Содержание

Введение 6
Условные обозначения 7
Схема класса и виртуальной сети 8

Модуль 1. Системное администрирование 10
1.1. Процесс загрузки ОС 10
1.1.1. Универсальный загрузчик GRU 11
1.1.2. Работа с командами загрузчика GRUB 14
1.2. Уровни выполнения 18
1.3. Определение характеристик оборудования и его совместимости .. 23
1.4. Установка ОС 25
1.4.1. Методы установки ОС Linux 25
1.4.2. Создание инсталляционного ресурса 26
1.4.3. Процесс установки 27
1.4.4. Автоматизированная установка kickstart 32
1.4.5. Установка с использованием технологии PXE 36
1.5. Управление пакетами программ 36
1.5.1. Менеджер пакетов RPM 37
1.5.2. Менеджер пакетов YUM 42
1.5.3. Создание пакетов RPM 44
1.6. Организация хранения данных 47
1.6.1. Создание и управление массивами RAID 48
1.6.2. Создание и управление логическими томами с помощью LVM .. 51
1.7. Резервное копирование и восстановление данных 52
1.7.1. Утилиты dump и restore 53
1.7.2. Утилита tar 54
1.7.3. Утилита rsync 55
1.8. Автоматизация выполнения задач 56
1.9. Журналирование системных событий 58
1.10. Мониторинг и оптимизация производительности 61
1.10.1. Процессор 61
1.10.2. Память 64
1.10.3. Подсистема дискового ввода-вывода 66
1.10.4. Комплексные средства мониторинга системных ресурсов 67
1.11. Установка программ из исходного кода 68

Модуль 2. Сетевое администрирование 76
2.1. Централизованное хранилище данных. Каталоги LDAP..... 76
2.1.1. Настройка конфигурационных файлов LDAP 78
2.1.2. Создание каталога LDAP..... 79
2.1.3. Подключение к серверу LDAP 80
2.2. Организация общего доступа к файлам (NFS, SMB) 82
2.2.1. Организация общего доступа на основе NFS 82
2.2.2. Организация общего доступа на основе Samba 86
2.3. Система разрешения имен (DNS). Автоматизация получения сетевых параметров (DHCP)..... 89
2.3.1. Сервис DNS..... 89
2.3.2. Сервис DHCP..... 95
2.4. Администрирование веб-сервера Apache 96
2.5. Администрирование прокси-сервера SQUID 101
2.6. Защищенное администрирование. Пакет OpenSSH..... 104
2.7. Система обмена почтовыми сообщениями. Хранилище почтовых данных 107
2.7.1 Конфигурирование Sendmail 107
2.7.2. Конфигурирование Dovecot 110
2.8. Администрирование общих сетевых сервисов (xinetd, FTP, NTP) 111
2.8.1. Сервис xinetd..... 111
2.8.2. Сервис FTP..... 112
2.8.3. Сервис NTP 114
2.9. Основы маршрутизации в сетях TCP/IP 115

Модуль 3. Организация информационной безопасности 118
3.1. Управление интеллектуальными списками доступа (SELinux) 118
3.1.1. Режим работы и политика механизма SELinux..... 119
3.1.2. Использование утилиты SELinux Troubleshooting Tool..... 120
3.1.3. Работа с контекстами безопасности 120
3.2. Организация межсетевого экрана (iptables) 121
3.3. Организация виртуальных частных сетей (VPN)..... 126
3.4. Аудит системных событий (auditd)..... 130

Приложения 135
Приложение 1. Конфигруационый файл загрузчика GURB2 135
Приложение 2. Описание основных категорий и групп пакетов ОС Linux..... 136
Приложение 3. Основные команды управления LVM 138
Приложение 4. Основные журнальные файлы ОС Linux 140

Практические работы 142
Описание виртуальных машин 142

Практическая работа 1. Системное администрирование ОС Linux 143
Упражнение 1.1. Изменение параметров загрузки ядра ОС Linux 143
Упражнение 1.2. Управление системными сервисами 144
Упражнение 1.3. Сетевая инсталляция ОС Linux с использованием сервиса NFS 145
Упражнение 1.4. Установка ядра ОС Linux 148
Упражнение 1.5. Создание отказоустойчивого и масштабируемого хранилища 149
Упражнение 1.6. Синхронизация данных 152
Упражнение 1.7. Централизованное хранилище журнальных файлов 156
Упражнение 1.8. Мониторинг системных ресурсов 157
Самостоятельные упражнения и дополнительные вопросы 159

Практическая работа 2. Сетевое администрирование ОС Linux 160
Упражнение 2.1. Сетевая аутентификация пользователей LDAP 160
Упражнение 2.2. Настройка монтирования домашнего каталога пользователя по требованию 163
Упражнение 2.3. Настройка динамического обновления записей DNS 165
Упражнение 2.4. Организация централизованного веб-доступа к документации ОС Linux. 169
Упражнение 2.5. Настройка аутентификации SSH по публичному ключу. 172
Упражнение 2.6. Создание защищенного почтового сервиса на основе MTA Sendmail и Dovecot. 174
Упражнение 2.7. Организация файлообменного сервиса FTP. 178
Упражнение 2.8. Синхронизации времени. 181
Самостоятельные упражнения и дополнительные вопросы 183

Практическая работа 3. Организация информационной безопасности ОС Linux 184
Упражнение 3.1. Анализ работы механизма SELinux 184
Упражнение 3.2. Настройка ограничений на подключение к Telnet 185
Упражнение 3.3. Организация межсетевого экрана 186
Самостоятельные упражнения и дополнительные вопросы 188
Упражнение 3.4. Организация контроля целостности файлов 189
Упражнение 3.5. Организация VPN тунеля 190

Введение

Данное пособие – второе в серии книг по Linux, издаваемых совместно ДМК-пресс (www.dmk-press.ru) и Softline Academy Alliance (www.it-academy.ru). Данный курс предполагает наличие у слушателей знаний в объеме курса RH-033 «Основы работы с ОС Red Hat Enterprise Linux» и является следующей ступенью в освоении ОС Linux. Курс позволяет получить знания и навыки, необходимые для успешного администрирования ОС Linux. В качестве основного рассматриваемого в данном курсе дистрибутива используется Red Hat Enterprise Linux 5 (RHEL)[1], являющийся коммерческим решением компании Red Hat.

