50 1190 0101

Утвержден

РУСБ.10153-02-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

РУСБ.10153-02 95 01-1

Листов 123

2024

Литера $О_1$

АННОТАЦИЯ

Настоящий документ является руководством администратора программного изделия РУСБ.10153-02 «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.10153-02 97 01-1 «Операционная система специального назначения «Astra Linux Special Edition». Руководство по КСЗ. Часть 1».

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

Руководство администратора состоит из двух частей:

- РУСБ.10153-02 95 01-1 «Операционная система специального назначения «Astra Linux Special Edition». Руководство администратора. Часть 1»;
- РУСБ.10153-02 95 01-2 «Операционная система специального назначения «Astra Linux Special Edition». Руководство администратора. Часть 2. Установка».

В настоящем документе приведено назначение и настройка ОС. Рассмотрены системные компоненты, службы и команды, базовые сетевые службы, управление программными пакетами, резервное копирование и восстановление данных, средства контроля целостности и централизованного протоколирования. Приведен список сообщений для администратора.

Требования к обеспечению безопасности среды функционирования, а также настройка параметров, необходимых для безопасной эксплуатации ОС, приведены в документе РУСБ.10153-02 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 `wiki.astralinux.ru`.

СОДЕРЖАНИЕ

РУСБ.10153-02 95 01-2 «Операционная система специального назначения «Astra Linux Special Edition». Руководство администратора. Часть 2. Установка и миграция»

1. АДМИНИСТРИРОВАНИЕ ОС

Административное управление в ОС отделено от общего доступа пользователей.

Большинство операций по настройке и администрированию ОС требуют прав 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 ОС создается пользователь, входящий в группу `astra-admin`. Пользователю, входящему в 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.

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, которые сохраняются до завершения его работы. Пользователю для получения прав суперпользователя не требуется завершать свою сессию и вновь входить в систему.

С помощью команды `su`, вводимой с параметром `-c`, пользователь может выполнять отдельные команды от имени учетной записи `root` без запуска командного интерпретатора `shell`.

При этом пользователь получает права и привилегии суперпользователя на ограниченное время, а именно, на время исполнения заданной команды. Например, при необходимости поменять атрибуты файла ввести команду от имени учетной записи root:

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

После ввода пароля учетной записи root команда chmod получит права и привилегии суперпользователя на выполнение заданного запроса, но при этом права и привилегии пользователя на выполнение других команд не изменятся.

Кроме выполнения команд от имени учетной записи root, команда su позволяет выполнять команды от имени любого другого пользователя, при этом для выполнения команды необходимо знать пароль этого пользователя. Если вход в систему выполнен от имени root, то при использовании su для выполнения команды от имени другого пользователя знание пароля данного пользователя не требуется — все команды от имени любого пользователя исполняются без запроса пароля.

При предоставлении прав на использование команды su следует учитывать, что для нее отсутствует механизм ограничения списка команд, разрешенных конкретному пользователю выполнять от имени учетной записи root. Таким образом, если у пользователя есть права на выполнение команды su, то он может выполнить от имени учетной записи root любые команды. Поэтому использование команды su должно быть разрешено только доверенным пользователям. Также рекомендуется при вводе команды использовать полное путевое имя /bin/su (вместо su).

Описание команды приведено в man su.

1.1.2. sudo

Команда sudo используется пользователем для запуска команд от имени учетной записи root.

В качестве параметров команда sudo принимает командную строку, которую следует выполнить с правами суперпользователя. При выполнении команды sudo просматривается конфигурационный файл /etc/sudoers, в котором приведен список пользователей, имеющих полномочия на запуск команды sudo, а также перечень команд, которые каждый из пользователей имеет право выполнять от имени учетной записи root. Если данному пользователю разрешено выполнять указанную им команду, то при выполнении команды sudo у пользователя запрашивается его пароль. Таким образом, для каждого пользователя установлен набор команд, которые он может выполнять от имени учетной записи root без необходимости вводить пароль учетной записи root.

При использовании `sudo` подсистемой регистрации событий регистрируется следующая информация: выполненные команды, вызвавшие их пользователи, из какого каталога вызывались команды, время вызова команд.

Для изменения файла `/etc/sudoers` используется команда `visudo`, запущенная от имени администратора.

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.10153-02 97 01-1.

1.2.1. Механизм привилегий

Механизм привилегий ОС предназначен для передачи отдельным пользователям прав выполнения определенных 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.

1.2.2. Механизм повышения полномочий

Механизм повышения полномочий позволяет повысить полномочия пользователя на время выполнения определенной программы.

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.

2. СИСТЕМНЫЕ КОМПОНЕНТЫ

2.1. Управление устройствами

2.1.1. Типы устройств

В ОС существует два типа устройств:

1) блочные устройства с произвольным доступом — данные, записанные в такие устройства, могут быть прочитаны (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).

Файлы устройств располагаются в каталоге `/dev`, для вывода списка файлов выполнить команду `ls`. При выполнении команды с параметром `-l` на экран монитора будет выведен список файлов с указанием в первой колонке типа файла и прав доступа к нему. Первый символ в первой колонке указывает на тип файла:

- `c` — символьное устройство;
- `b` — блочное устройство;
- `d` — каталог;
- `l` — символическая ссылка;
- «`-`» (дефис) — обычный файл.

Пример

Просмотр информации о файле, соответствующем звуковому устройству

```
ls -l /dev/dsp
```

Вывод команды:

```
crw-rw---- 1 root audio 14, 3 июл 1 13:05 /dev/dsp
```

Описание команды `ls` приведено в `man ls`.

Наличие файла устройства не означает, что данное устройство установлено 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 параметров и создания файлов для новых устройств.

2.1.2. Жесткие диски

При администрировании дисков могут возникнуть задачи по разделению жесткого диска на разделы, созданию и монтированию ФС, форматированию диска и др.

Разделение жесткого диска может использоваться для хранения разных операционных систем на одном жестком диске, для хранения пользовательских и системных файлов в разных дисковых разделах. Разделение жесткого диска упрощает резервное копирование и восстановление, а также повышает защищенность системных файлов от повреждений.

Для использования диска или раздела необходимо создать 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. Монтируемые вручную ФС должны быть размонтированы также 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;
- `/dev/sda`, `/dev/sdb` и т. д. — для дисков, использующих SCSI или SATA-интерфейс.

2.1.3. Разделы жесткого диска

Весь жесткий диск может быть разделен на несколько дисковых разделов, при этом каждый раздел в системе представлен как отдельный диск. Разделение используется, например, при работе с двумя операционными системами на одном жестком диске. При этом каждая операционная система использует для работы отдельный дисковый раздел и не взаимодействует с другими. Таким образом, две различные системы могут быть установлены на одном жестком диске.

2.1.3.1. Разбиение жесткого диска

Главная загрузочная запись MBR (Master Boot Record) диска содержит место для четырех основных (первичных) разделов, пронумерованных от 1 до 4.

Если необходимо добавить еще разделы на диск, то следует преобразовать основной раздел в расширенный (extended). Далее расширенный раздел разделяется на один или несколько логических разделов с номерами от 5 до 15. Логические разделы функционируют так же, как и основные, различие состоит в схеме их создания.

При установке ОС разбиение жесткого диска (дисков) осуществляется средствами программы-установщика. При работе с ОС для разбиения жесткого диска на разделы используется инструмент командной строки `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 раздела, затем удалить раздел, создать новый раздел и восстановить сохраненную информацию в новом разделе.

Описание инструмента `fdisk` приведено в `man fdisk`.

2.1.3.2. Файлы устройств и разделы

Каждому первичному и расширенному разделу соответствует отдельный файл устройства. Существует соглашение для имен подобных файлов, которое заключается в добавлении номера раздела к имени соответствующего файла устройства. Разделы с 1 по 4 являются первичными либо один из этих разделов является расширенным. Разделы с 5 по 15 являются логическими, на которые разбивается расширенный раздел. Например, `/dev/hda1` соответствует первому первичному разделу первого IDE-диска, а `/dev/sdb7` — третьему логическому разделу второго диска с интерфейсом SCSI или SATA.

2.1.4. Форматирование

Форматирование — это процесс записи специальных отметок на магнитную поверхность, которые используются для разделения дорожек и секторов. Новый диск не может использоваться без предварительного форматирования. Для IDE- и некоторых SCSI-дисков форматирование выполняется при их изготовлении и обычно не требуется повторение этой процедуры.

2.1.5. 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 для создания и управления массивами RAID.

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

Том LVM, с точки зрения ядра системы, использует унифицированные механизмы VFS и не нуждается в специальных конфигурациях ядра. В ОС обеспечивается полнофункциональное управление томами LVM, которое осуществляется стеком команд управления.

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 перемещать с одного физического устройства на другое, а их размер изменять. Физические устройства можно добавлять и удалять. Томам, управляемым посредством LVM, можно назначать любые текстовые названия, например `database` или `home`, а не служебные `sda` или `hda`.

2.2. Управление ФС

2.2.1. Общие сведения

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, 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, каталоги, системные и пользовательские файлы) и располагаются в ФС. Структура ФС имеет вид «перевернутого дерева», вершину которого называют корневым каталогом, в системе обозначается символом «/».

