

Глава 3

Основополагающие навыки работы в командной строке.

ЦЕЛИ СЕРТИФИКАЦИИ

3.01 Шел

3.02 Стандартные инструменты командной строки

3.03 Управление Текстовые файлы

3.04 Локальная онлайн документация

3.05 Сетевой учебник

3.06 Конфигурация сети и устранение неисправностей

✓ Двухминутная тренировка

Q & A Самопроверка

Экзамены Red Hat - сложная задача. В этой главе рассматриваются требования RHCSA, которые ранее были указаны в качестве предварительных условий для уже устаревшей сертификации RHCT. Многие из этих требований определяют базовые инструменты командной строки, связанные с сертификацией начального уровня, например те, которые предлагаются Linux Professional Institute.

Навыки командной строки больше не указаны в качестве предварительных условий, но они необходимы для достижения целей экзамена. Поскольку большинство кандидатов на экзамен RHCSA уже должны быть знакомы с инструментами командной строки, в этой главе рассматриваются связанные темы с минимальными подробностями. Если после прочтения этой главы вы почувствуете необходимость в дополнительных руководствах по этим темам, прекрасные книги по Linux, описанные в главе 1, могут вам в этом помочь.

Гуру Linux должны признать, что мы "упрощенно" представили ряд объяснений, чтобы сделать эту главу максимально короткой. Однако, поскольку большинство ИТ-специалистов являются специалистами, вы можете почувствовать некоторую неуверенность в отношении нескольких тем в этой главе. Это нормально. На самом деле, естественно, что многие опытные администраторы Linux не часто используют каждую команду. Многие кандидаты успешно могут заполнить пробелы в своих знаниях с помощью самостоятельного изучения и практики.

Внутри экзамена

Оболочки

Соответствующая цель экзамена RHCSA довольно общая:

- Получите доступ к приглашению оболочки и введите команды с правильным синтаксисом

Оболочкой по умолчанию для Linux является **bash**, «оболочка Bourne-Again». Фактически в первоначальном выпуске целей RHCSA было указано использование **bash**. Хотя многие Linux-гуру используют одну из множества доступных оболочек, на экзамене весьма вероятно, что вы столкнетесь с **bash**.

Какую бы оболочку вы ни предпочли, вам нужно знать, как перейти к приглашению оболочки и запускать обычные команды из этого приглашения. Некоторые основные команды описаны в некоторых других целях. Довольно просто открыть приглашение оболочки из консоли и в графическом интерфейсе.

Конвейеры и Перенаправление

Данные в оболочку и из нее часто рассматриваются в Linux как поток информации. Одним из основных навыков Linux является способность перенаправлять такие потоки. Как описано в требованиях RHCSA, это способность

- Использовать перенаправление ввода/вывода (>, >>, |, 2> и т. д.)

Операторы в скобках могут перенаправлять потоки из вывода команды, ошибки команды, файлов данных и многого другого.

Управление файлами и каталогами

Теперь, когда у вас есть доступ к командной строке, управление файлами и каталогами будет следующим. С помощью связанных команд вы можете перемещаться по дереву каталогов Linux, а также выполнять все задачи, предложенные в соответствующих заданиях:

- Создавать, удалять, копировать и перемещать файлы и каталоги
- Создавайте жесткие и символические ссылки

Анализ вывода текста

Большинство файлов конфигурации Linux являются текстовыми файлами. Важно понимать и анализировать поток текста при его отправке через оболочку. Такие инструменты, как команда **grep**, могут помочь вам сосредоточиться на необходимой информации. В этой главе вы узнаете, как достичь следующей цели:

- Используйте **grep** и регулярные выражения для анализа текста

Разнообразие локальной документации

Интернет недоступен во время экзаменов Red Hat, но это нормально. Google не твой единственный друг. Linux имеет отличную документацию, установленную с большинством пакетов. Командные руководства также доступны. Следующая цель главы описывает команды и каталог, связанный с большинством онлайн-документации по Linux:

- Найдите, прочитайте и используйте системную документацию, включая **man**, **info** и файлы в **/usr/share/doc**

Цели включают в себя интересное замечание:

- Примечание: Red Hat может использовать приложения во время экзамена, которые не включены в Red Hat Enterprise Linux, для оценки способностей кандидата для достижения этой цели.

Большинство разработчиков Linux следуют базовым параметрам, только что описанным для системной документации. Означает ли «примечание» в Red Hat, что они «скрывают» некоторую ключевую информацию на странице руководства или в файле в каталоге **/usr/share/doc**? Формулировка предполагает, что вы должны быть готовы к такому сценарию.

Использование текстовых редакторов

Чтобы настроить Linux, вам нужно знать, как редактировать текстовые файлы. А для новичков в Linux это требует другой парадигмы. Хотя текстовые процессоры, такие как OpenOffice.org Writer и Microsoft Word, могут сохранять файлы в текстовом формате, ошибка, связанная с файлом конфигурации ключа, может привести к невозможности загрузки системы

Linux, и эти редакторы могут внедрить скрытые данные или иным образом вызвать проблемы при использовании для простого редактирования текста. Следовательно, вам нужно знать, как справиться со следующей задачей с помощью стандартных утилит, не использующих графический интерфейс:

- Создавать и редактировать текстовые файлы

Управление Сетевыми сервисами

Хотя существуют отличные инструменты с графическим интерфейсом, помогающие управлять сетевыми службами, с такими инструментами слишком легко совершать ошибки. Инструменты командной строки могут помочь вам понять и управлять сетевыми службами напрямую или через соответствующие файлы конфигурации. Связанные с этим цели

- Запуск, остановка и проверка состояние сетевых служб

Конечно, эта цель требует базового понимания IP-сетей.

Конфигурация сети и разрешение имен

Разрешение имен зависит от баз данных имен хостов или полных доменных имен (FQDN), таких как **server1.example.com**, и **IP-адресов**, таких как **192.168.122.50**. Источниками, из которых Linux получает информацию о разрешении имен, обычно являются локальная база данных **/etc/hosts** с именами хостов и IP-адресами, а также доступные базы данных серверов службы доменных имен (DNS). Это интерпретация следующей цели RHCSA:

- Сконфигурируйте сеть и разрешение имени хоста статически или динамически

Когда экзамены RHCSA были впервые выпущены, это были две разные задачи при сертификации. Хотя эти цели уже официально не действуют, они предоставляют больше информации о том, что означает настройка сети и разрешения имен хостов:

- Управление сетевыми устройствами: понимание основы IP-сети/IP-маршрутизации, настройка IP-адресов/маршрута по умолчанию статически или динамически
- Управление разрешением имен: установите локальное имя хоста, настройте **/etc/hosts**, настройте использование существующего DNS-сервера.

Хотя устранение неполадок с сетью больше не является частью экзамена Red Hat начального уровня, способ решения проблем, связанных с конфигурацией сети и разрешением имени хоста, может помочь вам лучше понять, как работают сети.

ЦЕЛЬ СЕРТИФИКАЦИИ 3.01

Shells

Оболочка (**Shells**) - это пользовательский интерфейс. Текстовая оболочка также используется в качестве интерпретатора командной строки. В Linux оболочка является интерпретатором, который позволяет вам взаимодействовать с операционной системой с помощью различных команд. С правильными правами доступа к файлам вы можете настроить команды в сценариях так, чтобы они выполнялись по мере необходимости, даже среди ночи. Оболочки Linux могут обрабатывать команды в различной последовательности, в зависимости от того, как вы управляете вводом и выводом каждой команды. Способ интерпретации команд частично определяется переменными и параметрами, связанными с каждой оболочкой.

Оболочкой по умолчанию в Linux является **bash**, также известная как **Bourne-Again Shell**. Основное внимание к командам в этой книге основано на том, как они используются в bash. Тем не менее, существует ряд других оболочек, которые популярны у многих пользователей. Пока установлены соответствующие RPM, пользователи могут запускать любые из этих оболочек. При желании вы можете изменить оболочку по умолчанию для отдельных пользователей в файле **/etc/passwd**.

Другие оболочки (SHELLS)

Пользователи могут выбирать между четырьмя оболочками командной строки в RHEL 7. Хотя bash используется по умолчанию, давние пользователи Linux и Unix могут предпочесть что-то другое:

bash Оболочка **Bourne-Again** по умолчанию, основанная на интерпретаторе командной строки, первоначально разработанном Стивеном Борном.

ksh Оболочка **Korn**, разработанная Дэвидом Корном в **Bell Labs** в 1980-х годах, вобрала в себя лучшие черты оболочки Bourne и C.

tcsh Улучшенная версия оболочки Unix C.

zsh Сложная оболочка, похожая на оболочку Корна.

Эти оболочки находятся в каталоге **/bin**. Если пользователь предпочитает один из этих параметров в качестве оболочки по умолчанию, его легко изменить. Самый прямой способ - изменить оболочку по умолчанию в файле **/etc/passwd**. Например, строка, которая относится к одному из постоянных аккаунтов авторов:

michael:x:1000:1000:Michael Jang:/home/michael:/bin/bash

Например, чтобы изменить оболочку по умолчанию на **ksh**, измените **/bin/bash** на **/bin/ksh**. Вам также необходимо установить соответствующий пакет RPM для оболочки Korn. Управление пакетами будет рассмотрено в главе 7.

Виртуальные Терминалы

Если у вас есть доступ к консоли системы RHEL, вы можете использовать шесть виртуальных терминалов, чтобы открыть шесть независимых сеансов входа в систему. Однако по умолчанию активирован только один виртуальный терминал. Другие запросы на вход в систему запускаются динамически при переключении на неиспользуемый терминал. Виртуальные терминалы определяются файлом **logind.conf** в каталоге **/etc/systemd**. Посмотрите на этот файл. Вы увидите опцию **NAutoVTs**, которая определяет максимальное количество виртуальных терминалов, которые могут быть активированы. Виртуальные терминалы связаны с файлами устройств **/dev/tty1 - /dev/tty6**. Когда графический интерфейс настроен, он принимает значение **/dev/tty1**. Можно настроить больше виртуальных терминалов, ограниченных теми, которые разрешены для пользователя с правами администратора в файле **/etc/securetty**.

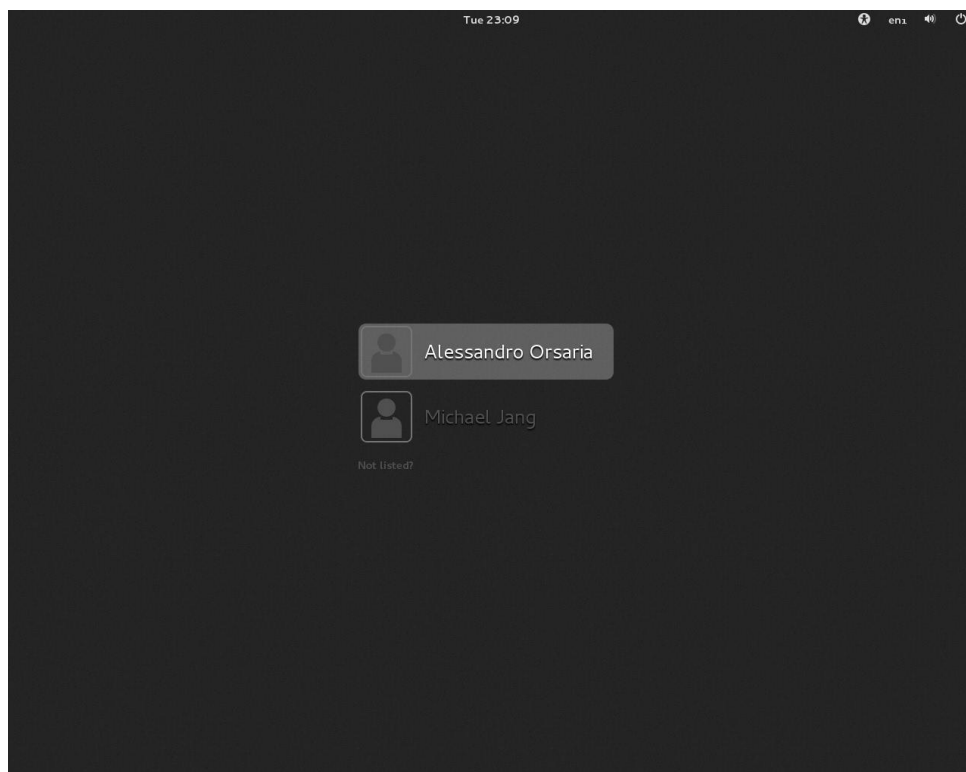
Обычно для переключения между виртуальными терминалами нажмите **alt** и функциональную клавишу, связанную с этим терминалом. Например, комбинация клавиш **alt-f2** перемещается на второй терминал. Однако в графическом интерфейсе RHEL комбинации клавиш **alt-fn** используются для предоставления других функций, таких как запуск инструмента «Запуск приложения» через **alt-f2**. Поэтому вам нужно нажать **ctrl-alt-fn** для перехода на n-ю виртуальную консоль из графического интерфейса.

!!!! EXAM Watch !!!!!

Несмотря на то, что это должно быть тривиально для большинства пользователей Linux, одной из целей RHCSA является «доступ к приглашению оболочки». Теперь вы должны знать, как настроить доступ к различным приглашениям оболочки.

!!!!

РИСУНОК 3-1 Первая консоль входа в систему с графическим интерфейсом



При входе в систему с текстовой консоли вы увидите следующее приглашение, которое немного зависит от выпуска RHEL, номера версии ядра и имени хоста системы:

Red Hat Enterprise Linux Server
Ядро 3.10.0-123.el7.x86_64 на x86_64
логин server1:

Графический вход в систему, который требует установки диспетчера отображения GNOME (GDM), является более интуитивным, как показано на рисунке 3-1.

Интерфейсы GUI Shell

После входа в GUI доступ к оболочке `bash` становится легким. Если вы находитесь в рабочей среде GNOME по умолчанию, щелкните **Приложения | Системные | Терминал (Applications | System tools «Utilities» | Terminal)**. Традиционно администраторы работали с консоли. Но во многих случаях работа с командной строкой из графического интерфейса может быть полезной, особенно с окнами, которые могут быть расположены рядом. Щелчок правой кнопкой мыши на экране терминала с графическим интерфейсом пользователя позволяет открывать дополнительные терминалы в разных окнах или на вкладках. Он также поддерживает копирование и вставку по мере необходимости. Снимки экрана командной строки для этой книги взяты из командной строки на основе графического интерфейса, частично потому, что темный текст на белом экране легче читать.

Различия между обычными и административными пользователями

То, что вы можете сделать в командной строке, зависит от привилегий, связанных с учетной записью для входа. Доступны две основные подсказки. Ниже приведен пример того, что вы можете увидеть, войдя в систему как обычный пользователь:

[michael@server1 ~]\$

Обратите внимание, как оно включает имя пользователя, имя хоста локальной системы, текущий каталог и приглашение \$. Подсказка \$ является стандартом для обычных пользователей. Как отмечено во введении к книге, примеры команд, запускаемых из учетной записи обычного пользователя, показывают только следующее:

\$

Напротив, взгляните на приглашение для пользователя с правами администратора в той же системе. Это должно выглядеть знакомо. За исключением имени учетной записи, единственное непротиворечивое отличие - это подсказка.

[root@server1 ~] #

В этой книге примеры команд, запускаемых от **учетной записи администратора root**, показывают следующее:

#

Помимо владения и разрешений, другие различия между обычными и административными учетными записями обсуждаются в главе 8.

Текстовые потоки и перенаправление команд

Linux использует три основных потока данных. Данные поступают, данные выходят, а ошибки отправляются в другом направлении. Эти потоки известны, как **стандартный ввод (stdin)**, **стандартный вывод (stdout)** и **вывод стандартных ошибок (stderr)**. Обычно ввод осуществляется с клавиатуры, тогда как стандартный вывод и стандартная ошибка выводятся на экран терминала. В следующем примере, когда вы запускаете **cat filename**, содержимое этого файла отправляется на экран как стандартный вывод (как и любые ошибки):

cat filename

Вы можете перенаправить каждый из этих потоков в файл или из файла. Например, если у вас есть программа с именем **database** и **datafile** с большим количеством данных, содержимое этого файла данных можно отправить в программу базы данных с помощью стрелки перенаправления влево (<). Как показано здесь, файл данных берется как стандартный ввод:

database < datafile

Стандартный ввод может также поступать с левой стороны команды. Например, если вам нужно прокрутить загрузочные сообщения, вы можете объединить команды **dmesg** и **less** конвейером:

dmesg | less

Вывод **dmesg** перенаправляется, как стандартный ввод утилите **less**, что позволяет вам прокручивать этот вывод, как если бы он был отдельным файлом.

Стандартный вывод так же легко перенаправить. Например, следующая команда использует стрелку перенаправления вправо (>) для отправки стандартного вывода команды **ls** в файл с именем **filelist**:

ls> filelist

Вы можете добавить стандартный вывод в конец существующего файла с помощью стрелки двойного перенаправления с помощью команды, такой, как **ls >> filelist**.

Если вы хотите сохранить сообщения об ошибках программы в файл, перенаправьте поток ошибок из него с помощью следующей команды:

program 2> err-list

Иногда вы можете отказаться от всех ошибок. Это может быть достигнуто путем перенаправления потока ошибок в специальный файл устройства **/dev/null**:

программа 2> / dev / null

Другим полезным оператором перенаправления является амперсанд и стрелка вправо (**&>**), которые отправляют, как стандартный вывод, так и ошибку в файл или устройство. Пример показан здесь:

program &> output-and-error

!!!! EXAM Watch !!!!!

Символы перенаправления команд, такие как **>**, **>>**, **2>** и **|** связаны с целью «перенаправление ввода/вывода» в задачах экзамена RHCSA.
!!!!

ЦЕЛЬ СЕРТИФИКАЦИИ 3.02

Стандартные инструменты командной строки

В то время как новые пользователи Linux могут предпочесть использовать графический интерфейс, наиболее эффективный способ администрирования Linux - это интерфейс командной строки. Несмотря на то, что доступны отличные инструменты с графическим интерфейсом, внешний вид этих инструментов сильно различается в зависимости от распределения. Напротив, если вы знаете стандартные инструменты командной строки, вы сможете работать в любом дистрибутиве Linux.

Помните, что в любом сеансе **bash** вы можете просмотреть историю предыдущих команд, используя клавиши со стрелками вверх и вниз, и **ctrl-R**, чтобы выполнить поиск. Вы также можете воспользоваться завершением текста, которое позволяет использовать клавишу табуляции почти, как подстановочный знак для завершения команды, имени файла или переменной (если текст начинается с символа \$).

!!!! EXAM Watch !!!!!

В этом разделе рассматриваются только самые основные команды, доступные в Linux. Он описывает только несколько возможностей каждой команды. Тем не менее, он позволяет вам «выдавать команды с правильным синтаксисом», как описано в целях RHCSA.
!!!!

