Администрирование систем Linux

Оригинал: Автор: Paul Cobbaut Дата публикации: 12 марта 2015 г. Перевод: А. Панин Дата перевода: 3 апреля 2015 г.

Оглавление

Глава 1. Вводная информация об управлении процессами

1.1. Терминология

1.1.1. Процесс

Процесс является скомпилированным исходным кодом, исполняющимся в текущее время в системе.

1.1.2. Идентификатор процесса PID

Каждый из процессов имеет идентификатор или PID (сокращение process id).

1.1.3. Идентификатор родительского процесса PPID

Каждый процесс имеет родительский процесс (который идентифицируется с помощью идентификатора родительского процесса или PPID). Дочерний процесс обычно запускается средствами родительского процесса.

1.1.4. Процесс init

Процесс init всегда имеет идентификатор PID, равный 1. Процесс init запускается средствами ядра операционной системы и, следовательно, не имеет родительского процесса. Процесс init

исполняет обязанности приемного родителя для осиротевших процессов.

1.1.5. Уничтожение процесса

Завершение работы процесса чаще всего достигается благодаря его уничтожению, поэтому в том случае, если вы хотите остановить исполнение процесса, вы должны уничтожить его.

1.1.6. Демон

Процессы, которые запускаются в момент загрузки системы и исполняются в течение всего времени функционирования системы, называются процессами-демонами или просто демонами. Демоны никогда не должны самостоятельно завершать свою работу.

1.1.7. Зомби

Уничтоженный процесс, информация о котором сохраняется в рамках системы, называется зомби-процессом. Вы не можете уничтожить зомби процессы, так как они уже были уничтожены ранее и не исполняются.

1.2. Базовые приемы управления процессами

1.2.1. Переменные командной оболочки $$ и $PPID

Некоторые переменные командной оболочки содержат информацию о процессах. Переменная будет хранить идентификатор используемого вами в данный момент процесса командной оболочки, а переменная $PPID — идентификатор соответствующего родительского процесса. На самом деле является параметром командной оболочки, а не переменной, так как вы не можете установить значение этого параметра.

Ниже приведен пример использования утилиты echo для вывода значений $$ и $PPID.

```
[paul@RHEL4b ~]$ echo $$ $PPID
4224 4223
```

1.2.2. Утилита pidof

Вы можете получить все идентификаторы процессов на основе известного имени процесса, воспользовавшись утилитой pidof.

```
root@rhel53 ~# pidof mingetty
2819 2798 2797 2796 2795 2794
```

1.2.3. Родительские и дочерние процессы

Иерархия процессов системы строится на отношении родительский процесс — дочерний процесс. Каждый процесс в системе имеет родительский процесс.

При запуске нового экземпляра командной оболочки bash вы можете воспользоваться утилитой echo, чтобы убедиться в том, что идентификатор процесса предыдущего экземпляра командной оболочки совпадает с идентификатором родительского процесса нового экземпляра командной оболочки. Таким образом, используемый ранее дочерний процесс в данном случае становится родительским процессом.

```
[paul@RHEL4b ~]$ bash
[paul@RHEL4b ~]$ echo $$ $PPID
4812 4224
```

Ввод команды exit приведет к завершению исполнения текущего процесса и возврату в командную оболочку с оригинальными значениями $$ и $PPID.

```
[paul@RHEL4b ~]$ echo $$ $PPID
4812 4224
[paul@RHEL4b ~]$ exit
exit
[paul@RHEL4b ~]$ echo $$ $PPID
4224 4223
[paul@RHEL4b ~]$
```

1.2.4. Системные вызовы fork и exec

Процесс инициирует исполнение другого процесса в два этапа. На первом этапе он создает свою точную копию с помощью системного вызова fork. После этого в рамках копии процесса выполняется системный вызов exec, предназначенный для замены скопированных данных родительского процесса на данные дочернего процесса.

```
[paul@RHEL4b ~]$ echo $$
```

4224
[paul@RHEL4b ~]$ bash
[paul@RHEL4b ~]$ echo $$ $PPID
5310 4224
[paul@RHEL4b ~]$

1.2.5. Команда exec

Благодаря команде exec вы можете инициировать исполнение нового процесса, минуя этап создания точной копии текущего процесса. В следующем примере осуществляется запуск командной оболочки Korn Shell (ksh) с последующей заменой на командную оболочку bash с помощью команды exec. Идентификатор процесса командной оболочки bash остается равным идентификатору процесса командной оболочки Korn Shell. При этом после завершения работы с дочерней командной оболочкой bash происходит перемещение в родительскую командную оболочку bash, а не в командную оболочку Korn shell (процесса которой более не существует).

[paul@RHEL4b ~]$ echo $$
4224 # PID процесса bash
[paul@RHEL4b ~]$ ksh
$ echo $$ $PPID
5343 4224 # PID процессов ksh и bash
$ exec bash
[paul@RHEL4b ~]$ echo $$ $PPID
5343 4224 # PID процессов bash и bash
[paul@RHEL4b ~]$ exit
exit
[paul@RHEL4b ~]$ echo $$
4224

1.2.6. Утилита ps

Одним из наиболее часто используемых инструментов для ознакомления со списком процессов в Linux является утилита ps. В следующем примере показаны отношения между тремя родительскими и дочерними процессами bash.

[paul@RHEL4b ~]$ echo $$ $PPID
4224 4223
[paul@RHEL4b ~]$ bash
[paul@RHEL4b ~]$ echo $$ $PPID
4866 4224
[paul@RHEL4b ~]$ bash
[paul@RHEL4b ~]$ echo $$ $PPID
4884 4866
[paul@RHEL4b ~]$ ps fx
PID TTY STAT TIME COMMAND
4223 ? S 0:01 sshd: paul@pts/0
4224 pts/0 Ss 0:00 _ -bash
4866 pts/0 S 0:00 _ bash
4884 pts/0 S 0:00 _ bash
4902 pts/0 R+ 0:00 _ ps fx
[paul@RHEL4b ~]$ exit
exit
[paul@RHEL4b ~]$ ps fx
PID TTY STAT TIME COMMAND
4223 ? S 0:01 sshd: paul@pts/0
4224 pts/0 Ss 0:00 _ -bash
4866 pts/0 S 0:00 _ bash
4903 pts/0 R+ 0:00 _ ps fx
[paul@RHEL4b ~]$ exit
exit
[paul@RHEL4b ~]$ ps fx
PID TTY STAT TIME COMMAND
4223 ? S 0:01 sshd: paul@pts/0
4224 pts/0 Ss 0:00 _ -bash
4904 pts/0 R+ 0:00 _ ps fx
[paul@RHEL4b ~]$

В Linux обычно используется команда ps fax. В Solaris же чаще всего используется команда ps -ef (которая также работает в Linux). Ниже приведен фрагмент вывода команды ps fax.

[paul@RHEL4a ~]$ ps fax
PID TTY STAT TIME COMMAND
1 ? S 0:00 init [5]
...
3713 ? Ss 0:00 /usr/sbin/sshd
5042 ? Ss 0:00 _ sshd: paul [priv]
5044 ? S 0:00 _ sshd: paul@pts/1
5045 pts/1 Ss 0:00 _ -bash
5077 pts/1 R+ 0:00 _ ps fax

1.2.7. Утилита pgrep

По аналогии с командой ps -C, вы также можете использовать утилиту pgrep для поиска информации о процессе по его имени.

[paul@RHEL5 ~]$ sleep 1000 &
[1] 32558
[paul@RHEL5 ~]$ pgrep sleep
32558
[paul@RHEL5 ~]$ ps -C sleep
PID TTY TIME CMD
32558 pts/3 00:00:00 sleep

Кроме того, вы можете осуществлять вывод списка найденных

идентификаторов процессов с соответствующими им именами процессов.

paul@laika:~$ pgrep -l sleep
9661 sleep

1.2.8. Утилита top

Другим популярным инструментом для получения информации о процессах в Linux является утилита top. Утилита top может упорядочивать список процессов в соответствии с создаваемой ими нагрузкой на центральный процессор, а также с другими параметрами. Кроме того, вы можете уничтожать процессы средствами утилиты top. Нажмите клавишу h в процессе работы с утилитой top для ознакомления со справочной информацией.

В случае неполадок утилита top является первым инструментом, который следует применять для диагностики системы, так как помимо информации о процессах она может предоставить информацию об использовании оперативной памяти и раздела подкачки вашей системы.

1.3. Отправка сигналов процессам

1.3.1. Утилита kill

Утилита kill позволяет уничтожить процесс (или остановить его исполнение). В примере ниже показана стандартная методика использования утилиты kill для остановки исполнения процесса с идентификатором 1942.

paul@ubuntu910:~$ kill 1942
paul@ubuntu910:~$

На самом деле в процессе использования утилиты kill осуществляется передача сигнала процессу.

1.3.2. Список сигналов

Исполняющиеся процессы могут принимать сигналы от других процессов, а также от пользователей. Вы можете получить список сигналов, воспользовавшись командой kill -l, причем в качестве параметра должна использоваться строчная буква l, а не число 1.

[paul@RHEL4a ~]$ kill -l
1) SIGHUP 2) SIGINT 3) SIGQUIT 4) SIGILL
5) SIGTRAP 6) SIGABRT 7) SIGBUS 8) SIGFPE
9) SIGKILL 10) SIGUSR1 11) SIGSEGV 12) SIGUSR2
13) SIGPIPE 14) SIGALRM 15) SIGTERM 17) SIGCHLD
18) SIGCONT 19) SIGSTOP 20) SIGTSTP 21) SIGTTIN
22) SIGTTOU 23) SIGURG 24) SIGXCPU 25) SIGXFSZ
26) SIGVTALRM 27) SIGPROF 28) SIGWINCH 29) SIGIO
30) SIGPWR 31) SIGSYS 34) SIGRTMIN 35) SIGRTMIN+1
36) SIGRTMIN+2 37) SIGRTMIN+3 38) SIGRTMIN+4 39) SIGRTMIN+5
40) SIGRTMIN+6 41) SIGRTMIN+7 42) SIGRTMIN+8 43) SIGRTMIN+9
44) SIGRTMIN+10 45) SIGRTMIN+11 46) SIGRTMIN+12 47) SIGRTMIN+13
48) SIGRTMIN+14 49) SIGRTMIN+15 50) SIGRTMAX-14 51) SIGRTMAX-13
52) SIGRTMAX-12 53) SIGRTMAX-11 54) SIGRTMAX-10 55) SIGRTMAX-9
56) SIGRTMAX-8 57) SIGRTMAX-7 58) SIGRTMAX-6 59) SIGRTMAX-5
60) SIGRTMAX-4 61) SIGRTMAX-3 62) SIGRTMAX-2 63) SIGRTMAX-1
64) SIGRTMAX
[paul@RHEL4a ~]$

1.3.3. Команда kill -1 (SIGHUP)

Первый сигнал SIGHUP (или HUP, или 1) часто используется в Linux для сообщения процессу о необходимости повторного чтения файла конфигурации. Таким образом, команда kill -1 1 приведет к принудительному повторному чтению файла конфигурации процессом init (процесс init всегда имеет идентификатор 1).

root@deb503:~# kill -1 1
root@deb503:~#

Решение о том, должен ли процесс осуществлять повторное чтение файла конфигурации в процессе работы или после перезапуска, принимается исключительно разработчиком соответствующего приложения. Перед отправкой данного сигнала пользователю следует ознакомиться с документацией, поставляемой совместно с приложением.

1.3.4. Команда kill -15 (SIGTERM)

Сигнал SIGTERM также часто называется стандартным сигналом уничтожения процесса. При использовании утилиты kill без указания сигнала предполагается, что исполняется команда kill -s 15.

Команды в примере ниже являются идентичными.

paul@ubuntu910:~$ kill 1942
paul@ubuntu910:~$ kill -15 1942

1.3.5. Команда kill -9 (SIGKILL)

Сигнал SIGKILL отличается от большинства сигналов тем, что он отправляется не процессу, а непосредственно ядру Linux. Команда kill -9 также называется надежной командной уничтожения процесса. Ядро операционной системы гарантированно завершит исполнение указанного процесса. Как разработчик вы не можете ни коим образом перехватить сигнал, генерируемый в результате исполнения команды kill -9.

root@rhel53 ~# kill -9 3342

1.3.6. Сигналы SIGSTOP и SIGCONT

Исполнение процесса может быть приостановлено в случае приема сигнала SIGSTOP. Данный сигнал может быть отправлен процессу с помощью команды kill -19 в Linux, причем в других

системах Unix номер сигнала может отличаться.

Приостановленный процесс не использует циклов центрального процессора, но находится в памяти и может быть реанимирован с помощью сигнала SIGCONT (отправляемого с помощью команды kill -18 в Linux).

Оба этих сигнала будут использоваться в главе, посвященной описанию процессов, исполняющихся в фоновом режиме.

1.3.7. Утилита pkill

Вы можете использовать утилиту pkill для завершения работы процесса на основе его имени.

```
[paul@RHEL5 ~]$ sleep 1000 &
[1] 30203
[paul@RHEL5 ~]$ pkill sleep
[1]+ Завершено sleep 1000
[paul@RHEL5 ~]$
```

1.3.8. Утилита killall

Утилита killall позволяет отправлять сигнал 15 всем процессам с определенным именем.

```
paul@rhel65:~$ sleep 8472 &
[1] 18780
paul@rhel65:~$ sleep 1201 &
[2] 18781
paul@rhel65:~$ jobs
[1]- Running sleep 8472 &
[2]+ Running sleep 1201 &
paul@rhel65:~$ killall sleep
[1]- Завершено sleep 8472
[2]+ Завершено sleep 1201
paul@rhel65:~$ jobs
paul@rhel65:~$
```

1.3.9. Утилита killall5

Аналог описанной выше утилиты из мира SysV, утилита killall5, может использоваться в процессе завершения работы системы. В данном примере показано, как утилита killall5 используется в дистрибутиве Red Hat Enterprise Linux 5.3 для уничтожения процессов при завершении работы системы.

```
root@rhel53 ~# grep killall /etc/init.d/halt
action $"Sending all processes the TERM signal..." /sbin/killall5 -15
action $"Sending all processes the KILL signal..." /sbin/killall5 -9
```

1.3.10. Утилита top

При работе с утилитой top кнопка k позволит вам выбрать сигнал и идентификатор процесса, которому этот сигнал будет отправлен. В примере ниже приведена строка, выводимая под строками с общей информацией о состоянии системы после нажатия клавиши k.

```
PID to signal/kill [default pid = 977] 1932
Send pid 1932 signal [15/sigterm] 9
```

1.4. Практическое задание: вводная информация об управлении процессами

1. Используйте утилиту ps для поиска информации о процессе с именем init.
2. Каков идентификатор процесса с именем init?
3. Используйте команду who i am для установления имени пользователя, под которым вы работаете с терминалом.
4. Располагая полученной ранее информацией о вашем имени пользователя, примените утилиту ps для поиска всех процессов, запущенных с помощью вашего терминала.
5. Каков идентификатор процесса вашей командной оболочки?
6. Каков идентификатор родительского процесса вашей командной оболочки?
7. Запустите в фоновом режиме два экземпляра процесса с помощью команд sleep 3342.
8. Определите идентификаторы всех процессов с именем sleep.
9. Выведите информацию только о двух запущенных ранее процессах sleep с помощью утилиты top. После этого завершите работу утилиты top.
10. Используйте стандартный сигнал уничтожения процесса для уничтожения одного из двух созданных ранее процессов с именами sleep.
11. Используйте одну команду для уничтожения всех процессов с именем sleep.

1.5. Корректная процедура выполнения практического задания: вводная информация об управлении процессами

1. Используйте утилиту ps для поиска информации о процессе с именем init.

```
root@rhel53 ~# ps -C init
PID TTY TIME CMD
1 ? 00:00:04 init
```

1. Каков идентификатор процесса с именем init?

1

1. Используйте команду who i am для установления имени пользователя, под которым вы работаете с терминалом.

```
root@rhel53 ~# who am i
paul pts/0 2010-04-12 17:44 (192.168.1.38)
```

1. Располагая полученной ранее информацией о вашем имени пользователя, примените утилиту ps для поиска всех процессов, запущенных с помощью вашего терминала.

oot@rhel53 ~# ps fax | grep pts/0
2941 ? S 0:00 _ sshd: paul@pts/0
2942 pts/0 Ss 0:00 _ -bash
2972 pts/0 S 0:00 _ su -
2973 pts/0 S 0:00 _ -bash
3808 pts/0 R+ 0:00 _ ps fax
3809 pts/0 R+ 0:00 _ grep pts/0
или может быть использована аналогичная команда
root@rhel53 ~# ps -ef | grep pts/0
paul 2941 2939 0 17:44 ? 00:00:00 sshd: paul@pts/0
paul 2942 2941 0 17:44 pts/0 00:00:00 -bash
root 2972 2942 0 17:45 pts/0 00:00:00 su -
root 2973 2972 0 17:45 pts/0 00:00:00 -bash
root 3816 2973 0 21:25 pts/0 00:00:00 ps -ef
root 3817 2973 0 21:25 pts/0 00:00:00 grep pts/0

1. Каков идентификатор процесса вашей командной оболочки?

2973 в примере выше, в вашем случае наверняка будет использоваться отличный идентификатор процесса

Команда echo $$ позволит получить аналогичное найденному значение

1. Каков идентификатор родительского процесса вашей командной оболочки?

2972 в примере выше, в вашем случае наверняка будет использоваться отличный идентификатор родительского процесса

В данном примере родительским процессом является процесс su -, в то время, как при работе в окружении рабочего стола gnome родительским процессом может оказаться процесс gnome-terminal

1. Запустите в фоновом режиме с помощью команд sleep 3342 два экземпляра процесса.

sleep 3342 &
sleep 3342 &

1. Определите идентификаторы всех процессов с именем sleep.

pidof sleep

1. Выведите информацию только о двух запущенных ранее процессах sleep с помощью утилиты top. После этого завершите работу утилиты top.

top -p pidx,pidy (замените pidx, pidy на действительные значения идентификаторов процессов)

1. Используйте стандартный сигнал уничтожения процесса для уничтожения одного из двух созданных ранее процессов с именами sleep.

kill pidx

1. Используйте одну команду для уничтожения всех процессов с именем sleep.

pkill sleep

Глава 2. Приоритеты процессов

2.1. Приоритеты процессов и значения nice

2.1.1. Введение

Все процессы в системе имеют соответствующие приоритеты и значения nice. Процессы с более высокими приоритетами будут получать больше процессорного времени, нежели остальные процессы с низкими приоритетами. Вы можете повлиять на распределение процессорного времени, воспользовавшись утилитами nice и renice.

2.1.2. Именованные программные каналы (утилита mkfifo)

Процессы могут обмениваться информацией друг с другом посредством именованных программных каналов. Эти именованные программные каналы могут создаваться с помощью утилиты mkfifo.

В примере ниже показана процедура создания четырех отдельных именованных программных каналов (в новой директории).

paul@ubuntu910:~$ mkdir procs
paul@ubuntu910:~$ cd procs/
paul@ubuntu910:~/procs$ mkfifo pipe33a pipe33b pipe42a pipe42b
paul@ubuntu910:~/procs$ ls -l
итого 0
prw-r—r-- 1 paul paul 0 2010-04-12 13:21 pipe33a
prw-r—r-- 1 paul paul 0 2010-04-12 13:21 pipe33b
prw-r—r-- 1 paul paul 0 2010-04-12 13:21 pipe42a
prw-r—r-- 1 paul paul 0 2010-04-12 13:21 pipe42b
paul@ubuntu910:~/procs$

2.1.3. Увлекательные манипуляции с утилитой cat

Для демонстрации методики использования утилит top и renice мы будем применять комбинацию из утилиты cat и созданных ранее именованных программных каналов с целью полного исчерпания вычислительных ресурсов центрального процессора.

Бинарный файл cat должен быть скопирован в текущую директорию с отличным именем. (Это позволит нам достаточно просто отслеживать созданные процессы с помощью утилиты top. Вы можете выполнить эти же действия и без копирования бинарного файла cat, но с использованием различных учетных записей пользователей. Или же вы можете просто отслеживать

значения идентификаторов каждого из созданных процессов.)

```
paul@ubuntu910:~/procs$ cp /bin/cat proj33
paul@ubuntu910:~/procs$ cp /bin/cat proj42
paul@ubuntu910:~/procs$ echo -n x | ./proj33 - pipe33a > pipe33b &
[1] 1670
paul@ubuntu910:~/procs$ ./proj33 <pipe33b >pipe33a &
[2] 1671
paul@ubuntu910:~/procs$ echo -n z | ./proj42 - pipe42a > pipe42b &
[3] 1673
paul@ubuntu910:~/procs$ ./proj42 <pipe42b >pipe42a &
[4] 1674
```

Команды, которые вы можете обнаружить в примере выше, предназначены для создания двух процессов с именами proj33, которые на самом деле будут являться экземплярами утилиты cat, выполняющими циклическую передачу символа x с помощью именованных программных каналов pipe33a и pipe33b. Те же самые манипуляции осуществляются процессом с именем proj42 с символом z.

2.1.4. Утилита top

Простой запуск утилиты top без передачи каких-либо параметров или аргументов приведет к выводу информации о параметрах системы и обо всех процессах в системе. При этом в верхней части вывода утилиты top может быть обнаружена следующая информация.

```
top - 13:59:29 up 48 min, 4 users, load average: 1.06, 0.25, 0.14
Tasks: 139 total, 3 running, 136 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.3 us, 99.7 sy, 0.0 ni, 0.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
Mem: 509352k total, 460040k used, 49312k free, 66752k buffers
Swap: 746980k total, 0k used, 746980k free, 247324k cached
```

Обратите внимание на нулевое значение времени простоя центрального процессора (cpu idle time 0.0 id). Оно объясняется тем, что наши процессы cat используют все вычислительные ресурсы центрального процессора. Результаты могут отличаться при использовании систем с четырьмя или большим количеством ядер центрального процессора.

2.1.5. Команда top -p

Команда top -p 1670,1671,1673,1674 из примера ниже позволяет выводить информацию о четырех процессах, каждый из которых использует примерно 25 процентов вычислительных ресурсов центрального процессора.

```
paul@ubuntu910:~$ top -p 1670,1671,1673,1674
PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
1674 paul 20 0 2972 616 524 S 26.6 0.1 0:11.92 proj42
1670 paul 20 0 2972 616 524 R 25.0 0.1 0:23.16 proj33
1671 paul 20 0 2972 616 524 S 24.6 0.1 0:23.07 proj33
1673 paul 20 0 2972 620 524 R 23.0 0.1 0:11.48 proj42
```

Все четыре процесса имеют одинаковый приоритет (столбец PR) и на равных соревнуются за время центрального процессора. В некоторых системах ядро Linux может использовать немного отличающиеся значения приоритетов процессов, но в любом случае в результате описанных выше манипуляций четыре процесса будут на равных соревноваться за время центрального процессора.

2.1.6. Утилита renice

Так как упомянутые ранее процессы уже исполняются, нам придется использовать утилиту renice для изменения их значений nice (столбец NI).

В примере ниже показана методика использования утилиты renice по отношению к обоим процессам с именами proj33.

```
paul@ubuntu910:~$ renice +8 1670
1670 (process ID) old priority 0, new priority 8
paul@ubuntu910:~$ renice +8 1671
1671 (process ID) old priority 0, new priority 8
```

Обычные пользователи могут устанавливать значения nice для своих процессов из диапазона от 0 до 20. Отрицательные значения nice может устанавливать исключительно пользователь root. Будьте осторожны при использовании отрицательных значений nice, так как в случае их установки управление системой с помощью клавиатуры или удаленной командной оболочки, работающей по протоколу ssh, может быть значительно осложнено или же вообще окажется невозможным.

2.1.7. Влияние значений nice на работу процессов

Влияние значения nice на работу процесса может может варьироваться. В примере ниже показан результат исполнения нашей команды renice +8. Обратите внимание на значения доли процессорного времени из столбца %CPU для каждого из процессов.

```
PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
1674 paul 20 0 2972 616 524 S 46.6 0.1 0:22.37 proj42
1673 paul 20 0 2972 620 524 R 42.6 0.1 0:21.65 proj42
1671 paul 28 8 2972 616 524 S 5.7 0.1 0:29.65 proj33
1670 paul 28 8 2972 616 524 R 4.7 0.1 0:29.82 proj33
```

Важно помнить о том, что значения nice менее важных процессов всегда должны быть выше соответствующих значений более важных процессов. При этом использование отрицательных значений nice может оказать негативное влияние на стабильность работы системы.

2.1.8. Утилита nice

Утилита nice функционирует аналогично утилите renice, но применяется в момент запуска приложения.

В примере ниже показана методика исполнения сценария с значением nice, равным пяти.

paul@ubuntu910:~$ nice -5 ./backup.sh

2.2. Практическое задание: приоритеты процессов

1. Создайте новую директорию и шесть именованных программных каналов в ней.
2. Инициируйте циклическую передачу символа посредством двух именованных программных каналов.
3. Используйте утилиты top и ps для вывода информации (pid, ppid, приоритет, значение nice, ...) о двух созданных процессах cat.
4. Инициируйте циклическую передачу другого символа посредством двух других именованных программных каналов, но в этот раз используйте утилиту nice для запуска приложений. Проверьте, соревнуются ли созданные процессы за все вычислительные ресурсы центрального процессора. (Не забывайте о том, что в случае необходимости вы можете создать еще два или большее количество процессов cat, использующих оставшиеся программные каналы).
5. Используйте утилиту ps для установления факта использования двумя новыми процессами cat измененных значений nice. Для этого используйте параметры -o и -C утилиты ps.
6. Используйте утилиту renice для увеличения значения nice процессов с 10 до 15. Обратите внимание на отличия в работе данных процессов и процессов со стандартными значениями nice.

2.3. Корректная процедура выполнения практического задания: приоритеты процессов

1. Создайте новую директорию и шесть именованных программных каналов в ней.

```
[paul@rhel53 ~]$ mkdir pipes ; cd pipes
[paul@rhel53 pipes]$ mkfifo p1 p2 p3 p4 p5 p6
[paul@rhel53 pipes]$ ls -l
итого 0
prw-rw-r-- 1 paul paul 0 Апр 12 22:15 p1
prw-rw-r-- 1 paul paul 0 Апр 12 22:15 p2
prw-rw-r-- 1 paul paul 0 Апр 12 22:15 p3
prw-rw-r-- 1 paul paul 0 Апр 12 22:15 p4
prw-rw-r-- 1 paul paul 0 Апр 12 22:15 p5
prw-rw-r-- 1 paul paul 0 Апр 12 22:15 p6
```

1. Инициируйте циклическую передачу символа посредством двух именованных программных каналов.

```
[paul@rhel53 pipes]$ echo -n x | cat - p1 > p2 &
[1] 4013
[paul@rhel53 pipes]$ cat <p2 >p1 &
[2] 4016
```

1. Используйте утилиты top и ps для вывода информации (pid, ppid, приоритет, значение nice, ...) о двух созданных процессах cat.

```
top (скорее всего, две верхних строки)
[paul@rhel53 pipes]$ ps -C cat
PID TTY TIME CMD
4013 pts/0 00:03:38 cat
4016 pts/0 00:01:07 cat
[paul@rhel53 pipes]$ ps fax | grep cat
4013 pts/0 R 4:00 | \_ cat - p1
4016 pts/0 S 1:13 | \_ cat
4044 pts/0 S+ 0:00 | \_ grep cat
```

1. Инициируйте циклическую передачу другого символа посредством двух других именованных программных каналов, но в этот раз используйте утилиту nice для запуска приложений. Проверьте, соревнуются ли созданные процессы за все вычислительные ресурсы центрального процессора. (Не забывайте о том, что в случае необходимости вы можете создать еще два или большее количество процессов cat, использующих оставшиеся программные каналы).

```
echo -n y | nice cat - p3 > p4 &
nice cat <p4 >p3 &
```

1. Используйте утилиту ps для установления факта использования двумя новыми процессами cat измененных значений nice. Для этого используйте параметры -o и -C утилиты ps.

```
[paul@rhel53 pipes]$ ps -C cat -o pid,ppid,pri,ni,comm
PID PPID PRI NI COMMAND
4013 3947 14 0 cat
4016 3947 21 0 cat
4025 3947 13 10 cat
4026 3947 13 10 cat
```

1. Используйте утилиту renice для увеличения значения nice процессов с 10 до 15. Обратите внимание на отличия в работе данных процессов и процессов со стандартными значениями nice.

```
[paul@rhel53 pipes]$ renice +15 4025
4025 (process ID) old priority 10, new priority 15
```

[paul@rhel53 pipes]$ renice +15 4026
4026 (process ID) old priority 10, new priority 15
[paul@rhel53 pipes]$ ps -C cat -o pid,ppid,pri,ni,comm
PID PPID PRI NI COMMAND
4013 3947 14 0 cat
4016 3947 21 0 cat
4025 3947 9 15 cat
4026 3947 8 15 cat

Глава 3. Фоновые задачи

3.1. Фоновые процессы

3.1.1. Команда jobs

Список задач, запущенных с помощью вашей текущей командной оболочки и выполняющихся в фоновом режиме, может быть выведен с помощью команды jobs. По умолчанию у вас не должно быть каких-либо задач, выполняющихся в фоновом режиме.

root@rhel53 ~# jobs
root@rhel53 ~#

Данная команда jobs будет использоваться несколько раз на протяжении данной главы.

3.1.2. Сочетание клавиш Control-Z

Исполнение некоторых процессов может быть остановлено с помощью комбинации клавиш Ctrl-Z. Данная комбинация клавиш позволяет отправить сигнал SIGSTOP ядру Linux, что приведет к остановке исполнения процесса.

В том случае, если данная комбинация клавиш используется при работе с текстовым редактором vi(m), процесс текстового редактора vi(m) продолжает исполняться в фоновом режиме. Процесс vi(m), исполняющийся в фоновом режиме, может быть обнаружен в выводе команды jobs.

[paul@RHEL4a ~]$ vi procdemo.txt
[5]+ Stopped vim procdemo.txt
[paul@RHEL4a ~]$ jobs
[5]+ Stopped vim procdemo.txt

3.1.3. Амперсанд (&)

Процессы, которые запускаются в фоновом режиме благодаря символу &, размещенному в конце строки команды, также отображаются в выводе команды jobs.

[paul@RHEL4a ~]$ find / > allfiles.txt 2> /dev/null &
[6] 5230
[paul@RHEL4a ~]$ jobs
[5]+ Stopped vim procdemo.txt
[6]- Running find / >allfiles.txt 2>/dev/null &
[paul@RHEL4a ~]$

3.1.4. Команда jobs -p

Команда jobs поддерживает интересный параметр -p, предназначенный для вывода идентификаторов процессов, исполняющихся в фоновом режиме.

[paul@RHEL4b ~]$ sleep 500 &
[1] 4902
[paul@RHEL4b ~]$ sleep 400 &
[2] 4903
[paul@RHEL4b ~]$ jobs -p
4902
4903
[paul@RHEL4b ~]$ ps jobs -p
PID TTY STAT TIME COMMAND
4902 pts/0 S 0:00 sleep 500
4903 pts/0 S 0:00 sleep 400
[paul@RHEL4b ~]$

3.1.5. Команда fg

Вызов команды fg позволяет перевести процесс из фонового режима в текущую командную оболочку. Номер фоновой задачи для переноса в текущую командную оболочку является параметром команды fg.

[paul@RHEL5 ~]$ jobs
[1] Running sleep 1000 &
[2]- Running sleep 1000 &
[3]+ Running sleep 2000 &
[paul@RHEL5 ~]$ fg 3
sleep 2000

3.1.6. Команда bg

Задачи из фонового режима, исполнение которых было приостановлено, могут быть снова запущены в фоновом режиме с помощью команды bg. Команда bg осуществляет отправку сигнала SIGCONT соответствующему процессу.

Ниже приведен пример запуска процесса sleep в фоновом режиме (исполнение которого было остановлено с помощью сочетания клавиш Ctrl-Z) с последующей повторной активацией этого процесса в фоновом режиме с помощью команды bg.

[paul@RHEL5 ~]$ jobs
[paul@RHEL5 ~]$ sleep 5000 &
[1] 6702
[paul@RHEL5 ~]$ sleep 3000
[2]+ Stopped sleep 3000
[paul@RHEL5 ~]$ jobs

[1]- Running sleep 5000 &
[2]+ Stopped sleep 3000
[paul@RHEL5 ~]$ bg 2
[2]+ sleep 3000 &
[paul@RHEL5 ~]$ jobs
[1]- Running sleep 5000 &
[2]+ Running sleep 3000 &
[paul@RHEL5 ~]$

3.2. Практическое задание: фоновые процессы

1. Используйте команду jobs, чтобы удостовериться в том, что никакие процессы не выполняются в фоновом режиме.
2. Используйте текстовый редактор vi для создания небольшого текстового файла. Переведите процесс vi в фоновый режим и остановите его исполнение.
3. Используйте команду jobs, чтобы убедиться в том, что исполнение процесса vi в фоновом режиме остановлено.
4. Выполните команду find / > allfiles.txt 2>/dev/null в фоновом режиме. Остановите исполнение этой команды в фоновом режиме перед тем, как будет закончен обход файловой системы.
5. Запустите два процесса sleep с длительным периодом ожидания в фоновом режиме.
6. Выведите информацию обо всех задачах, исполняющихся в фоновом режиме.
7. Используйте утилиту kill для остановки исполнения одного из процессов с именем sleep.
8. Инициируйте продолжение исполнения процесса find в фоновом режиме (убедитесь в том, что исполнение продолжилось).
9. Переведите один из процессов с именем sleep назад в текущую командную оболочку.
10. (Общий вопрос, ответ может быть дан при наличии свободного времени...) Дайте детальные пояснения по поводу источников чисел из следующего примера. В каких условиях переменные заменяются на их значения? Какая командная оболочка осуществляет эту замену?

[paul@RHEL4b ~]$ echo $$ $PPID
4224 4223
[paul@RHEL4b ~]$ bash -c "echo $$ $PPID"
4224 4223
[paul@RHEL4b ~]$ bash -c echo $$ $PPID
5059 4224
[paul@RHEL4b ~]$ bash -c echo $$ $PPID
4223: 4224: команда не найдена...

3.3. Корректная процедура выполнения практического задания: фоновые процессы

1. Используйте команду jobs, чтобы удостовериться в том, что никакие процессы не выполняются в фоновом режиме.

jobs (возможно одни из процессов cat все еще исполняется?)

1. Используйте текстовый редактор vi для создания небольшого текстового файла. Переведите процесс vi в фоновый режим и остановите его исполнение.

vi text.txt
(в процессе работы с vi нажмите ctrl-z)

1. Используйте команду jobs, чтобы убедиться в том, что исполнение процесса vi в фоновом режиме остановлено.

[paul@rhel53 ~]$ jobs
[1]+ Stopped vim text.txt

1. Выполните команду find / > allfiles.txt 2>/dev/null в фоновом режиме. Остановите исполнение этой команды в фоновом режиме перед тем, как будет закончен обход файловой системы.

[paul@rhel53 ~]$ find / > allfiles.txt 2>/dev/null
(нажмите ctrl-z)
[2]+ Stopped find / > allfiles.txt 2> /dev/null

1. Запустите два процесса sleep с длительным периодом ожидания в фоновом режиме.

sleep 4000 & ; sleep 5000 &

1. Выведите информацию обо всех задачах, исполняющихся в фоновом режиме.

[paul@rhel53 ~]$ jobs
[1]- Stopped vim text.txt
[2]+ Stopped find / > allfiles.txt 2> /dev/null
[3] Running sleep 4000 &
[4] Running sleep 5000 &

1. Используйте утилиту kill для остановки исполнения одного из процессов с именем sleep.

[paul@rhel53 ~]$ kill -SIGSTOP 4519
[paul@rhel53 ~]$ jobs
[1] Stopped vim text.txt
[2]- Stopped find / > allfiles.txt 2> /dev/null
[3] Running sleep 4000 &
[4]+ Stopped sleep 5000

1. Инициируйте продолжение исполнения процесса find в фоновом режиме (убедитесь в том, что исполнение продолжилось).

bg 2 (проверьте состояние задачи с указанным идентификатором в списке фоновых задач)

1. Переведите один из процессов с именем sleep назад в текущую командную оболочку.

fg 3 (и снова проверьте состояние задачи с указанным идентификатором в списке фоновых задач)

1. (Общий вопрос, ответ может быть дан при наличии свободного времени...) Дайте детальные пояснения по поводу источников чисел из следующего примера. В каких условиях переменные заменяются на их значения? Какая командная оболочка осуществляет эту замену?

```
[paul@RHEL4b ~]$ echo $$ $PPID
4224 4223
[paul@RHEL4b ~]$ bash -c "echo $$ $PPID"
4224 4223
[paul@RHEL4b ~]$ bash -c echo $$ $PPID
5059 4224
[paul@RHEL4b ~]$ bash -c echo $$ $PPID
4223: 4224: команда не найдена...
```

Текущая командная оболочка bash будет заменять имена переменных $$ и $PPID при разборе строки команды перед исполнением команды echo.

```
[paul@RHEL4b ~]$ echo $$ $PPID
4224 4223
```

Теперь имена переменных экранированы с помощью двойных кавычек, но текущая командная оболочка bash все также будет заменять имена переменных $$ и $PPID при разборе строки команды перед исполнением команды bash -c.

```
[paul@RHEL4b ~]$ bash -c "echo $$ $PPID"
4224 4223
```

А теперь имена переменных экранированы с помощью одинарных кавычек. Текущая командная оболочка bash не будет заменять имена переменных и $PPID. Команда bash -c будет выполнена перед заменой имен переменных на их значения. Вторая же командная оболочка bash будет осуществлять замену имен переменных и $PPID на их значения.

```
[paul@RHEL4b ~]$ bash -c echo $$ $PPID
5059 4224
```

При использовании обратных кавычек командная оболочка все также будет осуществлять замену имен переменных на их значения перед исполнением встроенной команды echo. Результатом исполнения этой команды echo будут являться идентификаторы двух процессов. Эти идентификаторы будут переданы команде bash -c. Но эти два числа не являются корректными командами!

```
[paul@RHEL4b ~]$ bash -c echo $$ $PPID
4223: 4224: команда не найдена...
```

Глава 4. Дисковые устройства

Прочитав данную главу, вы научитесь обнаруживать и идентифицировать жесткие диски. Данная информация необходима для подготовки к чтению следующей главы, в которой мы будем размещать разделы на этих устройствах.

4.1. Терминология

4.1.1. Пластина, головка, дорожка, цилиндр, сектор

В дисковых накопителях данные обычно хранятся на магнитных или оптических пластинах (disk platters). Пластины вращаются (с высокими скоростями). Данные считываются с помощью головок (heads), которые расположены очень близко к поверхности пластин, но не прикасаются к ним! Головки смонтированы на держателе (иногда называемом приводом или вилкой).

Данные записываются по концентрическим окружностям, называемым дорожками (tracks). Нулевой трек (обычно) располагается возле внешней стороны пластины. Время, необходимое для позиционирования головки над определенной дорожкой, называется временем перехода к дорожке (seek time). Обычно пластины располагаются одна над другой, поэтому набор дорожек, доступный при определенном позиционировании держателя головки формирует цилиндр (cylinder). Дорожки разделены на секторы по 512 байт с дополнительным неиспользуемым пространством (разрывом) между секторами на внешней стороне пластины.

При рассмотрении составляющих времени доступа (access time) для жесткого диска вы можете отметить, что большая часть времени уходит на позиционирование головок (около 65 %) и задержки вращения пластин (roatational latency) (около 30 %).

4.1.2. Интерфейсы IDE или SCSI

На самом деле следовало бы использовать заголовок «интерфейсы ATA или SCSI», так как устройство IDE является ATA-совместимым устройством. В большинстве настольных компьютеров используются устройства ATA, в большинстве серверов — устройства SCSI.

4.1.3. Интерфейс ATA

Контроллер ATA позволяет использовать два устройства на каждой из шин, причем одно из устройств должно быть ведущим, а другое — ведомым. В том случае, если ваш контроллер не поддерживает возможность автоматического распределения ролей устройств «cable select», вам придется производить соответствующие настройки самостоятельно с помощью джамперов.

После представления общественности интерфейса SATA (Serial

ATA) оригинальный интерфейс ATA был переименован в интерфейс Parallel ATA. Устройства чтения оптических дисков обычно используют интерфейс ATAPI, который на самом деле является интерфейсом ATA, использующим протокол обмена данными SCSI.

4.1.4. Интерфейс SCSI

Контроллер SCSI позволяет использовать более двух устройств. При использовании интерфейса SCSI (Small Computer System Interface) каждое устройство получает уникальный идентификатор scsi id. Контроллеру SCSI также требуется идентификатор scsi id, причем вы не должны использовать данный идентификатор для подключенного к контроллеру устройства SCSI.

Старый 8-битный интерфейс SCSI в наши дни называют narrow SCSI, а 16-битный интерфейс - wide SCSI. После удвоения частоты шины до 10MHz интерфейс стал называться fast SCSI. А после следующего удвоения частоты шины до 20MHz было введено название ultra SCSI. Обратитесь к статье https://ru.wikipedia.org/wiki/SCSI для ознакомления с остальными стандартами интерфейса SCSI.

4.1.5. Блочное устройство

Для жестких дисков с возможностью произвольного доступа к данным в рамках операционной системы реализован уровень абстракции под названием «блочное устройство» (block device), позволяющий осуществлять форматирование этих дисков с использованием блоков фиксированного размера (обычно равного 512 байтам). При этом доступ к каждому из блоков устройства может осуществляться независимо от доступа к любым другим блокам устройства.

```
[root@centos65 ~]# lsblk
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
sda 8:0 0 40G 0 disk
--sda1 8:1 0 500M 0 part /boot
--sda2 8:2 0 39.5G 0 part
--VolGroup-lv_root (dm-0) 253:0 0 38.6G 0 lvm /
--VolGroup-lv_swap (dm-1) 253:1 0 928M 0 lvm [SWAP]
sdb 8:16 0 72G 0 disk
sdc 8:32 0 144G 0 disk
```

Для обозначения файлов блочных устройств в выводе команды ls -l используется символ b.

```
[root@centos65 ~]# ls -l /dev/sd*
brw-rw----. 1 root disk 8, 0 Апр 19 10:12 /dev/sda
brw-rw----. 1 root disk 8, 1 Апр 19 10:12 /dev/sda1
brw-rw----. 1 root disk 8, 2 Апр 19 10:12 /dev/sda2
brw-rw----. 1 root disk 8, 16 Апр 19 10:12 /dev/sdb
brw-rw----. 1 root disk 8, 32 Апр 19 10:12 /dev/sdc
```

Обратите внимание на то, что в выводе команды ls -l символьные устройства, позволяющие работать исключительно с непрерывными потоками символов, обозначаются с помощью символа c. Также следует помнить о том, что оптический диск CDROM, соответствующий стандарту ISO 9660, использует блоки, размер каждого из которых равен 2048 байтам.

В старых жестких дисках (а также в дискетах) для доступа к секторам используется система адресации на основе номеров цилиндров, головок и секторов. При этом в большинстве современных дисков используется система адресации LBA (Logical Block Addressing).

4.1.6. Твердотельные накопители

Твердотельный накопитель или SSD (Solid State Drive) является блочным устройством без движущихся частей. Его можно сравнить с флеш-памятью. Устройства SSD превосходят в цене жесткие диски, но при этом обычно характеризуются гораздо меньшими значениями времени доступа к секторам.

В рамках данной книги для обозначения дисков с вращающимся шпинделем будут использоваться пиктограммы коричневого цвета, а для обозначения твердотельных накопителей — пиктограммы синего цвета.

4.2. Имена файлов устройств

4.2.1. Имена файлов устройств ATA (IDE)

Пути к файлам всех устройств ATA в вашей системе будут начинаться с /dev/hd с последующим буквенным символом. Ведущий жесткий диск, подключенный к первому контроллеру ATA будет представлен в системе с помощью файла устройства /dev/hda, ведомый — с помощью файла устройства /dev/hdb. Именами файлов устройств, подключенных ко второму контроллеру, будут /dev/hdc и /dev/hdd.

Таблица 4.1. Имена файлов устройств IDE

Контроллер	Роль устройства	Имя файла устройства
IDE 0	Ведущее	/dev/hda
	Ведомое	/dev/hdb
IDE 1	Ведущее	/dev/hdc
	Ведомое	/dev/hdd

Вполне вероятна ситуация, в которой в вашей системе будут созданы только файлы устройств /dev/hda и /dev/hdd. Первый файл

будет представлять единственный жесткий диск, а второй — привод для чтения дисков CDROM (по умолчанию настроенный как ведомое устройство).

4.2.2. Имена файлов устройств SCSI

Для файлов устройств SCSI используется аналогичная схема, но все пути к этим файлам начинаются с /dev/sd. В том случае, если для представления устройств, используемых в вашей системе, не хватит букв английского алфавита (устройство, представленное файлом /dev/sdz, не будет являться последним), в именах файлов устройств будет использоваться более одной буквы: /dev/sdaa, /dev/sdab и так далее. (Также позднее мы увидим, что тома LVM обычно представлены файлами устройств /dev/md0, /dev/md1, и.т.д.).

Ниже приведен пример распределения имен файлов устройств в Linux между устройствами SCSI. Добавление диска SCSI или RAID-контроллера с меньшим значением идентификатора scsi id приведет к изменению схемы распределения имен устройств (сдвигу вперед на одну букву из алфавита для устройств с большими значениями идентификаторов scsi id).

Таблица 4.2. Имена файлов устройств SCSI

Устройство	Идентификатор scsi id	Имя файла устройства
диск 0	0	/dev/sda
диск 1	1	/dev/sdb
RAID-контроллер 0	5	/dev/sdc
RAID-контроллер 1	6	/dev/sdd

В современной системе Linux файлы устройств /dev/sd* будут использоваться для представления устройств SCSI и SATA, а также карт памяти SD, накопителей с интерфейсом USB, устройств ATA/IDE (устаревших) и твердотельных накопителей.

4.3. Определение параметров дисковых устройств

4.3.1. Утилита fdisk

Процедура определения параметров дисковых устройств может быть начата с установления определенных ядром ОС типов устройств с помощью утилиты /sbin/fdisk. Ниже представлен результат использования данной утилиты на старом настольном компьютере с двумя дисками ATA/IDE, работающем под управлением дистрибутива Debian.

root@barry:~# fdisk -l | grep Disk
Disk /dev/hda: 60.0 GB, 60022480896 bytes
Disk /dev/hdb: 81.9 GB, 81964302336 bytes

А ниже приведен пример описания параметров дисков SATA и SCSI, которые установлены в сервере, работающем под управлением дистрибутива CentOS. Помните о том, что для представления дисков SATA также используется нотация имен файлов устройств, принятая для дисков SCSI, а именно, /dev/sd*.

[root@centos65 ~]# fdisk -l | grep Disk /dev/sd
Disk /dev/sda: 42.9 GB, 42949672960 bytes
Disk /dev/sdb: 77.3 GB, 77309411328 bytes
Disk /dev/sdc: 154.6 GB, 154618822656 bytes
Disk /dev/sdd: 154.6 GB, 154618822656 bytes

Ниже приведено описание параметров двух реальных дисков SCSI объемом по 72 ГБ, которые установлены в сервере, работающем под управлением дистрибутива RHEL4u3. Данный сервер соединен с сервером сетевого хранилища (NAS) с четырьмя дисками объемом в половину терабайта каждый. На дисках сервера сетевого хранилища размещены четыре раздела LVM (/dev/mdx), предназначенные для формирования программного RAID-массива.

[root@tsvtl1 ~]# fdisk -l | grep Disk
Disk /dev/sda: 73.4 GB, 73407488000 bytes
Disk /dev/sdb: 73.4 GB, 73407488000 bytes
Disk /dev/sdc: 499.0 GB, 499036192768 bytes
Disk /dev/sdd: 499.0 GB, 499036192768 bytes
Disk /dev/sde: 499.0 GB, 499036192768 bytes
Disk /dev/sdf: 499.0 GB, 499036192768 bytes
Disk /dev/md0: 271 MB, 271319040 bytes
Disk /dev/md2: 21.4 GB, 21476081664 bytes
Disk /dev/md3: 21.4 GB, 21467889664 bytes
Disk /dev/md1: 21.4 GB, 21476081664 bytes

Также вы можете использовать утилиту fdisk для получения информации о каждом отдельном дисковом устройстве.

[root@centos65 ~]# fdisk -l /dev/sdc
Disk /dev/sdc: 154.6 GB, 154618822656 bytes
255 heads, 63 sectors/track, 18798 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000000

Позднее мы будем использовать утилиту fdisk для выполнения таких опасных операций, как создание и удаление разделов жестких дисков.

4.3.2. Утилита dmesg

Сообщения, выводимые при загрузке ядра ОС, могут быть изучены и после загрузки ОС благодаря наличию утилиты dmesg. Ввиду того, что дисковые устройства определяются ядром ОС в процессе загрузки системы, вы также можете использовать утилиту dmesg для поиска информации о дисковых устройствах.

```
[root@centos65 ~]# dmesg | grep sd[a-z] | head
sd 0:0:0:0: [sda] 83886080 512-byte logical blocks: (42.9 GB/40.0 GiB)
sd 0:0:0:0: [sda] Write Protect is off
sd 0:0:0:0: [sda] Mode Sense: 00 3a 00 00
sd 0:0:0:0: [sda] Write cache: enabled, read cache: enabled, doesn't support \
DPO or FUA
sda: sda1 sda2
sd 0:0:0:0: [sda] Attached SCSI disk
sd 3:0:0:0: [sdb] 150994944 512-byte logical blocks: (77.3 GB/72.0 GiB)
sd 3:0:0:0: [sdb] Write Protect is off
sd 3:0:0:0: [sdb] Mode Sense: 00 3a 00 00
sd 3:0:0:0: [sdb] Write cache: enabled, read cache: enabled, doesn't support \
DPO or FUA
```

А ниже приведен другой пример фрагмента вывода утилиты dmesg, полученного при работе с компьютером, оснащенным жестким диском ATA объемом 200 ГБ.

```
paul@barry:~$ dmesg | grep -i "ata disk"
[ 2.624149] hda: ST360021A, ATA DISK drive
[ 2.904150] hdb: Maxtor 6Y080L0, ATA DISK drive
[ 3.472148] hdd: WDC WD2000BB-98DWA0, ATA DISK drive
```

В последнем, третьем примере приведен фрагмент вывода утилиты dmesg, полученный при работе с дистрибутивом RHEL 5.3.

```
root@rhel53 ~# dmesg | grep -i "scsi disk"
sd 0:0:2:0: Attached scsi disk sda
sd 0:0:3:0: Attached scsi disk sdb
sd 0:0:6:0: Attached scsi disk sdc
```

4.3.3. Утилита /sbin/lshw

Утилита lshw позволяет получить список параметров аппаратного обеспечения (название утилиты расшифровывается как «list hardware» - «получение списка параметров аппаратного обеспечения»). При передаче соответствующих параметров утилита lshw может вывести большой объем информации о дисковых устройствах (и созданных на этих устройствах разделах).

Ниже приведен сокращенный пример вывода данной утилиты, полученный при работе с дистрибутивом Debian 6:

```
root@debian6~# lshw -class volume | grep -A1 -B2 scsi
description: Linux raid autodetect partition
physical id: 1
bus info: scsi@1:0.0.0,1
logical name: /dev/sdb1
description: Linux raid autodetect partition
physical id: 1
bus info: scsi@2:0.0.0,1
logical name: /dev/sdc1
description: Linux raid autodetect partition
physical id: 1
bus info: scsi@3:0.0.0,1
logical name: /dev/sdd1
description: Linux raid autodetect partition
physical id: 1
bus info: scsi@4:0.0.0,1
logical name: /dev/sde1
vendor: Linux
physical id: 1
bus info: scsi@0:0.0.0,1
logical name: /dev/sda1
vendor: Linux
physical id: 2
bus info: scsi@0:0.0.0,2
logical name: /dev/sda2
description: Extended partition
physical id: 3
bus info: scsi@0:0.0.0,3
logical name: /dev/sda3
```

Данная утилита не поставляется в составе дистрибутивов Redhat и CentOS (но вы можете подключить дополнительный репозиторий программного обеспечения и установить ее).

4.3.4. Утилита /sbin/lsscsi

Утилита lsscsi выводит информацию обо всех устройствах SCSI (а также об эмулируемых устройствах SCSI) в удобном для чтения формате. В первом примере показан вывод утилиты lsscsi при работе с системой SPARC.

```
root@shaka:~# lsscsi
[0:0:0:0] disk Adaptec RAID5 V1.0 /dev/sda
[1:0:0:0] disk SEAGATE ST336605FSUN36G 0438 /dev/sdb
root@shaka:~#
```

Ниже приведен пример вывода утилиты lsscsi при работе с сервером сетевого хранилища QNAP NAS (в котором установлено четыре диска объемом 750 ГБ, причем сам сервер поддерживает

возможность загрузки операционной системы с накопителя с интерфейсом USB).

```
lroot@debian6~# lsscsi
[0:0:0:0] disk SanDisk Cruzer Edge 1.19 /dev/sda
[1:0:0:0] disk ATA ST3750330AS SD04 /dev/sdb
[2:0:0:0] disk ATA ST3750330AS SD04 /dev/sdc
[3:0:0:0] disk ATA ST3750330AS SD04 /dev/sdd
[4:0:0:0] disk ATA ST3750330AS SD04 /dev/sde
```

А в этом примере приведен классический вывод утилиты lsscsi.

```
root@debian6~# lsscsi -c
Attached devices:
Host: scsi0 Channel: 00 Target: 00 Lun: 00
Vendor: SanDisk Model: Cruzer Edge Rev: 1.19
Type: Direct-Access ANSI SCSI revision: 02
Host: scsi1 Channel: 00 Target: 00 Lun: 00
Vendor: ATA Model: ST3750330AS Rev: SD04
Type: Direct-Access ANSI SCSI revision: 05
Host: scsi2 Channel: 00 Target: 00 Lun: 00
Vendor: ATA Model: ST3750330AS Rev: SD04
Type: Direct-Access ANSI SCSI revision: 05
Host: scsi3 Channel: 00 Target: 00 Lun: 00
Vendor: ATA Model: ST3750330AS Rev: SD04
Type: Direct-Access ANSI SCSI revision: 05
Host: scsi4 Channel: 00 Target: 00 Lun: 00
Vendor: ATA Model: ST3750330AS Rev: SD04
Type: Direct-Access ANSI SCSI revision: 05
```

4.3.5. Файл /proc/scsi/scsi

Другой способ обнаружения дисковых устройств SCSI (или SATA) заключается в чтении файла /proc/scsi/scsi.

В данном примере показано содержимое упомянутого файла в системе SPARC с контроллером RAID 5 производства компании Adaptec.

```
root@shaka:~# cat /proc/scsi/scsi
Attached devices:
Host: scsi0 Channel: 00 Id: 00 Lun: 00
Vendor: Adaptec Model: RAID5 Rev: V1.0
Type: Direct-Access ANSI SCSI revision: 02
Host: scsi1 Channel: 00 Id: 00 Lun: 00
Vendor: SEAGATE Model: ST336605FSUN36G Rev: 0438
Type: Direct-Access ANSI SCSI revision: 03
root@shaka:~#
```

А это вывод команды cat /proc/scsi/scsi для упомянутого выше сервера сетевого хранилища QNAP (работающего под управлением Debian Linux).

```
root@debian6~# cat /proc/scsi/scsi
Attached devices:
Host: scsi0 Channel: 00 Id: 00 Lun: 00
Vendor: SanDisk Model: Cruzer Edge Rev: 1.19
Type: Direct-Access ANSI SCSI revision: 02
Host: scsi1 Channel: 00 Id: 00 Lun: 00
Vendor: ATA Model: ST3750330AS Rev: SD04
Type: Direct-Access ANSI SCSI revision: 05
Host: scsi2 Channel: 00 Id: 00 Lun: 00
Vendor: ATA Model: ST3750330AS Rev: SD04
Type: Direct-Access ANSI SCSI revision: 05
Host: scsi3 Channel: 00 Id: 00 Lun: 00
Vendor: ATA Model: ST3750330AS Rev: SD04
Type: Direct-Access ANSI SCSI revision: 05
Host: scsi4 Channel: 00 Id: 00 Lun: 00
Vendor: ATA Model: ST3750330AS Rev: SD04
Type: Direct-Access ANSI SCSI revision: 05
```

Обратите внимание на то, что в последних версиях дистрибутива Debian данная возможность деактивирована на уровне ядра ОС. Вы можете активировать ее (перед компиляцией ядра ОС) с помощью данного параметра:

```
# CONFIG_SCSI_PROC_FS is not set
```

В дистрибутивах Redhat и CentOS данный файл присутствует по умолчанию (в случае наличия устройств SCSI).

```
[root@centos65 ~]# cat /proc/scsi/scsi
Attached devices:
Host: scsi0 Channel: 00 Id: 00 Lun: 00
Vendor: ATA Model: VBOX HARDDISK Rev: 1.0
Type: Direct-Access ANSI SCSI revision: 05
Host: scsi3 Channel: 00 Id: 00 Lun: 00
Vendor: ATA Model: VBOX HARDDISK Rev: 1.0
Type: Direct-Access ANSI SCSI revision: 05
Host: scsi4 Channel: 00 Id: 00 Lun: 00
Vendor: ATA Model: VBOX HARDDISK Rev: 1.0
Type: Direct-Access ANSI SCSI revision: 05
```

4.4. Очистка жесткого диска

Перед продажей вашего старого жесткого диска через торговую площадку в сети Интернет следует удалить все хранившиеся на нем ранее данные. После простого изменения таблицы разделов, применения утилиты format из состава ОС Microsoft Windows или даже после применения утилиты mkfs некоторые специалисты все еще будут иметь возможность прочитать большую часть данных, хранившихся на диске.

```
root@debian6~# aptitude search foremost autopsy sleuthkit | tr -s ' '
```

p autopsy - graphical interface to SleuthKit
p foremost - Forensics application to recover data
p sleuthkit - collection of tools for forensics analysis

Хотя в техническом плане утилита /sbin/badblocks и предназначена для отслеживания поврежденных секторов на диске, вы также можете использовать ее для полного удаления всех данных с диска. Так как данная утилита в прямом смысле записывает данные в каждый сектор диска, процедура удаления данных может занять много времени!

```
root@RHELv4u2:~# badblocks -ws /dev/sdb
Testing with pattern 0xaa: done
Reading and comparing: done
Testing with pattern 0x55: done
Reading and comparing: done
Testing with pattern 0xff: done
Reading and comparing: done
Testing with pattern 0x00: done
Reading and comparing: done
```

В предыдущем примере осуществляется четырехкратная перезапись каждого из секторов на диске. Однократной перезаписи секторов диска с помощью такого инструмента, как dd, также вполне достаточно для уничтожения всех данных на диске.

Обратите внимание на то, что в примере ниже показан способ необратимого уничтожения всех данных, сохраненных на блочном устройстве.

```
[root@rhel65 ~]# dd if=/dev/zero of=/dev/sdb
```

4.5. Дополнительные параметры функционирования дисковых устройств

Подробное рассмотрение вопросов установки значений параметров функционирования жестких дисков (таких, как dma, gap, ...) выходит за рамки данного курса. Тем не менее, необходимо упомянуть о том, что для этих целей существует несколько утилит, причем утилиты hdparm и sdparm являются двумя наиболее часто используемыми из них.

Утилита hdparm может использоваться для вывода информации и установки значений параметров функционирования жестких дисков с интерфейсом ATA (SATA). Параметры -i и -I позволят вам получить дополнительную информацию о физических свойствах вашего устройства.

```
root@laika:~# hdparm /dev/sdb
/dev/sdb:
IO_support = 0 (default 16-bit)
readonly = 0 (off)
readahead = 256 (on)
geometry = 12161/255/63, sectors = 195371568, start = 0
```

Ниже приведена информация о жестком диске с интерфейсом IDE объемом в 200 ГБ, выводимая утилитой hdparm.

```
root@barry:~# hdparm /dev/hdd
/dev/hdd:
multcount = 0 (off)
IO_support = 0 (default)
unmaskirq = 0 (off)
using_dma = 1 (on)
keepsettings = 0 (off)
readonly = 0 (off)
readahead = 256 (on)
geometry = 24321/255/63, sectors = 390721968, start = 0
```

А это пример вывода утилиты sdparm из состава дистрибутива Ubuntu 10.10.

```
root@ubu1010:~# aptitude install sdparm
...
root@ubu1010:~# sdparm /dev/sda | head -1
/dev/sda: ATA FUJITSU MJA2160B 0081
root@ubu1010:~# man sdparm
```

Используйте утилиты hdparm и sdparm с особой осторожностью.

4.6. Практическое задание: дисковые устройства

О данном практическом задании: для того, чтобы получить практические навыки по работе с жесткими дисками, вам, разумеется, потребуется несколько жестких дисков. В том случае, если у вас в распоряжении нет физического жесткого диска, вы можете использовать виртуальные диски виртуальной машины vmware или VirtualBox. Инструктор должен помочь с подключением нескольких жестких дисков ATA и/или SCSI к виртуальной машине. Результаты выполнения данного практического задания могут использоваться при выполнении трех следующих практических заданий (посвященных работе разделами, файловыми системами, а также монтированию файловых систем).

Рекомендуется подключить к виртуальной машине три жестких диска объемом 1 ГБ и три жестких диска объемом 2 ГБ. Это позволит выполнить как текущее практическое задание, так и практические задания из следующих глав (посвященные работе с RAID-массивами, разделами LVM и iSCSI) без лишних сложностей.

1. Используйте утилиту dmesg для получения списка всех устройств жестких дисков, обнаруженных в процессе загрузки системы.
2. Используйте утилиту fdisk для вычисления общего объема всех жестких дисков, установленных в вашей системе.
3. Остановите виртуальную машину, добавьте три виртуальных диска с интерфейсом SCSI объемом в 1 ГБ

каждый и один виртуальный диск с интерфейсом IDE объемом в 400 МБ. Если это возможно, также добавьте еще один виртуальный диск с интерфейсом IDE объемом в 400 МБ.
4. Используйте утилиту dmesg для проверки корректности обнаружения новых дисковых устройств в процессе загрузки системы.
5. Проверьте наличие файлов устройств, соответствующих дисковым устройствам, в директории /dev.
6. Используйте утилиту fdisk (совместно с утилитой grep и специальным файлом устройства /dev/null) для вывода информации об объеме новых дисковых устройств.
7. Используйте утилиту badblocks для полной очистки одного из дисков наименьшего объема.
8. Рассмотрите содержимое файла /proc/scsi/scsi.
9. Если это возможно, установите утилиты lsscsi и lshw, после чего используйте их для получения списка дисковых устройств.

4.7. Корректная процедура выполнения практического задания: дисковые устройства

1. Используйте утилиту dmesg для получения списка всех устройств жестких дисков, обнаруженных в процессе загрузки системы.

Некоторые возможные варианты ответов...
dmesg | grep -i disk
Поиск жестких дисков с интерфейсом ATA: dmesg | grep hd[abcd]
Поиск жестких дисков с интерфейсом ATA: dmesg | grep -i "ata disk"
Поиск жестких дисков с интерфейсом SCSI: dmesg | grep sd[a-f]
Поиск жестких дисков с интерфейсом SCSI disks: dmesg | grep -i "scsi disk"

1. Используйте утилиту fdisk для вычисления общего объема всех жестких дисков, установленных в вашей системе.

fdisk -l

1. Остановите виртуальную машину, добавьте три виртуальных диска с интерфейсом SCSI объемом в 1 ГБ каждый и один виртуальный диск с интерфейсом IDE объемом в 400 МБ. Если это возможно, также добавьте еще один виртуальный диск с интерфейсом IDE объемом в 400 МБ.

Данное задание выполняется путем редактирования настроек виртуальной машины vmware или VirtualBox.

1. Используйте утилиту dmesg для проверки корректности обнаружения новых дисковых устройств в процессе загрузки системы.

Аналогично заданию 1.

1. Проверьте наличие файлов устройств, соответствующих дисковым устройствам, в директории /dev.

SCSI+SATA: ls -l /dev/sd*
ATA: ls -l /dev/hd*

1. Используйте утилиту fdisk (совместно с утилитой grep и специальным файлом устройства /dev/null) для вывода информации об объеме новых дисковых устройств.
   root@rhel53 ~# fdisk -l 2>/dev/null | grep [MGT]B

Disk /dev/hda: 21.4 GB, 21474836480 bytes
Disk /dev/hdb: 1073 MB, 1073741824 bytes
Disk /dev/sda: 2147 MB, 2147483648 bytes
Disk /dev/sdb: 2147 MB, 2147483648 bytes
Disk /dev/sdc: 2147 MB, 2147483648 bytes

1. Используйте утилиту badblocks для полной очистки одного из дисков наименьшего объема.

#Проверьте корректность имени устройства (/dev/sdc??), которое вы хотите очистить, перед вводом данной команды.
#
root@rhel53 ~# badblocks -ws /dev/sdc
Testing with pattern 0xaa: done
Reading and comparing: done
Testing with pattern 0x55: done
Reading and comparing: done
Testing with pattern 0xff: done
Reading and comparing: done
Testing with pattern 0x00: done
Reading and comparing: done

1. Рассмотрите содержимое файла /proc/scsi/scsi.

root@rhel53 ~# cat /proc/scsi/scsi
Attached devices:
Host: scsi0 Channel: 00 Id: 02 Lun: 00
Vendor: VBOX Model: HARDDISK Rev: 1.0
Type: Direct-Access ANSI SCSI revision: 05
Host: scsi0 Channel: 00 Id: 03 Lun: 00
Vendor: VBOX Model: HARDDISK Rev: 1.0
Type: Direct-Access ANSI SCSI revision: 05
Host: scsi0 Channel: 00 Id: 06 Lun: 00
Vendor: VBOX Model: HARDDISK Rev: 1.0
Type: Direct-Access ANSI SCSI revision: 05

1. Если это возможно, установите утилиты lsscsi и lshw, после чего используйте их для получения списка дисковых устройств.

Debian,Ubuntu: aptitude install lsscsi lshw
Fedora: yum install lsscsi lshw

root@rhel53 ~# lsscsi
[0:0:2:0] disk VBOX HARDDISK 1.0 /dev/sda
[0:0:3:0] disk VBOX HARDDISK 1.0 /dev/sdb
[0:0:6:0] disk VBOX HARDDISK 1.0 /dev/sdc

Глава 5. Разделы жестких дисков

В данной главе будет продолжено рассмотрение приемов работы с жесткими дисками, которые были подготовлены нами в предыдущей главе. На этот раз мы займемся созданием разделов на этих жестких дисках.

Прочитав данную главу, вы будете подготовлены к чтению следующей главы, в которой будет описываться процесс создания файловых систем в подготовленных разделах.

5.1. Информация о разделах жестких дисков

5.1.1. Первичные, расширенные и логические разделы

Для корректного функционирования дистрибутива Linux вам потребуется создать один или несколько разделов (partitions) на жестком диске. Далее будут приведены подробные пояснения относительно создания и использования разделов жестких дисков.

Геометрия раздела (geometry), а также его размер обычно описываются с помощью номеров начального и конечного цилиндра (а иногда начального и конечного сектора). Разделы могут быть первичными (primary, максимум четыре), расширенными (extended, максимум один) или логическими (logical, размещаются внутри расширенного раздела). Каждый раздел имеет поле типа, которое содержит соответствующий код. Данный код позволяет идентифицировать операционную систему компьютера или файловую систему раздела.

Таблица 5.1. Первичные, расширенные и логические разделы

Тип раздела	Порядковый номер
Первичный (максимум 4)	1-4
Расширенный (максимум 1)	1-4
Логический	5-

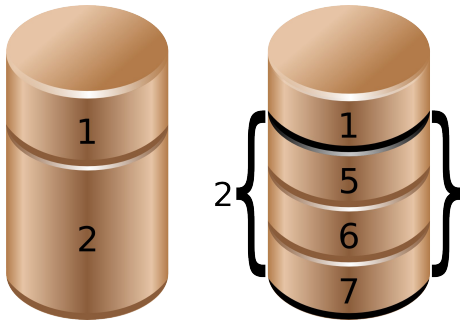
5.1.2. Имена файлов устройств, соответствующих разделам

Ранее мы говорили о том, что имена файлов устройств жестких дисков соответствуют шаблону /dev/hdx или /dev/sdx, причем x является буквенным символом, зависящим от аппаратной конфигурации. После этого символа должен следовать порядковый номер раздела, причем отсчет должен начинаться с числа 1. Следовательно, для обозначения четырех (возможных) первичных разделов могут использоваться числа от 1 до 4. Отсчет порядковых номеров логических разделов всегда начинается с 5. Таким образом, файл устройства /dev/hda2 представляет второй раздел на первом жестком диске с интерфейсом ATA, а файл /dev/hdb5 — первый логический раздел на втором жестком диске с интерфейсом ATA. Аналогичным образом происходит и присваивание имен и файлам устройств, соответствующих жестким дискам с интерфейсом SCSI. Например, файл устройства /dev/sdb3 соответствует третьему разделу второго жесткого диска с интерфейсом SCSI.

Таблица 5.2. Имена файлов устройств, соответствующих разделам

Раздел	Устройство
/dev/hda1	Первый первичный раздел на устройстве, представленном файлом /dev/hda
/dev/hda2	Второй первичный раздел или расширенный раздел на устройстве, представленном файлом /dev/hda
/dev/sda5	Первый логический раздел на устройстве, представленном файлом /dev/sda
/dev/sdb6	Второй логический раздел на устройстве, представленном файлом /dev/sdb

На рисунке ниже представлены схематичные изображения двух жестких дисков (с вращающимися шпинделями) с разделами. Обратите внимание на то, что расширенный раздел выступает в роли контейнера, содержащего логические разделы.



5.2. Обнаружение разделов жестких дисков

5.2.1. Команда fdisk -l

Рассмотрев приведенный в примере ниже вывод команды fdisk -l, вы можете обнаружить, что на жестком диске, представленном файлом устройства /dev/sdb, существуют два раздела. Первый раздел размещается до 31 цилиндра и является разделом подкачки Linux. Второй раздел имеет гораздо больший размер.

root@laika:~# fdisk -l /dev/sdb
Disk /dev/sdb: 100.0 GB, 100030242816 bytes
255 heads, 63 sectors/track, 12161 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Device Boot Start End Blocks Id System
/dev/sdb1 1 31 248976 82 Linux swap / Solaris
/dev/sdb2 32 12161 97434225 83 Linux
root@laika:~#

5.2.2. Файл /proc/partitions

Файл /proc/partitions содержит таблицу с основными и дополнительными номерами файлов устройств, соответствующих разделам жестких дисков, количеством блоков в рамках этих разделов и именами файлов устройств, расположенных в директории /dev. Изучив содержимое файла /proc/devices, убедитесь в том, что в данном списке содержатся основные номера, соответствующие типу каждого из устройств.

paul@RHELv4u4:~$ cat /proc/partitions
major minor #blocks name
3 0 524288 hda
3 64 734003 hdb
8 0 8388608 sda
8 1 104391 sda1
8 2 8281507 sda2
8 16 1048576 sdb
8 32 1048576 sdc
8 48 1048576 sdd
253 0 7176192 dm-0
253 1 1048576 dm-1

Основной номер устройства соответствует типу устройства (или драйверу) и может быть обнаружен в файле /proc/devices. В данном случае основной номер устройства 3 соответствует устройству с интерфейсом IDE, а 8 — устройству с интерфейсом SATA. Основной номер устройства описывает драйвер, который должен использоваться для взаимодействия с данным устройством.

Дополнительный номер устройства является уникальным идентификатором экземпляра устройства данного типа. Файл devices.txt в дереве исходного кода ядра Linux содержит полный список основных и дополнительных номеров устройств.

5.2.3. Parted и другие утилиты

Вас могут заинтересовать такие альтернативные утилите fdisk инструменты, как parted, cfdisk, sfdisk и gparted. При этом для осуществления манипуляций с разделами жестких дисков в рамках данного курса будет использоваться главным образом утилита fdisk.

Утилита parted рекомендуется разработчиками некоторых дистрибутивов Linux для работы с дисками, содержащими таблицы разделов формата GPT, а не MBR.

Ниже приведен пример использования утилиты parted при работе с дистрибутивом CentOS.

[root@centos65 ~]#
rpm -q parted
parted-2.1-21.el6.x86_64
[root@centos65 ~]#
parted /dev/sda
GNU Parted 2.1
Используется /dev/sda
Добро пожаловать в GNU Parted! Наберите help для просмотра списка команд.
(parted)
print
Модель: ATA VBOX HARDDISK (scsi)
Диск /dev/sda: 42.9GB
Размер сектора (логич./физич.): 512B/512B
Таблица разделов: msdos
Номер Начало Конец Размер Тип Файловая система Флаги
1 1049kB 525MB 524MB primary ext4 загрузочный
2 525MB 42.9GB 42.4GB primary lvm
(parted)

5.3. Создание разделов на новых дисках

В примере ниже будет рассматриваться ситуация покупки нового жесткого диска для нашей системы. После корректного подключения нового аппаратного обеспечения вы можете использовать утилиты fdisk и parted для создания необходимых разделов. В данном примере используется утилита fdisk, но утилита parted в подобной ситуации также является вполне подходящим инструментом.

5.3.1. Обнаружение жесткого диска

В первую очередь мы должны удостовериться в том, что ядро Linux имеет доступ к новому жесткому диску с помощью команды fdisk -l. Да, новый жесткий диск представлен файлом устройства /dev/sdb, но на нем пока не создано каких-либо разделов.

root@RHELv4u2:~# fdisk -l
Disk /dev/sda: 12.8 GB, 12884901888 bytes
255 heads, 63 sectors/track, 1566 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Device Boot Start End Blocks Id System
/dev/sda1 * 1 13 104391 83 Linux
/dev/sda2 14 1566 12474472+ 8e Linux LVM
Disk /dev/sdb: 1073 MB, 1073741824 bytes
255 heads, 63 sectors/track, 130 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Disk /dev/sdb doesn't contain a valid partition table

5.3.2. Открытие дискового устройства с помощью утилиты fdisk

После этого мы можем приступить к созданию раздела на жестком диске, представленном файлом устройства /dev/sdb, с помощью утилиты fdisk. Для начала мы должны передать путь к файлу устройства /devsdb в качестве параметра утилиты fdisk. Следует проявить крайнюю осторожность и убедиться еще раз в том, что вы создаете раздел именно на новом диске!!

root@RHELv4u2:~# fdisk /dev/sdb
Device contains neither a valid DOS partition table, nor Sun, SGI...
Building a new DOS disklabel. Changes will remain in memory only,
until you decide to write them. After that, of course, the previous
content won't be recoverable.
Warning: invalid flag 0x0000 of partition table 4 will be corrected...

5.3.3. Пустая таблица разделов

Работая с инструментом fdisk, вы можете выполнить команду p для того, чтобы ознакомиться с текущей таблицей разделов диска.

Command (m for help): p
Disk /dev/sdb: 1073 MB, 1073741824 bytes
255 heads, 63 sectors/track, 130 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Device Boot Start End Blocks Id System

5.3.4. Создание нового раздела

На данный момент на диске не существует разделов, поэтому мы должны выполнить команду n для создания нового раздела. Мы будем вводить символ p для создания первичного раздела и число 1 в качестве номера раздела, а также число 1 в качестве номера начального цилиндра и число 14 в качестве номера конечного цилиндра.

Command (m for help): n
Command action
e extended
p primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-130, default 1):
Using default value 1
Last cylinder or +size or +sizeM or +sizeK (1-130, default 130): 14

Теперь мы снова можем выполнить команду p для проверки внесенных нами изменений, причем эти изменения еще не записаны на диск, поэтому мы все еще можем отменить эту операцию! Так как внесенные изменения кажутся нам корректными, мы можем выполнить команду w для записи изменений на диск с последующим завершением работы утилиты fdisk.

Command (m for help): p
Disk /dev/sdb: 1073 MB, 1073741824 bytes
255 heads, 63 sectors/track, 130 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Device Boot Start End Blocks Id System
/dev/sdb1 1 14 112423+ 83 Linux
Command (m for help): w
The partition table has been altered!
Calling ioctl() to re-read partition table.
Syncing disks.
root@RHELv4u2:~#

5.3.5. Вывод информации о новом разделе

Давайте снова проверим с помощью команды fdisk -l, совпадает ли реальность с нашими ожиданиями. Разумеется, в примере ниже приводится информация о разделе на диске, представленном файлом устройства /dev/sdb.

root@RHELv4u2:~# fdisk -l
Disk /dev/sda: 12.8 GB, 12884901888 bytes
255 heads, 63 sectors/track, 1566 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Device Boot Start End Blocks Id System
/dev/sda1 * 1 13 104391 83 Linux
/dev/sda2 14 1566 12474472+ 8e Linux LVM
Disk /dev/sdb: 1073 MB, 1073741824 bytes
255 heads, 63 sectors/track, 130 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Device Boot Start End Blocks Id System
/dev/sdb1 1 14 112423+ 83 Linux
root@RHELv4u2:~#

5.4. О таблице разделов

5.4.1. Основная загрузочная запись

Информация из таблицы разделов (об основных и расширенных разделах) записывается в основную загрузочную запись (Master Boot Record или MBR). Вы можете использовать утилиту dd для копирования данных из основной загрузочной записи в файл.

В данном примере осуществляется копирование основной загрузочной записи с первого жесткого диска с интерфейсом SCSI.

dd if=/dev/sda of=/SCSIdisk.mbr bs=512 count=1

Этот же инструмент также может использоваться для удаления всей информации о разделах на диске. В данном примере область основной загрузочной записи заполняется нулевыми байтами.

dd if=/dev/zero of=/dev/sda bs=512 count=1

Или удаления всей информации из раздела на жестком диске или со всего диска.

dd if=/dev/zero of=/dev/sda

5.4.2. Утилита partprobe

Не забывайте о том, что после восстановления основной загрузочной записи с помощью утилиты dd вам придется сообщить ядру ОС о необходимости повторного чтения таблицы разделов жесткого диска с помощью утилиты partprobe. После запуска утилиты partprobe разделы жесткого диска могут снова использоваться.

