50 1190 0101

Утвержден

РУСБ.10015-01-УД

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

ОПЕРАЦИОННАЯ СИСТЕМА СПЕЦИАЛЬНОГО НАЗНАЧЕНИЯ «ASTRA LINUX SPECIAL EDITION»

Руководство администратора

РУСБ.10015-01 95 01

Листов 166

2010

АННОТАЦИЯ

Настоящий документ является руководством администратора операционной системы специального назначения «Astra Linux Special Edition» РУСБ.10015-01 (далее по тексту — ОС).

В документе приведено назначение, установка и настройка ОС. Рассмотрены системные компоненты, сервисы и команды, базовые сетевые службы, средства организации единого пространства пользователей (ЕПП), защищенная графическая подсистема, управление программными пакетами, резервное копирование и восстановление данных, система печати, защищенная система управления базами данных (СУБД), защищенные комплексы программ гипертекстовой обработки данных и электронной почты.

Приведен список сообщений для администратора.

Документ предназначен для администраторов системы и сети.

СОДЕРЖАНИЕ

1. НАЗНАЧЕНИЕ

Административное управление в ОС отделено от общего доступа пользователей.

ОС позволяет администратору (суперпользователю) выполнять над файлом или процессом любую операцию. Кроме того, некоторые системные вызовы (обращения к ядру) может выполнять только суперпользователь. Некоторые системные вызовы доступны всем пользователям, но имеют специальные опции для суперпользователя.

Примеры операций, которые может выполнить только суперпользователь:

– монтирование и размонтирование ФС;

– изменение корневого каталога процесса командой `chroot`;

– создание файлов устройств;

– установка системных часов;

– изменение принадлежности файлов;

– увеличение лимитов использования ресурсов и назначение приоритетов процессов;

– задание host-имени системы;

– конфигурирование сетевых интерфейсов.

1.1. Пароль суперпользователя

Пароль следует выбирать так, чтобы его нельзя было определить. Наиболее безопасный пароль состоит из случайной последовательности букв, знаков препинания и цифр.

Как правило, достаточно надежен пароль, состоящий из двух случайно выбранных слов, разделенных знаком препинания, или из первых букв какой-нибудь фразы, набранных в разных регистрах. Такой пароль соответствует требованиям, предъявляемым к паролям. Пароль суперпользователя должен состоять из восьми символов. Задавать более длинный пароль не имеет смысла, потому что ОС обрабатывает только первые восемь символов.

Пароль суперпользователя следует менять:

– минимум раз в три месяца;

– каждый раз, когда увольняется сотрудник организации, знающий пароль;

– когда безопасность системы поставлена под угрозу.

1.2. Доступ к учетной записи суперпользователя

Существует несколько способов доступа к учетной записи суперпользователя:

– вход в систему под именем `root`;

– использование команды `su`;

– использование команды `sudo`.

1.2.1. Вход в систему под именем root

При запросе системы на ввод имени и пароля администратор может ввести имя `root` и соответствующий пароль. После этого будет создана новая сессия и все команды, исполняемые в этой сессии, будут иметь права и привилегии суперпользователя.

Войдя в систему под именем `root`, пользователь получает неограниченные возможности на все время, пока открыта сессия. Закрыть сессию можно с помощью команды `exit`.

1.2.2. su

Команда `su` используется пользователем для запуска команд от имени другого пользователя. В том числе могут быть запущены команды от имени суперпользователя. В целях безопасности команду `su` могут использовать только члены группы `root`.

При запуске команды `su` без параметров подразумевается, что пользователь хочет запустить командный интерпретатор `shell` от имени суперпользователя. При этом система просит ввести его пароль. При вводе правильного пароля запускаемый интерпретатор команд получает права и привилегии суперпользователя, которые сохраняются до завершения его работы. Для получения прав суперпользователя пользователю не требуется завершать свою сессию и вновь входить в систему.

С помощью команды `su` пользователь может исполнять отдельные команды от имени суперпользователя без запуска командного интерпретатора `shell`. Для этого используется опция `-c`. Преимущество такого способа состоит в том, что пользователь получает права и привилегии суперпользователя на строго ограниченное время, а именно, на время исполнения заданной команды. Предположим, что требуется поменять атрибуты файла от имени суперпользователя. Тогда пользователь может написать:

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

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

Кроме выполнения команд от имени суперпользователя, команда `su` позволяет выполнять команды от имени любого другого пользователя. Для этого необходимо знать пароль этого пользователя. Если пользователь вошел в систему под именем `root` и выполняет команду `su`, то знание пароля пользователя не требуется. Тогда любые команды от имени любого пользователя исполняются свободно.

Недостаток команды `su` состоит в том, что, хоть она и дает права и привилегии суперпользователя на ограниченное время, но не регламентирует команды, разрешенные конкретному пользователю на запуск от имени суперпользователя. Таким образом, если у пользователя есть права на запуск команды `su`, то он может выполнить от имени су-

перпользователя любые команды. Поэтому запуск программы `su` должен быть разрешен только доверенным пользователям. Также рекомендуется при вводе команды использовать полное путевое имя `/bin/su`, а не просто `su`.

1.2.3. sudo

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

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

Для изменения администратором файла `/etc/sudoers` можно использовать специальную команду `visudo`.

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

1.3. Механизмы разделения полномочий

Механизмы разделения полномочий между суперпользователями позволяют вводить в ОС специальных привилегированных пользователей — системных администраторов, которые выполняют строго отведенные им роли (заданное суперпользователем множество разрешенных действий). Таким образом, при дальнейшей эксплуатации системы появляется возможность исключения из системы суперпользователя (роли, для которой разрешенными являются любые, допустимые с точки зрения реализации ОС, действия).

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

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

1.3.1. Механизм привилегий

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

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

Распределение (первоначальная настройка) привилегий выполняется только суперпользователем.

1.3.2. Механизм повышения полномочий

Механизм повышения полномочий позволяет повысить полномочия пользователя на время выполнения определенной программы. Настройка механизма может быть выполнена только суперпользователем.

1.3.3. Механизм автоматической установки ACL на файлы

Механизм автоматической установки ACL на файлы облегчает задачу администрирования, при которой пользователю предоставляется доступ к тем файловым объектам, к которым необходим доступ в соответствии с его ролью. Такую настройку выполняет суперпользователь.

2. УСТАНОВКА И НАСТРОЙКА ОС

2.1. Общие положения

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

2.2. Установка с DVD-диска (запуск программы установки)

Выполнение программы установки ОС начинается с ее запуска, а затем, после выбора во входном меню конкретных параметров пользовательского интерфейса, начинается работа самой программы в интерактивном или автоматическом режимах.

В самом начале загрузки программы установки на экране монитора появляется логотип ОС, меню, переключатель «Русский»–«English» (для изменения языка меню). Меню программы установки содержит следующие пункты:

1) «Графическая установка»;
2) «Установка»;
3) «Быстрая установка»;
4) «Режим восстановления».

В нижней части экрана приведен список функциональных клавиш, подключающих дополнительные возможности программы установки:

– **[F2]** — «Язык»;
– **[F3]** — «Клавиатура»;
– **[F4]** — «Режимы»;
– **[F5]** — «Спец. возможности»;
– **[F6]** — «Параметры».

Чтобы начать установку ОС, следует выбрать пункт «Графическая установка» или «Установка» с помощью клавиш со стрелками на клавиатуре и нажать **<Enter>** для запуска программы. Произойдет переход к программе установки в графическом или в текстовом режиме, соответственно.

Пункт «Быстрая установка» запускает программу установки в режиме с минимальным количеством действий пользователя, которые сводятся к ответам на вопросы, связанные с настройкой сети, разметкой жесткого диска и установкой пароля суперпользователя. Остальные шаги программы установки будут выполняться автоматически с использованием значений параметров установки по умолчанию.

Пункт «Режим восстановления» запускает ОС в текстовом режиме непосредственно

с DVD-диска с дистрибутивом для использования при восстановлении нарушенной работоспособности уже установленной ОС.

Если необходимо добавить какие-то параметры загрузки для программы установки или ядра, то следует нажать **<F6>**, а затем **<Esc>**. После этого на экране будет показана командная строка загрузки и можно будет ввести дополнительные параметры.

Программа установки в графическом и в текстовом режимах имеет одинаковую функциональность, т. к. в обоих случаях используются одни и те же модули, т. е. отличаются они только на уровне пользовательского интерфейса. Графическая программа обеспечивает поддержку в процессе установки несколько большего числа языков, управление в ней можно осуществлять с помощью мыши, а также на одном экране может быть выведено одновременно значительно большее количество информации.

ВНИМАНИЕ! Для программы установки в графическом режиме требуется гораздо больше памяти для запуска, чем для программы установки в текстовом режиме: 128 МБ. Если памяти не достаточно, то автоматически будет осуществлен переход в текстовый режим.

Если количество установленной памяти в системе меньше 64 МБ, то графическая программа установки может совсем не запуститься, в то время как текстовая программа установки будет работать. При малом количестве памяти в системе рекомендуется использовать текстовую программу установки с консольным интерфейсом.

Особенности использования стандартных клавиш (если пользователь использует клавиатуру вместо мыши):

– чтобы раскрыть свернутый список (например, выбор стран и континентов), следует использовать клавиши **<+>** и **<->**;

– если в списке можно выбрать более одного значения (например, выбор групп пакетов), после окончания выбора следует нажать на кнопку **[Продолжить]**, а нажимая **<Пробел>** для переключения выбора, не активировать **[Продолжить]**;

– чтобы перейти на другую консоль, следует использовать клавиши **<F1>**–**<F7>**. Например, чтобы перейти на VT2 (первая оболочка командной строки для отладки), следует нажать **<левый Alt+F2>**. Сама программа установки в графическом режиме работает на VT5, так что для обратного переключения следует использовать **<левый Alt+F5>**.

2.3. Графическая установка и первичная настройка

2.3.1. Последовательность основных шагов

Действия, которые необходимо выполнить для установки ОС:

1) загрузить программу установки ОС с носителя;

2) выбрать язык установки;

3) активировать (если есть) подключение к сети Ethernet;

4) создать и смонтировать дисковые разделы, на которые будет установлена ОС;

5) выбрать и установить необходимое дополнительное программное обеспечение (ПО);

6) установить и настроить системный загрузчик `GRUB`;

7) создать учетные записи суперпользователя (`root`) и первого обычного пользователя;

8) загрузить установленную ОС в первый раз.

2.3.2. Последовательность действий

Подробное описание последовательности действий при графической установке ОС и ее первичной настройке см. в инструкции, содержащейся в каталоге `/install-doc` на DVD-диске с дистрибутивом.

3. СИСТЕМНЫЕ КОМПОНЕНТЫ

3.1. Управление устройствами

3.1.1. Типы устройств

В ОС существует два типа устройств: блочные с прямым доступом (например, жесткие диски) и символьные (например, последовательные порты), некоторые из них могут быть последовательными, а некоторые — с прямым доступом. Каждое поддерживаемое устройство представляется в ФС файлом устройства. При выполнении операций чтения или записи с подобным файлом происходит обмен данными с устройством, на которое указывает этот файл. Такой способ доступа к устройствам позволяет не использовать специальные программы (а также специальные методы программирования, такие как работа с прерываниями).

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

```
ls -l
```

на экран выводится список файлов, причем в первой колонке содержится тип файла и права доступа к нему. Например, для просмотра файла, соответствующего последовательному порту, используется следующая команда:

```
ls -l /dev/cua0
crw-rw-rw- 1 root uucp 5, 64 Nov 30 1999 /dev/cua0
```

Первый символ в первой колонке `c`, показывает тип файла, в данном случае — символьное устройство. Для обычных файлов используется символ `-` (дефис), для каталогов — `d`, для блочных устройств — `b` (подробнее см. руководство `man` к команде `ls(1)`).

Наличие большого количества файлов устройств не означает, что эти устройства на самом деле установлены. Наличие файла `/dev/sda` ни о чем не говорит и совсем не означает, что на компьютере установлен жесткий диск SCSI. Это предусмотрено для облегчения установки программ и нового оборудования (нет необходимости искать нужные параметры и создавать файлы для новых устройств).

3.1.2. Жесткие диски

При администрировании дисков могут возникнуть вопросы разделения жесткого диска на разделы, создания ФС, монтирования ФС, форматирование диска и др.

Одна из причин разделения жесткого диска — это хранение разных ОС на одном жестком диске. Другая причина — хранение пользовательских и системных файлов в разных разделах, что упрощает резервное копирование и восстановление, а также защиту системных файлов от повреждений.

Создание ФС на диске или разделе также необходимо: в ОС диск ничего не значит,

пока на нем не установлена ФС. Только после этого возможна работа с файлами.

Монтирование различных ФС для формирования единой структуры каталогов как автоматически, так и вручную (ФС, монтируемые вручную, должны быть вручную размонтированы), и вопросы буферизации дисков и работы с виртуальной памятью также необходимы при работе с дисками.

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 одного интерфейса для связи с компьютером.

Каждый жесткий диск представлен отдельным файлом. Они известны как `/dev/hda` и `/dev/hdb`, соответственно. Для SCSI-дисков используются файлы `/dev/sda` и `/dev/sdb` и т. д. Подобные обозначения применяются и для других типов дисков.

3.1.3. Разделы жесткого диска

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

3.1.3.1. Расширенные и логические разделы

В ОС swap-область чаще всего размещается в отдельном разделе (а не в основном разделе ОС) для повышения скорости обмена.

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

3.1.3.2. Разбиение жесткого диска

Программа разбиения жесткого диска на разделы называется `fdisk`.

При работе с IDE-дисками загрузочный раздел (раздел, в котором находятся файлы, используемые при загрузке и само ядро) должен полностью располагаться в пределах первых 1024 цилиндров, потому как во время загрузки работа с диском происходит через BIOS (перед переходом системы в защищенный режим), а BIOS не может оперировать с цилиндрами, номер которых больше, чем 1024.

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 обычно требуется сначала сделать резервную копию всей необходимой информации, удалить раздел, создать новый раздел, а затем восстановить всю сохраненную информацию в новом разделе.

3.1.3.3. Файлы устройств и разделы

Каждому основному и расширенному разделу соответствует отдельный файл устройства. Существует соглашение для имен подобных файлов, которое состоит в добавлении номера раздела к имени файла самого диска. Разделы 1–4 являются основными (вне зависимости от того, сколько существует основных разделов), а разделы 5–8 — логическими (вне зависимости от того, к какому основному разделу они относятся). Например, `/dev/hda1` соответствует первому основному разделу первого IDE-диска, а `/dev/sda3` — третьему расширенному разделу второго SCSI-диска.

3.1.4. Форматирование

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

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

3.1.5. Программная организация дисковых разделов в RAID и тома LVM

Программный интерфейс для организации RAID-массивов поддерживается ядром. Для организации массивов уровней `RAID 0`, `RAID 1`, `RAID 5` и их сочетаний используется менеджер массивов `mdadm`, который также по запросу администратора на создание массива конфигурирует и запускает сервисный процесс `mdmonitor`. Отличие команд в ОС от их аналогов состоит в том, что после создания массива с помощью одной команды он способен работать, запускаясь автоматически после перезагрузки системы, и не требует ручного редактирования конфигурационных файлов.

LVM, с точки зрения ядра системы, использует унифицированные механизмы VFS, не нуждаясь в специальных конфигурациях ядра. В ОС обеспечивается полнофункциональное управление томами LVM, которое осуществляется стеком команд управления (около 30 программ).

3.2. Управление ФС

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

Перед тем, как раздел или диск могут быть использованы в качестве ФС, он должен быть инициализирован, а требуемые данные перенесены на этот диск. Этот процесс называется «созданием ФС».

ФС ОС соответствует типу Ext3, обеспечивает поддержку длинных имен, символических связей, а также обеспечивает поддержку ФС ISO9660, FAT (MS-DOS), NTFS и др. Также предусмотрена возможность представления имен файлов русскими буквами.

ФС являются основой для хранения всех данных в ОС. Программы, библиотеки, системные и пользовательские файлы — все они располагаются в ФС.

ОС состоит из нескольких каталогов и множества файлов.

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

После начальной установки ФС ОС может состоять, например, из следующих частей:

- `root`:
  - `/bin` — находятся выполняемые программы (точнее, их двоичные файлы). Они необходимы для работы системы. Многие команды ОС на самом деле являются программами из этого каталога;
  - `/dev` — расположены особые файлы, называемые «файлами устройств» (device files). С их помощью осуществляется доступ ко всем физическим устройствам, установленным в системе;
  - `/tmp` — используется для хранения временных файлов, создаваемых программами в процессе своей работы. Работая с программами, создающими много больших временных файлов, лучше иметь отдельную ФС, чем простой каталог корневой ФС;
  - `/etc` — содержит конфигурационные файлы ОС. Здесь находится файл паролей `passwd`, а также список ФС, подключаемых при начальной загрузке `fstab`. В этом же каталоге хранятся сценарии загрузки (startup scripts), список узлов (hosts) с их IP-адресами и множество других данных о конфигурации;
  - `/lib` — содержатся разделяемые библиотеки, используемые многими программами во время своей работы. Применяя разделяемые библиотеки, хранящиеся в общедоступном месте, можно уменьшить размер программ за счет повторного использования одного и того же кода;
  - `/proc` — является виртуальной ФС и используется для чтения из памяти информации о системе;
  - `/sbin` — хранятся системные двоичные файлы (большинство из них используется для нужд системного администрирования);
- `/usr` — хранятся различные программы и данные, не подлежащие изменению. Каталог `/usr` и его подкаталоги необходимы для функционирования ОС, т. к. содержат наиболее важные программы. Данный каталог почти всегда является отдельной

ФС;

– /var — содержатся изменяемые файлы (такие как log-файлы и др.);

– /home — состоит из личных каталогов пользователей. Общепринято иметь здесь отдельную ФС, чтобы обеспечить пользователям достаточное пространство для размещения своих файлов. Если пользователей в системе много, возможно, придется разделить этот каталог на несколько ФС. Тогда, например, можно создать подкаталоги /home/staff и /home/admin для персонала и администрации, соответственно, установить каждый как самостоятельную ФС и уже в них создавать рабочие каталоги пользователей.

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

```
ls -a
```

3.2.1. Установка

ФС устанавливается, т. е. инициализируется, при помощи команды mkfs. Она запускает требуемую программу в зависимости от типа устанавливаемой системы. Тип ФС указывается при помощи опции -t fstype (подробнее см. руководство man).

3.2.2. Монтирование

Перед работой с ФС она должна быть смонтирована. При этом ОС выполняет некоторые действия, обеспечивающие функционирование монтируемой системы. Так как все файлы в ОС принадлежат одной структуре каталогов, то эта операция обеспечивает работу с ФС, как с каталогом уже смонтированной.

Для монтирования (подключения) ФС к дереву каталогов ОС необходимо иметь место на жестком диске. Следует убедиться также, что каталог, к которому следует подключить ФС (точка подключения), действительно существует.

Предположим, что требуется монтировать файл ISO9660 к точке подключения /mnt. Каталог /mnt должен уже существовать, иначе подключение окончится неудачей. После подключения к этому каталогу все файлы и подкаталоги ФС появятся в нем. В противном случае каталог /mnt будет пустым.

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

```
df
```

Будет видна ФС и объем свободного пространства.

3.2.2.1. mount

В ОС для подключения ФС используется команда mount. Синтаксис команды:

```
mount device mountpoint
```

где device означает физическое устройство, которое необходимо подключить, а mountpoint — точку подключения.

В целях системной безопасности использовать команду mount могут только суперпользователи.

Кроме опций, указанных выше, команда mount может иметь в командной строке еще несколько опций, приведенных в таблице 1.

Таблица 1

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

Если необходимая опция не указана, mount попытается определить ее по файлу /etc/fstab.

Распространенные формы команды mount:

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

подключает раздел жесткого диска /dev/hdb3 к каталогу /mnt.

```
mount -vat nfs
```

подключает все ФС NFS, перечисленные в файле /etc/fstab.

Если правильно подключить ФС не удается, воспользоваться командой:

```
mount -vt device mountpoint
```

чтобы узнать, что именно пытается сделать команда mount. В этом случае она выполнит все действия, кроме подключения, и даст о них подробный отчет.

3.2.2.2. fstab

Если список используемых ФС изменяется редко (а это бывает в большинстве случаев), то удобно указать ОС подключать их сразу же при загрузке и отключать при завершении работы. Эти ФС перечисляются в специальном конфигурационном файле /etc/fstab по одной в строке. Поля в строках разделяются пробелами или символами табуляции. В

таблице 2 показаны поля файла /etc/fstab.

Таблица 2

Поле	Описание
ФС	Подключаемое блочное устройство или удаленная ФС
Точка подключения	Место подключения ФС. Чтобы сделать систему невидимой в дереве каталогов (например, для файлов подкачки), используется слово none
Тип	Указывает тип подключаемой ФС
Опции подключения	Список разделенных запятыми опций для подключаемой ФС, должен содержать, по крайней мере, тип подключения. Более подробную информацию см. в руководстве man команды mount
Периодичность резервного копирования	Указывает, как часто следует выполнять резервное копирование с помощью команды dump. Если это поле отсутствует, dump считает, что ФС не нуждается в резервном копировании
Номер прохода	Задает порядок проверки целостности ФС при загрузке с помощью команды fsck. Для корневой ФС следует указывать значение 1, для остальных — 2. Если значение не указано, целостность ФС при загрузке проверяться не будет

Рекомендуется подключать ФС во время загрузки через /etc/fstab вместо команды mount.

Пример файла fstab:

```
# file system  mount point     type         options         dump  pass
/dev/sda1      /              ext3         defaults        0     1
/dev/sda2      /usr           ext3         defaults        0     0
/dev/sda3      none           swap         sw              0     0
/dev/sdb1      /win           vfat         defaults        0     0
/proc          /proc          proc         defaults        0     0
/dev/sr0       /media/cdrom0  udf,iso9660  user,noauto     0     0
/dev/fd0       /media/floppy0 auto         rw,user,noauto  0     0
```

Комментарии в файле начинаются с символа #. В примере перечислено несколько ФС. Сначала идут две обычные ФС — дисковые разделы /dev/sda1 и /dev/sda2. Они имеют тип Ext3 и подключаются к корневому каталогу / и к каталогу /usr.

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

```
mount -a
```

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

Раздел подкачки `/dev/sda3` используется ядром ОС для организации виртуальной памяти. Он должен присутствовать в файле `/etc/fstab`, чтобы система знала, где он находится. Чтобы он не появлялся в дереве каталогов, точка подключения указана как `none`. Кроме того, разделы подкачки подключаются с опцией `sw`.

Виртуальная ФС `/proc` указывает на информационное пространство процессов в памяти. Соответствующий физический раздел для нее отсутствует.

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

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

3.2.3. Размонтирование

Для размонтирования (отключения) ФС используется команда `umount`. Отключение может понадобиться для проверки и восстановления ФС с помощью команды `fsck`. Удаленные ФС отключаются в случае неполадок в сети.

Команда `umount` имеет три основные формы:

```
umount device : mountpoint
umount -a
umount -t fstype
```

`device` означает физическое устройство, которое необходимо отключить, а `mountpoint` — имя каталога точки подключения (указывать только `device` или `mountpoint`). У команды `umount` всего два параметра. Параметр `-a` отключает все ФС, а параметр `-t fstype` — только ФС указанного типа.

Команда `umount` не отключает ФС, если они используются в текущий момент.

Например, если какую-либо ФС подключить к `/mnt` и попытаться выполнить команды:

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

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

3.3. Управление пользователями

3.3.1. Работа с пользователями

Управление пользователями означает добавление, удаление пользователей и определение их привилегий.

Управление пользователями предусматривает:

– добавление имен пользователей для возможности их работы в системе;

– 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.

После того как пользователь перестает работать с системой, его учетная запись должна быть удалена (вместе со всеми его файлами).

3.3.1.1. Добавление

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

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

В этой записи поля разделены двоеточиями, а значения этих полей приведены в таблице 3.

Таблица 3

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

Для добавления пользователя применяется команда `adduser` с параметром — именем добавляемого пользователя, например:

```
adduser User1
```

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

3.3.1.2. Установка пароля

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

1) ввести команду и регистрационное имя пользователя, например:

```
passwd User1
```

и нажать клавишу **<Enter>**;

2) после появления приглашения:

```
New password:
```

ввести пароль (он не будет отображаться на экране);

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

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

Необходимо периодически изменять пароль.

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

```
anna:Zie.89&"W*:123:21:Anna_M.:/usr/anna:/bin/bash
```

Второе поле записи содержит пароль в зашифрованном виде.

Примечание. Если пользователь забыл свой пароль, то администратор системы не может напомнить его пользователю, т. к. в явном виде пароль нигде не хранится. Поэтому действия по восстановлению доступа пользователя в систему сводятся к редактированию файла `/etc/passwd`: удалению второго поля записи пользователя и назначению нового пароля с помощью команды `passwd`.

3.3.1.3. Удаление

Есть несколько степеней удаления пользователя:

– лишение пользователя возможности входа в систему;

– удаление записи;

– удаление пользователя и всех его файлов.

Лишение пользователя возможности входа в систему полезно в случае его длительного перерыва в работе. На время отсутствия пользователя можно заменить содержимое второго поля его записи звездочкой. При этом все пользовательские файлы и каталоги остаются нетронутыми, но войти в систему под его именем становится невозможно. Удаление записи о пользователе из файла `/etc/passwd` с сохранением пользовательских файлов и каталогов имеет смысл при передаче файлов другому пользователю. При этом придется изменить владельца файлов бывшего пользователя с помощью команды `chown`. Удаление пользователя таким образом производится либо путем непосредственного редактирования файла `/etc/passwd`, либо с помощью команды:

```
userdellogin_name
```

Удаление пользователя и всех его файлов — это окончательное и полное удаление пользователя из системы с помощью команды:

```
find user_home_dir -exec rm {} \;
```

Затем следует удалить рабочий каталог пользователя с помощью команды:

```
rmdir user_home_dir
```

и запись о пользователе из файла `/etc/passwd`.

3.3.2. Работа с группами

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

Информация о группах содержится в файле `/etc/group`. Пример записи из этого файла:

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

Здесь имя группы — `admin`, идентификатор — `21`, членами группы являются `user1`, `user2`, `user3`. Пользователь может быть членом нескольких групп и переходить из одной в другую в процессе работы.

3.3.2.1. Добавление

Новая группа создается путем непосредственного редактирования файла `/etc/group`, ввода необходимой информации о группе. Каждой группе присваивается свой уникальный идентификационный номер (ОС при работе учитывает номер, а не имя группы), поэтому, если присвоить двум группам один номер, для ОС получится одна и та же группа.

3.3.2.2. Удаление

Удаление группы производится путем удаления записи о ней в файле `/etc/group`. Кроме того, необходимо не забыть переназначить группу для всех файлов удаляемой груп-

пы. Это можно сделать с помощью команды:

```
find / -gid gr oup-id find user-home-dir -exec chgrp newgroup {} \;
```

3.3.3. Рабочие каталоги пользователей

Рабочие каталоги пользователей следует разместить на одном компьютере в каталоге верхнего уровня. Таким образом, они будут достаточно логично сгруппированы, что в дальнейшем облегчит администрирование системы.

Например, можно определить `/home` как каталог верхнего уровня для рабочих каталогов пользователей.

3.4. Перезагрузка и останов

Перезагрузка необходима в следующих случаях:

1) при подключении нового устройства или если работающее устройство «зависает» так, что 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 систему «заклинило» так, что невозможно зарегистрироваться и правильно поставить диагноз.

Перезагрузку можно выполнить несколькими способами:

1) дать команду `shutdown`;

2) использовать команду `reboot`;

3) послать процессу `init` сигнал `TERM`;

4) уничтожить процесс `init`.

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

Останов системы можно выполнить несколькими способами:

1) выключить питание;

2) дать команду `shutdown`;

3) использовать команду `halt`;

4) послать процессу `init` сигнал `TERM`;

5) уничтожить процесс `init`.

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

Выключение питания может привести не только к потере данных и повреждению системных файлов. Есть риск повредить жесткий диск, если он относится к числу тех, на

которых перед отключением питания необходимо установить в соответствующее положение защитный переключатель либо провести парковку головок.

3.4.1. shutdown

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

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

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

Синтаксис команды:

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

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

В таблице 4 перечислены основные опции команды `shutdown`.

Таблица 4

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

Команда `shutdown` запрещает регистрацию пользователей, посылает всем пользователям предупреждающее сообщение, затем ожидает определенное в командной строке

время и посылает всем процессам сигнал SIGTERM. Затем вызывается команда halt или reboot — в зависимости от опций командной строки.

3.4.2. halt и reboot

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

```
shutdown -h
```

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

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

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

```
shutdown -r
```

3.4.3. init

Процесс init настолько важен для работы системы, что если его уничтожить, то компьютер автоматически перезагрузится. Лучше пользоваться командами shutdown и reboot.

3.4.3.1. Посылка сигнала TERM

Когда ОС получает сигнал TERM, она обычно уничтожает все пользовательские процессы, сервисные программы, процессы getty и возвращает систему в однопользовательский режим. Это средство использует команда shutdown.

Для того чтобы послать процессу сигнал, следует с помощью команды ps узнать идентификационный номер этого процесса; init — это всегда процесс номер один. Для отправки сигнала используется команда kill, например:

```
# sync
# kill -Term 1
```

3.4.3.2. Изменение уровня выполнения

ОС с многоуровневым процессом init можно дать указание перейти на конкретный уровень выполнения с помощью команды telinit, например:

```
telinit S
```

Эта команда переводит систему в однопользовательский режим, при этом не появляются предупреждающие сообщения, которые дает команда `shutdown`.

Команда `telinit` наиболее эффективна для тестирования изменений, внесенных в файл `inittab`. При указании аргумента `-q` процесс `init` повторно читает `inittab`.