Курс предназначен для системных администраторов и инженеров начального[2] уровня знаний, стремящихся освоить системное и сетевое администрирование ОС Linux, а также использовать данную ОС для решения задач информационной безопасности. После прохождения данного курса, при желании, слушатели могут успешно сертифицироваться на звания Red Hat Certified Technician (RHCT) и Red Hat Certified Engineer (RHCE)[3].

Предлагаемый вашему вниманию конспект лекций и практические работы — это основной учебный материал для проведения лекционных и практических занятий по курсу «Администрирование ОС Red Hat Enterprise Linux 5» в учебных центрах Softline Academy (www.it-academy.ru). Эти учебные центры создаются в рамках инициативы Softline Academy Alliance, цель которой объединить учебные заведения и организации, заинтересованные в качественной и эффективной подготовке студентов и молодых специалистов для работы в области IT.

Курс разработан преподавателями Учебного центра ВМК МГУ & Softline Academy (www.it-university.ru), который является первым в России авторизованным учебным центром программы Microsoft IT Academy и первой Академией Softline. Курс рассчитан на 40 академических часов и может быть освоен как самостоятельно, так и под руководством опытного преподавателя в любой из двадцати пяти Академий Softline, находящихся в восемнадцати регионах России.

Данный курс состоит из трех частей, изучение каждой из которых предполагает хорошее усвоение предыдущего материала.

Первая часть курса полностью посвящена системному администрированию и

---

[1] Здесь и далее по тексту под «ОС Linux» будет пониматься дистрибутив Red Hat Enterprise Linux 5. Начальная подготовка предполагают уверенное знание.

[2] Начальная подготовка предполагают уверенное знание основ ОС Linux и командного интерпретатора bash, полученных в ходе изучения курса RH-033.

[3] Основная информация по программе сертификации Red Hat представлена на сайте https://www.redhat.com/certification/

содержит материал, необходимый для успешной сдачи экзамена RHCT (RH202). Данная часть курса содержит в себе вопросы инсталляции ОС Linux, управления пакетами программ, обеспечения резервного копирования, организации хранилища данных, а также другие важные вопросы, связанные с системным администрированием.

Вторая часть курса ориентирована на администрирование сетевых служб и содержит материал, необходимый для успешной сдачи экзамена RHCE(RH302). В данной части курса рассматриваются вопросы организации каталогов LDAP, проектирования и развертывания веб-серверов на базе Apache, управления системами кеширования и фильтрации контента, администрирования общих сетевых служб, а также использования ОС Linux для решения задач маршрутизации данных в сетях TCP/IP.

В третьей части курса рассматриваются вопросы информационной безопасности, которые должны знать специалисты, сертифицирующиеся на звание RHCE и желающие продолжить дальнейшее обучение в области защиты данных.

После завершения обучения данному курсу вы будете обладать необходимыми знаниями, и уметь:

- устанавливать, настраивать и обновлять ОС Linux;
- устанавливать и настраивать дополнительные пакеты программ;
- устанавливать программы из исходного кода;
- определять характеристики серверного оборудования и выбирать оптимальную конфигурацию ОС Linux с учетом системных требований;
- эффективно управлять системными процессами и сервисами;
- администрировать учетные записи пользователей и групп;
- обеспечивать надежное хранение и резервирование данных;
- администрировать основные сетевые службы, поддерживаемые ОС Linux;
- настраивать маршрутизацию трафика в сетях TCP/IP, используя ОС Linux и соответствующие программы;
- организовывать безопасное функционирование системных и сетевых служб, используя механизм SELINUX;
- настраивать межсетевой экран;
- настраивать аудит системных и сетевых событий;
- обеспечивать ведение централизованного журнала событий;
- определять «узкие» места в производительности ОС Linux и приложений;
- восстанавливать ОС Linux в случае программных или аппаратных сбоев.

Условные обозначения

В данном пособии применяются следующие условные обозначения.

Имена файлов и папок начинаются со строчных букв (при работе в командной строке или графической оболочке регистр букв всегда имеет значение).

Аббревиатуры напечатаны ПРОПИСНЫМИ БУКВАМИ.

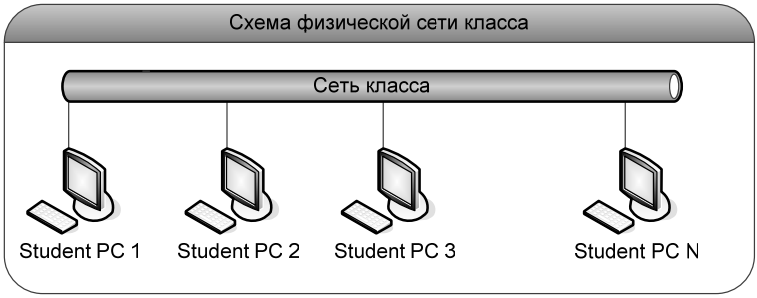
Листинги кода, примеры команд, а также текста, выводимого на экран, выделены `данным шрифтом`, причем ввод команды выделен **жирным шрифтом**.

Необязательные аргументы команд заключены в квадратные скобки (например: данный [`аргумент`] является необязательным).

Обязательные аргументы команд записываются без квадратных скобок.

Ключевые термины выделены ***полужирным курсивом***.

Схема класса и виртуальные машины



Класс, в котором выполняются практические работы, состоит из физических компьютеров, объединенных в локальную сеть с адресом 192.168.1.0/24 и имеющих динамическую IP-адресацию (DHCP). Все практические работы по данному курсу выполняются на виртуальных машинах, работающих под управлением ПО VMware Player.