[root@RHEL5 ~]# partprobe
[root@RHEL5 ~]#

5.4.3. Логические разделы

Таблица разделов жесткого диска не содержит информации о логических разделах. Следовательно, резервная копия основной загрузочной записи, созданная с помощью утилиты dd, будет содержать информацию исключительно о первичных и расширенных разделах жесткого диска. Для создания резервной копии основной загрузочной записи с информацией о логических разделах жесткого диска вы можете воспользоваться утилитой sfdisk.

В данном примере показана методика создания резервной копии таблицы всех разделов жесткого диска, включая логические, с записью данных в файл.

sfdisk -d /dev/sda > parttable.sda.sfdisk

А в данном примере осуществляется копирование основной загрузочной записи и информации обо всех логических разделах с устройства, представленного файлом /dev/sda, на устройство, представленное файлом /dev/sdb.

sfdisk -d /dev/sda | sfdisk /dev/sdb

5.5. Практическое задание: разделы жестких дисков

1. Используйте команду fdisk -l для вывода списка существующих разделов жестких дисков с информацией об их размерах.
2. Используйте команду df -h для вывода списка существующих разделов жестких дисков с информацией об их размерах.
3. Сравните вывод утилиты fdisk с выводом утилиты df.
4. Создайте основной раздел размером в 200 МБ на жестком диске малого объема.
5. Создайте основной раздел размером в 400 МБ и два логических раздела размером в 300 МБ каждый на жестком диске большого объема.
6. Используйте команды df -h и fdisk -l для проверки корректности выполненных вами действий.
7. Снова сравните вывод утилиты fdisk с выводом утилиты df. Присутствует ли в выводах обоих утилит информация о новых разделах?
8. С помощью утилиты dd создайте резервную копию основной загрузочной записи, которая содержит информацию о вашем основном разделе размером в 200 МБ.
9. Создайте резервную копию таблицы разделов, содержащей информацию о вашем первичном разделе размером в 400 МБ и о логических разделах размером в 300 МБ каждый. Убедитесь в том, что информация о логических разделах добавлена в файл резервной копии таблицы разделов.

10 (дополнительное задание). Удалите информацию обо всех ваших разделах с помощью утилиты fdisk. После этого восстановите ее из файлов резервных копий.

5.6. Корректная процедура выполнения практического задания: разделы жестких дисков

1. Используйте команду fdisk -l для вывода списка существующих разделов жестких дисков с информацией об их размерах.

С использованием учетной записи пользователя root: # fdisk -l

1. Используйте команду df -h для вывода списка существующих разделов жестких дисков с информацией об их размерах.

df -h

1. Сравните вывод утилиты fdisk с выводом утилиты df.

Информация о некоторых разделах жестких диск5ов будет присутствовать в выводах обоих утилит (возможно, одним из таких разделов будет раздел, представленный файлом устройства /dev/sda1 или /dev/hda1).

1. Создайте основной раздел размером в 200 МБ на жестком диске малого объема.

Выберите один из добавленных ранее дисков (в данном примере используется диск, представленный файлом устройства /dev/sdc).

root@rhel53 ~# fdisk /dev/sdc
...
Command (m for help): n
Command action
e extended
p primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-261, default 1): 1
Last cylinder or +size or +sizeM or +sizeK (1-261, default 261): +200m
Command (m for help): w
The partition table has been altered!
Calling ioctl() to re-read partition table.
Syncing disks.

1. Создайте основной раздел размером в 400 МБ и два логических раздела размером в 300 МБ каждый на жестком диске большого объема.

Выберите один из добавленных ранее дисков (в данном примере используется диск, представленный файлом устройства /dev/sdb).

fdisk /dev/sdb

при работе с утилитой fdisk : n p 1 +400m enter --- n e 2 enter enter --- n l +300m (дважды)

1. Используйте команды df -h и fdisk -l для проверки корректности выполненных вами действий.

fdisk -l ; df -h

1. Снова сравните вывод утилиты fdisk с выводом утилиты df. Присутствует ли в выводах обоих утилит информация о новых разделах?

Созданные разделы обнаруживаются с помощью утилиты fdisk.

При этом эти же разделы не обнаруживаются с помощью утилиты df.

1. С помощью утилиты dd создайте резервную копию основной загрузочной записи, которая содержит информацию о вашем основном разделе размером в 200 МБ.

dd if=/dev/sdc of=bootsector.sdc.dd count=1 bs=512

1. Создайте резервную копию таблицы разделов, содержащую информацию о вашем первичном разделе размером в 400 МБ и о логических разделах размером в 300 МБ каждый. Убедитесь в том, что информация о логических разделах добавлена в файл резервной копии таблицы разделов.

sfdisk -d /dev/sdb > parttable.sdb.sfdisk

10 (дополнительное задание). Удалите информацию обо всех ваших разделах с помощью утилиты fdisk. После этого восстановите ее из файлов резервных копий.

Глава 6. Файловые системы

После того, как вы закончите разделение жесткого диска на разделы, вы сможете создать файловую систему в каждом из созданных разделов.

В данной главе используются разделы жесткого диска, созданные в предыдущей главе, причем сама глава содержит информацию, необходимую для подготовки к чтению следующей главы, в которой мы будем монтировать созданные файловые системы.

6.1. Информация о файловых системах

Файловая система является способом организации файлов в рамках вашего раздела жесткого диска. Помимо сохранения данных в рамках файлов файловые системы обычно позволяют использовать директории и механизм контроля доступа, кроме того, они сохраняют метаданные, относящиеся к файлам, такие, как метки времени доступа и модификации, а также данные о владельце.

Ограничения имен файлов (максимальная длина, набор допустимых символов, ...) определяются типом выбранной вами файловой системы. Директории обычно реализуются с помощью файлов специального типа и вам предстоит разобраться в вопросах их реализации! Механизм доступа к файлам оперирует идентификаторами пользователей, владеющих файлами (а также идентификаторами групп пользователей, владеющих файлами и учитывает членство пользователей в группах) с учетом одного или большего количества списков контроля доступа.

6.1.1. Страница руководства man fs

Страница руководства, посвященной файловым системам, может быть открыта с помощью команды man fs.

[root@rhel65 ~]# man fs

6.1.2. Файл /proc/filesystems

Ядро Linux проинформирует вас о загруженных на данный момент драйверах файловых систем с помощью файла /proc/filesystems.

root@rhel53 ~# cat /proc/filesystems | grep -v nodev
ext2
iso9660
ext3

6.1.3. Файл /etc/filesystems

Файл /etc/filesystems содержит список автоматически определенных файловых систем (на случай использования утилиты mount без параметра -t).

Информация о данном файле содержится на странице руководства, которая может быть открыта с помощью команды man mount.

[root@rhel65 ~]# man mount

6.2. Часто используемые файловые системы

6.2.1. Файловые системы ext2 и ext3

В прошлом наиболее часто используемой файловой системой в Linux была файловая система ext2 («вторая расширенная файловая система»). Недостаток данной файловой системы заключался в значительных затратах времени на ее проверку.

Впоследствии на большинстве машин, работающих под управлением Linux, файловая система ext2 была заменена на файловую систему ext3. По большей части данные файловые системы идентичны, за исключением функции журналирования, которая присутствует только в файловой системе ext3.

Функция журналирования подразумевает запись всех изменений в файловой системе в первую очередь в журнал, расположенный на диске. Периодически осуществляемая синхронизация журнала с диском приводит к записи изменений в файловую систему. Функция журналирования позволяет поддерживать файловую систему в корректном состоянии, поэтому вам не придется осуществлять полную проверку файловой системы после некорректного выключения компьютера или проблем с питанием.

6.2.2. Создание файловых систем ext2 и ext3

Вы можете создать данные файловые системы с помощью утилит /sbin/mkfs и /sbin/mke2fs. Используйте команду mke2fs -j для создания файловой системы ext3.

Также вы можете преобразовать файловую систему ext2 в ext3 с помощью команды tune2fs -j. При этом вы можете смонтировать файловую систему ext3 как ext2, но в этом случае вы потеряете функцию журналирования. Не забывайте о необходимости выполнения команды mkinitrd в том случае, если устройство с данной файловой системой должно использоваться в процессе загрузки системы.

6.2.3. Файловая система ext4

Новейшим воплощением расширенной файловой системы является файловая система ext4, код для поддержки которой включен в состав ядра Linux в 2008 году. Файловая система ext4 поддерживает файлы большего размера (до 16 терабайт), а также позволяет создавать файловые системы на разделах больших размеров, чем в случае файловой системы ext3 (кроме того, она предоставляет множество дополнительных возможностей).

Процесс разработки данной файловой системы начался с доработки файловой системы ext3 с целью полной поддержки 64-битных операций. После того, как выяснилось, что в код должны быть внесены значительные изменения, разработчики приняли решение о присвоении созданной в итоге файловой системе имени ext4.

6.2.4. Файловая система xfs

Файловая система xfs используется по умолчанию в дистрибутиве Redhat Enterprise Linux 7. Это хорошо масштабируемая высокопроизводительная файловая система.

Файловая система xfs была создана специально для операционной системы Irix и в течение нескольких лет также использовалась в операционной системе FreeBSD. Она поддерживается ядром Linux, но редко используется в дистрибутивах, не имеющих отношения к дистрибутивам Redhat/CentOS.

6.2.5. Файловая система vfat

Файловая система vfat существует в нескольких форматах: fat12 для дискет, fat16 для операционной системы ms-dos и fat32 для поддержки дисков большего объема. Реализация файловой системы vfat из состава ядра Linux поддерживает все упомянутые форматы, но при этом в данной реализации отсутствует поддержка ряда возможностей, таких, как механизм контроля доступа и ссылки на файлы. Диски с файловой системой fat могут читаться средствами любой операционной системы и широко используются в цифровых камерах и накопителях с интерфейсом USB, причем данная файловая система очень удобна для осуществления обмена данными между операционными системами, установленными на компьютере домашнего пользователя.

6.2.6. Файловая система iso 9660

Файловая система iso 9660 является стандартной файловой системой для оптических дисков CD-ROM. Высока вероятность того, что вы также встретите данную файловую систему на вашем жестком диске в форме образов дисков CD-ROM (которые обычно имеют расширение .iso). Стандарт iso 9660 ограничивает длину имен файлов форматом 8.3. Это ограничение было воспринято негативно в мире Unix, в результате чего в стандарт было добавлено расширение Rock Ridge, позволяющее использовать имена файлов длиной в 255 символов, режимы доступа и идентификаторы владельцев файлов в стиле Unix, а также символьные ссылки. Другим расширением для стандарта iso9660 являются расширение Joliet, которое добавляет возможность

использования до 64 символов Unicode в именах файлов. Расширение El Torito также относится к стандарту iso 9660 и позволяет осуществлять загрузку системы с дисков CD-ROM.

6.2.7. Файловая система udf

При создании большинства оптических дисков (включая диски CD и DVD) в настоящее время используется файловая система udf, название которой расшифровывается как Universal Disk Format (универсальный дисковый формат).

6.2.8. Файловая система swap

Если принимать во внимание все детали, swap не является файловой системой. Но для использования раздела в качестве раздела подкачки он должен быть отформатирован и смонтирован как пространство для хранения данных подкачки.

6.2.9. Файловая система gfs

Кластеры на основе Linux обычно используют отдельную кластерную файловую систему, такую, как GFS, GFS2, ClusterFS, ...

6.2.10. Другие файловые системы...

В более старых системах Linux вы можете встретить файловую систему reiserfs. Также не исключено, что вы столкнетесь с файловой системой zfs, разработанной в компании Sun, или открытой файловой системой btrfs. Для описания возможностей последней, скорее всего, потребуется отдельная глава.

6.2.11. Файл /proc/filesystems

Файл /proc/filesystems содержит список поддерживаемых файловых систем. В момент монтировании файловой системы без явного указания ее типа утилита mount в первую очередь попытается идентифицировать файловую систему как одну из файловых систем, описанных в файле /etc/filesystems, после чего в случае необходимости попытается идентифицировать эту же файловую систему как одну из файловых систем, описанных в файле /proc/filesystems за исключением тех, для которых установлена метка nodev. В том случае, если в последней строке файла /etc/filesystems расположен только символ звездочки (*), будут использоваться оба упомянутых файла.

```
paul@RHELv4u4:~$ cat /proc/filesystems
nodev sysfs
nodev rootfs
nodev bdev
nodev proc
nodev sockfs
nodev binfmt_misc
nodev usbfs
nodev usbdevfs
nodev futexfs
nodev tmpfs
nodev pipefs
nodev eventpollfs
nodev devpts
ext2
nodev ramfs
nodev hugetlbfs
iso9660
nodev relayfs
nodev mqueue
nodev selinuxfs
ext3
nodev rpc_pipefs
nodev vmware-hgfs
nodev autofs
paul@RHELv4u4:~$
```

6.3. Создание файловой системы в разделе

На данный момент у нас в распоряжении имеется недавно созданный раздел жесткого диска. Список системных бинарных файлов, предназначенных для создания файловых систем, может быть сформирован с помощью утилиты ls.

```
[root@RHEL4b ~]# ls -lS /sbin/mk*
-rwxr-xr-x 3 root root 34832 апр 24 2006 /sbin/mke2fs
-rwxr-xr-x 3 root root 34832 апр 24 2006 /sbin/mkfs.ext2
-rwxr-xr-x 3 root root 34832 апр 24 2006 /sbin/mkfs.ext3
-rwxr-xr-x 3 root root 28484 окт 13 2004 /sbin/mkdosfs
-rwxr-xr-x 3 root root 28484 окт 13 2004 /sbin/mkfs.msdos
-rwxr-xr-x 3 root root 28484 окт 13 2004 /sbin/mkfs.vfat
-rwxr-xr-x 1 root root 20313 апр 10 2006 /sbin/mkinitrd
-rwxr-x--- 1 root root 15444 окт 5 2004 /sbin/mkzonedb
-rwxr-xr-x 1 root root 15300 май 24 2006 /sbin/mkfs.cramfs
-rwxr-xr-x 1 root root 13036 май 24 2006 /sbin/mkswap
-rwxr-xr-x 1 root root 6912 май 24 2006 /sbin/mkfs
-rwxr-xr-x 1 root root 5905 авг 3 2004 /sbin/mkbootdisk
[root@RHEL4b ~]#
```

Самое время прочитать страницы руководств утилит mkfs и mke2fs. С помощью приведенного ниже примера вы можете проследить процесс создания файловой системы ext2 в разделе, представленном файлом устройства /dev/sdb1. В реальной жизни вам также могут понадобиться такие параметры данной утилиты, как -m0 и -j.

```
root@RHELv4u2:~# mke2fs /dev/sdb1
mke2fs 1.35 (28-Feb-2004)
Filesystem label=
```

OS type: Linux
Block size=1024 (log=0)
Fragment size=1024 (log=0)
28112 inodes, 112420 blocks
5621 blocks (5.00%) reserved for the super user
First data block=1
Maximum filesystem blocks=67371008
14 block groups
8192 blocks per group, 8192 fragments per group
2008 inodes per group
Superblock backups stored on blocks:
8193, 24577, 40961, 57345, 73729
Writing inode tables: done
Writing superblocks and filesystem accounting information: done
This filesystem will be automatically checked every 37 mounts or
180 days, whichever comes first. Use tune2fs -c or -i to override

6.4. Настройка файловой системы

Вы можете использовать утилиту tune2fs для вывода списка параметров файловой системы и установки их значений. В первом примере показана методика вывода информации о зарезервированном для пользователя root пространстве в рамках раздела (текущее значение равно пяти процентам от общего размера раздела).

[root@rhel4 ~]# tune2fs -l /dev/sda1 | grep -i "block count"
Block count: 104388
Reserved block count: 5219
[root@rhel4 ~]#

А в примере ниже данное значение увеличивается до 10 процентов от общего размера раздела. Вы можете использовать утилиту tune2fs в процессе эксплуатации файловой системы, даже в том случае, если данная файловая система содержит корневую директорию (как в примере ниже).

[root@rhel4 ~]# tune2fs -m10 /dev/sda1
tune2fs 1.35 (28-Feb-2004)
Setting reserved blocks percentage to 10 (10430 blocks)
[root@rhel4 ~]# tune2fs -l /dev/sda1 | grep -i "block count"
Block count: 104388
Reserved block count: 10430
[root@rhel4 ~]#

6.5. Проверка файловой системы

Команда fsck позволяет задействовать утилиту-обертку, используемую для вызова утилит, осуществляющих проверку файловых систем.

[root@RHEL4b ~]# ls /sbin/fsck
/sbin/dosfsck /sbin/fsck /sbin/fsck.ext2 /sbin/fsck.msdos
/sbin/e2fsck /sbin/fsck.cramfs /sbin/fsck.ext3 /sbin/fsck.vfat
[root@RHEL4b ~]#

Значение из последнего столбца файла /etc/fstab используется в качестве флага проверки файловой системы в процессе загрузки операционной системы.

[paul@RHEL4b ~]$ grep ext /etc/fstab
/dev/VolGroup00/LogVol00 / ext3 defaults 1 1
LABEL=/boot /boot ext3 defaults 1 2
[paul@RHEL4b ~]$

Проверка смонтированной файловой системы в ручном режиме приведет к выводу предупреждения и завершению работы утилиты fsck.

[root@RHEL4b ~]# fsck /boot
fsck из util-linux 2.25.2
e2fsck 1.42.11 (09-Jul-2014)
/dev/sda1 is mounted.
e2fsck: Cannot continue, aborting.

Но после размонтирования файловой системы ext2 утилиты fsck и e2fsck могут успешно использоваться для ее проверки.

[root@RHEL4b ~]# fsck /boot
fsck из util-linux 2.25.2
e2fsck 1.42.11 (09-Jul-2014)
/boot: clean, 44/26104 files, 17598/104388 blocks
[root@RHEL4b ~]# fsck -p /boot
fsck из util-linux 2.25.2
/boot: clean, 44/26104 files, 17598/104388 blocks
[root@RHEL4b ~]# e2fsck -p /dev/sda1
/boot: clean, 44/26104 files, 17598/104388 blocks

6.6. Практическое задание: файловые системы

1. Выведите список файловых систем, известных вашей операционной системе.
2. Создайте файловую систему ext2 в разделе размером в 200 МБ.
3. Создайте файловую систему ext3 в одном из логических разделов размером в 300 МБ.
4. Создайте файловую систему ext4 в разделе размером в 400 МБ.
5. Установите размер резервируемого пространства для пользователя root в разделе с файловой системой ext3 равным 0 процентов от общего размера раздела.
6. Проверьте корректность выполненных действий с помощью утилит fdisk и df.
7. Выполните проверки всех недавно созданных файловых

систем.

6.7. Корректная процедура выполнения практического задания: файловые системы

1. Выведите список файловых систем, известных вашей операционной системе.

man fs
cat /proc/filesystems
cat /etc/filesystems (не во всех дистрибутивах Linux)

1. Создайте файловую систему ext2 в разделе размером в 200 МБ.

mke2fs /dev/sdc1 (замените sdc1 на корректное имя файла устройства, представляющего необходимый раздел жесткого диска)

1. Создайте файловую систему ext3 в одном из логических разделов размером в 300 МБ.

mke2fs -j /dev/sdb5 (замените sdb5 на корректное имя файла устройства, представляющего необходимый раздел жесткого диска)

1. Создайте файловую систему ext4 в разделе размером в 400 МБ.

mkfs.ext4 /dev/sdb1 (замените sdb1 на корректное имя файла устройства, представляющего необходимый раздел жесткого диска)

1. Установите размер резервируемого пространства для пользователя root в разделе с файловой системой ext3 равным 0 процентов от общего размера раздела.

tune2fs -m 0 /dev/sdb5

1. Проверьте корректность выполненных действий с помощью утилит fdisk и df.

Использование утилиты mkfs (mke2fs) не должно стать причиной принципиальных различий в выводе предложенных для осуществления диагностики утилит.

В то же время, принципиальные различия проявятся в следующей главе, посвященной монтированию файловых систем.

1. Выполните проверки всех недавно созданных файловых систем.

fsck /dev/sdb1
fsck /dev/sdc1
fsck /dev/sdb5

Глава 7. Монтирование файловых систем

После того, как вы создали файловую систему в разделе жесткого диска, вы можете смонтировать ее. Монтирование файловой системы делает ее доступной для использования обычно посредством специально созданной директории. Мы говорим о монтировании файловой системы, а не о монтировании раздела жесткого диска, ведь, как мы увидим позднее, у нас также имеется возможность монтирования файловых систем, которые созданы вне разделов жестких дисков.

Во всех системах Unix каждый файл и каждая директория являются частью одного большого дерева директорий файловой системы. Для доступа к файлу вы должны знать полный путь к этому файлу, начинающийся с корневой директории. При добавлении файловой системы в ходе обслуживания операционной системы вам придется предоставить доступ к этой файловой системе из какой-либо точки дерева директорий файловой системы. Директория, из которой будет доступна ваша файловая система, называется точкой монтирования.

7.1. Монтирование локальных файловых систем

7.1.1. Создание точки монтирования с помощью утилиты mkdir

В данном примере демонстрируется методика создания новой точки монтирования с помощью утилиты mkdir.

root@RHELv4u2:~# mkdir /home/project42

7.1.2. Утилита mount

После того, как точка монтирования создана, а файловая система присутствует на разделе жесткого диска, утилита mount может быть использована для монтирования файловой системы в директорию, которая станет точкой монтирования.

root@RHELv4u2:~# mount -t ext2 /dev/sdb1 /home/project42/

После монтирования пользователи получат доступ к файловой системе.

7.1.3. Файл /etc/filesystems

На самом деле использование параметра -t ext2 утилиты mount для явного указания типа файловой системы не всегда является необходимым. Утилита mount имеет возможность автоматической идентификации типов множества файловых систем.

В случае монтирования файловой системы без явного указания ее типа утилита mount в первую очередь попытается идентифицировать файловую систему как одну из файловых систем, перечисленных в файле /etc/filesystems. При этом утилита mount будет пропускать строки с директивой nodev.

paul@RHELv4u4:~$ cat /etc/filesystems
ext3
ext2
nodev proc
nodev devpts

iso9660
vfat
hfs

7.1.4. Файл /proc/filesystems

В том случае, если файла /etc/filesystems не существует или данный файл заканчивается строкой с одним символом *, утилита mount будет читать файл /proc/filesystems.

[root@RHEL52 ~]# cat /proc/filesystems | grep -v ^nodev
ext2
iso9660
ext3

7.1.5. Утилита umount

Вы можете отмонтировать файловую систему с помощью утилиты umount.

root@pasha:~# umount /home/reet

7.2. Вывод списка смонтированных файловых систем

Для вывода списка смонтированных файловых систем следует использовать команду mount. Также можно прочитать содержимое файлов /proc/mounts и /etc/mtab.

7.2.1. Утилита mount

Простейший и наиболее часто используемый способ вывода списка смонтированных файловых систем заключается в использовании утилиты mount без каких-либо аргументов.

root@RHELv4u2:~# mount | grep /dev/sdb
/dev/sdb1 on /home/project42 type ext2 (rw)

7.2.2. Файл /proc/mounts

Ядро ОС предоставляет информацию о смонтированных файловых системах посредством файла /proc/mounts, причем данный файл не хранится на жестком диске. При чтении информации из файла /proc/mounts осуществляется чтение информации, переданной непосредственно ядром ОС.

root@RHELv4u2:~# cat /proc/mounts | grep /dev/sdb
/dev/sdb1 /home/project42 ext2 rw 0 0

7.2.3. Файл /etc/mtab

Содержимое файла /etc/mtab не обновляется средствами ядра ОС, а поддерживается в актуальном состоянии средствами утилиты mount. Не редактируйте файл /etc/mtab вручную.

root@RHELv4u2:~# cat /etc/mtab | grep /dev/sdb
/dev/sdb1 /home/project42 ext2 rw 0 0

7.2.4. Утилита df

Более удобный для пользователя способ получения списка смонтированных файловых систем заключается в использовании утилиты df. Утилита df (название расшифровывается как diskfree — свободное пространство диска) имеет полезную дополнительную возможность, заключающуюся в выводе данных об объеме свободного пространства в каждой из смонтированных файловых систем, расположенных в разделах жестких дисков. Как и большинство утилит из состава Linux, утилита df поддерживает параметр -h, предназначенный для активации режима вывода данных в формате, облегчающем чтение.

root@RHELv4u2:~# df
Файловая система 1K-блоков Использовано Доступно Использовано% Смонтировано в
/dev/mapper/VolGroup00-LogVol00
11707972 6366996 4746240 58% /
/dev/sda1 101086 9300 86567 10% /boot
none 127988 0 127988 0% /dev/shm
/dev/sdb1 108865 1550 101694 2% /home/project42

root@RHELv4u2:~# df -h
Файловая система Размер Использовано Дост Использовано% Смонтировано в
/dev/mapper/VolGroup00-LogVol00
12G 6.1G 4.6G 58% /
/dev/sda1 99M 9.1M 85M 10% /boot
none 125M 0 125M 0% /dev/shm
/dev/sdb1 107M 1.6M 100M 2% /home/project42

7.2.5. Команда df -h

В приведенном ниже примере вывода команды df -h вы можете обнаружить информацию о размере файловой системы, свободном пространстве, использованном пространстве в гигабайтах и процентах, а также точке монтирования файловой системы, созданной в соответствующем разделе жесткого диска.

root@laika:~# df -h | egrep -e "(sdb2|File)"
Файловая система Размер Использовано Дост Использовано% Смонтировано в
/dev/sdb2 92G 83G 8.6G 91% /media/sdb2

7.2.6. Утилита du

Утилита du позволяет получить количественные показатели использования пространства жесткого диска для хранения файлов и директорий. Используя утилиту du с параметром, являющимся путем к точке монтирования, вы на самом деле будете получать информацию о пространстве раздела жесткого диска, занятом данными из файловой системы.

Хотя утилита du и может осуществлять рекурсивный обход дерева директорий с последующим выводом информации о каждой из директорий, параметр -s позволит вам получить обобщенную информацию о размере родительской директории.

Этот параметр чаще всего используется вместе с параметром -h. Поэтому команда du -sh по отношению к точке монтирования позволит получить информацию обо всем пространстве раздела диска, использованном файловой системой.

```
root@debian6~# du -sh /boot /srv/wolf
6.2M /boot
1.1T /srv/wolf
```

7.3. Процесс монтирования файловой системы от начала до конца

В приведенном ниже примере показана полная последовательность действий, начинающаяся с определения имени файла устройства жесткого диска (/dev/sdb) и заканчивающаяся монтированием созданной в разделе этого жесткого диска файловой системы в директорию /mnt.

```
[root@centos65 ~]#
dmesg | grep \[sdb\]
sd 3:0:0:0: [sdb] 150994944 512-byte logical blocks: (77.3 GB/72.0 GiB)
sd 3:0:0:0: [sdb] Write Protect is off
sd 3:0:0:0: [sdb] Mode Sense: 00 3a 00 00
sd 3:0:0:0: [sdb] Write cache: enabled, read cache: enabled, doesn't support \
DPO or FUA
sd 3:0:0:0: [sdb] Attached SCSI disk
[root@centos65 ~]#
parted /dev/sdb
(parted)
mklabel msdos
(parted)
mkpart primary ext4 1 77000
(parted)
print
Модель: ATA VBOX HARDDISK (scsi)
Диск /dev/sdb: 77.3GB
Размер сектора (логич./физич.): 512B/512B
Таблица разделов: msdos
Номер Начало Конец Размер Тип Файловая система Флаги
1 1049kB 77.0GB 77.0GB primary
(parted)
quit
[root@centos65 ~]#
mkfs.ext4 /dev/sdb1
mke2fs 1.41.12 (17-May-2010)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
4702208 inodes, 18798592 blocks
939929 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=4294967296
574 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
( вывод сокращен )
...
[root@centos65 ~]#
mount /dev/sdb1 /mnt
[root@centos65 ~]#
mount | grep mnt
/dev/sdb1 on /mnt type ext4 (rw)
[root@centos65 ~]#
df -h | grep mnt
/dev/sdb1 71G 180M 67G 1% /mnt
[root@centos65 ~]#
du -sh /mnt
20K /mnt
[root@centos65 ~]#
umount /mnt
```

7.4. Монтирование файловых систем на постоянной основе

До текущего момента мы монтировали все файловые системы вручную. Такой подход вполне приемлем, но файловая система остается смонтированной только до следующей перезагрузки системы. К счастью, существует способ уведомления вашего компьютера о необходимости монтирования определенных файловых систем в процессе загрузки операционной системы.

7.4.1. Файл /etc/fstab

Таблица монтируемых файловых систем, расположенная в файле /etc/fstab, содержит список файловых систем с параметрами, позволяющими монтировать каждую из них в процессе загрузки операционной системы.

Ниже приведен пример файла /etc/fstab.

```
root@RHELv4u2:~# cat /etc/fstab
/dev/VolGroup00/LogVol00 / ext3 defaults 1 1
LABEL=/boot /boot ext3 defaults 1 2
```

none /dev/pts devpts gid=5,mode=620 0 0
none /dev/shm tmpfs defaults 0 0
none /proc proc defaults 0 0
none /sys sysfs defaults 0 0
/dev/VolGroup00/LogVol01 swap swap defaults 0 0

Добавив в данный файл следующую строку, вы можете автоматизировать процесс монтирования созданной файловой системы.

/dev/sdb1 /home/project42 ext2 defaults 0 0

7.4.2. Команда mount /точка_монтирования

Благодаря добавлению элемента с список монтируемых файловых систем, расположенный в файле /etc/fstab, вы можете использовать упрощенный вариант команды mount. Команда из примера ниже позволяет осуществить поиск информации о файле устройства, представляющем связанный с переданной точкой монтирования раздел жесткого диска, в файле /etc/fstab с помощью утилиты mount .

root@rhel65:~# mount /home/project42

7.5. Безопасное монтирование файловых систем

Безопасное монтирование файловых систем становится возможным благодаря наличию нескольких специализированных параметров монтирования файловых систем. Эти параметры рассмотрены в примерах, приведенных ниже.

7.5.1. Параметр ro

Параметр ro позволяет смонтировать файловую систему в режиме только для чтения, что позволит предотвратить запись данных в нее любым пользователем.

root@rhel53 ~# mount -t ext2 -o ro /dev/hdb1 /home/project42
root@rhel53 ~# touch /home/project42/testwrite
touch: невозможно выполнить touch для ‘/home/project42/testwrite’: Файловая система доступна только для чтения

7.5.2. Параметр noexec

Параметр noexec позволяет предотвратить исполнение бинарных файлов и сценариев, расположенных в смонтированной файловой системе.

root@rhel53 ~# mount -t ext2 -o noexec /dev/hdb1 /home/project42
root@rhel53 ~# cp /bin/cat /home/project42
root@rhel53 ~# /home/project42/cat /etc/hosts
-bash: /home/project42/cat: Отказано в доступе
root@rhel53 ~# echo echo hello > /home/project42/helloscript
root@rhel53 ~# chmod +x /home/project42/helloscript
root@rhel53 ~# /home/project42/helloscript
-bash: /home/project42/helloscript: Отказано в доступе

7.5.3. Параметр nosuid

Параметр nosuid позволяет игнорировать биты setuid, установленные для бинарных файлов из смонтированной файловой системы.

Обратите внимание на то, что в случае использования данного параметра вы все также сможете устанавливать биты setuid для бинарных файлов из файловой системы.

root@rhel53 ~# mount -o nosuid /dev/hdb1 /home/project42
root@rhel53 ~# cp /bin/sleep /home/project42/
root@rhel53 ~# chmod 4555 /home/project42/sleep
root@rhel53 ~# ls -l /home/project42/sleep
-r-sr-xr-x 1 root root 19564 Июн 24 17:57 /home/project42/sleep

Но пользователи не смогут воспользоваться привилегиями, обусловленными наличием битов setuid.

root@rhel53 ~# su - paul
[paul@rhel53 ~]$ /home/project42/sleep 500 &
[1] 2876
[paul@rhel53 ~]$ ps -f 2876
UID PID PPID C STIME TTY STAT TIME CMD
paul 2876 2853 0 17:58 pts/0 S 0:00 /home/project42/sleep 500
[paul@rhel53 ~]$

7.5.4. Параметр noacl

Для предотвращения использования механизма управления правами доступа к файлам на основе списков контроля доступа следует использовать параметр noacl.

root@rhel53 ~# mount -o noacl /dev/hdb1 /home/project42

Описания других параметров монтирования файловых систем доступны на странице руководства для утилиты mount.

7.6. Монтирование удаленных файловых систем

7.6.1. Файловая система SMB/CIFS

Команда разработчиков проекта Samba (samba.org) осуществляет разработку системной службы для Unix/Linux, которая совместима с протоколом SMB/CIFS. Данный протокол используется главным образом компьютерами, работающими под управлением ОС производства компании Microsoft.

Соединение с сервером Samba (или компьютером, работающим под управлением ОС производства компании Microsoft) также может осуществляться с помощью утилиты mount.

В данном примере показана методика соединения с сервером, имеющим IP-адрес 10.0.0.42, для доступа к разделяемому ресурсу с именем data2.

[root@centos65 ~]# mount -t cifs -o user=paul //10.0.0.42/data2 /home/data2
Password:

[root@centos65 ~]# mount | grep cifs

Для корректного соединения с использованием описанного протокола необходимо установить дополнительные программные компоненты с помощью команды yum install cifs-client.

7.6.2. Файловая система NFS

Серверы Unix обычно используют протокол NFS (расшифровывается как Network File System — сетевая файловая система) для предоставления доступа к директориям по сети. Настройка сервера NFS будет обсуждаться позднее. Соединение же с сервером NFS со стороны клиента осуществляется также с помощью утилиты mount и очень похоже на соединение с локальным дисковым хранилищем.

Данная команда демонстрирует методику соединения с сервером NFS с именем server42, который предоставляет доступ к директории /srv/data. Точка монтирования (путь к которой приведен в конце строки команды) должна существовать.

[root@centos65 ~]# mount -t nfs server42:/srv/data /home/data

[root@centos65 ~]#

В том случае, если сервер с именем server42 имеет IP-адрес 10.0.0.42, вы также можете использовать запись:

[root@centos65 ~]# mount -t nfs 10.0.0.42:/srv/data /home/data

[root@centos65 ~]# mount | grep data

10.0.0.42:/srv/data on /home/data type nfs (rw,vers=4,addr=10.0.0.42,clienta\

ddr=10.0.0.33)

7.6.2. Специфичные для NFS параметры монтирования

bg: в случае неудачи при монтировании файловой системы повторять попытки в фоновом режиме.

fg: (используется по умолчанию) в случае неудачи при монтировании файловой системы повторять попытки, выводя информацию в текущую командную оболочку.

soft: прекратить процесс монтирования файловой системы после X неудачных попыток.

hard: (используется по умолчанию) продолжать предпринимать попытки монтирования файловой системы после неудач.

Комбинация параметров soft+bg позволяет осуществить наиболее быструю загрузку клиентской системы в случае неполадок с сервером NFS.

retrans=X: Попытаться соединиться с сервером X раз (по протоколу UDP).

tcp: Принудительно использовать протокол TCP (используется по умолчанию и всегда поддерживается).

udp: Принудительно использовать протокол UDP (не всегда поддерживается).

7.7. Практическое задание: монтирование файловых систем

1. Смонтируйте файловую систему, созданную в рамках малого раздела размером в 200 МБ, в точку монтирования /home/project22.
2. Смонтируйте файловую систему, созданную в рамках большого первичного раздела размером в 400 МБ, в точку монтирования /mnt, после чего скопируйте некоторые системные файлы в эту директорию (рекомендуется скопировать все содержимое директории /etc). После этого отмонтируйте файловую систему и повторно смонтируйте ее в точку монтирования /srv/nfs/salesnumbers в режиме только для чтения. Где оказались скопированные вами ранее файлы?
3. Проверьте корректность своих выполненных действий с помощью утилит fdisk, df и mount. Также изучите содержимое файлов /etc/mtab и /proc/mounts.
4. Сделайте так, чтобы монтирование обеих файловых систем осуществлялось автоматически на постоянной основе, после чего проверьте работоспособность использованного механизма монтирования.
5. Что случится, если вы смонтируете файловую систему в директорию, в которой уже содержатся файлы?
6. Что случится, если вы смонтируете две файловых системы в одну и ту же точку монтирования?

7 (дополнительное задание). Дайте пояснения относительно различий между данными утилитами: find, locate, updatedb, makewhatis, whereis, apropos, which и type.

8 (дополнительное задание). Выполните проверку файловой системы, созданной в разделе, который смонтирован в директорию /srv/nfs/salesnumbers.

7.8. Процедура корректного выполнения практического задания: монтирование файловых систем

1. Смонтируйте файловую систему, созданную в рамках малого раздела размером в 200 МБ, в точку монтирования /home/project22.

   mkdir /home/project22

mount /dev/sdc1 /home/project22

1. Смонтируйте файловую систему, созданную в рамках большого первичного раздела размером в 400 МБ, в точку монтирования /mnt, после чего скопируйте некоторые системные файлы в эту директорию (рекомендуется скопировать все содержимое директории /etc). После этого

отмонтируйте файловую систему и повторно смонтируйте ее в точку монтирования /srv/nfs/salesnumbers в режиме только для чтения. Где оказались скопированные вами ранее файлы?

mount /dev/sdb1 /mnt
cp -r /etc /mnt
ls -l /mnt
umount /mnt
ls -l /mnt
mkdir -p /srv/nfs/salesnumbers
mount /dev/sdb1 /srv/nfs/salesnumbers

Теперь вы можете обнаружить скопированные ранее файлы в директории /srv/nfs/salenumbers ...

Но физически эти файлы все также хранятся в рамках файловой системы ext3 на разделе жесткого диска, представленном файлом устройства /dev/sdb1

1. Проверьте корректность своих выполненных действий с помощью утилит fdisk, df и mount. Также изучите содержимое файлов /etc/mtab и /proc/mounts.

fdisk -l
df -h
mount

В выводах всех трех приведенных выше команд должна содержаться информация о смонтированных вами файловых системах и файлах устройств соответствующих разделов дисков.

grep project22 /etc/mtab
grep project22 /proc/mounts

1. Сделайте так, чтобы монтирование обеих файловых систем осуществлялось автоматически на постоянной основе, после чего проверьте работоспособность использованного механизма монтирования.

Добавьте следующие строки в файл /etc/fstab
/dev/sdc1 /home/project22 auto defaults 0 0
/dev/sdb1 /srv/nfs/salesnumbers auto defaults 0 0

1. Что случится, если вы смонтируете файловую систему в директорию, в которой уже содержатся файлы?

Уже имеющиеся файлы будут скрыты до момента отмонтирования файловой системы, к примеру, с помощью утилиты umount.

1. Что случится, если вы смонтируете две файловых системы в одну и ту же точку монтирования?

Будут доступны файлы только из последней смонтированной файловой системы.

7 (дополнительное задание). Дайте пояснения относительно различий между данными утилитами: find, locate, updatedb, makewhatis, whereis, apropos, which и type.

man find
man locate
...

8 (дополнительное задание). Выполните проверку файловой системы, созданной в разделе, который смонтирован в директорию /srv/nfs/salesnumbers.

umount /srv/nfs/salesnumbers (необязательная команда, которую рекомендуется выполнить)
fsck /dev/sdb

Глава 8. Инструменты для диагностики файловых систем

В данной главе описываются некоторые инструменты, которые могут использоваться для диагностики файловых систем помимо команд df -h и du -h. Эти инструменты позволят вам выявить множество проблем, связанных с файловыми системами и устройствами для хранения данных.

8.1. Утилита lsof

С помощью утилиты lsof вы можете получить список открытых файлов.

При использовании утилиты lsof без параметров будет выводиться список, содержащий все открытые файлы. В данном списке вы можете обнаружить строку команды (в данном случае это команда init), идентификатор созданного процесса в столбце PID (1) а также имя пользователя (root), с привилегиями которого была открыта корневая директория и файл /sbin/init. Данные из столбца FD (содержащего информацию о дескрипторе файла) говорят о том, что директория / являлась как корневой директорией (rtd), так и текущей рабочей директорией (cwd) при исполнении команды /sbin/init. В столбце FD может содержаться строка rtd, указывающая на то, что директория является корневой директорией, строка cwd, указывающая на то, что директория является текущей рабочей директорией и строка txt, указывающая не то, что открыт файл (включая файлы с данными и кодом).

root@debian7:~# lsof | head -4
COMMAND PID TID USER FD TYPE DEVICE SIZE/OFF NODE NAME
init 1 root cwd DIR 254,0 4096 2 /
init 1 root rtd DIR 254,0 4096 2 /
init 1 root txt REG 254,0 36992 130856 /sbin/init

В остальных случаях в столбце FD приводятся числовые значения дескрипторов с символами, соответствующими режимам открытия файлов, причем символ w соответствует режиму записи данных, r — режиму чтения данных, а u — режиму чтения и записи

данных. Вы можете вывести список файлов, открытых в рамках процесса с определенным идентификатором PID, воспользовавшись командой lsof -p PID. Для процесса init данная команда будет выглядеть следующим образом:

lsof -p 1

В приведенном ниже примере показана простая методика использования утилиты lsof для доказательства того, что текстовый редактор vi хранит файл с расширением .swp в открытом состоянии (даже в том случае, если исполнение соответствующего процесса приостанавливается в фоновом режиме) при работе с нашей недавно смонтированной файловой системой.

```
[root@RHEL65 ~]# df -h | grep sdb
/dev/sdb1 541M 17M 497M 4% /srv/project33
[root@RHEL65 ~]# vi /srv/project33/busyfile.txt
[1]+ Stopped vi /srv/project33/busyfile.txt
[root@RHEL65 ~]# lsof /srv/*
COMMAND PID USER FD TYPE DEVICE SIZE/OFF NODE NAME
vi 3243 root 3u REG 8,17 4096 12 /srv/project33/.busyfile.txt.swp
```

А здесь мы видим, что демон rsyslog открывает несколько файлов журналов для записи (как указано в строке FD).

```
root@debian7:~# lsof /var/log/*
COMMAND PID USER FD TYPE DEVICE SIZE/OFF NODE NAME
rsyslogd 2013 root 1w REG 254,0 454297 1308187 /var/log/syslog
rsyslogd 2013 root 2w REG 254,0 419328 1308189 /var/log/kern.log
rsyslogd 2013 root 5w REG 254,0 116725 1308200 /var/log/debug
rsyslogd 2013 root 6w REG 254,0 309847 1308201 /var/log/messages
rsyslogd 2013 root 7w REG 254,0 17591 1308188 /var/log/daemon.log
rsyslogd 2013 root 8w REG 254,0 101768 1308186 /var/log/auth.log
```

Вы можете указать имя интересующего вас пользователя в рамках команды lsof -u. В данном примере выводится информация о текущих рабочих директориях нескольких программ с интерфейсом командной строки, запущенных от лица пользователя paul.

```
[paul@RHEL65 ~]$ lsof -u paul | grep home
bash 3302 paul cwd DIR 253,0 4096 788024 /home/paul
lsof 3329 paul cwd DIR 253,0 4096 788024 /home/paul
grep 3330 paul cwd DIR 253,0 4096 788024 /home/paul
lsof 3331 paul cwd DIR 253,0 4096 788024 /home/paul
```

Совместно с параметром -u утилиты lsof также может быть использовать символ ^, соответствующий логической операции не. Исходя из этого, команда для получения информации обо всех открытых файлах, за исключением тех файлов, которые открыты пользователем root, будет выглядеть следующим образом:

lsof -u^root

8.2. Утилита fuser

С помощью утилиты fuser вы можете получить информацию о процессах, которые работают с указанной файловой системой.

В следующем примере мы приостанавливаем исполнение процесса текстового редактора vi в фоновом режиме и используем утилиту fuser для вывода идентификатора процесса, который работает с указанной файловой системой.

```
[root@RHEL65 ~]# jobs
[1]+ Stopped vi /srv/project33/busyfile.txt
[root@RHEL65 ~]# fuser -m /srv/project33/
/srv/project33/: 3243
```

В случае добавления параметра -u утилита также будет выводить имя пользователя, от лица которого был запущен процесс.

```
[root@RHEL65 ~]# fuser -m -u /srv/project33/
/srv/project33/: 3243(root)
```

Вы можете быстро уничтожить все процессы, которые работают с определенным файлом (или директорией), воспользовавшись параметром -k.

```
[root@RHEL65 ~]# fuser -m -k -u /srv/project33/
/srv/project33/: 3243(root)
[1]+ Killed vi /srv/project33/busyfile.txt
[root@RHEL65 ~]# fuser -m -u /srv/project33/
[root@RHEL65 ~]#
```

В данном примере выводится информация обо всех процессах, которые используют текущую директорию (в данном случае этими процессами являются bash и vi).

```
root@debian7:~/test42# vi file42
[1]+ Stopped vi file42
root@debian7:~/test42# fuser -v .
ПОЛЬЗ-ЛЬ PID ДОСТУП КОМАНДА
/root/test42: root 2909 ..c.. bash
root 3113 ..c.. vi
```

А в данном примере показано, что при использовании команды vi осуществляется доступ к файлу /usr/bin/vim.basic, как к исполняемому файлу.

```
root@debian7:~/test42# fuser -v $(which vi)
ПОЛЬЗ-ЛЬ PID ДОСТУП КОМАНДА
/usr/bin/vim.basic: root 3113 ...e. vi
```

В последнем примере показана методика поиска процесса, который осуществляет доступ к определенному файлу.

```
[root@RHEL65 ~]# vi /srv/project33/busyfile.txt
[1]+ Stopped vi /srv/project33/busyfile.txt
[root@RHEL65 ~]# fuser -v -m /srv/project33/busyfile.txt
```

ПОЛЬЗ-ЛЬ PID ДОСТУП КОМАНДА
/srv/project33/busyfile.txt:
root 13938 F.... vi
[root@RHEL65 ~]# ps -fp 13938
UID PID PPID C STIME TTY TIME CMD
root 13938 3110 0 15:47 pts/0 00:00:00 vi /srv/project33/busyfile.txt

8.3. Утилита chroot

Утилита chroot позволяет открыть командную оболочку с измененной корневой директорией. Все файлы, находящиеся вне этой директории, скрываются от пользователя.

В примере ниже мы будем предполагать, что наша система не может загрузиться (например, из-за проблемы с файлом /etc/fstab или невозможности монтирования корневой файловой системы).

Мы будем использовать live-систему (загруженную с накопителя CD/DVD/USB) для диагностики нашего сервера. Live-система не будет использовать наш основной жесткий диск в качестве устройства с корневой директорией.

root@livecd:~# df -h | grep root
rootfs 186M 11M 175M 6% /
/dev/loop0 807M 807M 0 100% /lib/live/mount/rootfs/filesystem.squashfs
root@livecd:~# mount | grep root
/dev/loop0 on /lib/live/mount/rootfs/filesystem.squashfs type squashfs (ro)

Для последующей проверки мы можем создать тестовый файл и директорию в нашей текущей корневой файловой системе.

root@livecd:~# touch /file42
root@livecd:~# mkdir /dir42
root@livecd:~# ls /
bin dir42 home lib64 opt run srv usr
boot etc initrd.img media proc sbin sys var
dev file42 lib mnt root selinux tmp vmlinuz

В первую очередь мы должны смонтировать корневую файловую систему с нашего диска (который является частью массива lvm, поэтому мы будем использовать файл устройства /dev/mapper вместо /dev/sda5).

root@livecd:~# mount /dev/mapper/packer—debian—7-root /mnt

Теперь мы готовы к использованию утилиты chroot для замены корневой файловой системы на файловую систему с нашего диска, смонтированную в директорию.

root@livecd:~# cd /mnt
root@livecd:/mnt# chroot /mnt
root@livecd:/# ls /
bin dev initrd.img lost+found opt run srv usr vmlinuz
boot etc lib media proc sbin sys vagrant
data home lib64 mnt root selinux tmp var

Наши тестовые файл и директория (file42 и dir42) не отображаются из-за того, что они находятся вне установленной с помощью утилиты chroot корневой директории.

Обратите внимание на то, что имя узла chroot-окружения идентично существующему имени узла.

Для завершения работы chroot-окружением следует использовать команду exit:

root@livecd:/# exit
exit
root@livecd:~# ls /
bin dir42 home lib64 opt run srv usr
boot etc initrd.img media proc sbin sys var
dev file42 lib mnt root selinux tmp vmlinuz

8.4. Утилита iostat

Утилита iostat периодически по прошествии заданного промежутка времени выводит статистические данные, связанные с использованием операций ввода-вывода. Также в вывод утилиты включены некоторые обобщенные показатели загрузки центрального процессора системы. В данном примере утилита iostat через каждые 10 секунд выводит статистические данные, на основе которых можно сделать вывод о том, что на диски, представленные файлами устройств /dev/sdc и /dev/sde, осуществляется интенсивная запись данных.

[root@RHEL65 ~]# iostat 10 3
Linux 2.6.32-431.el6.x86_64 (RHEL65) 06/16/2014 x86_64 (1 CPU)
avg-cpu: %user %nice %system %iowait %steal %idle
5.81 0.00 3.15 0.18 0.00 90.85
Device: tps Blk_read/s Blk_wrtn/s Blk_read Blk_wrtn
sda 42.08 1204.10 1634.88 1743708 2367530
sdb 1.20 7.69 45.78 11134 66292
sdc 0.92 5.30 45.82 7672 66348
sdd 0.91 5.29 45.78 7656 66292
sde 1.04 6.28 91.49 9100 132496
sdf 0.70 3.40 91.46 4918 132440
sdg 0.69 3.40 91.46 4918 132440
dm-0 191.68 1045.78 1362.30 1514434 1972808
dm-1 49.26 150.54 243.55 218000 352696
avg-cpu: %user %nice %system %iowait %steal %idle
56.11 0.00 16.83 0.10 0.00 26.95
Device: tps Blk_read/s Blk_wrtn/s Blk_read Blk_wrtn
sda 257.01 10185.97 76.95 101656 768
sdb 0.00 0.00 0.00 0 0

sdc 3.81 1.60 2953.11 16 29472
sdd 0.00 0.00 0.00 0 0
sde 4.91 1.60 4813.63 16 48040
sdf 0.00 0.00 0.00 0 0
sdg 0.00 0.00 0.00 0 0
dm-0 283.77 10185.97 76.95 101656 768
dm-1 0.00 0.00 0.00 0 0
avg-cpu: %user %nice %system %iowait %steal %idle
67.65 0.00 31.11 0.11 0.00 1.13
Device: tps Blk_read/s Blk_wrtn/s Blk_read Blk_wrtn
sda 466.86 26961.09 178.28 238336 1576
sdb 0.00 0.00 0.00 0 0
sdc 31.45 0.90 24997.29 8 220976
sdd 0.00 0.00 0.00 0 0
sde 0.34 0.00 5.43 0 48
sdf 0.00 0.00 0.00 0 0
sdg 0.00 0.00 0.00 0 0
dm-0 503.62 26938.46 178.28 238136 1576
dm-1 2.83 22.62 0.00 200 0
[root@RHEL65 ~]#

Дополнительные параметры позволяют указывать имена файлов устройств дисков для мониторинга операций ввода-вывода(в данном случае с временным интервалом в 5 секунд):

iostat sdd sde sdf 5

Или выводить статистику для разделов жестких дисков:

iostat -p sde -p sdf 5

8.5. Утилита iotop

Утилита iotop работает аналогично утилите top, но упорядочивает процессы в зависимости от интенсивности операций ввода-вывода, а не от нагрузки на центральный процессор.

По умолчанию утилита iotop будет выводить информацию обо всех процессах. В данном примере используется параметр iotop -o для вывода информации исключительно о тех процессах, которые осуществляют операции ввода-вывода.

[root@RHEL65 ~]# iotop -o
Total DISK READ: 8.63 M/s | Total DISK WRITE: 0.00 B/s
TID PRIO USER DISK READ DISK WRITE SWAPIN IO> COMMAND
15000 be/4 root 2.43 M/s 0.00 B/s 0.00 % 14.60 % tar cjf /srv/di...
25000 be/4 root 6.20 M/s 0.00 B/s 0.00 % 6.15 % tar czf /srv/di...
24988 be/4 root 0.00 B/s 7.21 M/s 0.00 % 0.00 % gzip
25003 be/4 root 0.00 B/s 1591.19 K/s 0.00 % 0.00 % gzip
25004 be/4 root 0.00 B/s 193.51 K/s 0.00 % 0.00 % bzip2

Используйте параметр -b для задействования режима вывода данных утилиты iotop в форме журнала (вместо стандартного интерактивного режима вывода данных).

[root@RHEL65 ~]# iotop -bod 10
Total DISK READ: 12.82 M/s | Total DISK WRITE: 5.69 M/s
TID PRIO USER DISK READ DISK WRITE SWAPIN IO COMMAND
25153 be/4 root 2.05 M/s 0.00 B/s 0.00 % 7.81 % tar cjf /srv/di...
25152 be/4 root 10.77 M/s 0.00 B/s 0.00 % 2.94 % tar czf /srv/di...
25144 be/4 root 408.54 B/s 0.00 B/s 0.00 % 0.05 % python /usr/sbi...
12516 be/3 root 0.00 B/s 1491.33 K/s 0.00 % 0.04 % [jbd2/sdc1-8]
12522 be/3 root 0.00 B/s 45.48 K/s 0.00 % 0.01 % [jbd2/sde1-8]
25158 be/4 root 0.00 B/s 0.00 B/s 0.00 % 0.00 % [flush-8:64]
25155 be/4 root 0.00 B/s 493.12 K/s 0.00 % 0.00 % bzip2
25156 be/4 root 0.00 B/s 2.81 M/s 0.00 % 0.00 % gzip
25159 be/4 root 0.00 B/s 528.63 K/s 0.00 % 0.00 % [flush-8:32]

А это пример использования утилиты iotop для отслеживания операций дискового ввода-вывода, осуществляемых в течение каждых десяти секунд от лица пользователя с именем vagrant (средствами единственного процесса, запущенного данным пользователем, но данное обстоятельство не принципиально и может не рассматриваться). Параметр -a позволяет накапливать статистические данные об операциях ввода-вывода.

[root@RHEL65 ~]# iotop -q -a -u vagrant -b -p 5216 -d 10 -n 10
Total DISK READ: 0.00 B/s | Total DISK WRITE: 0.00 B/s
TID PRIO USER DISK READ DISK WRITE SWAPIN IO COMMAND
5216 be/4 vagrant 0.00 B 0.00 B 0.00 % 0.00 % gzip
Total DISK READ: 818.22 B/s | Total DISK WRITE: 20.78 M/s
5216 be/4 vagrant 0.00 B 213.89 M 0.00 % 0.00 % gzip
Total DISK READ: 2045.95 B/s | Total DISK WRITE: 23.16 M/s
5216 be/4 vagrant 0.00 B 430.70 M 0.00 % 0.00 % gzip
Total DISK READ: 1227.50 B/s | Total DISK WRITE: 22.37 M/s
5216 be/4 vagrant 0.00 B 642.02 M 0.00 % 0.00 % gzip
Total DISK READ: 818.35 B/s | Total DISK WRITE: 16.44 M/s
5216 be/4 vagrant 0.00 B 834.09 M 0.00 % 0.00 % gzip
Total DISK READ: 6.95 M/s | Total DISK WRITE: 8.74 M/s
5216 be/4 vagrant 0.00 B 920.69 M 0.00 % 0.00 % gzip
Total DISK READ: 21.71 M/s | Total DISK WRITE: 11.99 M/s

8.6. Утилита vmstat

Хотя утилита vmstat и является инструментом, применяемым главным образом для мониторинга работы оперативной памяти, она заслуживает упоминания в данной главе ввиду того, что она также выводит информацию о суммарном объеме передаваемых благодаря операциям ввода-вывода данных для блочных устройств и пространства подкачки.

В данном примере показан вывод описанной утилиты, содержащий количественные показатели дисковой активности (в

столбцах под заголовком -----io----) без учета пространства подкачки.

```
[root@RHEL65 ~]# vmstat 5 10
procs ----------memory---------- ---swap-- -----io---- --system-- -----cpu-----
r b swpd free buff cache si so bi bo in cs us sy id wa st
0 0 5420 9092 14020 340876 7 12 235 252 77 100 2 1 98 0 0
2 0 5420 6104 13840 338176 0 0 7401 7812 747 1887 38 12 50 0 0
2 0 5420 10136 13696 336012 0 0 11334 14 1725 4036 76 24 0 0 0
0 0 5420 14160 13404 341552 0 0 10161 9914 1174 1924 67 15 18 0 0
0 0 5420 14300 13420 341564 0 0 0 16 28 18 0 0 100 0 0
0 0 5420 14300 13420 341564 0 0 0 0 22 16 0 0 100 0 0
...
[root@RHEL65 ~]#
```

Вы можете воспользоваться полезной возможностью утилиты vmstat, заключающейся в поддержке вывода значений объемов памяти в килобайтах, мегабайтах или даже в кибибайтах и мибибайтах благодаря наличию параметра -S (после которого должен следовать идентификатор единиц измерения k, K, m или M).

```
[root@RHEL65 ~]# vmstat -SM 5 10
procs ----------memory---------- ---swap-- -----io---- --system-- -----cpu-----
r b swpd free buff cache si so bi bo in cs us sy id wa st
0 0 5 14 11 334 0 0 259 255 79 107 2 1 97 0 0
0 0 5 14 11 334 0 0 0 2 21 18 0 0 100 0 0
0 0 5 15 11 334 0 0 6 0 35 31 0 0 100 0 0
2 0 5 6 11 336 0 0 17100 7814 1378 2945 48 21 31 0 0
2 0 5 6 11 336 0 0 13193 14 1662 3343 78 22 0 0 0
2 0 5 13 11 330 0 0 11656 9781 1419 2642 82 18 0 0 0
2 0 5 9 11 334 0 0 10705 2716 1504 2657 81 19 0 0 0
1 0 5 14 11 336 0 0 6467 3788 765 1384 43 9 48 0 0
0 0 5 14 11 336 0 0 0 13 28 24 0 0 100 0 0
0 0 5 14 11 336 0 0 0 0 20 15 0 0 100 0 0
[root@RHEL65 ~]#
```

Утилита vmstat будет обсуждаться и в других главах.

8.7. Практическое задание: инструменты для диагностики файловых систем

1. Очень важно научиться работать с описанными в главе инструментами перед тем, как возникнут реальные проблемы с файловыми системами. Данное практическое задание поспособствует ознакомлению с описанными инструментами, а также позволит составить представление о параметрах корректно функционирующих систем.
2. Прочитайте раздел, посвященный утилите fuser, а также страницу руководства данной утилиты. Используйте эту утилиту для получения информации об открытых вами файлах.
3. Прочитайте раздел, посвященный утилите lsof, а также страницу руководства данной утилиты. Используйте эту утилиту для получения информации об открытых вами файлах.
4. Загрузите live-систему на компьютере (виртуальном или реальном) и смените корневую директорию с помощью утилиты chroot.
5. Запустите одну или несколько задач, интенсивно использующих диск, после чего попробуйте отследить эти задачи с помощью утилит iostat и iotop (сравните вывод этих утилит с выводом утилиты vmstat).

8.8. Корректная процедура выполнения практического задания: инструменты для диагностики файловых систем

1. Очень важно научиться работать с описанными в главе инструментами перед тем, как возникнут реальные проблемы с файловыми системами. Данное практическое задание поспособствует ознакомлению с описанными инструментами, а также позволит составить представление о параметрах корректно функционирующих систем.
2. Прочитайте раздел, посвященный утилите fuser, а также страницу руководства данной утилиты. Используйте эту утилиту для получения информации об открытых вами файлах.

fuser -vm /home

1. Прочитайте раздел, посвященный утилите lsof, а также страницу руководства данной утилиты. Используйте эту утилиту для получения информации об открытых вами файлах.

lsof -u whoami

1. Загрузите live-систему на компьютере (виртуальном или реальном) и смените корневую директорию с помощью утилиты chroot.

```
mkdir /mnt/rootfs
mount /dev/sda6 /mnt/rootfs
chroot /mnt/rootfs
```

1. Запустите одну или несколько задач, интенсивно использующих диск, после чего попробуйте отследить эти задачи с помощью утилит iostat и iotop (сравните вывод этих утилит с выводом утилиты vmstat).

Iostat

iotop
vmstat

Глава 9. Знакомство с идентификаторами UUID

Идентификатор UUID (Universally Unique Identifier) используется для уникальной идентификации объектов. В соответствии со стандартом, длина идентификатора равна 128 битам, что позволяет любому пользователю без лишних сложностей создавать уникальные идентификаторы UUID.

В данной главе приводится краткий обзор идентификаторов UUID.

9.1. Об уникальных объектах

В составе устаревших версий дистрибутивов Linux поставлялась утилита vol_id, которая предназначалась для вывода идентификатора UUID указанной файловой системы.

root@debian5:~# vol_id --uuid /dev/sda1
193c3c9b-2c40-9290-8b71-4264ee4d4c82

В дистрибутиве RedHat Enterprise Linux 5 утилита vol_id была размещена по пути /lib/udev/vol_id, причем путь к директории с данной утилитой не добавлялся в список директорий для поиска бинарных файлов, являющийся значением переменной окружения $PATH. Синтаксис данной утилиты также немного отличался от синтаксиса аналогичной утилиты из состава дистрибутивов Debian/Ubuntu/Mint.

root@rhel53 ~# /lib/udev/vol_id -u /dev/hda1
48a6a316-9ca9-4214-b5c6-e7b33a77e860

Данная утилита не доступна после стандартной установки дистрибутивов RHEL 6 и Debian 6.

9.2. Утилита tune2fs

Для определения идентификатора UUID файловой системы следует использовать утилиту tune2fs.

[root@RHEL5 ~]# tune2fs -l /dev/sda1 | grep UUID
Filesystem UUID: 11cfc8bc-07c0-4c3f-9f64-78422ef1dd5c
[root@RHEL5 ~]# /lib/udev/vol_id -u /dev/sda1
11cfc8bc-07c0-4c3f-9f64-78422ef1dd5c

9.3. Утилита uuid

Возможности утилиты uuid, являющейся инструментом для генерации идентификаторов UUID, подробно описаны на странице руководства.

[root@rhel65 ~]# yum install uuid
(вывод сокращен)
[root@rhel65 ~]# man uuid

9.4. Идентификаторы UUID в файле /dev/fstab

Вы можете использовать идентификаторы UUID для универсальной уникальной идентификации разделов жестких дисков в рамках файла /etc/fstab. Имя файла устройства может меняться в зависимости от количества дисковых устройств, подключенных к системе в процессе ее загрузки, но идентификаторы UUID устройств не изменятся ни при каких обстоятельствах.

В первую очередь мы должны использовать утилиту tune2fs для того, чтобы выяснить идентификатор UUID раздела жесткого диска.

[root@RHEL5 ~]# tune2fs -l /dev/sdc1 | grep UUID
Filesystem UUID: 7626d73a-2bb6-4937-90ca-e451025d64e8

После этого следует убедиться в том, что данный идентификатор корректно используется в файле /etc/fstab, заменяя собой непостоянное имя файла устройства, представляющего раздел жесткого диска.

[root@RHEL5 ~]# grep UUID /etc/fstab
UUID=7626d73a-2bb6-4937-90ca-e451025d64e8 /home/pro42 ext3 defaults 0 0

Теперь мы можем смонтировать файловую систему с интересующего нас раздела жесткого диска в точку монтирования, объявленную в файле /etc/fstab.

[root@RHEL5 ~]# mount /home/pro42
[root@RHEL5 ~]# df -h | grep 42
/dev/sdc1 397M 11M 366M 3% /home/pro42

Реальная же проверка должна заключаться в отключении от системы диска, представленного файлом устройства /dev/sdb, перезагрузке системы и проверке результата. После загрузки диск, который ранее был представлен с помощью файла устройства /dev/sdc, будет представлен с помощью файла устройства /dev/sdb.

[root@RHEL5 ~]# tune2fs -l /dev/sdb1 | grep UUID
Filesystem UUID: 7626d73a-2bb6-4937-90ca-e451025d64e8

Но благодаря использованию идентификаторов UUID в файле /etc/fstab, в объявленную точку монтирования будет смонтирована файловая система с раздела того же диска, что и ранее.

[root@RHEL5 ~]# df -h | grep sdb
/dev/sdb1 397M 11M 366M 3% /home/pro42

9.5. Идентификаторы UUID для загрузочных устройств

Современные дистрибутивы Linux (Debian, Ubuntu, ...) используют системный загрузчик grub и идентификаторы UUID для идентификации разделов жестких дисков с корневыми файловыми

системами.

В данном примере показано, как имя файла устройства раздела с корневой файловой системой из директивы root=/dev/sda1 может быть заменено на идентификатор UUID этого же раздела.

```
title Ubuntu 9.10, kernel 2.6.31-19-generic
uuid f001ba5d-9077-422a-9634-8d23d57e782a
kernel /boot/vmlinuz-2.6.31-19-generic \
root=UUID=f001ba5d-9077-422a-9634-8d23d57e782a ro quiet splash
initrd /boot/initrd.img-2.6.31-19-generic
```

Пример выше содержит всего четыре строки. Строка, начинающаяся с директивы «root=», является продолжением строки kernel.

В дистрибутивах RHEL и CentOS после стандартной процедуры установки загрузка системы осуществляется с томов LVM.

9.6. Практическое задание: идентификаторы UUID и файловые системы

1. С помощью утилиты tune2fs определите идентификатор UUID одного из разделов вашего жесткого диска с файловой системой ext3 (также используйте утилиту vol_id в том случае, если вы работаете с дистрибутивом RHEL5).
2. Используйте полученный идентификатор UUID в файле /etc/fstab и проверьте его работоспособность с помощью утилиты mount.

3 (дополнительное задание). Также проверьте работоспособность полученного идентификатора, отключив от системы один из жестких дисков (с целью изменения имен файлов устройств, представляющих жесткие диски). Вы можете изменить настройки vmware/Virtualbox для отключения жесткого диска.

1. Найдите директиву «root=» в файле /boot/grub/menu.lst. (Позднее в рамках данного курса мы рассмотрим вопросы, касающиеся редактирования упомянутого файла).

5 (дополнительное задание для пользователей Ubuntu). Замените в файле /boot/grub/menu.lst путь к файлу устройства /dev/xxx, представляющему раздел жесткого диска с корневой файловой системой, на идентификатор UUID (используйте для этого дополнительную директиву). Проверьте возможность загрузки системы после внесения изменений в упомянутый файл.

9.7. Корректная процедура выполнения практического задания: идентификаторы UUID и файловые системы

1. С помощью утилиты tune2fs определите идентификатор UUID одного из разделов вашего жесткого диска с файловой системой ext3 (также используйте утилиту vol_id в том случае, если вы работаете с дистрибутивом RHEL5).

```
root@rhel55:~# /lib/udev/vol_id -u /dev/hda1
60926898-2c78-49b4-a71d-c1d6310c87cc
root@ubu1004:~# tune2fs -l /dev/sda2 | grep UUID
Filesystem UUID: 3007b743-1dce-2d62-9a59-cf25f85191b7
```

1. Используйте полученный идентификатор UUID в файле /etc/fstab и проверьте его работоспособность с помощью утилиты mount.

```
tail -1 /etc/fstab
UUID=60926898-2c78-49b4-a71d-c1d6310c87cc /home/pro42 ext3 defaults 0 0
```

3 (дополнительное задание). Также проверьте работоспособность полученного идентификатора, отключив от системы один из жестких дисков (с целью изменения имен файлов устройств, представляющих жесткие диски). Вы можете изменить настройки vmware/Virtualbox для отключения жесткого диска.

1. Найдите директиву «root=» в файле /boot/grub/menu.lst. (Позднее в рамках данного курса мы рассмотрим вопросы, касающиеся редактирования упомянутого файла).

```
paul@deb503:~$ grep [#] /boot/grub/menu.lst | grep root=
kernel /boot/vmlinuz-2.6.26-2-686 root=/dev/hda1 ro selinux=1 quiet
kernel /boot/vmlinuz-2.6.26-2-686 root=/dev/hda1 ro selinux=1 single
```

5 (дополнительное задание для пользователей Ubuntu). Замените в файле /boot/grub/menu.lst путь к файлу устройства /dev/xxx, представляющему раздел жесткого диска с корневой файловой системой, на идентификатор UUID (используйте для этого дополнительную директиву). Проверьте возможность загрузки системы после внесения изменений в упомянутый файл.

Глава 10. Знакомство с технологией RAID

10.1. Аппаратная или программная реализация

Избыточный массив независимых (а в оригинале недорогих) дисков (Redundant Array of Inedependent (Inexpensive) Disks) или RAID-массив может быть реализован аппаратно или программно. Аппаратный RAID-массив обходится более дорого, но позволяет достичь более высокой производительности. Программный RAID-массив более дешев и прост в управлении, но использует ресурсы центрального процессора и оперативной памяти вашего компьютера.

Десять лет назад никто не мог поспорить с утверждением о том, что аппаратный RAID-массив является более удачным решением, но положение дел изменилась после оптимизации таких

технологий, как mdadm, lvm или даже zfs с целью упрощения методики управления используемыми дисками. Хотя программные RAID-массивы и оказывают значительную нагрузку на центральный процессор, следует принимать во внимание и тот факт, что в течение последних лет производительность центральных процессоров неуклонно возрастала.

10.2. Уровни RAID-массивов

10.2.1. Дисковый массив RAID 0

В рамках дискового массива RAID 0 используются два или большее количество дисков, а сам массив обычно называется дисковым массивом с чередованием (stripping, stripe set или stripe volume). Данные разделяются на фрагменты, которые равномерно распределяются по каждому из дисков массива. Главное преимущество дискового массива RAID 0 заключается в возможности создания виртуальных дисков большего размера. Дисковый массив RAID 0 является единственным стандартизированным типом дискового массива RAID без избыточного копирования данных.

10.2.2. Дисковый массив RAID типа JBOD

В рамках дискового массива RAID типа JBOD используются два или большее количество дисков, а сам массив обычно называется дисковым массивом со связыванием (concatenating, spanning, spanned set или spanned volume). Данные записываются на первый диск до момента его полного заполнения. После этого данные записываются на следующий диск и так далее... Главное достоинство дискового массива RAID типа JBOD (расшифровывается как Just a Bunch of Disks — просто связка дисков) заключается в возможности создания виртуальных дисков большего размера. Дисковый массив RAID типа JBOD не предусматривает возможности избыточного копирования данных.

10.2.3. Дисковый массив RAID 1

В рамках дискового массива RAID 1 используются ровно два диска, а сам массив обычно называется дисковым массивом с зеркалированием (mirroring, mirror set или mirrored volume). Все данные, записываемые в дисковый массив, записываются на каждый диск. Главное достоинство дискового массива RAID 1 заключается в избыточной записи данных. Главный недостаток дискового массива данного типа заключается в потере как минимум половины доступного дискового пространства (другими словами, общая стоимость дисковых устройств возрастает как минимум вдвое).

10.2.4. А что насчет дисковых массивов RAID 2, 3 и 4?

Дисковый массив RAID 2 использует чередование данных на уровне битов, RAID 3 — на уровне байт, а RAID 4 работает аналогично RAID 5, но использует отдельный диск для хранения данных четности. Дисковый массив последнего типа обычно медленнее массива RAID 5, так как при каждой операции записи данных также осуществляется операция записи данных четности на выделенный для этих целей (замедляющий работу массива) диск. Вероятность обнаружения этих уровней RAID в составе систем, предназначенных для промышленной эксплуатации, чрезвычайно мала.

10.2.5. Дисковый массив RAID 5

В рамках дискового массива RAID 5 используются три или большее количество дисков, на каждом из которых хранятся фрагменты данных. При каждой операции записи данных в массив на один из дисков будут записываться данные четности. В отличие от дискового массива RAID 4 данные четности будут размещаться на всех дисках массива. Главное достоинство рассматриваемого дискового массива RAID 5 заключается в возможности восстановления всех данных в случае отказа одного из дисковых устройств.

10.2.6. Дисковый массив RAID 6

Дисковый массив RAID 6 очень похож на дисковый массив RAID 5, но в нем используются по две копии данных четности, следовательно дисковый массив RAID 6 защищает данные даже в случае отказа двух дисковых устройств. В рамках файловой системы zfs из состава операционной системы Oracle Solaris данный тип дискового массива носит имя raidz2 (также в рамках упомянутой файловой системы реализован тип дискового массива с названием raidz3, предусматривающий трехкратное копирование данных четности).

10.2.7. Дисковый массив RAID 0+1

Дисковый массив RAID 0+1 является зеркалируемым дисковым массивом (1) с чередованием данных (0). Это означает, что в первую очередь вам придется создать два дисковых массива с чередованием данных RAID 0, после чего объединить их в рамках одного зеркалируемого дискового массива RAID 1. К примеру, в том случае, если у вас имеется шесть дисков объемом в 100 ГБ, каждый из дисковых массивов с чередованием данных будет иметь объем в 300 ГБ. После объединения этих дисковых массивов в рамках одного дискового массива общий объем последнего будет равен 300 ГБ. Дисковый массив RAID 0+1 может пережить выход из строя одного из дисковых устройств. При этом он переживет выход из строя второго диска только в том случае, если этот диск содержит тот же набор фрагментов данных, что и вышедший из строя диск.

10.2.8. Дисковый массив RAID 1+0

Дисковый массив RAID 1+0 является дисковым массивом с

чередованием данных (0), состоящим из зеркалируемых дисковых массивов (1). К примеру, в том случае, если у вас имеется шесть дисков объемом в 100 ГБ каждый, в первую очередь вам придется создать три зеркалируемых дисковых массива объемом в 100 ГБ каждый. После этого вы можете объединить их в рамках одного дискового массива с разделением данных объемом в 300 ГБ. В случае использования описанной структуры дискового массива, он может пережить выход из строя до трех дисков с тем условием, что все вышедшие из строя диски не находятся в составе одного из зеркалируемых дисковых массивов.

10.2.9. Дисковый массив RAID 50

Дисковый массив RAID 50 является дисковым массивом с чередованием данных, состоящим из дисковых массивов RAID 5. Если у вас имеется девять дисков объемом в 100 ГБ каждый, вы можете создать три массива RAID 5 объемом в 200 ГБ каждый. После этого вы можете скомбинировать их в рамках одного большого дискового массива с чередованием данных.

10.2.10. Множество других типов дисковых массивов

Также на практике может использоваться множество других комбинаций описанных типов дисковых массивов RAID, таких, как RAID 30, 51, 60, 100, 150, ...

10.3. Создание программного массива RAID 5

10.3.1. Имеются ли в наличии три диска?

В первую очередь вам придется подключить к вашему компьютеру три диска. В данном примере мы добавили три новых диска объемом в восемь гигабайт каждый. Для проверки корректности подключения дисков следует воспользоваться командой fdisk -l.

```
[root@rhel6c ~]# fdisk -l 2> /dev/null | grep MB
Disk /dev/sdb: 8589 MB, 8589934592 bytes
Disk /dev/sdc: 8589 MB, 8589934592 bytes
Disk /dev/sdd: 8589 MB, 8589934592 bytes
```

10.3.2. Тип раздела fd

Следующий шаг заключается в создании раздела типа fd на каждом из подключенных дисков. Тип раздела fd позволяет задействовать технологию автоматического обнаружения разделов дисковых массивов RAID в Linux (Linux RAID autodetect). Рассмотрите приведенный ниже (сокращенный) пример:

```
[root@rhel6c ~]# fdisk /dev/sdd
...
Command (m for help): n
Command action
e extended
p primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-1044, default 1):
Using default value 1
Last cylinder, +cylinders or +size{K,M,G} (1-1044, default 1044):
Using default value 1044
Command (m for help): t
Selected partition 1
Hex code (type L to list codes): fd
Changed system type of partition 1 to fd (Linux raid autodetect)
Command (m for help): w
The partition table has been altered!
Calling ioctl() to re-read partition table.
Syncing disks.
```

10.3.3. Проверка корректности создания разделов на дисках

Теперь все три диска готовы к работе в составе дискового массива RAID 5, поэтому нам остается лишь передать системе информацию о том, как работать ними.

```
[root@rhel6c ~]# fdisk -l 2> /dev/null | grep raid
/dev/sdb1 1 1044 8385898+ fd Linux raid autodetect
/dev/sdc1 1 1044 8385898+ fd Linux raid autodetect
/dev/sdd1 1 1044 8385898+ fd Linux raid autodetec
```

10.3.4. Создание дискового массива RAID 5

Обычно следующий шаг заключается в создании таблицы разделов RAID в рамках файла /etc/raidtab. Но на сегодняшний день вы можете ограничиться только вызовом утилиты mdadm с передачей корректных параметров.

Приведенная ниже строка команды разделена на две строки для удобного размещения на странице данной книги, но вам в любом случае придется передать утилите все параметры в одной строке без использования символа обратного слэша (\).

```
[root@rhel6c ~]# mdadm --create /dev/md0 --chunk=64 --level=5
--raid-\
devices=3 /dev/sdb1 /dev/sdc1 /dev/sdd1
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md0 started.
```

В примере ниже приведен сокращенный вывод команды fdisk -l, содержащий информацию о созданном дисковом массиве RAID 5.

```
[root@rhel6c ~]# fdisk -l /dev/md0
Disk /dev/md0: 17.2 GB, 17172135936 bytes
2 heads, 4 sectors/track, 4192416 cylinders
Units = cylinders of 8 * 512 = 4096 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 65536 bytes / 131072 bytes
```

Disk identifier: 0x00000000
Disk /dev/md0 doesn’t contain a valid partition table
Мы можем использовать созданный дисковый массив RAID 5 при рассмотрении следующей темы: LVM.

10.3.5. Файл /proc/mdstat

Информация о состоянии устройств из состава дискового массива RAID может быть получена из файла /proc/mdstat. В данном примере показано содержимое этого файла, при рассмотрении которого можно сделать вывод о том, что в текущий момент идет процесс пересборки дискового массива RAID 5.

[root@rhel6c ~]# cat /proc/mdstat
Personalities : [raid6] [raid5] [raid4]
md0 : active raid5 sdd1[3] sdc1[1] sdb1[0]
16769664 blocks super 1.2 level 5, 64k chunk, algorithm 2 [3/2] [UU_]
[===========⇒.....] recovery = 62.8% (5266176/8384832) finish=0\
3min speed=139200K/sec

А в этом примере показано содержимое этого же файла, соответствующее активному состоянию дискового массива RAID 5.