4. СИСТЕМНЫЕ СЕРВИСЫ И КОМАНДЫ

4.1. Сервисы

Сервисы — это программы, которые запускаются и останавливаются через инициализационные скрипты, расположенные в каталоге `/etc/init.d`. Многие из этих сервисов запускаются на этапе старта ОС. `/sbin/service` обеспечивает интерфейс (взаимодействие) пользователя с инициализационными скриптами. А сами эти скрипты обеспечивают интерфейс для управления сервисами, предоставляя пользователю опции для запуска, остановки, перезапуска, запроса состояния сервиса и выполнения других воздействий на сервис. К примеру, инициализационный скрипт сервиса `syslog` имеет следующие опции:

```
/sbin/service syslog
Usage: httpd {start|stop|status |restart|condrestart }
```

В ОС можно просмотреть текущее состояние всех системных служб с помощью следующей опции команды `service`:

```
/sbin/service --status-all
acpid (pid 2481) is running...
anacron (pid 2647) is running...
atd (pid 2657) is running...
auditd (pid 2189) is running...
```

Информация об уровне выполнения этих сервисов, т. е. установка того, на каком из системных уровней выполнения запускается тот или иной сервис во время загрузки системы, может быть получена или изменена с помощью команды `chkconfig`. Например, для службы системного протоколирования `syslog` установки по умолчанию выглядят следующим образом:

```
/sbin/chkconfig --list syslog
syslog   0:off   1:off   2:on   3:on   4:on   5:on   6:off
```

Сервис `syslog` автоматически запускается при переходе на уровни 2–5. Для того чтобы отключить его запуск на уровнях 3 и 4, можно воспользоваться следующей опцией команды `chkconfig`:

```
/sbin/chkconfig --levels 34 syslog off
```

4.2. Команды

К основным системным командам ОС относятся:

- `ar` — создание и работа с библиотечными архивами;
- `at` — формирование или удаление отложенного задания;
- `awk` — язык обработки строковых шаблонов;
- `batch` — планирование команд в очереди загрузки;
- `bc` — строковый калькулятор;

– chfn — управление информацией учетной записи (имя, описание);
– chsh — управление выбором командного интерпретатора (по умолчанию для учетной записи);
– cut — разбивка файла на секции, задаваемые контекстными разделителями;
– df — вывод отчета об использовании дискового пространства;
– dmesg — вывод содержимого системного буфера сообщений;
– du — вычисление количества использованного пространства элементов ФС;
– echo — вывод содержимого аргументов на стандартный вывод;
– egrep — поиск в файлах содержимого согласно регулярных выражений;
– fgrep — поиск в файлах содержимого согласно фиксированных шаблонов;
– file — определение типа файла;
– find — поиск файла по различным признакам в иерархии каталогов;
– gettext — получение строки интернационализации из каталогов перевода;
– grep — вывод строки, содержащей шаблон поиска;
– groupadd — создание новой учетной записи группы;
– groupdel — удаление учетной записи группы;
– groupmod — изменение учетной записи группы;
– groups — вывод списка групп;
– gunzip — распаковка файла;
– gzip — упаковка файла;
– hostname — вывод и задание имени хоста;
– install — копирование файла с установкой атрибутов;
– ipcrm — удаление ресурса IPC;
– ipcs — вывод характеристик ресурса IPC;
– kill — прекращение выполнения процесса;
– killall — удаление процессов по имени;
– lpr — система печати;
– ls — вывод содержимого каталога;
– lsb_release — вывод информации о загрузочном модуле;
– m4 — макропроцессор;
– md5sum — генерация и проверка MD5-сообщения;
– 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 — позволяет работать процессу после выхода из системы;
– od — вывод содержимого файла в восьмеричном и других видах;
– passwd — смена пароля учетной записи;
– patch — применение файла описания изменений к оригинальному файлу;
– pidof — вывод идентификатора процесса по его имени;
– ps — вывод информации о процессах;
– renice — изменение уровня приоритета процесса;
– sed — строковый редактор;
– sendmail — транспорт системы электронных сообщений;
– sh — командный интерпретатор;
– shutdown — команда останова системы;
– su — изменение идентификатора запускаемого процесса;
– sync — сброс системных буферов на носители;
– tar — файловый архиватор;
– umount — размонтирование ФС;
– useradd — создание новой учетной записи или обновление существующей;
– userdel — удаление учетной записи и соответствующих файлов окружения;
– usermod — модификация информации об учетной записи;
– w — показывает, кто в настоящий момент работает в системе и с чем;
– who — вывод списка пользователей системы.

4.2.1. Средства архивирования файлов

Команды tar, cpio, gzip представляют собой традиционные инструменты создания резервных копий и архивирования ФС. 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.).

4.2.1.1. tar

Команда tar может работать с рядом дисковых накопителей. Она проста в использовании, надежна, позволяет прочесть архивы в ОС.

В таблице 5 приведены некоторые наиболее часто используемые опции команды tar. Чтобы узнать о других опциях, следует воспользоваться командой man.

Таблица 5

Опция	Назначение
c	Создает архив
x	Восстанавливает файлы из архива на устройстве, заданном по умолчанию или определенном опцией f
f name	Создает (или читает) архив с name, где name — имя файла или устройства, определенного в /dev, например /dev/rmt0
Z	Сжимает или распаковывает архив
z	Сжимает или распаковывает архив с помощью gzip
M	Создает многотомный архив
t	Создает список сохраненных в архиве файлов и выводит его на консоль
v	Выводит подробную информацию о процессе

Недостатки tar:

1) может выполнять только полное архивирование данных и неспособна обновить архив. Для этого необходимо воспользоваться возможностями языка программирования оболочки;

2) сама по себе не выполняет сжатие информации. Для сжатия полученного tar-файла используется либо опция r в командной строке, либо команда сжатия, например gzip.

Примеры:

1. Копирование каталога /home на специальный раздел жесткого диска /dev/hda4

```
tar -cf /dev/hda4 /home
```

В этом примере опция f определяет создание архива на устройстве /dev/hda4.

2. Применение сжатия при архивировании

```
tar -cvfzM /dev/hda4 /home | tee home index
```

В этом примере опция v заставляет tar выводить подробную информацию, опция z говорит о том, что архив должен быть сжат, а опция M — что команда tar должна создать многотомный архив. Список скопированных файлов направляется в home index.

3. Использование команды find для поиска измененных в течение определенного времени файлов

```
find /home -mtime -1 -type f -print > bkuplst
tar -cvfzM /dev/hda4 'cat bkuplst' | tee home index
```

В приведенном примере команда find использована для создания списка измененных за последний день файлов.

Для использования выводимого командой find списка путем ввода команды tar

команда `cat bkuplst` заключена в обратные кавычки в командной строке, что говорит оболочке о том, что вывод команды должен быть помещен в командную строку.

4. Восстановление из архива

```
tar -xv /home/dave/notes.txt
```

В этом примере файл `/home/dave/notes.txt` восстанавливается из архива (следует указать полное имя восстанавливаемого файла).

5. Использование команды `tar` для создания архивов в ФС ОС, а не только на устройствах для архивирования. Таким образом, можно архивировать группу файлов с их структурой каталогов в один файл. Для этого передать имя создаваемого файла с помощью опции `f` вместо имени устройства, например:

```
tar cvf /home/backup.tar /home/dave
```

С помощью `tar` архивируется каталог с вложенными подкаталогами.

При этом создается файл `/home/backup.tar`, содержащий архив каталога `/home/dave` и всех файлов в его подкаталогах.

Обычно при использовании команды `tar` стоит делать входом верхнего уровня каталог. В таком случае файлы при восстановлении будут располагаться в подкаталоге рабочего каталога и не будут его засорять.

Предположим, в рабочем каталоге имеется подкаталог `data`, содержащий несколько сотен файлов. В распоряжении есть два основных пути создания архива этого каталога. Можно войти в подкаталог и создать там 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 подкаталог не будет создан и все сотни файлов окажутся 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 создаваться уже в нем.

Можно автоматизировать выполнение всех этих 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 есть язык сценариев оболочки, которые также могут быть запущены с помощью cron (4.2.2.3).

4.2.1.2. cpio

cpio — это команда общего назначения для копирования файлов.

Ее можно использовать с опцией -o для создания резервных архивов и с опцией -i — для восстановления файлов. Команда получает информацию от стандартного устройства ввода и посылает выводимую информацию на стандартное устройство вывода.

cpio может архивировать любой набор файлов, может архивировать специальные файлы, хранит информацию более эффективно, чем tar, пропускает сбойные сектора или блоки при восстановлении данных, ее архивы могут быть восстановлены в ОС.

Недостатком команды cpio является то, что для обновления архива следует использовать язык программирования оболочки, чтобы создать соответствующий сценарий.

В таблице 6 приведены основные опции команды cpio.

Таблица 6

Опция	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

Узнать о других опциях можно с помощью команды man.

Примеры:

1. Копирование файлов из каталога /home на устройство /dev/hda4

```
ls /home | cpio -o > /dev/hda4
```

2. Восстановление файлов с устройства /dev/hda4 и создание списка в файле bkup.indx

```
cpio -it < /dev/hda4 > bkup.indx
```

3. Использование команды find для поиска измененных за последние сутки файлов

```
find /home -mtime 1 -type f -print | cpio -08 > /dev/hda4
```

4. Восстановление файла /home/dave/notes.txt с устройства /dev/hda4

```
echo "/home/dave/notes.txt" | cpio -i < /dev/hda4
```

Для восстановления файла с помощью cpio следует указывать его полное имя.

Все эти команды могут выполняться автоматически путем их размещения в файле crontab суперпользователя. Пример записи, выполняющей резервное копирование каталога /home ежедневно в 01:30:

```
30 01 *** is /home : cpio -o > /dev/hda4
```

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.

4.2.1.3. Набор программ Bacula

Bacula представляет собой набор программ, позволяющий системному администратору управлять процессами резервного копирования и восстановления данных, а также проверять резервные копии, в том числе в гетерогенных сетях.

Bacula — это сетевая клиент-серверная система резервного копирования. Программа обладает множеством возможностей, позволяющих легко находить и восстанавливать утраченные или поврежденные файлы. Из-за своей модульной архитектуры Bacula может масштабироваться от небольших автономных систем до больших сетей, состоящих из сотен компьютеров.

Bacula состоит из следующих составных частей:

– Bacula Director service — центральная программа, координирующая все выполняемые операции (функционирует в фоне);

– Bacula Console services — программа, позволяющая администратору взаимодействовать с центральной программой;

– Bacula File services — клиентская программа, устанавливаемая на каждый обслуживаемый компьютер;

– Bacula Storage services — программа, обычно функционирующая на компьютере, к которому присоединены внешние устройства для хранения резервных копий;

– Catalog services — программа, отвечающая за индексирование и организацию базы резервных данных.

Программа Bacula обеспечивает поддержку сохранения расширенных атрибутов каталогов и файлов и, при необходимости, их последующее восстановление.

4.2.2. Планирование запуска команд

4.2.2.1. 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.

Например, если необходимо запустить команды в 1:23, следует ввести:

```
at 1:23
lpr /usr/sales/reports/.
echo "Files printed"
```

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.

В примере в 1:23 будут распечатаны все файлы каталога `/usr/sales/reports`, и пользователю будет выведено сообщение на экран монитора.

После ввода всей команды, например, на мониторе отобразится следующая запись:

```
job 756603300.a at Tues Jan 21 01:23:00 2007
```

Это означает, что указанные команды будут запущены, как и было заказано, в 1:23. Здесь приведен также идентификатор задания (`756603300.a`), который понадобится, например, если необходимо отменить задание:

```
at –d 756603300.a
```

Если список команд находится в файле, например, `getdone` и необходимо запустить все перечисленные в нем команды в 10:00, следует воспользоваться одной из двух форм команды `at`:

```
at 10:00 < getdone
```

или:

```
at 10:00 –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 также определена дата, например:

```
at 17:00 Jan 24
lp /usr/sales/reports/
echo "Files printed"
```

Задания, определяемые администратором системы, помещаются в очередь, которую ОС периодически просматривает. Администратору необязательно находиться в систе-

ме для того, чтобы at отработала задания. В данном случае команда работает в фоновом режиме.

Для того чтобы просмотреть очередь заданий, ввести:

```
at -l
```

Если предыдущие примеры были запущены, то будет выведено:

```
job 756603300.a at Sat Dec 20 01:23:00 2007 job 756604200.a at Sat Jan 24
  17:00:00 2008
```

Администратор системы видит только свои задания по команде at.

Для удаления задания из очереди следует запустить at с опцией -d и номером удаляемого задания, например:

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

В таблице 7 показаны различные 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.

4.2.2.2. batch

batch позволяет ОС самой решить, когда наступает подходящий момент для запуска задачи, т. е. когда система не очень загружена, процессы запускаются в фоновом режиме. Формат команды batch представляет собой список команд для выполнения, следующих в строках за ней; заканчивается список комбинацией клавиш **<Ctrl+D>**. Можно также поместить список команд в файл и перенаправить его на стандартный ввод команды batch. Например, для сортировки набора файлов, печати результатов и вывода сообщения ввести:

```
batch
sort /usr/sales/reports ; lp
echo "Files printed"
```

В ответ на это система выведет, например:

```
job 7789001234.b at Fri Feb 21 11:43:09 1999
```

Дата и время, приведенные в сообщении, соответствуют нажатию клавиш **<Ctrl+D>**.

4.2.2.3. cron

Для регулярного запуска команд в ОС существует команда `cron`. Администратор системы определяет время и даты, когда должна запускаться та или иная программа 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 ничего для запуска не нашлось, `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 задач для `cron`, используется команда `crontab`. Для каждого пользователя с помощью этой команды создается его собственный файл `crontab` со списком заданий, находящийся в каталоге `/usr/spool/cron/crontabs` и имеющий то же имя, что и имя пользователя.

Примечание. Пользователи, которым разрешено давать задания команде `cron`, перечислены в файле `etc/cron/.d/cron.allow`. Хотя можно создать файл заданий для команды `cron` с помощью обычного текстового редактора, нельзя просто заменить им существующий файл задания (в каталоге `/usr/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. Команда выполняется тогда, когда текущее время соответствует приведенному шаблону. Шаблон состоит из пяти частей, разделенных пробелами или символами табуляции.

Синтаксис команд в файле `crontab`:

```
минуты часы день_месяца месяц_ года день_недели задание
```

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

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

Например, шаблон 02 00 01 * * говорит о том, что команда должна быть запущена в две минуты пополуночи (поле часов нулевое) каждого первого числа любого (первая звездочка) месяца, каким бы днем недели оно не было (вторая звездочка).

В таблице 8 приведены допустимые значения полей записей crontab.

Таблица 8

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

Можно создать любое количество команд для cron, их число ничем не ограничено. Например, необходимо сортировать и отправлять пользователю pav файл /usr/sales/weekly каждый понедельник в 7:30. Соответствующая запись будет выглядеть так:

```
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 в качестве разделяющего символа запятая. 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
```

Опции командной строки crontab приведены в таблице 9.

Таблица 9

Опция	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

В любом случае crontab работает с файлом согласно регистрационному имени.

В случае команды cron как администратор системы, так и пользователи несут от-

ветственность за корректное ее использование, которое не должно, например, вызвать перегрузку системы.

4.2.3. Администрирование многопользовательской и многозадачной среды

4.2.3.1. who

Для получения списка пользователей, работающих в ОС, используется команда `who`, перечисляющая идентификаторы активных пользователей, терминалы и время входа в систему.

Для получение списка зарегистрировавшихся в системе пользователей ввести команду `who`, и на экране появится список, например:

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

Команда `who` имеет несколько опций, однако здесь рассмотрены только две из них:

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

2) `-H` — заставляет команду выводить подробную информацию о пользователях; при этом выводит строку заголовка таблицы пользователей, столбцы которой показаны в таблице 10

Таблица 10

Поле	Описание
`NAME`	Имена пользователей
`LINE`	Использованные линии и терминалы
`TIME`	Время, прошедшее после регистрации пользователя в системе
`IDLE`	Время, прошедшее со времени последней активной работы пользователя
`PID`	Идентификатор процесса входной оболочки пользователя
`COMMENT`	Комментарий (если таковые имеются в файле `/etc/inittab`)

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

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

В список включен идентификатор процесса оболочки пользователя.

Для получения информации следует вызвать команду `finger`, передав ей имя пользователя (`finger username` или `finger usernameisdomain`, если нужна информация о пользователе на другом компьютере). Для получения дополнительной информации о команде `finger` см. руководство `man`.

4.2.3.2. ps

Для получения информации о состоянии процессов используется команда `ps`. Это команда выдает информацию о запущенных процессах: какие из них выполнены, какие вызвали проблемы в системе, как долго выполняется тот или иной процесс, какие он затребовал системные ресурсы, идентификатор процесса (который будет необходим, например, для прекращения работы процесса с помощью команды `kill`) и т. д. Вся эта информация полезна как для рядового пользователя, так и для системного администратора. Запущенная без опций 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 никак не взаимодействуют ни с экраном, ни с клавиатурой, команда `ps` остается основным средством наблюдения за ними.

Команда `ps` выводит четыре основных поля информации для каждого процесса (таблица 11).

Таблица 11

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

Для получения дополнительной информации о команде `ps` следует обратиться к руководству с помощью команды `man`.

4.2.3.3. nohup

Обычно дочерний процесс прекращается после родительского. Таким образом, если запущен фоновый процесс, он будет прекращен при выходе из системы. Для того чтобы процесс продолжал выполняться даже после выхода из системы, применяется команда `nohup`. Ее следует поместить в начало командной строки, например:

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

Эта команда заставляет ОС игнорировать выход из нее и продолжать выполнение до тех пор, пока процесс не закончится сам по себе. Таким образом, будет запущен процесс, который будет выполняться длительное время, не требуя контроля администратора системы.

4.2.3.4. nice

Команда `nice` позволяет запустить другую команду с предопределенным приоритетом выполнения, предоставляя администратору системы возможность определять приоритет при выполнении своих задач. При обычном запуске все задачи имеют один и тот же приоритет, и ОС равномерно распределяет между ними процессорное время. С помощью команды `nice` можно понизить приоритет какой-либо «неспешной» задачи, предоставив другим задачам больше процессорного времени. Повысить приоритет той или иной задачи имеет право только суперпользователь.

Синтаксис команды `nice`:

```
nice –number command
```

Уровень приоритета определяется параметром `number`, при этом большее его значение означает меньший приоритет команды. Значение по умолчанию равно 10, и `number` представляет собой число, на которое он должен быть уменьшен. Например, если запущен процесс сортировки:

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

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

```
nice –5 lp mail_list &
```

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

```
nice –10 lp mail_list &
```

Примечание. В случае команды `nice` тире означает знак опции.

Только суперпользователь может повысить приоритет того или иного процесса, применяя для этого отрицательное значение аргумента. Максимально возможный приоритет — 20; присвоить его процессу суперпользователь может с помощью команды:

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

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

4.2.3.5. renice

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

```
renice –number PID
```

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

```
ps –e : grep name
```

В данной команде необходимо заменить `name` именем интересующего процесса. Команда `grep` отфильтрует только те записи, в которых будет встречаться имя нужной

команды, и можно будет узнать идентификатор ее процесса. Если необходимо изменить приоритет всех процессов пользователя или группы пользователей, в команде `renice` используется идентификатор пользователя или группы.

Рассмотрим пример использования команды `renice`, предположив, что имя пользователя — `pav`:

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

Теперь, чтобы понизить приоритет процесса `find` с идентификатором `19940`, ввести:

```
renice -5 19940
```

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

– ее можно использовать только со своими процессами;

– суперпользователь может применить ее к любому процессу;

– только суперпользователь может повысить приоритет процесса.

4.2.3.6. kill

Иногда необходимо прекратить выполнение процесса, не дожидаясь его нормального завершения. Это может произойти в следующих случаях:

1) процесс использует слишком много времени процессора и ресурсов компьютера;

2) процесс работает слишком долго, не давая ожидаемых результатов;

3) процесс производит слишком большой вывод информации на экран или в файл;

4) процесс привел к блокировке терминала или другой сессии;

5) из-за ошибки оператора или программы используются не те файлы или параметры командной строки;

6) дальнейшее выполнение процесса бесполезно.

Если процесс работает не в фоновом режиме, нажатие клавиш **<Ctrl+C>** должно прервать его выполнение, но если процесс фоновый, это не поможет. В этом случае прервать его выполнение можно только с помощью команды `kill`.

Для завершения фонового процесса команда `kill` посылает процессу сигнал, требующий от процесса завершения. Для этого используются две формы:

```
kill PID(s)
kill -signal PID(s)
```

Для завершения процесса с идентификатором `127` ввести:

```
kill 127
```

Для того чтобы завершить процессы 115, 225 и 325, ввести:

```
kill 115 225 325
```

С помощью опции -signal можно, например, заставить процесс перечитать конфигурационные файлы без прекращения работы. Список доступных сигналов можно получить с помощью команды:

```
kill -l
```

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

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

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

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

и завершить работу оболочки на блокированном терминале.

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

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

```
kill PID
```

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

```
kill -9 PID
```

Команда:

```
kill -9
```

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

Для завершения всех фоновых процессов ввести:

```
kill 0
```

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

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

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

4.3. Графические утилиты

В состав рабочего стола Fly входит большое количество графических утилит, которые могут быть использованы для администрирования системы. Большинство из этих утилит представляет собой графические оболочки над соответствующими текстовыми утилитами командной строки.

Описание утилит см. в электронной справке.

5. БАЗОВЫЕ СЕТЕВЫЕ СЛУЖБЫ

5.1. Сеть 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 программы, установлено, что порты с номерами до 1024 могут использоваться только суперпользователем.

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 надежности, которое расставляет маршруты по приоритетам, если таблица содержит противоречивую информацию. Для направления пакета по конкретному адресу подбирается наиболее подходящий маршрут. Если нет ни такого маршрута, ни маршрута по умолчанию, то отправителю возвращается

ошибка: «network unreachable» (сеть недоступна).

Таблицу маршрутизации компьютера можно вывести на экран 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 и 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 интерфейсов, у него будет несколько сетевых адресов.

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

5.1.4.3. Настройка сетевых интерфейсов

Программа `ifconfig` используется для включения и выключения сетевого интерфейса, задания IP-адреса, широковещательного адреса и связанной с ним маски подсети, а также для установки других опций и 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 опция ...
```

Пример

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

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

```
netstat-i
```

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

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
```

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 аргумент `число пакетов`. Чтобы прекратить работу команды `ping`, необходимо нажать **<Ctrl+C>**.

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

Команда `netstat -i` показывает состояние сетевых интерфейсов.

Команда `netstat -r` выдает таблицу маршрутизации ядра.

Команда `netstat -s` выдает содержимое счетчиков, разбросанных по сетевым программам.

5.1.5.3. arp

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

Команда `arp -a` распечатывает содержимое таблицы соответствий.

5.2. Служба передачи файлов FTP

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

5.2.1. Клиентская часть

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`.

5.2.2. Сервер VSFTPD

В ОС программный пакет `vsftpd` устанавливается командой:

```
apt-get install vsftpd
```

Пакет также может быть установлен в процессе установки ОС. Для этого следует в окне программы установки «Выбор программного обеспечения» отметить группу пакетов «Сетевые сервисы».

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

Сама команда располагается в каталоге `/usr/sbin/vsftpd`.

5.2.2.1. Конфигурационный файл

После установки сервера `vsftpd` он сразу готов к работе с опциями по умолчанию. Если для работы сервера необходимы другие значения опций, следует отредактировать конфигурационный файл `/etc/vsftpd.conf`.

В файле `vsftpd.conf` представлены три вида опций:

– `BOOLEAN` — опции, которые могут содержать значения: `YES`, `NO`;

– `NUMERIC` — опции, содержащие различные цифровые значения (к примеру, время в секундах или номер порта соединения);

– `STRING` — опции, содержащие текстовую строку (к примеру, путь к каталогу на диске).

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

Не все опции следует указывать напрямую, иначе конфигурационный файл может вырасти до очень больших размеров. В большинстве случаев необходимо записать в файл всего лишь несколько строк, а для остальных настроек использовать значения по умолчанию.

Многие настройки зависят от других опций. Если те опции, от которых они зависят, отключены, то и данные настройки не будут работать. Некоторые опции являются взаимоисключающими и, следовательно, не будут работать в паре с другими такими включенными опциями.

5.3. Служба динамической конфигурации узла DHCP

На компьютере, выполняющем роль сервера динамической конфигурации сети, должна быть установлена служба `dhcpd`. Настройки этой службы хранятся в файле `/etc/dhcpd.conf`. Файл настройки содержит инструкции, которые определяют, какие подсети и узлы обслуживает сервер и какую информацию настройки он им предоставляет.

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

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

Описание некоторых параметров настройки сервера dhcpd, содержащихся в файле dhcpd.conf, приводится 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 так же, 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 `dhcp.conf`, приведены в таблице 13.

Таблица 13

Оператор	Описание
`group {[parameters] [options]}`	`group` группирует операторы `shared-network`, `subnet`, `host` и другие операторы `group`. Позволяет применять наборы параметров и опций 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 и опции, 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 опции, следующие за ключевым словом `option` в файле `dhcp.conf`, приведены в таблице 14.

Таблица 14

Опция	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 опция `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

Запуск службы dhcpd можно осуществить с помощью команды:

```
service dhcpd start
```

или включить одним из известных способов в список служб, запускаемых при старте системы.

5.4. Служба сетевого доступа к файловым системам NFS

Служба сетевого доступа к файловым системам NFS позволяет использовать ФС удаленных серверов и компьютеров. Доступ к ФС удаленных компьютеров обеспечивается с помощью нескольких программ на сторонах сервера и клиента.

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.

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 указанных клиентских компьютеров и какими правами доступа обладают клиентские компьютеры при выполнении операций на сервере. 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.

В составе ОС используется модифицированная версия службы NFS4 с поддержкой расширенных атрибутов (в том числе, мандатных меток), а также аутентификации и защиты данных с помощью механизма Kerberos.

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

понимать запись:

```
имя_компьютера: каталог
```

где имя_компьютера — имя сервера NFS, каталог — экспортированный каталог сервера службы NFS. Для удаленных ФС, которые являются частью постоянной конфигурации клиента, записи о монтируемых ФС службы NFS должны быть перечислены в файле /etc/fstab для автоматического монтирования во время начальной загрузки клиентского компьютера.

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

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

5.5. Служба доменных имен DNS

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

– прямой запрос — запрос на преобразование имени компьютера в IP-адрес;

– обратный запрос — запрос на преобразование IP-адреса в имя компьютера.

Настройки службы named хранятся в файлах каталога /etc/bind/ и, в первую очередь, в файле /etc/bind/named.conf.

5.6. Фильтр сетевых пакетов

При помощи фильтра сетевых пакетов можно осуществлять контроль сетевого трафика, проходящего через данный компьютер.

Фильтрацию пакетов выполняет фильтр пакетов iptables. Данный фильтр позволяет выполнять следующие задачи:

1) фильтрацию пакетов — это механизм, который, основываясь на некоторых правилах, разрешает или запрещает передачу информации, проходящей через него, с целью ограждения некоторой подсети от внешнего доступа, или, наоборот, для недопущения выхода наружу. Фильтр пакетов может определять правомерность передачи информации на основе только заголовков IP-пакетов, а может анализировать и их содержимое, т. е. использовать данные протоколов более высокого уровня;

2) трансляцию сетевых адресов (т. н. «маскарадинг») — это подмена некоторых параметров в заголовках IP-пакетов. Используется для сокрытия реальных IP-адресов компьютеров защищаемой ЛВС, а также для организации доступа из ЛВС с компьютерами, не имеющими реальных IP-адресов, к глобальной сети;

3) прозрачное проксирование — это переадресация пакетов на другой порт ком-

пьютера. Обычно используется для того, чтобы заставить пользователей из ЛВС пользоваться proxy-сервером маршрутизатора без дополнительного конфигурирования их клиентских программ.

Настройка рассмотренных механизмов (фильтрация пакетов, трансляция сетевых адресов и прозрачное проксирование) выполняется командой `iptables`.

5.6.1. Формирование правил

Каждое правило — это строка, содержащая в себе критерии, определяющие, подпадает ли пакет под заданное правило, и действие, которое необходимо выполнить в случае выполнения критерия.

Правила записываются следующим образом:

```
iptables [-t table] command [match] [target/jump]
```

Если в правило не включается спецификатор `[-t table]`, то по умолчанию предполагается использование таблицы `filter`, если же предполагается использование другой таблицы, то это требуется указать явно. Спецификатор таблицы также можно указывать в любом месте строки правила, однако более или менее стандартным считается указание таблицы в начале правила.

Далее, непосредственно за именем таблицы, должна стоять команда. Если спецификатора таблицы нет, то команда всегда должна стоять первой. Команда определяет действие `iptables`, например вставить, добавить в конец цепочки или удалить правило и т. п.

Раздел `matches` задает критерии проверки, по которым определяется, подпадает ли пакет под действие этого правила или нет. Здесь можно указать самые разные критерии — и IP-адрес источника пакета или сети, и сетевой интерфейс, и т. д.

`target` указывает, какое действие должно быть выполнено при условии выполнения критериев в правиле. Здесь можно заставить ядро передать пакет в другую цепочку правил, «сбросить» пакет и забыть про него, выдать на источник сообщение об ошибке и т. п.

5.6.1.1. Порядок прохождения таблиц и цепочек

Когда пакет приходит на сетевой фильтр, то он сначала попадает на сетевое устройство, перехватывается соответствующим драйвером и далее передается в ядро. Затем пакет проходит несколько таблиц и после передается либо локальному приложению, либо переправляется на другой компьютер. Порядок следования пакета приводится в таблице 15.

Таблица 15

Шаг	Таблица	Цепочка	Описание
1	=	=	Кабель
2	=	=	Сетевой интерфейс (например, eth0)
3	mangle	PREROUTING	Используется для внесения изменений в заголовок пакета, например для изменения битов TOS и пр.
4	nat	PREROUTING	Используется для трансляции сетевых адресов DNAT. SNAT выполняется позднее, в другой цепочке. Любого рода фильтрация в этой цепочке может производиться только в исключительных случаях
5	=	=	Принятие решения о дальнейшей маршрутизации, т. е. в этой точке решается, куда пойдет пакет — локальному приложению или на другой узел сети
6	filter	FORWARD	Попадают только те пакеты, которые идут на другой компьютер. Вся фильтрация транзитного трафика должна выполняться здесь. Через эту цепочку проходит трафик в обоих направлениях, поэтому обязательно учитывать это обстоятельство при написании правил фильтрации
7	nat	POSTROUTING	Предназначена в первую очередь для SNAT. Не использовать для фильтрации без особой необходимости. Здесь же выполняется и маскарадинг
8	=	=	Выходной сетевой интерфейс (например, eth1)
9	=	=	Кабель

Пакет проходит несколько этапов, прежде чем он будет передан далее. На каждом из них пакет может быть остановлен. Цепочку FORWARD проходят все пакеты, которые движутся через сетевой фильтр. В таблице 16 представлен порядок движения пакета, предназначенного локальному процессу/приложению.

Таблица 16

Шаг	Таблица	Цепочка	Описание
1	=	=	Кабель
2	=	=	Входной сетевой интерфейс (например, eth0)
3	mangle	PREROUTING	Обычно используется для внесения изменений в заголовок пакета, например для установки битов TOS и пр.
4	nat	PREROUTING	Преобразование адресов DNAT. Фильтрация пакетов здесь допускается только в исключительных случаях
5	=	=	Принятие решения о маршрутизации
6	filter	INPUT	Фильтрация входящего трафика. Все входящие пакеты, адресованные локальному приложению, проходят через эту цепочку, независимо от того, с какого интерфейса они поступили
7	=	=	Локальный процесс/приложение

Важно помнить, что пакеты идут через цепочку INPUT, а не через FORWARD. В таблице 17 представлен порядок движения пакетов, созданных локальными процессами.

Таблица 17

Шаг	Таблица	Цепочка	Описание
1	=	=	Локальный процесс
2	mangle	OUTPUT	Внесение изменений в заголовок пакета. Фильтрация, выполняемая в этой цепочке, может иметь негативные последствия
3	filter		
4	=	=	Принятие решения о маршрутизации. Здесь решается — куда пойдет пакет дальше
5	nat	POSTROUTING	Здесь выполняется SNAT. Не следует в этой цепочке производить фильтрацию пакетов во избежание нежелательных побочных эффектов. Однако и здесь можно останавливать пакеты, применяя политику по умолчанию — DROP
6	=	=	Сетевой интерфейс (например, eth0)
7	=	=	Кабель

mangle

Эта таблица предназначена для внесения изменений в заголовки пакетов, т. е. в этой таблице можно устанавливать биты TOS и т. д.

В этой таблице не следует производить любого рода фильтрацию, маскировку или преобразование адресов (DNAT, SNAT).

В этой таблице допускается выполнять действия, приведенные в таблице 18.

Таблица 18

Действие	Описание
TOS	Выполняет установку битов поля TOS в пакете. Это поле используется для назначения сетевой политики обслуживания пакета, т. е. задает желаемый вариант маршрутизации
TTL	Используется для установки значения поля TTL пакета.
MARK	Устанавливает специальную метку на пакет, которая затем может быть проверена другими правилами в iptables или другими программами, например iproute2. С помощью меток можно управлять маршрутизацией пакетов, ограничивать трафик и т. п.

Таблица имеет две цепочки:

– PREROUTING — используется для внесения изменений на входе в сетевой фильтр перед принятием решения о маршрутизации;

– OUTPUT — для внесения изменений в пакеты, поступающие от приложений внутри сетевой фильтр. Таблица mangle не должна использоваться для преобразования сетевых адресов или маскарадинга, поскольку для этих целей имеется таблица

nat.

nat

Эта таблица используется для выполнения преобразований сетевых адресов NAT. Только первый пакет из потока проходит через цепочки этой таблицы. Трансляция адресов или маскировка применяются ко всем последующим пакетам в потоке автоматически. Для этой таблицы характерны действия (таблица 19).

Таблица 19

Действие	Описание
DNAT	Производит преобразование адресов назначения в заголовках пакетов. Другими словами, этим действием перенаправляются пакеты на другие адреса, отличные от указанных в заголовках пакетов
SNAT	Используется для изменения исходных адресов пакетов. С помощью этого действия можно скрыть структуру локальной сети
MASQUERADE	Применяется в тех же целях, что и SNAT, но в отличие от последней дает более сильную нагрузку на систему. Происходит это потому, что каждый раз, когда требуется выполнение этого действия, производится запрос IP-адреса для указанного в действии сетевого интерфейса, в то время как для SNAT IP-адрес указывается непосредственно. Однако благодаря такому отличию, MASQUERADE может работать в случаях с динамическим IP-адресом

Таблица имеет две цепочки:

– PREROUTING — используется для внесения изменений в пакеты на входе в сетевой фильтр;

– OUTPUT — используется для преобразования пакетов, созданных приложениями внутри сетевой фильтр, перед принятием решения о маршрутизации.

filter

В этой таблице содержатся наборы правил для выполнения фильтрации пакетов. Пакеты могут пропускаться далее либо отвергаться в зависимости от их содержимого.

В таблице filter можно выполнить DROP, LOG, ACCEPT или REJECT без каких-либо сложностей, как в других таблицах. Имеется три встроенных цепочки:

– FORWARD — используется для фильтрации пакетов, идущих транзитом через сетевой фильтр;

– INPUT — проходят пакеты, которые предназначены локальным приложениям (сетевому фильтру);

– OUTPUT — используется для фильтрации исходящих пакетов, сгенерированных приложениями на самом сетевом фильтре.

5.6.1.2. Механизм трассировки соединений

Механизм трассировки соединений является частью сетевого фильтра iptables и устроен так, чтобы netfilter мог получить информацию о состоянии конкретного соединения. Наличие этого механизма позволяет создавать более надежные наборы правил.

В пределах iptables соединение может иметь одно из четырех базовых состояний: NEW, ESTABLISHED, RELATED и INVALID. Для управления пакетами на основе их состояния используется критерий --state. Трассировщик определяет четыре основных состояния каждого TCP- или UDP-пакета и некоторые дополнительные характеристики. Для TCP- и UDP-пакетов — это IP-адреса отправителя и получателя, порты отправителя и получателя.

Трассировка производится в цепочке PREROUTING. Это означает, что iptables производит все вычисления, связанные с определением состояния, в пределах этой цепочки. Когда отправляется инициирующий пакет в потоке, то ему присваивается состояние NEW, а когда возвращается пакет ответа, то состояние соединения изменяется на ESTABLISHED и т. д.

Таблица трассировки

Таблицу трассировщика можно найти в файле /proc/net/ip_conntrack. Здесь содержится список всех активных соединений. Если модуль ip_conntrack загружен, то команда cat /proc/net/ip_conntrack должна вывести:

```
tcp 6 117 SYN_SENT src=192.168.1.6 dst=192.168.1.9 sport=32775 dport=22
  [UNREPLIED] src=192.168.1.9 dst=192.168.1.6 sport=22 dport=32775 use=2
```

В этом примере содержится вся информация, которая известна трассировщику по конкретному соединению. Первое, что можно увидеть — это название протокола, в данном случае — tcp. Далее следует некоторое число в обычном десятичном представлении. После него следует число, определяющее «время жизни» (т. е. количество секунд, через которое информация о соединении будет удалена из таблицы) записи в таблице. В приведенном примере запись в таблице будет храниться еще 117 с, если через это соединение более не проследует ни одного пакета, в противном случае это значение будет установлено в значение по умолчанию для заданного состояния. Это число уменьшается на одну секунду.

Далее следует фактическое состояние соединения. В примере это состояние имеет значение SYN_SENT. Внутреннее представление состояния несколько отличается от внешнего. Значение SYN_SENT говорит о том, что через данное соединение проследовал единственный пакет TCP SYN. Далее расположены адреса отправителя и получателя, порты отправителя и получателя. Здесь же видно ключевое слово, которое сообщает о том, что ответного трафика через это соединение еще не было.

Приводится дополнительная информация по ожидаемому пакету, это IP-адреса отправителя/получателя (те же самые, только поменявшиеся местами, т. к. ожидается ответный пакет), то же касается и портов.

После получения пакетом ответа трассировщик снимет флаг `[unreplied]` и заменит его флагом `[assured]`. Этот флаг сообщает, что соединение установлено уверенно, и эта запись не будет стерта по достижении максимально возможного количества трассируемых соединений. Максимальное количество записей, которое может содержаться в таблице, зависит от значения по умолчанию, которое может быть установлено вызовом функции `ipsysctl`. Для объема ОЗУ 128 МБ — это значение соответствует 8192 записям, для 256 МБ — 16376. Можно посмотреть и изменить это значение через:

```
\underline{}\RUSBCourier{/proc/sys/net/ipv4/ip_conntrack_max}
```

Состояния

Сетевые пакеты могут иметь несколько различных состояний в пределах ядра в зависимости от типа протокола. Однако вне ядра имеется только четыре состояния, как было сказано выше. Параметры, описывающие состояние пакета, используются в критерии `--state`. Допустимыми являются: `NEW`, `ESTABLISHED`, `RELATED` и `INVALID`.

В таблице 20 подробно рассмотрены каждое из возможных состояний и приведены необходимые комментарии.

Таблица 20

Состояние	Описание
`NEW`	Признак `NEW` сообщает о том, что пакет является первым для данного соединения. Это означает, что это первый пакет в данном соединении, который увидел модуль трассировщика
`ESTABLISHED`	Признак `ESTABLISHED` говорит о том, что это не первый пакет в соединении. Для перехода в состояние `ESTABLISHED` необходимо, чтобы один компьютер передал пакет и получил на него ответ от другого компьютера. После получения ответа признак соединения `NEW` будет заменен на `ESTABLISHED`
`RELATED`	Соединение получает статус `RELATED`, если оно связано с другим соединением, имеющим признак `ESTABLISHED`, т. е. соединение инициировано из уже установленного соединения, имеющего признак `ESTABLISHED`
`INVALID`	Признак `INVALID` говорит о том, что пакет не может быть идентифицирован и поэтому не может иметь определенного статуса. Это может происходить по разным причинам, например, при нехватке памяти или при получении ICMP-сообщения, которое не соответствует какому-либо известному соединению. Наилучшим вариантом было бы применение действия `DROP` к таким пакетам

Таблицы

Опция `-t` указывает на используемую таблицу. По умолчанию используется таблица `filter`.

Команды

В таблице 21 приводится список команд, которые используются в `iptables`, и правила их использования. Посредством команд `iptables` узнает, что необходимо выполнить. Обычно предполагается одно из двух действий — это добавление нового правила в цепочку или удаление существующего правила из той или иной таблицы.

Таблица 21

Команда	Использование
`-A, --append`	Добавляет новое правило в конец заданной цепочки. Пример `iptables -A INPUT ...`
`-D, --delete`	Удаляет правило из цепочки. Команда имеет два формата записи, первый — когда задается критерий сравнения с опцией `-D`, второй — порядковый номер правила. Если задается критерий сравнения, то удаляется правило, которое имеет в себе этот критерий, если задается номер правила, то будет удалено правило с заданным номером. Счет правил в цепочках начинается с единицы. Пример `iptables -D INPUT --dport 80 -j DROP, iptables -D INPUT 1`
`-E, --rename-chain`	Выполняет переименование пользовательской цепочки. В примере цепочка `allowed` будет переименована в цепочку `disallowed`. Эти переименования не изменяют порядок работы. Пример `iptables -E allowed disallowed` Команда должна быть указана всегда
`-F, --flush`	Сброс (удаление) всех правил из заданной цепочки (таблицы). Если имя цепочки и таблицы не указывается, то удаляются все правила во всех цепочках. Пример `iptables -F INPUT`
`-I, --insert`	Вставляет новое правило в цепочку. Число, следующее за именем цепочки, указывает номер правила, перед которым следует вставить новое правило. В примере выше указывается, что данное правило должно быть первым в цепочке `INPUT`. Пример `iptables -I INPUT 1 --dport 80 -j ACCEPT`
`-L, --list`	Вывод списка правил в заданной цепочке. В нижеприведенном примере предполагается вывод правил из цепочки `INPUT`. Если имя цепочки не указывается, то выводится список правил для всех цепочек. Формат вывода зависит от наличия дополнительных ключей в команде, например `-n`, `-v` и пр. Пример `iptables -L INPUT`

Окончание таблицы 21

Команда	Использование
`-N, --new-chain`	Создается новая цепочка с заданным именем в заданной таблице. В нижеприведенном примере создается новая цепочка с именем `allowed`. Имя цепочки должно быть уникальным и не должно совпадать с зарезервированными именами цепочек и действий (`DROP`, `REJECT` и т. п.). Пример `iptables -N allowed`
`-P, --policy`	Определяет политику по умолчанию для заданной цепочки. Политика по умолчанию определяет действие, применяемое к пакетам, не попавшим под действие ни одного из правил в цепочке. В качестве политики по умолчанию допускается использовать `DROP`, `ACCEPT` и `REJECT`. Пример `iptables -P INPUT DROP`
`-R, --replace`	Данная команда заменяет одно правило другим. В основном она используется во время отладки новых правил. Пример `iptables -R INPUT 1 -s 192.168.0.1 -j`
`-X, --delete-chain`	Удаление заданной цепочки из заданной таблицы. Удаляемая цепочка не должна иметь правил и не должно быть ссылок из других цепочек на удаляемую цепочку. Если имя цепочки не указано, то будут удалены все цепочки, определенные командой `-N` в заданной таблице. Примеры: 1. `iptables -X allowed` 2. `iptables -P INPUT DROP`
`-Z, --zero`	Обнуление всех счетчиков в заданной цепочке. Если имя цепочки не указывается, то подразумеваются все цепочки. При использовании ключа `-v` совместно с командой `-L` на вывод будут поданы и состояния счетчиков пакетов, попавших под действие каждого правила. Допускается совместное использование команд `-L` и `-Z`. В этом случае будет выдан сначала список правил со счетчиками, а затем произойдет обнуление счетчиков

Ключи

Некоторые команды могут использоваться совместно с дополнительными ключами (таблица 22).

Таблица 22

Ключ	Описание
`c, --set-counters`	Используется вместе с командами `--insert`, `--append` и `--replace` при создании нового правила для установки счетчиков пакетов и байт в заданное значение. Например, ключ `--set-counters 20 4000` установит счетчик пакетов = 20, а счетчик байт = 4000

Окончание таблицы 22

Ключ	Описание
`--line-numbers`	Используется вместе с командой `--list`, включает режим вывода номеров строк при отображении списка правил командой `--list`. Номер строки соответствует позиции правила в цепочке
`--modprobe`	Может использоваться с любой командой, определяет команду загрузки модуля ядра. Данный ключ используется в случае, если команда `modprobe` находится вне пути поиска
`n, --numeric`	Используется вместе с командой `--list`. Заставляет `iptables` выводить IP-адреса и номера портов в числовом виде, предотвращая попытки преобразовать их в символические имена
`-v, --verbose`	Используется вместе с командами `--list`, `--append`, `--insert`, `--delete` и `--replace` для повышения информативности вывода. В случае использования с командой `--list` в вывод этой команды включаются так же имя интерфейса, счетчики пакетов и байт для каждого правила. Формат вывода счетчиков предполагает вывод, кроме цифр числа, еще и символьные множители `K` (x1000), `M` (x1,000,000) и `G` (x1,000,000,000). Для того чтобы заставить команду `--list` выводить полное число (без употребления множителей), требуется применять ключ `-x`, который описан ниже. При использовании с другими командами на вывод будет выдан подробный отчет о произведенной операции
`-x, --exact`	Используется вместе с командой `--list`. Для всех чисел в выходных данных выводятся их точные значения без округления и без применения множителей `K`, `M`, `G`. Важно то, что данный ключ используется только с командой `--list` и не применяется с другими командами

5.6.1.3. Критерии выделения пакетов

Выделяются следующие критерии:

1) общие — критерии, которые допустимо употреблять в любых правилах. Они не зависят от типа протокола и не требуют подгрузки модулей расширения. В эту группу добавлен критерий `--protocol`, несмотря на то, что он используется в некоторых специфичных от протокола расширениях. Например, при использовании TCP-критерия необходимо использовать и критерий `--protocol`, которому в качестве дополнительного ключа передается название протокола — TCP. Однако `--protocol` сам по себе является критерием, который используется для указания типа протокола;

2) неявные — это критерии, которые подгружаются неявно и становятся доступны, например, при указании критерия `--protocol`. Существует три автоматически подгружаемых расширения: TCP-, UDP- и ICMP-критерии. Загрузка этих расширений может производиться и явным образом с помощью ключа `-m`, `--match`, например:
`-m tcp`

3) перед использованием вышеописанных расширений они должны быть загружены явно, с помощью ключа `-m` или `--match`. Так, например, если использовать

критерии `--state`, то следует явно указать это в строке правила: `-m state` левее используемого критерия. Все отличие между явными и неявными критериями заключается только в том, что первые необходимо подгружать явно, а вторые подгружаются автоматически.

5.6.1.4. Действия и переходы

Действия и переходы сообщают правилу, что необходимо выполнить, если пакет соответствует заданному критерию. Чаще всего употребляются действия `ACCEPT` и `DROP`.

Описание переходов в правилах выглядит точно так же, как и описание действий, т. е. ставится ключ `-j` и указывается название цепочки правил, на которую выполняется переход. На переходы накладывается ряд ограничений, первое — цепочка, на которую выполняется переход, должна находиться в той же таблице, что и цепочка, из которой этот переход выполняется; второе — цепочка, являющаяся целью перехода должна быть создана до того, как на нее будут выполняться переходы.

Например, создать цепочку `tcp_packets` в таблице `filter` с помощью команды:

```
iptables -N tcp_packets
```

Теперь можно выполнять переходы на эту цепочку подобно:

```
iptables -A INPUT -p tcp -j tcp_packets
```

т. е., встретив пакет протокола TCP, `iptables` произведет переход на цепочку `tcp_packets` и продолжит движение пакета по этой цепочке. Если пакет достиг конца цепочки, то он будет возвращен в вызывающую цепочку (в примере — это цепочка `INPUT`) и движение пакета продолжится с правила, следующего за правилом, вызвавшем переход. Если к пакету во вложенной цепочке будет применено действие `ACCEPT`, то автоматически пакет будет считаться принятым и в вызывающей цепочке и уже не будет продолжать движение по вызывающим цепочкам. Однако пакет пойдет по другим цепочкам в других таблицах.

Действие — это предопределенная команда, описывающая действие, которое необходимо выполнить, если пакет совпал с заданным критерием. Например, можно применить действие `DROP` или `ACCEPT` к пакету. В результате выполнения одних действий пакет прекращает свое прохождение по цепочке, например `DROP` и `ACCEPT`; в результате других, после выполнения неких операций, продолжает проверку, например `LOG`; в результате работы третьих — даже видоизменяется, например `DNAT` и `SNAT`, `TTL` и `TOS`, но так же продолжает продвижение по цепочке.

ACCEPT

Если над пакетом выполняется действие `ACCEPT`, то пакет прекращает движение по цепочке (и всем вызвавшим цепочкам, если текущая цепочка была вложенной) и считается принятым, тем не менее, пакет продолжит движение по цепочкам в других таблицах и мо-

жет быть отвергнут там. Действие задается с помощью ключа `-j ACCEPT`. Дополнительных ключей не имеет.

DNAT

`DNAT` используется для преобразования адреса места назначения в IP-заголовке пакета. Если пакет подпадает под критерий правила, выполняющего `DNAT`, то этот пакет и все последующие пакеты из этого же потока будут подвергнуты преобразованию адреса назначения и переданы на требуемое устройство, компьютер или сеть.

Может выполняться только в цепочках `PREROUTING` и `OUTPUT` таблицы `nat` и во вложенных подцепочках.

Ключ для действия `DNAT` — `--to-destination`.

Пример

```
iptables -t nat -A PREROUTING -p tcp -d 15.45.23.67
--dport 80 -j DNAT --to-destination 192.168.1.1-192.168.1.10
```

Этот ключ указывает, какой IP-адрес должен быть подставлен в качестве адреса места назначения. В вышеприведенном примере во всех пакетах, пришедших на адрес `.45.23.67`, адрес назначения будет изменен на один из диапазона от `192.168.1.1` до `192.168.1.10`. Все пакеты из одного потока будут направляться на один и тот же адрес, а для каждого нового потока будет выбираться один из адресов в указанном диапазоне случайным образом. Можно также определить единственный IP-адрес. Можно дополнительно указать порт или диапазон портов, на который (которые) будет перенаправлен трафик. Для этого после IP-адреса через двоеточие указать порт, например:

```
--to-destination 192.168.1.1:80
```

а указание диапазона портов выглядит так:

```
--to-destination 192.168.1.1:80-100
```

Синтаксис действий `DNAT` и `SNAT` во многом схож. Указание портов допускается только при работе с протоколом `TCP` или `UDP`, при наличии опции `--protocol` в критерии.

DROP

`DROP` сбрасывает пакет и `iptables` забывает о его существовании. Сброшенные пакеты прекращают свое движение полностью, т. е. они не передаются в другие таблицы, как это происходит в случае с действием `ACCEPT`. Следует помнить, что данное действие может иметь негативные последствия, поскольку может оставлять незакрытые сокеты как на стороне сервера, так и на стороне клиента, наилучшим способом защиты будет использование действия `REJECT` особенно при защите от сканирования портов.

LOG

`LOG` служит для журналирования отдельных пакетов и событий. В журнал могут заноситься заголовки IP-пакетов и другая интересующая информация. Информация из жур-

нала может быть прочитана с помощью dmesg или syslogd, либо с помощью других команд.

Ключи действия LOG приведены в таблице 23.

Таблица 23

Ключ	Описание
--log-level	Используется для задания уровня журналирования. Можно задать следующие уровни: debug, info, notice, warning, warn, err, error, crit, alert, emerg и panic. Ключевое слово error означает то же самое, что и err, warn — warning и panic — emerg. Приоритет определяет различия в том, как будут заноситься сообщения в журнал. Все сообщения заносятся в журнал средствами ядра. Если установить строку kern.=info /var/log/iptables в файле syslog.conf, то все сообщения из iptables, использующие уровень info, будут заноситься в файл /var/log/iptables. Однако в этот файл попадут и другие сообщения, поступающие из других подсистем, которые используют уровень info. Пример iptables -A FORWARD -p tcp -j LOG --log-level debug
--log-prefix	Задает префикс, который будет стоять перед всеми сообщениями iptables. Сообщения со специфичным префиксом затем легко можно найти, к примеру, с помощью grep. Префикс может содержать до 29 символов, включая пробелы. Пример iptables -A INPUT -p tcp -j LOG --log-prefix "INPUT packets"
--log-tcp-sequence	Позволяет заносить в журнал номер TCP Sequence-пакета. Номер TCP Sequence идентифицирует каждый пакет в потоке и определяет порядок сборки потока. Этот ключ потенциально опасен для безопасности системы, если системный журнал разрешает доступ «на чтение» всем пользователям. Как и любой другой журнал, содержащий сообщения от iptables. Пример iptables -A INPUT -p tcp -j LOG --log-tcp-sequence
--log-tcp-options	Позволяет заносить в системный журнал различные сведения из заголовка TCP-пакета. Такая возможность может быть полезна при отладке. Этот ключ не имеет дополнительных параметров. Пример iptables -A FORWARD -p tcp -j LOG --log-tcp-options
--log-ip-options	Позволяет заносить в системный журнал различные сведения из заголовка IP-пакета. Во многом схож с ключом --log-tcp-options, но работает только с IP-заголовком. Пример iptables -A FORWARD -p tcp -j LOG --log-ip-options

MARK

MARK используется для установки меток для определенных пакетов. Это действие может выполняться только в пределах таблицы mangle. Установка меток обычно используется для нужд маршрутизации пакетов по различным маршрутам, для ограничения трафика и т. п. Метка пакета существует только в период времени, пока пакет не покинул брандмауэр, т. е. метка не передается по сети. Если необходимо как-то пометить пакеты, чтобы использовать маркировку на другом компьютере, то можно манипулировать битами поля TOS.

Ключ для действия MARK — --set- — устанавливает метку на пакет. После ключа --set-mark должно следовать целое число.

Пример

```
iptables -t mangle -A PREROUTING -p tcp --dport 22 -j MARK --set-mark 2
```

MASQUERADE

Маскарадинг подразумевает получение IP-адреса от заданного сетевого интерфейса, вместо прямого его указания, как это делается с помощью ключа --to-source в действии SNAT.

Действие MASQUERADE может быть использовано вместо SNAT, даже если имеется постоянный IP-адрес.

MASQUERADE допускается указывать только в цепочке POSTROUTING таблицы nat, так же как и действие SNAT. MASQUERADE имеет ключ, использование которого необязательно.

Ключ для действия MASQUERADE — --to-ports — используется для указания порта источника или диапазона портов исходящего пакета. Можно указать один порт, например:

```
--to-ports 1025
```

или диапазон портов:

```
--to-ports 1024-3100
```

Этот ключ можно использовать только в правилах, где критерий содержит явное указание на протокол TCP или UDP с помощью ключа --protocol.

Пример

```
iptables -t nat -A POSTROUTING -p TCP -j MASQUERADE --to-ports 1024-31000
```

REDIRECT

REDIRECT выполняет перенаправление пакетов и потоков на другой порт того же самого компьютера. К примеру, можно пакеты, поступающие на HTTP-порт перенаправить на порт HTTP-прокси. Действие REDIRECT очень удобно для выполнения прозрачного проксирования (transparent proxying), когда компьютеры в ЛВС даже не подозревают о суще-

ствовании прокси.

REDIRECT может использоваться только в цепочках PREROUTING и OUTPUT таблицы nat, а также выполняться в подцепочках.

Ключ для действия REDIRECT — --to-ports — определяет порт или диапазон портов назначения. Без указания ключа --to-ports перенаправления не происходит, т. е. пакет идет на тот порт, куда и был назначен. В примере, приведенном ниже, --to-ports 8080 указан один порт назначения.

Пример

```
iptables -t nat -A PREROUTING -p tcp --dport 80 -j REDIRECT --to-ports 8080
```

Если необходимо указать диапазон портов, то написать:

```
--to-ports 8080-8090
```

Этот ключ можно использовать только в правилах, где критерий содержит явное указание на протокол TCP или UDP с помощью ключа --protocol.

REJECT

REJECT используется, как правило, в тех же самых ситуациях, что и DROP, но в отличие от DROP, команда REJECT выдает сообщение об ошибке на компьютер, передавший пакет. Действие REJECT работает только в цепочках INPUT, FORWARD и OUTPUT (и во вложенных в них цепочках). Пока существует только единственный ключ, управляющий поведением команды REJECT.

Ключ для действия REJECT — --reject-with — указывает, какое сообщение необходимо передать в ответ, если пакет совпал с заданным критерием. При применении действия REJECT к пакету, сначала на компьютер-отправитель будет отослан указанный ответ, а затем пакет будет сброшен. Допускается использовать следующие типы ответов: icmp-net-unreachable, icmp-host-unreachable, icmp-port-unreachable, icmp-proto-unreachable, icmp-net-prohibited и icmp-host-prohibited. По умолчанию передается сообщение port-unreachable. Все вышеуказанные типы ответов являются ICMP error messages (сообщениями об ошибках). Тип ответа tcp-reset используется только для протокола TCP. Если указано значение tcp-reset, то действие REJECT передаст в ответ пакет TCP RST, который используется для закрытия TCP-соединения.

Пример

```
iptables -A FORWARD -p TCP --dport 22 -j REJECT --reject-with tcp-reset
```

RETURN

RETURN прекращает движение пакета по текущей цепочке правил и производит возврат в вызывающую цепочку, если текущая цепочка была вложенной, или, если текущая

цепочка лежит на самом верхнем уровне (например, INPUT), то к пакету будет применена политика по умолчанию. В качестве политики по умолчанию назначают действия ACCEPT или DROP.

SNAT

SNAT используется для преобразования сетевых адресов, т. е. изменение исходящего IP-адреса в IP-заголовке пакета. SNAT допускается выполнять только в таблице nat, в цепочке POSTROUTING. Другими словами, только здесь допускается преобразование исходящих адресов. Если первый пакет в соединении подвергся преобразованию исходящего адреса, то все последующие пакеты из этого же соединения будут преобразованы автоматически и не пойдут через эту цепочку правил.

Ключ для действия SNAT — --to-source — используется для указания адреса, присваиваемого пакету.

Пример

```
iptables -t nat -A POSTROUTING -o eth0 -j SNAT --to-source
194.236.50.155-194.236.50.160:1024-32000
```

Указывается IP-адрес, который будет подставлен в заголовок пакета в качестве исходящего. Если необходимо перераспределить нагрузку между несколькими брандмауэрами, то можно указать диапазон адресов, где начальный и конечный адреса диапазона разделяются дефисом, например:

```
194.236.50.155-194.236.50.160
```

Тогда конкретный IP-адрес будет выбираться из диапазона случайным образом для каждого нового потока. Дополнительно можно указать диапазон портов, которые будут использоваться только для нужд SNAT.

TOS

TOS используется для установки бит в поле TOS IP-заголовка. Поле TOS содержит восемь бит, которые используются для маршрутизации пакетов. Это одно из нескольких полей, используемых iproute2. Данное поле может обрабатываться различными маршрутизаторами с целью выбора маршрута движения пакета. Как уже указывалось выше, это поле, в отличие от MARK, сохраняет свое значение при движении по сети, а поэтому может использоваться для маршрутизации пакета. Данное действие допускается выполнять только в пределах таблицы mangle.

Ключ для действия TOS — --set-tos — определяет числовое значение в десятичном или шестнадцатеричном виде.

Пример

```
iptables -t mangle -A PREROUTING -p TCP --dport 22 -j TOS --set-tos 0x10
```

Поскольку поле TOS является 8-битным, то можно указать число в диапазоне от 0 до 255 (0x00–0xFF). Большинство значений этого поля никак не используются. Лучше использовать мнемонические обозначения: Minimize-Delay (16 или 0x10), Maximize-Throughput (8 или 0x08), Maximize-Reliability (4 или 0x04), Minimize-Cost (2 или 0x02) или Normal-Service (0 или 0x00). По умолчанию большинство пакетов имеют признак Normal-Service или нуль. Список мнемоник можно получить, выполнив команду:

```
iptables -j TOS -h
```

TTL

TTL используется для изменения содержимого поля TTL в IP-заголовке. Один из вариантов применения этого действия — устанавливать значение поля TTL во всех исходящих пакетах в одно и то же значение.

Действие TTL можно указывать только в таблице mangle и нигде больше.

Ключи для действия TTL приведены в таблице 24.

Таблица 24

Ключ	Описание
--ttl-set	Устанавливает поле TTL в заданное значение. Оптимальным считается значение около 64. Пример `iptables -t mangle -A PREROUTING -o eth0 -j TTL --ttl-set 64`
--ttl-dec	Уменьшает значение поля TTL на заданное число. Например, пусть входящий пакет имеет значение TTL, равное 53, выполняется команда --ttl-dec 3. Тогда пакет покинет компьютер с полем TTL, равным 49. Сетевой код автоматически уменьшит значение TTL на 1, поэтому фактически получается: 53 - 3 - 1 = 49. Пример `iptables -t mangle -A PREROUTING -o eth0 -j TTL --ttl-dec 1`
--ttl-inc	Увеличивает значение поля TTL на заданное число. Пусть поступает пакет с TTL, равным 53, тогда после выполнения команды --ttl-inc 4 на выходе с компьютера пакет будет иметь TTL, равный 56, не стоит забывать об автоматическом уменьшении поля TTL сетевым кодом ядра, т. е. фактически получается выражение: 53 + 4 - 1 = 56. Пример `iptables -t mangle -A PREROUTING -o eth0 -j TTL --ttl-inc 1`

ULOG

ULOG предоставляет возможность журналирования пакетов в пользовательское пространство. Оно заменяет традиционное действие LOG, базирующееся на системном журнале. При использовании этого действия пакет через сокеты netlink передается специальному демону, который может выполнять очень детальное журналирование в различных форматах (например, обычный текстовый файл) и к тому же поддерживает возможность добавления надстроек (плагинов) для формирования различных выходных форма-

тов и обработки сетевых протоколов.

Ключи для действия ULOG приведены в таблице 25.

Таблица 25

Ключ	Описание
--ulog-nlgroup	Сообщает ULOG, в какую группу netlink должен быть передан пакет. Всего существует 32 группы (от 1 до 32). Если необходимо передать пакет в пятую группу, то можно просто указать: --ulog-nlgroup 5 По умолчанию используется первая группа. Пример iptables -A INPUT -p TCP --dport 22 -j ULOG --ulog-nlgroup 2
--ulog-prefix	Имеет тот же смысл, что и аналогичная опция в действии LOG. Длина строки префикса не должна превышать 32 символа. Пример iptables -A INPUT -p TCP --dport 22 -j ULOG --ulog-prefix "SSH connection attempt: "
--ulog-cprange	Определяет, какую долю пакета, в байтах, надо передавать демону ULOG. Если указать число 100, как показано в примере, то демону будет передано только 100 Б из пакета, это означает, что демону будет передан заголовок пакета и некоторая часть области данных пакета. Если указать нуль, то будет передан весь пакет, независимо от его размера. Значение по умолчанию равно нулю. Пример iptables -A INPUT -p TCP --dport 22 -j ULOG --ulog-cprange 100
--ulog-qthreshold	Устанавливает величину буфера в области ядра. Например, если задать величину буфера, равной 10, как в примере, то ядро будет накапливать журналируемые пакеты во внутреннем буфере и передавать в пользовательское пространство группами по 10 пакетов. Пример iptables -A INPUT -p TCP --dport 22 -j ULOG --ulog-qthreshold 10

5.7. Настройка защищенного интерпретатора команд SSH

5.7.1. Общие сведения

Защищенный интерпретатор команд SSH — это клиент-серверная система для организации защищенных туннелей между двумя и более компьютеров. В таких туннелях защищаются все передаваемые данные, в т. ч. пароли, что особенно полезно при доступе к другому компьютеру по сети в режиме суперпользователя.

5.7.2. Сервер — служба sshd

Служба `sshd` запускается на этапе начальной загрузки из сценария `/etc/rc.d/init.d/sshd`. Этот сценарий, а также ссылки на него в виде сценариев запуска и останова службы создаются в процессе установки программы. По умолчанию служба прослушивает порт 22. Когда поступает запрос на подключение, он порождает дочерний процесс, который управляет передачей данных в рамках конкретного соединения.

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

Таблица 26

Параметр	Описание
`AllowGroups`	Задает разделенный пробелами список групп. Эти группы будут допущены в систему
`DenyGroups`	То же, что `AllowGroups`, только смысл проверки обратный. Записанные в этот параметр группы не будут допущены в систему
`AllowUsers`	Задает разделенный пробелами список пользователей. Только перечисленные пользователи получают доступ в систему. По умолчанию доступ разрешен всем пользователям
`DenyUsers`	То же, с противоположным смыслом проверки
`AFSTokenPassing`	Указывает на то, может ли маркер `AFS` пересылаться на сервер. По умолчанию — `yes`
`AllowTCPForwarding`	Указывает на то, разрешены ли запросы на переадресацию портов (по умолчанию — `yes`)
`Banner`	Отображает полный путь к файлу сообщения, выводимого перед аутентификацией пользователя
`ChallengeResponseAuthentication`	Указывает на то, разрешена ли аутентификация по методу «оклик — ответ». 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 (по умолчанию — `no`)

Продолжение таблицы 26

Параметр	Описание
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. (По умолчанию — /etc/ssh/ssh_host_key)
GatewayPorts	Указывает на то, могут ли удаленные компьютеры подключаться к портам, для которых клиент запросил переадресацию (по умолчанию — no)
HostbasedAuthentication	Указывает на то, разрешена ли аутентификация пользователей с проверкой файлов .rhosts и /etc/hosts.equivи открытого ключа компьютера. Значение по умолчанию — no
IgnoreRhosts	Указывает на то, игнорируются ли файлы $HOME/.rhosts и $HOME/.shosts. По умолчанию — yes
IgnoreUserKnownHosts	Указывает на то, игнорируется ли файл $HOME/.ssh/known_hosts в режимах аутентификации RhostsRSAAuthentication и HostbasedAuthentication (по умолчанию — no)
KeepAlive	Если равен yes (по умолчанию), демон sshd будет периодически проверять наличие связи с клиентом. В случае неуспешного завершения проверки соединение разрывается. Чтобы отключить этот механизм, надо задать параметр, равным no, в файле конфигурации и сервера, и клиента
KerberosAuthentication	Указывает на то, разрешена ли аутентификация с использованием Kerberos. По умолчанию — no
KerberosOrLocalPasswd	Указывает на то, должна ли использоваться локальная парольная аутентификация в случае неуспешной аутентификации на основе Kerberos
KerberosTgtPassing	Указывает на то, может ли структура TGT системы Kerberos пересылаться на сервер (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

Продолжение таблицы 26

Параметр	Описание
LogLevel	Задает степень подробности журнальных сообщений. Возможные значения: QUIET, FATAL, ERROR, INFO (по умолчанию), VERBOSE, DEBAG (не рекомендуется)
MACs	Задает разделенный запятыми список доступных алгоритмов MAC (код аутентификации сообщений), используемых для обеспечения целостности данных
MaxStartups	Задает максимальное число одновременных неаутентифицированных соединений с демоном sshd
PAMAuthenticationViaKbdInt	Указывает на то, разрешена ли парольная аутентификация с использованием модулей PAM (по умолчанию — no)
PasswordAuthentication	Если равен 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 параметр равен no, парольная аутентификация запрещена
PermitEmptyPasswords	Если равен yes, пользователи, не имеющие пароля, могут быть аутентифицированы службой sshd. Если параметр равен no (по умолчанию), пустые пароли запрещены
PermitRootLogin	Указывает на то, может ли пользователь root войти в систему с помощью команды ssh. Возможные значения: yes (по умолчанию), without-password, forced-command-only и no
PidFile	Задает путь к файлу, содержащему идентификатор главного процесса (по умолчанию — /var/run/sshd.pid)
Port	Задает номер порта, к которому подключается sshd. По умолчанию — 22
PrintLastLog	Указывает на то, должна ли служба sshd отображать сообщение о времени последнего доступа. По умолчанию — yes
PrintMotd	Указывает на то, следует ли после регистрации в системе отображать содержимое файла /etc/motd. По умолчанию — yes
Protocol	Задает разделенный запятыми список версий протокола, поддерживаемых службой sshd
PubKeyAuthentication	Указывает на то, разрешена ли аутентификация с использованием открытого ключа (по умолчанию — yes)
ReverseMappingCheck	Указывает на то, должен ли выполняться обратный поиск имен. По умолчанию — no

Окончание таблицы 26

Параметр	Описание
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 (по умолчанию — no)
X11Forwarding	Указывает на то, разрешена ли переадресация запросов к системе X Window (по умолчанию —no)
X11DisplayOffset	Задает номер первого дисплея (сервера) системы X Window, доступного демону sshd для переадресации запросов (по умолчанию — 10)
XAuthLocation	Задает путь к команде xauth (по умолчанию — /usr/X11R6/bin/xauth)

5.7.3. Клиент ssh

Клиентом является команда ssh. Синтаксис командной строки:

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

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

```
ssh 10.1.1.170
```

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

```
-l <имя_клиента>
```

Однако сервер будет согласовывать ключ сеанса (например, при беспарольной аутентификации по открытому ключу пользователя), проверяя открытые ключи в домашнем каталоге пользователя именно с этим именем на компьютере-клиенте. Если же используется парольная аутентификация, на компьютере-сервере должна существовать учетная

запись с таким именем. Использовать беспарольную аутентификацию по открытым ключам компьютера настоятельно не рекомендуется, т. к. при этом способе в системе должны существовать потенциально опасные файлы: /etc/hosts.equiv, /etc/shosts.equiv, $HOME/.rhosts, $HOME/.shosts.

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

В таблице 27 описаны параметры, которые могут присутствовать в файле $HOME/.ssh/config или /etc/ssh/ssh_config. Пустые строки и комментарии игнорируются.

Таблица 27

Параметр	Описание
CheckHostIP	Указывает на то, должна ли команда ssh проверять IP-адреса в файле known_hosts (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 (по умолчанию — no). Эта установка может быть переопределена с помощью опции командной строки -C
ConnectionAttempts	Задает число неудачных попыток подключения (одна в секунду), после чего произойдет завершение работы. Значение по умолчанию — 4
EscapeChar	Задает escape-символ, используемый для отмены специального назначения следующего символа в сеансах с псевдотерминалом. По умолчанию — ~. Значение none запрещает использование escape-символа
ForwardAgent	Указывает на то, будет ли запрос к команде ssh-agent переадресован на удаленный сервер (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 (по умолчанию — no)
GatewayPorts	Указывает на то, могут ли удаленные компьютеры подключаться к локальным портам, для которых включен режим переадресации (по умолчанию — no)
GlobalKnownHostsFile	Задает файл, в котором хранится глобальная база ключей компьютера (по умолчанию — /etc/ssh/ssh_known_hosts)

Продолжение таблицы 27

Параметр	Описание
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. Выбор 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 (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 отключить этот механизм, следует задать данный параметр, равным no, в файлах /etc/ssh/sshd_config и /etc/ssh/ssh_config (либо $HOME/.ssh/config)
KerberosAuthentication	Указывает на то, разрешена ли аутентификация с применением Kerberos
KerberosTgtPassing	Указывает на то, будет ли структура TGT системы Kerberos пересылаться на сервер
LocalForward	Требует значения в формате порт:узел:удаленный_порт. Указывает на то, что запросы к соответствующему локальному порту перенаправляются на заданный порт удаленного узла
LogLevel	Задает степень подробности журнальных сообщений команды ssh. Возможные значения: QUIET, FATAL, ERROR, INFO (по умолчанию), VERBOSE, DEBAG
MACs	Задает разделенный запятыми список доступных алгоритмов аутентификации сообщений для обеспечения целостности данных. Стандартный выбор: hmac-md5, hmac-sha1, hmac-ripemd160@openssh.com, hmac-sha1-96, hmac-md5-96
NumberOfPasswordPrompts	Задает число допустимых попыток ввести пароль (по умолчанию — 3)
PasswordAuthentication	Если равен yes (по умолчанию), то в случае необходимости команда ssh пытается провести парольную аутентификацию
Port	Задает номер порта сервера (по умолчанию — 22)

Окончание таблицы 27

Параметр	Описание
PreferredAuthentications	Задает порядок применения методов аутентификации (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 (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 опции 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
User	Задает пользователя, от имени которого следует регистрироваться в удаленной системе. Эта установка может быть переопределена с помощью опции командной строки -l
UserKnownHostsFile	Задает файл, который используется для автоматического обновления открытых ключей
XAuthLocation	Задает путь к команде xauth (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.

5.8. Настройка сервера единого сетевого времени NTP

5.8.1. Назначение

Сервер единого сетевого времени предназначен для синхронизации времени компьютера в ЛВС. В основе лежит сетевой протокол времени (Network Time Protocol, NTP). Алгоритм коррекции временной шкалы включает внесение задержек, коррекцию частоты часов и ряд механизмов, позволяющих достичь точности порядка нескольких миллисекунд, даже после длительных периодов, когда потеряна связь с синхронизирующими источниками. Для надежной защиты передаваемого сигнала используется аутентификация при помощи криптографических ключей. Целостность данных обеспечивается с помощью IP- и UDP-контрольных сумм.

5.8.2. Режимы работы

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

1) клиент-сервер — в этом режиме клиент посылает запрос серверу, который обрабатывает его и немедленно посылает ответ. Такой режим работы обеспечивает синхронизацию времени клиента со временем сервера, но сам сервер при этом с клиентом не синхронизируется. Режим «клиент-сервер» используется в тех случаях, когда нужна максимальная точность синхронизации времени и надежная защита передаваемой информации;
2) симметричный — может быть активным или пассивным:
   – в активном режиме каждый компьютер в сети периодически посылает сообщения другому компьютеру вне зависимости от ее достижимости и слоя. При этом компьютер оповещает о своем намерении синхронизировать и быть синхронизированным своим партнером. Адреса партнеров известны заранее. Этот режим обычно используется серверами с большим номером слоя;
   – в пассивном режиме адрес партнера заранее не известен. Взаимодействие в этом режиме начинается по прибытии сообщения от партнера (с неизвестным адресом), работающего в симметрично активном режиме, и сохраняется до тех пор, пока партнер достижим и функционирует в слое ниже или равном слою данного компьютера. Пассивный режим обычно используется первичными или вторичными серверами.

   Симметричный режим обеспечивает высокую надежность синхронизации, т. к. при выходе из строя одного из источников времени система автоматически переконфигурируется таким образом, чтобы исключить его из сети синхронизации;
3) широковещательный — в этом режиме один или более серверов времени рассы-

лают широковещательные сообщения, клиенты определяют время исходя из предположения, что задержка составляет несколько миллисекунд. Сервер при этом не принимает ответных ntp-сообщений.

Такой режим используется в быстрых локальных сетях с большим числом рабочих станций и без необходимости в высокой точности;

4) межсетевой — аналогичен широковещательному, но в отличие от него ntp-сообщения передаются не в рамках одной подсети, ограниченной локальным широковещательным адресом, а распространяются так же и в другие сети. Для работы службы единого времени в межсетевом режиме выделен специальный групповой IP-адрес (`224.0.1.1`), который используется как для серверов, так и для клиентов.

Межсетевой режим используется в сетях, разделенных на подсети с помощью маршрутизаторов и мостов, которые не способны ретранслировать широковещательные IP-дейтаграммы.

При реализации службы единого сетевого времени на сети системы могут играть четыре возможные роли:

1) серверы — предоставляют сервис времени другим системам;

2) равноправные узлы — многие серверы единого времени вступают в равноправные отношения с другими серверами того же уровня (`stratum level`). Если сервер второго уровня теряет связь со своим источником времени первого уровня, он может временно использовать сервис времени, предоставляемый равноправным узлом второго уровня;

3) опросные клиенты — регулярно опрашивают, как минимум, один сервер единого времени, сличают ответы серверов и синхронизируют системные часы по наиболее точному источнику времени;

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

Серверы второго уровня опрашивают серверы первого, получая от них текущее системное время. Рекомендуется, чтобы каждый сервер единого времени второго уровня сверялся, как минимум, с тремя серверами первого уровня для обеспечения надежности.

Демон `ntpd` будет автоматически опрашивать оба сервера первого уровня и синхронизироваться по источнику, который он считает наиболее точным. Чтобы еще более повысить надежность, каждый сервер второго уровня должен установить равноправные отношения, как минимум, еще с одним сервером второго уровня.

5.8.3. Настройка

Настройка и управление сервером `ntpd` осуществляется либо путем задания опций в командной строке, либо путем редактирования конфигурационного файла. Первый способ предоставляет ограниченные возможности настройки, второй — наиболее полные.

Во время своего запуска сервер `ntpd` читает конфигурационный файл, который обычно находится в каталоге `/etc`, но может быть перемещен в любой другой каталог (см. опцию командной строки `-c conffile`).

Формат файла аналогичен формату других конфигурационных файлов ОС: комментарии начинаются с символа # и действуют до конца строки, пустые строки игнорируются. Конфигурационные команды состоят из ключевого слова и следующих за ним аргументов, разделенных пробелами. Любая команда должна занимать строго одну строку. Аргументами могут быть имена и адреса хостов (в форме IP-адресов и доменных имен), целые и дробные числа, текстовые строки. Далее необязательные аргументы заключены в квадратные скобки — `[ ]`, альтернативные аргументы отделены символом `|`. Нотация вида `[...]` означает, что стоящий перед ней необязательный аргумент может повторяться несколько раз.

5.8.4. Установка

Действия, которые необходимо выполнить для установки сервера:

1) установить сервер NTP из соответствующего `deb-пакета` (при стандартной установке ОС сервер включается в состав пакетов по умолчанию);
2) добавить в переменную окружения `PATH` путь `/usr/ccs/bin`;
3) в `/etc/ntp/step-tickers` положить список опорных NTP-серверов;
4) для автоматического запуска создать символические ссылки `/etc/rc.d/rc{2,3}.d {K,S}31ntpd` на файл `ntpd`;
5) настроить файл `/etc/ntp.conf`;
6) подстроить приблизительное время часов вручную. Точность настройки не должна быть хуже 1000 с от реального времени;
7) перезапустить ОС.

5.8.5. Конфигурация

Настройка и управление сервером `ntpd` осуществляется либо путем задания опций в командной строке, либо путем редактирования конфигурационного файла.

Первый способ предоставляет ограниченные возможности настройки, второй — наиболее полные.

По умолчанию необходимый конфигурационный файл `ntp.conf` после установки находится в каталоге `/etc`.

5.8.5.1. Конфигурационный файл ntp.conf

Синтаксис:

```
server address [key key | autokey] [version version] [prefer]
[minpoll minpoll] [maxpoll maxpoll]
peer address [key key | autokey] [version version] [prefer]
[minpoll minpoll] [maxpoll maxpoll]
broadcast address [key key | autokey] [version version]
[minpoll minpoll] [ttl ttl]
manycastclient address [key key | autokey] [version version] [minpoll minpoll
[maxpoll maxpoll] [ttl ttl]
```

Описание команд приведено в таблице 28.

Таблица 28

Команда	Описание
`server`	Позволяет установить постоянное соединение (организовать постоянную ассоциацию) клиента с удаленным сервером. При этом локальное время может быть синхронизировано с удаленным сервером, но удаленный сервер не может синхронизировать свое время с локальным
`peer`	Устанавливается постоянное соединение (ассоциация) в симметрично-активном режиме с указанным удаленном сервером (`peer` — симметричным). В данном режиме локальные часы могут быть синхронизированы с удаленным симметричным сервером или удаленный сервер может синхронизироваться с локальными часами
`broadcast`	Организуется постоянная широковещательная ассоциация
`manycastclient`	Организуется межсетевой режим синхронизации с указанным групповым адресом
`vmanycast`	Указывает, что локальный сервер должен работать в клиентском режиме с удаленными серверами, которые обнаруживаются в процессе работы при помощи широковещательных/межсетевых пакетов

Описание опций команд приведено в таблице 29.

Таблица 29

Опция	Описание
`autokey`	Все отсылаемые пакеты включают аутентификационные поля, зашифрованные в автоматическом режиме
`key key`	Все отправляемые и принимаемые пакеты включают поля аутентификации, зашифрованные при помощи ключа шифрования с заданным идентификатором, значения которого составляют от 1 до 65534. По умолчанию поля аутентификации не используются
`minpoll minpoll`, `maxpoll maxpoll`	Указание временных задержек
`noselect`	Указывает, что сервер используется только в демонстративных целях

Окончание таблицы 29

Опция	Описание
`prefer`	Отмечает, что сервер является предпочтительным
`ttl ttl`	Данная опция используется только в широковещательном и межсетевом режимах. Указывает время жизни пакета
`version version`	Указывает версию протокола отправляемых пакетов (по умолчанию — 4)

5.8.5.2. Конфигурирование процесса аутентификации

Поддержка аутентификации позволяет клиенту службы единого времени удостовериться, что сервер является именно тем, за кого он себя выдает. Конфигурирование производится в файле `ntp.conf` с использованием дополнительных опций команд `peer`, `server`, `broadcast` и `multicast`.

– `autokey [logsec]` — указывает интервалы в секундах между генерациями нового ключа;

– `controlkey key` — указывает идентификатор ключа для использования командой `ntpq`;

– `keys keyfile` — указывает местонахождение файла, хранящего ключи и их идентификаторы, используемые командами `ntpd`, `ntpq` и `ntpdc`. Данная команда эквивалентна использованию опции `-k` командной строки;

– `keysdir путь_к_директории` — указывает путь к каталогу, хранящему ключи (по умолчанию). По умолчанию — `/usr/local/etc/`;

– `trustedkey key [...]` — указывает идентификаторы ключей, которые являются доверенными для аутентификации с симметричным ключом.

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

5.8.5.3. Конфигурация сервера уровней 1 и 2

Чтобы настроить конфигурацию сервера уровня 1, необходимо добавить в файл `/etc/ntp.conf` следующие строки:

```
server символический_IP_адрес
peer DNS_имя_соседнего_сервера_1
peer DNS_имя_соседнего_сервера_2
```

Символический IP-адрес в первой строке используется службой `ntpd` для того, чтобы определить, какого типа радиочасы подсоединены к системе. Конфигурация сервера уровня 2:

```
server DNS_имя_сервера_уровня_1
server DNS_имя_сервера_уровня_1
```

```
peer DNS_имя_соседнего_сервера_уровня_2
driftfile /etc/ntp.drift
broadcast _IP_адрес
```

Записи `server` определяют, какие серверы уровня 1 должен опрашивать данный сервер, чтобы воспользоваться сервисом времени.

Запись `peer` определяет равноправные отношения с другим сервером уровня 2.

Запись `driftfile` задает имя файла, который будет использоваться для отслеживания долгосрочного сдвига локальных часов.

Запись `broadcast` указывает демону `ntpd` регулярно сообщать вещательным клиентам сети об официальном времени.

5.8.6. Методы синхронизации системных часов

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

Команда `ntpdate`, выполняемая с опцией `-b`, опрашивает, как минимум, один сервер единого времени, затем синхронизирует системные часы с наиболее точным сервером единого сетевого времени. Выполняется только при запуске системы до того, как запускаются приложения.

После того, как во время загрузки команда `ntpdate` первоначально синхронизирует системные часы, демон `ntpd` постоянно работает в фоновом режиме, периодически опрашивая серверы службы единого времени, заданные в `/etc/ntp.conf`, и по мере необходимости «подкручивая» системные часы, чтобы поддерживать синхронизацию. Эти незначительные постепенные корректировки во времени должны быть прозрачными для приложений. Файл сдвига, определяемый в записи `driftfile`, используется для отслеживания различий между временем клиента и временем сервера. По мере стабилизации файла сдвига сервер будет опрашиваться все реже.

5.8.6.1. Команды командной строки

Синтаксис:

```
ntpd [-опции]
```

Команда `ntpd` является демоном ОС, который устанавливает и поддерживает системное время, синхронизируя его с остальными серверами единого времени. Демон `ntpd` обменивается сообщениями с одним или более серверами с установленной периодичностью.

Опции командной строки приведены в таблице 30.

Таблица 30

Опция	Описание
-4	Форсирование разрешения доменных имен в пространство имен протокола IP версии 4
-6	Использование пространства имен протокола IP версии 6
-A	Не использовать криптографические алгоритмы
-b	Разрешить клиенту синхронизировать системное время с вещательными клиентами
-c конфигурационный_файл	Указать имя и путь конфигурационного файла (по умолчанию — /etc/ntp.conf)
-d	Отладочный режим
-D уровень	Указать уровень отладки
-f driftfile	Указать имя файла сдвига частоты локальных системных часов (по умолчанию — /etc/ntp.drift). Эта опция аналогична команде driftfile в /etc/ntp.conf
-g	Обычно процесс ntpd завершается с соответствующим сообщением в файле журналирования, если локальное время отличается от реального времени более, чем на 1000 с. Данная опция позволяет устанавливать время без каких-либо ограничений, однако, это может быть сделано только один раз. Если порог будет превышен и после этой операции демон ntpd будет завершен с соответствующим сообщением в файл журнала. Эта опция может использоваться с опциями -q и -x
-i директория	Поменять корневой каталог на каталог, указанный в команде. Данная опция подразумевает, что сервер пытается при запуске понизить привилегии суперпользователя, иначе могут возникнуть некоторые проблемы с безопасностью. Это возможно, если ОС поддерживает работу сервера без полных привилегий root
-k keyfile	Указать имя и путь к файлу симметричного ключа (по умолчанию — /etc/ntp.keys). Эта опция аналогична команде keyfile в файле /etc/ntp.conf
-l путь_и_имя_файла	Указать имя и путь к файлу логического журнала. По умолчанию используется системный файл логического журнала. Данная опция эквивалентна команде logfile в конфигурационном файле /etc/ntp.conf
-L	Не прослушивать виртуальные IP-адреса. По умолчанию прослушиваются
-m	Разрешить клиенту синхронизировать межсетевые сервера IP версии 4 с групповым адресом 224.0.1.1
-n	Не использовать системный вызов fork
-N	Запускать ntpd с максимальным приоритетом

Окончание таблицы 30

Опция	Описание
`p файл_процесса`	Указать имя и путь к файлу, хранящему идентификатор процесса `ntpd` в системе. Данная опция эквивалентна команде `pidfile` в конфигурационном файле
`-P приоритет`	Указать приоритет запускаемого серверного процесса
`-q`	Завершить процесс `ntpd` сразу после синхронизации времени
`-r задержка_распространения_вещательного_пакета`	Задержка распространения вещательного пакета от сервера клиенту. Данная опция необходима только, если задержка не может быть вычислена автоматически протоколом NTP
`-s директория`	Указать путь к каталогу с файлами, создаваемыми командой подсчета статистики
`-u пользователь [:группа]`	Указать пользователя (группу), от чьего имени запускается процесс. Данная опция возможна только в ОС, в которой процесс `ntpd` может быть запущен без прав `root`
`-x`	Запустить процесс в обычном режиме. Локальное системное время корректируется процессом только, если «ошибка» составляет менее, чем установленная величина порога, которая по умолчанию составляет 128 мс. Данная опция устанавливает величину порога в 600 с

5.8.6.2. ntpq

Синтаксис:

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

Команда `ntpq` используется для мониторинга деятельности демона `ntpd` и определения производительности. Может быть запущена как в интерактивном режиме, так и с использованием опций командной строки. Она может получать и выводить на терминал список серверов того же уровня синхронизации в обычном формате, запрашивая все сервера.

Опции командной строки приведены в таблице 31.

Таблица 31

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

Интерактивные команды

Интерактивная команда состоит из командного слова и следующих за ним аргументов (возможно использование от 0 до 4 аргументов). Вывод результата выполнения команды направляется на стандартный вывод (stdout). Другими словами, можно перенаправлять вывод команды в файл, используя > имя_файла. Список интерактивных команд приведен в таблице 32.

Таблица 32

Команда	Описание
? [командное_слово] help1 [командное_слово]	Если задана опция ?, на терминал будет выдана информация о возможном использовании данной команды
addvars имя_переменной [= значение] [...] rmvars имя_переменной [...] clearvars	Данные, передаваемые протоколом NTP, содержат ряд сущностей вида: имя_переменной=значение. Команда ntpq поддерживает внутренний список, в котором данные встраиваются в контрольные сообщения. Команда addvars добавляет переменные в список, rmvars удаляет переменные из списка, clearvars полностью очищает список
cooked	Позволяет преобразовать вывод переменных и их значения в удобный для пользователя вид
debug more | less | off	Позволяет включить/выключить внутреннюю команду запросов
delay миллисекунды	Указывает временный интервал для добавления к временной отметке (timestamp), которая включается в запросы, требующие аутентификации. Это используется для возможности переконфигурации сервера
host имя_хоста	Устанавливает имя хоста, к которому будут отсылаться последующие запросы
hostnames [yes | no]	Если указывается yes, доменные имена хостов выводятся на терминал. Иначе выводятся на терминал численные адреса. По умолчанию стоит yes
keyid идентификатор_ключа	Позволяет указать номер ключа для использования его в запросах, требующих аутентификацию
ntpversion 1 | 2 | 3 | 4	Устанавливает номер версии NTP. По умолчанию используется протокол версии 6
passwd	Запрашивает пароль, который будет использоваться в запросах, требующих аутентификации
quit	Выход из интерактивного режима ntpq
raw	Заставляет выводить результаты запросов команды, как будто они пришли от удаленного сервера
timeout миллисекунды	Устанавливает временной интервал запросов серверам. По умолчанию составляет 5000 мс

Команды контрольных сообщений

Каждая ассоциация, известная NTP-серверу, имеет личный 16-битный целочисленный идентификатор. Ассоциация с идентификатором 0 играет особую роль — определяет системные переменные, чьи имена лежат вне локального пространства имен. Команды контрольных сообщений приведены в таблице 33

Таблица 33

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

5.8.6.3. ntpdate

Синтаксис:

```
ntpdate [ -опции]
```

Команда `ntpdate` устанавливает локальное системное время, используя NTP. Должна быть запущена с правами `root`. Возможен запуск как из командной строки (вручную), так и из стартового скрипта, выполняемого при загрузке ОС. Есть возможность выполнения `ntpdate` из сценария демона `cron`.

Данная команда завершается, если обнаруживается, что на том же хосте запущен сервер `ntpd`.

Опции командной строки приведены в таблице 34.

Таблица 34

Опция	Описание
`-4`	Форсирование разрешения доменных имен в пространство имен протокола IP версии 4
`-6`	Использование пространства имен протокола IP версии 6
`-a ключ`	Разрешение аутентификации и указание ключа для использования. По умолчанию аутентификация отключена
`-d`	Отладочный режим

Окончание таблицы 34

Опция	Описание
`-q`	Только запрос. Никаких изменений локальных часов не производится
`-t время_в_секундах`	Установка максимального времени ожидания ответа сервера

5.8.6.4. ntptrace

Синтаксис:

```
ntptrace [ -vdn ] [ -r retries ] [ -t timeout ] [ server ]
```

Программа `ntptrace` определяет, где сервера NTP получают время, и проходит по цепочке серверов до источника точного времени.

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

Опции командной строки приведены в таблице 35.

Таблица 35

Опция	Описание
`-d`	Отладочный режим
`-n`	В результатах запроса вместо доменных имен хостов выдаются их IP-адреса. Данная опция удобна, когда в сети отсутствует DNS
`-r retries`	Установка количества попыток передачи (по умолчанию — 5)
`-t временная_задержка`	Установка временной задержки передачи данных в секундах (по умолчанию — 2)
`-v`	Выдача многословной информации о NTP-серверах

5.8.6.5. fly-admin-ntp

В состав ОС входит графическая утилита `fly-admin-ntp`, которая позволяет администратору произвести большинство настроек системы NTP в графическом режиме (см. электронную справку).

5.9. Сетевая защищенная файловая система

5.9.1. Назначение и возможности

Для организации защищенных файловых серверов предназначена сетевая защищенная файловая система (СЗФС), в основу которой положена сетевая ФС CIFS, работающая по протоколу SMB/CIFS. Протокол СЗФС содержит в себе сообщения, которые передают информацию о стандартных и расширенных атрибутах (атрибутах безопасности), а также сообщения для передачи мандатной метки субъекта доступа.

Условием корректного функционирования СЗФС является использование механизма единого пространства пользователей, обеспечивающее в рамках данной ЛВС однознач-

ное соответствие между логическим именем пользователя и его идентификатором (а также именем группы и ее идентификатором) на всех компьютерах (рабочих станциях и серверах), на которых данный пользователь может работать. Для корректной работы СЗФС необходима синхронизация UID/GID в системах клиента и сервера, т. к. информация о пользователях и группах передается в сеть в численных значениях.

СЗФС состоит из сервера и клиента. Сервер представляет собой расширенный сервер Samba и выполняет следующие задачи:

1) управление разделяемыми ресурсами;

2) контроль доступа к разделяемым ресурсам. При подключении клиента сервер устанавливает мандатную метку процесса, обслуживающего запросы клиента, в соответствии с мандатной меткой этого клиента. Этим обеспечивается мандатный контроль доступа к разделяемым файлам на стороне сервера.

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

1) отображение каталогов и файлов смонтированного сетевого ресурса;

2) передача на сервер дополнительной информации о классификационной метке пользователя (процесса), работающего с разделяемым ресурсом.

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

СЗФС предоставляет следующие базовые возможности:

– разделение ФС ОС «Astra Linux Special Edition», ОС типа Windows и, наоборот;

– совместное использование принтеров, подключенных к ОС «Astra Linux Special Edition», ОС типа Windows и, наоборот.

5.9.2. Состав

СЗФС состоит из нескольких компонентов:

– `smbd` — сервисная служба, которая обеспечивает работу службы печати и разделения файлов для клиентов типа ОС Windows. Конфигурационные параметры сервисной службы `smbd` описываются в файле `smb.conf`;

– `nmbd` — сервисная служба, которая обеспечивает работу службы имен NetBIOS, а также может использоваться для запроса других сервисных служб имен;

– `smbclient` — сервисная служба, которая реализует клиент, используемый для доступа к другим серверам и для печати на принтерах, подключенных к серверам;

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

– smbstatus — команда, сообщающая, кто в настоящее время пользуется сервером smbd.

В состав ОС входит графическая утилита fly-admin-samba, которая устанавливается при установке smbd и позволяет настроить пользовательский доступ к ресурсам СЗФС (см. электронную справку).

5.9.3. Настройка

СЗФС устанавливается в процессе установки ОС.

Настройка СЗФС в ОС осуществляется посредством настройки параметров главного конфигурационного файла.

Главный конфигурационный файл называется .conf и находится в каталоге /etc.

Файл smb.conf состоит из именованных разделов, начинающихся с имени раздела в квадратных скобках (например, с [global]). Внутри каждого раздела находится ряд параметров в виде key = value. Файл конфигурации содержит три специальных раздела: [global], [homes] и [printers] и несколько пользовательских разделов.

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

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

Этот фрагмент определяет рабочую группу WORKGR1, к которой относится данный компьютер, а также описывает саму систему.

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

Этот фрагмент описывает тип системы печати, доступный на сервере администратора, а также местонахождение конфигурационного файла принтера.

Последняя строка говорит о том, что все принтеры, определенные в файле printcap, должны быть доступны в сети.