Учетные данные для регистрации на физическом компьютере слушателя (Host PC):

Имя пользователя: _____

Пароль: _____

Домен (если есть): _____

В качестве операционной системы ***хостовой машины*** (физической рабочей станции слушателя, на которой работает ПО VMware Player) используется ОС Microsoft Windows XP. В качестве операционной системы ***гостевых машин***, в которых непосредственно выполняются все практические работы, используется ОС Red Hat Enterprise Linux 5 (RHEL5).

У каждого слушателя имеется одна виртуальная машина rhel5.linux.lab. Данная виртуальная машина имеет один сетевой интерфейс (eth0), используемый для связи между виртуальными машинами и выхода в физическую сеть класса. Все практические работы выполняются на виртуальной машине rhel5.linux.lab.

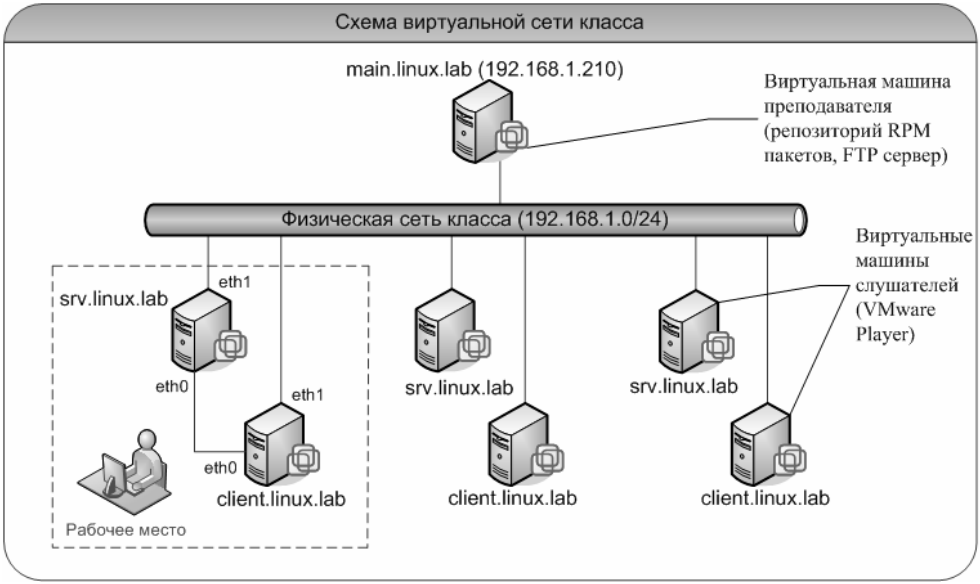


Схема виртуальной сети класса
main.linux.lab (192.168.1.210)
Виртуальная машина преподавателя (репозиторий RPM пакетов, FTP сервер)
Физическая сеть класса (192.168.1.0/24)
Виртуальные машины слушателей (VMware Player)
srv.linux.lab
eth1
eth0
eth1
eth0
client.linux.lab
Рабочее место
srv.linux.lab
client.linux.lab
srv.linux.lab
client.linux.lab

Модуль 1. Системное администрирование

Изучив данный модуль, вы научитесь:

- описывать процесс загрузки ОС Linux и настраивать загрузчик GRUB;
- описывать уровни выполнения (Runlevels);
- управлять загрузкой сервисов ОС Linux;
- описывать процедуру и варианты инсталляции ОС Linux;
- определять аппаратную конфигурацию оборудования, на котором установлена ОС Linux;
- управлять пакетами ОС Linux при помощи менеджеров пакетов **rpm** и **yum**;
- управлять дисковыми разделами, файловыми системами, RAID-группами и томами LVM;
- использовать основные методы резервного копирования данных в ОС Linux;
- автоматизировать выполнение задач при помощи планировщиков **cron** и **at**;
- настраивать журналирование системных событий;
- организовывать мониторинг системных ресурсов ОС Linux.

1.1. Процесс загрузки ОС

После включения компьютера происходит выполнение специального кода начальной загрузки. На персональных компьютерах данной код представлен в виде базовой подсистемы ввода/вывода – **BIOS** (Basic Input/Output System). Коду BIOS известно почти все об устройствах, расположенных на материнской плате: о контроллере жестких дисков, плате сетевого адаптера, контроллере клавиатуры и других устройствах. В настройках BIOS можно выбирать, с какого устройства следует начинать загрузку ОС. После того как подсистема BIOS определила, с какого устройства ей следует загружать ОС, считываются первые 512 байт данного устройства. Эта часть диска называется **загрузочной записью** (**MBR** – Master Boot Record). В данной записи хранится программа **первичной загрузки** (stage1), которая сообщает компьютеру о том, в каком разделе диска расположена программа вторичной загрузки (stage2 – непосредственный загрузчик ОС). В данном курсе в качестве такого загрузчика рассматривается **GRUB** (GRand Unified Boot Loader). Помимо программы вторичной загрузки существует еще дополнитель-

ная программа (stage1.5)[1], помогающая загрузчику ОС определить некоторые параметры файловой системы. Это дополнительная программа используется в том случае, если программа вторичной загрузки не была установлена непосредственно в загрузочную запись MBR.

После того, как управление передастся программе вторичной загрузки ОС (stage2), загрузчик GRUB приступает к своей основной обязанности – загрузке ОС. Если среди опций загрузки ядра отсутствует параметр **quiet**, то в процессе загрузки система будет отображать служебную информацию на экран. Данную служебную информацию можно просмотреть не только в процессе загрузки ОС, но и используя команду **dmesg** или файл **/var/log/dmesg**.

Вывод служебной информации существенно зависит от оборудования, на котором установлена ОС Linux. В данной информации можно выделить следующие основные показатели:

- версия ядра ОС;
- количество распознанной оперативной памяти;
- количество процессоров (**CPU0**, **CPU1** и т.д.);
- статус режима работы механизма **SELinux**[2];
- параметры загрузки ядра ОС;
- количество свободной памяти отданной под начальный диск памяти (**initramfs**);
- жесткие диски и соответствующие им разделы (например, **/dev/sda**);
- сетевые платы (например, **eth0**);
- смонтированные файловые системы (например, **ext3**);
- разделы подкачки и их размер.

