat`, `lpmove`, `cancel`, а также с помощью графической утилиты `fly-admin-printer`.

Система печати ОС решает следующие задачи:

1) монопольная постановка задания в очередь на печать. Данная функция предполагает невозможность вывода документа на печать в обход системы печати;

2) маркировка каждого напечатанного листа. Каждый лист сопровождается автоматической маркировкой (учетными атрибутами документа).

ВНИМАНИЕ! Для обеспечения нормальной работы пользователя с сетевыми службами должна быть явно задана его метка безопасности (диапазон уровней конфиденциальности, категорий конфиденциальности и уровни целостности) с помощью соответствующих утилит, даже если ему не доступны уровни и категории выше 0. Дополнительная информация приведена в документе РУСБ.10152-02 97 01-1.

12.2. Установка

Основные компоненты защищенного комплекса программ печати и маркировки документов устанавливаются автоматически при установке ОС.

Графическая утилита маркировки документов `fly-print-station` устанавливается путем выполнения команды:

```
apt install fly-print-station
```

Графическая утилита для редактирования шаблонов маркировки `fly-admin-marker` устанавливается путем выполнения команды:

```
apt install fly-admin-marker
```

В случае необходимости возможно вручную установить защищенный комплекс программ печати и маркировки документов, выполнив команду:

```
apt install fly-print-station parsec-cups-client fly-admin-printer
     fly-admin-printer-mac fly-admin-marker
```

12.3. Настройка

Настройка защищенного комплекса программ печати и маркировки документов выполняется путем корректировки конфигурационных файлов `/etc/cups/cupsd.conf` и `/etc/cups/cups-files.conf`. Копии конфигурационных файлов, устанавливаемые вместе с пакетом, размещаются в `/usr/share/cups` (файлы `cupsd.conf.default` и `cups-files.conf.default`). Данные файлы могут использоваться при необходимости вернуть комплекс программ печати и маркировки документов в исходное состояние.

Предварительная настройка защищенного комплекса программ печати и маркировки документов должна выполняться от имени учетной записи администратора с использованием механизма `sudo`.

Ряд действий по администрированию CUPS (добавление и удаление принтеров, изменение политики для принтера, установка мандатных атрибутов для принтера) может выполняться от имени пользователя, входящего в локальную группу администраторов печати `lpadmin`. Данная группа администраторов печати указана в качестве значения параметра `SystemGroup` в файле `/etc/cups/cups-files.conf`.

Основные пользовательские настройки содержатся в файлах конфигурации `client.conf` и `~/.cups/lpoptions`.

12.3.1. Настройка для работы с локальной базой безопасности

Для разрешения серверу CUPS удаленно принимать задания и команды необходимо от имени учетной записи администратора через механизм sudo:

1) выполнить следующие команды:

```
cupsctl --remote-admin --share-printers --remote-any
cupsctl ServerAlias=*
cupsctl DefaultPolicy=authenticated
cupsctl DefaultAuthType=Basic
```

2) осуществить перезапуск сервера системы печати, выполнив команды:

```
systemctl stop cups
systemctl start cups
```

В файле конфигурации клиента client.conf должен быть задан один параметр ServerName, определяющий имя сервера печати, например:

```
ServerName computer.domain
```

12.3.2. Настройка для работы в ЕПП

Для работы системы печати в ЕПП необходимо выполнение следующих условий:

1) наличие в системах, на которых функционируют сервер и клиенты системы печати, установленного пакета клиента FreeIPA — client.domain.ipa;

2) разрешение имен должно быть настроено таким образом, чтобы имя системы разрешалось, в первую очередь, как полное имя (например, myserver.example.ru);

3) клиент FreeIPA должен быть настроен на используемый FreeIPA домен в соответствии с 8.3.6.

Для проведения операций по настройке FreeIPA и администрированию Kerberos необходимо знание паролей администраторов FreeIPA и Kerberos.

12.3.2.1. Настройка сервера печати

Для выполнения действий по управлению принтерами и очередями печати необходимо создать в FreeIPA учетную запись администратора печати ipa_print_admin и добавить ее в локальную группу администраторов печати на сервере печати, выполнив команду:

```
sudo gpasswd -a ipa_print_admin lpadmin
```

Для обеспечения совместной работы сервера печати с FreeIPA необходимо:

1) на контроллере домена добавить службу ipp:

```
ipa service-add ipp/printserver.domain.ipa
```

2) выгрузить таблицу ключей для службы:

```
sudo ipa-getkeytab -p ipp/printserver.domain.ipa@DOMAIN.IPA -k
    /tmp/ipp.keytab
```

3) если сервер CUPS установлен не на контроллере домена, то необходимо перенести таблицу ключей на printserver.domain.ipa в /tmp:

```
sudo scp /tmp/ipp.keytab admin@printserver.domain.ipa:/tmp
```

4) на компьютере, где установлен сервер CUPS, добавить ключи в хранилище Kerberos:

```
admin@printserver:~$ sudo ktutil
ktutil:  rkt /tmp/ipp.keytab
ktutil:  wkt /etc/krb5.keytab
ktutil:  l
slot KVNO Principal
---- ---- ---------------------------------------------------------------------
1    1           ipp/printserver.domain.ipa@DOMAIN.IPA
2    1           ipp/printserver.domain.ipa@DOMAIN.IPA
ktutil:  q

admin@printserver:~$ sudo klist -kte /etc/krb5.keytab
Keytab name: FILE:/etc/krb5.keytab
KVNO Timestamp           Principal
---- ------------------- ------------------------------------------------------
1 18.05.2020 12:10:17 host/printserver.domain.ipa@DOMAIN.IPA
    (aes256-cts-hmac-sha1-96)
1 18.05.2020 12:10:17 host/printserver.domain.ipa@DOMAIN.IPA
    (aes128-cts-hmac-sha1-96)
1 18.05.2020 13:10:27 ipp/printserver.domain.ipa@DOMAIN.IPA
    (aes256-cts-hmac-sha1-96)
1 18.05.2020 13:10:27 ipp/printserver.domain.ipa@DOMAIN.IPA
    (aes128-cts-hmac-sha1-96)
```

ВНИМАНИЕ! С включенной проверкой целостности администрировать сервер печати можно только локально.

Для настройки сервера печати CUPS от имени учетной записи администратора с использованием механизма sudo:

1) выполнить следующие команды:

```
cupsctl --remote-admin --share-printers --remote-any
cupsctl ServerAlias=*
cupsctl DefaultPolicy=authenticated
cupsctl ServerName=printserver.domain.ipa
cupsctl MacEnable=On
cupsctl DefaultAuthType=Negotiate
```

2) в конфигурационном файле /etc/cups/cups-files.conf раскомментировать строку:

```
MarkerUser ipp
```

3) в конфигурационном файле /etc/cups/cupsd.conf заменить строки:

```
Port 631
Listen /var/run/cups/cups.sock
```

на строку:

```
Listen 0.0.0.0:631
```

4) осуществить перезапуск сервера системы печати, выполнив команду:

```
systemctl restart cups
```

ВНИМАНИЕ! В конфигурационном файле защищенного сервера печати из состава изделия /etc/cups/cupsd.conf не допускается установка значения None параметра DefaultAuthType (отключение аутентификации) и внесение изменений в параметры политики PARSEC, не соответствующих эксплуатационной документации.

Далее выполнить вход на сервере печати от имени учетной записи, входящей в локальную группу администраторов печати на сервере печати lpadmin, и настроить принтеры. Настройка принтеров может быть выполнена с использованием утилиты fly-admin-printer (см. электронную справку). После запуска утилиты необходимо указать, что для выполнения привилегированных действий не используется учетная запись root, и затем выполнять действия по настройке.

12.3.2.2. Настройка клиента системы печати

Для настройки клиента системы печати необходимо:

1) создать конфигурационный файл /etc/cups/client.conf;

2) задать в конфигурационным файле /etc/cups/client.conf для параметра ServerName в качестве значения имя сервера системы печати, например, printserver.domain.ipa.

12.4. Настройка принтера и управление печатью

12.4.1. Общие положения

Установку и настройку принтера следует производить после завершения установки и первоначальной настройки ОС.

При печати через локальный сервер печати данные сначала формируются на локальном сервере, как для любой другой задачи печати, после чего посылаются на принтер, подключенный к данному компьютеру.

Системные каталоги, определяющие работу системы печати ОС, содержат файлы, которые не являются исполняемыми и содержат необходимую для драйвера принтера

информацию (используемое физическое устройство, удаленный компьютер и принтер для удаленной печати):

- /etc/cups/printers.conf — содержит описания принтеров в ОС;
- /etc/cups/ppd/<имя_очереди>.ppd — содержит описания возможностей принтера, которые используются при печати заданий и при настройке принтеров;
- /var/log/cups/error_log — поступает протокол работы принтера. В этом файле могут находиться сообщения об ошибках сервера печати или других программ системы печати;
- /var/log/cups/access_log — регистрируются все запросы к серверу печати;
- /var/log/cups/page_log — поступают сообщения, подтверждающие успешную обработку страниц задания фильтрами и принтером.

Далее термин «принтер» в настоящем подразделе используется для обозначения принтера, соответствующего одной записи в файле /etc/cups/printers.conf. Под термином «физический принтер» подразумевается устройство, с помощью которого производится вывод информации на бумажный носитель. В файле /etc/cups/printers.conf может быть несколько записей, описывающих один физический принтер различными способами.

12.4.2. Команды управления печатью

В систему печати ОС включены файлы, предоставляющие командный интерфейс пользователя в стиле BSD и System V. Перечень файлов приведен в таблице 59.

Таблица 59

Файл	Описание
/usr/bin/lpr	Постановка заданий в очередь. Совместима с командой lpr системы печати BSD UNIX
/usr/bin/lp	Постановка заданий в очередь. Совместима с командой lp системы печати System V UNIX
/usr/bin/lpq	Просмотр очередей печати
/usr/sbin/lpc	Управление принтером. Является частичной реализацией команды lpc системы печати BSD UNIX
/usr/bin/lprm	Отмена заданий, поставленных в очередь на печать
/usr/sbin/cupsd	Сервер печати
/usr/sbin/lpadmin	Настройка принтеров и классов принтеров
/usr/sbin/lpmove	Перемещение задания в другую очередь
/usr/bin/fly-admin-printer	Настройка системы печати, установка и настройка принтеров, управление заданиями

Описание данных команд приведено на страницах руководства man.

CUPS предоставляет утилиты командной строки для отправления заданий и проверки состояния принтера. Команды `lpstat` и `lpc status` также показывают сетевые принтеры (`принтер@сервер`), когда разрешен обзор принтеров.

Команды администрирования System V предназначены для управления принтерами и классами. Средство администрирования `lpc` поддерживается только в режиме чтения для проверки текущего состояния очередей печати и планировщика.

Остановить работу службы печати можно с помощью команды:

```
systemctl stop cups
```

Запустить службу печати можно с помощью команды:

```
systemctl start cups
```

12.4.2.1. lp

С помощью команды `lp` выполняется передача задачи принтеру, т. е. задача ставится в очередь на печать. В результате выполнения этой команды файл передается серверу печати, который помещает его в каталог `/var/spool/cups/`.

12.4.2.2. lpq

Команда `lpq` предназначена для проверки очереди печати, используемой LPD, и вывода состояния заданий на печать, указанных при помощи номера задания либо системного идентификатора пользователя, которому принадлежит задание (владельца задания). Команда выводит для каждого задания имя его владельца, текущий приоритет задания, номер задания и размер задания в байтах, без параметров выводит состояние всех заданий в очереди.

12.4.2.3. lprm

Команда `lprm` предназначена для удаления задания из очереди печати. Для определения номера задания необходимо использовать команду `lpq`. Удалить задание может только его владелец или администратор печати.

12.4.2.4. lpadmin

Команда `lpadmin` также используется для настройки принтера в ОС.

Ее запуск с параметром `-p` используется для добавления или модификации принтера:

```
/usr/sbin/lpadmin -p printer [параметры]
```

Основные параметры команды `lpadmin` приведены в таблице 60.

Таблица 60

Параметр	Описание
`-c class`	Добавляет названный принтер к классу принтеров `class`. Если класс не существует, то он создается

Окончание таблицы 60

Параметр	Описание
`-m model`	Задает стандартный драйвер принтера, обычно файл `PPD`. Файлы `PPD` обычно хранятся в каталоге `/usr/share/cups/model/`. Список всех доступных моделей можно вывести командой `lpinfo` с параметром `-m`
`-r class`	Удаляет указанный принтер из класса `class`. Если в результате класс становится пустым, он удаляется
`-v device-uri`	Указывает адрес устройства для связи с принтером
`-D info`	Выдает текстовое описание принтера
`-E`	Разрешает использование принтера и включает прием заданий
`-L location`	Выводит расположение принтера
`-P ppd-file`	Указывает локальный файл `PPD` для драйвера принтера

Для данной команды существуют также параметры по регулированию политики лимитов и ограничений по использованию принтеров и политики доступа к принтерам.

Запуск команды `lpadmin` с параметром `-x` используется для удаления принтера:

```
/usr/sbin/lpadmin -x printer
```

12.4.3. Графическая утилита настройки сервера печати

Утилита `fly-admin-printer` предназначена для настройки печати в графическом режиме. Позволяет в режиме администратора печати устанавливать, настраивать и удалять принтеры и классы принтеров, а также настраивать сервер печати и управлять заданиями на печать. В режиме обычного пользователя позволяет устанавливать настройки печати и параметры принтера, а также управлять заданиями на печать (удалять, приостанавливать, возобновлять печать и устанавливать отложенную печать). Для вызова привилегированных действий запрашивается авторизация. Подробную информацию по использованию утилиты `fly-admin-printer` см. в электронной справке.

Для установки драйверов принтеров производства Hewlett Packard рекомендуется использовать утилиту `hp-setup`.

12.5. Маркировка документа

При поступлении задания на печать считывается метка безопасности сетевого соединения и копируется в атрибут задания `mac-job-mac-label`.

При печати задания с нулевой меткой безопасности (нулевой иерархический уровень конфиденциальности, без неиерархических категорий конфиденциальности и нулевой неиерархический уровень целостности) маркировка листов не выполняется и печать осуществляется в штатном режиме. При этом атрибут принтера `mac-printer-mac-min` должен быть нулевым, иначе задание на печать будет завершено с ошибкой.

При печати задания с ненулевой меткой безопасности оно принудительно переводится сервером печати в состояние «отложено» до проведения привилегированным пользователем маркировки выводимых на печать листов. Файлы заданий (в каталоге `/var/spool/cups`) маркируются согласно мандатному контексту документа.

ВНИМАНИЕ! Задания печати администратора печати (учетная запись, входящая в группу `lpadmin`), отправляются сразу на печать без задержки на маркировку.

Для печати заданий с ненулевой меткой безопасности необходимо соответствующим образом настроить принтер, а также маркеры печати (при необходимости). Описание настройки принтеров, маркеров печати и порядка маркировки приведено в РУСБ.10152-02 97 01-1.

ВНИМАНИЕ! Мандатный контекст задания должен находиться в диапазоне между минимальным и максимальным мандатным контекстом принтера, на который отправлено задание. Если метка безопасности задания ненулевая, но не попадает в множество разрешенных меток для данного принтера, заданных атрибутами `mac-printer-mac-min` и `mac-printer-mac-max`, то задание на печать будет завершено с ошибкой.

ВНИМАНИЕ! Контроль уровня целостности работает только для локальных соединений (через Unix Domain Socket). Любому соединение по TCP/IP будет присваиваться низкий уровень целостности. Поэтому для возможности печати с удаленного компьютера необходимо разрешить принтеру печать с низкой целостностью.

Маркировка осуществляется «наложением» маркеров с учетными атрибутами документа, включающими:

- уровень конфиденциальности документа;
- номер экземпляра;
- количество листов в экземпляре;
- дату вывода документа на печать;
- номер каждого входящего документа;
- имя исполнителя;
- имя пользователя, производившего печать.

Система печати является инвариантной по отношению к приложениям, которые обращаются к службе печати. Это означает, что приложения, выводящие на печать, должны учитывать маркировку листов и оставлять для этого свободное место. В противном случае маркеры могут наложиться на фрагменты печатаемой информации.

Маркировка задания выполняется в пять этапов:

1) блокировка задания. Если задание в процессе маркировки другим пользователем или соединением, то выдается ошибка;

2) проверка наличия и установка атрибутов задания;

3) с помощью переменных маркировки запрос у пользователя атрибутов задания;

4) выставление атрибутов задания, полученных на предыдущем этапе;

5) непосредственно маркировка задания.

Маркировка документов при использовании локальной базы осуществляется от имени пользователя, входящего в группу `lpmac`. Если группа отсутствует в системе, она должна быть создана.

Маркировка документов в ЕПП осуществляется от имени доменного пользователя, входящего в локальную группу `lpmac` на сервере печати. Для добавления пользователя в локальную группу `lpmac` необходимо на сервере печати выполнить команду:

```
sudo gpasswd -a ipa_marker_user lpmac
```

Маркировка документа выполняется с помощью инструмента командной строки `markjob`, описанного в 12.6, или с помощью графической утилиты `fly-print-station`, описанной в 12.7.

12.6. Маркировка документа в командной строке

Маркировка документа в командной строке выполняется с помощью инструмента `markjob`. Инструмент `markjob` требует наличия утилиты `lpq`, входящей в состав пакета `cups-bsd`.

Для маркировки с помощью `markjob` выполнить команду:

```
markjob -m
```

или

```
markjob
```

Подробное описание инструмента `markjob` приведено в `man markjob`.

В процессе работы инструмента `markjob` у пользователя запрашиваются настроенные атрибуты для маркера печати, например:

- `mac-inv-num` — инвентарный номер;
- `mac-owner-phone` — телефон исполнителя;
- `mac-workplace-id` — идентификатор рабочего места;
- `mac-distribution` — список рассылки.

При вводе списка рассылки адреса разделяются символом «^». Если в значении списка рассылки используется пробел, то значение атрибута необходимо взять в кавычки целиком.

Пример

Выдается запрос на ввод списка рассылки:

```
Enter mac-distribution - Distribution list, addresses separated by '^':
```

Ввести список рассылки:

```
"В дело^В адрес"
```

После выполнения маркировки в очереди формируются два дополнительных задания в состояние «отложено»: первое (с меньшим номером) представляет собой промаркированный документ, а второе (с большим номером) — размещаемую на обороте последнего листа документа маркировку.

Для печати промаркированного документа необходимо возобновить печать первого отложенного задания. Затем на обороте последнего листа документа печатается маркировка путем возобновления выполнения второго дополнительного задания.

При выполнении маркировки от имени пользователя, входящего в группу `lpmac`, возможно получение сообщения:

```
Невозможно выполнить запрос: запрещено
```

В данном случае необходимо выполнить команду `id` от имени пользователя, выполняющего маркировку, и повторно запустить инструмент `markjob`.

Если ведение журнала маркировки включено, то после завершения задания данные маркировки будут записаны в него. Описание журнала маркировки приведено в 12.9.

12.7. Графическая утилита управления печатью

Для печати документов с маркировкой используется графическая утилита `fly-print-station`. Утилита предназначена для управления заданиями на печать, для маркировки документов, отправленных на печать, а также для просмотра журнала маркировки.

Описание использования утилиты приведено в электронной справке.

12.8. Маркировка нескольких экземпляров документа

Для печати нескольких экземпляров документа с ненулевой меткой безопасности пользователь должен отправить на печать только одну копию документа.

