

Глава 5

Процесс загрузки

ЦЕЛИ СЕРТИФИКАЦИИ

5.01 BIOS и UEFI

5.02. Загрузчики и GRUB 2

5.03 Между GRUB 2 и логином

5.04 Контроль за целью

5.05 Синхронизация времени

✓ Двухминутная тренировка

Q & A Самопроверка

В этой главе основное внимание уделяется тому, что происходит с момента включения системы до момента появления приглашения для входа в систему. Это называется процессом загрузки. Когда RHEL 7 установлен правильно, **BIOS/UEFI** указывает на конкретное мультимедийное устройство. Предполагая, что это локальный жесткий диск, таблица разделов **GUID (GPT)** или основная загрузочная запись (**MBR**) этого устройства указывает **на загрузчик GRUB 2**. Когда в **GRUB 2** выбран вариант загрузки RHEL 7, соответствующие команды указывают и **инициализируют ядро Linux**, которое затем **запускает systemd**, первый процесс Linux. Затем процесс **systemd** инициализирует систему и активирует соответствующие системные единицы. Когда Linux загружается в определенную цель, он запускает ряд блоков, включая клиент, связанный с протоколом сетевого времени (NTP). Вы можете настроить этот процесс.

Понимание процесса загрузки

Цели, связанные с процессом загрузки, были объединены в экзамен RHCSA. Возможно, самым основным навыком, связанным с процессом загрузки, является понимание команд, которые запускают и останавливают процесс загрузки, таких как **systemctl poweroff** и **systemctl reboot**:

- Загрузка, перезагрузка и нормальное отключение системы. (**Boot, reboot, and shut down a system normally**)

Конечно, это начинается с того, как система включена. В этой главе вы познакомитесь с **целями (targets – группа юнитов) systemd**, которые заменяют традиционные уровни выполнения в RHEL 6 и других старых дистрибутивах Linux. Из стандартного загрузочного меню RHEL 7 необходимо знать, как

- Загрузка систем в ручную с использованием различных целей (targets)

С этой целью тесно связан этот вопрос:

- Прерывание процесса загрузки, чтобы получить доступ к системе

Если вы уже знакомы с однопользовательским режимом в RHEL 6, вы должны понимать, что «доступ» в однопользовательском режиме - это доступ без пароля к корневой учетной записи администратора в ограниченной среде. Вы должны быть в состоянии достичь той же цели в RHEL 7 и получить доступ к системе для восстановления утерянного пароля **root** или устранения неполадок во время процесса загрузки.

Также тесно связана с этим задачи, сфокусированы на конфигурации различных целей (**targets**):

- Настроить системы для автоматической загрузки в заранее выбранном варианте (**targets**).

Поскольку Linux - это сетевая операционная система, и, поскольку большинство пользователей не могут многое сделать без сети, важно знать, как

- Настроить сетевые службы для автоматического запуска при загрузке

Следующая цель RHCSA строго связана с предыдущей:

- Запуск и остановка служб и настройка служб для автоматического запуска при загрузке

Сосредоточившись на процессе загрузки, вы также узнаете, как

- Модифицировать системный загрузчик

С этими целями тесно связаны и часть процесса загрузки - это цели, связанные с тем, как монтируются файловые системы, это будет описано в **главе 6**.

Служба сетевого времени

В этой главе описывается конфигурация NTP на основе следующей цели:

- Настройте систему для использования служб времени

ЦЕЛЬ СЕРТИФИКАЦИИ 5.01

BIOS и UEFI

Хотя это и не является обязательным условием или требованием к экзамену Red Hat, базовое понимание **BIOS** и **UEFI** является фундаментальным навыком для всех серьезных пользователей компьютеров. **UEFI** заменил **BIOS** во многих современных системах и может сделать намного больше. Но поскольку **UEFI** поддерживает изменения в загрузочных носителях аналогичным образом, функциональность для наших целей та же.

Из-за разнообразия программного обеспечения **BIOS/UEFI** это обсуждение носит общий характер. Невозможно предоставить какие-либо пошаговые инструкции по изменению широкого спектра доступных меню **BIOS/UEFI**. В любом случае, такие инструкции не имеют прямого отношения ни к администрированию Linux, ни к каким-либо экзаменам Red Hat. Тем не менее, эти навыки могут помочь вам загрузиться с разных установочных носителей Linux, получить доступ к параметрам виртуализации по умолчанию и многое другое.

Базовая конфигурация системы

Когда компьютер включен, первое, что выполняется, это **BIOS/UEFI**. На основе настроек, хранящихся в постоянной, доступной только для чтения памяти, система **BIOS/UEFI** выполняет серию диагностики для обнаружения и подключения ключевых контроллеров и ЦП. Это называется автоматическим самотестированием (**Power On Self Test POST**). Если во время этого процесса вы слышите звуковые сигналы, может возникнуть проблема с оборудованием, например, неправильно подключенный контроллер жесткого диска. Затем система **BIOS/UEFI** ищет подключенные устройства, такие как видеокарта.

После обнаружения графического оборудования вы можете увидеть экран, похожий на **рисунок 5-1**, который отображает другое аппаратное обеспечение, как обнаружено, проверено и подтверждено.

Если ваша система имеет меню **UEFI**, она может включать в себя доверенный платформенный модуль (**Trusted Platform Module TPM**). Хотя он создан для повышения безопасности в системе, это вызвало разногласия в сообществе с открытым исходным кодом из-за проблем с конфиденциальностью и блокировкой поставщиков. Многие профессионалы открытого исходного кода работают над тем, чтобы свести к минимуму такие проблемы через группу **Open Trusted Computing (OpenTC)** Европейского Союза. RHEL 7 использует аппаратные возможности **TPM** для повышения безопасности системы.

По завершении работы **BIOS/UEFI** передается управление устройству загрузки, обычно первому жесткому диску. Первый этап загрузчика **GRUB 2** обычно копируется в **MBR** или **GUID Partition Table (GPT)**. Он служит указателем на другую информацию из меню **GRUB 2**. В этот момент должен появиться экран системного загрузчика.

РИСУНОК 5-1 Меню инициализации BIOS

```
F2 = System Setup
F10 = Lifecycle Controller
F11 = Boot Manager
Force PXE Boot Requested via Attribute

Initializing Serial ATA devices...
Port J: PLDS DVD+/-RW DS-8A8SH

Initializing Intel(R) Boot Agent XE v2.3.27
PXE 2.1 Build 092 (WfM 2.0)

PowerEdge Expandable RAID Controller BIOS
Copyright(c) 2014 LSI Corporation
Press <Ctrl><R> to Run Configuration Utility
F/W Initializing Devices 26%
```

Меню запуска

Как правило, единственной причиной доступа к меню **BIOS/UEFI** во время экзаменов Red Hat является загрузка с разных носителей, таких как компакт-диск, дискета или USB-ключ. Во многих случаях вы можете обойти этот процесс.

Иногда все, что вы видите после **POST**, - пустой экран. **BIOS/UEFI** часто настраивается таким образом. В этом случае вам нужно будет угадать, основываясь на вашем опыте доступа к меню загрузки или **BIOS**.

Во многих случаях загрузочные меню доступны напрямую, нажимая клавишу, например, **esc**, **del**, **f1**, **f2** или **f12**. Такие загрузочные меню могут иметь записи, похожие на следующие:

Boot Menu

1. Removable Devices
2. Hard Drive
3. CD-ROM Drive
4. USB Drive
5. Built-In LAN

Из этого или похожего меню вы должны иметь возможность выбрать нужное устройство загрузки с помощью стрелки и клавиш ввода. Если это не работает, вам придется использовать меню **BIOS/UEFI**, чтобы загрузиться с нужного диска.

Доступ к загрузчикам Linux

Как отмечено в главе 2, загрузчик по умолчанию - **GRUB 2**, а первая часть его (известная как этап 1) устанавливается в таблице **MBR** или **GUID** диска по умолчанию. Обычно **BIOS** должен автоматически запускать загрузчик с сообщением, аналогичным

Red Hat Enterprise Linux Server, with Linux 3.10.0-123.el7.x86_64
Red Hat Enterprise Linux Server, with Linux 0-rescue-662ce234911596f1a75

...

The selected entry will be started automatically in 5s.

В качестве альтернативы, если вы нажмете клавишу до завершения этих пяти секунд, **GRUB** представит меню, подобное показанному на **рисунке 5-2**.

Если в систему входит более одного ядра Linux или более одной операционной системы, может быть доступно несколько вариантов, которые вы можете выделить с помощью стрелок вверх и стрелок вниз. Чтобы загрузить Linux из выделенной опции, нажмите клавишу ввода.

На старых компьютерах (до 21 века) некоторые **BIOS** не смогли найти ваш загрузчик, если он не был расположен в первых 1024 цилиндрах жесткого диска. По этой причине раздел, в котором настроен каталог **/boot**, обычно является первым доступным основным разделом. RHEL 7 поддерживает традиционный формат разметки **MBR** и новый формат таблицы **GUID (GPT)**. В то время как схема разделения **MBR** поддерживает максимальный размер **2 ТБ** на диск, **GPT** не имеет такого ограничения. Однако для загрузки **RHEL** с диска с разметкой разделов **GPT** вам нужна система с интерфейсом прошивки **UEFI**, а не обычная прошивка **BIOS**. Вы должны проверить у своего поставщика оборудования, если **UEFI** поддерживается вашей системой.

РИСУНОК 5-2 Меню GRUB



```
Red Hat Enterprise Linux Server, with Linux 3.10.0-123.el7.x86_64
Red Hat Enterprise Linux Server, with Linux 0-rescue-662ce234911596f1a75→
```

Use the ↑ and ↓ keys to change the selection.
Press 'e' to edit the selected item, or 'c' for a command prompt.

ЦЕЛЬ СЕРТИФИКАЦИИ 5.02

Загрузчики и GRUB 2

Стандартный загрузчик, связанный с **Red Hat Enterprise Linux (RHEL)**, - это **GRUB 2**, версия **GRand Unified Bootloader 2**. Как следует из требований экзамена Red Hat, для экзамена RHCSA необходимо знать, как использовать меню **GRUB 2** для загрузки в различные цели и диагностировать и исправлять сбои загрузки, возникающие из-за ошибок загрузчика. В **GRUB** версии 1, которая использовалась по умолчанию в RHEL 6, связанный файл конфигурации был относительно прост для понимания и настройки. Однако, хотя меню **GRUB 2.0** похоже на

то, что вы видели в RHEL 6, шаги, необходимые для настройки этого загрузчика, совершенно иные, как вы увидите далее в этой главе.

GRUB, GRand Unified Bootloader

Red Hat внедрила **GRUB 2** в качестве единственного загрузчика для своих дистрибутивов Linux. Обычно он настроен для загрузки в настроенное ядро по умолчанию. **GRUB 2** находит конфигурацию в каталоге **/boot** и отображает в меню, которое будет похоже на **рисунк 5-2**. Вы можете использовать меню GRUB 2 для загрузки любой операционной системы, обнаруженной в процессе установки Linux, или любой другой операционной системы, добавленной в соответствующие файлы конфигурации.

GRUB 2 является гибким. Мало того, что конфигурация может быть легко сгенерирована из CLI, но также она может быть отредактирована непосредственно из меню GRUB 2. В меню, показанном на рис. 5-2, вы можете нажать **Е**, чтобы временно отредактировать файл, или нажать **С**, чтобы открыть командную строку **GRUB 2**. Этот раздел посвящен загрузке различных системных целей (**targets**).

Загрузка в разные цели (targets)

Чтобы передать параметр в ядро через GRUB 2, нажмите **Е** в первом меню GRUB. Это позволяет редактировать параметры загрузки, отправленные в ядро. Найдите строку, которая начинается с директивы **linux16**. Прокрутите вниз с помощью клавиши со стрелкой вниз, если это необходимо. Затем вы можете увидеть строку команд, подобную следующей:

```
linux16 /vmlinuz-3.10.0-123.el7.x86_64 root=/dev/mapper/rhel-root
ro rd.lvm.lv=rhel/root vconsole.font=latarcyrheb-sun16
rd.lvm.lv=rhel/swap crashkernel=auto vconsole.keymap=uk rhgb
quiet LANG=en_GB.UTF-8
```

Да, это много вещей, которые будут объяснены в ближайшее время. Для RHCSA важно то, что вы можете добавить больше параметров ядра в конец этой строки. Например, если вы добавляете строку **systemd.unit=emergency.target** в конец этой строки и нажимаете **ctrl-x**, Linux запускается в режиме операции, называемом *аварийной целью* (**emergency target**), при котором запускается оболочка аварийного восстановления.

Из аварийной цели (**emergency target**) введите **exit**. Система перейдет к цели по умолчанию, которая обычно является многопользовательской или графической целью. Если вы внесли изменения или исправили какие-либо разделы, следующим шагом будет перезагрузка компьютера с помощью команды **systemctl reboot**. В какой-то момент изменения, сделанные во время экзамена Red Hat, должны быть протестированы с перезагрузкой.

!!!! On the Job !!!!!