[root@rhel6c ~]# cat /proc/mdstat
Personalities : [raid6] [raid5] [raid4]
md0 : active raid5 sdd1[3] sdc1[1] sdb1[0]
16769664 blocks super 1.2 level 5, 64k chunk, algorithm 2 [3/3] [UUU]

10.3.6. Команда mdadm --detail

Используйте команду mdadm --detail для получения информации о виртуальном устройстве дискового массива RAID.

[root@rhel6c ~]# mdadm --detail /dev/md0
/dev/md0:
Version : 1.2
Creation Time : Sun Jul 17 13:48:41 2011
Raid Level : raid5
Array Size : 16769664 (15.99 GiB 17.17 GB)
Used Dev Size : 8384832 (8.00 GiB 8.59 GB)
Raid Devices : 3
Total Devices : 3
Persistence : Superblock is persistent
Update Time : Sun Jul 17 13:49:43 2011
State : clean
Active Devices : 3
Working Devices : 3
Failed Devices : 0
Spare Devices : 0
Layout : left-symmetric
Chunk Size : 64K
Name : rhel6c:0 (local to host rhel6c)
UUID : c10fd9c3:08f9a25f:be913027:999c8e1f
Events : 18
Number Major Minor RaidDevice State
0 8 17 0 active sync /dev/sdb1
1 8 33 1 active sync /dev/sdc1
3 8 49 2 active sync /dev/sdd1

10.3.7. Деактивация программного дискового массива RAID

Информация о программном дисковом массиве присутствует в файле /proc/mdstat тогда, когда этот массив активен. Полная деактивация (остановка) массива, после которой вы сможете использовать диски из его состава для других целей, может быть осуществлена с помощью утилиты mdadm.

[root@rhel6c ~]# mdadm --stop /dev/md0
mdadm: stopped /dev/md0

Теперь на дисках могут быть созданы новые разделы.

10.3.8. Дополнительное чтение

Обратитесь к странице руководства утилиты mdadm для получения дополнительной информации. Ниже приведен пример команды, предназначенной для замены раздела с вышедшего из строя жесткого диска на раздел с нового жесткого диска в рамках дискового массива RAID.

mdadm /dev/md0 --add /dev/sdd1 --fail /dev/sdb1 --remove /dev/sdb1

10.4. Практическое задание: RAID

1. Добавьте три виртуальных диска объемом в 1 ГБ каждый в виртуальную машину.
2. Создайте программный дисковый массив RAID 5 на основе этих трех дисков. (Не обязательно создавать файловую систему на результирующем виртуальном диске.)
3. С помощью утилиты fdisk и файловой системы /proc удостоверьтесь в корректности создания дискового массива RAID 5.
4. Деактивируйте дисковый массив RAID 5 и удалите созданные разделы с дисков.
5. Создайте дисковый массив RAID 1 для зеркалирования данных на двух дисках.

10.5. Корректная процедура выполнения практического задания: RAID

1. Добавьте три виртуальных диска объемом в 1 ГБ каждый в виртуальную машину.
2. Создайте программный дисковый массив RAID 5 на основе этих трех дисков. (Не обязательно создавать файловую систему на результирующем виртуальном диске.)

3. С помощью утилиты fdisk и файловой системы /proc удостоверьтесь в корректности создания дискового массива RAID 5.
4. Деактивируйте дисковый массив RAID 5 и удалите созданные разделы с дисков.
5. Создайте дисковый массив RAID 1 для зеркалирования данных на двух дисках.

```
[root@rhel6c ~]# mdadm --create /dev/md0 --level=1 --raid-devices=2 \
/dev/sdb1 /dev/sdc1
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md0 started.
[root@rhel6c ~]# cat /proc/mdstat
Personalities : [raid6] [raid5] [raid4] [raid1]
md0 : active raid1 sdc1[1] sdb1[0]
8384862 blocks super 1.2 [2/2] [UU]
[====⇒.............] resync = 20.8% (1745152/8384862) \
finish=0.5min speed=218144K/sec
```

Глава 11. Управление логическими томами

Большинство реализаций систем управления логическими томами LVM позволяет осуществлять группировку физических носителей, изменение размеров логических томов и миграцию данных.

Группировка физических носителей — это специальное обозначение операции группировки множества блочных устройств (жестких дисков, а также устройств iSCSI и.т.д.) в рамках одного виртуального логического устройства для хранения данных. При возникновении необходимости в расширении этой логической группы устройств в нее впоследствии могут быть добавлены дополнительные блочные устройства (включая их разделы).

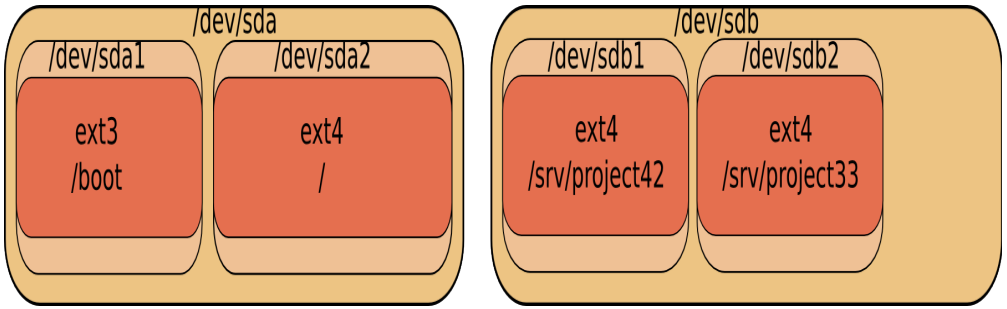
Размеры томов LVM в этой физической группе устройств не зависят от размеров отдельных устройств. Общий размер группы устройств является пределом.

Одной из наиболее ценных возможностей системы управления логическими томами LVM является возможность изменения размеров логических томов. Вы можете увеличить размер логического тома LVM в некоторых случаях даже без перезагрузки системы. Кроме того, вы можете инициировать миграцию данных с выходящего из строя жесткого диска, зеркалирование данных, а также создавать снимки разделов.

11.1. Знакомство с LVM

11.1.1. Недостатки стандартных разделов

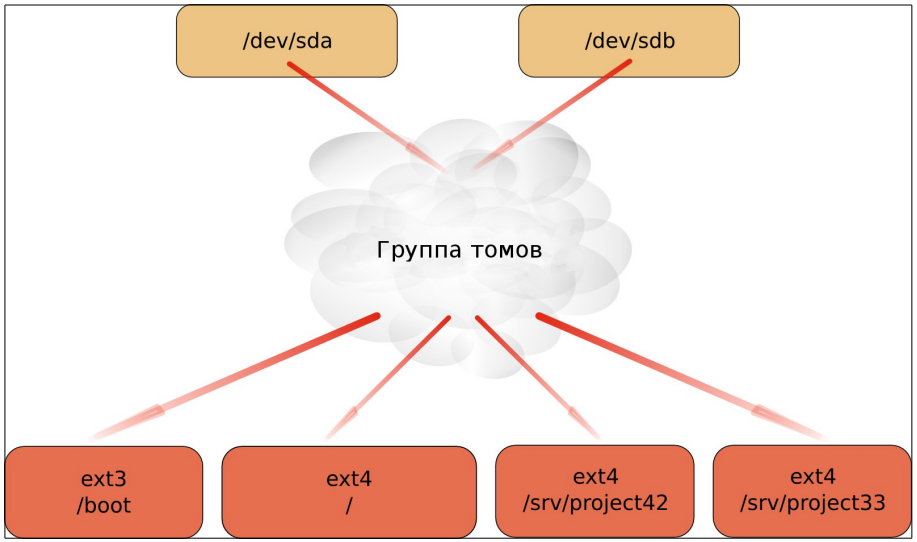
Существует несколько недостатков стандартных разделов жестких дисков, которые чаще всего выявляются в процессе эксплуатации последних. Рассмотрим систему с дисками малого и большого размера, на которых созданы разделы таким образом, как показано на иллюстрации ниже. Первый диск (представленный файлом устройства /dev/sda) разделен на два раздела, а второй диск (представленный файлом устройства /dev/sdb) — также разделен на два раздела, причем часть диска остается не размеченной.



Продолжая рассмотрение приведенного выше примера, подумайте о том, каким образом можно поступить в случае возникновения необходимости увеличения дискового пространства, доступного по пути /srv/project42. Какое действие вы можете предпринять? Любое ваше решение будет связано в размонтированием файловой системы, созданием резервной копии хранящихся в ней данных, удалением и повторным созданием разделов, восстановлением данных из резервной копии и повторным монтированием файловой системы.

11.1.2. Решение в случае использования LVM

В случае использования LVM будет создана виртуальная прослойка между смонтированными файловыми системами и аппаратными устройствами. Эта виртуальная прослойка позволит администратору увеличить размер смонтированной файловой системы в процессе ее эксплуатации. При корректном использовании LVM не возникает необходимости в размонтировании файловой системы для увеличения ее размера в ручном режиме.



11.2. Терминология LVM

11.2.1. Физический том (physical volume — pv)

Физическим томом является любое блочное устройство (файл устройства, соответствующий диску, разделу на диске, дисковому массиву RAID или даже устройству iSCSI). Все эти устройства могут быть добавлены в группу томов.

Команды, используемые для управления физическими томами, всегда начинаются с символов pv.

[root@centos65 ~]# pv
pvchange pvck pvcreate pvdisplay pvmove pvremove
pvresize pvs pvscan

11.2.2. Группа томов (volume group — vg)

Группа томов является уровнем абстракции между блочными устройствами и логическими томами.

Команды, используемые для управления группами томов, начинаются с символов vg.

[root@centos65 ~]# vg
vgcfgbackup vgconvert vgextend vgmknodes vgs
vgcfgrestore vgcreate vgimport vgreduce vgscan
vgchange vgdisplay vgimportclone vgremove vgsplit
vgck vgexport vgmerge vgrename

11.2.3. Логический том (logical volume - lv)

Логический том создается в рамках группы томов. Каждый логический том может содержать файловую систему, которая может монтироваться. Методика использования логических томов аналогична методике использования разделов дисков и предусматривает работу с тем же набором стандартных утилит (mkfs, mount, fsck, df, ...).

Команды, используемые для управления логическими томами, начинаются с символов lv.

[root@centos65 ~]# lv
lvchange lvextend lvmdiskscan lvmsar lvresize
lvconvert lvm lvmdump lvreduce lvs
lvcreate lvmchange lvmetad lvremove lvscan
lvdisplay lvmconf lvmsadc lvrename

11.3. Пример: работа с LVM

В данном примере показана методика использования дискового устройства (в нашем примере дисковое устройство представлено файлом устройства /dev/sdc, но оно также может быть представлено и файлом устройства /dev/sdb, и любым другим файлом устройства, соответствующим другому дисковому устройству или разделу на таком устройстве) при работе с технологией LVM и даны пояснения относительно создания группы томов (vg), а также создания и использования логического тома (vg/lvol0).

В первую очередь необходимо создать физические тома, которые впоследствии будут объединены в рамках группы томов с помощью утилиты pvcreate. Данная утилита модифицирует раздел дискового устройства таким образом, чтобы он мог использоваться в рамках групп томов. В примере ниже показано, как задействовать диск SCSI при работе с технологией LVM.

root@RHEL4:~# pvcreate /dev/sdc
Physical volume "/dev/sdc" successfully created

Примечание: Технология LVM будет отлично работать и в случае использования устройств вместо специализированных разделов на них, но другая операционная система, установленная на этом же компьютере (или в этой же сети хранения данных) наверняка не сможет определить, что устройство используется в рамках группы томов LVM и, скорее всего, пометит блочное устройство как устройство без данных! Вы можете избежать такого поведения сторонней операционной системы, создав раздел на всей доступной поверхности дискового устройства и использовав утилиту pvcreate по отношению к этому разделу, а не к дисковому устройству.

После этого следует воспользоваться утилитой vgcreate для создания группы томов, содержащей том с одного дискового устройства. Учтите, что в эту группу томов впоследствии могут быть добавлены тома с других устройств.

root@RHEL4:~# vgcreate vg /dev/sdc
Volume group "vg" successfully created

Последний шаг заключается в использовании утилиты lvcreate для создания логического тома.

root@RHEL4:~# lvcreate --size 500m vg
Logical volume "lvol0" created

Теперь в логическом томе, представленном файлом устройства /dev/vg/lvol0, может быть создана файловая система ext3, которая впоследствии может монтироваться и использоваться по назначению.

root@RHELv4u2:~# mke2fs -m0 -j /dev/vg/lvol0
mke2fs 1.35 (28-Feb-2004)
Filesystem label=
OS type: Linux
Block size=1024 (log=0)
Fragment size=1024 (log=0)
128016 inodes, 512000 blocks
0 blocks (0.00%) reserved for the super user
First data block=1
Maximum filesystem blocks=67633152
63 block groups
8192 blocks per group, 8192 fragments per group
2032 inodes per group
Superblock backups stored on blocks:
8193, 24577, 40961, 57345, 73729, 204801, 221185, 401409
Writing inode tables: done
Creating journal (8192 blocks): done
Writing superblocks and filesystem accounting information: done
This filesystem will be automatically checked every 37 mounts or
180 days, whichever comes first. Use tune2fs -c or -i to override.
root@RHELv4u2:~# mkdir /home/project10
root@RHELv4u2:~# mount /dev/vg/lvol0 /home/project10/
root@RHELv4u2:~# df -h | grep proj
/dev/mapper/vg-lvol0 485M 11M 474M 3% /home/project10

Логический том очень похож на раздел дискового устройства; в нем может быть создана файловая система, которая впоследствии может монтироваться и использоваться для хранения данных пользователей.

11.4. Пример: расширение логического тома

Логический том может быть расширен без отмонтирования файловой системы. Возможность расширения логического тома определяется возможностями используемой рамках этого тома файловой системы. Смонтированные тома с файловыми системами vfat и ext2 не могут быть расширены, поэтому в данном примере мы будем использовать файловую систему ext3.

Утилита fdisk выводит информацию о недавно добавленных дисках SCSI, которые будут содержать данные нашего тома LVM. Впоследствии этот том может быть расширен. В первую очередь давайте рассмотрим параметры этих дисков.

Disk /dev/sdb doesn't contain a valid partition table
Disk /dev/sdc doesn't contain a valid partition table
Disk /dev/sdb: 1181 MB, 1181115904 bytes
Disk /dev/sdc: 429 MB, 429496320 bytes

Вы уже знаете о том, как создаются разделы на дисковых устройствах, поэтому при ознакомлении со следующим фрагментом вывода можете сделать вывод о том, что разделы созданы лишь на первом диске (точнее один первичный раздел большого размера), в то время, как второй диск остался не тронутым.

Disk /dev/sdc doesn't contain a valid partition table
Disk /dev/sdb: 1181 MB, 1181115904 bytes
/dev/sdb1 1 143 1148616 83 Linux
Disk /dev/sdc: 429 MB, 429496320 bytes

Также вы знаете о том, как готовить диски к работе с технологией LVM с помощью утилиты pvcreate, а также о том, как создавать группу томов с помощью утилиты vgcreate. В данном примере в группу томов с именем vg2 добавляется как диск с разделами, так и диск без них.

[root@RHEL5 ~]# pvcreate /dev/sdb1
Physical volume "/dev/sdb1" successfully created
[root@RHEL5 ~]# pvcreate /dev/sdc
Physical volume "/dev/sdc" successfully created
[root@RHEL5 ~]# vgcreate vg2 /dev/sdb1 /dev/sdc
Volume group "vg2" successfully created

Вы можете использовать утилиту pvdisplay, чтобы убедиться в том, что как диск, так и раздел принадлежат к одной и той же группе томов.

[root@RHEL5 ~]# pvdisplay | grep -B1 vg2
PV Name /dev/sdb1
VG Name vg2
PV Name /dev/sdc
VG Name vg

Кроме того, вы должны быть знакомы как с утилитой lvcreate, предназначенной для создания логических томов, так и с утилитой mke2fs, предназначенной для создания файловой системы ext3 в рамках этих томов.

[root@RHEL5 ~]# lvcreate --size 200m vg2
Logical volume "lvol0" created
[root@RHEL5 ~]# mke2fs -m20 -j /dev/vg2/lvol0
...

Как вы можете увидеть, нам удалось создать и смонтировать

логический том размер которого, в соответствии с информацией от утилиты df, составляет практически 200 МБ.

```
[root@RHEL5 ~]# mkdir /home/resizetest
[root@RHEL5 ~]# mount /dev/vg2/lvol0 /home/resizetest/
[root@RHEL5 ~]# df -h | grep resizetest
194M 5.6M 149M 4% /home/resizetest
```

Расширение тома осуществляется достаточно просто с помощью утилиты lvextend.

```
[root@RHEL5 ~]# lvextend -L +100 /dev/vg2/lvol0
Extending logical volume lvol0 to 300.00 MB
Logical volume lvol0 successfully resized
```

Но, как вы можете увидеть, возникла небольшая проблема: кажется утилита df не может вывести информацию о полном размере расширенного тома. Это происходит из-за того, что файловая система создавалась в рамках тома, размер которого увеличился после расширения.

```
[root@RHEL5 ~]# df -h | grep resizetest
194M 5.6M 149M 4% /home/resizetest
```

Однако, в случае использования утилиты lvdisplay мы можем увидеть реальный размер тома после расширения.

```
[root@RHEL5 ~]# lvdisplay /dev/vg2/lvol0 | grep Size
LV Size 300.00 MB
```

Для завершения процесса расширения тома необходимо воспользоваться утилитой resize2fs, которая позволяет распространить файловую систему во всем пространстве, выделенном для логического тома.

```
[root@RHEL5 ~]# resize2fs /dev/vg2/lvol0
resize2fs 1.39 (29-May-2006)
Filesystem at /dev/vg2/lvol0 is mounted on /home/resizetest; on-line re\
sizing required
Performing an on-line resize of /dev/vg2/lvol0 to 307200 (1k) blocks.
The filesystem on /dev/vg2/lvol0 is now 307200 blocks long.
```

Поздравляю, вы только что успешно расширили логический том.

```
[root@RHEL5 ~]# df -h | grep resizetest
291M 6.1M 225M 3% /home/resizetest
[root@RHEL5 ~]#
```

11.5. Пример: изменение размера физического тома

Ниже представлено краткое описание методики изменения размера физического тома средствами LVM (после изменения его размера с помощью утилиты fdisk). Описание начинается с получения информации о разделе дискового устройства размером в 100 МБ, представленном файлом устройства /dev/sde1. Мы заранее использовали утилиту fdisk для создания и проверки размера этого раздела.

```
[root@RHEL5 ~]# fdisk -l 2>/dev/null | grep sde1
/dev/sde1 1 100 102384 83 Linux
[root@RHEL5 ~]#
```

Теперь мы можем использовать утилиту pvcreate для создания физического тома, а также утилиту pvs для проверки корректности его создания.

```
[root@RHEL5 ~]# pvcreate /dev/sde1
Physical volume "/dev/sde1" successfully created
[root@RHEL5 ~]# pvs | grep sde1
/dev/sde1 lvm2 -- 99.98M 99.98M
[root@RHEL5 ~]#
```

На следующем шаге следует использовать утилиту fdisk для увеличения размера раздела (на самом деле будет осуществляться удаление этого раздела и повторное создание раздела с дополнительными цилиндрами, который будет представлен файлом устройства /dev/sde1).

```
[root@RHEL5 ~]# fdisk /dev/sde
Command (m for help): p
Disk /dev/sde: 858 MB, 858993152 bytes
64 heads, 32 sectors/track, 819 cylinders
Units = cylinders of 2048 * 512 = 1048576 bytes
Device Boot Start End Blocks Id System
/dev/sde1 1 100 102384 83 Linux
Command (m for help): d
Selected partition 1
Command (m for help): n
Command action
e extended
p primary partition (1-4)
p
Partition number (1-4):
Value out of range.
Partition number (1-4): 1
First cylinder (1-819, default 1):
Using default value 1
Last cylinder or +size or +sizeM or +sizeK (1-819, default 819): 200
Command (m for help): w
The partition table has been altered!
Calling ioctl() to re-read partition table.
Syncing disks.
[root@RHEL5 ~]#
```

Если мы используем сейчас утилиты fdisk и pvs для проверки размеров раздела и физического тома, мы обнаружим различие в

значениях. В рамках LVM все еще используется старый размер раздела.

```
[root@RHEL5 ~]# fdisk -l 2>/dev/null | grep sde1
/dev/sde1 1 200 204784 83 Linux
[root@RHEL5 ~]# pvs | grep sde1
/dev/sde1 lvm2 -- 99.98M 99.98M
[root@RHEL5 ~]#
```

Утилита pvresize, примененная по отношению к физическому тому, уведомит LVM об изменении размера раздела дискового устройства. Информация о корректном размере также может быть получена с помощью утилиты pvs.

```
[root@RHEL5 ~]# pvresize /dev/sde1
Physical volume "/dev/sde1" changed
1 physical volume(s) resized / 0 physical volume(s) not resized
[root@RHEL5 ~]# pvs | grep sde1
/dev/sde1 lvm2 -- 199.98M 199.98M
[root@RHEL5 ~]#
```

11.6. Пример: зеркалирование логического тома

Начнем работу с создания трех физических томов LVM. После этого проверим корректность создания и размеры этих томов с помощью утилиты pvs. Мы создаем три физических диска по той причине, что в LVM два диска используются для хранения данных с зеркалированием, а третий том — для хранения данных журнала зеркалирования!

```
[root@RHEL5 ~]# pvcreate /dev/sdb /dev/sdc /dev/sdd
Physical volume "/dev/sdb" successfully created
Physical volume "/dev/sdc" successfully created
Physical volume "/dev/sdd" successfully created
[root@RHEL5 ~]# pvs
PV VG Fmt Attr PSize PFree
/dev/sdb lvm2 -- 409.60M 409.60M
/dev/sdc lvm2 -- 409.60M 409.60M
/dev/sdd lvm2 -- 409.60M 409.60M
```

После этого создадим группу томов и проверим корректность ее создания с помощью ранее использованной утилиты pvs. Учтите, что три физических тома теперь принадлежат к группе томов с именем vg33, причем размеры этих томов округляются в меньшую сторону (на шаг, равный размеру экстента, в нашем случае 4МБ).

```
[root@RHEL5 ~]# vgcreate vg33 /dev/sdb /dev/sdc /dev/sdd
Volume group "vg33" successfully created
[root@RHEL5 ~]# pvs
PV VG Fmt Attr PSize PFree
/dev/sda2 VolGroup00 lvm2 a- 15.88G 0
/dev/sdb vg33 lvm2 a- 408.00M 408.00M
/dev/sdc vg33 lvm2 a- 408.00M 408.00M
/dev/sdd vg33 lvm2 a- 408.00M 408.00M
[root@RHEL5 ~]#
```

Последний шаг заключается в создании логического тома с помощью утилиты lvcreate. Обратите внимание на параметр -m 1, предназначенный для выделения одного дискового устройства для хранения данных журнала зеркалирования. Также обратите внимание на изменение значений свободного пространства на каждом из трех физических томов!

```
[root@RHEL5 ~]# lvcreate --size 300m -n lvmir -m 1 vg33
Logical volume "lvmir" created
[root@RHEL5 ~]# pvs
PV VG Fmt Attr PSize PFree
/dev/sda2 VolGroup00 lvm2 a- 15.88G 0
/dev/sdb vg33 lvm2 a- 408.00M 108.00M
/dev/sdc vg33 lvm2 a- 408.00M 108.00M
/dev/sdd vg33 lvm2 a- 408.00M 404.00M
```

Вы можете получить информацию о состоянии процесса зеркалирования данных, воспользовавшись утилитой lvs. В данном примере создана 100-процентная копия данных.

```
[root@RHEL5 ~]# lvs vg33/lvmir
LV VG Attr LSize Origin Snap% Move Log Copy%
lvmir vg33 mwi-ao 300.00M lvmir_mlog 100.00
```

11.7. Пример: снимок логического тома

Снимком называется виртуальная копия всех данных, хранящихся на разделе дискового устройства в данный момент времени. Снимок логического раздела будет содержать копию всех измененных файлов с используемого для создания снимка логического тома.

В примере ниже создается снимок логического тома под названием bigLV.

```
[root@RHEL5 ~]# lvcreate -L100M -s -n snapLV vg42/bigLV
Logical volume "snapLV" created
[root@RHEL5 ~]#
```

Изучив вывод утилиты lvs, вы можете убедиться в том, что снимок snapLV действительно является снимком логического тома bigLV. Спустя непродолжительное время после создания снимка, происходит несколько модификаций данных на логическом томе bigLV (0.02 процента).

```
[root@RHEL5 ~]# lvs
LV VG Attr LSize Origin Snap% Move Log Copy%
bigLV vg42 owi-a- 200.00M
snapLV vg42 swi-a- 100.00M bigLV 0.02
[root@RHEL5 ~]#
```

Но в процессе последующего использования тома bigLV объем

модификаций увеличивается. Это означает, что в рамках логического тома снимка должен храниться больший объем оригинальных данных (10.22 процента).

```
[root@RHEL5 ~]# lvs | grep vg42
bigLV vg42 owi-ao 200.00M
snapLV vg42 swi-a- 100.00M bigLV 10.22
[root@RHEL5 ~]#
```

Теперь вы можете использовать обычные инструменты для резервного копирования данных (dump, tar, cpio, ...) с целью создания резервной копии логического тома снимка. Эта резервная копия будет содержать все данные, хранящиеся в рамках логического тома bigLV в момент создания снимка. После создания резервной копии вы можете удалить логический том снимка.

```
[root@RHEL5 ~]# lvremove vg42/snapLV
Do you really want to remove active logical volume "snapLV"? [y/n]: y
Logical volume "snapLV" successfully removed
[root@RHEL5 ~]#
```

11.8. Проверка существующих физических томов

11.8.1. Утилита lvmdiskscan

Для получения списка блочных устройств, которые могут использоваться для работы с технологией LVM следует использовать утилиту lvmdiskscan. В примере ниже для вывода информации исключительно об устройствах SCSI используется утилита grep.

```
[root@RHEL5 ~]# lvmdiskscan | grep sd
/dev/sda1 [ 101.94 MB]
/dev/sda2 [ 15.90 GB] LVM physical volume
/dev/sdb [ 409.60 MB]
/dev/sdc [ 409.60 MB]
/dev/sdd [ 409.60 MB] LVM physical volume
/dev/sde1 [ 95.98 MB]
/dev/sde5 [ 191.98 MB]
/dev/sdf [ 819.20 MB] LVM physical volume
/dev/sdg1 [ 818.98 MB]
[root@RHEL5 ~]#
```

11.8.2. Утилита pvs

Простейший способ получения информации о том, используется ли технология LVM для работы с определенными устройствами, заключается в использовании утилиты pvs. При рассмотрении фрагмента вывода данной утилиты, приведенного в примере ниже, можно сделать вывод о том, что технология LVM используется исключительно для работы с разделом дискового устройства, представленным файлом устройства /dev/sda2. В выводе утилиты также содержится информация о том, что раздел дискового устройства, представленный файлом устройства /dev/sda2, состоит в группе томов с именем Volgroup00 и имеет размер, равный практически 16 Гб. Кроме того, в выводе утилиты содержится информация о дисковых устройствах, представленных файлами устройств /dev/sdc и /dev/sdd, которые состоят в группе томов с именем vg33. Технология LVM может использоваться и для работы с дисковым устройством, представленным файлом устройства /dev/sdb, но это устройство на данный момент не относится к каким-либо группам томов.

```
[root@RHEL5 ~]# pvs
PV VG Fmt Attr PSize PFree
/dev/sda2 VolGroup00 lvm2 a- 15.88G 0
/dev/sdb lvm2 -- 409.60M 409.60M
/dev/sdc vg33 lvm2 a- 408.00M 408.00M
/dev/sdd vg33 lvm2 a- 408.00M 408.00M
[root@RHEL5 ~]#
```

11.8.3. Утилита pvscan

Утилита pvscan позволяет осуществить сканирование всех дисковых устройств в поисках существующих физических томов. Данная утилита выводит информацию, аналогичную той, которую выводит утилита pvs, а также строку с информацией об общем размере обнаруженных физических томов.

```
[root@RHEL5 ~]# pvscan
PV /dev/sdc VG vg33 lvm2 [408.00 MB / 408.00 MB free]
PV /dev/sdd VG vg33 lvm2 [408.00 MB / 408.00 MB free]
PV /dev/sda2 VG VolGroup00 lvm2 [15.88 GB / 0 free]
PV /dev/sdb lvm2 [409.60 MB]
Total: 4 [17.07 GB] / in use: 3 [16.67 GB] / in no VG: 1 [409.60 MB]
[root@RHEL5 ~]#
```

11.8.4. Утилита pvdisplay

Используйте утилиту pvdisplay для получения дополнительной информации о физическом томе. Также вы можете использовать утилиту pvdisplay без аргументов для вывода информации обо всех физических томах (LVM).

```
root@RHEL5 ~]# pvdisplay /dev/sda2
--- Physical volume ---
PV Name /dev/sda2
VG Name VolGroup00
PV Size 15.90 GB / not usable 20.79 MB
Allocatable yes (but full)
PE Size (KByte) 32768
Total PE 508
Free PE 0
Allocated PE 508
```

PV UUID TobYfp-Ggg0-Rf8r-xtLd-5XgN-RSPc-8vkTHD
[root@RHEL5 ~]#

11.9. Проверка существующих групп томов

11.9.1. Утилита vgs

По аналогии с утилитой pvs, утилита vgs может использоваться для вывода краткой информации обо всех группах томов. В выводе данной утилиты, приведенном в примере ниже, содержится информация только об одной группе томов с именем VolGroup00, размер которой равен практически 16 Гб.

```
[root@RHEL5 ~]# vgs
VG #PV #LV #SN Attr VSize VFree
VolGroup00 1 2 0 wz--n- 15.88G 0
[root@RHEL5 ~]#
```

11.9.2. Утилита vgscan

Утилита vgscan позволяет осуществить сканирование всех дисковых устройств в поисках существующих групп томов. Также данная утилита обновляет содержимое файла /etc/lvm/.cache. В данном файле содержится список всех устройств, для работы с которыми на данный момент используется технология LVM.

```
[root@RHEL5 ~]# vgscan
Reading all physical volumes. This may take a while...
Found volume group "VolGroup00" using metadata type lvm2
[root@RHEL5 ~]#
```

В случае использования технологии LVM утилита vgscan будет запускаться автоматически в процессе загрузки системы, поэтому в том случае, если вы будете осуществлять горячее подключение дисковых устройств, вам придется самостоятельно запускать утилиту vgscan для добавления в файл /etc/lvm/.cache информации о новых устройствах.

11.9.3. Утилита vgdisplay

Утилита vgdisplay предоставляет более подробную информацию о группе томов (или обо всех группах томов, если вы не будете передавать ей каких-либо аргументов).

```
[root@RHEL5 ~]# vgdisplay VolGroup00
--- Volume group ---
VG Name VolGroup00
System ID
Format lvm2
Metadata Areas 1
Metadata Sequence No 3
VG Access read/write
VG Status resizable
MAX LV 0
Cur LV 2
Open LV 2
Max PV 0
Cur PV 1
Act PV 1
VG Size 15.88 GB
PE Size 32.00 MB
Total PE 508
Alloc PE / Size 508 / 15.88 GB
Free PE / Size 0 / 0
VG UUID qsXvJb-71qV-9l7U-ishX-FobM-qptE-VXmKIg
[root@RHEL5 ~]#
```

11.10. Проверка существующих логических томов

11.10.1. Утилита lvs

Используйте утилиту lvs для получения краткой информации обо всех существующих логических томах. В выводе данной утилиты, приведенном в примере ниже, вы можете обнаружить информацию о двух логических томах с именами LogVol00 и LogVol01.

```
[root@RHEL5 ~]# lvs
LV VG Attr LSize Origin Snap% Move Log Copy%
LogVol00 VolGroup00 -wi-ao 14.88G
LogVol01 VolGroup00 -wi-ao 1.00G
[root@RHEL5 ~]#
```

11.10.2. Утилита lvscan

Утилита lvscan позволяет осуществить сканирование всех дисковых устройств в поисках существующих логических томов.

```
[root@RHEL5 ~]# lvscan
ACTIVE '/dev/VolGroup00/LogVol00' [14.88 GB] inherit
ACTIVE '/dev/VolGroup00/LogVol01' [1.00 GB] inherit
[root@RHEL5 ~]#
```

11.10.3. Утилита lvdisplay

Более подробная информация о логических томах может быть получена с помощью утилиты lvdisplay(1).

```
[root@RHEL5 ~]# lvdisplay VolGroup00/LogVol01
--- Logical volume ---
LV Name /dev/VolGroup00/LogVol01
VG Name VolGroup00
LV UUID RnTGK6-xWsi-t530-ksJx-7cax-co5c-A1KlDp
LV Write Access read/write
LV Status available
# open 1
LV Size 1.00 GB
```

Current LE 32
Segments 1
Allocation inherit
Read ahead sectors 0
Block device 253:1
[root@RHEL5 ~]#

11.11. Управление логическими томами

11.11.1. Утилита pvcreate

Используйте утилиту pvcreate для добавления устройств, работа с которыми должна осуществляться посредством технологии LVM. В данном примере показана методика перевода дискового устройства (или устройства из аппаратного массива RAID) в разряд устройств LVM.

```
[root@RHEL5 ~]# pvcreate /dev/sdb
Physical volume "/dev/sdb" successfully created
[root@RHEL5 ~]#
```

В следующем примере показана методика перевода раздела дискового устройства в разряд устройств, работа с которыми должна осуществляться посредством технологии LVM.

```
[root@RHEL5 ~]# pvcreate /dev/sdc1
Physical volume "/dev/sdc1" successfully created
[root@RHEL5 ~]#
```

Также вы можете передавать утилите pvcreate пути к множеству файлов устройств, представляющих дисковые устройства и их разделы. В данном примере для работы посредством технологии LVM выделяются три диска.

```
[root@RHEL5 ~]# pvcreate /dev/sde /dev/sdf /dev/sdg
Physical volume "/dev/sde" successfully created
Physical volume "/dev/sdf" successfully created
Physical volume "/dev/sdg" successfully created
[root@RHEL5 ~]#
```

11.11.2. Утилита pvremove

Используйте утилиту pvremove для удаления физических томов из списка физических томов LVM. Соответствующие устройства не должны использоваться.

```
[root@RHEL5 ~]# pvremove /dev/sde /dev/sdf /dev/sdg
Labels on physical volume "/dev/sde" successfully wiped
Labels on physical volume "/dev/sdf" successfully wiped
Labels on physical volume "/dev/sdg" successfully wiped
[root@RHEL5 ~]#
```

11.11.3. Утилита pvresize

После того, как вы воспользовались утилитой fdisk для изменения размера раздела на диске, вы должны использовать утилиту pvresize для определения нового размера физического тома, представляющего рассматриваемый раздел диска, на уровне LVM.

```
[root@RHEL5 ~]# pvresize /dev/sde1
Physical volume "/dev/sde1" changed
1 physical volume(s) resized / 0 physical volume(s) not resized
```

11.11.4. Утилита pvchange

С помощью утилиты pvchange вы можете предотвратить резервирование физического тома в рамках новой группы томов или логического тома. Это обстоятельство может оказаться полезным в том случае, если вы планируете впоследствии удалить этот физический том.

```
[root@RHEL5 ~]# pvchange -xn /dev/sdd
Physical volume "/dev/sdd" changed
1 physical volume changed / 0 physical volumes not changed
[root@RHEL5 ~]#
```

В следующем примере содержится описание методики повторной активации физического тома для его последующего резервирования, которая может быть осуществлена в случае необходимости отмены вашего предыдущего решения.

```
[root@RHEL5 ~]# pvchange -xy /dev/sdd
Physical volume "/dev/sdd" changed
1 physical volume changed / 0 physical volumes not changed
[root@RHEL5 ~]#
```

11.11.5. Утилита pvmove

С помощью утилиты pvmove вы можете перемещать логические тома в рамках группы томов в сторонние физические тома. Эта операция должна выполняться перед удалением физического тома.

```
[root@RHEL5 ~]# pvs | grep vg1
/dev/sdf vg1 lvm2 a- 816.00M 0
/dev/sdg vg1 lvm2 a- 816.00M 816.00M
[root@RHEL5 ~]# pvmove /dev/sdf
/dev/sdf: Moved: 70.1%
/dev/sdf: Moved: 100.0%
[root@RHEL5 ~]# pvs | grep vg1
/dev/sdf vg1 lvm2 a- 816.00M 816.00M
/dev/sdg vg1 lvm2 a- 816.00M 0
```

11.12. Управление группами томов

11.12.1. Утилита vgcreate

Используйте утилиту vgcreate для создания группы томов. Вы можете сразу же передать информацию о путях к файлам устройств физических томов, которые должны быть добавлены в создаваемую группу томов.

```
[root@RHEL5 ~]# vgcreate vg42 /dev/sde /dev/sdf
Volume group "vg42" successfully created
[root@RHEL5 ~]#
```

11.12.2. Утилита vgextend

Используйте утилиту vgextend для расширения существующей группы томов путем добавления в нее нового физического тома.

```
[root@RHEL5 ~]# vgextend vg42 /dev/sdg
Volume group "vg42" successfully extended
[root@RHEL5 ~]#
```

11.12.3. Утилита vgremove

Используйте утилиту vgremove для удаления групп томов LVM. Удаляемые группы томов не должны использоваться.

```
[root@RHEL5 ~]# vgremove vg42
Volume group "vg42" successfully removed
[root@RHEL5 ~]#
```

11.12.4. Утилита vgreduce

Используйте утилиту vgreduce для удаления физического тома из группы томов.

В следующем примере осуществляется добавление физического тома, представленного файлом устройства /dev/sdg, в группу томов с именем vg1 средствами утилиты vgextend. После этого данный логический том удалятся из группы томов с помощью утилиты vgreduce.

```
[root@RHEL5 ~]# pvs | grep sdg
/dev/sdg lvm2 -- 819.20M 819.20M
[root@RHEL5 ~]# vgextend vg1 /dev/sdg
Volume group "vg1" successfully extended
[root@RHEL5 ~]# pvs | grep sdg
/dev/sdg vg1 lvm2 a- 816.00M 816.00M
[root@RHEL5 ~]# vgreduce vg1 /dev/sdg
Removed "/dev/sdg" from volume group "vg1"
[root@RHEL5 ~]# pvs | grep sdg
/dev/sdg lvm2 -- 819.20M 819.20M
```

11.12.5. Утилита vgchange

Используйте утилиту vgchange для изменения значений параметров группы томов.

В данном примере показана методика предотвращения добавления или удаления физических томов из группы томов с именем vg1.

```
[root@RHEL5 ~]# vgchange -xn vg1
Volume group "vg1" successfully changed
[root@RHEL5 ~]# vgextend vg1 /dev/sdg
Volume group vg1 is not resizable.
```

Также вы можете использовать утилиту vgchange для изменения значений большинства других параметров группы томов. В данном примере осуществляется изменение максимального количества логических томов, а также максимального количества физических томов, которые могут функционировать в рамках группы томов с именем vg1.

```
[root@RHEL5 ~]# vgdisplay vg1 | grep -i max
MAX LV 0
Max PV 0
[root@RHEL5 ~]# vgchange -l16 vg1
Volume group "vg1" successfully changed
[root@RHEL5 ~]# vgchange -p8 vg1
Volume group "vg1" successfully changed
[root@RHEL5 ~]# vgdisplay vg1 | grep -i max
MAX LV 16
Max PV 8
```

11.12.6. Утилита vgmerge

Объединение двух групп томов в рамках одной группы может осуществляться с помощью утилиты vgmerge. В следующем примере происходит объединение группы томов с именем vg2 с группой томов с именем vg1 с сохранением всех значений параметров группы томов с именем vg1.

```
[root@RHEL5 ~]# vgmerge vg1 vg2
Volume group "vg2" successfully merged into "vg1"
[root@RHEL5 ~]#
```

11.13. Управление логическими томами

11.13.1. Утилита lvcreate

Используйте утилиту lvcreate для создания логических томов в рамках группы томов. В данном примере в рамках группы томов с именем vg42 создается логический том размером в 8 Гб.

```
[root@RHEL5 ~]# lvcreate -L5G vg42
Logical volume "lvol0" created
[root@RHEL5 ~]#
```

Как вы видите, LVM автоматически присваивает логическому разделу имя lvol0. В следующем примере в рамках группы томов с именем vg42 создается логический том с именем MyLV размером в 200 Мб.

```
[root@RHEL5 ~]# lvcreate -L200M -nMyLV vg42
Logical volume "MyLV" created
[root@RHEL5 ~]#
```

В следующем примере выполняется та же операция, но используется отличный синтаксис команды.

```
[root@RHEL5 ~]# lvcreate --size 200M -n MyLV vg42
Logical volume "MyLV" created
```

[root@RHEL5 ~]#
А в этом примере создается логический том, который будет занимать ровно 10 процентов дискового пространства группы томов.
[root@RHEL5 ~]# lvcreate -l 10%VG -n MyLV2 vg42
Logical volume "MyLV2" created
[root@RHEL5 ~]#
В этом же примере создается логический том, который будет занимать 30 процентов оставшегося в рамках группы томов свободного дискового пространства.
[root@RHEL5 ~]# lvcreate -l 30%FREE -n MyLV3 vg42
Logical volume "MyLV3" created
[root@RHEL5 ~]#

11.13.2. Утилита lvremove

Используйте утилиту lvremove для удаления логических томов из группы томов. Для удаления логического тома потребуется имя группы томов.
[root@RHEL5 ~]# lvremove vg42/MyLV
Do you really want to remove active logical volume "MyLV"? [y/n]: y
Logical volume "MyLV" successfully removed
[root@RHEL5 ~]#
В случае удаления множества логических томов будут выводиться запросы подтверждений перед удалением каждого из указанных томов.
[root@RHEL5 ~]# lvremove vg42/MyLV vg42/MyLV2 vg42/MyLV3
Do you really want to remove active logical volume "MyLV"? [y/n]: y
Logical volume "MyLV" successfully removed
Do you really want to remove active logical volume "MyLV2"? [y/n]: y
Logical volume "MyLV2" successfully removed
Do you really want to remove active logical volume "MyLV3"? [y/n]: y
Logical volume "MyLV3" successfully removed
[root@RHEL5 ~]#

11.13.3. Утилита lvextend

Расширение логического тома может осуществляться достаточно просто средствами утилиты lvextend. В следующем примере логический том размером в 200 Мб расширяется на 100 Мб.
[root@RHEL5 ~]# lvdisplay /dev/vg2/lvol0 | grep Size
LV Size 200.00 MB
[root@RHEL5 ~]# lvextend -L +100 /dev/vg2/lvol0
Extending logical volume lvol0 to 300.00 MB
Logical volume lvol0 successfully resized
[root@RHEL5 ~]# lvdisplay /dev/vg2/lvol0 | grep Size
LV Size 300.00 MB
В следующем примере создается логический том размером в 100 Мб, который впоследствии расширяется до 500 Мб.
[root@RHEL5 ~]# lvcreate --size 100M -n extLV vg42
Logical volume "extLV" created
[root@RHEL5 ~]# lvextend -L 500M vg42/extLV
Extending logical volume extLV to 500.00 MB
Logical volume extLV successfully resized
[root@RHEL5 ~]#
А в данном примере размер логического тома удваивается.
[root@RHEL5 ~]# lvextend -l+100%LV vg42/extLV
Extending logical volume extLV to 1000.00 MB
Logical volume extLV successfully resized
[root@RHEL5 ~]#

11.13.4. Утилита lvrename

Переименование логического тома осуществляется с помощью утилиты lvrename. В данном примере осуществляется смена имени логического тома из группы томов vg42 с extLV на bigLV.
[root@RHEL5 ~]# lvrename vg42/extLV vg42/bigLV
Renamed "extLV" to "bigLV" in volume group "vg42"
[root@RHEL5 ~]#

11.14. Практическое задание: LVM

1. Создайте группу томов, которая будет содержать все пространство дискового устройства и раздел с другого дискового устройства.
2. Создайте два логических тома (малого и большего размера) в рамках данной группы томов. Создайте в них файловую систему ext3, смонтируйте их и скопируйте какие-либо файлы на них.
3. Попытайтесь получить информацию о заполнении созданных томов, воспользовавшись утилитами fdisk, mount, pvs, vgs, lvs, pvdisplay, vgdisplay, lvdispaly и df. Получили ли вы какую-либо информацию о томах LVM от утилиты fdisk?
4. Расширьте логический том малого размера на 50 процентов и проверьте корректность выполнения операции!
5. Рассмотрите другие утилиты, имена которых начинаются с символов vg*, pv* и lv*.
6. Создайте зеркальный логический том (mirrored) и логический том с чередованием (striped).
7. Преобразуйте линейный логический том (linear) в зеркальный (mirrored).
8. Преобразуйте зеркальный логический том (mirrored) в линейный (linear).
9. Создайте снимок логического тома, а также резервную копию этого снимка. После этого удалите некоторые файлы с логического тома и восстановите том из резервной копии

его снимка.

10. Переместите вашу группу томов на другое дисковое устройство (оставьте логические тома смонтированными).
11. При наличии времени разделите группу томов с помощью утилиты vgsplit, после чего снова объедините результирующие группы с помощью утилиты vgmerge.

11.15. Корректная процедура выполнения практического задания: LVM

1. Создайте группу томов, которая будет содержать все пространство дискового устройства и раздел с другого дискового устройства.

Шаг 1: выбор дисковых устройств
root@rhel65:~#
fdisk -l | grep Disk
Disk /dev/sda: 8589 MB, 8589934592 bytes
Disk identifier: 0x00055ca0
Disk /dev/sdb: 1073 MB, 1073741824 bytes
Disk identifier: 0x00000000
Disk /dev/sdc: 1073 MB, 1073741824 bytes
Disk identifier: 0x00000000
...
В данном случае я выбрал дисковые устройства, представленные файлами устройств /dev/sdb и /dev/sdc.
Шаг 2: создание разделов на устройстве, представленном файлом устройства /dev/sdc
root@rhel65:~#
fdisk /dev/sdc
Device contains neither a valid DOS partition table, nor Sun, SGI or OSF disk\
label
Building a new DOS disklabel with disk identifier 0x94c0e5d5.
Changes will remain in memory only, until you decide to write them.
After that, of course, the previous content won't be recoverable.
Warning: invalid flag 0x0000 of partition table 4 will be corrected by w(rite)
Warnin DOS-compatible mode is deprecated. It's strongly
g recommended to
switch off the mode (command 'c') and change display units to
sectors (command 'u').
Command (m for help): n
Command action
e extended
p primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-130, default 1):
Using default value 1
Last cylinder, +cylinders or +size{K,M,G} (1-130, default 130):
Using default value 130
Command (m for help): w
The partition table has been altered!
Calling ioctl() to re-read partition table.
Syncing disks.
Шаг 3: Работа с утилитами pvcreate и vgcreate
root@rhel65:~# pvcreate /dev/sdb /dev/sdc1
Physical volume "/dev/sdb" successfully created
Physical volume "/dev/sdc1" successfully created
root@rhel65:~# vgcreate VG42 /dev/sdb /dev/sdc1
Volume group "VG42" successfully created

1. Создайте два логических тома (малого и большего размера) в рамках данной группы томов. Создайте в них файловую систему ext3, смонтируйте их и скопируйте какие-либо файлы на них.

root@rhel65:~# lvcreate --size 200m --name LVsmall VG42
Logical volume "LVsmall" created
root@rhel65:~# lvcreate --size 600m --name LVbig VG42
Logical volume "LVbig" created
root@rhel65:~# ls -l /dev/mapper/VG42-LVsmall
lrwxrwxrwx. 1 root root 7 Апр 20 20:41 /dev/mapper/VG42-LVsmall
→ ../dm-2
root@rhel65:~# ls -l /dev/VG42/LVsmall
lrwxrwxrwx. 1 root root 7 Апр 20 20:41 /dev/VG42/LVsmall → ../dm-2
root@rhel65:~# ls -l /dev/dm-2
brw-rw----. 1 root disk 253, 2 Апр 20 20:41 /dev/dm-2
root@rhel65:~# mkfs.ext3 /dev/mapper/VG42-LVsmall
mke2fs 1.41.12 (17-May-2010)
Filesystem label=
OS type: Linux
Block size=1024 (log=0)
Fragment size=1024 (log=0)
Stride=0 blocks, Stripe width=0 blocks
51200 inodes, 204800 blocks
10240 blocks (5.00%) reserved for the super user
First data block=1
Maximum filesystem blocks=67371008
25 block groups
8192 blocks per group, 8192 fragments per group

2048 inodes per group
Superblock backups stored on blocks:
8193, 24577, 40961, 57345, 73729
Writing inode tables: done
Creating journal (4096 blocks): done
Writing superblocks and filesystem accounting information: done
This filesystem will be automatically checked every 39 mounts or
180 days, whichever comes first. Use tune2fs -c or -i to override.
root@rhel65:~# mkfs.ext3 /dev/VG42/LVbig
mke2fs 1.41.12 (17-May-2010)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
38400 inodes, 153600 blocks
7680 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=159383552
5 block groups
32768 blocks per group, 32768 fragments per group
7680 inodes per group
Superblock backups stored on blocks:
32768, 98304
Writing inode tables: done
Creating journal (4096 blocks): done
Writing superblocks and filesystem accounting information: done
This filesystem will be automatically checked every 25 mounts or
180 days, whichever comes first. Use tune2fs -c or -i to override.
Монтирование файловых систем и копирование файлов:
root@rhel65:~# mkdir /srv/LVsmall
root@rhel65:~# mkdir /srv/LVbig
root@rhel65:~# mount /dev/mapper/VG42-LVsmall /srv/LVsmall
root@rhel65:~# mount /dev/VG42/LVbig /srv/LVbig
root@rhel65:~# cp -r /etc /srv/LVsmall/
root@rhel65:~# cp -r /var/log /srv/LVbig/

1. Попытайтесь получить информацию о заполнении созданных томов, воспользовавшись утилитами fdisk, mount, pvs, vgs, lvs, pvdisplay, vgdisplay, lvdispaly и df. Получили ли вы какую-либо информацию о томах LVM от утилиты fdisk?

Используйте все упомянутые утилиты (ниже приведены примеры использования только двух из них), и ответьте «нет» на поставленный вопрос.

root@rhel65:~# df -h
Filesystem Size Used Avail Use% Mounted on
/dev/mapper/VolGroup-lv_root
6.7G 1.4G 5.0G 21% /
tmpfs 246M 0 246M 0% /dev/shm
/dev/sda1 485M 77M 383M 17% /boot
/dev/mapper/VG42-LVsmall
194M 30M 154M 17% /srv/LVsmall
/dev/mapper/VG42-LVbig
591M 20M 541M 4% /srv/LVbig
root@rhel65:~# mount | grep VG42
/dev/mapper/VG42-LVsmall on /srv/LVsmall type ext3 (rw)
/dev/mapper/VG42-LVbig on /srv/LVbig type ext3 (rw)

1. Расширьте логический том малого размера на 50 процентов и проверьте корректность выполнения операции!

root@rhel65:~# lvextend VG42/LVsmall -l+50%LV
Extending logical volume LVsmall to 300.00 MiB
Logical volume LVsmall successfully resized
root@rhel65:~#
resize2fs /dev/mapper/VG42-LVsmall
resize2fs 1.41.12 (17-May-2010)
Filesystem at /dev/mapper/VG42-LVsmall is mounted on /srv/LVsmall; on-line res\
izing required
old desc_blocks = 1, new_desc_blocks = 2
Performing an on-line resize of /dev/mapper/VG42-LVsmall to 307200 (1k) blocks.
The filesystem on /dev/mapper/VG42-LVsmall is now 307200 blocks long.
root@rhel65:~# df -h | grep small
/dev/mapper/VG42-LVsmall
291M 31M 246M 12% /srv/LVsmall
root@rhel65:~#

1. Рассмотрите другие утилиты, имена которых начинаются с символов vg*, pv* и lv*.
2. Создайте зеркальный логический том (mirrored) и логический том с чередованием (striped).
3. Преобразуйте линейный логический том (linear) в зеркальный (mirrored).
4. Преобразуйте зеркальный логический том (mirrored) в линейный (linear).
5. Создайте снимок логического тома, а также резервную копию этого снимка. После этого удалите некоторые файлы с логического тома и восстановите том из резервной копии его снимка.

6. Переместите вашу группу томов на другое дисковое устройство (оставьте логические тома смонтированными).
7. При наличии времени разделите группу томов с помощью утилиты vgsplit, после чего снова объедините результирующие группы с помощью утилиты vgmerge.

По прочтении данной главы вы научитесь настраивать сервер, являющийся целевым сервером iSCSI (iSCSI target), и клиент, являющийся инициатором iSCSI (iSCSI initiator).

Глава 12. Устройства iSCSI

12.1. Терминология iSCSI

iSCSI является протоколом, который позволяет передавать команды SCSI посредством протокола IP. Это означает, что вы можете использовать локальные файлы устройств SCSI (такие, как /dev/sdb) в условиях отсутствия представленных ими дисковых накопителей в локальном компьютере.

Компьютер, в котором установлены физические устройства для хранения данных, называется целевым сервером iSCSI (iSCSI Target). Каждое отдельное адресуемое устройство iSCSI на целевом сервере получает индивидуальный номер LUN.

Компьютер-клиент iSCSI, который взаимодействует с целевым сервером называется инициатором (iSCSI Initiator). Инициатор отправляет команды SCSI по протоколу IP вместо непосредственного взаимодействия с аппаратным устройством. Инициатор должен соединяться с целевым сервером посредством сети.

12.2. Целевой сервер iSCSI в дистрибутиве RHEL/CentOS

В данном разделе описана методика настройки целевого сервера iSCSI в дистрибутивах RHEL6, RHEL7 и CentOS.

Настройку следует начинать с установки пакета с программными компонентами, выполняющими функции целевого сервера iSCSI.

yum install scsi-target-utils

В данном примере в файле /etc/tgt/targets.conf указаны пути к файлам устройств, представляющим три локальных диска, которым должны быть присвоены номера LUN.

```
<target iqn.2008-09.com.example:server.target2>
direct-store /dev/sdb
direct-store /dev/sdc
direct-store /dev/sdd
incominguser paul hunter2
</target>
```

Перезапуск службы:

```
[root@centos65 ~]# service tgtd start
Starting SCSI target daemon: [ OK ]
```

Стандартным локальным портом целевого сервера iSCSI должен являться порт номер 3260, причем при необходимости вы можете удостовериться в этом с помощью утилиты netstat.

```
[root@server1 tgt]# netstat -ntpl | grep tgt
tcp 0 0 0.0.0.0:3260 0.0.0.0:* LISTEN 1670/tgtd
tcp 0 0 :::3260 :::* LISTEN 1670/tgtd
```

Теперь при использовании команды tgt-admin -s вы можете получить подробную информацию об устройствах с тремя номерам LUN (а также о контроллере с номером LUN 0).

```
[root@server1 tgt]# tgt-admin -s
Target 1: iqn.2014-04.be.linux-training:server1.target1
System information:
Driver: iscsi
State: ready
I_T nexus information:
LUN information:
LUN: 0
Type: controller
SCSI ID: IET 00010000
SCSI SN: beaf10
Size: 0 MB, Block size: 1
Online: Yes
Removable media: No
Prevent removal: No
Readonly: No
Backing store type: null
Backing store path: None
Backing store flags:
LUN: 1
Type: disk
SCSI ID: IET 00010001
SCSI SN: VB9f23197b-af6cfb60
Size: 1074 MB, Block size: 512
Online: Yes
Removable media: No
Prevent removal: No
Readonly: No
Backing store type: rdwr
Backing store path: /dev/sdb
Backing store flags:
LUN: 2
Type: disk
```

SCSI ID: IET 00010002
SCSI SN: VB8f554351-a1410828
Size: 1074 MB, Block size: 512
Online: Yes
Removable media: No
Prevent removal: No
Readonly: No
Backing store type: rdwr
Backing store path: /dev/sdc
Backing store flags:
LUN: 3
Type: disk
SCSI ID: IET 00010003
SCSI SN: VB1035d2f0-7ae90b49
Size: 1074 MB, Block size: 512
Online: Yes
Removable media: No
Prevent removal: No
Readonly: No
Backing store type: rdwr
Backing store path: /dev/sdd
Backing store flags:
Account information:
ACL information:
ALL

12.3. Инициатор iSCSI в дистрибутиве RHEL/CentOS

В данном разделе описана методика настройки инициатора iSCSI в дистрибутивах RHEL6, RHEL7 и CentOS.

Настройку следует начинать с установки пакета с программными компонентами, выполняющими функции инициатора iSCSI.

[root@server2 ~]# yum install iscsi-initiator-utils

После этого необходимо запросить у целевого сервера iSCSI список идентификаторов всех целевых серверов.

[root@server2 ~]# iscsiadm -m discovery -t sendtargets -p 192.168.1.95:3260
Starting iscsid: [OK]
192.168.1.95:3260,1 iqn.2014-04.be.linux-training:centos65.target1

Мы получили информацию о целевом сервере с идентификатором iqn.2014-04.be.linux-training:centos65.target1.

Используем этот идентификатор iqn для указания имени пользователя и пароля (paul и hunter2 соответственно), которые были ранее установлены нами на целевом сервере.

[root@server2 iscsi]# iscsiadm -m node --targetname iqn.2014-04.be.linux-tra\
ining:centos65.target1 --portal "192.168.1.95:3260" --op=update --name node.\
session.auth.username --value=paul
[root@server2 iscsi]# iscsiadm -m node --targetname iqn.2014-04.be.linux-tra\
ining:centos65.target1 --portal "192.168.1.95:3260" --op=update --name node.\
session.auth.password --value=hunter2
[root@server2 iscsi]# iscsiadm -m node --targetname iqn.2014-04.be.linux-tra\
ining:centos65.target1 --portal "192.168.1.95:3260" --op=update --name node.\
session.auth.authmethod --value=CHAP

В дистрибутивах RHEL и CentOS эта информация хранится в файлах из директории /var/lib/iscsi/nodes/.

[root@server2 iscsi]# grep auth /var/lib/iscsi/nodes/iqn.2014-04.be.linux-tr\
aining\:centos65.target1/192.168.1.95\,3260\,1/default
node.session.auth.authmethod = CHAP
node.session.auth.username = paul
node.session.auth.password = hunter2
node.conn[0].timeo.auth_timeout = 45
[root@server2 iscsi]#

После перезапуска службы iscsi в нашей системе появятся три новых устройства.

[root@server2 iscsi]# fdisk -l | grep Disk
Disk /dev/sda: 42.9 GB, 42949672960 bytes
Disk identifier: 0x0004f229
Disk /dev/sdb: 1073 MB, 1073741824 bytes
Disk identifier: 0x00000000
Disk /dev/sdc: 1073 MB, 1073741824 bytes
Disk identifier: 0x00000000
Disk /dev/sdd: 1073 MB, 1073741824 bytes
Disk identifier: 0x00000000
Disk /dev/sde: 2147 MB, 2147483648 bytes
Disk identifier: 0x00000000
Disk /dev/sdf: 2147 MB, 2147483648 bytes
Disk identifier: 0x00000000
Disk /dev/sdg: 2147 MB, 2147483648 bytes
Disk identifier: 0x00000000
Disk /dev/mapper/VolGroup-lv_root: 41.4 GB, 41448112128 bytes
Disk identifier: 0x00000000
Disk /dev/mapper/VolGroup-lv_swap: 973 MB, 973078528 bytes
Disk identifier: 0x00000000

[root@server2 iscsi]# service iscsi restart
Stopping iscsi: [OK]
Starting iscsi: [OK]
[root@server2 iscsi]# fdisk -l | grep Disk
Disk /dev/sda: 42.9 GB, 42949672960 bytes
Disk identifier: 0x0004f229
Disk /dev/sdb: 1073 MB, 1073741824 bytes
Disk identifier: 0x00000000
Disk /dev/sdc: 1073 MB, 1073741824 bytes
Disk identifier: 0x00000000
Disk /dev/sdd: 1073 MB, 1073741824 bytes
Disk identifier: 0x00000000
Disk /dev/sde: 2147 MB, 2147483648 bytes
Disk identifier: 0x00000000
Disk /dev/sdf: 2147 MB, 2147483648 bytes
Disk identifier: 0x00000000
Disk /dev/sdg: 2147 MB, 2147483648 bytes
Disk identifier: 0x00000000
Disk /dev/mapper/VolGroup-lv_root: 41.4 GB, 41448112128 bytes
Disk identifier: 0x00000000
Disk /dev/mapper/VolGroup-lv_swap: 973 MB, 973078528 bytes
Disk identifier: 0x00000000
Disk /dev/sdh: 1073 MB, 1073741824 bytes
Disk identifier: 0x00000000
Disk /dev/sdi: 1073 MB, 1073741824 bytes
Disk identifier: 0x00000000
Disk /dev/sdj: 1073 MB, 1073741824 bytes
Disk identifier: 0x00000000

Вы можете получить информацию о состоянии службы iscsi, воспользовавшись следующей командой:

service iscsi status

12.4. Целевой сервер iSCSI в дистрибутиве Debian

Процесс подготовки программного обеспечения для целевого сервера в дистрибутивах Ubuntu и Debian заключается в установке пакета программного обеспечения iscsitarget в обоих дистрибутивах, а также в установке дополнительного пакета программного обеспечения iscsitarget-dkms для сборки модулей ядра ОС исключительно в дистрибутиве Debian.

root@debby6:~# aptitude install iscsitarget
Следующие НОВЫЕ пакеты будут установлены:
iscsitarget
0 пакетов обновлено, 1 установлено новых, 0 пакетов отмечено для удаления, и 0 пакетов не обновлено.
Необходимо получить 69.4 kB архивов. После распаковки 262 kB будет занято.
Получить: 1 http://ftp.belnet.be/debian/ squeeze/main iscsitarget i386 1.4.20.2-1\
[69.4 kB]
Получено 69.4 кБ in 0c (415 кБ/c)
Выбор ранее не выбранного пакета iscsitarget.
(Чтение базы данных ... на данный момент установлено 36441 файлов и каталогов.)
Распаковывается iscsitarget (from .../iscsitarget_1.4.20.2-1_i386.deb) ...
Обрабатываются триггеры для man-db ...
Настраивается пакет iscsitarget (1.4.20.2-1) ...
iscsitarget not enabled in "/etc/default/iscsitarget", not starting... (warning).

При работе с дистрибутивом Debian 6 вам также придется выполнить команду aptitude install iscsitarget-dkms для установки программных компонентов, предназначенных для сборки модулей ядра ОС. В случае дистрибутива Debian 5 аналогичная команда будет выглядеть следующим образом: aptitude install iscsitarget-modules-`uname -a`. В дистрибутиве Ubuntu все необходимые модули ядра ОС включены в состав основного пакета программного обеспечения.

Целевой сервер iSCSI по умолчанию деактивирован, поэтому нам придется активировать его.

root@debby6:~# cat /etc/default/iscsitarget
ISCSITARGET_ENABLE=false
root@debby6:~# vi /etc/default/iscsitarget
root@debby6:~# cat /etc/default/iscsitarget
ISCSITARGET_ENABLE=true

12.5. Использование технологии iSCSI для доступа к файлам, созданным с помощью утилиты dd

Вы можете использовать тома LVM (/dev/md0/lvol0), физические разделы дисков (/dev/sda), устройства RAID (/dev/md0), а также обычные файлы в качестве источников данных. В данном примере мы будем использовать файлы, созданные с помощью утилиты dd.

Ниже показана методика создания трех файлов небольших размеров (100 МБ, 200 МБ и 300 МБ).

root@debby6:~# mkdir /iscsi
root@debby6:~# dd if=/dev/zero of=/iscsi/lun1.img bs=1M count=100
100+0 записей получено
100+0 записей отправлено
скопировано 104857600 байт (105 MB), 0.315825 с, 332 MB/c
root@debby6:~# dd if=/dev/zero of=/iscsi/lun2.img bs=1M

count=200
200+0 записей получено
200+0 записей отправлено
скопировано 209715200 байт (210 MB), 1.08342 с, 194 MB/с
root@debby6:~# dd if=/dev/zero of=/iscsi/lun3.img bs=1M count=300
300+0 записей получено
300+0 записей отправлено
скопировано 314572800 байт (315 MB), 1.36209 с, 231 MB/с
Мы должны объявить эти три файла источниками данных для целевого сервера iSCSI в файле /etc/iet/ietd.conf (в некоторых случаях расположенном по пути /etc/ietd.conf).
root@debby6:/etc/iet# cp ietd.conf ietd.conf.original
root@debby6:/etc/iet# > ietd.conf
root@debby6:/etc/iet# vi ietd.conf
root@debby6:/etc/iet# cat ietd.conf
Target iqn.2010-02.be.linux-training:storage.lun1
IncomingUser isuser hunter2
OutgoingUser
Lun 0 Path=/iscsi/lun1.img,Type=fileio
Alias LUN1
Target iqn.2010-02.be.linux-training:storage.lun2
IncomingUser isuser hunter2
OutgoingUser
Lun 0 Path=/iscsi/lun2.img,Type=fileio
Alias LUN2
Target iqn.2010-02.be.linux-training:storage.lun3
IncomingUser isuser hunter2
OutgoingUser
Lun 0 Path=/iscsi/lun3.img,Type=fileio
Alias LUN3
Нам также необходимо добавить информацию о наших устройствах в файл /etc/initiators.allow.
root@debby6:/etc/iet# cp initiators.allow initiators.allow.original
root@debby6:/etc/iet# >initiators.allow
root@debby6:/etc/iet# vi initiators.allow
root@debby6:/etc/iet# cat initiators.allow
iqn.2010-02.be.linux-training:storage.lun1
iqn.2010-02.be.linux-training:storage.lun2
iqn.2010-02.be.linux-training:storage.lun3
Самое время запустить настроенный сервер:
root@debby6:/etc/iet# /etc/init.d/iscsitarget start
Starting iSCSI enterprise target service:.
.
root@debby6:/etc/iet#
Для проверки корректности активации устройств для хранения данных можно прочитать содержимое файлов из директории /proc/net/iet:
root@debby6:/etc/iet# cat /proc/net/iet/volume
tid:3 name:iqn.2010-02.be.linux-training:storage.lun3
lun:0 state:0 iotype:fileio iomode:wt blocks:614400 blocksize:\
512 path:/iscsi/lun3.img
tid:2 name:iqn.2010-02.be.linux-training:storage.lun2
lun:0 state:0 iotype:fileio iomode:wt blocks:409600 blocksize:\
512 path:/iscsi/lun2.img
tid:1 name:iqn.2010-02.be.linux-training:storage.lun1
lun:0 state:0 iotype:fileio iomode:wt blocks:204800 blocksize:\
512 path:/iscsi/lun1.img
root@debby6:/etc/iet# cat /proc/net/iet/session
tid:3 name:iqn.2010-02.be.linux-training:storage.lun3
tid:2 name:iqn.2010-02.be.linux-training:storage.lun2
tid:1 name:iqn.2010-02.be.linux-training:storage.lun1

12.6. Инициатор iSCSI в дистрибутиве Ubuntu

В первую очередь мы должны установить клиентское программное обеспечение iSCSI (разумеется, не на том компьютере, который будет выполнять роль целевого сервера).
root@ubu1104:~# aptitude install open-iscsi
Чтение списков пакетов...
Построение дерева зависимостей...
Чтение информации о состоянии...
Чтение информации о расширенных состояниях...
Инициализация состояний пакетов...
Следующие НОВЫЕ пакеты будут установлены:
open-iscsi open-iscsi-utils{a}
После этого мы изменим настройки клиентского программного обеспечения iSCSI для его автоматического запуска.
root@ubu1104:/etc/iscsi# cp iscsid.conf iscsid.conf.original
root@ubu1104:/etc/iscsi# vi iscsid.conf
root@ubu1104:/etc/iscsi# grep ^node.startup iscsid.conf
node.startup = automatic
Или же, вы можете осуществить запуск этого клиентского программного обеспечения в ручном режиме.
root@ubu1104:/etc/iscsi/nodes# /etc/init.d/open-iscsi start
- Starting iSCSI initiator service iscsid [OK]
- Setting up iSCSI targets [OK]

root@ubu1104:/etc/iscsi/nodes#
Теперь мы можем соединиться с целевым сервером и использовать утилиту iscsiadm для получения информации об

устройствах, к которым он предоставляет доступ.

```
root@ubu1104:/etc/iscsi# iscsiadm -m discovery -t st -p 192.168.1.31
192.168.1.31:3260,1 iqn.2010-02.be.linux-training:storage.lun2
192.168.1.31:3260,1 iqn.2010-02.be.linux-training:storage.lun1
192.168.1.31:3260,1 iqn.2010-02.be.linux-training:storage.lun3
```

Эта же утилита iscsiadm может использоваться и для редактирования файлов из директории /etc/iscsi/nodes/.

```
root@ubu1104:/etc/iscsi# iscsiadm -m node --targetname "iqn.2010-02.be.linu\
x-training:storage.lun1" --portal "192.168.1.31:3260" --op=update --name no\
de.session.auth.authmethod --value=CHAP
root@ubu1104:/etc/iscsi# iscsiadm -m node --targetname "iqn.2010-02.be.linu\
x-training:storage.lun1" --portal "192.168.1.31:3260" --op=update --name no\
de.session.auth.username --value=isuser
root@ubu1104:/etc/iscsi# iscsiadm -m node --targetname "iqn.2010-02.be.linu\
x-training:storage.lun1" --portal "192.168.1.31:3260" --op=update --name no\
de.session.auth.password --value=hunter2
```

Описанные выше действия следует повторить и для двух других устройств.

После этого необходимо перезапустить системную службу инициатора для соединения с целевым сервером.

```
root@ubu1104:/etc/iscsi/nodes# /etc/init.d/open-iscsi restart
```

- Disconnecting iSCSI targets [OK]
- Stopping iSCSI initiator service [OK]
- Starting iSCSI initiator service iscsid [OK]
- Setting up iSCSI targets

Используйте команду fdisk -i для получения информации о трех новых устройствах iSCSI.

```
root@ubu1104:/etc/iscsi/nodes# fdisk -l 2> /dev/null | grep Disk
Disk /dev/sda: 17.2 GB, 17179869184 bytes
Disk identifier: 0x0001983f
Disk /dev/sdb: 209 MB, 209715200 bytes
Disk identifier: 0x00000000
Disk /dev/sdd: 314 MB, 314572800 bytes
Disk identifier: 0x00000000
Disk /dev/sdc: 104 MB, 104857600 bytes
Disk identifier: 0x00000000
```

После этого на стороне целевого сервера может быть получена информация об активных сессиях.

```
root@debby6:/etc/iet# cat /proc/net/iet/session
tid:3 name:iqn.2010-02.be.linux-training:storage.lun3
    sid:5348024611832320 initiator:iqn.1993-08.org.debian:01:8983ed2d770
        cid:0 ip:192.168.1.35 state:active hd:none dd:none
tid:2 name:iqn.2010-02.be.linux-training:storage.lun2
    sid:4785074624856576 initiator:iqn.1993-08.org.debian:01:8983ed2d770
        cid:0 ip:192.168.1.35 state:active hd:none dd:none
tid:1 name:iqn.2010-02.be.linux-training:storage.lun1
    sid:5066549618344448 initiator:iqn.1993-08.org.debian:01:8983ed2d770
        cid:0 ip:192.168.1.35 state:active hd:none dd:none
root@debby6:/etc/iet#
```

12.7. Использование устройств iSCSI

Между устройствами SCSI и iSCSI не существует принципиальных различий после их корректного соединения с системой: поверхность дисковых устройств должна разделяться на разделы, в этих разделах должны создаваться файловые системы, после чего созданные файловые системы должны монтироваться.

```
root@ubu1104:/etc/iscsi/nodes# history | tail -13
 94 fdisk /dev/sdc
 95 fdisk /dev/sdd
 96 fdisk /dev/sdb
 97 mke2fs /dev/sdb1
 98 mke2fs -j /dev/sdc1
 99 mkfs.ext4 /dev/sdd1
100 mkdir /mnt/is1
101 mkdir /mnt/is2
102 mkdir /mnt/is3
103 mount /dev/sdb1 /mnt/is1
104 mount /dev/sdc1 /mnt/is2
105 mount /dev/sdd1 /mnt/is3
106 history | tail -13
root@ubu1104:/etc/iscsi/nodes# mount | grep is
/dev/sdb1 on /mnt/is1 type ext2 (rw)
/dev/sdc1 on /mnt/is2 type ext3 (rw)
/dev/sdd1 on /mnt/is3 type ext4 (rw)
```

12.8. Практическое задание: устройства iSCSI

1. Настройте целевые серверы (используя устройства LVM и SCSI в качестве источников данных) и инициатор, который будет соединяться с обоими серверами.

12.9. Корректная процедура выполнения практического задания: устройства iSCSI

1. Настройте целевые серверы (используя устройства LVM и SCSI в качестве источников данных) и инициатор, который будет соединяться с обоими серверами.

Данная методика применима в случае работы с дистрибутивами Debian/Ubuntu/Mint. Для ознакомления с нюансами, которые могут возникнуть в случае работы с дистрибутивами RHEL/CentOS, следует обратиться к разделам с теоретической информацией.

Договоритесь (с сидящим рядом студентом) о том, какой из компьютеров будет выполнять функции целевого сервера, а какой — инициатора.

Действия, которые должны быть выполнены на компьютере, выполняющем роль целевого сервера:

В первую очередь с помощью стандартных инструментов вашего дистрибутива для установки программного обеспечения следует установить пакет программного обеспечения iscsitarget. После этого вы должны использовать знания, полученные в ходе чтения предыдущей главы, для настройки логического тома (/dev/vg/lvol0), а также главы, посвященной технологии RAID, для настройки устройства RAID, которое будет представлено файлом устройства /dev/md0. Далее вы должны выполнить следующую команду для модификации файла конфигурации:

vi /etc/default/iscsitarget (Установите значение true для параметра enable)

Добавьте информацию о ваших устройствах в файл конфигурации /etc/iet/ietf.conf:

root@debby6:/etc/iet# cat ietd.conf
Target iqn.2010-02.be.linux-training:storage.lun1
IncomingUser isuser hunter2
OutgoingUser
Lun 0 Path=/dev/vg/lvol0,Type=fileio
Alias LUN1
Target iqn.2010-02.be.linux-training:storage.lun2
IncomingUser isuser hunter2
OutgoingUser
Lun 0 Path=/dev/md0,Type=fileio
Alias LUN2

Добавьте идентификаторы обоих устройств в файл конфигурации /etc/iet/initiators.allow:

root@debby6:/etc/iet# cat initiators.allow
iqn.2010-02.be.linux-training:storage.lun1
iqn.2010-02.be.linux-training:storage.lun2

Действия, которые должны быть выполнены на компьютере, выполняющим роль инициатора:

Установите пакет программного обеспечения open-iscsi и запустите соответствующий демон.

После этого используйте команду iscsiadm -m discovery -t st целевой-ip для получения информации об устройствах iSCSI на целевом сервере.

Отредактируйте файлы из директории /etc/iscsi/nodes/ таким образом, как было описано в разделах теоретической части главы. После этого перезапустите демон iSCSI и выполните команду fdisk -i для получения информации об используемых устройствах iSCSI.

Глава 13. Знакомство с технологией резервирования каналов передачи данных

13.1. Установка необходимых программных компонентов

В дистрибутивах RHEL и CentOS необходимо установить пакет программного обеспечения с именем device-mapper-multipath.

yum install device-mapper-multipath

После установки этого пакета в директории /usr/share/doc/device-mapper-multipath-0.4.9/ появится шаблон файла конфигурации с именем multipath.conf.

Файл конфигурации /etc/multipath.conf не будет создан до тех пор, пока вы не проведете инициализацию специализированного демона с помощью утилиты mpathconf.

[root@server2 ~]# mpathconf --enable --with_multipathd y
Starting multipathd daemon: [OK]
[root@server2 ~]# wc -l /etc/multipath.conf
99 /etc/multipath.conf

13.2. Настройка механизма резервирования каналов передачи данных

Теперь вы можете либо вручную отредактировать файл /etc/multipath.conf, либо использовать утилиту mpathconf для его автоматического редактирования.

