Впервые опубликовано на developerWorks 14.11.2007

Учебное пособие для экзамена LPI 102, тема 111: Задачи администрирования

Администрирование для специалистов (LPIC-1) тема 111

Учетные записи пользователей и групп

Как вы знаете из учебного пособия "Подготовка к экзамену LPI 101 (тема 104) Устройства, файловые системы Linux и стандарт Filesystem Hierarchy Standard", Linux является многопользовательской системой, в которой каждый пользователь принадлежит одной *основной* группе и, возможно, нескольким дополнительным группам. В Linux права на файлы тесно связаны с идентификаторами пользователей (id) и группами. Вспомните, что можно войти в систему в качестве какого-либо пользователя и при помощи команд `su` или `sudo -s` стать другим пользователем и что можно воспользоваться командой `whoami` для проверки текущего действительного id и командой `groups`, чтобы узнать, какой группе вы принадлежите. Из этого раздела вы узнаете, как создавать и удалять пользователей и группы и управлять ими. Также вы узнаете о содержащихся в каталоге /etc файлах, в которых хранится информация о пользователях и группах.

Добавление и удаление пользователей и групп

Вы добавляете пользователя в систему Linux при помощи команды `useradd` и удаляете при помощи команды `userdel`. Подобным образом, вы добавляете или удаляете группы при помощи команд `groupadd` и `groupdel`.

Добавление пользователя или группы

Администрирование пользователей и групп в современных системах Linux обычно производится при помощи соответствующего графического интерфейса. Обычно доступ к нему можно получить через меню для системного администрирования. Существует большое разнообразие этих интерфейсов, так что интерфейс, присутствующий в вашей системе, может выглядеть не так, как в примере, приведенном здесь, но основные понятия и команды будут похожи.

Давайте начнем с добавления пользователя в графической системе Fedora Core 5 и затем рассмотрим приведенные выше команды. В системе Fedora Core 5 с десктопом GNOME выберите **Система > Администрирование > Пользователи и Группы** и нажмите кнопку **Добавить пользователя**.

На рисунке 1 изображены окно User Manager (Менеджер пользователей) и окно Create New User (Создать пользователя), содержащее основную информацию для нового пользователя по имени 'john'. Были введены полное имя пользователя (Full name) John Doe, и пароль (Password). Оболочка (Login Shell) /bin/bash предоставлена по умолчанию. В системах Fedora по умолчанию создается новая группа, имя которой совпадает с именем пользователя, в нашем случае 'john', и домашний каталог /home/john.

В листинге 1 показан пример использования команды `id` для просмотра основной информации о новом пользователе. Как вы можете видеть, john имеет идентификатор пользователя 503 и соответствующую группу john с номером 503. john является членом только этой группы.

Листинг 1. Просмотр информации об id пользователя

```
[root@pinguino ~]# id john
uid=503(john) gid=503(john) groups=503(john)
```

Для выполнения той же задачи из командной строки воспользуйтесь командами `groupadd` и `useradd` для создания группы и пользователя, а затем командой `passwd`, чтобы установить пароль для вновь созданного пользователя. Для выполнения всех этих команд необходимы привилегии пользователя root. Основные приемы использования этих команд для добавления другого пользователя, jane, показаны в листинге 2.

Листинг 2. Добавление пользователя jane

```
[root@pinguino ~]# groupadd jane
[root@pinguino ~]# useradd -c "Jane Doe" -g jane -m jane
[root@pinguino ~]# passwd jane
Changing password for user jane.
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully.
[root@pinguino ~]# id jane
uid=504(jane) gid=504(jane) groups=504(jane)
[root@pinguino ~]# ls -ld /home/jane
drwx------ 3 jane jane 4096 Jun 25 18:22 /home/jane
```

В этих двух примерах и id пользователя, и id группы имеют значение выше 500. В некоторых современных системах пользовательские id начинаются не с 500, а с 1000. Обычно эти значения являются признаком обычных пользователей, в то время как значения ниже 500 (или 1000, если в системе отсчет обычных пользователей начинается с 1000) зарезервированы для *системных пользователей*. Действующие точки отсечения установлены в /etc/login.defs как `UID_MIN` и `GID_MIN`.

В листинге 2, приведенном выше, команда `groupadd` имеет один параметр, `jane`, имя добавляемой группы. Имена групп должны начинаться с букв нижнего регистра или знаков подчеркивания и обычно содержат только эти символы, а также дефисы или черточки. Опции, которые можно указать этой команде, показаны в таблице 3.

Таблица 3. Опции команды groupadd

Опция	**Назначение**
`-f`	Выйти со статусом успешного выполнения, если группа уже существует. Удобна при написании скриптов, когда нет необходимости проверять, существует ли группа, прежде чем пытаться ее создавать.
`-g`	Задать id группы вручную. По умолчанию используется самое маленькое значение, не меньше, чем `GID_MIN`, причем больше, чем id любой из существующих групп. id групп обычно уникальны и не должны быть отрицательными
`-o`	Разрешить группу с неуникальным id.
`-K`	Может использоваться для отмены значений по умолчанию, хранящихся в файле /etc/login.defs.

В листинге 2, приведенном выше, команда `useradd` имеет один параметр, `jane`, имя добавляемого пользователя, а также опции `-c`, `-g` и `-m`. Наиболее часто употребляемые опции команды `useradd` показаны в таблице 4.