Затем пользователь, осуществляющий маркировку, должен выполнить следующую последовательность действий:

1) получить номер задания для маркировки, выполнив команду:

```
lpq -a
```

2) задать число копий для печати, выполнив команду:

```
lpattr -j <номер_задания> -s copies=<число_копий>
```

3) произвести маркировку с помощью инструмента `markjob` или графической утилиты `fly-print-station`.

После выполнения маркировки в очереди формируются по два дополнительных задания для каждого экземпляра документа, располагаемых в очереди последовательно. Первое (с меньшим номером) представляет собой промаркированный экземпляр документа, а второе (с большим номером) — маркировку, размещаемую на обороте последнего

листа экземпляра документа. Для печати экземпляра документа необходимо возобновить выполнение первого соответствующего ему задания, что приведет к печати промаркированного экземпляра документа. Затем на обороте последнего листа экземпляра документа печатается маркировка посредством возобновления выполнения второго соответствующего экземпляру документа дополнительного задания.

12.9. Журнал маркировки

При установки для параметра `MacJournal` значение `on` в конфигурационном файле `/etc/cups/cupsd.conf` ведется журнал маркировки. По умолчанию журнал записывается в базу данных SQLITE `/var/spool/cups/parsec/marking-journal.sqlite`.

Включить журнал маркировки возможно путем редактирования конфигурационного файла или выполнив команду от имени администратора:

```
cupsctl MacJournal=On
```

Просмотр журнала возможен с использованием графической утилиты `fly-print-station` и графической утилиты `fly-admin-printer` с установленным плагином `fly-admin-printer-mac`.

13. ЗАЩИЩЕННАЯ СИСТЕМА УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ

В качестве защищенной СУБД в составе ОС используется PostgreSQL, доработанная в соответствии с требованием интеграции с ОС в части мандатного управления доступом к информации. Описание реализации мандатного управления доступом к информации в защищенной СУБД PostgreSQL приведено в РУСБ.10152-02 97 01-1.

СУБД PostgreSQL предназначена для создания и управления реляционными БД и предоставляет многопользовательский доступ к расположенным в них данным.

Данные в реляционной БД хранятся в отношениях (таблицах), состоящих из строк и столбцов. При этом единицей хранения и доступа к данным является строка, состоящая из полей, идентифицируемых именами столбцов. Кроме таблиц существуют другие объекты БД (виды, процедуры и т. п.), которые предоставляют доступ к данным, хранящимся в таблицах.

Подробное описание работы с защищенной СУБД приведено в документе РУСБ.10152-02 95 01-2.

14. ЗАЩИЩЕННЫЙ КОМПЛЕКС ПРОГРАММ ЭЛЕКТРОННОЙ ПОЧТЫ

В качестве защищенного комплекса программ электронной почты используется сервер электронной почты, состоящий из агента передачи электронной почты (Mail Transfer Agent, MTA) Exim4, агента доставки электронной почты (Mail Delivery Agent, MDA) Dovecot и клиента электронной почты (Mail User Agent, MUA) Mozilla Thunderbird, доработанных для реализации следующих дополнительных функциональных возможностей:

- интеграции с ядром ОС и базовыми библиотеками для обеспечения разграничения доступа;
- реализации мандатного управления доступом к почтовым сообщениям;
- автоматической маркировки создаваемых почтовых сообщений, отражающих уровень их конфиденциальности;
- регистрации попыток доступа к почтовым сообщениям.

Агент передачи электронной почты использует протокол SMTP и обеспечивает решение следующих задач:

- доставку исходящей почты от авторизованных клиентов до сервера, который является целевым для обработки почтового домена получателя;
- прием и обработку почтовых сообщений доменов, для которых он является целевым;
- передачу входящих почтовых сообщений для обработки агентом доставки электронной почты.

Агент доставки электронной почты предназначен для решения задач по обслуживанию почтового каталога и предоставления удаленного доступа к почтовому ящику по протоколу IMAP.

Клиент электронной почты — это прикладное ПО, устанавливаемое на рабочем месте пользователя и предназначенное для получения, создания, отправки и хранения сообщений электронной почты пользователя.

14.1. Состав

Защищенный комплекс программ электронной почты состоит из следующих пакетов:

- `exim4-daemon-heavy` — агент передачи сообщений Exim4. Пакет `exim4-daemon-light` не поддерживает работу с классификационными метками, отличными от `0:0`;
- `dovecot-imapd` — агент доставки сообщений Dovecot. Работает только по протоколу IMAP, протокол POP3 отключен. Серверная часть защищенного комплекса программ электронной почты использует в качестве почтового хранилища MailDir

(mailbox не поддерживает работу с классификационными метками, отличными от 0:0);

- thunderbird — клиент электронной почты Mozilla Thunderbird.

14.2. Настройка серверной части

Настройки по умолчанию:

1) прием почтовых сообщений по протоколу SMTP только от MUA из доменов relay-domens и из подсети;

2) отправка почтовых сообщений по протоколу SMTP в соответствии с DNS;

3) хранение локальной почты в MailDir в /var/mail/%u, где %u — локальная часть адресата;

4) выдача локальных почтовых сообщений по протоколу IMAP.

ВНИМАНИЕ! Для обеспечения нормальной работы пользователя с сетевыми службами должна быть явно задана его классификационная метка (диапазон уровней конфиденциальности и категорий конфиденциальности) с помощью соответствующих утилит, даже если ему не доступны уровни и категории выше 0. Дополнительная информация приведена в документе РУСБ.10152-02 97 01-1.

ВНИМАНИЕ! Редактирование конфигурационных файлов и выполнение команд по настройке необходимо выполнять от имени учетной записи администратора с использованием механизма sudo.

14.2.1. Настройка агента доставки сообщений

Настройка агента доставки сообщений Dovecot осуществляется путем правки конфигурационного файла /etc/dovecot/dovecot.conf и конфигурационных файлов в каталоге /etc/dovecot/conf.d.

В файле /etc/dovecot/dovecot.conf необходимо задать список интерфейсов, с которых будут приниматься соединения, и установить протокол IMAP, например:

```
protocols = imap
listen =  192.168.2.55
```

Для настройки аутентификации с использованием PAM в конфигурационном файле /etc/dovecot/conf.d/10-auth.conf необходимо установить:

```
disable_plaintext_auth = no
auth_mechanisms = plain
```

Агент доставки сообщений Dovecot для PAM-аутентификации использует сценарий PAM, содержащийся в конфигурационном файле /etc/pam.d/dovecot. PAM-сценарий для Dovecot включает common-auth и common-account. По умолчанию в ОС для фиксации числа неверных попыток входа пользователей применяется PAM-модуль pam_tally. Использование pam_tally в секции auth в файле /etc/pam.d/common-auth обеспечивает

увеличение счетчика неверных попыток входа пользователя при начале процесса аутентификации. Для сброса счетчика неверных попыток входа пользователя после успешной аутентификации в Dovecot необходимо в сценарий PAM, содержащийся в конфигурационном файле `/etc/pam.d/dovecot`, добавить использование `pam_tally` в секции `account`. PAM-сценарий для Dovecot будет иметь следующий вид:

```
@include common-auth
@include common-account
@include common-session
account required pam_tally.so
```

В случае когда SSL не будет использоваться в конфигурационном файле `/etc/dovecot/conf.d/10-ssl.conf`, необходимо установить:

```
ssl = no
```

Для настройки встроенного в MDA Dovecot сервера SASL, к которому будет обращаться MTA Exim4 для аутентификации пользователей, в конфигурационном файле `/etc/dovecot/conf.d/10-master.conf` в секцию `service auth` необходимо добавить:

```
unix_listener auth-client {
mode = 0600
user = Debian-exim
}
```

Перезапустить MDA Dovecot, выполнив команду:

```
systemctl restart dovecot
```

14.2.2. Настройка агента передачи сообщений

Для настройки агента передачи сообщений Exim4 требуется инициировать переконфигурирование пакета `exim4-config`, для этого выполнить в эмуляторе терминала команду:

```
sudo dpkg-reconfigure exim4-config
```

В появившемся окне настройки для указанных ниже параметров необходимо установить следующие значения:

1) «Общий тип почтовой конфигурации:» — выбрать пункт «интернет-сайт; прием и отправка почты напрямую, используя SMTP»;

2) «Почтовое имя системы:» — ввести имя домена;

3) «IP-адреса, с которых следует ожидать входящие соединения:» — ввести IP-адрес сервера;

4) «Другие места назначения, для которых должна приниматься почта:» — ввести имя домена;

5) «Домены, для которых доступна релейная передача почты:» — оставить пустым;

6) «Машины, для которых доступна релейная передача почты:» — оставить пустым;

7) «Сокращать количество DNS-запросов до минимума:» — выбрать пункт «Нет»;

8) «Метод доставки локальной почты:» — выбрать пункт «Maildir формат в /var/mail/»;

9) «Разделить конфигурацию на маленькие файлы:» — выбрать пункт «Да».

Если возникла необходимость изменить расположение каталога `/var/spool/exim4`, убедиться, что каталог `exim4`, подкаталоги `db input`, `msglog`, файлы `db/retry`, `db/retry.lockfile` имеют метки безопасности `0:::EHOLE`. Если это не так, установить соответствующие метки на указанные каталоги и файлы командами:

```
sudo cd new_dir
sudo pdpl-file 0:::EHOLE . db input msglog db/retry db/retry.lockfile
```

Если возникла необходимость изменить расположение каталога хранилища локальной почты `/var/mail`, убедиться, что на новый каталог установлены права `1777`, если это не так, установить командой:

```
sudo chmod 1777 new_dir
```

Для нормальной работы `exim4-daemon-heavy` необходимо в каталоге `/var/mail` удалить файл с именем пользователя, созданного при установке системы.

В каталоге `/etc/exim4/conf.d/auth` необходимо создать файл с именем `05_dovecot_login` и следующим содержимым:

```
dovecot_plain:
        driver = dovecot
        public_name = plain
        server_socket = /var/run/dovecot/auth-client
        server_set_id = $auth1
```

Для запрета отправки писем без аутентификации необходимо в конфигурационном файле `/etc/exim4/conf.d/acl/30_exim4-config_check_rcpt` в начало секции `acl_check_rcpt` добавить строки:

```
deny
    message = "Auth required"
    hosts = *:+relay_from_hosts
    !authenticated = *
```

Настройку сквозной авторизации для сервера и клиента, работающих в рамках ЕПП, см. в 14.4.

Настроить автоматический запуск службы MTA Exim4, выполнив команду:

```
sudo systemctl enable exim4
```

Перезапустить MTA Exim4, выполнив команду:

```
systemctl restart exim4
```

14.3. Настройка клиентской части

Первичное создание для пользователя учетной записи сервера электронной почты в MUA Mozilla Thunderbird должно производиться с нулевой классификационной меткой (значение уровня конфиденциальности `0`, категорий конфиденциальности нет). Далее для каждой конкретной классификационной метки (значение уровня и набор категорий) создание учетной записи необходимо повторить.

При создании учетной записи пользователя в MUA Mozilla Thunderbird необходимо выбрать тип используемого сервера входящей почты IMAP.

При настройке учетной записи установить в параметрах сервера и параметрах сервера исходящей почты:

- «Защита соединения:» — из выпадающего списка выбрать «Нет»;
- «Метода аутентификации:» — выбрать «Обычный пароль».

14.4. Настройка для работы в ЕПП

Для обеспечения совместной работы сервера электронной почты с ALD и FreeIPA должны быть установлены:

- агент передачи сообщений Exim4 — из пакета `exim4-daemon-heavy`;
- агент доставки сообщений Dovecot — из пакета `dovecot-imapd`;
- пакет `dovecot-gssapi` поддержки GSSAPI-аутентификации для MDA Dovecot;
- клиент Mozilla Thunderbird — из пакета `thunderbird`.

14.4.1. Настройка для работы со службой FreeIPA

Для настройки совместной работы сервера электронной почты с FreeIPA должно быть предварительно выполнено:

- установлен сервер контроллера домена FreeIPA (например, домен `astra.mta`);
- на отдельном компьютере установлен почтовый сервер, введенный в домен FreeIPA (например, сервер `exim1.astra.mta` с IP-адресом `192.168.32.3`).

14.4.1.1. Настройка почтового сервера

Установить на почтовом сервере необходимые пакеты следующей командой:

```
sudo apt install exim4-daemon-heavy dovecot-imapd dovecot-gssapi
```

При установке пакетов `dovecot-imapd` и `dovecot-gssapi` создается файл `/etc/dovecot/conf.d/10-master.conf`. В секции `service auth` этого файла необходимо добавить следующие строки:

```
unix_listener auth-client {
mode = 0600
user = Debian-exim
}
```

После внесения изменений следует выполнить команду для реконфигурации Exim:

```
sudo dpkg-reconfigure exim4-config
```

В появившемся окне настройки для указанных ниже параметров необходимо установить следующие значения:

1) «Общий тип почтовой конфигурации:» — выбрать пункт «доставка только локальной почты; доступа к сети нет»;

2) «Почтовое имя системы:» — ввести имя домена, например, «astra.mta»;

3) «IP-адреса, с которых следует ожидать входящие соединения:» — указать IP-адрес сервера или оставить поле пустым;

4) «Другие места назначения, для которых должна приниматься почта:» — ввести имя домена, например, «astra.mta»;

5) «Машины, для которых доступна релейная передача почты:» — указать IP-адреса, например, «192.168.32.0/24»;

6) «Сокращать количество DNS-запросов до минимума:» — выбрать пункт «Нет»;

7) «Метод доставки локальной почты:» — выбрать пункт «Maildir формат в /var/mail/»;

8) «Разделить конфигурацию на маленькие файлы:» — выбрать пункт «Да».

В журнале Exim (файл `/var/log/exim4/paniclog`) могут появляться сообщения об ошибках вида:

```
Failed to create spool file /var/spool/exim4//input//1jb2ok-00031u-5R-D:
Operation not permitted
```

В этом случае следует исправить права доступа к каталогу `/var/spool/exim4`:

```
sudo chown -R Debian-exim:Debian-exim /var/spool/exim4/
```

14.4.1.2. Регистрация почтовых служб на контроллере домена

На контроллере домена необходимо добавить принципалов служб:

- `imap/exim1.astra.mta@ASTRA.MTA`
- `smtp/exim1.astra.mta@ASTRA.MTA`

Это можно сделать через web-интерфейс FreeIPA, перейдя «Идентификация — Службы» и нажав кнопку **[Добавить]** (см. рис. 8).

Также возможно из командной строки, предварительно получив полномочия администратора домена:

```
sudo kinit admin
sudo ipa service-add imap/exim1.astra.mta@ASTRA.MTA
sudo ipa service-add smtp/exim1.astra.mta@ASTRA.MTA
```

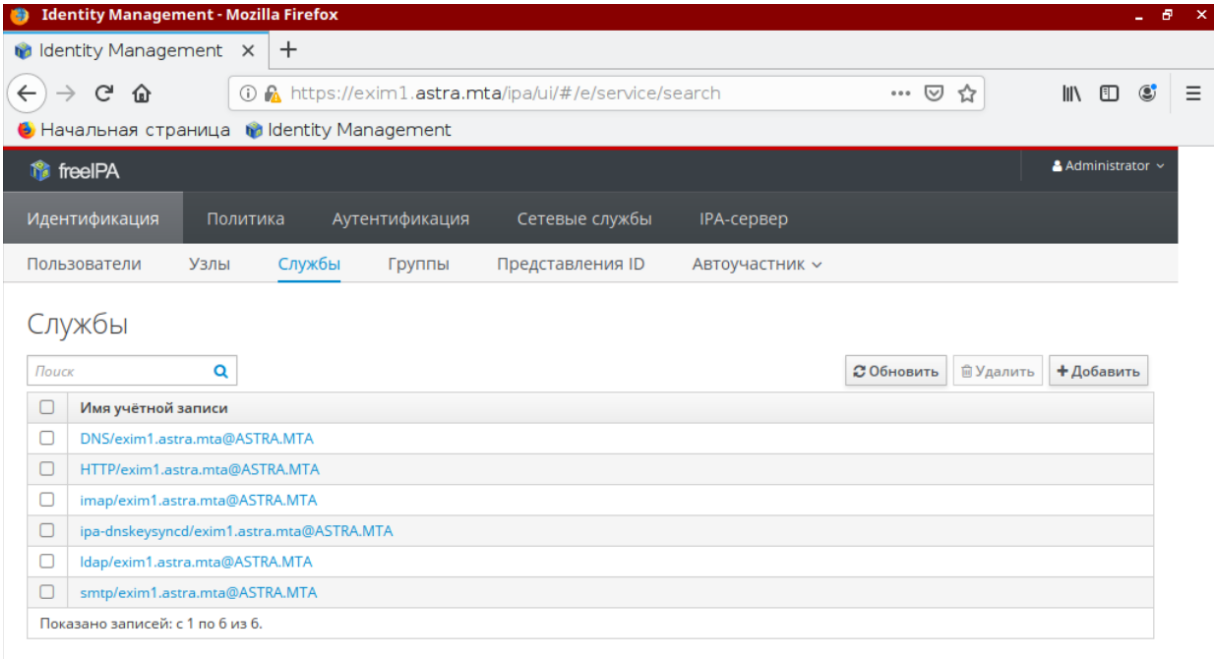

Рис. 8

14.4.1.3. Получение таблицы ключей на почтовом сервере

На почтовом сервере следует получить таблицу ключей для службы `imap`, затем добавить таблицу ключей для службы `smtp`:

```
sudo kinit admin
sudo ipa-getkeytab --principal=imap/exim1.astra.mta@ASTRA.MTA --keytab=/var/lib
     /dovecot/dovecot.keytab
sudo ipa-getkeytab --principal=smtp/exim1.astra.mta@ASTRA.MTA --keytab=/var/lib
     /dovecot/dovecot.keytab
```

Проверить полученную таблицу ключей:

```
sudo klist -k /var/lib/dovecot/dovecot.keytab
```

Вывод в терминале будет иметь следующий вид:

```
Keytab name: FILE:/var/lib/dovecot/dovecot.keytab
KVNO Principal
----------------------------------------------------------------------------
1 imap/exim1.astra.mta@ASTRA.MTA
1 imap/exim1.astra.mta@ASTRA.MTA
1 smtp/exim1.astra.mta@ASTRA.MTA
1 smtp/exim1.astra.mta@ASTRA.MTA
```

После этого следует выдать пользователю `dovecot` права на чтение файла ключа Kerberos:

```
sudo setfacl -m u:dovecot:x /var/lib/dovecot
sudo setfacl -m u:dovecot:r /var/lib/dovecot/dovecot.keytab
```

Далее убедиться, что в конфигурационном файле /etc/dovecot/dovecot.conf отключено использование протоколов POP3, и отключить неиспользуемые протоколы, оставив только IMAP:

```
protocols = imap
```

После этого следует выполнить настройки в конфигурационном файле /etc/dovecot/conf.d/10-auth.conf:

- для отключения передачи при аутентификации пароля открытым текстом установить:

```
disable_plaintext_auth = yes
```

- для настройки аутентификации посредством Kerberos с использованием метода GSSAPI установить:

```
auth_gssapi_hostname = exim1.astra.mta
auth_krb5_keytab = /var/lib/dovecot/dovecot.keytab
auth_mechanisms = gssapi
```

Затем перезапустить Dovecot:

```
sudo systemctl restart dovecot
```

14.4.1.4. Настройка авторизации через Kerberos

Для настройки аутентификации в Exim следует создать конфигурационный файл /etc/exim4/conf.d/auth/33_exim4-dovecot-kerberos-ipa со следующим содержимым:

```
dovecot_gssapi:
driver = dovecot
public_name = GSSAPI
server_socket = /var/run/dovecot/auth-client
server_set_id = $auth1
```

Далее запустить сервер Exim и разрешить его автоматический запуск после перезагрузки:

```
sudo systemctl start exim4
sudo systemctl enable exim4
```

После настройки авторизации через Kerberos в домене FreeIPA требуется настройка параметров почтового сервера (параметров пересылки почты) и настройка клиентской части на клиентах.