Анализ данной информации может помочь в случае возникновения проблем загрузки ОС Linux.

1.1.1. Универсальный загрузчик GRUB

Одним из вариантов загрузчика ОС Linux является программа **GRUB**. После того как программа, хранящаяся в загрузочной записи, сообщит компьютеру, что загрузчик ОС находится в разделе **/boot**, управление передается загрузчику ОС и на экран выводится его основное меню. Меню загрузчика GRUB позволяет настраивать различные варианты загрузки ядра ОС Linux, выбирать операционную систему, которая должна быть загружена, а также выполнять операции по восстановлению системы, используя собственный интерфейс командной строки. Основные команды, которые используются для управления загрузчиком GRUB, приведены табл. 1.1.

---

1 В версии загрузчика GRUB 2 данная программа отсутствует ввиду наличия собственного режима восстановления (rescue mode).

2 Механизм **SELinux** используется для контроля доступа приложения к ресурсам ОС.

Таблица 1.1. Основные команды работы в меню загрузчика GRUB

Команда	Описание
a	Используется для изменения параметров загрузки ядра.
b	Используется для загрузки ОС (выделенная запись меню GRUB).
c	Используется для входа на интерфейс командной строки.
d	Используется для удаления текущей записи меню GRUB.
e	Используется для редактирования меню GRUB.
o	Используется для вставки пустой строки под текущей записью в меню GRUB (режим редактирования меню GRUB).
O	Используется для вставки пустой строки над текущей записью в меню GRUB (режим редактирования меню GRUB).

Параметры меню загрузчика GRUB записываются в файл **/boot/grub/menu.lst**, который можно отредактировать после окончания процесса загрузки ОС. По умолчанию конфигурация загрузчика GRUB содержится в файле **/boot/grub/grub.conf**, однако в каталоге **/etc** существует символическая ссылка **/etc/grub.conf** на данный файл, так что редактирование конфигурации загрузчика возможно и через файл **/boot/grub/grub.conf**, и через ссылку **/etc/grub.conf**.

Загрузчик GRUB является универсальным загрузчиком и позволяет загружать не только ОС Linux, но и другие ОС, включая Windows, используя вызов сторонних загрузчиков.

Для изменения параметров[3] загрузки ядра используется команда **a**, присутствующая в меню GRUB. Например, если в конец строки загрузки ядра добавить параметр **single**, то система начнет загружаться в режиме **одного пользователя** (single-user). В случае если ОС некорректно распознает количество установленной оперативной памяти, необходимо добавить параметр **mem=XM**, где X обозначает количество мегабайт оперативной памяти.

В случае возникновения проблем с загрузкой ОС Linux в первую очередь необходимо проверить конфигурацию загрузчика GRUB, содержащуюся в файле **/boot/grub/grub.conf**. В следующем листинге приведено типовое содержание данного конфигурационного файла.

```
# grub.conf generated by anaconda
#
# Note that you do not have to rerun grub after making changes to this
file
# NOTICE:  You have a /boot partition.  This means that
#          all kernel and initrd paths are relative to /boot/, eg.
#          root (hd0,0)
#          kernel /vmlinuz-version ro root=/dev/sda2
#          initrd /initrd-version.img
#boot=/dev/sda
default=0
```

3 Полный список параметров ядра ОС Linux содержится по адресу http://www.kernel.org/doc/Documentation/kernel-parameters.txt

```
timeout=5
splashimage=(hd0,0)/grub/splash.xpm.gz
hiddenmenu
title Red Hat Enterprise Linux Server (2.6.18-53.el5)
        root (hd0,0)
        kernel /vmlinuz-2.6.18-53.el5 ro root=LABEL=/ rhgb quiet
        initrd /initrd-2.6.18-53.el5.img
```

В первой строке файла сообщается о том, что он был создан программой **Anaconda**, являющейся штатной программой-установщиком ОС Linux. В следующей строке содержится предупреждение, напоминающее о том, что изменения, сделанные в файле **grub.conf**, не будут записаны в загрузочную запись MBR, поскольку данный файл автоматически привязан к программе первичной загрузки stage1, расположенной в загрузочной записи MBR.

Строка со словом Notice появляется в том случае, если каталог **/boot** был сделан отдельным разделом. Каталог **/boot** ассоциирован с параметром **root**, в данном случае **root (hd0,0)**. Однако на самом деле корневой каталог / находится во втором разделе устройства **sda** и имеет абсолютный путь устройства **/dev/sda2**. Загрузчик GRUB использует свои правила именования физических устройств, отличающиеся от стандартных соглашений, принятых в ОС Linux. Имя устройства в загрузчике GRUB выглядит, например, так: **(hd0,0)**. Первое числовое значение определяет номер физического диска (нумерация ведется с нуля), а второе – номер раздела (правила нумерации аналогичны). В рассматриваемом листинге запись **(hd0,0)** соответствует первому разделу первого жесткого диска – **/dev/sda1**. К данному устройству монтируется каталог **/boot**, содержащий файл ядра ОС, образ памяти **initrd** и конфигурационные файлы загрузчика GRUB. Последняя закомментированная строка файла **grub.conf** сообщает, что загрузочным устройством, содержащим загрузочную запись MBR, является устройство **/dev/sda**.

После закомментированных строк записаны непосредственно команды загрузчика GRUB. Первая команда **default**, обозначает номер заголовка меню (**title**), который необходимо использовать для загрузки по умолчанию. Нумерация заголовков меню начинается с 0. В листинге присутствует лишь один заголовок **title**, при выборе которого будет осуществлена загрузка ОС **Red Hat Enterprise Linux Server (2.6.18-53.el5)**. Команда **timeout** обозначает время задержки в секундах, по истечении которого автоматически будет загружена ОС, указанная в директиве **default**. Директива **splashimage** указывает на файл, содержащий графические примитивы для отображения меню. Директива **hiddenmenu** обозначает, что по умолчанию меню скрыто от пользователя и вместо него на экране отображается строка вида **Booting Red Hat Enterprise Linux Server (2.6.18-53.el5) in 5 seconds...** Далее следуют команды описания расположения корневого раздела /, в данном случае это **(hd0,0)**, расположения файла ядра ОС Linux **vmlinuz-2.6.18-53.el5**, с указанием соответствующих параметров загрузки ядра, расположения файла ***начального образа памяти*** **initrd-2.6.18-53.el5.img**. Начальный образ памяти (Initial RAM disk) в процессе загрузки ОС создает временную файловую систему, содержащую модули ядра и программы, необходимые для монтирования файловых систем и запуска всех остальных системных процессов.