Таблица 4. Опции команды useradd

Опция	**Назначение**
`-b` `--basedir`	Базовый каталог по умолчанию, в котором создаются домашние каталоги пользователей. Обычно это /home, а пользовательские каталоги — /home/$USER.
`-c` `--comment`	Текстовая строка для описания id, содержащая, например, полное имя пользователя.
`-d` `--home`	Предоставляет определенное имя каталога для домашнего каталога.
`-e` `--expiredate`	Дата, когда учетная запись потеряет силу или будет заблокирована. Задается в формате YYYY-MM_DD.
`-g` `--gid`	Имя или номер начальной группы регистрации пользователя. Группа должна существовать, и поэтому в листинге 2 группа jane была создана раньше пользователя jane.
`-G` `--groups`	Список дополнительных групп, которым принадлежит пользователь. Группы перечисляются через запятую.
`-K`	Может использоваться для отмены значений по умолчанию, хранящихся в файле /etc/login.defs.
`-m` `--create-home`	Создает домашний каталог пользователя, если он не существует. Копирует скелетные файлы и другие каталоги из /etc/skel в домашний каталог.
`-o` `--non-unique`	Позволяет создать пользователя с неуникальным id.
`-p` `--password`	Шифрованный пароль. Если пароль не определен, по умолчанию учетная запись заблокирована. Вместо того чтобы создавать шифрованный пароль и определять его в команде `useradd`, обычно для создания пароля вы будете использовать команду `passwd`.
`-s` `--shell`	Имя login shell пользователя, если оно отличается от login shell по умолчанию.
`-u` `--uid`	Неотрицательное цифровое значение id пользователя, которое должно быть уникальным, если не определено иначе опцией `-o`. По умолчанию используется самое маленькое значение, не меньше, чем `UID_MIN`, причем больше, чем id любого из существующих пользователей.

Примечания:

1. В некоторых системах, в том числе в дистрибутивах Fedora и Red Hat, имеются расширения в виде команд для создания пользователей. Например, в системах Fedora и Red Hat по умолчанию для пользователя создается новая группа, и для запрета этой функции при выполнении команды `useradd` используется опция `-n`. Следует знать, что такие различия возможны, и при возникновении сомнений обращаться к страницам руководства man вашей системы.
2. В системах SUSE для доступа к графическому интерфейсу администрирования пользователей и групп используется YaST или YaST2.
3. Графические интерфейсы могут использоваться для решения дополнительных задач, например, для создания файла для почты пользователя в каталоге /var/spool/mail.

Удаление пользователя или группы

Удаление пользователя или группы значительно проще, чем их создание, поскольку имеет меньше опций. Фактически, команде `groupdel` для удаления группы требуется только имя группы; эта команда не имеет опций. Вы не можете удалить группу, если она является основной группой пользователя. Если для удаления пользователей и групп вы используете графический интерфейс, действия очень похожи на команды, показанные выше.

Воспользуйтесь командой `userdel`, чтобы удалить пользователя. Опция `-r` или `--remove` дает указание удалить домашний каталог пользователя и все его содержимое вместе с пользовательской почтой. Когда вы удаляете пользователя, группа, имеющая то же имя, что и пользователь, также удаляется, если переменная `USERGROUPS_ENAB` в файле /etc/login.defs установлена в положение yes, но это будет сделано, только если группа не является основной для другого пользователя.

В листинге 3 вы видите пример удаления группы, когда несколько пользователей разделяют одну и ту же основную группу. Здесь другой пользователь, jane2, был предварительно добавлен в систему с той же группой, что и jane.

Листинг 3. Удаление пользователей и групп

```
root@pinguino:~# groupdel jane
groupdel: cannot remove user's primary group.
root@pinguino:~# userdel -r jane
userdel: Cannot remove group jane which is a primary group for another user.
root@pinguino:~# userdel -r jane2
root@pinguino:~# groupdel jane
```

Примечания:

1. Для удаления пользователей и их групп команда `userdel` может быть использована с опцией `-f` или `--force`. Эта опция опасна, поэтому использовать ее следует только в крайнем случае. Прежде чем сделать это, внимательно прочтите руководство man.
2. Следует знать, что если вы удаляете пользователя или группу и в файловой системе есть файлы, принадлежащие этому пользователю или группе, эти файлы автоматически не удаляются или присваиваются другому пользователю или группе.

Приостановка и изменение учетных записей

Теперь, когда вы можете создать или удалить id пользователя или группу, у вас может возникнуть потребность изменить их.

Изменение учетных записей пользователей

Предположим, пользователь john хочет иметь в качестве оболочки по умолчанию tcsh. В графическом интерфейсе вы, как правило, найдете способ отредактировать данные о пользователе (или группе) или просмотреть свойства объекта.

Для изменения учетной записи пользователя из командной строки используйте команду `usermod`. Можно использовать большинство опций, которые используются с командой `useradd`, за исключением того, что для пользователя нельзя создать или наполнить содержимым новый домашний каталог. Если необходимо изменить имя пользователя,

используйте опцию `-l` или `--login` в сочетании с новым именем. Возможно, вы захотите переименовать домашний каталог, чтобы он соответствовал id пользователя. У вас также может возникнуть необходимость переименовать другие элементы, такие как почтовые spool-файлы. Наконец, если login shell изменен, может возникнуть необходимость изменить некоторые связанные с ним profile-файлы. В листинге 4 показан пример операций, которые необходимо выполнить, чтобы изменить пользователя john на john2 с /bin/tcsh в качестве shell по умолчанию и переименовать домашний каталог на /home/john2.