В RHEL 7 команды **shutdown**, **reboot** и **halt** являются символическими ссылками на **systemctl**. Они имеют тот же эффект, что и команды **systemctl poweroff**, **systemctl reboot** и **systemctl halt** соответственно.

!!!!!!

!!!! EXAMEN Watch !!!!

Изменения должны выдержать перезагрузку во время экзамена RHCSA, поэтому вам нужно хотя бы один раз перезагрузить систему, чтобы убедиться, что каждое из требований выполнено даже после перезагрузки.

!!!!!!

В определенной степени концепция целей **systemd** аналогична концепции уровней выполнения в RHEL 6 и подробно описана далее в этой главе. Пока все, что вам нужно знать, это то, что когда RHEL 7 настроен для загрузки в графический интерфейс, он по умолчанию настроен для загрузки в графическую цель (**graphical target**). Эту цель (**target**) можно изменить, добавив строку **systemd.unit=name.target** в конец командной строки ядра.

Если вы столкнулись с проблемой загрузки системы в GUI, первое, что нужно попробовать, это добавить **systemd.unit=multi-user.target** в конец командной строки ядра. В случае успеха он загрузит RHEL 7 в текстовом режиме с помощью входа в систему из командной строки.

Если вам нужен прямой доступ к оболочке восстановления (**recovery**), добавьте строку **systemd.unit=rescue.target** в конец командной строки ядра. В редких случаях некоторые системы испытывают такие проблемы, что они не загружаются в цель спасения (**rescue target**). В этом случае доступны два других варианта:

- **systemd.unit=emergency.target** Файловая система не монтируется, кроме корневой файловой системы в режиме только для чтения.
- **init=/sysroot/bin/sh** Запускает оболочку и монтирует корневую файловую систему в режиме только для чтения; не требует пароля.

Аварийным и спасательным целям **требуется пароль root** для входа в систему и получения полных прав администратора **root**. Если вы потеряли пароль **root**, вам нужно добавить строку **init=/sysroot/bin/sh** или **rd.break** в конец командной строки ядра и выполнить процедуру, показанную в **упражнении 5-2**. Так как он поддерживает полные административные привилегии, включая изменения корневого административного пароля, важно защитить паролем меню GRUB 2. Тот, кто может изменить порядок загрузки, может добиться того же с загрузочным USB-накопителем, поэтому важно также защитить BIOS или UEFI, чтобы система загружала только локальный диск без пароля.

Теперь вы должны понимать, как загружаться в разные цели в процессе загрузки. Как определено в руководстве по подготовке к экзамену Red Hat, это явно описывается как требование RHCSA:

- Загрузите системы в разные цели (**targets**) вручную

!!!! EXAME Watch !!!!

Экзамены Red Hat являются «закрытой книгой». Несмотря на то, что вам разрешено использовать всю документацию, которую можно найти в вашей установке RHEL, во время процедур восстановления или аварийной ситуации у вас может не быть доступа к страницам руководства или другим ресурсам документации. Поэтому крайне важно, чтобы вы выполняли упражнения, описанные в этой главе, без помощи какой-либо документации. Вы должны запомнить шаги для загрузки в аварийную оболочку или для восстановления пароля **root**; в противном случае у вас могут быть проблемы не только во время экзамена RHCSA, но и в реальной жизни, когда вы выполняете свои обязанности в качестве системного администратора Linux.

!!!!

УПРАЖНЕНИЕ 5-1

Загрузка в различные цели (**target**)

Одним из ключевых навыков является умение загружаться с другой целью (**target**) **systemd**. В этом упражнении предполагается, что вы настроили **RHEL 7** в главе 2, которая устанавливает графическую цель (**target**) по умолчанию. Выполните команду **ls -l /etc/systemd/system/default.target** для проверки. Если текущая система отражает значения по

умолчанию, этот файл должен быть символической ссылкой на файл **graphical.target** в каталоге **/usr/lib/systemd/system**. В качестве альтернативы выполните следующую команду:

```
# systemctl get-default
```

Он должен вернуть строку «**graphical.target**». Теперь вы можете начать упражнение.

1. Перезагрузите систему, используя команду **reboot**.
2. Когда вы увидите следующее сообщение, обязательно нажмите любую клавишу для доступа к меню **GRUB**:

The selected entry will be started automatically in 5s.

3. Нажмите **E**, чтобы редактировать текущий пункт меню.
4. Прокрутите вниз с помощью клавиши со стрелкой вниз, чтобы найти строку, начинающуюся с **linux16**. Сначала удалите параметры ядра **rhgb quiet**. Затем в конце строки введите **systemd.unit=multi-user.target** и нажмите **ctrl-x**, чтобы загрузить это ядро.
5. Смотреть загрузочные сообщения. Какой вид экрана входа вы видите?
6. Войдите в эту систему. Вы можете использовать любую существующую учетную запись пользователя.
7. Запустите команду **reboot**, чтобы перезапустить эту систему.
8. Повторите шаги со 2 по 4, но загрузите эту систему в цель восстановления (**rescue target**), передав опцию **systemd.unit=rescue.target** ядру.
9. Смотреть загрузочные сообщения. Какой вид экрана входа вы видите? Какие файловые системы смонтированы?
10. Повторите шаги 2–4, но загрузите эту систему в аварийную цель (**emergency target**), передав опцию **systemd.unit=emergency.target** ядру.
11. Смотреть загрузочные сообщения. Какой вид экрана входа вы видите? Вы должны войти в систему вообще? Какие файловые системы смонтированы?
12. Повторите шаги со 2 по 4, но на этот раз добавьте **rd.break** в строку ядра.
13. Смотреть загрузочные сообщения. Какой вид экрана входа вы видите? Вы должны войти в систему вообще? Смонтирована ли корневая файловая система с жесткого диска?
14. Выполните **exit**, чтобы продолжить последовательность загрузки.
15. Повторите шаги 2–4, но загрузите эту систему в аварийную оболочку (**emergency shell**), передав строку **init=/sysroot/bin/sh**.
16. Смотреть загрузочные сообщения. Какой вид экрана входа вы видите?
17. Введите **reboot**, чтобы выйти из системы и перезагрузить систему.

УПРАЖНЕНИЕ 5-2

Восстановить пароль root

Если вы загружаете систему RHEL 7 в восстановление (**rescue**) или аварийную (**emergency**) цель (**target**), вам будет предложено ввести пароль **root**. Но что, если вы забыли пароль? В этом упражнении показаны шаги, необходимые для сброса утерянного пароля для пользователя **root**. В процессе восстановления пароля у вас, вероятно, не будет доступа к документации. Следовательно, вы должны практиковать следующую процедуру, пока не сможете использовать ее в условиях кризиса:

1. Используйте следующую команду, чтобы изменить пароль **root** на случайную строку. Эта команда скрывает от вас случайный пароль:

```
# pwmake 128 | passwd --stdin root
```

2. Выйдите из сеанса. Попробуйте войти снова как пользователь **root**. Вы не должны иметь возможность войти в систему со старым известным паролем **root**.
3. Перезагрузите сервер.
4. Когда вы увидите следующее сообщение, нажмите клавишу для доступа к меню GRUB:

The selected entry will be started automatically in 5s.

5. Нажмите **E**, чтобы редактировать текущий пункт меню.
6. Прокрутите вниз с помощью клавиши со стрелкой вниз, чтобы найти строку, начинающуюся с **linux16**. Нажмите **ctrl-e** или **end**, чтобы перейти к концу строки, а затем введите строку **rd.break**.
7. Нажмите **Ctrl-X** для загрузки системы.
8. Директива **rd.break** прерывает последовательность загрузки до того, как корневая файловая система будет правильно смонтирована. Подтвердите это, запустив **ls /sysroot**. Если вы знаете содержимое корневой файловой системы, вывод должен выглядеть знакомым.
9. Перемонтируйте файловую систему **root /sysroot** в режим чтения-записи и измените корневой каталог на **/sysroot**:

```
# mount -o remount,rw /sysroot
# chroot /sysroot
```

10. Изменить пароль root:

```
# passwd
```

11. Поскольку **SELinux** не работает, команда **passwd** не сохраняет контекст файла **/etc/passwd**. Чтобы убедиться, что файл **/etc/passwd** помечен в правильном контексте **SELinux**, дайте Linux команду перемаркировать все файлы при следующей загрузке с помощью следующей команды:

```
# touch /.autorelabel
```

12. Введите команду **exit**, чтобы закрыть ограничение **chroot**, а затем снова введите команду **exit**, чтобы перезагрузить систему.
13. **SELinux** может занять несколько минут, чтобы перемаркировать все файлы. Как только вы получите приглашение для входа в систему, убедитесь, что вы можете войти в систему как пользователь **root**.

Изменить системный загрузчик

RHCSA, в частности, требует, чтобы вы знали, как «модифицировать системный загрузчик». Это означает, что вам нужно уметь подробно настраивать GRUB 2. Конфигурация доступна в файле **/etc/grub2.cfg**, который является символической ссылкой, указывающей на **/boot/grub2/grub.cfg** в системах, настроенных в режиме **BIOS**, или **/boot/efi/EFI/redhat/grub.cfg** для серверов, использующих менеджер загрузки **UEFI**. В оставшейся части этой главы мы будем предполагать, что вы используете традиционную систему на основе BIOS или систему с поддержкой UEFI в режиме BIOS. Мы будем ссылаться на **/boot/grub2/grub.cfg** в качестве стандартного пути к файлу конфигурации.

Файл **grub.cfg** состоит из раздела заголовка и различных разделов меню, по одному для каждого ядра, установленного в системе. Выдержка из файла показана на **рисунке 5-3**. Каждый блок меню содержит две строки, начинающиеся с директив **linux16** и **initrd16**. процесс

загрузки. Как вы видели в предыдущем разделе, **линия linux16** особенно важна. Это запись, которую вы можете редактировать в интерактивном режиме во время процесса загрузки, чтобы передавать дополнительные параметры ядра или загружаться в цель **target systemd** не по умолчанию.

РИСУНОК 5-3. Отрывок файла grub.cfg

```
menuentry 'Red Hat Enterprise Linux Server (3.10.0-123.13.2.el7.x86_64) 7.0 (Maipo)' --class
red --class gnu-linux --class gnu --class os --unrestricted $menuentry_id_option 'gnulinux-
3.10.0-123.el7.x86_64-advanced-d055418f-1ff6-46bf-8476-b391e82a6f51' {
    load_video
    set gfxpayload=keep
    insmod gzio
    insmod part_msdos
    insmod xfs
    set root='hd0,msdos1'
    if [ x$feature_platform_search_hint = xy ]; then
        search --no-floppy --fs-uuid --set=root --hint='hd0,msdos1' 26740bbd-3aea-44b9-94
9d-c2ed4017f193
    else
        search --no-floppy --fs-uuid --set=root 26740bbd-3aea-44b9-949d-c2ed4017f193
    fi
    linux16 /vmlinuz-3.10.0-123.13.2.el7.x86_64 root=/dev/mapper/rhel-root ro rd.lvm.lv=
rhel/root vconsole.font=latarcyrheb-sun16 rd.lvm.lv=rhel/swap crashkernel=auto vconsole.key
map=uk rhgb quiet LANG=en_GB.UTF-8
    initrd16 /initramfs-3.10.0-123.13.2.el7.x86_64.img
}
menuentry 'Red Hat Enterprise Linux Server (3.10.0-123.el7.x86_64) 7.0 (Maipo)' --class red
--class gnu-linux --class gnu --class os --unrestricted $menuentry_id_option 'gnulinux-3.10.
0-123.el7.x86_64-advanced-d055418f-1ff6-46bf-8476-b391e82a6f51' {
    load_video
    set gfxpayload=keep
    insmod gzio
    insmod part_msdos
    insmod xfs
    set root='hd0,msdos1'
```

Хотя количество опций и директив в файле **grub.cfg** может показаться огромным, не паникуйте. Вам никогда не нужно трогать этот файл напрямую. Правильный подход заключается в создании новой версии этого файла с помощью инструмента **grub2-mkconfig**, основанного на файле конфигурации **/etc/default/grub** и сценарии в каталоге **/etc/grub.d/**.

Файл **/etc/default/grub** гораздо проще для понимания, безопаснее и удобнее для редактирования, чем файл **grub.cfg**. После внесения изменений в **/etc/default/grub** создайте новый файл конфигурации **GRUB**, запустив

```
# grub2-mkconfig -o /boot/grub2/grub.cfg
```

!!!! On the Job !!!!

Не редактируйте вручную файл **/etc/grub2/grub.cfg**. Этот файл автоматически генерируется, когда ядро установлено или обновлено, и в результате любые прямые настройки этого файла будут потеряны. Используйте **grub2-mkconfig** и файл **/etc/default/grub** для внесения изменений в **grub.cfg**.
!!!!