Параметры загрузки ядра указываются через пробел. По умолчанию используются следующие параметры **ro root=LABEL=/ rhgb quiet**.

Первый параметр сообщает о том, что каталог **/boot** открывается в режиме чтения: это делается с целью защиты раздела от случайной записи неверных данных. Второй параметр обозначает, что корневой каталог ассоциирован с меткой **/ (root=LABEL=/)**. Последние два параметра **rhgb** и **quiet** указывают на то, что, во-первых, следует использовать графическое меню загрузчика GRUB и, во-вторых, не отображать служебные сообщения в процессе загрузки ОС.

1.1.2. Работа с командами загрузчика GRUB

Загрузчик GRUB имеет собственный набор команд, предназначенный для его настройки. Запустить интерфейс для ввода данных команд можно из меню загрузчика, выполнив команду **c,** или непосредственно из командного интерпретатора, выполнив команду **grub**. Для просмотра всех возможных команд загрузчика GRUB используется клавиша **<TAB>**. Командный интерфейс загрузчика GRUB имеет базовые возможности, присутствующие в командном интерпретаторе bash; в частности, можно использовать автодополнение ввода и историю введенных команд.

В следующем листинге приведены примеры использования команд загрузчика GRUB.

```
grub> find /grub/stage1
 (hd0,0)
grub> root (hd0,<TAB>
 Possible partitions are:
   Partition num: 0,  Filesystem type is ext2fs, partition type 0x83
   Partition num: 1,  Filesystem type is ext2fs, partition type 0x83
   Partition num: 2,  Filesystem type unknown, partition type 0x82
   Partition num: 4,  Filesystem type unknown, partition type 0x8e
grub> root (hd0,1)
 Filesystem type is ext2fs, partition type 0x83
grub> root
 (hd0,1): Filesystem type is ext2fs, partition type 0x83
grub> find (hd0,0)/grub/grub.conf
 (hd0,0)
 (hd0,1)
grub> cat (hd0,1)/etc/redhat-release
Red Hat Enterprise Linux Server release 5.1 (Tikanga)
```

В первой строке листинга используется команда **find** для поиска устройства, содержащего программу первичной загрузки **starge1**. Далее при помощи команды **root** осуществляется поиск корневого раздела / (команда **root (hd0,<TAB>**), его ассоциация с физическим устройством **(hd0,1)** (команда **root (hd0,1)**) и вывода типа его файловой системы[4] (команда **root**). В предпоследней команде осуществляется поиск физических устройств, которые содержат файл **grub.conf**. Следует заметить, что в выводе данной команды содержится два устройства **(hd0,0)** и

4 Большинство типов файловых систем, записанных в в шестнадцатеричном формате, можно посмотреть по адресу http://www.essdatarecovery.com/hexcodes.asp или при помощи команды **fdisk**.

(hd0,1), соответствующие разделам **/boot** и /, поскольку файл **grub.conf** содержится в обоих разделах. Последняя команда используется для вывода содержимого текстового файла **/etc/redhat-release**, содержащего информацию о релизе ОС Linux.

Описанные выше особенности отчасти применимы к новой версии загрузчика GRUB – **GRUB2**, имеющей модульную архитектуру и обладающей большей переносимостью. Загрузчик **GRUB2** в настоящее время находится в активной разработке и имеет следующие особенности:

- Поддержка скриптования (циклы, переменные, функции);
- Динамическая загрузка модулей с жесткого диска при необходимости (RAID, LVM);
- Интернационализация (отображение не ASCII–символов);
- Собственный командный режим восстановления;
- Графический интерфейс.

Наиболее важные отличия **GRUB2** от предыдущей версии (**GRUB legacy**) следующие:

- Настройка меню загрузчика выполняется через файл **grub.cgf** (файл **menu.lst** отсутствует);
- Нумерация разделов начинается с 1 (**(hd0,1)** – соответствует первому разделу на жестком диске);
- Добавлены дополнительные команды[5] и изменены названия некоторых стандартных команд предыдущей версии.

Загрузчик **GRUB2** имеет следующие конфигурационные файлы и каталоги:

– **/etc/default/grub**[6] – позволяет устанавливать глобальные параметры загрузчика (например, при указании параметра **GRUB_CMDLINE_LINUX=acpi=off**, ко всем командам «**linux**» в файле **grub.cfg** будет добавлен параметр **acpi=off**).

– **/etc/grub.d** – содержит набор сценариев командной оболочки, необходимых для генерации файла **grub.cfg**. Здесь присутствуют сценарии, помогающие определить имеющиеся операционные системы, а также добавить собственные сценарии пользователя в меню загрузкичка. Порядок представления результатов сценариев в меню загрузчика определяется первыми числовыми символами в имени файла сценария (например, результат выполнения сценария **00_header**, будет присутствовать в меню первым).

– **/boot/grub/grub.cfg** – основной конфигурационный файл, содержащий информацию о загружаемых системах.