14.4.2. Настройка для работы со службой ALD

Предложенная конфигурация сервера электронной почты предоставляет возможность организации совместной работы с ALD с использованием для аутентификации поль-

зователей посредством Kerberos метода GSSAPI на основе встроенного в Dovecot сервера SASL.

Для обеспечения совместной работы сервера электронной почты, состоящего из перечисленных выше средств, с ALD необходимо выполнение следующих условий:

1) наличие в системах, на которых функционируют MTA, MDA и MUA, установленного пакета клиента ALD `ald-client`;

2) разрешение имен должно быть настроено таким образом, чтобы имя системы разрешалось, в первую очередь, как полное имя (например, `myserver.example.ru`);

3) клиент ALD должен быть настроен на используемый ALD домен (см. 8.2.3);

4) в процессе установки MTA Exim4 необходимо указать, что для хранения сообщений электронной почты должен быть использован формат `Maildir` в домашнем каталоге и конфигурация разделена на небольшие файлы.

Для проведения операций по настройке ALD и администрированию Kerberos необходимо знание паролей администраторов ALD и Kerberos.

14.4.2.1. Сервер

Для обеспечения работы сервера электронной почты, включающего MDA Dovecot, установленный из пакета `dovecot-imapd` и настроенный согласно 14.2.1, и MTA Exim4, установленный из пакета `exim4-daemon-heavy` и настроенный согласно 14.2.2, необходимо:

1) создать в БД ALD с помощью утилиты администрирования ALD принципала, соответствующего установленному MDA Dovecot. Принципал создается с автоматически сгенерированным случайным ключом:

```
ald-admin service-add imap/server.my_domain.org
```

2) ввести созданного принципала в группу служб `mac`, используя следующую команду:

```
ald-admin sgroup-svc-add imap/server.my_domain.org --sgroup=mac
```

3) ввести созданного принципала в группу служб `mail`, используя следующую команду:

```
ald-admin sgroup-svc-add imap/server.my_domain.org --sgroup=mail
```

4) создать файл ключа Kerberos для MDA Dovecot с помощью утилиты администрирования ALD `ald-client`, используя следующую команду:

```
ald-client update-svc-keytab imap/server.my_domain.org
   --ktfile="/var/lib/dovecot/dovecot.keytab"
```

5) создать в БД Kerberos принципала, соответствующего установленному MTA Exim4. Принципал создается с автоматически сгенерированным случайным ключом:

```
ald-admin service-add smtp/server.my_domain.org
```

6) ввести созданного принципала в группу служб `mac`, используя следующую команду:

```
ald-admin sgroup-svc-add smtp/server.my_domain.org --sgroup=mac
```

7) ввести созданного принципала в группу служб `mail`, используя следующую команду:

```
ald-admin sgroup-svc-add smtp/server.my_domain.org --sgroup=mail
```

8) создать файл ключа Kerberos для MTA Exim4 с помощью утилиты администрирования ALD `ald-client`, используя следующую команду:

```
sudo ald-client update-svc-keytab smtp/server.my_domain.org
  --ktfile="/var/lib/dovecot/dovecot.keytab"
```

9) предоставить пользователю `dovecot` права на чтение файл ключа Kerberos, выполнив команды:

```
sudo setfacl -m u:dovecot:x /var/lib/dovecot
sudo setfacl -m u:dovecot:r /var/lib/dovecot/dovecot.keytab
```

10) в конфигурационном файле `/etc/dovecot/dovecot.conf` отключить использование протоколов POP3, установив:

```
protocols = imap
```

11) в конфигурационном файле `/etc/dovecot/conf.d/10-auth.conf` установить:

```
auth_krb5_keytab = /var/lib/dovecot/dovecot.keytab
```

12) для отключения передачи при аутентификации пароля открытым текстом в конфигурационном файле `/etc/dovecot/conf.d/10-auth.conf` установить:

```
disable_plaintext_auth = yes
```

13) для настройки аутентификации посредством Kerberos с использованием метода GSSAPI в конфигурационном файле `/etc/dovecot/conf.d/10-auth.conf` установить:

```
auth_mechanisms = gssapi
auth_gssapi_hostname = server.my_domain.org
```

14) для настройки встроенного в MDA Dovecot сервера SASL, к которому будет обращаться MTA Exim4 для аутентификации пользователей, в конфигурационном файле `/etc/dovecot/conf.d/10-master.conf` в секцию `service auth` необходимо добавить:

```
unix_listener auth-client {
mode = 0600
user = Debian-exim
}
```

15) перезапустить MDA Dovecot, выполнив команду:

```
systemctl restart dovecot
```

16) для настройки аутентификации пользователей в MTA Exim4 посредством Kerberos с использованием метода GSSAPI и встроенного в Dovecot сервера SASL создать конфигурационный файл `/etc/exim4/conf.d/auth/33_exim4-dovecot-kerberos-ald` со следующим содержимым:

```
dovecot_gssapi:
driver = dovecot
public_name = GSSAPI
  server_socket = /var/run/dovecot/auth-client
server_set_id = $auth1
```

Если раннее MTA Exmin4 был настроен для использования PAM-аутентификации, то необходимо в каталоге `/etc/exim4/conf.d/auth` удалить файл с именем `05_dovecot_login`

17) для запрета отправки писем без аутентификации в конфигурационном файле `/etc/exim4/conf.d/acl/30_exim4-config_check_rcpt` в начало секции `acl_check_rcpt` добавить строки:

```
deny
    message = "Auth required"
    hosts = *:+relay_from_hosts
    !authenticated = *
```

18) перезапустить MTA Exim4, выполнив команду:

```
systemctl reload exim4
```

14.4.2.2. Клиент

Для обеспечения возможности работы MUA Mozilla Thunderbird с ЕПП необходимо создать учетную запись пользователя в ALD, например, при помощи команды:

```
ald-admin user-add user1
```

Первичное создание для пользователя `user1` учетной записи в MUA Mozilla Thunderbird должно производится с нулевой классификационной меткой (значение уровня конфиденциальности `0`, категорий конфиденциальности нет). Далее для каждой конкретной классификационной метки (значение уровня конфиденциальности и набор категорий конфиденциальности) создание учетной записи необходимо повторить.

При создании учетной записи пользователя в MUA Mozilla Thunderbird необходимо выбрать тип используемого сервера входящей почты IMAP.

При настройке учетной записи установить в параметрах сервера и параметрах сервера исходящей почты:

- «Защита соединения:» — из выпадающего списка выбрать «Нет»;
- «Метода аутентификации:» — выбрать «Kerberos/GSSAPI».

15. СРЕДСТВА ЦЕНТРАЛИЗОВАННОГО ПРОТОКОЛИРОВАНИЯ И АУДИТА

15.1. Аудит

Для аудита ОС могут использоваться системные лог-файлы различных служб и программ. Основное расположение этих файлов — системный каталог `/var/log`.

Аудит событий постановки/снятия с контроля целостности исполняемых модулей и файлов данных, а также событий неудачного запуска неподписанных файлов осуществляется в системном журнале `/var/log/kern.log` и `ksystemlog`.

Аудит событий создания/удаления/изменения настроек учетных записей пользователей и начала/окончания сеансов работы учетных записей пользователей осуществляется в системном журнале `/var/log/auth.log`.

Аудит событий изменения полномочий для учетных записей по доступу к информации осуществляется в системном журнале `/var/log/auth.log` и `ksystemlog`.

Аудит событий смены аутентифицирующей информации учетных записей осуществляется в системном журнале `/var/log/auth.log`.

Аудит событий выдачи печатных (графических) документов на бумажный носитель осуществляется в системных журналах `/var/log/cups/page_log` и `/var/spool/cups/parsec`.

Аудит отслеживания удаления журналов аудита `parsec` отображается первой записью в том же файле журнала `ksystemlog`.

Также в ОС для регистрации событий безопасности используется подсистема протоколирования, описание которой приведено в РУСБ.10152-02 97 01-1.

15.2. Подсистема регистрации событий

Подсистема регистрации событий включает следующие инструменты:

1) менеджер и маршрутизатор событий `syslog-ng` — принимает события из различных источников (события от `auditd`, файлы, прикладное ПО), проводит их обработку и, в зависимости от конфигурации, сохраняет в файл, отправляет по сети и т.д.;

2) модуль `syslog-ng-mod-astra` — модуль для `syslog-ng`, выполняющий дополнительную обработку и фильтрацию событий;

3) `astra-event-watcher` — демон уведомления пользователя о событиях, обработанных менеджером `syslog-ng`;

4) журнал событий `ksystemlog` — просмотр и анализ событий.

Установка выполняется командой:

```
sudo apt install syslog-ng syslog-ng-mod-python syslog-ng-mod-astra
astra-event-watcher
```

Работа модуля `syslog-ng-mod-astra` настраивается в конфигурационных файлах:

1) `/etc/astra-syslog.conf` — список регистрируемых событий;

2) `/var/cache/astra-syslog/` — каталог с файлами настроек по умолчанию для каждого события.

Модуль `syslog-ng-mod-astra` информацию о событиях регистрирует в файлах:

1) `/var/log/astra/events` — лог-файл в формате `json` регистрируемых событий (попытки запуска неподписанных файлов, успешная и неуспешная авторизация, данные о пользовательских сессиях и др.). Доступен для чтения только администратору;

2) `/var/log/astra/prevlogin<username>` — лог-файл формате `json` сводной статистики предыдущих входов в систему пользователя `<username>`. Включает данные о последней завершенной сессии пользователя, а также количество успешных и неуспешных входов пользователя со времени начала ведения статистики. Доступен для чтения только пользователю `<username>`.

Настройка отображения уведомлений демона `astra-event-watcher` выполняется в файле `/usr/share/knotifications5/astra-event-watcher.notifyrc`.

15.3. Средства централизованного протоколирования

Для решения задач централизованного протоколирования и анализа журналов аудита, а также организации распределенного мониторинга сети, жизнеспособности и целостности серверов используется программное решение Zabbix, реализованное на web-сервере Apache, СУБД (MySQL, Oracle, PostgreSQL, SQLite) и языке сценариев PHP.

Zabbix предоставляет гибкий механизм сбора данных. Все отчеты и статистика Zabbix, а также параметры настройки компонентов Zabbix доступны через web-интерфейс. В web-интерфейсе реализован следующий функционал:

- вывод отчетности и визуализация собранных данных;
- создание правил и шаблонов мониторинга состояния сети и узлов;
- определение допустимых границ значений заданных параметров;
- настройка оповещений;
- настройка автоматического реагирования на события безопасности.

15.3.1. Архитектура

Zabbix состоит из следующих основных программных компонентов:

1) сервер — является основным компонентом, который выполняет мониторинг, взаимодействует с прокси и агентами, вычисляет триггеры, отправляет оповещения. Является главным хранилищем данных конфигурации, статистики, а также оперативных данных;

2) агенты — разворачиваются на наблюдаемых системах для активного мониторинга за локальными ресурсами и приложениями и для отправки собранных данных серверу или прокси;

3) прокси — может собирать данные о производительности и доступности от имени сервера. Прокси является опциональной частью Zabbix и может использоваться для снижения нагрузки на сервер;

4) база данных — вся информация о конфигурации, а также собранные Zabbix данные, хранятся в базе данных;

5) web-интерфейс — используется для доступа к Zabbix из любого места и с любой платформы.

15.3.2. Сервер

Для установки сервера с СУБД PostgreSQL выполнить команду:

```
apt-get install zabbix-server-pgsql zabbix-frontend-php
```

Для создания базы данных сервера используются скрипты по созданию базы данных для PostgreSQL, например:

```
psql -U <username>
create database zabbix;
\q
cd database/postgresql
psql -U <username> zabbix < schema.sql
psql -U <username> zabbix < images.sql
psql -U <username> zabbix < data.sql
```

Далее необходимо импортировать исходную схему и данные сервера на PostgreSQL:

```
zcat /usr/share/doc/zabbix-server-pgsql/create.sql.gz | psql -U <username> zabbix
```

Для настройки базы данных сервера откорректировать конфигурационный файл `zabbix_server.conf`.

Пример

```
vi /etc/zabbix/zabbix_server.conf
DBHost=localhost
DBName=zabbix
DBUser=zabbix
DBPassword=<пароль>
```

В параметре `DBPassword` указывается пароль пользователя PosgreSQL.

Основные параметры конфигурационного файла сервера приведены в таблице 61.

Таблица 61

Параметр	Описание
AllowRoot	Разрешение серверу запускаться от имени пользователя root. Если не разрешено (значение «0») и сервер запускается от имени root, сервер попробует переключиться на пользователя zabbix. Не влияет, если сервер запускается от имени обычного пользователя. Значение по умолчанию — 0
CacheSize	Размер кэша конфигурации в байтах для хранения данных узлов сети, элементов данных и триггеров. Возможные значения от 128 КБ до 8 ГБ, значение по умолчанию — 8 МБ
CacheUpdateFrequency	Частота выполнения процедуры обновления кэша конфигурации, в секундах. Возможные значения от 1 до 3600 сек, значение по умолчанию — 60 сек
DBHost	Имя хоста базы данных. В случае пустой строки PostgreSQL будет использовать сокет. Значение по умолчанию — localhost
DBName	Обязательный параметр. Имя базы данных
DBPassword	Пароль к базе данных
DBPort	Порт базы данных, когда не используется localhost. Значение по умолчанию — 3306
DBSchema	Имя схемы
DBUser	Пользователь базы данных
HousekeepingFrequency	Частота выполнения автоматической процедуры очистки базы данных от устаревшей информации, в часах. Возможные значения от 0 до 24 ч, значение по умолчанию — 1

Сервер работает как демон. Для запуска сервера выполнить команду:

```
systemctl start zabbix-server
```

Соответственно для остановки, перезапуска и просмотра состояния сервера используются следующие команды:

```
systemctl stop zabbix-server
systemctl restart zabbix-server
systemctl status zabbix-server
```

ВНИМАНИЕ! Для работы сервера необходима кодировка UTF-8 иначе некоторые текстовые элементы данных могут быть интерпретированы некорректно.

В таблице 62 приведены основные параметры, используемые при управлении сервером.

Таблица 62

Параметр	Описание
-c --config <файл>	Путь к файлу конфигурации. Значение по умолчанию /usr/local/etc/zabbix_server.conf

Окончание таблицы 62

Параметр	Описание
`-R --runtime-control <параметр>`	Выполнение административных функций
`config_cache_reload`	Перезагрузка кэша конфигурации. Игнорируется, если кэш загружается в данный момент: `zabbix_server -c /usr/local/etc/zabbix_server.conf -R config_cache_reload`
`housekeeper_execute`	Запуск процедуры очистки базы данных. Игнорируется, если процедура очистки выполняется в данный момент `zabbix_server -c /usr/local/etc/zabbix_server.conf -R housekeeper_execute`

`log_level_increase[=<цель>]`	Увеличение уровня журналирования, действует на все процессы, если цель не указана. В качестве цели может быть указан идентификатор процесса, тип процесса или тип и номер процесса: `zabbix_server -c /usr/local/etc/zabbix_server.conf -R log_level_increase`
	`zabbix_server -c /usr/local/etc/zabbix_server.conf -R log_level_increase=1234`
	`zabbix_server -c /usr/local/etc/zabbix_server.conf -R log_level_increase=poller,2`

`log_level_decrease[=<цель>]`	Уменьшение уровня журналирования, действует на все процессы, если цель не указана. В качестве цели может быть указан идентификатор процесса, тип процесса или тип и номер процесса: `zabbix_server -c /usr/local/etc/zabbix_server.conf -R log_level_decrease="http poller"`

15.3.3. Агенты

Агенты могут выполнять пассивные и активные проверки.

При пассивной проверке агент отвечает на запрос от сервера или прокси.

При активной проверке агент получает от сервера перечень данных для мониторинга, затем осуществляет сбор данных согласно полученному перечню и периодически отправляет собранные данные серверу.

Выбор между пассивной и активной проверкой осуществляется выбором соответствующего типа элемента данных. Агент обрабатывает элементы данных типов «Zabbix агент» и «Zabbix агент (активный)».

Для установки агента в UNIX-системах выполнить команду:

```
apt-get install zabbix-agent
```

Основные параметры конфигурационного файла агента UNIX приведены в таблице 63.

Таблица 63

Параметр	Описание
AllowRoot	Разрешение серверу запускаться от имени пользователя root. Если не разрешено (значение «0») и сервер запускается от имени root, сервер попробует переключиться на пользователя zabbix. Не влияет, если сервер запускается от имени обычного пользователя. Значение по умолчанию — 0
EnableRemoteCommands	Указывает разрешены ли удаленные команды с сервера: - 0 — не разрешены; - 1 — разрешены
Hostname	Уникальное, регистрозависимое имя машины. Требуется для активных проверок и должно совпадать с именем машины, указанным на сервере
ListenIP	Список IP-адресов, разделенных запятой, которые агент должен слушать
ListenPort	Порт, который необходимо слушать для подключений с сервера
LogFile	Имя файла журнала. Обязательный параметр, если тип журнала указан file (см. параметр LogType)
LogType	Тип вывода журнала: - file — запись журнала в файл, указанный в параметре LogFile; - system— запись журнала в syslog; - console — вывод журнала в стандартный вывод
Server	Список IP-адресов или имен серверов, разделенных запятой. Входящие соединения будут приниматься только с адресов, указанных в параметре
TLSAccept	Обязательный параметр если заданы TLS-сертификат или параметры PSK, в противном случае — нет. Указывает какие входящие подключения принимаются. Используется пассивными проверками. Можно указывать несколько значений, разделенных запятой: - unencrypted — принимать подключения, не использующие криптографические ключи (по умолчанию); - psk — принимать подключения с TLS и pre-shared ключом (PSK); - cert — принимать подключения с TLS и сертификатом
TLSConnect	Обязательный параметр если заданы TLS-сертификат или параметры PSK, в противном случае — нет. Как агент должен соединяться с сервером или прокси. Используется активными проверками. Можно указать только одно значение: - unencrypted — подключаться без использования криптографических ключей (по умолчанию); - psk — подключаться, используя TLS и pre-shared ключ (PSK); - cert — подключаться, используя TLS и сертификат

Окончание таблицы 63

Параметр	Описание
User	Использование привилегий указанного (существующего) пользователя системы. Значение по умолчанию — zabbix. Актуально только если запускается от имни пользователя root и параметр AllowRoot не разрешен

Агент UNIX работает как демон, для запуска выполнить команду:

```
systemctl start zabbix-agent
```

Соответственно для остановки, перезапуска и просмотра состояния агента UNIX используются следующие команды:

```
systemctl stop zabbix-agent
systemctl restart zabbix-agent
systemctl status zabbix-agent
```

В среде Windows агент работает как служба. Агент Windows распространяется в виде zip-архива. Агент bin\win64\zabbix_agentd.exe и файл конфигурации conf\zabbix_agentd.win.conf из zip-архива необходимо скопировать в один каталог, например, C:\zabbix.

При необходимости откорректировать конфигурационный файл c:\zabbix\zabbix_agentd.win.conf.

Основные параметры конфигурационного файла агента Windows приведены в таблице 64.