[root@server2 ~]# grep user_friendly_names /etc/multipath.conf
user_friendly_names yes
user_friendly_names yes
[root@server2 ~]# mpathconf --enable --user_friendly_names n
[root@server2 ~]# grep user_friendly_names /etc/multipath.conf
user_friendly_names no
user_friendly_names yes

```
[root@server2 ~]# mpathconf --enable --user_friendly_names y
[root@server2 ~]# grep user_friendly_names /etc/multipath.conf
user_friendly_names yes
# user_friendly_names yes
```

13.3. Сетевые соединения

В данном примере используются три сетевых соединения, поэтому следует убедиться в том, что целевой сервер iSCSI доступен посредством каждого из соединений.

```
root@server1 tgt]# ifconfig | grep -B1 192.168
eth1 Link encap:Ethernet HWaddr 08:00:27:4E:AB:8E
inet addr:192.168.1.98 Bcast:192.168.1.255 Mask:255.255.255.0
eth2 Link encap:Ethernet HWaddr 08:00:27:3F:A9:D1
inet addr:192.168.2.98 Bcast:192.168.2.255 Mask:255.255.255.0
eth3 Link encap:Ethernet HWaddr 08:00:27:94:52:26
inet addr:192.168.3.98 Bcast:192.168.3.255 Mask:255.255.255.0
```

Это же утверждение справедливо и для инициатора, поддерживающего технологию резервирования каналов передачи данных.

```
[root@server2 ~]# ifconfig | grep -B1 192.168
eth1 Link encap:Ethernet HWaddr 08:00:27:A1:43:41
inet addr:192.168.1.99 Bcast:192.168.1.255 Mask:255.255.255.0
eth2 Link encap:Ethernet HWaddr 08:00:27:12:A8:70
inet addr:192.168.2.99 Bcast:192.168.2.255 Mask:255.255.255.0
eth3 Link encap:Ethernet HWaddr 08:00:27:6E:99:9B
inet addr:192.168.3.99 Bcast:192.168.3.255 Mask:255.255.255.0
```

13.4. Запуск системных служб multipathd и iscsi

Пришло время запустить (или перезапустить) системные службы multipathd и iscsi:

```
[root@server2 ~]# service multipathd restart
Stopping multipathd daemon: [ OK ]
Starting multipathd daemon: [ OK ]
[root@server2 ~]# service iscsi restart
Stopping iscsi: [ OK ]
Starting iscsi: [ OK ]
```

Ниже приведен вывод утилиты fdisk в случае использования в файле конфигурации демона значения yes параметра friendly_names. Три последних устройства являются используемыми устройствами с возможностью резервирования каналов передачи данных.

```
[root@server2 ~]# fdisk -l | grep Disk
Disk /dev/sda: 42.9 GB, 42949672960 bytes
Disk identifier: 0x0004f229
Disk /dev/sdb: 1073 MB, 1073741824 bytes
Disk identifier: 0x00000000
Disk /dev/sdc: 1073 MB, 1073741824 bytes
Disk identifier: 0x00000000
Disk /dev/sdd: 1073 MB, 1073741824 bytes
Disk identifier: 0x00000000
Disk /dev/sde: 2147 MB, 2147483648 bytes
Disk identifier: 0x00000000
Disk /dev/sdf: 2147 MB, 2147483648 bytes
Disk identifier: 0x00000000
Disk /dev/sdg: 2147 MB, 2147483648 bytes
Disk identifier: 0x00000000
Disk /dev/mapper/VolGroup-lv_root: 41.4 GB, 41448112128 bytes
Disk identifier: 0x00000000
Disk /dev/mapper/VolGroup-lv_swap: 973 MB, 973078528 bytes
Disk identifier: 0x00000000
Disk /dev/sdh: 1073 MB, 1073741824 bytes
Disk identifier: 0x00000000
Disk /dev/sdi: 1073 MB, 1073741824 bytes
Disk identifier: 0x00000000
Disk /dev/sdj: 1073 MB, 1073741824 bytes
Disk identifier: 0x00000000
Disk /dev/sdl: 1073 MB, 1073741824 bytes
Disk identifier: 0x00000000
Disk /dev/sdn: 1073 MB, 1073741824 bytes
Disk identifier: 0x00000000
Disk /dev/sdk: 1073 MB, 1073741824 bytes
Disk identifier: 0x00000000
Disk /dev/sdm: 1073 MB, 1073741824 bytes
Disk identifier: 0x00000000
Disk /dev/sdp: 1073 MB, 1073741824 bytes
Disk identifier: 0x00000000
Disk /dev/sdo: 1073 MB, 1073741824 bytes
Disk identifier: 0x00000000
Disk /dev/mapper/mpathh: 1073 MB, 1073741824 bytes
Disk identifier: 0x00000000
Disk /dev/mapper/mpathi: 1073 MB, 1073741824 bytes
Disk identifier: 0x00000000
Disk /dev/mapper/mpathj: 1073 MB, 1073741824 bytes
Disk identifier: 0x00000000
[root@server2 ~]#
```

13.5. Список каналов передачи данных

Вы можете получить список соединений, предназначенных для передачи данных, а также используемых устройств с возможностью резервирования каналов передачи данных, воспользовавшись командой multipath -ll.

```
[root@server2 ~]# multipath -ll
mpathj (1IET 00010001) dm-4 Reddy,VBOX HARDDISK
size=1.0G features=0 hwhandler=0 wp=rw
|-+- policy=round-robin 0 prio=1 status=active
| `- 13:0:0:1 sdh 8:112 active ready running
|-+- policy=round-robin 0 prio=1 status=enabled
| `- 12:0:0:1 sdi 8:128 active ready running
`-+- policy=round-robin 0 prio=1 status=enabled
`- 14:0:0:1 sdm 8:192 active ready running
mpathi (1IET 00010003) dm-3 Reddy,VBOX HARDDISK
size=1.0G features=0 hwhandler=0 wp=rw
|-+- policy=round-robin 0 prio=1 status=active
| `- 13:0:0:3 sdk 8:160 active ready running
|-+- policy=round-robin 0 prio=1 status=enabled
| `- 12:0:0:3 sdn 8:208 active ready running
`-+- policy=round-robin 0 prio=1 status=enabled
`- 14:0:0:3 sdp 8:240 active ready running
mpathh (1IET 00010002) dm-2 Reddy,VBOX HARDDISK
size=1.0G features=0 hwhandler=0 wp=rw
|-+- policy=round-robin 0 prio=1 status=active
| `- 12:0:0:2 sdl 8:176 active ready running
|-+- policy=round-robin 0 prio=1 status=enabled
| `- 13:0:0:2 sdj 8:144 active ready running
`-+- policy=round-robin 0 prio=1 status=enabled
`- 14:0:0:2 sdo 8:224 active ready running
[root@server2 ~]#
```

Идентификаторы IET (iSCSI Enterprise Target) должны совпадать с соответствующими идентификаторами на стороне целевого сервера.

```
[root@server1 ~]# tgt-admin -s | grep -e LUN -e IET -e dev
LUN information:
LUN: 0
SCSI ID: IET 00010000
LUN: 1
SCSI ID: IET 00010001
Backing store path: /dev/sdb
LUN: 2
SCSI ID: IET 00010002
Backing store path: /dev/sdc
LUN: 3
SCSI ID: IET 00010003
Backing store path: /dev/sdd
```

13.6. Использование устройства

При работе с устройствами с возможностью резервирования каналов передачи данных используются такие стандартные команды, как mkfs, mkdir, mount:

```
[root@server2 ~]# mkfs.ext4 /dev/mapper/mpathi
mke2fs 1.41.12 (17-May-2010)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
65536 inodes, 262144 blocks
13107 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=268435456
8 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
32768, 98304, 163840, 229376
Writing inode tables: done
Creating journal (8192 blocks): done
Writing superblocks and filesystem accounting information: done
This filesystem will be automatically checked every 38 mounts or
180 days, whichever comes first. Use tune2fs -c or -i to override.
[root@server2 ~]# mkdir /srv/multipath
[root@server2 ~]# mount /dev/mapper/mpathi /srv/multipath/
[root@server2 ~]# df -h /srv/multipath/
Filesystem Size Used Avail Use% Mounted on
/dev/mapper/mpathi 1008M 34M 924M 4% /srv/multipath
```

13.7. Практическое задание: технология резервирования каналов передачи данных

1. Вместе с сидящим рядом студентом примите решение о том, кто на своем компьютере будет настраивать целевой сервер iSCSI, а кто - инициатор iSCSI и демон резервирования каналов передачи данных. Осуществите настройку программных компонентов, необходимых для задействования механизма резервирования каналов передачи данных таким образом, как описано в разделах с теоретической информацией.
2. Раскомментируйте большую секцию defaults в файле конфигурации /etc/multipath.conf и деактивируйте параметр friendly_names в этом же файле. Проверьте, работоспособен ли после этих манипуляций механизм резервирования каналов передачи данных. Вам наверняка придется обратиться к страницам руководства для утилиты

/lib/dev/scsi_id и файла конфигурации multipath.conf.

13.8. Корректная процедура выполнения практического задания: технология резервирования каналов передачи данных

1. Вместе с сидящим рядом студентом примите решение о том, кто на своем компьютере будет настраивать целевой сервер iSCSI, а кто - инициатор iSCSI и демон резервирования каналов передачи данных. Осуществите настройку программных компонентов, необходимых для задействования механизма резервирования каналов передачи данных таким образом, как описано в разделах с теоретической информацией.

Обратитесь к разделу с теоретической информацией...

1. Раскомментируйте большую секцию defaults в файле конфигурации /etc/multipath.conf и деактивируйте параметр friendly_names в этом же файле. Проверьте, работоспособен ли после этих манипуляций механизм резервирования каналов передачи данных. Вам наверняка придется обратиться к страницам руководства для утилиты /lib/dev/scsi_id и файла конфигурации multipath.conf.

vi multipath.conf

Удалите символ # перед объявлением большой секции defaults.

Добавьте символ # перед строкой friendly_names active.

Добавьте параметр --replace-whitespace в список параметров утилиты scsi_id.

```
defaults {
udev_dir /dev
polling_interval 10
path_selector "round-robin 0"
path_grouping_policy multibus
getuid_callout "/lib/udev/scsi_id --whitelisted --replace\
-whitespace --device=/dev/%n"
prio const
path_checker readsector0
rr_min_io 100
max_fds 8192
rr_weight priorities
failback immediate
no_path_retry fail
user_friendly_names no
}
```

Теперь (после перезапуска системной службы) список соединений и устройств будет выглядеть следующим образом:

```
root@server2 etc]# multipath -ll
1IET_00010001 dm-8 Reddy,VBOX HARDDISK
size=1.0G features='0' hwhandler='0' wp=rw
`-+- policy=round-robin 0 prio=1 status=active
|- 17:0:0:1 sdh 8:112 active ready running
|- 16:0:0:1 sdi 8:128 active ready running
`- 15:0:0:1 sdn 8:208 active ready running
1IET_00010003 dm-10 Reddy,VBOX HARDDISK
size=1.0G features=0 hwhandler=0 wp=rw
`-+- policy=round-robin 0 prio=1 status=active
|- 17:0:0:3 sdl 8:176 active ready running
|- 16:0:0:3 sdm 8:192 active ready running
`- 15:0:0:3 sdp 8:240 active ready running
1IET_00010002 dm-9 Reddy,VBOX HARDDISK
size=1.0G features=0 hwhandler=0 wp=rw
`-+- policy=round-robin 0 prio=1 status=active
|- 17:0:0:2 sdj 8:144 active ready running
|- 16:0:0:2 sdk 8:160 active ready running
`- 15:0:0:2 sdo 8:224 active ready running
```

Вы запретили доступ к своим устройствам?

vi multipath.conf

-→ search for blacklist:

```
add
devnode "^sd[a-g]"
```

Глава 14. Системный загрузчик

В данной главе кратко описывается процесс загрузки компьютера (с 32- или 64-битным центральным процессором архитектуры Intel), работающего под управлением Linux.

Системы, загрузка которых осуществляется средствами системного загрузчика lilo, на сегодняшний день встречаются достаточно редко, поэтому в соответствующем разделе приведено лишь сжатое описание данного системного загрузчика.

На сегодняшний день наиболее часто используемым в системах Linux с длительным сроком поддержки системным загрузчиком является системный загрузчик grub даже несмотря на то, что он и не развивается в рамках проекта, непосредственно связанного с разработкой ядра Linux. Такие дистрибутивы операционных систем, как FreeBSD и Solaris также используют системный загрузчик grub.

Следует отметить, что системный загрузчик grub не ограничивается архитектурой центрального процессора Intel. Он может загружать операционные системы и в случае использования компьютеров с центральными процессорами архитектуры PowerPC.

Обратите внимание на то, что хотя системный загрузчик grub и устанавливается по умолчанию в дистрибутиве Debian, на данный момент в большинстве дистрибутивов идет медленная миграция на системный загрузчик grub2.

14.1. Терминология

Точная последовательность операций, выполняемых при включении компьютера, зависит от его аппаратной архитектуры (аппаратная архитектура Intel x86 отличается от аппаратной архитектуры Sun Sparc и других аппаратных архитектур), системного загрузчика (системный загрузчик grub отличается от системного загрузчика lilo) и операционной системы (Linux, Solaris, BSD, и т.д.). В данной главе по большей части рассматривается процесс загрузки операционной системы Linux с помощью системного загрузчика grub на компьютере с аппаратной архитектурой Intel x86.

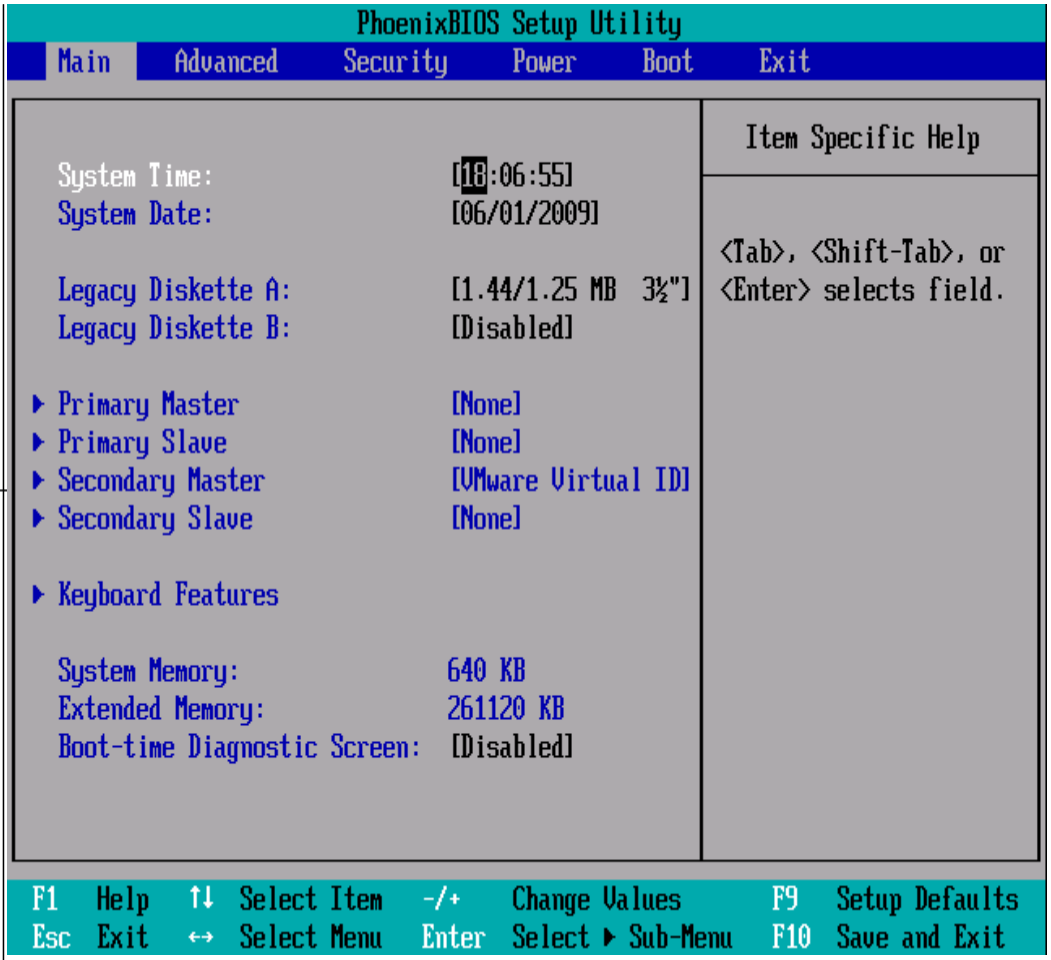
14.1.1. Самостоятельное тестирование системы в момент запуска

Загрузка компьютера начинается в момент включения его блока питания (и это не шутка). Первый выполняемый процесс называется POST (Power On Self Test — самостоятельное тестирование системы в момент запуска). В том случае, если тестирование завершается успешно, начинается загрузка BIOS. Если же не все так гладко, вы можете не услышать ничего, услышать повторяющуюся последовательность звуковых сигналов, увидеть на экране монитора сообщение об ошибке или в некоторых случаях увидеть дым, идущий из корпуса компьютера (после сгорания комплектующих появляется отвратительный запах!).

14.1.2. BIOS

Все компьютеры с аппаратной архитектурой Intel x86 используют BIOS (Basic Input/Output system — базовая система ввода/вывода) для выявления, идентификации и инициализации аппаратного обеспечения. После этого BIOS осуществляет поиск загрузочного устройства. Таким устройством может быть гибкий диск, жесткий диск, CDROM, сетевая карта или устройство для хранения данных с интерфейсом USB.

В процессе функционирования BIOS вы можете увидеть на экране сообщение о том, какую клавишу следует нажать для перехода в режим установки значений параметров BIOS (обычно это либо клавиша Del, либо клавиша F2).



14.1.3. Openboot

Системы Sun Sparc используют openboot для тестирования аппаратного обеспечения и загрузки операционных систем. Bill Callkins описывает openboot в своих книгах, посвященных администрированию систем Solaris. Подробное описание возможностей openboot не соответствует целям данного курса.

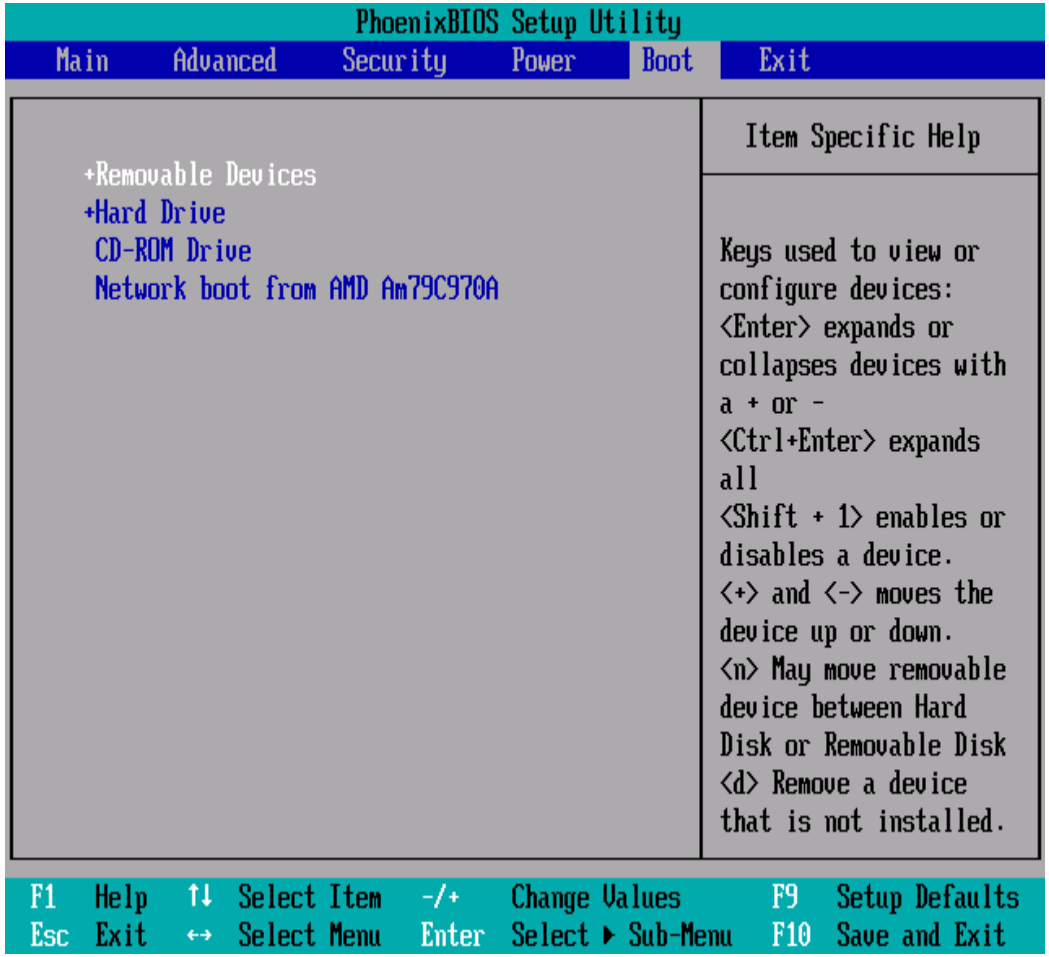
14.1.4. Пароль для загрузки системы

BIOS позволяет установить пароль для загрузки системы. Не забывайте этот пароль, так как в том случае, если вы забудете его, вам придется открывать корпус компьютера и производить специальные манипуляции для его сброса. В некоторых случаях вам может быть предоставлена возможность установки пароля

для загрузки системы, а так же отдельного пароля для защиты от модификации значений параметров BIOS.

14.1.5. Загрузочное устройство

BIOS будет проверять доступные устройства для хранения данных в поисках загрузочного устройства в том порядке, который был задан в режиме модификации значений параметров BIOS. На компьютерах, находящихся в промышленной эксплуатации, операционная система чаше всего загружается с жесткого диска.



14.1.6. Основная загрузочная запись

Основная загрузочная запись (Master Boot Record или MBR) находится в первом секторе жесткого диска. В этой области диска хранится информация о первичных разделах жесткого диска и его активном разделе.

Основная загрузочная запись имеет размер, равный 512 байтам и может быть скопирована с помощью утилиты dd.

dd if=/dev/sda of=bootsect.mbr count=1 bs=512

14.1.7. Системный загрузчик

Код из основной загрузочной записи исполняется средствами BIOS, причем в основной загрузочной записи может быть размещен как код полноценного системного загрузчика, так и вспомогательный код для запуска системного загрузчика.

При рассмотрении с помощью утилиты od данных, находящихся в области основной загрузочной записи, можно обнаружить информацию о системном загрузчике.

paul@laika:~$ sudo dd if=/dev/sda count=1 bs=16 skip=24 2>/dev/null|od -c

0000000 376 G R U B \0 G e o m \0 H a r d

0000020

Существует множество системных загрузчиков, но наиболее часто используемым на системах с аппаратной архитектурой Intel системным загрузчиком является системный загрузчик grub, который во многих случаях приходит на замену системному загрузчику lilo. При установке Linux на систему с аппаратной архитектурой sparc вы можете выбрать системный загрузчик silo. Для загрузки систем с аппаратной архитектурой Itanium используется системный загрузчик elilo. В системах с аппаратными архитектурами IBM S/390 и zSeries используется системный загрузчик z/IPL. В системах с аппаратной архитектурой Alpha используется системный загрузчик milo, а в системах с аппаратной архитектурой PowerPC — системный загрузчик yaboot (расшифровывается как yet another boot loader — еще один системный загрузчик).

Для создания загрузочных дисков CD и DVD обычно используется системный загрузчик syslinux.

14.1.8. Ядро ОС

Цель процесса загрузки компьютера заключается в загрузке операционной системы или только ядра операционной системы. Обычный системный загрузчик, такой, как grub, осуществляет копирование данных ядра операционной системы с жесткого диска в оперативную память, после чего передает управление компьютером непосредственно этому ядру операционной системы (исполняет код ядра операционной системы).

После того, как ядро операционной системы загружено, системный загрузчик лишается контроля над ним. С этого момента ядро операционной системы осуществляет управление системой.

После обсуждения системных загрузчиков мы продолжим рассматривать процесс загрузки системы и обсудим системы инициализации, с помощью которых осуществляется запуск системных демонов.

14.2. Системный загрузчик grub

14.2.1. Файл конфигурации /boot/grub/grub.cfg

Дистрибутив Debian перешел на системный загрузчик grub2, который будет обсуждаться в следующем разделе. Главным файлом конфигурации меню загрузки системного загрузчика grub2 является файл с именем grub.cfg.

```
root@debian7:~# ls -l /boot/grub/grub.cfg
-r—r--r-- 1 root root 2453 май 13 17:22 /boot/grub/grub.cfg
root@debian7:~#
```

14.2.2. Файл конфигурации /boot/grub/grub.conf

В таких дистрибутивах, как Red Hat Enterprise Linux 6, для конфигурации меню загрузки все еще используется файл с именем grub.conf, причем на данный файл установлены две символьные ссылки: /boot/grub/menu.lst и /etc/grub.conf.

```
[root@centos65 ~]# ls -l /boot/grub/menu.lst
lrwxrwxrwx. 1 root root 11 мар 7 11:53 /boot/grub/menu.lst → ./grub.conf
[root@centos65 ~]# ls -l /boot/grub/grub.conf
-rw-------. 1 root root 1189 май 5 11:47 /boot/grub/grub.conf
[root@centos65 ~]#
```

Содержимое этого файла на данный момент (в дистрибутиве RHEL 6.5) выглядит аналогичным образом:

```
[root@centos65 ~]# more /boot/grub/grub.conf
# 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/mapper/VolGroup-lv_root
# initrd /initrd-[generic-]version.img
#boot=/dev/sda
default=0
timeout=5
splashimage=(hd0,0)/grub/splash.xpm.gz
hiddenmenu
title CentOS (2.6.32-431.11.2.el6.x86_64)
root (hd0,0)
kernel /vmlinuz-2.6.32-431.11.2.el6.x86_64 ro root=/dev/mapper/VolGr\
oup-lv_root rd_NO_LUKS LANG=en_US.UTF-8 rd_NO_MD rd_LVM_LV=VolGroup/lv_swap \
SYSFONT=latarcyrheb-sun16 crashkernel=auto rd_LVM_LV=VolGroup/lv_root KEYBO\
ARDTYPE=pc KEYTABLE=us rd_NO_DM rhgb quiet
initrd /initramfs-2.6.32-431.11.2.el6.x86_64.img
title CentOS (2.6.32-431.el6.x86_64)
root (hd0,0)
kernel /vmlinuz-2.6.32-431.el6.x86_64 ro root=/dev/mapper/VolGroup-l\
v_root rd_NO_LUKS LANG=en_US.UTF-8 rd_NO_MD rd_LVM_LV=VolGroup/lv_swap SYSFO\
NT=latarcyrheb-sun16 crashkernel=auto rd_LVM_LV=VolGroup/lv_root KEYBOARDTY\
PE=pc KEYTABLE=us rd_NO_DM rhgb quiet
initrd /initramfs-2.6.32-431.el6.x86_64.img
[root@centos65 ~]#
```

14.2.3. Команды меню

Команды меню должны быть расположены в начале файла конфигурации меню системного загрузчика grub.

default

Команда default устанавливает выбранный по умолчанию элемент меню загрузки. Первый элемент меню загрузки имеет идентификатор 0.

```
default=0
```

Описание каждого элемента меню или станса начинается с директивы title.

fallback

В том случае, если загрузка выбранного по умолчанию с помощью команды default элемента меню загрузки завершается неудачей, будет осуществляться загрузка элемента этого же меню, выбранного с помощью команды fallback.

```
fallback=1
```

timeout

В случае использования команды timeout системный загрузчик будет ожидать заданное количество секунд перед загрузкой выбранного по умолчанию с помощью команды default элемента меню загрузки.

```
timeout=5
```

hiddenmenu

Команда hiddenmenu позволяет скрыть меню загрузки системного загрузчика grub до момента нажатия пользователем клавиши Esc перед истечением времени ожидания, заданного с

помощью команды timeout.

hiddenmenu

title

С команды title вы можете начинать описание нового элемента меню загрузки или станса.

title CentOS (2.6.32-431.11.2.el6.x86_64)

password

С помощью команды password вы можете добавить этап проверки пароля в процессе функционирования системного загрузчика grub для предотвращения выбора окружения загрузки операционной системы в интерактивном режиме.

password --md5 1Ec.id/$T2C2ahI/EG3WRRsmmu/HN/

Используйте интерактивную командную оболочку системного загрузчика grub для создания хэша пароля.

grub> md5crypt

Password: **

Encrypted: 1Ec.id/$T2C2ahI/EG3WRRsmmu/HN/

14.2.4. Команды, используемые в стансах

Каждая операционная система или ядро операционной системы, которое вы хотите загрузить, с помощью системного загрузчика grub должно быть описано в рамках станса (stanza), являющегося по своей сути набором из нескольких строк данных конфигурации. Ниже приведены описания некоторых наиболее часто используемых в рамках станс команд.

boot

В техническом плане команда boot является единственной обязательной командой для исполнения строки команды системного загрузчика grub. Эта команда не имеет каких-либо параметров и может использоваться исключительно в качестве последней команды станса.

boot

kernel

Команда kernel позволяет указать путь к файлу ядра ОС. Под загрузкой Linux обычно подразумевается загрузка исполняемого файла ядра zImage, сжатого с помощью утилиты gzip, или загрузка исполняемого файла ядра bzImage, сжатого с помощью утилиты bzip2.

В данном примере показан вариант использования команды kernel для загрузки ядра ОС в дистрибутиве Debian.

kernel /boot/vmlinuz-2.6.17-2-686 root=/dev/hda1 ro

А это пример использования команды kernel для загрузки ядра ОС в дистрибутиве RHEL 5.

kernel /vmlinuz-2.6.18-128.el5 ro root=/dev/VolGroup00/LogVol00 rhgb quiet

Все параметры команды kernel могут обрабатываться как самим ядром ОС, так и любой другой программой (которая будет запущена позднее) путем чтения данных из файла /proc/cmdline.

initrd

Во многих случаях при установке дистрибутива Linux появляется необходимость в подключении начального виртуального диска в процессе загрузки. Эта операция описывается на уровне системного загрузчика grub с помощью команды initrd.

Ниже приведен пример использования упомянутой команды в дистрибутиве Debian 4.0.

initrd /boot/initrd.img-2.6.17-2-686

И аналогичный пример для дистрибутива Red Hat Enterprise Linux 5.

initrd /initrd-2.6.18-128.el5.img

root

Команда root принимает в качестве параметра описание раздела устройства для хранения данных с файлами корневой файловой системы.

Команда root позволяет указать необходимый жесткий диск и его раздел, причем запись hd0 соответствует первому жесткому диску, hd1 — второму жесткому диску и.т.д. Аналогичная нумерация используется по отношению к разделам жесткого диска, следовательно, запись hd0,0 будет указывать на первый раздел первого жесткого диска, запись hd0,1 — на второй раздел этого же жесткого диска и.т.д.

root (hd0,0)

savedefault

Команда sevedefault может быть использована вместе с командой default saved для изменения поведения меню. Благодаря использованию описанной комбинации команд станс, соответствующий загруженной операционной системе, будет выбран по умолчанию при следующей загрузке.

default saved
timeout 10
title Linux
root (hd0,0)
kernel /boot/vmlinuz
savedefault
title DOS
root (hd0,1)
makeactive
chainloader +1
savedefault

14.2.5. Последовательная загрузка операционной системы

В случае использования системного загрузчика grub у вас есть

два варианта загрузки операционной системы: непосредственная загрузка операционной системы или ее последовательная загрузка средствами нескольких системных загрузчиков (chainloading). При последовательной загрузке операционной системы системный загрузчик grub осуществляет загрузку кода из загрузочного сектора раздела (который содержит данные операционной системы).

Некоторые устаревшие операционные системы требуют использования первичного раздела, отмеченного как активный раздел. Так как активным разделом может быть только один раздел устройства для хранения данных, системный загрузчик grub может отметить нужный раздел соответствующим образом непосредственно перед последовательной загрузкой операционной системы.

В примере ниже показана методика отметки первого основного раздела жесткого диска как активного средствами системного загрузчика grub.

```
root (hd0,0)
makeactive
```

Последовательная загрузка операционной системы связана с загрузкой средствами системного загрузчика grub системного загрузчика другой операционной системы. Команда chainloader принимает один параметр: количество секторов для чтения и загрузки. Для операционных систем DOS и OS/2 достаточно одного сектора. Обратите внимание на то, что операционная система DOS требует, чтобы загрузочный/корневой раздел был активным!

Ниже приведен полный пример последовательной загрузки устаревшей операционной системы.

```
title MS-DOS 6.22
root (hd0,1)
makeactive
chainloader +1
```

14.2.6. Простые примеры станс

Это пример станса для загрузки дистрибутива Debian 4.

```
title Debian GNU/Linux, kernel 2.6.17-2-686
root (hd0,0)
kernel /boot/vmlinuz-2.6.17-2-686 root=/dev/hda1 ro
initrd /boot/initrd.img-2.6.17-2-686
```

А это пример станса для загрузки дистрибутива Red Hat Enterprise Linux 5.

```
title Red Hat Enterprise Linux Server (2.6.18-128.el5)
root (hd0,0)
kernel /vmlinuz-2.6.18-98.el5 ro root=/dev/VolGroup00/LogVol00 rhgb quiet
initrd /initrd-2.6.18-98.el5.img
```

14.2.7. Редактирование станс средствами системного загрузчика grub в процессе загрузки системы

В процессе загрузки системы при показе меню системного загрузчика grub вы можете нажать клавишу e для редактирования текущего станса. Эта возможность позволит вам передать дополнительные параметры ядру ОС.

Одним из таких параметров, который может оказаться полезным в случае утраты пароля пользователя root, является параметр single. Этот параметр позволит вам загрузить ядро ОС в однопользовательском режиме (хотя в некоторых дистрибутивах вам все равно придется ввести пароль пользователя root).

```
kernel /boot/vmlinuz-2.6.17-2-686 root=/dev/hda1 ro single
```

Другим параметром для сброса пароля пользователя root является параметр init=/bin/bash.

```
kernel /boot/vmlinuz-2.6.17-2-686 root=/dev/hda1 ro init=/bin/bash
```

Обратите внимание на то, что в некоторых дистрибутивах данный параметр может быть деактивирован на этапе компиляции ядра ОС.

14.2.8. Установка системного загрузчика grub

Используйте утилиту grub-install для установки системного загрузчика grub. Этой утилите потребуется параметр, указывающий путь к файлу устройства для хранения данных, на котором будет перезаписан загрузочный сектор.

```
# grub-install /dev/hda
```

Вам придется выполнять описанное действие в ручном режиме крайне редко, так как системный загрузчик grub устанавливается в процессе установки операционной системы и не должен переустанавливаться при изменении конфигурации системы (как в случае с системным загрузчиком lilo).

14.3. Системный загрузчик grub2

14.3.1. grub 2.0?

Главным конфигурационным файлом в данной версии системного загрузчика является файл /boot/grub/grub.cfg. Несмотря на то, что синтаксис этого файла может показаться знакомым, его не нужно редактировать (так как этот файл генерируется автоматически!).

```
root@debian7:~# ls -l /boot/grub/grub.cfg
-r—r--r-- 1 root root 2453 Май 13 17:22 /boot/grub/grub.cfg
root@debian7:~# head -3 /boot/grub/grub.cfg
#
# DO NOT EDIT THIS FILE
#
```

14.3.2. Файл конфигурации дополнительных элементов меню загрузки /etc/grub.d/40_custom

Файл /etc/grub.d/40_custom может модифицироваться с целью добавления в меню загрузки дополнительных элементов. Соответствующие этим элементам данные впоследствии могут быть автоматически добавлены в основной файл конфигурации системного загрузчика grub.

```
root@debian7:~# ls -l /etc/grub.d/40_custom
-rwxr-xr-x 1 root root 214 июл 3 2013 /etc/grub.d/40_custom
root@debian7:~# cat /etc/grub.d/40_custom
#!/bin/sh
exec tail -n +3 $0
# This file provides an easy way to add custom menu entries. Simply type the
# menu entries you want to add after this comment. Be careful not to change
# the exec tail line above.
```

14.3.3. Файл конфигурации /etc/default/grub

Файлом конфигурации, позволяющим изменить поведение новой версии системного загрузчика grub, является файл /etc/default/grub.

```
root@debian7:~# head /etc/default/grub
# If you change this file, run update-grub afterwards to update
# /boot/grub/grub.cfg.
# For full documentation of the options in this file, see:
# info -f grub -n Simple configuration
GRUB_DEFAULT=0
GRUB_TIMEOUT=5
GRUB_DISTRIBUTOR=lsb_release -i -s 2> /dev/null || echo Debian
GRUB_CMDLINE_LINUX_DEFAULT="quiet"
GRUB_CMDLINE_LINUX="debian-installer=en_US"
```

14.3.4. Утилита update-grub

При любой модификации файла конфигурации /etc/default/grub вам придется использовать утилиту update-grub с целью записи изменений в основной файл конфигурации системного загрузчика.

```
root@debian7:~# vi /etc/default/grub
root@debian7:~# update-grub
Generating grub.cfg ...
Found linux image: /boot/vmlinuz-3.2.0-4-amd64
Found initrd image: /boot/initrd.img-3.2.0-4-amd64
done
```

14.4. Системный загрузчик lilo

14.4.1. Linux loader

Системный загрузчик lilo в прошлом являлся наиболее часто используемым системным загрузчиком в дистрибутивах Linux, но в последнее время наметилась тенденция его замены на такие системные загрузчики, как описанный выше grub и недавно выпущенный grub2.

14.4.2. Файл конфигурации lilo.conf

Ниже приведен пример содержимого файла конфигурации lilo.conf. Параметр delay позволяет установить длительность периода ожидания перед загрузкой операционной системы в десятых долях секунды. Следовательно, установленная длительность периода ожидания равна 3 секундам, а не 30, как может показаться на первый взгляд!

```
boot = /dev/hda
delay = 30
image = /boot/vmlinuz
root = /dev/hda1
label = Red Hat 5.2
image = /boot/vmlinuz
root = /dev/hda2
label = S.U.S.E. 8.0
other = /dev/hda4
table = /dev/hda
label = MS-DOS 6.22
```

В файле конфигурации содержатся три примера станс. С помощью первого станса осуществляется загрузка дистрибутива Red Hat с первого раздела первого жесткого диска (hda1). Второй станс предназначен для загрузки дистрибутива Suse 8.0 со следующего раздела того же диска. Последний станс предназначен для загрузки операционной системы MS-DOS.

14.5. Практическое задание: системный загрузчик

1. Выясните, какой из системных загрузчиков используется на вашем компьютере: lilo, grub или grub2. Выполняйте только те практические задания, которые выполнимы в рамках вашей системы.
2. Создайте копию исполняемого файла ядра ОС, файла виртуального диска initrd и файла System.map в директории /boot. Разместите копии в той же директории /boot, заменив в именах 2.x или 3.x на 4.0 (просто представьте, что состоялся выпуск ядра Linux версии 4.0).
3. Добавьте в файл конфигурации системного загрузчика grub станс для загрузки файлов версии 4.0. Убедитесь в том, что вы используете отличное название пункта меню загрузки.
4. Установите длительность периода ожидания перед загрузкой операционной системы, равную 30 секундам.
5. Перезагрузите компьютер и проверьте работоспособность

нового станса.

14.6. Корректная процедура выполнения практического задания: системный загрузчик

1. Выясните, какой из системных загрузчиков используется на вашем компьютере: lilo, grub или grub2. Выполняйте только те практические задания, которые выполнимы в рамках вашей системы.
2. Создайте копию исполняемого файла ядра ОС, файла виртуального диска initrd и файла System.map в директории /boot. Разместите копии в той же директории /boot, заменив в именах 2.x или 3.x на 4.0 (просто представьте, что состоялся выпуск ядра Linux версии 4.0).

```
[root@centos65 boot]# uname -r
2.6.32-431.11.2.el6.x86_64
[root@centos65 boot]# cp System.map-2.6.32-431.11.2.el6.x86_64 System.map-4.0
[root@centos65 boot]# cp vmlinuz-2.6.32-431.11.2.el6.x86_64 vmlinuz-4.0
[root@centos65 boot]# cp initramfs-2.6.32-431.11.2.el6.x86_64.img initramfs-4.0\
img
```

Не забывайте о том, что файл initrd (или initramfs) имеет расширение .img.

1. Добавьте в файл конфигурации системного загрузчика grub станс для загрузки файлов версии 4.0. Убедитесь в том, что вы используете отличное название пункта меню загрузки.

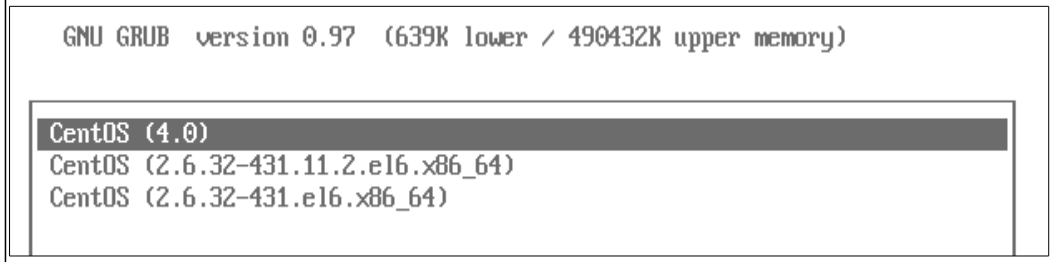
```
[root@centos65 grub]# cut -c1-70 menu.lst | tail -12
title CentOS (4.0)
root (hd0,0)
kernel /vmlinuz-4.0 ro root=/dev/mapper/VolGroup-lv_root rd_NO_LUKS
L
initrd /initramfs-4.0.img
title CentOS (2.6.32-431.11.2.el6.x86_64)
root (hd0,0)
kernel /vmlinuz-2.6.32-431.11.2.el6.x86_64 ro
root=/dev/mapper/VolGro
initrd /initramfs-2.6.32-431.11.2.el6.x86_64.img
title CentOS (2.6.32-431.el6.x86_64)
root (hd0,0)
kernel /vmlinuz-2.6.32-431.el6.x86_64 ro root=/dev/mapper/VolGroup-
lv
initrd /initramfs-2.6.32-431.el6.x86_64.img
[root@centos65 grub]#
```

1. Установите длительность периода ожидания перед загрузкой операционной системы, равную 30 секундам.

```
[root@centos65 grub]# vi menu.lst
[root@centos65 grub]# grep timeout /boot/grub/grub.conf
timeout=30
```

1. Перезагрузите компьютер и проверьте работоспособность нового станса.

```
[root@centos65 grub]# reboot
```



Выберите ваш станс и в том случае, если произойдет загрузка операционной системы, вы выполнили практическое задание корректно.

Глава 15. Инициализация системы и уровни исполнения

Во многих дистрибутивах Unix и Linux используются сценарии инициализации, которые предназначены для запуска системных демонов таким же образом, как это делалось в системе Unix System V. В данной главе приведены подробные пояснения относительно принципа работы этих сценариев.

Системы инициализации запускают демоны силами сценариев, причем каждый из сценариев осуществляет запуск одного демона, а каждый следующий сценарий ожидает завершения исполнения предыдущего сценария. Этот последовательный процесс запуска демонов является достаточно медленным и, несмотря на то, что медленная загрузка системы не является проблемой в случае серверов, время непрерывной работы которых измеряется в годах, в результате недавних масштабных проектов по развертыванию Linux на настольных компьютерах выяснилось, что процесс загрузки системы вызывает нарекания со стороны пользователей.

Для увеличения скорости загрузки Linux силами компании Canonical была разработана система инициализации upstart, которая была впервые использована в дистрибутиве Ubuntu. В операционной системе Solaris система инициализации init также

использовалась вплоть до версии Solaris 9, а для версии Solaris 10 силами компании Sun была разработана система инициализации Service Management Facility. Обе системы инициализации осуществляют запуск демонов в параллельном режиме и могут заменять сценарии инициализации SysV. Также ведется работа по реализации системы инициализации initng (init next generation).

В 2014 году ведущую роль начинает играть система инициализации systemd, которая после дистрибутива Fedora была внедрена в дистрибутивах RHEL7 и CentOS7, а также была выбрана как наиболее предпочтительная замена для системы инициализации init в дистрибутиве Debian. По этой причине в конце данной главы приведено краткое описание системы инициализации systemd.

15.1. Инициализация системы

15.1.1. Процесс с идентификатором 1

Системный загрузчик передает контроль над системой ядру ОС. После непродолжительного периода времени ядро ОС запускает демон системы инициализации. Этот демон системы инициализации (/sbin/init) является первым демоном, запущенным в рамках системы, поэтому соответствующий процесс получает идентификатор 1 (PID 1). Демон системы инициализации никогда не завершает свою работу.

15.1.2. Параметры конфигурации в файле /etc/inittab

После того, как исполняется бинарный файл /sbin/init, в первую очередь осуществляется чтение конфигурационного файла /etc/inittab. В данном файле демон будет искать значение переменной initdefault (равное 3 в примере ниже).

[paul@rhel4 ~]$ grep ^id /etc/inittab
id:3:initdefault:

15.1.3. Переменная initdefault

С помощью значения переменной initdefault указывается стандартный уровень исполнения (default runlevel). В некоторых дистрибутивах Linux в файле /etc/inittab приводится краткое описание уровней исполнения подобное приведенному ниже переведенному описанию из соответствующего файла дистрибутива Red Hat Enterprise Linux 4.

Стандартный уровень исполнения. Уровни исполнения, используемые в RHS:
0 - Отключение системы (НЕ УСТАНАВЛИВАЙТЕ данное значение initdefault)
1 - Однопользовательский режим
2 - Многопользовательский режим без поддержки NFS (Аналогичен уровню 3 в том случае, если не используется сетевое соединение)
3 - Полноценный многопользовательский режим
4 - Не используется
5 - X11
6 - Перезагрузка (НЕ УСТАНАВЛИВАЙТЕ данное значение initdefault)

Уровень исполнения 0 соответствует отключению системы. Уровень исполнения 1 используется для устранения неполадок, так как осуществить вход в систему может исключительно пользователь root, причем для входа в систему может использоваться исключительно консоль. Уровень исполнения 3 типичен для серверов, а уровень исполнения 5 — для настольных компьютеров (на которых вход в систему осуществляется в графическом режиме). За исключением уровней исполнения 0, 1 и 6, различные уровни исполнения могут отличаться в зависимости от дистрибутива. К примеру, в дистрибутиве Debian и производных дистрибутивах Linux на уровнях исполнения 2 и 5 имеется возможность входа в систему с использованием сетевого соединения и графического интерфейса. Исходя из этого, следует всегда сверяться с корректным описанием уровней исполнения вашей системы.

15.1.4. Сценарий sysinit

Сценарий /etc/rc.d/rc.sysinit

Следующая строка файла конфигурации /etc/inittab в дистрибутиве Red Hat и производных дистрибутивах выглядит следующим образом:

si::sysinit:/etc/rc.d/rc.sysinit

Эта запись означает, что независимо от выбранного уровня исполнения система инициализации будет исполнять сценарий /etc/rc.d/rc.sysinit. Этот сценарий осуществляет инициализацию аппаратного обеспечения, устанавливает значения некоторых основных переменных окружения, заполняет файл /etc/mtab в процессе монтирования файловых систем, подключает раздел подкачки, а также выполняет другие необходимые действия.

[paul@rhel ~]$ egrep -e"# Ini" -e"# Sta" -e"^# Che" /etc/rc.d/rc.sysinit
Check SELinux status (проверка состояния SELinux)
Initialize hardware (инициализация аппаратного обеспечения)
Start the graphical boot, if necessary; /usr may not be mounted yet... (начало загрузки в графическом режиме, если необходимо; файловая система /usr может быть еще не смонтирована...)
Initialiaze ACPI bits (инициализация ACPI)
Check filesystems (проверка файловых систем)
Start the graphical boot, if necessary and not done yet. (начало загрузки в графическом режиме, если необходимо и еще не выполнено)

Check to see if SELinux requires a relabel (проверка необходимости повторной расстановки меток SELinux)
Initialize pseudo-random number generator (инициализация генератора псевдослучайных чисел)
Start up swapping. (подключение раздела подкачки)
Initialize the serial ports. (инициализация последовательных портов)

Приведенная выше команда egrep может быть заменена на аналогичную приведенную ниже команду grep:

grep "^# \(Ini\|Sta\|Che\)".

Сценарий /etc/init.d/rcS

В дистрибутиве Debian после строки с переменной initdefault используется следующая строка:

si::sysinit:/etc/init.d/rcS

Сценарий /etc/init.d/rcS в дистрибутиве Debian будет выполняться всегда (независимо от выбранного уровня исполнения). Данный сценарий на самом деле осуществляет исполнение всех сценариев из директории /etc/rcS.d/ с сортировкой имен этих сценариев в алфавитном порядке.

root@barry:~# cat /etc/init.d/rcS
#! /bin/sh
#
rcS
#
Call all S??* scripts in /etc/rcS.d/ in numerical/alphabetical order (Исполнение сценариев S??* из директории /etc/rcS.d/ с сортировкой имен в числовом/алфавитном порядке)
#
exec /etc/init.d/rc S

15.1.5. Сценарии инициализации

Демон инициализации продолжит чтение конфигурационного файла /etc/inittab и перейдет к приведенной ниже секции в случае использования дистрибутива Debian Linux.

l0:0:wait:/etc/init.d/rc 0
l1:1:wait:/etc/init.d/rc 1
l2:2:wait:/etc/init.d/rc 2
l3:3:wait:/etc/init.d/rc 3
l4:4:wait:/etc/init.d/rc 4
l5:5:wait:/etc/init.d/rc 5
l6:6:wait:/etc/init.d/rc 6

В дистрибутиве Red Hat Enterprise Linux соответствующая секция является идентичной за исключением использования директории init.d вместо rc.d.

l0:0:wait:/etc/rc.d/rc 0
l1:1:wait:/etc/rc.d/rc 1
l2:2:wait:/etc/rc.d/rc 2
l3:3:wait:/etc/rc.d/rc 3
l4:4:wait:/etc/rc.d/rc 4
l5:5:wait:/etc/rc.d/rc 5
l6:6:wait:/etc/rc.d/rc 6

В обоих случаях демон инициализации должен осуществить запуск сценария инициализации с единственным параметром, который соответствует уровню исполнения. На самом деле поля в конфигурационном файле /etc/inittab разделены с помощью символов двоеточия. Второе поле описывает уровень исполнения на котором команда из данной строки должна быть выполнена. Таким образом, в обоих случаях происходит выполнение только одной из семи команд в зависимости от текущего уровня исполнения, установленного с помощью переменной initdefault.

15.1.6. Директории для хранения сценариев инициализации

Если вы рассмотрите содержимое одной из директорий /etc/rcX.d/, вы можете обнаружить множество сценариев (или ссылок на сценарии), имена которых начинаются либо с буквы K, либо с буквы S в верхнем регистре.

[root@RHEL52 rc3.d]# ls -l | tail -4
lrwxrwxrwx 1 root root 19 окт 11 2008 S98haldaemon → ../init.d/haldaemon
lrwxrwxrwx 1 root root 19 окт 11 2008 S99firstboot → ../init.d/firstboot
lrwxrwxrwx 1 root root 11 янв 21 04:16 S99local → ../rc.local
lrwxrwxrwx 1 root root 16 янв 21 04:17 S99smartd → ../init.d/smartd

Директории /etc/rcX.d/ содержат только ссылки на сценарии, расположенные в директории /etc/init.d/. Благодаря ссылкам появляется возможность использования сценариев с отличными именами. При переходе на новый уровень исполнения все сценарии, имена которых начинаются с буквы K или буквы S в верхнем регистре начинают исполняться после сортировки по именам в алфавитном порядке. Сценарии, имена которых начинаются с буквы K, будут исполняться в первую очередь с передачей единственного параметра stop. Остальные сценарии, имена которых начинаются с буквы S, будут исполняться с передачей единственного параметра start.

Все эти операции выполняются силами сценария /etc/rc.d/rc в дистрибутиве Red Hat и сценария /etc/init.d/rc в дистрибутиве Debian.

15.1.7. Демоны mingetty

Описание демонов mingetty в конфигурационном файле /etc/inittab

Практически в конце конфигурационного файла /etc/inittab

находится секция с описанием условий запуска и перезапуска нескольких демонов mingetty.

[root@RHEL4b ~]# grep getty /etc/inittab

Run gettys in standard runlevels (Запуск gettys на различных уровнях исполнения)

1:2345:respawn:/sbin/mingetty tty1
2:2345:respawn:/sbin/mingetty tty2
3:2345:respawn:/sbin/mingetty tty3
4:2345:respawn:/sbin/mingetty tty4
5:2345:respawn:/sbin/mingetty tty5
6:2345:respawn:/sbin/mingetty tty6

Демоны mingetty и исполняемый файл /bin/login

Демон /sbin/mingetty выводит сообщение в виртуальной консоли и позволяет вам ввести идентификатор пользователя. После этого он выполняет бинарный файл /bin/login с передачей введенного идентификатора пользователя. Программа /bin/login проверяет, присутствует ли информация о пользователе в файле /etc/passwd и запрашивает пароль (а также проверяет его корректность). В том случае, если пароль корректен, программа /bin/login передает управление командной оболочке, установленной в файле /etc/passwd.

Перезапуск демонов mingetty

Демон инициализации осуществляет запуск демонов mingetty и отслеживает их состояние вплоть до завершения их работы (это происходит тогда, когда пользователь прекращает работу с командной оболочкой и осуществляет выход из системы). Когда это случается, демон инициализации осуществляет повторный запуск новой копии демона mingetty. Исходя из этого, даже в том случае, если вы принудительно завершите работу демона mingetty, он будет автоматически перезапущен.

В приведенном ниже примере показано, как демон инициализации перезапускает демоны mingetty. Обратите внимание на идентификаторы двух последних процессов mingetty.

[root@RHEL52 ~]# ps -C mingetty
PID TTY TIME CMD
2407 tty1 00:00:00 mingetty
2408 tty2 00:00:00 mingetty
2409 tty3 00:00:00 mingetty
2410 tty4 00:00:00 mingetty
2411 tty5 00:00:00 mingetty
2412 tty6 00:00:00 mingetty

При использовании утилиты kill для завершения работы двух последних демонов mingetty демон инициализации будет обрабатывать сигналы завершения работы дочерних процессов и снова запускать их (с отличными идентификаторами процессов).

[root@RHEL52 ~]# kill 2411 2412
[root@RHEL52 ~]# ps -C mingetty
PID TTY TIME CMD
2407 tty1 00:00:00 mingetty
2408 tty2 00:00:00 mingetty
2409 tty3 00:00:00 mingetty
2410 tty4 00:00:00 mingetty
2821 tty5 00:00:00 mingetty
2824 tty6 00:00:00 mingetty

Деактивация демонов mingetty

Вы можете деактивировать демоны mingetty для определенного терминала, удалив номер уровня исполнения из второго поля соответствующей строки файла /etc/inittab. Не забудьте о необходимости сообщить демону инициализации об изменении его конфигурационного файла с помощью команды kill -1 1.

В примере ниже показана методика деактивации демонов mingetty для терминалов tty3 и tty6 на уровнях исполнения 4 и 5.

[root@RHEL52 ~]# grep getty /etc/inittab

Run gettys in standard runlevels (запуск gettys на различных уровнях исполнения)

1:2345:respawn:/sbin/mingetty tty1
2:2345:respawn:/sbin/mingetty tty2
3:23:respawn:/sbin/mingetty tty3
4:23:respawn:/sbin/mingetty tty4
5:23:respawn:/sbin/mingetty tty5
6:23:respawn:/sbin/mingetty tty6

15.2. Daemon или demon?

Демон (daemon) является процессом, который выполняется в фоновом режиме без связи с графическим интерфейсом или терминалом. Обычно демоны запускаются при загрузке системы и находятся в рабочем состоянии вплоть до момента завершения работы системы. В современной технической документации демоны чаще всего называются службами (services).

При разговоре о демонах Unix не следует путать слова daemons и demons. Evi Nemeth, соавтор книги «Unix System Administration Handbook» говорил о демонах следующее:

Многие уравнивают слово «daemon» со словом «demon», намекая на некоторую адскую связь между системами Unix и преисподней. Это утверждение свидетельствует о вопиющем непонимании. На самом деле, слово «daemon» является более старой формой слова «demon»; слово daemon не имеет определенного отношения к добру или злу, а вместо этого служит для описания характера или личностных качеств человека. Древнегреческая концепция «личного демона» была схожа с современной концепцией «ангела-хранителя»...

15.3. Запуск и остановка демонов

Сценарии, имена которых начинаются с букв K и S, являются ссылками на реальные сценарии из директории /etc/init.d/. Эти сценарии могут также использоваться в процессе функционирования системы для для запуска и остановки демонов (или служб). Большинство из них принимает следующие параметры: start, stop, restart, status.

К примеру, в данном случае мы осуществляем перезапуск демона samba.

root@laika:~# /etc/init.d/samba restart

- Stopping Samba daemons... [OK]
- Starting Samba daemons... [OK]

Вы можете добиться аналогичного результата в дистрибутиве RHEL/Fedora, использовав утилиту service.

[root@RHEL4b ~]# service smb restart
Shutting down SMB services: [OK]
Shutting down NMB services: [OK]
Starting SMB services: [OK]
Starting NMB services: [OK]

Также вы можете обратить внимание на утилиты chkconfig и update-rc.d.

15.4. Утилита chkconfig

Утилита chkconfig предназначена для избавления системных администраторов от необходимости вручную редактировать ссылки и сценарии в директориях /etc/init.d и /etc/rcX.d/.

15.4.1. Команда chkconfig --list

В данном случае мы используем утилиту chkconfig для вывода списка состояний службы на различных уровнях исполнения. Как вы видите, демон (или служба) crond активируется только на уровнях исполнения со 2 по 5.

[root@RHEL52 ~]# chkconfig --list crond
crond 0:off 1:off 2:on 3:on 4:on 5:on 6:off

Если вы сравните приведенный выше вывод с приведенным ниже результатом поиска ссылок на сценарии, вы можете обнаружить, что отключенное состояние (off) соответствует ссылке на сценарий с именем, начинающимся на K, а включенное состояние (on) — ссылкой на сценарий с именем, начинающимся на букву S.

[root@RHEL52 etc]# find ./rc?.d/ -name *crond -exec ls -l {} \;|cut -b40-
/rc6.d/K60crond → ../init.d/crond

15.4.2. Установка уровней исполнения для службы

Обратившись к приведенному ниже примеру, вы можете узнать о том, как использовать утилиту chkconfig для деактивации (или активации) службы на определенном уровне исполнения.

В этом примере показано, как деактивировать службу crond на уровне исполнения 3.

[root@RHEL52 ~]# chkconfig --level 3 crond off
[root@RHEL52 ~]# chkconfig --list crond
crond 0:off 1:off 2:on 3:off 4:on 5:on 6:off

А в этом примере показано, как активировать эту же службу crond на уровнях исполнения 3 и 4.

[root@RHEL52 ~]# chkconfig --level 34 crond on
[root@RHEL52 ~]# chkconfig --list crond
crond 0:off 1:off 2:on 3:on 4:on 5:on 6:off

15.4.3. Настройка утилиты chkconfig

Каждый сценарий в директории /etc/init.d/ может содержать дополнительные строки (и комментарий), сообщающие утилите chkconfig от том, что следует сделать со службой в той или иной ситуации. Строка, начинающаяся с # chkconfig: содержит список уровней исполнения, на которых служба должна запускаться (2345), за которым следуют значения приоритетов запуска (90) и остановки (60) службы.

[root@RHEL52 ~]# head -9 /etc/init.d/crond | tail -5
chkconfig: 2345 90 60
description: cron is a standard UNIX program that runs user-specified
programs at periodic scheduled times. vixie cron adds a
number of features to the basic UNIX cron, including better
security and more powerful configuration options.

15.4.4. Активация и деактивация служб

Службы могут быть активированы и деактивированы на всех уровнях исполнения с помощью единственной команды. Уровни исполнения 0, 1 и 6 всегда связаны с остановкой служб (или вызовом сценариев с передачей параметра stop) даже в том случае, если имя сценария начинается с буквы S в верхнем регистре.

[root@RHEL52 ~]# chkconfig crond off
[root@RHEL52 ~]# chkconfig --list crond
crond 0:off 1:off 2:off 3:off 4:off 5:off 6:off
[root@RHEL52 ~]# chkconfig crond on
[root@RHEL52 ~]# chkconfig --list crond
crond 0:off 1:off 2:on 3:on 4:on 5:on 6:off

15.5. Утилита update-rc.d

15.5.1. Об утилите update-rc.d

В дистрибутиве Debian эквивалентом утилиты chkconfig является утилита с именем update-rc.d. Эта утилита предназначена главным образом для использования в рамках сценариев, поэтому в том случае, если вы предпочитаете инструменты с графическим

интерфейсом, следует обратить внимание на утилиту bum.

В том случае, если в директории /etc/rcX.d/ присутствуют ссылки на сценарии, утилита update-rc.d не будет выполнять никаких действий. Описанное поведение утилиты объясняется необходимостью предотвращения перезаписи пост-установочными сценариями, использующими утилиту update-rc.d, изменений, внесенных системным администратором.

```
root@barry:~# update-rc.d cron remove
update-rc.d: /etc/init.d/cron exists during rc.d purge (use -f to force)
```

Как вы можете увидеть в следующем примере, условия запуска демона crond никоим образом не изменились.

```
root@barry:~# find /etc/rc?.d/ -name *cron -exec ls -l {} \;|cut -b44-
/etc/rc0.d/K11cron → ../init.d/cron
/etc/rc1.d/K11cron → ../init.d/cron
/etc/rc2.d/S89cron → ../init.d/cron
/etc/rc3.d/S89cron → ../init.d/cron
/etc/rc4.d/S89cron → ../init.d/cron
/etc/rc5.d/S89cron → ../init.d/cron
/etc/rc6.d/K11cron → ../init.d/cron
```

15.5.2. Удаление системной службы

В данном случае мы удалим упоминание о системной службе cron на всех уровнях исполнения. Помните о том, что корректный способ деактивации системной службы заключается в размещении в директориях всех уровней исполнения ссылки на сценарий с именем, начинающимся с буквы K.

```
root@barry:~# update-rc.d -f cron remove
Removing any system startup links for /etc/init.d/cron ...
/etc/rc0.d/K11cron
/etc/rc1.d/K11cron
/etc/rc2.d/S89cron
/etc/rc3.d/S89cron
/etc/rc4.d/S89cron
/etc/rc5.d/S89cron
/etc/rc6.d/K11cron
root@barry:~# find /etc/rc?.d/ -name *cron -exec ls -l {} \;|cut -b44-
root@barry:~#
```

15.5.3. Активация системной службы

В данном примере показана методика использования утилиты update-rc.d для активации системной службы на уровнях исполнения 2, 3, 4 и 5, а также деактивации системной службы на уровнях исполнения 0, 1 и 6.

```
root@barry:~# update-rc.d cron defaults
Adding system startup for /etc/init.d/cron ...
/etc/rc0.d/K20cron -> ../init.d/cron
/etc/rc1.d/K20cron -> ../init.d/cron
/etc/rc6.d/K20cron -> ../init.d/cron
/etc/rc2.d/S20cron -> ../init.d/cron
/etc/rc3.d/S20cron -> ../init.d/cron
/etc/rc4.d/S20cron -> ../init.d/cron
/etc/rc5.d/S20cron -> ../init.d/cron
```

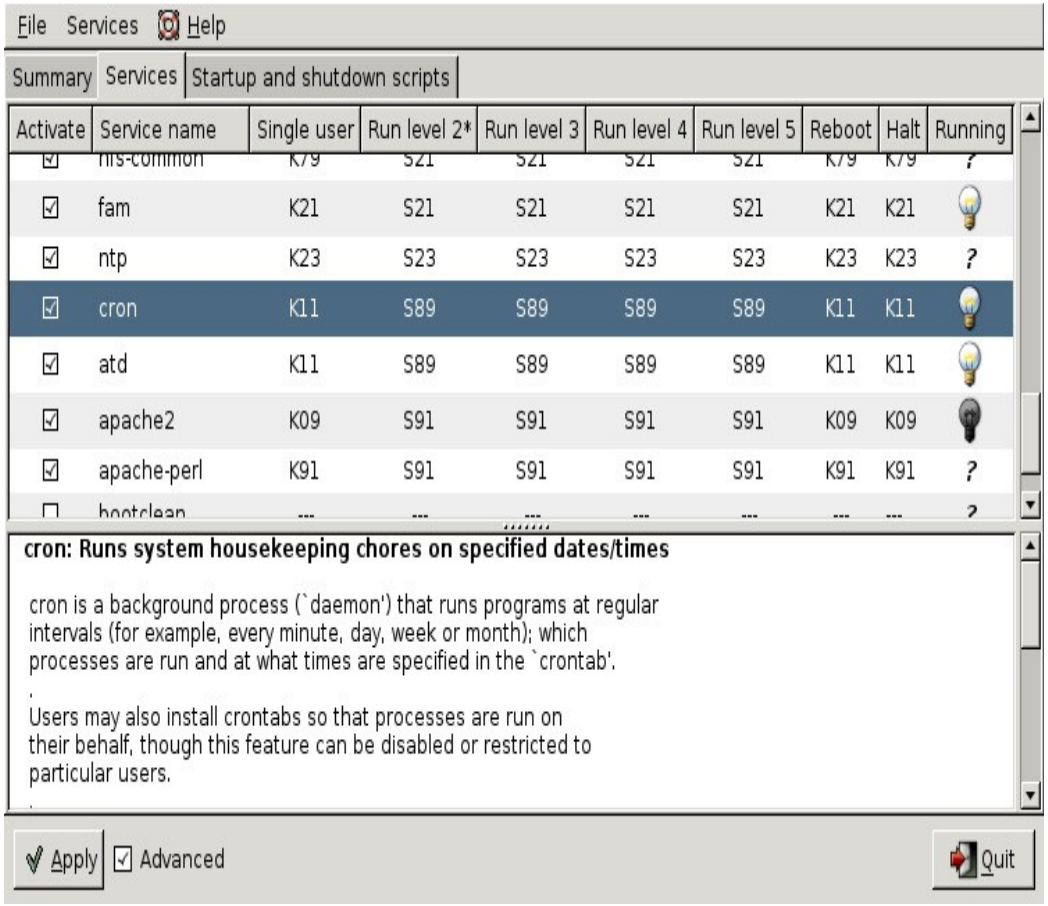
15.5.4. Нестандартная конфигурация системной службы

А в данном примере показана методика использования разработанной вами нестандартной конфигурации демона crond.

```
root@barry:~# update-rc.d -n cron start 11 2 3 4 5 . stop 89 0 1 6 .
Adding system startup for /etc/init.d/cron ...
/etc/rc0.d/K89cron -> ../init.d/cron
/etc/rc1.d/K89cron -> ../init.d/cron
/etc/rc6.d/K89cron -> ../init.d/cron
/etc/rc2.d/S11cron -> ../init.d/cron
/etc/rc3.d/S11cron -> ../init.d/cron
/etc/rc4.d/S11cron -> ../init.d/cron
/etc/rc5.d/S11cron -> ../init.d/cron
```

15.6. Утилита bum

На рисунке показано окно утилиты bum, работающей в расширенном режиме.



15.7. Уровни исполнения

15.7.1. Вывод информации об уровне исполнения

Вы можете узнать текущий уровень исполнения с помощью команд runlevel и who -r.

Команда runlevel является стандартной командой Linux и выводит информацию о предыдущем и текущем уровнях исполнения. В том случае, если предыдущего уровня исполнения не было, в качестве номера будет выведена буква N.

[root@RHEL4b ~]# runlevel
N 3

История появления команды who -r связана с системами Unix семидесятых годов прошлого века, но команда все еще работоспособна в современных системах Linux.

[root@RHEL4b ~]# who -r
run-level 3 Июл 28 09:15 last=S

15.7.2. Изменение уровня исполнения

Вы можете перейти на другой уровень исполнения с помощью команды telinit. В Linux файл /sbin/telinit обычно является ссылкой (жесткой) на бинарный файл /sbin/init.

В данном примере показана методика перехода с уровня исполнения 2 на уровень исполнения 3 без перезагрузки.

root@barry:~# runlevel
N 2
root@barry:~# init 3
root@barry:~# runlevel
2 3

15.7.3. Утилита /sbin/shutdown

Утилита shutdown используется для корректного завершения работы системы.

Стандартными аргументами, передаваемыми утилите shutdown, являются -a, -t. -h и -r.

Аргумент -a принуждает утилиту /sbin/shutdown использовать файл конфигурации /etc/shutdown.allow. Параметр -t используется для установки количества секунд, которые должны пройти между отправкой процессам сигналов TERM и KILL. Параметр -h позволяет завершить работу системы вместо перехода к уровню исполнения 1. Параметр -r сообщает утилите /sbin/shutdown о необходимости осуществления перезагрузки после завершения работы системы.

В данном примере показана методика использования утилиты shutdown с пятисекундным интервалом между отправкой сигналов TERM и KILL.

root@barry:~# shutdown -t5 -h now

Аргумент now является меткой времени. В качестве этого аргумента может использоваться запись +м, где вместо символа м должно указываться количество минут по истечении которых работа системы должна быть завершена (очевидно, что аргумент now является псевдонимом аргумента +0). Утилита также примет метку времени в формате чч:мм вместо сдвига по времени в формате +м.

15.7.4. Утилиты halt, reboot и poweroff

Бинарный файл, реализующий функции утилиты /sbin/reboot, является тем же самым бинарным файлом, который используется и для реализации функций утилит /sbin/halt и /sbin/poweroff. В зависимости от используемой команды он может вести себя по-разному.

При работе с уровнями исполнения 0 или 6 команды halt, reboot и poweroff сообщат ядру ОС о необходимости осуществления отключения, перезагрузки или завершения работы системы.

Если же уровень исполнения не является уровнем 0 или 6, при вводе команды reboot в случае использования учетной записи пользователя root на самом деле произойдет вызов утилиты shutdown с параметром -r, а при вводе команды poweroff — отключение питания в процессе отключения системы.

15.7.5. Файл журнала /var/log/wtmp

При использовании команд halt, reboot и poweroff происходит запись данных в файл журнала /var/log/wtmp. Для просмотра файла журнала /var/log/wtmp удобно использовать утилиту last.

```
[root@RHEL52 ~]# last | grep reboot
reboot system boot 2.6.18-128.el5 Fri May 29 11:44 (192+05:01)
reboot system boot 2.6.18-128.el5 Wed May 27 12:10 (06:49)
reboot system boot 2.6.18-128.el5 Mon May 25 19:34 (1+15:59)
reboot system boot 2.6.18-128.el5 Mon Feb 9 13:20 (106+21:13)
```

15.7.6. Сочетание клавиш Ctrl+Alt+Del

После того, как сценарий rc завершит исполнение всех описанных сценариев для управления службами, демон инициализации init продолжит читать файл конфигурации /etc/inittab. В следующей строке описано поведение демона инициализации в том случае, если пользователь нажмет клавиши Alt+Ctrl+Del на клавиатуре.

Ниже приведена команда, используемая в дистрибутиве Debian 4.0.

```
root@barry:~# grep -i ctrl /etc/inittab
# What to do when CTRL-ALT-DEL is pressed. (Что делать при нажатии сочетания клавиш CTRL-ALT-DEL.)
ca:12345:ctrlaltdel:/sbin/shutdown -t1 -a -r now
```

Данная команда очень похожа на стандартную команду, используемую в дистрибутиве Red Hat Enterprise Linux 5.2.

```
[root@RHEL52 ~]# grep -i ctrl /etc/inittab
# Trap CTRL-ALT-DELETE (Перехват сочетания клавиш CTRL-ALT-DELETE)
ca::ctrlaltdel:/sbin/shutdown -t3 -r now
```

Одно очевидное отличие между данным командами заключается в том, что в дистрибутиве Debian принудительно используется файл конфигурации /etc/shutdown.allow, в то время, как в дистрибутиве Red Hat любому пользователю, который может нажать сочетание клавиш Ctrl+Alt+Delete, разрешается использовать утилиту shutdown.

15.7.7. ИБП и отключение питания компьютера

```
[root@RHEL52 ~]# grep ^p /etc/inittab
pf::powerfail:/sbin/shutdown -f -h +2 "Power Failure; System Shutting Down" («Питание компьютера отключено; Работа системы завершается»)
pr:12345:powerokwait:/sbin/shutdown -c "Power Restored; Shutdown Cancelled" («Питание компьютера восстановлено; Процесс завершения работы системы прерван»)
```

Демон инициализации будет читать из файла конфигурации команды, которые должны исполняться в случае отключения питания компьютера (powerfailure), восстановления питания компьютера (powerok) и нажатия сочетания клавиш Ctrl+Alt+Del. Процесс демона инициализации init никогда не будет прекращать отслеживание сбоев по питанию компьютера, а также нажатия упомянутого сочетания трех клавиш на клавиатуре.

```
root@barry:~# grep ^p /etc/inittab
pf::powerwait:/etc/init.d/powerfail start
pn::powerfailnow:/etc/init.d/powerfail now
po::powerokwait:/etc/init.d/powerfail stop
```

15.8. Менеджер инициализации systemd

Можно предположить с высокой вероятностью, что менеджер инициализации systemd заменит все стандартные функции компонентов систем инициализации init/runlevel/rc. Разработчики дистрибутивов Red Hat и Debian в 2014 году приняли решение о том, что systemd заменит стандартные системы инициализации в следующих выпусках их дистрибутивов (RHEL/CentOS7 и Debian 8).

В примере ниже показан процесс systemd с идентификатором 1, исполняющийся в ходе работы с дистрибутивом RHEL7.

```
[root@rhel7 ~]# ps fax | grep systemd | cut -c1-76
1 ? Ss 0:01 /usr/lib/systemd/systemd --switched-root --system
505 ? Ss 0:00 /usr/lib/systemd/systemd-journald
545 ? Ss 0:00 /usr/lib/systemd/systemd-udevd
670 ? Ss 0:00 /usr/lib/systemd/systemd-logind
677 ? Ssl 0:00 /bin/dbus-daemon --system --address=systemd: --no
2662 pts/1 S+ 0:00 \_ grep --color=auto systemd
[root@rhel7 ~]#
```

В дистрибутиве Debian 8 (еще не выпущенном в сентябре 2014 года) используются лишь некоторые компоненты системы инициализации systemd, но при этом процессом с идентификатором 1 все также является процесс init.

```
root@debian8:~# ps fax | grep systemd
2042 ? S 0:00 /sbin/cgmanager --daemon -m name=systemd
10127 pts/4 S+ 0:00 | \_ grep systemd
2777 ? S 0:00 /lib/systemd/systemd-logind
root@debian8:~#
```

15.8.1. Цели systemd

Первой командой, которую нужно запомнить, является команда systemctl list-units --type=target (или ее более короткий вариант systemctl -t target). Она позволяет вывести список различных целей менеджера инициализации, присутствующих в системе.

[root@rhel7 ~]# systemctl list-units --type=target
UNIT LOAD ACTIVE SUB DESCRIPTION
basic.target loaded active active Basic System
cryptsetup.target loaded active active Encrypted Volumes
getty.target loaded active active Login Prompts
graphical.target loaded active active Graphical Interface
local-fs-pre.target loaded active active Local File Systems (Pre)
local-fs.target loaded active active Local File Systems
multi-user.target loaded active active Multi-User System
network.target loaded active active Network
nfs.target loaded active active Network File System Server
paths.target loaded active active Paths
remote-fs.target loaded active active Remote File Systems
slices.target loaded active active Slices
sockets.target loaded active active Sockets
swap.target loaded active active Swap
sysinit.target loaded active active System Initialization
timers.target loaded active active Timers
LOAD = Reflects whether the unit definition was properly loaded. (Отражает корректность загрузки описания элемента.)
ACTIVE = The high-level unit activation state, i.e. generalization of SUB. (Состояние активации элемента верхнего уровня, т.е., обобщение данных из столбца SUB.)
SUB = The low-level unit activation state, values depend on unit type. (Состояние активации элемента нижнего уровня, значения зависят от типа элемента.)
16 loaded units listed. Pass --all to see loaded but inactive units, too. (выведена информация о 16 элементах. Передайте параметр --all для вывода информации также и о неактивных элементах.)
To show all installed unit files use systemctl list-unit-files. (Для показа всех установленных файлов элементов используйте команду systemctl list-unit-files.)
[root@rhel7 ~]#
Цели введены в качестве замены уровней исполнения и описывают определенные точки, которые должны быть достигнуты при загрузке системы. К примеру, цель graphical.target достигается тогда, когда вы получаете возможность работы с графическим интерфейсом, а цель nfs.target требует наличия функционирующего сервера nfs.
Для перехода к цели (к примеру, к цели multi-user.target) мы должны использовать команду systemctl isolate (вместо эквивалентной команды init 3, предназначенной для изменения уровня исполнения).
[root@rhel7 ~]# ps fax | wc -l
169
[root@rhel7 ~]# systemctl isolate multi-user.target
[root@rhel7 ~]# ps fax | wc -l
129
[root@rhel7 ~]#
Для изменения цели, установленной по умолчанию, мы должны снова использовать команду systemctl (вместо редактирования файла /etc/inittab).
[root@rhel7 ~]# systemctl enable multi-user.target --force
rm /etc/systemd/system/default.target
ln -s /usr/lib/systemd/system/multi-user.target '/etc/systemd/system/default\
target'
[root@rhel7 ~]#
Данная команда удаляет файл /etc/systemd/system/default.target и заменяет его на символьную ссылку на файл цели multi-user.target.

15.8.2. Зависимости systemd

Зависимости больше не описываются с помощью последовательности сценариев, отсортированной в алфавитном порядке, а задаются с помощью ссылок на файлы служб, расположенные в директории /etc/systemd/system/. К примеру, ниже приведен список файлов служб, которые должны быть активированы для достижения цели multi-user.target в дистрибутиве Red Hat Enterprise Linux 7.
[root@rhel7 ~]# ls /etc/systemd/system/multi-user.target.wants/
abrt-ccpp.service hypervkvpd.service postfix.service
abrtd.service hypervvssd.service remote-fs.target
abrt-oops.service irqbalance.service rhsmcertd.service
abrt-vmcore.service ksm.service rngd.service
abrt-xorg.service ksmtuned.service rpcbind.service
atd.service libstoragemgmt.service rsyslog.service
auditd.service libvirtd.service smartd.service
avahi-daemon.service mdmonitor.service sshd.service
chronyd.service ModemManager.service sysstat.service
crond.service NetworkManager.service tuned.service
cups.path nfs.target vmtoolsd.service
[root@rhel7 ~]#
Дистрибутив Debian 8 пока не полностью мигрировал на systemd.
root@debian8:~# ls /etc/systemd/system/multi-user.target.wants/
anacron.service binfmt-support.service pppd-dns.service ssh.service
atd.service fancontrol.service remote-fs.target
avahi-daemon.service lm-sensors.service rsyslog.service
Обычные сценарии инициализации из директории rc заменены на файлы служб. Используйте команду systemctl list-units -t service

--all (или systemctl -at service) для получения списка всех файлов служб в вашей системе.

```
[root@rhel7 ~]# systemctl -at service | head -5 | column -t | cut -c1-78
UNIT LOAD ACTIVE SUB DESCRIPTION
abrt-ccpp.service loaded active exited Install ABRT coredump
abrt-oops.service loaded active running ABRT kernel log
abrt-vmcore.service loaded inactive dead Harvest vmcores for
abrt-xorg.service loaded active running ABRT Xorg log
[root@rhel7 ~]#
```

А это пример вывода информации о состоянии службы sshd.

```
[root@rhel7 ~]# systemctl status sshd.service
sshd.service - OpenSSH server daemon
Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled)
Active: active (running) since Wed 2014-09-10 13:42:21 CEST; 55min ago
Main PID: 1400 (sshd)
CGroup: /system.slice/sshd.service
--1400 /usr/sbin/sshd -D
Sep 10 13:42:21 rhel7 systemd[1]: Started OpenSSH server daemon.
Sep 10 13:42:21 rhel7 sshd[1400]: Server listening on 0.0.0.0 port 22.
Sep 10 13:42:21 rhel7 sshd[1400]: Server listening on
    port 22.
[root@rhel7 ~]#
```

15.8.3. Службы systemd

Утилиты chkconfig и service признаны устаревшими. Они заменены на утилиту systemctl.