Листинг 4. Изменение пользователя

```
[root@pinguino ~]# usermod -l john2 -s /bin/tcsh -d /home/john2 john
[root@pinguino ~]# ls -d ~john2
ls: /home/john2: No such file or directory
[root@pinguino ~]# mv /home/john /home/john2
[root@pinguino ~]# ls -d ~john2
/home/john2
```

Примечания:

1. Если вам необходимо изменить дополнительные группы пользователя, вы должны определить полный список дополнительных групп. Не существует команды, чтобы просто добавить или удалить единственную группу для пользователя.
2. Существуют ограничения на изменение имени или id для пользователя, который зарегистрирован в системе или который выполняет какие-либо процессы. За подробностями обратитесь к страницам руководства man.
3. Если вы меняете номер пользователя, у вас может возникнуть желание изменить владельца файлов и каталогов этого пользователя в соответствии с новым номером.

Изменение групп

Ни для кого не является сюрпризом, что команда `groupmod` используется для изменения информации о группе. При помощи опции `groupmod` можно изменить номер, а при помощи опции `-n` — имя группы.

Листинг 5. Переименование группы

```
[root@pinguino ~]# ls -ld ~john2
drwx------ 3 john2 john 4096 Jun 26 18:29 /home/john2
[root@pinguino ~]# groupmod -n john2 john
[root@pinguino ~]# ls -ld ~john2
drwx------ 3 john2 john2 4096 Jun 26 18:29 /home/john2
```

В листинге 5 обратите внимание, что, когда мы использовали команду `groupmod` для изменения имени группы, имя группы для домашнего каталога пользователя john2 чудесным образом изменилось. Вы удивлены? Это не удивительно, поскольку в файловой системе группы представлены их номерами, а не именами. Однако, если вы изменяете номер группы, вы должны обновить всех пользователей, для которых эта группа является основной, кроме того у вас может возникнуть желание обновить файлы и каталоги, принадлежащие этой группе, в соответствии с новым номером (так же, как было сказано выше, где речь шла об изменении номера пользователя). В листинге 6 показано, как изменить номер группы для john2 на 505,

обновить учетную запись пользователя и произвести соответствующие изменения для всех файлов, входящих в файловую систему /home. Вы, вероятно, захотите изменить номера пользователей и групп, если это вообще возможно.

Листинг 6. Переименование группы

```
[root@pinguino ~]# groupmod -g 505 john2
[root@pinguino ~]# ls -ld ~john2
drwx------ 3 john2 503 4096 Jun 26 18:29 /home/john2
[root@pinguino ~]# id john2
uid=503(john2) gid=503 groups=503
[root@pinguino ~]# usermod -g john2 john2
[root@pinguino ~]# id john2
uid=503(john2) gid=505(john2) groups=505(john2)
[root@pinguino ~]# ls -ld ~john2
drwx------ 3 john2 503 4096 Jun 26 18:29 /home/john2
[root@pinguino ~]# find /home -gid 503 -exec chgrp john2 {} \;
[root@pinguino ~]# ls -ld ~john2
drwx------ 3 john2 john2 4096 Jun 26 18:29 /home/john2
```

Пароли пользователей и групп

Вам уже встречалась команда `passwd`, которая используется для изменения пароля пользователя. Пароль является уникальным (или должен быть таковым) для пользователя и может быть изменен пользователем. Как вы уже видели, пользователь root может изменить пароль любого пользователя.

Группы также могут иметь пароли, и для их установки используется команда `gpasswd`. Наличие пароля группы позволяет пользователям временно войти в группу при помощи команды `newgrp`, если им известен пароль группы. Конечно, наличие пароля, известного нескольким пользователям в некоторой степени проблематично, поэтому необходимо оценить преимущества добавления пользователя в группу при помощи команды `usermod` в сравнении с проблемой безопасности при слишком большом количестве людей, знающих пароль группы.

Приостановка и блокирование учетных записей

Если необходимо запретить пользователю регистрацию в системе, можно *приостановить* или *заблокировать* учетную запись при помощи команды `usermod` с опцией `-L`. Для разблокирования учетной записи используется опция `-U`. В листинге 7 показано, как заблокировать учетную запись john2, и что произойдет, если john2 попытается зарегистрироваться в системе. Обратите внимание, что когда учетная запись john2 разблокируется, восстанавливается ее прежний пароль.

Листинг 7. Блокирование учетной записи

```
[root@pinguino ~]# usermod -L john2
[root@pinguino ~]# ssh john2@pinguino
john2@pinguino's password:
Permission denied, please try again.
```

Мы кратко упомянули о возможности использования команды `passwd` для установки пароля пользователя, но и эта команда, и команды `usermod` и `chage` могут выполнять множество задач, связанных с учетными записями пользователей. Некоторые их опции показаны в таблице 5. За более подробной информацией об этих и других опциях обратитесь к соответствующим страницам руководства man.

Таблица 5. Команды и опции для изменения учетных записей пользователей

Опция команды			**Назначение**
Usermode	Passwd	Chage	
`-L`	`-l`	`N/A`	Блокирует или приостанавливает действие учетной записи.
`-U`	`-u`	`N/A`	Разблокирует учетную запись.
`N/A`	`-d`	`N/A`	Блокирует учетную запись путем отмены ее пароля.
`-e`	`-f`	`-E`	Устанавливает дату прекращения полномочий для учетной записи.
`N/A`	`-n`	`-m`	Минимальное время действия пароля в днях.
`N/A`	`-x`	`-M`	Максимальное время действия пароля в днях.
`N/A`	`-w`	`-W`	Число дней, за которое появляется предупреждение о необходимости изменить пароль.
`-f`	`-i`	`-l`	Число дней после того, как пароль потеряет силу, но до того, как учетная запись будет отключена.
`N/A`	`-S`	`-l`	Вывод краткого сообщения о статусе текущей учетной записи.

Управление базами данных пользователей и групп

Основные репозитории, содержащие информацию о пользователях и группах, — это четыре файла в каталоге /etc.