Ниже приведен подробный анализ типичной версии файла **/etc/default/grub**:

```
GRUB_TIMEOUT=5
GRUB_DISTRIBUTOR="$(sed 's, release .*$,g' /etc/system-release)"
GRUB_DEFAULT=saved
GRUB_DISABLE_SUBMENU=true
GRUB_TERMINAL_OUTPUT="console"
GRUB_CMDLINE_LINUX="rd.lvm.lv=rhel/root vconsole.font=latarcyrheb-sun16
rd.lvm.lv=rhel/swap crashkernel=auto vconsole.keymap=uk rhgb quiet"
GRUB_DISABLE_RECOVERY="true"
```

В первой строке переменная **GRUB_TIMEOUT** указывает время в секундах, прежде чем **GRUB 2** автоматически загрузит операционную систему по умолчанию. Вы можете прервать обратный отсчет, нажав любую клавишу на клавиатуре. Если для этой переменной задано значение **0**, **GRUB 2** не будет отображать список загружаемых ядер, если только вы не нажмете и не удержите буквенно-цифровую клавишу во время начального экрана **BIOS**.

Значение переменной **GRUB_DISTRIBUTOR** возвращает «**Red Hat Enterprise Linux Server**» в стандартной установке RHEL и отображается перед каждой загрузочной записью ядра. Вы можете изменить эту запись на любую строку по вашему выбору.

Следующая директива **GRUB_DEFAULT** связана с ядром по умолчанию, которое **GRUB 2** загружает при загрузке. Значение «сохранено» (**saved**) указывает **GRUB 2** искать переменную **save_entry** в файле **/boot/grub2/grubenv**. Эта переменная обновляется с именем самого последнего ядра каждый раз, когда устанавливается новое ядро.

Вы можете обновить переменную **save_entry** и дать команду **GRUB 2** загрузить другое ядро по умолчанию с помощью команды **grub2-set-default**. В качестве примера,

grub2-set-default 1

устанавливает второй пункт меню в **/etc/grub2.cfg**, как ядро по умолчанию. Это может немного сбивать с толку, потому что **GRUB 2** начинает считать с **0**. Следовательно, **grub2-set-default 0** указывает на первый доступный пункт меню в **/etc/grub2.cfg**. Аналогично, команда **grub2-set-default 1** указывает на вторую запись ядра и т. д., если она включена в файл конфигурации.

Следующая строка конфигурации в **/etc/default/grub** определяет переменную **GRUB_DISABLE_SUBMENU**. По умолчанию для этого параметра установлено значение «**true**», чтобы отключить любые записи подменю при загрузке. Затем следует директива **GRUB_TERMINAL_OUTPUT**, которая указывает **GRUB 2** использовать текстовую консоль в качестве выходного терминала по умолчанию. Последняя переменная, определенная в файле, это **GRUB_DISABLE_RECOVERY**, которая отключает создание пунктов меню восстановления.

Директива **GRUB_CMDLINE_LINUX** более интересна. Он определяет параметры для передачи в ядро Linux. Например, **rd.lvm.lv** сообщает имена логических томов, в которых находятся корневая файловая система и раздел подкачки. Следующие параметры, **vconsole.font** и **vconsole.keymap**, содержат список шрифтов по умолчанию и карту клавиатуры соответственно. Опция **crashkernel** используется для резервирования некоторой памяти для **kdump**, который вызывается для захвата дампа ядра, если система выйдет из строя. Наконец, в конце строки, директивы **rhgb quiet** включают графическую загрузку Red Hat и скрывают сообщения о загрузке по умолчанию. Если вы хотите включить вывод информации при загрузке, уберите опцию **quiet** из этой строки.

Как обновить GRUB

Если вы ранее установили другой загрузчик в **MBR**, например **Microsoft NTLDR** или **BOOTMGR**, просто запустите команду **grub2-install**. Если он не записывает автоматически указатель **GRUB 2** в **MBR** или если доступно несколько жестких дисков, вам может потребоваться включить устройство жесткого диска, например **/dev/sdb**. Также возможно установить **GRUB 2** на портативный привод; просто укажите устройство с помощью команды.

Когда файл конфигурации **GRUB 2** генерируется с использованием **grub2-mkconfig**, никаких дополнительных команд не требуется. Указатель из **MBR** автоматически считывает текущую версию файла **/boot/grub2/grub.cfg**.

GRUB 2 Командная строка

Ошибка в **grub.cfg** может привести к невозможности загрузки системы. Например, если **GRUB 2** определит неправильный том в качестве корневого раздела (**/**), **Linux** будет зависать

во время процесса загрузки. Другие ошибки конфигурации в **/boot/grub2/grub.cfg** могут привести к панике ядра во время процесса загрузки.

Теперь, когда вы проанализировали файл конфигурации **GRUB 2**, вы, вероятно, можете визуализировать некоторые последствия ошибок в этом файле. Если некоторые имена файлов или разделов неверны, **GRUB 2** не сможет найти важные файлы, такие, как **ядро Linux**. Если файл конфигурации **GRUB 2** полностью отсутствует, вы увидите приглашение, похожее на это:

```
grub>
```

Вы можете получить доступ к командной строке **GRUB 2**, нажав клавишу с при отображении меню. Чтобы просмотреть список доступных команд, нажмите клавишу табуляции в приглашении **grub>** или введите команду **help**.

Завершение команды также доступно. Например, если вы не помните имя файла ядра, введите **linux /**, а затем нажмите клавишу табуляции, чтобы просмотреть доступные файлы в **/boot** каталоге.

Вы сможете найти все обнаруженные жесткие диски на стандартном ПК из меню **BIOS/UEFI** с помощью команды **ls**. В качестве примера давайте найдем раздел **/boot** и файл **grub.cfg** в этой конкретной системе. По умолчанию каталог **/boot** монтируется в отдельный раздел. Сначала запустите **ls** в командной строке **grub>**:

```
grub> ls
(proc) (hd0) (hd0, msdos1) (hd0, msdos2)
```

Строка **hd0** обозначает первый жесткий диск, тогда как **msdos1** - это первый раздел, созданный в формате **MBR (msdos)**. Если сервер был разделен с использованием более нового формата раздела **GPT**, **GRUB 2** идентифицирует первый раздел, как **gpt1**, а не как **msdos1**. Аналогично, **hd0, msdos2** обозначает второй раздел на первом жестком диске.

Затем используйте эту информацию, чтобы найти файл **grub.cfg**:

```
grub> ls (hd0, msdos1) /grub2/grub.cfg
grub.cfg
```

Если файл не находится в указанном разделе, вы увидите сообщение об ошибке **“error: file '/grub2/grub.cfg' not found”** error «ошибка: файл» **/grub2/grub.cfg** «не найден». Вы также можете увидеть **“error: unknown filesystem”** «ошибка: неизвестная файловая система», если указанный раздел не содержит допустимой файловой системы.

Мы знаем, что каталог **/boot** включен (**hd0, msdos1**). Чтобы подтвердить местоположение **grub.cfg**, выполните следующую команду:

```
grub> cat (hd0, msdos1) /grub2/grub.cfg
```

Вы должны увидеть содержимое файла **grub.cfg** в выводе. Нажмите клавишу, чтобы прокрутить содержимое файла, пока не вернетесь к командной строке **GRUB 2**.

Есть еще один способ идентифицировать раздел с помощью каталога **/boot**. Запустить команда **search.file** для поиска **grub.cfg**:

```
grub> search.file /grub2/grub.cfg
```

GRUB 2 должен вернуть раздел с каталогом **/boot**. В данном случае это первый раздел на первом жестком диске:

```
hd0, msdos1
```

Теперь вы можете использовать эти команды из файла конфигурации GRUB 2 для загрузки **Linux** из командной строки **grub>**. Если корневой каталог верхнего уровня обычно монтируется в разделе, вы можете даже подтвердить содержимое файла **/etc/fstab** с помощью команды, подобной следующей:

```
grub> cat (hd0,msdos2)/etc/fstab
```

Если корневая файловая система находится на томе **LVM**, предыдущая команда вернет сообщение «ошибка: неизвестная файловая система». Чтобы решить эту проблему, загрузите модуль **LVM** с помощью следующей команды:

```
grub> insmod lvm
```

Теперь команда **ls** также должна включать в свой вывод логические тома:

```
grub> ls  
(proc) (hd0) (hd0,msdos2) (hd0,msdos1) (lvm/rhel-root) (lvm/rhel-swap)
```

Наконец, чтобы распечатать содержимое **/etc/fstab**, выполните следующую команду:

```
grub> cat (lvm/rhel-root)/etc/fstab
```

УПРАЖНЕНИЕ 5-3

Использование командной строки GRUB 2

В этом упражнении вы загрузите RHEL 7 вручную. Посмотрите на содержимое **/etc/grub2.cfg** и определите нужные команды в разделе. Теперь выполните следующие действия:

1. Загрузите систему. Когда вы увидите следующую строку в верхней части экрана, нажмите любую клавишу для доступа к меню **GRUB 2**:

The selected entry will be started automatically in 5s.

2. Нажмите **C** для интерфейса командной строки на основе **GRUB**. Вы должны увидеть приглашение **grub>**.
3. Загрузите модуль **LVM**, введя следующую команду:

```
grub> insmod lvm
```

4. Перечислите все разделы и логические тома:

```
grub> ls
```

5. Определите корневой раздел. Это может быть названо что-то вроде **(lvm/rhel-root)**. Вам может понадобиться использовать метод проб и ошибок, чтобы выяснить это (например, пытаясь отобразить файл **/etc/fstab** со всех имен устройств, ранее перечисленных **GRUB 2**).

```
grub> cat (lvm/rhel-root) /etc/fstab
```

6. Установите корневую переменную для устройства, которое вы определили как устройство, содержащее корневую файловую систему:

```
grub> set root=(lvm/rhel-root)
```

7. Введите команду **linux**, которая указывает ядро и корневой каталог. Да, это длинная очередь; однако, вы можете использовать завершение команды (нажмите клавишу табуляции), чтобы сделать это быстрее. Кроме того, единственными важными частями строки являются файл ядра и расположение корневого каталога верхнего уровня.

```
linux (hd0,msdos1)/vmlinuz-3.10.0-123.el7.x86_64 root=/dev/mapper/rhel-root
```

8. Введите команду **initrd**, которая указывает начальную команду **RAM**-диска и местоположение файла. Опять же, вы можете использовать клавишу табуляции для завершения имени файла.

```
initrd (hd0,msdos1)/initramfs-3.10.0-123.el7.x86_64.img
```

9. Теперь введите команду загрузки. Если эта команда выполнена успешно, **Linux** теперь должен загрузить выбранное ядро и начальный **RAM**-диск, как если бы вы выбрали эту опцию в меню конфигурации **GRUB 2**.

Переустановите GRUB 2

В некоторых ситуациях вам может потребоваться переустановить GRUB 2 с нуля. Это может произойти, если **grub2-mkconfig** не работает, или если генерируемый файл конфигурации содержит ошибки из-за поврежденного или неправильного файла скрипта. В этом случае вам необходимо переустановить **RPM-пакет grub2-tools**. Прежде чем продолжить эту операцию, перечислите и удалите все конфигурационные файлы **GRUB 2** и файлы сценариев. Это можно сделать с помощью следующих команд:

```
# rpm -qc grub2-tools
/etc/default/grub
/etc/grub.d/00_header
/etc/grub.d/10_linux
/etc/grub.d/20_linux_xen
/etc/grub.d/20_ppc_terminfo
/etc/grub.d/30_os-prober
/etc/grub.d/40_custom
/etc/grub.d/41_custom
# rm -f /etc/default/grub
# rm -f /etc/grub.d/*
```

Затем переустановите **GRUB 2**, выполнив следующую команду:

```
# yum reinstall grub2-tools
```

(Полное введение в команды **rpm** и **yum** вы найдете в главе 7.)

Наконец, заново создайте файл конфигурации **grub.cfg**. На машинах, на которых выполняется традиционная прошивка BIOS, команда **grub2-mkconfig** будет выглядеть так:

```
# grub2-mkconfig -o /boot/grub2/grub.cfg
```

Конечно, если файл конфигурации **GRUB 2** отсутствует, и вы не смогли загрузить систему, чтобы даже отобразить меню **GRUB 2**, вам, возможно, придется прибегнуть к опции, известной как режим восстановления.

Опция для загрузки из GRUB 2: Rescue Mode

Задачи устранения неполадок, связанные с предыдущей версией руководства по подготовке к экзамену **RHCE**, показали, что вам необходимо иметь возможность восстановиться после полного сбоя загрузки, например, если файл конфигурации **GRUB 2** был поврежден или отсутствует. Другими словами, если вы попытались загрузиться непосредственно из приглашения **grub>**, описанного ранее, и потерпели неудачу, вам может потребоваться использовать опцию, известную как режим восстановления (**rescue mode**). Для этого требуется доступ к установочному DVD или сетевому загрузочному диску.

!!!! EXAM Watch !!!!!

Задачи **RHCSA** и **RHCE** больше не включают требования, связанные с режимом спасения. Однако, поскольку спасение не загружаемых систем является важным навыком, его можно включить в будущие версии одного из этих экзаменов.
!!!!