Таблица 64

Параметр	Описание
EnableRemoteCommands	Указывает разрешены ли удаленные команды с сервера: - 0 — не разрешены; - 1 — разрешены
Hostname	Уникальное, регистрозависимое имя машины. Требуется для активных проверок и должно совпадать с именем машины, указанным на сервере
ListenIP	Список IP-адресов, разделенных запятой, которые агент должен слушать
ListenPort	Порт, который необходимо слушать для подключений с сервера
LogFile	Имя файла журнала. Обязательный параметр, если тип журнала указан file (см. параметр LogType)
LogType	Тип вывода журнала: - file — запись журнала в файл, указанный в параметре LogFile; - system— запись журнала в Журнал событий Windows; - console — вывод журнала в стандартный вывод

Окончание таблицы 64

Параметр	Описание
Server	Список IP-адресов или имен серверов, разделенных запятой. Входящие соединения будут приниматься только с адресов, указанных в параметре
TLSAccept	Обязательный параметр если заданы TLS-сертификат или параметры PSK, в противном случае — нет. Указывает какие входящие подключения принимаются. Используется пассивными проверками. Можно указывать несколько значений, разделенных запятой: - unencrypted — принимать подключения, не использующие криптографические ключи (по умолчанию); - psk — принимать подключения с TLS и pre-shared ключом (PSK); - cert — принимать подключения с TLS и сертификатом
TLSConnect	Обязательный параметр если заданы TLS-сертификат или параметры PSK, в противном случае — нет. Как агент должен соединяться с сервером или прокси. Используется активными проверками. Можно указать только одно значение: - unencrypted — подключаться без использования криптографических ключей (по умолчанию); - psk — подключаться, используя TLS и pre-shared ключом (PSK); - cert — подключаться, используя TLS и сертификат
User	Использование привилегий указанного (существующего) пользователя системы. Значение по умолчанию — zabbix. Актуально только если запускается от имни пользователя root и параметр AllowRoot не разрешен

Для установки агента Windows как службы используется следующая команда:

```
C:\> c:\zabbix\zabbix_agentd.exe -c c:\zabbix\zabbix_agentd.win.conf -i
```

В таблице 65 приведены основные параметры, используемые при управлении агентом.

Таблица 65

Параметр	Описание
Агент UNIX и Windows	
-c --config <файл_конфигурации>	Путь к файлу конфигурации, размещенному в каталоге, отличном от заданного по умолчанию. В UNIX путь по умолчанию /usr/local/etc/zabbix_agentd.conf. В Windows — c:\zabbix_agentd.conf
-p --print	Вывод известных данных и выход
-t --test <ключ_элемента_данных>	Тестирование указанного элемента данных и выход
Агент UNIX	
-R --runtime-control <параметр>	Выполнение административных функций для изменения уровня журналирования у процессов агента

Окончание таблицы 65

Параметр	Описание

`log_level_increase[=<цель>]`	Увеличение уровня журналирования, действует на все процессы, если цель не указана. В качестве цели может быть указан идентификатор процесса, тип процесса или тип и номер процесса:
	`zabbix_agentd -R log_level_increase`
	`zabbix_agentd -R log_level_increase=1234`
	`zabbix_agentd -R log_level_increase=listener,2`

`log_level_decrease[=<цель>]`	Уменьшение уровня журналирования, действует на все процессы, если цель не указана. В качестве цели может быть указан идентификатор процесса, тип процесса или тип и номер процесса: `zabbix_agentd -R log_level_decrease="active checks"`
Агент Windows	
`-m --multiple-agents`	Использование нескольких экземпляров агента (с `-i,-d,-s,-x` функциями). Для отделения имени экземпляров служб каждое имя службы будет в значении `Hostvalue` из указанного файла конфигурации
`-i --install`	Установка агента как службы
`-d --uninstall`	Удаление службы агента
`-s --start`	Запуск службы агента
`x --stop`	Остановка службы агента

15.3.4. Прокси

Для прокси требуется отдельная база данных. Для установки прокси с PostgreSQL выполнить команду:

```
apt-get install zabbix-proxy-pgsql
```

Для создания базы данных прокси используются скрипты по созданию базы данных для PostgreSQL, например:

```
psql -U <username>
create database zabbix;
\q
cd database/postgresql
psql -U <username> zabbix < schema.sql
```

Далее необходимо импортировать исходную схему и данные прокси на PostgreSQL:

```
zcat /usr/share/doc/zabbix-proxy-pgsql/create.sql.gz | psql -U <username>
   zabbix
```

Для настройка базы данных прокси изменить конфигурационный файл `zabbix_proxy.conf`.

Пример

```
vi /etc/zabbix/zabbix_proxy.conf
```

```
DBHost=localhost
DBName=zabbix
DBUser=zabbix
DBPassword=<пароль>
```

В параметре DBPassword указать пароль пользователя PosgreSQL.

Основные параметры конфигурационного файла прокси приведены в таблице 66.

Таблица 66

Параметр	Описание
AllowRoot	Разрешение прокси запускаться от имени пользователя root. Если не разрешено (значение «0») и прокси запускается от имени root, прокси попробует переключиться на пользователя zabbix. Не влияет, если прокси запускается от имени обычного пользователя. Значение по умолчанию — 0
CacheSize	Размер кэша конфигурации в байтах для хранения данных узлов сети, элементов данных и триггеров. Возможные значения от 128 КБ до 8 ГБ, значение по умолчанию — 8 МБ
ConfigFrequency	Часто получения данных конфигурации от сервера, в секундах. Параметр активного прокси. Игнорируется пассивными прокси (см. параметр ProxyMode). Возможные значения от 1 до 604800 сек, значение по умолчанию — 3600 сек
DBHost	Имя хоста базы данных. В случае пустой строки PostgreSQL будет использовать сокет. Значение по умолчанию — localhost
DBName	Обязательный параметр. Имя базы данных. Должна отличаться от базы данных сервера
DBPassword	Пароль к базе данных
DBPort	Порт базы данных, когда не используется localhost. Значение по умолчанию — 3306
DBSchema	Имя схемы
DBUser	Пользователь базы данных
DataSenderFrequency	Частота отправки собранных значений серверу, в секундах. Параметр активного прокси. Игнорируется пассивными прокси (см. параметр ProxyMode). Возможные значения от 1 до 3600 сек, значение по умолчанию — 1 сек
Hostname	Уникальное регистрозависимое имя прокси
HousekeepingFrequency	Частота выполнения автоматической процедуры очистки базы данных от устаревшей информации, в часах. Возможные значения от 0 до 24 ч, значение по умолчанию — 1
ProxyMode	Режим работы прокси: - 0 — прокси в активном режиме; - 1 — прокси в пассивном режиме

Окончание таблицы 66

Параметр	Описание
Server	IP-адрес или имя сервера для доступа к данным конфигурации с сервера. Параметр активного прокси, игнорируется пассивными прокси (см. ProxyMode)
TLSAccept	Обязательный параметр если заданы TLS-сертификат или параметры PSK, в противном случае — нет. Указывает какие входящие подключения принимаются от сервера. Используется пассивным прокси, игнорируется активным прокси. Можно указывать несколько значений, разделенных запятой: - unencrypted — принимать подключения, не использующие криптографические ключи (по умолчанию); - psk — принимать подключения с TLS и pre-shared ключом (PSK); - cert — принимать подключения с TLS и сертификатом
TLSConnect	Обязательный параметр если заданы TLS-сертификат или параметры PSK, в противном случае — нет. Как прокси должен соединяться с сервером. Используется активным прокси, игнорируется пассивным прокси. Можно указать только одно значение: - unencrypted — подключаться без использования криптографических ключей (по умолчанию); - psk — подключаться, используя TLS и pre-shared ключом (PSK); - cert — подключаться, используя TLS и сертификат

Прокси работает как демон. Для запуска прокси выполнить команду:

```
systemctl start zabbix-proxy
```

Соответственно для остановки, перезапуска и просмотра состояния прокси используются следующие команды:

```
systemctl stop zabbix-proxy
systemctl restart zabbix-proxy
systemctl status zabbix-proxy
```

В таблице 67 приведены основные параметры командной строки zabbix-proxy.

Таблица 67

Параметр	Описание
-c --config <файл>	Путь к файлу конфигурации. Значение по умолчанию /etc/zabbix/zabbix_proxy.conf
-R --runtime-control <параметр>	Выполнение административных функций
config_cache_reload	Перезагрузка кэша конфигурации. Игнорируется, если кэш загружается в данный момент. Активный прокси подключится к серверу и запросит данные конфигурации: zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R config_cache_reload

Окончание таблицы 67

Параметр	Описание
`housekeeper_execute`	Запуск процедуры очистки базы данных. Игнорируется, если процедура очистки выполняется в данный момент: `zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R housekeeper_execute`

`log_level_increase[=<цель>]`	Увеличение уровня журналирования, действует на все процессы, если цель не указана. В качестве цели может быть указан идентификатор процесса, тип процесса или тип и номера процесса:
	`zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R log_level_increase`
	`zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R log_level_increase=1234`
	`zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R log_level_increase=poller,2`

`log_level_decrease[=<цель>]`	Уменьшение уровня журналирования, действует на все процессы, если цель не указана. В качестве цели может быть указан идентификатор процесса, тип процесса или тип и номера процесса: `zabbix_proxy -c /usr/local/etc/zabbix_proxy.conf -R log_level_decrease="http poller"`

15.3.5. Web-интерфейс

Настройка и управление работой Zabbix осуществляется посредством web-интерфейса.

Установка web-интерфейса производится путем копирования `php`-файлов в папку HTML web-сервера. Далее необходимо:

1) ввести URL Zabbix `http://<ip_или_имя_сервера>/zabbix` в браузере — откроется первая страница помощника установки web-интерфейса;

2) указать данные для подключения к базе данных. База данных должна быть создана;

3) указать данные сервера;

4) подтвердить данные для настройки;

5) скачать конфигурационный файл и поместить его в каталог `conf/` (если web-сервер имеет право на запись в каталог `conf/`, файл будет сохранен автоматически);

6) завершить установку.

Для входа по умолчанию используется имя пользователя `Admin` и пароль `zabbix`.

16. РЕЗЕРВНОЕ КОПИРОВАНИЕ И ВОССТАНОВЛЕНИЕ ДАННЫХ

Система резервного копирования является составной частью плана восстановления системы.

Резервное копирование выполняется с целью обеспечения возможности восстановления отдельных файлов или ФС в целом с минимальными затратами труда и времени в случае утери рабочей копии информации. Резервные копии должны создаваться периодически, в соответствии с заранее установленным графиком (см. 16.2).

Процесс резервного копирования должен быть максимально автоматизирован и требовать наименьшего участия со стороны администратора системы.

Резервное копирование — это процесс, влияющий на работоспособность системы. Резервное копирование и восстановление увеличивает текущую нагрузку на систему, что может вызывать замедление работы системы. Кроме того, в зависимости от вида резервного копирования и восстановления, может потребоваться монопольный доступ к системе или полная остановка ее работы.

Основная идея резервного копирования — создание копий критической части содержания резервируемой системы. Основными исключениями, как правило, не входящими в процедуру резервного копирования функционирующей ОС, являются каталоги, содержащие служебные данные, меняющиеся в процессе функционирования (`/dev`, `/media`, `/mnt`, `/parsecfs`, `/proc`, `/run`, `/sys`, `/tmp`), а также сетевые каталоги (смонтированная NFS, Samba и прочие виды сетевых данных).

Элементы системы резервного копирования должны включать необходимое оборудование, носители резервных копий и ПО. Для хранения резервных копий могут быть использованы различные носители информации: дисковые накопители, отчуждаемые носители информации или специально выделенные разделы жесткого диска. Тип и количество носителей определяются используемым оборудованием, объемами обрабатываемых данных и выбранной схемой резервирования данных. ПО резервного копирования и восстановления из состава ОС включает утилиты командной строки и распределенные системы управления хранилищами данных:

1) комплекс программ Bacula (16.3);
2) утилита копирования `rsync` (16.4);
3) утилиты архивирования `tar`, `cpio`, `gzip` (16.5).

ВНИМАНИЕ! Для восстановления мандатных атрибутов файлов из резервных копий процесс должен иметь PARSEC-привилегию `0x1000` (`PARSEC_CAP_UNSAFE_SETXATTR`). Привилегия может быть получена с использованием утилиты `execaps`:

```
sudo execaps -c 0x1000 tar .......
```

ВНИМАНИЕ! Восстановление расширенных атрибутов файлов с использованием `unsecure_setxattr` возможно только в случае, если атрибуты восстанавливаются с помощью системного вызова `setxattr` путем установки атрибута `security.PDPL`. Использование `unsecure_setxattr` не влияет на возможность изменения мандатных атрибутов файлов системными вызовами `pdpl_set_path`, `pdpl_set_fd`.

Комплекс программ Bacula позволяет системному администратору управлять процессами резервного копирования и восстановления данных, находить и восстанавливать утраченные или поврежденные файлы, а также проверять резервные копии, в том числе в гетерогенных сетях.

Утилита `rsync` предоставляет возможности для локального и удаленного копирования (резервного копирования) или синхронизации файлов и каталогов с минимальными затратами трафика.

Утилиты командной строки `tar`, `cpio`, `gzip` представляют собой традиционные инструменты создания резервных копий и архивирования ФС.

Порядок выполнения операций резервного копирования и восстановления объектов ФС с сохранением и восстановлением мандатных атрибутов и атрибутов аудита описан в РУСБ.10152-02 97 01-1.

16.1. Виды резервного копирования

Существуют следующие виды резервного копирования:

- полное резервное копирование — сохранение резервной копии всех файлов системы. Процедура занимает много времени и требует место для хранения большого объема. Как правило, выполняется в тех случаях, когда не влияет на основную работу системы, или для создания базовой резервной копии данных. В дальнейшем может выполняться дифференциальное или инкрементное резервное копирование;
- дифференциальное резервное копирование — сохранение копий изменившихся с последнего полного резервного копирования файлов. Требования к объему хранения и времени создания меньше, чем при полном копировании. Время восстановления незначительно за счет прямой перезаписи файлов;
- инкрементное резервное копирование — сохранение изменений файлов с момента последнего инкрементного копирования. Требует минимального количества времени и места для создания копии, но усложняет последующее восстановление, поскольку необходимо последовательное восстановление всех инкрементных копий с момента последнего полного резервного копирования.

16.2. Планирование резервного копирования

Планирование резервного копирования заключается в рассмотрении и определении следующих вопросов:

- что именно и как часто должно архивироваться;
- какие виды резервного копирования и на какие носители должны применяться;
- как часто и каким образом будут восстанавливаться файлы при необходимости;
- каким образом пользователи могут запросить ранее сохраненные файлы.

План резервного копирования должен периодически пересматриваться для отражения текущих изменений в системе, используемых технологиях или условиях функционирования.

16.2.1. Составление расписания резервного копирования

При составлении расписания резервного копирования определяется что, когда и на каком носителе должно сохраняться.

Должна существовать возможность восстановления любого файла в любой момент времени. Например, требуется восстановить файл не более, чем однодневной давности. Для этого может использоваться комбинация полного и обновляемого (дифференциального или инкрементного) резервного копирования. Полное резервное копирование позволяет сохранить копии всех файлов системы, обновляемое — только изменившиеся со времени последнего архивирования. Обновляемое может иметь несколько уровней, например, обновление по отношению к последней обновляемой резервной копии.

Для восстановления отдельных файлов при таком многоуровневом расписании может понадобиться полная резервная копия, если файл не изменялся в течение месяца; копия первого уровня, если файл не изменялся в течение недели; копия второго уровня при ежедневной работе с этим файлом. Такая схема несколько сложнее, однако требует меньших ежедневных затрат времени.

Примечание. Расписание резервного копирования должно быть доведено до пользователей.

16.2.2. Планирование восстановления системы

При составлении плана резервного копирования должен быть определен план действий на случай аварийной ситуации, как при необходимости может быть восстановлена система или отдельные файлы, где хранятся и насколько доступны носители с резервными копиями и не могут ли они потерять работоспособность при неполадках на компьютере.

Примечание. Необходимо периодически выполнять проверку исправности носителей с архивами резервных копий. Проверка может включать в себя чтение содержимого копии после сохранения или выборочную проверку файлов резервной копии.

16.3. Комплекс программ Bacula

Bacula — это сетевая клиент-серверная система резервного копирования. Благодаря модульной архитектуре Bacula может масштабироваться от небольших автономных систем до больших сетей, состоящих из сотен компьютеров.

Bacula состоит из следующих основных компонентов:

- Bacula Director — центральная программа, координирующая все выполняемые операции (функционирует в фоне);
- Bacula Console — консоль Bacula, позволяющая администратору взаимодействовать с центральной программой;
- Bacula File — клиентская программа, устанавливаемая на каждый обслуживаемый компьютер;
- Bacula Storage — программа, обычно функционирующая на компьютере, к которому присоединены внешние устройства для хранения резервных копий;
- Catalog — программа, отвечающая за индексирование и организацию базы резервных данных.

Программа Bacula обеспечивает поддержку сохранения расширенных атрибутов каталогов и файлов и, при необходимости, их последующее восстановление (см. РУСБ.10152-02 97 01-1).

Все действия выполняются от имени учетной записи администратора с использованием механизма `sudo`.

Порядок использования Bacula описан на примере системы со следующей инфраструктурой:

- выделенный сервер `bakula1.my.dom` с IP-адресом `11.11.11.21` для функционирования Bacula Director — главный сервер, осуществляющий резервное копирование;
- выделенный сервер `bakula2.my.dom` с IP-адресом `11.11.11.22` для функционирования Bacula Storage — машина, на которой будут размещаться резервные копии данных;
- персональный компьютер `bakula3.my.dom` с IP-адресом `11.11.11.23` для функционирования Bacula File — машина, с которой будут копироваться данные и на которую будут восстанавливаться резервные копии данных.

16.3.1. Подготовка инфраструктуры

Для подготовки инфраструктуры к управлению системой резервного копирования необходимо выполнить следующие действия:

1) установить PostgreSQL на сервер, где будет работать Bacula Director:

```
aptitude install postgresql-11
```

2) установить pgadmin3 на сервер, где будет работать Bacula Director:

```
aptitude install pgadmin3
```

3) предполагается, что на всех машинах изначально установлены все пакеты, касающиеся Bacula, из состава ОС. Через менеджер пакетов Synaptic по ключевому слову «bacula» необходимо установить все пакеты, кроме тех, где в названии фигурирует «-sqlite3».

При настройке Bacula в появившемся интерфейсе настройки совместимости с БД в качестве имени БД необходимо указать bacula и пароль bacula.

В случае возникновения ошибки игнорировать ее на данном этапе, БД будет настроена позднее;

4) подготовить БД для Bacula выполнив следующие действия:

- в файле /etc/postgresql/11/main/postgresql.conf указать listen_addresses = '*';
- в файле /etc/postgresql/11/main/pg_hba.conf внести необходимые изменения, для простоты можно указать метод trust для всех соединений, удалить любую дополнительную конфигурацию после метода типа mod=;
- обязательно добавить host с IP-адресом, где будет работать bacula-dir. В случае если все демоны Bacula будут установлены на одну машину, указывать IP-адрес не обязательно, т.к. работа будет идти через localhost.