/etc/passwd
: файл *паролей*, содержащий основную информацию о пользователях

/etc/shadow
: файл *теневых паролей*, содержащий шифрованные пароли

/etc/group
: файл *групп*, содержащий основную информацию о группах и принадлежащих этим группам пользователях

/etc/gshadow
: файл *теневых групп*, содержащий шифрованные пароли групп

Эти файлы обновляются при помощи команд, которые вы уже видели в этом учебном пособии, кроме того, после того как мы обсудим сами эти файлы, вам встретятся другие команды для работы с ними. Все эти файлы являются простыми текстовыми файлами. Вообще, вы не должны редактировать их непосредственно. Для их обновления используются специальные инструменты, так что они должным образом блокируются и поддерживаются в синхронном состоянии.

Обратите внимание, что файлы passwd и group являются *затеняемыми*. Это сделано из соображений безопасности. Сами файлы passwd и group должны быть доступными для чтения для всех, а зашифрованные пароли — недоступными для чтения для всех. Поэтому зашифрованные пароли хранятся в теневых файлах, и эти файлы доступны для чтения только пользователю root. Необходимый доступ для изменения аутентификационных данных обеспечивается при помощи suid-программы, которая имеет полномочия пользователя root, но может быть запущена любым пользователем. Убедитесь, что в системе установлены соответствующие права доступа. В листинге 8 показан пример.

Листинг 8. Права доступа к базам данных пользователей и групп

```
[ian@pinguino ~]$ ls -l /etc/passwd /etc/shadow /etc/group /etc/gshadow
-rw-r--r-- 1 root root  701 Jun 26 19:04 /etc/group
-r-------- 1 root root  580 Jun 26 19:04 /etc/gshadow
-rw-r--r-- 1 root root 1939 Jun 26 19:43 /etc/passwd
-r-------- 1 root root 1324 Jun 26 19:50 /etc/shadow
```

Примечание: Несмотря на то, что все еще существует техническая возможность работы без теневых файлов паролей и групп, эта возможность почти некогда не используется и пользоваться ею не рекомендуется.

Файл /etc/passwd

Файл /etc/passwd содержит одну строку для каждого пользователя системы. В листинге 9 показано несколько примеров строк.

Листинг 9. Записи из файла /etc/password

```
root:x:0:0:root:/root:/bin/bash
jane:x:504:504:Jane Doe:/home/jane:/bin/bash
john2:x:503:505:John Doe:/home/john2:/bin/tcsh
```

Каждая строка содержит семь полей, разделенных двоеточиями (:), как показано в таблице 6.

Таблица 6. Поля файла /etc/passwd

Поле	**Назначение**
Имя пользователя (Username)	Имя, используемое для входа в систему. Например, john2.
Пароль (Password)	Зашифрованный пароль. Если используется зашифрованный пароль, это поле содержит единичный символ x.
id пользователя (UID)	Число, используемое для представления этого пользователя в системе. Например, 503 для пользователя john2.
id группы (GID)	Число, используемое для представления этой основной группы пользователя в системе. Например, 505 для пользователя john2.
Комментарий (GECOS)	Необязательное поле, используемое для описания пользователя. Например, "John Doe". Это поле может содержать несколько разделенных запятыми записей. Оно также используется такой программой как `finger`. Название поля GECOS сложилось исторически. Подробнее см. в `man 5 passwd`.
Домашний каталог (Home)	Абсолютный путь для домашнего каталога пользователя. Например, /home/john2.
Командная ободочка (shell)	Программа, которая автоматически запускается при входе пользователя в систему. Обычно это интерактивный shell, такой как /bin/bash или /bin/tcsh, но это может быть и другая программа, не обязательно интерактивный shell.

Файл /etc/group

Файл /etc/group содержит одну строку для каждой группы системы. В листинге 10 показано несколько примеров строк.

Листинг 10. Записи в /etc/group

```
root:x:0:root
jane:x:504:john2
john2:x:505:
```

Каждая строка содержит четыре поля, разделенных двоеточиями (:), как показано в таблице 7.

Таблица 7. Поля файла /etc/group

Поле	**Назначение**
Имя группы (Groupname)	Имя этой группы. Например, john2.
Пароль (Password)	Зашифрованный пароль. Если используется зашифрованный пароль группы, это поле содержит единичный символ x.
id группы (GID)	Число, используемое для представления этой группы в системе. Например, 505 для группы john2.
Члены (Members)	Разделенный запятыми список членов группы, за исключением тех членов, для которых это группа является основной.

Теневые файлы

Файл /etc/shadow должен быть доступен для чтения только для пользователя root. Он содержит зашифрованные пароли наряду с паролем и информацией о времени истечения действия учетной записи. Информацию о значении полей см. в man-странице (`man 5 shadow`). Пароли могут быть зашифрованы при помощи DES, но чаще для шифрования используется MD5. Алгоритм DES использует 7 младших битов из первых 8 символов пароля пользователя, представленных в виде 56-битного ключа, в то время как алгоритм MD5 использует весь пароль. В любом случае пароли кодированы при помощи *salt-кода*, так что из двух идентичных паролей не будут сгенерированы одинаковые зашифрованные значения. В листинге 11 показано, как установить одинаковые пароли для пользователей jane и john2, и затем показан результат шифрования паролей при помощи MD5 в файле /etc/shadow.