Для этого загрузитесь с одного из этих параметров мультимедиа. Вы должны увидеть экран установки со следующими параметрами:

Install Red Hat Enterprise Linux 7.0

Test this media & install Red Hat Enterprise Linux 7.0

Troubleshooting

Выберите вариант устранения неполадок и нажмите ввод. Вы увидите второй экран со следующими параметрами:

Install Red Hat Enterprise Linux 7.0 in basic graphics mode

Rescue a Red Hat Enterprise Linux system

Run a memory test

Boot from local drive

Return to main menu

Выберите опцию **Rescue a Red Hat Enterprise Linux** и нажмите enter. В режиме восстановления работает стабильная минимальная версия операционной системы RHEL 7 на локальном компьютере. По сути, это текстовая версия носителя «Live DVD», доступная в других дистрибутивах Linux, таких как Knoppix, Ubuntu и, да, даже в дистрибутиве Scientific Linux для восстановления сборки.

!!!! On the Job !!!!!

Для **RHEL 7** лучше всего использовать **RHEL 7 Rescue Media**. На таких носителях используется ядро, скомпилированное Red Hat, настроенное для поддерживаемого программного обеспечения. Тем не менее, такие опции, как Knoppix, превосходны.
!!!!

Вы можете использовать спасательную среду для восстановления незагружаемых систем. Если вы использовали режим восстановления в RHEL 6, вы должны чувствовать себя здесь комфортно. В большинстве случаев следующий шаг, который вы видите, показан на рисунке 5-4.

Опция **Continue**, как показано на рис. 5-5, монтирует все обнаруженные тома в качестве подкаталогов в каталоге **/mnt/sysimage**. Опция Только для чтения (**Read-Only**) монтирует обнаруженные тома в режиме только для чтения (**Read-Only**). Опция Пропустить (**Skip**) перемещается прямо к интерфейсу командной строки. Выберите Продолжить (**Continue**). После подтверждения вы увидите приглашение оболочки, как показано на рисунке 5-6.

РИСУНОК 5-4 Варианты аварийной среды (rescue environment)

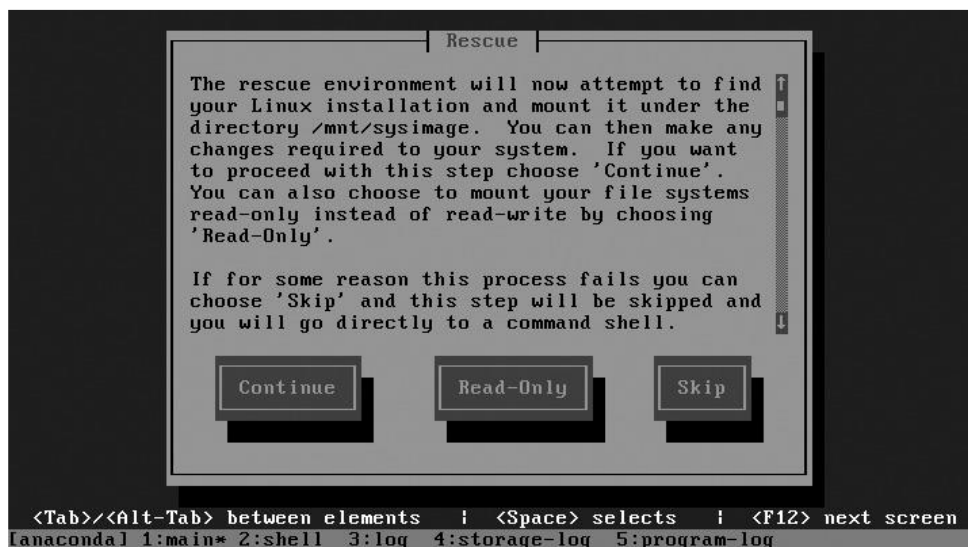


РИСУНОК 5-5. Монтирование корневой файловой системы в среде аварийного восстановления. (rescue environment)

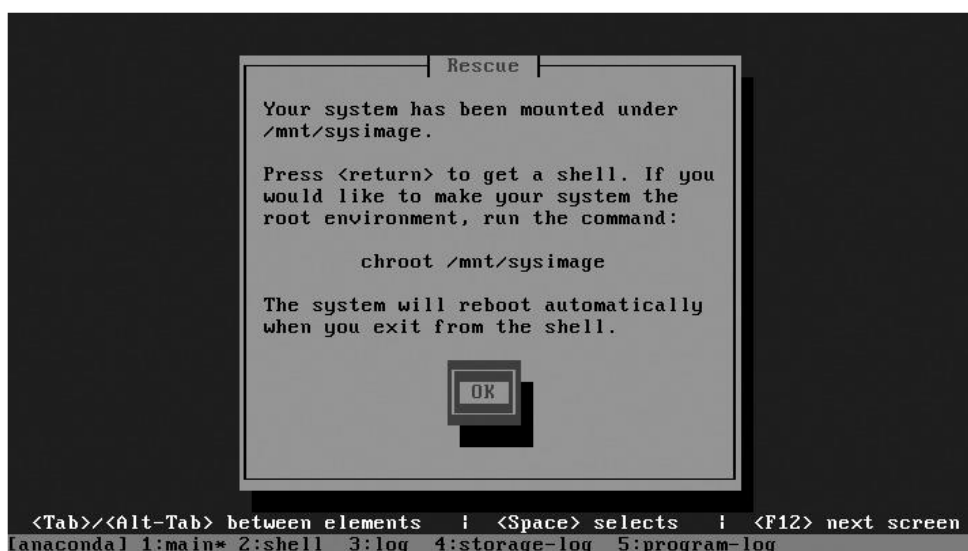


РИСУНОК 5-6 Оболочка спасательной среды

```
Starting installer, one moment...
anaconda 19.31.79-1 for Red Hat Enterprise Linux 7.0 started.

Your system is mounted under the /mnt/sysimage directory.
When finished please exit from the shell and your system will reboot.

sh-4.2#
```

```
[anaconda] 1:main* 2:shell 3:log 4:storage-log 5:program-log
```

В командной строке введите команду **chroot /mnt/sysimage**. Поскольку обычный корневой каталог верхнего уровня для системы смонтирован в каталоге **/mnt/sysimage**, команда **chroot** изменяет корневой каталог, как если бы файловая система **/mnt/sysimage** была смонтирована в **/**.

Практикуйте то, что вы узнали о **GRUB 2** в этом разделе. Это может помочь вам справиться с реальной проблемой - и Red Hat действительно говорит, что их экзамены заполнены «реальными задачами». Однако не думайте, что у вас есть доступ к CD или DVD во время Red Hat. экзамен. Если аварийный носитель недоступен, это должно означать, что существует, как минимум один альтернативный метод, который можно использовать для решения проблемы.

ЦЕЛЬ СЕРТИФИКАЦИИ 5.03

Между GRUB 2 и логином

В этом разделе представлен общий обзор процесса загрузки после того, как загрузчик GRUB 2 обнаружит ядро. Если вы понимаете этот процесс, вы можете диагностировать самые разные проблемы с загрузкой. Сообщения, связанные с ядром, обеспечивают пошаговое представление процесса.

Загрузка Linux зависит от временной файловой системы, известной как начальный **(initial) RAM-диск**. После завершения процесса загрузки управление переходит к **systemd**, известному, как **первый процесс**. В этом разделе будет подробно описано содержимое **systemd** через конфигурацию модулей и целей (**targets**).

!!!! On the Job !!!!

Большинство дистрибутивов Linux, включая RHEL 7, заменили Upstart и SysVinit новым диспетчером службы **systemd**.
!!!!

В этом разделе вы также рассмотрите команды, которые позволяют вам перезагрузить (**reboot**) компьютер и завершить работу (**shut down**) системы в обычном режиме.

!!!! On the Job !!!!

В **systemd** философию Unix, согласно которой «все является файлом» можно перефразировать как «все является единицей (**unit**)». Единицы (**units**) - это основные строительные блоки **systemd**.
!!!!

Ядра (Kernels) и начальный (Initial)RAM-диск

После выбора ядра в меню конфигурации **GRUB 2 Linux** передает обязанности по загрузке ядру с помощью **начального RAM-диска**, также известного по имени файла в каталоге **/boot, initramfs**. Как следует из его названия, это на самом деле файловая система.

В процессе загрузки Linux загружает эту временную файловую систему в вашу оперативную память. Затем Linux загружает драйверы оборудования и **запускает первый процесс, systemd**.

Затем **systemd** активирует все системные блоки для **initrd.target** и монтирует корневую файловую систему в **/sysroot**. Наконец, **systemd** перезапускает себя в новом корневом каталоге и активирует все юниты (**units**) для цели (**targets**) по умолчанию (мы рассмотрим более подробно юниты и цели в следующем разделе).

Чтобы узнать больше, отключите директиву **quiet** для нужного ядра в файле конфигурации **GRUB**. Загрузите вашу систему. Смотрите, как сообщения быстро проходят через экран. После входа в систему вы можете просмотреть эти сообщения в файле **/var/log/dmesg** или с помощью команды **dmesg**.

Вы можете найти больше информации в журнале **systemd**. Отобразите его содержимое с помощью команды **journalctl**. То, что вы видите, зависит от оборудования и конфигурации локальной системы. Ключевые сообщения включают в себя следующее:

- Версия ядра.
- Статус **SELinux**, если активен. По умолчанию **SELinux** сначала запускается в разрешающем режиме, пока настроенная политика (принудительное применение) не будет загружена в конце процесса загрузки.
- Объем распознанной оперативной памяти (который не обязательно соответствует фактическому объему установленной оперативной памяти).
- Процессоры.
- Командная строка ядра, указывающая логический том или корневую файловую систему.
- Освобождение памяти, связанной с исходным **RAM-диском (initramfs)**.
- Жесткие диски и разделы (в соответствии с именами их устройств, такими, как **/dev/sda** или **/dev/vda1**).
- Активные файловые системы.
- **Swap** разделы.

Файл журнала заполнен отличной информацией. Если система загружает не то ядро, вы увидите это здесь. Если **Linux** не использует настроенный вами раздел, вы также увидите его здесь (косвенно). Если **SELinux** не загружается должным образом, вы увидите его в сообщениях в конце файла.

!!!! Exam Watch !!!!!

Помните, что экзамены Red Hat - это не экзамены на оборудование. Если вы обнаружили проблему с ключевым аппаратным компонентом, таким как сетевая карта (которая не может быть решена какой-либо командой **Linux**), сообщите об этом своему инструктору/проктору экзамена. Однако не удивляйтесь, если она ответит, что это не аппаратная проблема.

!!!!

Первый процесс, цели (targets) и Units

Ядро Linux продолжает процесс загрузки, вызывая первый процесс, **systemd**. В RHEL 7 унаследованный процесс инициализации сконфигурирован с символической ссылкой на **systemd**.

Единицы (**units**) являются основными строительными блоками **systemd**. Наиболее распространенными являются сервисные единицы (**service units**), которые имеют расширение **.service** и активируют системный сервис. Чтобы показать список всех сервисных единиц, введите следующую команду:

```
# systemctl list-units --type=service --all
```

Флаг **--all** включает в себя все юниты, а не только активные. Существуют другие типы модулей, такие, как модули монтирования и автмонтирования, которые управляют точками монтирования; модули пути, которые активируют службу при изменении пути в файловой системе (например, в **каталоге spool**); блоки сокетов, которые запускают службу только при установлении клиентского соединения (если вы использовали демон **xinetd**, это похоже на то, как **xinetd** запускает службы по требованию); и многое, многое другое.

Единица (**unit**) специального типа - это **target unit**, которая используется для группировки других системных единиц и для перевода системы в другое состояние. Чтобы вывести список всех целевых (**targets**) объектов, введите следующую команду:

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

Наиболее важные **target unit** описаны в таблице 5-1.

ТАБЛИЦА 5-1 The systemd Target Units

Target Unit	Описание
emergency.target	Emergency(Аварийный) shell ; только / файловая система монтируется в режиме только для чтения.
graphical.target	Цель по умолчанию для многопользовательских графических систем.
multi-user.target	Неграфическая многопользовательская система.
rescue.target	Emergency(Аварийный) shell ; все файловые системы смонтированы.

В **systemd targets** выполняют ту же функцию, что и уровни выполнения в предыдущих дистрибутивах RHEL. В RHEL 6 было доступно семь уровней запуска, от 0 до 6. Службы Linux были организованы на уровне запуска. Каждый уровень выполнения был связан с уровнем функциональности.

Например, на уровне выполнения 1 только одному пользователю было разрешено войти в систему Linux. Режим X11, также известный как уровень запуска 5, использовался для запуска Linux с экраном входа в систему с графическим интерфейсом, если были установлены соответствующие пакеты. Таблица 5-2 сравнивает системные цели (**targets**) и уровни выполнения, определенные в RHEL 6.

Запустите следующую команду:

```
# ls -l /usr/lib/systemd/system/runlevel?.target
```

Обратите внимание на символические ссылки в выводе. Посмотрите, как файлы, такие как **runlevel0.target**, **runlevel1.target** и т. д. связаны с системными целями (**targets**), такими как **poweroff.target** и **rescue.target**. Эти ссылки обеспечивают обратную совместимость со старыми уровнями выполнения SysV. Вы даже можете сослаться на **graphical.target**, как **runlevel5.target** и **multi-user.target**, как **runlevel3.target**.