Пример

Файл pg_hba.conf

```
local all postgres trust
local all all trust
host all all 127.0.0.1/32 trust
host all all 11.11.11.21/24 trust
```

- выполнить запуск БД:

```
pg_ctlcluster 11 main restart
```

- присвоить пароль postgres:

```
passwd postgres
```

- присвоить для Bacula пароль bacula:

```
passwd bacula
```

- создать пользователя БД для работы с Bacula (выполнять не от имени учетной записи администратора):

```
# psql template1 postgres
postgres=# CREATE ROLE bacula;
postgres=# ALTER USER bacula PASSWORD 'bacula';
```

```
postgres=# ALTER USER bacula LOGIN SUPERUSER CREATEDB CREATEROLE;
```

5) создать БД bacula (выполнять не от имени учетной записи администратора):

- выполнить pgadmin3;
- указать имя template1, пользователя postgres, пароль postgres;
- в секции Роли входа добавить роль входа bacula. Создать БД bacula, владельцем назначить bacula;

6) на сервере bakula1.my.dom необходимо запустить скрипты, которые создадут все необходимые таблицы и привилегии, предварительно отредактировав их:

- в скрипте /usr/share/bacula-director/make_postgresql_tables внести следующие изменения:
  - в строке db_name указать имя –bacula;
  - в строке psql после psql вписать –U bacula;
- в скрипте /usr/share/bacula-director/grant_postgresql_privileges внести следующие изменения:
  - в строке db_user указать имя –bacula;
  - в строке db_name указать имя –bacula;
  - в строке db_password указать пароль bacula;
  - в строке $bindir/psql после psql вписать –U bacula;
- сохранить изменения и выполнить скрипты:

```
make_postgresql_tables
grant_postgresql_privileges
```

7) на машине, где будет работать Bacula Storage, необходимо создать каталог /back, в котором будут храниться резервные копии данных, и присвоить каталогу владельца bacula:

```
mkdir /back
chown –R bacula /back
```

8) на машине, где будет работать Bacula File, необходимо создать каталог /etc2, в который будут восстанавливаться данные из резервной копии:

```
mkdir /etc2
```

Если подготовительные настройки выполнены корректно, БД стартует без ошибок и скрипты выполнились без ошибок, то можно приступить к настройке Bacula.

16.3.2. Настройка Bacula

Подготовка Bacula к работе заключается в настройке каждого компонента в отдельности и последующей настройке их взаимодействия.

16.3.2.1. Настройка Bacula Director

Настройка Bacula Director осуществляется путем корректировки конфигурационного файла `/etc/bacula/bacula-dir` сервера `bakula1.my.dom`.

В первую очередь необходимо определить основные параметры в секции `Director`. На начальном этапе важно установить параметры `Name` и `Password`. `Name` задает уникальное имя Bacula Director, а `Password` — пароль, который будет использоваться при соединениях `BC` с `DD`. Остальные параметры можно оставить со значениями по умолчанию:

```
Director { # define myself
Name = bacula-dir
DIRport = 9101 # where we listen for UA connections
QueryFile = "/etc/bacula/scripts/query.sql"
WorkingDirectory = "/var/lib/bacula"
PidDirectory = "/var/run/bacula"
Maximum Concurrent Jobs = 1
Password = "1" # Console password
Messages = Daemon
DirAddress = 11.11.11.21
}
```

Следующей группой параметров, которые необходимо определить, является секция `Catalog`. В ней необходимо указать реквизиты доступа к БД, а также назначить уникальное имя данного Bacula Catalog с помощью параметра `Name`:

```
Catalog {
Name = MyCatalog
# Uncomment the following line if you want the dbi

PS. driver
# dbdriver = "dbi:sqlite3"; dbaddress = 127.0.0.1; dbport =
dbname = "bacula"; dbuser = "bacula"; dbpassword = "bacula"
DB Address = 11.11.11.21
}
```

Далее необходимо определить `SD`, на который будет производиться передача данных для дальнейшей записи на устройство хранения. Когда Bacula Storage настроен и готов к работе, необходимо определить реквизиты доступа к нему в секции `Storage` файла `bacula-dir.conf`. Основные параметры:

1) `Name` — уникальное имя, использующееся для адресации секции `Storage` в рамках файла `bacula-dir.conf`;

2) Device и MediaType — дублируют одноименные параметры файла bacula-sd.conf;

3) Password — содержит пароль, который будет использоваться при подключении к Bacula Storage:

```
Storage {
Name = File
# Do not use "localhost" here
Address = 11.11.11.22 # N.B. Use a fully qualified name here
SDPort = 9103
Password = "1"
Device = FileStorage
Media Type = File
}
```

Секция Pool определяет набор носителей информации и параметры, используемые SD при их обработке. Каждый Pool взаимодействует с устройством хранения данных, поэтому необходимо создать столько пулов, сколько определено устройств хранения. Фактически если для каждого Bacula File определено отдельное устройство, то для каждого FD необходимо определить и Pool. Основные параметры:

1) Name — определяет уникальное имя пула;

2) Pool Type — определяет тип, для резервных копий должен быть установлен в значение Backup;

3) Maximum Volume Jobs — рекомендуется установить в значение 1. Данное значение указывает, что в рамках одного носителя данных могут быть размещены резервные данные, полученные в ходе выполнения только одного задания. Если размер созданной резервной копии много меньше размера носителя, то имеет смысл сохранять на него копии, которые будут создаваться в будущем. Но если говорится о файлах, то желательно придерживаться правила «один файл — одна копия», т.е. в одном файле Bacula должны храниться резервные данные, которые были сформированы в рамках выполнения одного задания. Для каждого последующего будут создаваться новые файлы;

4) Volume Retention — время, по прошествии которого данные о резервной копии, хранящейся на носителе, будут удалены из каталога. Для обеспечения работоспособности Bacula при указании значения данного параметра необходимо учитывать, что информация обо всех зарезервированных файлах хранится в БД, по записи на каждый файл. Если резервируются тысячи файлов, то за непродолжительное время БД станет огромной, что может затруднить работу Bacula. Поэтому важно своевре-

менно очищать БД от устаревшей информации. При этом сам носитель информации не будет очищен автоматически. Он будет промаркирован как устаревший, но всегда можно будет использовать его для восстановления данных в ручном режиме;

5) `Maximum Volumes` — максимальное количество носителей (в данном случае файлов), доступных в пуле;

6) `Recycle` — указывает на необходимость повторного использования носителей, помеченных как устаревшие. При этом реальная перезапись носителя произойдет лишь в случае, когда свободных носителей не останется. Свободные носители определяются из параметра `Maximum Volumes`;

7) `AutoPrune` — указывает на необходимость удаления устаревших записей из Bacula Catalog автоматически после завершения выполнения очередного задания;

8) `Label Format` — определяет префикс, который будет использован Bacula для маркирования носителей информации, в данном случае — для именования файлов;

9) `Storage` — указывает на имя устройства хранения данных, указанного в параметре `Name` секции `Storage` файла `bacula-dir.conf`.

```
Pool {
Name = Default
Pool Type = Backup
Recycle = yes # Bacula can automatically recycle Volumes
AutoPrune = yes # Prune expired volumes
Volume Retention = 1 month # one year
Maximum Volume Jobs = 1
Maximum Volumes = 32
Storage = File
Label Format = "volume-"
}
```

Секция `FileSet` позволяет предопределить несколько наборов резервируемых файлов. Например, один набор для Windows, другой — для Linux или один для серверов, а другой — для рабочих станций. Параметр `Name` определяет уникальное имя набора.

Секция `Include` содержит пути к резервируемым файлам/каталогам, а `Exclude` — пути к файлам и каталогам, которые необходимо исключить из списка резервируемых. В секции `Include` возможна секция `Options`, в которой определяются параметры резервирования. Основные параметры:

1) `signature` — указывает алгоритм вычисления контрольных сумм файлов;

2) `compression` — указывает алгоритм компрессии файлов;

3) `recurse` — указывает на необходимость рекурсивного резервирования, включая подкаталоги и файлы;

4) File — указывает копируемый каталог;

5) xattrsupport — указывает на возможность включения поддержки расширенных атрибутов, это обязательный параметр для работы с метками безопасности:

```
FileSet {
Name = "Catalog"
Include {
Options {
signature = MD5
compression = GZIP
# recurse = yes
aclsupport = yes
xattrsupport = yes
}
File = /etc
}
}
```

Все настройки связываются воедино с помощью секции Job, в которой дается задание планировщику по выполнению резервирования данных. Основные параметры:

1) Type — указывает на тип задания. Типов существует несколько. Здесь достаточно указать Backup;

2) Schedule — указывает на предопределенное расписание, согласно которому будет выполняться резервирование данных. Все расписания определены в файле bacula-dir.conf;

3) Where — указывает на каталог, в котором будут восстанавливаться данные из резервной копии;

4) Write Bootstrap — указывает путь к файлу, в который будет записываться информация, с помощью которой данные могут быть восстановлены из резервной копии без наличия подключения к Bacula Catalog. Вместо %n будет подставлено значение параметра Name:

```
Schedule {
Name = "DailyCycle"
Run = Full daily at 16:10
# Run = Differential 2nd-5th sun at 23:05
Run = Incremental mon-sat at 23:05
}

Job {
```

```
Name = "RestoreFiles"
Type = Restore
Client= bacula-fd
FileSet="Catalog"

Storage = File
Pool = Default
Messages = Standard
Where = /etc2
}

Job {
Name = "BackupCilent1"
Type = Backup
Client = bacula-fd
FileSet = "Catalog"
Schedule = "DailyCycle"
Messages = Standard
Pool = Default
Write Bootstrap = "/var/lib/bacula/Client1.bsr"
Priority = 1
}
```

Затем необходимо указать параметры единственного Агента:

```
Client {
Name = bacula-fd
Address = 11.11.11.23
FDPort = 9102
Catalog = MyCatalog
Password = "1" # password for FileDaemon
File Retention = 30 days # 30 days
Job Retention = 6 months # six months
AutoPrune = yes # Prune expired Jobs/Files
}
```

Остальные секции (`Job`, `JobDefs`, `Client` и `Console`) необходимо закомментировать. Трафик данных будет идти по портам, указанным в конфигурационных файлах каждого из компонентов Bacula.

Настроить доступ к DD со стороны Bacula Console в файле /etc/bacula/bconsole.conf сервера bakula1.my.dom:

```
Director {
Name = bacula-dir
DIRport = 9101
address = 11.11.11.21
Password = "1"
}
```

На машине, где будет функционировать Bacula Director, следует удалить пакеты bacula-sd и bacula-fd:

```
apt-get remove bacula-sd
apt-get remove bacula-fd
```

Конфигурационные файлы bacula-sd и bacula-fd в /etc/bacula следует переименовать либо удалить.

Службы bacula-sd и bacula-fd остановить:

```
systemctl stop bacula-sd
systemctl stop bacula-fd
```

16.3.2.2. Настройка Bacula Storage

Bacula Storage отвечает за непосредственную работу с устройством хранения данных. Bacula поддерживает широкий спектр устройств от оптических дисков до полнофункциональных ленточных библиотек. В описываемой системе используется самый распространенный вариант — жесткий диск с существующей файловой системой (например, ext3).

Для настройки Bacula Storage необходимо на сервере bakula2.my.dom отредактировать конфигурационный файл /etc/bacula/bacula-sd.conf.

В секции основных параметров Storage определить параметр Name, который задает уникальное имя Bacula Storage. Для остальных параметров возможно оставить значения по умолчанию.

Секция Director необходима для указания уникального имени DD и пароля, с которым данный DD может подключаться к SD. Секций Director в файле может быть несколько, что дает возможность использовать единый сервер хранения данных для нескольких систем резервирования. Все остальные секции Director, найденные в файле, необходимо закомментировать:

```
Storage { # definition of myself
Name = bacula-sd
SDPort = 9103 # Director's port
WorkingDirectory = "/var/lib/bacula"
Pid Directory = "/var/run/bacula"
```

```
Maximum Concurrent Jobs = 20
SDAddress = 11.11.11.22
}

Director {
Name = bacula-dir
Password = "1"
}
```

Основные настройки, определяющие взаимодействие с устройствами хранения, находятся в секции `Device`. Параметры, необходимые для хранения резервных копий в рамках существующей ФС, примонтированной в каталог `/back`:

1) `Name` — определяет уникальное имя подключенного устройства. Если планируется создавать изолированные друг от друга резервные копии для каждого из Bacula File, то необходимо создать несколько секций `Device` с уникальными именами. В противном случае резервируемые файлы со всех `FD` будут размещаться в одном и том же файле, что может затруднить дальнейшее обслуживание системы;

2) `Media Type` — определяет произвольное уникальное имя, которое будет использоваться Bacula при восстановлении данных. Согласно ему определяется устройство хранения, с которого будет производиться восстановление. Если резервные копии хранятся в файлах, то для каждой секции `Device` должен быть задан уникальный `Media Type`;

3) `Archive Device` — указывает путь к файлу устройства в каталоге `/dev` или путь к каталогу, в котором будут размещаться резервные копии;

4) `Device Type` — определяет тип устройства. Для размещения в существующей ФС указывается `File`;

5) `Random Access` — указывает на возможность случайной (непоследовательной) адресации. Для файлов указывается `Yes`;

6) `RemovableMedia` — указывает, возможно ли извлечение устройства хранения. Необходимо для ленточных устройств, приводов оптических дисков и т.д. Для файлов устанавливается в значение `No`;

7) `LabelMedia` — указывает на необходимость автоматического маркирования носителей информации:

```
Device {
Name = FileStorage
Media Type = File
Archive Device = /back
LabelMedia = yes; # lets Bacula label unlabeled media
```

```
Random Access = Yes;
AutomaticMount = yes; # when device opened, read it
RemovableMedia = no;
AlwaysOpen = no;
}
```

На машине, где будет функционировать Bacula Storage, следует удалить пакет bacula-fd:

```
apt-get remove bacula-fd
```

Конфигурационный файл bacula-fd в /etc/bacula следует переименовать либо удалить.

Службу bacula-fd остановить:

```
systemctl stop bacula-fd
```

16.3.2.3. Настройка Bacula File

Для настройки Bacula File на рабочей станции bakula3.my.dom используется конфигурационный файл /etc/bacula/bacula-fd. Для базовой настройки достаточно определить параметры секций Director и FileDaemon.

В секции Director указывается пароль, который будет использовать DD при подключении к FD. Секций Director в файле может быть несколько, все остальные секции Director, найденные в файле, необходимо закомментировать:

```
Director {
Name = bacula-dir
Password = "1"
}
```

В секции FileDaemon указываются настройки FD, в ней необходимо определить параметр Name, в котором указывается уникальное имя Bacula File:

```
FileDaemon { # this is me
Name = bacula-fd
FDport = 9102 # where we listen for the director
WorkingDirectory = /var/lib/bacula
Pid Directory = /var/run/bacula
Maximum Concurrent Jobs = 20
FDAddress = 11.11.11.23
}
```

На машине, где будет функционировать Bacula File, следует удалить пакет bacula-sd:

```
apt-get remove bacula-sd
```

Конфигурационный файл `bacula-sd` в `/etc/bacula` следует переименовать либо удалить.

Службу `bacula-sd` следует остановить:

```
systemctl stop bacula-sd
```

Далее необходимо запустить все компоненты соответствующими командами, выполненными на соответствующих серверах:

```
systemctl restart bacula-director
systemctl restart bacula-sd
systemctl restart bacula-fd
```

16.3.2.4. Проверка Bacula

После настройки Bacula Director, Bacula Storage и Bacula File программа Bacula готова к работе. Управление Bacula осуществляется через `bconsole`. Настройки каталогов, заданий, расписаний и прочие задаются в конфигурационных файлах.

Для тестовой проверки необходимо:

- выполнить `bconsole`;
- выполнить `run`;
- выбрать `job 1`;
- войти в меню, набрав `mod`;
- выбрать `1` (Level);
- выбрать `1` (Full);
- подтвердить выполнение, набрав `yes`.

В результате будет создана резервная копия данных в каталоге `/back` на машине с Bacula Storage.

Для восстановления объектов ФС с установленными мандатными атрибутами необходимо запустить консоль управления Bacula с PARSEC-привилегией 0x1000, выполнив команду:

```
sudo execaps -c 0x1000 -- bconsole
```

Для восстановления данных из резервной копии необходимо:

- выполнить `restore`;
- выбрать пункт `12`;
- ввести номер `job id`;
- указать параметр маркировки `mark *`;
- подтвердить выполнение командой `done`.

Данные из резервной копии будут восстановлены в каталоге `/etc2` на машине с Bacula File.

Также управление Bacula возможно с помощью графической утилиты `bacula-console-qt`.

16.4. Утилита копирования `rsync`

Все действия при использовании команды `rsync` выполняются от имени учетной записи администратора с использованием механизма `sudo`.

В таблице 68 приведены некоторые наиболее часто используемые параметры команды `rsync`.

Таблица 68

Параметр	Назначение
`-v, --verbose`	Подробный вывод
`-z, --compress`	Сжимать трафик
`-r, --recursive`	Выполнять копирование рекурсивно
`-p, --perms`	Сохранять дискретные права доступа
`-t, --times`	Сохранять время доступа к файлам
`-g, --group`	Сохранять группу
`-o, --owner`	Сохранять владельца
`-A, --acls`	Сохранять списки контроля доступа ACL (включает `-p`)
`-X, --xattrs`	Сохранять расширенные атрибуты (в том числе мандатные атрибуты)

Подробное описание команды приведено в `man` для `rsync`.

Пример

Следующая команда сделает копию домашнего каталога на 192.168.0.1

```
sudo rsync -vzrptgoAX /home/ admin@192.168.0.1:/home_bak
```

В данном примере должен быть создан каталог `/home_bak` на сервере и установлены на него максимальные метки с `ccnr`.

ВНИМАНИЕ! Не рекомендуется использовать параметр `-l` для копирования символических ссылок при создании резервной копии домашних каталогов пользователей.

16.5. Утилиты архивирования

При создании архива командами `tar` и `gzip` передается список файлов и каталогов, указываемых как параметры командной строки. Любой указанный каталог просматривается рекурсивно. При создании архива с помощью команды `cpio` ей предоставляется список объектов (имена файлов и каталогов, символические имена любых устройств, гнезда доменов UNIX, поименованные каналы и т. п.).

Все действия при использовании команд `tar`, `cpio` и `gzip` выполняются от имени учетной записи администратора с использованием механизма `sudo`.

Подробное описание команд приведено в руководстве man для tar, cpio и gzip.

16.5.1. tar

Команда tar может работать с рядом дисковых накопителей, позволяет просматривать архивы в ОС.

В таблице 69 приведены основные параметры команды tar.

Таблица 69

Опция	Назначение
--acls	Сохраняет (восстанавливает) списки контроля доступа (ACL) каталогов и файлов, вложенных в архив
-c, --create	Создает архив
-x, --extract, --get	Восстанавливает файлы из архива на устройстве, заданном по умолчанию или определенном параметром f
--xattrs	Сохраняет (восстанавливает) расширенные атрибуты каталогов и файлов, вложенных в архив
-f, --file name	Создает (или читает) архив с name, где name — имя файла или устройства, определенного в /dev, например, /dev/rmt0
-Z, --compress, --uncompress	Сжимает или распаковывает архив с помощью compress
-z, --gzip, --gunzip	Сжимает или распаковывает архив с помощью gzip
-M, --multi-volume	Создает многотомный архив
-t, --list	Выводит список сохраненных в архиве файлов
-v, --verbose	Выводит подробную информацию о процессе

Подробное описание команды приведено в man для tar.

В примерах приведены варианты использования команды tar.

Примеры:

1. Копирование каталога /home на специальный раздел жесткого диска /dev/hda4

```
tar -cf /dev/hda4 /home
```

Параметр f определяет создание архива на устройстве /dev/hda4.

2. Применение сжатия при архивировании

```
tar -cvfz /dev/hda4 /home | tee home.index
```

Параметр v заставляет tar выводить подробную информацию, параметр z указывает на сжатие архива с помощью утилиты gzip. Список скопированных файлов направляется в home.index.