Почти все команды Linux включают опции (или «переключатели, ключи») и аргументы. Опции команды позволяют вам изменить поведение команды и обычно начинаются с одной или двух черточек (например, **ls -a** или **ls --all**). Аргументы указывают файлы, устройства или другие цели, с которыми должна работать команда. В этой главе рассматриваются только несколько команд. Если вы менее знакомы с некоторыми из них, используйте их справочные страницы. Изучите параметры команды. Попробуйте их использовать! Только с практикой,

практикой и большей практикой вы сможете по-настоящему понять всю мощь некоторых из этих команд.

Две основные группы команд используются для управления файлами Linux. Одна группа поможет вам обойти файлы и каталоги Linux. Другая группа на самом деле делает что-то творческое с файлами. Эти команды будут рассмотрены в следующих разделах, но сначала мы рассмотрим некоторые основные концепции файловой системы.

Концепции файлов и каталогов

Как отмечалось ранее, все в Linux может быть сведено к файлу. Каталоги - это специальные типы файлов, которые служат контейнерами для других файлов. Для навигации и поиска важных файлов, вам нужны некоторые основные команды и концепции, чтобы рассказать вам, где вы находитесь и как перейти от каталога к каталогу. Наиболее важной командой является **pwd**; переменной, которая всегда ведет в домашний каталог пользователя, является тильда (~); и концепцией, которая описывает, где вы находитесь в дереве каталогов Linux - это путь. С ними тесно связан поиск в каталогах когда вводится команда, основанная на переменной окружения, известной как **PATH**. Как только эти понятия будут поняты, вы сможете перемещаться между каталогами с помощью команды **cd**.

PWD

В интерфейсе командной строки текущий каталог может находиться либо в корневом каталоге (/) верхнего уровня, либо в подкаталоге. Команда **pwd** идентифицирует текущий каталог. Попробуйте. Выполнение команды выдаст вам имя каталога относительно корневого каталога верхнего уровня (/). Зная эту информацию, вы можете при необходимости перейти в другой каталог. Кстати, **pwd** - это сокращение от печать рабочего каталога (которое не имеет ничего общего с современными принтерами, но учитывает дни, когда вывод печатался на телетайпе). Например, когда пользователь michael запускает эту команду в своем домашнем каталоге, он получает следующий вывод:

```
/home/michael
```

Тильда (~)

После стандартного входа в систему каждый пользователь Linux попадает в домашний каталог. Тильду (~) можно использовать для представления домашнего каталога любого активного в данный момент пользователя. Например, когда пользователь входит в систему, он попадает в свой домашний каталог **/home/john**. В отличие от этого, домашний каталог пользователя **root** - **/root**.

Таким образом, эффект от команды **cd ~** зависит от вашего имени пользователя. Например, если вы вошли в систему как пользователь **mj**, команда **cd ~** переместится в каталог **/home/mj**. Если вы вошли в систему как пользователь **root**, эта команда перейдет в каталог **/root**. Вы можете получить список содержимого вашего домашнего каталога из любой точки дерева каталогов используя команду **ls ~**. Команды **cd** и **ls** описаны вкратце. Когда вы входите в систему, как пользователь **root** и запускаете команду **ls**, вы должны увидеть следующее:

```
anaconda-ks.cfg initial-setup-ks.cfg
```

Кстати, эти файлы описывают то, что произошло в процессе установки, установленные пакеты, а также пользователей и группы, добавленные в локальную систему. Файл **anaconda-ks.cfg** важен для автоматических установок **Kickstart**, как описано в Главе 2.

Пути к каталогам

Вам нужно знать две концепции путей при работе с каталогами Linux: **абсолютные и относительные пути**. Абсолютный путь описывает полную структуру каталогов в терминах

каталога верхнего уровня, **root (/)**. Относительный путь основан на текущем каталоге. Относительные пути не включают косую черту впереди.

Разница между абсолютным и относительным путями важна. Особенно, когда вы запускаете команду, абсолютные пути очень важны. В противном случае команды, ссылающиеся на неправильный каталог, могут привести к непредвиденным последствиям. Например, допустим, что вы находитесь в корневом каталоге верхнего уровня и сделали резервную копию каталога **/home**, используя его относительный путь к корню (**/**). Если вы оказались в каталоге **/home** при восстановлении этой резервной копии, файлы для пользователя **michael**, например, будут восстановлены в каталог **/home/home/michael**.

Напротив, если каталог **/home** был сохранен с использованием абсолютного пути, текущий рабочий каталог не имеет значения, когда вы восстанавливаете эти файлы. Эта резервная копия будет восстановлена в правильные каталоги.

Окружение PATH

Строго говоря, при выполнении команды вы должны указать полный путь к этой команде. Например, поскольку команда **ls** находится в каталоге **/bin**, пользователи должны запустить команду **/bin/ls**, чтобы вывести список файлов в текущем каталоге. С помощью PATH, переменной среды, это не требуется. оболочка, такая как **bash**, автоматически ищет в каталогах, перечисленных в **пользовательской переменной PATH**, команду, которую пользователь только что набрал в командной строке. Переменные среды постоянны от консоли к консоли.

Чтобы определить PATH для текущей учетной записи пользователя, выполните команду **echo \$PATH**. Вы должны увидеть серию каталогов в выводе. Различия между PATH для обычного пользователя и для корневого пользователя в RHEL 7 сузились:

```
$echo $PATH
```

```
/usr/local/bin:/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/home/michael/.local/bin:/home/michael/bin
```

```
# echo $PATH
```

```
/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/root/bin
```

Каталоги в **PATH** для обычных пользователей и пользователей с правами администратора немного отличаются. Различия имеют значение, потому что каталоги ищутся по порядку. Например, команда **system-config-keyboard** доступна как из каталога **/usr/bin**, так и из **/usr/sbin**. Как видно из стандартного PATH для обычных и корневых пользователей, используемая версия различается из-за различий в PATH.

PATH определяется глобальными текущими настройками в файле **/etc/profile** или скриптами в каталоге **/etc/profile.d**. Вы можете заметить различия между **PATH**, настроенным для идентификатора пользователя (UID) 0, и всеми другими пользователями. UID 0 соответствует пользователю с правами администратора.

Помимо глобальных настроек, **PATH** для отдельных пользователей можно настроить с помощью соответствующей записи в домашнем каталоге этого пользователя, в скрытых файлах с именем **~/.bash_profile** или **~/.profile**.

cd

В Linux легко менять каталоги. Просто используйте **cd** и приведите абсолютный путь к нужному каталогу. Если вы используете относительный путь, просто помните, что назначение зависит от текущего рабочего каталога.

По умолчанию команда **cd** сама перемещается в ваш домашний каталог. Тильда не требуется для этой команды. Другой общий ярлык - две последовательные точки (**..**) для представления каталога, который находится на один уровень выше в иерархии. Таким образом, **cd ..** перемещается в родительский каталог текущего рабочего каталога.

Списки файлов и ls

Теперь, когда вы ознакомились с этими командами, которые могут перемещаться из одного каталога в другой, пришло время посмотреть, какие файлы существуют в каталоге, и это является областью действия команды **ls**.

Команда Linux **ls** с правильными опциями (ключами) может быть довольно мощной. Правильный тип **ls** может рассказать вам все о файле, например, время последней модификации, время последнего доступа и размер. Это может помочь вам организовать распечатку файлов практически в любом желаемом порядке. Важные варианты этой команды включают **ls -a** для отображения скрытых файлов, **ls -l** для отображения длинных списков, **ls -t** для вывода списка, отсортированного по времени модификации, и **ls -i** для номеров инодов (inode - это внутренние структуры данных в файловой системе, которые хранят информация о файле). Другими полезными параметрами команды являются **-g**, чтобы изменить порядок листинга, и **-R**, чтобы рекурсивно вывести список содержимого всех подкаталогов.

Вы можете комбинировать опции (ключи); мы часто используем команду **ls -ltr** для отображения рекурсивного длинного списка с последними самыми последними измененными файлами. Ключ **-d**, в сочетании с другими, может дать вам больше информации о текущем каталоге или о каталоге, который вы передали в качестве аргумента команде **ls**.

Одной из важных функций, которая возвращает контексты **SELinux**, является команда **ls -Z**. Посмотрите на вывод на рисунке 3-2. Выходные данные **system_u**, **object_r**, **var_t** и **s0** демонстрируют текущие контексты **SELinux** отмеченных файлов. Во время экзамена RHCSA (и RHCE) вы должны будете сконфигурировать систему с включенным **SELinux**. Начиная с главы 4, эта книга описывает, как **SELinux** можно настроить для каждой установленной службы.

Команды создания файла

Для создания новых файлов используются две команды: **touch** и **cp**. Кроме того, вы можете позволить текстовому редактору, например **vi**, создать новый файл. Конечно, хотя команды **ln**, **mv** и **rm** не создают файлы, они управляют ими связанными способами.

РИСУНОК 3-2 Текущие контексты SELinux

```
[root@server1 ~]# \ls -Z /var/
drwxr-xr-x. root root system_u:object_r:acct_data_t:s0 account
drwxr-xr-x. root root system_u:object_r:var_t:s0 adm
drwxr-xr-x. root root system_u:object_r:var_t:s0 cache
drwxr-xr-x. root root system_u:object_r:kdump_crash_t:s0 crash
drwxr-xr-x. root root system_u:object_r:var_t:s0 db
drwxr-xr-x. root root system_u:object_r:var_t:s0 empty
drwxr-xr-x. root root system_u:object_r:public_content_t:s0 ftp
drwxr-xr-x. root root system_u:object_r:games_data_t:s0 games
drwx--x--x. gdm gdm system_u:object_r:xserver_log_t:s0 gdm
drwxr-xr-x. root root system_u:object_r:var_t:s0 gopher
drwxr-xr-x. root root system_u:object_r:var_t:s0 kerberos
drwxr-xr-x. root root system_u:object_r:var_lib_t:s0 lib
drwxr-xr-x. root root system_u:object_r:var_t:s0 local
lrwxrwxrwx. root root system_u:object_r:var_lock_t:s0 lock -> ../run/lock
drwxr-xr-x. root root system_u:object_r:var_log_t:s0 log
lrwxrwxrwx. root root system_u:object_r:mail_spool_t:s0 mail -> spool/mail
drwxr-xr-x. root root system_u:object_r:var_t:s0 nis
drwxr-xr-x. root root system_u:object_r:var_t:s0 opt
drwxr-xr-x. root root system_u:object_r:var_t:s0 preserve
lrwxrwxrwx. root root system_u:object_r:var_run_t:s0 run -> ../run
drwxr-xr-x. root root system_u:object_r:var_spool_t:s0 spool
drwxrwxrwt. root root system_u:object_r:tmp_t:s0 tmp
drwxr-xr-x. root root system_u:object_r:var_t:s0 var
drwxr-xr-x. root root system_u:object_r:httpd_sys_content_t:s0 www
drwxr-xr-x. root root system_u:object_r:var_yp_t:s0 yp
[root@server1 ~]# █
```

touch

Возможно, самый простой способ создать новый файл - использовать команду **touch**. Например, команда **touch abc** создает пустой файл с именем **abc** в локальном каталоге. Команда **touch** также используется для изменения времени последней модификации файла. Например, попробуйте следующие три команды:

```
# ls -l /etc/passwd
# touch /etc/passwd
# ls -l /etc/passwd
```

Обратите внимание на временные метки, указанные в выходных данных каждой команды **ls -l**, и сравните их с текущей датой и временем, возвращаемым по дате. После запуска команды **touch** временная метка **/etc/passwd** обновляется до текущей даты и времени.

cp

Команда **cp (copy)** позволяет вам взять содержимое одного файла и поместить копию с тем же или другим именем в каталог по вашему выбору. Например, команда **cp file1 file2** берет содержимое файла **file1** и сохраняет содержимое файла **file2** в текущем каталоге. Одна из опасностей **cp** заключается в том, что он может легко перезаписывать файлы в разных каталогах, не предлагая вам убедиться, что это именно то, что вы действительно хотели сделать.

Другое использование команды **cp** - копирование нескольких источников в один каталог назначения. В этом случае синтаксис **cp file1 file2 ... dir**.

Команда **cp** с ключом **-a** поддерживает рекурсивные изменения и сохраняет все атрибуты файла, такие как разрешения, владельцы и временные метки. Например, следующая команда копирует все подкаталоги указанного каталога вместе со связанными файлами в **/mnt/backup**:

```
# cp -a /home/michael/. /mnt/backup/
```

mv

Хотя в Linux нет команды «переименовать», вы можете использовать **mv**. Команда **mv** помещает другую метку в файл. Например, команда **mv file1 file2** меняет имя *file1* на *file2*. Если вы не перемещаете файл в другую файловую систему, все в этом файле, включая номер индекса, остается неизменным. Команда **mv** работает и с каталогами.

ln

Связанные файлы позволяют пользователям обращаться к одному и тому же файлу под разными именами. Когда связанные файлы являются устройствами, они могут представлять более распространенные имена, такие, как **/dev/cdrom**. Ссылки на файлы могут быть **жесткими** или **символическими**.

Жесткие ссылки - это записи каталога, которые указывают на один и тот же индекс. Они должны быть созданы в одной файловой системе. Вы можете удалить файл с жесткой связью в одном каталоге, и он все еще будет существовать в другом каталоге (файлы удаляются только тогда, когда число записей **dentry**, указывающих на них, **достигает 0**, что отслеживается счетчиком на файл). Например, следующая команда создает жесткую ссылку из фактического файла конфигурации **Samba** на **файл smb.conf** в локальном каталоге:

```
# ln /etc/samba/smb.conf smb.conf
```

С другой стороны, **символическая ссылка** служит перенаправлением; Когда вы открываете файл, созданный с помощью символической ссылки, эта ссылка перенаправляет вас на исходный файл. Если вы удалите исходный файл, файл будет потерян. Хотя символическая ссылка сохранится, просто она будет указывать на несуществующий файл. Следующая команда является примером того, как вы можете создать файл с символическими связями:

```
# ln -s /etc/samba/smb.conf smb.conf
```

rm

Команда **rm** несколько опасна. В командной строке **Linux** нет мусорной корзины. Поэтому, если вы удалите файл с помощью команды **rm**, восстановить этот файл в лучшем случае будет сложно.

Команда **rm** является мощной. Например, когда мы загрузили исходные файлы для ядра Linux несколько тысяч файлов были включены в каталог **/root/rpmbuild/BUILD/kernel-3.10.0-123.el7**. Очевидно, что нецелесообразно удалять эти файлы по одному. Поэтому команда **rm** включает в себя несколько мощных ключей. Следующая команда удаляет все эти файлы за один раз:

```
# rm -rf /root/rpmbuild/BUILD/kernel-3.10.0-123.el7
```

Ключ -r работает рекурсивно, а **ключ -f** отменяет любые меры предосторожности, такие как **ключ -i**, показанный в выходных данных команды **alias** для пользователя с правами администратора. Это все еще довольно опасная команда. Например, простая ошибка ввода, такая, как вставка пробела между первой косой чертой и именем каталога, как показано здесь, сначала приведет к удалению каждого файла, начиная с корневого каталога верхнего уровня (/), прежде чем искать

Подкаталог **root/rpmbuild/BUILD/kernel-3.10.0-123.el7**:

```
# rm -rf / root/rpmbuild/BUILD/kernel-3.10.0-123.el7
```

Это приведет к удалению всех файлов в системе, включая любые точки монтирования.

Создание и удаление каталогов

Команды **mkdir** и **rmdir** используются для создания и удаления каталогов. Способы использования этих команд зависят от уже обсужденных концепций абсолютных и относительных путей. Например, следующая команда создает подкаталог **test** для текущего каталога. Если вы находитесь в каталоге **/home/michael**, полный путь будет **/home/michael/test**.

```
# mkdir test
```

В качестве альтернативы следующая команда создает каталог **/ test**:

```
# mkdir /test
```

При желании вы можете использовать следующую команду для создания серии каталогов:

```
# mkdir -p test1/test2/test3
```

Эта команда эквивалентна следующим командам:

```
# mkdir test1
```

```
# mkdir test1/test2
```

```
# mkdir test1/test2/test3
```

И наоборот, команда **rmdir** удаляет каталог, только если он пуст. Если вы выполняете очистку после предыдущих команд **mkdir**, здесь также **полезен ключ -p**. Следующая команда удаляет указанный каталог и подкаталоги, если все каталоги пусты:

```
# rmdir -p test1/test2/test3
```

alias

Команда **alias** может быть использована для упрощения нескольких команд. Для пользователя с правами администратора псевдонимы по умолчанию обеспечивают некоторую безопасность. Чтобы увидеть псевдонимы для текущего пользователя, выполните команду **alias**. Следующий вывод показывает **alias** Red Hat по умолчанию для пользователя **root**:

```
alias cp='cp -i'
alias egrep='egrep --color=auto'
alias fgrep='fgrep --color=auto'
alias grep='grep --color=auto'
alias l.='ls -d .* --color=auto'
alias ll='ls -l --color=auto'
alias ls='ls --color=auto'
alias mv='mv -i'
alias rm='rm -i'
alias which='alias | /usr/bin/which --tty-only --read-alias --show-dot --show-tilde'
```

Некоторые из этих **alias** помогают защитить ключевые файлы от ошибок. **Ключ -i** запрашивает подтверждение у пользователя перед удалением или перезаписью файла командой **cp**, **mv** или **rm**. Просто знайте, **ключ -f заменяет -i** для отмеченных команд.

Wildcards Подстановочный знак

Иногда вы можете не знать точное имя файла или точное условие поиска. Именно тогда подстановочный знак удобен, особенно с командами, описанными всюду по книге. Три основных подстановочных знака показаны в таблице 3-1.

ТАБЛИЦА 3-1 Подстановочные знаки в оболочке

Подстановочный знак	Описание
*	Любое количество символов (или вообще никаких символов). Например, команда ls ab* вернет следующие имена файлов, при условии, что они существуют в текущем каталоге: ab, abc, abcd .
?	Один единственный символ. Например, ls ab? Команда вернет следующие имена файлов, при условии, что они существуют в текущем каталоге: abc, abd, abe .
[]	Диапазон вариантов. Например, команда ls ab[123] вернет следующие имена файлов, при условии, что они существуют в текущем каталоге: ab1, ab2, ab3 . В качестве альтернативы команда ls ab[X-Z] возвращает следующие имена файлов, при условии, что они существуют в текущем каталоге: abX, abY, abZ

!!!! On the Job !!!

Использование подстановочных знаков в мире Linux иногда называют глобальным (globbing.)

!!!!