В зависимости от параметров, заданных в процессе установки ОС, каталоги могут относиться к различным ФС.

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

- /bin (/usr/bin) — содержит исполняемые файлы, необходимые для работы системы. Многие команды ОС являются программами из этого каталога;
- /boot — содержит необходимую информацию для загрузки системы: ядро (ядра), образ initrd, файлы загрузчика;
- /dev — содержит файлы устройств (device files). С их помощью осуществляется доступ к физическим устройствам, установленным в системе;
- /root — рабочий (домашний) каталог суперпользователя;
- /tmp — используется для хранения временных файлов, создаваемых программами в процессе своей работы. При работе с программами, создающими много больших временных файлов, рекомендуется иметь отдельную ФС;
- /etc — содержит конфигурационные файлы ОС, в т.ч. файл паролей passwd и список ФС fstab, монтируемых при начальной загрузке. В этом же каталоге хранятся сценарии загрузки (startup scripts), список узлов (hosts) с их IP-адресами и другие данные о конфигурации;
- /lib (/usr/lib) — содержит разделяемые библиотеки, используемые 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 (/usr/sbin) — содержит исполняемые файлы, используется для системного администрирования и требующие для запуска права суперпользователя);
- /usr — содержит 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 — содержит изменяемые файлы, например log-файлы и др.;
- /home — содержит рабочие (домашние) каталоги пользователей. Рекомендуется создавать в качестве отдельной ФС, чтобы обеспечить пользователям достаточное пространство для размещения своих файлов. Если пользователей в системе много, возможно разделить этот каталог на несколько ФС. Например, можно создать подкаталоги /home/staff и /home/admin соответственно для персонала и администраторов, установить каждый подкаталог как отдельную ФС и уже в них создавать рабочие каталоги пользователей.

В рабочих каталогах пользователей также содержатся некоторые конфигурационные файлы, которые являются скрытыми и изменяются редко. Файл становится скрытым, если поставить точку в начале имени файла. При выводе списка файлов командой ls для отображения в том числе скрытых файлов использовать параметр -a:

```
ls -a
```

Для обеспечения совместной работы пользователей в ОС создаются автоматически при установке совместно используемые каталоги, доступ к которым разрешен всем пользователям:

- `/tmp` — каталог временных файлов. Содержимое каталога не сохраняется после перезагрузки ОС;
- `/var/tmp` — каталог временных файлов. Содержимое каталога сохраняется после перезагрузки ОС;
- `/dev/shm` — каталог разделяемой памяти, используется для обмена временными рабочими данными через разделяемую оперативную память. Содержимое каталога не сохраняется после перезагрузки ОС;
- `/run/mount` — каталог временного монтирования пользовательских устройств, используется для автоматического монтирования с помощью сценариев подключаемых пользовательских устройств;
- `/var/cache` — каталог для кеширования данных.

Также при установке дополнительного ПО могут создаваться собственные совместно используемые каталоги данного ПО.

Совместно используемые каталоги предназначены для создания в них файловых объектов, доступных всем пользователям. Применяемый в ОС механизм дискреционного управления доступом позволяет всем пользователям выполнять создание файловых объектов в совместно используемые каталогах, а также поиск принадлежащих другим пользователям файловых объектов в совместно используемых каталогах. Чтение и изменение не принадлежащих пользователю файловых объектов ограничивается дискреционными правами доступа, заданным для данного объекта. Дополнительно применяется специальное ограничение дискреционного доступа sticky-бит, запрещающее удалять и переименовывать не принадлежащие пользователю файловые объекты.

2.2.2. Создание

ФС создается при помощи команды `mkfs`. Команда запускает требуемую программу в зависимости от типа создаваемой ФС. Тип ФС задается параметром `-t`.

Пример

```
mkfs -t ext2 /dev/hdb1
```

Описание команды приведено в `man mkfs`.

2.2.3. Монтирование

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

Перед монтированием ФС к дереву каталогов ОС необходимо убедиться, что существует каталог (точка монтирования), в который будет осуществляться монтирование ФС, иначе монтирование завершится неудачно.

После успешного монтирования ФС в каталог в нем появятся все файлы и подкаталоги ФС. В противном случае каталог будет пустым.

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

Для получения информации об имеющихся в ОС файловых системах используется инструмент командной строки `df`. Описание инструмента приведено в `man df`.

2.2.3.1. mount

В ОС для монтирования ФС используется инструмент командной строки `mount`. По умолчанию в целях обеспечения безопасности информации использовать инструмент `mount` может только администратор.

Синтаксис:

```
mount [параметр[параметр]] [<устройство>] [<точка_монтирования>]
```

где `<устройство>` — устройство, которое необходимо примонтировать;
`<точка_монтирования>` — имя каталога, в который требуется примонтировать устройство.

Параметры, дополнительно используемые с инструментом `mount`, приведенные в таблице 1.

Таблица 1

Параметр	Описание
`-f`	Имитировать монтирование ФС. Выполняются все действия, кроме системного вызова для монтирования ФС
`-v`	Вывести подробный отчет о действиях, выполняемых командой
`-w`	Примонтировать ФС с доступом для чтения и записи
`-r`	Примонтировать ФС с доступом только для чтения
`-n`	Выполнить монтирование без записи в файл `/etc/mtab`
`-t <тип_ФС>`	Задать тип монтируемой ФС
`-a`	Подключить все ФС, перечисленные в `/etc/fstab`
`-o <параметр>`	Задать параметры монтирования ФС, параметры в списке разделяются запятыми. Список возможных параметров приведен в `man mount`

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

Примеры:

1. Монтирование раздела жесткого диска /dev/hdb3 в каталог /mnt:

```
mount /dev/hdb3 /mnt
```

2. Монтирование всех ФС типа NFS, перечисленных в файле /etc/fstab:

```
mount -vat nfs
```

Если правильно примонтировать ФС не удается, то для получения отчета о результатах выполнения команды mount выполнить команду:

```
mount -vf <устройство> <точка_монтирования>
```

В данном случае команда выполняет все действия, кроме монтирования, и выводится подробный отчет о каждом шаге выполнения команды.

Описание команды mount приведено в man mount.

2.2.3.2. fstab

В конфигурационном файле /etc/fstab указываются ФС для монтирования и перечисляются параметры их монтирования.

В файле /etc/fstab каждой ФС соответствует запись в одной строке. Поля в строках разделяются пробелами или символами табуляции. В таблице 2 приведено описание полей файла /etc/fstab.

Таблица 2

Поле	Описание
<file system> (файловая система)	Подключаемое блочное устройство или удаленная ФС
<mount point> (точка монтирования)	Каталог монтирования ФС. Чтобы сделать систему невидимой в дереве каталогов (например, для файлов подкачки), используется слово none
<type> (тип)	Указывает тип монтируемой ФС
<options> (параметры монтирования)	Список разделенных запятыми параметров для монтируемой ФС. Должен содержать, по крайней мере, тип монтирования. Более подробную информацию см. в руководстве man команды mount
<dump> (периодичность резервного копирования)	Указывает, как часто следует выполнять резервное копирование с помощью команды dump. Если в поле стоит значение 0, то резервное копирование ФС не выполняется

Окончание таблицы 2

Поле	Описание
<pass> (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, без использования 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 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 #.

В файле fstab параметр defaults поля <options> указывает, что при монтировании ФС будет применен набор параметров по умолчанию, а именно — ФС будет примонтирована 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 монтироваться с помощью команды mount -a; биты UID и GID файлов в ФС интерпретируются; обычным пользователям не разрешено подключать данную ФС.

Раздел подкачки (в примере /dev/sda5) 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 монтирования в

файле fstab указывается none. Кроме того, разделы подкачки монтируются с параметром sw.

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

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

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

2.2.4. Размонтирование

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

Инструмент umount имеет следующий синтаксис:

```
umount <устройство>
umount <точка_монтирования>
umount -a
umount -t <тип_ФС>
```

где <устройство> — устройство, которое необходимо размонтировать;
<точка_монтирования> — имя каталога, от которого необходимо отмонтировать;
-a — размонтировать все ФС;
-t — размонтировать только ФС указанного типа <тип_ФС>.

Инструмент umount не размонтирует ФС, если она используется в текущий момент. Например, если 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:

```
umount -f /cdrom
```

Для размонтирования и извлечения из устройств сменных носителей информации используется инструмент командной строки `eject`.

Инструмент командной строки `fuser` отображает сведения о процессах, использующих ФС:

```
fuser -v  <точка_монтирования>
```

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

```
fuser -km <точка_монтирования>
```

Описание инструментов `umount`, `eject` и `fuser` приведено, соответственно, в `man umount`, `man eject` и `man fuser`.

2.3. Управление пользователями

2.3.1. 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.

2.3.1.1. Добавление пользователя

Для добавления пользователя применяется инструмент командной строки `adduser` с указанием в качестве параметра имени добавляемого пользователя:

```
adduser <имя_пользователя>
```

Команда `adduser` добавляет пользователя, создает рабочий каталог пользователя, 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 необходимы каждому пользователю. Обычно это персональ-

ные конфигурационные файлы для настройки оболочки, например .profile, .bashrc и .bash_logout.

При добавлении пользователя в систему в файле /etc/passwd добавляется запись вида:

```
login_name:encrypted_password:user_ID:group_ID:user_information:
     login_directory:login_shell
```

Описание полей записи приведено в таблице 3.