3. Использование команды find для поиска измененных в течение одного дня файлов в каталоге /home и создание архива home.new.tar с этими файлами:

```
find /home -mtime 1 -type f -exec tar -rf home.new.tar {} \;
```

4. Если надо посмотреть содержимое архива, то можно воспользоваться параметром -t команды tar:

```
tar -tf home.new.tar
```

5. Для извлечения файлов из архива необходимо указать путь к архиву либо устройству и путь к месту извлечения. Если архив (каталога /home) был создан командой:

```
tar -czf /tmp/home.tar /home
```

то извлекать его надо командой:

```
tar -xzf /tmp/home.tar /
```

6. Использование команды tar для создания архивов в ФС ОС, а не только на устройствах для архивирования (можно архивировать группу файлов с их структурой каталогов в один файл, для чего передать имя создаваемого файла с помощью параметра f вместо имени устройства)

```
tar cvf /home/backup.tar /home/dave
```

С помощью tar архивируется каталог с вложенными подкаталогами.

При этом создается файл /home/backup.tar, содержащий архив каталога /home/dave и всех файлов в его подкаталогах.

Обычно при использовании команды tar стоит делать входом верхнего уровня каталог. В таком случае файлы при восстановлении будут располагаться в подкаталоге рабочего каталога.

Предположим, в рабочем каталоге имеется подкаталог data, содержащий несколько сотен файлов. Существует два основных пути создания архива этого каталога. Можно войти в подкаталог и создать в нем архив, например:

```
pwd
/home/dave
cd data
pwd
/home/dave/data
tar cvf .. /data.tar *
```

Будет создан архив в каталоге /home/dave, содержащий файлы без указания их расположения в структуре каталогов. При попытке восстановить файлы из архива data.tar подкаталог не будет создан, и все файлы будут восстановлены в текущем каталоге.

Другой путь состоит в создании архива каталога, например:

```
pwd
/home/dave
```

```
tar cvf data.tar data
```

Будет создан архив каталога, в котором первой будет следовать ссылка на каталог. При восстановлении файлов из такого архива будет создан подкаталог в текущем каталоге, и файлы будут восстанавливаться в нем.

Можно автоматизировать выполнение данных команд, поместив их в файл `crontab` суперпользователя. Например, следующая запись в файле `crontab` выполняет резервное копирование каталога `/home` ежедневно в 01:30:

```
30 01 *** tar -cvfz /dev/hda4 /home > home index
```

При необходимости более сложного архивирования используется язык сценариев оболочки, которые также могут быть запущены с помощью `cron` (см. 4.3.1.2).

Порядок использования команды `tar` для сохранения и восстановления мандатных атрибутов файлов описан в РУСБ.10152-02 97 01-1.

16.5.2. cpio

Для копирования файлов используется команда общего назначения `cpio`.

Команда используется с параметром `-o` для создания резервных архивов и с параметром `-i` — для восстановления файлов. Команда получает информацию от стандартного устройства ввода и посылает выводимую информацию на стандартное устройство вывода.

Команда `cpio` может использоваться для архивирования любого набора файлов и специальных файлов. Команда `cpio` сохраняет информацию эффективнее, чем `tar`, пропускает сбойные сектора или блоки при восстановлении данных, архивы могут быть восстановлены в ОС.

Недостатком команды `cpio` является необходимость использовать язык программирования оболочки для создания соответствующего сценария, чтобы обновить архив.

В таблице 70 приведены основные параметры команды `cpio`.

Таблица 70

Параметр	Назначение
`-o`	Создание архива в стандартное устройство вывода
`-l`	Восстановление файлов из архива, передаваемого на стандартное устройство ввода
`-t`	Создание списка содержимого стандартного устройства ввода

Подробное описание команды приведено в `man cpio`.

Примеры:

1. Копирование файлов из каталога `/home` в архив `home.cpio`

```
find /home/* | cpio -o > /tmp/home.cpio
```

2. Восстановление файлов из архива `home.cpio` с сохранением дерева каталогов и создание списка в файле `bkup.index`

```
cpio -id < /tmp/home.cpio > bkup.index
```

3. Использование команды `find` для поиска измененных за последние сутки файлов и сохранение их в архив `home.new.cpio`

```
find /home -mtime 1 -type f | cpio -o > /tmp/home.new.cpio
```

4. Восстановление файла `/home/dave/notes.txt` из архива `home.cpio`

```
cpio -id /home/dave/notes.txt < home.cpio
```

Для восстановления файла с помощью `cpio` следует указывать его полное имя.

Можно автоматизировать выполнение данных команд, поместив их в файл `crontab` суперпользователя. Например, следующая запись в файле `crontab` выполняет резервное копирование каталога `/home` ежедневно в 01:30:

```
30 01 *** ls /home : cpio -o > /tmp/home.cpio
```

При необходимости более сложного резервного копирования можно создать соответствующий сценарий оболочки. Запуск подобных сценариев также может быть осуществлен посредством `cron`.

Создание резервных копий означает определение политики создания резервных копий для снижения потерь и восстановления информации после возможной аварии системы.

17. СРЕДСТВА РАЗГРАНИЧЕНИЯ ДОСТУПА К ПОДКЛЮЧАЕМЫМ УСТРОЙСТВАМ

В ОС поддерживается разграничение доступа к символьным и блочным устройствам, для которых в каталоге `/dev` создаются файлы устройств. Разграничение доступа реализуется с использованием генерации правил менеджера устройств `udev`. Для разграничения доступа к устройствам типа видеокарт, сетевых карт и т.д. данный метод не используется.

Для решения задачи разграничения доступа к устройствам на основе генерации правил менеджера устройств `udev` в ОС реализованы:

- средства разграничения доступа к устройствам на основе правил `udev`;
- средства регистрации (учета) устройств.

Средства разграничения доступа к устройствам на основе генерации правил `udev` обеспечивают дискреционное и мандатное управление доступом пользователей к устройствам, подключаемым, в первую очередь, через интерфейс USB: сканерам, съемным накопителям, видеокамерам и т. п. Описание правил генерации и порядок их применение приведены в 17.2-17.6.

Средства регистрации устройств обеспечивают учет подключаемых устройств и съемных носителей в системе, установку дискреционных и мандатных атрибутов доступа пользователей к устройствам и создание дополнительных правил доступа к устройству (например, ограничение на подключение устройства только в определенный USB-порт), Описание порядка регистрации устройств приведено в 17.8.

17.1. Монтирование съемных накопителей

При монтировании блочных устройств используется утилита `mount`, модифицированная для монтирования устройства владельцем или пользователем, входящим в указанную группу. В процессе монтирования от имени пользователя ожидается два параметра: наименование файла устройства и наименование точки монтирования. Остальные параметры монтирования выбираются из файлов `/etc/fstab` и `/etc/fstab.pdac` с использованием регулярных выражений. При этом монтирование ФС съемных накопителей от имени пользователя (в т. ч. с использованием графической утилиты `fly-fm`) осуществляется в каталог `/run/user/$uid/media`, а с использованием командной строки — в каталог, указанный в файле `/etc/fstab`.

Для предоставления локальным пользователям возможности монтирования ФС съемных накопителей необходимо наличие в файле `/etc/fstab` следующей записи:

```
/dev/s*  /home/*/media/*  auto owner,group,noauto,noexec 0 0
```

Для предоставления пользователям ALD возможности монтирования ФС съемных накопителей необходимо наличие в файле `/etc/fstab` следующей записи:

```
/dev/s* /ald_home/*/media/* auto owner,group,noauto,noexec 0 0
```

Для одновременного предоставления локальным пользователям и пользователям ALD возможности монтирования ФС съемных накопителей необходимо наличие в файле /etc/fstab следующей записи:

```
/dev/s*  /*home/*/media/* auto owner,group,noauto,noexec 0 0
```

По умолчанию для монтирования различных ФС, содержащихся в учтенных разделах на блочных устройствах USB-накопителей, в файл /etc/fstab.pdac включены следующие записи:

```
/dev/*fat  /run/user/*/media/*  auto
owner,group,noauto,nodev,noexec,iocharset=utf8,defaults  0  0
/dev/*ntfs*  /run/user/*/media/*  auto
owner,group,noauto,nodev,noexec,iocharset=utf8,defaults  0  0
/dev/sd*ext*  /run/user/*/media/*  auto
owner,group,noauto,nodev,noexec,defaults  0  0
```

По умолчанию для монтирования различных ФС, содержащихся на учтенных CD/DVD-дисках, в файл /etc/fstab.pdac включены следующие записи:

```
/dev/s*udf  /run/user/*/media/*  udf
owner,group,nodev,noexec,noauto,defaults  0  0
/dev/s*iso9660  /run/user/*/media/*  iso9660
owner,group,nodev,noexec,noauto,defaults  0  0
```

По умолчанию монтирование ФС, содержащихся в неучтенных разделах на блочных устройствах USB-накопителей, разрешено пользователям, входящим в группу floppy. В данном случае монтирование будет осуществляться в соответствии со следующей записью из файла /etc/fstab.pdac:

```
/dev/sd*  /run/user/*/media/*  auto
owner,group,noauto,nodev,noexec,iocharset=utf8,defaults  0  0
```

Для возможности монтирования ФС ext*, содержащихся в неучтенных разделах на блочных устройствах USB-накопителей, необходимо в файле /etc/fstab.pdac из записи для устройств /dev/sd* удалить неподдерживаемый для данной ФС параметр монтирования iocharset=utf8:

```
/dev/sd*  /run/user/*/media/*  auto
owner,group,noauto,nodev,noexec,defaults  0  0
```

Для монтирования пользователями ФС, содержащихся на неучтенных CD/DVD-дисках, в конец файла /etc/fstab необходимо включить следующую запись:

```
/dev/sr*  /*home/*/media/*  udf,iso9660  user,noauto  0  0
```

ВНИМАНИЕ! При монтировании ФС, поддерживающей атрибуты UNIX и расширенные атрибуты, права доступа на файл учтенного устройства не будут совпадать с правами доступа в ФС. Использование мандатных атрибутов будет ограничено атрибутами, установленными для файла устройства.

ВНИМАНИЕ! Использование учтенного USB-носителя с ФС VFAT возможно только при входе в систему на том уровне конфиденциальности, который назначен администратором для этого устройства.

ВНИМАНИЕ! При включении режима работы с отчуждаемыми носителями с конфиденциальной информацией все непривилегированные пользователи должны быть исключены из группы `floppy`.

ВНИМАНИЕ! При включении режима работы с CD/DVD-дисками с конфиденциальной информацией все непривилегированные пользователи должны быть исключены из группы `cdrom`.

ВНИМАНИЕ! Использование учтенного USB-носителя с ФС ext4 (ext3) возможно пользователями на разных доступных им уровнях конфиденциальности. При этом администратор должен зарегистрировать носитель для данного пользователя на требуемых уровнях и создать на ФС носителя систему каталогов с необходимыми уровнями конфиденциальности. Например, для обеспечения работы на нескольких уровнях на USB-носителе с ФС ext4 администратор может использовать следующий сценарий, задав необходимые переменные `USERNAME` и `DEVICE`:

```
USERNAME="user"
DEVICE="/dev/sdc1"
mkfs.ext4 $DEVICE
mkdir -p /media/usb
mount $DEVICE /media/usb
#multilevel
pdpl-file 3:0:-1:ccnr /media/usb/
mkdir /media/usb/{0,1,2,3}
pdpl-file 0:0:0:0 /media/usb/0
pdpl-file 1:0:0:0 /media/usb/1
pdpl-file 2:0:0:0 /media/usb/2
pdpl-file 3:0:0:0 /media/usb/3
chown -R ${USERNAME}:${USERNAME} /media/usb/{0,1,2,3}
ls -la /media/usb/
pdp-ls -M /media/usb/
umount /media/usb
```

17.2. Перехват события менеджером устройств udev

Менеджер устройств `udev` перехватывает события, возникающие при изменении статуса подключенных устройств. Основные события:

- подключение устройства (событие `add`);
- отключение устройства (событие `remove`).

Перехват событий осуществляется с помощью правил udev. Файлы с правилами, перехватывающими события, располагаются в следующих каталогах, при этом правила обрабатываются в порядке их нахождения в каталогах в следующей последовательности:

1) правила из каталога /lib/udev/rules.d/;
2) правила из каталога /run/udev/rules.d/;
3) правила из каталога /etc/udev/rules.d/.

Перед выполнением правил файлы упорядочиваются по алфавиту. Файлы с одинаковыми именами переписываются последним найденным файлом, т.е. файл, найденный в последнем каталоге (/etc/udev/rules.d/) заменит собой ранее найденный файл. Имя файла правила имеет расширение .rules.

Пример

Правило перехвата события /etc/udev/rules.d/99-local.rules

```
KERNEL=="sd[a-z][0-9]", SUBSYSTEMS=="usb", ACTION=="add", RUN+="/bin/systemctl
start usb-mount@%k.service"
KERNEL=="sd[a-z][0-9]", SUBSYSTEMS=="usb", ACTION=="remove",
RUN+="/bin/systemctl stop usb-mount@%k.service"
```

Данное правило обрабатывает события подключения (add) и отключения (remove) дисковых устройств с именами, начинающимися с букв sd, после которых следует одна любая строчная буква ([a-z]) и одна цифра ([0-9]).

Правило при этом не выполняет прямых действий, а вызывает системную службу usb-mount@%k.service, то есть вызывает сценарий обработки события как системную службу.

При выполнении сценария обработки событий служба udev вместо переменной %k подставляет имя устройства, т.е. при подключении, например, устройства /dev/sdb1 будет выполняться команда:

```
/bin/systemctl start usb-mount@sdb1.service
```

При вызове службы, в имени которой содержится символ @, системная служба вызова служб разделит это имя на части и передаст часть, находящуюся после символа @, как параметр вызываемой службы. Т.е. вызов:

```
systemctl start usb-mount@sdb1.service
```

будет обработан как вызов службы usb-mount с параметрами start и sdb1.

Подробное описание параметров и переменных правил udev приведено в руководстве man udev.

17.3. Разграничение доступа к устройствам на основе генерации правил udev

Разграничение доступа к устройству осуществляется на основе генерации правил для менеджера устройств udev, которые хранятся в соответствующих файлах в каталогах /etc/udev/rules.d и /run/udev/rules.d. Генерация правил осуществляется автоматически для символьных и блочных устройств с использованием базы учета устройств, ведущейся в локальной системе (файл /etc/parsec/PDAC/devices.cfg) или в ALD/FreeIPA (см. раздел 8).

Для устройств, учитываемых в локальной базе, генерация правила осуществляется при сохранении информации об устройстве с использованием утилиты fly-admin-smc. Для устройств, учитываемых в базе ALD или FreeIPA, генерация правил осуществляется PAM-модулем pam_ald_mac при входе пользователя в систему. При этом правила генерируются для всех устройств, учтенных в базе, вне зависимости от имени пользователя, осуществляющего вход в систему, и имени хоста, на котором выполняется вход.

Пример

Правило для съемного USB-накопителя

```
ENV{ID_SERIAL}=="JetFlash_TS256MJF120_OYLIXNA6-0:0", OWNER="user",
GROUP="users" PDPL="3:0:f:0!:" AUDIT="0:0x0:0x0"
```

Правило для съемного USB-накопителя с серийным номером JetFlash_TS256MJF120_OYLIXNA6-0:0 разрешает использование данного накопителя владельцу устройства (пользователю user) и пользователям, входящим в группу users. Для устройства установлены мандатные атрибуты:

- уровень конфиденциальности — 3;
- уровень целостности — 0;
- категории — f;
- роли и административные роли отсутствуют,

а флаги аудита не установлены.

17.4. Вызов сценария обработки события как системной службы

Для вызова системных служб используются соответствующие юниты службы systemd, расположенные в каталоге /etc/systemd/system/.

Юнит с именем usb-mount@.service для вызова службы перехвата события udev может быть записан в следующем виде:

```
[Unit]
Description=Mount USB Drive on %i
[Service]
```

```
Type=oneshot
RemainAfterExit=true
ExecStart=/usr/local/bin/usb-mount.sh add %i
ExecStop=/usr/local/bin/usb-mount.sh remove %i
```

Данный юнит при выполнении команд `start` и `stop` вызывает исполняемый файл сценария обработки события `/usr/local/bin/usb-mount.sh`.

При вызове сценария вместо параметра `%i` будет подставлена часть имени вызова службы, находящаяся после символа `@`.

17.5. Сценарий обработки события

Сценарий обработки события может быть размещен в любом каталоге.

Пример

Сценарий обработки события `/usr/local/bin/usb-mount.sh`

```
# Этот сценарий вызывается из системного юнита как сценарий обработки
# подключения/отключения накопителей.
usage() {
echo "Использование: $0 {add|remove} device_name (например, sdb1)"
exit 1
}

if [[ $# -ne 2 ]]; then
usage
fi

ACTION=$1
DEVBASE=$2
DEVICE="/dev/${DEVBASE}"

# Проверяем, не примонтировано ли уже устройство
MOUNT_POINT=$(/bin/mount | /bin/grep ${DEVICE} | /usr/bin/awk
     '{ print $3 }')

do_mount() {
if [[ -n ${MOUNT_POINT} ]]; then
echo "Предупреждение: ${DEVICE} уже смонтировано в ${MOUNT_POINT}"
exit 1
fi
```

```
# Получаем информацию об устройстве : метка $ID_FS_LABEL, идентификатор
# $ID_FS_UUID, и тип файловой системы $ID_FS_TYPE
eval $(/sbin/blkid -o udev ${DEVICE})

# Создаем точку монтирования:
LABEL=${ID_FS_LABEL}
if [[ -z "${LABEL}" ]]; then
LABEL=${DEVBASE}
elif /bin/grep -q " /media/${LABEL} " /etc/mtab; then
# Если точка монтирования уже существует изменяем имя:
LABEL+="-${DEVBASE}"
fi
MOUNT_POINT="/media/${LABEL}"
echo "Точка монтирования: ${MOUNT_POINT}"
/bin/mkdir -p ${MOUNT_POINT}

# Глобальные параметры монтирования
OPTS="rw,relatime"

# Специфические параметры монтирования:
if [[ ${ID_FS_TYPE} == "vfat" ]]; then
OPTS+=",users,gid=100,umask=000,shortname=mixed,utf8=1,flush"
fi

if ! /bin/mount -o ${OPTS} ${DEVICE} ${MOUNT_POINT}; then
echo "Ошибка монтирования ${DEVICE} (статус = $?)"
/bin/rmdir ${MOUNT_POINT}
exit 1
fi

echo "**** Устройство ${DEVICE} смонтировано в ${MOUNT_POINT} ****"
}

do_unmount() {
if [[ -z ${MOUNT_POINT} ]]; then
echo "Предупреждение: ${DEVICE} не смонтировано"
else
/bin/umount -l ${DEVICE}
```

```
echo "**** Отмонтировано ${DEVICE}"
fi

# Удаление пустых каталогов
for f in /media/* ; do
if [[ -n $(/usr/bin/find "$f" -maxdepth 0 -type d -empty) ]]; then
if ! /bin/grep -q " $f " /etc/mtab; then
echo "**** Удаление точки монтирования $f"
/bin/rmdir "$f"
fi
fi
done
}

case "${ACTION}" in
add) do_mount ;;
remove) do_unmount ;;
*) usage ;;
esac
```

После создания файла сценария сделать его исполнимым, выполнив от имени администратора команду:

```
chmod +x /usr/local/bin/usb-mount.sh
```

17.6. Порядок генерации правил udev для учета съемных накопителей

Съемный накопитель всегда является блочным устройством (`block`). Съемный накопитель всегда является устройством типа «диск» (`disk`) или типа «дисковый раздел» (`partition`), при этом правила МРД, применяемые для реализации учета съемных накопителей, применяются к дисковыми разделами.