Для создания конфиграционного файла **grub.cfg** используется команда **grub-mkconfig**:

5 Сравнение команд загрузчиков GRUB2 и GRUB legacy представлено здесь: http://grub.enbug.org/CommandList

6 Дополнительные параметры файла **/etc/default**/grub описаны в приложении 1

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

В результате выполнения данной команды в каталоге **/boot/grub** будет создан файл **grub,cfg** примерно следующего содержания:

```
### BEGIN /etc/grub.d/00_header ###
set default=0
set timeout=5
set root=(hd0,3)
search --fs-uuid --set a1ae9df0-b8d3-4fe2-ad81-75a3c2f03c5d
if font /usr/share/grub/ascii.pff ; then
  set gfxmode=640x480
  insmod gfxterm
  insmod vbe
  terminal gfxterm
fi
### END /etc/grub.d/00_header ###
### BEGIN /etc/grub.d/05_debian_theme ###
set menu_color_normal=cyan/blue
set menu_color_highlight=white/blue
### END /etc/grub.d/05_debian_theme ###
### BEGIN /etc/grub.d/10_hurd ###
### END /etc/grub.d/10_hurd ###
### BEGIN /etc/grub.d/10_linux ###
menuentry "Debian GNU/Linux, linux 2.6.26-2-686" {
        set root=(hd0,2)
        search --fs-uuid --set dc47f706-7cee-4b03-8bd0-5efbf1996d9f
        linux   /vmlinuz-2.6.26-2-686 root=UUID=a1ae9df0-b8d3-4fe2-ad81-
75a3c2f03c5d ro
        initrd  /initrd.img-2.6.26-2-686
}
menuentry "Debian GNU/Linux, linux 2.6.26-2-686 (single-user mode)" {
        set root=(hd0,2)
        search --fs-uuid --set dc47f706-7cee-4b03-8bd0-5efbf1996d9f
        linux   /vmlinuz-2.6.26-2-686 root=UUID=a1ae9df0-b8d3-4fe2-ad81-
75a3c2f03c5d ro single
        initrd  /initrd.img-2.6.26-2-686
}
### END /etc/grub.d/10_linux ###
```

Как видно из листинга, формат записи изменился существенно: для задания параметров меню и указания загрузочного раздела (**root**), содержащего образ ядра и модули **GRUB**, используется команда **set**. Для поиска файловой системы, содержащей образ ядра, используется команда **search** (взамен команды **find**). Для загрузки ядра ОС в оперативную память используются команды **linux** или **multiboot** (замена команды **root**). Нетрудно заметить, что в конфигурационном файле присутствует оператор **if**, содержащий набор команд, загружающий встроенный терминал загрузчика **GRUB2** – **gfxterm**. Параметр **--fs-uuid** определяет уникальный идентификатор **UUID**[7] загрузочного раздела.

7 **Universally Unique Identifier** (UUID) используется для уникальной идентификации различной информации. Например, в случае использования **UUID** для устройств, перенос данных на другое устройство или смена номера устройства (LUN) не повлияет на корректную идентификацию загрузочного раздела в процессе загрузки ОС.

В случае изменения конфигурационных файлов в каталоге **/etc/grub.d** или в файле **/etc/default/grub,** следует выполнить команду **update-grub2** для обновления файла меню загрузчика. Данная команда на самом деле вызывает команду **grub-mkconfig -o /boot/grub/grub.cfg**, которая выполняет несколько специальных сценариев и формирует файл **/boot/grub/grub.cfg**.

Как было сказано ранее основная особенность загрузчика **GRUB2** – модульная архитектура. Для загрузки необходимых модулей используется команда **insmod <название_модуля>**. Например, если раздел **/boot** расположен поверх **LVM** тома, то в описании пункта меню в файле **grub.cfg** необходимо добавить команду загрузки модуля **lvm**:

```
insmod lvm
```

и указать загрузочный раздел, используя следующую нотацию:

```
set root=(lvm_группа-lvm_том)
```

Важной особенностью загрузчика **GRUB2** является наличие собственного режима восстановления rescue mode, использующегося в случае возникновения проблем загрузки основной части загрузчика – программы **stage2.** Если загрузчик не сможет считать параметры меню из файла **grub.cfg,** но сможет определить системные диски, пользователь попадает в нормальный командный режим, отличающийся следующим приглашением:

```
sh:grub>
```

Если в системе присутствуют более серьёзные проблемы, то загружается режим восстановления со следующим приглашением:

```
grub rescue>
```

Режим **rescue mode** является, по сути, усеченным нормальным режимом с ограниченным набором команд. Прежде чем пытаться что-то восстанавливать стоит попробовать загрузить нормальный режим, используя соответствующий модуль и переменную **prefix**, хранящуюся в **MBR** и содержащую расположение основной программы **stage2** и модулей:

```
grub rescue> set prefix=(hdX,Y)/grub
grub rescue> insmod (hdX,Y)/grub/normal.mod
rescue:grub> normal
```

В данном листинге X – соответствует номеру устройства, а Y – соответствует номеру раздела. Для расширения возможностей консоли необходимо загрузить модуль **linux.mod:**

```
grub rescue> insmod (hdX,Y)/grub/linux.mod
```

После этого станет доступна стандартная консоль GRUB2, в которой следует выполнить дальнейшую загрузку ОС:

```
set root=(hdX,Y)
linux /boot/vmlinuz-2.6.18-53.el5 ro root=LABEL=/
initrd /initrd-2.6.18-53.1.21.el5.img
boot
```

Данная последовательность команд стандартная в большинстве случаев: сначала определяется загрузочный раздел, содержащий образы ядра и ram-диска **initrd**, затем выполняется последовательная загрузка данных образов в память с указанием необходимх параметров и, наконец, дальнейшее управление процессом загрузки передается ядру при помощи команды **boot**.

1.2. Уровни выполнения

После того, как ядро было загружено загрузчиком **GRUB**, оно выполняет базовую настройку необходимых устройств, в том числе загрузку ram-диска **initrd**, содержащего драйверы устройств и позволяющего смонтировать корневой раздел. Затем загрузка ОС Linux продолжается вызовом процесса **init.** Данный процесс запускает скрипт **/etc/rc.d/rc.sysinit**, который осуществляет загрузку дополнительных модулей ядра (например, драйвера поддержки программных RAID-массивов), проверку файловой системы корневого раздела и монтирование корневого раздела в режиме чтения/записи. Далее процесс **init** на основании данных, содержащихся в файле **/etc/inittab**, определяет, какой уровень выполнения (runlevel) необходимо использовать, и последовательно продвигается от уровня 0 к уровню по умолчанию, указанному в файле **/etc/inittab** директивой `id:5:initdefault:`.