В данном примере показан новый способ запуска и остановки системной службы.

```
[root@rhel7 ~]# systemctl start crond.service
[root@rhel7 ~]# systemctl show crond.service | grep State
LoadState=loaded
ActiveState=active
SubState=running
UnitFileState=enabled
[root@rhel7 ~]# systemctl stop crond.service
[root@rhel7 ~]# systemctl show crond.service | grep State
LoadState=loaded
ActiveState=inactive
SubState=dead
UnitFileState=enabled
[root@rhel7 ~]#
```

А это новый способ остановки и деактивации системной службы.

```
[root@rhel7 ~]# systemctl stop crond.service
[root@rhel7 ~]# systemctl disable crond.service
rm /etc/systemd/system/multi-user.target.wants/crond.service
[root@rhel7 ~]# systemctl show crond.service | grep State
LoadState=loaded
ActiveState=inactive
SubState=dead
UnitFileState=disabled
[root@rhel7 ~]#
```

В данном примере показан способ повторной активации и запуска системной службы.

```
[root@rhel7 ~]# systemctl enable crond.service
ln -s /usr/lib/systemd/system/crond.service '/etc/systemd/system/multi-user.\
target.wants/crond.service'
[root@rhel7 ~]# systemctl start crond.service
[root@rhel7 ~]# systemctl show crond.service | grep State
LoadState=loaded
ActiveState=active
SubState=running
UnitFileState=enabled
[root@rhel7 ~]#
```

15.8.4. Отправка сигналов при работе с systemd

Вы можете использовать менеджер инициализации systemd для прекращения работы проблемных служб.

```
[root@rhel7 ~]# systemctl show crond.service | grep State
LoadState=loaded
ActiveState=active
SubState=running
UnitFileState=enabled
[root@rhel7 ~]# systemctl kill -s SIGKILL crond.service
[root@rhel7 ~]# systemctl show crond.service | grep State
LoadState=loaded
ActiveState=failed
SubState=failed
UnitFileState=enabled
[root@rhel7 ~]#
```

15.8.5. Завершение работы системы средствами systemd

Команды poweroff, halt и reboot считаются устаревшими, поэтому на данный момент при их использовании происходит вызов утилиты systemctl. В приведенной ниже таблице в левом столбце перечислены устаревшие команды, а в правом — эквивалентные команды на основе утилит из комплекта поставки systemd.

Таблица 15.1. Управление питанием системы средствами systemd

Устаревшая команда	Команда в случае использования systemd
poweroff	systemctl poweroff
reboot	systemctl reboot
halt	systemctl halt
pm-suspend	systemctl suspend
pm-hibernate	systemctl hibernate

15.8.6. Удаленное использование systemd

Утилита systemctl имеет встроенный механизм для удаленного управления, который может использоваться в том случае, если в удаленной системе функционирует сервер ssh.

В данном примере показана методика использования утилиты systemctl для проверки состояния службы на удаленном сервере, работающем под управлением дистрибутива RHEL.

```
[root@rhel7 ~]# systemctl -H root@192.168.1.65 status sshd
root@192.168.1.65's password:
sshd.service - OpenSSH server daemon
Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled)
Active: active (running) since Thu 2014-09-11 13:04:10 CEST; 16min ago
Process: 1328 ExecStartPre=/usr/sbin/sshd-keygen (code=exited, status=0/SUCCE\
SS)
Main PID: 1363 (sshd)
CGroup: /system.slice/sshd.service
[root@rhel7 ~]#
```

15.8.7. Комплект поставки systemd включает большее количество утилит

Существуют и другие инструменты...

```
systemd-analyze systemd-loginctl
systemd-ask-password systemd-machine-id-setup
systemd-cat systemd-notify
systemd-cgls systemd-nspawn
systemd-cgtop systemd-run
systemd-coredumpctl systemd-stdio-bridge
systemd-delta systemd-sysv-convert
systemd-detect-virt systemd-tmpfiles
systemd-inhibit systemd-tty-ask-password-agent
```

Например, с помощью команды systemd-analyze blame можно получить список всех служб с временем загрузки каждой из них.

```
[root@rhel7 ~]# systemd-analyze blame | head
1.977s firewalld.service
1.096s tuned.service
993ms postfix.service
939ms iprinit.service
925ms vboxadd-x11.service
880ms firstboot-graphical.service
839ms accounts-daemon.service
829ms network.service
822ms iprupdate.service
795ms boot.mount
[root@rhel7 ~]#
```

15.9. Практическое задание: системы инициализации

1. Измените конфигурационный файл /etc/inittab таким образом, чтобы производился перезапуск только двух демонов mingetty. Завершите работу остальных демонов mingetty и удостоверьтесь в том, что они не были перезапущены.
2. Запустите дистрибутив Red Hat Enterprise Linux в виртуальной машине. Перейдите на уровень исполнения 5, выполните команду, позволяющую вывести информацию о текущем и предыдущем уровне исполнения, после чего перейдите на уровень исполнения 3.
3. Выясните, выполняет ли сценарий sysinit установку или изменение значения переменной окружения PATH на ваших компьютерах?
4. Выведите список всех сценариев из директории init.d, которые запускаются при достижении уровня исполнения 2.
5. Разработайте сценарий, который ведет себя аналогично сценарию для управления демоном из директории /etc/init.d/. Он должен содержать условный переход для обработки аргументов start/stop/restart и status. Протестируйте ваш сценарий!
6. Используйте утилиту chkconfig для того, чтобы ваш сценарий запускал демон при достижении уровней исполнения 3, 4 и 5 и останавливал его при достижении любого другого уровня исполнения.

15.10. Корректная процедура выполнения практического задания: системы инициализации

1. Измените конфигурационный файл /etc/inittab таким образом, чтобы производился перезапуск только двух демонов mingetty. Завершите работу остальных демонов mingetty и удостоверьтесь в том, что они не были

перезапущены.

Завершение работы демонов mingetty приведет к их перезапуску средствами демона инициализации init. Вы можете отредактировать файл /etc/inittab так, как показано ниже. Не забудьте также выполнить команду kill -1 1.

```
[root@RHEL5 ~]# grep tty /etc/inittab
# Run gettys in standard runlevels
1:2345:respawn:/sbin/mingetty tty1
2:2345:respawn:/sbin/mingetty tty2
3:2:respawn:/sbin/mingetty tty3
4:2:respawn:/sbin/mingetty tty4
5:2:respawn:/sbin/mingetty tty5
6:2:respawn:/sbin/mingetty tty6
[root@RHEL5 ~]#
```

1. Запустите дистрибутив Red Hat Enterprise Linux в виртуальной машине. Перейдите на уровень исполнения 5, выполните команду, позволяющую вывести информацию о текущем и предыдущем уровне исполнения, после чего перейдите на уровень исполнения 3.

init 5 (изменение уровня исполнения можно отследить по выводу в консоли)

runlevel

init 3 (и снова вы можете отследить изменение уровня исполнения по выводу в консоли)

1. Выясните, выполняет ли сценарий sysinit установку или изменение значения переменной окружения PATH на ваших компьютерах?

В дистрибутиве Red Hat следует использовать утилиту grep для поиска обращений к переменной окружения PATH в файле сценария /etc/rc.sysinit, а в дистрибутиве Debian следует осуществить поиск обращений к данной переменной как в файле /etc/rc.local, так и в файле /etc/init/rc.local. С высокой вероятностью вы можете ответить «нет» на поставленный вопрос, хотя в дистрибутиве RHEL5 сценарий rc.sysinit и устанавливает значение переменной окружения HOSTNAME.

```
[root@RHEL5 etc]# grep HOSTNAME rc.sysinit
```

1. Выведите список всех сценариев из директории init.d, которые запускаются при достижении уровня исполнения 2.

```
root@RHEL5 ~# chkconfig --list | grep '2:on'
```

2. Разработайте сценарий, который ведет себя аналогично сценарию для управления демоном из директории /etc/init.d/. Он должен содержать условный переход для обработки аргументов start/stop/restart и status. Протестируйте ваш сценарий!

Сценарий должен выглядеть аналогично приведенному ниже сценарию.

```
#!/bin/bash
#
# chkconfig: 345 99 01
# description: демонстрационный сценарий pold
#
# /etc/init.d/pold
#
case "$1" in
start)
echo -n "Запуск pold..."
sleep 1;
touch /var/lock/subsys/pold
echo "выполнено."
echo демон pold запущен >> /var/log/messages
    stop)
echo -n "Остановка pold..."
sleep 1;
rm -rf /var/lock/subsys/pold
echo "выполнено."
echo демон pold остановлен >> /var/log/messages
    *)
echo "Использование: /etc/init.d/pold {start|stop}"
exit 1
    esac
exit 0
```

Команда touch /var/lock/subsys/pold необходима и должна осуществлять создание файла с именем, идентичным имени сценария в том случае, если вы хотите задействовать сценарий в последовательности прекращения работы демонов (для этого также необходимо создать ссылку на сценарий с именем K01pold).

1. Используйте утилиту chkconfig для того, чтобы ваш сценарий запускал демон при достижении уровней исполнения 3, 4 и 5 и останавливал его при достижении любого другого уровня исполнения.

```
chkconfig --add pold
```

Приведенная выше команда будет рабоать только в том случае, если в сценарии pold присутствуют строки # chkconfig: и # description:.

Глава 16. Планирование исполнения задач

Администраторы систем Linux используют демон atd для

планирования однократного исполнения задач. Многократное исполнение задач удобнее планировать с помощью демона crond. В двух следующих разделах мы обсудим оба упомянутых инструмента.

16.1. Планирование однократного исполнения задач с помощью демона atd

16.1.1. Утилита at

Простое планирование исполнения задачи может осуществляться с помощью утилиты at. В данном примере показана методика планирования исполнения команды date в 22:01 и команды sleep в 22:03.

```
root@laika:~# at 22:01
at> date
at> <EOT>
job 1 at Wed Aug 1 22:01:00 2007
root@laika:~# at 22:03
at> sleep 10
at> <EOT>
job 2 at Wed Aug 1 22:03:00 2007
root@laika:~#
```

В реальной жизни, надеюсь, вы будете планировать исполнение более полезных команд :-)

16.1.2. Утилита atq

Проверка установленного для исполнения задач времени может осуществляться достаточно просто с помощью команды atq или at -l.

```
root@laika:~# atq
1 Wed Aug 1 22:01:00 2007 a root
2 Wed Aug 1 22:03:00 2007 a root
root@laika:~# at -l
1 Wed Aug 1 22:01:00 2007 a root
2 Wed Aug 1 22:03:00 2007 a root
root@laika:~#
```

Утилита at принимает в качестве меток времени такие английские слова, как tomorrow и teatime, которые позволяют запланировать исполнение команды на следующий день и на четыре часа вечера соответственно.

```
root@laika:~# at 10:05 tomorrow
at> sleep 100
at> <EOT>
job 5 at Thu Aug 2 10:05:00 2007
root@laika:~# at teatime tomorrow
at> tea
at> <EOT>
job 6 at Thu Aug 2 16:00:00 2007
root@laika:~# atq
6 Thu Aug 2 16:00:00 2007 a root
5 Thu Aug 2 10:05:00 2007 a root
root@laika:~#
```

16.1.3. Утилита atrm

Запланированные к исполнению задачи могут быть удалены из очереди с помощью утилиты atrm.

```
root@laika:~# atq
6 Thu Aug 2 16:00:00 2007 a root
5 Thu Aug 2 10:05:00 2007 a root
root@laika:~# atrm 5
root@laika:~# atq
6 Thu Aug 2 16:00:00 2007 a root
root@laika:~#
```

16.1.4. Файлы конфигурации at.allow и at.deny

Также вы можете использовать файлы конфигурации /etc/at.allow и /etc/at.deny для четкого указания пользователей, которые могут или не могут планировать исполнение задач с помощью демона atd.

Файл /etc/at.allow может содержать список пользователей, которым позволено осуществлять планирование исполнения задач с помощью демона atd. В том случае, если файла /etc/at.allow не существует, демон atd может использоваться любым пользователем, имя которого не внесено в файл /etc/at.deny.

Если же не существует ни одного из этих файлов, то демон atd по прямому назначению может использовать любой пользователь.

16.2. Планирование многократного исполнения задач с помощью демона crond

16.2.1. Файл конфигурации crontab

Команда crontab(1) может использоваться для модификации файла конфигурации crontab(5). Каждый пользователь может использовать свой файл конфигурации crontab для планирования исполнения задач в определенное время. Это время может быть задано с помощью значений в пяти полях в следующем порядке: минуты, часы, день месяца, месяц и день недели. В том случае, если поле содержит символ звездочки (*), считается, что в этом поле записаны все доступные значения.

Приведенная в примере ниже запись означает следующее: необходимо исполнять сценарий с именем script42 в два часа и восемь минут дня каждый месяц и каждый день недели.

```
8 14 * * * script42
```

Исполнение сценария с именем script8472 каждый месяц первого числа в 25 минут поле полуночи.

25 0 1 * * script847

Исполнение сценария с именем script33 через каждые две минуты в воскресенье (воскресенью соответствует как значение 0, так и значение 7).

*/2 * * * 0

Вместо значений в этих пяти полях вы также можете использовать одну из следующих директив: @reboot (перезагрузка), @yearly или @annually (ежегодно), @monthly (ежемесячно), @weekly (еженедельно), @daily или @midnight (ежедневно), @hourly (ежечасно).

16.2.2. Команда crontab

Пользователи не должны непосредственно редактировать файл crontab; вместо этого они должны использовать команду crontab -e, которая приведет к открытию текстового редактора, путь к которому задан с помощью переменной окружения EDITOR или VISUAL. Пользователи могут выводить содержимое своих таблиц задач демона crond с помощью команды crontab -l.

16.2.3. Файлы конфигурации cron.allow и cron.deny

Демон crond читает таблицы задач, принимая во внимание содержимое файлов конфигурации /etc/cron.allow и /etc/cron.deny.

Эти файлы оказывают точно такое же влияние на процесс планирования задач, как и файлы at.allow и at.deny. В том случае, если файл cron.allow существует, ваше имя пользователя должно быть занесено в него, ведь в противном случае вы не сможете воспользоваться демоном crond. В том же случае, если файла cron.allow не существует, вашего имени пользователя не должно быть в файле cron.deny в том случае, если вы планируете пользоваться демоном crond.

16.2.4. Файл конфигурации /etc/crontab

Файл конфигурации /etc/crontab содержит информацию о задачах, которые должны исполняться ежечасно/ежедневно/еженедельно/ежегодно. Его содержимое выглядит аналогично содержимому, представленному в примере ниже.

```
SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
20 3 * * * root run-parts --report /etc/cron.daily
40 3 * * 7 root run-parts --report /etc/cron.weekly
55 3 1 * * root run-parts --report /etc/cron.monthly
```

16.2.5. Директории /etc/cron.*

Директории, перечисленные в следующем примере, содержат описания задач, которые должны исполняться в моменты времени, описанные в файле конфигурации /etc/crontab. Директория /etc/cron.d предназначена для специальных случаев, когда при планировании исполнения задач требуется более гибкое описание периодов времени, чем ежечасно/ежедневно/еженедельно/ежемесячно.

```
paul@laika:~$ ls -ld /etc/cron.*
drwxr-xr-x 2 root root 4096 2008-04-11 09:14 /etc/cron.d
drwxr-xr-x 2 root root 4096 2008-04-19 15:04 /etc/cron.daily
drwxr-xr-x 2 root root 4096 2008-04-11 09:14 /etc/cron.hourly
drwxr-xr-x 2 root root 4096 2008-04-11 09:14 /etc/cron.monthly
drwxr-xr-x 2 root root 4096 2008-04-11 09:14 /etc/cron.weekly
```

16.2.6. Файл /etc/anachrontab

Обратите внимание на то, что в дистрибутиве Red Hat для планирования ежедневного, еженедельного и ежемесячного исполнения задач с помощью демона crond используется файл anacrontab.

```
root@rhel65:/etc# cat anacrontab
# /etc/anacrontab: configuration file for anacron
# See anacron(8) and anacrontab(5) for details.
SHELL=/bin/sh
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
# the maximal random delay added to the base delay of the jobs
RANDOM_DELAY=45
# the jobs will be started during the following hours only
START_HOURS_RANGE=3-22
#period in days delay in minutes job-identifier command
1 5 cron.daily nice run-parts /etc/cron.daily
7 25 cron.weekly nice run-parts /etc/cron.weekly
@monthly 45 cron.monthly nice run-parts /etc/cron.monthly
root@rhel65:/etc#
```

16.3. Практическое задание: планирование исполнения задач

1. Запланируйте исполнение двух задач с помощью утилиты at, выведите очередь задач демона atd и удалите задачу из нее.
2. Работая с учетной записью обычного пользователя, используйте команду crontab -e для планирования исполнения сценария через каждые четыре минуты.
3. Работая с учетной записью пользователя root, выведите содержимое файла crontab, принадлежащего обычному пользователю.
4. Снова перейдя к использованию учетной записи обычного пользователя, удалите ваш файл crontab.

5. Рассмотрите используемые демоном crond файлы и поддиректории в директории /etc и попытайтесь разобраться с их назначением. Какое действие выполняется с помощью команды run-parts?

16.4. Корректная процедура выполнения практического задания: планирование исполнения задач

6. Запланируйте исполнение двух задач с помощью утилиты at, выведите очередь задач демона atd и удалите задачу из нее.

```
root@rhel55 ~# at 9pm today
at> echo пора идти спать >> /root/todo.txt
at> <EOT>
job 1 at 2010-11-14 21:00
root@rhel55 ~# at 17h31 today
at> echo время для ланча >> /root/todo.txt
at> <EOT>
job 2 at 2010-11-14 17:31
root@rhel55 ~# atq
2 2010-11-14 17:31 a root
1 2010-11-14 21:00 a root
root@rhel55 ~# atrm 1
root@rhel55 ~# atq
2 2010-11-14 17:31 a root
root@rhel55 ~# date
Вс ноя 14 17:31:01 CET 2010
root@rhel55 ~# cat /root/todo.txt
время для ланча
```

1. Работая с учетной записью обычного пользователя, используйте команду crontab -e для планирования исполнения сценария через каждые четыре минуты.

```
paul@rhel55 ~$ crontab -e
no crontab for paul - using an empty one
crontab: installing new crontab
```

1. Работая с учетной записью пользователя root, выведите содержимое файла crontab, принадлежащего обычному пользователю.

```
root@rhel55 ~# crontab -l -u paul
*/4 * * * * echo date >> /home/paul/crontest.txt
```

1. Снова перейдя к использованию учетной записи обычного пользователя, удалите ваш файл crontab.

```
paul@rhel55 ~$ crontab -r
paul@rhel55 ~$ crontab -l
no crontab for paul
```

1. Рассмотрите используемые демоном crond файлы и поддиректории в директории /etc и попытайтесь разобраться с их назначением. Какое действие выполняется с помощью команды run-parts?

Команда run-parts позволяет выполнить сценарий в заданной директории.

Глава 17. Журналирование событий

В данной главе обсуждаются три отдельных вопроса.

Во-первых, рассматривается механизм журналирования входов в систему: обсуждаются способы получения информации о том, кто, когда и с какого узла осуществил вход в систему. Также рассматриваются способы получения информации о том, кому не удалось осуществить вход в систему, кто не смог воспользоваться утилитой su или ssh.

Во-вторых, рассматривается процесс настройки демона syslog, а также его тестирования с помощью утилиты logger.

Последняя часть главы в основном посвящена механизму ротации фалов журналов, а также содержит пояснения относительно использования команд tail -f и watch для отслеживания изменений файлов журналов.

17.1. Журналирование входов в систему

С целью облегчения процесса отслеживания входов пользователей в систему Linux может записывать необходимые данные в файлы журналов /var/log/wtmp, /var/log/btmp, /var/run/utmp и /var/log/lastlog.

17.1.1. Файл журнала /var/run/utmp (who)

Используйте утилиту who для просмотра содержимого файла /var/run/utmp. Эта утилита выводит информацию о пользователях, осуществивших вход в систему и в данный момент работающих с ней. Обратите внимание на то, что файл utmp находится в директории /var/run, а не /var/log.

```
[root@rhel4 ~]# who
paul pts/1 Feb 14 18:21 (192.168.1.45)
sandra pts/2 Feb 14 18:11 (192.168.1.42)
inge pts/3 Feb 14 12:01 (192.168.1.33)
els pts/4 Feb 14 14:33 (192.168.1.19)
```

17.1.2. Файл журнала /var/log/wtmp (last)

Содержимое файла журнала /var/log/wtmp обновляется силами программы login. Используйте утилиту last для просмотра содержимого файла журнала /var/run/wtmp.

```
[root@rhel4a ~]# last | head
paul pts/1 192.168.1.45 Wed Feb 14 18:39 still logged in
```

reboot system boot 2.6.9-42.0.8.ELs Wed Feb 14 18:21 (01:15)
nicolas pts/5 pc-dss.telematic Wed Feb 14 12:32 - 13:06 (00:33)
stefaan pts/3 pc-sde.telematic Wed Feb 14 12:28 - 12:40 (00:12)
nicolas pts/3 pc-nae.telematic Wed Feb 14 11:36 - 12:21 (00:45)
nicolas pts/3 pc-nae.telematic Wed Feb 14 11:34 - 11:36 (00:01)
dirk pts/5 pc-dss.telematic Wed Feb 14 10:03 - 12:31 (02:28)
nicolas pts/3 pc-nae.telematic Wed Feb 14 09:45 - 11:34 (01:48)
dimitri pts/5 rhel4 Wed Feb 14 07:57 - 08:38 (00:40)
stefaan pts/4 pc-sde.telematic Wed Feb 14 07:16 - down (05:50)
[root@rhel4a ~]#

Утилита last также может использоваться и для получения информации о последних перезагрузках.

[paul@rekkie ~]$ last reboot
reboot system boot 2.6.16-rekkie Mon Jul 30 05:13 (370+08:42)
wtmp begins Tue May 30 23:11:45 2006

17.1.3. Файл журнала /var/log/lastlog (lastlog)

Используйте утилиту lastlog для просмотра содержимого файла /var/log/lastlog.

[root@rhel4a ~]# lastlog | tail
tim pts/5 10.170.1.122 Вт фев 13 09:36:54 +0100 2007
rm pts/6 rhel4 Вт фев 13 10:06:56 +0100 2007
henk Никогда не входил в систему
stefaan pts/3 pc-sde.telematic Ср фев 14 12:28:38 +0100 2007
dirk pts/5 pc-dss.telematic Ср фев 14 10:03:11 +0100 2007
arsene Никогда не входил в систему
nicolas pts/5 pc-dss.telematic Ср фев 14 12:32:18 +0100 2007
dimitri pts/5 rhel4 Ср фев 14 07:57:19 +0100 2007
bashuserrm pts/7 rhel4 Чт фев 13 10:35:40 +0100 2007
kornuserrm pts/5 rhel4 Чт фев 13 10:06:17 +0100 2007
[root@rhel4a ~]#

17.1.4. Файл журнала /var/log/btmp (lastb)

Также существует утилита lastb, предназначенная для вывода содержимого файла /var/log/btmp. Содержимое этого файла обновляется программой login при вводе некорректного пароля, следовательно, он содержит информацию о неудачных попытках входа в систему. В файловых системах множества компьютеров данный файл может отсутствовать, в результате чего неудачные попытки входа в систему не будут журналироваться.

[root@RHEL4b ~]# lastb
lastb: невозможно открыть /var/log/btmp: Нет такого файла или каталога
[root@RHEL4b ~]#

Обычно данный файл удаляют по той причине, что пользователи иногда по ошибке вводят свой пароль вместо имени учетной записи, следовательно, читаемый всеми файл является потенциальной угрозой безопасности системы. Вы можете активировать механизм журналирования неудачных попыток входа в систему, просто создав файл с упомянутым именем. В этом случае исполнение команды chmod o-r /var/log/btmp позволит повысить защищенность системы.

[root@RHEL4b ~]# touch /var/log/btmp
[root@RHEL4b ~]# ll /var/log/btmp
-rw-r—r-- 1 root root 0 июл 30 06:12 /var/log/btmp
[root@RHEL4b ~]# chmod o-r /var/log/btmp
[root@RHEL4b ~]# lastb
btmp begins Пн июл 30 06:12:19 2007
[root@RHEL4b ~]#

Информация о попытках ввода некорректных паролей при использовании утилит ssh, rlogin или su не сохраняется в файле /var/log/btmp. В нем сохраняется исключительно информация о попытках ввода некорректного пароля при работе с терминалами.

[root@RHEL4b ~]# lastb
HalvarFl tty3 Mon Jul 30 07:10 - 07:10 (00:00)
Maria tty1 Mon Jul 30 07:09 - 07:09 (00:00)
Roberto tty1 Mon Jul 30 07:09 - 07:09 (00:00)
btmp begins Mon Jul 30 07:09:32 2007
[root@RHEL4b ~]#

17.1.5. Журналирование входов в систему с использованием утилит su и ssh

В зависимости от дистрибутива в файловой системе вашего компьютера вы также можете обнаружить файл /var/log/secure, который заполнен сообщениями от вспомогательных модулей auth и/или authpriv демона syslog. Этот файл журнала должен содержать информацию о неудачных попытках входа в систему с использованием утилиты su и/или ssh. В некоторых дистрибутивах данная информация сохраняется в файле /var/log/auth.log, поэтому следует проверить конфигурацию демона syslog.

[root@RHEL4b ~]# cat /var/log/secure
Jul 30 07:09:03 sshd[4387]: Accepted publickey for paul from ::ffff:19\
2.168.1.52 port 33188 ssh2
Jul 30 05:09:03 sshd[4388]: Accepted publickey for paul from ::ffff:19\
2.168.1.52 port 33188 ssh2
Jul 30 07:22:27 sshd[4655]: Failed password for Hermione from ::ffff:1\
92.168.1.52 port 38752 ssh2
Jul 30 05:22:27 sshd[4656]: Failed password for Hermione from ::ffff:1\
92.168.1.52 port 38752 ssh2

Jul 30 07:22:30 sshd[4655]: Failed password for Hermione from ::ffff:1\
92.168.1.52 port 38752 ssh2
Jul 30 05:22:30 sshd[4656]: Failed password for Hermione from ::ffff:1\
92.168.1.52 port 38752 ssh2
Jul 30 07:22:33 sshd[4655]: Failed password for Hermione from ::ffff:1\
92.168.1.52 port 38752 ssh2
Jul 30 05:22:33 sshd[4656]: Failed password for Hermione from ::ffff:1\
92.168.1.52 port 38752 ssh2
Jul 30 08:27:33 sshd[5018]: Invalid user roberto from ::ffff:192.168.1\
52
Jul 30 06:27:33 sshd[5019]: input_userauth_request: invalid user rober\
to
Jul 30 06:27:33 sshd[5019]: Failed none for invalid user roberto from \
::ffff:192.168.1.52 port 41064 ssh2
Jul 30 06:27:33 sshd[5019]: Failed publickey for invalid user roberto \
from ::ffff:192.168.1.52 port 41064 ssh2
Jul 30 08:27:36 sshd[5018]: Failed password for invalid user roberto f\
rom ::ffff:192.168.1.52 port 41064 ssh2
Jul 30 06:27:36 sshd[5019]: Failed password for invalid user roberto f\
rom ::ffff:192.168.1.52 port 41064 ssh2
[root@RHEL4b ~]#

Вы можете активировать данный механизм журналирования самостоятельно, указав путь к произвольному файлу путем добавления следующей строки в файл конфигурации syslog.conf.

auth.,authpriv. /var/log/customsec.log

17.2. Демон журналирования событий syslogd

17.2.1. О демоне syslogd

Стандартный метод журналирования событий в Linux связан с использованием демона syslogd. Демон syslogd был разработан Eric Allman для использования совместно с агентом передачи почты sendmail, но вскоре стал одним из множества стандартных приложений Unix, архитектура которого гораздо позднее была описана в рамках стандарта RFC 3164. Демон принимает сообщения от множества приложений (и системных утилит) по протоколу UDP на порту 514 и может дописывать сообщения в файлы журнала, выводить их, показывать сообщения в терминалах и передавать данные журнала другим демонам syslogd, работающим на других машинах. Конфигурация демона syslogd осуществляется с помощью файла /etc/syslog.conf.

17.2.2. О демоне rsyslogd

Новый демон журналирования событий носит имя rsyslogd (reliable and extended syslogd - надежный и расширяемый syslogd) и использует файл конфигурации /etc/rsyslogd.conf. Обратная совместимость синтаксиса данного файла конфигурации сохранена.

В каждой строке файла конфигурации используется идентификатор системы (facility) для определения источника сообщения. Также она содержит описание приоритета (priority) для определения важности сообщения и описание действия (action) для принятия решения о том, что нужно сделать с данным сообщением.

17.2.3. Модули

Новый демон rsyslog предоставляет много дополнительных возможностей, набор которых может быть расширен с помощью модулей. Модули позволяют, например, осуществлять экспорт сообщений из журнала syslog в базу данных.

Обратитесь к страницам руководств для получения дополнительной информации (после того, как вы закончите чтение данной главы).

root@rhel65:/etc# man rsyslog.conf
root@rhel65:/etc# man rsyslogd
root@rhel65:/etc#

17.2.4. Идентификаторы систем (facilities)

На странице руководства, доступной после исполнения команды man rsyslog.conf, можно найти информацию о различных стандартных идентификаторах систем для классификации сообщений от определенных демонов, таких, как mail, lpr, news и kern(el). Идентификаторы систем local0 и local7 могут использоваться для классификации сообщений от системных утилит (или любых подключенных к сети устройств, которые поддерживают механизм сообщений демона syslogd). Ниже приведен список всех идентификаторов систем из файла конфигурации rsyslog.conf версии 1.3. Ключевое слово security является устаревшим.

auth (security)
authpriv
cron
daemon
ftp
kern
lpr mail
mark (только для внутреннего использования)
news
syslog
user
uucp
local0-7

17.2.5. Описания приоритетов (priorities)

Наиболее важное сообщение может иметь описание приоритета emerg, вслед за которыми идут описания приоритетов alert и crit. Сообщения с наиболее низкими приоритетами имеют описания приоритетов info и debug. При указании минимального приоритета для журналирования сообщений будет осуществляться журналирование сообщений и с более высоким приоритетом. Вы можете использовать префикс = перед описанием приоритета для получения только тех сообщений, которые соответствуют этому описанию. Также вы можете использовать описание приоритета .none для предотвращения выполнения определенного действия при приеме любого сообщения с определенным идентификатором системы.

Ниже приведен список приоритетов в порядке возрастания. Ключевые слова warn, error и panic являются устаревшими.

debug
info
notice
warning (warn)
err (error)
crit
alert
emerg (panic)

17.2.6. Действия (actions)

Стандартным действием является отправка сообщения пользователю, имя которого записано в качестве действия. Если в качестве действия приводится строка с префиксом /, демон rsyslogd будет отправлять сообщение в файл (который может быть как обычным файлом, так и файлом устройства принтера или терминала). Префикс @ позволяет осуществлять отправку сообщения другому серверу с исполняющимся на нем демоном syslogd. Ниже приведен список всех возможных действий.

root,user1
передача сообщений пользователям с именами из списка, разделенными с помощью запятой

- передача сообщения всем пользователям, осуществившим вход в систему

/ запись сообщения в файл (который может представлять принтер, консоль, терминал, ...)
-/ запись сообщения в файл без принудительной записи данных каждого сообщения на диск
| запись сообщения в именованный программный канал
@ передача сообщения по сети другому демону syslog, исполняющемуся на узле с заданным именем

В дополнение вы можете использовать префикс - перед описаниями действий для исключения принудительной записи данных в файл на диске после журналирования каждого из событий.

17.2.7. Конфигурация

Ниже приведен пример фрагмента файла конфигурации /etc/rsyslog.conf для обработки нестандартных сообщений с идентификатором системы local4.

local4.crit /var/log/critandabove
local4.=crit /var/log/onlycrit
local4.* /var/log/alllocal4

17.2.8. Перезапуск демона rsyslogd

Не забудьте перезапустить демон после модификации его файла конфигурации.

root@rhel65:/etc# service rsyslog restart
Shutting down system logger: [OK]
Starting system logger: [OK]
root@rhel65:/etc#

17.3. Утилита logger

Утилита logger может использоваться для генерации тестовых сообщений для демона syslogd. Также вы можете использовать ее в сценариях командной оболочки. Ниже приведен пример команд, используемых для тестирования демона syslogd с помощью утилиты logger.

[root@rhel4a ~]# logger -p local4.debug "l4 debug"
[root@rhel4a ~]# logger -p local4.crit "l4 crit"
[root@rhel4a ~]# logger -p local4.emerg "l4 emerg"
[root@rhel4a ~]#

Результаты тестирования демона журналирования событий с помощью утилиты logger.

[root@rhel4a ~]# cat /var/log/critandabove
Feb 14 19:55:19 rhel4a paul: l4 crit
Feb 14 19:55:28 rhel4a paul: l4 emerg
[root@rhel4a ~]# cat /var/log/onlycrit
Feb 14 19:55:19 rhel4a paul: l4 crit
[root@rhel4a ~]# cat /var/log/alllocal4
Feb 14 19:55:11 rhel4a paul: l4 debug
Feb 14 19:55:19 rhel4a paul: l4 crit
Feb 14 19:55:28 rhel4a paul: l4 emerg
[root@rhel4a ~]#

17.4. Просмотр журналов событий

Вы можете использовать команду tail -f для просмотра последних строк файла журнала. Параметр -f позволяет

динамически выводить строки, которые добавляются в файл журнала в реальном времени.

paul@ubu1010:~$ tail -f /var/log/udev
SEQNUM=1741
SOUND_INITIALIZED=1
ID_VENDOR_FROM_DATABASE=nVidia Corporation
ID_MODEL_FROM_DATABASE=MCP79 High Definition Audio
ID_BUS=pci
ID_VENDOR_ID=0x10de
ID_MODEL_ID=0x0ac0
ID_PATH=pci-0000:00:08.0
SOUND_FORM_FACTOR=internal

Также вы можете автоматически повторять вызовы утилит, размещая перед ними вызов утилиты watch. Примером может служить следующая команда:

[root@rhel6 ~]# watch who

Данный подход аналогичен описанному выше, ведь в результате вывод утилиты who обновляется на экране через каждые две секунды.

Every 2.0s: who Sun Jul 17 15:31:03 2011
root tty1 2011-07-17 13:28
paul pts/0 2011-07-17 13:31 (192.168.1.30)
paul pts/1 2011-07-17 15:19 (192.168.1.30)

17.5. Ротация журналов событий

Размер большинства файлов журналов событий неуклонно растет. Для ограничения размеров этих файлов может использоваться утилита logrotate, предназначенная для ротации, сжатия, удаления и отправки по электронной почте файлов журналов событий. Дополнительная информация об утилите logrotate может быть получена из основного файла конфигурации /etc/logrotate.conf. Отдельные файлы конфигурации могут находиться в директории /etc/logrotate.d/.

Ниже приведено содержимое стандартного файла конфигурации logrotate.conf из состава дистрибутива Red Hat.

root@rhel65:/etc# cat logrotate.conf
see "man logrotate" for details (обратитесь к странице руководства man logrotate за подробностями)
rotate log files weekly (еженедельная ротация файлов журналов)
weekly
keep 4 weeks worth of backlogs (сохранение файлов журналов четырехнедельной давности)
rotate 4
create new (empty) log files after rotating old ones (создание новых (пустых) файлов журналов перед ротацией старых)
create
use date as a suffix of the rotated file (использование даты в качестве суффикса имени файла, подвергающегося ротации)
dateext
uncomment this if you want your log files compressed (раскомментируйте эту строку если вы хотите, чтобы ваши файлы журналов подвергались компрессии)
#compress
RPM packages drop log rotation information into this directory (информация о ротации файлов журналов, извлеченная из пакетов программного обеспечения формата RPM, должна размещаться в данной директории)
include /etc/logrotate.d
# no packages own wtmp and btmp — we'll rotate them here (ни один из пакетов программного обеспечения не содержит информации о ротации файлов журналов wtmp и btmp - мы приведем параметры их ротации здесь)
/var/log/wtmp {
monthly
create 0664 root utmp
minsize 1M
rotate 1
}
/var/log/btmp {
missingok
monthly
create 0600 root utmp
rotate 1
}
# system-specific logs may be also be configured here. (параметры ротации специфичных для системы файлов журналов также могут быть приведены здесь.)

root@rhel65:/etc#

17.6. Практическое задание: журналирование событий

1. Выведите содержимое файла журнала /var/run/utmp с помощью специально предназначенной для этой цели утилиты (без использования утилиты cat или текстового редактора vi).
2. Выведите аналогичным образом содержимое файла журнала /var/log/wtmp.
3. Используйте утилиты lastlog и lastb и сделайте вывод о различии этих утилит.
4. Исследуйте файл конфигурации демона журналирования

событий syslogd с целью выяснения пути к к файлу журнала событий, который содержит информацию о неудачных попытках удаленного входа в систему с помощью клиента ssh.

5. Настройте демон журналирования событий syslogd таким образом, чтобы сообщения с идентификатором системы и приоритетом local4.error и сообщения с более высокими приоритетами и этим же идентификатором системы размещались в файле журнала /var/log/l4e.log, а сообщения с идентификатором системы и исключительным приоритетом local4.info — в файле журнала /var/log/l4i.log. Проверьте корректность настройки с помощью утилиты logger!
6. Настройте демон журналирования событий syslogd таким образом, чтобы в файле журнала /var/log/Mysu.log размещались все сообщения, сгенерированные утилитой su в ходе получения привилегий пользователя root. Проверьте корректность настройки!
7. Настройте отправку сообщений с идентификатором системы local5 на сервер вашего соседа, на котором исполняется демон syslogd. Проверьте корректность настройки.
8. Разработайте сценарий, который будет вызывать утилиту logger для отправки сообщений с идентификатором системы local4 демону журналирования событий через каждые 15 секунд (текст сообщений должен отличаться). Используйте команду tail -f для отслеживания изменений ваших локальных файлов журналов событий.

17.7. Корректная процедура выполнения практического задания: журналирование событий

1. Выведите содержимое файла журнала /var/run/utmp с помощью специально предназначенной для этой цели утилиты (без использования утилиты cat или текстового редактора vi).

   who
2. Выведите аналогичным образом содержимое файла журнала /var/log/wtmp.

last

1. Используйте утилиты lastlog и lastb и сделайте вывод о различии этих утилит.

lastlog: выводит информацию о последних входах пользователей в систему

lastb: выводит информацию о неудачных попытках входа в систему

1. Исследуйте файл конфигурации демона журналирования событий syslogd с целью выяснения пути к к файлу журнала событий, который содержит информацию о неудачных попытках удаленного входа в систему с помощью клиента ssh.

RHEL: root@rhel53 ~# grep authpriv /etc/syslog.conf
authpriv.* /var/log/secure
Debian/Ubuntu: /var/log/auth.log
Дистрибутивы Ubuntu 9.10 и Debian Lenny переведены на использование демона журналирования событий rsyslog.
root@ubuntu910:~# grep authpriv /etc/rsyslog.d/50-default.conf
auth,authpriv.* /var/log/auth.log
root@deb503:~# grep authpriv /etc/rsyslog.conf
auth,authpriv.* /var/log/auth.log

1. Настройте демон журналирования событий syslogd таким образом, чтобы сообщения с идентификатором системы и приоритетом local4.error и сообщения с более высокими приоритетами и этим же идентификатором системы размещались в файле журнала /var/log/l4e.log, а сообщения с идентификатором системы и исключительным приоритетом local4.info — в файле журнала /var/log/l4i.log. Проверьте корректность настройки с помощью утилиты logger!

echo local4.error /var/log/l4e.log >> /etc/syslog.conf
echo local4.=info /var/log/l4i.log >> /etc/syslog.conf
/etc/init.d/syslog restart
logger -p local4.error "l4 error test"
logger -p local4.alert "l4 alert test"
logger -p local4.info "l4 info test"
cat /var/log/l4e.log
cat /var/log/l4i.log

1. Настройте демон журналирования событий syslogd таким образом, чтобы в файле журнала /var/log/Mysu.log размещались все сообщения, сгенерированные утилитой su в ходе получения привилегий пользователя root. Проверьте корректность настройки!

echo authpriv.* /var/log/Mysu.log >> /etc/syslog.conf
Данная директива позволит записывать в файл журнала событий не только данные, касающиеся использования утилиты su.

1. Настройте отправку сообщений с идентификатором системы local5 на сервер вашего соседа, на котором исполняется демон syslogd. Проверьте корректность настройки.

В дистрибутиве RHEL5 следует отредактировать файл /etc/sysconfig/syslog для активации режима приема сообщений от удаленных узлов.
В дистрибутиве RHEL7 следует раскомментировать две следующие строки файла /etc/rsyslog.conf для приема сообщений syslog с использованием протокола UDP.
Provides UDP syslog reception

$ModLoad imudp
$UDPServerRun 514
В Debian/Ubuntu следует отредактировать файл /etc/default/syslog или /etc/default/rsyslog.
На клиентском компьютере:
logger -p local5.info "Тестовое сообщение local5 для соседа"

1. Разработайте сценарий, который будет вызывать утилиту logger для отправки сообщений с идентификатором системы local4 демону журналирования событий через каждые 15 секунд (текст сообщений должен отличаться). Используйте команду tail -f для отслеживания изменений ваших локальных файлов журналов событий.

root@rhel53 scripts# cat logloop
#!/bin/bash
for i in seq 1 10
do
logger -p local4.info "local4.info тест номер $i"
sleep 15
done
root@rhel53 scripts# chmod +x logloop
root@rhel53 scripts# ./logloop &
[1] 8264
root@rhel53 scripts# tail -f /var/log/local4.all.log
Mar 28 13:13:36 rhel53 root: local4.info тест номер 1
Mar 28 13:13:51 rhel53 root: local4.info тест номер 2
...

Глава 18. Управление распределением памяти

Из данной главы вы узнаете о том, как управлять распределением оперативной памяти и кэшей.

В начале главы рассматриваются простые команды, предназначенные для вывода информации о состоянии памяти: free -om, top и cat /proc/meminfo.

После них описывается методика управления пространством подкачки, причем в описании используются такие понятия, как перемещение страницы памяти в область подкачки (swapping), перемещение страницы памяти в оперативную память (paging) и виртуальная память (virtual memory).

Последняя часть главы посвящена утилите vmstat, которая используется для мониторинга использования пространства подкачки.

18.1. Вывод информации о памяти и кэшах

18.1.1. Файл /proc/meminfo

Просмотрев содержимое файла /proc/meminfo, вы можете получить достаточно большой объем информации об использовании памяти вашего компьютера, работающего под управлением Linux.

paul@ubu1010:~$ cat /proc/meminfo
MemTotal: 3830176 kB
MemFree: 244060 kB
Buffers: 41020 kB
Cached: 2035292 kB
SwapCached: 9892 kB
...

Первая строка содержит информацию об общем объеме физической оперативной памяти, вторая строка — информацию об объеме неиспользованной оперативной памяти. В строке, начинающейся со слова «Buffers» содержится информация об объеме оперативной памяти, использованной для буферизации файлов, со слова «cached» — информация об объеме оперативной памяти, используемой в качестве кэша, а со слова «SwapCached» — информация об объеме памяти из пространства подкачки, используемой в качестве кэша. Данный файл также содержит много другой информации, рассмотрение которой в рамках данного курса является нецелесообразным.

18.1.2. Утилита free

Утилита free выводит информацию из файла /proc/meminfo в более удобном для чтения формате. В примере ниже показан вывод рассматриваемой утилиты, содержащий краткую информацию об использовании памяти со значениями в мегабайтах.

paul@ubu1010:~$ free -om
total used free shared buffers cached
Mem: 3740 3519 221 0 42 1994
Swap: 6234 82 6152

8.1.3.Утилита top

Обычно утилита top применяется для выявления процессов, которые наиболее интенсивно используют ресурс центрального процессора, но при этом данная утилита также выводит информацию о состоянии памяти в четвертой и пятой строке (эти строки могут скрываться и восстанавливаться с помощью клавиши m).

Ниже приведен пример вывода утилиты top после запуска на рассматриваемой системе ubu1010.

top - 10:44:34 up 16 days, 9:56, 6 users, load average: 0.13, 0.09,

0.12
Tasks: 166 total, 1 running, 165 sleeping, 0 stopped, 0 zombie
Cpu(s): 5.1%us, 4.6%sy, 0.6%ni, 88.7%id, 0.8%wa, 0.0%hi, 0.3%si, 0.0%st
Mem: 3830176k total, 3613720k used, 216456k free, 45452k buffers
Swap: 6384636k total, 84988k used, 6299648k free, 2050948k cached

18.2. Управление пространством подкачки

18.2.1. Информация о пространстве подкачки

Если операционной системе требуется объем памяти, превышающий объем имеющейся физической оперативной памяти, она может задействовать пространство подкачки. Пространство подкачки располагается на более медленном, но более дешевом устройстве для хранения данных. Обратите внимание на то, что хотя жесткие диски и используются чаще всего в качестве хранилищ для пространств подкачки, их значения времени доступа к данным в сотни тысяч раз хуже соответствующих значений оперативной памяти.

Пространство подкачки может быть представлено файлом, разделом жесткого диска или комбинацией файлов и разделов жесткого диска. Вы можете получить информацию об используемых пространствах подкачки выполнив либо команду free, либо команду cat /proc/swaps.

paul@ubu1010:~$ free -o | grep -v Mem
total used free shared buffers cached
Swap: 6384636 84988 6299648
paul@ubu1010:~$ cat /proc/swaps
Filename Type Size Used Priority
/dev/sda3 partition 6384636 84988 -1

Необходимый объем пространства подкачки зависит главным образом от предназначения рассматриваемого компьютера.

18.2.2. Создание раздела подкачки

Вы можете активировать и деактивировать пространство подкачки с помощью утилит swapon и swapoff соответственно. Новое пространство подкачки может быть создано с помощью утилиты mkswap. В приведенном ниже примере показан процесс создания и активации раздела подкачки.

root@RHELv4u4:~# fdisk -l 2> /dev/null | grep hda
Disk /dev/hda: 536 MB, 536870912 bytes
/dev/hda1 1 1040 524128+ 83 Linux
root@RHELv4u4:~# mkswap /dev/hda1
Setting up swapspace version 1, size = 536702 kB
root@RHELv4u4:~# swapon /dev/hda1

После выполнения описанных действий вы можете заметить, что в файле /proc/swaps содержится информация о каждом из пространств подкачки по отдельности, а при использовании команды free -om выводится только общая информация, отформатированная для более удобного чтения.

root@RHELv4u4:~# cat /proc/swaps
Filename Type Size Used Priority
/dev/mapper/VolGroup00-LogVol01 partition 1048568 0 -1
/dev/hda1 partition 524120 0 -2
root@RHELv4u4:~# free -om
total used free shared buffers cached
Mem: 249 245 4 0 125 54
Swap: 1 535 0 1535

18.2.3. Создание файла подкачки

Ниже приведен еще одни пример, иллюстрирующий процесс создания файла подкачки. При работе с операционной системой Solaris вы можете использовать утилиту mkfile вместо утилиты dd.

root@RHELv4u4:~# dd if=/dev/zero of=/smallswapfile bs=1024 count=4096
4096+0 записей получено
4096+0 записей отправлено
root@RHELv4u4:~# mkswap /smallswapfile
Setting up swapspace version 1, size = 4190 kB
root@RHELv4u4:~# swapon /smallswapfile
root@RHELv4u4:~# cat /proc/swaps
Filename Type Size Used Priority
/dev/mapper/VolGroup00-LogVol01 partition 1048568 0 -1
/dev/hda1 partition 524120 0 -2
/smallswapfile file 4088 0 -3

18.2.4. Описание пространства подкачки в файле /etc/fstab

Если вы хотите использовать созданные пространства подкачки на постоянной основе, не забудьте добавить их описание в файл /etc/fstab. Строки, которые следует добавить в данный файл, должны быть аналогичны приведенным ниже.

/dev/hda1 swap swap defaults 0 0
/smallswapfile swap swap defaults 0 0

18.3. Мониторинг использования памяти с помощью утилиты vmstat

Для получения информации об использовании пространства подкачки может использоваться утилита vmstat.

В примере ниже приведен вывод утилиты vmstat с значениями в мегабайтах.

paul@ubu1010:~$ vmstat -S m
procs ---------memory-------- ---swap-- -----io---- -system- ----cpu----
r b swpd free buff cache si so bi bo in cs us sy id wa

0 0 87 225 46 2097 0 0 2 5 14 8 6 5 89 1

В следующем примере приведен вывод утилиты vmstat после исполнения пользователем root (в другом терминале) команды find /. Утилита find выполняет множество операций дискового ввода-вывода (в столбцах bi и bo расположены счетчики прочитанных и записанных блоков диска соответственно). В данных обстоятельствах нет повода для использования пространства подкачки.

```
paul@ubu1010:~$ vmstat 2 100
procs ----------memory---------- ---swap-- -----io---- -system-- ----cpu----
r b swpd free buff cache si so bi bo in cs us sy id wa
0 0 84984 1999436 53416 269536 0 0 2 5 2 10 6 5 89 1
0 0 84984 1999428 53416 269564 0 0 0 0 1713 2748 4 4 92 0
0 0 84984 1999552 53416 269564 0 0 0 0 1672 1838 4 6 90 0
0 0 84984 1999552 53424 269560 0 0 0 14 1587 2526 5 7 87 2
0 0 84984 1999180 53424 269580 0 0 0 100 1748 2193 4 6 91 0
1 0 84984 1997800 54508 269760 0 0 610 0 1836 3890 17 10 68 4
1 0 84984 1994620 55040 269748 0 0 250 168 1724 4365 19 17 56 9
0 1 84984 1978508 55292 269704 0 0 126 0 1957 2897 19 18 58 4
0 0 84984 1974608 58964 269784 0 0 1826 478 2605 4355 7 7 44 41
0 2 84984 1971260 62268 269728 0 0 1634 756 2257 3865 7 7 47 39
```

А в примере ниже представлен вывод утилиты vmstat при исполнении (в дистрибутиве RHEL6) простой программы, повторяющей поведение приложения с утечкой памяти. Теперь вы можете обнаружить большое количество страниц памяти в пространстве подкачки (или «страниц памяти, перемещенных в пространство подкачки»).

```
[paul@rhel6c ~]$ vmstat 2 100
procs ----------memory-------- ---swap-- ----io---- --system-- -----cpu-----
r b swpd free buff cache si so bi bo in cs us sy id wa st
0 3 245208 5280 232 1916 261 0 0 42 27 21 0 1 98 1 0
0 2 263372 4800 72 908 143840 128 0 1138 462 191 2 10 0 88 0
1 3 350672 4792 56 992 169280 256 0 1092 360 142 1 13 0 86 0
1 4 449584 4788 56 1024 95880 64 0 606 471 191 2 13 0 85 0
0 4 471968 4828 56 1140 44832 80 0 390 235 90 2 12 0 87 0
3 5 505960 4764 56 1136 68008 16 0 538 286 109 1 12 0 87 0
```

Для симуляции утечки памяти (и принудительного переноса страниц памяти в пространство подкачки) был использован приведенный ниже код. Этот код был найден в энциклопедии Wikipedia, причем имя автора не было указано.

```
paul@mac:~$ cat memleak.c
#include <stdlib.h>
int main(void)
{
while (malloc(50));
return 0;
}
```

18.4. Практическое задание: память

1. Используйте утилиту dmesg для получения информации об общем объеме оперативной памяти вашего компьютера.
2. Используйте утилиту free для вывода информации об использовании памяти с значениями в килобайтах (а после этого в мегабайтах).
3. Работая с операционной системой, запущенной в виртуальной машине, создайте раздел подкачки (для этой цели вам может потребоваться дополнительный виртуальный диск).
4. Добавьте в систему файл подкачки размером в 20 мегабайт.
5. Разместите описания всех созданных пространств подкачки в файле /etc/fstab и активируйте их. Проверьте, монтируются ли они после перезагрузки системы.
6. Используйте утилиту free для проверки использования созданных пространств подкачки.

7 (дополнительное задание). Выведите информацию об использовании созданных пространств подкачки в условиях запуска приложения с утечкой памяти с помощью команд vmstat и free -s.

18.5. Корректная процедура выполнения практического задания: память

1. Используйте утилиту dmesg для получения информации об общем объеме оперативной памяти вашего компьютера.

```
dmesg | grep Memory
```

1. Используйте утилиту free для вывода информации об использовании памяти с значениями в килобайтах (а после этого в мегабайтах).

```
free ; free -m
```

1. Работая с операционной системой, запущенной в виртуальной машине, создайте раздел подкачки (для этой цели вам может потребоваться дополнительный виртуальный диск).

```
mkswap /dev/sdd1 ; swapon /dev/sdd1
```

1. Добавьте в систему файл подкачки размером в 20 мегабайт.

```
dd if=/dev/zero of=/swapfile20mb bs=1024 count=20000
mkswap /swapfile20mb
swapon /swapfile20mb
```

1. Разместите описания всех созданных пространств подкачки в файле /etc/fstab и активируйте их. Проверьте, монтируются ли они после перезагрузки системы.

root@computer# tail -2 /etc/fstab
/dev/sdd1 swap swap defaults 0 0
/swapfile20mb swap swap defaults 0 0

1. Используйте утилиту free для проверки использования созданных пространств подкачки.

free -om

7 (дополнительное задание). Выведите информацию об использовании созданных пространств подкачки в условиях запуска приложения с утечкой памяти с помощью команд vmstat и free -s.

Глава 19. Мониторинг использования ресурсов системы

Процесс мониторинга использования ресурсов системы заключается в сборе информации об использовании ресурсов памяти, центрального процессора, сети и устройств для хранения данных. Вы должны начать мониторинг использования ресурсов вашей системы как можно быстрее для того, чтобы иметь возможность установления соответствующих базовых показателей. Убедитесь в том, что вы установили все базовые показатели использования ресурсов вашей системы! Эти базовые показатели важны, так как они позволят вам обнаружить последовательный или внезапный рост потребления ресурсов системы и, соответственно, последовательное (или внезапное) сокращение доступных ресурсов системы. Эта информация позволяет планировать масштабирование системы с целью расширения или сокращения доступных ресурсов.

Давайте рассмотрим некоторые инструменты, которые могут использоваться для мониторинга использования ресурсов системы помимо рассмотренных ранее команд ps fax, df -h, free -om и du -sh.

19.1. Четыре базовых типа ресурсов системы

Четырьмя базовыми типами ресурсов системы являются:

- Ресурсы центрального процессора
- Ресурсы сети
- Ресурсы оперативной памяти
- Ресурсы устройств для хранения данных

19.2. Утилита top

В начале процесса мониторинга использования ресурсов системы вы можете прибегнуть к использованию утилиты top. Данная утилита позволяет осуществлять мониторинг использования ресурсов оперативной памяти, центрального процессора и пространства подкачки. Утилита top автоматически обновляет вывод. После запуска данной утилиты вы можете использовать множество команд, таких, как команда k, предназначенная для завершения работы процессов, команды t или m, предназначенные для показа и скрытия информации о процессах и использовании памяти или команда 1, предназначенная для переключения между режимами вывода информации об использовании ресурсов каждого отдельного центрального процессора или всех центральных процессоров.

top - 12:23:16 up 2 days, 4:01, 2 users, load average: 0.00, 0.00, 0.00
Tasks: 61 total, 1 running, 60 sleeping, 0 stopped, 0 zombie
Cpu(s): 0.3% us, 0.5% sy, 0.0% ni, 98.9% id, 0.2% wa, 0.0% hi, 0.0% si
Mem: 255972k total, 240952k used, 15020k free, 59024k buffers
Swap: 524280k total, 144k used, 524136k free, 112356k cached
PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
1 root 16 0 2816 560 480 S 0.0 0.2 0:00.91 init
2 root 34 19 0 0 0 S 0.0 0.0 0:00.01 ksoftirqd/0
3 root 5 -10 0 0 0 S 0.0 0.0 0:00.57 events/0
4 root 5 -10 0 0 0 S 0.0 0.0 0:00.00 khelper
5 root 15 -10 0 0 0 S 0.0 0.0 0:00.00 kacpid
16 root 5 -10 0 0 0 S 0.0 0.0 0:00.08 kblockd/0
26 root 15 0 0 0 0 S 0.0 0.0 0:02.86 pdflush
...

Вы можете настроить утилиту top таким образом, чтобы осуществлялся вывод исключительно выбранных вами столбцов или вывод параметров единственного процесса, который вас интересует.

[paul@RHELv4u3 ~]$ top p 3456 p 8732 p 9654

19.3. Утилита free

Утилита free является стандартным инструментом из состава различных дистрибутивов Linux, предназначенным для осуществления мониторинга объема неиспользованной памяти. Вы можете использовать утилиту free для вывода информации через каждые x секунд, но вывод не будет таким же идеальным, как у предыдущего инструмента.

[paul@RHELv4u3 gen]$ free -om -s 10
total used free shared buffers cached
Mem: 249 222 27 0 50 109
Swap: 511 0 511
total used free shared buffers cached
Mem: 249 222 27 0 50 109
Swap: 511 0 511
[paul@RHELv4u3 gen]$

19.4. Утилита watch

Более интересных результатов можно достичь при комбинировании утилиты free с утилитой watch. Эта утилита может исполнять команды с задержкой, а также выделять изменения в выводе (в случае использования параметра -d).

```
[paul@RHELv4u3 ~]$ watch -d -n 3 free -om
...
Every 3.0s: free -om Sat Jan 27 12:13:03 2007
total used free shared buffers cached
Mem: 249 230 19 0 56 109
Swap: 511 0 511
```

19.5. Утилита vmstat

Для получения статистики использования ресурсов центрального процессора, устройств хранения данных и памяти в однострочном формате в составе дистрибутивов имеется утилита vmstat. В примере ниже показан вывод утилиты vmstat, обновляющей значения параметров 100 раз в течение каждых двух секунд (или до момента нажатия сочетания клавиш Ctrl-C). В столбце r приводится количество процессов, ожидающих доступности ресурсов центрального процессора, а количество процессов в неактивном состоянии приводится в столбце b. Объем станиц памяти в пространстве подкачки остается неизменным и равным 144 килобайтам, объем свободной оперативной памяти резко снизился с 16.7 МБ до 12.9 МБ. Для получения информации об остальных данных, выводимых утилитой vmstat, следует обратиться к странице руководства man vmstat.

```
[paul@RHELv4u3 ~]$ vmstat 2 100
procs ----------memory--------- --swap-- ---io--- --system-- ---cpu----
r b swpd free buff cache si so bi bo in cs us sy id wa
0 0 144 16708 58212 111612 0 0 3 4 75 62 0 1 99 0
0 0 144 16708 58212 111612 0 0 0 0 976 22 0 0 100 0
0 0 144 16708 58212 111612 0 0 0 0 958 14 0 1 99 0
1 0 144 16528 58212 111612 0 0 0 18 1432 7417 1 32 66 0
1 0 144 16468 58212 111612 0 0 0 0 2910 20048 4 95 1 0
1 0 144 16408 58212 111612 0 0 0 0 3210 19509 4 97 0 0
1 0 144 15568 58816 111612 0 0 300 1632 2423 10189 2 62 0 36
0 1 144 13648 60324 111612 0 0 754 0 1910 2843 1 27 0 72
0 0 144 12928 60948 111612 0 0 312 418 1346 1258 0 14 57 29
0 0 144 12928 60948 111612 0 0 0 0 977 19 0 0 100 0
0 0 144 12988 60948 111612 0 0 0 0 977 15 0 0 100 0
0 0 144 12988 60948 111612 0 0 0 0 978 18 0 0 100 0
[paul@RHELv4u3 ~]$
```

19.6. Утилита iostat

Утилита iostat может выводить статистическую информацию об использовании ресурсов устройств для хранения данных и центрального процессора. Использованный в примере ниже параметр -d сообщает утилите iostat о необходимости вывода информации об использовании ресурсов устройств для хранения данных (500 раз в течение каждых двух секунд). В первом блоке данных приведена статистическая информация об использовании описанных ресурсов с момента последней перезагрузки системы.

```
[paul@RHELv4u3 ~]$ iostat -d 2 500
Linux 2.6.9-34.EL (RHELv4u3.localdomain) 01/27/2007
Device: tps Blk_read/s Blk_wrtn/s Blk_read Blk_wrtn
hdc 0.00 0.01 0.00 1080 0
sda 0.52 5.07 7.78 941798 1445148
sda1 0.00 0.01 0.00 968 4
sda2 1.13 5.06 7.78 939862 1445144
dm-0 1.13 5.05 7.77 939034 1444856
dm-1 0.00 0.00 0.00 360 288
Device: tps Blk_read/s Blk_wrtn/s Blk_read Blk_wrtn
hdc 0.00 0.00 0.00 0 0
sda 0.00 0.00 0.00 0 0
sda1 0.00 0.00 0.00 0 0
sda2 0.00 0.00 0.00 0 0
dm-0 0.00 0.00 0.00 0 0
dm-1 0.00 0.00 0.00 0 0
...
[paul@RHELv4u3 ~]$
```

Вы можете получать больший объем статистических данных, использовав команду iostat -d -x, или получать исключительно статистические данные, касающиеся центрального процессора, воспользовавшись командой iostat -c.

```
[paul@RHELv4u3 ~]$ iostat -c 5 500
Linux 2.6.9-34.EL (RHELv4u3.localdomain) 01/27/2007
avg-cpu: %user %nice %sys %iowait %idle
0.31 0.02 0.52 0.23 98.92
avg-cpu: %user %nice %sys %iowait %idle
0.62 0.00 52.16 47.23 0.00
avg-cpu: %user %nice %sys %iowait %idle
2.92 0.00 36.95 60.13 0.00
avg-cpu: %user %nice %sys %iowait %idle
0.63 0.00 36.63 62.32 0.42
avg-cpu: %user %nice %sys %iowait %idle
0.00 0.00 0.20 0.20 99.59
[paul@RHELv4u3 ~]$
```

19.7. Утилита mpstat

При работе с мультипроцессорными машинами для получения

статистических данных об использовании ресурсов всех или выбранных центральных процессоров может использоваться утилита mpstat.

```
paul@laika:~$ mpstat -P ALL
Linux 2.6.20-3-generic (laika) 02/09/2007
CPU %user %nice %sys %iowait %irq %soft %steal %idle intr/s
all 1.77 0.03 1.37 1.03 0.02 0.39 0.00 95.40 1304.91
0 1.73 0.02 1.47 1.93 0.04 0.77 0.00 94.04 1304.91
1 1.81 0.03 1.27 0.13 0.00 0.00 0.00 96.76 0.00
paul@laika:~$
```

19.8. Утилиты sadc и sar

Утилита sadc предназначена для записи данных, связанных с использованием ресурсов системы, в файлы /var/log/sa??, где символы ?? заменяются на номер текущего дня месяца. По умолчанию демон crond исполняет сценарий sal через каждые 10 минут, а сценарий sal, в свою очередь, запускает на секунду утилиту sadc. Также ежедневно перед полночью демон crond исполняет сценарий sa2, который, в свою очередь, запускает утилиту sar. Утилита sar читает данные, собранные утилитой sadc в течение дня, и помещает сгенерированный на их основе отчет в файл /var/log/sa/sar??. Эти отчеты, созданные утилитой sar, содержат большой объем статистических данных.

Также вы можете использовать утилиту sar для вывода фрагмента собранных статистических данных. В примере ниже приведен такой фрагмент статистических данных, относящихся к центральному процессору.

```
[paul@RHELv4u3 sa]$ sar -u | head
Linux 2.6.9-34.EL (RHELv4u3.localdomain) 01/27/2007
12:00:01 AM CPU %user %nice %system %iowait %idle
12:10:01 AM all 0.48 0.01 0.60 0.04 98.87
12:20:01 AM all 0.49 0.01 0.60 0.06 98.84
12:30:01 AM all 0.49 0.01 0.64 0.25 98.62
12:40:02 AM all 0.44 0.01 0.62 0.07 98.86
12:50:01 AM all 0.42 0.01 0.60 0.10 98.87
01:00:01 AM all 0.47 0.01 0.65 0.08 98.80
01:10:01 AM all 0.45 0.01 0.68 0.08 98.78
[paul@RHELv4u3 sa]$
```

Существуют и другие полезные параметры утилиты sar, такие, как sar -I PROC, позволяющий выводить информацию об обработке запросов прерываний для каждого из обработанных запросов прерываний и для каждого центрального процессора, или sar -r, управляющий обработкой статистических данных, связанных с памятью. Для ознакомления с другими параметрами утилиты sar обратитесь к странице руководства man sar.

19.9. Утилита ntop

Утилита ntop не будет присутствовать в системе после стандартной процедуры установки дистрибутива Red Hat. После запуска данная утилита начнет генерацию подробных отчетов о сетевом трафике, которые доступны в формате html по адресу http://localhost:3000.

19.10. Утилита iftop

Утилита iftop выводит статистическую информацию об использовании пропускной способности определенных сетевых устройств различными соединениями, представленными сокетами. На серверах, работающих под управлением дистрибутива Red Hat, данная утилита также недоступна по умолчанию.

```
  1.91Mb 3.81Mb 5.72Mb 7.63Mb 9.54Mb
  ------------|------------|------------|------------|-------|----
  laika.local ⇒ barry 4.94Kb 6.65Kb 69.9Kb
  <= 7.41Kb 16.4Kb 766Kb
  laika.local ⇒ ik-in-f19.google.com 0b 1.58Kb 14.4Kb
  <= 0b 292b 41.0Kb
  laika.local ⇒ ik-in-f99.google.com 0b 83b 4.01Kb
  <= 0b 83b 39.8Kb
  laika.local ⇒ ug-in-f189.google.com 0b 42b 664b
  <= 0b 42b 406b
  laika.local ⇒ 10.0.0.138 0b 0b 149b
  <= 0b 0b 256b
  laika.local ⇒ 224.0.0.251 0b 0b 86b
  <= 0b 0b 0b
  laika.local ⇒ ik-in-f83.google.com 0b 0b 39b
  <= 0b 0b 21b
```

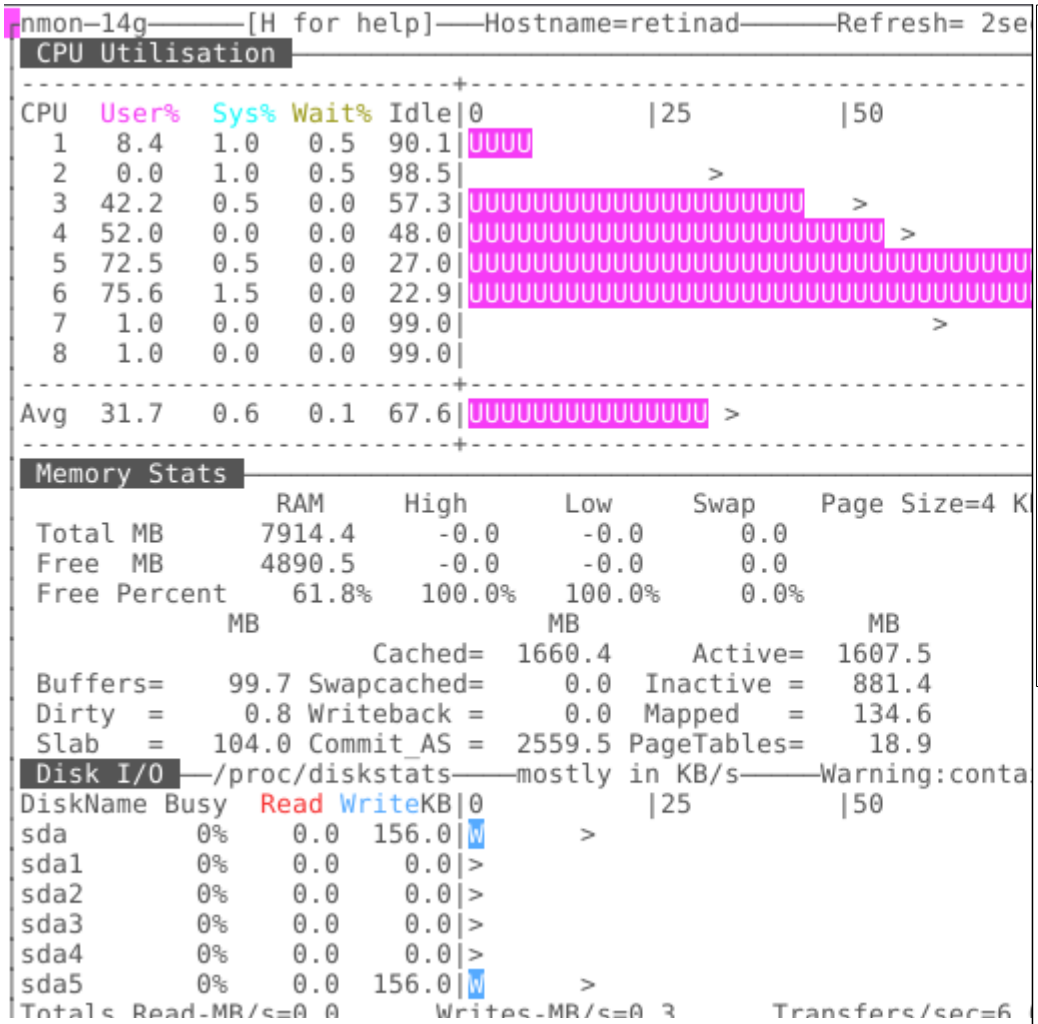
19.11. Утилита iptraf

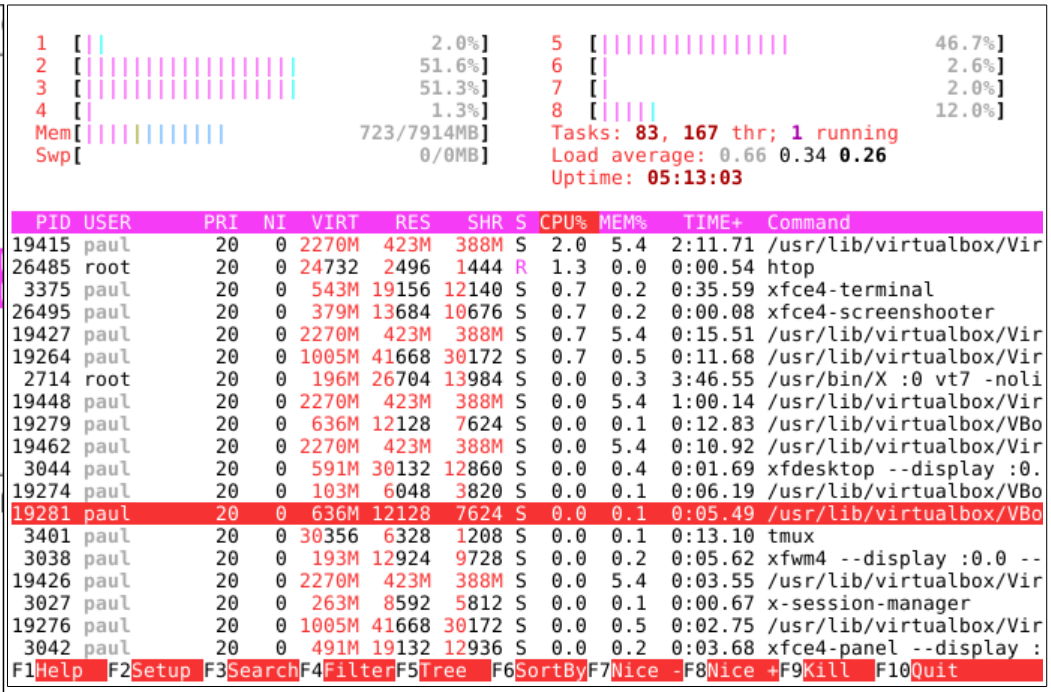
Используйте утилиту iptraf для получения вывода с цветовой разметкой, содержащего информацию о проходящем через сетевые карты IP-трафике.

```
[root@centos65 ~]# iptraf
[root@centos65 ~]# iptraf -i eth0
```

19.12. Утилита nmon

Другим популярным и распространенным инструментом для отслеживания сетевой активности является утилита nmon.





19.13. Утилита htop

Вы можете использовать утилиту htop вместо утилиты top.

Глава 20. Управление пакетами программного обеспечения

В большинстве дистрибутивов Linux используются системы управления пакетами программного обеспечения, позволяющие взаимодействовать с сетевыми репозиториями программного обеспечения, содержащими тысячи пакетов. Эти системы позволяют без лишних сложностей устанавливать и удалять приложения, компоненты операционной системы, документацию, а также другие программные компоненты.

В первую очередь в главе будут рассматриваться пакеты программного обеспечения дистрибутива Debian с расширением .deb, а также инструменты, предназначенные для работы с этими пакетами, а именно dpkg, apt-get и aptitude. Пакеты программного обеспечения упомянутого формата и соответствующие инструменты используются в дистрибутивах Debian, Ubuntu, Mint, а также во всех производных дистрибутивах.

После этого будут рассматриваться пакеты программного обеспечения дистрибутива Red Hat с расширением .rpm, а также инструменты, предназначенные для работы с этими пакетами, а именно rpm и yum. Пакеты программного обеспечения

упомянутого формата и соответствующие инструменты используются в дистрибутивах Red Hat, Fedora, CentOS, а также во всех производных дистрибутивах.

20.1. Терминология

20.1.1. Репозиторий

Программное обеспечение и документация для вашего дистрибутива Linux могут быть получены в формате пакетов из одного или нескольких идентичных распределенных репозиториев. Пакеты из репозитория протестированы на работоспособность и могут достаточно просто устанавливаться в систему и удаляться из нее с помощью программы установки пакетов программного обеспечения с графическим интерфейсом или интерфейсом командной строки.

20.1.2. Пакеты программного обеспечения с расширением .deb

В дистрибутивах Debian, Ubuntu, Mint и во всех производных от Debian и Ubuntu дистрибутивах используются пакеты программного обеспечения с расширением .deb. Для управления программным обеспечением в данных дистрибутивах могут использоваться инструменты aptitude и apt-get, которые, в свою очередь, используют низкоуровневую утилиту dpkg.

20.1.3. Пакеты программного обеспечения с расширением .rpm

В дистрибутивах Red Hat, Fedora, CentOS, OpenSUSE, Mandriva, Red Flag, а также других дистрибутивах используются пакеты программного обеспечения с расширением .rpm. Инструментами для управления программным обеспечением в этих дистрибутивах являются yum и rpm.

20.1.4. Зависимости

Некоторым пакетам программного обеспечения для корректного функционирования требуются другие пакеты программного обеспечения. Такие инструменты, как apt-get, aptitude и yum позволяют устанавливать все необходимые зависимости в процессе установки пакетов программного обеспечения. В случае использования утилит dpkg и rpm или сборки программного обеспечения из исходных кодов вам придется установить все зависимости самостоятельно.

20.1.5. Программное обеспечение с открытым исходным кодом

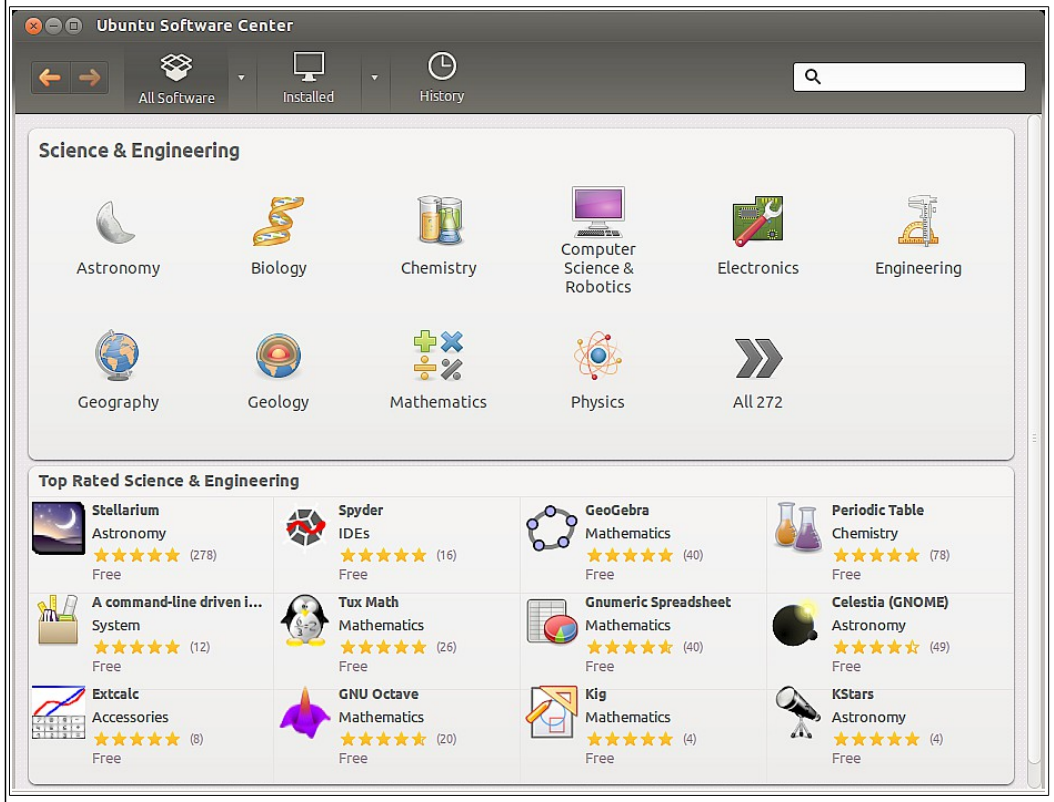
Репозитории содержат широкий спектр программного обеспечения с открытым исходным кодом. Нередко в исходный код программного обеспечения вносятся изменения, направленные на лучшую интеграцию данного программного обеспечения с вашим дистрибутивом. В большинстве дистрибутивов также осуществляется распространение результирующего модифицированного исходного кода программного обеспечения в форме пакетов из репозиториев исходного кода.

При этом у вас также остается возможность посещения вебсайта интересующего вас проекта (samba.org, apache.org, github.com, ...) и скачивания ванильного исходного кода (т.е., исходного кода без изменений, внесенных разработчиками дистрибутива).

20.1.6. Управление программным обеспечением с помощью приложения с графическим интерфейсом

Для удобства конечных пользователей существует несколько приложений с графическим интерфейсом, предназначенных для управления программным обеспечением в процессе использования окружения рабочего стола (вам следует искать пункт меню «Установка / удаление программного обеспечения» или что-то похожее).