Назначение мандатных атрибутов съемному накопителю выполняется при его подключении, при этом операции подключения выполняются раздельно для самого накопителя и для всех находящихся на этом накопителе дисковых разделов.

Правила `udev` применяются к устройствам при совпадении заданных в правиле параметров и параметров устройства. Все параметры подключенного устройства можно просмотреть, выполнив команду:

```
sudo udevadm info --query=property --name=/dev/<имя_устройства>
```

При генерации правил для блочных устройств не рекомендуется использовать параметры, относящиеся к подключению этих устройств к ОС (например, параметры DEVNAME, ID_BUS и др.), так как они:

- могут повторяться для разных устройств (присвоение имени sdX);
- могут зависеть от порядка подключения устройств (присвоение имени sdX);
- могут изменяться при изменении аппаратной конфигурации;
- могут отличаться на разных доменных компьютерах, имеющих разную аппаратную конфигурацию.

Параметры, применимые для идентификации съемных накопителей типа «диск» (также применимы к устройствам типа «дисковый раздел», которыми наследуются от устройства «диск»): ID_VENDOR, ID_VENDOR_ID, ID_VENDOR_ENC, ID_MODEL, ID_MODEL_ID, ID_MODEL_ENC, ID_SERIAL, ID_SERIAL_SHORT.

Дополнительно к устройствам типа «дисковый раздел» применимы параметры: ID_FS_LABEL, ID_FS_LABEL_ENC, ID_PART_ENTRY_NUMBER, ID_FS_TYPE, ID_FS_USAGE, ID_FS_UUID, ID_FS_UUID_ENC, ID_FS_VERSION, ID_PART_ENTRY_NUMBER.

Основным минимальным параметром идентификации съемного накопителя является его серийный номер (ID_SERIAL или ID_SERIAL_SHORT).

Для идентификации накопителей при использовании оборудования разных моделей и разных производителей можно использовать набор параметров «Производитель» — «Модель» — «Серийный номер» (например, ID_VENDOR, ID_MODEL, ID_SERIAL или ID_VENDOR_ID, ID_MODEL_ID, ID_SERIAL и т.д.).

С учетом того, что на одном устройстве может располагаться несколько дисковых разделов, в дополнение к параметрам идентификации накопителя для идентификации дисковых разделов можно использовать метку файловой системы (ID_FS_LABEL), универсальный идентификатор файловой системы UUID (ID_FS_UUID) и номер раздела на накопителе (ID_PART_ENTRY_NUMBER).

Пример

Правило идентификации дискового раздела по серийному номеру устройств и UUID

```
# отсекаются ненужные устройства - вероятность несовпадения серийного номера
# выше, правило сработает чаще

ENV{ID_SERIAL}!="SanDisk_Cruzer_Glide_XXXXXXXXXXXXX-0:0", GOTO="END"
ENV{ID_FS_UUID}!="0047-C44D", GOTO="END"

# отсекаются ненужные события
ACTION!="add",              GOTO="END"
```

```
ENV{SUBSYSTEM}!="block",   GOTO="END"
ENV{DEVTYPE}!="partition", GOTO="END"

# настройка правил Parsec
OWNER="user", GROUP="root", MODE="740", PDPL="0:0:0x0:0x0!:", AUDIT="o:0x0:0x0"
ENV{ID_FS_TYPE}=="?*", SYMLINK+="%k_$env{ID_FS_TYPE}",
RUN+="/bin/ln -f /dev/%k /dev/%k_$env{ID_FS_TYPE}"

LABEL="END"
```

17.7. Отладка правил

Включение вывода отладочных сообщений в файл `/var/log/syslog`:

```
udevadm control -l debug
```

Тестовая отработка правил `udev` без их загрузки:

```
udevadm test /dev/sdb1
```

Мониторинг событий `udev`:

```
udevadm monitor -k -u -p
```

Путь к устройству:

```
udevadm info -q path -n /dev/sdd1
```

Полная информация об устройстве:

```
udevadm info -a -p $(udevadm info -q path -n /dev/sdd1)
```

17.8. Регистрация устройств

Регистрация устройств в локальной базе учета устройств осуществляется с использованием графической утилиты управления политикой безопасности `fly-admin-smc`.

Регистрация устройств в базе учета устройств ALD осуществляется с использованием графической утилиты управления политикой безопасности `fly-admin-smc` (`fly-admin-ald`) или утилиты командной строки `ald-admin`.

Регистрация устройств в базе учета устройств FreeIPA осуществляется с использованием web-интерфейса контроллера домена путем создания записей об этих устройствах и глобальных правил. Про этом графическая утилита управления политикой безопасности `fly-admin-smc` позволяет скопировать атрибуты регистрируемых устройств в web-интерфейс FreeIPA через буфер обмена.

Устройства идентифицируются на основе атрибутов менеджера устройств `udev`. В большинстве случаев достаточно использовать серийный номер `ID_SERIAL`. В случае, когда использование для идентификации устройства серийного номера невозможно, необходимо выбрать один или несколько других атрибутов, обеспечивающих идентификацию устройства.

Для предоставления локальным пользователям и пользователям ALD доступа к устройствам (USB-накопители, сканеры, оптические носители) по классификационной метке необходимо выполнить следующие действия:

1) запустить от имени администратора через механизм `sudo` утилиту управления политикой безопасности `fly-admin-smc` (см. электронную справку) и выбрать в дереве объектов в боковой панели «Устройства и правила – Устройства»;

2) нажать кнопку **[Создать новый элемент]** на панели инструментов. Дождаться появления графического окна и подключить устройство одним из следующих способов в зависимости от типа устройства:

- подключить USB-накопитель к USB-порту компьютера;
- подключить кабель USB-сканера к USB-порту компьютера;
- вставить оптический носитель в устройство чтения CD/DVD-дисков.

3) в появившемся перечне выбрать устройство и открыть его «Свойства»;

4) в списке свойств устройства должны быть отмечены строки следующего вида:

- для USB-накопителей (отмечено по умолчанию):

```
ID_SERIAL       Значение
```

- для сканеров (отмечено по умолчанию):

```
ID_SERIAL       Значение
PRODUCT         Значение
```

- для оптических носителей (отмечено по умолчанию):

```
ID_SERIAL       Значение
```

позволяет идентифицировать устройства, на которых будет осуществляться работа с оптическими носителями, и:

```
ID_FS_LABEL     Значение
```

позволяет идентифицировать оптический носитель.

При необходимости можно выбрать другие свойства;

5) добавить устройство, нажав кнопку **[Да]**;

6) в поле «Наименование» указать наименование устройства;

7) во вкладке «Общие» необходимо выбрать пользователя, группу (владельца устройства) и задать права доступа для пользователя, группы и всех остальных;

8) указать классификационную метку, для этого во вкладке «МРД» выбрать иерархический уровень конфиденциальности и указать набор неиерархических категорий конфиденциальности;

9) назначить параметры регистрации событий, связанных с устройством. Для этого во вкладке «Аудит» необходимо выбрать событие и результат («Успех», «Отказ»), подлежащие регистрации;

10) назначить дополнительные наборы правил для устройства из списка правил, созданных во вкладке боковой панели «Устройства и правила — Правила» (в данной вкладке создается набор правил для менеджера устройств udev (см. 17.3);

11) применить изменения, нажав кнопку **[Применить изменения]** на панели инструментов.

Для предоставления пользователям FreeIPA доступа к USB-накопителям по классификационной метке необходимо выполнить следующие действия:

1) запустить от имени администратора через механизм sudo утилиту управления политикой безопасности fly-admin-smc (см. электронную справку) и выбрать в дереве объектов в боковой панели «Устройства и правила – Устройства»;

2) нажать кнопку **[Создать новый элемент]** на панели инструментов. Дождаться появления графического окна и подключить USB-накопитель к USB-порту компьютера;

3) в появившемся перечне выбрать устройство и открыть его «Свойства»;

4) в списке свойств устройства должны быть отмечены строки следующего вида:

```
ID_SERIAL       Значение
```

5) скопировать значение правила;

6) в web-интерфейсе контроллера домена FreeIPA перейти «Политика — Политика PARSEC» и в выпадающем списке выбрать «Registerd device»;

7) задать имя регистрируемого носителя, права для пользователя и группы, в поле «Device attributes» вставить скопированное в fly-admin-smc правило, установить флаг «Device is ON» и сохранить правило, нажав кнопку **[Добавить]**;

8) прервать процедуру создания локального правила в fly-admin-smc без сохранения изменений;

9) для подготовки USB-носителя к работе в ненулевой сессии на одном уровне конфиденциальности:

а) создать сценарий singlelevel.sh со следующим текстом, задав соответствующие значения для параметров USERNAME и DEVICE:

```
#!/bin/bash
USERNAME="user"
DEVICE="/dev/sdc1"
mkfs.ext4 $DEVICE
mkdir -p /media/usb
mount $DEVICE /media/usb
#one level
pdpl-file 2:0:0:0 /media/usb/
chown -R ${USERNAME}:${USERNAME} /media/usb/
```

```
ls -la /media/usb/
pdp-ls -M /media/usb/
umount /media/usb
```

б) сделать скрипт исполняемым, выполнив команду:

```
sudo chmod +x singlelevel.sh
```

в) в web-интерфейсе управления доменом FreeIPA («Политика — Политика PARSEC») создать глобальное правило использования требуемого устройства для соответствующего уровня;

10) для подготовки USB-носителя к работе в ненулевой сессии на нескольких уровнях конфиденциальности:

а) создать сценарий `multilevel.sh` со следующим текстом, задав соответствующие значения для параметров `USERNAME` и `DEVICE`:

```
#!/bin/bash
USERNAME="user"
DEVICE="/dev/sdc1"
mkfs.ext4 $DEVICE
mkdir -p /media/usb
mount $DEVICE /media/usb
#multilevel
pdpl-file 3:0:-1:ccnr /media/usb/
mkdir /media/usb/{0,1,2,3}
pdpl-file 0:0:0:0 /media/usb/0
pdpl-file 1:0:0:0 /media/usb/1
pdpl-file 2:0:0:0 /media/usb/2
pdpl-file 3:0:0:0 /media/usb/3
chown -R ${USERNAME}:${USERNAME} /media/usb/{0,1,2,3}
ls -la /media/usb/
pdp-ls -M /media/usb/
umount /media/usb
```

б) сделать скрипт исполняемым, выполнив команду:

```
sudo chmod +x multilevel.sh
```

в) в web-интерфейсе управления доменом FreeIPA («Политика — Политика PARSEC») создать глобальные правила использования требуемого устройства для каждого из созданных уровней.

После переподключения устройства владелец устройства или пользователи из группы смогут монтировать устройство, при этом на точку монтирования будет устанавли-

ваться указанная классификационная метка (иерархический уровень конфиденциальности и неиерархические категории конфиденциальности).

ВНИМАНИЕ! В случае если включен мандатный контроль целостности, то действия по предоставлению пользователям доступа к устройствам должны осуществляться от имени администратора на высоком уровне целостности (по умолчанию 63).

18. ПОДДЕРЖКА СРЕДСТВ ДВУХФАКТОРНОЙ АУТЕНТИФИКАЦИИ

Повышение надежности аутентификации возможно путем применения многофакторной аутентификации, т. е. аутентификации, в процессе которой используются аутентификационные факторы нескольких типов.

К факторам, которые могут быть использованы, относятся:

- ввод пароля или PIN-кода;
- ввод одноразовых паролей (скрэтч-карты);
- предоставление физического устройства или носителя, содержащего аутентификационную информацию (смарт-карта, USB-токен и т. п.);
- предоставление биометрической информации (отпечатки пальцев, изображение сетчатки глаза и т. п.).

На практике в большинстве случаев используется двухфакторная аутентификация на основе ввода пароля с одновременным предоставлением пользователем физического устройства (носителя), содержащего дополнительную аутентификационную информацию. Дополнительной аутентификационной информацией в этом случае обычно является размещенный на устройстве сертификат пользователя.

Для обеспечения двухфакторной аутентификации с помощью внешнего носителя используются следующие средства и технологии:

- PKCS (Public Key Cryptography Standard) — группа стандартов криптографии с открытым ключом, в частности, стандарты PKCS-11, PKCS-12, PKCS-15, относящиеся к работе с криптографическими токенами;
- X.509 — стандарт, определяющий форматы данных и процедуры распределения открытых ключей с помощью сертификатов с цифровыми подписями, которые предоставляются удостоверяющими центрами сертификации (Certification Authority (CA));
- OpenSC — набор программных утилит и библиотек для работы с носителями аутентификационной информации пользователя (смарт-карты, USB-токены), содержащие функции аутентификации, криптографии и цифровой подписи. Поддерживает стандарты PKCS-11, PKCS-15;
- OpenCT — набор драйверов устройств для работы с носителями аутентификационной информации (устаревший);
- OpenSSL — программное средство для работы с криптографическим протоколом SSL/TLS. Позволяет создавать ключи RSA, DH, DSA и сертификаты X.509, подписывать их, формировать файлы сертификатов CSR и CRT. Также имеется возможность тестирования SSL/TLS соединений. Поддерживает механизм динамически подключаемых библиотек алгоритмов защитного преобразования данных, т.е. механизм

подключения внешних модулей, содержащих дополнительные алгоритмы. С использованием указанного механизма обеспечивает работу с алгоритмами защитного преобразования данных в соответствии с требованиями ГОСТ (пакет библиотеки алгоритмов ГОСТ `libgost-astra`);

- PC/SC — набор спецификаций для доступа к смарт-картам;
- PKINIT (Public Key Cryptography for Initial Authentication in Kerberos) — стандарт использования криптографии с открытым ключом в качестве фактора аутентификации в протоколе аутентификации Kerberos (см. 8.1.4).

Двухфакторная аутентификация может применяться как в случае использования локальной аутентификации, так и в случае использования ЕПП.

18.1. Аутентификация с открытым ключом (инфраструктура открытых ключей)

При доступе к ресурсам информационных систем часто используются криптографические механизмы, основанные на ассиметричных криптографических алгоритмах и сертификатах открытого ключа. Применение указанных механизмов в информационных системах обеспечивается инфраструкторой открытых ключей PKI, которая включает в себя набор аппаратных и программных средств, политик и процедур создания, управления, распространения, использования и отзыва цифровых сертификатов.

В основе PKI лежит использование криптографической системы с открытым ключом и несколько основных принципов:

- закрытый ключ известен только его владельцу;
- удостоверяющий центр создает сертификат открытого ключа, таким образом удостоверяя этот ключ;
- никто не доверяет друг другу, но все доверяют удостоверяющему центру;
- удостоверяющий центр подтверждает или опровергает принадлежность открытого ключа заданному лицу, которое владеет соответствующим закрытым ключом.

Аутентификация на основе ключей использует два ключа, один «открытый» (публичный ключ), который доступен каждому, и второй «закрытый» (секретный ключ), который доступен только владельцу. В процессе аутентификации используются криптографические алгоритмы с открытым ключом для проверки подлинности пользователя. При этом секретный ключ находится непосредственно у пользователя, а открытый ключ по защищенным каналам связи передается в те системы, которые должны с его помощью проверять подлинность пользователя.

В качестве электронного представления ключей используются цифровые сертификаты. Сертификат является удостоверением принадлежности открытого ключа. Цифровой

сертификат устанавливает и гарантирует соответствие между открытым ключом и его владельцем. Сертификаты выдаются специальными уполномоченными организациями — CA. Сертификаты могут быть использованы не только для аутентификации, но и для предоставления избирательных прав доступа, в том числе и права подписи других сертификатов.

В рамках изолированной информационной системы средством выработки и подписывания цифровых сертификатов могут быть использованы различные программные средства, например, `openssl`. В этом случае такое средство может выступать в роли локального удостоверяющего центра для создания ключевых пар и сертификатов клиентов и серверов системы.

18.2. Средства поддержки двухфакторной аутентификации

18.2.1. Общие сведения

В ОС поддерживается механизм двухфакторной аутентификации пользователей с использованием токенов (USB-ключей, криптографических ключей).

Для аутентификации пользователей используется модуль `pam-csp`, реализованный на основе стандартного PAM-модуля `libpam-csp`.

PAM-модуль `libpam-csp` обрабатывает два события:

- аутентификация пользователя;
- смена пароля пользователя.

Для доступа к токенам используется стандартная библиотека `opensc-pkcs11`, позволяющая модулю `libpam-csp` работать с любыми токенами различных производителей, поддерживающими эту библиотеку.

Контроль пользовательской сессии осуществляется с использованием службы `csp-monitor`. Для взаимодействия службы с токенами используется библиотека `opensc-pkcs11`.

Служба `csp-monitor` принимает от `pam_csp` по шине DBus сообщения о входе и выходе пользователя с использованием токена и поддерживает список текущих пользовательских сессий с информацией об использованных для входа токенах.

Служба `csp-monitor` осуществляет мониторинг подключений и отключений USB-устройств и, в случае если какой-либо токен из числа участвующих в аутентификации пользователя был вынут, блокирует все сессии данного пользователя. Для разблокировки сессии пользователь должен подключить токен и ввести PIN-код.

Служба `csp-monitor` управляется как юнит `systemd`. Для просмотра статуса службы выполнить команду:

```
systemctl status csp-monitor
```

ВНИМАНИЕ! При использовании решения `pam_csp` совместно с FreeIPA для параметра доменной политики паролей «минимальный срок действия пароля» должно быть задано значение 0.

18.2.2. Настройка клиентской машины

Для установки модуля `libpam-csp` выполнить установку соответствующего пакета от имени администратора командой:

```
apt install libpam-csp
```

Далее необходимо задать команду принудительной смены пароля. Для локальных пользователей на компьютере пользователя выполнить команду от имени администратора:

```
passwd --expire <имя_пользователя>
```

Для доменных пользователей необходимо использовать соответствующие инструменты администрирования домена.

При установке пакета `libpam-csp` автоматически будет установлен пакет для службы `csp-monitor`.

Во время установки пакета модуль `pam_csp` регистрируется первым в цепочках PAM-модулей в двух PAM-профилях:

```
/etc/pam.d/common-auth
/etc/pam.d/common-password
```

18.2.3. Инициализация токена

Процесс инициализации токена одинаков для локальных и доменных пользователей.

До передачи токена пользователю выполняется его подготовка на компьютере администратора, ответственного за подготовку.

Для выполнения подготовки токена на компьютере должны быть установлены пакеты:

- `opensc-pkcs11` версии не ниже 0.19.0-2;
- `ifd-rutokens` версии не ниже 1.0.4 (для Rutoken S и Rutoken ECP);
- пакеты других интерфейсных модулей, необходимые для используемой модели токена.

Для установки пакета `opensc-pkcs11` выполнить от имени администратора команду:

```
sudo apt install opensc-pkcs11
```

Установка интерфейсных модулей выполняется в соответствии с инструкциями производителей соответствующих токенов.

Процедура инициализации зависит от используемой модели токена.