```
;Раскомментируйте это поле, если вам нужен гостевой вход
;guest = pcguest
log file = /var/log/samba-log.%m
max log size = 50
```

В этом фрагменте определяется, будет ли сервер поддерживать гостевой вход. Следующие два параметра определяют работу с журнальными файлами. Параметр m со-

общает службе Samba, что для каждого клиента ведется свой файл, а последняя строка говорит о том, что максимальный размер создаваемого журнального файла — 50 КБ.

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

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

Параметр `comment` выводится для клиента при запросе о доступных ресурсах; параметр `browseable` определяет, как выводить ресурс в списке просмотра. Параметр `read only` определяет, может ли пользователь создавать и изменять файлы в своем рабочем каталоге при подключении по сети. Параметр `create mask` определяет права доступа для вновь создаваемых файлов в рабочем каталоге пользователя.

В разделе `[printers]` описаны параметры управления печатью при отсутствии иного явного описания. Используется для предоставления доступа к принтерам, определенным в файле `/etc/` (данная возможность в ОС заблокирована по умолчанию, для чего закомментированы все строки раздела `[printers]`).

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

Параметры `comment`, `browseable`, `create mode` описаны выше, см. раздел `[homes]`.

Параметр `path` определяет местонахождение файла спулера при печати через SMB.

Параметр `printable` определяет, может ли использоваться данный ресурс для печати, параметр `guest ok` — может ли воспользоваться принтером гостевой пользователь.

После настройки параметров сервера по умолчанию можно создать разделяемые

каталоги, доступ к которым могут получать определенные пользователи, группы пользователей или все пользователи. Рассмотрим пример создания разделяемого каталога с доступом только для одного пользователя. Для этого необходимо создать отдельный раздел файла smb.conf и заполнить его необходимой информацией (обычно это пользователь, каталог и конфигурационная информация).

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

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

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

Примечание. Применение testparm не дает гарантии, что все сервисы и ресурсы, описанные в конфигурации, доступны и будут корректно работать.

Синтаксис testparm:

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

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

Если ошибки не будут обнаружены, на экране появится примерно следующее сообщение (в случае обнаружения ошибок о них будет предоставлена полная информация):

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

При нажатии **<Enter>** testparm протестирует каждый раздел, определенный в кон-

фигурационном файле.

5.9.4. Запуск сервера

Сервер состоит из двух сервисных команд — `smbd` и `nmbd`. `smbd` обеспечивает работу службы разделения файлов и принтеров, а `nmbd` поддерживает имена NetBIOS.

Сервер запускается либо из инициализирующих сценариев, либо из `inetd` в качестве системного сервиса.

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

```
/etc/rc.d/init.d/samba start:stop
```

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

Опции командной строки `smbclient` позволяют сделать запрос о разделяемых ресурсах или перенести файлы.

Например, для запроса списка доступных ресурсов на удаленном сервере `win.netwhart.com` используется командная строка:

```
slibclient -L -I win.netwhart.com
```

Здесь опция `-L` указывает, что требуется вывести список разделяемых ресурсов, а опция `-I` — что указанное далее имя следует рассматривать как имя DNS, а не NetBIOS.

Для пересылки файла необходимо сначала подключиться к серверу с использованием команды:

```
smbclient '\\WORKGR1\PUBLIC' -I win.netwhart.com -U tackett
```

Параметр `\\WORKGR1\PUBLIC` определяет удаленный сервис на другом компьютере (обычно это каталог ФС или принтер). Опция `-U` позволяет определить имя пользователя для подключения к ресурсу (при этом, если необходимо, СЗФС запросит соответствующий пароль). После подключения появится приглашение:

```
Smb:\
```

где `\` означает текущий рабочий каталог.

В этой командной строке можно указать команды для передачи файлов и работы с ними (см. руководство `man`).

6. СРЕДСТВА ОРГАНИЗАЦИИ ЕДИНОГО ПРОСТРАНСТВА ПОЛЬЗОВАТЕЛЕЙ

6.1. Общие сведения

Организация ЕПП обеспечивает:

– сквозную аутентификацию в сети;

– централизацию хранения информации об окружении пользователей;

– централизацию хранения настроек системы защиты информации на сервере;

– централизацию управления серверами DNS и DHCP.

Сетевая аутентификация и централизация хранения информации об окружении пользователя подразумевает использование двух основных механизмов: поддержки кросс-платформенных серверных приложений для обеспечения безопасности (NSS) (6.2) и подгружаемых аутентификационных модулей (PAM) (6.3).

Для реализации удаленной аутентификации наиболее оптимальным решением является использование службы каталогов LDAP (6.4.3) в качестве источника данных для базовых системных сервисов на базе механизмов NSS и PAM.

Сквозная доверенная аутентификация реализуется технологией Kerberos (6.5).

Централизация хранения информации об окружении пользователей подразумевает и централизованное хранение домашних каталогов пользователей. Для этого используется СЗФС CIFS (см. 5.9).

6.2. Механизм NSS

Механизм NSS предоставляет всем программам и службам, функционирующим на локальном компьютере системную информацию через соответствующие программные вызовы. Он обращается к конфигурационному файлу `/etc/nsswitch.conf`, в котором указаны источники данных для каждой из системных служб. Краткое описание системных служб приведено в таблице 36.

Таблица 36

Сервис	Источник данных по умолчанию	Описание
`passwd`	`/etc/passwd`	Окружение пользователя (домашний каталог, идентификатор пользователя и пр.)
`shadow`	`/etc/shadow`	Пароли пользователей
`group`	`/etc/group`	Принадлежность пользователей группам
`hosts`	`/etc/hosts`	Соответствие имен хостов адресам
`services`	`/etc/services`	Характеристики сетевых сервисов (порт, тип транспортного протокола)

Каждый из базовых системных сервисов поддерживает ряд библиотечных про-

граммных вызовов, таких как getpwent, getspent, getgrent, getservent. При выполнении данных программных вызовов производится поиск в конфигурационном файле /etc/nsswitch.conf источника данных соответствующего сервиса (например, passwd для получения домашнего каталога пользователя). По умолчанию в качестве источника данных системных сервисов используются соответствующие конфигурационные файлы в каталоге /etc (источник files). NSS при получении имени источника данных из конфигурационного файла /etc/nsswitch.conf осуществляет поиск программной разделяемой библиотеки в каталоге /lib с именем libnss_<имя_источника_данных>-<версия_библиотеки>.so, где в качестве имени источника данных выступает строка, полученная из /etc/nsswitch.conf. Например, при вызове getpwent, при условии, что в /etc/nsswitch.conf находится строка:

```
passwd : files
```

будет вызвана соответствующая функция из библиотеки /lib/libnss_files.so.

6.3. Механизм PAM

Механизмы PAM имеют схожие с механизмами NSS принципы работы. В каталоге /etc/pam.d расположены конфигурационные файлы PAM для соответствующих сервисов, в том числе и для login (авторизованный вход в систему). В конфигурационном файле сервиса дана информация по проведению аутентификации.

Модули PAM вызываются при выполнении следующих функций:

1) auth — аутентификации;
2) account — получения привилегий доступа;
3) password — управления паролями;
4) session — сопровождения сессий.

Для выполнения каждой функции может быть перечислено несколько модулей PAM, которые будут вызываться последовательно, образуя стек PAM для данной задачи. Каждый вызываемый модуль возвращает в стек результат своей работы: или успешный (PAM_SUCCESS), или неуспешный (PAM_AUTH_ERR), или игнорирующий (PAM_IGNORE), или иной. Для каждого вызова может быть указан набор управляющих флагов в виде соответствия кода возврата и того, как результат работы модуля скажется на обработке всей сервисной задачи, например ignore, ok, die. Для управления аутентификацией используются следующие флаги:

– requisite — немедленное прекращение дальнейшего выполнения сервисной задачи с общим неуспешным результатом в случае неуспешного результата выполнения данного модуля;

– required — требование удачного выполнения этого модуля одновременно с выполнением всех остальных, перечисленных в данной сервисной задаче;

– `sufficient` — немедленное прекращение дальнейшего выполнения сервисной задачи с общим позитивным результатом, в случае позитивного результата выполнения данного модуля и всех предыдущих с флагом `required` в стеке задачи, если же модуль вернул негативный результат, то его значение игнорируется;

– `optional` — выполнение данного модуля никак не сказывается на результате всей задачи, но играет дополнительную информационную роль.

В отличие от механизма NSS механизм PAM позволяет исключительно проводить аутентификацию, т. е. подтверждать или опровергать введенную аутентификационную информацию.

6.4. Служба каталогов LDAP

6.4.1. Протокол доступа

LDAP — это протокол, используемый для доступа к информации, хранящейся на распределенных в сети серверах. Данная информация представляет собой данные, хранящиеся в атрибутах. При этом предполагается, что такие данные чаще читаются, чем модифицируются. LDAP основан на клиент-серверной модели взаимодействия. Общая модель данного протокола состоит в том, что клиент выполняет операции протокола на серверах. Клиент передает запрос, описывающий операцию, которая должна быть выполнена сервером. Сервер выполняет необходимые операции в каталоге. После завершения операции (операций) сервер возвращает клиенту ответ, содержащий результаты или ошибки. Следует заметить, что хотя требуется, чтобы серверы возвращали ответы всякий раз, когда такие ответы определены в протоколе, не существует требования синхронного поведения клиентов или серверов. Запросы и ответы для нескольких операций могут пересылаться между клиентом и сервером в любом порядке, однако клиенты должны получить ответ на каждый свой запрос. Информация на сервере LDAP представляет собой совокупность записей, которые содержат набор атрибутов и сгруппированы в древовидную иерархическую структуру.

Запись идентифицируется глобально уникальным именем (DN) подобно имени домена в структуре DNS. Каталог является специализированной БД, которая может использоваться в повседневной жизни — телефонная книга, программа передач и т. п. Предполагается, что данные каталога достаточно статичны. Классическим примером подобной специализированной БД является сервис DNS.

6.4.2. База каталога DIB

Информация, хранящаяся в каталоге, называется информационной базой каталога (DIB). Пользователь каталога, который может быть как человеком, так и компьютером, получает доступ к каталогу посредством клиента. Клиент от имени пользователя каталога

взаимодействует с одним или более серверами. Сервер хранит фрагмент DIB.

DIB содержит два типа информации:

– пользовательская — информация, предоставляемая пользователям и, быть может, изменяемая ими;

– административная и функциональная — информация, используемая для администрирования и/или функционирования каталога.

Множество записей, представленных в DIB, организовано иерархически в структуру дерева, известную как информационное дерево каталога (DIT).

Каждый атрибут, хранящийся в каталоге LDAP, имеет определенный синтаксис (например, тип данных), который накладывает ограничения на структуру и формат его значений. Сравнение значений не является частью определения синтаксиса, а задается отдельно определяемыми правилами соответствия. Правила соответствия специфицируют аргумент, значение утверждения, которое также имеет определенный синтаксис.

6.4.3. Аутентификация пользователей

Службы каталогов LDAP могут быть использованы в качестве источника данных для базовых системных сервисов на базе механизмов NSS и PAM.

В результате вся служебная информация пользователей сети может располагаться на выделенном сервере в распределенной гетерогенной сетевой среде. Добавление новых сетевых пользователей в этом случае производится централизованно на сервере службы каталогов. Сетевые сервисы, поддерживающие возможность аутентификации пользователей (Web, FTP, почта), могут вместо локальных учетных записей использовать тот же каталог LDAP проверки аутентификационной информации. Администратор сети может централизованно управлять конфигурацией сети, в т. ч. разграничивать доступ к сетевым сервисам.

Благодаря предоставлению информации LDAP в иерархической древовидной форме разграничение доступа в рамках службы каталогов LDAP может быть основано на введении доменов. В качестве домена в данном случае будет выступать поддерево службы каталогов LDAP. Сервисы LDAP позволяют разграничивать доступ пользователей к разным поддеревьям каталога, хотя по умолчанию в ОС реализуется схема одного домена.

6.5. Доверенная аутентификация Kerberos

Kerberos является протоколом, обеспечивающим централизованную идентификацию пользователей и применяющим техническое маскирование данных для противодействия различным видам атак.

Основным компонентом системы Kerberos является центр распределения ключей (KDC). Программы, настроенные на взаимодействие с Kerberos, называются «керберизо-

ванными приложениями». KDC отвечает за аутентификацию в некоторой области Kerberos. В процессе работы система Kerberos выдает билеты (tickets) на использование различных служб.

Сервером Kerberos называется компьютер, на котором выполняется серверная программа Kerberos, или сама программа KDC. Клиент Kerberos — это компьютер или программа, которые получают билет от сервера Kerberos. Обычно действия системы Kerberos инициирует пользователь, отправляющий запрос на получение услуг от некоторого сервера приложения (например, сервера почты). Kerberos предоставляет билеты принципалам, в роли которых выступают пользователи или серверные программы. Для описания принципала применяется идентификатор, состоящий из трех компонентов: основы (`primary`), экземпляра (`instance`) и области (`realm`). Данный идентификатор имеет вид:

`основа/экземпляр@область`

Система Kerberos выполняет следующие задачи:

1) обеспечение аутентификации в сети. Для предотвращения несанкционированного доступа к службам сервер должен иметь возможность идентифицировать пользователей. Кроме того, в некоторых средах важно, чтобы клиент мог идентифицировать серверы. Это исключит работу пользователей с фальшивыми серверами, созданными для незаконного сбора конфиденциальной информации;

2) защиту паролей. Открытость паролей, используемых в ряде сетевых служб, создает угрозу безопасности системы, т. к. они могут быть перехвачены и использованы для незаконного доступа к системе. Для решения данной проблемы используется техническое маскирование билетов Kerberos.

Технология Kerberos представляет собой механизм аутентификации пользователей и сервисов, основным достоинством которой является повышенная защищенность при использовании в сети, которая достигается механизмом защищенного обмена билетами между пользователями, сервисами и сервером учетных записей Kerberos. При данном механизме пароли пользователей по сети не передаются, что обеспечивает повышенную защищенность от сетевых атак. С помощью механизма открытых и закрытых ключей, а также синхронизации часов клиентских компьютеров с сервером Kerberos обеспечивается уникальность билетов и их защищенность от подделки.

В ОС используется реализация MIT Kerberos;

3) обеспечение однократной регистрации в сети. Система Kerberos дает возможность пользователю работать с сетевыми сервисами, пройдя лишь единожды аутентификацию на своем компьютере. При этом для обмена с приложениями дополнительно вводить пароль не требуется.

Локальные системы учетных записей пользователей и система ЕПП существуют в ОС параллельно. Различие между ними проводится с помощью разграничения диапазонов UID (значения UID меньшие, чем 2500, относятся к локальным пользователям, а большие или равные 2500 — к пользователям ЕПП).

6.6. Централизация хранения атрибутов СЗИ в распределенной сетевой среде

В среде ОС пользователю поставлен в соответствие ряд атрибутов, характеризующих его мандатные права. Концепция ЕПП подразумевает хранение системной информации о пользователе (в т. ч. и его права МАС) централизованно. В данном случае вся информация хранится в службе каталогов LDAP.

Информация о мандатных атрибутах пользователей хранится локально в соответствующих конфигурационных файлах. При изменении конфигурации системы для использования в сетевом контексте мандатные права пользователей должны переместиться вслед за окружением пользователя (идентификаторы пользователей, групп, домашние каталоги и пр.) в службу каталогов LDAP. Доступ к мандатным атрибутам пользователей осуществляется с использованием программной библиотеки `parsec`. Данная библиотека получает из соответствующего конфигурационного файла информацию об источнике данных для мандатных СЗИ системы. По умолчанию используются локальные текстовые файлы. При работе ОС в сетевом контексте в качестве источника данных выступает служба каталогов LDAP. Переключение контекста производится путем правки соответствующего конфигурационного файла.

6.7. Служба Astra Linux Directory

Служба Astra Linux Directory (ALD) представляет собой систему управления ЕПП.

Таким образом, ALD является надстройкой над технологиями LDAP, Kerberos 5, CIFS и обеспечивает автоматическую настройку всех необходимых файлов конфигурации служб, реализующих перечисленные технологии, а так же предоставляет интерфейс управления и администрирования.

Все необходимые компоненты службы ALD входят в состав следующих пакетов:

– `ald-client` — клиентская часть ALD. Содержит утилиту конфигурирования клиентского компьютера `ald-client` и утилиту автоматического обновления пользовательских билетов `-renew-tickets`. Пакет должен устанавливаться на все клиентские компьютеры, входящие в домен;

– `ald-admin` — содержит утилиту `ald-admin` и утилиту администрирования БД ALD. Пакет должен устанавливаться на компьютеры, с которых будет осуществляться администрирование БД ALD. При установке данного пакета также устанавливается клиентская часть;

– ald-server — серверная часть ALD. Содержит утилиту конфигурации сервера ald-init. Пакет должен устанавливаться на сервер домена. При установке данного пакета также устанавливается ald-admin и, соответственно, клиентская часть.

В руководстве man подробно описаны все возможности указанных утилит.

Для поддержки централизации хранения атрибутов СЗИ в распределенной сетевой среде предназначены дополнительные пакеты ALD, первая часть наименования которых соответствует одному из основных пакетов:

– ald-client-parsec — расширение, необходимое клиентской части ALD;

– ald-admin-parsec — расширение утилиты администрирования БД ALD.

– ald-server-parsec — расширение, необходимое для организации хранения атрибутов СЗИ на сервере ALD;

Без установки пакетов расширения совместно с соответствующими основными пакетами невозможна централизация хранения атрибутов СЗИ в распределенной сетевой среде, что может привести к невозможности входа пользователей в систему.

В состав ОС входит графическая утилита fly-admin-ald, которая позволяет администратору произвести управление ЕПП в графическом режиме (см. электронную справку).

6.7.1. Настройка

Настройка всех компонентов ALD осуществляется автоматически утилитами конфигурирования.

Настройки сервера и клиентов ALD содержатся в файле /etc/ald/ald.conf.

После изменения данного файла необходимо выполнить команду commit-config для того, чтобы изменения вступили в силу:

```
ald-init commit-config (на сервере)
ald-client commit-config (на клиентах)
```

Формат файла:

```
ИМЯ_ПАРАМЕТРА=значение # Комментарий
```

В файле для системы ALD задаются следующие параметры:

– VERSION — для текущей версии должно быть установлено значение 1.3;

– DOMAIN — имя домена. Должно быть задано в формате:

```
.example.ru
```

для сервера ALD. Если данный параметр меняется, то необходимо заново инициализировать сервер командой:

```
ald-init init
```

Можно также воспользоваться командами:

```
ald-init backup-ldif
ald-init restore-backup-ldif
```

для переименования домена;

– SERVER — полное имя серверного компьютера ALD.

Пример

```
my-server.example.ru
```

MINIMUM_UID

Минимальный номер глобального пользователя. Пользователи с номером меньше данного считаются локальными и аутентифицируются через локальные файлы /etc/passwd и /etc/shadow.

Примечание. Для нормальной работы домена не рекомендуется пересечение по номерам локальных и глобальных пользователей и групп. Не рекомендуется задавать MINIMUM_UID меньше 1000;

– TICKET_MAX_LIFE=10h — максимальное время жизни билета Kerberos (если его не обновлять). Формат параметра: NNd (дни), или NNh (часы), или NNm (минуты).

При входе в домен пользователь получает билет. При выходе из домена билет уничтожается. Если билет не обновлять, то после истечения срока действия билета пользователь потеряет доступ к своему домашнему каталогу. Чтобы восстановить доступ, ему придется выполнить команду kinit или зайти в систему заново. Чтобы доступ не был потерян, билет следует периодически обновлять (до истечения срока действия). Настроить автоматическое обновление можно с помощью утилиты ald-renew-ticket.

Для удобства можно настроить данный параметр на большое количество времени, например 30d. Но это менее безопасно;

– TICKET_MAX_RENEWABLE_LIFE=7d — максимальное обновляемое время жизни билета Kerberos. Формат параметра: NNd (дни), или NNh (часы), или NNm (минуты).

По истечении данного срока билет не может быть обновлен. Данный параметр должен быть больше, чем параметр TICKET_MAX_LIFE.

Примечание. Для клиентских компьютеров параметры TICKET_MAX_LIFE и TICKET_MAX_RENEWABLE_LIFE определяются как наименьшие значения этих параметров, заданных в файлах ald.conf на сервере и на клиентском компьютере;

– NETWORK_FS_TYPE — определяет, какая сетевая ФС будет использоваться для глобальных пользовательских домашних каталогов. Возможные значения:

  – none — сетевая ФС не используется. Работает только аутентификация глобальных пользователей. Используются локальные домашние каталоги пользователей. (Следующие параметры, относящиеся к сетевой ФС, игнорируются);

  – cifs — используется Samba/CIFS;

– SERVER_EXPORT_DIR — (только для сервера). Задает абсолютный путь к каталогу на сервере, где будет располагаться хранилище домашних каталогов. Данный каталог будет экспортирован по Samba/CIFS;

– CLIENT_MOUNT_DIR — задает абсолютный путь к точке монтирования хранилища домашних каталогов на клиентских компьютерах;

– SERVER_FS_KRB_MODES — (только для сервера). Задает режимы экспорта сервера Samba/CIFS (перечисленные через запятую). Возможные режимы:

  – krb5 — только Kerberos-аутентификация;

  – krb5i — (integrity) аутентификация и проверка целостности (подпись) пакетов.

Должен быть указан хотя бы один режим;

– CLIENT_FS_KRB_MODE — задает Kerberos-режим монтирования на клиентском компьютере. Должен быть указан один из режимов: krb5 или krb5i;

– SERVER_ON — включает/выключает сервер. Присвоенное значение может быть 0 или 1.

Если на клиентском компьютере SERVER_ON=0, это аналогично CLIENT_ON=0.

Если на сервере SERVER_ON=0, то:

  – домашние каталоги не экспортируются;

  – разрешение имен по LDAP выключается в nsswitch.conf;

  – все принципалы Kerberos деактивируются (allow_tickets=0);

  – службы LDAP, Samba, Kerberos, nss-ldapd останавливаются;

  – служба nscd перезапускается;

– CLIENT_ON — включает/выключает клиентскую часть ALD. Присвоенное значение может быть 0 или 1.

Если CLIENT_ON=0, то:

  – домашние каталоги не монтируются;

  – разрешение имен по LDAP выключается в nsswitch.conf;

  – служба nscd перезапускается.

Пример файла /etc/ald/ald.conf:

```
VERSION=1.3
DOMAIN=.example.ru
SERVER=my-server.example.ru
MINIMUM_UID=2500
TICKET_MAX_LIFE=10h
TICKET_MAX_RENEWABLE_LIFE=7d
NETWORK_FS_TYPE=cifs
```

```
SERVER_EXPORT_DIR=/ald_export_home
CLIENT_MOUNT_DIR=/ald_home
SERVER_FS_KRB_MODES=krb5,krb5i
CLIENT_FS_KRB_MODE=krb5i
SERVER_ON=1
CLIENT_ON=1
```

6.8. Настройка сетевых служб

Ряд сетевых служб, таких как СУБД PostgreSQL, электронная почта, обработка гипертекстовых документов (web), система печати и др. для работы в ЕПП должны быть соответствующим образом настроены. Как правило, настройка заключается в обеспечении возможности использования этими службами сквозной аутентификации и получения необходимой информации из БД ALD.

Примечание. При выполнении настройки сетевых служб потребуется использование учетной записи суперпользователя `root`. Не рекомендуется осуществлять переключение в режим суперпользователя командой `su`. Необходимо использовать команду:

```
# su -
```

6.8.1. СУБД PostgreSQL

Для работы СУБД PostgreSQL с ALD необходимо выполнение следующих условий:

1) наличие в системах, на которых функционируют сервер и клиенты СУБД PostgreSQL, установленного пакета клиента ALD — `ald-client`;

2) разрешение имен должно быть настроено таким образом, чтобы имя системы разрешалось, в первую очередь, как полное имя (например, `myserver.example.ru`);

3) клиент ALD должен быть настроен на используемый ALD домен (см. 6.7.1).

Для проведения операций по настройке ALD и администрированию Kerberos необходимо знание паролей администраторов ALD и Kerberos.

6.8.1.1. Сервер

Для обеспечения совместной работы сервера СУБД PostgreSQL с ALD необходимо, чтобы сервер СУБД PostgreSQL функционировал как сервис Kerberos. Выполнение данного условия требует наличия в БД Kerberos принципала для сервера СУБД PostgreSQL, имя которого задается в формате:

```
servicename/hostname@realm
```

где имя сервиса `servicename` соответствует имени учетной записи пользователя, от которой осуществляется функционирование сервера СУБД PostgreSQL (по умолчанию — `postgres`), и указывается в конфигурационном файле сервера PostgreSQL как значение параметра `krb_srvname`. В качестве значения `hostname` указывается полное доменное

имя системы, на которой функционирует сервер СУБД PostgreSQL, а в качестве значения realm — имя домена ALD.

Таким образом, для обеспечения совместной работы сервера СУБД PostgreSQL с ALD необходимо:

1) создать в БД ALD с помощью утилиты администрирования ALD принципала, соответствующего устанавливаемому серверу PostgreSQL. Принципал создается с автоматически сгенерированным случайным ключом:

```
ald-admin service-add postgres/server.my_domain.org
```

2) ввести созданного принципала в группу сервисов mac, используя следующую команду:

```
ald-admin sgroup-svc-add postgres/server.my_domain.org --sgroup=mac
```

3) создать файл ключа Kerberos для сервера СУБД PostgreSQL с помощью утилиты администрирования ALD ald-client, используя следующую команду (пример приведен для кластера БД по умолчанию):

```
ald-client update-svc-keytab postgres/server.my_domain.org
   --ktfile="/etc/postgresql/8.4/main/krb5.keytab"
```

Полученный файл должен быть доступен серверу СУБД PostgreSQL по пути, указанному в конфигурационном параметре krb_server_keyfile (в данном случае — /etc/postgresql/8.4/main/krb5.keytab). Права доступа к этому файлу должны позволять читать его пользователю, от имени которого работает сервер СУБД PostgreSQL (как правило, владельцем файла назначается пользователь postgres);

4) сменить владельца полученного на предыдущем шаге файла krb5.keytab на пользователя posgtres, выполнив следующую команду:

```
chown postgres /etc/postgresql/8.4/main/krb5.keytab
```

5) задать в конфигурационном файле сервера СУБД PostgreSQL /etc/postgreslq/8.4/main/postgresql.conf для приведенных ниже параметров соответствующие значения:

```
krb_server_keyfile = '/etc/postgresql/8.4/main/krb5.keytab'
krb_srvname = 'postgres'
```

6) указать для внешних соединений в конфигурационном файле сервера СУБД PostgreSQL /etc/postgreslq/8.4/main/pg_hba.conf метод аутентификации gss.

Пример

```
host all all 192.168.32.0/24 gss
```

6.8.1.2. Клиент

Общие условия, при которых обеспечивается совместное функционирование клиентов СУБД PostgreSQL с ALD, см. в 6.8.1. Кроме того, сервер СУБД PostgreSQL должен быть также настроен соответствующим образом (см. 6.8.1.1). Для настройки клиента СУБД PostgreSQL необходимо:

1) создать в БД ALD учетную запись пользователя, зарегистрированного в СУБД PostgreSQL (например, `pgusername`). Дополнительная информация приведена в руководстве `man` на ALD;

2) задать в качестве значения параметра соединения `krbsrvname` имя сервиса `servicename`, использованное при создании принципала сервера СУБД PostgreSQL. Имя принципала сервера СУБД PostgreSQL задается в формате:

`servicename/hostname@realm`

где `servicename` — обычно имя учетной записи сервера СУБД PostgreSQL, использованное при создании принципала сервера СУБД PostgreSQL (по умолчанию — `postgres`), а `hostname` — полное доменное имя системы, на которой функционирует сервер СУБД PostgreSQL.

6.8.2. Система обмена сообщениями электронной почты

Для обеспечения совместной работы системы обмена сообщениями электронной почты (СЭП) с ALD предлагается использовать ее в следующем составе:

– агент передачи сообщений СЭП (MTA) — Exim 4, установленный из пакета `exim4-daemon-heavy`;

– агент доставки сообщений СЭП (MDA) — Dovecot, установленный из пакета `dovecot-imapd`;

– клиент СЭП (MUA) — Mozilla Thunderbird, установленный из пакета `thunderbird`.

Предложенная конфигурация СЭП предоставляет возможность организации совместной работы с ALD с использованием для аутентификации пользователей посредством Kerberos метода GSSAPI на основе встроенного в Dovecot сервера SASL.

Для обеспечения совместной работы СЭП, состоящей из перечисленных выше средств, с ALD необходимо выполнение следующих условий:

1) наличие в системах, на которых функционируют MTA, MDA и MUA, установленного пакета клиента ALD — `ald-client`;

2) разрешение имен должно быть настроено таким образом, чтобы имя системы разрешалось, в первую очередь, как полное имя (например, `myserver.example.ru`);

3) клиент ALD должен быть настроен на используемый ALD домен (см. 6.7.1);

4) в процессе установки MTA Exim 4 необходимо указать, что для хранения сооб-

щений электронной почты должен быть использован формат `Maildir` в домашнем каталоге и конфигурация разделена на небольшие файлы.

Для проведения операций по настройке ALD и администрированию Kerberos необходимо знание паролей администраторов ALD и Kerberos.

6.8.2.1. Сервер

Для обеспечения работы сервера СЭП, включающего MDA Dovecot, установленный из пакета `dovecot-imapd` и настроенный (13.3.1), и MTA Exim 4, установленный из пакета `exim4-daemon-heavy` и настроенный (13.3.2), необходимо:

1) создать в БД ALD с помощью утилиты администрирования ALD принципала, соответствующего установленному MDA Dovecot. Принципал создается с автоматически сгенерированным случайным ключом:

```
ald-admin service-add imap/server.my_domain.org
```

2) ввести созданного принципала в группу сервисов `mac`, используя следующую команду:

```
ald-admin sgroup-svc-add imap/server.my_domain.org --sgroup=mac
```

3) ввести созданного принципала в группу сервисов `mail`, используя следующую команду:

```
ald-admin sgroup-svc-add imap/server.my_domain.org --sgroup=mail
```

4) создать файл ключа Kerberos для MDA Dovecot с помощью утилиты администрирования ALD `ald-client`, используя следующую команду:

```
ald-client update-svc-keytab imap/server.my_domain.org
   --ktfile="/var/lib/dovecot/dovecot.keytab"
```

5) создать в БД Kerberos принципала, соответствующего установленному MTA Exim 4. Принципал создается с автоматически сгенерированным случайным ключом:

```
ald-admin service-add smtp/server.my_domain.org
```

6) ввести созданного принципала в группу сервисов `mac`, используя следующую команду:

```
ald-admin sgroup-svc-add smtp/server.my_domain.org --sgroup=mac
```

7) ввести созданного принципала в группу сервисов `mail`, используя следующую команду:

```
ald-admin sgroup-svc-add smtp/server.my_domain.org --sgroup=mail
```

8) создать файл ключа Kerberos для MTA Exim 4 с помощью утилиты администрирования ALD `ald-client`, используя следующую команду:

```
ald-client update-svc-keytab smtp/server.my_domain2.org
  --ktfile="/var/lib/dovecot/dovecot.keytab"
```

9) в конфигурационном файле `/etc/dovecot/dovecot.conf` отключить использование протоколов POP3, установив:

```
protocols = imap
```

10) в конфигурационном файле /etc/dovecot/dovecot.conf установить:

```
auth_krb5_keytab = /var/lib/dovecot/dovecot.keytab
```

11) для отключения передачи при аутентификации пароля открытым текстом в конфигурационном файле /etc/dovecot/dovecot.conf установить:

```
disable_plaintext_auth = yes
```

12) в конфигурационном файле /etc/dovecot/dovecot.conf отключить поддержку SSL/TLS установив значение no для параметра ssl:

```
ssl = no
```

13) для настройки аутентификации посредством Kerberos с использованием метода GSSAPI в секции auth default конфигурационного файла /etc/dovecot/dovecot.conf установить:

```
mechanisms = gssapi
```

14) для настройки встроенного в Dovecot сервера SASL, к которому будет обращаться MTA Exim 4, в секции auth default конфигурационного файла /etc/dovecot/dovecot.conf установить:

```
socket listen {
client {
# The client socket is generally safe to export to everyone.
# Typical use is to export it to your SMTP server
# so it can do SMTP AUTH lookups using it.
path = /var/run/dovecot/auth-client
user = Debian-exim
mode = 0600
}
}
```

15) перезапустить MDA Dovecote, выполнив команду:

```
service dovecot restart
```

16) для настройки аутентификации пользователей в MTA Exim 4 посредством Kerberos с использованием метода GSSAPI и встроенного в Dovecot сервера SASL создать конфигурационный файл /etc/exim4/conf.d/auth/33_exim4-dovecot-kerberos-ald со следующим содержимым:

```
dovecot_gssapi:
driver = dovecot
public_name = GSSAPI
server_socket = /var/run/dovecot/auth-client
server_set_id = $auth1
```

17) для запрета отправки писем без аутентификации создать конфигурационный файл `/etc/exim4/conf.d/acl/30_exim4-config_check_rcpt_auth_req` со следующим содержимым:

```
deny message = "Auth required"
hosts = *:+relay_from_hosts
!authenticated = *
```

18) перезапустить MTA Exim 4, выполнив команду:

```
/etc/init.d/exim4 reload
```

Для работы сервера СЭП необходимо установить корректный порядок запуска сервисов (13.3.3).

6.8.2.2. Клиент

Для обеспечения возможности работы MUA Mozilla Thunderbird с ALD необходимо:

1) создать в ALD учетную запись пользователя при помощи команды:

```
ald-admin user-add user1
```

2) при создании учетной записи пользователя СЭП в MUA Mozilla Thunderbird необходимо выбрать тип используемого сервера входящей почты IMAP;

3) при настройке учетной записи установить в параметре «Защита соединения» для сервера и сервера исходящей значение «Нет»;

4) при настройке учетной записи установить в параметрах сервера и параметрах сервера исходящей почты использование метода аутентификации «Kerberos / GSSAPI».

6.8.3. Web-сервер Apache 2.2

Для обеспечения совместной работы web-сервера Apache 2.2 с ALD необходимо:

– наличие в системе, на которой функционирует web-сервер, установленного пакета клиента ALD — `ald-client`;

– разрешение имен должно быть настроено таким образом, чтобы имя системы разрешалось, в первую очередь, как полное имя (например, `myserver.example.ru`);

– клиент ALD должен быть настроен на используемый ALD домен (см. 6.7.1);

– в системе должен быть установлен модуль web-сервера Apache 2.2 `auth_kerb` из пакета `libapache2-mod-auth-kerb`.

Наличие модуля web-сервера Apache 2.2 `auth_kerb` предоставляет возможность организации совместной работы с ALD с использованием для аутентификации пользователей посредством Kerberos метода GSSAPI.

Для проведения операций по настройке ALD и администрированию Kerberos необходимо знание паролей администраторов ALD и Kerberos.

Для обеспечения возможности работы web-сервера Apache 2.2 с ALD необходимо:

1) активировать модуль web-сервера Apache 2.2 `auth_kerb` при помощи команды:

```
a2enmod auth_kerb
```

2) в конфигурационных файлах виртуальных хостов web-сервера Apache 2.2 для областей, требующих авторизации, указать:

```
AuthType Kerberos
KrbAuthRealms REALM
KrbServiceName HTTP/server.my_domain.org
Krb5Keytab /etc/apache2/keytab
KrbMethodNegotiate on
KrbMethodK5Passwd off
require valid-user
```

3) создать в БД ALD с помощью утилиты администрирования ALD принципала, соответствующего настраиваемому web-серверу Apache. Принципал создается с автоматически сгенерированным случайным ключом:

```
ald-admin service-add HTTP/server.my_domain.org
```

4) ввести созданного принципала в группу сервисов `mac`, используя следующую команду:

```
ald-admin sgroup-svc-add HTTP/server.my_domain.org --sgroup=mac
```

5) создать файл ключа Kerberos для web-сервера Apache с помощью утилиты администрирования ALD `ald-client`, используя следующую команду:

```
ald-client update-svc-keytab HTTP/server.my_domain.org
   --ktfile="/etc/apache2/keytab"
```

Полученный файл должен быть доступен web-серверу Apache по пути, указанному в конфигурационном параметре `Krb5Keytab` (в данном случае — `/etc/apache2/keytab`). Права доступа к этому файлу должны позволять читать его пользователю, от имени которого работает web-сервер Apache (как правило, владельцем файла назначается пользователь `www-data`);

6) сменить владельца полученного на предыдущем шаге файла `keytab` на пользователя `www-data`, выполнив следующую команду:

```
chown www-data /etc/apache2/keytab
```

7) сделать файл `/etc/apache2/keytab` доступным на чтение для остальных пользователей:

```
chmod 644 /etc/apache2/keytab
```

8) перезапустить web-сервер Apache, выполнив команду:

```
/etc/init.d/apache2 restart
```

Браузер пользователя должен поддерживать аутентификацию `negotiate`. В последних версиях браузера Konqueror данная поддержка присутствует автоматически. В

браузере Mozilla Firefox в настройках, доступных по адресу `about:config`, необходимо указать, для каких серверов доступна аутентификация `negotiate`. Для выполнения данной настройки необходимо задать маски доменов или в общем случае http- и https-соединения в качестве значений параметра `network.negotiate-auth.trusted-uris`, вставив, например, значение `http://`, `https://`.

6.8.4. Система печати

Для работы системы печати необходимо выполнение следующих условий:

1) наличие в системах, на которых функционируют сервер и клиенты системы печати, установленного пакета клиента ALD — `ald-client`;

2) разрешение имен должно быть настроено таким образом, чтобы имя системы разрешалось, в первую очередь, как полное имя (например, `myserver.example.ru`);

3) клиент ALD должен быть настроен на используемый ALD домен (см. 6.7.1).

Для проведения операций по настройке ALD и администрированию Kerberos необходимо знание паролей администраторов ALD и Kerberos.

6.8.4.1. Сервер системы печати

Для обеспечения совместной работы сервера печати с ALD необходимо, чтобы сервер печати функционировал как сервис Kerberos. Выполнение данного условия требует наличия в БД Kerberos принципала для сервера печати, имя которого задается в формате: `servicename/hostname@realm`

Таким образом, для обеспечения совместной работы сервера печати с ALD необходимо:

1) создать в БД ALD с помощью утилиты администрирования ALD принципала, соответствующего серверу печати. Принципал создается с автоматически сгенерированным случайным ключом:

```
ald-admin service-add ipp/server.my_domain
```

2) ввести созданного принципала в группу сервисов `mac`, используя следующую команду:

```
ald-admin sgroup-svc-add ipp/server.my_domain --sgroup=mac
```

3) создать файл ключа Kerberos для сервера печати с помощью утилиты администрирования ALD `ald-client`, используя следующую команду:

```
ald-client update-svc-keytab ipp/server.my_domain
```

4) от имени суперпользователя выполнить следующие команды:

```
cupsctl --remote-admin --remote-printers --remote-any
cupsctl ServerAlias=*
cupsctl DefaultPolicy=default
cupsctl MarkerUser=ipp
```

```
cupsctl Port=631
cupsctl ServerName=server.my_domain
cupsctl MacEnable=On
cupsctl DefaultAuthType=Negotiate
```

5) задать в конфигурационном файле сервера системы печати /etc/cups/cupsd.conf для политики <Policy parsec> в секции <Limit Send-Document Send-URI> для параметра AuthType значение None:

```
<Limit Send-Document Send-URI>
  AuthType None
  Require user @OWNER @SYSTEM
  Order deny,allow
    </Limit>
```

6) осуществить перезапуск сервера системы печати, выполнив команды:

```
service cups stop
service cups start
```

Пользователь, от имени которого будут выполняться команды по маркировке, должен входить в группу lpmac. В его домашнем каталоге необходимо создать файл .bashrc со следующим содержимым:

```
id > /dev/null
```

Далее приведен пример создания файла для пользователя lpadmin:

```
echo "id > /dev/null" > /ald_export_home/lpadmin/.bashrc
```

Подробная информация по маркировке документов приведена в 10.2. Информация о печати нескольких копий документов с ненулевым мандатным уровнем приведена в 10.3.

Далее выполнить вход на сервере печати от имени учетной записи, входящей в группу ALD lpadmin, и настроить принтеры. Настройка принтеров может быть выполнена с использованием утилиты fly-admin-printer (см. электронную справку). После запуска утилиты необходимо указать, что для выполнения привилегированных действий не используется учетная запись root, и затем выполнять действия по настройке.

Дополнительная информация по системе печати приведена в разделе 10.

6.8.4.2. Клиент системы печати

Общие условия, при которых обеспечивается совместное функционирование клиентов системы печати с ALD, см. в 6.8.4. Кроме того, сервер печати должен быть также настроен соответствующим образом (см. 6.8.4.1). Для настройки клиента системы печати необходимо:

1) создать конфигурационный файл /etc/cups/client.conf;

2) задать в конфигурационным файле /etc/cups/client.conf для параметра ServerName в качестве значения имя сервера системы печати, например

server.my_domain.

7. ЗАЩИЩЕННАЯ ГРАФИЧЕСКАЯ ПОДСИСТЕМА

7.1. Общие сведения

Защищенная графическая подсистема в составе ОС функционирует с использованием графического сервера Xorg.

В подсистему входит также рабочий стол Fly, который состоит из оконного менеджера и большого набора графических утилит как пользовательских, так и административных.

По умолчанию в графическую подсистему встроена мандатная защита.

Для установки пакетов графической подсистемы следует в процессе работы программы установки ОС отметить в окне «Выбор программного обеспечения» строку «Рабочий стол Fly». В этом случае рабочий стол Fly установится с настройками по умолчанию, и в процессе загрузки установленной системы после окончания работы системного загрузчика произойдет переход к графической программе «Вход в систему» (`fly-dm`). После завершения процедуры аутентификации на экране монитора появится графический рабочий стол.

7.2. Рабочий стол Fly

В состав рабочего стола Fly входит оконный менеджер и большое количество графических утилит, которые могут быть использованы для администрирования ОС. Большинство из этих утилит представляет собой графические оболочки над соответствующими текстовыми утилитами командной строки.

Основные графические утилиты для администратора системы:

– «Управление ЕПП» (`fly-admin-ald`) — управление БД ALD;

– «Панель управления» (`fly-admin-center`) — панель управления;

– «Планировщик задач» (`fly-admin-cron`) — планировщик задач;

– «Дата и время» (`fly-admin-date`) — установка даты и времени;

– «Менеджер устройств» (`fly-admin-device-manager`) — управление некоторыми системными устройствами;

– «Настройка DHCP-сервера» (`fly-admin-dhcp`) — настройка сервера DHCP;

– «Вход в систему» (`fly-admin-dm`) — настройка входа в систему;

– «Переменные окружения» (`fly-admin-env`) — редактирование переменных окружения;

– «Сетевой фильтр» (`fly-admin-firewall`) — настройка сетевого фильтра;

– «FTP» (`fly-admin-ftp`) — настройка сервера FTP;

– «Коррекция гаммы» (`fly-admin-gamma`) — установка цветового баланса монитора;

– «Загрузчик GRUB» (`fly-admin-grubeditor`) — настройка загрузчика ОС;

– «Проверка целостности системы» (`fly-admin-int-check`) — проверка целостности системы;

– «Киоск» (`fly-admin-kiosk`) — создание и настройка профилей для режима kiosk;

– «Настройка NTP» (`fly-admin-ntp`) — настройка сервера времени и управление им;

– «Менеджер пакетов» (`fly-admin-package`) — управление программными пакетами;

– «Санкции Policykit» (`fly-admin-policykit`) — настройка полномочий Policykit;

– «Средства администрирования СУБД» (`fly-admin-postgres`) — управление СУБД PostgreSQL;

– «Управление питанием» (`fly-admin-power`) — управление системой питания;

– «Менеджер печати» (`fly-admin-printer`) — настройка сервера печати, а также локальных и сетевых принтеров;

– «Обработка подключения устройств» (`fly-admin-reflex`) — настройка службы обработки «горячего» подключения устройств;

– «Уровни запуска» (`fly-admin-runlevel`) — настройка уровней запуска сервисов ОС;

– «Общие папки Samba» (`fly-admin-samba`) — настройка сервера Samba;

– «Сканеры (настройка сканеров)» (`fly-admin-scan`) — проверка установленных в системе сканеров;

– «Управление локальной политикой безопасности» (`fly-admin-smc`) — управление локальной политикой безопасности (управление протоколированием, привилегиями и мандатными атрибутами пользователей, работа с пользователями и группами);

– «Журнал безопасности» (`fly-admin-viewaudit`) — поиск и просмотр записей системы протоколирования;

– «Доступ к X-серверу» (`fly-admin-xhost`) — настройка удаленного доступа к X-серверу;

– «Системные альтернативы» (`fly-alternatives`) — управление системными альтернативами;

– «Менеджер шрифтов» (`fly-admin-fonts`) — управление шрифтами;

– «Редактор горячих клавиш» (`fly-hotkeys`) — установка клавиш быстрого доступа;

– «Приложения для типов файлов» (`fly-mimeapps-service`) — установка приложений для типов файлов;

– «Менеджер обновлений» (`fly-update-notifier`) — проверка наличия обновлений ОС.

Описание утилит приведено в электронной справке.

7.3. Мандатное разграничение доступа

Мандатная защита, встроенная в рабочий стол Fly, позволяет администратору устанавливать отдельно для каждого пользователя разрешенный диапазон мандатных уровней и категорий. Для этой цели следует использовать графическую утилиту `fly-admin-smc`.

После того, как пользователь, для которого установлены возможные мандатные уровни и категории, отличные от нуля, войдет в систему, ему будет предложено установить конкретный мандатный уровень и конкретную категорию для данной сессии в пределах разрешенных диапазонов. Выбранные значения этих параметров можно будет проверить с помощью индикатора в виде кружка с числом внутри, расположенного в системном лотке в правом нижнем углу рабочего стола (рис. 1). Для получения информационного сообщения следует навести курсор на этот индикатор.

Рис. 1

8. УПРАВЛЕНИЕ ПРОГРАММНЫМИ ПАКЕТАМИ

В ОС используются программные пакеты (далее по тексту — пакеты) в формате `.deb`. Для управления пакетами в режиме командной строки (или в эмуляторе терминала в графическом режиме) предназначены набор команд нижнего уровня `dpkg` и комплекс программ высокого уровня `apt` (для которого имеется текстовая псевдографическая оболочка `aptitude`). В графическом режиме управлять пакетами можно с помощью утилиты `fly-admin-package` (рабочий стол Fly), программ `Kpackage` (рабочий стол KDE) и `synaptic` (универсальная графическая оболочка для `apt`).

По умолчанию обычный пользователь не имеет права использовать эти инструменты. Для всех операций с пакетами (за исключением некоторых случаев получения информации о пакетах) необходимы права администратора (суперпользователя).

8.1. Набор команд dpkg

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

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

Если пакет (например, `iptables_1.4.2-6_amd64.deb`), который необходимо установить, помещен в рабочий каталог (например, `/home/user1`) или находится на смонтированном внешнем носителе, следует выполнить следующую команду:

```
dpkg -i /home/user1/iptables_1.4.2-6_amd64.deb
```

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

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

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

Для приведенного выше примера команда будет выглядеть следующим образом:

```
dpkg -r iptables
```

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

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

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

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

8.2. Комплекс программ apt

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

8.2.1. Настройка доступа к архивам пакетов

Информация о сетевых и локальных архивах пакетов для комплекса программ apt содержится в файле /etc/apt/sources.list. В этом файле находится список источников пакетов, который используется программами для определения местоположения архивов. Список источников разрабатывается для поддержки любого количества активных источников и различных видов этих источников. В данном файле перечисляется по одному источнику на строку, где источники следуют в порядке убывания их приоритета.

Пример файла sources.list:

```
deb cdrom:[OS Astra Linux 1.1 smolensk - amd64 DVD]/ smolensk main contrib
  non-free
deb ftp://ftp-server.cct.rbt/distr/astra/stable/ smolensk main contrib
  non-free
deb file:/home/user/distr/astra/stable/ smolensk main contrib non-free
deb http://web-server.cct.rbt/distr/astra/stable/ smolensk main contrib
  non-free
```

При установке ОС с дистрибутива строка deb cdrom... автоматически записывается в этот список.

Включить эту строку в данный список можно также при помощи команды:

```
apt-cdrom add
```

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

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

8.2.2. Установка и удаление пакетов

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

```
apt-cache show iptables
```

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

Обновить содержимое локальной БД можно при помощи команды:

```
apt-get install update
```

Эту операцию необходимо выполнять при каждом изменении как списка источников пакетов, так и содержимого этих источников (например, при переходе к использованию обновленной версии ОС).

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

```
apt-get install upgrade
```

Обновление старой версии ОС до новой (без переустановки) производится при помощи команды:

```
apt-get install dist-upgrade
```

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

```
apt-get install <значимая_часть_имени_пакета>
```

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

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

```
apt-get remove <значимая_часть_имени_пакета>
```

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

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

8.3. Пересмотр прав доступа к файлам

Во время установки пакета права доступа к файлам назначаются автоматически, и установочный сценарий корректно определяет права доступа к каждому файлу пакета. Однако следует пересмотреть их и решить, разрешено ли работать с пакетом тем пользователям, для которых он предназначен, и не может ли злоумышленник воспользоваться им для проникновения в систему.

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.

Примечание. По умолчанию в ОС для всех пользователей заблокирована возможность изменения атрибутов любого неисполняемого файла на исполняемый. Это обеспечивается передачей ядру значения «1» для соответствующих параметров:

```
/proc/sys/fs/user_noacl
```

```
/proc/sys/fs/user_nox
/proc/sys/kernel/user_nosetuid
```

Администратор может снять это ограничение, откорректировав файл `/etc/sysctl.conf`, в котором следует записать следующие строки:

```
fs.user_noacl=0
fs.user_nox=0
kernel.user_nosetuid=0
```

После этого необходимо выполнить команду:

```
/sbin/sysctl  -p  /etc/sysctl.conf
```

При следующей загрузке системы будут использоваться новые значения параметров, и ограничение на изменение атрибутов файла для пользователей будет отменено.

8.4. Удаление приложения

Не всегда достаточно просто стереть с диска файлы приложения и удалить его каталог. Драйверы и другие приложения должны быть корректно отключены во избежание проблем в дальнейшем. Мониторинг системы в процессе установки и ведение журнала установки позволяют корректно удалить приложение, ставшее ненужным.

9. РЕЗЕРВНОЕ КОПИРОВАНИЕ И ВОССТАНОВЛЕНИЕ ДАННЫХ

9.1. Вопросы

Цель резервного сохранения данных заключается в том, чтобы иметь возможность восстановить с минимальными затратами труда и времени отдельные файлы или всю ФС.

С точки зрения администратора системы, процесс резервного сохранения должен быть как можно больше автоматизирован и требовать минимального участия.

При рассмотрении вопросов, связанных с резервным сохранением данных, следует определить:

– полное или обновляемое архивирование;
– резервное сохранение ФС;
– типы носителей для резервного сохранения;
– влияние резервного сохранения на работоспособность системы;
– программы резервного копирования.

9.1.1. Полное и обновляемое архивирование

Полное архивирование означает сохранение всех файлов. Это процедура займет немало времени и огромное количество носителей информации для копирования всего содержимого системы. В большинстве случаев нет необходимости ежедневного выполнения этой процедуры.

При обновляемом архивировании сохраняются только те файлы, которые добавились или изменились со времени последнего сохранения.

9.1.2. Резервное сохранение ФС

Резервное сохранение всех ФС абсолютно необходимо. Естественно, в первую очередь, оно должно касаться тех ФС, которые используются наиболее активно; другие ФС можно резервировать реже.

9.1.3. Выбор носителей для резервного сохранения

В зависимости от имеющихся в распоряжении устройств для резервного сохранения данных можно использовать специально выделенный раздел жесткого диска.

9.1.4. Влияние резервного сохранения на работоспособность системы

Резервное копирование — это процесс, который, как и любой другой, будет увеличивать текущую нагрузку на систему, что может вызывать недовольство пользователей.

9.2. План резервного копирования

Составить план резервного копирования и определить: что именно должно архивироваться, как часто и каким образом будут восстанавливаться файлы при необходимости.

Определить, каким образом пользователи могут запросить ранее сохраненные файлы.

Периодически пересматривать план резервного копирования.

9.2.1. Составление расписания резервного копирования

При составлении расписания резервного сохранения определить: что, когда и на каком носителе должно сохраняться.

Должна существовать возможность восстановления любого файла в любой момент времени. Реально восстановить файл не более, чем однодневной давности. Для этого можно использовать комбинацию полного и обновляемого архивирования. Полное архивирование позволяет сохранить копии всех файлов системы, обновляемое — только изменившиеся со времени последнего архивирования. Обновляемое архивирование может иметь несколько уровней, например обновление по отношению к последнему обновляемому архивированию.

Для восстановления отдельных файлов при таком многоуровневом расписании может понадобиться полный архив, если файл не изменялся в течение месяца; архив первого уровня, если файл не изменялся в течение недели; архив второго уровня при ежедневной работе с этим файлом. Такая схема несколько сложнее, чем в предыдущем примере, однако требует меньших ежедневных затрат времени.

Расписание архивирования следует довести до пользователей.

9.2.2. Проверка архивов

Следует проверять свои архивы. Эта проверка может включать в себя чтение содержимого архива после сохранения или выборочную проверку файлов из архива.

9.2.3. Планирование восстановления системы

Определить план действий на случай аварийной ситуации и то, как при необходимости можно восстановить систему или отдельные файлы, где хранятся и насколько доступны носители с резервными копиями и не могут ли они потерять работоспособность при неполадках на компьютере.

10. СИСТЕМА ПЕЧАТИ

Одним из основных сервисов, предоставляемых ОС, является сервис печати, позволяющий осуществлять печать документов в соответствии с требованиями, предъявляемыми к защищенным ОС.

10.1. Устройство системы печати

В ОС используется система печати CUPS, которая:

– управляет заданиями на печать;

– исполняет административные команды;

– предоставляет информацию о состоянии принтеров локальным и удаленным программам;

– информирует пользователей, если это требуется.

Планировщик — это сервер, который управляет списком доступных принтеров и направляет задания на печать, как требуется, используя подходящие фильтры и выходные буферы (backends).

Файлами конфигурации являются:

– файл конфигурации сервера;

– файлы определения принтеров и классов;

– типы MIME и файлы правил преобразования;

– файлы описания PostScript-принтеров (PPD).

Конфигурационный файл сервера очень похож на файлы конфигурации web-сервера и определяет все свойства управления доступом.

Файлы описания принтеров и классов перечисляют доступные очереди печати и классы. Классы принтеров — наборы принтеров. Задания, посланные классу принтеров, направляются к первому доступному принтеру данного класса.

Очередь печати — механизм, который позволяет буферизовать и организовать задания, посылаемые на принтер. Необходимость организации такого механизма обуславливается тем, что принтер является медленно действующим устройством, и задания не могут быть распечатаны мгновенно. Очевидно, что в многопользовательской среде возникает конкуренция со стороны пользователей при доступе к принтерам, поэтому задания необходимо располагать в очереди. Для этого используется буферный каталог `/var/spool/cups/`.

Файлы типов MIME перечисляют поддерживаемые MIME-типы (`text/plain`, `application/postscript` и т. д.) и правила для автоматического обнаружения формата файла. Они используются сервером для определения поля `Content-Type` для GET- и HEAD-запросов и обработчиком запросов IPP, чтобы определить тип файла.

Правила преобразования MIME перечисляют доступные фильтры. Фильтры используются, когда задание направляется на печать, таким образом, приложение может послать файл удобного (для него) формата системе печати, которая затем преобразует документ в требуемый печатный формат. Каждый фильтр имеет относительную «стоимость», связанную с ним, и алгоритм фильтрования выбирает набор фильтров, который преобразует файл в требуемый формат с наименьшей общей «стоимостью».

Файлы PPD описывают возможности всех типов принтеров. Для каждого принтера имеется один PPD-файл. Файлы PPD для не-PostScript-принтеров определяют дополнительные фильтры посредством атрибута `cupsFilter` для поддержки драйверов принтеров.

В ОС стандартным языком описания страниц является язык PostScript и большинство прикладных программ (редакторы, браузеры) генерируют программы печати на этом языке. Когда необходимо напечатать ASCII-текст, программа печати может быть ASCII-текстом. Имеется возможность управления размером шрифтов при печати ASCII-текста. Управляющая информация используется для контроля доступа пользователя к принтеру и аудита печати. Также имеется возможность печати изображений в форматах GIF, JPEG, PNG, TIFF и документов в формате PDF.

Фильтр — программа, которая читает из стандартного входного потока или из файла, если указано его имя. Все фильтры поддерживают общий набор опций, включающий имя принтера, идентификатор задания, имя пользователя, имя задания, число копий и опции задания. Весь вывод направляется в стандартный выходной поток.

Фильтры предоставлены для многих форматов файлов и включают, в частности, фильтры файлов изображения и растровые фильтры PostScript, которые поддерживают принтеры, не относящиеся к типу PostScript. Иногда несколько фильтров запускаются параллельно для получения требуемого формата на выходе.

Программа `backend` — это специальный фильтр, который отправляет печатаемые данные устройству или через сетевое соединение. В состав системы печати включены фильтры для поддержки устройств, подключаемых с помощью параллельного и последовательного интерфейсов, а также шины USB.

Клиентские программы используются для управления заданиями и сервером печати.

Управление заданиями включает:

– формирование;

– передачу серверу печати;

– мониторинг и управление заданиями в очереди на печать.

Управление сервером включает:

– запуск/остановку сервера печати;

– запрещение/разрешение постановки заданий в очередь;

– запрещение/разрешение вывода заданий на принтер.

Основные пользовательские настройки содержатся в файлах конфигурации `client.conf` и `~/.cups/lpoptions`.

Для удаленного использования сервера печати необходимо от имени суперпользователя выполнить следующие команды:

```
cupsctl --remote-admin --remote-printers --remote-any
cupsctl ServerAlias=*
```

В случае использования сервера печати в ЕПП, необходимо задание соответствующего типа аутентификации: для работы в ЕПП значение параметра должно быть `DefaultAuthType Negotiate`, без использования ЕПП значение параметра должно быть `DefaultAuthType Basic`.

В файле конфигурации клиента `client.conf` должен быть задан один параметр `ServerName`, определяющий имя сервера печати.

Пример