Таблица 3

Поле	Описание
login_name	Регистрационное имя учетной записи пользователя (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, то он может в процессе работы менять группу (если это разрешено администратором)
user_information	Информация о пользователе, например его фамилия, 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)

Описание файла /etc/passwd приведено в man 5 passwd.

Команда adduser представляет собой файл сценария bash, находящийся в каталоге /usr/sbin.

Для изменения информации о пользователе используется инструмент командой строки chfn.

Описание adduser и chfn приведено, соответственно, в man adduser и man chfn.

2.3.1.2. Установка пароля пользователя

Для установки пароля пользователя предназначена команда passwd. Задавать пароль необходимо для каждого пользователя. После входа в систему пользователь может изменить свой пароль. Для установки пароля пользователя выполнить следующее:

1) ввести команду и имя пользователя, например:

```
passwd ivanov
```

и нажать клавишу **<Enter>**;

2) после появления приглашения:

```
Новый пароль:
```

ввести пароль и нажать клавишу **<Enter>**;

3) ввести повторно пароль после появления соответствующего сообщения и нажать клавишу **<Enter>**.

Пароль будет преобразован и внесен в файл `/etc/shadow`.

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

Описание команды приведено в `man passwd`.

Пример

Запись в файле `/etc/passwd`

```
ivanov:x:123:121:Petr Ivanov:/home/ivanov:/bin/bash
```

Второе поле записи содержит ссылку на пароль в преобразованном виде.

Примечание. Пароль пользователя не хранится в явном виде. Если пользователь забыл свой пароль, то администратор системы не может его восстановить. Для восстановления доступа пользователя в систему администратор может задать новый пароль для пользователя с помощью команды `passwd`.

2.3.1.3. Удаление пользователя

В ОС доступно несколько вариантов удалить пользователя:

- лишить пользователя возможности входа в систему;
- удалить учетную запись пользователя;
- удалить учетную запись пользователя и все его файлы и каталоги.

Лишение пользователя возможности входа в систему может быть использовано в случае его длительного перерыва в работе. На время отсутствия пользователя можно заблокировать его учетную запись с помощью команды:

```
usermod -L <имя_пользователя>
```

После выполнения команды вход в систему от имени указанного пользователя будет недоступен, при этом все пользовательские файлы и каталоги сохраняются.

Для разблокировки учетной записи необходимо выполнить команду:

```
usermod -U <имя_пользователя>
```

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

```
usermod -l <новое_имя_пользователя>  <имя_пользователя>
```

Примечание. Имена домашнего каталога и почтового ящика при 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 <имя_пользователя>
```

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

Также удаление пользователя, его домашнего каталога и файлов могут быть выполнены вручную путем последовательного выполнения следующих команд:

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

```
find / -user <имя_пользователя> -exec rm -r {} \;
```

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

3) 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`.

2.3.1.4. Неудачный вход в систему

Инструмент командной строки `faillog` используется для управления счетчиком неудачных попыток и их ограничения. Также инструмент отображает журнал неудачных попыток входа в систему (файл `/var/log/faillog`).

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 счетчика неудачных попыток входа необходимо использовать параметр `-r`.

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

2.3.2. Работа с группами

Каждый пользователь является членом группы. Разным группам можно назначить разные возможности и привилегии.

Пользователь может состоять в нескольких группах и переходить из одной в другую в процессе работы.

Информация о группах содержится в файле `/etc/group`. Описание файла `/etc/group` приведено в `man 5 group`.

2.3.2.1. Добавление

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

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

где `Admin` — имя группы;
`x` — пароль в преобразованном виде. Если поле пустое, то пароль не требуется;
`21` — уникальный идентификатор группы;
`user1, user2, user3` — участники группы.

Добавление группы производится с помощью команды:

```
addgroup <имя_группы>
```

Также новую группу можно создать путем редактирования файла `/etc/group`, добавив в нем строку с необходимой информацией 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 инструмента `addgroup` и файла `/etc/group` приведено, соответственно, в `man addgroup` и `man 5 group`.

2.3.2.2. Удаление

Удаление группы производится с помощью команды:

```
delgroup <имя_группы>
```

Также удалить группу можно путем редактирования файла `/etc/group`, удалив записи о группе.

Описание команды `delgroup` приведено в `man delgroup`.

2.3.3. 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.

2.4. Перезагрузка и выключение

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 одним из способов:

1) использовать команду `shutdown` с параметром `-r` в соответствии с 2.4.1;
2) использовать команду `reboot` в соответствии с 2.4.2;
3) использовать команду `init 6`.

Выключение компьютера предполагает корректное завершение работы системы (останов), позволяющее избежать потерь информации и сбоев ФС.

Выключение компьютера можно выполнить несколькими способами:

1) использовать команду `shutdown` (см. 2.4.1);
2) использовать команду `halt` (см. 2.4.2);
3) использовать команду `init 0`.

Работая с ОС, не рекомендуется выключать питание компьютера без предварительного завершения работы с использованием соответствующих инструментов ОС, т. к. ОС хранит информацию ФС в оперативной памяти и при отключении питания информация может быть потеряна, а ФС повреждена.

Выключение питания также может повредить жесткий диск, если установленный в системе жесткий диск перед отключением питания требует установки в соответствующее положение защитный переключатель либо выполнения парковки головок.

2.4.1. shutdown

Команда `shutdown` позволяет безопасно и корректно инициировать завершение работы системы, выключение, перезагрузку или возврат в однопользовательский режим.

В качестве параметра команды `shutdown` возможно задать время ожидания перед завершением работы системы. Во время ожидания команда посылает зарегистрированным пользователям через постепенно укорачивающиеся промежутки времени предупреждения о завершении работы системы. По умолчанию сообщения содержат информацию о завершении работы и времени, оставшемся до завершения работы. При желании администратор может добавить собственное сообщение, например с информацией о причине останова и о времени, в течение которого вход в систему будет невозможен.

Параметры команды `shutdown` позволяют задать определенное действие для компьютера: остановиться, перейти в однопользовательский режим или перезагрузиться. Дополнительно возможно указать, следует ли перед перезагрузкой проверить диски с помощью команды `fsck`.

Синтаксис команды:

```
shutdown [<параметр>] [<время>] [<сообщение>]
```

где `<параметр>` — параметр, определяющий действие команды (без параметра команда выполняет выключение компьютера);
`<время>` — время завершения работы системы в формате `чч:мм`. Значение может быть также задано в формате `+m`, где `m` — количество минут ожидания до завершения работы. Значение `+0` может быть заменено словом `now`;
`<сообщение>` — сообщение, посылаемое всем пользователям, зарегистрированным в системе в момент запуска команды.

В таблице 4 перечислены основные параметры команды `shutdown`.

Таблица 4

Параметр	Описание
`-k`	Послать предупреждение без реального завершения работы системы
`-r`	Перезагрузить компьютер после завершения работы
`-h`	Выключить компьютер после завершения работы
`-n`	Не синхронизировать диски. Этот параметр следует использовать 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`	Отменить уже запущенный процесс завершения работы. Параметр `<время>` при этом не может быть использован

Описание команды приведено в `man shutdown`.

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

2.4.2. halt и reboot

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

```
halt
```

или

```
shutdown -H
```

Команда регистрирует останов, уничтожает несущественные процессы, осуществляет системный вызов `sync`, ожидает завершения операций записи ФС, а затем прекращает работу ядра.

При выполнении команды `halt` с параметром `-n`:

```
halt -n
```

вызов `sync` подавляется. Данная команда используется после исправления корневого раздела программой `fsck` для того, чтобы ядро не могло затереть исправления старыми версиями суперблока.

При выполнении команды `halt` с параметром `-q`:

```
halt -q
```

инициируется немедленный останов, без синхронизации, уничтожения процессов и записи в файлы регистрации.

Команда `reboot` выполняет все основные операции, необходимые для останова системы (аналогично команде `halt`), а затем перезагружает компьютер с нуля. Для вызова команды выполнить в командной строке:

```
reboot
```

или

```
shutdown -r
```

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

3. СИСТЕМНЫЕ СЛУЖБЫ, СОСТОЯНИЯ И КОМАНДЫ

3.1. 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.

3.1.1. Управление службами

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.

Отличительной особенностью systemd является использование контрольных групп Linux, обеспечивающих иерархическую структуризацию служб: любая запущенная служба помещается в отдельную контрольную группу с уникальным идентификатором. Служба, запуская другую зависимую службу, становится родительской службой, а зависимая служба — дочерней. Дочерняя служба автоматически включается в группу с тем же идентификатором, что и родительская. Непривилегированные службы не могут изменить свое положение в иерархии. При штатном завершении работы родительской службы будут завершены и все ее дочерние службы.

Информация о менеджере systemd также приведена в `man systemd`.

Описание использования менеджера systemd для управления доступом приведено в РУСБ.10153-02 97 01-1.

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.

Юниты менеджера systemd располагаются в каталогах `/etc/systemd/system`, `/run/systemd/system`, `/usr/lib/systemd/system`, а также в пользовательских каталогах.

Приоритет выполнения юнитов зависит от их расположения:

- `/usr/lib/systemd/system/` — юниты из установленных пакетов, имеют минимальный приоритет;

- /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.

Также в ОС доступен механизм управления службами systemV, сохраненный для обеспечения совместимости. Менеджер systemV управляет сценариями запуска в каталогах /etc/init.d, /etc/rc{0-6,S}.d.

Таким образом, администратор ОС может использовать два инструмента для управления службами:

1) /usr/sbin/service (команда service) — устаревший инструмент, работающий только со службами, сценарии управления которых находятся в каталогах /etc/init.d, /etc/rc{0-6,S}.d;
2) /bin/systemctl (команда systemctl) — инструмент для управления всеми службами.

Инструменты обеспечивают интерфейс пользователя с юнитами/сценариями. Юниты/сценарии обеспечивают интерфейс управления службами, предоставляя администратору 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 проверки доступных параметров с помощью команды service, выполненной с названием сценария в качестве параметра.

Пример

Команда запроса доступных параметров для службы cron:

```
sudo /usr/sbin/service cron
```

Результат выполнения команды:

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

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:

```
sudo /usr/sbin/service --status-all
```

Пример

Вывод команды `/usr/sbin/service --status-all` проверки состояния служб:

```
[ + ]  acpi-support
[ + ]  acpid
[ - ]  anacron
...
```

Юниты systemd имеют фиксированный набор параметров, оформленных в виде параметров команды `systemctl`, например, `start, stop, reload, restart` и т.д.

Для просмотра списка установленных юнитов выполнить команду:

```
sudo 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  <тип_юнита>
```

Для получения списка юнитов, которые менеджер systemd загрузил и пробовал загрузить, не зависимо от их состояния в текущий момент, используется команда `systemctl` с параметром `-a`.

Пример

Для получения списка юнитов типа `service`, которые загрузил и пробовал загрузить менеджер systemd, выполнить команду:

```
systemctl -t service -a
```

Результат выполнения команды:

```
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
```

...

Основные параметры для использования с инструментом командной строки `systemctl` приведены в таблице 5.

Таблица 5

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`)

Полное описание команды `systemctl` приведено в `man systemctl`.

3.1.2. 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 секции, соответствующие конкретному типу юнита.

Секция [Unit] содержит описание юнита, а также информацию о зависимостях при запуске юнита. Основные параметры секции:

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

Секция [Service] присутствует в юнитах службы и может содержать следующие параметры, определяющие запуск службы:

1) Type= — определяет тип запуска службы:

а) simple — служба считается запущенной, когда завершился основной процесс службы (процесс, определенный в ExecStart=, считается основным процессом). Не рекомендуется использовать данный тип, если другие службы зависят от очередности при запуске данной службы. Исключение — активация сокета;

б) forking — служба считается запущенной, когда основной (родительский) процесс службы создал дочерний процесс, при этом родительский процесс завершился. Дочерний процесс продолжает функционировать в качестве основного. Рекомендуется использовать данный тип для запуска классических демонов. Потребуется также задать значение параметра PIDFile= для отслеживания основного процесса;

в) oneshot — похож на тип simple, используется для сценариев, которые завершаются после выполнения одного задания;

г) notify — похож на тип simple, но служба запускается после отправки менеджеру systemd сигнала о своей готовности;

д) dbus — похож на тип simple, но ожидает появления в системной шине DBus шины, указанной в BusName=;

е) idle — менеджер systemd отложит выполнение службы и запустит ее после запуска остальных служб;

2) PIDFile= — расположение pid-файла службы;

3) ExecStart= — указывает на команду, которая должна быть выполнена при запуске службы;

4) ExecStop= — указывает на команды, которые должны быть выполнены для завершения службы, запущенной в ExecStart=;

5) ExecReload= — указывает на команду, которая должна быть выполнена для перезапуска службы;

6) Restart= — определяет перезапуск службы в случае самостоятельного или принудительного завершения основного процесса или при возникновении ошибки;

7) RemainAfterExit — позволяет считать службу активной даже в случае, если все ее процессы завершились. Значение по умолчанию no (нет).

Общие параметры, которые могут содержаться в секциях [Service], [Socket], [Mount], [Swap]:

1) WorkingDirectory= — рабочий каталог службы;

2) User= — пользователь, от имени которого будет запущена служба;

3) Group= — группа, от имени которой будет запущена служба;

4) OOMScoreAdjust= — приоритет завершения процесса при нехватке памяти, где 1000 — максимальное значение, означающее полный запрет на завершение процесса;

5) KillMode= — указывает на порядок завершения процессов данного юнита.

3.2. 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 `multi-user.target` (с уровням выполнения 2, 3 и 4) — 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*
```

Пример вывода команды:

```
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 SystemV, 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
```

или команда:

```
sudo init <уровень_выполнения>
```

Данные команды изменят только текущий уровень выполнения и их действие не повлияет на последующие загрузки системы.

Пример

Для перехода в целевое состояние командой `systemctl` выполнить:

```
systemctl isolate multi-user.target
```

Для перехода в целевое состояние командой `init` выполнить:

```
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 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 текущего уровня выполнения выполнить команду:

```
sudo runlevel
```

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

```
sudo systemctl set-default <имя_состояния>.target
```

В новое состояние по умолчанию система будет переведена после перезагрузки. Для принудительного перевода системы в нужное состояние без перезагрузки используется команда `systemctl` с параметром `isolate` и именем целевого состояния (имя состояния может быть указано без расширения `.target`). или команда `init` с указанием уровня выполнения.

Для обеспечения совместимости с более ранними реализациями помимо запуска/остановки юнитов, определенных в файлах `.target`, при переводе системы в другое целевое состояние 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 systemctl`, `man init`.

3.3. Системные команды

Основные системные команды ОС приведены в таблице 6.

Таблица 6

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 строки (в т.ч. в файлах), содержащей заданное регулярное выражение

Продолжение таблицы 6

Команда	Назначение
fgrep	Поиск строки (в т.ч. в файлах), содержащей заданный фиксированный шаблон
file	Определение типа файла
find	Поиск файла по различным признакам в иерархии каталогов
gettext	Получение строки интернационализации из каталогов перевода
grep	Поиск строки (в т.ч. в файлах), 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
hostname	Вывод и задание имени хоста
install	Копирование файла с установкой атрибутов
ipcrm	Удаление средства IPC
ipcs	Вывод информации о средствах IPC
kill	Отправка процессу сигнала прекращения выполнения
killall	Отправка всем процессам с указанным именем сигнала прекращения выполнения
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

Окончание таблицы 6

Команда	Назначение
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
who	Вывод списка пользователей системы

Описание команд приведено на страницах руководства man.

3.3.1. Планирование запуска команд

3.3.1.1. 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 7 показаны варианты использования команды at.

Таблица 7

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.

3.3.1.2. 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> <день_недели> <задание>
```

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

Примечание. Символ * означает соответствие любому корректному значение.

Пример

Шаблон:

```
02 00 01 * *
```

определяет, что команда должна быть запущена в 00 часов 2 минуты каждого первого числа любого месяца (символ * в четвертом поле) независимо от дня недели (символ * в пятом поле).

В таблице 8 приведены допустимые значения полей записей `crontab`.

Таблица 8

Поле	Диапазон
<минуты>	00–59
<часы>	00–23 (полночь — 00)
<день_месяца>	01–31
<месяц>	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 9.

Таблица 9

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.

3.3.2. Администрирование многопользовательской и многозадачной среды

3.3.2.1. who

Для получения списка пользователей, работающих в ОС, используется инструмент командной строки who. Результатом выполнения команды является список, содержащий идентификаторы активных пользователей, терминалы и время входа в систему.

Пример

Результат выполнения команды who:

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

Основные параметры команды who:

1) -u — вывести список пользователей с указанием времени бездействия (символ «.» (точка) означает, что пользователь активно работал в последнюю минуту, old — что последний раз нажатие клавиш было более суток назад);
2) -H — вывести подробную информацию о пользователях. При этом выводится строка заголовка таблицы пользователей, описание столбцов приведено в таблице 10.

Таблица 10

Поле	Описание
ИМЯ	Имя пользователя
ЛИНИЯ	Использованные линии и терминалы
ВРЕМЯ	Время, прошедшее после регистрации пользователя в системе
IDLE	Время, прошедшее со времени последней активной работы пользователя
PID	Идентификатор процесса входной оболочки пользователя
КОММЕНТАРИЙ	Комментарий

Пример

Выполнение команды who с параметрами -u и -H:

```
who -uH
```

Результат выполнения команды:

```
ИМЯ   ЛИНИЯ    ВРЕМЯ           IDLE   PID     КОММЕНТАРИЙ
root console Dec 12 08:00   .      10340
```

Подробное описание команды приведено в man who.

3.3.2.2. 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 11 приведены четыре основных поля информации для каждого процесса, выводимые командой `ps`.

Таблица 11

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

Подробное описание команды приведено в `man ps`.

3.3.2.3. 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`.

3.3.2.4. nice

Команда `nice` позволяет предопределять приоритет выполнения процесса (фонового или переднего плана) во время его запуска.

При запуске все процессы имеют одинаковый приоритет и ОС равномерно распределяет между ними процессорное время. С помощью команды `nice` можно понизить приоритет выбранного процесса, предоставив другим процессам больше процессорного времени.