В файле **/etc/inittab** содержится информация обо всех уровнях выполнения, о виртуальных терминалах, загрузочных сценариях и других системных параметрах. ***Уровнями выполнения*** (runlevels) в ОС Linux называются несколько специальных режимов работы, на каждом из которых происходит запуск или остановка определенных процессов-демонов. Описание уровней выполнения ОС Linux приведено в табл. 1.2.

Таблица 1.2. Уровни выполнения ОС Linux

Уровень	Значение
0	Остановка системы.
1	Запуск ОС в режиме одного пользователя (single-user mode) без поддержки сетевых сервисов. Загрузка ОС и последующая работа производятся в режиме командной строки.
2	Запуск ОС в многопользовательском режиме без поддержки сетевых сервисов. Загрузка ОС и последующая работа производятся в режиме командной строки.
3	Запуск ОС в многопользовательском режиме с поддержкой сетевых сервисов. Загрузка ОС и последующая работа производятся в режиме командной строки.
4	Не используется.
5	Запуск ОС в многопользовательском режиме с поддержкой сетевых сервисов. Загрузка ОС и последующая работа производятся в графическом режиме.
6	Перезапуск системы.

Смена уровней выполнения скриптом **/etc/rc.d/rc**, который запускает все остальные системные процессы, посредством обращения к символьным ссылкам init-скриптов, содержащихся в каталогах **rc0.d** - **rc6.d**. Имена ссылок начинаются с префикса **S** или **K**, за которым следует порядковый номер и имя процесса-демона, управляемого данным init-скриптом (например, **S55sshd**). При переходе на заданный уровень выполнения, скрипт **rc** выполняет все скрипты с префиксом **S** в порядке возрастания порядковых номеров, причем выполнение скриптов осуществляется с параметром **start**. При выходе с текущего уровня выполняются все скрипты с префиксом **K** в порядке возрастания порядковых номеров, причем выполнение скриптов осуществляется с параметром **stop**. Например, при переходе с 3-го на 5-й уровень выполняются все скрипты каталога **/etc/rc.d/rc5.d** с префиксом **K**, а затем выполняются все скрипты с префиксом **S** того же каталога.

В следующем листинге приведен частичный вывод каталога **/etc/rc.d/rc3.d**, который соответствует третьему функциональному уровню загрузки. Из данного листинга видно, что демон **sshd** запускается под номером **55** после того, как все предыдущие сервисы, содержащие префикс **S**, будут запущены.

```
lrwxrwxrwx 1 root root 16 Dec  9 17:19 K02dhcdbd -> ../init.d/dhcdbd
lrwxrwxrwx 1 root root 14 Dec  9 17:21 S26hidd -> ../init.d/hidd
lrwxrwxrwx 1 root root 16 Dec  9 17:22 S28autofs -> ../init.d/autofs
lrwxrwxrwx 1 root root 15 Dec  9 17:21 S44acpid -> ../init.d/acpid
lrwxrwxrwx 1 root root 15 Dec  9 17:21 S50hplip -> ../init.d/hplip
lrwxrwxrwx 1 root root 14 Dec  9 17:20 S55sshd -> ../init.d/sshd
```

Часть строк файла **/etc/inittab** представлена в следующем листинге.

```
id:5:initdefault:
si::sysinit:/etc/rc.d/rc.sysinit
l0:0:wait:/etc/rc.d/rc 0
. . . . . . . . . . . .
l6:6:wait:/etc/rc.d/rc 6
ca::ctrlaltdel:/sbin/shutdown -t3 -r now
pf::powerfail:/sbin/shutdown -f -h +2 "Power Failure; System Shutting Down"
pr:12345:powerokwait:/sbin/shutdown -c "Power Restored; Shutdown Cancelled"
1:2345:respawn:/sbin/mingetty tty1
. . . . . . . . . . . . . . . . .
6:2345:respawn:/sbin/mingetty tty6
x:5:respawn:/etc/X11/prefdm -nodaemon
```

В первой **строке** содержится директива определения функционального уровня по умолчанию (**initdefault**). Из данного листинга видно, что по умолчанию система функционирует на уровне 5, который ассоциирован с работой в графическом режиме. Каждая строка файла **/etc/inittab** записывается в определенном формате и состоит из четырех полей:

идентификатор:уровень:действие:команда

- ***идентификатор*** – поле, состоящее из 1-4 символов, обозначающих функционал данной записи.
- ***уровень*** – поле, содержащее список уровней, для которых применяется

указанная команда. Например, запись 345 означает выполнение команды на уровнях 3, 4 и 5.

- ***действие*** – специальное действие, которое должен выполнить процесс **init** прежде чем перейти к считыванию следующей строки файла **/etc/inittab**. Например, запись **wait** говорит о том, что процесс **init** должен дождаться завершения выполнения указанной команды. Запись **respawn** говорит о том, что процесс **init** должен повторно выполнить команду в случае ее завершения.
- ***команда*** – в данном поле указана команда, которую необходимо выполнить, со всеми ее аргументами и опциями.

Помимо определения уровней выполнения, в файле **/etc/inittab** содержится определение ряда дополнительных команд и виртуальных терминалов. Первая из команд определяет, каким образом в системе будет осуществляться перехват клавиш **<CTRL+ALT+DELETE>**:

```
ca::ctrlaltdel:/sbin/shutdown -t3 -r now
```

Следующие две команды определяют поведение системы в случае получения аварийного сигнала powerfail (SIGPWR):

```
pf::powerfail:/sbin/shutdown -f -h +2 "Power Failure; System Shutting Down"
pr:12345:powerokwait:/sbin/shutdown -c "Power Restored; Shutdown Cancelled"
```

Далее в файле **/etc/inittab** следует определение виртуальных терминалов. ***Виртуальный терминал*** – это сеанс командной строки, в котором выполняется основная работа с системой. По умолчанию, в ОС Linux определено шесть виртуальных терминалов, однако это количество может быть увеличено до 12. Поскольку ОС Linux является многопользовательской системой, несколько разных пользователей могут одновременно работать из нескольких терминалов, включая графическую оболочку.