Ниже приведен снимок окна Центра приложений Ubuntu из состава дистрибутива Ubuntu 12.04. Приложения с графическим интерфейсом не будут обсуждаться в рамках данной книги.



20.2. Управление пакетами программного обеспечения формата deb

20.2.1. О пакетах программного обеспечения формата deb

Большинство пользователей применяет утилиту aptitude или apt-get для управления программным обеспечением своих дистрибутивов Linux семейства Debian/Ubuntu. Обе упомянутые утилиты используют низкоуровневую утилиту dpkg, а сами являются основой для приложения synaptic и других приложений с графическим интерфейсом.

20.2.2. Команда dpkg -l

Утилита dpkg является низкоуровневым инструментом, предназначенным для работы с пакетами программного обеспечения формата deb. Ниже приведена команда для получения списка всех установленных пакетов программного обеспечения, которая вначале выполняется на сервере, работающем под управлением дистрибутива Debian.

```
root@debian6:~# dpkg -l | wc -l
265
```

Сравните этот вывод с выводом этой же команды, выполненной на настольном компьютере, который работает под управлением дистрибутива Ubuntu.

```
root@ubu1204~# dpkg -l | wc -l
2527
```

20.2.3. Команда dpkg -l $имя_пакета

Ниже приведен пример команды для получения информации об отдельном пакете программного обеспечения. Символы ii в начале строки говорят о том, что пакет программного обеспечения установлен.

```
root@debian6:~# dpkg -l rsync | tail -1 | tr -s ' '
ii rsync 3.0.7-2 fast remote file copy program (like rcp)
```

20.2.4. Команда dpkg -S

Вы можете найти пакет программного обеспечения, из которого был извлечен определенный файл, находящийся в файловой системе вашего компьютера, с помощью команды dpkg -S. В данном примере показана методика поиска пакета программного обеспечения, из которого были извлечены три файла, которые можно обнаружить на среднестатистическом работающем под управлением дистрибутива Debain сервере.

```
root@debian6:~# dpkg -S /usr/share/doc/tmux/ /etc/ssh/ssh_config /sbin/ifconfig
tmux: /usr/share/doc/tmux/
openssh-client: /etc/ssh/ssh_config
net-tools: /sbin/ifconfig
```

20.2.5. Команда dpkg -L

Также вы можете получить список всех файлов, которые устанавливаются в процессе установки определенной программы. Ниже приведен список файлов, устанавливаемых в процессе установки пакета программного обеспечения tmux.

```
root@debian6:~# dpkg -L tmux
/.
/etc
/etc/init.d
/etc/init.d/tmux-cleanup
/usr
/usr/share
/usr/share/lintian
/usr/share/lintian/overrides
/usr/share/lintian/overrides/tmux
/usr/share/doc
/usr/share/doc/tmux
/usr/share/doc/tmux/TODO.gz
/usr/share/doc/tmux/FAQ.gz
/usr/share/doc/tmux/changelog.Debian.gz
/usr/share/doc/tmux/NEWS.Debian.gz
/usr/share/doc/tmux/changelog.gz
/usr/share/doc/tmux/copyright
/usr/share/doc/tmux/examples
/usr/share/doc/tmux/examples/tmux.vim.gz
/usr/share/doc/tmux/examples/h-boetes.conf
/usr/share/doc/tmux/examples/n-marriott.conf
/usr/share/doc/tmux/examples/screen-keys.conf
/usr/share/doc/tmux/examples/t-williams.conf
/usr/share/doc/tmux/examples/vim-keys.conf
/usr/share/doc/tmux/NOTES
/usr/share/man
/usr/share/man/man1
/usr/share/man/man1/tmux.1.gz
/usr/bin
/usr/bin/tmux
```

10.2.6. Основные функции утилиты dpkg

Вы можете использовать команду dpkg -i для установки пакета программного обеспечения и команду dpkg -r для удаления пакета программного обеспечения, но в этом случае вам придется вручную разрешить все зависимости. Гораздо проще пользоваться такими инструментами, как apt-get и aptitude.

20.3. Утилита apt-get

Утилита apt-get используется для управления пакетами программного обеспечения в дистрибутиве Debian с 1998 года. На сегодняшний день в рамках дистрибутива Debian, а также множества основанных на Debain дистрибутивов осуществляется активная поддержка утилиты apt-get, хотя некоторые эксперты и заявляют, что утилита aptitude обрабатывает зависимости гораздо лучше, чем apt-get.

Обе упомянутые утилиты используют одни и те же конфигурационные файлы и могут заменять друг друга; если вы обнаружите упоминание об утилите apt-get в какой-либо документации, можете смело заменять apt-get на aptitude.

Сначала рассмотрим утилиту apt-get, после чего перейдем к обсуждению утилиты aptitude в следующем разделе.

20.3.1. Команда apt-get update

При исполнении команды apt-get update осуществляется автоматическая загрузка файлов со списками имен, версий и краткими описаниями всех пакетов программного обеспечения, доступных из всех репозиториев, которые активированы в вашей системе.

В выводе из примера ниже можно обнаружить несколько репозиториев с строками URL, начинающимися с be.archive.ubuntu.com из-за того, что данный компьютер установлен в Бельгии. В вашем случае строки URL могут отличаться.

```
root@ubu1204~# apt-get update
Игн http://be.archive.ubuntu.com precise InRelease
Игн http://extras.ubuntu.com precise InRelease
Игн http://security.ubuntu.com precise-security InRelease
Игн http://archive.canonical.com precise InRelease
Игн http://be.archive.ubuntu.com precise-updates InRelease
...
В кэше http://be.archive.ubuntu.com precise-backports/main Translation-en
В кэше http://be.archive.ubuntu.com precise-backports/multiverse Translation-en
В кэше http://be.archive.ubuntu.com precise-backports/restricted Translation-en
В кэше http://be.archive.ubuntu.com precise-backports/universe Translation-en
Получено 13.7 МБ за 8с (1682 кБ/с)
Чтение списков пакетов... Готово
root@ubu1204~#
```

Выполняйте команду apt-get update каждый раз перед выполнением других операций с пакетами программного обеспечения.

20.3.2. Команда apt-get upgrade

Одна из наиболее полезных возможностей утилиты apt-get заключается в безопасном обновлении всего установленного на данный момент на вашем компьютере программного обеспечения с помощью единственной команды.

```
root@debian6:~# apt-get upgrade
Чтение списков пакетов... Готово
Построение дерева зависимостей
Чтение информации о состоянии... Готово
обновлено 0, установлено 0 новых пакетов, для удаления отмечено 0 пакетов, и 0 пакетов не обновлено.
root@debian6:~#
```

В примере выше показана ситуация, в которой все программное обеспечение уже обновлено до последних версий доступных в моей версии дистрибутива.

20.3.3. Команда apt-get clean

Утилита apt-get хранит копии загруженных пакетов программного обеспечения в директории /var/cache/apt/archives, что хорошо видно в примере ниже.

```
root@ubu1204~# ls /var/cache/apt/archives/ | head
accountsservice_0.6.15-2ubuntu9.4_i386.deb
apport_2.0.1-0ubuntu14_all.deb
apport-gtk_2.0.1-0ubuntu14_all.deb
apt_0.8.16~exp12ubuntu10.3_i386.deb
apt-transport-https_0.8.16~exp12ubuntu10.3_i386.deb
apt-utils_0.8.16~exp12ubuntu10.3_i386.deb
bind9-host_1%3a9.8.1.dfsg.P1-4ubuntu0.4_i386.deb
chromium-browser_20.0.1132.47~r144678-0ubuntu0.12.04.1_i386.deb
chromium-browser-l10n_20.0.1132.47~r144678-0ubuntu0.12.04.1_all.deb
chromium-codecs-ffmpeg_20.0.1132.47~r144678-0ubuntu0.12.04.1_i386.deb
```

Команда apt-get clean позволяет удалить все файлы из этой директории.

```
root@ubu1204~# apt-get clean
root@ubu1204~# ls /var/cache/apt/archives/*.deb
ls: невозможно получить доступ к /var/cache/apt/archives/*.deb: Нет такого файла или каталога
```

20.3.4. Команда apt-cache search

Используйте команду apt-cache search для проверки доступности пакета программного обеспечения. В данном примере мы ищем пакет с именем rsync.

root@ubu1204~# apt-cache search rsync | grep ^rsync
rsync - fast, versatile, remote (and local) file-copying tool
rsyncrypto - rsync friendly encryption

20.3.5. Команда apt-get install

Вы можете установить одно или несколько приложений, разместив имена в строке команды после apt-get install. В примере ниже показан процесс установки пакета программного обеспечения с именем rsync.

root@ubu1204~# apt-get install rsync
Чтение списков пакетов... Готово
Построение дерева зависимостей
Чтение информации о состоянии... Готово
НОВЫЕ пакеты, которые будут установлены:
rsync
обновлено 0, установлено 1 новых пакетов, для удаления отмечено 0 пакетов, и 8 пакетов не обновлено.
Необходимо скачать 299 кБ архивов.
После данной операции, объём занятого дискового пространства возрастёт на 634 kB.
Получено:1 http://be.archive.ubuntu.com/ubuntu/ precise/main rsync i386 3.0.9-1ubuntu1 [299 kB]
Получено 299 кБ за 0с (740 кБ/с)
Выбор ранее не выбранного пакета rsync.
(Чтение базы данных ... на данный момент установлено 323649 файлов и каталогов.)
Распаковывается пакет rsync (из файла .../rsync_3.0.9-1ubuntu1_i386.deb) ...
Обрабатываются триггеры для man-db ...
Обрабатываются триггеры для ureadahead ...
Настраивается пакет (3.0.9-1ubuntu1) ...
Removing any system startup links for /etc/init.d/rsync ...
root@ubu1204~#

20.3.6. Команда apt-get remove

Вы можете удалить одно или несколько приложений, разместив их имена в строке команды после apt-get remove. В примере ниже показан процесс удаления пакета программного обеспечения с именем rsync.

root@ubu1204~# apt-get remove rsync
Чтение списков пакетов... Готово
Построение дерева зависимостей
Чтение информации о состоянии... Готово
Пакеты, которые будут УДАЛЕНЫ:
rsync ubuntu-standard
обновлено 0, установлено 0 новых пакетов, для удаления отмечено 2 пакетов, и 8 пакетов не обновлено.
После данной операции, объём занятого дискового пространства уменьшится на 692 kB.
Хотите продолжить [Д/н]? y
(Чтение базы данных ... на данный момент установлено 323681 файлов и каталогов.)
Удаляется пакет ubuntu-standard ...
Удаляется пакет rsync ...
- Stopping rsync daemon rsync [OK]
  Обрабатываются триггеры для ureadahead ...

Обрабатываются триггеры для man-db ...
root@ubu1204~#
Обратите внимание на то, что при удалении пакета с помощью рассматриваемой команды некоторые конфигурационные денные не удаляются.
root@ubu1204~# dpkg -l rsync | tail -1 | tr -s ' '
rc rsync 3.0.9-1ubuntu1 fast, versatile, remote (and local) file-copying tool

20.3.7. Команда apt-get purge

Вы можете удалить полностью одно или несколько приложений, разместив их имена в строке команды после apt-get purge. В ходе полного удаления приложения будут также удаляться все существующие файлы конфигурации, имеющие отношение к данному приложению. В примере ниже показан процесс полного удаления пакета программного обеспечения с именем rsync.

root@ubu1204~# apt-get purge rsync
Чтение списков пакетов... Готово
Построение дерева зависимостей
Чтение информации о состоянии... Готово
Пакеты, которые будут УДАЛЕНЫ:
rsync*
обновлено 0, установлено 0 новых пакетов, для удаления отмечено 1 пакетов, и 8 пакетов не обновлено.
После данной операции, объём занятого дискового пространства уменьшится на 0 kB.
Хотите продолжить [Д/н]? y
(Чтение базы данных ... на данный момент установлено 323651 файлов и каталогов.)
Удаляется пакет rsync ...
Вычищаются файлы настройки пакета rsync ...
Обрабатываются триггеры для ureadahead ...
root@ubu1204~#
Обратите внимание на то, что утилита dpkg не обладает информацией о полностью удаленном пакете помимо той, что

пакет был удален и в системе не осталось файлов конфигурации.

root@ubu1204~# dpkg -l rsync | tail -1 | tr -s ' '
un rsync <нет> (описание недоступно)

20.4. Утилита aptitude

Большинство пользователей применяет утилиту aptitude для управления пакетами программного обеспечения в дистрибутивах Debain, Mint и Ubuntu.

Команда для синхронизации списков программного обеспечения с репозиториями:

aptitude update

Команда для обновления всех программных компонентов до последних версий в дистрибутиве Debian.

aptitude upgrade

Команда для обновления всех программных компонентов до последних версий в дистрибутивах Ubuntu и Mint.

aptitude safe-upgrade

Команда для установки приложения со всеми зависимостями:

aptitude install $имя_пакета

Команда для поиска в репозиториях приложений, имена или описания которых содержат указанную строку:

aptitude search $строка

Команда для удаления приложения:

aptitude remove $имя_пакета

Команда для удаления приложения вместе со всеми его файлами конфигурации:

aptitude purge $имя_пакета

20.5. Директория apt

И утилита apt-get, и утилита aptitude используют одни и те же файлы конфигурации, расположенные в директории /etc/apt. Следовательно, в случае добавления репозитория для установки программного обеспечения с помощью одной из упомянутых утилит, вы сможете устанавливать программное обеспечение из добавленного репозитория и с помощью другой утилиты.

20.5.1. Файл конфигурации /etc/apt/sources.list

Список ресурсов, используемый утилитами apt-get и aptitude, расположен в файле /etc/apt/sources.list. Данный файл содержит строки URL для доступа по протоколам http и ftp к ресурсами, с которых могут быть получены пакеты для используемого дистрибутива.

На моем сервере, работающем под управлением дистрибутива Debian, данный файл выглядит следующим образом:

root@debian6:~# cat /etc/apt/sources.list
deb http://ftp.be.debian.org/debian/ squeeze main
deb-src http://ftp.be.debian.org/debian/ squeeze main
deb http://security.debian.org/ squeeze/updates main
deb-src http://security.debian.org/ squeeze/updates main
squeeze-updates, previously known as volatile
deb http://ftp.be.debian.org/debian/ squeeze-updates main
deb-src http://ftp.be.debian.org/debian/ squeeze-updates main

На моем настольном компьютере, работающем под управлением дистрибутива Ubuntu, используется в четыре раза больше сетевых репозиториев.

root@ubu1204~# wc -l /etc/apt/sources.list
63 /etc/apt/sources.list

Ввиду того, что в рамках данной книги не рассматриваются некоторые аспекты работы со списками репозиториев apt, вам следует разобраться с ними самостоятельно, обратив особое внимание на такие команды, как add-apt-repository, apt-key и aprops apt.

20.6. Управление пакетами программного обеспечения формата rpm

20.6.1. О пакетах программного обеспечения формата rpm

Менеджер пакетов программного обеспечения Red Hat (Red Hat package manager) может использоваться посредством утилиты rpm с интерфейсом командной строки в терминале или посредством приложения с графическим интерфейсом, вызываемого с помощью пункта меню «Приложения — Системные настройки — Установка/Удаление приложений». Выполните команду rpm --help для ознакомления с некоторыми параметрами утилиты.

При формировании имен файлов пакетов программного обеспечения формата rpm используется следующий формат: название_программного_компонента-версия.платформа.rpm.

20.6.2. Команда rpm -qa

Для получения списка установленных пакетов программного обеспечения следует использовать команду rpm -qa.

[root@RHEL52 ~]# rpm -qa | grep samba
system-config-samba-1.2.39-1.el5
samba-3.0.28-1.el5_2.1
samba-client-3.0.28-1.el5_2.1
samba-common-3.0.28-1.el5_2.1

20.6.3. Команда rpm -q

Чтобы убедиться в том, что определенный пакет программного обеспечения был установлен, вы можете использовать команду rpm -q.

root@RHELv4u4:~# rpm -q gcc
gcc-3.4.6-3
root@RHELv4u4:~# rpm -q laika

пакет laika не установлен

20.6.4. Команда rpm -Uhv

Для установки или обновления пакета программного обеспечения следует использовать параметры -Uvh рассматриваемой утилиты. Параметр -U аналогичен параметру -i, предназначенному для установки пакета программного обеспечения, но в случае его использования более старые версии пакетов программного обеспечения будут удаляться. Параметры -vh предназначены для активации режима вывода, более удобного для чтения.

root@RHELv4u4:~# rpm -Uvh gcc-3.4.6-3

20.6.5. Команда rpm -e

Для удаления пакета программного обеспечения следует использовать параметр -e рассматриваемой утилиты.

root@RHELv4u4:~# rpm -e gcc-3.4.6-3

В случае использования команды rpm -e утилита rpm будет осуществлять проверку зависимостей, предотвращая тем самым возможность случайного удаления пакетов программного обеспечения, которые требуются для корректной работы программных компонентов из других пакетов программного обеспечения.

[root@RHEL52 ~]# rpm -e gcc-4.1.2-42.el5
ошибка: Неудовлетворенные зависимости:
gcc = 4.1.2-42.el5 нужен для (установлен)gcc-c++-4.1.2-42.el5.i386
gcc = 4.1.2-42.el5 нужен для (установлен)gcc-gfortran-4.1.2-42.el5.i386
gcc нужен для (установлен)systemtap-0.6.2-1.el5_2.2.i386

20.6.6. Директория /var/lib/rpm

База данных менеджера пакетов программного обеспечения Red Hat расположена в директории /var/lib/rpm. Эта база данных содержит все метаданные, относящиеся к пакетам программного обеспечения, которые были установлены (с помощью утилиты rpm). В ней хранится информация обо всех файлах из пакетов программного обеспечения, благодаря наличию которой имеется возможность полного удаления программного обеспечения.

20.6.7. Утилита rpm2cpio

Мы можем использовать утилиту rpm2cpio для преобразования пакета программного обеспечения формата rpm в архив формата cpio.

[root@RHEL53 ~]# file kernel.src.rpm
kernel.src.rpm: RPM v3 src PowerPC kernel-2.6.18-92.1.13.el5
[root@RHEL53 ~]# rpm2cpio kernel.src.rpm > kernel.cpio
[root@RHEL53 ~]# file kernel.cpio
kernel.cpio: ASCII cpio archive (SVR4 with no CRC)

Но для чего это может понадобиться?

Возможно, просто для того, чтобы ознакомиться со списком имен файлов, хранящихся в пакете программного обеспечения формата rpm.

[root@RHEL53 ~]# rpm2cpio kernel.src.rpm | cpio -t | head -5
COPYING.modules
Config.mk
Module.kabi_i686
Module.kabi_i686PAE
Module.kabi_i686xen

Или для извлечения какого-либо файла из пакета программного обеспечения формата rpm.

[root@RHEL53 ~]# rpm2cpio kernel.src.rpm | cpio -iv Config.mk
Config.mk
246098 блоков

20.7. Утилита yum

20.7.1. Информация об утилите yum

Модифицированный инструмент для обновления дистрибутива Yellowdog Linux (Yellowdog Updater, Modified — yum) является утилитой с интерфейсом командной строки, предназначенной для упрощения работы с пакетами программного обеспечения формата rpm. Данная утилита устанавливается по умолчанию в процессе установки таких дистрибутивов, как Fedora и Red Hat Enterprise Linux начиная с версии 5.2.

20.7.2. Команда yum list

Выполните команду yum list available для получения списка доступных для установки пакетов программного обеспечения. Параметр available является необязательным.

root@rhel65:/etc# yum list | wc -l
This system is receiving updates from Red Hat Subscription Management.
3935
root@rhel65:/etc#

Выполните команду yum list $имя_пакета для получения списка всех доступных версий указанного пакета программного обеспечения (из различных репозиториев).

[root@rhel55 ~]# yum list samba
Loaded plugins: rhnplugin, security
Установленные пакеты
samba.i386 3.0.33-3.28.el5 installed
Доступные пакеты
samba.i386 3.0.33-3.29.el5_5 rhel-i386-server-5

20.7.3. Команда yum search

Для осуществления поиска пакета, имя или описание которого

содержит заданную строку, следует использовать команду yum search $строка.

[root@rhel55 ~]# yum search gcc44
Loaded plugins: rhnplugin, security
========== N/S matched: gcc44 ==========
gcc44.i386 : Preview of GCC version 4.4
gcc44-c.i386 : C support for GCC version 4.4
gcc44-gfortran.i386 : Fortran support for GCC 4.4 previe

20.7.4. Команда yum provides

Для поиска пакета программного обеспечения, содержащего файл с определенным именем (что может понадобиться в процессе компиляции программного обеспечения из исходных кодов), следует использовать команду yum provides $имя_файла.

root@rhel65:/etc# yum provides /usr/share/man/man5/passwd.5.gz
Loaded plugins: product-id, subscription-manager
This system is receiving updates from Red Hat Subscription Management.
rhel-6-server-cf-tools-1-rpms | 2.8 kB 00:00
rhel-6-server-rpms | 3.7 kB 00:00
man-pages-3.22-12.el6.noarch : Man (manual) pages from the Linux Documenta...
Источник : rhel-6-server-rpms
Совпадения с:
Имя файла : /usr/share/man/man5/passwd.5.gz
man-pages-3.22-20.el6.noarch : Man (manual) pages from the Linux Documenta...
Источник : rhel-6-server-rpms
Совпадения с:
Имя файла : /usr/share/man/man5/passwd.5.gz
man-pages-3.22-17.el6.noarch : Man (manual) pages from the Linux Documenta...
Источник : rhel-6-server-rpms
Совпадения с:
Имя файла : /usr/share/man/man5/passwd.5.gz
man-pages-3.22-20.el6.noarch : Man (manual) pages from the Linux Documenta...
Источник : installed
Совпадения с:
Имя файла : Provides-match: /usr/share/man/man5/passwd.5.gz
root@rhel65:/etc#

20.7.5. Команда yum install

Для установки приложения следует использовать команду yum install $имя_пакета. Утилита yum позаботится об установке всех необходимых зависимостей.

[root@rhel55 ~]# yum install sudo
Loaded plugins: rhnplugin, security
Setting up Install Process
Разрешение зависимостей
-→ Проверка сценария
--→ Пакет sudo.i386 0:1.7.2p1-7.el5_5 помечен для установки
-→ Проверка зависимостей окончена
Зависимости определены
Package Архитектура Версия Репозиторий Размер
Установка:
sudo i386 1.7.2p1-7.el5_5 rhel-i386-server-5 230 k
Итого за операцию
Установить 1 пакет
Объем загрузки: 230 k
Is this ok [y/N]: y
Downloading Packages:
sudo-1.7.2p1-7.el5_5.i386.rpm | 230 kB 00:00
Running rpm_check_debug
Running Transaction Test
Finished Transaction Test
Transaction Test Succeeded
Running Transaction
Установка : sudo 1/1
Установлено:
sudo.i386 0:1.7.2p1-7.el5_5
Выполнено!

В рамках данной команды вы можете использовать более одного параметра.

yum install $имя_пакета1 $имя_пакета2 $имя_пакета3

20.7.6. Команда yum update

Для обновления всех приложений путем загрузки и установки новых пакетов программного обеспечения, следует использовать команду yum update. Все программное обеспечение, которое было установлено в помощью утилиты yum, будет обновлено до последних версий, доступных в репозитории.

yum update

Если вы желаете обновить только один пакет программного обеспечения, вы можете использовать команду yum update $имя_пакета.

[root@rhel55 ~]# yum update sudo
Loaded plugins: rhnplugin, security
Skipping security plugin, no data
Setting up Update Process

Разрешение зависимостей
Skipping security plugin, no data
--> Проверка сценария
---> Пакет sudo.i386 0:1.7.2p1-7.el5_5 помечен как обновление
--> Проверка зависимостей окончена
Dependencies Resolved
Package Архитектура Версия Репозиторий Размер
Обновление:
sudo i386 1.7.2p1-7.el5_5 rhel-i386-server-5 230 k
Итого за операцию
Обновить 1 пакет
Объем загрузки: 230 k
Is this ok [y/N]: y
Downloading Packages:
sudo-1.7.2p1-7.el5_5.i386.rpm | 230 kB 00:00
Running rpm_check_debug
Running Transaction Test
Finished Transaction Test
Transaction Test Succeeded
Running Transaction
Обновление : sudo 1/2
Очистка : sudo 2/2
Обновлено:
sudo.i386 0:1.7.2p1-7.el5_5
Выполнено!

20.7.7. Группы программного обеспечения yum

Выполните команду yum grouplist для ознакомления со списком всех доступных групп программного обеспечения.

[root@rhel55 ~]# yum grouplist
Loaded plugins: rhnplugin, security
Setting up Group Process
Installed Groups:
Administration Tools
Authoring and Publishing
DNS Name Server
Development Libraries
Development Tools
Editors
GNOME Desktop Environment
GNOME Software Development
Graphical Internet
Graphics
Legacy Network Server
Legacy Software Development
Legacy Software Support
Mail Server
Network Servers
Office/Productivity
Printing Support
Server Configuration Tools
System Tools
Text-based Internet
Web Server
Windows File Server
X Software Development
X Window System
Available Groups:
Engineering and Scientific
FTP Server
Games and Entertainment
Java Development
KDE (K Desktop Environment)
KDE Software Development
MySQL Database
News Server
OpenFabrics Enterprise Distribution
PostgreSQL Database
Sound and Video
Done

Для установки приложений, объединенных в рамках группы, следует использовать команду yum groupinstall $имя_группы.

[root@rhel55 ~]# yum groupinstall Sound and video
Loaded plugins: rhnplugin, security
Setting up Group Process
Пакет alsa-utils-1.0.17-1.el5.i386 уже установлен и это последняя версия
Пакет sox-12.18.1-1.i386 уже установлен и это последняя версия
Пакет 9:mkisofs-2.01-10.7.el5.i386 уже установлен и это последняя версия
Пакет 9:cdrecord-2.01-10.7.el5.i386 уже установлен и это последняя версия
Пакет cdrdao-1.2.1-2.i386 уже установлен и это последняя версия
Разрешение зависимостей
--> Проверка сценария
---> Пакет cdda2wav.i386 9:2.01-10.7.el5 помечен для установки
---> Пакет cdparanoia.i386 0:alpha9.8-27.2 помечен для установки
---> Пакет sound-juicer.i386 0:2.16.0-3.el5 помечен для установки
--> Обработка зависимостей: libmusicbrainz >= 2.1.0 пакета:

sound-juicer
--> Обработка зависимостей: libmusicbrainz.so.4 пакета: sound-juicer
---> Пакет vorbis-tools.i386 1:1.1.1-3.el5 помечен для установки
--> Обработка зависимостей: libao >= 0.8.4 пакета: vorbis-tools
--> Обработка зависимостей: libao.so.2 пакета: vorbis-tools
--> Проверка сценария
---> Пакет libao.i386 0:0.8.6-7 помечен для установки
---> Пакет libmusicbrainz.i386 0:2.1.1-4.1 помечен для установки
--> Проверка зависимостей окончена
...

Обратитесь к странице руководства утилиты yum для получения дополнительной информации об управлении группами программного обеспечения средствами yum.

20.7.8. Файл конфигурации /etc/yum.conf и репозитории

Данные для доступа к репозиториям, которые используются утилитой yum, приводится в рамках файла конфигурации /etc/yum/yum.conf и файлов описания репозиториев из директории /etc/yum/repos.d/.

Конфигурация утилиты yum осуществляется в рамках файла конфигурации /etc/yum.conf. Данный файл должен содержать путь к файлу журнала и директории для кэширования данных, загружаемых утилитой yum, а также может содержать данные для доступа к репозиториям.

Не так давно утилита yum получила возможность обработки множества файлов описания репозиториев с расширением .repo, каждый из которых может содержать данные для доступа к репозиториям. Эти файлы с расширением .repo должны быть помещены в директорию /etc/yum.repos.d/.

Одним из наиболее важных флагов утилиты yum является флаг enablerepo. Используйте следующую команду, если вы желаете использовать репозиторий, который не активирован по умолчанию.

yum $операция $имя_пакета --enablerepo=$имя_репозитория

В примере приведено необходимое содержимое файла описания репозитория с расширением .repo: MyRepo.repo

name=Мой репозиторий
baseurl=http://путь/к/репозиторию
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-MyRep

20.8. Утилита alien

Утилита alien является экспериментальным инструментом, который предназначен для преобразования форматов пакетов rpm и deb (и других).

Ниже приведен пример использования утилиты alien для преобразования пакета программного обеспечения формата rpm в пакет программного обеспечения формата deb.

paul@barry:~$ ls -l netcat*
-rw-r—r-- 1 paul paul 123912 2009-06-04 14:58 netcat-0.7.1-1.i386.rpm
paul@barry:~$ alien --to-deb netcat-0.7.1-1.i386.rpm
netcat_0.7.1-2_i386.deb generated
paul@barry:~$ ls -l netcat*
-rw-r—r-- 1 paul paul 123912 2009-06-04 14:58 netcat-0.7.1-1.i386.rpm
-rw-r—r-- 1 root root 125236 2009-06-04 14:59 netcat_0.7.1-2_i386.deb

В реальной жизни следует использовать пакет с утилитой netcat из репозитория вашего дистрибутива или пакет программного обеспечения формата deb с вебсайта проекта.

20.9. Загрузка программного обеспечения в обход репозитория дистрибутива

При загрузке любого программного обеспечения в обход репозитория дистрибутива первым и наиболее важным шагом является чтение поставляемого в комплекте файла README!

Обычно в файле README даются пояснения относительно ваших действий после загрузки пакета программного обеспечения. С большой вероятностью для загрузки вам будет предложен файл архива с расширением .tar.gz или .tgz. Прочитайте документацию, после чего переместите загруженный файл архива в отдельную директорию. Вы можете использовать следующую команду, чтобы выяснить, в какую директорию по умолчанию будет распакован архив.

tar tvzpf $загруженный_файл.tgz

Если вы распакуете данный архивный файл с помощью команды tar xzf, будет создана директория с именем $имя_приложения-версия.

tar xzf $applicationName.tgz

Замените параметр z на j в том случае, если файл архива имеет расширение .tar.bz2. Приемы работы с утилитами tar, gzip и bzip2 подробно описаны в книге «Фундаментальные основы Linux».

Если вы загрузили файл с расширением .deb, для установки программного обеспечения придется использовать утилиту dpkg, а в случае загрузки файла с расширением .rpm установка программного обеспечения может осуществляться с помощью утилиты rpm.

20.10. Компиляция программного обеспечения из исходного кода

При загрузке исходного кода любого программного компонента с

целью последующей компиляции и установки первым и наиболее важным шагом является чтение поставляемого в комплекте файла README!

Чаще всего процесс сборки программного обеспечения из исходного кода может быть разделен на три шага: запуск сценария конфигурации ./configure с последующим запуском процесса сборки make (которая по сути и является компиляцией) и выполнением команды установки make install для копирования результирующих файлов в необходимые директории файловой системы.

/configure
make
make install

20.11. Практическое задание: управление пакетами программного обеспечения

1. Проверьте, установлены ли пакеты программного обеспечения с именами gcc, sudo и wesnoth в вашей системе.
2. Используйте утилиту yum или aptitude для поиска и установки пакетов программного обеспечения с именами scp, tmux и man-pages. Удалось ли найти в репозитории вашего дистрибутива все перечисленные пакеты.
3. Найдите в сети Интернет информацию о программном обеспечении webmin и выясните способ установки этого программного обеспечения в систему.
4. Если есть время, найдите пакеты программного обеспечения samba, а также пакеты с документацией проекта samba в формате pdf (тысячи страниц документации в рамках двух файлов pdf).

20.12. Корректная процедура выполнения практического задания: управление пакетами программного обеспечения

1. Проверьте, установлены ли пакеты программного обеспечения с именами gcc, sudo и wesnoth в вашей системе.

В дистрибутиве RedHat/CentOS:
rpm -qa | grep gcc
rpm -qa | grep sudo
rpm -qa | grep wesnoth
В дистрибутиве Debian/Ubuntu:
dpkg -l | grep gcc
dpkg -l | grep sudo
dpkg -l | grep wesnoth

1. Используйте утилиту yum или aptitude для поиска и установки пакетов программного обеспечения с именами scp, tmux и man-pages. Удалось ли найти в репозитории вашего дистрибутива все перечисленные пакеты.

В дистрибутиве RedHat/CentOS:
yum search scp
yum search tmux
yum search man-pages
В дистрибутиве Debian/Ubuntu:
aptitude search scp
aptitude search tmux
aptitude search man-pages

1. Найдите в сети Интернет информацию о программном обеспечении webmin и выясните способ установки этого программного обеспечения в систему.

Поисковая система Google приведет вас на вебсайт webmin.com.

С данного вебсайта могут быть загружены пакеты программного обеспечения различных форматов, поэтому в зависимости от дистрибутива следует выбрать файл с расширением .rpm, .deb или .tgz.

1. Если есть время, найдите пакеты программного обеспечения samba, а также пакеты с документацией проекта samba в формате pdf (тысячи страниц документации в рамках двух файлов формата pdf).

Глава 21. Общая информация о сетях

Несмотря на то, что в данной главе не рассматриваются вопросы администрирования систем Linux, она содержит общую информацию о сетях, которая может оказаться полезной при диагностике ваших сетей, связывающих работающие под управлением Linux компьютеры.

21.1. Уровни сетевых моделей

21.1.1. Семь уровней сетевой модели OSI

При разговоре об уровнях сетевых моделей чаще всего упоминаются семь уровней сетевой модели OSI (прикладной уровень (application layer), уровень представления (presentation layer), сеансовый уровень (session layer), транспортный уровень (transport layer), сетевой уровень (network layer), канальный уровень (data link layer) и физический уровень (physical layer)). В данной главе мы будем подробно обсуждать только уровни 2 и 3 и ограничимся краткими описаниями остальных уровней данной модели. Такое разделение приоритетов объясняется тем, что понимание назначения перечисленных уровней является наиболее важным для понимания принципов работы сетей. Вы можете услышать от администраторов сетей такие описания, как «это устройство, работающее на 2 уровне» или «это

широковещательная передача данных на 3 уровне» и вы должны понимать, о чем идет речь.

21.1.2. Четыре уровня сетевой модели DoD

Сетевая модель DoD (или сетевая модель TCP/IP) состоит всего из четырех уровней и, грубо говоря, ее уровень сетевого доступа (network access layer) может быть поставлен в соответствие уровням 1 и 2 сетевой модели OSI (физическому и канальному уровням), межсетевой уровень (internet layer, IP) — сетевому уровню сетевой модели OSI, транспортный уровень (host-to-host layer, TCP/UDP) - уровню 4 сетевой модели OSI (транспортному уровню), а уровень приложений (application layer) — уровням 5, 6 и 7 сетевой модели OSI.

Ниже приведена таблица соответствия уровней сетевых моделей OSI и DoD с примерами некоторых сетевых протоколов и приложений/устройств.

<table>
<tr><th>Модель OSI</th><th>Модель DoD</th><th colspan="2">Протоколы</th><th>Устройства/прилож.</th></tr>
<tr><td>уровни 5, 6, 7</td><td>уровень приложений</td><td colspan="2">dns, dhcp, ntp, snmp, https, ftp,ssh, telnet, http, pop3 и другие</td><td>веб-сервер, почтовый сервер, браузер, почтовый клиент</td></tr>
<tr><td>уровень 4</td><td>транспортный уровень</td><td>tcp</td><td>udp</td><td>сетевой шлюз</td></tr>
<tr><td>уровень 3</td><td>межсетевой уровень</td><td colspan="2">ip, icmp, igmp</td><td>маршрутизатор, межсетевой экран, коммутатор уровня 3</td></tr>
<tr><td>уровень 2</td><td rowspan="2">уровень сетевого доступа</td><td colspan="2">arp (mac), rarp</td><td>мост, коммутатор уровня 2</td></tr>
<tr><td>уровень 1</td><td colspan="2">ethernet, token ring</td><td>хаб</td></tr>
</table>

21.1.3. Краткое описание физического уровня сетевой модели OSI

На физическом уровне или уровне 1 сетевой модели OSI рассматриваются напряжения, электрические сигналы и механические соединения. Для построения некоторых сетей все еще могут использоваться коаксиальные кабели, но большая часть сетей переведена на использование кабелей UTP (категории 5 и выше) с коннекторами RJ45.

На данном уровне сетевой модели рассматриваются такие устройства, как повторители и хабы. У вас нет возможности обнаружения в сети повторителя или хаба с помощью специального программного обеспечения. Единственной задачей упомянутых устройств является усиление электрических сигналов, проходящих по кабелям. Пассивные хабы являются усилителями с множеством портов, которые усиливают входящий электрический сигнал и передают его через все соединенные с выходными портами кабели. Активные хабы выполняют аналогичную задачу путем приема битов данных и их повторной передачи без какой-либо интерпретации этих битов.

На данном уровне рассматриваются такие сетевые технологии, как CSMA/CD и Token ring.

Это все, что следует сказать об уровне 1 сетевой модели OSI в рамках данной книги.

21.1.4. Краткое описание канального уровня сетевой модели OSI

На канальном уровне или уровне 2 сетевой модели OSI рассматриваются фреймы данных. Каждый фрейм данных содержит поле CRC (cyclic redundancy check - циклический избыточный код). В рамках протокола Ethernet (802.3) каждая сетевая карта идентифицируется с помощью уникального 48-битного MAC-адреса (media access control address - адрес доступа к среде).

На данном уровне мы можем обнаружить упоминания о таких устройствах, как мосты и коммутаторы. Мост является более сложным устройством, чем хаб, ведь он может принимать решения на основе MAC-адресов компьютеров из сети. Коммутатор также обрабатывает MAC-адреса.

В данной книге с целью более подробного исследования данного уровня сетевой модели мы будем обсуждать такие утилиты, как arp и ifconfig.

21.1.5. Краткое описание сетевого уровня сетевой модели OSI

На уровне 3 сетевой модели OSI рассматриваются пакеты IP. На данном уровне сетевой модели каждому устройству выдается уникальный 32-битный IP-адрес. Но протокол IP не является

единственным протоколом данного уровня сетевой модели, ведь существуют и такие протоколы данного уровня, как ICMP, IGMP, IPv6 и другие. Полный список протоколов находится в файле /etc/protocols.

На данном уровне мы можем обнаружить упоминания о таких устройствах, как маршрутизаторы и коммутаторы уровня 3 сетевой модели, причем данные устройства обрабатывают (и имеют) IP-адреса.

В случае использования стека протоколов TCP/IP данный уровень обычно называют межсетевым уровнем (intenet layer).

21.1.6. Краткое описание транспортного уровня сетевой модели OSI

Мы будем обсуждать протоколы TCP и UDP в контексте уровня 4 сетевой модели OSI. В рамках сетевой модели DoD данный уровень называется транспортным уровнем.

21.1.7. Об уровнях 5, 6 и 7 сетевой модели OSI

Приложение, работающее со стеком протоколов TCP/IP, работает на уровнях 5, 6 и 7 сетевой модели OSI. Описание различий между этими уровнями сетевой модели выходит за рамки данного курса.

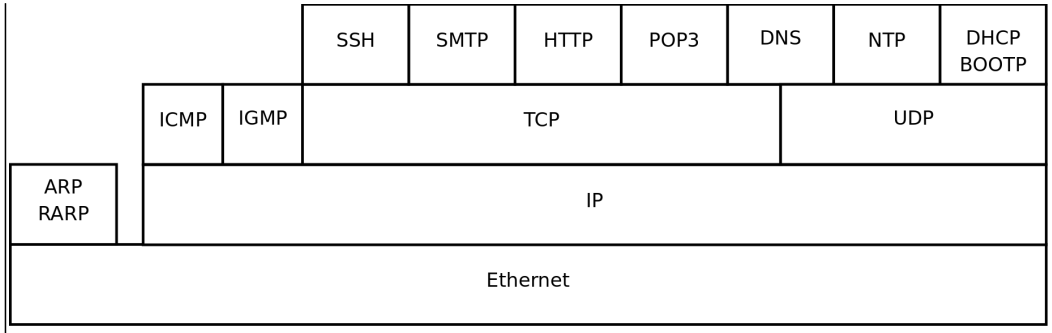
21.1.8. Уровни сетевой модели OSI в рамках данной книги

Описание наложения уровней сетевой модели OSI в рамках данной книги базируется на описании протоколов фрейма (protocols in frame) сниффера wireshark. При захвате пакета DHCP мы увидим следующую информацию в окне сниффера.

При захвате пакетов NTP (Network Time Protocol — сетевой протокол для синхронизации часов) мы получим следующую строку, на основе которой можно сделать вывод о том, что протокол NTP должен быть размещен рядом с протоколом BOOTP на приведенной ниже схеме.

После захвата пакетов ARP, распространяемых с помощью широковещательной передачи, можно сделать вывод о том, что протокол ARP должен быть размещен рядом с протоколом IP на приведенной ниже схеме. Все упомянутые протоколы будут рассматриваться более подробно в рамках данной главы.

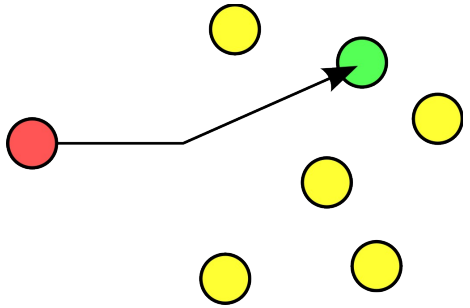
Ниже приведена схема распределения сетевых протоколов, составленная на основе анализа данных, предоставленных сниффером wireshark. Она содержит некоторые чрезвычайно часто используемые протоколы, которые будут обсуждаться в рамках данной книги. Разумеется, на данной схеме не представлены все существующие сетевые протоколы.



21.2. Одноадресная, многоадресная, широковещательная и произвольная передача данных

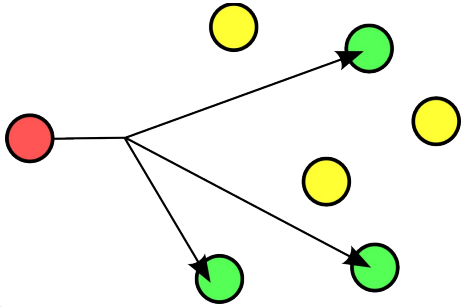
21.2.1. Одноадресная передача данных

Одноадресная передача данных (unicast) осуществляется компьютером в том случае, если требуется передать данные только одному другому компьютеру (или узлу). Нередко компьютеры осуществляют большое количество одноадресных передач данных одновременно.



21.2.2. Многоадресная передача данных

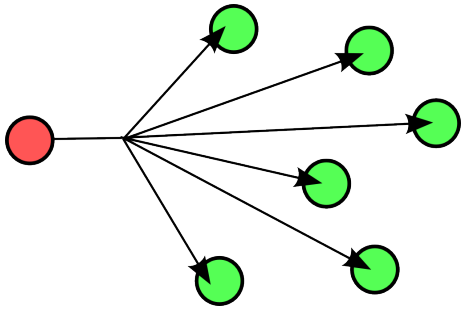
Многоадресная передача данных (multicast) предназначена для доставки одних и тех же данных группе компьютеров.



В качестве примеров использования многоадресной передачи данных можно привести приложение Realpalyer (работающее с файлами с расширением .sdp) и протокол RIPv2 (протокол маршрутизации).

21.2.3. Широковещательная передача данных

Широковещательная передача данных (broadcast) предназначена для доставки данных всем устройствам в сети.

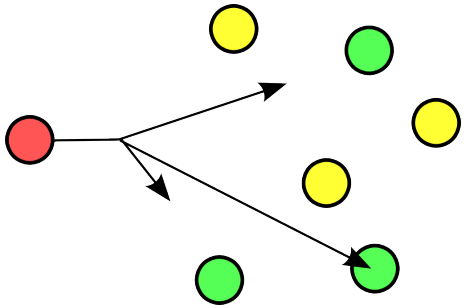


В качестве примера использования данного типа передачи данных можно привести общедоступные трансляции BBC (British Broadcasting Corporation — Британская вещательная корпорация). Широковещательная передача данных чаще всего осуществляется в рамках локальной сети.

Следует помнить о том, что описание механизма широковещательной передачи данных на уровне 2 сетевой модели OSI кардинально отличается от описания этого же механизма на уровне 3 сетевой модели OSI. На уровне 2 благодаря широковещательной передаче данные доставляются всем сетевым картам в одном сегменте сети (который не распространяется за рамки маршрутизаторов), в то время, как на уровне 3 данные принимаются всеми узлами, расположенными в одной подсети.

21.2.4. Произвольная передача данных

Корневые серверы имен сети Интернет используют механизм произвольной передачи данных (anycast). В случае произвольной передачи данных данные доставляются наиболее близко расположенному (географически) узлу из четко установленной группы узлов.



Благодарю анонимного пользователя энциклопедии Wikipedia, который создал приведенные выше иллюстрации и опубликовал их как публичное достояние.

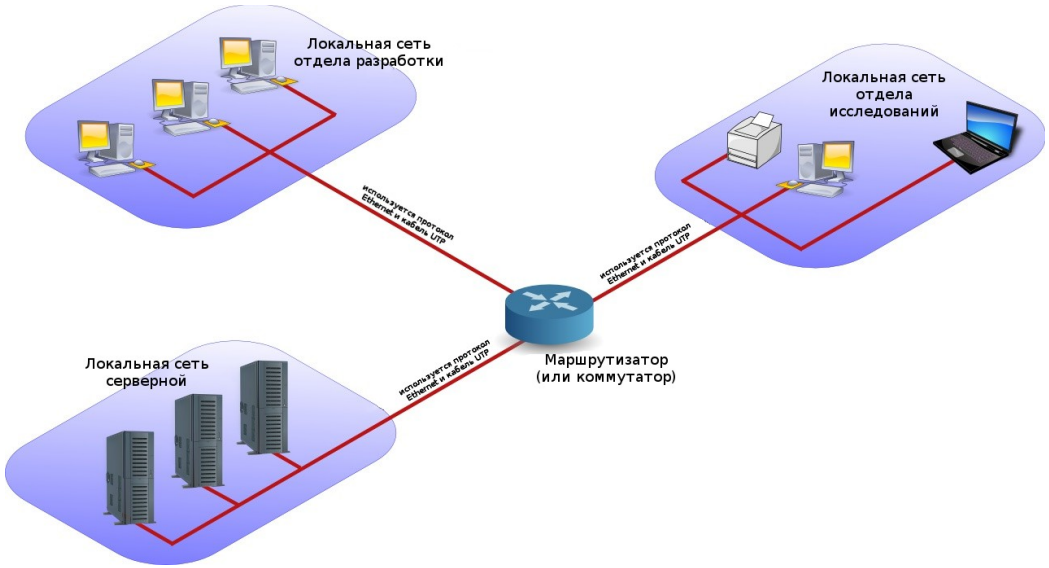
21.3. Локальные, глобальные и региональные сети

Аббревиатура LAN используется для обозначения локальных сетей (local area networks) в противоположность аббревиатуре WAN, используемой для обозначения глобальных сетей (wide area networks). Разница между двумя упомянутыми типами сетей заключается в дистанции между подключенными к ним компьютерами, а не в их количестве. Некоторые протоколы, такие, как ATM разработаны специально для глобальных сетей, другие, такие, как Ethernet — для локальных сетей.

21.3.1. Локальная сеть

Аббревиатура LAN (Local Area Network) используется для обозначения локальных сетей. Сеть данного типа может быть развернута в рамках одной комнаты, одного этажа здания или даже одного большого здания. Мы говорим о локальной сети тогда, когда подключенные к сети компьютеры расположены близко друг к другу. Также вы можете считать локальной сетью сеть, в которой все компьютеры соединены друг с другом по протоколу Ethernet.

Локальная сеть может состоять из множества локальных сетей меньших размеров. На рисунке ниже изображены три малых локальных сети, которые образуют одну большую локальную сеть.



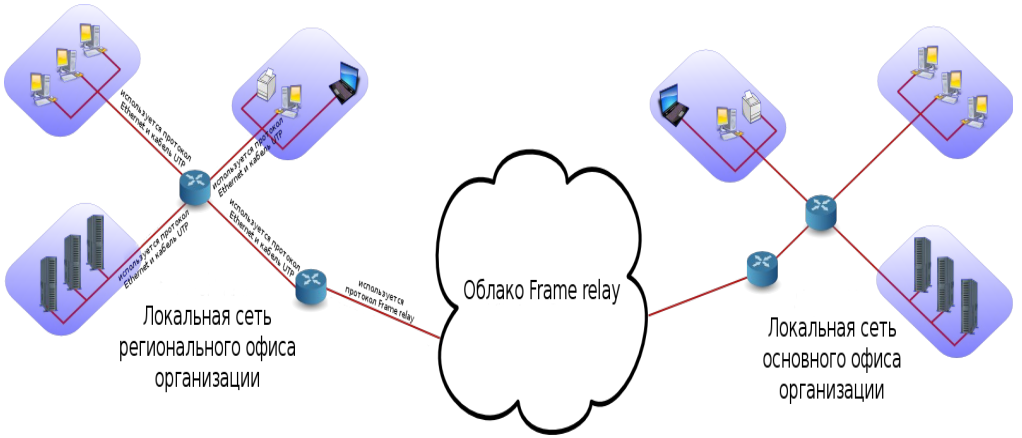
21.3.2. Региональная сеть

Обозначаемая с помощью аббревиатуры MAN (Metropolitan Area Network) региональная сеть является промежуточным вариантом между локальной и глобальной сетями и обычно охватывает несколько зданий в одном и том же кампусе или городе. Для организации региональной сети могут использоваться протоколы FDDI, Ethernet или другие протоколы.

21.3.3. Глобальная сеть

Обозначаемая с помощью аббревиатуры WAN (Wide Area Network) глобальная сеть является сетью с большими дистанциями между компьютерами (или узлами). Эти узлы обычно соединяются с помощью выделенных каналов для передачи данных. При организации глобальной сети для соединения компьютеров (и сетей) используется не протокол Ethernet, а такие протоколы, как FDDI, Frame Relay, ATM и X.25.

На рисунке ниже показана локальная сеть регионального офиса, которая соединена по протоколу Frame Relay с основным офисом организации.



Акроним WAN также используется для обозначения таких сетей со значительной областью покрытия, как сеть Интернет.

Также широко известна технология WAN, разработанная компанией Cisco. Эта компания выпускает маршрутизаторы, которые объединяют множество локальных сетей по протоколу WAN.

21.3.4. Персональная сеть

Для обозначения вашей домашней сети используется аббревиатура PAN (Personal Area Network). Персональная беспроводная сеть обозначается с помощью аббревиатуры WPAN.

21.4. Интернет, интранет и экстранет

Интернет является глобальной сетью. Эта сеть объединяет множество сетей, использующих стек протоколов TCP/IP.

Основой сети Интернет являлась сеть Arpanet. Сеть Arpanet была создана в 1969 году, причем в этом году к сети были подключены всего четыре компьютера. В 1971 году посредством сети Arpanet было отправлено первое сообщение электронной почты. Трафик сообщений электронной почты составлял 75 процентов от всего трафика сети Arpanet в 1973 году. Кроме того, в 1973 году был представлен протокол FTP, а также осуществлено подключение к сети первых европейских стран (Норвегии и Великобритании). По информации на 2009 год возможностью доступа к сети Интернет обладают 25 процентов населения планеты Земля. По оценкам в 2011 году Английский язык использовался всего на четверти всех страниц сети Интернет.

Сеть интранет является частной сетью, использующей стек протоколов TCP/IP. Интранет использует те же протоколы, что и Интернет, но доступ к сети имеет лишь персонал одной организации.

Сеть экстранет аналогична сети интеранет за исключением того, что к сети также имеет доступ и персонал некоторых доверенных организаций (партнеры/клиенты/поставщики/...).

21.5. Стек протоколов TCP/IP

21.5.1. История разработки стека протоколов TCP/IP

Разработка стека протоколов TCP/IP была начата в шестидесятых годах прошлого века специалистами из министерства обороны США. В восьмидесятых годах многие коммерческие организации разрабатывали свои стеки протоколов: в компании IBM был создан стек протоколов SNA, в компании Novell разрабатывался стек протоколов IPX/SPX, специалисты компании Microsoft разработали стек протоколов NetBeui, а компания Apple внедрила стек протоколов Appletalk. Все технологии из восьмидесятых годов прекратили свое существование в девяностых годах. К концу девяностых годов практически все компьютеры мира могли работать со стеком протоколов TCP/IP.

По моему мнению, главной причиной успеха стека протоколов TCP/IP, в отличие от всех других протоколов, стала его открытость. Каждый разработчик имеет возможность создать и использовать свою собственную реализацию набора протоколов TCP/IP.

21.5.2. Стандарты RFC (Request For Comment)

Протоколы, которые используются для реализации сети Интернет описаны в стандартах RFC. Стандарт RFC или Request For Comment (Рабочие предложения) описывает низкоуровневые принципы работы всех протоколов, используемых для реализации сети Интернет. Организация IETF (Internet Engineering Task Force — Инженерный совет Интернета) публикует информацию о данных протоколах начиная с 1986 года.

Официальный вебсайт стандартов RFC находится по адресу http://www.rfc-editor.org. Данный вебсайт содержит все стандартны RFC в текстовом формате, к примеру, стандарт RFC 2132 (описывающий принцип работы протоколов dhcp и bootp) доступен по адресу http://

www.rfc-editor.org/rfc/rfc2132.txt.

21.5.3. Многообразие протоколов

Для установки надежных соединений используется протокол TCP, при этом протокол UDP не предусматривает возможности установки соединений, но работает гораздо быстрее. Сообщения об ошибках, доставляемые по протоколу ICMP, используются утилитой ping, а управление группами многоадресной передачи осуществляется с помощью протокола IGMP.

Все эти протоколы можно идентифицировать с помощью значения поля протокола заголовка IP, причем список всех допустимых значений находится в файле /etc/protocols.

```
paul@debian5:~$ grep tcp /etc/protocols
tcp 6 TCP # transmission control protocol
```

21.5.3. Многообразие сетевых служб

Сетевые карты идентифицируются уникальным образом с помощью своих MAC-адресов, узлы — с помощью IP-адресов, а приложения — с помощью номеров портов.

Стандартные протоколы уровня приложений, такие, как smtp, http, ssh, telnet и ftp используют фиксированные номера портов. Список стандартных номеров портов для различных протоколов приведен в файле /etc/services.

```
paul@ubu1010:~$ grep ssh /etc/services
ssh 22/tcp # SSH Remote Login Protocol
ssh 22/udp
```

Глава 22. Настройка сетевых интерфейсов

В данной главе описывается процесс настройки сетевых интерфейсов, соответствующих сетевым картам, для последующего использования стека протоколов TCP/IP.

22.1. Следует ли использовать для настройки инструменты с графическим интерфейсом

В состав современных дистрибутивов Linux часто включаются приложения с графическим интерфейсом, предназначенные для настройки параметров сетевых соединений. Некоторые люди жалуются на то, что данные приложения нарушают настройки сетевых соединений в том случае, если они используются одновременно с инструментами с интерфейсом командной строки. Наиболее известными примерами приложений, которые никоим образом не обрабатывают настройки сетевых соединений, сделанные с помощью инструментов с интерфейсом командной строки, являются демон Network Manager (который нередко заменяется на демон wicd), а также приложение для настройки системы yast.

Так как целью данного курса является изучение приемов администрирования серверов, мы будем считать, что администрирование серверов Linux всегда осуществляется с помощью приложений с интерфейсом командной строки.

В данной главе рассматриваются исключительно вопросы использования инструментов с интерфейсом командной строки для настройки сетевых интерфейсов системы!

К сожалению, не существует единого набора команд и файлов конфигурации в директории /etc, которые могли бы использоваться во всех дистрибутивах Linux. Мы рассмотрим

вопросы настройки сетей в двух (значительных, но отличающихся) семействах дистрибутивов Linux.

Начнем с рассмотрения аспектов настройки сетевых интерфейсов в дистрибутивах Debian/Ubintu, после чего перейдем к рассмотрению аналогичных аспектов в дистрибутивах Fedora/RHEL.

22.2. Настройка сетевых интерфейсов в дистрибутивах Debain/Ubuntu

22.2.1. Файл конфигурации /etc/network/interfaces

Файл /etc/network/interfaces является основным файлом настроек сетевых интерфейсов, соответствующих сетевым картам, в дистрибутивах Debain/Ubuntu.

Клиент DHCP

В примере ниже показано, что наша текущая система Ubuntu 11.04 настроена таким образом, что клиент DHCP используется для сетевого интерфейса eth0 (соответствующего первой сетевой карте).

```
root@ubu1104srv:~# cat /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).
# The loopback network interface
auto lo
iface lo inet loopback
# The primary network interface
auto eth0
iface eth0 inet dhcp
```

Использование клиента DHCP является отличной практикой в случае настройки клиентских машин, но в случае настройки серверов чаще всего требуется использовать фиксированные IP-адреса.

Фиксированный IP-адрес

В примере ниже показано содержимое файла /etc/network/interfaces в случае использования фиксированного IP-адреса.

```
root@ubu1104srv:~# cat /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).
# The loopback network interface
auto lo
iface lo inet loopback
# The primary network interface
auto eth0
iface eth0 inet static
address 192.168.33.100
network 192.168.33.0
netmask 255.255.255.0
gateway 192.168.33.1
```

В примере также показано, что в файле конфигурации помимо единственного IP-адреса могут использоваться дополнительные параметры. Обратитесь к странице руководства interfaces(5) для ознакомления с принципами установки значений таких параметров, как gateway, netmask или каких-либо других.

22.2.2. Утилита /sbin/ifdown

Рекомендуется деактивировать сетевой интерфейс перед изменением его конфигурации (хотя это и не обязательно). Данная операция может быть осуществлена с помощью утилиты ifdown.

Утилита не выведет какого-либо сообщения при деактивации сетевого интерфейса с фиксированным IP-адресом. Однако после деактивации сетевого интерфейса информация о нем больше не будет содержаться в выводе утилиты ifconfig.

```
root@ubu1104srv:~# ifdown eth0
root@ubu1104srv:~# ifconfig
lo Link encap:Local Loopback
inet addr:127.0.0.1 Mask:255.0.0.0
inet6 addr: ::1/128 Scope:Host
UP LOOPBACK RUNNING MTU:16436 Metric:1
RX packets:106 errors:0 dropped:0 overruns:0 frame:0
TX packets:106 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:11162 (11.1 KB) TX bytes:11162 (11.1 KB)
```

Сетевой интерфейс, который был деактивирован, не может использоваться для соединения с сетью без повторной активации.

22.2.3. Утилита /sbin/ifup

Ниже приведен вывод утилиты ifup при активации сетевого интерфейса eth0 с использованием клиента DHCP. (Обратите внимание на то, что данный вывод был получен в дистрибутиве Ubuntu 10.10, при этом в дистрибутиве Ubuntu 11.04 утилита ifup не генерирует вывода по умолчанию).

```
root@ubu1010srv:/etc/network# ifup eth0
Internet Systems Consortium DHCP Client V3.1.3
Copyright 2004-2009 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/
Listening on LPF/eth0/08:00:27:cd:7f:fc
Sending on LPF/eth0/08:00:27:cd:7f:fc
Sending on Socket/fallback
DHCPREQUEST of 192.168.1.34 on eth0 to 255.255.255.255 port 67
DHCPNAK from 192.168.33.100
DHCPDISCOVER on eth0 to 255.255.255.255 port 67 interval 3
```

DHCPOFFER of 192.168.33.77 from 192.168.33.100
DHCPREQUEST of 192.168.33.77 on eth0 to 255.255.255.255 port 67
DHCPACK of 192.168.33.77 from 192.168.33.100
bound to 192.168.33.77 — renewal in 95 seconds.
ssh stop/waiting
ssh start/running, process 1301
root@ubu1010srv:/etc/network#

Подробности использования программных компонентов для работы с протоколом DHCP освещены в отдельной главе курса «Администрирование серверов Linux».

22.3. Настройка сетевых интерфейсов в дистрибутивах Red Hat/Fedora

22.3.1. Файл конфигурации /etc/sysconfig/network

Файл /etc/sysconfig/network является глобальным файлом конфигурации (для всех сетевых карт). Он позволяет установить, хотим ли мы использовать сетевое соединение (NETWORKING=yes|no), желаемое имя узла (HOSTNAME=), а также адрес шлюза (GATEWAY=).

```
[root@rhel6 ~]# cat /etc/sysconfig/network
NETWORKING=yes
HOSTNAME=rhel6
GATEWAY=192.168.1.1
```

Существуют и некоторые другие параметры, значения которых могут быть установлены в рамках данного файла, причем информация об этих параметрах приведена в текстовом файле /usr/share/doc/initscripts-*/sysconfig.txt.

22.3.2. Файлы конфигурации /etc/sysconfig/network-scripts/ifcfg-*

Каждая сетевая карта может быть индивидуально настроена в рамках файла конфигурации /etc/sysconfig/network-scripts/ifcfg-*. Если вы используете всего одну сетевую карту, для ее настройки, скорее всего, будет использоваться файл конфигурации /etc/sysconfig/network-scripts/ifcfg-eth0.

Клиент DHCP

Ниже приведено содержимое файла конфигурации /etc/sysconfig/network-scripts/ifcfg-eth0 в случае использования клиента DHCP (BOOTPROTO=»dhcp»). Также обратите внимание на параметр NM_CONTROLLED, который предназначен для запрета управления этой сетевой картой демоном Network Manager. Данный параметр, в отличие от многих других, не описан (и даже не упомянут) в текстовом файле /usr/share/doc/initscripts-*/sysconfig.txt.

```
[root@rhel6 ~]# cat /etc/sysconfig/network-scripts/ifcfg-eth0
DEVICE="eth0"
HWADDR="08:00:27:DD:0D:5C"
NM_CONTROLLED="no"
BOOTPROTO="dhcp"
ONBOOT="yes"
```

В качестве значения переменной BOOTPROTO может использоваться либо строка dhcp, либо строка bootp, любые другие значения будут интерпретироваться как значение static, которое подразумевает отказ от специальных протоколов для установки параметров сетевого интерфейса при его активации.

Фиксированный IP-адрес

Ниже приведен пример содержимого файла конфигурации /etc/sysconfig/network-scripts/ifcfg-eth0 в случае использования фиксированного IP-адреса.

```
[root@rhel6 ~]# cat /etc/sysconfig/network-scripts/ifcfg-eth0
DEVICE="eth0"
HWADDR="08:00:27:DD:0D:5C"
NM_CONTROLLED="no"
BOOTPROTO="none"
IPADDR="192.168.1.99"
NETMASK="255.255.255.0"
GATEWAY="192.168.1.1"
ONBOOT="yes"
```

Параметр HWADDR предназначен для гарантированного использования заданных значений параметров для определенной сетевой карты при наличии нескольких сетевых карт в системе. Оно не может использоваться для присваивания произвольного MAC-адреса сетевой карте. Для этого вам придется установить значение переменной MACADDR. Не используйте переменные HWADDR и MACADDR в рамках одного файла конфигурации ifcfg-ethx.

Параметры BROADCAST= и NETWORK= из предыдущих версий дистрибутивов RHEL/Fedora являются устаревшими.

22.3.3. Утилиты /sbin/ifup и /sbin/ifdown

Утилиты ifup и ifdown позволяют активировать или деактивировать сетевой интерфейс с использованием описанных выше файлов конфигурации. Данные утилиты ведут себя аналогично соответствующим утилитам из состава дистрибутивов Debian и Ubuntu.

```
[root@rhel6 ~]# ifdown eth0 && ifup eth0
[root@rhel6 ~]# ifconfig eth0
eth0 Link encap:Ethernet HWaddr 08:00:27:DD:0D:5C
inet addr:192.168.1.99 Bcast:192.168.1.255 Mask:255.255.255.0
inet6 addr: fe80::a00:27ff:fedd:d5c/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:2452 errors:0 dropped:0 overruns:0 frame:0
TX packets:1881 errors:0 dropped:0 overruns:0 carrier:0
```

collisions:0 txqueuelen:1000
RX bytes:257036 (251.0 KiB) TX bytes:184767 (180.4 KiB)

22.4. Утилита ifconfig

При использовании утилиты /sbin/ifconfig без аргументов будет выведен список всех активных сетевых интерфейсов, включая интерфейсы адаптеров беспроводных сетей и петлевой интерфейс. В примере ниже IP-адрес сетевого интерфейса eth0 не установлен.

```
root@ubu1010:~# ifconfig
eth0 Link encap:Ethernet HWaddr 00:26:bb:5d:2e:52
UP BROADCAST MULTICAST MTU:1500 Metric:1
RX packets:0 errors:0 dropped:0 overruns:0 frame:0
TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)
Interrupt:43 Base address:0xe000
eth1 Link encap:Ethernet HWaddr 00:26:bb:12:7a:5e
inet addr:192.168.1.30 Bcast:192.168.1.255 Mask:255.255.255.0
inet6 addr: fe80::226:bbff:fe12:7a5e/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:11141791 errors:202 dropped:0 overruns:0 frame:11580126
TX packets:6473056 errors:3860 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:3476531617 (3.4 GB) TX bytes:2114919475 (2.1 GB)
Interrupt:23
lo Link encap:Local Loopback
inet addr:127.0.0.1 Mask:255.0.0.0
inet6 addr: ::1/128 Scope:Host
UP LOOPBACK RUNNING MTU:16436 Metric:1
RX packets:2879 errors:0 dropped:0 overruns:0 frame:0
TX packets:2879 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:486510 (486.5 KB) TX bytes:486510 (486.5 KB)
```

Также вы можете использовать утилиту ifconfig для получения информации об одной сетевой карте.

```
[root@rhel6 ~]# ifconfig eth0
eth0 Link encap:Ethernet HWaddr 08:00:27:DD:0D:5C
inet addr:192.168.1.99 Bcast:192.168.1.255 Mask:255.255.255.0
inet6 addr: fe80::a00:27ff:fedd:d5c/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:2969 errors:0 dropped:0 overruns:0 frame:0
TX packets:1918 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:335942 (328.0 KiB) TX bytes:190157 (185.7 KiB)
```

В том случае, если для учетных записей обычных пользователей директория /sbin не включена в список директорий с исполняемыми файлами, передаваемый с помощью переменной окружения $PATH, вам придется использовать полный путь к утилите, как и в дистрибутиве Debian.

```
paul@debian5:~$ /sbin/ifconfig eth3
eth3 Link encap:Ethernet HWaddr 08:00:27:ab:67:30
inet addr:192.168.1.29 Bcast:192.168.1.255 Mask:255.255.255.0
inet6 addr: fe80::a00:27ff:feab:6730/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:27155 errors:0 dropped:0 overruns:0 frame:0
TX packets:30527 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:13095386 (12.4 MiB) TX bytes:25767221 (24.5 MiB)
```

22.4.1. Активация и деактивация сетевых интерфейсов

Вы можете использовать утилиту ifconfig также для активации и деактивации сетевых интерфейсов. Главное отличие данной утилиты от утилиты ifup состоит в том, что в случае использования команды ifconfig eth0 up сетевой интерфейс будет повторно активирован с сохранением (текущей) конфигурации, в то время, как в случае использования утилиты ifup будет осуществляться чтение соответствующего файла конфигурации, который содержит (возможно обновленные) значения параметров конфигурации, с последующим использованием прочитанных значений параметров конфигурации в процессе активации сетевого интерфейса.

```
[root@rhel6 ~]# ifconfig eth0 down
[root@rhel6 ~]# ifconfig eth0 up
[root@rhel6 ~]# ifconfig eth0
eth0 Link encap:Ethernet HWaddr 08:00:27:DD:0D:5C
inet addr:192.168.1.99 Bcast:192.168.1.255 Mask:255.255.255.0
inet6 addr: fe80::a00:27ff:fedd:d5c/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:2995 errors:0 dropped:0 overruns:0 frame:0
TX packets:1927 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:339030 (331.0 KiB) TX bytes:191583 (187.0 KiB)
```

22.4.2. Установка IP-адреса

С помощью утилиты ifconfig вы также можете временно установить IP-адрес сетевого интерфейса. Этот IP-адрес будет использоваться до следующего цикла активации/деактивации сетевого интерфейса с помощью утилит ifup/ifdown или до следующей перезагрузки системы.

```
[root@rhel6 ~]# ifconfig eth0 | grep 192
```

inet addr:192.168.1.99 Bcast:192.168.1.255 Mask:255.255.255.0
[root@rhel6 ~]# ifconfig eth0 192.168.33.42 netmask 255.255.0.0
[root@rhel6 ~]# ifconfig eth0 | grep 192
inet addr:192.168.33.42 Bcast:192.168.255.255 Mask:255.255.0.0
[root@rhel6 ~]# ifdown eth0 && ifup eth0
[root@rhel6 ~]# ifconfig eth0 | grep 192
inet addr:192.168.1.99 Bcast:192.168.1.255 Mask:255.255.255.0

22.4.3. Установка MAC-адреса

Кроме того, вы также можете использовать утилиту ifconfig для установки MAC-адреса сетевого интерфейса, отличного от того, который был установлен при изготовлении вашей сетевой карты. В примере ниже показан процесс его установки.

[root@rhel6 ~]# ifconfig eth0 | grep HWaddr
eth0 Link encap:Ethernet HWaddr 08:00:27:DD:0D:5C
[root@rhel6 ~]# ifconfig eth0 hw ether 00:42:42:42:42:42
[root@rhel6 ~]# ifconfig eth0 | grep HWaddr
eth0 Link encap:Ethernet HWaddr 00:42:42:42:42:42

22.4.4. Демон dhclient

Демон /sbin/dhclient обычно исполняется в домашних или клиентских системах, работающих под управлением Linux. Данный демон позволяет получать параметры подключения сетевого интерфейса от DHCP-сервера. В том случае, если в настройках вашего сетевого адаптера есть указание на использование протоколов DHCP или BOOTP, утилита /sbin/ifup будет автоматически осуществлять запуск демона dhclient.

При обновлении данных подключения демон dhclient будет заменять IP-адрес, установленный с помощью утилиты ifconfig!

22.5. Имя узла

Каждый узел получает имя, называемое именем узла (hostname) и обычно находящееся в пространстве имен DNS и формирующее полностью определенное имя домена (Fully Qualified Domain Name — FQDN).

В данном примере показана методика установки имени узла и его проверки с помощью утилиты hostname в случае работы с дистрибутивами Red Hat/Fedora.

[root@rhel6 ~]# grep rhel /etc/sysconfig/network
HOSTNAME=rhel6
[root@rhel6 ~]# hostname
rhel6

Для установки имени узла в дистрибутивах Ubuntu/Debian используется файл /etc/hostname.

paul@ubu1010:~$ cat /etc/hostname
ubu1010
paul@ubu1010:~$ hostname
ubu1010

Во всех дистрибутивах Linux вы можете изменить имя узла с помощью команды hostname $новое_имя_узла. В этом случае имя узла будет изменяться не на постоянной основе.

[root@rhel6 ~]# hostname server42
[root@rhel6 ~]# hostname
server42

В любой системе Linux вы также можете использовать утилиту sysctl для вывода и изменения имени узла.

[root@rhel6 ~]# sysctl kernel.hostname
kernel.hostname = server42
[root@rhel6 ~]# sysctl kernel.hostname=rhel6
kernel.hostname = rhel6
[root@rhel6 ~]# sysctl kernel.hostname
kernel.hostname = rhel6
[root@rhel6 ~]# hostname
rhel6

22.6. Утилита arp

Поиск соответствий между IP- и MAC-адресами осуществляется с помощью протокола ARP второго уровня сетевой модели OSI. Содержимое таблиц ARP может выводиться с помощью одноименной утилиты arp. В примере ниже показан список имен и адресов компьютеров, с которыми не так давно обменивался данными рассматриваемый компьютер.

root@barry:~# arp -a
? (192.168.1.191) at 00:0C:29:3B:15:80 [ether] on eth1
agapi (192.168.1.73) at 00:03:BA:09:7F:D2 [ether] on eth1
anya (192.168.1.1) at 00:12:01:E2:87:FB [ether] on eth1
faith (192.168.1.41) at 00:0E:7F:41:0D:EB [ether] on eth1
kiss (192.168.1.49) at 00:D0:E0:91:79:95 [ether] on eth1
laika (192.168.1.40) at 00:90:F5:4E:AE:17 [ether] on eth1
pasha (192.168.1.71) at 00:03:BA:02:C3:82 [ether] on eth1
shaka (192.168.1.72) at 00:03:BA:09:7C:F9 [ether] on eth1
root@barry:~#

Узел с именем «anya» является межсетевым экраном Cisco, с именем «faith» —

лазерным принтером, с именем «kiss» — медиа-плеером Kiss DP600, с именем «laika» - ноутбуком, а с именами «Agapi», «Shaka» и «Pasha» - серверами SPARC. С помощью символа знака вопроса обозначен сервер под управлением дистрибутива Red Hat Enterprise Linux, работающий в виртуальной машине.

Вы можете использовать команду arp -d для удаления элемента из таблицы ARP.

[root@rhel6 ~]# arp

Address HWtype HWaddress Flags Mask Iface
ubu1010 ether 00:26:bb:12:7a:5e C eth0
anya ether 00:02:cf:aa:68:f0 C eth0
[root@rhel6 ~]# arp -d anya
[root@rhel6 ~]# arp
Address HWtype HWaddress Flags Mask Iface
ubu1010 ether 00:26:bb:12:7a:5e C eth0
anya (incomplete) eth0
[root@rhel6 ~]# ping anya
PING anya (192.168.1.1) 56(84) bytes of data.
64 bytes from anya (192.168.1.1): icmp_seq=1 ttl=254 time=10.2 ms
...
[root@rhel6 ~]# arp
Address HWtype HWaddress Flags Mask Iface
ubu1010 ether 00:26:bb:12:7a:5e C eth0
anya ether 00:02:cf:aa:68:f0 C eth0

22.7. Утилита route

Вы можете ознакомиться с локальной таблицей маршрутизации компьютера, воспользовавшись утилитой /sbin/route (а также с помощью команды netstat -r).

root@RHEL4b ~]# netstat -r
Kernel IP routing table
Destination Gateway Genmask Flags MSS Window irtt Iface
192.168.1.0 * 255.255.255.0 U 0 0 0 eth0
[root@RHEL4b ~]# route
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
192.168.1.0 * 255.255.255.0 U 0 0 0 eth0
[root@RHEL4b ~]#

Оказывается, на данном компьютере не установлен адрес шлюза, поэтому мы можем воспользоваться командой route add default gw для непосредственной установки адреса шлюза, используемого по умолчанию.

[root@RHEL4b ~]# route add default gw 192.168.1.1
[root@RHEL4b ~]# route
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
192.168.1.0 * 255.255.255.0 U 0 0 0 eth0
default 192.168.1.1 0.0.0.0 UG 0 0 0 eth0
[root@RHEL4b ~]#

Если вы не добавите адрес шлюза в один из описанных в начале данной главы файлов конфигурации сетевых интерфейсов из директории /etc/, ваш компьютер будет забывать этот адрес после каждой перезагрузки.

22.8. Утилита ping

Если удаленный узел обнаруживается в сети с помощью утилиты ping, все необходимые для функционирования стека протоколов TCP/IP настройки были выполнены корректно.

[root@RHEL4b ~]# ping 192.168.1.5
PING 192.168.1.5 (192.168.1.5) 56(84) bytes of data.
64 bytes from 192.168.1.5: icmp_seq=0 ttl=64 time=1004 ms
64 bytes from 192.168.1.5: icmp_seq=1 ttl=64 time=1.19 ms
64 bytes from 192.168.1.5: icmp_seq=2 ttl=64 time=0.494 ms
64 bytes from 192.168.1.5: icmp_seq=3 ttl=64 time=0.419 ms
--- 192.168.1.5 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3009ms
rtt min/avg/max/mdev = 0.419/251.574/1004.186/434.520 ms, pipe 2
[root@RHEL4b ~]#

22.9. Дополнительная информация: утилита ethtool

Для вывода и изменения значений рабочих параметров сетевой карты следует использовать утилиту ethtool. Результаты работы утилиты зависят от возможностей вашей сетевой карты. При рассмотрении представленного в примере вывода можно сделать вывод, что сетевая карта осуществляет автоматическое согласование пропускной способности.

root@laika:~# ethtool eth0
Settings for eth0:
Supported ports: [TP]
Supported link modes: 10baseT/Half 10baseT/Full
100baseT/Half 100baseT/Full
1000baseT/Full
Supports auto-negotiation: Yes
Advertised link modes: 10baseT/Half 10baseT/Full
100baseT/Half 100baseT/Full
1000baseT/Full
Advertised auto-negotiation: Yes
Speed: 1000Mb/s
Duplex: Full
Port: Twisted Pair
PHYAD: 0
Transceiver: internal
Auto-negotiation: on
Supports Wake-on: pumbg
Wake-on: g
Current message level: 0x00000033 (51)
Link detected: yes

А в данном примере показана методика использования утилиты ethtool с целью перевода сетевой карты с режима работы с

пропускной способностью в 1000МБит к режиму работы с пропускной способностью в 100МБит и назад. Учтите, что для возврата сетевой карты в режим работы с пропускной способностью в 1000Мбит потребуется некоторое время.

root@laika:~# ethtool eth0 | grep Speed
Speed: 1000Mb/s
root@laika:~# ethtool -s eth0 speed 100
root@laika:~# ethtool eth0 | grep Speed
Speed: 100Mb/s
root@laika:~# ethtool -s eth0 speed 1000
root@laika:~# ethtool eth0 | grep Speed
Speed: 1000Mb/s

22.10. Практическое задание: настройка сетевых интерфейсов

1. Проверьте, исполняется ли на вашем компьютере демон dhclient.
2. Выведите информацию о вашем текущем IP-адресе (или адресах).
3. Выведите содержимое файла конфигурации, в котором записан используемый IP-адрес.
4. Выполните действия, перечисленные в одном из разделов с описанием процесса настройки сетевых интерфейсов, для перехода от использования клиента DHCP к использованию фиксированного IP-адреса. Используйте полученный ранее IP-адрес для того, чтобы избежать конфликтов!
5. Установили ли вы корректный адрес шлюза при выполнении предыдущего задания? Если нет, сделайте это сейчас.
6. Проверьте корректность установки адреса шлюза.
7. Проверьте возможность соединения с адресом шлюза, а также корректность работы шлюза.
8. Измените два последних символа в MAC-адресе вашей сетевой карты.
9. Какие порты используются приложениями, работающими по протоколам http, pop3, ssh, telnet, nntp и ftp?
10. Поясните, почему для передачи сообщений электронной почты и доступа к вебсайтам используется протокол TCP, а не UDP.
11. Выведите имя узла вашего компьютера.
12. Выведите список IP-адресов узлов, с которыми ваш компьютер недавно обменивался данными.