Приоритет выполнения процесса может изменяться от -20 (наивысший приоритет) до 19 (наименьший приоритет). По умолчанию приоритет каждого процесса равен 10.

Повышение приоритета процесса осуществляется от имени администратора.

Синтаксис команды `nice`:

```
nice -<число> <команда>
```

Параметр `<число>` определяет на какое значение должен быть изменен приоритет выбранного процесса. Чем больше значение параметра `<число>`, тем меньше будет приоритет выбранного процесса.

Пример

Для процесса сортировки, запущенного командой:

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

необходимо повысить приоритет над процессом печати.

Для этого необходимо запустить процесс печати с уменьшенным приоритетом, выполнив команду:

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

Или назначить процессу печати самый низкий приоритет, выполнив команду:

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

Для назначения процессу максимального приоритета -20 необходимо от имени администратора выполнить команду:

```
nice --30 <команда> &
```

Подробное описание команды приведено в `man nice`.

3.3.2.5. renice

Команда `renice` позволяет изменить приоритет запущенного процесса. Повышение приоритета процесса осуществляется от имени администратора.

Синтаксис команды `renice`:

```
renice -<число> <PID>
```

где `PID` — идентификатор процесса.

Определить PID можно с помощью команды `ps`:

```
ps -e | grep <имя_процесса>
```

Команда `grep` отфильтрует записи по имени нужного процесса.

Возможно изменить приоритет всех процессов пользователя или группы пользователей, для этого в команде `renice` используется идентификатор пользователя или группы.

Пример

Для изменения приоритета процесса текущего пользователя (`pav`) необходимо:

1) отобразить идентификаторы всех процессов, запущенных текущим пользователем, выполнив команду:

```
ps -ef | grep $LOGNAME
```

Результат выполнения команды:

```
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 {};
```

2) уменьшить приоритет процесса `find` с идентификатором 19940, выполнив команду:

```
renice -5 19940
```

Подробное описание команды приведено в `man renice`.

3.3.2.6. kill

Команда `kill` отправляет сигнал указанному процессу или процессам. Каждый сигнал имеет номер и название. Для просмотра всех сигналов необходимо выполнить команду:

```
kill -l
```

Синтаксис команды `kill`:

```
kill [-<сигнал>] <PID_1> [<PID_2> [...]]
```

где `<сигнал>` — номер сигнала или его название. Если параметр не задан, то по умолчанию будет применен сигнал с номером 15 (`SIGTERM`) на завершение выполнения процесса;
`<PID_n>` — идентификатор процесса.

С помощью параметра `<сигнал>` можно, например, дать указание процессу перечитать конфигурационные файлы без прекращения работы.

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

Примеры:

1. Завершить процесс с идентификатором 127:

```
kill -SIGTERM 127
```

или:

```
kill -15 127
```

2. Завершить процессы с идентификаторами 127 и 240:

```
kill 127 240
```

Для завершения процесса, только что запущенного в фоновом режиме, необходимо выполнить команду:

```
kill $!
```

Для завершения всех фоновых процессов необходимо выполнить команду:

```
kill 0
```

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 приводит к завершению дочерних (кроме запущенных с помощью `nohup`). Однако для полной уверенности в завершении всех процессов, связанных с данным, следует указывать их в команде `kill`.

Некоторые процессы могут игнорировать посылаемые им сигналы, включая сигнал 15 (`SIGTERM`). Сигнал с номером 9 (`SIGKILL`) не может быть проигнорирован процессом, и процесс будет принудительно завершен. Например, если процесс не завершился после выполнения команды:

```
kill <PID_процесса>
```

то необходимо выполнить команду:

```
kill -9 <PID_процесса>
```

После выполнения команды процесс завершится без возможности корректно закрыть файлы, что может привести к потере данных.

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`.

4. УПРАВЛЕНИЕ ПРОГРАММНЫМИ ПАКЕТАМИ

В ОС используются программные пакеты (далее по тексту — пакеты) в формате DEB (файлы с расширением `.deb`). Для управления пакетами предназначены набор команд нижнего уровня `dpkg` и комплекс программ высокого уровня `apt`, `apt-cache` и `aptitude`.

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.

4.1. dpkg

Инструмент командной строки `dpkg` предназначен для операций с пакетами на локальном уровне. С помощью `dpkg` можно устанавливать и удалять пакеты, собирать пакеты из исходных текстов, получать информацию о конкретном пакете и об установленных в системе пакетах.

Для установки пакета необходимо выполнить команду:

```
dpkg -i <полное_имя_пакета>
```

Пример

Для установки пакета `iptables_1.4.21-2_amd64.deb`, расположенного в домашнем каталоге пользователя `/home/user1`, выполнить следующую команду:

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

В случае нарушения зависимостей будет выведено сообщение об ошибке, в котором будут перечислены все необходимые пакеты, которые следует установить для разрешения обязательных зависимостей.

Для удаления пакета с сохранением его конфигурационных, пользовательских и других файлов (в случае, если данный пакет не связан зависимостями с другими установленными пакетами) следует выполнить команду:

```
dpkg -r <имя_пакета>
```

Пример

Для удаления пакета `iptables_1.4.21-2_amd64.deb` необходимо выполнить команду:

```
dpkg -r iptables
```

Для удаления пакета и его конфигурационных, пользовательских и других файлов (в случае, если данный пакет не связан зависимостями с другими установленными пакетами) следует выполнить команду:

```
dpkg -P <имя_пакета>
```

Пример

Для удаления пакета `iptables_1.4.21-2_amd64.deb` необходимо выполнить команду:

```
dpkg -P iptables
```

При удалении пакета с зависимостями с другими пакетами будет отображено сообщение об ошибке с перечнем зависимостей.

Подробное описание команды приведено в `man dpkg`.

4.2. apt

Инструмент командной строки `apt` предназначен для выполнения операций с пакетами (при наличии доступа к сетевым или локальным репозиториям): устанавливать, удалять, обновлять, разрешать зависимости. А также искать пакеты по заданным критериям и просматривать подробную информацию о пакете.

4.2.1. Настройка доступа к репозиториям

Информация о сетевых и локальных репозиториях содержится в файле `/etc/apt/sources.list`. В файле указывается список источников пакетов, который используется программами для определения местоположения репозиториев. 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 1.8_x86-64 DVD ]/ 1.8_x86-64 contrib main
     non-free
deb https://dl.astralinux.ru/astra/stable/1.8_x86-64/repository-main/
     1.8_x86-64 main contrib non-free
deb https://dl.astralinux.ru/astra/stable/1.8_x86-64/repository-update/
     1.8_x86-64 main contrib non-free
deb https://dl.astralinux.ru/astra/stable/1.8_x86-64/uu/last/
     repository-update/ 1.8_x86-64 main contrib non-free
deb https://dl.astralinux.ru/astra/stable/1.8_x86-64/repository-base/
     1.8_x86-64 main contrib non-free
```

При установке ОС с DVD-диска строка `deb cdrom...` автоматически записывается в файл `sources.list`. Добавить данную строку в список источников также можно при помощи команды:

```
apt-cdrom add
```

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

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

4.2.2. Установка и удаление пакетов

После установки ОС создается локальная БД с информацией обо всех пакетах, которые находились на DVD-диске с дистрибутивом, и репозиторий установленных пакетов. Информацию о каждом установленном пакете можно просмотреть.

Пример

Для просмотра информации о пакете `iptables` выполнить команду:

```
apt show iptables
```

Обновить содержимое локальной БД можно при помощи команды:

```
apt update
```

Данную команду необходимо выполнять при каждом изменении списка источников пакетов или при изменении содержимого этих источников.

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

```
apt dist-upgrade
```

Установка отдельного пакета (если он отсутствовал в системе) выполняется командой:

```
apt install <имя_пакета>
```

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

Удаление пакета (с сохранением его конфигурационных файлов) выполняется командой:

```
apt remove <имя_пакета>
```

Для удаления пакета вместе с его конфигурационными файлами (кроме конфигурационных файлов из домашних каталогов пользователей) применяется команда:

```
apt remove --purge <имя_пакета>
```

Полное описание инструмента `apt` приведено в `man apt`.

5. БАЗОВЫЕ СЕТЕВЫЕ СЛУЖБЫ

5.1. Протокол TCP/IP

5.1.1. 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.

5.1.2. 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`.

5.1.3. Маршрутизация

5.1.3.1. 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 метрики, в котором указано значение приоритета маршрута на определенном сетевом интерфейсе (если таблица содержит противоречивую информацию). Для направления пакета по конкретному адресу подбирается

наиболее подходящий маршрут. Если нет такого маршрута и нет маршрута по умолчанию, то отправителю возвращается ошибка «network unreachable» (сеть недоступна).

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

5.1.3.2. 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.

5.1.4. Создание сети TCP/IP

Процесс создания сети TCP/IP состоит из следующих этапов:

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

5.1.4.1. 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.).

5.1.4.2. Назначение IP-адресов

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

Существует возможность создания виртуального сетевого интерфейса (loopback), который реализован программно и не связан с оборудованием, но при этом полностью интегрирован во внутреннюю сетевую инфраструктуру компьютерной системы.

Для того, чтобы можно было обращаться к компьютерам по их именам рекомендуется использовать службу DNS, также соответствие между именем и IP-адресом может быть задано в файле `/etc/hosts` на компьютере, с которого выполняется обращение.

5.1.4.3. 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
```

Параметр `up` включает интерфейс, а параметр `down` выключает его.