Переключение между терминалами осуществляется при помощи клавиш **<ALT+Fn>**, где **n** обозначает номер виртуального терминала (от 1 до 12). Например, для переключения на третий виртуальный терминал необходимо нажать **<ALT+F3>**.

Последним звеном процесса загрузки ОС Linux является скрипт **/etc/rc.d/rc.local**, который предназначен для добавления пользовательских сценариев запуска.

Управление уровнями выполнения является очень важным аспектом администрирования ОС Linux. Вы должны уметь настраивать уровень выполнения, используемый системой по умолчанию, а также определять, какие из системных процессов должны запускаться автоматически на данном уровне.

По умолчанию, если при инсталляции была установлена графическая оболочка, ОС Linux загружается на 5-й уровень выполнения, о чем свидетельствует запись **id:5:initdefault:** файла **/etc/inittab**. Для изменения уровня по умолчанию необходимо изменить второе поле данной записи. Сделанные измене-

ния вступят в силу только после перезагрузки системы. Для того чтобы изменить текущий уровень выполнения без перезагрузки системы необходимо запустить следующую команду от пользователя **root**:

```
init <уровень>
```

Параметр **уровень** задается числом от 0 до 6.

Для настройки автозапуска процессов в ОС Linux используются следующие команды и утилиты:

- chkconfig;
- ntsysv;
- service configuration tool.

Команда **chkconfig** может использоваться как для просмотра текущей конфигурации уровней выполнения, так и для их настройки и предоставляет наиболее простой механизм осуществления следующих действий:

- просмотр текущих настроек сервисов;
- изменение текущих настроек сервисов;
- добавление/удаление сервисов из структуры каталогов **/etc/rc.d**;
- включение/отключение автоматического запуска сервисов.

Команда **chkconfig** имеет следующий синтаксис:

```
chkconfig --list [имя сервиса]
chkconfig --add <имя сервиса>
chkconfig --del <имя сервиса>
chkconfig [--level <уровни>] <имя сервиса> <on|off|reset>
```

Необязательный параметр `имя сервиса` обозначает название системного процесса или сервиса, сценарий которого присутствует в каталоге **/etc/rc.d/init.d**, причем имя сервиса должно совпадать с названием соответствующего сценария.

Опция `--list` используется для просмотра текущих настроек сервиса.

Опции `--add` и `--del` используются для добавления и удаления символьных ссылок сервисов в структуру каталогов /etc/rc.d.

Опция `--level` определяет, для каких уровней выполнения необходимо выполнить указанные далее действия.

Параметры `on, off, reset` используются для изменения статуса запуска сервиса.

При указании параметра **off** сервис исключается из автозапуска в процессе загрузки ОС. При указании параметра **on** сервис запускается в процессе загрузки ОС. При указании параметра **reset** статус запуска сервиса устанавливается согласно изначальной конфигурации, указанной в файле сценария.

В следующем листинге приведены примеры использования команды **chkconfig** для управления сервисами.

```
# chkconfig --list|grep sendmail
sendmail        0:off   1:off   2:on    3:on    4:on    5:on    6:off
# chkconfig --level 245 sendmail off
```

```
# chkconfig --list|grep sendmail
sendmail        0:off   1:off   2:off   3:on    4:off   5:off   6:off
# chkconfig sendmail reset
# chkconfig --list|grep sendmail
sendmail        0:off   1:off   2:on    3:on    4:on    5:on    6:off
```

В первой строке осуществляется вывод текущих настроек запуска сервиса **sendmail**, из которых видно, что данный сервис запускается на уровнях **2**, **3**, **4** и **5**. Во второй строке осуществляется отключение автозапуска сервиса **sendmail** на уровнях **2**, **4** и **5**. Далее при помощи параметра **reset** осуществляется возврат первоначальных настроек автозапуска сервиса **sendmail** согласно записи (# chkconfig: 2345 80 30), указанной в файле сценария **/etc/rc.d/init.d/sendmail**.

Для добавления нового сервиса в структуру каталогов **/etc/rc.d** необходимо, чтобы фай сценария сервиса имел как минимум две обязательные записи:

1. **# chkconfig: 2345 80 30**
   Числовое значение, следующее за символом «**:**», говорит о том, что в каталогах **/etc/rc.d/rc2.d**, **/etc/rc.d/rc3.d**, **/etc/rc.d/rc4.d** и **/etc/rc.d/rc5.d** (уровнях **2**, **3**, **4** и **5**) необходимо создать символьные ссылки на данный сценарий, причем, ссылкам с префиксом **S** будет присвоен порядковый номер **80**, а ссылкам с префиксом **K** – порядковый номер **30**.
2. **# description: <текст>**
   В данной записи дается формальное описание сервиса.

Например, для сервиса **sendmail** записи, считываемые утилитой **chkconfig**, имеют следующий вид:

```
# chkconfig: 2345 80 30
# description: Sendmail is a Mail Transport Agent, which is the program \
# that moves mail from one machine to another.
```

После создания файла сценария сервиса, содержащего записи **chkconfig**, его необходимо скопировать в каталог **/etc/init.d** и сделать его исполняемым. Затем, используя команду **chkconfig –add <название_сценария>**, создать символьные ссылки в соответствующих каталогах **/etc/rc.d/rc*.d**.

Для управления автозапуском сервисов в интерактивном режиме используется команда **ntsysv**, имеющая следующий синтаксис:

```
ntsysv [--back] [--level <уровни>]
```

Для указания уровней выполнения, на которых необходимо настроить автозапуск сервисов, используется необязательная опция **--level**. В случае запуска данной команды без параметров на терминале будет отображено окно настройки сервисов для текущего функционального уровня.

Для управления сервисами в графическом режиме используется программа **service configuration tool**. Данную программу можно запустить непосредственно из командной строки, выполнив команду **system-config-services**, или из графической оболочки, выбрав ярлык **Services**, расположенный на вкладке **System → Administration → Services**.

Конец ознакомительного фрагмента.
Приобрести книгу можно
в интернет-магазине «Электронный универс»
(e-Univers.ru)