Цели (**targets**) контролируются единицами (**units**), организованными в файлы **units**. Хотя **targets** по умолчанию определена в **/etc/systemd/system**, вы можете переопределить настройку по умолчанию во время процесса загрузки из меню **GRUB 2**.

ТАБЛИЦА 5-2 RHEL 6 Runlevels and RHEL 7 systemd Targets

Runlevel	Systemd Target	Описание
0	poweroff.target	Остановить систему
1	rescue.target	Однопользовательский режим для обслуживания и ремонта
2	multi-user.target	Многопользовательский, без NFS
3	multi-user.target	Полный многопользовательский режим
4	multi-user.target	Не используется в RHEL 6
5	graphical.target	X11 GUI с поддержкой сети
6	reboot.target	Перезагрузите систему

Каждая цель (**targets**) может быть связана с несколькими системными единицами (**systemd units**). Каждый модуль может запускать или останавливать службы Linux, такие как печать (**cupsd**), планирование (**crond**), веб-сервер Apache (**httpd**), файловый сервер **Samba** (**smbd**) и другие. После настройки процесс загрузки запускается и останавливает системные модули по вашему выбору. Эти единицы (**units**) известны как зависимости. Чтобы получить

список всех зависимостей модуля по умолчанию **graphical.target**, выполните следующую команду:

```
# systemctl list-dependencies graphical.target
```

Targets по умолчанию указывается в виде символической ссылки из файла `/etc/systemd/system/default.target` на **multi-user.target** или **graphical.target**. Вы также можете использовать команду **systemctl** для получения текущего значения **targets** по умолчанию или для изменения текущих настроек, как показано здесь:

```
# systemctl get-default  
graphical.target  
# systemctl set-default multi-user.target  
rm '/etc/systemd/system/default.target'  
ln -s '/usr/lib/systemd/system/multi-user.target' '/etc/systemd/system/default.target'
```

Как видно из этого вывода, команда **systemctl set-default multi-user.target** создает символическую ссылку из `/etc/systemd/system/default.target`.

Переключение между Targets

Теперь, когда вы изучили различные **targets**, доступные в RHEL 7, пришло время изучить, как переключаться между ними. В более ранних версиях RHEL это функционально эквивалентно переключению уровней выполнения. Сначала установите **target** по умолчанию с помощью следующей команды:

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

RHEL 7 обычно загружается либо в **graphical.target**, либо в **multi-user.target**. После входа в систему от имени администратора вы можете перейти к другой цели с помощью команды **systemctl isolate**. Например, следующая команда перемещает систему к многопользовательской **target**:

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

После завершения этой команды снова запустите команду **systemctl get-default**. Вывод подтверждает, что цель по умолчанию не изменилась:

```
graphical.target
```

Теперь попробуйте что-нибудь еще. Как вы думаете, что происходит при выполнении следующей команды?

```
# systemctl isolate poweroff.target
```

Перезагрузка и выключение системы в обычном режиме

Команды, необходимые для перезагрузки и выключения системы, просты. Как только что предлагалось в предыдущем разделе, следующие команды предоставляют один из способов выключения и перезагрузки системы, соответственно:

```
# systemctl poweroff  
# systemctl reboot
```

В целях наследования Red Hat создал символические ссылки из следующих команд на **systemctl**. Эти команды работают так же, как и в предыдущих версиях RHEL.

```
# shutdown
# reboot
```

systemd заменяет Upstart и SysVinit

Процесс **systemd** - это первый процесс, запускаемый при загрузке. Он берет на себя ответственность за активацию всех услуг. Он заменяет традиционный демон **init** и систему **Upstart**, которая также заменяет **init** и была демоном **init** по умолчанию в RHEL 6. Дизайн и философия **Upstart** очень похожи на старую систему **SysVinit**, которая для активации использует сценарии **init**. сервисы, и о концепции уровней выполнения, которая была представлена в предыдущих разделах.

Напротив, **systemd** представляет много новых инструментов и может делать гораздо больше, сохраняя совместимость с **SysVinit**. Дизайн **systemd** основан на оптимальной эффективности. Во-первых, при загрузке **systemd** активирует только те службы, которые строго необходимы, тогда как другие запускаются по требованию. Например, **systemd** запускает службу печати **CUPS** только тогда, когда задание печати отправляется в очередь **/var/spool/cups**. Кроме того, **systemd** распараллеливает инициализацию сервисов.

В результате процесс загрузки под **systemd** происходит быстрее. Чтобы отобразить время, необходимое для загрузки вашей системы, выполните следующую команду:

```
# systemd-analyze time
Startup finished in 506ms (kernel) + 1.144s
(initrd) + 6.441s (userspace) = 8.092s.
```

Эти выходные данные показывают время, необходимое для инициализации ядра, плюс время для загрузки исходного **RAM-диска (initrd)** и время для активации модулей **systemd** (пользовательское пространство). Общее время составляет 8,092 секунды. Но это еще не все. Вы можете отобразить подробный отчет о времени, необходимом для активации каждого модуля **systemd**, выполнив **systemd-analyze blame**. Пример показан на **рисунке 5-7**.

Числа на **Рисунке 5-7** не равны общему времени пространства пользователя, указанному **systemd-analyze time**. Это происходит потому, что **systemd** запускает несколько сервисов одновременно.

РИСУНОК 5-7 Время инициализации системных модулей

```
[root@server1 ~]# systemd-analyze blame
4.604s kdump.service
2.610s postfix.service
2.333s firewalld.service
1.629s tuned.service
1.370s network.service
1.365s plymouth-quit-wait.service
953ms iprupdate.service
845ms accounts-daemon.service
729ms avahi-daemon.service
699ms iprinit.service
666ms ModemManager.service
638ms systemd-logind.service
596ms lvm2-monitor.service
568ms rsyslog.service
560ms rtkit-daemon.service
536ms nfs-lock.service
520ms iprdump.service
482ms NetworkManager.service
477ms libvirtd.service
442ms gdm.service
412ms chronyd.service
338ms ksmtuned.service
324ms netcf-transaction.service
```

Но это еще не все. Хотя глубокое знание всех функций **systemd** выходит за рамки экзамена RHCSA, системные администраторы могут воспользоваться некоторыми из его возможностей.

Некоторые разработчики Linux утверждают, что **systemd** делает слишком много и нарушает философию Unix по написанию программ, которые «делают одно и делают это хорошо». Однако на сегодняшний день **systemd** был принят большинством основных дистрибутивов Linux.

Логирование

Процесс **systemd** включает в себя мощную систему регистрации. Вы можете отобразить все собранные журналы с помощью команды **journalctl**. По умолчанию журнальные лог-файлы временно хранятся в оперативной памяти в кольцевом буфере в каталоге **/run/log/journal**. Чтобы заставить Linux постоянно записывать лог-файлы журнала на диск, выполните следующие команды:

```
# mkdir /var/log/journal
# chgrp systemd-journal /var/log/journal
# chmod 2775 /var/log/journal
# systemctl restart systemd-journald.service
```

Как только постоянное ведение журнала включено, вы можете отображать сообщения журнала от конкретной загрузки с помощью ключа **-b**: **journalctl -b 0** отображает сообщения журнала с момента последней загрузки, **journalctl -b 1** от загрузки до последней и т. д. Вам не нужно переключаться между различными файлами журнала, потому что **journalctl** автоматически объединяет доступные данные из текущего и всех ротэйт файлов журнала.

Вы также можете фильтровать сообщения журнала по их приоритету, используя опцию **-p** в команде. Например, **journalctl -p warning** отображает все сообщения с уровнем приоритета «**warning**» или выше. Сообщения журнала с уровнем приоритета «**warning**» отображаются жирным шрифтом, тогда как сообщения с уровнями приоритета «**err**» и выше отображаются красным цветом.

Контрольные группы (cgroups)

Контрольные группы (или **cgroups**) - это функция ядра Linux, позволяющая группировать процессы и контролировать или ограничивать использование их ресурсов (таких как ЦП, память и т. д.). В **systemd** **cgroups** в основном используются для отслеживания процессов и обеспечения того, чтобы все процессы, принадлежащие службе, были остановлены при остановке службы.

В традиционной системе **SysVinit** сложно идентифицировать службу, связанную с процессом. На самом деле, сервисы часто запускают несколько процессов. При остановке службы **SysVinit** эта служба не сможет завершить все зависимые (дочерние) процессы. Вы застряли либо с остановкой всех зависимых процессов вручную (с помощью команд **ps** и **kill**), либо с принятием системы с потерянными процессами в неизвестном состоянии до следующей перезагрузки.

Чтобы устранить это ограничение, **systemd** помечает процессы, связанные со службой, с помощью **cgroups**. Таким образом, **systemd** использует **cgroups** для уничтожения всех процессов в группе, если это необходимо.

Команда **systemd-cgls** отображает иерархию **cgroup** в древовидном формате, как показано на рисунке 5-8. Из выдержки на рис. 5-8 вы можете идентифицировать **cgroups**, такие как **rsyslog.service** и **avahi-daemon.service**, а также процессы, которые они породили. Обратите внимание на однозначное соответствие между **cgroups** и **systemd service units**.

РИСУНОК 5-8 Иерархия cgroup

```

├─1673 pickup -l -t unix -u
├─1674 qmgr -l -t unix -u
├─rsyslog.service
├─633 /usr/sbin/rsyslogd -n
├─rhsmcertd.service
├─1283 /usr/bin/rhsmcertd
├─NetworkManager.service
├─828 /usr/sbin/NetworkManager --no-daemon
├─avahi-daemon.service
├─632 avahi-daemon: running [server1.local]
├─643 avahi-daemon: chroot helpe
├─crond.service
├─698 /usr/sbin/crond -n
├─pcscd.service
├─1324 /usr/sbin/pcscd --foreground --auto-exit
├─dbus.service
├─683 /bin/dbus-daemon --system --address=systemd: --nofork --nopidfile --sy
├─firewalld.service
├─629 /usr/bin/python -Es /usr/sbin/firewalld --nofork --nopid
├─iprdump.service
├─746 /sbin/iprdump --daemon
├─iprupdate.service
├─714 /sbin/iprupdate --daemon
lines 120-142/142 (END)

```

РИСУНОК 5-9 Зависимости между systemd units

```

default.target
├─accounts-daemon.service
├─gdm.service
├─iprdump.service
├─iprinit.service
├─iprupdate.service
├─network.service
├─rhnsd.service
├─rtkit-daemon.service
├─systemd-readahead-collect.service
├─systemd-readahead-replay.service
├─systemd-update-utmp-runlevel.service
├─multi-user.target
├─abrt-ccpp.service
├─abrt-oops.service
├─abrt-vmcore.service
├─abrt-xorg.service
├─abrt-d.service
├─atd.service
├─auditd.service
├─avahi-daemon.service
├─brandbot.path
├─chronyd.service
lines 1-23

```

Зависимости

Традиционная система **SysVinit** запускает сервисы последовательно. Напротив, **systemd** может активировать сервисы параллельно, отслеживая все зависимости между модулями. Команда **systemctl list-dependencies** отображает дерево со всеми зависимостями между модулями. Выдержка из вывода показана на рисунке 5-9.

Вы можете показать зависимости для любого доступного модуля. Зависимые блоки должны быть запущены в первую очередь. Например, следующая команда показывает модули, которые должны быть запущены до службы **rsyslog**:

```
# systemctl list-dependencies rsyslog.service
```

Системные Units

Первый процесс **systemd**. Процесс **systemd** использует различные файлы конфигурации для запуска других процессов. Вы можете найти эти файлы конфигурации в следующих каталогах:

/etc/systemd/system и **/usr/lib/systemd/system**.

Файлы конфигурации по умолчанию хранятся в **/usr/lib/systemd/system**. Пользовательские файлы, хранящиеся в **/etc/systemd/system**, заменяют эти файлы. Не меняйте файлы в **/usr/lib/systemd**

Каталог **/system**. Любые обновления программного обеспечения могут перезаписывать эти файлы. Мы уже обсуждали сервисные и целевые подразделения, но есть и другие.

Таблица 5-3 дает краткое описание всех доступных типов единиц.

Изучите содержимое каталога **/usr/lib/systemd/system**. Каждый файл содержит конфигурацию модуля **systemd**, тип которого соответствует расширению имени файла. Например, файл **graphical.target** определяет конфигурацию для целевого модуля графического входа в систему, тогда как файл **rsyslog.service** включает в себя конфигурацию **unit** обслуживания **rsyslog**.