Листинг 11. Пароли в /etc/shadow

```
[root@pinguino ~]# echo lpic1111 |passwd jane --stdin
Changing password for user jane.
passwd: all authentication tokens updated successfully.
[root@pinguino ~]# echo lpic1111 |passwd john2 --stdin
Changing password for user john2.
passwd: all authentication tokens updated successfully.
[root@pinguino ~]# grep "^j" /etc/shadow
jane:$1$eGO/KGQY$ZJl.ltYtVw0sv.C50rqUu/:13691:0:99999:7:::
john2:$1$grkxo6ie$J2muvoTpwo3dZAYYTDYNu.:13691:0:180:7:29::
```

Лидирующий `$1$` означает пароль MD5, а salt — это поле переменной длины до 8 символов, заканчивающееся следующим символом `$`. Оставшаяся строка из 22 символов — это зашифрованный пароль.

Средства администрирования пользователей и групп

Вам уже встречалось несколько команд для манипуляций с файлами учетных записей и групп и их теневыми файлами. Сейчас вы узнаете о:

- Администраторах групп

- Командах редактирования файлов паролей и групп
- Программах преобразования

Администраторы групп

При каких-то обстоятельствах у вас может возникнуть желание, чтобы не только root, но и другие пользователи могли администрировать одну или несколько групп, добавляя или удаляя членов группы. В листинге 12 показано, как root может добавить пользователя jane в качестве администратора для группы john2 и затем jane, в свою очередь, может добавить пользователя ian в качестве пользователя.

Листинг 12. Добавление администраторов и членов группы

```
[root@pinguino ~]# gpasswd -A jane john2
[root@pinguino ~]# su - jane
[jane@pinguino ~]$ gpasswd -a ian john2
Adding user ian to group john2
[jane@pinguino ~]$ id ian;id jane
uid=500(ian) gid=500(ian) groups=500(ian),505(john2)
uid=504(jane) gid=504(jane) groups=504(jane)
```

Вы можете с удивлением заметить, что, несмотря на то, что jane является администратором группы john2, она не является ее членом. Исследование структуры файла /etc/gshadow показывает, почему это произошло. Как показано в таблице 8, файл /etc/gshadow содержит четыре поля для каждой записи. Обратите внимание, что третье поле — это разделенный запятыми список администраторов группы.

Таблица 8. Поля файла /etc/gshadow

Поле	**Назначение**
Имя группы (Groupname)	Имя этой группы. Например, john2.
Пароль (Password)	Поле используется для хранения зашифрованного пароля, если у группы имеется пароль. Если группа не имеет пароля, здесь можно увидеть 'x', '!' или '!!'.
Администраторы (Admins)	Разделенный запятыми список администраторов группы.
Члены (Members)	Разделенный запятыми список членов группы.

Как можно заметить, список администраторов и список членов — это два отдельных поля. Опция `-A` команды `gpasswd` позволяет пользователю root добавлять администраторов группы, в то время как опция `-M` позволяет пользователю root добавлять членов. Опция `-a` (заметьте, что используется нижний регистр) позволяет администратору добавлять члена, в то время как опция `-d` позволяет администратору удалять пользователя. Дополнительные опции позволяют удалить пароль группы. Подробнее см. в страницах руководства man.

Команды редактирования файлов паролей и групп

Хотя следующих двух команд нет в списке целей LPI, необходимо знать, что при помощи команды `vipw` можно безопасно редактировать файл /etc/passwd, а при помощи команды `vigr` безопасно редактировать файл /etc/group. Эти команды заблокируют необходимые файлы на то время, пока при помощи редактора `vi` будут производиться изменения. Если вы вносите изменения в файл /etc/passwd, команда `vipw` подскажет, что необходимо проверить, не нужно ли обновить и файл /etc/shadow. Подобным образом, если вы обновляете файл /etc/group при помощи команды `vigr`, вы получите подсказку, что необходимо обновить и файл /etc/gshadow. Если необходимо удалить администраторов группы, необходимо использовать команду `vigr`, поскольку команда `gpasswd` позволяет только добавлять администраторов.

Программы преобразования

Другие четыре другие связанные команды также не перечислены в целях LPI. Это команды `pwconv`, `pwunconv`, `grpconv` и `grpunconv`. Они используются для преобразования файлов теневых паролей и групп в нетеневые и обратно. У вас может никогда не возникнуть необходимости в этих командах, но вы должны знать об их существовании. Подробности см. в страницах руководства man.

Ограниченные учетные записи и учетные записи специального назначения

В соответствии с соглашением, системные пользователи обычно имеют id меньше, чем 100, а пользователь root имеет id, равный 0. Автоматическая нумерация обычных пользователей начинается со значения `UID_MIN`, установленного в файле /etc/login.defs, это значение обычно установлено в 500 или 1000.

Помимо учетных записей обычных пользователей и учетной записи пользователя root, обычно в системе бывает несколько учетных записей специального назначения для демонов, таких как FTP, SSH, mail, news и т.д. В листинге 13 показано несколько записей из файла /etc/passwd для этих учетных записей.

Листинг 13. Ограниченные учетные записи и учетные записи специального назначения

```
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
news:x:9:13:news:/etc/news:
uucp:x:10:14:uucp:/var/spool/uucp:/sbin/nologin
gopher:x:13:30:gopher:/var/gopher:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:99:99:Nobody:/:/sbin/nologin
apache:x:48:48:Apache:/var/www:/sbin/nologin
ntp:x:38:38::/etc/ntp:/sbin/nologin
```

Такие учетные записи часто управляют файлами, но к ним невозможно получить доступ путем обычной регистрации в системе. Поэтому обычно они имеют login shell, определенный как /sbin/nologin или /bin/false, чтобы попытки зарегистрироваться в системе терпели неудачу.