22.11. Корректная процедура выполнения практического задания: настройка сетевых интерфейсов

1. Проверьте, исполняется ли на вашем компьютере демон dhclient.

paul@debian5:~$ ps fax | grep dhclient

1. Выведите информацию о вашем текущем IP-адресе (или адресах).

paul@debian5:~$ /sbin/ifconfig | grep 'inet '
inet addr:192.168.1.31 Bcast:192.168.1.255 Mask:255.255.255.0
inet addr:127.0.0.1 Mask:255.0.0.0

1. Выведите содержимое файла конфигурации, в котором записан используемый IP-адрес.

В дистрибутивах Ubuntu/Debian: cat /etc/network/interfaces
В дистрибутивах RedHat/Fedora: cat /etc/sysconfig/network-scripts/ifcfg-eth*

1. Выполните действия, перечисленные в одном из разделов с описанием процесса настройки сетевых интерфейсов, для перехода от использования клиента DHCP к использованию фиксированного IP-адреса. Используйте полученный ранее IP-адрес для того, чтобы избежать конфликтов!

В дистрибутивах Ubuntu/Debian:
ifdown eth0
vi /etc/network/interfaces
ifup eth0
В дистрибутивах RedHat/Fedora:
ifdown eth0
vi /etc/sysconfig/network-scripts/ifcfg-eth0
ifup eth0

1. Установили ли вы корректный адрес шлюза при выполнении предыдущего задания? Если нет, сделайте это сейчас.
2. Проверьте корректность установки адреса шлюза.

paul@debian5:~$ /sbin/route
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
192.168.1.0 * 255.255.255.0 U 0 0 0 eth0
default 192.168.1.1 0.0.0.0 UG 0 0 0 eth0

1. Проверьте возможность соединения с адресом шлюза, а также корректность работы шлюза.
   paul@debian5:~$ ping -c3 192.168.1.1

PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
64 bytes from 192.168.1.1: icmp_seq=1 ttl=254 time=2.28 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=254 time=2.94 ms
64 bytes from 192.168.1.1: icmp_seq=3 ttl=254 time=2.34 ms
--- 192.168.1.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2008ms
rtt min/avg/max/mdev = 2.283/2.524/2.941/0.296 ms

1. Измените два последних символа в MAC-адресе вашей

сетевой карты.

[root@rhel6 ~]# ifconfig eth0 hw ether 08:00:27:ab:67:XX

1. Какие порты используются приложениями, работающими по протоколам http, pop3, ssh, telnet, nntp и ftp?

root@rhel6 ~# grep ^'http ' /etc/services
http 80/tcp www www-http # WorldWideWeb HTTP
http 80/udp www www-http # HyperText Transfer Protocol
root@rhel6 ~# grep ^'smtp ' /etc/services
smtp 25/tcp mail
smtp 25/udp mail
root@rhel6 ~# grep ^'ssh ' /etc/services
ssh 22/tcp # The Secure Shell (SSH) Protocol
ssh 22/udp # The Secure Shell (SSH) Protocol
root@rhel6 ~# grep ^'telnet ' /etc/services
telnet 23/tcp
telnet 23/udp
root@rhel6 ~# grep ^'nntp ' /etc/services
nntp 119/tcp readnews untp # USENET News Transfer Protocol
nntp 119/udp readnews untp # USENET News Transfer Protocol
root@rhel6 ~# grep ^'ftp ' /etc/services
ftp 21/tcp
ftp 21/udp fsp fspd

1. Поясните, почему для передачи сообщений электронной почты и доступа к вебсайтам используется протокол TCP, а не UDP.

Так как протокол TCP является надежным, а протокол UDP не является таковым.

1. Выведите имя узла вашего компьютера.

paul@debian5:~$ hostname
debian5

1. Выведите список IP-адресов узлов, с которыми ваш компьютер недавно обменивался данными.

root@rhel6 ~# arp -a
? (192.168.1.1) at 00:02:cf:aa:68:f0 [ether] on eth2
? (192.168.1.30) at 00:26:bb:12:7a:5e [ether] on eth2
? (192.168.1.31) at 08:00:27:8e:8a:a8 [ether] on eth2

Глава 23. Перехват сетевого трафика

Администратор сети должен уметь работать со сниффером, таким, как wireshark или tcpdump, для диагностирования проблем сети.

Студенту также придется нередко прибегать к использованию сниффера для того, чтобы разобраться в принципах функционирования сетей. В данной главе описываются соответствующие методики перехвата сетевого трафика.

23.1. Приложение wireshark

23.1.1. Установка wireshark

В данном примере приведена команда для установки приложения wireshark в дистрибутивах, использующих пакеты программного обеспечения с расширением .deb (включая Debian, Mint, Xubuntu и другие дистрибутивы).

root@debian8:~#
Чтение списков пакетов... Готово
Построение дерева зависимостей
Чтение информации о состоянии... Готово

1. (вывод сокращен)

В дистрибутивах, использующих пакеты программного обеспечения с расширением .rpm, таких, как CentOS, RHEL и Fedora, для установки приложения wireshark может использоваться утилита yum.

[root@centos7 ~]#
yum install wireshark
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile

1. (вывод сокращен)

23.1.2. Выбор сетевого интерфейса

При запуске приложения wireshark в первый раз вам придется выбрать сетевой интерфейс. Вы увидите диалог, который выглядит аналогично приведенному на иллюстрации ниже.

Вероятна ситуация, при которой доступных сетевых интерфейсов попросту не окажется, ведь в некоторых дистрибутивах перехват сетевого трафика может осуществляться исключительно пользователем root. В этом случае вам придется запустить приложение wireshark от лица пользователя root с помощью команды sudo wireshark.

Или же вы можете последовать общим рекомендациям и использовать утилиту tcpdump или какой-либо другой инструмент для перехвата трафика и записи данных в файл. Любые перехваченные данные могут быть проанализированы позднее с помощью приложения wireshark.

23.1.3. Минимизация трафика

В процессе перехвата сетевого трафика в течение очень коротких промежутков времени могут генерироваться тысячи пакетов. Очевидно, что такой объем данных затруднит анализ трафика. Попытайтесь выйти из этого положения, изолировав ваш сниффер в рамках сети. Предпочтительным вариантом является перехват трафика, проходящего через сетевой интерфейс изолированной виртуальной сети, находящейся под вашим полным контролем.

Если вы изучаете инструменты для перехвата сетевого трафика в домашних условиях, для минимизации трафика будет полезно завершить работу всех приложений для работы с сетью, запущенных на вашем компьютере, а также отсоединить компьютер от других компьютеров и от таких устройств, как смартфоны и планшеты.

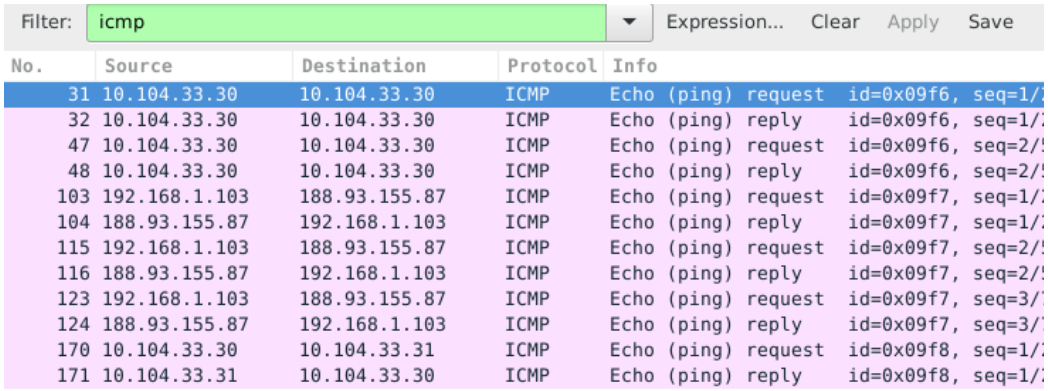
И все же более важным инструментом для минимизации трафика являются фильтры, которые будут обсуждаться в следующем разделе.

23.1.4. Перехват трафика, генерируемого утилитой ping

Я запустил сниффер и захватил все пакеты, переданные по сети в результате исполнения трех команд ping (не имеет смысла выполнять эти команды от лица пользователя root):

```
root@debian7:~# ping -c2 ns1.paul.local
PING ns1.paul.local (10.104.33.30) 56(84) bytes of data.
64 bytes from 10.104.33.30: icmp_req=1 ttl=64 time=0.010 ms
64 bytes from 10.104.33.30: icmp_req=2 ttl=64 time=0.023 ms
--- ns1.paul.local ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 0.010/0.016/0.023/0.007 ms
root@debian7:~# ping -c3 linux-training.be
PING linux-training.be (188.93.155.87) 56(84) bytes of data.
64 bytes from antares.ginsys.net (188.93.155.87): icmp_req=1 ttl=56 time=15.6 ms
64 bytes from antares.ginsys.net (188.93.155.87): icmp_req=2 ttl=56 time=17.8 ms
64 bytes from antares.ginsys.net (188.93.155.87): icmp_req=3 ttl=56 time=14.7 ms
--- linux-training.be ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 14.756/16.110/17.881/1.309 ms
root@debian7:~# ping -c1 centos7.paul.local
PING centos7.paul.local (10.104.33.31) 56(84) bytes of data.
64 bytes from 10.104.33.31: icmp_req=1 ttl=64 time=0.590 ms
--- centos7.paul.local ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.590/0.590/0.590/0.000 ms
```

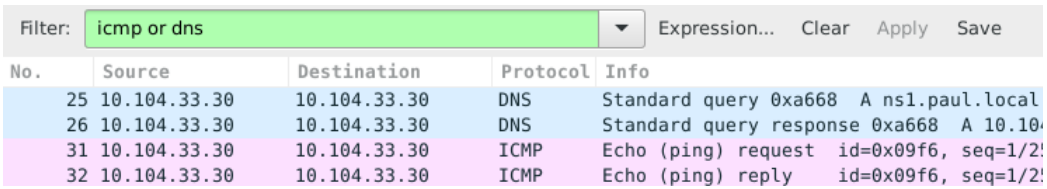
В общей сложности из сети было захвачено более чем 200 пакетов. Все станет гораздо очевиднее в том случае, если введете строку icmp в поле фильтра и нажмете кнопку «Применить» («Apply»).



23.1.5. Перехват трафика, генерируемого утилитой ping и клиентом dns

Работая с той же сессией захвата данных, применим отличный фильтр. Мы хотим отслеживать трафик, относящийся как к протоколу dns, так и к протоколу icmp, поэтому нам придется ввести названия двух упомянутых протоколов в поле фильтра.

Для захвата данных, относящихся к двум рассматриваемым протоколам, в поле фильтра должна быть введена строка «dns or icmp». В случае ввода строки «dns and icmp» не будет выведено информации о каких-либо пакетах, так как не существует пакетов, относящихся к обоим упомянутыми протоколам.

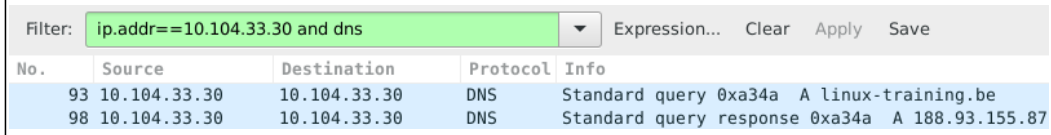


При рассмотрении приведенной выше иллюстрации можно заметить, что пакеты 25 и 26 имеют исходные и целевые IP-адреса 10.104.33.30. Это объясняется тем, что клиент DNS работает на том же компьютере, что и сервер DNS.

Аналогичная ситуация наблюдается и в случае пакетов 31 и 32, ведь с помощью утилиты ping осуществляется отправка пакетов рабочей системе, на которой запущена данная утилита.

23.1.6. Определенный IP-адрес

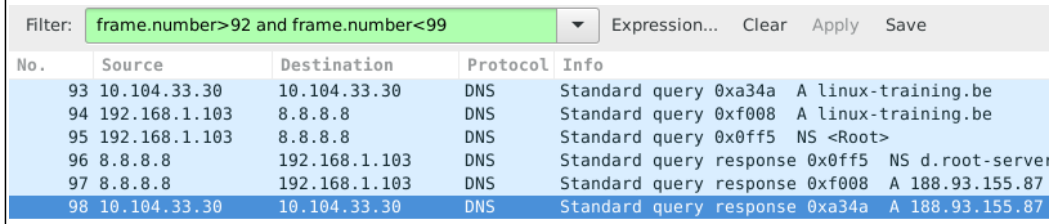
В данном случае осуществляется фильтрация пакетов, относящихся к протоколу DNS и содержащих определенный IP-адрес. В качестве фильтра используется строка «ip.addr==10.104.33.30 and dns». Директива and сообщает приложению о том, что следует выводить информацию о каждом пакете, соответствующем двум условиям.



Пакет 93 содержит запрос DNS, направленный на получение записи типа A домена linux-training.be. Пакет 98 содержит ответ от сервера DNS. Как вы думаете, что происходило после отправки пакета 93 и до приема пакета 98? Попытайтесь ответить на этот вопрос перед чтением следующего раздела (при работе с различными системами всегда полезно пытаться предсказывать наступающие события и проверять корректность своих предсказаний).

23.1.7. Фильтрация на основе фреймов

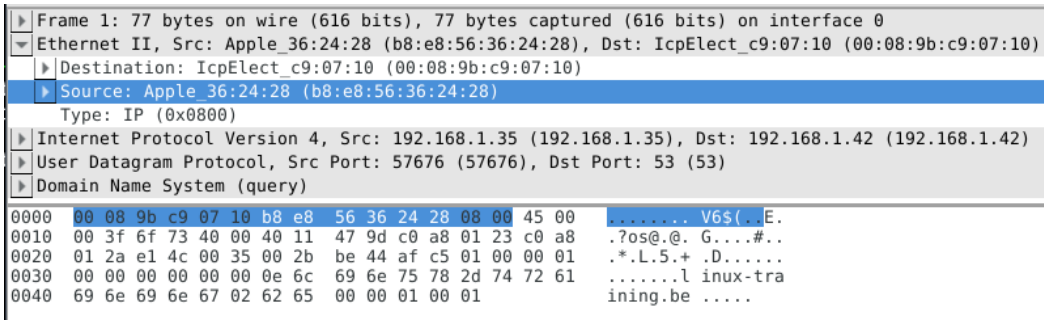
Корректным термином, используемым для обозначения перехваченного пакета, является термин фрейм (из-за того, что мы осуществляем перехват пакетов на уровне 2 сетевой модели OSI). Таким образом, для вывода информации о пакетах с определенными номерами, следует использовать директиву frame.number в поле фильтра.



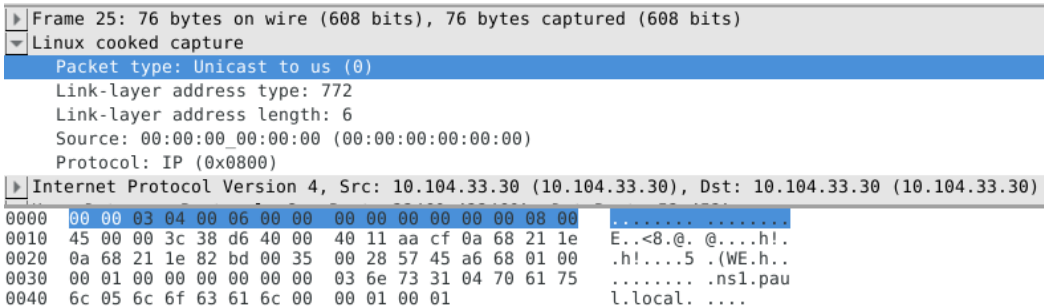
23.1.8. Исследование содержимого пакетов

Средняя панель окна сниффера может быть раскрыта. При выборе строки в рамках данной панели вы можете увидеть соответствующие значения байт в поле нижней панели.

На иллюстрации ниже показана средняя панель окна сниффера с выбранным адресом моего ноутбука.

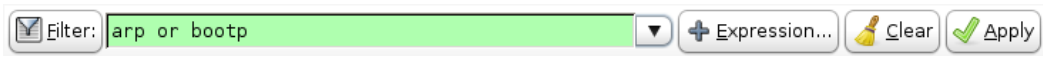


Учтите, что описанная выше техника отлично работает при перехвате трафика, передаваемого через один сетевой интерфейс. Если же вы перехватываете трафик, к примеру, с помощью команды tcpdump -i any, вы столкнетесь с методом перехвата пакетов «Linux cooked capture».

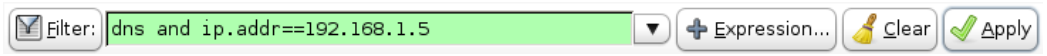


23.1.9. Другие примеры фильтров

Вы можете комбинировать два описания протоколов с помощью директивы логической операции or («ИЛИ»), расположенной между ними. На иллюстрации ниже показан способ захвата исключительно пакетов, относящихся к протоколам ARP и BOOTP (или DHCP).



А на следующей иллюстрации показан способ перехвата пакетов, относящихся к протоколу DNS и содержащих определенный IP-адрес.



23.2. Утилита tcpdump

В случае работы с интерфейсом командной строки системы перехват пакетов может осуществляться с помощью утилиты tcpdump. Ниже приведены некоторые примеры ее использования.

При использовании команды tcpdump host $ip будет выводиться информация обо всем трафике, относящимся к определенному узлу (в данном случае с IP-адресом 192.168.1.38).

root@ubuntu910:~# tcpdump host 192.168.1.38
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes

Перехват трафика, относящегося исключительно к протоколу ssh (протокол TCP, порт 22), может осуществляться с помощью команды tcpdump tcp port $порт. Длина строк вывода урезана до 76 символов для более удобного чтения.

root@deb503:~# tcpdump tcp port 22
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth1, link-type EN10MB (Ethernet), capture size 96 bytes
14:22:20.716313 IP deb503.local.37973 > rhel53.local.ssh: P 666050963:66605
14:22:20.719936 IP rhel53.local.ssh > deb503.local.37973: P 1:49(48) ack 48
14:22:20.720922 IP rhel53.local.ssh > deb503.local.37973: P 49:113(64) ack
14:22:20.721321 IP rhel53.local.ssh > deb503.local.37973: P 113:161(48) ack
14:22:20.721820 IP deb503.local.37973 > rhel53.local.ssh: . ack 161 win 200
14:22:20.722492 IP rhel53.local.ssh > deb503.local.37973: P 161:225(64) ack
14:22:20.760602 IP deb503.local.37973 > rhel53.local.ssh: . ack 225 win 200
14:22:23.108106 IP deb503.local.54424 > ubuntu910.local.ssh: P 467252637:46
14:22:23.116804 IP ubuntu910.local.ssh > deb503.local.54424: P 1:81(80) ack
14:22:23.116844 IP deb503.local.54424 > ubuntu910.local.ssh: . ack 81 win 2
^C
10 packets captured
10 packets received by filter
0 packets dropped by kernel

Та же операция, но с записью захваченных данных в файл, может осуществляться с помощью команды tcpdump -w

$имя_файла.

root@ubuntu910:~# tcpdump -w sshdump.tcpdump tcp port 22

tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes

^C

17 packets captured

17 packets received by filter

0 packets dropped by kernel

С помощью команды tcpdump -r $имя_файла может быть выведено содержимое файла, созданного в предыдущем примере.

root@ubuntu910:~# tcpdump -r sshdump.tcpdump

Множество других примеров использования рассматриваемой утилиты может быть найдено на странице руководства man tcpdump.

23.3. Практическое задание: перехват сетевого трафика

1. Установите приложение wireshark в вашу систему (работающую не в виртуальной машине).
2. Используйте утилиту ping генерации трафика между вашим и каким-либо другим компьютером.
3. Начните перехват сетевого трафика.
4. С помощью фильтра осуществите вывод информации исключительно о тех пакетах, которые содержат ответы на запросы от утилиты ping.
5. Теперь передайте утилите ping имя домена (такое, как www.linux-training.be) и попытайтесь перехватить пакеты с запросом и ответом DNS. Какой DNS-сервер был использован? Был ли использован протокол TCP или UDP для передачи запроса и ответа?
6. Найдите закрытый вебсайт, имеющий форму запроса пароля. Попытайтесь войти на него, использовав имя пользователя paul и пароль hunter2 в процессе работы сниффера. Теперь попытайтесь найти введенные имя пользователя и пароль в захваченных с помощью сниффера данных.

23.4. Корректная процедура выполнения практического задания: перехват сетевого трафика

1. Установите приложение wireshark в вашу систему (работающую не в виртуальной машине).

В дистрибутивах Debain/Ubuntu: aptitude install wireshark

В дистрибутивах Red Hat/Mandriva/Fedora: yum install wireshark

1. Используйте утилиту ping генерации трафика между вашим и каким-либо другим компьютером.

ping $ip_адрес

1. Начните перехват сетевого трафика.

(sudo) wireshark

Выберите сетевой интерфейс (вероятно, eth0)

1. С помощью фильтра осуществите вывод информации исключительно о тех пакетах, которые содержат ответы на запросы от утилиты ping.

Введите icmp (без кавычек) в поле фильтра и нажмите кнопку «Применить» («Apply»).

1. Теперь передайте утилите ping имя домена (такое, как www.linux-training.be) и попытайтесь перехватить пакеты с запросом и ответом DNS. Какой DNS-сервер был использован? Был ли использован протокол TCP или UDP для передачи запроса и ответа?

В первую очередь запустите сниффер.

Введите dns в поле фильтра и нажмите кнопку «Применить» («Apply»).

root@ubuntu910:~# ping www.linux-training.be

PING www.linux-training.be (88.151.243.8) 56(84) bytes of data.

64 bytes from fosfor.openminds.be (88.151.243.8): icmp_seq=1 ttl=58 time=14.9 ms

64 bytes from fosfor.openminds.be (88.151.243.8): icmp_seq=2 ttl=58 time=16.0 ms

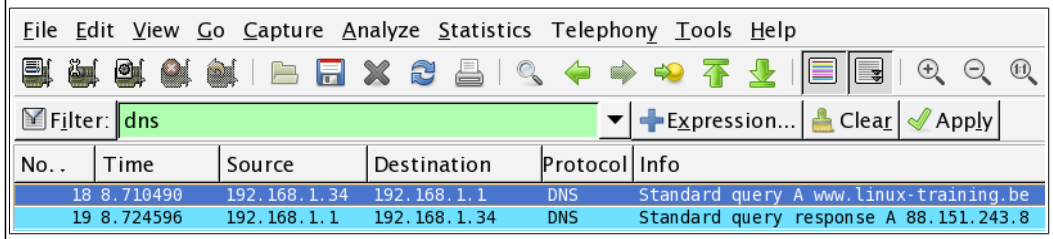
^C

--- www.linux-training.be ping statistics ---

2 packets transmitted, 2 received, 0% packet loss, time 1002ms

rtt min/avg/max/mdev = 14.984/15.539/16.095/0.569 ms

Окно приложения wireshark должно выглядеть аналогичным образом.



На основе информация из окна приложения wireshark можно сделать вывод о том, что запрос DNS передавался с помощью пакета UDP, после чего несложно дать ответы на поставленные вопросы.

1. Найдите закрытый вебсайт, имеющий форму запроса пароля. Попытайтесь войти на него, использовав имя пользователя paul и пароль hunter2 в процессе работы сниффера. Теперь попытайтесь найти введенные имя

пользователя и пароль в захваченных с помощью сниффера данных.

Глава 24. Добавление IP-адресов и связывание сетевых интерфейсов

Иногда приходится использовать более одного IP-адреса при работе с одной сетевой картой, причем в этом случае говорят о добавлении IP-адресов сетевого интерфейса (binding IP addresses).

Ядро Linux также может активировать множество сетевых карт, использующих один и тот же IP-адрес, причем данная технология называется связыванием сетевых интерфейсов (bonding).

В данной главе описываются процессы добавления IP-адресов и связывания сетевых интерфейсов в наиболее популярных дистрибутивах Linux.

24.1. Добавление IP-адресов сетевых интерфейсов в дистрибутивах Red Hat/Fedora

24.1.1. Добавление дополнительных IP-адресов

Для связывания более чем одного IP-адреса с одним и тем же сетевым интерфейсом следует использовать файл конфигурации сетевого интерфейса с именем ifcfg-eth0:0, где вместо второго значения 0 может использоваться любое число. В подобных файлах конфигурации обязательны к использованию всего две директивы.

```
[root@rhel6 ~]# cat /etc/sysconfig/network-scripts/ifcfg-eth0:0
DEVICE="eth0:0"
IPADDR="192.168.1.133"
[root@rhel6 ~]# cat /etc/sysconfig/network-scripts/ifcfg-eth0:1
DEVICE="eth0:0"
IPADDR="192.168.1.142"
```

24.1.2. Активация дополнительных IP-адресов

Для активации виртуального сетевого интерфейса следует использовать утилиту ifup, для его деактивации — утилиту ifdown.

```
[root@rhel6 ~]# ifup eth0:0
[root@rhel6 ~]# ifconfig | grep 'inet '
inet addr:192.168.1.99 Bcast:192.168.1.255 Mask:255.255.255.0
inet addr:192.168.1.133 Bcast:192.168.1.255 Mask:255.255.255.0
inet addr:127.0.0.1 Mask:255.0.0.0
[root@rhel6 ~]# ifup eth0:1
[root@rhel6 ~]# ifconfig | grep 'inet '
inet addr:192.168.1.99 Bcast:192.168.1.255 Mask:255.255.255.0
inet addr:192.168.1.133 Bcast:192.168.1.255 Mask:255.255.255.0
inet addr:192.168.1.142 Bcast:192.168.1.255 Mask:255.255.255.0
inet addr:127.0.0.1 Mask:255.0.0.0
```

24.1.3. Проверка корректности добавления дополнительных IP-адресов

Для проверки корректности добавления и активации дополнительных IP-адресов сетевого интерфейса следует либо использовать утилиту ping на другом компьютере, либо использовать утилиту ifconfig таким образом, как показано в примере ниже.

```
[root@rhel6 ~]# ifconfig
eth0 Link encap:Ethernet HWaddr 08:00:27:DD:0D:5C
inet addr:192.168.1.99 Bcast:192.168.1.255 Mask:255.255.255.0
inet6 addr: fe80::a00:27ff:fedd:d5c/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:1259 errors:0 dropped:0 overruns:0 frame:0
TX packets:545 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:115260 (112.5 KiB) TX bytes:84293 (82.3 KiB)
eth0:0 Link encap:Ethernet HWaddr 08:00:27:DD:0D:5C
inet addr:192.168.1.133 Bcast:192.168.1.255 Mask:255.255.255.0
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
eth0:1 Link encap:Ethernet HWaddr 08:00:27:DD:0D:5C
inet addr:192.168.1.142 Bcast:192.168.1.255 Mask:255.255.255.0
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
```

24.2. Добавление IP-адресов сетевых интерфейсов в дистрибутивах Debian/Ubuntu

24.2.1. Добавление дополнительных IP-адресов

Указание дополнительных IP-адресов для одной и той же сетевой карты осуществляется в рамках файла конфигурации /etc/network/interfaces путем добавления устройств с именами eth0:x. Кроме того, при редактировании данного файла конфигурации обязательно указание масок сетей сетевых интерфейсов.

```
debian5:~# cat /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).
# The loopback network interface
auto lo
iface lo inet loopback
# The primary network interface
iface eth0 inet static
address 192.168.1.34
network 192.168.1.0
netmask 255.255.255.0
gateway 192.168.1.1
```

```
auto eth0
auto eth0:0
iface eth0:0 inet static
address 192.168.1.233
netmask 255.255.255.0
auto eth0:1
iface eth0:1 inet static
address 192.168.1.242
netmask 255.255.255.0
```

24.2.2. Активация дополнительных IP-адресов

Используйте утилиту ifup для активации дополнительных IP-адресов.

```
debian5:~# ifup eth0:0
debian5:~# ifup eth0:1
```

24.2.3. Проверка корректности добавления дополнительных IP-адресов

Для проверки корректности добавления и активации дополнительных IP-адресов следует либо использовать утилиту ping с другого компьютера, либо использовать утилиту ifconfig таким образом, как показано в примере ниже.

```
debian5:~# ifconfig | grep 'inet '
inet addr:192.168.1.34 Bcast:192.168.1.255 Mask:255.255.255.0
inet addr:192.168.1.233 Bcast:192.168.1.255 Mask:255.255.255.0
inet addr:192.168.1.242 Bcast:192.168.1.255 Mask:255.255.255.0
inet addr:127.0.0.1 Mask:255.0.0.0
```

24.3. Связывание сетевых интерфейсов в дистрибутивах RedHat/Fedora

Начнем с выполнения команды ifconfig -a для получения списка всех сетевых карт, установленных в компьютере.

```
[root@rhel6 network-scripts]# ifconfig -a | grep Ethernet
eth0 Link encap:Ethernet HWaddr 08:00:27:DD:0D:5C
eth1 Link encap:Ethernet HWaddr 08:00:27:DA:C1:49
eth2 Link encap:Ethernet HWaddr 08:00:27:40:03:3B
```

В данном примере мы будем связывать сетевые интерфейсы eth1 и eth2.

Назовем нашу связку сетевых интерфейсов bond0 и добавим это название в файл конфигурации утилиты modprobe для того, чтобы ядро ОС автоматически загружало модуль bonding в момент, когда мы будем активировать соответствующий сетевой интерфейс.

```
[root@rhel6 network-scripts]# cat /etc/modprobe.d/bonding.conf
alias bond0 bonding
```

После этого мы должны будем создать файл конфигурации /etc/sysconfig/network-scripts/ifcfg-bond0 для сохранения параметров конфигурации нашего сетевого интерфейса bond0.

```
[root@rhel6 network-scripts]# pwd
/etc/sysconfig/network-scripts
[root@rhel6 network-scripts]# cat ifcfg-bond0
DEVICE=bond0
IPADDR=192.168.1.199
NETMASK=255.255.255.0
ONBOOT=yes
BOOTPROTO=none
USERCTL=no
```

На следующем этапе мы должны будем создать два файла конфигурации, по одному для каждой сетевой карты, используемой для создания виртуального сетевого интерфейса bond0.

```
[root@rhel6 network-scripts]# cat ifcfg-eth1
DEVICE=eth1
BOOTPROTO=none
ONBOOT=yes
MASTER=bond0
SLAVE=yes
USERCTL=no
[root@rhel6 network-scripts]# cat ifcfg-eth2
DEVICE=eth2
BOOTPROTO=none
ONBOOT=yes
MASTER=bond0
SLAVE=yes
USERCTL=no
```

Наконец, мы активируем сетевой интерфейс с помощью команды ifup bond0.

```
[root@rhel6 network-scripts]# ifup bond0
[root@rhel6 network-scripts]# ifconfig bond0
bond0 Link encap:Ethernet HWaddr 08:00:27:DA:C1:49
inet addr:192.168.1.199 Bcast:192.168.1.255 Mask:255.255.255.0
inet6 addr: fe80::a00:27ff:feda:c149/64 Scope:Link
UP BROADCAST RUNNING MASTER MULTICAST MTU:1500 Metric:1
RX packets:251 errors:0 dropped:0 overruns:0 frame:0
TX packets:21 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:39852 (38.9 KiB) TX bytes:1070 (1.0 KiB)
```

Информация о виртуальном сетевом интерфейсе bond0 также будет присутствовать в одноименном файле из директории /proc/net/bonding.

```
[root@rhel6 network-scripts]# cat /proc/net/bonding/bond0
```

Ethernet Channel Bonding Driver: v3.5.0 (November 4, 2008)
Bonding Mode: load balancing (round-robin)
MII Status: up
MII Polling Interval (ms): 0
Up Delay (ms): 0
Down Delay (ms): 0
Slave Interface: eth1
MII Status: up
Link Failure Count: 0
Permanent HW addr: 08:00:27:da:c1:49
Slave Interface: eth2
MII Status: up
Link Failure Count: 0
Permanent HW addr: 08:00:27:40:03:3b

24.4. Связывание сетевых интерфейсов в дистрибутивах Debian/Ubuntu

Начнем с выполнения команды ifconfig -a для получения списка всех сетевых карт, установленных в компьютере.

```
debian5:~# ifconfig -a | grep Ethernet
eth0 Link encap:Ethernet HWaddr 08:00:27:bb:18:a4
eth1 Link encap:Ethernet HWaddr 08:00:27:63:9a:95
eth2 Link encap:Ethernet HWaddr 08:00:27:27:a4:92
```

В данном примере мы будем связывать сетевые интерфейсы eth1 и eth2.

Также нам придется установить пакет программного обеспечения ifenslave.

```
debian5:~# aptitude search ifenslave
p ifenslave - Attach and detach slave interfaces to a bonding device
p ifenslave-2.6 - Attach and detach slave interfaces to a bonding device
debian5:~# aptitude install ifenslave
Чтение списков пакетов... Готово
...
```

После этого нам придется отредактировать файл конфигурации /etc/network/interfaces, добавив в него информацию о виртуальном сетевом интерфейсе bond0.

```
debian5:~# tail -7 /etc/network/interfaces
iface bond0 inet static
address 192.168.1.42
netmask 255.255.255.0
gateway 192.168.1.1
slaves eth1 eth2
bond-mode active-backup
bond_primary eth1
```

В более старых версиях дистрибутивов Debian/Ubintu вам придется вручную выполнить команду modprobe bonding, но в современных версиях данных дистрибутивов этого уже не требуется. Используйте утилиту ifup для активации сетевого интерфейса и проверьте его работоспособность.

```
debian5:~# ifup bond0
debian5:~# ifconfig bond0
bond0 Link encap:Ethernet HWaddr 08:00:27:63:9a:95
inet addr:192.168.1.42 Bcast:192.168.1.255 Mask:255.255.255.0
inet6 addr: fe80::a00:27ff:fe63:9a95/64 Scope:Link
UP BROADCAST RUNNING MASTER MULTICAST MTU:1500 Metric:1
RX packets:212 errors:0 dropped:0 overruns:0 frame:0
TX packets:39 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:31978 (31.2 KiB) TX bytes:6709 (6.5 KiB)
```

Информация о виртуальном сетевом интерфейсе bond0 также будет присутствовать в одноименном файле из директории /proc/net/bonding/.

```
debian5:~# cat /proc/net/bonding/bond0
Ethernet Channel Bonding Driver: v3.2.5 (March 21, 2008)
Bonding Mode: fault-tolerance (active-backup)
Primary Slave: eth1
Currently Active Slave: eth1
MII Status: up
MII Polling Interval (ms): 0
Up Delay (ms): 0
Down Delay (ms): 0
Slave Interface: eth1
MII Status: up
Link Failure Count: 0
Permanent HW addr: 08:00:27:63:9a:95
Slave Interface: eth2
MII Status: up
Link Failure Count: 0
Permanent HW addr: 08:00:27:27:a4:92
```

24.5. Практическое задание: добавление IP-адресов и связывание сетевых интерфейсов

1. Свяжите дополнительный IP-адрес с одной из ваших сетевых карт. Проверьте его работоспособность (может ли ваш сосед использовать данный IP-адрес для входа в вашу систему с использованием протокола ssh)!
2. Используйте утилиту ifdown для деактивации данного IP-адреса.
3. Убедитесь в том, что ваш сосед также успешно связал

дополнительный IP-адрес с сетевой картой перед продолжением выполнения заданий.

4. Добавьте дополнительную сетевую карту (или две) в вашу виртуальную машину и используйте информацию из теоретической части главы для связывания двух сетевых карт.

24.6. Корректная процедура выполнения практического задания: добавление IP-адресов и связывание сетевых интерфейсов

1. Свяжите дополнительный IP-адрес с одной из ваших сетевых карт. Проверьте его работоспособность (может ли ваш сосед использовать данный IP-адрес для входа в вашу систему с использованием протокола ssh)!

В дистрибутивах RedHat/Fedora:

Следует добавить файлы конфигурации /etc/sysconfig/network-scripts/ifcfg-ethX:X, содержащие описания виртуальных сетевых интерфейсов, таким образом, как показано в теоретической части главы.

В дистрибутивах Debian/Ubuntu:

Следует модифицировать файл конфигурации /etc/network/interfaces, добавив в него описания виртуальных сетевых интерфейсов таким образом, как показано в теоретической части главы.

1. Используйте утилиту ifdown для деактивации данного IP-адреса.

ifdown eth0:0

1. Убедитесь в том, что ваш сосед также успешно связал дополнительный IP-адрес с сетевой картой перед продолжением выполнения заданий.

ping $дополнительный_ip_адрес_соседа
или
ssh $дополнительный_ip_адрес_соседа

1. Добавьте дополнительную сетевую карту (или две) в вашу виртуальную машину и используйте информацию из теоретической части главы для связывания двух сетевых карт.

В дистрибутивах RedHat/Fedora:

Следует добавить в директорию /etc/sysconfig/network-scripts файлы ifcfg-ethX и ifcfg-bondX, заполненные таким образом, как показано в теоретической части главы. Также не забудьте о необходимости редактирования файла конфигурации утилиты modprobe.

В дистрибутивах Debian/Ubuntu:

Следует модифицировать файл /etc/network/interfaces таким образом, как показано в теоретической части главы. Не забудьте о необходимости установки пакета программного обеспечения ifenslave.

Глава 25. Клиент и сервер ssh

Безопасная командная оболочка (secure shell, ssh) является набором инструментов, использующих защищенный протокол для взаимодействия с удаленными системами Linux.

В данной главе проведен обзор наиболее часто используемых команд, связанных с сервером sshd и клиентом ssh.

25.1. О протоколе ssh

Избегайте использования утилит telnet, rlogin и rsh для удаленного подключения к вашим серверам. Эти инструменты используют устаревшие протоколы, не предусматривающие шифрования данных сессии входа в систему, следовательно, ваши имя пользователя и пароль могут быть перехвачены с помощью таких инструментов, как wireshark или tcpdump. Для безопасного подключения к вашим серверам следует использовать протокол ssh.

Протокол ssh предусматривает два уровня защиты. Во-первых ваше соединение шифруется, а во-вторых, перед установкой соединения осуществляется двухсторонняя аутентификация.

Соединение по протоколу ssh всегда начинается с криптографического рукопожатия, после которого начинается передача зашифрованного с помощью симметричного алгоритма на транспортном уровне трафика. Другими словами, шифрование данных в туннеле начинается до того момента, когда вы начнете что-либо писать.

После установки защищенного соединения осуществляется аутентификация пользователя (с использованием имени/пароля пользователя или публичного/закрытого ключей) и начинается передача данных по защищенному соединению.

Протокол ssh предусматривает возможность сохранения данных о серверах, с которыми осуществлялось взаимодействие (поэтому при каких-либо подозрениях о ненадежности устанавливаемого соединения вы будете немедленно уведомлены).

Пакет openssh сопровождается разработчиками проекта OpenBSD и распространяется в составе множества операционных систем (вполне возможно, что это самый популярный пакет в мире).

25.1.2. Директория для файлов конфигурации /etc/ssh/

Файлы конфигурации клиента и сервера ssh находятся в директории /etc/ssh. В следующих разделах мы обсудим большинство файлов конфигурации из данной директории.

25.1.3. Версии протокола ssh

Существуют две версии протокола ssh (а именно, версии 1 и 2). Избегайте использовании версии 1 везде, где это возможно, так как данная версия содержит ряд известных уязвимостей. Вы можете управлять версией протокола, используемой клиентом ssh, с помощью файла конфигурации /etc/ssh/ssh_config, а демоном, являющимся сервером openssh — с помощью файла конфигурации /etc/ssh/sshd_config.

paul@ubu1204:/etc/ssh$ grep Protocol ssh_config
Protocol 2,1
paul@ubu1204:/etc/ssh$ grep Protocol sshd_config
Protocol 2

25.1.4. Публичные и закрытые ключи

Протокол ssh предусматривает использование известной системы публичных и закрытых ключей. Ниже приведено сжатое объяснение принципа использования данных ключей; дополнительная информация может быть найдена в энциклопедии Wikipedia.

http://en.wikipedia.org/wiki/Public-key_cryptography

Представьте, что два человека, Элис и Боб, хотят пообщаться друг с другом. Воспользовавшись публичными и закрытыми ключами, они могут общаться с шифрованием данных и аутентификацией.

Если Элис захочет отправить зашифрованное сообщение Бобу, она использует публичный ключ Боба. Ранее Боб должен был передать свой публичный ключ Элис, но не передавать свой закрытый ключ никому! Так как закрытого ключа Боба нет ни у кого кроме самого Боба, Элис может быть уверена в том, что зашифрованное сообщение не сможет прочитать никто кроме Боба.

Если Боб захочет проверить, действительно ли данное сообщение было отправлено Элис, он использует публичный ключ Элис для проверки того, действительно ли Элис подписала свое сообщение с помощью своего закрытого ключа. Так как закрытого ключа Элис нет ни у кого кроме самой Элис, Боб может быть уверен в том, что сообщение было отправлено Элис.

25.1.5. Алгоритмы rsa и dsa

В данной главе не приводятся пояснения относительно реализации криптографических алгоритмов, а даются лишь рекомендации по поводу задействования алгоритмов rsa и dsa при работе с инструментами из комплекта поставки ssh. Дополнительная информация об упомянутых алгоритмах может быть найдена в энциклопедии Wikipedia:

http://en.wikipedia.org/wiki/RSA_(algorithm)
http://en.wikipedia.org/wiki/Digital_Signature_Algorithm

25.2. Вход в удаленную систему

В следующем примере показана методика использования клиента ssh для входа в удаленную систему Linux. Локальный пользователь с именем paul осуществляет вход в удаленную систему под именем admin42.

paul@ubu1204:~$ ssh admin42@192.168.1.30
The authenticity of host 192.168.1.30 (192.168.1.30) can't be established.
RSA key fingerprint is b5:fb:3c:53:50:b4:ab:81:f3:cd:2e:bb:ba:44:d3:75.
Are you sure you want to continue connecting (yes/no)?

Как вы видите, пользователь paul представлен с помощью соответствующего отпечатка ключа аутентификации rsa (rsa autentification fingerprint) на стороне удаленной системы. Локальный пользователь может принять его, введя слово yes. Как мы увидим позднее, данный отпечаток ключа аутентификации будет добавлен в файл ~/.ssh/known_hosts.

paul@ubu1204:~$ ssh admin42@192.168.1.30
The authenticity of host 192.168.1.30 (192.168.1.30) can't be established.
RSA key fingerprint is b5:fb:3c:53:50:b4:ab:81:f3:cd:2e:bb:ba:44:d3:75.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 192.168.1.30 (RSA) to the list of known hosts.
admin42@192.168.1.30's password:
Welcome to Ubuntu 12.04 LTS (GNU/Linux 3.2.0-26-generic-pae i686)
- Documentation: https://help.ubuntu.com/

1 package can be updated.
0 updates are security updates.
Last login: Wed Jun 6 19:25:57 2012 from 172.28.0.131
admin42@ubuserver:~$

Пользователь может завершить работу с удаленной системой, выполнив команду exit или использовав сочетание клавиш Ctrl+D.

admin42@ubuserver:~$ exit
logout
Connection to 192.168.1.30 closed.
paul@ubu1204:~$

25.3. Выполнение команд в удаленной системе

В примере ниже показана методика выполнения единственной команды pwd в удаленной системе. В данном случае нет необходимости в явном завершении работы с системой с помощью команды exit.

paul@ubu1204:~$ ssh admin42@192.168.1.30 pwd

admin42@192.168.1.30's password:
/home/admin42
paul@ubu1204:~$

25.4. Утилита scp

Утилита scp работает аналогично утилите cp, но при этом позволяет использовать протокол ssh при задании путей к исходному и целевому файлам. Ниже приведен пример копирования файла /etc/hosts с удаленного сервера в домашнюю директорию пользователя paul.

paul@ubu1204:~$ scp admin42@192.168.1.30:/etc/hosts /home/paul/serverhosts
admin42@192.168.1.30's password:
hosts 100% 809 0.8KB/s 00:00

А это обратный пример копирования локального файла в директорию на удаленном сервере.

paul@ubu1204:$ scp /serverhosts admin42@192.168.1.30:/etc/hosts.new
admin42@192.168.1.30's password:
serverhosts 100% 809 0.8KB/s 00:00

25.5. Настройка соединений по протоколу ssh без использования паролей

Для настройки соединения по протоколу ssh на основе публичных/закрытых ключей без использования паролей следует в первую очередь воспользоваться утилитой ssh-keygen с целью генерации пары ключей без использования фразы пароля, после чего скопировать ваш сгенерированный публичный ключ на удаленный сервер. Давайте поэтапно выполним описанные действия.

В следующем примере мы будем настраивать соединение по протоколу ssh без использования паролей для Элис и Боба. У Элис есть учетная запись на сервере, работающем под управлением дистрибутива Red Hat Enterprise Linux, а Боб использует дистрибутив Ubuntu на своем ноутбуке. Боб желает предоставить Элис доступ к своей системе по протоколу ssh с использованием публичных и закрытых ключей. Это означает, что даже в том случае, если Боб сменит свой пароль для входа в систему, используемую на ноутбуке, у Элис все так же будет доступ к его системе.

25.5.1. Утилита ssh-keygen

В примере ниже показано, как Элис использовала утилиту ssh-keygen для генерации пары ключей. Элис не вводила ключевую фразу.

[alice@RHEL5 ~]$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/alice/.ssh/id_rsa):
Created directory /home/alice/.ssh.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/alice/.ssh/id_rsa.
Your public key has been saved in /home/alice/.ssh/id_rsa.pub.
The key fingerprint is:
9b:ac:ac:56:c2:98:e5:d9:18:c4:2a:51:72:bb:45:eb alice@RHEL5
[alice@RHEL5 ~]$

Вы можете использовать команду ssh-keygen -t dsa аналогичным образом.

25.5.2. Директория ~/.ssh

Помимо того, что утилита ssh-keygen генерирует публичные и закрытые ключи, она также создает скрытую директорию .ssh с необходимыми правами доступа. Вы можете создать директорию .ssh вручную, но в этом случае вам придется выполнить команду chmod 700 для установки необходимых прав доступа к ее содержимому! В противном случае клиент ssh откажется от использования ключей (так как использование общедоступных закрытых ключей не является безопасной практикой!).

Как вы видите, содержимое директории .ssh, находящейся в домашней директории Элис, надежно защищено.

[alice@RHEL5 ~]$ ls -ld .ssh
drwx------ 2 alice alice 4096 май 1 07:38 .ssh
[alice@RHEL5 ~]$

Боб использует дистрибутив Ubuntu дома. Он решил самостоятельно создать директорию .ssh, поэтому ему придется позаботиться и о ее безопасности.

bob@laika:~$ mkdir .ssh
bob@laika:~$ ls -ld .ssh
drwxr-xr-x 2 bob bob 4096 2008-05-14 16:53 .ssh
bob@laika:~$ chmod 700 .ssh/
bob@laika:~$

25.5.3. Файлы ключей id_rsa и id_rsa.pub

Утилита ssh-keygen генерирует файлы ключей в директории .ssh. Файл публичного ключа носит имя /.ssh/id_rsa.pub. Файл закрытого ключа носит имя /.ssh/id_rsa.

[alice@RHEL5 ~]$ ls -l .ssh/
итого 16
-rw------- 1 alice alice 1671 май 1 07:38 id_rsa
-rw-r—r-- 1 alice alice 393 май 1 07:38 id_rsa.pub

При использовании алгоритма dsa вместо алгоритма rsa файлы будут носить имена id_dsa и id_dsa.pub.

25.5.4. Распространение публичных ключей по другим системам

Элис решила использовать утилиту scp для копирования файла публичного ключа со своего сервера на ноутбук Боба.

[alice@RHEL5 .ssh]$ scp id_rsa.pub bob@192.168.48.92:~/.ssh/authorized_keys
bob@192.168.48.92's password:
id_rsa.pub 100% 393 0.4KB/s 00:00

Проявляйте осторожность при копировании второго файла публичного ключа! Вы должны не перезаписывать файл первого публичного ключа, а добавить данные из файла нового публичного ключа в файл ~/.ssh/authorized_keys!

cat id_rsa.pub >> ~/.ssh/authorized_keys

Элис также могла бы использовать утилиту ssh-copy-id таким образом, как показано в примере ниже.

ssh-copy-id -i .ssh/id_rsa.pub bob@192.168.48.92

25.5.5. Файл authorized_keys

Вы можете создать файл с именем authorized_keys в директории ~/.ssh вашей системы. Данный файл может содержать один или большее количество публичных ключей от людей, которым вы доверяете. Эти доверенные лица могут использовать свои закрытые ключи для подтверждения своей идентичности и получения доступа к вашей учетной записи по протоколу ssh (без ввода пароля). В примере показано содержимое файла authorized_keys из системы Боба, в котором сохранен публичный ключ Элис.

bob@laika:~$ cat .ssh/authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAQEApCQ9xzyLzJes1sR+hPyqW2vyzt1D4zTLqk\
MDWBR4mMFuUZD/O583I3Lg/Q+JIq0RSksNzaL/BNLDou1jMpBe2Dmf/u22u4KmqlJBfDhe\
yTmGSBzeNYCYRSMq78CT9I9a+y6x/shucwhaILsy8A2XfJ9VCggkVtu7XlWFDL2cum08/0\
mRFwVrfc/uPsAn5XkkTscl4g21mQbnp9wJC40pGSJXXMuFOk8MgCb5ieSnpKFniAKM+tEo\
/vjDGSi3F/bxu691jscrU0VUdIoOSo98HUfEf7jKBRikxGAC7I4HLa+/zX73OIvRFAb2hv\
tUhn6RHrBtUJUjbSGiYeFTLDfcTQ== alice@RHEL5

25.5.6. Соединение по протоколу ssh без использования паролей

Теперь Элис может использовать клиент ssh для соединения с ноутбуком Боба без использования пароля. Принимая во внимание возможность использования протокола ssh для выполнения команд на уделенных системах, можно сделать вывод о том, что приложения из различных систем могут объединяться каналами для выполнения полезной работы.

[alice@RHEL5 ~]$ ssh bob@192.168.48.92 "ls -l .ssh"
итого 4
-rw-r—r-- 1 bob bob 393 2008-05-14 17:03 authorized_keys
[alice@RHEL5 ~]$

25.6. Перенаправление сессии оконной системы X11 по протоколу ssh

Другой популярной возможностью клиента ssh является возможность перенаправления сессии оконной системы X11 (X11 forwarding) по протоколу ssh с помощью команды ssh -X.

Ниже приведен пример перенаправления сессии оконной системы X11 по протоколу ssh: пользователь paul входит со своего компьютера под именем greet в удаленную систему с целью запуска приложения с графическим интерфейсом mozilla-thunderbird. Несмотря на то, что приложение будет исполняться на удаленном компьютере от лица пользователя greet, окно приложения будет показано на мониторе, подключенном к локальному компьютеру пользователя paul.

paul@debian5:~/PDF$ ssh -X greet@greet.dyndns.org -p 55555
Warning: Permanently added the RSA host key for IP address \
81.240.174.161 to the list of known hosts.
Password:
Linux raika 2.6.8-2-686 #1 Tue Aug 16 13:22:48 UTC 2005 i686 GNU/Linux
Last login: Thu Jan 18 12:35:56 2007
greet@raika:~$ ps fax | grep thun
greet@raika:~$ mozilla-thunderbird &
[1] 30336

25.7. Отладка соединений по протоколу ssh

Используйте команду ssh -v для получения отладочной информации о попытке соединения по протоколу ssh.

paul@debian5:~$ ssh -v bert@192.168.1.192
OpenSSH_4.3p2 Debian-8ubuntu1, OpenSSL 0.9.8c 05 Sep 2006
debug1: Reading configuration data /home/paul/.ssh/config
debug1: Reading configuration data /etc/ssh/ssh_config
debug1: Applying options for *
debug1: Connecting to 192.168.1.192 [192.168.1.192] port 22.
debug1: Connection established.
debug1: identity file /home/paul/.ssh/identity type -1
debug1: identity file /home/paul/.ssh/id_rsa type 1
debug1: identity file /home/paul/.ssh/id_dsa type -1
debug1: Remote protocol version 1.99, remote software version

OpenSSH_3
debug1: match: OpenSSH_3.9p1 pat OpenSSH_3.*
debug1: Enabling compatibility mode for protocol 2.0
...

25.8. Сервер sshd

Исполняемый файл сервера ssh носит имя sshd и содержится в пакете программного обеспечения openssh-server.

root@ubu1204~# dpkg -l openssh-server | tail -1
ii openssh-server 1:5.9p1-5ubuntu1 secure shell (SSH) server,...

25.9. Ключи сервера sshd

Файлы публичных ключей, используемые сервером sshd, располагаются в директории /etc/ssh и могут читаться всеми пользователями. Файлы закрытых ключей могут читаться исключительно пользователем root.

root@ubu1204~# ls -l /etc/ssh/ssh_host_*
-rw------- 1 root root 668 июн 7 2011 /etc/ssh/ssh_host_dsa_key
-rw-r—r-- 1 root root 598 июн 7 2011 /etc/ssh/ssh_host_dsa_key.pub
-rw------- 1 root root 1679 июн 7 2011 /etc/ssh/ssh_host_rsa_key
-rw-r—r-- 1 root root 390 июн 7 2011 /etc/ssh/ssh_host_rsa_key.pub

25.10. Демон ssh-agent

При генерации ключей с помощью утилиты ssh-keygen у вас имеется возможность ввести ключевую фразу для ограничения доступа к этим ключам. Для того, чтобы не вводить ключевую фразу каждый раз при использовании файлов закрытых ключей, вы можете добавить закрытый ключ в базу данных демона ssh-agent с помощью утилиты ssh-add.

В большинстве дистрибутивов Linux демон ssh-agent запускается автоматически после входа пользователя в систему.

root@ubu1204~# ps -ef | grep ssh-agent
paul 2405 2365 0 08:13 ? 00:00:00 /usr/bin/ssh-agent...

В данном примере показана методика использования утилиты ssh-add для вывода списка открытых ключей, которые соответствуют закрытым ключам, добавленным в базу данных демона ssh-agent.

paul@debian5:~$ ssh-add -L
ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAQEAvgI+Vx5UrIsusZPl8da8URHGsxG7yiv
v3/\
...
wMGqa48Kelwom8TGb4Sgcwpp/VO/ldA5m+BGCw== paul@deb503

25.11. Практическое задание: ssh

1. Убедитесь в том, что у вас есть доступ к двум компьютерам, работающим под управлением Linux, или объединитесь с соседом для выполнения этого практического задания. В этом практическом задании мы будем называть один из используемых компьютеров сервером.
2. Установите сервер sshd на компьютере, выступающем в роли сервера.
3. Проверьте файлы конфигурации на наличие директив, позволяющих использовать исключительно версию 2 протокола ssh.
4. Используйте клиент ssh для входа в используемую на сервере систему, вывода информации о вашей текущей директории и завершения работы с системой.
5. Используйте утилиту scp для копирования файла с вашего компьютера на сервер.
6. Используйте утилиту scp для копирования файла с сервера на ваш компьютер.
7. (Необязательное задание, которое может быть выполнено только в том случае, если у вас установлены дистрибутивы Linux с программными компонентами для реализации графических интерфейсов.) Установите пакет программного обеспечения xeyes на сервере и используйте клиент ssh для запуска утилиты xeyes на сервере с показом окна утилиты на мониторе клиентского компьютера.
8. (Необязательное задание, которое может быть выполнено при тех же условиях, что и предыдущее.) Создайте закладку в запущенном на сервере браузере Firefox, после чего завершите работу браузера на клиентском компьютере и сервере. Используйте команду ssh -X для запуска браузера Firefox на компьютере вашего соседа с показом окна на мониторе вашего компьютера. Видите ли вы закладку, созданную вашим соседом?
9. Используйте утилиту ssh-keygen для создания пары ключей без использования ключевой фразы. Настройте соединение по протоколу ssh без использования паролей между вашим компьютером и компьютером вашего соседа (или между клиентом и сервером).
10. Проверьте корректность прав доступа к файлам ключей сервера sshd; файлы публичных ключей должны читаться любым пользователем, а файлы закрытых ключей — только пользователем root.
11. Проверьте, исполняется ли демон ssh-agent.
12. (Необязательное задание) Защитите доступ к паре ваших файлов ключей с помощью ключевой фразы, после чего добавьте эти ключи в базу данных демона ssh-agent и проверьте работоспособность вашего соединения по протоколу ssh без использования пароля.

25.12. Корректная процедура выполнения практического задания: ssh

1. Убедитесь в том, что у вас есть доступ к двум компьютерам, работающим под управлением Linux, или объединитесь с соседом для выполнения этого практического задания. В этом практическом задании мы будем называть один из используемых компьютеров сервером.
2. Установите сервер sshd на компьютере, выступающем в роли сервера.

apt-get install openssh-server (в дистрибутивах Ubuntu/Debian)
yum -y install openssh-server (в дистрибутивах Centos/Fedora/Red Hat)

1. Проверьте файлы конфигурации на наличие директив, позволяющих использовать исключительно версию 2 протокола ssh.

grep Protocol /etc/ssh/ssh*_config

1. Используйте клиент ssh для входа в используемую на сервере систему, вывода информации о вашей текущей директории и завершения работы с системой.
   user@client$ ssh имя_пользователя@ip_адрес_сервера

user@server$ pwd
/home/user
user@server$ exit

1. Используйте утилиту scp для копирования файла с вашего компьютера на сервер.

scp локальный_файл имя_пользователя@ip_адрес_сервера:~

1. Используйте утилиту scp для копирования файла с сервера на ваш компьютер.

scp имя_пользователя@ip_адрес_сервера:~/файл_с_сервера .

1. (Необязательное задание, которое может быть выполнено только в том случае, если у вас установлены дистрибутивы Linux с программными компонентами для реализации графических интерфейсов.) Установите пакет программного обеспечения xeyes на сервере и используйте клиент ssh для запуска утилиты xeyes на сервере с показом окна утилиты на мониторе клиентского компьютера.

На сервере:
apt-get install xeyes
На клиенте:
ssh -X имя_пользователя@ip_адрес_сервера
xeyes

1. (Необязательное задание, которое может быть выполнено при тех же условиях, что и предыдущее) Создайте закладку в запущенном на сервере браузере Firefox, после чего завершите работу браузера на клиентском компьютере и сервере. Используйте команду ssh -X для запуска браузера Firefox на компьютере вашего соседа с показом окна на мониторе вашего компьютера. Видите ли вы закладку, созданную вашим соседом?
2. Используйте утилиту ssh-keygen для создания пары ключей без использования ключевой фразы. Настройте соединение по протоколу ssh без использования паролей между вашим компьютером и компьютером вашего соседа (или между клиентом и сервером).

Решение подробно описано в разделе «Настройка соединений по протоколу ssh без использования паролей» теоретической части главы.

1. Проверьте корректность прав доступа к файлам ключей сервера sshd; файлы публичных ключей должны читаться любым пользователем, а файлы закрытых ключей — только пользователем root.

ls -l /etc/ssh/ssh_host_*

1. Проверьте, исполняется ли демон ssh-agent.

ps fax | grep ssh-agent

1. (Необязательное задание) Защитите доступ к паре ваших файлов ключей с помощью ключевой фразы, после чего добавьте эти ключи в базу данных демона ssh-agent и проверьте работоспособность вашего соединения по протоколу ssh без использования пароля.

Информация может быть найдена на страницах следующих руководств:
man ssh-keygen
man ssh-agent
man ssh-add

Глава 26. Краткая информация о сетевой файловой системе

Сетевая файловая система (Network File System или NFS) с восьмидесятых годов прошлого века позволяет предоставлять доступ к директориям локальной файловой системы другим компьютерам из сети.

В данной главе будут рассматриваться вопросы настройки сервера и клиента NFS.

26.1. Версии протокола NFS

Устаревшие версии протокола NFS 2 и 3 по умолчанию не сохраняют состояния (используют протокол UDP, хотя и могут работать по протоколу TCP). Более новая версия протокола NFS 4

поддерживает возможность сохранения состояния, имеет лучшую производительность и более надежную защиту.

Версия 4 протокола NFS была описана в стандартах RFC 3010 в 2000 году и RFC 3530 в 2003 году, а также требует использования протокола TCP (на порту 2049). Кроме того, она поддерживает механизм аутентификации пользователей Kerberos в качестве варианта системы контроля доступа к монтируемой файловой системе. Аутентификация в версиях протокола NFS 2 и 3 происходила исключительно на уровне узлов.

26.2. Утилита rpcinfo

Клиенты соединяются с сервером с помощью технологии удаленного вызова процедур (в Linux для этих целей задействуется демон portmap). Рассмотрите вывод утилиты rpcinfo, чтобы убедиться в том, что обработчики запросов NFS и соответствующие службы исполняются.

```
root@RHELv4u2:~# /etc/init.d/portmap status
portmap (pid 1920) is running...
root@RHELv4u2:~# rpcinfo -p
program vers proto port
100000 2 tcp 111 portmapper
100000 2 udp 111 portmapper
100024 1 udp 32768 status
100024 1 tcp 32769 status
root@RHELv4u2:~# service nfs start
Starting NFS services: [ OK ]
Starting NFS quotas: [ OK ]
Starting NFS daemon: [ OK ]
Starting NFS mountd: [ OK ]
```

Вывод той же утилиты rpcinfo после запуска служб, ответственных за работу с сетевой файловой системой.

```
root@RHELv4u2:~# rpcinfo -p
program vers proto port
100000 2 tcp 111 portmapper
100000 2 udp 111 portmapper
100024 1 udp 32768 status
100024 1 tcp 32769 status
100011 1 udp 985 rquotad
100011 2 udp 985 rquotad
100011 1 tcp 988 rquotad
100011 2 tcp 988 rquotad
100003 2 udp 2049 nfs
100003 3 udp 2049 nfs
100003 4 udp 2049 nfs
100003 2 tcp 2049 nfs
100003 3 tcp 2049 nfs
100003 4 tcp 2049 nfs
100021 1 udp 32770 nlockmgr
100021 3 udp 32770 nlockmgr
100021 4 udp 32770 nlockmgr
100021 1 tcp 32789 nlockmgr
100021 3 tcp 32789 nlockmgr
100021 4 tcp 32789 nlockmgr
100005 1 udp 1004 mountd
100005 1 tcp 1007 mountd
100005 2 udp 1004 mountd
100005 2 tcp 1007 mountd
100005 3 udp 1004 mountd
100005 3 tcp 1007 mountd
```

26.3. Настройка сервера

Сервер NFS настраивается с помощью файла конфигурации /etc/exports. Также вам может понадобиться какой-либо механизм синхронизации идентификаторов пользователей между компьютерами в случае масштабного использования сетевой файловой системы (возможно, таким механизмом является LDAP).

Параметр rootsquash позволяет заменять идентификатор пользователя root (0) на идентификатор пользователя nobody (или аналогичного пользователя). Параметр sync позволяет активировать механизм принудительной записи данных на диск перед завершением выполнения запроса от клиента.

26.4. Файл конфигурации /etc/exports

Ниже приведен пример файла конфигурации /etc/exports с пояснениями относительно синтаксиса.

```
paul@laika:~$ cat /etc/exports
# Everyone can read this share (Каждый пользователь имеет возможность чтения данных из файлов разделяемой файловой системы)
/mnt/data/iso *(ro)
# Only the computers named pasha and barry can readwrite this one (Читать и записывать данные в файлы экспортируемой файловой системы и могут исключительно пользователи компьютеров pasha и barry)
/var/www pasha(rw) barry(rw)
# same, but without root squashing for barry (Аналогично, но пользователю root компьютера barry разрешается работать с файловой системой без изменения идентификатора пользователя)
/var/ftp pasha(rw) barry(rw,no_root_squash)
# everyone from the netsec.local domain gets access (Каждый пользователь узла с доменом netsec.local получает полный доступ
```

к файловой системе)
/var/backup *.netsec.local(rw)
ro for one network, rw for the other (Для пользователей из одной сети разрешено только чтение из файловой системы, из другой — чтение и запись в файловую систему)
/var/upload 192.168.1.0/24(ro) 192.168.5.0/24(rw)
Более новые версии протокола NFS требуют явной активации механизма проверки принадлежности запрошенных файлов к экспортируемым поддиректориям файловых систем с помощью параметра subtree_check (или его явной деактивации с помощью параметра no_subtree_check). В этом случае файл конфигурации /etc/exports будет выглядеть аналогичным образом:
root@debian6 ~# cat /etc/exports
Everyone can read this share (Каждый пользователь имеет возможность чтения данных из файлов разделяемой файловой системы)
/srv/iso *(ro,no_subtree_check)
Only the computers named pasha and barry can readwrite this one (Читать и записывать данные в файлы экспортируемой файловой системы и могут исключительно пользователи компьютеров pasha и barry)
/var/www pasha(rw,no_subtree_check) barry(rw,no_subtree_check)
same, but without root squashing for barry (Аналогично, но пользователю root компьютера barry разрешается работать с файловой системой без изменения идентификатора пользователя)
/var/ftp pasha(rw,no_subtree_check) barry(rw,no_root_squash,no_subtree_check)

26.5. Утилита exportfs

Вам не придется перезапускать сервер NFS для экспорта файловых систем, соответствующих добавленным в файл конфигурации описаниями. Для этой цели вы можете просто воспользоваться командой exportfs -va. Она позволяет дописать пути директориям в файл /var/lib/nfs/etab, после чего они будут немедленно экспортированы.

```
root@debian6 ~# exportfs -va
exporting pasha:/var/ftp
exporting barry:/var/ftp
exporting pasha:/var/www
exporting barry:/var/www
exporting *:/srv/iso
```

26.6. Настройка клиента

Ранее мы рассматривали утилиту mount и файл /etc/fstab.

```
root@RHELv4u2:~# mount -t nfs barry:/mnt/data/iso /home/project55/
root@RHELv4u2:~# cat /etc/fstab | grep nfs
barry:/mnt/data/iso /home/iso nfs defaults 0 0
root@RHELv4u2:~#
```

Ниже приведен другой простой пример. Представьте, что разработчики проекта project55 сообщили вам о том, что им требуется несколько файлов образов дисков CD-ROM, причем эти файлы уже размещены вами в директории, экспортируемой средствами сервера NFS. Вы должны выполнить следующую команду для монтирования этой разделяемой директории в точку монтирования /home/project55.

```
root@RHELv4u2:~# mount -t nfs 192.168.1.40:/mnt/data/iso /home/project55/
root@RHELv4u2:~# ls -lh /home/project55/
итого 3.6G
drwxr-xr-x 2 1000 1000 4.0K янв 16 17:55 RHELv4u1
drwxr-xr-x 2 1000 1000 4.0K янв 16 14:14 RHELv4u2
drwxr-xr-x 2 1000 1000 4.0K янв 16 14:54 RHELv4u3
drwxr-xr-x 2 1000 1000 4.0K янв 16 11:09 RHELv4u4
-rw-r—r-- 1 root root 1.6G окт 13 15:22 sled10-vmwarews5-vm.zip
root@RHELv4u2:~#
```

26.7. Практическое задание: вводная информация о сетевой файловой системе

1. Создайте две директории с какими-либо файлами. Используйте сервер NFS для экспорта первой директории в режиме только для чтения, а второй — в режиме для чтения и записи. Ваш сосед должен смонтировать данные директории и проверить корректность расстановки прав доступа.
2. Выясните имя пользователя, владеющего файлами, которые создал ваш сосед в разделяемой файловой системе.
3. Ограничьте доступ к разделяемой директории, указав IP-адрес или имя узла компьютера вашего соседа.

Глава 27. Краткая информация о сетевых службах

27.1. Вводная информация о межсетевом экране iptables

27.1.1. Межсетевой экран iptables

В состав ядра Linux включен межсетевой экран с сохранением состояний, носящий имя iptables. Для остановки межсетевого экрана iptables в дистрибутиве Red Hat следует использовать утилиту service.

```
root@RHELv4u4:~# service iptables stop
```

Flushing firewall rules: [OK]
Setting chains to policy ACCEPT: filter [OK]
Unloading iptables modules: [OK]
root@RHELv4u4:~#

Простейший способ настройки межсетевого экрана iptables заключается в использовании инструмента с графическим интерфейсом, такого, как утилита kmyfirewall из состава окружения рабочего стола KDE или утилита Security Level Configuration Tool. Последнюю утилиту вы можете найти в меню приложений окружения рабочего стола в разделе «Системные утилиты — Безопасность» или запустить вручную, выполнив команду system-config-securitylevel в командной оболочке bash. Упомянутые инструменты позволяют осуществлять элементарную настройку межсетевого экрана. У вас имеется возможность принятия решения о том, следует ли активировать межсетевой экран, а также о том, какие стандартные порты должны быть открыты при активации межсетевого экрана. Вы можете даже добавить некоторые нестандартные порты. После того, как вы завершите настройку межсетевого экрана, в дистрибутиве Red Hat соответствующие правила будут записаны в файл конфигурации межсетевого экрана /etc/sysconfig/iptables.

root@RHELv4u4:~# cat /etc/sysconfig/iptables
Firewall configuration written by system-config-securitylevel
Manual customization of this file is not recommended.
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
:RH-Firewall-1-INPUT - [0:0]
-A INPUT -j RH-Firewall-1-INPUT
-A FORWARD -j RH-Firewall-1-INPUT
-A RH-Firewall-1-INPUT -i lo -j ACCEPT
-A RH-Firewall-1-INPUT -p icmp --icmp-type any -j ACCEPT
-A RH-Firewall-1-INPUT -p 50 -j ACCEPT
-A RH-Firewall-1-INPUT -p 51 -j ACCEPT
-A RH-Firewall-1-INPUT -p udp --dport 5353 -d 224.0.0.251 -j ACCEPT
-A RH-Firewall-1-INPUT -p udp -m udp --dport 631 -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
-A RH-F...NPUT -m state --state NEW -m tcp -p tcp --dport 22 -j ACCEPT
-A RH-F...NPUT -m state --state NEW -m tcp -p tcp --dport 80 -j ACCEPT
-A RH-F...NPUT -m state --state NEW -m tcp -p tcp --dport 21 -j ACCEPT
-A RH-F...NPUT -m state --state NEW -m tcp -p tcp --dport 25 -j ACCEPT
-A RH-Firewall-1-INPUT -j REJECT --reject-with icmp-host-prohibited
COMMIT
root@RHELv4u4:~#

Для запуска службы межсетевого экрана следует выполнить команду service iptables start. С помощью утилиты chkconfig вы можете изменить настройки системы для запуска межсетевого экрана iptables в процессе загрузки системы.

root@RHELv4u4:~# service iptables start
Applying iptables firewall rules: [OK]
root@RHELv4u4:~# chkconfig iptables on
root@RHELv4u4:~#

Одной из наиболее полезных возможностей межсетевого экрана iptables является возможность вывода подробной информации о состоянии межсетевого экрана в случае запроса этой информации с помощью команды service iptables status.

root@RHELv4u4:~# service iptables status
Table: filter
Chain INPUT (policy ACCEPT)
target prot opt source destination
RH-Firewall-1-INPUT all — 0.0.0.0/0 0.0.0.0/0
Chain FORWARD (policy ACCEPT)
target prot opt source destination
RH-Firewall-1-INPUT all — 0.0.0.0/0 0.0.0.0/0
Chain OUTPUT (policy ACCEPT)
target prot opt source destination
Chain RH-Firewall-1-INPUT (2 references)
target prot opt source destination
ACCEPT all — 0.0.0.0/0 0.0.0.0/0
ACCEPT icmp — 0.0.0.0/0 0.0.0.0/0 icmp type 255
ACCEPT esp — 0.0.0.0/0 0.0.0.0/0
ACCEPT ah — 0.0.0.0/0 0.0.0.0/0
ACCEPT udp — 0.0.0.0/0 224.0.0.251 udp dpt:5353
ACCEPT udp — 0.0.0.0/0 0.0.0.0/0 udp dpt:631
ACCEPT all — 0.0.0.0/0 0.0.0.0/0 state RELATED,ESTABLISHED
ACCEPT tcp — 0.0.0.0/0 0.0.0.0/0 state NEW tcp dpt:22
ACCEPT tcp — 0.0.0.0/0 0.0.0.0/0 state NEW tcp dpt:80
ACCEPT tcp — 0.0.0.0/0 0.0.0.0/0 state NEW tcp dpt:21
ACCEPT tcp — 0.0.0.0/0 0.0.0.0/0 state NEW tcp dpt:25
REJECT all — 0.0.0.0/0 0.0.0.0/0 reject-with icmp-host-prohibited
root@RHELv4u4:~#

Для создания с нуля правил межсетевого экрана необходимо понимать принципы функционирования стека протоколов TCP/IP. Отличные руководства по написанию правил межсетевого экрана iptables могут быть найдены в сети по ссылкам: http://iptables-tutorial.frozentux.net/iptables-tutorial.html и http://tldp.org/HOWTO/IP-Masquerade-HOWTO/.

27.2. Практическое задание: межсетевой экран iptables

1. Проверьте, выполняется ли служба межсетевого экрана в вашей системе.
2. Остановите службу межсетевого экрана.

27.3. Корректная процедура выполнения практического задания: межсетевой экран iptables

1. Проверьте, выполняется ли служба межсетевого экрана в вашей системе.

root@rhel55 ~# service iptables status | head
Table: filter
Chain INPUT (policy ACCEPT)
num target prot opt source destination
1 RH-Firewall-1-INPUT all — 0.0.0.0/0 0.0.0.0/0
Chain FORWARD (policy ACCEPT)
num target prot opt source destination
1 RH-Firewall-1-INPUT all -- 0.0.0.0/0 0.0.0.0/0
Chain OUTPUT (policy ACCEPT)

1. Остановите службу межсетевого экрана.

root@rhel55 ~# service iptables stop
Flushing firewall rules: [OK]
Setting chains to policy ACCEPT: filter [OK]
Unloading iptables modules: [OK]
root@rhel55 ~# service iptables status
Firewall is stopped.

27.4. Супердемоны xinetd и inetd

27.4.1. Информация о супердемонах

В прошлом, когда объем оперативной памяти компьютеров был строго ограничен, был разработан суперсервер, предназначенный для прослушивания всех сокетов с заданными номерами портов и запуска соответствующего демона при необходимости. Такие службы, как swat, telnet и ftp обычно обслуживаются подобным суперсервером. Супердемон xinetd является более современной реализацией супердемона inetd. Мы будем обсуждать вопросы настройки обоих упомянутых супердемонов.

Даже в не очень современных дистрибутивах, таких, как RHEL5 и Ubuntu 10.04, супердемоны inetd или xinetd не активируются по умолчанию до того момента, когда какое-либо из приложений явным образом потребует их активации.

27.4.2. Какой из супердемонов используется: inetd или xinetd?

В первую очередь следует установить, используется ли в вашей системе супердемон inetd или супердемон xinetd. В данном дистрибутиве Debian 4.0 Etch используется супердемон inetd.

root@barry:~# ps fax | grep inet
3870 ? Ss 0:00 /usr/sbin/inetd

А в данном дистрибутиве Red Hat Enterprise Linux 4 с обновлением 4 используется супердемон xinetd.

[root@RHEL4b ~]# ps fax | grep inet
3003 ? Ss 0:00 xinetd -stayalive -pidfile /var/run/xinetd.pid

Оба демона имеют идентичный набор функций (прослушивание множества портов, запуск сторонних демонов при необходимости), но они используют отличные файлы конфигурации.

27.4.3. Супердемон xinetd

Демон xinetd обычно называется супердемоном, так как он осуществляет прием множества входящих соединений и запускает сторонние демоны по мере необходимости. При приеме запроса на соединение демон xinetd будет в первую очередь проверять файлы со списками доступа к службам по протоколу TCP (TCP wrappers) (/etc/hosts.allow и /etc/hosts.deny), после чего передаст управление соединением стороннему демону. Данный супердемон настраивается с помощью основного файла конфигурации /etc/xinetd.conf, а также специальных файлов конфигурации из директории /etc/xinetd.d. Давайте рассмотрим содержимое основного файла конфигурации /etc/xinetd.conf.

paul@RHELv4u2:~$ cat /etc/xinetd.conf
#
Simple configuration file for xinetd
#
Some defaults, and include /etc/xinetd.d/
defaults
{
instances = 60
log_type = SYSLOG authpriv
log_on_success = HOST PID
log_on_failure = HOST
cps = 25 30
}
includedir /etc/xinetd.d
paul@RHELv4u2:~$

В соответствии с значениями параметров из данного файла конфигурации, демон xinetd может обрабатывать 60 клиентских запросов одновременно. Он использует вспомогательный программный компонент authpriv для журналирования IP-адреса узла и идентификатора процесса успешно запущенного демона. Если служба (или протокол, связанный с демоном) получает более 25 запросов в секунду, он приостанавливает передачу последующих запросов на 30 секунд.

Директория /etc/xinetd.d содержит более специфичные конфигурационные файлы. Давайте также рассмотрим один из них.

```
paul@RHELv4u2:~$ ls /etc/xinetd.d
amanda chargen-udp echo klogin rexec talk
amandaidx cups-lpd echo-udp krb5-telnet rlogin telnet
amidxtape daytime eklogin kshell rsh tftp
auth daytime-udp finger ktalk rsync time
chargen dbskkd-cdb gssftp ntalk swat time-udp
paul@RHELv4u2:~$ cat /etc/xinetd.d/swat
# default: off
# description: SWAT is the Samba Web Admin Tool. Use swat \
# to configure your Samba server. To use SWAT, \
# connect to port 901 with your favorite web browser.
service swat
{
port = 901
socket_type = stream
wait = no
only_from = 127.0.0.1
user = root
server = /usr/sbin/swat
log_on_failure += USERID
disable = yes
}
paul@RHELv4u2:~$
```

Параметры служб должны быть приведены в файле /etc/services. Параметр port соответствует порту службы, причем его значение должно совпадать с соответствующим значение порта службы из файла /etc/services. Параметр socket_type должен иметь значение stream в случае служб, работающих с протоколом TCP (и значение dgram в случае служб, работающих с протоколом UDP). Параметр log_on_failure += позволяет добавлять идентификатор пользователя в запись журнала, формат которой описывается в файле конфигурации /etc/xinetd.conf. Значением последнего параметра disable может быть строка yes или no. Установка значения no приведет к активации службы!