ТАБЛИЦА 5-3 Системный Типы объектов (Unit Types)

Тип Unit	Описание
Target	Группа Units . Он используется в качестве точки синхронизации при запуске для определения набора Units , которые должны быть активированы.
Service	Service , такой, как демон, веб-сервер Apache .
Socket	IPС или сетевой сокет, используемый для активации службы при получении трафика через прослушивающий сокет (аналогично активации служб по требованию, выполняемой демоном xinetd).
Device	Блок устройства, например диск или раздел.
Mount	Точка монтирования файловой системы, управляемая systemd .
Automount	Точка автомонтирования файловой системы, управляемая systemd .
Swap	Раздел подкачки, который должен быть активирован systemd .
Path	Путь (Path), отслеживаемый systemd , используется для активации службы при изменении пути.
Timer	Таймер, управляемый systemd , используется для активации службы по истечении таймера.
Snapshot	Используется для создания снимка состояния времени выполнения systemd .
Slice	Группа системных ресурсов (таких как процессор, память и т. д.), Которые могут быть назначены устройству через интерфейс cgroup .
Scope	Unit для организации и управления использованием ресурсов набора системных процессов.

Вы можете перечислить все активные модули **systemd**, используя следующую команду:

```
# systemctl list-unit
```

Ключевое слово **list-units** является необязательным, поскольку оно используется по умолчанию. Если вы хотите включить неактивные, обслуживаемые и неисправные модули, добавьте ключ команды **--all**. Выдержка из вывода команды показана на **рисунке 5-10**.

В выходных данных в первом столбце указано имя устройства, а во втором столбце указано, правильно ли загружен модуль. В третьем столбце отображается состояние устройства: активен, неактивен, неисправен или обслуживается. Следующий столбец содержит более подробную информацию. Наконец, последний столбец показывает краткое описание устройства.

Принимая во внимание, что команда **systemctl list-units** дает во время выполнения снимок состояния каждого модуля, следующая команда показывает, включен ли модуль при запуске:

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

Пример выходных данных показан на **рисунке 5-11**. Как видите, юниты могут быть «включены» (**enabled**) или «отключены» (**disabled**). Существует также другое состояние, называемое «статическим» (**static**), что означает, что юнит включен, и его нельзя отключить вручную.

РИСУНОК 5-10 systemd units

```
systemd-as-...sword-plymouth.path loaded active waiting Forward Password Requests to Plym
systemd-ask-password-wall.path loaded active waiting Forward Password Requests to Wall
session-4.scope loaded active running Session 4 of user alex
abrt-ccpp.service loaded active exited Install ABRT coredump hook
abrt-oops.service loaded active running ABRT kernel log watcher
abrt-vmcore.service loaded inactive dead Harvest vmcores for ABRT
abrt-xorg.service loaded active running ABRT Xorg log watcher
abrttd.service loaded active running ABRT Automated Bug Reporting Tool
accounts-daemon.service loaded active running Accounts Service
alsa-restore.service loaded inactive dead Restore Sound Card State
alsa-state.service loaded active running Manage Sound Card State (restore
alsa-store.service loaded inactive dead Store Sound Card State
atd.service loaded active running Job spooling tools
auditd.service loaded active running Security Auditing Service
avahi-daemon.service loaded active running Avahi mDNS/DNS-SD Stack
bluetooth.service loaded active running Bluetooth service
brandbot.service loaded inactive dead Flexible Branding Service
chronyd.service loaded active running NTP client/server
colord.service loaded active running Manage, Install and Generate Colo
cpupower.service loaded inactive dead Configure CPU power related setti
crond.service loaded active running Command Scheduler
cups.service loaded active running CUPS Printing Service
dbus.service loaded active running D-Bus System Message Bus
dm-event.service loaded inactive dead Device-mapper event daemon
lines 53-76/250 30%
```

РИСУНОК 5-11 Установленные файлы units

```
dbus-org.freedesktop.timedatел.service static
dbus.service static
debug-shell.service disabled
display-manager.service enabled
dm-event.service disabled
dmraid-activation.service enabled
dnsmasq.service disabled
dracut-cmdline.service static
dracut-initqueue.service static
dracut-mount.service static
dracut-pre-mount.service static
dracut-pre-pivot.service static
dracut-pre-trigger.service static
dracut-pre-udev.service static
dracut-shutdown.service static
ebtables.service disabled
emergency.service static
fcoe.service disabled
firewalld.service enabled
firstboot-graphical.service disabled
fprintd.service static
gdm.service enabled
getty@.service enabled
halt-local.service static
lines 66-89
```

Виртуальные терминалы и экраны входа

Терминалы входа в Linux являются виртуальными терминалами. Большинство систем Linux, включая RHEL 7, сконфигурированы с шестью стандартными виртуальными терминалами командной строки. Эти консоли пронумерованы от 1 до 6. При настройке с графическим интерфейсом и менеджером входа RHEL 7 заменяет графический экран входа для первого виртуального терминала.

Что это все значит? В Linux вы можете переключаться между виртуальными терминалами с помощью комбинация клавиш **Alt-Function**. Например, **alt-f2** приводит вас ко второму виртуальному терминалу. Вы можете переключаться между соседними виртуальными терминалами, нажимая **Alt-стрелка вправо** или **Alt-стрелка влево**. Например, чтобы перейти от виртуального терминала 2 к виртуальному терминалу 3, нажмите **Alt-стрелку вправо**. Если вы находитесь в виртуальном терминале с графическим интерфейсом, добавьте клавишу **ctrl**. Так в RHEL 7, если графический интерфейс установлен и вы находитесь в первом виртуальном терминале, вы нажимаете **ctrl-alt-f2**, чтобы перейти ко второму виртуальному терминалу.

Когда вы входите в обычный виртуальный терминал, Linux возвращает оболочку командной строки. Оболочка по умолчанию для пользователя определена в файле **/etc/passwd**, описанном в **главе 6**. Когда вы входите в виртуальный терминал с графическим интерфейсом, Linux возвращает настроенный рабочий стол с графическим интерфейсом. Для получения дополнительной информации о графическом интерфейсе Linux см. **Главу 8**.

В RHEL 6 виртуальные терминалы настраивались в файлах в **/etc/sysconfig/init** и в каталоге **/etc/init**. Теперь, когда **systemd** заменил **Upstart**, они определяются файлом **logind.conf** в каталоге **/etc/systemd**.

Виртуальные терминалы воплощают в жизнь многопользовательские возможности Linux. На работе (или во время экзамена Red Hat) вы можете просмотреть справочную страницу на одном терминале, скомпилировать программу на другом и отредактировать файл конфигурации на третьем виртуальном терминале. Другие подключенные пользователи могут делать то же самое одновременно.

ЦЕЛЬ СЕРТИФИКАЦИИ 5.04

Управление Target

С помощью **systemd** управление службами Red Hat Enterprise Linux настраивается в зависимости от **target**. Поскольку **systemd** содержит ссылки на уровни выполнения для обратной совместимости с **SysVinit**, вы все равно можете ссылаться на уровни выполнения, перечисленные в **таблице 5-2**, с помощью таких команд, как **init** и **telinit**. Тем не менее, вы должны ознакомиться с **targets**, потому что это стандартный метод активации служб при загрузке.

Linux очень настраиваемый. Поэтому имеет смысл, что системные блоки (**units**), которые запускаются в каждом **target**, могут быть настроены. Хотя инструменты GUI доступны для настройки модулей **systemd**, их настройка из интерфейса командной строки обычно выполняется намного быстрее.

Функциональность Target (задачи, цели)

Как описано ранее, основные функциональные возможности каждой цели перечислены в файлах конфигурации в каталогах **/etc/systemd/systemd** и **/usr/lib/systemd/system**. Например, давайте начнем с **target** по умолчанию, которая в системе RHEL 7 с графическим входом в систему

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

Система знает, что **graphical.target** является значением по умолчанию благодаря символической ссылке из файла **/etc/systemd/system/default.target** в файл **graphical.target** находится в **/etc/systemd/system/default.target**. Посмотрите на один из этих файлов. Выдержка показана здесь:

[Unit]

Description=Graphical Interface
Documentation=man:systemd.special(7)
Requires=multi-user.target
After=multi-user.target
Conflicts=rescue.target
Wants=display-manager.service
AllowIsolate=yes

Это означает, что **target** может включать в себя другие **targets**. В этом случае **graphical.target** является надмножеством **multi-user.target**. После запуска всех **units** **systemd** в **multi-user.target** **graphical.target** активирует **display-manager.service**, как указано в директиве **Wants** в файле конфигурации **graphical.target**.

Другие сервисы, запущенные **graphical.target**, могут быть перечислены в подкаталоге **graphical.target.wants** в **/etc/systemd/system** и **/usr/lib/systemd/system**. При установке по умолчанию **RHEL 7** мы видим следующие файлы:

```
# ls /etc/systemd/system/graphical.target.wants
accounts-daemon.service rtkit-daemon.service
```

Это символические ссылки на файлы конфигурации **unit** сервисов **Accounts** и **RealtimeKit**.

Внутри Units **systemd**

Системные юниты активируются всякий раз, когда система перемещается к другому **target**. Поэтому юниты, связанные с **target** по умолчанию, выполняются во время процесса загрузки. Соответствующие юниты также запускаются при смене **target**; например, когда вы запускаете команду **systemctl isolate multi-user.target** из **graphical.target**, Linux останавливает все сервисные юниты, которые были запущены графическим **target**.

Но вы можете контролировать системные юниты напрямую. Например, проверьте содержимое файла **rsyslog.service** из каталога **/usr/lib/systemd/system**, как показано на рисунке 5-12.

РИСУНОК 5-12 Файл конфигурации сервисного юнита **rsyslog**

```
[root@server1 ~]# cat /usr/lib/systemd/system/rsyslog.service
[Unit]
Description=System Logging Service
Requires=syslog.socket

[Service]
Type=notify
EnvironmentFile=-/etc/sysconfig/rsyslog
ExecStart=/usr/sbin/rsyslogd -n $SYSLOGD_OPTIONS
StandardOutput=null

[Install]
WantedBy=multi-user.target
;Alias=syslog.service
[root@server1 ~]# █
```

Конфигурация начинается с раздела **Unit**, который содержит описание сервиса. Затем следует настройка службы, которая включает тип службы, указатель на файл с некоторыми переменными среды, которые настраивают поведение службы, основной исполняемый файл, запускаемый для активации службы, и директива, которая отправляет все стандартные выходные данные службы. в **/dev/null**.

Наконец, директива **WantedBy** сообщает нам, что эта служба будет активирована при загрузке, когда система войдет в **multi-user target**.

Теперь запустите следующую команду:

```
# systemctl status rsyslog.service
```

Если вы указываете имя **unit** без расширения, по умолчанию **systemd** предполагает, что это **service unit**. Следовательно, короткая версия предыдущей команды

```
# systemctl status rsyslog
```

Эта команда должна возвращать выходные данные, аналогичные показанным на **рис. 5-13**, включая состояние сервисного блока, его ID основного процесса и до 10 самых последних строк журнала. Если некоторые строки журнала обрезаются, используйте **ключ -l**, чтобы отобразить их полностью.

Вы можете остановить службу, выполнив такую команду:

```
# systemctl stop rsyslog.service
```

РИСУНОК 5-13 Отображение статуса service

```
[root@server1 ~]# systemctl status rsyslog.service
rsyslog.service - System Logging Service
   Loaded: loaded (/usr/lib/systemd/system/rsyslog.service; enabled)
   Active: active (running) since Thu 2015-01-22 08:47:34 GMT; 8h ago
   Main PID: 634 (rsyslogd)
   CGroup: /system.slice/rsyslog.service
           └─634 /usr/sbin/rsyslogd -n

Jan 22 08:47:33 server1.example.net systemd[1]: Starting System Logging Service.
Jan 22 08:47:34 server1.example.net systemd[1]: Started System Logging Service.
```

ТАБЛИЦА 5-4 Сервис systemctl Команды управления

Команда	Описание
start	Запускает службу, если она в данный момент не работает.
stop	Останавливает службу, если она работает в данный момент.
restart	Останавливается, а затем запускает службу.
reload	Если поддерживается, загружается текущая версия файла (ов) конфигурации. Служба не останавливается, и клиенты, которые ранее подключились, не запускаются.
try-restart	Останавливает, а затем перезапускает службу, только если она уже запущена.
condrestart	То же, что и попытаться перезапустить.
try-restart	Перечисляет текущее рабочее состояние службы.

В качестве альтернативы команда **systemctl** может использоваться с параметрами, показанными в таблице 5-4; например, следующая команда перезагружает файл конфигурации SSH без остановки или запуска службы:

```
# systemctl reload sshd.service
```

Настройка сервиса

Команда **systemctl** дает вам простой способ включить службу для **target** по умолчанию. Сначала попробуйте следующую команду:

```
# systemctl list-unit-files --type = service
```

Это дает вывод, аналогичный **рисунку 5-11**, но ограниченный обслуживаемыми юнитами. Вы увидите весь список установленных служб в системе, а также их статус активации при загрузке.