Настройка окружения

Установка переменных окружения и отмена установок

При создании нового пользователя вы обычно устанавливаете множество переменных в соответствии с вашими частными потребностями. Эти переменные обычно устанавливаются в предоставляемых новым пользователям профайлах, таких как .bash_profile и .bashrc, или в общесистемных профайлах /etc/profile и /etc/bashrc. В листинге 14 показан пример, как установить системное приглашение PS1 в /etc/profile на системе Ubuntu 7.04. Первый оператор `if` проверяет, установлена ли переменная PS1, что показывает, что это интерактивный shell, поскольку для неинтерактивного shell приглашение не требуется. Второй оператор `if` проверяет, установлена ли переменная окружения BASH. Если да, устанавливается приглашение и /etc/bash.bashrc (обратите внимание на точку). Если переменная BASH не установлена, проверяется, запущена ли она от имени root (id=0), и устанавливается приглашение # или $ соответственно.

Листинг 14. Установка переменных окружения

```
if [ "$PS1" ]; then
  if [ "$BASH" ]; then
    PS1='\u@\h:\w\$ '
    if [ -f /etc/bash.bashrc ]; then
        . /etc/bash.bashrc
    fi
  else
    if [ "`id -u`" -eq 0 ]; then
      PS1='# '
    else
      PS1='$ '
    fi
  fi
fi
```

В учебном пособии Подготовка к экзамену LPI 102: Командные оболочки, написание скриптов, программирование и компиляция (LPI exam 102 prep: Shells, scripting, programming, and compiling) дается подробная информация о командах, используемых для установки переменных окружения и отмены установок, а также информация о том, как и когда используются различные профайлы.

Настраивая пользовательские окружения, следует учитывать два важных момента:

1. Чтение файла /etc/profile происходит только во время регистрации в системе и не происходит при запуске каждого нового shell'а.
2. Функции и псевдонимы не наследуются новыми shell'ами. Поэтому обычно вы будете устанавливать их и ваши переменные окружения в /etc/bashrc или в собственный профайл пользователя.

Linux Standard Base (LSB) предусматривает, что дополнительные скрипты могут быть расположены не только в системных профайлах /etc/profile и /etc/bashrc, но и в каталоге /etc/profile.d. Эти скрипты служат источником при создании интерактивного login shell. Они обеспечивают удобный способ разделения настроек для различных программ. В листинге 15 показан пример.

Листинг 15. Файл /etc/profile.d/vim.sh из Fedora 7

```
[if [ -n "$BASH_VERSION" -o -n "$KSH_VERSION" -o -n "$ZSH_VERSION" ]; then
  [ -x //usr/bin/id ] || return
  [ `//usr/bin/id -u` -le 100 ] && return
  # for bash and zsh, only if no alias is already set
  alias vi >/dev/null 2>&1 || alias vi=vim
fi
```

Помните, что обычно вы должны `экспортировать` все переменные, установленные в профайле; иначе они не будут доступны командам, запускаемым в новом shell'е.

Поддержка скелетных каталогов для новых учетных записей пользователей

Из раздела Добавление и удаление пользователей и групп вы узнали, как можно создать или наполнить содержимым новый домашний каталог пользователя. Источником для этого нового каталога служит поддерево, корнем которого является /etc/skel. В листинге 16 показаны файлы этого поддерева для системы Fedora 7. Обратите внимание, что большинство файлов начинается с точки, поэтому для их просмотра необходимо использовать опцию `-a`. Опция `-R` рекурсивно выводит подкаталоги, а опция `-L` — соответствующие символьные ссылки.

Листинг 16. Файл /etc/skel из Fedora 7

```
[ian@lyrebird ~]$ ls -aRL /etc/skel
/etc/skel:
.  ..  .bash_logout  .bash_profile  .bashrc  .emacs  .xemacs

/etc/skel/.xemacs:
.  ..  init.el
```

Обратите внимание, что вдобавок к файлам .bash_logout, .bash_profile и .bashrc, которые вы могли ожидать увидеть для Bash shell, этот пример содержит информацию о профайле для редакторов emacs и xemacs. Если вам необходима информация о функциях различных profile-файлов, обратитесь к учебному пособию Подготовка к экзамену LPI 102: Командные оболочки, написание скриптов, программирование и компиляция (LPI exam 102 prep: Shells, scripting, programming, and compiling).

В листинге 17 показан файл /etc/skel/.bashrc для системы Fedora 7. В разных релизах и разных дистрибутивах этот файл может быть различным, но он дает представление о том, какие пользовательские установки по умолчанию можно сделать.

Листинг 17. Файл /etc/skel/.bashrc из Fedora 7

```
# .bashrc

# Source global definitions
if [ -f /etc/bashrc ]; then
        . /etc/bashrc
fi

# User specific aliases and functions
```

Как можно видеть, источником является глобальный /etc/bashrc, затем могут быть добавлены любые специфичные для пользователя инструкции. В листинге 18 показан фрагмент файла /etc/bashrc, в котором скрипты .sh подгружаются из /etc/profile.d.

Листинг 18. Подгрузка скриптов .sh из /etc/profile.d

```
        for i in /etc/profile.d/*.sh; do
                if [ -r "$i" ]; then
                        . $i
        fi
        done
        unset i
```

Обратите внимание, что после выполнения цикла установки для переменной `i` отменены.

Установка путей поиска команды

Ваши профайлы по умолчанию часто содержат переменные `PATH` для частных функций или для продуктов, которые вы могли установить. Можно включить их в скелетные файлы /etc/skel, изменить /etc/profile, /etc/bashrc или создать файл /etc/profile.d, если он используется в вашей системе. Если вы изменяете системные файлы, убедитесь, что ваши изменения сохранятся после любых обновлений системы. В листинге 19 показано, как добавить новый каталог /opt/productxyz/bin в начало или конец существующего `PATH`.