Обратитесь к страницам руководств man xinetd и man xinetd.conf для ознакомления с другими параметрами конфигурации рассматриваемого супердемона.

27.4.4. Супердемон inetd

Данный супердемон использует всего один конфигурационный файл /etc/inetd.conf. Каждый протокол или демон, запросы к которому принимаются, описывается с помощью одной строки в данном файле.

```
root@barry:~# grep ftp /etc/inetd.conf
tftp dgram udp wait nobody /usr/sbin/tcpd /usr/sbin/in.tftpd /boot/tftp
root@barry:~#
```

Вы можете деактивировать службу в рамках описанного выше файла конфигурации inetd.conf, добавив символ # в начале соответствующей строки. Ниже приведен пример деактивированного веб-интерфейса vmware (в данном случае принимались соединения на порту 902 по протоколу TCP).

```
paul@laika:~$ grep vmware /etc/inetd.conf
#902 stream tcp nowait root /usr/sbin/vmware-authd vmware-authd
```

27.5. Практическое задание: супердемоны inetd и xinetd

1. Проверьте все находящиеся в вашем распоряжении системы и установите, используются ли в этих системах супердемон xinetd или супердемон inetd.
2. Изучите файлы конфигурации супердемонов.
3. (В том случае, если у вас имеется возможность установки пакета программного обеспечения службы telnet, замените все упоминания службы swat на telnet в следующих заданиях.) Установлены ли файлы, относящиеся к службе swat, в вашей системе? Если нет установите соответствующие пакеты программного обеспечения и обратите внимание на изменения, внесенные в файлы конфигурации супердемона (x)inetd. Активирована ли по умолчанию служба swat?
4. Деактивируйте службу swat и попытайтесь проверить ее работоспособность, установив соединение с соответствующим портом. Снова активируйте службу swat и попытайтесь проверить ее работоспособность, установив соединение с соответствующим портом.

Глава 28. Ядро Linux

28.1. О ядре Linux

28.1.1. Версии ядра

В 1991 году Linus Torvalds разработал первую версию ядра Linux. Он выложил исходный код ядра в открытый доступ, после чего другие люди начали вносить свой вклад в его развитие. В разработке одной из версий ядра (а именно, версии 2.6.27, выпущенной в ноябре 2008 года) поучаствовало более 4000 разработчиков.

Для основных версий ядра Linux используются четные и нечетные номера. Ранее существовало разделение версий, в соответствии с которым версии 2.0, 2.2, 2.4 и 2.6 считались

стабильными. В то же время версии 2.1, 2.3 и 2.5 считались нестабильными (или разрабатываемыми). Начиная с выпуска версии ядра 2.6.0 в январе 2004 года вся разработка велась в рамках ветки 2.6. Версия ядра 2.7.0 так и не была выпущена и, судя по заявлениям главного разработчика ядра, которым все также является Linus Torvalds, схема разделения версий ядра на четные/стабильные и нечетные/разрабатываемые не используется и не будет использоваться в будущем.

28.1.2. Команда uname -r

Для того, чтобы узнать текущую версию ядра Linux, следует использовать команду uname -r таким образом, как показано ниже.

В первом примере показано ядро Linux основной версии 2.6 и дополнительной версии 24. Остальная часть строки -22-generic является специфичной для дистрибутива (в данном случае используется дистрибутив Ubuntu).

```
paul@laika:~$ uname -r
2.6.24-22-generic
```

При выполнении этой же команды в дистрибутиве Red Hat Entrprise Linux можно получить информацию о более старом ядре (версии 2.6.18), причем строка -92.1.17.el5 также является специфичной для дистрибутива.

```
[paul@RHEL52 ~]$ uname -r
2.6.18-92.1.17.el5
```

28.1.3. Файл /proc/cmdline

Параметры, которые были переданы ядру ОС в процессе загрузки системы, могут быть получены из файла /proc/cmdline.

```
paul@RHELv4u4:~$ cat /proc/cmdline
ro root=/dev/VolGroup00/LogVol00 rhgb quiet
```

28.1.4. Однопользовательский режим

В случае передачи ядру Linux параметра single будет осуществляться загрузка системы в однопользовательском режиме. В данном случае после загрузки ядра Linux будет запущена командная оболочка bash с привилегиями пользователя root (ввода пароля не потребуется).

В некоторых дистрибутивах данная возможность заблокирована (на этапе компиляции ядра ОС).

28.1.5. Параметр init=/bin/bash

Обычно ядро ОС запускает бинарный файл, путь к которому передан с помощью параметра init, создавая таким образом первый процесс-демон. Добавление параметра init=/bin/bash к строке параметров ядра позволяет вызывать командную оболочку bash (снова с правами пользователя root без необходимости ввода пароля).

28.1.6. Файл журнала /var/log/messages

Ядро ОС в процессе загрузки передает демону syslogd информацию о множестве выполняемых действий, которая сохраняется в файле журнала /var/log/messages. Из данного файла журнала можно извлечь информацию о процессе загрузки ядра, включая информацию обо всех устройствах, которые были идентифицированы в процессе загрузки системы.

```
[root@RHEL53 ~]# grep -A16 "syslogd 1.4.1:" /var/log/messages|cut -b24-
syslogd 1.4.1: restart.
kernel: klogd 1.4.1, log source = /proc/kmsg started.
kernel: Linux version 2.6.18-128.el5 (mockbuild@hs20-bc1-5.build.red...
kernel: BIOS-provided physical RAM map:
kernel: BIOS-e820: 0000000000000000 - 000000000009f800 (usable)
kernel: BIOS-e820: 000000000009f800 - 00000000000a0000 (reserved)
kernel: BIOS-e820: 00000000000ca000 - 00000000000cc000 (reserved)
kernel: BIOS-e820: 00000000000dc000 - 0000000000100000 (reserved)
kernel: BIOS-e820: 0000000000100000 - 000000001fef0000 (usable)
kernel: BIOS-e820: 000000001fef0000 - 000000001feff000 (ACPI data)
kernel: BIOS-e820: 000000001feff000 - 000000001ff00000 (ACPI NVS)
kernel: BIOS-e820: 000000001ff00000 - 0000000020000000 (usable)
kernel: BIOS-e820: 00000000fec00000 - 00000000fec10000 (reserved)
kernel: BIOS-e820: 00000000fee00000 - 00000000fee01000 (reserved)
kernel: BIOS-e820: 00000000fffe0000 - 0000000100000000 (reserved)
kernel: 0MB HIGHMEM available.
kernel: 512MB LOWMEM available.
```

А в данном примере показан фрагмент файла /var/log/messages, который может использоваться для извлечения информации о дисковом устройстве, представленном файлом устройства /dev/sda.

```
[root@RHEL53 ~]# grep sda /var/log/messages | cut -b24-
kernel: SCSI device sda: 41943040 512-byte hdwr sectors (21475 MB)
kernel: sda: Write Protect is off
kernel: sda: cache data unavailable
kernel: sda: assuming drive cache: write through
```

kernel: SCSI device sda: 41943040 512-byte hdwr sectors (21475 MB)
kernel: sda: Write Protect is off
kernel: sda: cache data unavailable
kernel: sda: assuming drive cache: write through
kernel: sda: sda1 sda2
kernel: sd 0:0:0:0: Attached scsi disk sda
kernel: EXT3 FS on sda1, internal journal

28.1.7. Утилита dmesg

Утилита dmesg выводит все сообщения ядра ОС (с момента его последней загрузки системы).

[root@RHEL53 ~]# dmesg | head
Linux version 2.6.18-128.el5 (mockbuild@hs20-bc1-5.build.redhat.com)
BIOS-provided physical RAM map:
BIOS-e820: 0000000000000000 - 000000000009f800 (usable)
BIOS-e820: 000000000009f800 - 00000000000a0000 (reserved)
BIOS-e820: 00000000000ca000 - 00000000000cc000 (reserved)
BIOS-e820: 00000000000dc000 - 0000000000100000 (reserved)
BIOS-e820: 0000000000100000 - 000000001fef0000 (usable)
BIOS-e820: 000000001fef0000 - 000000001feff000 (ACPI data)
BIOS-e820: 000000001feff000 - 000000001ff00000 (ACPI NVS)
BIOS-e820: 000000001ff00000 - 0000000020000000 (usable)

Следовательно, в случае поиска информации о дисковом устройстве, представленном файлом устройства /dev/sda, с помощью утилиты dmesg, будут выведены только сообщения ядра ОС с момента последней загрузки системы.

[root@RHEL53 ~]# dmesg | grep sda
SCSI device sda: 41943040 512-byte hdwr sectors (21475 MB)
sda: Write Protect is off
sda: Mode Sense: 5d 00 00 00
sda: cache data unavailable
sda: assuming drive cache: write through
SCSI device sda: 41943040 512-byte hdwr sectors (21475 MB)
sda: Write Protect is off
sda: Mode Sense: 5d 00 00 00
sda: cache data unavailable
sda: assuming drive cache: write through
sda: sda1 sda2
sd 0:0:0:0: Attached scsi disk sda
EXT3 FS on sda1, internal journal

28.2. Исходный код ядра Linux

28.2.1. Ресурс ftp.kernel.org

Официальное хранилище архивов исходного ядра Linux расположено по адресу ftp.kernel.org. Данный ресурс содержит архивы исходного кода всех официальных выпусков ядра Linux начиная с 1991 года. Доступ ко всем архивам исходного кода, а также к файлам со списками изменений и файлам с исправлениями может осуществляться по протоколам http, ftp и rsync. Дополнительная информация может быть получена на вебсайте www.kernel.org.

Каждый человек может воспользоваться ftp-клиентом для осуществления анонимного входа на ресурс ftp.kernel.org.

paul@laika:~$ ftp ftp.kernel.org
Connected to pub3.kernel.org.
220 Welcome to ftp.kernel.org.
Name (ftp.kernel.org:paul): anonymous
331 Please specify the password.
Password:
230- Welcome to the
230-
230- LINUX KERNEL ARCHIVES
230- ftp.kernel.org

Архивы исходного кода всех версий ядра Linux расположены по пути pub/linux/kernel/.

ftp> ls pub/linux/kernel/v*
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
drwxrwsr-x 2 536 536 4096 Mar 20 2003 v1.0
drwxrwsr-x 2 536 536 20480 Mar 20 2003 v1.1
drwxrwsr-x 2 536 536 8192 Mar 20 2003 v1.2
drwxrwsr-x 2 536 536 40960 Mar 20 2003 v1.3
drwxrwsr-x 3 536 536 16384 Feb 08 2004 v2.0
drwxrwsr-x 2 536 536 53248 Mar 20 2003 v2.1
drwxrwsr-x 3 536 536 12288 Mar 24 2004 v2.2
drwxrwsr-x 2 536 536 24576 Mar 20 2003 v2.3
drwxrwsr-x 5 536 536 28672 Dec 02 08:14 v2.4
drwxrwsr-x 4 536 536 32768 Jul 14 2003 v2.5
drwxrwsr-x 7 536 536 110592 Dec 05 22:36 v2.6
226 Directory send OK.
ftp>

28.2.2. Директория /usr/src

Исходный код ядра ОС на вашем локальном компьютере должен быть расположен в директории /usr/src. Учтите, что структура поддиректорий директории /usr/src может отличаться в зависимости от используемого вами дистрибутива.

В первую очередь давайте рассмотрим структуру поддиректорий директории /usr/src в дистрибутиве Debian. В данном случае в этой

директории можно обнаружить две версии полного исходного кода ядра Linux. При поиске определенного файла (e1000_main.c) исходного кода с помощью утилиты find можно получить полный путь к этому файлу.

paul@barry:~$ ls -l /usr/src/
drwxr-xr-x 20 root root 4096 2006-04-04 22:12 linux-source-2.6.15
drwxr-xr-x 19 root root 4096 2006-07-15 17:32 linux-source-2.6.16
paul@barry:~$ find /usr/src -name e1000_main.c
/usr/src/linux-source-2.6.15/drivers/net/e1000/e1000_main.c
/usr/src/linux-source-2.6.16/drivers/net/e1000/e1000_main.c

Данная структура поддиректорий очень похожа на структуру поддиректорий директории /usr/src в дистрибутиве Ubuntu за тем исключением, что в данном случае в директории расположен исходный код одной версии ядра (и эта версия более новая).

paul@laika:~$ ls -l /usr/src/
drwxr-xr-x 23 root root 4096 2008-11-24 23:28 linux-source-2.6.24
paul@laika:~$ find /usr/src -name "e1000_main.c"
/usr/src/linux-source-2.6.24/drivers/net/e1000/e1000_main.c

Теперь давайте рассмотрим содержимое директории /usr/src в дистрибутиве Red Hat Entrprise Linux.

[paul@RHEL52 ~]$ ls -l /usr/src/
drwxr-xr-x 5 root root 4096 Dec 5 19:23 kernels
drwxr-xr-x 7 root root 4096 Oct 11 13:22 redhat

Нам придется погрузиться немного глубже в дерево директорий для того, чтобы добраться до исходного кода ядра ОС в дистрибутиве Red Hat!

[paul@RHEL52 ~]$ cd /usr/src/redhat/BUILD/
[paul@RHEL52 BUILD]$ find . -name "e1000_main.c"
/kernel-2.6.18/linux-2.6.18.i686/drivers/net/e1000/e1000_main.c

28.2.3. Загрузка исходного кода ядра ОС

Debian

Установка пакета программного обеспечения с исходным кодом ядра ОС в дистрибутиве Debian осуществляется достаточно просто с помощью команды aptitude install linux-source. В первую очередь вы можете осуществить поиск всех пакетов программного обеспечения с именами linux-source таким образом, как показано в примере ниже.

root@barry:~# aptitude search linux-source
v linux-source -
v linux-source-2.6 -
id linux-source-2.6.15 - Linux kernel source for version 2.6.15
i linux-source-2.6.16 - Linux kernel source for version 2.6.16
p linux-source-2.6.18 - Linux kernel source for version 2.6.18
p linux-source-2.6.24 - Linux kernel source for version 2.6.24

После этого вы можете использовать команду aptitude install для загрузки и установки пакета программного обеспечения с исходным кодом ядра Linux из состава дистрибутива Debian.

root@barry:~# aptitude install linux-source-2.6.24

После того, как утилита aptitude закончит работу, вы обнаружите новый файл с именем /usr/src/linux-source-<версия>.tar.bz2 в файловой системе вашего компьютера.

root@barry:/usr/src# ls -lh
drwxr-xr-x 20 root root 4.0K 2006-04-04 22:12 linux-source-2.6.15
drwxr-xr-x 19 root root 4.0K 2006-07-15 17:32 linux-source-2.6.16
-rw-r—r-- 1 root root 45M 2008-12-02 10:56 linux-source-2.6.24.tar.bz2

Ubuntu

Дистрибутив Ubuntu основывается на дистрибутиве Debian, причем в нем также может использоваться утилита aptitude, поэтому задача полностью аналогична предыдущей.

root@laika:~# aptitude search linux-source
i linux-source - Linux kernel source with Ubuntu patches
v linux-source-2.6 -
i A linux-source-2.6.24 - Linux kernel source for version 2.6.24

root@laika:~# aptitude install linux-source

И после того, как утилита aptitude завершит свою работу, мы получим файл /usr/src/linux-source-<версия>.tar.bz2.

root@laika:~# ll /usr/src
итого 45M
-rw-r—r-- 1 root root 45M 2008-11-24 23:30 linux-source-2.6.24.tar.bz2

Red Hat Enterprise Linux

Ранее пакет программного обеспечения с исходным кодом ядра ОС располагался на четвертом установочном диске с пакетами исходного кода дистрибутива Red Hat Enerprise Linux. Файл пакета программного обеспечения носит имя kernel-2.6.9-42.EL.src.rpm (пример для дистрибутива RHEL версии 4 с обновлением 4). Также данный пакет доступен по адресу ftp://ftp.redhat.com/pub/redhat/linux/enterprise/5Server/en/os/SRPMS (пример для дистрибутива RHEL 5).

Для загрузки пакета исходного кода ядра ОС дистрибутива RHEL следует использовать длинную команду (команда должна располагаться в одной строке без завершающего символа \).

wget ftp://ftp.redhat.com/pub/redhat/linux/enterprise/5Server/en/os/\
SRPMS/kernel-uname -r.src.rpm

После того, как утилита wget закончит свою работу, вы получите файл с расширением .rpm размером в 60 Мб.

[root@RHEL52 src]# ll
итого 60M

-rw-r—r-- 1 root root 60M дек 5 20:54 kernel-2.6.18-92.1.17.el5.src.rpm
drwxr-xr-x 5 root root 4.0K дек 5 19:23 kernels
drwxr-xr-x 7 root root 4.0K окт 11 13:22 redhat

Нам придется выполнить дополнительные действия для того, чтобы использовать исходный код ядра ОС из пакета программного обеспечения по назначению.

В первую очередь мы должны выполнить команду rpm -i kernel-2.6.9-42.EL.src.rpm для установки загруженного пакета программного обеспечения дистрибутива Red Hat.

[root@RHEL52 src]# ll
итого 60M
-rw-r—r-- 1 root root 60M дек 5 20:54 kernel-2.6.18-92.1.17.el5.src.rpm
drwxr-xr-x 5 root root 4.0K дек 5 19:23 kernels
drwxr-xr-x 7 root root 4.0K окт 11 13:22 redhat
[root@RHEL52 src]# rpm -i kernel-2.6.18-92.1.17.el5.src.rpm

После этого мы должны перейти в директорию SPECS и задействовать утилиту rpmbuild.

[root@RHEL52 ~]# cd /usr/src/redhat/SPECS
[root@RHEL52 SPECS]# rpmbuild -bp -vv --target=i686 kernel-2.6.spec

Утилита rpmbuild поместит исходный код ядра Linux дистрибутива RHEL в директорию /usr/src/redhat/BUILD/kernel-<версия>/.

[root@RHEL52 kernel-2.6.18]# pwd
/usr/src/redhat/BUILD/kernel-2.6.18
[root@RHEL52 kernel-2.6.18]# ll
итого 20K
drwxr-xr-x 2 root root 4.0K дек 6 2007 config
-rw-r—r-- 1 root root 3.1K дек 5 20:58 Config.mk
drwxr-xr-x 20 root root 4.0K дек 5 20:58 linux-2.6.18.i686
drwxr-xr-x 19 root root 4.0K сен 20 2006 vanilla
drwxr-xr-x 8 root root 4.0K дек 6 2007 xen

28.3. Файлы, используемые в процессе загрузки ядра ОС

28.3.1. Файл vmlinuz

Файл vmlinuz из директории /boot является сжатым исполняемым файлом ядра ОС.

paul@barry:~$ ls -lh /boot | grep vmlinuz
-rw-r—r-- 1 root root 1.2M 2006-03-06 16:22 vmlinuz-2.6.15-1-486
-rw-r—r-- 1 root root 1.1M 2006-03-06 16:30 vmlinuz-2.6.15-1-686
-rw-r—r-- 1 root root 1.3M 2008-02-11 00:00 vmlinuz-2.6.18-6-686
paul@barry:~$

28.3.2. Файл initrd

Ядро ОС использует файл initrd (Initial RAM disk — диск в оперативной памяти для начальной инициализации) в процессе загрузки системы. Данный диск монтируется до загрузки ядра ОС и может содержать дополнительные драйверы и модули ядра ОС. На самом деле, данный файл является сжатым архивом формата CPIO, поэтому вы можете просмотреть его содержимое следующим образом.

root@RHELv4u4:/boot# mkdir /mnt/initrd
root@RHELv4u4:/boot# cp initrd-2.6.9-42.0.3.EL.img TMPinitrd.gz
root@RHELv4u4:/boot# gunzip TMPinitrd.gz
root@RHELv4u4:/boot# file TMPinitrd
TMPinitrd: ASCII cpio archive (SVR4 with no CRC)
root@RHELv4u4:/boot# cd /mnt/initrd/
root@RHELv4u4:/mnt/initrd# cpio -i | /boot/TMPinitrd
4985 блоков
root@RHELv4u4:/mnt/initrd# ls -l
итого 76
drwxr-xr-x 2 root root 4096 фев 5 08:36 bin
drwxr-xr-x 2 root root 4096 фев 5 08:36 dev
drwxr-xr-x 4 root root 4096 фев 5 08:36 etc
-rwxr-xr-x 1 root root 1607 фев 5 08:36 init
drwxr-xr-x 2 root root 4096 фев 5 08:36 lib
drwxr-xr-x 2 root root 4096 фев 5 08:36 loopfs
drwxr-xr-x 2 root root 4096 фев 5 08:36 proc
lrwxrwxrwx 1 root root 3 фев 5 08:36 sbin → bin
drwxr-xr-x 2 root root 4096 фев 5 08:36 sys
drwxr-xr-x 2 root root 4096 фев 5 08:36 sysroot
root@RHELv4u4:/mnt/initrd#

28.3.3. Файл System.map

Файл System.map содержит таблицу символов и изменяется при каждой компиляции ядра ОС. Таблица символов также присутствует в файле /proc/kallsyms (до выпуска ядра Linux версии 2.6 данный файл носил имя /proc/ksyms).

root@RHELv4u4:/boot# head System.map-uname -r
00000400 A __kernel_vsyscall
0000041a A SYSENTER_RETURN_OFFSET
00000420 A __kernel_sigreturn
00000440 A __kernel_rt_sigreturn
c0100000 A _text
c0100000 T startup_32
c01000c6 t checkCPUtype
c0100147 t is486
c010014e t is386

```
c010019f t L6
root@RHELv4u4:/boot# head /proc/kallsyms
c0100228 t _stext
c0100228 t calibrate_delay_direct
c0100228 t stext
c0100337 t calibrate_delay
c01004db t rest_init
c0100580 t do_pre_smp_initcalls
c0100585 t run_init_process
c01005ac t init
c0100789 t early_param_test
c01007ad t early_setup_test
root@RHELv4u4:/boot#
```

28.3.4. Файл .config

Последним файлом, копируемым в директорию /boot, является файл со значениями параметров конфигурации ядра ОС, используемый в процессе его компиляции. Данный файл не должен находиться директории /boot для корректного функционирования ядра ОС, но размещение его копии в данной директории является обычной практикой. Благодаря его наличию появляется возможность осуществления повторной компиляции ядра ОС с использованием тех же значений параметров конфигурации, которые были использованы для компиляции существующего работоспособного ядра ОС.

28.4. Модули ядра Linux

28.4.1. Информация о модулях ядра Linux

Ядро Linux является монолитным ядром с поддержкой загружаемых модулей. Данные модули содержат части ядра ОС, обычно используемые в качестве драйверов устройств, реализаций файловых систем и сетевых протоколов. В большинстве случаев необходимые модули ядра ОС загружаются автоматически и динамически без вмешательства системного администратора.

28.4.2. Директория /lib/modules

Модули ядра ОС хранятся в директории /lib/modules/<версия-ядра-ос>. Для хранения модулей каждой версии ядра Linux, которая была скомпилирована для вашей системы, создается отдельная директория.

```
paul@laika:~$ ll /lib/modules/
итого 12K
drwxr-xr-x 7 root root 4.0K 2008-11-10 14:32 2.6.24-16-generic
drwxr-xr-x 8 root root 4.0K 2008-12-06 15:39 2.6.24-21-generic
drwxr-xr-x 8 root root 4.0K 2008-12-05 12:58 2.6.24-22-generic
```

28.4.3. Файл <модуль>.ko

Имя файла, содержащего код модуля ядра ОС, обычно оканчивается на .ko. В данном примере показано расположение файлов модулей ядра ОС, предназначенных для поддержки протокола isdn.

```
paul@laika:~$ find /lib/modules -name isdn.ko
/lib/modules/2.6.24-21-generic/kernel/drivers/isdn/i4l/isdn.ko
/lib/modules/2.6.24-22-generic/kernel/drivers/isdn/i4l/isdn.ko
/lib/modules/2.6.24-16-generic/kernel/drivers/isdn/i4l/isdn.ko
```

28.4.4. Утилита lsmod

Для ознакомления со списком загруженных в текущий момент модулей ядра ОС, следует использовать утилиту lsmod. В выводе утилиты вы можете обнаружить информацию об имени каждого из загруженных модулей, его размере, значении счетчика использования модуля, а также именах других модулей, использующих данный модуль.

```
[root@RHEL52 ~]# lsmod | head -5
Module Size Used by
autofs4 24517 2
hidp 23105 2
rfcomm 42457 0
l2cap 29505 10 hidp,rfcomm
```

28.4.5. Файл /proc/modules

Файл /proc/modules содержит список всех модулей, загруженных ядром ОС. Список модулей ядра ОС из данного файла является слишком длинным для того, чтобы приводить его в данной книге, поэтому давайте используем утилиту grep для поиска модуля с именем vm.

Мы видим, что одновременно загружены модуля ядра ОС vmmon и vmnet. Вы можете получить ту же самую информацию и с помощью утилиты lsmod. На самом деле, утилита lsmod всего лишь читает содержимое файла /proc/modules и выводит его с соответствующим форматированием.

```
paul@laika:~$ cat /proc/modules | grep vm
vmnet 36896 13 - Live 0xffffffff88b21000 (P)
vmmon 194540 0 - Live 0xffffffff88af0000 (P)
paul@laika:~$ lsmod | grep vm
vmnet 36896 13
vmmon 194540 0
paul@laika:~$
```

28.4.6. Зависимости модулей ядра ОС

Некоторые модули ядра ОС зависят от других модулей ядра ОС. В следующем примере вы можете увидеть, что модуль ядра ОС с именем nfsd зависит от модулей с именами exportfs, lockd и sunrpc.

paul@laika:~$ cat /proc/modules | grep nfsd
nfsd 267432 17 - Live 0xffffffff88a40000
exportfs 7808 1 nfsd, Live 0xffffffff88a3d000
lockd 73520 3 nfs,nfsd, Live 0xffffffff88a2a000
sunrpc 185032 12 nfs,nfsd,lockd, Live 0xffffffff889fb000
paul@laika:~$ lsmod | grep nfsd
nfsd 267432 17
exportfs 7808 1 nfsd
lockd 73520 3 nfs,nfsd
sunrpc 185032 12 nfs,nfsd,lockd
paul@laika:~$

28.4.7. Утилита insmod

Модули ядра ОС могут быть загружены в ручном режиме с помощью утилиты insmod. Это очень простой (и устаревший) способ загрузки модулей ядра ОС. В примере ниже показан процесс загрузки модуля ядра ОС с именем fat (реализующего поддержку файловой системы fat).

root@barry:/lib/modules/2.6.17-2-686# lsmod | grep fat
root@barry:/lib/modules/2.6.17-2-686# insmod kernel/fs/fat/fat.ko
root@barry:/lib/modules/2.6.17-2-686# lsmod | grep fat
fat 46588 0

Утилита insmod не определяет зависимости между модулями ядра ОС, поэтому в следующем примере нам не удалось загрузить с помощью данной утилиты модуль ядра ОС с именем isdn (так как модуль ядра ОС с именем isdn зависит от модуля ядра ОС с именем slhc).

[root@RHEL52 drivers]# pwd
/lib/modules/2.6.18-92.1.18.el5/kernel/drivers
[root@RHEL52 kernel]# insmod isdn/i4l/isdn.ko
insmod: error inserting isdn/i4l/isdn.ko: -1 Unknown symbol in module

28.4.8. Утилита modinfo

Как вы можете увидеть в представленном ниже выводе утилиты modinfo, модуль ядра ОС с именем isdn зависит от модуля ядра ОС с именем slhc.

[root@RHEL52 drivers]# modinfo isdn/i4l/isdn.ko | head -6
filename: isdn/i4l/isdn.ko
license: GPL
author: Fritz Elfert
description: ISDN4Linux: link layer
srcversion: 99650346E708173496F6739
depends: slhc

28.4.9. Утилита modprobe

Большим преимуществом утилиты modprobe перед утилитой insmod является возможность первой утилиты загружать все необходимые модули ядра ОС, в то время, как вторая утилита требует загрузки всех необходимых модулей ядра ОС в ручном режиме. Другое преимущество данной утилиты заключается в отсутствии необходимости указания полного пути к файлу загружаемого модуля ядра ОС.

В данном примере показан способ загрузки модуля ядра ОС с именем isdn с помощью утилиты modprobe с автоматической загрузкой модуля ядра ОС с именем shlc в фоновом режиме.

[root@RHEL52 kernel]# lsmod | grep isdn
[root@RHEL52 kernel]# modprobe isdn
[root@RHEL52 kernel]# lsmod | grep isdn
isdn 122433 0
slhc 10561 1 isdn
[root@RHEL52 kernel]#

28.4.10. Файл /lib/modules/<версия-ядра-ос>/modules.dep

Зависимости модулей ядра ОС описаны в рамках файла с именем modules.dep.

[root@RHEL52 2.6.18-92.1.18.el5]# pwd
/lib/modules/2.6.18-92.1.18.el5
[root@RHEL52 2.6.18-92.1.18.el5]# head -3 modules.dep
/lib/modules/2.6.18-92.1.18.el5/kernel/drivers/net/tokenring/3c359.ko:
/lib/modules/2.6.18-92.1.18.el5/kernel/drivers/net/pcmcia/3c574_cs.ko:
/lib/modules/2.6.18-92.1.18.el5/kernel/drivers/net/pcmcia/3c589_cs.ko:

28.4.11. Утилита depmod

Файл modules.dep может быть обновлен (повторно создан) с помощью утилиты depmod. В следующем примере не было добавлено никаких новых модулей ядра ОС, поэтому утилита depmod сгенерировала идентичный файл.

root@barry:/lib/modules/2.6.17-2-686# ls -l modules.dep
-rw-r—r-- 1 root root 310676 2008-03-01 16:32 modules.dep
root@barry:/lib/modules/2.6.17-2-686# depmod
root@barry:/lib/modules/2.6.17-2-686# ls -l modules.dep
-rw-r—r-- 1 root root 310676 2008-12-07 13:54 modules.dep

28.4.12. Утилита rmmod

По аналогии с утилитой insmod, утилита rmmod на сегодняшний день практически не используется.

[root@RHELv4u3 ~]# modprobe isdn
[root@RHELv4u3 ~]# rmmod slhc
ERROR: Module slhc is in use by isdn
[root@RHELv4u3 ~]# rmmod isdn

[root@RHELv4u3 ~]# rmmod slhc
[root@RHELv4u3 ~]# lsmod | grep isdn
[root@RHELv4u3 ~]#

28.4.13. Команда modprobe -r

В отличие от утилиты rmmod, утилита modprobe будет автоматически удалять неиспользуемые модуля ядра ОС.

[root@RHELv4u3 ~]# modprobe isdn
[root@RHELv4u3 ~]# lsmod | grep isdn
isdn 133537 0
slhc 7233 1 isdn
[root@RHELv4u3 ~]# modprobe -r isdn
[root@RHELv4u3 ~]# lsmod | grep isdn
[root@RHELv4u3 ~]# lsmod | grep slhc
[root@RHELv4u3 ~]#

28.4.14. Файл конфигурации /etc/modprobe.conf

Файл конфигурации /etc/modprobe.conf и специальные файлы конфигурации из директории /etc/modprobe.d могут содержать псевдонимы (для использования людьми), а также параметры (для зависимых модулей ядра ОС), используемые утилитой modprobe.

[root@RHEL52 ~]# cat /etc/modprobe.conf
alias scsi_hostadapter mptbase
alias scsi_hostadapter1 mptspi
alias scsi_hostadapter2 ata_piix
alias eth0 pcnet32
alias eth2 pcnet32
alias eth1 pcnet32

28.5. Компиляция ядра ОС

28.5.1. Дополнительная версия

Перейдите в директорию /usr/src/redhat/BUILD/kernel-2.6.9/linux-2.6.9/ и измените дополнительную версию ядра ОС, заменив значение параметра extraversion в файле с именем Makefile.

[root@RHEL52 linux-2.6.18.i686]# pwd
/usr/src/redhat/BUILD/kernel-2.6.18/linux-2.6.18.i686
[root@RHEL52 linux-2.6.18.i686]# vi Makefile
[root@RHEL52 linux-2.6.18.i686]# head -4 Makefile
VERSION = 2
PATCHLEVEL = 6
SUBLEVEL = 18
EXTRAVERSION = -paul2008

28.5.2. Команда make mrproper

Теперь необходимо удалить результаты предыдущих сборок ядра ОС с помощью команды make mrproper. В том случае, если это ваша первая сборка после загрузки исходного кода, вы не должны выполнять это действие.

[root@RHEL52 linux-2.6.18.i686]# make mrproper
CLEAN scripts/basic
CLEAN scripts/kconfig
CLEAN include/config
CLEAN .config .config.old

28.5.3. Файл .config

Теперь скопируйте работоспособный файл с именем .config из директории /boot в директорию с исходным кодом вашего ядра ОС. Этот файл содержит значения параметров конфигурации, которые использовались при компиляции вашего текущего ядра ОС. Грубо говоря, данный файл содержит информацию о том, должны ли компилироваться определенные модули ядра ОС.

[root@RHEL52 linux-2.6.18.i686]# cp /boot/config-2.6.18-92.1.18.el5 .config

28.5.4. Команда make menuconfig

Теперь следует выполнить команду make menuconfig (или команду make xconfig в случае использования графического интерфейса). Запущенная утилита позволит вам установить, необходимо ли компилировать определенный модуль (m), включить код модуля в состав ядра ОС (*) или вообще не компилировать его (что позволит сократить размер ядра ОС). Если вы откажитесь от компиляции слишком большого количества модулей, ваше ядро ОС не будет работать корректно. Данные конфигурации будут сохранены в скрытом файле с именем .config.

[root@RHEL52 linux-2.6.18.i686]# make menuconfig

28.5.5. Команда make clean

Выполните команду make clean для подготовки исходного кода ядра ОС к компиляции. Команда make clean позволяет удалить большую часть сгенерированных файлов, но сохраняет ваши файлы конфигурации ядра ОС. Исполнение команды make mrproper на данном этапе приведет к уничтожению файла .config, который был создан в результате исполнения команды make menuconfig.

[root@RHEL52 linux-2.6.18.i686]# make clean

28.5.6. Команда make bzImage

А теперь следует выполнить команду make bzImage, присесть и расслабиться, ожидая окончания процесса компиляции ядра ОС. Вы также можете использовать команду time make bzImage, чтобы получить информацию о времени, необходимым для компиляции ядра ОС, которая может понадобиться для планирования непродолжительной прогулки во время следующей компиляции.

[root@RHEL52 linux-2.6.18.i686]# time make bzImage
HOSTCC scripts/basic/fixdep

HOSTCC scripts/basic/docproc
HOSTCC scripts/kconfig/conf.o
HOSTCC scripts/kconfig/kxgettext.o
...

Исполнение данной команды завершится выводом информации о пути к результирующему файлу bzImage (а также времени компиляции, если вы также использовали команду time).

Kernel: arch/i386/boot/bzImage is ready (#1)
real 13m59.573s
user 1m22.631s
sys 11m51.034s
[root@RHEL52 linux-2.6.18.i686]#

На данном этапе вы можете вручную скопировать результирующий файл в директорию /boot с помощью команды cp arch/i386/boot/bzImage /boot/vmlinuz-<версия-ядра-ОС>.

28.5.7. Команда make modules

Теперь следует выполнить команду make modules. На компиляцию всех модулей ядра ОС уйдет от 20 до 50 минут.

[root@RHEL52 linux-2.6.18.i686]# time make modules
CHK include/linux/version.h
CHK include/linux/utsrelease.h
CC [M] arch/i386/kernel/msr.o
CC [M] arch/i386/kernel/cpuid.o
CC [M] arch/i386/kernel/microcode.o

28.5.8. Команда make modules_install

Для копирования всех скомпилированных модулей ядра ОС в директорию /lib/modules следует всего лишь выполнить команду make modules_install (на установку модулей уйдет примерно 20 секунд). В примере ниже приведен список содержимого директории модулей перед исполнением упомянутой команды.

[root@RHEL52 linux-2.6.18.i686]# ls -l /lib/modules/
итого 20
drwxr-xr-x 6 root root 4096 окт 15 13:09 2.6.18-92.1.13.el5
drwxr-xr-x 6 root root 4096 ноя 11 08:51 2.6.18-92.1.17.el5
drwxr-xr-x 6 root root 4096 дек 6 07:11 2.6.18-92.1.18.el5
[root@RHEL52 linux-2.6.18.i686]# make modules_install

А это список содержимого этой же директории после исполнения этой команды. Обратите внимание на то, что в процессе исполнения команды make modules_install была создана новая директория для модулей нового ядра ОС.

[root@RHEL52 linux-2.6.18.i686]# ls -l /lib/modules/
итого 24
drwxr-xr-x 6 root root 4096 окт 15 13:09 2.6.18-92.1.13.el5
drwxr-xr-x 6 root root 4096 ноя 11 08:51 2.6.18-92.1.17.el5
drwxr-xr-x 6 root root 4096 дек 6 07:11 2.6.18-92.1.18.el5
drwxr-xr-x 3 root root 4096 дек 6 08:50 2.6.18-paul2008

28.5.9. Директория /boot

Нам все также необходимо скопировать исполняемый файл ядра ОС, файл System.map и наш файл со значениями параметров конфигурации ядра ОС в директорию /boot. Строго говоря, копировать файл со значениями параметров конфигурации ядра ОС с именем .config совсем не обязательно, но он может пригодиться в будущем при компиляции ядра ОС.

[root@RHEL52]# pwd
/usr/src/redhat/BUILD/kernel-2.6.18/linux-2.6.18.i686
[root@RHEL52]# cp System.map /boot/System.map-2.6.18-paul2008
[root@RHEL52]# cp .config /boot/config-2.6.18-paul2008
[root@RHEL52]# cp arch/i386/boot/bzImage /boot/vmlinuz-2.6.18-paul2008

28.5.10. Утилита mkinitrd

Ядро ОС обычно использует файл initrd в процессе загрузки. Мы можем использовать утилиту mkinitrd для генерации этого файла. Убедитесь в том, что вы используете корректное имя ядра ОС!

[root@RHEL52]# pwd
/usr/src/redhat/BUILD/kernel-2.6.18/linux-2.6.18.i686
[root@RHEL52]# mkinitrd /boot/initrd-2.6.18-paul2008 2.6.18-paul2008

28.5.11. Системный загрузчик

На этом компиляция ядра ОС закончена, но не следует забывать о необходимости создания дополнительного станса для отображения пункта в меню системного загрузчика grub или lilo.

28.6. Компиляция отдельного модуля ядра ОС

28.6.1. Файл исходного кода hello.c

Небольшая программа на языке С послужит основой для нашего модуля ядра ОС.

```
[root@rhel4a kernel_module]# cat hello.c
#include <linux/module.h>
#include <section>
int init_module(void)
{
printk(KERN_INFO "Инициализация модуля Hello World...\n");
return 0;
}
void cleanup_module(void)
{
printk(KERN_INFO "Завершение работы модуля Hello World... \n");
}
```

28.6.2. Файл Makefile

Файл для сборки модуля ядра ОС будет содержать следующие строки.

```
[root@rhel4a kernel_module]# cat Makefile
obj-m += hello.o
all:
make -C /lib/modules/$(shell uname -r)/build M=$(PWD) modules
clean:
make -C /lib/modules/$(shell uname -r)/build M=$(PWD) clean
```

Для компиляции модуля необходимы только два описанных файла.

```
[root@rhel4a kernel_module]# ll
итого 16
-rw-rw-r-- 1 paul paul 250 фев 15 19:14 hello.c
-rw-rw-r-- 1 paul paul 153 фев 15 19:15 Makefile
```

28.6.3. Команда make

В примере ниже представлен результат исполнения команды make.

```
[root@rhel4a kernel_module]# make
make -C /lib/modules/2.6.9-paul-2/build M=~/kernel_module modules
make[1]: Entering dir... '/usr/src/redhat/BUILD/kernel-2.6.9/linux-2.6.9'
CC [M] /home/paul/kernel_module/hello.o
Building modules, stage 2.
MODPOST
CC /home/paul/kernel_module/hello.mod.o
LD [M] /home/paul/kernel_module/hello.ko
make[1]: Leaving dir... '/usr/src/redhat/BUILD/kernel-2.6.9/linux-2.6.9'
[root@rhel4a kernel_module]#
```

Теперь у нас больше файлов.

```
[root@rhel4a kernel_module]# ll
итого 172
-rw-rw-r-- 1 paul paul 250 фев 15 19:14 hello.c
-rw-r—r-- 1 root root 64475 фев 15 19:15 hello.ko
-rw-r—r-- 1 root root 632 фев 15 19:15 hello.mod.c
-rw-r—r-- 1 root root 37036 фев 15 19:15 hello.mod.o
-rw-r—r-- 1 root root 28396 фев 15 19:15 hello.o
-rw-rw-r-- 1 paul paul 153 фев 15 19:15 Makefile
[root@rhel4a kernel_module]#
```

28.6.4. Файл hello.c

Используйте утилиту modinfo для проверки того, действительно ли файл с расширением .ko является модулем ядра ОС.

```
[root@rhel4a kernel_module]# modinfo hello.ko
filename: hello.ko
vermagic: 2.6.9-paul-2 SMP 686 REGPARM 4KSTACKS gcc-3.4
depends:
[root@rhel4a kernel_module]#
```

Отлично, значит мы можем загрузить наш модуль ядра ОС с именем «hello».

```
[root@rhel4a kernel_module]# lsmod | grep hello
[root@rhel4a kernel_module]# insmod ./hello.ko
[root@rhel4a kernel_module]# lsmod | grep hello
hello 5504 0
[root@rhel4a kernel_module]# tail -1 /var/log/messages
Feb 15 19:16:07 rhel4a kernel: Инициализация модуля Hello World...
[root@rhel4a kernel_module]# rmmod hello
[root@rhel4a kernel_module]#
```

Наконец мы можем обнаружить небольшой сюрприз в файле журнала /var/log/messages.

```
[root@rhel4a kernel_module]# tail -2 /var/log/messages
Feb 15 19:16:07 rhel4a kernel: Инициализация модуля Hello World...
Feb 15 19:16:35 rhel4a kernel: Завершение работы модуля Hello World...
[root@rhel4a kernel_module]#
```

Глава 29. Работа с разделяемыми библиотеками

29.1. Вводная информация о разделяемых библиотеках

При разговоре о библиотеках мы будем иметь в виду динамически связываемые библиотеки (или разделяемые объекты). Они являются бинарными файлами, содержащими код функций, которые не исполняются как приложения, а могут использоваться из других бинарных файлов.

Несколько программ могут использовать одну и ту же разделяемую библиотеку. Имя файла разделяемой библиотеки обычно начинается с префикса lib, после которого должно быть записано само имя разделяемой библиотеки, расширение .so и, наконец, номер версии разделяемой библиотеки.

29.2. Директории /lib и /usr/lib

При просмотре содержимого директорий /lib и /usr/lib можно обнаружить большое количество символьных ссылок. Имена большинства файлов разделяемых библиотек содержат подробные описания номеров версий, причем для этих файлов также

создаются символьные ссылки с именами, содержащими лишь указания на номера основных версий.

root@rhel53 ~# ls -l /lib/libext*
lrwxrwxrwx 1 root root 16 фев 18 16:36 /lib/libext2fs.so.2 → libext2fs.so.2.4
-rwxr-xr-x 1 root root 113K июн 30 2009 /lib/libext2fs.so.2.4

29.3. Утилита ldd

Работоспособность многих программ зависят от наличия определенных разделяемых библиотек в системе. Вы можете получить информацию о необходимых для корректной работы программы разделяемых библиотеках с помощью утилиты ldd.

В примере показан список разделяемых библиотек, необходимых для корректной работы утилиты su.

paul@RHEL5 ~$ ldd /bin/su
linux-gate.so.1 => (0x003f7000)
libpam.so.0 => /lib/libpam.so.0 (0x00d5c000)
libpam_misc.so.0 => /lib/libpam_misc.so.0 (0x0073c000)
libcrypt.so.1 => /lib/libcrypt.so.1 (0x00aa4000)
libdl.so.2 => /lib/libdl.so.2 (0x00800000)
libc.so.6 => /lib/libc.so.6 (0x00ec1000)
libaudit.so.0 => /lib/libaudit.so.0 (0x0049f000)
/lib/ld-linux.so.2 (0x4769c000)

29.4. Утилита ltrace

Утилита ltrace позволяет ознакомиться со списком всех вызовов функций разделяемых библиотек, осуществленных определенной программой. В приведенном ниже примере используется параметр -p, позволяющий получить информацию лишь о суммарном количестве вызовов функций (каждая функция может вызываться множество раз), а также параметр -l, позволяющий выводить информацию о вызовах функций лишь из одного файла разделяемой библиотеки. Все это делается для того, чтобы узнать, какие библиотечные функции вызываются в процессе исполнения команды su - serena с привилегиями пользователя root.

root@deb503:~# ltrace -c -l /lib/libpam.so.0 su - serena
serena@deb503:~$ exit
logout
% time seconds usecs/call calls function
------ ----------- ----------- --------- --------------------
70.31 0.014117 14117 1 pam_start
12.36 0.002482 2482 1 pam_open_session
5.17 0.001039 1039 1 pam_acct_mgmt
4.36 0.000876 876 1 pam_end
3.36 0.000675 675 1 pam_close_session
3.22 0.000646 646 1 pam_authenticate
0.48 0.000096 48 2 pam_set_item
0.27 0.000054 54 1 pam_setcred
0.25 0.000050 50 1 pam_getenvlist
0.22 0.000044 44 1 pam_get_item
------ ----------- ----------- --------- --------------------
100.00 0.020079 11 total

29.5. Команды dpkg -S и debsums

Имя пакета программного обеспечения дистрибутива Debian/Ubuntu, который содержит указанную разделяемую библиотеку, может быть установлено следующим образом.

paul@deb503:/lib$ dpkg -S libext2fs.so.2.4
e2fslibs: /lib/libext2fs.so.2.4

После этого вы можете проверить целостность всех файлов, установленных в систему из данного пакета программного обеспечения, с помощью утилиты debsums.

paul@deb503:~$ debsums e2fslibs
/usr/share/doc/e2fslibs/changelog.Debian.gz OK
/usr/share/doc/e2fslibs/copyright OK
/lib/libe2p.so.2.3 OK
/lib/libext2fs.so.2.4 OK

Если файл разделяемой библиотеки поврежден, переустановите пакет программного обеспечения с помощью команды aptitude reinstall $имя_пакета.

root@deb503:~# aptitude reinstall e2fslibs
Чтение списков пакетов... Готово
Построение дерева зависимостей
Чтение информации о состоянии... Готово
Чтение информации о расширенных состояниях... Готово
Инициализация состояний пакетов... Готово
Reading task descriptions... Готово
Следующие пакеты будут ПЕРЕУСТАНОВЛЕНЫ:
e2fslibs
...

29.6. Команды rpm -qf и rpm -V

Имя пакета программного обеспечения дистрибутива RedHat/Fedora, который содержит указанную разделяемую библиотеку, может быть установлено следующим образом.

paul@RHEL5 ~$ rpm -qf /lib/libext2fs.so.2.4
e2fsprogs-libs-1.39-8.el5

После этого вы можете использовать команду rpm -V для проверки целостности всех файлов, установленных из данного пакета программного обеспечения. В выводе из расположенного ниже примера содержится информация о том, что размер (Size) и метка времени модификации (Time stamp) файла разделяемой

библиотеки изменились с момента его установки в систему.

root@rhel53 ~# rpm -V e2fsprogs-libs
prelink: /lib/libext2fs.so.2.4: prelinked file size differs
S.?....T /lib/libext2fs.so.2.4

В подобной ситуации вы можете использовать команду yum reinstall $имя_пакета для замены существующего в системе файла разделяемой библиотеки на его оригинальную версию.

root@rhel53 lib# yum reinstall e2fsprogs-libs
Loaded plugins: rhnplugin, security
Setting up Reinstall Process
Разрешение зависимостей
--> Проверка сценария
---> Пакет e2fsprogs-libs.i386 0:1.39-23.el5 помечен для удаления
---> Пакет e2fsprogs-libs.i386 0:1.39-23.el5 помечен для обновления
--> Проверка зависимостей окончена
...

Теперь при проверке целостности файлов из пакета программного обеспечения не выводится информации о каких-либо проблемах с файлом разделяемой библиотеки.

root@rhel53 lib# rpm -V e2fsprogs-libs
root@rhel53 lib#

29.7. Трассировка вызовов библиотечных функций с помощью утилиты strace

Более детальная трассировка вызовов всех функций из разделяемых библиотек может осуществляться с помощью утилиты strace. Начнем с создания файла, доступного только для чтения.

root@deb503:~# echo hello > 42.txt
root@deb503:~# chmod 400 42.txt
root@deb503:~# ls -l 42.txt
-r-------- 1 root root 6 2011-09-26 12:03 42.txt

После этого откроем данный файл с помощью текстового редактора vi, разместив в строке команды перед командой исполнения бинарного файла vi команду исполнения утилиты strace с записью данных трассировки в отдельный файл. Это позволит нам получить файл с информацией о всех вызовах функций разделяемых библиотек, осуществленных из бинарного файла vi.

root@deb503:~# strace -o strace.txt vi 42.txt

Созданный нами файл может быть открыт только для чтения, но мы все равно попытаемся изменить его содержимое и использовать директиву :w! для принудительной записи данных в него. После этого мы закроем текстовый редактор vi и перейдем к рассмотрению содержимого файла журнала трассировки вызовов функций разделяемых библиотек.

root@deb503:~# grep chmod strace.txt
chmod("42.txt", 0100600) = -1 ENOENT (No such file or directory)
chmod("42.txt", 0100400) = 0
root@deb503:~# ls -l 42.txt
-r-------- 1 root root 12 2011-09-26 12:04 42.txt

Обратите внимание на то, что текстовый редактор vi осуществлял изменение прав доступа к файлу дважды. Файл журнала трассировки вызовов функций разделяемых библиотек содержит большой объем информации, поэтому его содержимое не приводится в полном объеме в данной книге.

root@deb503:~# wc -l strace.txt
941 strace.txt

Глава 30. Резервные копии данных

30.1. Информация о ленточных накопителях

Не забывайте о том, что имена файлов устройств, грубо говоря, не имеют решающего значения, так как ядро операционной системы использует основной и дополнительный идентификаторы для поиска соответствующего устройства! Обратитесь к странице руководства утилиты mknod и файлу devices.txt из архива исходного кода ядра Linux для получения дополнительной информации.

30.1.1. Ленточные устройства с интерфейсом SCSI

В официальном списке имен файлов устройств Linux (http://www.lanana.org/docs/device-list/) можно обнаружить имена файлов устройств ленточных накопителей с интерфейсом SCSI (символьные устройства с основным идентификатором 9 — major 9 char). Ленточные накопители с интерфейсом SCSI представлены файлами устройств с именами, начинающимися с /dev/st и нумеруются начиная с 0 для первого ленточного накопителя.

/dev/st0 Первый ленточный накопитель
/dev/st1 Второй ленточный накопитель
/dev/st2 Третий ленточный накопитель

Для предотвращения автоматической перемотки ленты используйте файлы устройств с префиксом n в именах.

/dev/nst0 Первый ленточный накопитель без автоматической перемотки ленты
/dev/nst1 Второй ленточный накопитель без автоматической перемотки ленты
/dev/nst2 Третий ленточный накопитель без автоматической перемотки ленты

По умолчанию ленточные накопители с интерфейсом SCSI в Linux

используют наиболее мощный из поддерживаемых аппаратный алгоритм сжатия данных. Для снижения степени сжатия добавьте букву l (низкая степень сжатия — low), m (средняя степень сжатия — medium) или a (автоматическая установка степени сжатия — auto) к имени файла устройства.

/dev/st0l Первый ленточный накопитель с низкой степенью сжатия данных

/dev/st0m Первый ленточный накопитель со средней степенью сжатия данных

/dev/nst2m Третий ленточный накопитель со средней степенью сжатия данных без автоматической перемотки ленты

30.1.2. Ленточные накопители с интерфейсом IDE

В официальном списке имен файлов устройств Linux (http://www.lanana.org/docs/device-list/) можно обнаружить имена файлов устройств ленточных накопителей с интерфейсом IDE (символьные устройства с основным идентификатором 37 — major 37 char). Ленточные накопители с интерфейсом IDE представлены файлами устройств с именами, начинающимися с /dev/ht и нумеруются с 0 для первого ленточного накопителя. Имена файлов устройств могут модифицироваться с целью отключения автоматической перемотки ленты и указания степени сжатия таким же образом, как в случае с ленточными накопителями с интерфейсом SCSI.

/dev/ht0 Первый ленточный накопитель с интерфейсом IDE

/dev/nht0 Второй ленточный накопитель с интеб6рфейсом IDE без автоматической перемотки ленты

/dev/ht0m Первый ленточный накопитель с интерфейсом IDE со средней степенью сжатия данных

30.1.3. Утилита mt

Для управления вашими ленточными накопителями следует использовать утилиту mt (Magnetic Tape — магнитный накопитель). Некоторые примеры использования данной утилиты приведены ниже.

Команда для получения информации о состоянии ленточного накопителя.

mt -f /dev/st0 status

Команда для перемотки ленты...

mt -f /dev/st0 rewind

Команда для перемотки и извлечения ленты...

mt -f /dev/st0 eject

Команда для очистки ленты...

mt -f /dev/st0 erase

30.2. Сжатие данных

Сжатие данных в процессе создания их резервных копий может оказаться весьма полезным. Двумя наиболее популярными инструментами для сжатия обычных файлов в Linux являются утилиты gzip/gunzip и bzip2/bunzip2. Вы можете увидеть пример использования утилиты gzip ниже; обратите внимание на то, что данная утилита добавляет расширение .gz к имени сжатого файла.

```
paul@RHELv4u4:~/test$ ls -l allfiles.tx*
-rw-rw-r-- 1 paul paul 8813553 фев 27 05:38 allfiles.txt
paul@RHELv4u4:~/test$ gzip allfiles.txt
paul@RHELv4u4:~/test$ ls -l allfiles.tx*
-rw-rw-r-- 1 paul paul 931863 фев 27 05:38 allfiles.txt.gz
paul@RHELv4u4:~/test$ gunzip allfiles.txt.gz
paul@RHELv4u4:~/test$ ls -l allfiles.tx*
-rw-rw-r-- 1 paul paul 8813553 фев 27 05:38 allfiles.txt
paul@RHELv4u4:~/test$
```

В общем случае утилита gzip обрабатывает данные гораздо быстрее, чем утилита bzip2, но последняя утилита гораздо качественнее сжимает данные. Давайте сравним две этих утилиты.

```
paul@RHELv4u4:~/test$ cp allfiles.txt bllfiles.txt
paul@RHELv4u4:~/test$ time gzip allfiles.txt
real 0m0.050s
user 0m0.041s
sys 0m0.009s
paul@RHELv4u4:~/test$ time bzip2 bllfiles.txt
real 0m5.968s
user 0m5.794s
sys 0m0.076s
paul@RHELv4u4:~/test$ ls -l ?llfiles.tx*
-rw-rw-r-- 1 paul paul 931863 фев 27 05:38 allfiles.txt.gz
-rw-rw-r-- 1 paul paul 708871 май 12 10:52 bllfiles.txt.bz2
paul@RHELv4u4:~/test$
```

30.3. Утилита tar

Название утилиты tar расшифровывается как Tape ARchive (ленточный архив). Данный инструмент используется для приема и передачи данных (обмен данными обычно осуществляется с ленточными накопителями или обычными фалами). Параметр c используется для создания архива формата tar (или файла формата tar), а параметр f — для указания имени файла формата tar. С помощью команды из примера ниже создается резервная копия директории /etc, которая сохраняется в файл /backup/etc.tar.

```
root@RHELv4u4:~# tar cf /backup/etc.tar /etc
root@RHELv4u4:~# ls -l /backup/etc.tar
-rw-r—r-- 1 root root 47800320 май 12 11:47 /backup/etc.tar
root@RHELv4u4:~#
```

Для сжатия данных не потребуются программные каналы, ведь утилита tar поддерживает параметр z, предназначенный для сжатия данных с использованием алгоритма gzip, а также параметр j для сжатия данных с использованием алгоритма bzip2.

root@RHELv4u4:~# tar czf /backup/etc.tar.gz /etc
root@RHELv4u4:~# tar cjf /backup/etc.tar.bz2 /etc
root@RHELv4u4:~# ls -l /backup/etc.ta*
-rw-r—r-- 1 root root 47800320 май 12 11:47 /backup/etc.tar
-rw-r—r-- 1 root root 6077340 май 12 11:48 /backup/etc.tar.bz2
-rw-r—r-- 1 root root 8496607 май 12 11:47 /backup/etc.tar.gz
root@RHELv4u4:~#

Параметр t используется для вывода списка содержимого архива формата tar. Режим подробного вывода активируется с помощью параметра v (данный параметр также полезен в том случае, если вы хотите видеть информацию об обрабатываемых файлах в процессе их архивации).

root@RHELv4u4:~# tar tvf /backup/etc.tar
drwxr-xr-x root/root 0 2007-05-12 09:38:21 etc/
-rw-r—r-- root/root 2657 2004-09-27 10:15:03 etc/warnquota.conf
-rw-r—r-- root/root 13136 2006-11-03 17:34:50 etc/mime.types
drwxr-xr-x root/root 0 2004-11-03 13:35:50 etc/sound/
…

Для вывода информации об определенном файле из архива формата tar, следует использовать параметр t с передачей имени интересующего файла (без начального символа /).

root@RHELv4u4:~# tar tvf /backup/etc.tar etc/resolv.conf
-rw-r—r-- root/root 77 2007-05-12 08:31:32 etc/resolv.conf
root@RHELv4u4:~#

Используйте параметр x для извлечения всех файлов или определенного файла из архива. Помните о том, что по умолчанию утилита tar будет извлекать файлы в текущую директорию.

root@RHELv4u4:~# tar xvf /backup/etc.tar etc/resolv.conf
etc/resolv.conf
root@RHELv4u4:~# ls -l /etc/resolv.conf
-rw-r—r-- 2 root root 40 май 12 12:05 /etc/resolv.conf
root@RHELv4u4:~# ls -l etc/resolv.conf
-rw-r—r-- 1 root root 77 май 12 08:31 etc/resolv.conf
root@RHELv4u4:~#

Вы можете активировать режим сохранения прав доступа к файлам с помощью параметра p. Для задания файлов и директорий, которые не должны быть добавлены в архив, следует использовать параметр --exclude.

root ~# tar cpzf /backup/etc_with_perms.tgz /etc
root ~# tar cpzf /backup/etc_no_sysconf.tgz /etc --exclude /etc/sysconfig
root ~# ls -l /backup/etc_*
-rw-r—r-- 1 root root 8434293 май 12 12:48 /backup/etc_no_sysconf.tgz
-rw-r—r-- 1 root root 8496591 май 12 12:48 /backup/etc_with_perms.tgz
root ~#

Также вы можете создать текстовый файл с именами файлов и директорий для архивации, после чего передать данный файл утилите tar, использовав параметр -T.

root@RHELv4u4:~# find /etc -name *.conf > files_to_archive.txt
root@RHELv4u4:~# find /home -name *.pdf >> files_to_archive.txt
root@RHELv4u4:~# tar cpzf /backup/backup.tgz -T files_to_archive.txt

Утилита tar может принимать имена файлов от утилиты find посредством специальной утилиты xargs.

find /etc -type f -name "*.conf" | xargs tar czf /backup/confs.tar.gz

Вы можете использовать утилиту tar для копирования директории, причем данная команда будет более эффективной, чем команда cp -r.

(cd /etc; tar -cf - .) | (cd /backup/copy_of_etc/; tar -xpf -)

Другой пример использования утилиты tar предназначен для безопасного копирования директории по сети.

(cd /etc;tar -cf - .)|(ssh user@srv 'cd /backup/cp_of_etc/; tar -xf - ')

Утилита tar может использоваться совместно с утилитой gzip и клиентом ssh для копирования файла на удаленный сервер по протоколу ssh.

cat backup.tar | gzip | ssh bashuser@192.168.1.105 "cat - > backup.tgz"

Команда для сжатия файла резервной копии формата tar при передаче по сети с сохранением несжатого файла на целевом узле.

cat backup.tar | gzip | ssh user@192.168.1.105 "gunzip|cat - > backup.tar"

Аналогичная предыдущей команда со сжатием данных силами клиента ssh.

cat backup.tar | ssh -C bashuser@192.168.1.105 "cat - > backup.tar"

30.4. Типы резервных копий данных

В Linux используются многоуровневые инкрементальные резервные копии данных с жестким разделением уровней. Полная резервная копия данных является резервной копией данных уровня 0. Резервная копия данных более высокого уровня x будет содержать информацию обо всех изменениях данных с момента создания резервной копии данных предыдущего уровня x-1.

Представьте, что вы создали полную резервную копию данных

(уровня 0) в понедельник, а резервную копию уровня 1 — во вторник, следовательно, созданная во вторник резервная копия данных будет содержать информацию обо всех изменениях данных начиная с понедельника. Созданная в среду резервная копия данных уровня 2 будет содержать информацию обо всех изменениях данных начиная со вторника (предыдущий уровень резервной копии данных 2-1). Резервная копия данных уровня 3, созданная в четверг, будет содержать информацию обо всех изменениях данных начиная со среды (предыдущий уровень резервной копии данных 3-1). Другая резервная копия данных уровня 3, созданная в пятницу, будет также содержать информацию обо всех изменениях данных начиная со среды. Резервная копия данных уровня 2, созданная в субботу, будет содержать информацию обо всех изменениях данных с момента создания резервной копии данных предыдущего уровня 1, то есть, со вторника.

30.5. Утилиты dump и restore

Хотя утилита dump и похожа на утилиту tar, она значительно отличается от нее тем, что работает с данными на уровне файловой системы. В то время, как утилита tar принимает списки файлов для создания резервной копии, утилита dump самостоятельно осуществляет поиск файлов для создания резервной копии, исследуя файловую систему ext2. Файлы, обнаруженные утилитой dump, будут копироваться на ленточный накопитель или в обычный файл. В том случае, если на целевом устройстве не хватает свободного места для сохранения резервной копии данных (заканчивается лента), она разделяется на множество томов.

Извлечение файлов из созданной с помощью утилиты dump резервной копии осуществляется с помощью утилиты restore. В примере ниже полная резервная копия уровня 0 двух разделов жесткого диска сохраняется на ленточный носитель с помощью ленточного накопителя с интерфейсом SCSI. Необходимо использовать файл устройства ленточного накопителя без автоматической перемотки ленты для того, чтобы резервные копии разделов жесткого диска записывались на ленту друг за другом.

```
dump 0f /dev/nst0 /boot
dump 0f /dev/nst0 /
```

Вывод списка файлов архива, созданного с помощью утилиты dump, осуществляется с помощью команды dump -t, причем вы также можете осуществлять сравнение файлов с помощью команды dump -C.

Вы можете предотвратить резервное копирование произвольных файлов, изменив значения соответствующих атрибутов этих файлов с помощью утилиты chattr. Атрибут файла d в файловых системах семейства ext сообщает утилите dump о необходимости пропуска данного файла даже при создании полной резервной копии файловой системы. В следующем примере приведена команда, используемая для предотвращения резервного копирования файла /etc/hosts.

```
chattr +d /etc/hosts
```

Для полного восстановления файловой системы с помощью утилиты restore следует использовать параметр -r. Такой подход может оказаться полезным в случае возникновения необходимости в изменении размера файловой системы или изменения размера блока файловой системы. Перед восстановлением данных из резервной копии вы должны создать новую файловую систему, смонтировать ее в определенную директорию и перейти в эту директорию. Следует выполнять действия в такой последовательности, как показано в примере ниже.

```
mke2fs /dev/hda3
mount /dev/hda3 /mnt/data
cd /mnt/data
restore rf /dev/nst0
```

Для извлечения из резервной копии одного файла или директории используйте параметр -x.

```
restore -xf /dev/st0 /etc
```

30.6. Утилита cpio

Утилита cpio (Copy Input and Output) значительно отличается от утилит tar и dump. Она может принимать список имен файлов от сторонней утилиты, но при этом осуществляет последовательное копирование оригиналов этих файлов. Это обстоятельство позволяет достаточно просто использовать данную утилиту в сочетании с утилитой find! Некоторые примеры ее использования приведены ниже.

Утилита find передает имена файлов утилите cpio, которая, в свою очередь, помещает файлы в архив.

```
find /etc -depth -print | cpio -oaV -O archive.cpio
```

Аналогичная команда со сжатием результирующего архива с помощью утилиты gzip.

```
find /etc -depth -print | cpio -oaV | gzip -c > archive.cpio.gz
```

Теперь данные архива передаются по протоколу ssh (осуществляется резервное копирование файлов в сжатый архив на другой машине).

```
find /etc -depth -print|cpio -oaV|gzip -c|ssh server "cat - > etc.cpio.gz"
```

Утилита find передает имена файлов утилите cpio | утилита cpio передает файлы в архиве клиенту ssh | клиент ssh передает файлы в архиве утилите cpio утилита cpio извлекает файлы из архива.

find /etc -depth -print | cpio -oaV | ssh user@host cpio -imVd

Аналогичная команда, с помощью которой выполняется обратное действие: осуществляется копирование директории с удаленного узла на локальную машину.

ssh user@host "find path -depth -print | cpio -oaV" | cpio -imVd

30.7. Утилита dd

30.7.1. Информация об утилите dd

Некоторые люди используют утилиту dd для создания резервных копий данных. Эта утилита является очень мощным инструментом, но созданные с помощью нее резервные копии данных могут восстанавливаться на схожих по параметрам разделах дисковых устройств или дисковых устройствах. Однако, существует множество вариантов продуктивного использования утилиты dd. Ниже приведены некоторые примеры ее использования.

30.7.2. Создание файла образа оптического диска CDROM

Простейший способ создания файла образа любого оптического диска с расширением .iso связан с использованием утилиты dd. Параметр if позволяет указать путь к исходному файлу, of — к целевому файлу. После этого для создания копии оптического диска на основе его файла образа с расширением .iso может быть использован любой инструмент для записи образов оптических дисков.

dd if=/dev/cdrom of=/путь/к/файлу/cdrom.ISO

30.7.3. Создание файла образа гибкого диска

Возможно, данный пример на сегодняшний день и не является актуальным, но на всякий случай следует упомянуть о том, что утилита dd может использоваться для создания файла образа гибкого диска объемом в 1.44 Мб. Размер блока файловой системы задается с помощью параметра bs, а в качестве значения параметра count используется количество блоков для копирования.

dd if=/dev/floppy of=/путь/к/файлу/floppy.img bs=1024 count=1440

30.7.4. Копирование основной загрузочной записи

Утилита dd может использоваться для копирования основной загрузочной записи (Master Boot Record — MBR) жесткого диска, представленного соответствующим файлом устройства, к примеру, /dev/hda, в файл.

dd if=/dev/hda of=/MBR.img bs=512 count=1

30.7.5. Копирование файлов

В данном примере показана методика использования утилиты dd для копирования файлов. Для файла с именем summer.txt создается копия с именем copy_of_summer.txt.

dd if=/summer.txt of=/copy_of_summer.txt

30.7.6. Создание файлов образов жестких дисков или их разделов

Кому нужна утилита ghost, если утилита dd может создавать образы разделов жестких дисков (со сжатием).

dd if=/dev/hdb2 of=/image_of_hdb2.IMG

dd if=/dev/hdb2 | gzip > /image_of_hdb2.IMG.gz

30.7.7. Создание файлов определенного размера

Утилита dd может использоваться для создания файла любого размера. В первом примере создается файл размером в один мебибайт, а во втором — в один мегабайт.

dd if=/dev/zero of=file1MB count=1024 bs=1024

dd if=/dev/zero of=file1MB count=1000 bs=1024

30.7.8. Сохранение копии оптического диска CDROM на удаленном сервере

Разумеется, существует бесконечное количество вариантов использования комбинации утилит ssh и bzip2. В данном примере сжатый с помощью утилиты bzip2 файл образа оптического диска CDROM сохраняется на удаленном сервере.

dd if=/dev/cdrom |bzip2|ssh user@host "cat - > /backups/cd/cdrom.iso.bz2"

30.8. Утилита split

Утилита split может успешно использоваться для разделения файлов на фрагменты меньшего размера. Данное разделение может оказаться полезным для записи фрагментов файла на множество носителей в том случае, если не разделенный на фрагменты файл не умещается на единственном носителе. В примере ниже показан метод разделения файла размером в 5000 байт на три файла меньшего размера, причем размер каждого из результирующих файлов не превышает 2000 байт.

```
paul@laika:~/test$ ls -l
итого 8
-rw-r—r-- 1 paul paul 5000 2007-09-09 20:46 bigfile1
paul@laika:~/test$ split -b 2000 bigfile1 splitfile.
paul@laika:~/test$ ls -l
итого 20
-rw-r—r-- 1 paul paul 5000 2007-09-09 20:46 bigfile1
-rw-r—r-- 1 paul paul 2000 2007-09-09 20:47 splitfile.aa
-rw-r—r-- 1 paul paul 2000 2007-09-09 20:47 splitfile.ab
-rw-r—r-- 1 paul paul 1000 2007-09-09 20:47 splitfile.ac
```

30.9. Практическое задание: резервные копии данных

!! Проявите осторожность при использовании параметров утилиты tar и указании пути к файлу резервной копии данных,

ведь в случае ошибки могут быть повреждены файлы вашей системы !!

1. Создайте директорию (или раздел жесткого диска, если желаете) для хранения резервных копий данных. Используйте путь (или смонтируйте файловую систему из созданного раздела жесткого диска в директорию) /mnt/backup.

2a. Используйте утилиту tar для создания файла резервной копии данных директории /etc с именем /mnt/backup/etc_<дата>.tgz, причем этот файл должен быть сжат с использованием алгоритма gzip. (Замените <дата> на текущую дату.)

2b. Используйте утилиту tar для создания файла резервной копии данных директории /bin с именем /mnt/backup/bin_<дата>.tar.bz2, причем этот файл должен быть сжат с использованием алгоритма bzip2.

2c. Выберите по файлу из директорий /etc и /bin и проверьте, была ли создана резервная копия этих файлов.

2d. Извлеките два выбранных файла из архивов в вашу домашнюю директорию.

3a. Создайте директорию для хранения файлов резервных копий данных вашего соседа и сделайте ее доступной по пути /mnt/<имя_компьютера_соседа>.

3b. Используйте комбинацию из клиента ssh и утилиты tar для сохранения резервной копии данных вашей директории /boot на компьютере вашего соседа в директории /mnt/<имя_вашего_компьютера>.

4a. Используйте комбинацию из утилит find и cpio для создания архива содержимого директории /etc формата cpio.

4b. Выберите файл из директории /etc и извлеките его из архива формата cpio в вашу домашнюю директорию.

1. Используйте утилиту dd и клиент ssh для сохранения файла резервной копии данных основной загрузочной записи жесткого диска на компьютере вашего соседа.
2. (Задание должно выполняться на реальном компьютере.) Создайте и смонтируйте файл образа установочного диска Ubuntu с расширением .iso.
3. Используйте комбинацию из утилит dd и gzip для создания виртуального образа одного из ваших разделов жесткого диска в другом разделе.
4. Используйте утилиту dd для создания файла размером в пять мегабайт в директории ~/testsplit с именем biggest. После этого разделите этот файл на файлы размером до двух мегабайт.

```
mkdir testsplit
dd if=/dev/zero of=~/testsplit/biggest count=5000 bs=1024
split -b 2000000 biggest parts
```

Приложение A. Дисковые квоты

A.1. Информация о дисковых квотах

Для ограничения доступного пользователю дискового пространства вы можете устанавливать дисковые квоты. Для этого необходимо добавить параметры монтирования usrquota и/или grpquota в файл /etc/fstab для одной или нескольких файловых систем.

```
root@RHELv4u4:~# cat /etc/fstab | grep usrquota
/dev/VolGroup00/LogVol02 /home ext3 usrquota,grpquota 0 0
```

После этого необходимо повторно смонтировать соответствующую файловую систему.

```
root@RHELv4u4:~# mount -o remount /home
```

Следующий шаг заключается в создании файлов quota.user и/или quota.group. Эти файлы (называемые файлами квот) должны содержать таблицу распределения дискового пространства для рассматриваемой файловой системы. Для выполнения данного действия используйте команду quotacheck.

```
root@RHELv4u4:~# quotacheck -cug /home
root@RHELv4u4:~# quotacheck -avug
```

Параметр -c предназначен для создания дисковой квоты, параметр u — для указания на то, что создается квота для пользователя, параметр v — для указания на то, что создается дисковая квота для группы пользователей, параметр a — для проверки активации всех механизмов квот для файловых систем, описанных в рамках файла /etc/fstab, а параметр v — для активации режима вывода подробной информации. На следующем шаге следует отредактировать отдельные дисковые квоты пользователей с помощью утилиты edquota или установить общую дисковую квоту для файловой системы с помощью команды edquota -t. Данный инструмент позволит вам установить жесткое (hard, реальное ограничение) и гибкое (soft, ограничение с допуском) ограничения количества блоков и структур inode. Утилита quota позволяет проверить, была ли установлена дисковая квота для пользователя. Также вы можете получить отличный отчет в случае использования утилиты repquota.

Финальным шагом (перед тем, как пользователи начнут жаловаться на недостаток дискового пространства) является активация дисковых квот с помощью утилиты quotaon(1).

```
root@RHELv4u4:~# quotaon -vaug
```

Используйте утилиту quotaoff для того, чтобы пользователи перестали жаловаться на ограничения.

```
root@RHELv4u4:~# quotaoff -vaug
```

A.2. Практическое задание: дисковые квоты

1. Активируйте дисковую квоту для одного из новых разделов жесткого диска. Ограничьте доступное для одного из ваших пользователей дисковое пространство 10 мегабайтами.
2. Проверьте корректность установки дисковой квоты, скопировав множество файлов на соответствующий раздел жесткого диска.

Приложение B. Краткая информация о протоколе VNC

B.1. Информация о протоколе VNC

Возможность доступа к рабочему столу по протоколу VNC может быть активирована в окружениях рабочего стола GNOME или KDE на странице настроек удаленного рабочего стола (Remote Desktop Preferences). Протокол VNC может использоваться для доступа к рабочему столу вашего компьютера с другого компьютера, причем вы также можете использовать данный протокол для доступа к рабочему столу другого пользователя. Последний вариант использования протокола может оказаться полезным для демонстрации пользователям методики выполнения тех или иных задач. Главным преимуществом протокола VNC является независимость от используемых операционных систем, причем имеется множество приложений, которые поддерживают данный протокол (realvnc, tightvnc, xvnc, ...) и предназначены для использования в операционных системах Solaris, Linux, BSD и других.

B.2. Сервер VNC

Методика первоначального запуска сервера VNC.

```
[root@RHELv4u3 conf]# rpm -qa | grep -i vnc
vnc-server-4.0-8.1
vnc-4.0-8.1
[root@RHELv4u3 conf]# vncserver :2
You will require a password to access your desktops.
Password:
Verify:
xauth: creating new authority file /root/.Xauthority
New RHELv4u3.localdomain:2 (root) desktop is RHELv4u3.localdomain:2
Creating default startup script /root/.vnc/xstartup
Starting applications specified in /root/.vnc/xstartup
Log file is /root/.vnc/RHELv4u3.localdomain:2.log
[root@RHELv4u3 conf]#
```

B.3. Клиент VNC

Теперь вы можете использовать клиент vncviewer с другой машины для соединения с вашим сервером VNC. По умолчанию будет использоваться простой графический интерфейс...

```
paul@laika:~$ vncviewer 192.168.1.49:2
VNC viewer version 3.3.7 - built Nov 20 2006 13:05:04
Copyright © 2002-2003 RealVNC Ltd.
Copyright © 1994-2000 AT&T Laboratories Cambridge.
See http://www.realvnc.com for information on VNC.
VNC server supports protocol version 3.8 (viewer 3.3)
Password:
VNC authentication succeeded
Desktop name "RHELv4u3.localdomain:2 (root)"
Connected to VNC server, using protocol version 3.3
...
```

Если вы не желаете использовать простой оконный менеджер twm, вы можете закомментировать две последних строки файла ~/.vnc/xstartup и добавить строку gnome-session & для использования окружения рабочего стола GNOME по умолчанию при соединении с системой посредством протокола VNC.

```
[root@RHELv4u3 ~]# cat .vnc/xstartup
#!/bin/sh
# Uncomment the following two lines for normal desktop:
# (Раскомментируйте две следующие строки для использования полноценного окружения рабочего стола:)
# unset SESSION_MANAGER
# exec /etc/X11/xinit/xinitrc
[ -x /etc/vnc/xstartup ] && exec /etc/vnc/xstartup
[ -r $HOME/.Xresources ] && xrdb $HOME/.Xresources
xsetroot -solid grey
vncconfig -iconic &
# xterm -geometry 80x24+10+10 -ls -title "$VNCDESKTOP Desktop" &
# twm &
gnome-session &
[root@RHELv4u3 ~]#
```

Не забудьте перезапустить ваш сервер VNC после модификации данного файла.

```
[root@RHELv4u3 ~]# vncserver -kill :2
Killing Xvnc process ID 5785
[root@RHELv4u3 ~]# vncserver :2
New RHELv4u3.localdomain:2 (root) desktop is RHELv4u3.localdomain:2
Starting applications specified in /root/.vnc/xstartup
```

Log file is /root/.vnc/RHELv4u3.localdomain:2.log

B.4. Практическое задание: краткая информация о протоколе VNC

1. Используйте протокол VNC для доступа к рабочему столу другой машины.

Приложение C. Лицензия

GNU Free Documentation License
Version 1.3, 3 November 2008
Copyright © 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

1. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.
This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.
We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.
A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.
A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.
The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.
The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.
A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent.
An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".
Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The "publisher" means any person or entity that distributes copies of the Document to the public.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

1. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

1. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

1. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license

notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document’s license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version’s license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

1. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number.

Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

1. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

1. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

1. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

1. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

1. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number.

If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

1. RELICENSING

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

"Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this

License
  somewhere other than this MMC, and subsequently incorporated in whole
  or in part into the MMC, (1) had no cover texts or invariant sections,
  and (2) were thus incorporated prior to November 1, 2008.
  The operator of an MMC Site may republish an MMC contained in the site
  under CC-BY-SA on the same site at any time before August 1, 2009,
  provided the MMC is eligible for relicensing.