Описание команды приведено в `man ifconfig`.

5.1.4.4. 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`.

5.1.5. Проверка и отладка сети

5.1.5.1. 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`.

5.1.5.2. 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`.

5.1.5.3. 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`.

5.2. Протокол FTP

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

5.2.1. 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`.

5.2.2. Служба vsftpd сервера FTP

В ОС служба `vsftpd` устанавливается командой:

```
apt install vsftpd
```

После установки службы `vsftpd` по умолчанию в конфигурационном файле `/etc/vsftpd.conf` указаны параметры для работы с включенной IPv4- и IPv6-адресацией — для параметров `listen` и `listen_ipv6` установлены следующие значения:

```
listen=NO
listen_ipv6=YES
```

Для приема соединения как от клиентов IPv4, так и от клиентов IPv6 достаточно, чтобы для параметра `listen_ipv6` было установлено значение `YES`, при этом значение параметра `listen` всегда будет интерпретироваться как `YES`.

Если в системе включена IPv6-адресация, то служба `vsftpd` запускается автоматически без дополнительных настроек.

Если в системе отключена IPv6-адресация или необходимо использовать только IPv4-адресацию, то для запуска службы `vsftpd` требуется отредактировать файл `/etc/vsftpd.conf` — для параметров `listen` и `listen_ipv6` должны быть установлены следующие значения:

```
listen=YES
listen_ipv6=NO
```

Для настройки `vsftpd` не требуется указывать все доступные параметры, а достаточно указать только те, значения которых следует переопределить. Параметры, не указанные явным образом в файле `/etc/vsftpd.conf`, будут принимать значения по умолчанию. Значения, принимаемые по умолчанию, приведены в `man vsftpd.conf`.

В конфигурационном файле `/etc/vsftpd.conf` присутствуют параметры, которые зависят от других параметров. Если один параметр, от которого зависит другой, отключен, то и зависимый параметр также будет отключен. Например, если параметр `local_enable`, позволяющий авторизоваться локальным пользователям, будет отключен, то зависящий от него параметр `local_umask` также будет отключен.

Для запуска службы `vsftpd` с параметрами, отличными от указанных в файле `/etc/vsftpd.conf`, необходимо использовать инструмент командной строки `vsftpd`. Таким образом, если значение параметра, переданное в командной строке, не совпадает с указанным в конфигурационном файле, то будет применено значение параметра, указанное в командной строке, так как оно имеет приоритет над указанным в конфигурационном файле. Значения параметров, указанных в командной строке, применяются последовательно.

После установки службы `vsftpd` создается каталог с документацией службы `/usr/share/doc/vsftpd`, где каталог `examples` содержит примеры конфигурационного файла `vsftpd.conf`.

Описание службы `vsftpd` и файла `/etc/vsftpd.conf` приведено в `man vsftpd` и `man vsftpd.conf` соответственно.

5.3. Протокол DHCP

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

DHCP-сервер представлен пакетом `isc-dhcp-server` для его быстрой настройки.

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

```
sudo apt install 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. Каждая строка задана шаблоном «параметр — значение». Параметр может быть общим или стоять перед `option`. Параметр, следующий за `option`, — это ключ настройки, он состоит 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 12.

Таблица 12

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 12

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 13.

Таблица 13

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 13

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 14.

Таблица 14

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`.

5.4. Протокол NFS

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

В ОС используется реализация службы NFS, работающая 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.

5.4.1. 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 15.

Таблица 15

Параметр	Описание
`rw`	Предоставляет права на чтение и запись
`ro`	Предоставляет права только на чтение

Окончание таблицы 15

Параметр	Описание
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
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 совместно используемого каталога /nfsshare

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

ВНИМАНИЕ! Использование пробелов между IP-адресом/именем клиента и правами его доступа в файле `/etc/exports` 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
```

5.4.2. 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 совместно используемые ресурсы. Список доступных ресурсов можно проверить, выполнив команду:

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

Для монтирования совместно используемого 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`.

5.5. 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.

5.5.1. 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`.

5.5.2. Настройка сервера службы доменных имен `named`

Конфигурационные параметры службы `named` хранятся в файлах каталога `/etc/bind/`, перечень конфигурационных файлов приведен в таблице 16.

Таблица 16 – 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`

Окончание таблицы 16

Файл	Описание
/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`».

5.5.3. 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`.

5.6. 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 разработчика wiki.astralinux.ru.

5.6.1. 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 17. Пустые строки, а также строки, начинающиеся с #, игнорируются. Названия параметров не чувствительны к регистру символов.

Таблица 17

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
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

Продолжение таблицы 17

Параметр	Описание
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
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

Продолжение таблицы 17

Параметр	Описание
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, DEBUG (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
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

Окончание таблицы 17

Параметр	Описание
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

5.6.2. Клиент ssh

В роли клиента выступает инструмент командной строки ssh. Синтаксис команды:

```
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]
```

Подробное описание параметров инструмента приведено man ssh. 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.

Клиент ssh может заходить на сервер sshd под любым именем, используя параметр:

```
-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.

Инструментssh 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 18 описаны параметры, которые могут присутствовать в файле $HOME/.ssh/config или /etc/ssh/ssh_config. Пустые строки и комментарии игнорируются.

Таблица 18

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
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

Продолжение таблицы 18

Параметр	Описание
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. По умолчанию глобальная база ключей компьютера хранится в файле /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
KerberosAuthentication	Указывает на то, разрешена ли аутентификация с применением Kerberos

Продолжение таблицы 18

Параметр	Описание
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, DEBUG
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
UsePrivilegedPort	Указывает на то, можно ли использовать привилегированный порт для установления исходящих соединений. Значение по умолчанию no

Окончание таблицы 18

Параметр	Описание
`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.

5.7. Службы точного времени

В состав ОС входят следующие службы точного времени:

1) службы, использующие протокол синхронизации времени NTP:
   а) `systemd-timesyncd` — клиентская служба синхронизации времени, используется в ОС по умолчанию. Описание службы приведено в 5.7.1;
   б) `chronyd` — клиент и сервер протокола точного времени NTP. Описание службы приведено в 5.7.2;
2) служба времени высокой точности PTP (Precision Time Protocol) — описание службы приведено в 5.7.3.

При настройке служб времени используются термины для обозначения времени, приведенные в таблице 19.

Таблица 19

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)

Окончание таблицы 19

Термин	Описание	Пример
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

5.7.1. Служба systemd-timesyncd

Служба `systemd-timesyncd` предназначена для использования в роли клиента и не может выполнять функции сервера точного времени. Подходит для синхронизации времени с доверенным сервером времени в локальной сети. Может применяться в системах, где не требуется высокая точность синхронизации времени. Поддерживает только упрощенный протокол передачи времени.

5.7.1.1. Установка и настройка

Служба `systemd-timesyncd` устанавливается автоматически, если при установке ОС был выбран для установки компонент «Консольные утилиты».

Служба синхронизации запускается автоматически, если при установке ОС для настройки времени была выбрана синхронизация времени по сети.

Для запуска службы синхронизации времени вручную и для ее добавления в автозапуск выполнить команду:

```
systemctl enable systemd-timesyncd
```

Служба не может работать одновременно со службами `ntpd` или `chronyd` (не будет выполняться синхронизация времени). Служба завершает свою работу без сообщений об ошибке, если обнаружит на компьютере:

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

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

Состояние службы `systemd-timesyncd` можно проверить командой:

```
systemctl status systemd-timesyncd
```

Также для проверки статуса службы синхронизации времени можно использовать команду:

```
timedatectl status
```

Примечание. Необходимо учитывать, что `timedatectl` может использоваться и с другими службами синхронизации времени. Таким образом, если просматривать статус службы времени, то отображается статус запущенной в данной момент службы. А при использовании `timedatectl`, например, для запуска службы времени будет запущена установленная в системе служба.

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)
System clock synchronized: yes
NTP service: active
RTC in local TZ: no
```

Автоматический запуск службы отключается командой:

```
systemctl disable systemd-timesyncd
```

или

```
timedatectl set-ntp false
```

5.7.1.2. Выбор серверов времени

Основные и резервные серверы времени указываются в конфигурационных файлах службы `systemd-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= — список имен основных серверов единого времени, разделенный пробелами. Объединяется со списком имен, полученных от службы systemd-networkd. По умолчанию список пустой и используются резервные серверы времени, указанные в параметре FallbackNTP=;
2) FallbackNTP= — список имен резервных серверов единого времени, разделенный пробелами.

Служба systemd-timesyncd перебирает по очереди серверы из основного списка и, если не удалось связаться ни с одним из серверов, обращается к серверам из резервного списка.

По умолчанию для службы systemd-timesyncd указаны российские серверы точного времени ВНИИФТРИ.

Дополнительно служба systemd-timesyncd может получать 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.