```
ServerName computer.domain
```

В общем случае вывод данных на принтер происходит следующим образом:

1) программа формирует запрос на печать задания к серверу печати;

2) сервер печати принимает подлежащие печати данные, формирует в буферном каталоге файлы с содержимым задания и файлы описания задания, при этом задание попадает в соответствующую очередь печати;

3) сервер печати просматривает очереди печати для незанятых принтеров, находит в них задания и запускает конвейер процессов, состоящий из фильтров и заканчивающийся выходным буфером (backend), информация из которого поступает в принтер посредством драйверов ОС;

4) контроль и мониторинг процесса печати выполняется с помощью программ `lpq`, `lpc`, `lprm`, `lpstat`, `lpmove`, `cancel`, а также с помощью графической утилиты `fly-admin-printer`.

Система печати ОС решает следующие задачи:

1) монопольная постановка задания в очередь на печать. Данная функция предполагает невозможность вывода документа на печать в обход системы печати;

2) маркировка каждого напечатанного листа. Каждый лист сопровождается автоматической маркировкой (учетными атрибутами документа).

10.2. Маркировка документов

Маркировка печатных листов осуществляется «наложением» маркеров с учетными атрибутами документа, включающими:

– уровень конфиденциальности документа;

– номер экземпляра;

– количество листов в экземпляре;

– дату вывода документа на печать;

– номер каждого входящего документа;

– имя исполнителя;

– имя пользователя, производившего печать на станции печати.

Система печати является инвариантной по отношению к приложениям, которые обращаются к сервису печати. Это, в частности, означает, что приложения, выводящие на печать, должны учитывать маркировку листов и оставлять для этого свободное место. В противном случае маркеры могут затереть фрагменты печатаемой информации.

В каталогах `/usr/share/cups/psmarker` и `/usr/share/cups/fonarik` хранятся файлы с настойками маркеров печати. Настройка элементов маркировки осуществляется редактированием следующих файлов:

– `/usr/share/cups/psmarker/marker.template` — описание элементов маркера, проставляемых на первой, каждой и последней странице;

– `/usr/share/cups/psmarker/marker.defs` — описание положения элементов маркера на странице;

– `/usr/share/cups/fonarik/fonarik.template` — описание элементов маркировки, проставляемых на обороте последней страницы, при количестве экземпляров меньшем либо равном пяти;

– `/usr/share/cups/fonarik/fonarik_gt_5.template` — описание элементов маркировки, проставляемых на обороте последней страницы, при количестве экземпляров больше пяти;

– `/usr/share/cups/fonarik/fonarik.defs` — описание положения элементов маркера на странице;

Для изменения положения маркера, проставляемого на первой, каждой и последней странице необходимо в файле `/usr/share/cups/psmarker/marker.defs` изменить значение параметра:

– `MarkerTopShift` — для верхнего элемента маркера;

– `MarkerBottomShift` — для нижнего элемента маркера;

– `MarkerLeftShift` — для левого элемента маркера;

– `MarkerRightShift` — для правого элемента маркера.

Для изменения положения маркера, проставляемого на обороте последней страницы, необходимо в файле /usr/share/cups/fonarik/fonarik.defs изменить значение параметра:

– FonarikTopShift — для верхнего элемента маркера;

– FonarikBottomShift — для нижнего элемента маркера;

– FonarikLeftShift — для левого элемента маркера;

– FonarikRightShift — для правого элемента маркера.

Любые изменения содержания и формата маркера страниц может производить только суперпользователь.

Для выполнения маркировки и администрирования должны быть созданы группы lpmac и lpadmin.

Пользователь, от имени которого будут выполняться команды по маркировке, должен входить в группу lpmac.

Ряд действий по администрированию CUPS (добавление и удаление принтеров, изменение политики для принтера, установка мандатных атрибутов для принтера) может выполняться от имени пользователя, входящего в группу lpadmin.

Для печати документов с ненулевым мандатным контекстом необходимо соответствующим образом настроить принтер. Данная настройка осуществляется с использованием утилиты fly-admin-printer. В закладке «МАС» необходимо установить политику parsec, а также допустимый диапазон мандатных уровней и категорий. Дополнительная информация об утилите fly-admin-printer приведена в электронной справке.

После отправки пользователем на печать документа с ненулевым мандатным контекстом в очереди сервера печати формируется задание.

Для печати документа необходимо выполнить его маркировку. Маркировка выполняется вызовом скрипта markjob, который требует наличия утилиты lpq, входящей в состав пакета cups-bsd.

В процессе выполнения скрипта markjob у пользователя запрашиваются следующие атрибуты маркера:

1) mac-inv-num – инвентарный номер;

2) mac-owner-phone – телефон исполнителя;

3) mac-workplace-id – идентификатор рабочего места;

4) mac-distribution – список рассылки.

Если в значении атрибута используется пробел, то значение атрибута необходимо взять в кавычки.

Пример

Выдается запрос на ввод списка рассылки:

```
# Enter mac-distribution - Distribution list, addresses separated by '^':
```

Вводится список рассылки:

```
"В дело"
```

После выполнения маркировки в очереди формируются два дополнительных задания, первое (с меньшим номером) представляет собой промаркированный документ, а второе (с большим номером) — маркировку, размещаемую на обороте последнего листа документа. Необходимо возобновить выполнение первого задания, что приведет к печати промаркированного документа. Затем на обороте последнего листа документа печатается маркировка посредством возобновления выполнения второго дополнительного задания.

При выполнении маркировки от имени пользователя, входящего в группу `lpmac`, возможно получение сообщения:

```
Невозможно выполнить запрос: запрещено
```

В данном случае необходимо выполнить команду `id` от имени пользователя, выполняющего маркировку, и повторно запустить скрипт маркировки `markjob`.

10.3. Печать нескольких экземпляров документа с ненулевым мандатным уровнем

Для печати нескольких экземпляров документа с ненулевым мандатным уровнем пользователь должен отправить на печать только одну копию документа.

Пользователь, осуществляющий маркировку, должен выполнить следующую последовательность действий:

1) получить номер задания для маркировки, выполнив команду:

```
lpq -a
```

2) задать число копий для печати, выполнив команду:

```
lpattr -j <номер_задания> -s copies=<число_копий>
```

3) произвести маркировку, выполнив скрипт `markjob`.

При вводе списка рассылки адреса разделяются символом `'^'`. Если в значении списка рассылки используется пробел, то значение атрибута необходимо взять в кавычки целиком.

Пример

Выдается запрос на ввод списка рассылки:

```
# Enter mac-distribution - Distribution list, addresses separated by '^':
```

Вводится список рассылки:

```
"В дело^В адрес"
```

После выполнения маркировки в очереди формируются по два дополнительных задания для каждого экземпляра документа, располагаемых в очереди последовательно. Первое (с меньшим номером) представляет собой промаркированный экземпляр докумен-

та, а второе (с большим номером) — маркировку, размещаемую на обороте последнего листа экземпляра документа. Для печати экземпляра документа необходимо возобновить выполнение первого соответствующего ему задания, что приведет к печати промаркированного экземпляра документа. Затем на обороте последнего листа экземпляра документа печатается маркировка посредством возобновления выполнения второго соответствующего экземпляру документа дополнительного задания.

10.4. Установка и настройка принтера

Установку и настройку принтера следует производить после завершения установки и первоначальной настройки ОС. Для получения дополнительной информации о каких-либо командах можно обратиться к руководству `man`.

10.4.1. Общие положения

При печати через локальный сервер печати данные сначала формируются на локальном сервере, как для любой другой задачи печати, после чего посылаются на принтер, подключенный к данному компьютеру.

Вся информация, необходимая для драйвера принтера (используемое физическое устройство, удаленный компьютер и принтер для удаленной печати), содержится в файлах `/etc/cups/printers.conf` и `/etc/cups/ppd/<имя_очереди>.ppd`.

Далее термин «принтер» в этом разделе используется для обозначения принтера, соответствующего одной записи в файле `/etc/cups/printers.conf`. Под термином «физический принтер» подразумевается устройство, с помощью которого производится печать на бумаге. В файле `/etc/cups/printers.conf` может быть несколько записей, описывающих один физический принтер различными способами.

10.4.2. Команды управления печатью

В систему печати ОС включены файлы, предоставляющие командный интерфейс пользователя в стиле BSD и SystemV (таблица 37).

Таблица 37

Файл	Описание
`/usr/bin/lpr`	Постановка заданий в очередь. Совместима с командой `lpr` системы печати BSD UNIX
`/usr/bin/lp`	Постановка заданий в очередь. Совместима с командой `lp` системы печати System V UNIX
`/usr/bin/lpq`	Просмотр очередей печати
`/usr/sbin/lpc`	Управление принтером. Является частичной реализацией команды `lpc` системы печати BSD UNIX
`/usr/bin/lprm`	Отмена заданий, поставленных в очередь на печать

Окончание таблицы 37

Файл	Описание
/usr/sbin/cupsd	Сервер печати
/usr/sbin/lpadmin	Настройка принтеров и классов принтеров
/usr/sbin/lpmove	Перемещение задания в другую очередь
/usr/bin/fly-admin-printer	Настройка системы печати, установка и настройка принтеров, управление заданиями

CUPS предоставляет утилиты командной строки для отправления заданий и проверки состояния принтера. Команды `lpstat` и `lpc status` также показывают сетевые принтеры (`принтер@сервер`), когда разрешен обзор принтеров.

Команды администрирования System V предназначены для управления принтерами и классами. Средство администрирования (`lpc`) поддерживается только в режиме чтения для проверки текущего состояния очередей печати и планировщика.

С помощью команды `lp` выполняется передача задачи принтеру, т. е. задача ставится в очередь на печать. В результате выполнения этой команды файл передается серверу печати, который помещает его в каталог `/var/spool/cups/`.

С помощью команды `lpq` отображается содержимое каталога `/var/spool/cups/` для конкретного принтера. Особое значение имеет ID задания, т. к. без него, к примеру, нельзя отменить задание печати, находящееся в состоянии ожидания.

С помощью `lpq` определяется ранг каждой задачи в очереди. Значение `active` относится к файлу, который распечатывается в данный момент.

С помощью команды `lprm` задание удаляется из очереди, т. е. ненапечатанные файлы удаляются из каталога спулинга. В качестве аргумента можно указать ID задачи, полученный с помощью команды `lpq`, или использовать символ `-`; в этом случае отменяются все принадлежащие пользователю задания.

Если это сделает суперпользователь, будут отменены все задачи, стоящие в очереди на принтер. Для отмены суперпользователем заданий, принадлежащих определенному пользователю, необходимо указать его имя.

Остановить работу сервиса печати можно с помощью команды:

```
service cups stop
```

Запустить сервис печати можно с помощью команды:

```
service cups start
```

10.4.2.1. lpq

Команда `lpq` предназначена для проверки очереди печати (используемой `lpd`) и вывода состояния заданий на печать, указанных при помощи номера задания либо системного идентификатора пользователя, которому принадлежит задание. Она выводит для

каждого задания имя его владельца, текущий приоритет задания, номер задания и размер задания в байтах, без параметров выводит состояние всех заданий в очереди.

10.4.2.2. lprm

Команда `lprm` предназначена для удаления задания из очереди печати. Для определения номера задания необходимо использовать команду `lpq`. Для того чтобы удалить задание, необходимо быть его владельцем или суперпользователем.

Системные каталоги, определяющие работу системы печати ОС, также содержат файлы, которые не являются исполняемыми:

– `/etc/cups/printers.conf` — содержит описания принтеров в ОС;

– `/etc/cups/ppd/<имя_очереди>.ppd` — содержит описания возможностей принтера, которые используются при печати заданий и при настройке принтеров;

– `/var/log/cups/error_log` — поступает протокол работы принтера. В этом файле могут находиться сообщения об ошибках сервера печати или других программ системы печати;

– `/var/log/cups/access_log` — регистрируются все запросы к серверу печати;

– `/var/log/cups/page_log` — поступают сообщения, подтверждающие успешную обработку страниц задания фильтрами и принтером.

10.4.2.3. lpadmin

Настроить принтер в ОС можно также с помощью команды `lpadmin`.

Ее запуск с опцией `-p` — для добавления или модификации принтера:

```
/usr/sbin/lpadmin -p printer [опции]
```

Основные опции команды `lpadmin` приведены в таблице 38.

Таблица 38

Опция	Описание
`-c class`	Добавляет названный принтер к классу принтеров `class`. Если класс не существует, то он создается
`-m model`	Задает стандартный драйвер принтера, обычно файл PPD. Файлы PPD обычно хранятся в каталоге `/usr/share/cups/model/`. Список всех доступных моделей можно вывести командой `lpinfo` с опцией `-m`
`-r class`	Удаляет указанный принтер из класса `class`. Если в результате класс становится пустым, он удаляется
`-v device-uri`	Указывает адрес устройства для связи с принтером
`-D info`	Выдает текстовое описание принтера
`-E`	Разрешает использование принтера и включает прием заданий
`-L location`	Выводит расположение принтера
`-P ppd-file`	Указывает локальный файл PPD для драйвера принтера

Для данной команды существуют также опции по регулированию политики лимитов и ограничений по использованию принтеров и политики доступа к принтерам.

Запуск команды `lpadmin` с опцией `-x` — для удаления принтера:

```
/usr/sbin/lpadmin -x printer
```

10.4.2.4. fly-admin-printer

Утилита `fly-admin-printer` предназначена для настройки печати в графическом режиме. Позволяет в режиме суперпользователя устанавливать, настраивать и удалять принтеры и классы принтеров, а также настраивать сервер печати и управлять заданиями на печать. В режиме обычного пользователя позволяет устанавливать настройки печати и опции принтера, а также управлять заданиями на печать (удалять, приостанавливать, возобновлять печать и устанавливать отложенную печать). Для вызова привилегированных действий запрашивается авторизация. Подробную информацию по использованию утилиты `fly-admin-printer` см. в электронной справке.

11. ЗАЩИЩЕННАЯ СИСТЕМА УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ

11.1. Назначение

В качестве защищенной СУБД в составе ОС используется PostgreSQL, доработанная в соответствии с требованием интеграции с ОС в части мандатного разграничения доступа к информации.

СУБД PostgreSQL предназначена для создания и управления реляционными БД и предоставляет многопользовательский доступ к расположенным в них данным.

Данные в реляционной БД хранятся в отношениях (таблицах), состоящих из строк и столбцов. При этом единицей хранения и доступа к данным является строка, состоящая из полей, идентифицируемых именами столбцов. Кроме таблиц, существуют другие объекты БД (виды, процедуры и т. п.), которые предоставляют доступ к данным, хранящимся в таблицах.

Для работы СУБД на диске выделяется область для хранения БД, называемая «кластером БД». Кластер БД является набором БД, управляемых одним экземпляром сервера СУБД. Настройка работы отдельного экземпляра сервера СУБД так же определяется в рамках кластера соответствующими конфигурационными файлами.

Корректная работа с СУБД предполагает использование механизма ЕПП.

11.2. Состав

СУБД PostgreSQL состоит из нескольких компонентов:

– `postgresql` — сервисная служба, реализующая непосредственно сервер БД;

– `libpq` — клиентская библиотека, предоставляющая доступ к серверу СУБД;

– набор серверных утилит для управления работой сервера и создания кластеров БД;

– набор клиентских утилит для создания и управления БД.

В состав ОС входит графическая утилита `fly-admin-postgres`, предназначенная для администрирования БД СУБД PostgreSQL, включая управления мандатным разграничением доступа к объектам БД (см. электронную справку).

11.3. Настройка

Настройка сервера СУБД осуществляется установкой параметров в конфигурационном файле `postgresql.conf`. В дополнение к файлу `postgresql.conf` в PostgreSQL используется еще два конфигурационных файла, которые контролируют аутентификацию клиента (11.4). По умолчанию все эти три файла находятся в каталоге данных кластера БД или в соответствующем кластеру конфигурационном каталоге, например `/etc/postgreslq/8.4/main`. За расположение указанных файлов отвечают конфигура-

ционные параметры, приведенные в таблице 39.

Таблица 39

Параметр	Описание
data_directory	Определяет каталог для хранения данных
config_file	Определяет основной конфигурационный файл сервера (postgresql.conf). Значение этого параметра может быть задано только в командной строке postgres
hba_file	Определяет конфигурационный файл для аутентификации по узлам (pg_hba.conf)
ident_file	Определяет конфигурационный файл для аутентификации по методу ident (pg_ident.conf)
external_pid_file	Определяет имя дополнительного файла с идентификатором процесса, который сервер создает для использования программами администрирования сервера

Для настройки работы сервера с мандатным разграничением доступа существует ряд конфигурационных параметров, указываемых в конфигурационном файле postgresql.conf. Описание параметров приведено в документе РУСБ.10015-01 97 01-1 «Операционная система специального назначения «Astra Linux Special Edition». Руководство по КСЗ. Часть 1».

При использовании относительного пути для задания значений этих параметров путь будет отсчитываться от каталога, в которой запущен postgres.

За установку соединений отвечают конфигурационные параметры, приведенные в таблице 40.

Таблица 40

Параметр	Описание
listen_addresses	Определяет TCP/IP-адреса, по которым сервер должен ожидать соединения от клиентских приложений. Значение формируется в виде перечня разделенных запятой имен узлов и/или числовых IP-адресов. Специальный знак * соответствует всем доступным IP-адресам. Если список пуст, сервер не слушает ни один IP-интерфейс. В этом случае установка соединения с сервером возможна только с использованием доменных сокетов UNIX. По умолчанию значением параметра является localhost, которое позволяет создавать только локальные loopback-соединения
port	Определяет TCP-порт, на котором сервер должен ожидать соединения от клиентских приложений (по умолчанию — 5432). Следует отметить, что для всех IP-адресов, указанных в listen_addresses (string), используется один и тот же порт

Окончание таблицы 40

Параметр	Описание
`max_connections`	Определяет максимальное число одновременных соединений с сервером СУБД
`superuser_reserved_connections`	Определяет число соединений, зарезервированных для подключения суперпользователей PostgreSQL
`unix_socket_group`	Определяет группу, владеющую доменным сокетом UNIX. Владельцем сокета всегда является пользователь, запускающий сервер. Используется в комбинации с параметром `unix_socket_permissions`, как дополнительный механизм контроля доступа для соединений в домене UNIX
`unix_socket_permissions`	Определяет права доступа для доменного сокета UNIX. Для доменного сокета UNIX имеет значение только разрешение на запись, и следовательно нет необходимости устанавливать или удалять разрешение на чтение или выполнение. Данный механизм контроля доступа не зависит от механизма, описанного в 11.4

11.4. Аутентификация клиента СУБД

При попытке соединения с сервером СУБД клиентское приложение указывает пользователя СУБД PostgreSQL, от имени которого осуществляется подключение. В пределах окружения SQL активное имя пользователя СУБД определяет права на объекты БД.

PostgreSQL предлагает несколько различных методов аутентификации клиента. Метод, используемый для аутентификации конкретного клиентского соединения, может быть выбран на основе адреса узла сети клиента, БД и пользователя.

Несмотря на то, что имена пользователей СУБД PostgreSQL логически отделены от имен пользователей ОС, в которой запущен сервер, в соответствии с требованиями по защите информации от несанкционированного доступа требуется сопоставление пользователей СУБД пользователям ОС. Таким образом, при настройке аутентификации в СУБД следует использовать только методы аутентификации, в которых осуществляется подобное сопоставление. Для других пользователей осуществляется доступ только к незащищенной информации.

Условием корректного функционирования СУБД является использование механизма ЕПП (см. раздел 6). Для этого в качестве метода аутентификации должен быть указан `gss` и проведена соответствующая настройка сервера и клиента СУБД PostgreSQL (см. 6.8.1).

11.4.1. Файл конфигурации pg_hba.conf

Аутентификация клиентов настраивается в конфигурационном файле `pg_hba.conf`, расположенном в каталоге данных кластера БД. По умолчанию файл

pg_hba.conf устанавливается при инициализации каталога данных с помощью initdb. Однако данный конфигурационный файл для аутентификации может быть размещен в другом месте (см. конфигурационный параметр hba_file).

Файл pg_hba.conf представляет собой набор записей, каждая из которых находится в отдельной строке. Пустые строки игнорируются, также как и любой текст после символа #, обозначающего комментарии. Запись состоит из нескольких полей, разделенных пробелами и/или символами табуляции. Поле может содержать символ пробела при условии, что все значение поля заключено в кавычки. Запись не может занимать более одной строки.

Каждая запись определяет тип соединения, диапазон IP-адресов клиента (когда это имеет значение для типа соединения), имя БД, имя пользователя и метод аутентификации, который следует использовать для соединений, соответствующих этим параметрам. При аутентификации используется первая запись с подходящим типом соединения, адресом клиента, запрашиваемой БД и именем пользователя. При этом возможность перехода через запись или возврата отсутствует: в случае когда выбрана какая-либо запись, и аутентификация не выполнена, прочие записи не рассматриваются. При отсутствии подходящий для аутентификации записи доступ запрещен.

Запись может иметь один из следующих семи форматов:

```
local      database  user  auth-method  [auth-options]
host       database  user  CIDR-address  auth-method  [auth-options]
hostssl    database  user  CIDR-address  auth-method  [auth-options]
hostnossl  database  user  CIDR-address  auth-method  [auth-options]
host       database  user  IP-address  IP-mask  auth-method  [auth-options]
hostssl    database  user  IP-address  IP-mask  auth-method  [auth-options]
hostnossl  database  user  IP-address  IP-mask  auth-method  [auth-options]
```

Запись local используется при попытке соединения через доменные сокеты UNIX. Если нет записи такого типа, доступ через доменные сокеты UNIX запрещен.

Запись host соответствует попытке установить соединение по TCP/IP. Записи host могут применяться как для соединений, использующих SSL, так и для не использующих SSL. Следует отметить, что удаленные соединения по TCP/IP возможны, только если при запуске сервера было указано соответствующее значение для конфигурационного параметра listen_addresses, т. к. по умолчанию сервер ожидает TCP/IP-соединения только на локальном адресе обратной связи localhost.

Запись hostssl соответствует попыткам соединения по TCP/IP с обязательным использованием SSL. Для применения данной опции необходимо, чтобы сервер был собран с поддержкой SSL.

Запись hostnossl реализует логику, противоположную записи hostssl, и соответствует соединениям по TCP/IP, не использующим SSL.

Поле database определяет, какие имена БД соответствуют этой записи. Значение all соответствует всем БД. Значение sameuser указывает, что запись подходит, если запрашиваемая БД имеет то же имя, что и запрашивающий пользователь. Значение samerole указывает, что запрашивающий пользователь должен быть членом роли с тем же именем, что и запрашиваемая БД. Кроме того, в значении может быть указано имя определенной БД PostgreSQL. Может быть указано несколько имен БД, перечисленных через запятую. Может быть указан отдельный файл, содержащий имена БД, посредством предшествующего имени файла символа @.

Поле user определяет имена пользователей БД, соответствующие данной записи. Значение all определяет, что запись действует для всех пользователей СУБД. В других случаях указывается имя определенного пользователя БД или имя группы, предваряемое символом +. Следует отметить, что в PostgreSQL не существует реального отличия между пользователями и группами. Символ + в действительности определяет соответствие записи любым ролям, которые прямо или опосредованно являются членами данной роли, в то время как имя без символа + означает соответствие записи только данной конкретной роли. Может быть указано несколько имен пользователей, перечисленных через запятую. Может быть указан отдельный файл, содержащий имена пользователей, посредством предшествующего имени файла символа @.

Поле CIDR-address определяет диапазон IP-адресов клиентского компьютера, которому соответствует данная запись. Поле содержит IP-адрес в стандартной нотации с использованием разделенных точками десятичных чисел и длину маски CIDR. IP-адреса могут быть указаны только в числовом формате, имена домена и узла не принимаются. Длина маски показывает число старших битов, которые должны совпадать с соответствующими битами IP-адреса клиента. Биты в IP-адресе правее битов определенных маской должны быть нулевыми. Не допускается наличие любых пробелов между IP-адресом, символом / и длиной маски CIDR. Далее приведены типичные примеры значений поля CIDR-address.

Примеры:

1. Значение поля для конкретного узла сети

```
172.20.143.89/32
```

2. Значение поля для сети класса C

```
172.20.143.0/24
```

3. Значение поля для сети класса A

```
10.6.0.0/16
```

Для задания конкретного узла сети необходимо указывать в маске CIDR значение

32 для формата IPv4 или 128 для формата IPv6. Пропуск в сетевых адресах завершающих нулей не допускается.

IP-адрес, заданный в формате IPv4, будет подходить и для соединений с соответствующим адресом в формате IPv6. Например, значение 127.0.0.1 соответствует адресу в формате IPv6 ::ffff:127.0.0.1. Значение, заданное в формате IPv6, будет подходить только для адресов в формате IPv6, даже если представленный адрес содержится в диапазоне IPv4–IPv6. Следует отметить, что значения поля в формате IPv6 будут отброшены в случае, когда системная библиотека C не поддерживает формат адресов IPv6. Данное поле используется только в записях host, hostssl и hostnossl.

Поля IP-address и IP-mask используются как альтернатива нотации CIDR-address. Вместо указания длины маски в отдельном столбце IP-mask задается действительная маска. Например, значение 255.0.0.0 поля IP-mask соответствует маске CIDR длиной 8 в формате IPv4, а значение 255.255.255.255 поля IP-mask соответствует маске CIDR длиной 32. Данные поля используются только в записях host, hostssl и hostnossl.

Поле auth-method определяет метод аутентификации, используемый соединениями, соответствующими данной записи. Далее приведены возможные значения поля:

– trust — безусловно разрешает соединение. Данный метод позволяет любому, кто может соединиться с сервером СУБД PostgreSQL, входить под любой учетной записью СУБД PostgreSQL без пароля. Должно использоваться только при разработке;

– reject — безусловно отвергает попытки соединений. Использование значения полезно при необходимости отфильтровать отдельные узлы сети из группы;

– gss — требует использования для аутентификации пользователей GSSAPI;

– pam — требует использования для аутентификации сервиса PAM, предоставляемого ОС.

После поля auth-method может быть указана одна или несколько опций метода аутентификации (поля auth-options) в форме пар name=value (имя=значение). Файлы, включенные посредством символа @, воспринимаются как списки имен, разделенные запятыми или пробелами. Комментарии вводятся символом #, аналогично файлу pg_hba.conf. Так же разрешены вложенные конструкции с использованием символа @. Если имя файла, следующее за символом @, не является абсолютным путем, то оно определяется относительно каталога, содержащей файл, в котором находится ссылка на данное имя файла. Порядок записей файла pg_hba.conf имеет значение вследствие последовательной проверки записей при каждой попытке соединения. В общем случае первые записи должны содержать наиболее жесткие ограничения параметров соединения и

наименее безопасные методы аутентификации, в последующих записях должны содержаться менее жесткие ограничения для параметров соединения и более безопасные методы аутентификации. Например, можно использовать метод аутентификации `trust` для локальных TCP/IP-соединений, но требовать пароль для удаленных TCP/IP-соединений. В данном случае запись, определяющая метод аутентификации `trust` для соединений с `127.0.0.1`, должна предшествовать записи, требующей аутентификации с паролем для широкого диапазона допустимых IP-адресов клиентов.

Файл `pg_hba.conf` считывается при запуске и при получении основным процессом сервера сигнала `SIGHUP`. После редактирования данного файла во время работы сервера для повторного считывания файла необходимо отправить серверу сигнал, используя команду:

```
pg_ctl reload
```

или:

```
kill -HUP
```

Для подключения к определенной БД пользователь должен не только пройти проверку в соответствии с записями `pg_hba.conf`, но и иметь привилегию `CONNECT` для данной БД. Таким образом, установку ограничений на подключение пользователей к БД проще осуществлять добавляя или отбирая привилегию `CONNECT`, а не создавая правила в записях `pg_hba.conf`.

Далее приведены примеры записей файла `pg_hba.conf`.

Примеры:

1.
```
# В отсутствие предыдущих строк типа host, следующие две строки
# запрещают все соединения с 192.168.54.1 (поскольку запись будет
# проверена в первую очередь), но разрешают соединения используя
# Kerberos 5 с любого другого адреса. Нулевая маска означает,
# что ни один бит IP-адреса не рассматривается, следовательно
# подходит любой узел.
#
# TYPE  DATABASE    USER   CIDR-ADDRESS          METHOD
host    all         all    192.168.54.1/32       reject
host    all         all    0.0.0.0/0             krb5
```

2.
```
# Разрешить пользователям с узлов 192.168.x.x подключаться
# к любой базе данных, если они проходят проверку ident.
# Если, например, сервис ident сообщает, что имя пользователя
# "bryanh", и он запрашивает разрешение на соединение как
# пользователь PostgreSQL "guest1", то соединение разрешается,
# если в pg_ident.conf есть запись для карты "omicron",
```

```
# разрешающая "bryanh" соединяться как "guest1".
#
# TYPE  DATABASE    USER    CIDR-ADDRESS          METHOD
host    all         all     192.168.0.0/16        ident map=omicron
```

3. `# Столбец DATABASE также может содержать списки и имена файлов:`

```
# TYPE  DATABASE       USER    CIDR-ADDRESS        METHOD
local   db1,db2,@demodbs  all                         md5
```

11.4.2. Карты имен пользователей pg_ident.conf

При использовании внешней системы аутентификации, например GSSAPI, имя пользователя ОС, которая инициирует соединение, может не совпадать с именем пользователя БД, которое необходимо использовать для соединения. В подобном случае может быть применена карта имен пользователей ОС для отображения имен пользователей ОС в имена пользователей БД. Для использования карты имен пользователей необходимо указать map=map-name в поле опций файла pg_hba.conf. Опция поддерживается всеми методами аутентификации, которые получают внешние имена пользователей. Поскольку может быть необходимо использовать различные отображения для различных соединений, имя используемой карты задается в параметре map-name в файле pg_hba.conf для указания карты, применяемой для каждого конкретного соединения.

Карты отображений имен пользователей определены в файле карт pg_ident.conf, расположенный в каталоге данных кластера. Файл карт содержит записи следующего общего вида:

```
map-name system-username database-username
```

Комментарии и пробелы обрабатываются аналогично файлу pg_hba.conf. Поле map-name задает произвольное имя, которое будет использовано для ссылки на карту в файле pg_hba.conf. В двух других полях указывается имя пользователя ОС и соответствующее имя пользователя БД. Одно имя, заданное в поле map-name, может быть неоднократно использовано для указания множества отображений имен пользователей в одной карте.

Ограничения относительно количества пользователей БД, соответствующих данному пользователю ОС и наоборот, отсутствуют. Следовательно, запись в карте должна задаваться в значении, что данному пользователю ОС разрешено подключение как такому-то пользователю БД, а не в значении, что они эквивалентны. Подключение будет разрешено, если существует некоторая запись в карте, определяющая, что имя пользователя, полученное от внешней системы аутентификации, соответствует имени пользователя БД, указанному в запросе на подключение.

Если поле system-username начинается с символа /, то оставшаяся часть содер-

жимого поля воспринимается как регулярное выражение. Регулярные выражения в картах имен пользователей всегда рассматривались как дополнительная возможность. Регулярное выражение может включать одно заключенное в скобки подвыражение, на которое можно ссылаться в поле database-name, используя \1. Данная возможность позволяет выполнять отображение множества имен пользователей в одной строке, что особенно полезно для простых синтаксических подстановок. Далее приведен пример использования регулярных выражений в карте имен пользователей.

Пример

Файл pg_ident.conf

```
mymap   /(.*)@mydomain.com   \1
mymap   /(.*)@otherdomain.com guest
```

Записи в данном примере будут исключать доменную часть имени для пользователей, у которых системное имя пользователя завершается @mydomain.com, и разрешать всем пользователям, у которых системное имя завершается @otherdomain.com, входить от имени учетной записи guest. Следует помнить, что по умолчанию регулярное выражение может осуществлять сопоставление для части строки. Целесообразно использовать ^ и $, как показано в приведенном выше примере, для принудительного выполнения сопоставления для системного имени пользователя в целом.

Файл pg_ident.conf считывается при запуске и при получении сервером сигнала SIGHUP. По завершении редактирования данного файла во время работы системы необходимо отправить серверу сигнал посредством команды:

```
pg_ctl reload
```

или:

```
kill -HUP
```

для повторного считывания файла.

Далее приведен пример файла pg_ident.conf, который может быть использован в сочетании с записью в файле pg_hba.conf, приведенной ранее. В соответствии с записями файлов pg_ident.conf и pg_hba.conf любой пользователь, выполнивший вход в систему в IP-сети 192.168 с именем, отличным от bryanh, ann или robert, доступ не получит. Системный пользователь robert сможет получить доступ при подключении с именем пользователя PostgreSQL bob, а не robert или каким-либо еще. Системный пользователь ann может подключиться только как ann. Системный пользователь bryanh сможет подключиться как bryanh или как guest1.

Пример

Файл pg_ident.conf