Поиск файлов

Большинство пользователей, которые некоторое время изучают Linux, знакомятся с ключевыми файлами. Например, **named.conf** - это файл конфигурации ключей для стандартных DNS-серверов (Служба доменных имен), основанный на домене имен Интернета в Беркли (BIND). Но не многие люди помнят, что образец файла **named.conf** со всеми видами полезных советов по настройке можно найти в каталоге **/usr/share/doc/bind-*/sample/etc**.

Для этого есть две основные команды для поиска файлов: **find** и **locate**.

find

Команда **find** выполняет поиск нужных файлов в каталогах и подкаталогах. Например, если вы хотите найти каталог с примером файла конфигурации **DNS named.conf**, вы можете использовать следующую команду, которая запустит поиск в корневом каталоге:

```
# find / -name named.conf
```

Но скорость этого поиска зависит от памяти и скорости диска, доступных в локальной системе. В качестве альтернативы, если вы знаете, что этот файл находится в дереве подкаталога **/usr**, вы можете запустить этот каталог с помощью следующей команды:

```
# find /usr -name named.conf
```

Эта команда теперь должна быстрее найти нужный файл.

locate

Если это занимает слишком много времени, RHEL позволяет настроить базу данных установленных файлов и каталогов. Поиск с помощью команды **locate** происходит практически мгновенно, и поиск по местоположению не требует полного имени файла. Недостатком является то, что база данных команды **locate** обычно обновляется только один раз каждый день, как указано в скрипте **/etc/cron.daily/mlocate**.

Поскольку ежедневные задания выполняются только раз в 24 часа, этого недостаточно, особенно во время 2,5-часового экзамена. К счастью, указанный сценарий может быть

выполнен непосредственно из интерфейса командной строки пользователем с правами администратора. Просто введите полный путь к файлу, как если бы это была команда:

```
# /etc/cron.daily/mlocate
```

!!!! Exam Watch !!!!!

Когда вы сдаете экзамены в Red Hat, вы можете вначале запустить **updatedb** или отмеченный скрипт **mlocate**, чтобы быстрее найти нужные файлы в дальнейшем. !!!!!

ЦЕЛЬ СЕРТИФИКАЦИИ 3.03

Управление текстовыми файлами

Linux и Unix обычно управляются через серию текстовых файлов. Администраторы Linux обычно не используют графические редакторы для управления этими файлами конфигурации. Такие редакторы, как OpenOffice.org Writer или Microsoft Word, обычно либо сохраняют файлы в двоичном формате, либо изменяют кодировку текстовых файлов. Если текстовые файлы не сохранены в их первоначальном формате, внесенные изменения могут привести к невозможности загрузки системы Linux.

Команды Linux были настроены для управления текстовыми файлами как потоками данных. Вы видели такие инструменты, как стрелки перенаправления и пайпс. Тем не менее, эти данные могут быть огромными без инструментов, которые могут сортировать эти данные. Еще до того, как файлы отредактированы, важно знать, как читать эти файлы в интерфейсе командной строки.

Команды для чтения текстовых потоков

Ранее вы просматривали такие команды, как **cd**, **ls** и **pwd**, которые могут помочь вам обойти файлы Linux. С такими командами, как **find** и **locate**, вы узнали, как определить местоположение нужных файлов.

Теперь пришло время начать чтение, копирование и перемещение файлов. Большинство файлов конфигурации Linux являются текстовыми файлами. Linux редакторы - это текстовые редакторы. Команды Linux предназначены для чтения текстовых файлов. Чтобы определить типы файлов в текущем каталоге, попробуйте команду **file ***.

cat

Самая основная команда для чтения файлов - это **cat**. Команда **cat filename** выполняет прокрутку текста в файла **filename**. Он также работает с несколькими именами файлов; он объединяет имена файлов, которые вы можете перечислить, как один непрерывный вывод на ваш экран. Вы можете перенаправить вывод на имя файла по вашему выбору, как описано в разделе «Текстовые потоки и перенаправление команд».

less и more

Большие файлы требуют командных утилит, которые могут помочь вам прокрутить текст файла на досуге. Эти утилиты известны как пейджеры, и наиболее распространенными являются все **more** и **less**. С помощью команды **more filename** вы можете пролистывать текст файла от начала до конца по одному экрану за раз. С помощью команды **less filename** вы можете прокручивать в обоих направлениях один и тот же текст с помощью кнопок «вверх», «вниз» и «стрелка». Обе команды поддерживают поиск в стиле **vi**. Поскольку команды **less** и **more** не изменяют файлы, они являются отличным способом прокрутки и поиска элементов в большом текстовом файле, например в журнале ошибок.

Например, чтобы выполнить поиск в основном файле `/var/log/messages`, выполните следующую команду:

```
# less /var/log/messages
```

После этого вы сможете прокручивать файл журнала вверх и вниз для получения важной информации. Вы можете использовать косую черту и вопросительный знак для поиска вперед или назад по файлу. Например, после запуска только что показанной команды вы попадете на экран, аналогичный показанному на **рисунке 3-3**.

РИСУНОК 3-3 команда `less` и `/var/log/messages`

```
Dec 28 09:36:31 server1 NetworkManager: DHCPREQUEST on eth0 to 255.255.255.255 p
ort 67 (xid=0x14b16e00)
Dec 28 09:36:31 server1 NetworkManager[829]: <info> (eth0): DHCPv4 state changed
nbi -> preinit
Dec 28 09:36:31 server1 dhclient[1707]: DHCPACK from 192.168.122.1 (xid=0x14b16e
00)
Dec 28 09:36:31 server1 NetworkManager: DHCPACK from 192.168.122.1 (xid=0x14b16e
00)
Dec 28 09:36:31 server1 dhclient[1707]: bound to 192.168.122.225 -- renewal in 1
441 seconds.
Dec 28 09:36:31 server1 NetworkManager: bound to 192.168.122.225 -- renewal in 1
441 seconds.
Dec 28 09:36:31 server1 NetworkManager[829]: <info> (eth0): DHCPv4 state changed
preinit -> reboot
Dec 28 09:36:31 server1 NetworkManager[829]: <info> address 192.168.122.225
Dec 28 09:36:31 server1 NetworkManager[829]: <info> plen 24 (255.255.255.0)
Dec 28 09:36:31 server1 NetworkManager[829]: <info> gateway 192.168.122.1
Dec 28 09:36:31 server1 NetworkManager[829]: <info> server identifier 192.168.
122.1
Dec 28 09:36:31 server1 NetworkManager[829]: <info> lease time 3600
Dec 28 09:36:31 server1 NetworkManager[829]: <info> hostname 'server1'
Dec 28 09:36:31 server1 NetworkManager[829]: <info> nameserver '192.168.122.1'
Dec 28 09:36:31 server1 NetworkManager[829]: <info> Activation (eth0) Stage 5 of
:
```

Для поиска в файле термина «IPv4 tunneling» введите следующее в команде `less`:

```
/ IPv4 tunneling
```

Чтобы искать в обратном направлении, подставьте `? для /`.

P.S.

Нажатие символа `n` продолжает поиск по указанному шаблону далее, символ `N` продолжит поиск к началу вывода.

Команда `less` имеет еще одну функцию, недоступную для таких команд, как `more` и `cat`: она может читать текстовые файлы, сжатые в формате `gzip`, обычно показанном с расширением `.gz`. Например, страницы руководства, связанные со многими стандартными командами, которые запускаются в оболочке, можно найти в каталоге `/usr/share/man/man1`. Все файлы в этом каталоге сжаты в формате `.gz`. Тем не менее, команда `less` может читать эти файлы.

И это указывает на действие команды `man`. Другими словами, эти две команды функционально эквивалентны:

```
# man cat
```

```
# less /usr/share/man/man1/cat.1.gz
```

head и tail

Команды **head** и **tail** являются отдельными инструментами, которые работают практически одинаково. По умолчанию команда **head filename** просматривает первые 10 строк файла; команда **tail filename** просматривает последние 10 строк файла. Вы можете указать количество отображаемых строк с помощью ключа **-nxy**. Например, команда **tail -n 15 /etc/passwd** выводит список последних 15 строк файла **/etc/passwd**.

Команда **tail** может быть особенно полезна при возникновении проблем. Например, если существует постоянная проблема с неудачными попытками входа в систему, следующая команда отслеживает указанный файл и отображает новые строки на экране по мере записи новых записей журнала:

```
# tail -f /var/log/secure
```

Команды для обработки текстовых потоков

Текстовый поток - это движение данных. Например, команда **cat filename** отправляет данные из файла **filename** в терминал. Когда эти файлы становятся большими, удобно иметь команды, которые могут фильтровать и иным образом обрабатывать эти потоки текста.

Linux включает в себя простые команды, которые помогут вам искать, проверять или сортировать содержимое файла. Кроме того, есть специальные файлы, которые содержат другие; некоторые из этих файлов-контейнеров в разговорной речи называются «**tarballs**».

!!!! On the Job !!!!

tarballs - это распространенный способ распространения пакетов Linux. Пакеты обычно распространяются в сжатом формате с расширением **.tar.gz** или **.tgz**, объединенным в пакет в одном файле.

!!!!

sort

Вы можете отсортировать содержимое файла несколькими способами. По умолчанию команда сортировки сортирует содержимое в алфавитном порядке, в зависимости от первой буквы в каждой строке. Например, команда **sort/etc/passwd** сортирует всех пользователей (включая тех, которые связаны с конкретными службами и т. д.) По имени пользователя.

grep

Команда **grep** использует поисковый запрос для просмотра файла. Возвращает полную строку, содержащую поисковый запрос. Например, **grep 'Michael Jang' /etc/passwd** ищет имя этого автора в файле **/etc/passwd**.

Вы можете использовать регулярные выражения в команде **grep**. Регулярные выражения предоставляют мощный способ задания сложных шаблонов поиска. Некоторые символы, которые имеют особое значение внутри регулярного выражения, показаны в **таблице 3-2**. Если вы хотите, чтобы метасимвол терял свое особое значение и воспринимался буквально, перед ним ставьте черную черту (****).

Команда **grep** поддерживает несколько полезных ключей. Чтобы сделать поиск нечувствительным к регистру, вы можете передать **опцию -i** в командную строку. **Опция -E** позволяет использовать расширенный синтаксис регулярного выражения. Другой интересный **ключ -v**, который переворачивает логику сопоставления, то есть он говорит **grep** выбирать только строки, которые не соответствуют регулярному выражению.

В качестве примера, предположим, что вы хотите выбрать только те строки из **/etc/nsswitch.conf**, которые не являются пустыми и не содержат комментариев (то есть они не начинаются с символа **#**). Это может быть достигнуто с помощью следующей команды:

```
# grep -v '^$' /etc/nsswitch.conf | grep -v '^#'
```

ТАБЛИЦА 3-2 Специальные символы в регулярных выражениях

Метасимвол	Описание
.	Любой отдельный символ. Часто используется с множителем * для обозначения любого количества символов.
[]	Соответствует любому отдельному символу, заключенному в квадратные скобки. Например, команда grep 'jo[ah]n' /etc/passwd вернет все строки в /etc/passwd , содержащие строку joan или john .
?	Соответствует предыдущему элементу ноль или один раз. Например, команда grep -E 'ann?a' /etc/passwd вернет все строки в /etc/passwd , которые содержат строку ana или anna .
+	Сопоставьте предыдущий элемент один или несколько раз. Например, команда grep -E 'j[a-z]+n' /etc/passwd вернет все строки в /etc/passwd , содержащие буквы j и n , с одной или несколькими строчными буквами между ними. Следовательно, это регулярное выражение будет соответствовать строкам, таким как joan , john , а также jason и jonathan .
*	Соответствует предыдущему элементу ноль или более раз. Например, команда grep 'jo[a-z]*n' /etc/passwd вернет все строки в /etc/passwd , которые содержат строку jo , за которой следуют ноль или более строчных букв и заканчиваются символом n . Поэтому предыдущее регулярное выражение будет соответствовать строкам, таким как jon , joan или john .
^	Соответствует началу строки. Например, команда grep 'jo[a-z]*n' /etc/passwd вернет все строки в /etc/passwd , которые начинаются с последовательности символов bin .
\$	Подходим конец строки. Например, команда grep '/bin/[kz]sh\$' /etc/passwd вернет все строки в /etc/passwd , которые заканчиваются последовательностью /bin/ksh or /bin/zsh (то есть все записи, соответствующие пользователям, которые установили Korn или Zsh в качестве оболочек по умолчанию).

Обратите внимание, как первая команда **grep** выбирает все строки, которые не являются пустыми (регулярное выражение, которое соответствует пустой строке, **равно** **^\$**, то есть начало строки и непосредственно конец строки). Затем выходные данные передаются второй команде **grep**, которая исключает все строки, начинающиеся с символа хеша.

Тот же результат можно получить с помощью одного экземпляра **grep** и ключа **-e**, который позволяет указать несколько шаблонов поиска для одной команды:

```
# grep -v -e '^$' -e '^#' /etc/nsswitch.conf
```

Для получения дополнительной информации о регулярных выражениях введите **man 7 regex**.

diff

Одним из полезных вариантов поиска различий между файлами является команда **diff**. Если вы только что использовали инструмент, такой как редактор соединений Network Manager, описанный далее в этой главе, он изменит файл, такой как **ifcfg-eth0**, в каталоге **/etc/sysconfig/network-scripts**.

Если вы создали резервную копию этого файла **ifcfg-eth0**, команда **diff** может определить различия между этими двумя файлами. Например, следующая команда определяет различия между файлом **ifcfg-eth0** в каталогах **/root** и **/etc/sysconfig/network-scripts**:

```
# diff /root/ifcfg-eth0 /etc/sysconfig/network-scripts/ifcfg-eth0
```

wc

Команда **wc**, сокращение от количества слов (**word count**), может возвращать количество строк, слов и символов в файле. Варианты **wc** просты; например, **wc -w filename** возвращает количество слов в этом файле.

sed

Команда **sed**, сокращение от (**stream editor**) потоковый редактор, позволяет вам искать и изменять указанные слова или даже текстовые потоки в файле. Например, следующая команда заменяет первый экземпляр слова «**Windows**» на «**Linux**» в каждой строке файла **opsys** и записывает результат в файл **newopsys**:

```
# sed 's/Windows/Linux/' opsys > newopsys
```

Однако этого может быть недостаточно. Если в файле **opsys** имеется более одного экземпляра «**Windows**», он не меняет второй экземпляр этого слова. Но вы можете исправить это, добавив суффикс для глобального, во всём файле изменения (**global(g)**):

```
# sed 's/Windows/Linux/g' opsys > newopsys
```

В следующем примере будет показано, что все общие ресурсы **Samba**, настроенные с помощью директивы **writable = yes**, поменяют значения на противоположные:

```
# sed 's/writable = yes/writable = no/g' /etc/samba/smb.conf > ~/smb.conf
```

Конечно, вам следует просмотреть результаты в файле **/root/smb.conf** перед тем, как перезаписать исходный файл **/etc/samba/smb.conf**.

awk

Команда **awk**, названная по имени ее разработчиков (Aho, Weinberger и Kernighan), больше чем команда, **awk** является скорее полноценным языком программирования. Он может идентифицировать строки с помощью ключевого слова и считывать текст из указанного столбца в этой строке. Типичным примером является файл **/etc/passwd**. Например, следующая команда будет считывать четвертое поле в **/etc/passwd** (идентификатор группы) каждого пользователя со списком «**mike**»:

```
# awk -F : '/mike/ {print $4}' /etc/passwd
```

Редактировать текстовые файлы на консоли

В первоначальной версии целей RHCSA указано использование редактора **vim**. Строго говоря, не имеет значения, какой текстовый редактор вы используете для редактирования текстовых файлов. Тем не менее, мы считаем, что вам нужно знать, как использовать редактор **vim**, и, очевидно, некоторые в Red Hat с этим согласны. Редактор **vim** - это сокращение от «vi, улучшено». После установки редактор **vim** можно запустить с помощью команды **vi**. Далее мы называем этот текстовый редактор **vi**.

Мы считаем, что каждый администратор должен иметь хотя бы базовые знания о **vi**. Хотя **emacs** также будет правильным выбором, **vi** может помочь вам сохранить испорченную систему. Если вам когда-нибудь понадобится восстановить критический файл конфигурации с помощью аварийного загрузочного носителя, **vi**, возможно, будет единственным доступным вам редактором.

Хотя RHEL 7 также включает в себя доступ к более интуитивному **nano** - редактору, знание команд **vi** может помочь вам быстрее находить и редактировать ключевые разделы текстовых файлов. Хотя аварийный носитель RHEL поддерживает больше консольных редакторов, **vi** является одним из наиболее многофункциональных и эффективных редакторов, доступных в Linux.

Вы должны знать, как использовать два основных **режима vi**: **команда и вставка (command and insert.)**. Когда вы используете **vi**, чтобы открыть файл, он открывается в командном режиме. Некоторые команды запускают режим вставки. Открыть файл легко: просто используйте команду **vi filename**. Пример **vi** с файлом **/etc/nsswitch.conf** показан на рисунке 3-4.

Следующее является только кратким введением в редактор **vi**. Для получения дополнительной информации вы можете обратиться к ряду книг по этой теме, а также к учебному пособию, которое вы можете запустить с помощью команды **vimtutor**.

РИСУНОК 3-4 Редактор **vi** с **/etc/nsswitch.conf**

```
passwd:      files sss
shadow:      files sss
group:       files sss
#initgroups: files

#hosts:      db files nisplus nis dns
hosts:       files dns myhostname

# Example - obey only what nisplus tells us...
#services:   nisplus [NOTFOUND=return] files
#networks:   nisplus [NOTFOUND=return] files
#protocols:  nisplus [NOTFOUND=return] files
#rpc:        nisplus [NOTFOUND=return] files
#ethers:     nisplus [NOTFOUND=return] files
#netmasks:   nisplus [NOTFOUND=return] files

bootparams:  nisplus [NOTFOUND=return] files

ethers:      files
netmasks:    files
networks:    files
protocols:   files
rpc:         files
```

vi Командный режим

В командном режиме вы можете делать все с текстовым файлом, кроме как редактировать его. Опции в командном режиме являются обширными и разнообразными, они являются предметом ряда текстов длиной в книгу. Параметры в командном режиме **vi** делятся на семь категорий:

- **Открыть (Open)** Чтобы открыть файл в редакторе **vi** из интерфейса командной строки, выполните команду **vi filename**.
- **Поиск (Search)** Для прямого поиска начните с обратной **косой черты (/)**, за которой следует поисковый запрос. Помните, Linux чувствителен к регистру, так что если вы ищете «**Michael**» в **/etc/passwd**, используйте команду **/Michael** (не **/michael**). Для обратного поиска начните с вопросительного знака (?).
- **Запись (Write)** Чтобы сохранить изменения, используйте команду **w**. Вы можете комбинировать команды; например **:wq** записывает файл и выходит из **vi**.
- **Закрыть (Close)** Чтобы выйти из **vi**, используйте команду **:q**.
- **Отказаться от (Abandon)** Если вы хотите отказаться от каких-либо изменений в редактируемом файле, используйте команду **:q!**.
- **Редактировать (Edit)** Вы можете использовать ряд команд для редактирования файлов с помощью **vi**, например, **x**, который удаляет выделенный в данный момент символ, **dw**,

который удаляет выделенное в настоящий момент слово, и **dd**, который удаляет текущую строку. Помните, **yy** копирует текущую строку в буфер, **p** помещает текст из буфера, а **u (undo)** восстанавливает текст из предыдущего изменения.