5.7.2. Служба chronyd

Служба точного времени chronyd рекомендована к применению вместо службы ntpd. Служба chronyd может выступать в роли клиента и сервера протокола сетевого времени NTP и позволяет:

- быстрее синхронизировать системные часы;
- использовать аппаратные отметки времени, что обеспечивает более точную синхронизацию времени;
- не прекращать работу службы синхронизации, обнаружив слишком большое отклонение времени, а попытаться выполнить коррекцию времени;
- работать, если порт 123 закрыт для исходящих запросов.

Службы chronyd обеспечивает надежную работу синхронизации при нестабильных сетевых соединениях, частичной доступности или перегрузки сети.

5.7.2.1. Установка

При установке ОС служба `chronyd` автоматически не устанавливается. Для установки службы требуется установить пакет `chrony` (при этом будет удален пакет установленной ранее службы `systemd-timesyncd`):

```
apt install chrony
```

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

5.7.2.2. Настройка

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

По умолчанию для службы `chronyd` указаны российские серверы точного времени ВНИИФТРИ.

Для настройки работы службы `chronyd` в режиме сервера времени (т.е. чтобы служба отвечала клиентам на запросы) необходимо отредактировать (или при его отсутствии — создать) конфигурационный файл `/etc/chrony/chrony.conf`. В конфигурационном файле требуется добавить строку с разрешениями клиентам подключаться к серверу NTP.

Пример

Настройка разрешений подключаться к NTP-серверу:

1) разрешить всем клиентам:

```
allow
```

2) разрешить только клиенту с определенным IP-адресом:

```
allow 10.10.12.5
```

3) разрешить клиентам определенной сети:

```
allow 10.10.12
```

Более подробная информация о настройке конфигурационного файла `/etc/chrony/chrony.conf` приведена в `man chrony.conf`.

После редактирования конфигурационного файла перезапустить службу `chronyd`:

```
systemctl restart chronyd
```

5.7.3. 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 (`chronyd`) и службы времени высокой точности PTP.

5.7.3.1. Проверка оборудования

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

Проверить, поддерживает ли сетевая карта аппаратные отметки времени, можно из командной строки с помощью инструмента `ethtool`. Для этого необходимо:

1) установить пакет `ethtool`, если он не был установлен ранее, выполнив команду:

```
apt install ethtool
```

2) проверить оборудование, выполнив команду:

```
ethtool -T eth0
```

Если сетевая карта не поддерживает аппаратные отметки времени, возможно настроить и использовать службу времени высокой точности на основе программных отметок времени, но это повлечет снижение точности. Настройка использования сетевых карт без аппаратной поддержки отметок времени приведена в 5.7.3.3.

5.7.3.2. Установка службы PTP

Служба времени высокой точности PTP устанавливается из пакета `linuxptp` командой:

```
apt install linuxptp
```

5.7.3.3. Настройка службы ptp4l

Для включения службы `ptp4l` необходимо раскомментировать в конфигурационном файле `/etc/linuxptp/timemaster.conf` секцию домена точного времени `[ptp_domain 0]` и указать данные сетевой карты.

Пример

Настройки домена точного времени, использующего интерфейс `eth0`:

```
[ptp_domain 0]
interfaces eth0
delay 10e-6
```

Домен точного времени обслуживается службой `ptp4l`.

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

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

```
time_stamping software
```

Подробное описание настроек конфигурационного файла приведено в `man pdp4l`.

5.7.3.4. Настройка службы timemaster

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

Необходимо разрешить автоматический запуск службы `timemaster` при старте ОС, выполнив команду:

```
systemctl enable timemaster
```

Подробно параметры настройки описаны в `man timemaster`.

5.7.3.5. 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, служба `phc2sys` запускается автоматически с нужными параметрами.

При работе с сетевыми картами, не поддерживающими аппаратные отметки времени, служба `phc2sys` не запускается.

5.7.3.6. Запуск службы PTP

После завершения настройки запуск всех служб осуществляется командой:

```
systemctl start timemaster
```

Служба `timemaster` запустит все остальные службы.

Пример

Проверка состояния службы `timemaster`:

```
systemctl status timemaster
```

Результат выполнения команды при штатном функционировании и наличии аппаратной поддержки:

```
timemaster.service - Synchronize system clock to NTP and PTP time sources
Loaded: loaded (/lib/systemd/system/timemaster.service; enabled; preset:
      enabled)
Active: active (running) since Thu 2024-02-22 12:33:42 MSK; 2s ago
Docs: man:timemaster
Main PID: 32390 (timemaster)
Tasks: 3 (limit: 4001)
Memory: 1.1M
CPU: 12ms
CGroup: /system.slice/timemaster.service
   32390 /usr/sbin/timemaster -f /etc/linuxptp/timemaster.conf
   32391 /usr/sbin/chronyd -n -f /var/run/timemaster/chrony.conf
   32392 /usr/sbin/ptp4l -l 5 -f /var/run/timemaster/ptp4l.0.conf -H -i
        eth0
   32393 /usr/sbin/phc2sys -l 5 -a -r -R 1.00 -z /var/run/timemaster/
         ptp4l.0.socket -n 0 -E ntpshm -M 0
```

5.7.3.7. 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.

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
```

5.7.4. Ручная синхронизация времени ntpdate

Инструмент `ntpdate` применяется для проверки работы сервера времени и/или синхронизации с ним системного времени.

Ручная синхронизация времени может применяться для:

1) проверки, независимой от запущенных служб времени, степени рассинхронизации времени;
2) проверки доступности серверов через порт 123.

Инструмент устанавливается в ОС по умолчанию.

Запускать необходимо с правами `root`. Возможен запуск как из командной строки (вручную), так и из стартового сценария, 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
```

Синтаксис команды:

```
ntpdate [-параметры] <NTP-сервер>
```

Основные параметры инструмента приведены в таблице 20.

Таблица 20

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 служба `ntpd`, и, если служба `ntpd` 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
```

Более подробная информация приведена в `man ntpdate`.

5.8. Средство создания защищенных каналов

Для создания между компьютерами сети защищенных каналов типа точка-точка или сервер-клиент 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.

ВНИМАНИЕ! OpenVPN не является сертифицированным криптографическим средством защиты информации и не может применяться в целях криптографической защиты информации. Основное назначение OpenVPN в составе ОС — обеспечение целостности заголовка IP-пакетов при передаче по сетям связи.

Для обеспечения безопасности управляющего канала и потока данных OpenVPN использует библиотеку `OpenSSL` (устанавливается автоматически при установке ОС). При этом OpenVPN использует алгоритмы защитного преобразования `OpenSSL` 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`.

5.8.1. Установка

Для установки OpenVPN необходимо:

1) на компьютере, предназначенном на роль сервера OpenVPN, и на клиентских компьютерах установить пакет `openvpn`:

```
apt install openvpn
```

2) на компьютере, предназначенном на роль сервера, для управления службой `openvpn` установить инструмент командной строки `astra-openvpn-server`:

```
apt install astra-openvpn-server
```

Примечание. При установке инструмента командной строки `astra-openvpn-server` будет автоматически установлен и настроен пакет алгоритмов защитного преобразования ГОСТ `libgost-astra`.

5.8.2. Управление службой openvpn

5.8.2.1. Параметры инструмента командной строки

Команды, используемые с инструментом командной строки `astra-openvpn-server`, приведены в таблице 21.

Таблица 21

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

Продолжение таблицы 21

Параметр	Описание
`status`	Проверить службу. После выполнения данной команды другие команды не выполняются
`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-адрес `10.8.0.0` и маска `255.255.255.0`), например: `astra-openvpn-server server "10.8.0.0 255.255.255.0"`
`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 центра аутентификации
`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

Окончание таблицы 21

Параметр	Описание
Генерация и отзыв ключей клиентов	
`client <имя_клиента>`	Создать ключи и сертификаты для указанного клиента
`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.

5.8.2.2. 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 центр аутентификации, размещается в каталоге:

`/etc/openvpn/openvpn-certificates`

- сертификат открытого ключа центра аутентификации:

`/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 аутентификации 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 21) 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 21).

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 центра аутентификации не выполняется.

5.8.2.3. Генерация сертификатов и ключей

При использовании собственного центра аутентификации 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 21).

Команда генерации ключей клиента несовместима с параметрами сервера и командами управления сервером (см. таблицу 21).

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 центром аутентификации.

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 центра аутентификации (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 21). В таблице 21 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 21).

5.8.2.4. 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 21).

При выполнении данной команды:

- сертификат клиента в базе данных центра аутентификации будет помечен как «отозванный»;
- будет создан (или обновлен ранее созданный) список отозванных сертификатов;
- новый список отозванных сертификатов будет скопирован в каталог `/etc/openvpn/keys`, сервер OpenVPN будет автоматически перезапущен для применения обновлений.

5.8.2.5. 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 центра аутентификации;
- удаляются все копии сертификатов сервера и клиентов;
- создается новый центр аутентификации;
- 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.

5.8.2.6. 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 командой:

```
apt 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 аутентификации, 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`). 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`.

5.8.3. 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
```

5.9. 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.

5.9.1. Состав

В состав Ansible входят модули, обеспечивающие развертывание, контроль и управление компонентами удаленных машин. Перечень основных модулей приведен в таблице 22.

Таблица 22

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`

5.9.2. Установка и настройка Ansible

На управляющей и управляемых машинах должен быть установлен Python.

Дополнительно для работы Ansible необходимы следующие Python-модули на управляющей машине:

- python-yaml;
- paramiko;
- python-jinja2.

Установка модулей осуществляется путем выполнения команды:

```
apt install python-yaml python-jinja2 python-paramiko python-crypto
```

Для установки Ansible выполнить команду:

```
apt 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 сценария, 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.