Для инициализации токена Rutoken S выполнить последовательно следующие команды:

```
pkcs15-init --erase-card
```

```
pkcs15-init --create-pkcs15 --so-pin "87654321" --so-puk "" --pin "12345678"
pkcs15-init --store-pin --label "User PIN" --auth-id 02 --pin "12345678"
     --puk ""
```

Для инициализации токена Rutoken ECP выполнить последовательно следующие команды:

```
pkcs15-init --erase-card -p rutoken_ecp
pkcs15-init --create-pkcs15 --so-pin "87654321" --so-puk ""
pkcs15-init --store-pin --label "User PIN" --auth-id 02 --pin "12345678"
     --puk "" --so-pin "87654321" --finalize
```

Проверить, что токен успешно инициализирован, можно с помощью команды:

```
pkcs15-tool -D
```

18.2.4. Использование токена

При первичном использовании токена для входа в свою сессию пользователь должен подключить токен и в соответствующих полях ввести свои логин и пароль. При появлении окна с дополнительным приглашением:

```
Supply token PIN:
```

ввести PIN токена (текущий PIN токена пользователю сообщает администратор).

Далее пользователю будет предложено сменить PIN:

```
Supply new token PIN:
Retype new token PIN:
```

При этом можно указать новый PIN (рекомендуется), введя его два раза, или два раза нажать клавишу **<Enter>**, чтобы оставить текущий PIN (не рекомендуется).

После первичного ввода PIN произойдет генерация нового случайного пароля, его назначение учетной записи пользователя и будет выполнен вход в систему. В дальнейшем в токене будет храниться 16-символьный пароль, недоступный без знания PIN.

При последующих входах в систему пользователю нужно подключить токен и далее в соответствующих полях ввести логин и PIN токена.

Пример

Диалог при терминальном входе

```
login: user
Supply token PIN:
```

При необходимости сменить пароль пользователь должен подключить токен, войти в систему и затем:

1) при первичном входе — выполнить в командной строке `passwd`. При этом будут запрошены текущие пароль и PIN:

```
passwd
```

```
Введите ПИН-код :
Введите текущий пароль :
Введите новый ПИН-код :
Введите новый ПИН-код еще раз :
```

2) при последующих входах — выполнить в командной строке `passwd`. При этом будет запрошен текущий PIN:

```
passwd
Введите ПИН-код :
Введите новый ПИН-код :
Введите новый ПИН-код еще раз :
```

Для локального пользователя администратор может подготовить токен со сгенерированным паролем заранее. Для этого следует подключить токен и выполнить команду:

```
passwd <имя_пользователя>
Введите ПИН-код :
Введите новый ПИН-код :
Введите новый ПИН-код еще раз :
ПИН-код успешно изменен.
passwd: пароль успешно изменен
```

18.2.5. Разблокировка сессии с ненулевой меткой конфиденциальности с помощью PIN-кода

Токен возможно использовать для входа в сессию с ненулевым уровнем конфиденциальности. При этом для того чтобы функция разблокировки сессии по PIN-коду работала корректно, необходимо произвести следующие настройки:

1) присвоить сокету `/var/run/pcscd/pcscd.comm` привилегию `PARSEC_CAP_PRIV_SOCK`, добавив в раздел `[Socket]` файла `/lib/systemd/system/pcscd.socket` строку:

```
CapabilitiesParsec=PARSEC_CAP_PRIV_SOCK
```

2) перезапустить службу `pcscd`:

```
sudo systemctl daemon-reload
sudo systemctl stop pcscd.service
sudo systemctl stop pcscd.socket
sudo systemctl start pcscd.service
```

3) обеспечить корректную работу модуля `pam_p11`, входящего в состав `libpam-p11`. Для этого в каталоге `/home` каждого доступного пользователю уровня конфиденциальности должен находиться файл `.euid/authorized_certificates`. Данный файл можно после настройки модуля `pam_p11` скопировать из каталога `/home` поль-

зователя нулевого уровня конфиденциальности в каталоги `/home` других уровней конфиденциальности.

18.3. Управление сертификатами

Для обеспечения аутентификации с открытым ключом в информационной системе необходимо иметь набор ключевых пар и сертификатов ресурсов сети (серверов или служб) и ее клиентов (пользователей). Формирование и подписывание сертификатов выполняется с помощью удостоверяющего центра информационной системы. Процедура получения необходимого набора сертификатов заключается в следующем:

1) формируются ключи и корневой сертификат удостоверяющего центра;
2) для каждого сервера или клиента генерируется ключевая пара;
3) на основе полученной ключевой пары формируется заявка (запрос) на сертификат;
4) с помощью удостоверяющего центра по заявке выписывается сертификат;
5) полученная ключевая пара и сертификат сохраняются в соответствующие места системы.

Генерация ключевых пар и работа с сертификатами осуществляется в соответствии с инструкциями производителя соответствующего токена.

18.4. Настройка доменного входа (ЕПП)

При использовании ЕПП для аутентификации пользователей применяется доверенная аутентификация Kerberos (см. 8.1.4). По умолчанию аутентификации производится по паролю пользователя. В тоже время существует стандарт использования защитного преобразования с открытым ключом в качестве фактора аутентификации в протоколе аутентификации Kerberos PKINIT (Public Key Cryptography for Initial Authentication in Kerberos). Это позволяет применять сертификаты и, следовательно, устройства PKCS-11 для аутентификации по Kerberos.

Для используемого варианта Kerberos (MIT Kerberos V5) возможности PKINIT реализуются пакетом расширения `krb5-pkinit`. При этом для проведения аутентификации используется подгружаемый модуль аутентификации `libpam-krb5`.

ВНИМАНИЕ! Перед настройкой доменного входа с помощью сертификатов с устройств PKCS-11 должны быть выполнены следующие условия:

1) установлена и соответствующим образом настроена служба домена;
2) настроен домен ЕПП и созданы необходимые пользователи;
3) на компьютеры домена установлен пакет расширения `krb5-pkinit`;
4) получен или создан корневой сертификат CA.

19. СООБЩЕНИЯ АДМИНИСТРАТОРУ И ВЫЯВЛЕНИЕ ОШИБОК

19.1. Диагностические сообщения

При возникновении проблем в процессе функционирования ОС появляются диагностические сообщения трех типов: информационные, предупреждающие и сообщения об ошибках (примеры приведены в таблицах 71– 73, соответственно). Администратор должен проанализировать диагностические сообщения и принять меры по устранению появившихся проблем.

Таблица 71 – Информационные сообщения

Сообщение ОС	Описание	Файл
`Setting hostname to <>`	Установка имени хоста <>	`hostname`
`Setting domainname to <>`	Установка имени домена как <>	`hostname`
`Statistics dump initiated`	Вывод статистики запущен	`named`
`Query logging is now on`	Регистрация очередей включена	`named`
`Query logging is now off`	Регистрация очередей выключена	`named`
`Unknown host`	Неизвестный хост	`dnsquery`
`Non reloadable zone`	Не перезагружаемая зона	`named`
`Reconfig initiated`	Переконфигурирование запущено	`named`
`Zone not found`	Зона не найдена	`named`

Таблица 72 – Предупреждающие сообщения

Сообщение ОС	Описание	Действия по устранению проблемы	Файл
`<>: You can't change the DNS domain name with this command`	Неверное использование команды	Использовать соответствующую команду	`hostname`
`Could not find any active network interfaces`	Активные сетевые интерфейсы не найдены	Активировать сетевой интерфейс	`sendmail`
`You must be root to change the host name`	Недостаточно прав для изменения имени хоста	Обратиться к администратору	`dnsdomainname`

Таблица 73 – Сообщения об ошибках

Сообщение ОС	Описание	Действия по устранению проблемы	Файл
`Unknown server error`	Неизвестная ошибка сервера	Изменить права доступа	`dnsquery`
`Resolver internal error`	Внутренняя ошибка резольвера	Изменить права доступа	`dnsquery`

Окончание таблицы 73

Сообщение ОС	Описание	Действия по устранению проблемы	Файл
`Superblock last mount time (значение времени) is in the future`	Неверная установка времени	См. 19.3	См. 19.3

19.2. Выявление ошибок

В состав ОС входит инструмент `sosreport`, предназначенный для сбора информации о конфигурации системы и диагностических данных о работе ОС и ее компонентов. Инструмент включает модули для сбора информации о работе отдельных подсистем и программ из состава ОС.

На основе собранных данных создается диагностический архив с отчетом, который может храниться локально, централизованно или отправляться техническим специалистам. Дополнительно возможно создавать XML/HTML-отчеты.

Перечень основных параметров, используемых с инструментом `sosreport`, приведен в таблице 74.

Таблица 74

Параметр	Описание
`-l`	Вывести список доступных модулей и их параметры. Модули, которые не могут использоваться с текущей конфигурацией, выводятся отдельно
`-n <имя_модуля>`	Отключить указанный модуль. Отключение нескольких модулей выполняется повторением параметра или заданием списка модулей через запятую
`-e <имя_модуля>`	Включить указанный модуль. Включение нескольких модулей выполняется повторением параметра или заданием списка модулей через запятую
`-o <имя_модуля>`	Включить только указанный модуль (неуказанные модули будут автоматически отключены). Включение нескольких модулей выполняется повторением параметра или заданием списка модулей через запятую
`-k <имя_модуля>.<параметр_модуля> [=<значение>]`	Задать параметры модуля. Включает указанный параметр модуля, может также задавать значение параметра модуля
`-a`	Установить для всех логических параметров всех включенных модулей значение `True`
`-v`	Увеличить детализацию протоколирования. Может выполняться несколько раз для добавления дополнительных сообщений

Продолжение таблицы 74

Параметр	Описание
`--no-postproc`	Отключить постобработку собранных данных для всех модулей. В архиве с собранными данными не будет замаскирована/очищена конфиденциальная информация. Такие данные, как пароли, SSH-ключи, сертификаты будут сохранены в виде простого текста. Чтобы отключить постобработку для определенного модуля, использовать с параметром `-k` параметр `postproc` модуля, например `-k logs.postproc=off`
`-s <корневая_файловая_система>`	Указать другую корневую файловую систему. Возможно использовать для создания отчета работы контейнера или образа
`-c {auto/always/never}`	Установить режим использования `chroot`. Когда используется `-s`, команды по умолчанию выполняются с заданной файловой системой (если только они не отключены определенным модулем). Параметр `-c` переопределяет использование заданной корневой файловой системы: - значение `always` — всегда использовать корневую файловую систему, заданную параметром `-s`; - `never` — никогда не использовать корневую файловую систему, заданную параметром `-s` (команды всегда будут выполняться в пространстве хоста)
`--tmp-dir <путь>`	Задать временный каталог для копирования данных и архива отчета
`--list-profiles`	Вывести список доступных профилей и включенных в них модулей
`-p <имя_профиля>`	Выполнить модули, включенные в указанный профиль. Несколько профилей могут быть заданы через запятую, при этом будут выполнены модули всех указанных профилей
`--log-size`	Установить ограничение на размер (в МиБ) набора журналов. Ограничение применяется отдельно для каждого набора журналов, собранных любым модулем
`--all-logs`	Собрать все возможные журналы данных, включая из незаданных областей, игнорируя любые ограничения размера. При этом размер отчетов может быть увеличен
`-z <метод_сжатия>`	Задать метод сжатия отчета
`--encrypt-pass <пароль>`	Аналогично `--encrypt-key`, но защита архива выполняется установкой пароля
`--batch`	Создать архив отчета без интерактивных запросов пользователю

Окончание таблицы 74

Параметр	Описание
`--case-id <идентификатор_архива>`	Задать идентификатор архива. Может содержать цифры, латинские буквы, запятые и точки

Более подробное описание инструмента доступно в `man sosreport`.

Для использования инструмента `sosreport` в графическом режиме доступна утилита `fly-sosreport`. Описание утилиты приведено в электронной справке.

19.3. Циклическая перезагрузка компьютера по причине неверной установки времени

При возникновении сбоя, связанного с циклической перезагрузкой компьютера, необходимо во время загрузки ОС при появлении на экране заставки с мерцающей надписью «Astra Linux Special Edition» нажать клавишу **<Esc>**. Если среди отобразившихся сообщений есть сообщение вида:

```
/dev/sda1: Superblock last mount time (Wed Feb 15 12:41:05 2017,
            now = Mon Feb 15 12:45:37 2016) is in the future.
```

то сбой связан с неверной установкой времени.

Для устранения сбоя необходимо войти в меню настройки загрузчика и проверить выставленное системное время. Если системное время отстает от реального, то, возможно, это связано с отказом элемента питания системной платы. В этом случае необходимо заменить элемент питания на системной плате в соответствии с указаниями инструкции к техническому средству и установить корректное системное время.

Если системное время в меню настроек загрузчика установлено верно, но циклическая перезагрузка продолжается, то сбой может быть связан с неверным переводом времени на будущую дату и обратно. Данный сбой происходит если установить системное время на будущую дату, затем загрузить ОС и установить верное текущее время или сразу установить системное время на прошедшую дату. Для устранения данного сбоя необходимо:

1) в меню настроек загрузчика установить системное время на будущую дату, при этом дата должна быть позже даты, указанной в сообщении об ошибке при загрузке;

2) загрузить ОС;

3) создать файл `/etc/ef2fsck.conf` с содержимым:

```
[options]
broken_system_clock = true
```

4) создать файл `/etc/initramfs-tools/hooks/e2fsck-conf.sh` с содержимым:

```
#!/bin/sh
```

```
PREREQ=""
prereqs()
{
   echo "$PREREQ"
}

case $1 in
prereqs)
   prereqs
   exit 0
   ;;
esac

. /usr/share/initramfs-tools/hook-functions
CONFFILE=/etc/e2fsck.conf
CONFDIR=`dirname "$CONFFILE"`
if [ -f "$CONFFILE" ]
then
   mkdir -p ${DESTDIR}${CONFDIR}
   cp $CONFFILE ${DESTDIR}${CONFDIR}
fi
```

5) в терминале выполнить команду:

```
sudo update-initramfs -u
```

6) перезагрузить ОС и установить текущее время в качестве системного.

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

БД — база данных
ЕПП — единое пространство пользователей
КСЗ — комплекс средств защиты
ЛВС — локальная вычислительная сеть
МКЦ — мандатный контроль целостности
НСД — несанкционированный доступ
ОС — операционная система специального назначения «Astra Linux Special Edition»
ПО — программное обеспечение
СЗИ — средства защиты информации
СЗФС — сетевая защищенная файловая система
СУБД — система управления базами данных
УЦ — удостоверяющий центр
ФС — файловая система

AD — Active Directory (служба каталогов)
ACL — Access Control List (список контроля доступа)
ALD — Astra Linux Directory (единое пространство пользователей)
API — Application Programming Interface (программный интерфейс приложения)
ARP — Address Resolution Protocol (протокол разрешения адресов)
BIND — Berkeley Internet Name Domain (пакет программного обеспечения для поддержки DNS, разработанный в Калифорнийском университете, г. Беркли)
BOOTP — Bootstrap Protocol (простой протокол динамической конфигурации хоста)
BSD — Berkeley Software Distribution (программное изделие Калифорнийского университета)
CA — Certification Authority (удостоверяющий центр)
CephFS — Ceph File System (файловая система Ceph)
CIFS — Common Internet File System (общий протокол доступа к файлам Интернет)
DC — Domain Controller (контроллер домена)
DHCP — Dynamic Host Configuration Protocol (протокол динамической конфигурации хоста)
DIB — Directory Information Base (информационная база каталога)
DIT — Directory Information Tree (информационное дерево каталога)
DN — Distinguished Name (уникальное имя)
DNS — Domain Name System (система доменных имен)
FTP — File Transfer Protocol (протокол передачи файлов)

FQDN — Fully Qualified Domain Name (полностью определенное имя домена)
GID — Group Identifier (идентификатор группы)
HTTP — HyperText Transfer Protocol (протокол передачи гипертекста)
IDE — Integrated Drive Electronics (встроенный интерфейс накопителей)
IMAP — Internet Message Access Protocol (протокол доступа к сообщениям в сети Интернет)
IP — Internet Protocol (межсетевой протокол)
IPA — Identity, Policy, and Audit (система по управлению идентификацией пользователей, задания политик доступа и аудита для сетей на базе Linux и Unix)
IPC — InterProcess Communication (межпроцессное взаимодействие)
KDC — Key Distribution Center (центр распределения ключей)
KRA — Key Recovery Authority (служба восстановления ключей)
LDAP — Lightweight Directory Access Protocol (легковесный протокол доступа к службам каталогов)
LPR — Line Printer Remote (удаленный линейный принтер)
LVM — Logical Volume Manager (менеджер логических томов)
MAC — Mandatory Access Control (мандатное управление доступом)
MDA — Mail Delivery Agent (агент доставки электронной почты)
MDS — Metadata Server (сервер метаданных)
MIT — Massachusetts Institute of Technology (Массачусетский Технологический Институт)
MON — Monitor (монитор)
MTA — Mail Transfer Agent (агент пересылки сообщений)
MTU — Maximum Transfer Unit (максимальная единица передачи)
MUA — Mail User Agent (клиент электронной почты)
NAT — Network Address Translation (преобразование сетевых адресов)
NFS — Network File System (сетевая файловая система)
NIS — Network Information Service (сетевая информационная служба)
NSS — Name Service Switch (диспетчер службы имен)
NTP — Network Time Protocol (протокол сетевого времени)
OSD — Object Storage Device (устройство хранения объектов)
PAM — Pluggable Authentication Modules (подключаемые модули аутентификации)
PKI — Public Key Infrastructure (инфраструктура открытых ключей)
PTP — Precision Time Protocol (протокол точного времени)
POP3 — Post Office Protocol Version 3 (почтовый протокол, версия 3)
RADOS — Reliable Autonomic Distributed Object Store (безотказное автономное распределенное хранилище объектов)

RBD — RADOS block device (блочное устройство)

RFC — Request For Comments (общее название технических стандартов сети Интернет)

RPC — Remote Procedure Call (удаленный вызов процедур)

RTS — Real Time Clock (время, установленное в аппаратных часах компьютера)

SASL — Simple Authentication and Security Layer (простая аутентификация и слой безопасности)

SATA — Serial ATA (последовательный интерфейс обмена данными с накопителями информации, является развитием интерфейса IDE)

SCSI — Small Computer System Interface (системный интерфейс малых компьютеров)

SMB — Server Message Block (блок сообщений сервера)

SPICE — Simple Protocol for Independent Computing Environments (простой протокол для независимой вычислительной среды)

SQL — Structured Query Language (язык структурированных запросов)

SSH — Secure Shell Protocol (протокол защищенной передачи информации)

SSL — Secure Sockets Layer (протокол защищенных сокетов)

SSSD — System Security Services Daemon (системный демон, управляющий доступом к удаленным каталогам и механизмам аутентификации)

TCP — Transmission Control Protocol (протокол управления передачей данных)

TLS — Transport Layer Security (протокол защиты транспортного уровня)

TTL — Time To Live (время жизни IP-пакета)

UDP — User Datagram Protocol (протокол пользовательских дейтаграмм)

UID — User Identifier (идентификатор пользователя)

URI — Uniform Resource Identifier (унифицированный идентификатор ресурса)

UTC — Universal Time Coordinated (универсальное скоординированное время)

VFS — Virtual File System (виртуальная файловая система)

VIP — Virtual IP-adress (виртуальный IP-адрес)

VNC — Virtual Network Computing (система удаленного доступа к рабочему столу компьютера)

VPN — Virtual Private Network (виртуальная частная сеть)

VRRP — Virtual Redundancy Routing Protocol (сетевой протокол виртуального резервирования маршрутизаторов, предназначенный для увеличения доступности)

XCA — X window system Certification Authority (графический инструмент создания и управления удостоверяющим центром)

Лист регистрации изменений									
Изм.	Номера листов (страниц)				Всего листов (страниц) в документе	Номер документа	Входящий номер сопроводительного документа и дата	Подпись	Дата
	измененных	замененных	новых	аннулированных					
1		Все				РУСБ.13-21			30.07.21