- **Вставка (Insert)** Ряд команд позволяет запустить режим вставки, **включая i**, чтобы начать вставку текста в текущей позиции редактора, и **o**, чтобы открыть новую строку непосредственно под текущей позицией курсора.

Основное в редактирование текста

В современных системах Linux редактировать файлы с помощью **vi** легко. Просто используйте обычные навигационные клавиши (клавиши со стрелками, страница вверх и страница вниз), а затем одну из основных команд, таких, как **i** или **o**, чтобы запустить режим вставки **vi**, и введите свои изменения непосредственно в файл.

Когда вы закончите с режимом вставки, нажмите клавишу **esc**, чтобы вернуться в командный режим. Затем вы можете сохранить свои изменения или отказаться от них и выйти из **vi**.

!!!! On the Job !!!!

Существует несколько специализированных вариантов команды **vi**. Три - **vipw**, **vigw** и **visudo**, которые редактируют **/etc/passwd**, **/etc/group** и **/etc/sudoers** соответственно.

Команды **vipw -s** и **vigr -s** редактируют файлы **/etc/shadow** и **/etc/gshadow** соответственно.
!!!!

УПРАЖНЕНИЕ 3-1

Использование **vi** для создания нового пользователя

В этом упражнении вы создадите нового пользователя, отредактировав файл **/etc/passwd** с помощью текстового редактора **vi**. Хотя существуют и другие способы создания новых пользователей Linux, это упражнение поможет вам проверить свои навыки с помощью **vi** и интерфейса командной строки.

1. Откройте интерфейс командной строки Linux. Войдите в систему как пользователь **root** и введите команду **vipw**. Эта команда использует редактор **vi** для открытия **/etc/passwd**.
2. Перейдите в конец файла. Есть несколько способов сделать это в командном режиме, включая клавишу со стрелкой вниз, клавишу вниз страницы и команду **G**.
3. Определите линию, связанную с обычным пользователем. Если вы только что создали нового пользователя, это должна быть последняя строка в файле с **UID-номера** от **1000 и выше**. Если обычного пользователя еще не существует, укажите первую строку, которая должна быть связана с пользователем с правами администратора, с номером 0 в третьем и четвертом столбце.
4. Сделайте одну копию этой строки. Если вы уже знакомы с **vi**, вы должны знать, что вы можете скопировать всю строку в буфер с помощью команды **yy**. Строка помещается в буфер. Затем вы можете восстановить или вставить в эту строку столько раз, сколько пожелаете, с помощью команды **p**.
5. Измените имя пользователя, идентификатор пользователя, идентификатор группы, комментарий пользователя и домашний каталог для нового пользователя. Подробную информацию о каждой записи см. В главе 8. Например, на следующем рисунке это соответствует **weedle**, **1001**, **1001**, **Tweedle Dee**, и **/home/tweedle**. Убедитесь, что имя пользователя также соответствует домашнему каталогу.

```

rpc:x:32:32:Rpcbind Daemon:/var/lib/rpcbind:/sbin/nologin
rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin
nfsnobody:x:65534:65534:Anonymous NFS User:/var/lib/nfs:/sbin/nologin
named:x:25:25:Named:/var/named:/sbin/nologin
oprofile:x:16:16:Special user account to be used by OProfile:/var/lib/oprofile:/
sbin/nologin
tcpdump:x:72:72:::/sbin/nologin
usbmuxd:x:113:113:usbmuxd user:/sbin/nologin
colord:x:998:996:User for colord:/var/lib/colord:/sbin/nologin
abrt:x:173:173::/etc/abrt:/sbin/nologin
chrony:x:997:995::/var/lib/chrony:/sbin/nologin
libstoragemgmt:x:996:994:daemon account for libstoragemgmt:/var/run/lsm:/sbin/no
login
qemu:x:107:107:qemu user:/sbin/nologin
radvd:x:75:75:radvd user:/sbin/nologin
rtkit:x:172:172:RealtimeKit:/proc:/sbin/nologin
saslauthd:x:995:76:"Saslauthd user":/run/saslauthd:/sbin/nologin
ntp:x:38:38::/etc/ntp:/sbin/nologin
pulse:x:171:171:PulseAudio System Daemon:/var/run/pulse:/sbin/nologin
gdm:x:42:42::/var/lib/gdm:/sbin/nologin
gnome-initial-setup:x:993:991::/run/gnome-initial-setup:/sbin/nologin
michael:x:1000:1000:Michael Jang:/home/michael:/bin/bash
tweedle:x:1001:1001:Tweedle Dee:/home/tweedle:/bin/bash

```

6. Вернитесь в командный режим, **нажав** клавишу **esc**. Сохраните файл с помощью команды **:w**, а затем выйдите с помощью команды **:q**. (Вы можете объединить две команды в **vi**; в следующий раз, когда вы внесете изменение и хотите сохранить и выйти, запустите команду **:wq**.)
7. Вы должны увидеть следующее сообщение:

**You have modified /etc/passwd.
 You may need to modify /etc/shadow for consistency.
 Please use the command 'vipw -s' to do so.**

Это сообщение можно игнорировать, потому что следующий шаг добавляет соответствующие информация в файл **/etc/shadow**. Тем не менее, вам не нужно изменять **/etc/shadow** напрямую.

8. От имени пользователя **root** выполните команду **passwd newuser**. Назначьте новый пароль на ваш выбор. Для этого примера новый пользователь - **tweedle**.
9. Процесс еще не завершен; каждому пользователю нужна группа. Для этого запустите команду **vigr**. Повторите предыдущие шаги, которые скопировали соответствующую строку в конце файла. Обратите внимание, что имена групп и идентификаторы групп обычно идентичны их именам пользователей и идентификаторам пользователей.
10. Все, что вам нужно изменить для новой записи, это имя группы и идентификационный номер группы. Основываясь на информации, показанной на предыдущем рисунке, это будет имя группы из твида и номер группы 1001.
11. Повторите вышеупомянутую команду **:wq**, чтобы закрыть **vi** и сохранить изменения.
12. Обратите внимание на следующее сообщение:

**You have modified /etc/group.
 You may need to modify /etc/gshadow for consistency.
 Please use the command 'vigr -s' to do so.**

13. Как было предложено, выполните команду **vigr -s**, чтобы открыть файл **/etc/gshadow**. Вы заметите, что в этом файле меньше информации. Как только копия сделана из соответствующей строки, все, что вам нужно сделать, это изменить имя группы.
14. Повторите вышеупомянутую команду **:wq**, чтобы закрыть **vi** и сохранить изменения. На самом деле вы получите сообщение о том, что файл доступен только для чтения. Вы должны выполнить **:wq!** в этом случае для записи в этот файл «только для чтения», переопределит текущие настройки.

15. Дополнительные действия необходимы для правильной настройки нового пользователя, связанного с домашним каталогом этого пользователя и стандартными файлами из каталога `/etc/skel`. Для получения дополнительной информации см. **Главу 8**.

Если вам не нравится `vi`

По умолчанию, когда вы запускаете такие команды, как `edquota` и `crontab`, в редакторе `vi` открываются связанные файлы настроек квот и заданий `crontab`. Если вам абсолютно не нравится редактор `vi`, редактор по умолчанию можно изменить с помощью следующей команды:

```
# export EDITOR=/bin/nano
```

Чтобы изменить редактор по умолчанию для всех пользователей, добавьте предыдущую строку в файл конфигурации `/etc/environment`. Вам не нужно использовать редактор `vi` для изменения `/etc/environment`; вместо этого следующая команда добавляет указанную команду в конец файла `/etc/environment`:

```
# echo 'export EDITOR=/bin/nano' >> /etc/environment
```

Поскольку **nano-редактор** довольно интуитивно понятен, как показано на **рисунке 3-5**, инструкции не будут представленным в этой книге. Полное руководство доступно по адресу www.nano-editor.org/dist/v2.3/nano.html.

Подобные изменения могут быть сделаны, если вы предпочитаете другой редактор, например, `emacs`.

РИСУНОК 3-5 nano-редактор с `/etc/nsswitch.conf`



```
GNU nano 2.3.1 File: /etc/nsswitch.conf

#
# /etc/nsswitch.conf
#
# An example Name Service Switch config file. This file should be
# sorted with the most-used services at the beginning.
#
# The entry '[NOTFOUND=return]' means that the search for an
# entry should stop if the search in the previous entry turned
# up nothing. Note that if the search failed due to some other reason
# (like no NIS server responding) then the search continues with the
# next entry.
#
# Valid entries include:
#
#      nisplus          Use NIS+ (NIS version 3)
#      nis              Use NIS (NIS version 2), also called YP
#      dns              Use DNS (Domain Name Service)
#      files            Use the local files
#      db               Use the local database (.db) files

[ Read 64 lines ]

^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

Редактировать текстовые файлы в графическом интерфейсе

Без сомнения, экзамены Red Hat стали более дружелюбными для пользователей графического интерфейса. Текстовый редактор `gedit` даже на короткое время был включен в задачи RHCSA. Более традиционные администраторы Linux, возможно, были в ужасе. (Редактор `gedit` с тех пор был удален из целей для сдачи экзаменов.)

Если редактор `gedit` не установлен в вашей системе, выполните команду `yum install gedit`. После установки `gedit` вы можете запустить его, нажав **Приложения | Стандартные | Текстовый редактор (Applications | Accessories | Text Editor (gedit))**. Поскольку это

текстовый графический редактор с интуитивно понятным интерфейсом, его использование тривиально. Не заикивайтесь на редакторах; они просто инструменты на экзаменах и в реальной жизни.

Однако, если вы редактируете файлы конфигурации на удаленных системах, возможно, что у вас не будет доступа к **gedit** в этой системе, особенно если там не установлен графический интерфейс. Конечно, вы можете установить графический интерфейс и использовать перенаправление X в любой системе Red Hat. Но многие администраторы настраивают виртуальные машины без графического интерфейса для экономии места и снижения рисков безопасности.

ЦЕЛЬ СЕРТИФИКАЦИИ 3.04

Локальная онлайн документация

Хотя во время экзаменов Red Hat доступ в Интернет запрещен, в системе RHEL 7 уже имеется много документации. Она начинается со страниц **man** справочника, которые документируют параметры и настройки, связанные с большинством команд и многими файлами конфигурации. Больше помощи можно получить в информационных документах которые устанавливаются вместе с пакетами программ. Хотя не все команды и файлы имеют такие документы, когда они доступны, они предоставляют еще больше информации.

Многие пакеты содержат обширную документацию в каталоге **/usr/share/doc**. Просто примените команду **ls** к этому каталогу. Каждый подкаталог содержит информацию о возможностях каждого связанного пакета.

Когда тебе нужна помощь

Первое, что мы обычно делаем, когда нам нужна помощь с командой, запускаем ее отдельно. Если требуется дополнительные опции или ключи, команда запрашивает дополнительную информацию. В качестве примера рассмотрим вывод следующей команды:

```
$ yum
```

Если такой подход не работает, как правило, некоторая помощь доступна с ключами **-h** или **--help**. Иногда ошибка при использовании дополнительных параметров приводит к некоторым подсказкам; вывод следующей команды предлагает допустимые опции для команды **cd**:

```
$ cd -h
```

```
bash: cd: -h: invalid option
```

```
cd: usage: cd [-L|[-P [-e]]] [dir]
```

РИСУНОК 3-6 вывод краткой справки команды **ls**


```
[alex@server1 ~]$ ls --help
Usage: ls [OPTION]... [FILE]...
List information about the FILES (the current directory by default).
Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.

Mandatory arguments to long options are mandatory for short options too.
-a, --all                do not ignore entries starting with .
-A, --almost-all        do not list implied . and ..
    --author              with -l, print the author of each file
-b, --escape              print C-style escapes for nongraphic characters
    --block-size=SIZE    scale sizes by SIZE before printing them; e.g.,
                        '--block-size=M' prints sizes in units of
                        1,048,576 bytes; see SIZE format below
-B, --ignore-backups     do not list implied entries ending with ~
-c                        with -lt: sort by, and show, ctime (time of last
                        modification of file status information);
                        with -l: show ctime and sort by name;
                        otherwise: sort by ctime, newest first
-C                        list entries by columns
    --color[=WHEN]       colorize the output; WHEN can be 'never', 'auto',
                        or 'always' (the default); more info below
-d, --directory          list directories themselves, not their contents
-D, --dired               generate output designed for Emacs' dired mode
-f                        do not sort, enable -aU, disable -ls --color
```

Иногда **ключ -h** более полезен; взгляните на вывод команды **fdisk -h**. Но **опция -h** не всегда показывает сообщение справки; в этом случае вам поможет **ключ --help**. Посмотрите на **рис. 3-6** в качестве примера, который отображает вывод команды **ls --help**.

Разнообразие man страниц

Мало кто может запомнить множество ключей (опций) для каждой команды. Это одна из причин, почему документация по командам так важна. Большинство команд Linux документированы в формате, известном, как **man-страница**. Если вы запустите команду **man** самостоятельно без аргументов, RHEL вернет следующее сообщение:

Какая справочная страница вам нужна? (What manual page do you want?)

Например, скажем, вам нужно настроить физический том, но вы забыли ключи (опции), связанные с командой **lvextend**. Чтобы просмотреть **man-страницу** для этой команды, запустите **man lvextend**. Как и во многих других описаний команд, есть секция **EXAMPLES**, подобный показанному на **рисунке 3-7**. Если вы ранее выполняли команду **lvextend**, этот раздел может помочь вам восстановить информацию по команде в вашей памяти.

Такие справочные страницы доступны для большинства файлов конфигурации и команд. Однако может случиться и другое, вы не уверены в названии справочной страницы? В этом случае могут помочь команды **whatis** и **apropos**. Например, чтобы найти справочные страницы с «**nfs**» в заголовке, выполните следующую команду:

```
# whatis nfs
```

РИСУНОК 3-7 Примеры со страницы руководства **man lvextend**

```
--use-policies
    Resizes the logical volume according to configured policy. See
    lvm.conf(5) for some details.
```

Examples

Extends the size of the logical volume "vg01/lvol10" by 54MiB on physical volume /dev/sdk3. This is only possible if /dev/sdk3 is a member of volume group vg01 and there are enough free physical extents in it:

```
lvextend -L +54 /dev/vg01/lvol10 /dev/sdk3
```

Extends the size of logical volume "vg01/lvol01" by the amount of free space on physical volume /dev/sdk3. This is equivalent to specifying "-l +100%PVS" on the command line:

```
lvextend /dev/vg01/lvol01 /dev/sdk3
```

Extends a logical volume "vg01/lvol01" by 16MiB using physical extents /dev/sda:8-9 and /dev/sdb:8-9 for allocation of extents:

```
lvextend -L+16M vg01/lvol01 /dev/sda:8-9 /dev/sdb:8-9
```

SEE ALSO

```
fsadm(8), lvm(8), lvm.conf(5), lvcreate(8), lvconvert(8), lvreduce(8),
lvresize(8), lvchange(8)
```

Если вы хотите найти **man-страницы с nfs** в описании, следующая команда может идентифицировать связанные команды:

```
# apropos nfs
```

Однако, если вы только что установили службу, такую как **httpd**, связанную с **веб-сервером Apache**, такие команды, как **whatis httpd** и **apropos apachectl**, вероятно, не предоставят никакой информации. Эти команды работают из базы данных в каталоге **/var/cache/man**. Вы можете обновить эту базу данных с помощью задания **man-db.cron** в каталоге **/etc/cron.daily**. Поскольку этот сценарий уже выполняется, следующая команда обновляет базу данных **man-страниц**:

```
# /etc/cron.daily/man-db.cron
```

Или

```
#mandb
```

Если вы столкнулись с ситуацией, например, во время экзамена Red Hat, когда соответствующая страница руководства не установлена, для этого есть как минимум три возможные причины. Соответствующий функциональный программный пакет может быть не установлен. Пакет RPM с именем **man-pages** также может быть не установлен. В некоторых случаях есть пакет, специально предназначенный для документации, который должен быть установлен отдельно. Например, есть пакет **system-config-users-doc**, который включает документацию на основе графического интерфейса пользователя для инструмента **конфигурации User Manager**. Существует также пакет **httpd-manual**, установленный отдельно от **веб-сервера Apache**.

В некоторых случаях доступно несколько справочных страниц. Посмотрите на следующий вывод команды **whatis smbpasswd**:

```
smbpasswd (5) - The Samba encrypted password file
```

```
smbpasswd (8) - change a user's SMB password
```

Числа (5) и (8) связаны с различными разделами справочных страниц. Если вас интересуют подробности, они отображаются в выводе команды **man man**. Страница **man**, отображаемая по умолчанию, является страницей **man**, связанной с командой **smbpasswd**. В этом случае, если вам нужна страница справочника для зашифрованного файла паролей **smbpasswd**, выполните следующую команду:

\$ man 5 smbpasswd

Для выхода из справочной страницы нажмите **q**.

Справочные руководства

Список доступных информационных руководств несколько ограничен. Однако освещение некоторых тем (например, **оболочки bash**) обычно более обширное, чем соответствующая справочная страница. Чтобы получить полный список информационных документов, выполните команду `ls /usr/share/info`. Когда информационное руководство недоступно, запросом по умолчанию является связанная справочная страница.

Чтобы узнать больше о **оболочке bash**, выполните команду `pinfo bash`. Пользовательский интерфейс `pinfo` похож на текстовый **веб-браузер Lynx** и является более удобной альтернативой традиционной команде `info`. Как показано на **рисунке 3-8**, информационные руководства организованы в разделы. Чтобы получить доступ к разделу, переместите курсор к записи, отмеченной звездочкой, и нажмите ввод.

Чтобы выйти с информационной страницы, **нажмите q**.

РИСУНОК 3-8 Пример справочного руководства

```
File: bash.info, Node: Top, Next: Introduction, Prev: (dir), Up: (dir)

This manual is meant as a brief introduction to features found in
Bash. The Bash manual page should be used as the definitive reference
on shell behavior.