Команда **systemctl** может сделать больше. С помощью этой команды вы можете изменить состояние загрузки определенного сервиса. Например, следующая команда проверяет, настроена ли служба Postfix для запуска при загрузке:

```
# systemctl list-unit-files | grep postfix.service
postfix.service enabled
```

Эквивалентная команда

```
# systemctl is-enabled postfix.service
enabled
```

Это указывает на то, что **почтовый сервер Postfix** настроен на запуск **default target**. Если вы хотите убедиться, что служба **Postfix** не запускается **default target**, выполните следующую команду:

```
# systemctl disable postfix.service
```

Снова введите команду **systemctl list-unit-files**, чтобы подтвердить изменение. Чтобы снова включить его для **default target**, выполните ту же команду, подставив **enable** вместо **disable**, как показано здесь:

```
# systemctl enable postfix.service
```

Когда вы включаете (**enable**) службу, команда включения **systemctl** создает символическую ссылку в каталоге **/etc/systemd/system/multi-user.target.wants**, которая указывает на соответствующий файл конфигурации модуля в **/usr/lib/systemd/system**. При желании вы можете включить или отключить службы вручную, создав символические ссылки в соответствующих каталогах **systemd**. Однако использование **systemctl** является предпочтительным способом, поскольку он менее подвержен ошибкам.

Когда служба отключена, вы все равно можете запускать и останавливать ее вручную с помощью команд **systemctl start** и **stop**. Это означает, что команда **systemctl disable** не предотвращает случайный запуск пользователем службы по ошибке. Если вы хотите отключить (**disable**) сервисный юнит при загрузке и убедиться, что он больше не может быть запущен, вы должны использовать команду **mask**, как показано здесь:

```
# systemctl mask postfix.service
ln -s '/dev/null' '/etc/systemd/system/postfix.service'
```

Как показано, эта команда создает символическую ссылку в **/etc/systemd/system** на **postfix.service**, который указывает на **/dev/null**. Файл конфигурации в **/etc/systemd/system** всегда имеет приоритет над соответствующим файлом в **/usr/lib/systemd/system**. Следовательно, в результате файл **postfix.service** по умолчанию в **/usr/lib/systemd/system** «маскируется» символической ссылкой в **/etc/systemd/system** на **/dev/null**.

ЦЕЛЬ СЕРТИФИКАЦИИ 5.05

Синхронизация времени

Конфигурация клиента сетевого протокола времени (**NTP**) проста. Поэтому в этом разделе представлен обзор файлов конфигурации и соответствующих командных инструментов.

Существуют веские причины для того, чтобы разные системы работали на одних и тех же часах. Например, веб-сервер и клиент, регистрирующийся в разное время, могут чрезвычайно затруднить устранение неполадок. Некоторые службы используют точные метки времени. Например, смещение времени более пяти минут может привести к сбою аутентификации клиента **Kerberos**.

RHEL 7 включает **RPM** для двух демонов NTP: **ntpd** и **chronyd**. Не устанавливайте оба. Как правило, **ntpd** рекомендуется для систем, которые всегда подключены к сети, таких как серверы, тогда как **chronyd** является предпочтительным выбором для виртуальных и мобильных систем. Мы опишем конфигурацию службы синхронизации времени по умолчанию, **chronyd**. Но сначала мы объясним, как настроить часовой пояс.

Конфигурация часового пояса

Каждая система, реальная или виртуальная, запускается с аппаратных часов. Время на этих часах может зависеть от мощности батареи; со временем батареи теряют заряд, а многие аппаратные часы теряют время. Процесс установки на RHEL 7 обычно устанавливает аппаратные часы на местное время, а не на **UTC**. Тем не менее, **UTC** (которое по существу идентично среднему времени по Гринвичу или по Гринвичу) обычно является наилучшей настройкой для серверов, чтобы избежать проблем при переходе на летнее время.

Каждая система RHEL 7 включает часовой пояс, настроенный в файле **/etc/localtime**. Это символическая ссылка, которая указывает на один из файлов часовых поясов **/usr/share/zoneinfo**. Например, **/etc/localtime** должен указывать **/usr/share/zoneinfo/America/Los_Angeles**, если вы находитесь в Калифорнии.

Вместо того, чтобы вручную устанавливать символическую ссылку на файл часового пояса, вы можете использовать утилиту **timedatectl**. Если вы запустите команду без аргументов, она покажет сводку текущих настроек времени, включая текущее время, часовой пояс и состояние **NTP**. Некоторые примеры выходных данных показаны на рисунке 5-14.

Вы можете отобразить список доступных часовых поясов, выполнив следующую команду:

```
# timedatectl list-timezones
```

Затем, чтобы переключиться на другой часовой пояс, запустите **timedatectl** с командой **set-timezone**. Вот пример:

```
# timedatectl set-timezone America/Los_Angeles
```

РИСУНОК 5-14 Настройки даты и времени

```
[root@server1 ~]# timedatectl
Local time: Thu 2015-01-22 19:38:06 GMT
Universal time: Thu 2015-01-22 19:38:06 UTC
RTC time: Thu 2015-01-22 19:38:06
Timezone: Europe/London (GMT, +0000)
NTP enabled: yes
NTP synchronized: yes
RTC in local TZ: no
DST active: no
Last DST change: DST ended at
Sun 2014-10-26 01:59:59 BST
Sun 2014-10-26 01:00:00 GMT
Next DST change: DST begins (the clock jumps one hour forward) at
Sun 2015-03-29 00:59:59 GMT
Sun 2015-03-29 02:00:00 BST
[root@server1 ~]# █
```

Синхронизировать время с chronyd

Файл конфигурации **chronyd** по умолчанию, **/etc/chrony.conf**, настроен для подключения к нескольким публичным серверам из проекта пула **NTP**. При совместном использовании демон **chronyd** минимизирует временные ошибки.

```
server 0.rhel.pool.ntp.org iburst
server 1.rhel.pool.ntp.org iburst
server 2.rhel.pool.ntp.org iburst
server 3.rhel.pool.ntp.org iburst
```

Пользователи кто использует дистрибутивы, такие как CentOS, увидят разные имена хостов, например **0.centos.pool.ntp.org**. Показанный здесь параметр конфигурации **iburst** ускоряет начальную синхронизацию при запуске службы **chronyd**.

Чтобы настроить **chronyd** для синхронизации с другим **NTP-сервером**, просто измените директивы сервера в **/etc/chrony.conf** и перезапустите **chronyd**:

```
# systemctl restart chronyd
```

Вы можете отобразить информацию об источниках текущего времени с помощью команды **chronyc sources -v**. Пример показан на **рисунке 5-15**.

РИСУНОК 5-15 Статистика NTP-сервера

```
[root@server1 ~]# chronyc sources -v
210 Number of sources = 4

.-- Source mode '^' = server, '=' = peer, '#' = local clock.
/ .- Source state '*' = current synced, '+' = combined , '-' = not combined,
/   '?' = unreachable, 'x' = time may be in error, '~' = time too variable.
||
||                                     .- xxxx [ yyyy ] +/- zzzz
||                                     /   xxxx = adjusted offset,
||                                     |   yyyy = measured offset,
||                                     |   zzzz = estimated error.
||                                     \
||      Log2(Polling interval) - .-
||
MS Name/IP address         Stratum Poll Reach LastRx Last sample
=====
^* kvm1.websters-computers.c 2  6   77   33  +717us[ -992us] +/-  26ms
^+ static.132.14.76.144.clie 2  6   37   97  +3483us[+1774us] +/-  67ms
^+ ntp-ext.cosng.net         2  6   77   31  -2928us[-2928us] +/-  42ms
^+ ghost-networks.de         2  6   77   32  -1419us[-1419us] +/-  51ms
[root@server1 ~]#
```

Синхронизируйте время с ntpd

Базовая конфигурация демона **ntpd** проста. Во-первых, вы должны убедиться, что **chronyd** остановлен и отключен при загрузке, потому что вы не можете одновременно запускать **chronyd** и **ntpd** на одной машине:

```
# systemctl stop chronyd.service
# systemctl disable chronyd.service
```

Затем установите RPM-пакет **ntp**:

```
# yum install ntp
```

Файл конфигурации **ntpd** по умолчанию **/etc/ntp.conf**. Он похож на файл **/etc/chronyd.conf** и содержит четыре директивы сервера, указывающие на публичные серверы, которые являются частью проекта пула **NTP**. Вы можете настроить конфигурацию или запустить **ntpd** с настройками по умолчанию. После того, как вы внесли изменения в файл, запустите и включите **ntpd**:

```
# systemctl start ntpd.service
# systemctl enable ntpd.service
```

Чтобы отобразить информацию об источниках **NTP**, выполните команду **ntpq -p**.

РЕЗЮМЕ СЕРТИФИКАЦИИ

В этой главе описан основной процесс загрузки системы RHEL. Он начинается с **аппаратного POST** и продолжается с **системой BIOS** или **UEFI**. Как только загрузочный носитель найден, процесс переходит на первый этап загрузчика **GRUB 2**. Меню **GRUB 2** позволяет выбрать и настроить ядро для загрузки.

После того, как вы выбрали опцию, **GRUB 2** передает управление ядру. Ядро загружает временную файловую систему, известную как начальный **RAM-диск**. После загрузки необходимых драйверов и файловых систем вы можете просмотреть журнал **systemd** с помощью команды **journalctl**. Затем ядро выполняет первый процесс, также известный как **systemd**.

Службы Linux контролируются **systemd targets**, которые объединяют другие системные юниты. **Default target** настроена как символическая ссылка в каталоге **/etc/systemd/system**, и файлы конфигурации **юнитов** хранятся в этом каталоге и в **/usr/lib/systemd/system**. Состояние **systemd units** можно настроить и запросить с помощью команды **systemctl**. **Target systemd** связаны с другими **target** и файлами конфигурации юнитов. **systemctl** также может использоваться для **start, stop, restart, reload systemd units** и многого другого.

Вам может потребоваться настроить локальные системы в качестве клиентов **NTP**. Службой **NTP** по умолчанию в RHEL 7 является **chronyd**.

Пару минут самопроверки.

Здесь некоторые из ключевых моментов целей сертификации в главе 5.

BIOS и UEFI

- Хотя это не является частью экзамена, важно знать основы **BIOS** и **UEFI**.
- Вы можете изменить последовательность загрузки из меню **BIOS/UEFI**.
- Как только **BIOS/UEFI** обнаружит назначенный загрузочный диск (и), он передает управление в **GRUB 2** через основную загрузочную запись (**MBR**) или таблицу разделов **GUID (GPT)** соответствующий диск.

Загрузчики и GRUB 2

- RHEL 7 использует **GRUB 2**.
- Файл конфигурации **GRUB 2** состоит из разделов.
- В меню **GRUB 2** вы можете загрузить **systemd target**, отличную от заданной по умолчанию.
- Вы даже можете загрузиться из меню **GRUB 2** в спасательную оболочку (**rescue shell**), которая предоставляет административный доступ **root** без пароля учетной записи.
- Файл конфигурации **GRUB 2** определяет ядро, том корневого каталога и начальный **RAM-диск** для каждой операционной системы.

- Если файл конфигурации **GRUB 2** отсутствует, вы можете загрузиться с **grub>** запросить информацию о разделе каталога **/boot**, файле ядра Linux, корневой каталог верхнего уровня и исходный файл RAM-диска.

Между GRUB 2 и логином

- Вы можете анализировать загрузочные сообщения с помощью команды **journalctl**.
- **system targets** по умолчанию настраиваются, как символическая ссылка из каталога **/etc/systemd/system**.
- Процесс **systemd** заменил **Upstart** и **SysVinit** в качестве первого процесса. Процесс имеет файлы конфигурации в каталогах **/etc/systemd/system** и **/usr/lib/systemd/system**.
- После загрузки ядра управление передаётся **systemd**, также известному как первый процесс.

Контроль Target

- **Default target**, настроенная в **/etc/systemd/system**, активирует системные **units** в каталоге **/usr/lib/systemd/system**.
- Target юниты могут включать другие **targets** и юниты (**units**), которые нужно активировать.
- Вы можете использовать **systemctl** для управления службой с помощью **start**, **stop**, **restart**, **reload** и другими командами.
- Службы, которые запускаются в каждой **target**, также можно контролировать с помощью **systemctl** и команды **enable/disable**.

Синхронизация времени

- Инструмент **timedatectl** можно использовать для проверки текущего времени, даты, часового пояса и статус **сервиса NTP**.
- Службой **NTP** по умолчанию в RHEL 7 является **chronyd**. Сервис синхронизирует время с серверами и настраивается в файле **/etc/chrony.conf**.
- Альтернативой **chronyd** является **ntpd**, который сохраняет свои настройки в файл **/etc/ntp.conf**.
- Не запускайте **chronyd** и **ntpd** одновременно.

САМООЦЕНОЧНЫЙ ТЕСТ

Следующие вопросы помогут оценить ваше понимание материалов, представленных в этой главе. Поскольку на экзаменах Red Hat не появляется вопросов с несколькими вариантами ответов, вопросы с несколькими вариантами ответов не появляются в этой книге. Эти вопросы исключительно проверяют ваше понимание главы. Это нормально, если у вас есть другой способ выполнения задачи. Получение результатов, а не запоминание пустяков - вот на что рассчитывает Red Hat экзамены. На многие из этих вопросов может быть более одного ответа.

BIOS и UEFI

1. На какой части загрузочного жесткого диска обычно находится первая ступень загрузчика GRUB 2?

Загрузчики и GRUB 2

2. Когда вы увидите меню конфигурации GRUB 2, какую команду вы бы использовали для изменения конфигурации?