Листинг 19. Добавление каталога в путь поиска

```
PATH="$PATH${PATH:+:}/opt/productxyz/bin"
PATH="/opt/productxyz/bin${PATH:+:}$PATH"
```

Хотя строгого требования не существует, выражение `${PATH:+:}` вставляет в путь разделитель (двоеточие), только если переменная `PATH` не установлена или равна нулю.

Системные журналы

Управление типом и уровнем журналируемой информации

Функция системного журналирования в системе Linux обеспечивает системное журналирование и перехват сообщений ядра. Журналирование может осуществляться на локальной системе или пересылаться на удаленную систему, кроме того, в конфигурационном файле /etc/syslog.conf возможна тонкая регулировка уровня журналирования. Журналирование осуществляется при помощи демона `syslogd`, который обычно получает входную информацию при помощи сокета /dev/log, как показано в листинге 20.

Листинг 20. Сокет /dev/log

```
ian@pinguino:~$ ls -l /dev/log
srw-rw-rw- 1 root root 0 2007-07-05 15:42 /dev/log
```

В случае локального журналирования главным файлом обычно является /var/log/messages, но в большинстве инсталляций используются и многие другие файлы, которые могут быть тщательно настроены. Например, у вас может возникнуть желание выделить сообщения, порождаемые системой электронной почты.

Конфигурационный файл syslog.conf

Файл syslog.conf является главным конфигурационным файлом для демона syslogd. Журналирование базируется на сочетании facility (категория) и priority (приоритет). Существуют следующие категории: auth (или security), authpriv, cron, daemon, ftp, kern, lpr, mail, mark, news, syslog, user, uucp, а также с local0 по local7. Ключевое слово `auth` должно использоваться вместо `security`, а ключевое слово `mark` предназначено для внутреннего использования.

Приоритеты (в порядке возрастания):

1. debug
2. info
3. notice
4. warning (или warn)
5. err (или error)
6. crit
7. alert
8. emerg (или panic)

Ключевые слова, помещенные в скобки (warn, error и panic), сейчас признаны устаревшими.

Правила журналирования определяются записями в syslog.conf. Каждое правило имеет поле селектор и поле действие, которые разделены одним или более пробелами или символами табуляции. Поле селектор устанавливает категории и приоритеты, которые используют правило, а поле действие устанавливает журналируемое действие для категории и приоритетов. По умолчанию выбирается действие для определенного уровня и для всех более высоких уровней, хотя можно ограничить журналирование определенным уровнем. Каждый селектор состоит из категории и приоритета, разделенных точкой. Для данного действия могут быть определены несколько категорий, разделенных запятыми. Для данного действия могут быть определены несколько пар категория/приоритет, разделенных точкой с запятой. В

листинге 21 показан пример несложного syslog.conf.

Листинг 21. Пример syslog.conf

```
# Log all kernel messages to the console.
# Logging much else clutters up the screen.
#kern.*                                                 /dev/console

# Log anything (except mail) of level info or higher.
# Don't log private authentication messages!
*.info;mail.none;authpriv.none;cron.none                /var/log/messages

# The authpriv file has restricted access.
authpriv.*                                              /var/log/secure

# Log all the mail messages in one place.
mail.*                                                  -/var/log/maillog

# Log cron stuff
cron.*                                                  /var/log/cron

# Everybody gets emergency messages
*.emerg                                                 *

# Save news errors of level crit and higher in a special file.
uucp,news.crit                                          /var/log/spooler

# Save boot messages also to boot.log
local7.*                                                /var/log/boot.log
```

Примечания:

- Как и во многих конфигурационных файлах, строки, начинающиеся с #, и пустые строки игнорируются.
- Символ * может использоваться для указания всех категорий или всех приоритетов.
- Специальное ключевое слово none указывает, что журналирование для этой категории не должно быть выполнено для этого действия.
- Дефис перед именем файла (как -/var/log/maillog в этом примере) указывает, что после каждой записи журнал не должен синхронизироваться. В случае аварии системы вы можете потерять информацию, но отключение синхронизации позволит повысить производительность.

В общем, действия упоминаются как "log-файлы", хотя они и не должны действительно быть файлами. В таблице 9 дается описание возможных типов log-файлов.

Таблица 9. Действия в syslog.conf

Поле	**Назначение**
Обычный файл	Задайте полное имя пути, начиная со слеша (/). Поставьте перед ним дефис (-), чтобы отменить синхронизацию файла после каждой записи. Это может привести к потере информации, но повысить производительность
Именованные каналы	Размещение перед именем файла символа канала (|) позволит использовать fifo (first in — first out, первый пришел — первый вышел) или именованный канал (named pipe) в качестве приемника для сообщений. Прежде чем запускать (или перезапускать) `syslogd`, необходимо создать fifo при помощи команды `mkfifo`. Иногда fifo используются для отладки.
Терминал и консоль	Терминал, такой как /dev/console.
Удаленная машина	Чтобы сообщения пересылались на другой хост, поместите перед именем хоста символ (@). Обратите внимание, что сообщения не пересылаются с принимающего хоста.
Список пользователей	Разделенный запятыми список пользователей, получающих сообщения (если пользователь зарегистрирован в системе). Сюда часто включается пользователь root.
Все зарегистрированные пользователи	Чтобы известить всех зарегистрированных пользователей при помощи команды `wall`, используйте символ звездочки (*).