* Menu:

* Introduction::          An introduction to the shell.
* Definitions::          Some definitions used in the rest of this
                           manual.
* Basic Shell Features::  The shell "building blocks".
* Shell Builtin Commands:: Commands that are a part of the shell.
* Shell Variables::      Variables used or set by Bash.
* Bash Features::        Features found only in Bash.
* Job Control::          What job control is and how Bash allows you
                           to use it.
* Command Line Editing:: Chapter describing the command line
                           editing features.
* Using History Interactively:: Command History Expansion
* Installing Bash::       How to build and install Bash on your system.
* Reporting Bugs::       How to report bugs in Bash.
* Major Differences From The Bourne Shell:: A terse list of the differences

Viewing line 39/44, 88%
```

Подробная документация в /usr/share/doc

Список документации, доступной в каталоге `/usr/share/doc`, впечатляет. Но качество документации зависит от работы ее разработчиков. Подкаталоги содержат имя и номер версии установленного пакета. Некоторые из этих подкаталогов содержат только один файл, обычно называемый **COPYING**, в котором указана лицензия, под которой было выпущено данное программное обеспечение. Например, большинство пакетов **system-config-*** содержат копию **GNU GPL** в файле **COPYING** в соответствующем каталоге `/usr/share/doc`.

Иногда каталог документации содержит полезные примеры. Например, подкаталог `sudo-*/` содержит примеры файлов конфигурации и директив для административного контроля, которые могут быть полезны, когда вы настраиваете администраторов с различными привилегиями.

Документация может включать в себя целые руководства в формате HTML. Для примера, взгляните на подкаталог `pam-*/`, который включает в себя полное онлайн-руководство для системы **Pluggable Authentication Modules (PAM)**, рассмотренное в **Главе 10**.

ЦЕЛЬ СЕРТИФИКАЦИИ 3.05

Учебник по сети

ТСР/ІР - это серия протоколов, организованных по уровням, известная как набор протоколов. Он был разработан для Unix и в конечном итоге принят в качестве стандарта для общения в Интернете. **ІР-адреса** помогают вам общаться через сеть. Доступно множество инструментов и конфигураций **ТСР/ІР**, которые помогут вам управлять сетью.

Как и в предыдущих разделах этой главы, приведенные здесь утверждения являются упрощенными. Поэтому, если вы находите этот раздел излишним и/или неполным, прочитайте ссылки, приведенные в главе 1. Linux создан для работы в сети, и нет практического способа сдать экзамен Red Hat, если вы не разберетесь в деталях организации сети передачи данных.

Несмотря на то, что в настоящее время сети по-прежнему сосредоточены на адресации **ІР версии 4**, некоторые организации обязались перейти к сетям **ІР версии 6 (IPv6)**. В этом разделе основное внимание уделяется **IPv4**, тогда как **IPv6** будет рассмотрен в главе 12. Однако большинство файлов конфигурации и инструментов, используемых для IPv4, также применимы к IPv6.

Сети IPv4

Каждый компьютер, который общается в сети, должен иметь свой **собственный ІР-адрес**. Некоторые адреса назначаются навсегда определенному компьютеру; они известны как статические адреса. Другие арендуются у DHCP-сервера на ограниченный срок; они известны как динамические ІР-адреса.

Адреса IPv4 являются 32-разрядными двоичными числами и обычно выражаются в десятичной записи (например, 192.168.122.50), причем каждый десятичный октет представляет 8 бит. ІР-адрес состоит из двух частей: адреса сети (или подсети) и части хоста. До публикации RFC 1517 в 1993 году Инженерной рабочей группой по Интернету (www.ietf.org) ІР-адреса были распределены по разным классам, которые определяли размер сети и хост-часть адреса.

Сегодня ІР-адреса обычно анализируются с использованием бесклассовой логики. Маска подсети, а не класс адреса, используется для определения сетевой и хостовой частей ІР-адреса. Старая **схема классовой адресации**, представленная в RFC 791, **показана в таблице 3-3**. Некоторые концепции RFC 791 до сих пор остаются на практике; Например, диапазон ІР-адресов 224.0.0.0–239.255.255.255 используется для мультикаст адресов.

Кроме того, ряд частных ІР-адресов не следует назначать никаким компьютерам, напрямую подключенным к Интернету. Наиболее распространенные диапазоны частных сетей определены в RFC 1918 и связаны с сетевыми адресами **10.0.0.0–10.255.255.255**, **172.168.16.0–172.168.31.255** и **192.168.0.0–192.168.255.255**. Кроме того, сетевые адреса от **127.0.0.0 до 127.255.255.255** используются для обратной связи на локальном хосте.

ТАБЛИЦА 3-3 Классы ІР-адресов

Класс	Диапазон ІР	Запись
A	1.1.1.0-127.255.255.255	Позволяет сети до 16 777 214 хостов
B	128.0.0.0-191.255.255.255	Позволяет сети до 65 534 хостов
C	192.0.0.0-223.255.255.255	Позволяет сети до 254 компьютеров
D	224.0.0.0-239.255.255.255	Зарезервировано для групповой рассылки
E	240.0.0.0-255.255.255.255	Зарезервировано для экспериментального использования

Сети и Маршрутизация

Как мы уже обсуждали в предыдущем разделе, IP-адрес состоит из двух частей: префикса сети и идентификатора хоста. Чтобы определить сеть и хост-часть, IP-адреса связаны с маской подсети (также известной как маска сети или префикс). Это 32-разрядное число, состоящее из последовательности двоичных единиц, за которыми следуют нули.

Маска подсети может быть представлена в той же десятичной записи, которая используется для адресов IPv4. Например, 255.255.255.0 - маска подсети, состоящая из 24 двоичных единиц и 8 нулей. Альтернативная нотация известна как нотация бесклассовой междоменной маршрутизации (CIDR) и состоит из **символа косой черты (/)**, за которым следует число, указывающее количество битов в маске сети. Например, маска подсети 255.255.255.0 может быть записана, как **/24 в нотации CIDR**.

Учитывая IP-адрес и маску подсети, все, что вам нужно сделать, чтобы определить сетевую часть IP-адреса, - это обеспечить **логическое И (AND)** между IP-адресом и маской сети. Например, учитывая **IP-адрес 192.168.122.50 с netmask /24**, первые три байта адреса (192.168.122) представляют сетевую часть, тогда как последний байт (50) является идентификатором хоста.

Три ключевых IP-адреса определяют сеть: сетевой адрес, широковещательный адрес и маску подсети. Сетевой адрес всегда является первым IP-адресом в диапазоне; широковещательный адрес всегда является последним адресом в том же диапазоне. Маска подсети, как вы уже видели, помогает вашему компьютеру определить сеть и хост-часть IP-адреса. Вы можете назначить IP-адреса между сетью и широковещательными адресами (не включая эти адреса) любому компьютеру в сети.

В качестве примера, давайте определим диапазон адресов для частной сети. Начните с адреса частной сети 192.168.122.0 и маски подсети 255.255.255.0. Исходя из этих двух адресов, широковещательный адрес составляет 192.168.122.255, а диапазон IP-адресов, которые вы можете назначить в этой конкретной сети, составляет от 192.168.122.1 до 192.168.122.254. Эта маска подсети также определяется числом связанных битов 24. Другими словами, данная сеть может быть представлена как 192.168.122.0/24.

IP-адреса назначаются сетевым интерфейсам. **Узел с несколькими сетевыми интерфейсами, который пересылает трафик через разные сети, называется маршрутизатором.** IP-хосты, разделенные маршрутизатором другими группами IP-хостов, должны находиться в разных сетях.

С сетью и сетевыми масками связана концепция шлюза. Это IP-адрес, который определяет соединение между локальной сетью и другими сетями. Хотя этот IP-адрес шлюза является частью локальной сети, этот адрес назначается маршрутизатору с IP-адресом в другой сети, такой как общедоступный Интернет. IP-адрес шлюза обычно настраивается в таблице маршрутизации для локальной системы, как это **определено командой ip route**, описанной в следующем разделе.

Инструменты и Команды

Для управления набором протоколов TCP / IP на вашем компьютере Linux доступно значительное количество инструментов. В предыдущих версиях Red Hat Enterprise Linux некоторыми из наиболее важных команд управления сетью были **ifconfig**, **arp**, **netstat** и **route**. Эти команды устарели. **Инструмент ip** поддерживает более продвинутые функции. Чтобы упростить переход к **инструменту ip**, в **таблице 3-4** приведен список устаревших команд, а также их эквивалентные команды **ip**.

!!!! On the Job !!!

По умолчанию Red Hat Enterprise Linux 7 называет сетевые интерфейсы в зависимости от их физического местоположения (enoX и emX для встроенных сетевых интерфейсов, а также enpXsY и rXpY для слотов PCI). RHEL 7 будет использовать традиционный метод перечисления eth0, eth1, ... только как запасной вариант. Следовательно, вы можете найти первый встроенный сетевой интерфейс с именем eno1, в то время как интерфейс, расположенный на шине PCI 3, слот 0, будет называться enp3s0.

!!!!

ТАБЛИЦА 3-4 Команды **ifconfig**, **arp** и **netstat** с их эквивалентными командами **ip**

Устаревшая команда	Эквивалент команд в RHEL 7	Описание
ifconfig	ip [-s] link ip addr	Показывает информацию о статусе канала и IP-адресе для всех сетевых интерфейсов.
ifconfig eth0 192.168.122.150 netmask 255.255.255.0	ip addr add 192.168.122.150/24 dev eth0	Назначает IP-адрес и маску сети интерфейсу eth0
arp	ip neigh	Показывает таблицу ARP
route netstat -r	ip route	Отображает таблицу маршрутизации
netstat -tulnpa	ss -tulnpa	Показывает все сокеты для прослушивания и не прослушивания, а также программу, к которой они принадлежат

Другими важными сетевыми командами являются **ping** и **traceroute**, которые часто используются для диагностики и устранения неполадок в сети.

Но это всего лишь инструменты. В следующем разделе вы изучите файлы Red Hat, которые определяют команды, которые вызываются для автоматической настройки сетей в процессе загрузки.

ping и **traceroute**

Команда **ping** позволяет вам проверить соединение. Его можно применять локально, внутри сети и в разных сетях в Интернете. Для целей этого раздела предположим, что ваш IP-адрес - 192.168.122.50, а адрес шлюза в локальной сети - 192.168.122.1. Если у вас возникли проблемы с подключением к хосту, попробуйте следующие команды **ping** по порядку. Первым шагом является проверка целостности **TCP/IP** на вашем компьютере:

ping 127.0.0.1

Обычно **ping** постоянно работает в Linux; Вам нужно **нажать Ctrl-C**, чтобы остановить эту команду. Если вам необходимо проверить правильность подключения к локальной сети, выполните **ping** IP-адреса локальной сетевой карты:

ping 192.168.122.50

Если это работает, пропингуйте адрес другого компьютера в вашей сети. Затем начните отслеживать маршрут в Интернет. пропингуйте адрес сетевого шлюза (в этом случае 192.168.122.1). Если возможно, пропингуйте адрес подключения сети к Интернету, который будет на другой стороне шлюза. Это может быть публичный IP-адрес вашего роутера в интернете. Наконец, пингуйте адрес компьютера, который, как вы знаете, активен в Интернете.

Вы можете заменить имена хостов, такие как **www.google.com**, на **IP-адрес**. Если имя хоста не работает, вероятно, существует проблема с базой данных имен хостов и IP-адресов, более известной как **служба доменных имен (DNS)**. Это также может указывать на проблему с файлом конфигурации **/etc/hosts**.

Команда **traceroute** автоматизирует только что описанный процесс, отслеживая путь к месту назначения. Например, следующая команда находит **путь к IP-адресу 192.168.20.5**:

```
traceroute -n 192.168.20.5
traceroute to 192.168.20.5 (192.168.20.5), 30 hops max, 60 byte packets
1 192.168.122.1 0.204 ms 0.152 ms 0.148 ms
2 192.168.1.1 1.826 ms 2.413 ms 4.050 ms
3 192.168.20.5 2.292 ms 2.630 ms 2.554 ms
```

Посмотрите на параметр **-n** в этой команде. Это указывает **traceroute** отображать **IP-адреса**, а не имена хостов. Команда также показывает время прохождения в оба конца (RTT) для достижения каждого прыжка по пути. По умолчанию для каждого перехода отправляются три разных зонда.

Обратите внимание, что для некоторых параметров команды **traceroute** требуются права суперпользователя. Другой командой, которая служит той же цели и не подвергается этому ограничению, является **tracepath**.

!!!! On the Job !!!

По умолчанию **traceroute** использует пакеты зондов UDP с возрастающим значением времени жизни (TTL) в заголовке IP, чтобы найти путь к данному месту назначения. Иногда межсетевой экран вдоль пути может блокировать пакеты UDP. В этом случае вы можете попытаться запустить **traceroute** с параметром **-I** или **-T**, чтобы включить тестовые пакеты ICMP или TCP соответственно.

!!!!

Просмотр текущих сетевых адаптеров с помощью команды ip

Команда **ip** может отображать текущее состояние активных сетевых адаптеров. Он также может быть использован для назначения сетевых адресов и многое другое. Запустите команду **ip link show**, чтобы просмотреть состояние связи активных сетевых адаптеров в системе. При желании включите ключ **-s**, если вы хотите отображать статистику о производительности сети.

Чтобы просмотреть информацию об IP-адресе, попробуйте команду **ip address show**, которая выдает тот же вывод **ip link show**, но также включает IP-адреса и их свойства. Перечисленная ниже команда **ip address show eth0** отражает текущую конфигурацию первого сетевого адаптера Ethernet:

```
# ip addr show eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state □
UP qlen 1000
    link/ether 52:54:00:40:1e:6a brd ff:ff:ff:ff:ff:ff
    inet 192.168.122.50/24 brd 192.168.122.255 scope global eth0
        valid_lft forever preferred_lft forever
    inet6 fe80::2e0:4cff:fee3:d106/64 scope link
        valid_lft forever preferred_lft forever
```

Команда **ip** является гибкой. Например, команда **ip a s** функционально эквивалентна **ip addr show** или **ip address show**.

Настройте сетевой адаптер с помощью ip

Вы также можете использовать команду **ip** для назначения IP-адреса. Например, следующая команда добавляет указанный IP-адрес и маску сети к сетевому адаптеру **eth0**:

```
# ip addr add 192.168.122.150/24 dev eth0
```

Первый аргумент, **192.168.122.150/24**, указывает новый IP-адрес и маску сети. Следующий аргумент, **dev eth0**, сообщает вам, какое устройство настраивается. Чтобы убедиться, что изменение сработало, снова запустите команду **ip addr show eth0**.

При правильных настройках команда **ip** может изменять дополнительные параметры сети. Некоторые из этих вариантов показаны в таблице 3-5.

ТАБЛИЦА 3-5 Параметры команды **ip**

Команда	Описание
ip link set dev device up	Активирует указанный интерфейс.
ip link set dev device down	Деактивирует указанный интерфейс.
ip addr flush dev device	Удаляет все IP-адреса из указанного интерфейса.
ip link set dev device txqlen N	Изменяет длину очереди передачи для указанного интерфейса.
ip link set dev device mtu N	Устанавливает максимальную единицу передачи как N, в байтах.
ip link set dev device promisc on	Активирует смешанный режим. Это позволяет сетевому адаптеру читать все полученные пакеты, а не только пакеты, адресованные хосту. Может использоваться для анализа сети на наличие проблем или для расшифровки сообщений между другими узлами.
ip link set dev device promisc off	Деактивирует смешанный режим.

Конечно, вы должны убедиться, что изменения сохраняются после перезагрузки, будь то для экзамена или для сервера, которым вы хотите управлять удаленно. Это зависит от соответствующих изменений в файлах конфигурации в каталоге **/etc/sysconfig/network-scripts**, который будет описан ниже. Кроме того, **любые изменения, сделанные с помощью команды ip, по определению являются временными.**

Активировать и деактивировать сетевые адаптеры

Можно использовать команду **ip** для активации и деактивации сетевых адаптеров. Например, следующие команды отключают и повторно активируют первый адаптер Ethernet:

```
# ip link set dev eth0 down
# ip link set dev eth0 up
```

Однако пара более интуитивно понятных скриптов предназначена для управления сетевыми адаптерами: **ifup** и **ifdown**. В отличие от команды **ip**, они вызывают соответствующие файлы конфигурации и сценарии в каталоге **/etc/sysconfig/network-scripts**.

Например, команда **ifup eth0** активирует сетевой адаптер Ethernet с именем **eth0** на основе файла конфигурации **ifcfg-eth0** и сценария **ifup-eth** в каталоге **/etc/sysconfig/network-scripts**.

IP как инструмент диагностики

Протокол разрешения адресов (ARP) связывает аппаратный адрес сетевого интерфейса (MAC) с IP-адресом. Команда **ip neigh** отображает таблицу оборудования и IP-адресов на локальном компьютере. Эта команда может помочь обнаружить проблемы, такие как повторяющиеся адреса в сети. Такие проблемы могут возникать с неправильно настроенными системами или клонированными виртуальными машинами. При необходимости команда **ip neigh** может установить или изменить записи таблицы ARP вручную. Поскольку аппаратные

адреса не маршрутизируются, таблица ARP ограничена локальной сетью. Вот пример выходных данных команды, показывающий все записи ARP в локальной базе данных:

```
# ip neigh show
192.168.122.150 dev eth0 lladdr 52:a5:cb:54:52:a2 REACHABLE
192.168.100.100 dev eth0 lladdr 00:a0:c5:e2:49:02 STALE
192.168.122.1 dev eth0 lladdr 00:0e:2e:6d:9e:67 REACHABLE
```

В первом столбце выходных данных перечислены известные IP-адреса в локальной сети, затем интерфейс, к которому подключен сосед, и его адрес канального уровня (MAC-адрес). Последняя запись показывает, доступен ли аппаратный адрес соседа. Запись STALE может указывать на то, что истекло время ожидания кэша ARP с момента последнего просмотра пакета с этого хоста. Если таблица ARP пуста, недавних подключений к другим системам в локальной сети не существует.

Таблицы маршрутизации с `ip route`

Команда `ip` универсальна. Одна важная версия этой команды, `ip route`, отображает таблицы маршрутизации. Это функционально эквивалентно устаревшей команде `route`. При запуске с ключом `-r` (маршрут `ip -r`) эта команда ищет в файлах `/etc/hosts` и DNS-серверах имена хостов, а не числовые IP-адреса.

Таблица маршрутизации для локальной системы обычно включает ссылку на адрес шлюза по умолчанию. Например, посмотрите на следующий вывод команды `ip route`:

```
default via 192.168.122.1 dev eth0 proto static metric 1024
192.168.122.0/24 dev eth0 proto kernel scope link src 192.168.122.50
```

Устаревшая команда `netstat -nr` должна отображать ту же таблицу. Для этой таблицы маршрутизации IP-адрес шлюза - 192.168.122.1. Любые сетевые пакеты с пунктом назначения, отличным от сети 192.168.122.0, отправляются через адрес шлюза (другими словами, адрес уровня 2 для этого шлюза ищется и помещается в кадр как MAC-адрес назначения). Система с адресом шлюза, обычно маршрутизатором, пересылает этот пакет следующему маршрутизатору в соответствии со своей таблицей маршрутизации, пока не доберется до маршрутизатора, который напрямую подключен к месту назначения.

Динамически настраивать IP-адреса с помощью `dhclient`

Хотя имя команды время от времени менялось, функциональность осталась прежней. Команда `dhclient`, используемая с именем устройства сетевой карты, например `eth0`, вызывает сервер протокола динамической конфигурации хоста (DHCP) для получения IP-адреса и других параметров:

```
# dhclient eth0
```

Как правило, параметры сети, настроенные через сервер DHCP, включают IP-адрес, маску сети, адрес шлюза для доступа к внешним сетям и IP-адрес любых DNS-серверов для этой сети.

Другими словами, команда `dhclient eth0` не только назначает информацию об IP-адресе способом, описанным ранее с помощью команды `ip`, но также устанавливает маршрут по умолчанию для таблицы маршрутизации, показанной с помощью команды `ip route`. Кроме того, он добавляет IP-адрес DNS-сервера в файл конфигурации `/etc/resolv.conf`.

Отображение сетевых подключений с помощью `ss`

Команда **ss** заменяет устаревший инструмент **netstat** для отображения сетевых подключений. При правильной комбинации командных ключей он может отображать прослушивающие и не прослушивающие сокеты TCP и UDP. Одна команда, которую мы хотели бы использовать,

ss -tuna4

РИСУНОК 3-9 Вывод команды ss -tuna4

```
[root@server1 ~]# ss -tuna4
Netid  State      Recv-Q Send-Q      Local Address:Port      Peer Address:Port
tcp    UNCONN     0       0              *:68                    *:*
tcp    UNCONN     0       0              *:111                   *:*
tcp    UNCONN     0       0              *:123                   *:*
tcp    UNCONN     0       0      127.0.0.1:323          *:*
tcp    UNCONN     0       0              *:609                   *:*
tcp    UNCONN     0       0              *:43630                 *:*
tcp    UNCONN     0       0      127.0.0.1:659          *:*
tcp    UNCONN     0       0              *:45931                 *:*
tcp    UNCONN     0       0              *:5353                  *:*
tcp    UNCONN     0       0              *:61050                 *:*
tcp    LISTEN     0      100      127.0.0.1:25           *:*
tcp    LISTEN     0      128              *:52991                 *:*
tcp    LISTEN     0      128              *:111                   *:*
tcp    LISTEN     0      128              *:22                    *:*
tcp    LISTEN     0      128      127.0.0.1:631          *:*
tcp    ESTAB      0       0  192.168.122.50:22      192.168.122.1:43910
[root@server1 ~]#
```

где команда показывает все **(-a)** сетевые сокеты, используя IPv4 **(-4)** и оба протокола **TCP (-t)** и **UDP (-u)** в **числовом (-n) формате**. Если указан ключ **-p**, **ss** также покажет PID процесса, использующего каждый сокет. **Рисунок 3-9** иллюстрирует выходные данные на базовом сервере.

В конце выходных данных обратите внимание на одноранговый адрес 192.168.122.1:43910. Номер порта 43910 - это просто порт источника на удаленном сервере. Соответствующий локальный адрес 192.168.122.50:22 указывает номер порта 22 (локальная служба SSH) для соединения с 192.168.122.1. Вы также можете увидеть вторую запись с тем же номером порта, которая идентифицирует связанный демон SSH, прослушивающий соединения. Другие строки в этом выводе идентифицируют другие службы прослушивания.

ЦЕЛЬ СЕРТИФИКАЦИИ 3.06

Конфигурация сети и устранение неполадок

Теперь, когда вы ознакомились с основами IP-адресации и соответствующими командами, пришло время взглянуть на файлы конфигурации. Эти файлы конфигурации определяют, запускается ли сеть во время процесса загрузки. Если сеть активирована, эти файлы также определяют, настроены ли адреса и маршруты статически, как задокументировано, или динамически с помощью таких команд, как **dhclient**.

Базовая конфигурация сети только подтверждает, что системы могут общаться через их IP-адреса. Но этого недостаточно. Если вы указываете на системы, такие как **server1.example.com**, или на URL-адреса, такие как **www.mheducation.com**, конфигурации сети недостаточно, если разрешение имени хоста не работает.

!!!! On the Job !!!!

Некоторые из наиболее распространенных причин сетевых проблем носят физический характер. В этом разделе предполагается, что вы проверили все сетевые подключения. На **ВМ** это означает, что виртуальная сетевая карта не была случайно удалена на **ВМ** или на физическом хосте.

!!!!

Файлы конфигурации сети

Если возникают проблемы с конфигурацией сети, необходимо проверить текущее состояние сети. Для этого выполните следующую команду:

```
# systemctl status network
```

RHEL 7 использует службу, известную как **Network Manager**, для мониторинга и управления настройками сети. Используя инструмент командной строки **nmcli**, вы можете взаимодействовать с **Network Manager** и отображать текущее состояние сетевых устройств:

```
# nmcli dev status
```

Команда должна перечислить настроенные и активные устройства. Если ключевое устройство, такое, как **eth0**, не указано как подключенное, возможно, ваше сетевое подключение не работает или устройство не настроено. Ключевые файлы конфигурации находятся в каталоге **/etc/sysconfig/network-scripts**.

Иногда случаются ошибки. Если вы деактивировали адаптер или просто потеряли беспроводное соединение, попробуйте что-нибудь простое: перезагрузите сеть. Следующая команда перезапускает сеть с текущими файлами конфигурации:

```
# systemctl restart network.
```

!!!! On the Job !!!

Всегда используйте **systemctl** для выполнения сетевого сценария. Никогда не запускайте RHEL 7 сценарий **/etc/init.d/network** напрямую, потому что он может выполняться не корректно.

!!!!

Если простой перезапуск сетевых служб не работает, то пришло время заняться файлами. В каталоге **/etc/sysconfig/network-scripts** Red Hat Enterprise Linux хранит и получает сетевую информацию. С доступными инструментами настройки Red Hat вам не нужно трогать эти файлы, но приятно знать, что они там есть. Несколько типичных файлов показаны в **таблице 3-6**.

/etc/sysconfig/network

Если вы запустите команду **ip addr show** и не увидите выходных данных, это означает, что все сетевые устройства в настоящее время неактивны. Первое, что нужно проверить в этом случае, это содержимое файл конфигурации сети в **/etc/sysconfig/network**. Это довольно простой файл, который обычно содержит одну или две строки конфигурации. В системах, которые настроены на получение информации об адресации через **DHCP**, этот файл обычно пуст.

Если файл **/etc/sysconfig/network** содержит параметр **NETWORKING=no**, тогда сценарий **/etc/init.d/network** не активирует сетевые устройства. Еще одна директива, связанная с сетью, которая может появиться - это **GATEWAY**, если это один и тот же IP-адрес для всех сетевых устройств. В противном случае эта конфигурация поддерживается либо командой **dhclient**, либо задается в информации об IP-адресе для конкретного сетевого устройства в каталоге **/etc/sysconfig/network-scripts**.

Таблица 3-6 файлы в каталоге **/etc/sysconfig/network-scripts**

Файл в /etc/sysconfig/network-scripts	Описание
---------------------------------------	----------

ifcfg-lo	Настраивает устройство обратной связи, виртуальное устройство, которое используется для сетевого взаимодействия на локальном хосте.
ifcfg-*	Каждый установленный сетевой адаптер, такой как em1, получает свой собственный сценарий ifcfg-* . Например, eth0 - это файл ifcfg-eth0 . Этот файл содержит информацию об IP-адресе, необходимую для идентификации этого адаптера в сети.
network-functions	Этот сценарий содержит функции, используемые другими сетевыми сценариями для переключения сетевых интерфейсов up и down .
ifup-* and ifdown-*	Эти скрипты активируют и деактивируют назначенные им протоколы. Например, ifup-ppp вызывает устройство PPP, обычно телефонный модем.

/etc/sysconfig/network-scripts/ifcfg-lo

Говоря о каталоге **/etc/sysconfig/network-scripts**, возможно, основой сети является адрес обратной связи. Эта информация настраивается в файле **ifcfg-lo** в этом каталоге. Содержимое файла может помочь вам понять, как файлы в этом каталоге используются для сетевых устройств. По умолчанию в этом файле должны отображаться следующие записи, начиная с имени устройства обратной связи:

DEVICE=lo

За ним следуют **IP-адрес (IPADDR)**, **маска сети (NETMASK)** и **IP-адрес сети (NETWORK)**, а также соответствующий широковещательный адрес (**BROADCAST**):

IPADDR=127.0.0.1

NETMASK=255.0.0.0

NETWORK=127.0.0.0

BROADCAST=127.255.255.255

Следующие записи указывают, будет ли устройство активировано во время процесса загрузки, и общее имя устройства:

ONBOOT=yes

NAME=loopback

/etc/sysconfig/network-scripts/ifcfg-eth0

То, что вы видите в файле **ifcfg-eth0**, зависит от того, как был настроен первый сетевой адаптер **Ethernet**. Например, посмотрите на ситуацию, когда сеть была настроена только для целей установки. Если вы не настроили сеть при установке имени хоста во время процесса установки графического интерфейса, сеть не будет настроена в системе. В этом случае файл **ifcfg-eth0** будет содержать как минимум следующие две директивы:

HWADDR="F0:DE:F3:06:C6:DB"

TYPE=Ethernet

Конечно, если сеть не была настроена во время процесса установки, нет никаких причин для активации интерфейса во время загрузки:

ONBOOT="no"

По умолчанию RHEL 7 использует службу, известную как **Network Manager**, для управления настройками сети. Чтобы убедиться, что он работает, выполните команду **systemctl status NetworkManager**. Диспетчер сети включает в себя **nmcli**, инструмент командной строки для контроля состояния службы и применения изменений конфигурации сети.

Вместо того, чтобы изменять конфигурацию через **nmcli**, вы можете напрямую изменить файл конфигурации устройства. Для этого файл конфигурации, показанный на **рис. 3-10**, содержит руководство.

Большинство из этих директив просты. Они определяют устройство как сетевую карту Ethernet с именем **eth0**, используя определенный IP-адрес, маску сети, шлюз по умолчанию и DNS-сервер. Конечно, если вы предпочитаете использовать DHCP-сервер, статический сетевой адрес информация, указанная в последних пяти строках файла, будет опущена, а следующая директива будет изменена:

BOOTPROTO=dhcp

После сохранения файла вы все равно должны уведомить **Network Manager** об изменениях. Это достигается путем запуска следующих команд (**con** - сокращение от **connection**):

```
# nmcli con reload
# nmcli con down eth0
# nmcli con up eth0
```

Вскоре вы увидите, как использовать инструмент командной строки **Network Manager** для изменения конфигурации сетевого устройства.

Другие файлы /etc/sysconfig/network-scripts

Большинство файлов в каталоге **/etc/sysconfig/network-scripts** на самом деле являются скриптами. Другими словами, это исполняемые файлы, основанные на серии текстовых команд. Большинство этих сценариев основаны на командах **ifup** и **ifdown**, настроенных для типа сетевого устройства. Если необходимо настроить специальный маршрут, параметры конфигурации получают в этом каталоге свой специальный файл с именем, например **route-eth0**. Этот специальный маршрут будет указывать шлюз **к паре удаленный сетевой адрес/маска сети**. Один пример, основанный на системах, описанных в главе 1, может включать следующую директиву:

192.168.100.0/24 via 192.168.122.1

Инструменты настройки сети

Red Hat включает в себя несколько инструментов, которые можно использовать для настройки сетевых устройств в RHEL 7. Первым является инструмент командной строки **Network Manager**, **nmcli**. Если вы предпочитаете текстовый графический инструмент, **nmtui** можно запустить с виртуального терминала. В качестве альтернативы, редактор соединений Network Manager - это приложение **GTK+ 3**, которое можно запустить из командной строки графического интерфейса с помощью команды **nm-connection-editor**. Оболочка GNOME также включает графическую утилиту, которую можно открыть, нажав **Приложения | Разное | Сетевые соединения (Applications | Sundry | Network Connections)**.

Все упомянутые инструменты **взаимодействуют с Network Manager**, системным сервисом, который отвечает за управление сетевыми устройствами.

Инструмент настройки nmcli

Диспетчер сети может хранить различные профили, также известные как соединения, для одного и того же сетевого интерфейса. Это позволяет вам переключаться с одного профиля на другой. Например, вы можете иметь домашний профиль и рабочий профиль для адаптера Ethernet ноутбука и переключаться между ними в зависимости от сети, к которой вы подключены.

Вы можете отобразить все настроенные соединения в **Network Manager**, выполнив следующую команду:

```
# nmcli con show
```

NAME	UUID	TYPE	DEVICE
eth0	394f6436-5524-4154-b26e-6649b4d29027	802-3-ethernet	eth0

Чтобы показать, как можно использовать **nmcli** для настройки другого профиля соединения, давайте создадим новое соединение для **eth0**:

```
# nmcli con add con-name "eth0-work" type ethernet ifname eth0
```

Затем можно настроить статический IP-адрес и шлюз по умолчанию, как показано здесь:

```
nmcli con mod "eth0-work" ipv4.addresses "192.168.20.100/24 192.168.20.1"
```

Вы можете запустить **nmcli con show connection-id** для отображения текущих настроек соединения. Дополнительные свойства можно изменить с помощью инструмента командной строки **Network Manager**. Например, чтобы добавить DNS-сервер в соединение **eth-work**, выполните

```
# nmcli con mod "eth0-work" +ipv4.dns 192.168.20.1
```

Наконец, чтобы переключиться на новый профиль подключения, запустите

```
# nmcli con up "eth0-work"
```

Соединение может быть предотвращено от автоматического запуска при загрузке с помощью следующей команды:

```
# nmcli con mod "eth0-work" connection.autoconnect no
```

Инструмент настройки nmtui

Как следует из названия, этот инструмент предоставляет текстовый пользовательский интерфейс и может быть запущен из терминала командной строки. Просто запустите команду **nmtui**. С помощью инструмента консоли вам нужно будет нажимать клавишу **Tab**, чтобы переключаться между опциями(выборам), и пробел или клавишу ввода, чтобы выбрать выделенную опцию.

Нажимайте клавишу со стрелкой вниз, пока не будет выделен «Выход», а затем нажмите клавишу ввода. Сейчас сделайте резервную копию файла **ifcfg-eth0** из каталога **/etc/sysconfig/network-scripts**. На основании команды **diff** на рис. 3-11 сравнивается содержимое карты **eth0**, использующей протокол DHCP, настроенной в процессе установки, с платой, использующей статическую IP-адресацию, настроенной с помощью инструмента **nmtui**.

Директивы, показанные на рисунке 3-11, описаны в таблице 3-7.

РИСУНОК 3-11 Различия между статической и динамической конфигурацией сети