```
# MAPNAME     IDENT-USERNAME    PG-USERNAME
```

```
omicron       bryanh            bryanh
omicron       ann               ann
# на этих машинах bob соответствует имени пользователя robert
omicron       robert            bob
# bryanh также может подключаться как guest1
omicron       bryanh            guest1
```

12. ЗАЩИЩЕННЫЙ КОМПЛЕКС ПРОГРАММ ГИПЕРТЕКСТОВОЙ ОБРАБОТКИ ДАННЫХ

12.1. Общие сведения

Комплекс гипертекстовой обработки документов HTTP — это ПО, осуществляющее взаимодействие по HTTP-протоколу между сервером и браузерами: прием запросов, поиск указанных файлов и передача их содержимого, выполнение приложений на сервере и передача клиенту результатов их выполнения. Комплекс представлен web-севером Apache и браузером Firefox. Web-сервер Apache, входящий в состав ОС, не допускает возможности анонимного использования ресурсов web-сервера и требует обязательной настройки авторизации пользователей.

12.2. Настройка сервера

Сразу после установки сервер настроен и готов к приему запросов на всех сетевых интерфейсах на 80 порту. Если по каким-то причинам он не работоспособен, следует проверить минимально необходимые настройки сервера. В файле `/etc/apache2/ports.conf` должны быть указаны параметры:

```
NameVirtualHost *:80
Listen 80
```

В каталоге `/etc/apache2/sites-available` должны находиться файлы с настройками виртуальных хостов и как минимум один из них должен быть разрешен к использованию командой:

```
a2ensite config_filename
```

Минимальное содержимое таких файлов с конфигурациями виртуальных хостов выглядит следующим образом:

```
<VirtualHost *:80>
        ServerAdmin webmaster@localhost
        ServerName server.domain.name
        DocumentRoot /path/to/root/dir/
        <Directory /path/to/root/dir/>
                Options Indexes FollowSymLinks MultiViews
                AllowOverride None
        </Directory>
        ErrorLog /var/log/apache2/error.log
        LogLevel warn
        CustomLog /var/log/apache2/access.log combined
</VirtualHost>
```

После окончания правки конфигурационных файлов необходимо перезапустить сервер командой:

```
/etc/init.d/apache2 restart
```

12.3. Настройка авторизации

Настройку сквозной авторизации для сервера и клиента, работающих в рамках ЕПП, см. в 6.8.3. Если не настроена авторизация через Kerberos, по умолчанию для всех ресурсов будет использоваться авторизация через PAM, при этом будет использоваться пользовательская БД, прописанная в настройках ОС. Логин и пароль пользователя будут передаваться от пользователя к серверу в открытом виде с использованием метода аутентификации Basic. Для корректного функционирования авторизации через PAM пользователю, от которого работает web-сервер (по умолчанию — `www-data`), необходимо выдать права на чтение информации из БД пользователей и сведений о мандатных метках.

Например, добавить права на чтение файла `/etc/shadow`:

```
usermod -a -G shadow www-data
```

и права на чтение каталога `/etc/parsec/macdb`:

```
setfacl -d -m u:www-data:r /etc/parsec/macdb
setfacl -R -m u:www-data:r /etc/parsec/macdb.
setfacl -m u:www-data:rx /etc/parsec/macdb
```

13. ЗАЩИЩЕННЫЙ КОМПЛЕКС ПРОГРАММ ЭЛЕКТРОННОЙ ПОЧТЫ

13.1. Общие сведения

В качестве защищенного комплекса программ электронной почты используется сервер электронной почты, состоящий из агента передачи электронной почты Exim, агента доставки электронной почты Dovecot и клиента электронной почты Thunderbird, доработанных для реализации следующих дополнительных функциональных возможностей:

– интеграции с ядром ОС и базовыми библиотеками для обеспечения разграничения доступа;

– реализации мандатного разграничения доступа к почтовым сообщениям;

– автоматической маркировки создаваемых почтовых сообщений, отражающих уровень их конфиденциальности;

– регистрации попыток доступа к почтовым сообщениям.

Агент передачи электронной почты использует протокол SMTP и обеспечивает решение следующих задач:

– доставку исходящей почты от авторизованных клиентов до сервера, который является целевым для обработки почтового домена получателя;

– прием и обработку почтовых сообщений доменов, для которых он является целевым;

– передачу входящих почтовых сообщений для обработки агентом доставки электронной почты.

Агент доставки электронной почты предназначен для решения задач по обслуживанию почтового каталога и предоставления удаленного доступа к почтовому ящику по протоколу IMAP.

Клиент электронной почты — прикладное ПО, устанавливаемое на рабочем месте пользователя и предназначенное для получения, написания, отправки и хранения сообщений электронной почты пользователя.

13.2. Состав

Защищенный комплекс программ электронной почты состоит из следующих пакетов:

– `exim4-daemon-heavy` — агент передачи сообщений СЭП (MTA) Exim 4. `exim4-daemon-ligth` не поддерживает работу с мандатными метками, отличными от 0:0;

– `dovecot-imapd` — агент доставки сообщений СЭП (MDA) Dovecot. Работает только по протоколу IMAP, протокол POP3 отключен. Серверная часть СЭП в за-

щищенном исполнении использует в качестве почтового хранилища MailDir, mailbox не поддерживает работу с мандатными метками, отличными от 0:0;

– `thunderbird` — клиент СЭП (MUA) Mozilla Thunderbird.

13.3. Настройка серверной части

Настройки по умолчанию:

– прием почты по протоколу SMTP, только от MUA из доменов `relay-domens` и из подсети;

– отправка почты по протоколу SMTP в соответствии с DNS;

– хранение локальной почты в MailDir в `/var/mail/%u`, где `%u` — локальная часть адресата;

– выдача локальной почты по протоколу IMAP.

13.3.1. Настройка агента доставки сообщений

Настройка агента доставки сообщений СЭП (MTA) Dovecot осуществляется путем правки конфигурационного файла `/etc/dovecot/dovecot.conf`.

В файле необходимо задать список интерфейсов, с которых будут приниматься соединения, например:

```
listen =  192.168.2.55
```

Необходимо отключить поддержку SSL/TLS установив значение `no` для параметра `ssl`:

```
ssl = no
```

Для настройки аутентификации без использования Kerberos в секции `auth default` конфигурационного файла `/etc/dovecot/dovecot.conf` установить:

```
mechanisms = plain
```

13.3.2. Настройка агента передачи сообщений

Для настройки агента передачи сообщений СЭП (MTA) Exim 4 требуется инициировать переконфигурированиие пакета `exim4-config`, для этого выполнить в эмуляторе терминала команду:

```
#dpkg-reconfigure exim4-config
```

В появившемся окне настройки для указанных ниже параметров необходимо установить следующие значения:

– общий тип почтовой конфигурации: интернет-сайт; прием и отправка почты напрямую, используя SMTP;

– почтовое имя системы: <имя_домена>;

– IP-адреса, с которых следует ожидать входящие соединения: <IP-адрес_сервера> (например, `192.168.32.1`);

– другие места назначения, для которых должна приниматься почта: <имя_домена>;

– домены, для которых доступна релейная передача почты: <оставить пустым>;

– компьютеры, для которых доступна релейная передача почты: <оставить пустым>;

– сокращать количество DNS-запросов до минимума: <Нет>;

– метод доставки локальной почты: Maildir — формат в домашнем каталоге;

– разделить конфигурацию на маленькие файлы: <Да>.

Если возникла необходимость изменить расположение каталога /var/spool/exim4, убедиться, что каталог exim4, подкаталоги db input msglog, файлы db/retry db/retry.lockfile имеют мандатные метки 0:0:equ, если это не так, установить соответствующие метки на указанные каталоги и файлы командами:

```
#cd new_dir
#chmac 0:0:equ . db input msglog db/retry db/retry.lockfile
```

Если возникла необходимость изменить расположение каталога хранилища локальной почты /var/mail, убедиться, что на новый каталог установлены права 1777, если это не так, установить командой:

```
#chmod 1777 new_dir
```

Настройку сквозной авторизации для сервера и клиента, работающих в рамках ЕПП, см. 6.8.2.

13.3.3. Настройка порядка запуска сервисов СЭП

Для работы сервера СЭП необходимо установить корректный порядок запуска сервисов:

– сервис exim4 должен запускаться до сервиса dovecot;

– сервис exim4 должен останавливаться до сервиса dovecot.

Установка корректного порядка запуска служб может быть выполнена с помощью следующей последовательности команд:

```
# update-rc.d -f dovecot remove
# update-rc.d -f exim4 remove
# update-rc.d dovecot defaults 30
# update-rc.d exim4 defaults 31 29
```

13.4. Настройка клиентский части

При создании учетной записи пользователя СЭП в MUA Mozilla Thunderbird необходимо выбрать тип используемого сервера входящей почты IMAP. При настройке учетной записи установить в параметрах сервера и параметрах сервера исходящей почты:

– защита соединения «Нет»;

– использование метода аутентификации «Обычный пароль».

14. СРЕДСТВА КОНТРОЛЯ ЦЕЛОСТНОСТИ

Для решения задач контроля целостности в ОС реализованы:

– средство подсчета контрольных сумм файлов и оптических дисков (14.1);

– средство контроля соответствия дистрибутиву (14.2);

– средства регламентного контроля целостности (14.3);

– средства создания замкнутой программной среды (14.4).

14.1. Средство подсчета контрольных сумм файлов и оптических дисков

Для подсчета контрольных сумм файлов и оптических дисков в состав ОС включена утилита командной строки `gostsum`. Для вывода информации о синтаксисе утилиты `gostsum` необходимо выполнить команду:

```
gostsum -h
```

Синтаксис:

```
gostsum [-b buffer_size] [input file name][-o output file name]
    [-d device[iso file]]
```

Опции приведены в таблице 41.

Таблица 41

Опция	Описание
`-b`	Устанавливает размер блоков, которыми будет считываться файл
`input file name`	Задает имя файла для подсчета контрольной суммы (по умолчанию — стандартный поток ввода)
`-o`	Задает имя файла для вывода контрольной суммы (по умолчанию — стандартный поток вывода)
`-d`	Задает имя файла устройства чтения оптических дисков (файла с образом оптического диска) для подсчета контрольной суммы

Далее приведен пример подсчета контрольной суммы оптического диска:

```
gostsum -d /dev/cdrom
```

14.2. Средство контроля соответствия дистрибутиву

Средство контроля соответствия дистрибутиву предоставляет возможность для контроля соответствия объектов файловой системы ОС дистрибутиву ОС. Для обеспечения контроля целостности объектов ФС ОС (в том числе СЗИ) в состав дистрибутива входит файл `gostsums.txt` со списком контрольных сумм всех файлов, входящих в пакеты программ дистрибутива. Используя графическую утилиту `fly-admin-int-check`, можно провести вычисление контрольных сумм файлов системы и проверку соответствия полученных контрольных сумм файлов системы эталонным контрольным суммам. Более подробное описание утилиты см. в электронной справке.

14.3. Средства регламентного контроля целостности

Организация регламентного контроля целостности ОС, прикладного ПО и СЗИ обеспечивается набором программных средств на основе «Another File Integrity Checker». В указанном наборе программных средств реализована возможность для проведения периодического периодического (с использованием системного планировщика заданий `cron`) вычисления контрольных сумм файлов и соответствующих им атрибутов расширенной подсистемы безопасности PARSEC (мандатных атрибутов и атрибутов расширенной подсистемы протоколирования) с последующим сравнением вычисленных значений с эталонными. В указанном наборе программных средств реализовано использование библиотеки `libgost`, обеспечивающей подсчет контрольных сумм в соответствии с ГОСТ Р 34.11-94.

Эталонные значения контрольных сумм и атрибутов фалов хранятся в БД. База контрольных сумм и атрибутов может быть создана при помощи команды:

```
afick –i
```

Для вычисления контрольных сумм могут использоваться алгоритмы: MD5-Digest, SHA1 и ГОСТ Р 34.11-94.

14.3.1. Настройка

Для настройки достаточно параметров, которые указаны в конфигурационном файле по умолчанию (`etc/afick.conf`). Кроме различных путей, например, к файлам БД:

```
database:=/var/lib/afick/afick
```

где содержится указание о том, какие файлы/каталоги подвергаются контролю целостности и с какими правилами.

Правило PARSEC выглядит следующим образом:

```
PARSEC = p+d+i+n+u+g+s+b+md5+m+e+t
```

где `p+d+i+n+u+g+s+b+md5+m` означает слежение за всеми стандартными атрибутами файла и использование хэш-функции MD5-Digest для слежения за целостностью содержимого файлов. `+e+t` означает контроль расширенных атрибутов: мандатной метки и флагов аудита, соответственно. Контроль ACL осуществляется при установке флага `+g`.

Правило GOST выглядит следующим образом:

```
GOST = p+d+i+n+u+g+s+b+gost+m+e+t
```

где `p+d+i+n+u+g+s+b+gost+m` означает слежение за всеми стандартными атрибутами файла и использование хэш-функции ГОСТ Р 34.11-94 для слежения за целостностью содержимого файлов. `+e+t` означает контроль расширенных атрибутов: мандатной метки и флагов аудита, соответственно. Контроль ACL осуществляется при установке флага `+g`.

Правило для каталогов:

```
DIR = p+i+n+u+g
```

Правило означает слежение за правами доступа, метаданными, количеством ссылок и другими стандартными атрибутами (подробнее см. /etc/afick.conf).

В файле конфигурации задаются пути к файлам и каталогам, контролируемых afick, например:

```
/boot             GOST
/lib/modules      PARSEC
/sbin             PARSEC
/lib64/security   PARSEC
/lib/security     PARSEC
/usr/sbin         PARSEC
/etc/security     PARSEC
/etc/pam.d        PARSEC
```

Кроме того, на выбор администратора представлен ряд дополнительных путей с правилами. Соответствующие строки помечены знаком комментария # и могут быть активированы снятием этого знака.

При запуске afick с параметром -i:

```
afick -i
```

будет создан файл /var/lib/afick/afick. Это и есть БД формата ndbm. Если посмотреть ее содержимое, то можно обнаружить набор строк, каждая из которых — имя файла и далее через пробел его атрибуты и сигнатуры.

БД защищается системой разграничения доступа.

При запуске AFICK автоматически установит ежедневное задание для CRON. Файл с заданием находится в /etc/cron.daily/afick_cron.

Параметр report_url:=stdout задает местоположение файла-отчета.

В конфигурационном файле есть простой язык макросов, который используется при определении переменных для заданий системного планировщика заданий cron.

14.4. Средства создания замкнутой программной среды

В ОС реализован механизм, обеспечивающий проверку неизменности и подлинности загружаемых исполняемых файлов формата ELF. Проверка производится на основе контрольных сумм файлов, вычисляемых в соответствии с ГОСТ Р 34.11-94, и ЭЦП, реализованной в соответствии с ГОСТ Р 34.10-2001, которые внедрены в исполняемые файлы формата ELF в процессе сборки ОС. Средства создания замкнутой программной среды предоставляют возможность внедрения цифровой подписи в исполняемые файлы формата ELF, входящие в состав устанавливаемого СПО (14.4.3).

Механизм контроля целостности исполняемых файлов и разделяемых библиотек формата ELF при запуске программы на выполнение реализован в модуле ядра ОС

digsig_verif.

Модуль digsig_verif является невыгружаемым модулем ядра Linux, который может функционировать в одном из следующих режимов:

– исполняемым файлам и разделяемым библиотекам с неверной ЭЦП, а также без ЭЦП загрузка на исполнение запрещается (штатный режим функционирования);

– исполняемым файлам и разделяемым библиотекам с неверной ЭЦП, а также без ЭЦП загрузка на исполнение разрешается, при этом выдается сообщение об ошибке проверки ЭЦП (режим для проверки ЭЦП в СПО);

– ЭЦП при загрузке исполняемых файлов и разделяемых библиотек не проверяется (отладочный режим для тестирования СПО).

14.4.1. Настройка модуля digsig_verif

Для изменения режима функционирования модуля digsig_verif необходимо отредактировать файл /etc/digsig/digsig_initramfs.conf в стартовом загрузочном образе initrd (14.4.2).

Для использования отладочного режима для тестирования СПО необходимо установить для параметра DIGSIG_LOAD_KEYS значение 0:

```
DIGSIG_LOAD_KEYS=0
```

Для использования режима для проверки ЭЦП в СПО необходимо установить для параметра DIGSIG_LOAD_KEYS значение 1:

```
DIGSIG_LOAD_KEYS=1
DIGSIG_ENFORCE=0
```

Для использования штатного режима функционирования необходимо установить следующие значения параметров:

```
DIGSIG_LOAD_KEYS=1
DIGSIG_ENFORCE=1
```

Управление модулем digsig_verif может осуществляться через интерфейс sysfs с использованием файлов:

– /sys/digsig/enforce — проверка и переключение режима работы;

– /sys/digsig/key — файл загрузки главного ключа;

– /sys/digsig/additional — файл загрузки дополнительных ключей.

Каждый дополнительный ключ, использованный для подписывания СПО (14.4.3), необходимо скопировать в каталог /etc/digsig/ в стартовом загрузочном образе initrd (14.4.2) с использованием команды:

```
# cp /<каталог>/<файл ключа> /tmp/initrd/unpacked/etc/digsig/
```

Для загрузки дополнительного ключа модулем digsig_verif необходимо отредактировать файл скрипта /etc/digsig/digsig_initramfs в стартовом загрузочном образе initrd (14.4.2), добавив в него после строки:

```
cat /etc/digsig/key_for_signing.gpg > /sys/digsig/additional 2>/dev/null
```

строку следующего вида:

```
cat /etc/digsig/<файл ключа> >> /sys/digsig/additional 2>/dev/null
```

14.4.2. Внесение изменений в стартовый загрузочный образ initrd

Для внесения изменений в стартовый загрузочный образ `initrd` необходимо выполнить следующие действия:

1) скопировать стартовый загрузочный образ `initrd` в отдельный каталог, выполнив команды:

```
# mkdir /tmp/initrd
# cp /boot/initrd.img-2.6.34-3-generic /tmp/initrd/
```

2) распаковать стартовый загрузочный образ `initrd`, выполнив команды:

```
# mkdir /tmp/initrd/unpacked
# cd /tmp/initrd/unpacked
# gunzip < ../initrd.img-2.6.34-3-generic | cpio -i --make-directories
```

После выполнения команд в каталоге `/tmp/initrd/unpacked` окажется содержимое стартового загрузочного образа;

3) внести необходимые изменения в стартовый загрузочный образ;

4) запаковать стартовый загрузочный образ, выполнив команды:

```
# cd /tmp/initrd/unpacked
# find | cpio -H newc -o | gzip -9 > ../initrd.img-2.6.34-3-generic
```

5) скопировать полученный образ на карту памяти, выполнив команду:

```
# cp /tmp/initrd/initrd.img-2.6.34-3-generic /boot/
```

14.4.3. Подписывание СПО

В модуле ядра `digsig_verif` реализован механизм, позволяющий использовать несколько ключей при подписывании файлов формата ELF.

Порядок использования ключей для `digsig_verif`:

1) главный ключ записывается в `/sys/digsig/key`;

2) дополнительные ключи записываются в `/sys/digsig/additional`.

Все дополнительные ключи должны быть подписаны главным ключом. Дополнительные ключи генерируются и передаются потребителю разработчиком ОС по заявке.

Подписывание пакетов должно осуществляться на инструментальной вычислительной машине под управлением ОС. Для выполнения подписывания необходимо, чтобы в инструментальной среде были установлены следующие пакеты: `bsign`, `binutils`, `gzip`, `lzma`, `bzip2`, `gnupg`.

Подписывание пакетов должно осуществляться от имени пользователя `root`. Закрытый ключ для подписывания должны быть импортирован в набор ключей пользователя

root командой:

```
gpg: ключ 078AC4F1: секретный ключ импортирован
gpg: ключ 078AC4F1: открытый ключ "CCT RusBITech (Key for signing)
      <mail@rusbitech.ru>" импортирован
gpg: Всего обработано: 1
gpg:                импортировано: 1
gpg:     прочитано секретных ключей: 1
gpg: импортировано секретных ключей: 1
```

Для подписывания пакетов, может быть использован скрипт, текст которого представлен ниже. Скрипт должен запускаться от имени пользователя root. В качестве первого аргумента должен быть указан полный путь до каталога, содержащего пакеты, которые необходимо подписать. В качестве второго аргумента должен быть указан каталог в который будут помещаться подписанные пакеты.

```
#!/bin/bash

DIR_BIN=$1
DIR_SIGNED=$2

TMP_DIR=$DIR_SIGNED/tmp$$

if [ -z $DIR_BIN ] ; then
echo "Specify original directory as first argument."
exit 1
fi

if [ -z $DIR_SIGNED ] ; then
echo "Specify destination directory as second argument."
exit 1
fi

if [ ! -e $DIR_BIN ] ; then
echo "Original directory $DIR_BIN doesn't exist."
exit 1
fi

list_of_packages=`find $DIR_BIN -type f -name "*.deb"`
list_of_udebs=`find $DIR_BIN -type f -name "*.udeb"`
#echo $list_of_packages
```

```
for i in $list_of_packages ; do
pack_name=`echo $i | awk '{sub(/^.+\//,"",$0) ; a=$0; print a}'`
mkdir -p $TMP_DIR/{control,data}
cp $i $TMP_DIR
pushd $TMP_DIR
ar x $pack_name

# Definig archives type
data_arch_type=`ls data.tar* | cut -d'.' -f3`
control_arch_type=`ls control.tar* | cut -d'.' -f3`
popd

# Unpack data archive
pushd $TMP_DIR/data
case $data_arch_type in
gz)
tar --same-permissions --same-owner -xzf ../data.tar.gz
;;
bz2)
tar --same-permissions --same-owner -xjf ../data.tar.bz2
;;
lzma)
tar --same-permissions --same-owner --lzma -xf ../data.tar.lzma
;;
esac
popd

# Unpack control archive
pushd $TMP_DIR/control
case $control_arch_type in
gz)
tar --same-permissions --same-owner -xzf ../control.tar.gz
;;
bz2)
tar --same-permissions --same-owner -xjf ../control.tar.bz2
;;
lzma)
```

```
tar --same-permissions --same-owner --lzma -xf ../control.tar.lzma
;;
esac
popd

# Sign files
pushd $TMP_DIR/data
for file in `find . -type f` ; do
oldstat=`stat -c %a $file`
bsign -s --pgoptions "--default-key=A42E56D6" $file
bsign -V $file | grep -v "not ELF64"
bsign -w $file | grep -v "not ELF64"
newstat=`stat -c %a $file`
[$newstat!=$oldstat] && echo "BSIGN_CHMOD_ERROR in $file" >> ${SIGN_LOG} 2>&1
done
popd

# Counting md5sums
pushd $TMP_DIR/control
if [ -e ./md5sums ] ; then
filenames=`cat md5sums | awk -F'  ' '{print $2}'`
popd
pushd $TMP_DIR/data
for j in $filenames
do
echo `md5sum $j` >> $TMP_DIR/control/md5sums.new
done
sed -e 's/\ /\ \ /g' $TMP_DIR/control/md5sums.new >
$TMP_DIR/control/md5sums.new_mod
popd
mv -f $TMP_DIR/control/md5sums.new_mod $TMP_DIR/control/md5sums
rm -f $TMP_DIR/control/md5sums.new
fi

# Packing back in deb
pushd $TMP_DIR/data
case $data_arch_type in
gz)
```

```
tar --same-permissions --same-owner -czf ../data.tar.gz .
;;
bz2)
tar --same-permissions --same-owner -cjf ../data.tar.bz2 .
;;
lzma)
tar --same-permissions --same-owner --lzma -cf ../data.tar.lzma .
;;
esac
popd

pushd $TMP_DIR/control
case $control_arch_type in
gz)
tar --same-permissions --same-owner -czvf ../control.tar.gz .
;;
bz2)
tar --same-permissions --same-owner -cjvf ../control.tar.bz2 .
;;
lzma)
tar --same-permissions --same-owner --lzma -cvf ../control.tar.lzma .
;;
esac
popd
pushd $TMP_DIR
ar rcs $TMP_DIR/$pack_name debian-binary control.tar.$control_arch_type
    data.tar.$data_arch_type
cp $pack_name  $DIR_SIGNED/
popd
rm -rf $TMP_DIR
done

for j in $list_of_udebs ; do
    cp $j $DIR_SIGNED
done
```

Для корректной работы скрипта в строке

```
# bsign -s --pgoptions "--default-key=A42E56D6" $file
```

необходимо указать идентификатор ключа (слово A42E56D6), с помощью которого

необходимо подписать пакет. Идентификатор ключа можно получить, используя команду:

```
# gpg --list-keys
```

Далее приведен пример выводы команды `gpg --list-keys`

Пример

```
/root/.gnupg/pubring.gpg
------------------------
pub    256E/A42E56D6 2010-06-16
uid                  CCT NPO RusBITech (Key for signing) <mail@rusbitech.ru>
```

Пример вызова скрипта для подписывания исполняемых модулей формата ELF, находящихся в каталоге `/tmp/orig_packs`, с помещением результатов подписывания в каталог `/tmp/signed_packs`, приведен ниже.

Пример

```
# sign.sh /tmp/orig_packs /tmp/signed_packs
```

Дополнительный ключ пользователя копируется в каталог `/etc/digsig/` ОС, под управлением которой будет функционировать СПО пользователя. Для загрузки дополнительного ключа пользователя модулем `digsig_verif` необходимо отредактировать файл скрипта `/etc/digsig/digsig_initramfs` (см. 14.4.1)

Для проверки правильности ЭЦП файла формата `ELF` используется утилита `bsign`:

```
keys@debian:~$ bsign -w test_elf
version: 1
id: bsign v1.0
hash: cf37 e2f4 6999 28d6 d486 67f6 1ba6 92ed 5173 570c 6b05 66c7 1605 7a16
5b8d 3df8
signature_size: 96
signature:
88 5e 04 00 22 5e 00 06 05 02 4a 3b 20 ec 00 0a
09 10 f6 9a 9f a1 ba 70 38 34 eb 33 00 ff 5b 29
aa fb 48 4f c8 86 f1 74 c0 9a c9 dc 4d 64 5b 92
c5 cb 77 c8 82 df 33 16 f7 19 7e 7d 29 94 00 ff
52 99 5a 33 86 f1 3f ed 98 3a c9 38 96 be ca dd
f9 c1 64 eb 69 06 4f 8c 81 f2 9e 2a 76 ec e7 8f
signer: BA703834EB3300FF
timestamp: 19 Jun 2009 09:23:56 (1245389036)
bsign: good hash found in 'test_elf'.
```

Подписанный файл формата `ELF` может выполняться.

```
keys@debian:~$ ./test_elf
hello world!
keys@debian:~$
```

15. СООБЩЕНИЯ АДМИНИСТРАТОРУ

При возникновении проблем в процессе функционирования ОС появляются диагностические сообщения трех типов: информационные, предупреждающие и сообщения об ошибках (примеры приведены в таблицах 42– 44, соответственно). Администратор должен проанализировать диагностические сообщения и принять меры по устранению появившихся проблем.

Таблица 42 – Информационные сообщения

Сообщение ОС	Что означает сообщение	Файл
`Setting hostname to <>`	Установка имени хоста <>	`hostname`
`Setting domainname to <>`	Установка имени домена как <>	`hostname`
`Statistics dump initiated`	Вывод статистики запущен	`named`
`Query logging is now on`	Регистрация очередей включена	`named`
`Query logging is now off`	Регистрация очередей выключена	`named`
`Unknown host`	Неизвестный хост	`dnsquery`
`Non reloadable zone`	Неперезагружаемая зона	`named`
`Reconfig initiated`	Переконфигурирование запущено	`named`
`Zone not found`	Зона не найдена	`named`

Таблица 43 – Предупреждающие сообщения

Сообщение ОС	Что означает сообщение	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 44 – Сообщения об ошибках

Сообщение ОС	Что означает сообщение	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`

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

БД — база данных
ЕПП — единое пространство пользователей
КСЗ — комплекс средств защиты
ЛВС — локальная вычислительная сеть
ОЗУ — оперативное запоминающее устройство
ОС — операционная система
ПО — программное обеспечение
СЗИ — средства защиты информации
СЗФС — сетевая защищенная файловая система
СУБД — система управления базами данных
СЭП — система обмена сообщениями электронной почты
ФС — файловая система

ACL — Access Control List (список контроля доступа)
ALD — Astra Linux Directory (единое пространство пользователей)
ARP — Address Resolution Protocol (протокол разрешения адресов)
BOOTP — Bootstrap Protocol (простой протокол динамической конфигурации хоста)
BSD — Berkeley Software Distribution (программное изделие Калифорнийского университета)
CIFS — Common Internet File System (общий протокол доступа к файлам Интернет)
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 (служба доменных имен)
FTP — File Transfer Protocol (протокол передачи файлов)
GID — Group Identifier (идентификатор группы)
HBA — Host-based Authentication (аутентификация на основе адресов узлов сети)
HTTP — HyperText Transfer Protocol (протокол передачи гипертекстовых файлов)
ICMP — Internet Control Message Protocol (протокол управляющих сообщений в сети Интернет)
IDE — Integrated Drive Electronics (встроенный интерфейс накопителей)
IMAP — Internet Message Access Protocol (протокол доступа к сообщениям в сети Интер-

нет)

IP — Internet Protocol (протокол Интернет)

IPC — InterProcess Communication (межпроцессное взаимодействие)

KDC — Key Distribution Center (центр распределения ключей)

LDAP — Lightweight Directory Access Protocol (легковесный протокол доступа к сервисам каталогов)

LPR — Line Printer Remote (удаленный линейный принтер)

LVM — Logical Volume Manager (менеджер логических томов)

MAC — Mandatory Access Control (мандатное управление доступом)

MDA — Mail Delivery Agent (агент доставки электронной почты)

MTA — Mail Transfer Agent (агент пересылки сообщений)

MTU — Maximum Transfer Unit (максимальная единица передачи)

MUA — Mail User Agent (клиент электронной почты)

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 (сетевой протокол времени)

PAM — Pluggable Authentication Modules (подгружаемые аутентификационные модули)

POP3 — Post Office Protocol Version 3 (почтовый протокол, версия 3)

RFC — Request For Comments (общее название технических стандартов сети Интернет)

SASL — Simple Authentication and Security Layer (простая аутентификация и слой безопасности)

SCSI — Small Computer System Interface (системный интерфейс малых компьютеров)

SMB — Session Message Block (блок сессионных сообщений)

SQL — Structured Query Language (язык структурированных запросов)

SSL — Secure Sockets Layer (протокол защищенных сокетов)

SSH — Secure Shell Protocol (протокол передачи информации в зашифрованном виде)

TCP — Transmission Control Protocol (протокол передачи данных)

TOS — Type of Service (тип сервиса)

TTL — Time To Live (время жизни IP-пакета)

UDP — User Datagram Protocol (протокол пользовательских дейтаграмм)

UID — User Identifier (идентификатор пользователя)

UTC — Universal Time Coordinated (универсальное скоординированное время)

VFS — Virtual File System (виртуальная файловая система)

Лист регистрации изменений

Изм.	Номера листов (страниц)				Всего листов (страниц) в докум.	№ документа	Входящий № сопроводительного докум. и дата	Подп.	Дата
	измененных	замененных	новых	аннулированных					