5.9.3. 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 23.

Таблица 23

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

Окончание таблицы 23

Параметр	Описание
`remote_user` (`user`)	Имя пользователя для авторизации на удаленном узле

6. СРЕДСТВА АУДИТА И РЕГИСТРАЦИИ СОБЫТИЙ

6.1. Аудит

В ОС отправка и регистрация информации о событиях в системе осуществляется в соответствии со стандартом Syslog. Стандарт определяет формат сообщений о событиях и правила их передачи и регистрации в журналах. Основное расположение файлов журналов — системный каталог `/var/log`.

Аудит основных системных событий с момента запуска ОС ведется в системном журнале `/var/log/syslog`.

Аудит событий постановки/снятия с контроля целостности исполняемых модулей и файлов данных, а также событий неудачного запуска неподписанных файлов осуществляется в журнале ядра `/var/log/kern.log`.

Аудит событий создания/удаления/изменения настроек учетных записей пользователей и начала/окончания сеансов работы учетных записей пользователей осуществляется в журнале `/var/log/auth.log`.

Аудит событий изменения для учетных записей полномочий по доступу к информации осуществляется в журнале `/var/log/auth.log`.

Аудит событий смены аутентифицирующей информации учетных записей осуществляется в журнале `/var/log/auth.log`.

Для аудита ОС также могут использоваться журналы различных служб и программ.

Для регистрации событий безопасности в ОС используется служба аудита `auditd`, описание которой приведено в РУСБ.10153-02 97 01-1, и подсистема регистрации событий (см. 6.2).

6.2. Подсистема регистрации событий

В ОС реализована подсистема регистрации событий, которая собирает информацию о событиях, выполняет ее регистрацию и предоставляет инструменты для просмотра собранных данных и реагирования на события. Регистрация событий безопасности выполняется с учетом требований ГОСТ Р 59548-2022.

Сбор и регистрацию событий осуществляет служба `syslog-ng`. Служба `syslog-ng` принимает информацию о событиях из различных источников (события от `auditd`, собственные подключаемые модули, файлы, прикладное ПО и др.), выполняет фильтрацию и обработку полученных данных, регистрирует события в журнал `/parsec/log/astra/events`, а также, в зависимости от конфигурации, может сохранять в файл, отправлять по сети и т.д.

Для управления подсистемой регистрации событий используются инструменты командной строки `astra-admin-events` и `astra-event-viewer`. Порядок использования инструментов приведен на соответствующих страницах помощи:

```
astra-admin-events -h
astra-event-viewer -h
```

Подробное описание подсистемы регистрации событий приведено в РУСБ.10153-02 97 01-1.

7. 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. 7.2).

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`, `/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) утилита копирования `rsync` (7.3);
2) утилиты архивирования `tar`, `cpio`, `gzip` (7.4).

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 атрибутов аудита описан в РУСБ.10153-02 97 01-1.

7.1. 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.

7.2. 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 изменений как в системе, так и в используемых технологиях или условиях функционирования.

7.2.1. 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 ин-

кремментного) 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 временных затрат.

Примечание. Расписание резервного копирования должно быть доведено до пользователей.

7.2.2. Планирование восстановления системы

При составлении плана резервного копирования следует определить:

1) план действий на случай аварийной ситуации;
2) как при необходимости может быть восстановлена система или отдельные файлы;
3) где хранятся и насколько доступны носители с резервными копиями и не могут ли они быть повреждены при сбоях 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.

7.3. Утилита копирования `rsync`

Все действия при использовании команды `rsync` выполняются от имени учетной записи администратора с использованием механизма `sudo`.

В таблице 24 приведены некоторые наиболее часто используемые параметры команды `rsync`.

Таблица 24

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

Окончание таблицы 24

Параметр	Назначение
`-o, --owner`	Сохранять владельца
`-A, --acls`	Сохранять списки контроля доступа ACL (включает `-p`)

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
```

ВНИМАНИЕ! Не рекомендуется использовать параметр `-l` для копирования символических ссылок при создании резервной копии домашних каталогов пользователей.

7.4. 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`.

7.4.1. tar

Команда `tar` может работать с рядом дисковых накопителей, позволяет просматривать архивы в ОС.

В таблице 25 приведены основные параметры команды `tar`.

Таблица 25

Параметр	Назначение
`--acls`	Сохраняет (восстанавливает) списки контроля доступа (ACL) каталогов и файлов, вложенных в архив
`-c, --create`	Создает архив

Окончание таблицы 25

Параметр	Назначение
-x, --extract, --get	Восстанавливает файлы из архива на устройстве, заданном по умолчанию или определенном параметром f
-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` следует 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. 3.3.1.2).

7.4.2. 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. Она 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 26 приведены основные параметры команды `cpio`.

Таблица 26

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.

8. СООБЩЕНИЯ АДМИНИСТРАТОРУ И ВЫЯВЛЕНИЕ ОШИБОК

8.1. Диагностические сообщения

При возникновении проблем в процессе функционирования ОС появляются диагностические сообщения трех типов: информационные, предупреждающие и сообщения об ошибках (примеры приведены в таблицах 27–29, соответственно). Администратор должен проанализировать диагностические сообщения и принять меры по устранению появившихся проблем.

Таблица 27 – Информационные сообщения

Сообщение ОС	Описание	Файл
`Setting hostname to <>`	Установка имени хоста <>	`hostname`
`Setting domainname to <>`	Установка имени домена <>	`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`	Неперезагружаемая зона	`named`
`Reconfig initiated`	Переконфигурирование запущено	`named`
`Zone not found`	Зона не найдена	`named`

Таблица 28 – 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 29 – 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 29

Сообщение ОС	Описание	Действия по устранению проблемы	Файл
`Superblock last mount time (значение времени) is in the future`	Неверная установка времени	См. 8.3	См. 8.3

8.2. 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 30.

Таблица 30

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 30

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`	Собирать данные всех возможных журналов регистрации событий, включая из незаданных областей и игнорируя ограничения по размеру. В данном случае может быть значительно увеличен размер отчетов
`-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 30

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`.

8.3. 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 BIOS 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 BIOS 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 BIOS 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 ТЕРМИНОВ

Закрытый ключ — сохраняемый в тайне ключ из ключевой пары, принадлежащий владельцу и не подлежащий распространению.

Ключ — параметр в виде последовательности псевдослучайных чисел (не предназначен для защиты информации в контексте использования для целей, установленных в документации изделия; к ключам не предъявляются требования по источнику псевдослучайных чисел, криптографической стойкости, времени действия и т. п.).

Ключевая пара — упорядоченная пара математически однозначно связанных ключей, определяющих взаимосвязанные защитные преобразования.

Открытый ключ — ключ из ключевой пары, который может быть сделан общедоступным.

Сертификат открытого ключа — артефакт, содержащий открытый ключ, информацию о владельце ключа и подтверждающий принадлежность открытого ключа владельцу, защищенный с применением закрытого ключа.

Хеш — строка бит, являющаяся выходным результатом функции хеширования.

Центр аутентификации — программный компонент, реализующий возможность подтверждения подлинности ключей с помощью сертификатов.

Цифровая подпись — результат преобразования хеша для его защиты от несанкционированного доступа с использованием закрытого ключа (не предназначена для криптографической защиты информации).

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

БД — база данных
ВМ — виртуальная машина
ЕПП — единое пространство пользователей
КСЗ — комплекс средств защиты
ЛВС — локальная вычислительная сеть
НСД — несанкционированный доступ
ОС — операционная система специального назначения «Astra Linux Special Edition»
ПО — программное обеспечение
СЗИ — средства защиты информации
ФС — файловая система
ЦА — центр аутентификации

AD — Active Directory (служба каталогов)
ACL — Access Control List (список контроля доступа)
API — Application Programming Interface (программный интерфейс приложения)
ARP — Address Resolution Protocol (протокол разрешения адресов)
BIOS — Basic Input-Output system (базовая система ввода-вывода)
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 (центр аутентификации)
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)
KVM — Kernel-based Virtual Machine (программное решение, обеспечивающее виртуализацию в среде Linux на платформе, которая поддерживает аппаратную виртуализацию на базе Intel VT (Virtualization Technology) либо AMD SVM (Secure Virtual Machine))
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)
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)
OCI — Open Container Initiative (проект, который разрабатывает открытые стандарты для сред контейнеризации)
OSD — Object Storage Device (устройство хранения объектов)
PAM — Pluggable Authentication Modules (подключаемые модули аутентификации)
PID — Process Identifier (идентификатор процесса)
PKI — Public Key Infrastructure (инфраструктура открытых ключей)
PTP — Precision Time Protocol (протокол точного времени)
POP3 — Post Office Protocol Version 3 (почтовый протокол, версия 3)
QEMU — Quick Emulator (средства эмуляции аппаратного обеспечения)
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 (системная служба, управляющая 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)
UEFI — Unified Extensible Firmware Interface (унифицированный расширяемый микропрограммный интерфейс)
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)

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					