```
[root@server1 ~]# diff ifcfg-eth0 /etc/sysconfig/network-scripts/ifcfg-eth0
2c2
< BOOTPROTO=dhcp
---
> BOOTPROTO=none
8a9,13
> IPADDR=192.168.122.50
> PREFIX=24
> GATEWAY=192.168.122.1
> DNS1=192.168.122.1
> DOMAIN=example.com
10,11d14
< PEERDNS=yes
< PEERROUTES=yes
[root@server1 ~]# █
```

ТАБЛИЦА 3-7 Директивы конфигурации сети в каталоге /etc/sysconfig/network-scripts

Директива	Описание
DEVICE	Сетевое устройство; eth0 - это первый сетевой интерфейс Ethernet .
NAME	Имя профиля подключения интерфейса, используемого Network Manager.
UUID	Универсальный уникальный идентификатор для устройства.
HWADDR	Аппаратный (MAC) адрес для сетевого устройства.
TYPE	Тип сети; должен быть установлен « Ethernet » для устройства Ethernet .
ONBOOT	Директива, которая указывает, запускается ли сетевое устройство во время процесса загрузки.
BOOTPROTO	Может быть установлено « none » для статической конфигурации или « dhcp » для получения IP-адресов от DHCP-сервера.
IPADDR0	Статический IP-адрес; дополнительные IP-адреса могут быть указаны с помощью переменных IPADDR1, IPADDR2,...
PREFIX	Маска сети в формате CIDR (т.е. /24)
GATEWAY0	IP-адрес шлюза по умолчанию.
DEFROUTE	Двоичная директива для установки интерфейса в качестве маршрута по умолчанию.
DNS1	IP-адрес первого DNS-сервера .
DOMAIN	Указывает список поиска домена в /etc/resolv.conf .
PEERDNS	Двоичная директива, позволяющая модифицировать /etc/resolv.conf .
IPV6INIT	Двоичная директива, которая позволяет использовать адресацию IPv6.
USERCTL	Двоичная директива, позволяющая пользователям управлять сетевым устройством.
IPV4_FAILURE_FATAL	Двоичная директива; если установлено значение « no », при подключении к сетям IPv6 позволяет завершить настройку IPv6 в случае сбоя конфигурации IPv4.

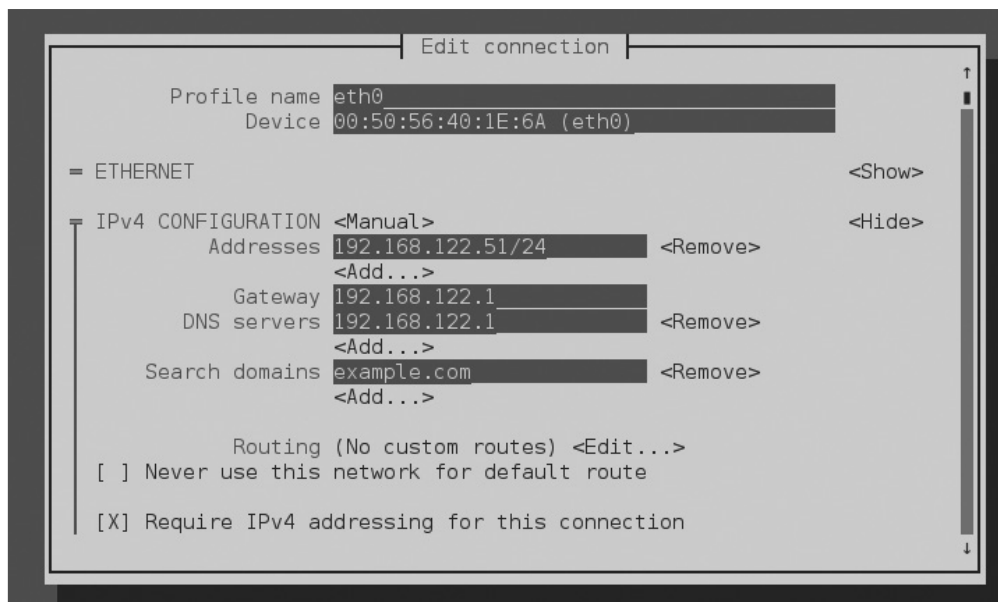
УПРАЖНЕНИЕ 3-2

Настройте сетевую карту

В этом упражнении вы настроите первую сетевую карту **Ethernet** с помощью инструмента текстового интерфейса пользователя **Network Manager**. Все, что вам нужно, это интерфейс командной строки. Неважно, находится ли командная строка в графическом

интерфейсе или виртуальном терминале. Чтобы настроить сетевую карту, выполните следующие действия:

1. **Создайте резервную копию** текущего файла конфигурации для первой карты **Ethernet**. Обычно это **ifcfg-eth0** в каталоге **/etc/sysconfig/network-scripts**. Для других имен интерфейса, таких как **em1**, замените соответственно. (Подсказка: используйте команду **cp**, а не команду **mv**.)
2. Запустите команду **nmtui**.
3. В появившемся меню выберите **Изменить соединение (Edit)** должно быть выделено **Соединение (Connection)**. При необходимости нажимайте стрелки или клавиши табуляции до тех пор, пока не выберете нужный вариант. Затем нажмите ввод.
4. На появившемся экране должна быть выделена первая сетевая карта Ethernet. Если это так, нажмите Enter.
5. В показанном здесь окне **Изменить соединение (Edit)** в разделе **Конфигурация IPv4 (IPv4 Configuration)** может быть выбран параметр **Автоматически (Automatic)**. Если это так, выделите его и нажмите ввод; затем выберите **Вручную (Manual)**.



6. Выделите опцию **Показать (Show)** справа от **Конфигурации IPv4** и нажмите Enter. Откроется окно с текущими настройками IPv4.
7. Введите информацию об IP-адресе для системы. Настройки, отображаемые в окне, основаны на настройках, описанных в главе 1 для системы server1.example.com. По завершении выделите **ОК** и нажмите ввод.
8. Вы вернулись на экран устройства. Убедитесь, что **Выход (Quit)** выделен, и нажмите Enter.
9. Деактивируйте, а затем повторно активируйте первую карту **Ethernet** с помощью команд **ifdown eth0** и **ifup eth0** и проверьте результат с помощью команд **ip addr show eth0** и **ip route**. Конфигурация сетевой карты и связанная таблица маршрутизации должны отражать новые настройки.
10. Чтобы восстановить исходную конфигурацию, восстановите файл **ifcfg-eth0** в **/etc/sysconfig/network-scripts** и перезапустите сеть с помощью команды **systemctl restart network**.

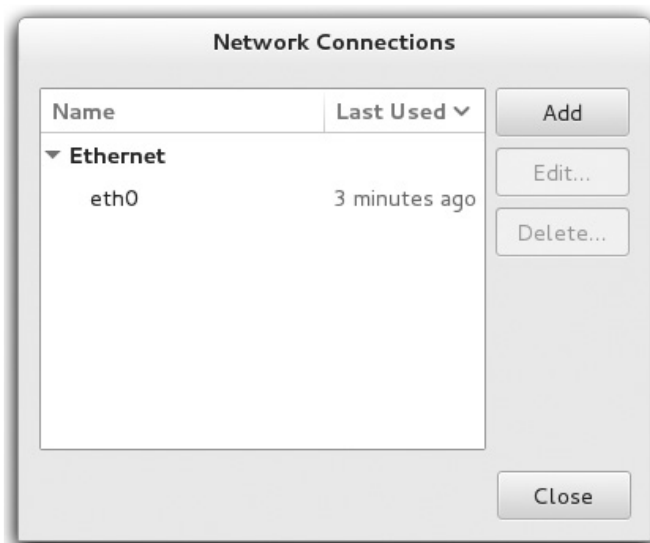
Редактор соединений Network Manager

Теперь вы будете работать со стандартным графическим инструментом управления сетью для RHEL 7, инструментом редактора соединений **Network Manager**. Благодаря

количеству пользователей, подключенных к нескольким сетевым подключениям, **Network Manager** разработан так, чтобы переключение между, скажем, беспроводным и Ethernet-соединением было максимально плавным. Но это больше относится к портативным системам, чем к серверам. Для наших целей все, что вам нужно знать, это как настроить сетевую карту с помощью этого инструмента.

Редактор соединений Network Manager не является чем-то новым, потому что он был в использовании в **Fedora** в течение нескольких лет. Он работает только в графическом интерфейсе. Чтобы запустить его, вы можете выполнить команду **nm-connection-editor**. Откроется инструмент «Сетевые соединения» (**Network Connections**), показанный на **рисунке 3-12**.

РИСУНОК 3-12 Редактор соединений Network Manager



Как видно из рисунка, инструмент перечисляет профиль подключения для первого сетевого интерфейса Ethernet. Он также поддерживает настройку других типов сетей, включая беспроводные, мобильные широкополосные карты (например, используемые для подключения к сетям 3G и 4G) и соединения цифровой абонентской линии (DSL). На обычном сервере основное внимание уделяется надежным соединениям, которые по-прежнему основаны на стандартном проводном устройстве Ethernet.

Выделите профиль подключения первого устройства **Ethernet (eth0)** и нажмите «**Изменить**» (**Edit**); затем выберите вкладку «**Параметры IPv4**» (**IPv4 Settings**). Откроется окно, показанное на **рисунке 3-13**. Если предварительно не настроено, предполагается, что сетевой интерфейс получит свои параметры конфигурации от DHCP-сервера.

Щелкните раскрывающееся текстовое поле «**Метод**» (**Method**). Хотя поддерживается настройка сетевой карты несколькими различными способами, в данном случае интерес представляет только способ «**Вручную**» (**Manual**). Выберите эту опцию, и раздел адреса в окне больше не должен быть отключен. Теперь добавьте информацию об IP-адресе для системы. На основе системы server1.example.com, описанной в главе 1, здесь приведены соответствующие параметры:

- IP address 192.168.122.50
- Network mask 255.255.255.0 (24 in CIDR notation is an acceptable equivalent in this field)
- Gateway address 192.168.122.1
- DNS server 192.168.122.1
- Search domains example.com
- Require IPv4 Addressing for This Connection to Complete Deselected

При правильном вводе конфигурация, связанная с первой платой Ethernet, называется именем подключения, указанным на **рисунке 3-13**. Для этой конфигурации параметры сохраняются в файле **ifcfg-eth0** в каталоге **/etc/sysconfig/network-scripts**.

РИСУНОК 3-13 Редактирование соединения Ethernet в редакторе соединений Network Manager



Настроить разрешение имен

Последним элементом конфигурации сети обычно **является разрешение имен**. Другими словами, имеет ли локальная система информацию, необходимую для перевода доменных имен, таких? как **mheducation.com**, в **IP-адреса**, такие как **198.45.24.143**?

Разрешение имен было легким, когда Unix впервые разрабатывался. Когда предшественник Интернета был впервые введен в эксплуатацию, во всемирной компьютерной сети было четыре хоста, по одному компьютеру в каждом из четырех разных университетов. Было легко настроить статический файл со списком каждого из их имен и соответствующих адресов. Этот файл превратился в то, что в Linux называется **/etc/hosts**.

Но сейчас Интернет стал более сложным. Хотя вы можете попытаться настроить базу данных каждого доменного имени и IP-адреса в Интернете в файле **/etc/hosts**, это займет почти вечность и не будет масштабируемым решением. Вот почему большинство пользователей **устанавливают соединения с DNS-серверами**. На RHEL 7 это все еще задокументировано в файле конфигурации **/etc/resolv.conf**. Как RHCE, вы должны знать, как **настроить DNS-сервер** только для кэширования; этот вопрос рассматривается в **главе 13**. Конфигурация DNS-сервера не является требованием RHCSA.

В небольших сетях некоторые администраторы настраивают файл **/etc/hosts** в качестве базы данных для имени каждой системы и IP-адреса в локальной сети. При желании администраторы могут даже настроить несколько IP-адресов доменов в Интернете, хотя эта настройка будет нарушена, если какой-либо из этих доменов Интернета внес изменения DNS самостоятельно.

Но если вы настроили соединение с DNS-сервером и системами в **/etc/hosts**, что сначала ищется? Именно поэтому файл конфигурации **/etc/nsswitch.conf** определяет порядок поиска для различных баз данных службы имен, включая имена хостов.

Файлы конфигурации имени хоста

RHEL 7 включает как минимум четыре файла конфигурации имени хоста: **/etc/hostname**, **/etc/hosts**, **/etc/resolv.conf** и **/etc/nsswitch.conf**. Эти четыре файла, вместе взятые, содержат локальное имя хоста, локальную базу данных имен хостов и IP-адресов, IP-адрес одного или нескольких DNS-серверов и порядок, в котором эти базы данных рассматриваются.

/etc/nsswitch.conf

Файл **/etc/nsswitch.conf** определяет приоритеты поиска в базе данных для всего, от аутентификации до служб имен. Как и файл переключения сервера имен, он включает следующую запись, которая определяет, какая база данных сначала ищется:

hosts: files dns

Когда система получает запрос на поиск имени хоста, такого как **outsider1.example.org**, предыдущая директива означает, что файл **/etc/hosts** ищется первым. Если это имя не найдено в **/etc/hosts**, следующим шагом будет **поиск** доступных настроенных **DNS-серверов**, обычно использующих те, которые настроены в файле **/etc/resolv.conf**.

Библиотека распознавателя (**resolver**) также использует информацию в файле **/etc/host.conf**. Запись в этом файле просто

multi on

который говорит системе возвращать все IP-адреса в **/etc/hosts**, которые сопоставлены с одним и тем же именем хоста, вместо того, чтобы возвращать только первую запись.

/etc/hosts

Файл **/etc/hosts** представляет собой статическую базу данных **имен хостов/полных доменных имен (hostnames/FQDNs)** и **IP-адресов**. Он подходит для небольших относительно статичных сетей. Тем не менее, это может быть проблемой для сетей, где часто происходят изменения. Каждый раз, когда система добавляется или удаляется, вам придется изменять этот файл - не только в локальной системе, но и в любой другой системе в этой сети.

Он хорошо подходит для локальных сетевых систем, созданных в **главе 1**. Простая версия файла может содержать следующие записи:

```
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4
::1 localhost localhost.localdomain localhost6 localhost6.localdomain6
192.168.122.50 server1.example.com server1
192.168.122.150 tester1.example.com tester1
192.168.100.100 outsider1.example.org outsider1
```

В некоторых случаях вы можете настроить несколько записей для IP-адреса. Например, следующие записи могут быть добавлены для указания IP-адресов для веб-серверов и FTP-серверов:

```
192.168.122.50 www.example.com
192.168.122.150 ftp.example.com
```

/etc/resolv.conf

Стандартный файл для документирования местоположения DNS-серверов по-прежнему остаётся **/etc/resolv.conf**. Как правило, он имеет одну или две записи, аналогичные следующим:

```
search example.com
nameserver 192.168.122.1
```

Директива **search** добавляет имя домена **example.com** к поиску простых имен хостов. Директива **nameserver** указывает **IP-адрес настроенного DNS-сервера**. Если вы сомневаетесь в работоспособности DNS-сервера, выполните одну из следующих команд:

```
# dig @192.168.122.1 mheducation.com
# host mheducation.com 192.168.122.1
```

При необходимости замените IP-адрес, связанный с директивой сервера имен в вашем **/etc/resolv.conf** файл. В этом файле вы можете указать до трех директив сервера имен (**nameserver**).

!!!! On the Job !!!!

Возможно, нет смысла редактировать файл **/etc/resolv.conf** напрямую. Если вы настроили DNS-серверы с помощью другого инструмента, такого как **nmcli**, редактор подключений, или с помощью директивы **DNS1** в файле конфигурации устройства, **Network Manager** будет перезаписывать любые изменения, которые вы вносите при непосредственном редактировании этого файла, если вы не переопределите это поведение с помощью **PEERDNS** в файл **ifcfg**.

!!!!

Параметры конфигурации имени хоста

Во время загрузки сетевая служба ищет файл **/etc/hostname** для определения значения локального имени хоста. Имя хоста должно быть установлено как полное доменное имя, например **tester1.example.com**. Как предлагалось ранее, это простой файл, в котором имя хоста задокументировано с помощью следующей директивы:

```
tester1.example.com
```

Конечно, вы можете изменить значение имени хоста с помощью команды **hostname newname**. Однако это изменение носит временный характер и не отражается в файле **/etc/hostname**. Чтобы сделать изменение постоянным, используйте команду **hostnamectl set-hostname newname**.

СЦЕНАРИЙ И РЕШЕНИЕ	
Сеть не работает.	Проверьте физические соединения. Запустите ip link show , чтобы проверить активные интерфейсы. Запустите команду systemctl status network .
Невозможно получить доступ к удаленным системам.	Используйте команды ping и traceroute для проверки доступа к локальным, а затем к удаленным IP-адресам.
Текущие настройки сети приводят к конфликтам.	Проверьте конфигурацию сетевого устройства файлы в /etc/sysconfig/network-scripts . Просмотрите настройки с помощью редактора соединений Network Manager .

Сетевые настройки не соответствуют.	Проверьте конфигурацию сетевого устройства в файлах /etc/sysconfig/network-scripts . Просмотрите настройки с помощью редактора соединений Network Manager . Сценарий предполагает желание статической конфигурации сети, поэтому рассмотрите соответственно.
Имя хоста не распознано.	Просмотрите /etc/hostname , выполните команду hostname и просмотрите /etc/hosts на предмет согласованности.
Имена удаленных хостов не распознаны.	Просмотрите /etc/hosts и /etc/nsswitch.conf . Проверьте /etc/resolv.conf для соответствующего IP-адреса DNS-сервера. Запустите команду dig , чтобы проверить разрешение DNS

РЕЗЮМЕ СЕРТИФИКАЦИИ

Глава охватывает основные инструменты командной строки, ранее связанные с предварительными условиями экзамена Red Hat. Поскольку эти цели были включены в основную часть RHCSA, они были объединены с конфигурацией сети, чтобы вы могли практиковать эти инструменты командной строки.

Командная строка начинается с оболочки, которая позволяет взаимодействовать с операционной системой с помощью различных команд. Хотя в целях не указана **оболочка (shell)**, оболочкой по умолчанию в большинстве дистрибутивов Linux, включая RHEL 7, **является bash**. Вы можете запустить командную строку на одной из консолей по умолчанию или на терминале в графическом интерфейсе. В командной строке **bash** вы можете управлять файлами и каталогами, через которые настроен и организован Linux. Поскольку файлы конфигурации Linux в основном представлены в текстовом формате, их можно настраивать, искать и изменять с помощью различных команд. Текстовые файлы Linux можно анализировать как потоки данных, которые можно интерпретировать и обрабатывать. Чтобы редактировать текстовый файл, вам нужен текстовый редактор, например, **vim** или **gedit**.

Онлайн документация в Linux обширна. Помощь начинается с командных ключей, таких, как **-h** и **--help**, которые предоставляют подсказки о том, какие ключи можно применить с командой. Документация по командам продолжается в **man** и **info** страницами. Многие пакеты содержат обширные файлы документации в каталоге **/usr/share/info** во многих, возможно, в большинстве случаев, вам не нужен доступ в Интернет, чтобы найти необходимый подсказки.

Linux по своей сути является сетевой операционной системой. Сетевые устройства, такие как **eth0**, могут быть настроены как с адресами IPv4, так и с адресами IPv6. Обзор и настройка сети включают команды **ip**, **ifup**, **ifdown** и **dhclient**. Дополнительные связанные команды включают **ss**, **ping** и **traceroute**. Связанные файлы конфигурации начинаются с файла **/etc/sysconfig/network**. Отдельные устройства настраиваются в файлах каталога **/etc/sysconfig/network-scripts**. Устройства сети могут быть настроены с помощью команд **nmcli** и **nmtui** в консоли и в редакторе соединений сетевого менеджера (**Network Manager Connections Editor**).

Пару минут тренировки

Вот некоторые из ключевых моментов целей сертификации в Главе 3.

Shells

- По умолчанию оболочка Linux - **bash**.
- По умолчанию доступно шесть виртуальных терминалов командной строки; если установлен графический интерфейс захватывает первый виртуальный терминал

- Вы можете открыть несколько терминалов командной строки в графическом интерфейсе.
- Оболочка shells работают с тремя потоками данных: **stdin**, **stdout** и **stderr**. С этой целью, перенаправление команд означает, что потоки данных могут управляться операторами такие, как **>**, **>>**, **<**, **|** и **2>**.

Стандартные инструменты командной строки

- Все в Linux может быть сведено к файлу.
- Такие команды, как **pwd** и **cd**, могут помочь в навигации по каталогам.
- Такие понятия, как пути к каталогам, **PATH** и **тильда (~)** могут помочь вам понимать и использовать команды в оболочке (**shells**).
- Основные команды позволяют вам найти нужные файлы и прочитать содержимое файла. Эти команды включают в себя **ls**, **find** и **locate**.
- Команды создания и удаления файлов включают в себя **touch**, **cp**, **ln**, **mv** и **rm**; соответствующие команды создания и удаления каталогов: **mkdir** и **rmdir**.
- Команды можно настроить с помощью команды **alias**.

Управление текстовыми файлами

- Linux управляется через серию текстовых файлов конфигурации.
- Текстовые файлы можно читать, как потоки данных с помощью таких команд, как **cat**, **less**, **more**, **head** и **tail**.
- Новые файлы могут быть созданы, скопированы, перемещены, связаны и удалены командами **touch**, **cp**, **mv**, **ln** и **rm**. Команды можно настроить с помощью команды **alias**.
- Фильтры файлов, такие, как команды **sort**, **grep**, **wc**, **sed** и **awk**, поддерживают обработку текстовых потоков.
- Понимание текстовых редакторов является критическим навыком. Более ранняя версия RHCSA в целях экзаменов указано использование **vim** и **gedit**.

Локальная онлайн документация

- Если вам нужна подсказка для команды, попробуйте выполнить ее самостоятельно; в качестве альтернативы, попробуйте команду с ключами **-h** или **--help**.
- Справочные страницы (**man**) команд часто содержат примеры; с помощью команд **whatis** и **apropos** можно искать справочные страницы на разные темы.
- Если для команды или файла доступно информационное руководство, вы найдете его в каталоге **/usr/share/info**.
- Многие пакеты включают обширную документацию и примеры в каталоге **/usr/share/doc**.

Сетевой учебник

- **IPv4-адреса имеют 32 бита**. Существует пять классов адресов IPv4 и три разных набора частных IPv4-адресов, подходящих для **настройки TCP/IP** в частной локальной сети.
- **Маска подсети (subnet mask)** (также известная как маска сети или префикс) используется для поиска сети и части хоста IP-адреса.
- Такие инструменты, как **ping**, **traceroute**, **tracert**, **ip** и **ss**, помогут вам диагностировать проблемы по локальной сети.
- Файлы конфигурации разрешения имен, такие, как **/etc/resolv.conf**, определяют, как система находит правильный IP-адрес; файл **/etc/resolv.conf** может быть настроен из сервера DHCP с помощью команды **dhclient** или с помощью диспетчера сети (**Network Manager**).

Конфигурация сети и устранение неполадок

- Отдельные сетевые устройства настраиваются в каталоге `/etc/sysconfig/network-scripts`.
- Инструменты конфигурации сети включают консольные команды **nmcli** и **nmtui**, а также редактор соединений **Network Manager**.
- Файлы конфигурации разрешения имен включают `/etc/nsswitch.conf`, `/etc/hosts` и `/etc/resolv.conf`.

САМОПРОВЕРКА

Следующие вопросы помогут вам оценить ваше понимание материала, представленного в этом

глава. Поскольку на экзаменах Red Hat нет вопросов с несколькими вариантами ответов, нет вариантов с несколькими вариантами ответов вопросы в этой книге. Эти вопросы исключительно проверяют ваше понимание главы. Получение результатов, а не запоминание пустяков, это то, что рассчитывает на экзамены Red Hat.

Shells

1. Как называется оболочка Linux по умолчанию?

2. Какая комбинация клавиш из графического интерфейса перемещается на виртуальную консоль 3?

Стандартные инструменты командной строки

3. Какая отдельная команда создает серию каталогов `/abc/def/ghi/jkl`?

4. Какой символ представляет домашний каталог текущего пользователя?

Управление текстовыми файлами

5. Какая команда перечисляет последние 10 строк файла `/var/log/messages`?

6. Какая команда возвращает строки с термином **Linux** из файла `/var/log/dmesg`?

Локальная онлайн документация

7. Какая команда ищет в базе данных **man-страниц** руководства, которые ссылаются на команду **passwd** и конфигурационный файл ?

8. Если для гипотетической команды и файла **abcde** имеются справочные страницы **man**, в разделах 5 и 8 введите команде, которая обязательно вызовет справочные страницы **man** из раздела 5.

Сетевой учебник

9. При адресации IPv4 с сетевым адресом 192.168.100.0 и широковещательным адресом 192.168.100.255, каков диапазон назначаемых IP-адресов?

10. Учитывая адреса, описанные в вопросе 9, какая команда назначает адрес IPv4 192.168.100.100 к сетевому устройству eth0?

Конфигурация сети и устранение неполадок

11. Какой полный путь к файлу конфигурации с именем хоста локальной системы?

12. Каков полный путь по умолчанию к файлу конфигурации, связанному с адаптером Ethernet eth0 для локальной системы?

Лабораторная работа

Во время экзаменов Red Hat задания будут представлены в электронном виде. Таким образом, эта книга также представляет большинство лабораторий в электронном виде. Для получения дополнительной информации см. Раздел «Лабораторные вопросы» в конце **главы 3**.

Хотя больше нет отдельного раздела по устранению неполадок в экзаменах **RHCSA/RHCE**, навыки устранения неполадок по-прежнему полезны при практических экзаменах и на работе. Таким образом, первые три лабораторных занятия помогут вам настроить сценарий устранения неполадок. Стандартное предупреждение: эти лабораторные работы, даже при правильном выполнении, могут сделать систему непригодной для использования. Поэтому не запускайте эти лаборатории в рабочих системах. В худшем случае, вы должны быть готовы переустановить RHEL 7 в этой системе.

Для первых трех лабораторных работ вам нужно будет выполнить сценарии из каталога **Chapter3/ DVD-диска**, входящего в комплект книги. Например, скрипт для первой лабораторной работы называется **Ch3Lab1**. В любом случае вам необходимо скопировать скрипты в тестовую систему **server1.example.com**. Следующие шаги предполагают, что система находится на виртуальной машине на основе **KVM**. Не просматривайте содержимое скопированных сценариев или файлов или файлов в каталоге **/root/backup** до тех пор, пока лабораторная работа не будет завершена. Следующие шаги являются общими для этих первых трех лабораторий:

1. Откройте **KVM Virtual Machine Manager** из командной строки графического интерфейса с помощью команды **virt-manager**.
2. Подключитесь к локальной системе (**QEMU**).
3. Дважды щелкните виртуальную машину с системой **server1.example.com**. В появившемся окне нажмите **Вид | Подробности (View | Details)**.
4. Вставьте DVD для книги. Используйте параметры, которые появляются, чтобы подключить привод CD/DVD к виртуальной машине. Вернитесь в консоль для виртуальной машины, нажав **Вид | Консоль (View | Console)**.
5. Загрузите систему **server1.example.com**. Смонтируйте DVD книги с помощью команды **mount /dev/cdrom /media**.
6. Войдите в систему с учетной записью администратора.

7. Убедитесь, что вы находитесь в каталоге **/root** с помощью команды **cd /root**. Скопируйте отмеченные сценарии (и связанные тестовые файлы) с DVD-диска в домашний каталог пользователя **root** с помощью команды **cp /media/Chapter3/Ch3Lab* ~.**
8. Эти лабораторные работы должны работать, даже если каталог **/root/backup** существует, хотя в этом случае вы увидите сообщение об ошибке.
9. Убедитесь, что сценарии являются исполняемыми; выполните команду **chmod + x Ch3Lab?**.

Лабораторная работа 1

Эта лабораторная работа основана на сценарии **Ch3Lab1**, доступном в подкаталоге **Chapter3/**, как описано ранее. Если вы следовали инструкциям в начале этого раздела, скрипт должен быть доступен в каталоге **/root**. Перейдите в этот каталог и выполните следующие действия:

1. Выполните сценарий для этой лабораторной работы с помощью команды **./Ch3Lab1**.
2. Деактивируйте текущее сетевое устройство. Если это **eth0**, запустите команду **ifdown eth0**. Если вы не уверены, выполните команду **ip link show**; замените **eth0** при необходимости.
3. Перезапустите сеть с помощью команды **systemctl restart network**. Что происходит? Вы можете подключиться? Начните просматривать файлы конфигурации сети, пока не сможете выявить и устранить проблему.
4. Если вы сразу не видите проблему, просмотрите различные файлы конфигурации, описанные в этой главе. Возможно, вам придется внести пару изменений в файл конфигурации.
5. Как только вы определили и устранили проблему в соответствующем файле конфигурации, вы будете готовы выполнить команду **systemctl restart network** и увидеть успешные сообщения.
6. Убедитесь, что конфигурация системы работает после перезапуска сценария **/etc/init.d/network** и перезагрузки системы (system reboot).
7. Если у вас все еще есть проблемы, посмотрите файл в каталоге **/root/backup**. Если вы не уверены, какой файл был изменен, выполните команду **ls -ltr ~/backup**. Это будет последний файл, указанный в выводе. Даже если у вас нет проблем, посмотрите на файл в этом каталоге. Возможно, вы что-то забыли.

Лабораторная работа 2

Эта лабораторная работа основана на скрипте **Ch3Lab2**, доступном в подкаталоге **Chapter3/**, как описано ранее. Если вы следовали инструкциям в начале этого раздела, скрипт должен быть доступен в каталоге **/root**. Перейдите в этот каталог и выполните следующие действия.

Помните, что в этой лабораторной работе предполагается, что первый адаптер **Ethernet** в системе активен, как устройство **eth0**. Если это не так в вашей тестовой системе, скрипт, используемый в этой лаборатории, не будет работать.

1. Выполните сценарий, связанный с этой лабораторной работой, с помощью команды **./Ch3Lab2**.
2. Проверьте соединения с локальными и удаленными системами, такими как имя хоста или полное доменное имя для локальной системы и IP-адрес сервера имен в файле **/etc/resolv.conf**. Попробуйте команду **ping** в зависимости от ситуации.
3. Просмотрите соответствующие файлы конфигурации сети. Запустите команду **ip addr show**. Просмотрите информацию, приведенную для согласованности. Можете ли вы определить проблему еще?

4. Если вы еще не можете определить проблему, откройте редактор подключений **Network Manager**. Проверьте **проводные соединения Ethernet**. Если вы видите несоответствие, устраните его.
5. Убедитесь, что конфигурация системы работает после перезагрузки системы.
6. Если у вас все еще есть проблемы, посмотрите файл в каталоге **/root/backup**. Если вы не уверены, какой файл был изменен, запустите команду **ls -ltr ~ /backup**. Это будет последний файл, указанный в выводе.
7. После завершения этой лабораторной работы восстановите последний файл, скопированный в каталог **/root/backup**, в его исходное местоположение. Стандартный каталог описан в основной части этой главы.

Лабораторная работа 3

Эта лабораторная работа основана на скрипте **Ch3Lab3**, доступном в подкаталоге **Chapter3/**, как описано ранее. Если вы следовали инструкциям в начале этого раздела, скрипт должен быть доступен в каталоге **/root**. Перейдите в этот каталог и выполните следующие действия.

Помните, что в этой лабораторной работе предполагается, что **первый адаптер Ethernet** в системе активен, как **устройство eth0**. Если это не так в вашей тестовой системе, скрипт, используемый в этой лаборатории, не будет работать.

1. Выполните сценарий для этой лабораторной работы с помощью команды **./Ch3Lab3**.
2. Проверьте соединения с локальными и удаленными системами, такими как имя хоста или полное доменное имя для локальной системы или IP-адрес сервера имен, указанный в файле **/etc/resolv.conf**. Используйте команду **ping** по необходимости.
3. Просмотрите соответствующие файлы конфигурации сети. Запустите команду **ip link show**. Можете ли вы определить проблему еще?
4. Если вы еще не можете определить проблему, попробуйте команду **ip addr show**. Видите ли вы разницу в выходе? Чем ты занимаешься?
5. Убедитесь, что конфигурация системы работает после перезапуска сетевой службы, а также перезагрузки системы.

Лабораторная работа 4

В этой лабораторной работе вы создадите файл **/etc/hosts** для разных систем в локальной сети.

Инструкции в этой лабораторной работе основаны на тестовых системах, описанных в главе 1.

1. Сделайте резервную копию файла **/etc/hosts** в соответствующий каталог, например **/root/backup**.
2. Откройте файл **/etc/hosts**.
3. Если вы видите запись с локальным именем хоста, не меняйте эту запись.
4. Добавьте записи в **/etc/hosts** для систем **tester1** и **outsider1**.
5. Когда все отмеченные системы работают, проверьте результат. Если эти системы находятся на виртуальных машинах, это может частично зависеть от активации переадресации IP, как описано в главе 1. Выполните команду **ping**, сначала на каждом IP-адресе в **/etc/hosts**, а затем на каждом имени хоста в **/etc/hosts**.

Лабораторная работа 5

В этой лабораторной работе вы будете использовать инструмент редактора соединений **Network Manager** из консоли. Но прежде чем, сделать это, не забудьте создать резервную копию соответствующих файлов конфигурации. Поскольку это консольный инструмент, вам придется использовать клавиши **TAB** и **ENTER** или пробел, чтобы сделать выбор.

1. Сделайте резервную копию файлов конфигурации сети для первого адаптера Ethernet из каталога **/etc/sysconfig/network-scripts** с помощью команды, подобной следующей:
cp /etc/sysconfig/network-scripts/ifcfg-eth0 /root/backup
2. Откройте инструмент настройки **nmtui** из командной строки с помощью команды с тем же именем.
3. В появившемся меню консоли выберите «Редактировать соединение» и нажмите «ENTER».
4. В появившемся меню выберите первую сетевую карту Ethernet и нажмите ENTER.
5. Внесите изменения в текущую конфигурацию. Предполагая конфигурацию статического IP-адреса для одной из систем, описанных в главе 1, вы сможете внести незначительные изменения в IP-адрес. Например, вы можете изменить IPv4-адрес `server1.example.com` с `192.168.122.50` на `192.168.122.51`. Выделите OK и нажмите ENTER.
6. Вы вернулись на экран устройства. Выделите опцию Quit и нажмите ENTER.
7. Отключите, а затем повторно активируйте первую карту Ethernet с помощью команд **ifdown eth0** и **ifup eth0** и проверьте результат с помощью команды **ip addr show**.
8. Используйте команду **diff**, чтобы проанализировать разницу между файлами **ifcfg-eth0** в каталогах **/etc/sysconfig/network-scripts** и в **/root directories**.
9. Если вы действительно заинтересованы в различиях, сравните файлы, созданные со статическим и динамическим IP-адресом. В инструменте **nmtui** это основано на установке конфигурации IP-адреса на Автоматический в Шаге 5.
10. Чтобы восстановить исходную конфигурацию, восстановите файл **ifcfg-eth0** в каталог **/etc/sysconfig/network-scripts** и перезапустите сеть с помощью команды **systemctl restart network**.

Лабораторная работа 6

Повторите процесс, описанный в лабораторной работе 6, с помощью редактора сетевых подключений. Чтобы открыть его, запустите графический интерфейс и нажмите **Приложения | Разное | Сетевые соединения (Applications | Sundry | Network Connections)**; альтернативно, из командной строки запустите команду **nm-connection-editor**. Эта команда также работает с удаленной консоли, если вы вошли в систему с помощью команды **ssh -X**, описанной в главе 2.

ОТВЕТЫ НА САМОПРОВЕРКУ

Shells

1. По умолчанию для Linux используется оболочка **bash**, также известная как оболочка Bourne-Again.
2. В графическом интерфейсе комбинация клавиш, которая перемещается на виртуальную консоль 3, называется **ctrl-alt-f3**.

Стандартные инструменты командной строки

3. Единственная команда, которая создает серию каталогов **/abc/def/ghi/jkl**:
mkdir -p /abc/def/ghi/jkl.
4. Символом, обозначающим домашний каталог текущего пользователя, является **тильда (~)**.

Управление текстовыми файлами

5. Команда, в которой перечислены последние 10 строк файла **/var/log/messages**, является **tail -n 10 /var/log/messages** . Поскольку по умолчанию используется 10 строк, **tail /var/log/messages** также приемлемо.
6. Команда, которая возвращает строки с термином **Linux** из файла **/var/log/dmesg**, - это **grep Linux /var/log/dmesg**. Допустимы другие варианты, такие как **cat /var/log/dmesg | grep Linux** .

Локальная онлайн документация

7. Команда, которая ищет в базе данных **man-страниц** руководства, ссылающиеся на команду **passwd** и конфигурационный файл - **whatis passwd**. Команды **apropos** и **man -k** идут далее, потому что они содержат **man-страницы** с текстом "**passwd**" в команде или описании.
8. Команда, которая **вызывает man-страницу** из раздела **5** для гипотетической команды **abcde** и файл – это **man 5 abcde**.

Сетевой учебник

9. Диапазон назначаемых IP-адресов в указанной сети IPv4 составляет от 192.168.100.1 до 192.168.100.254.
10. Учитывая адреса, описанные в вопросе 9, команда, которая назначает адрес IPv4 192.168.100.100 для сетевого устройства eth0 - это **ip addr add 192.168.100.100/24 dev eth0**.

Конфигурация сети и устранение неполадок

11. Полный путь к файлу конфигурации с именем хоста локальной системы: **/etc/hostname**.
12. Полный путь к файлу конфигурации, связанному с первым адаптером Ethernet для локальной системы называется **/etc/sysconfig/network-scripts/ifcfg-eth0**. Если вы используете разные профили подключения в Сетевой менеджер, вы найдете файл для каждого профиля подключения в каталоге **/etc/sysconfig/network-scripts**.

ОТВЕТЫ ЛАБОРАТОРНОЙ РАБОТЫ

Лабораторная работа 1

Эта лабораторная работа проверяла ситуацию, когда сеть была деактивирована с наиболее безобидными настройками, **директива NETWORKING** в файле **/etc/sysconfig/network**. Когда установлено в **no**, этот параметр деактивирует сеть в системе. Больше ничего не изменилось; информация об IP-адресе для конкретного сетевые карты все еще верны. Конечно, вы все равно можете активировать сеть с помощью других средств, но если директива **NETWORKING=no** в указанном файле, такие изменения не выдержат перезагрузки.

Сценарий, использованный в этой лабораторной работе, сохранил оригинальную копию в каталоге **/etc/sysconfig/network** в **/root/backup**. Теперь, когда лабораторная работа завершена, вы можете восстановить этот файл в его исходное местоположение. Будьте осторожны, неизменный флаг был применен к скопированному файлу; чтобы удалить его из каталога **/root/backup**, вам сначала нужно убрать неизменный флаг с помощью команды **chattr -i**.

Лабораторная работа 2

Эта лабораторная работа устанавливает недопустимую конфигурацию IP-адреса для **первого адаптера Ethernet, eth0**. Стандарт для системы, настроенные в главе 1, основаны на сети 192.168.122.0/24. Файл конфигурации в каталоге **/etc/sysconfig/network-scripts** может

иметь несколько другое имя, в зависимости от того, как этот адаптер был настроен. Исходный файл из этого каталога был перемещен в каталог **/root/backup**. Если ваши усилия по воссозданию этого файла конфигурации не удаются, восстановите исходный файл конфигурации из резервной копии каталога **/root/backup**.

Помните, что неизменяемый флаг был применен к скопированному файлу; удалить его из директории **/root/backup**, сначала нужно убрать неизменный флаг с помощью команды **chattr -i**. На самом деле, прежде чем будете выполнять Лабораторную работу 3, вам нужно будет выполнить следующую команду:

```
#chattr -i /root/backup/*
```

Лабораторная работа 3

Эта лабораторная работа деактивирует первое устройство **Ethernet** в системе и работает, если устройство имеет имя **eth0** по умолчанию.

Лабораторная работа 4

В этой лабораторной работе вы настраивали файл **/etc/hosts** на каждой из систем, описанных в главе 1. За исключением настройки локальной системы, добавленные администратором сети, данные в **/etc/hosts** на всех трех системах могут быть идентичны. В частности, этот файл должен содержать следующие записи:

```
192.168.122.50 server1.example.com server1
192.168.122.150 tester1.example.com tester1
192.168.100.100 outsider1.example.org outsider1
```

Неважно, что системы находятся в разных IP-сетях. Пока есть путь маршрутизации между системами данные в каждом файле **/etc/hosts** будут работать. Кроме того, дублирование с данными, вставленными в Диспетчер сети не является проблемой, если данные согласованы. На самом деле, можно настроить несколько имен для IP-адреса; например, если мы настроим веб-сервер в системе 192.168.122.50, мы можем добавить следующую запись в **/etc/hosts**:

```
192.168.122.50 www.example.com
```

Лабораторная работа 5 и 6

Обязательно изучите общие параметры, связанные с директивами в файле конфигурации, такие как

HWADDR, BOOTPROTO и DNS1.