3. Какую строку вы бы добавили в командную строку **linux16** для загрузки в аварийную цель?

4. Если вы видите директиву **set root = 'hd0, msdos1'** в файле конфигурации **GRUB 2**, на каком разделе такое каталог **/boot**? Предположим, что файл конфигурации **GRUB 2** настроен правильно.

Между GRUB 2 и логином

5. Какая временная файловая система загружается прямо из меню **GRUB 2**?

6. Какую команду из одного слова вы можете использовать для чтения сообщений журнала **systemd**?

7. В каких каталогах вы можете найти файлы конфигурации, связанные с первым процессом?

8. Как вы переключаетесь с графической цели на многопользовательскую(**graphical target to the multi-user target**)?

Контроль Target

9. Какая команда перечисляет **target** по умолчанию?

10. Назовите три команды, которые обычно запускаются из **systemctl** для контроля состояния системных юнитов.

11. Какая команда перечисляет состояние всех системных юнитов, доступных в настоящее время в локальной системе, в том числе те, которые не активны?

Синхронизация времени

12. Какая команда перечисляет текущее время, часовой пояс и статус службы NTP?

13. Какой файл конфигурации используется **chronyd**?

ВОПРОСЫ ДЛЯ ЛАБОРАТОРНОЙ РАБОТЫ

Некоторые из этих лабораторных работ включают установочные упражнения. Вы должны выполнять эти упражнения только на тестовых машинах. Вторая лабораторная работа главы 1 устанавливает KVM для этой цели.

Red Hat представляет свои экзамены в электронном виде. По этой причине лаборатории в этой и будущих главах доступны из средств массовой информации, которая сопровождает книгу. Посмотрите в подкаталоге **Chapter5/** лабораторная работа для этой главы. Задачи доступны в форматах .doc, .html и .txt. Если вы еще не настроили RHEL 7 в системе обратитесь к первой лабораторной работе главы 2 за инструкциями по установке. Ответы для каждой лаборатории следуйте ответам на самотестирование для заполнения вопросов.

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 5.

Для **лабораторной работы 4** вам потребуется выполнить сценарий из каталога **Chapter5/ DVD-диска**, входящего в комплект книги, в тестовой системе **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. Ниже перечислены шаги, общие для первых четырех лабораторных работ.

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 **View | Details**.
4. Вставьте **DVD** из книги. Используйте отображаемые параметры для подключения дисководов **CD/DVD** к виртуальной машине. Вернитесь в консоль для виртуальной машины, нажав **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/Chapter5/Ch5Lab* ~.**
8. Эти лабораторные работы должны работать, даже если каталог **/root/backup** существует, хотя в этом случае вы увидите сообщение об ошибке.
9. Убедитесь, что скрипты исполняемые; запустить команду **chmod +x Ch5Lab?**.

Лабораторная работа 1

Эта Лабораторная работа ориентирована на рассмотрение **targets** и процесс загрузки. Вы загрузите систему в **emergency target** с полным отображением всех загрузочных сообщений. Затем вы настроите загрузку системы по умолчанию в **multi-user target**.

1. На какой части загрузочного жесткого диска обычно находится первая ступень загрузчика GRUB 2?
2. Включите локальную систему. В процессе загрузки, когда вы видите следующее сообщение (имя операционной системы и номер версии могут отличаться), нажмите клавишу:

The selected entry will be started automatically in 5 seconds....3.

3. Нажмите **E**, чтобы редактировать текущий пункт меню.
4. Прокрутите вниз, чтобы обнаружить строку, начинающуюся с **linux16**.
5. Что нужно изменить и добавить в эту командную строку для загрузки в **emergency target**?
6. Внесите необходимые изменения, а затем продолжите загрузку в **emergency target**.
7. Перезагрузите систему в **graphical target**. Какова **target** по умолчанию?
8. Измените символическую ссылку **/etc/systemd/system/default.target**, чтобы система нормально загружалась в **multi-user target**.
9. Перезагрузите систему. Как вы подтвердите, что изменения сработали?
10. Восстановите исходную цель по умолчанию.

Лабораторная работа 2

В этой лабораторной работе вы измените пароль администратора. Но вот поворот: предположим, что вы не знаете текущую версию этого пароля. Что вы сделаете?

Лабораторная работа 3

В этой лабораторной работе вы установите время ожидания GRUB 2 до 10 секунд. Прежде чем начать, лучше сделать резервную копию файла. Например, следующая команда создает резервную копию файла в домашнем каталоге пользователя **root** (**/root**):

```
# cp /boot/grub2/grub.cfg ~
```

В качестве дополнительной задачи измените конфигурацию **GRUB 2**, чтобы включить подробные загрузочные сообщения при загрузке. Чтобы подтвердить результат, перезагрузите систему.

Лабораторная работа 4

Будьте осторожны с этой лабораторией. Шаги могут сделать эту систему не загружаемой, если вы не понимаете навыки, описанные в этой главе. Всегда существует риск того, что ключевые файлы конфигурации не будут должным образом сохранены, что означает риск потери всех данных в системе. Если вы понимаете эти риски, переходите к этой лабораторной работе.

Эта лаборатория основана на сценарии **Ch5Lab4**, описанном ранее. Если вы следовали инструкциям в начале этого раздела, скрипт должен быть доступен в вашем **/root** каталоге. Перейдите в этот каталог и выполните следующие шаги:

1. Войдите в учетную запись **root**. Если вы еще не там, перейдите в каталог **/root**. Выполните сценарий для этой лабораторной работы с помощью команды **./Ch5Lab4**.
2. Перезагрузите систему.

3. Когда вы увидите приглашение **grub>**, используйте навыки, описанные в этой главе, чтобы идентифицировать диск и раздел с каталогом **/boot**.
4. Где это применимо, используйте функции завершения команды в приглашении **grub>**. Это особенно полезно с командами **linux** / и **initrd** /.
5. Помните, что корневой каталог верхнего уровня определяется директиве **root** с командой **kernel**.
6. Не забудьте импортировать модуль **LVM**.
7. После ввода местоположения исходного **RAM-диска** выполните команду загрузки в приглашении **grub>**.
8. Если ваши усилия будут успешными, система загрузится нормально. В разделе «Лабораторные ответы» вы увидите, как восстановить резервную копию файла конфигурации **GRUB 2**.
9. Если ваши попытки не увенчались успехом, загрузите систему с установочного DVD или сетевого загрузочного CD и выберите Устранение неполадок, как описано в основной части главы.

Лабораторная работа 5

Эта лабораторная работа ориентирована на активные терминалы. Обычно Linux включает шесть активных терминалов. Если графический интерфейс пользователя установлен и активен с помощью диспетчера графического отображения, эта система запускается вместо первого активного терминала. Чтобы просмотреть, вы можете переключаться между активными терминалами с помощью клавиши **ALT** и функциональной клавиши, связанной с номером клеммы, такой как **ALT-F1**. Если вы находитесь в графическом интерфейсе, вам нужно добавить клавишу **CTRL** в комбинацию.

1. Сделайте резервную копию текущих версий файла конфигурации **/etc/systemd/logind.conf**. Логическим местоположением является домашний каталог текущего пользователя. Если вы совершите серьезную ошибку, вы сможете восстановить систему из резервной копии. В худшем случае вы сможете восстановить эти файлы из резервной копии, загрузившись в аварийную (**rescue**) или аварийную (**emergency**) **target**, как описано в лабораторной работе 1 и в основной части главы.
2. Теперь ограничьте активные консоли терминалами 1 и 2.
3. В случае успеха вы сможете получить доступ только к этим двум терминалам в локальной системе. Можете ли вы восстановить исходную конфигурацию?

ОТВЕТЫ НА САМОПРОВЕРКУ

BIOS и UEFI

1. Для того чтобы **BIOS/UEFI** передал управление **Linux**, необходимо определить основную загрузочную запись (**MBR**) или таблицу разделов **GUID (GPT)** загрузочного жесткого диска.

Загрузчики и GRUB 2

2. Из меню **GRUB 2** команда, которая изменяет конфигурацию, - **e**.
3. Чтобы загрузиться в аварийную цель из командной строки **GRUB 2 linux16**, вы должны добавить строку **systemd.unit = emergency.target**.
4. Директива **set root = 'hd0, msdos1'** документирует каталог **/boot** в первом разделе первого жесткого диска.

Между GRUB 2 и логином

5. Временная файловая система, загруженная из меню **GRUB 2**, также является исходной файловой системой **RAM-диска(initial RAM disk)** так же известный по имени файла, **initramfs**.
6. Командой из одного слова, которую вы можете использовать для чтения сообщений журнала **systemd**, является **journalctl**.
7. Файлы конфигурации, связанные с первым процессом, находятся в каталогах **/etc/systemd/system** и **/usr/lib/systemd/system**.
8. Команда для переключения с графического **target** на многопользовательский **target - systemdctl isolate multi-user.target**.

Контроль Target

9. Командой, которая перечисляет **target** по умолчанию, является **systemctl get-default**.
10. Типичные команды, которые можно запустить из **systemctl**, включают **start, stop, restart, reload, enable, disable**, и многое другое.
11. Команда **systemctl list-units --all** (или просто **systemctl --all**) перечисляет состояние всех **юнитов**, в том числе неактивные.

Синхронизация времени

12. Команда **timedatectl** отображает текущее время, часовой пояс и состояние службы **NTP**.
13. Файл конфигурации **chronyd** - это **/etc/chrony.conf**.

ОТВЕТЫ НА ВОПРОСЫ ЛАБОРАТОРНОЙ РАБОТЫ

Да, есть много систем Linux, которые работают годами без перезагрузки. Но перезагрузки иногда требуется, например, когда установлены более новые ядра. Так что при настройке системы **Linux**, убедитесь, что любые изменения пережили перезагрузку. На экзамене Red Hat вы не получите кредит, если ваши изменения не пережили перезагрузку.

Лабораторная работа 1

В случае успеха эта лабораторная работа покажет вам, как изменить **target** по умолчанию, а также сравнительную важность опций в загрузчике GRUB. Помните, что вы можете изменить цель по умолчанию через команду **systemctl**

```
# systemctl set-default multi-user.target
```

или вручную, изменив символическую ссылку **/etc/systemd/system/default.target**:

```
# rm -f /etc/systemd/system/default.target
```

```
# ln -s /usr/lib/systemd/system/multi-user.target /etc/systemd/system/default.target
```

Лабораторная работа 2

Эта лабораторная работа такая же, как упражнение 5-2. Практикуйтесь с процедурой восстановления пароля **root**, пока вы не знакомы со всеми этапами, и вам не нужно полагаться на документацию. Помните, экзамены Red Hat являются «закрытой книгой».

Лабораторная работа 3

После завершения этой лабораторной работы вы должны были изменить переменные **GRUB_TIMEOUT** и **GRUB_CMDLINE_LINUX** в **/etc/default/grub**, как показано здесь:

```
GRUB_TIMEOUT=10
GRUB_CMDLINE_LINUX="rd.lvm.lv=rhel/root vconsole.font=latacyrheb-sun16 \
rd.lvm.lv=rhel/swap crashkernel=auto vconsole.keymap=uk rhgb"
```

Обратите внимание, что ключевое слово «**quiet**» было удалено из **GRUB_CMDLINE_LINUX**, чтобы включить подробный вывод сообщений при загрузке.

Затем выполните команду **grub2-mkconfig** для создания нового файла конфигурации GRUB:

```
# grub2-mkconfig -o /boot/grub2/grub.cfg
```

Чтобы действительно проверить результат, перезагрузите систему. Что происходит? В конце, отмените ваши изменения.

Лабораторная работа 4

Сценарий, выполненный в этой лабораторной работе, переместил файл конфигурации **grub.cfg** в каталог **/root/backup**. Если вы хорошо понимаете GRUB 2, вы должны были загрузить систему из приглашения **grub>**.

В противном случае вы можете восстановить файл **grub.cfg**, загрузившись в **rescue target**, описанную в этой глав. Из командной строки **rescue target** вы сможете восстановить исходную конфигурацию с помощью следующих команд:

```
# chroot /mnt/sysimage
# cp /root/backup/grub.cfg /boot/grub2/
```

Или создайте новый файл конфигурации **grub.cfg** с помощью команды **grub2-mkconfig**.

Лабораторная работа 5

Можно настроить до 12 виртуальных терминалов, которые соответствуют количеству доступных функциональных клавиш на большинстве клавиатур. Если вы хотите настроить 12 виртуальных терминалов (и это было бы интересной проблемой для экзамена RHCSA) посмотрите файл **/etc/securetty** и соответствующие страницы руководства команды с **man -k securetty**. Один из способов выполнить задачи в этой лабораторной работе - выполнить следующие шаги:

1. Откройте файл **/etc/systemd/logind.conf**. Измените следующую директиву, чтобы ограничить активные консоли терминалами 1 и 2:

```
NAutoVTs = 2
```

2. Чтобы проверить результат, перезагрузите систему.
3. Что происходит? Можете ли вы войти в терминалы 3, 4, 5 и 6?
4. По завершении просто не забудьте восстановить исходную версию файла **/etc/systemd/logind.conf**.