Можно поставить перед приоритетом знак !, чтобы показать, что действие не должно применяться, начиная с этого уровня и выше. Подобным образом, перед приоритетом можно поставить знак =, чтобы показать, что правило применяется только к этому уровню, или !=, чтобы показать, что правило применяется ко всем уровням, кроме этого. В листинге 22 показано несколько примеров, а страница руководства man для syslog.conf содержит множество других примеров.

Листинг 22. Другие примеры syslog.conf

```
# Store all kernel messages in /var/log/kernel.
# Send critical and higher ones to remote host pinguino and to the console
# Finally, Send info, notice and warning messages to /var/log/kernel-info
#
kern.*                      /var/log/kernel
kern.crit                   @pinguino
kern.crit                   /dev/console
kern.info;kern.!err         /var/log/kernel-info

# Store all mail messages except info priority in /var/log/mail.
mail.*;mail.!=info          /var/log/mail
```

Автоматическая ротация и архивирование журналов

При всем многообразии журналов вы должны иметь возможность контролировать их размер. Это делается при помощи команды `logrotate`, которая обычно выполняется демоном cron. Работа демона cron описана в этом пособии ниже в разделе Планирование задач. Главная цель команды `logrotate` состоит в том, чтобы периодически создавать резервные копии журналов и начинать новые журналы. Сохраняется несколько поколений журналов, и, когда завершается срок жизни журнала последнего поколения, он может быть заархивирован. Результат может быть отправлен по почте, например, ответственному за ведение архивов.

Для определения порядка ротации и архивирования журналов используется конфигурационный файл /etc/logrotate.conf. Для разных журналов можно задать разную периодичность, например, ежедневно, еженедельно или ежемесячно, кроме того, можно регулировать количество накапливаемых поколений, а также указать, будут ли копии архивов отправляться ответственному за ведение архивов и, если будут, когда. В листинге 23 показан пример файла /etc/logrotate.conf.

Листинг 23. Пример файла /etc/logrotate.conf

```
# rotate log files weekly
weekly

# keep 4 weeks worth of backlogs
rotate 4

# create new (empty) log files after rotating old ones
create

# uncomment this if you want your log files compressed
#compress

# packages drop log rotation information into this directory
include /etc/logrotate.d

# no packages own wtmp, or btmp -- we'll rotate them here
/var/log/wtmp {
    missingok
    monthly
    create 0664 root utmp
    rotate 1
}

/var/log/btmp {
    missingok
    monthly
    create 0664 root utmp
    rotate 1
}

# system-specific logs may be configured here
```

Глобальные опции размещаются в начале файла logrotate.conf. Они используются по умолчанию, если где-то в другом месте не задано ничего более определенного. В нашем примере ротация журналов происходит еженедельно и резервные копии сохраняются в течение четырех недель. Как только производится ротация журнала, на месте старого журнала автоматически создается новый. Обратите внимание, что файл logrotate.conf может

содержать спецификации из других файлов. Так, в него включаются все файлы из /etc/logrotate.d.

В этом примере также содержатся специальные правила для /var/log/wtmp и /var/log/btmp, ротация которых происходит ежемесячно. Если файлы отсутствуют, сообщение об ошибке не выдается. Создается новый файл и сохраняется только одна резервная копия.

В этом примере по достижении резервной копией последнего поколения она удаляется, поскольку не определено, что следует с ней делать.

Примечание: В файлы /var/log/wtmp и /var/log/btmp записываются удачные и неудачные попытки регистрации в системе соответственно. В отличие от большинства журналов, эти файлы не являются чисто текстовыми. Просмотреть их содержимое можно при помощи команд `last` или `lastb`. Подробную информацию от этих командах см. в их страницах руководства man.

Резервные копии журналов могут также создаваться, когда журналы достигают определенного размера, и могут быть созданы скрипты из наборов команд для выполнения до или после операции резервного копирования. В листинге 24 показан более сложный пример.

Листинг 24. Другой пример конфигурации logrotate

```
/var/log/messages {
    rotate 5
    mail logsave@pinguino
    size 100k
     postrotate
        /usr/bin/killall -HUP syslogd
    endscript
}
```

В этом примере ротация /var/log/messages производится по достижении им размера 100 КБ. Накапливается пять резервных копий, и когда истекает срок жизни самой старой резервной копии, она отсылается по почте на адрес logsave@pinguino. Командное слово `postrotate` включает скрипт, перезапускающий демон syslogd после завершения ротации путем отправки сигнала HUP. Командное слово `endscript` необходимо для завершения скрипта, а также в случае, если имеется скрипт `prerotate`. Более полную информацию см. в страницах руководства man для logrotate.

Изучение и мониторинг журналов с целью выявления повышенной активности

Записи в журналах обычно содержат метку времени, имя хоста, на котором выполняется описываемый процесс, и имя процесса. В листинге 25 показано несколько строк из файла /var/log/messages, содержащих записи для gconfd, ntpd, init и yum.

Листинг 25. Пример записей в журнале

```
Jul  5 15:28:24 lyrebird gconfd (root-2832): Exiting
Jul  5 15:31:06 lyrebird ntpd[2063]: synchronized to 87.98.219.90, stratum 2
Jul  5 15:31:06 lyrebird ntpd[2063]: kernel time sync status change 0001
Jul  5 15:31:24 lyrebird init: Trying to re-exec init
Jul  5 15:31:24 lyrebird yum: Updated: libselinux.i386 2.0.14-2.fc7
Jul  5 15:31:24 lyrebird yum: Updated: libsemanage.i386 2.0.3-4.fc7
Jul  5 15:31:25 lyrebird yum: Updated: cups-libs.i386 1.2.11-2.fc7
Jul  5 15:31:25 lyrebird yum: Updated: libXfont.i386 1.2.9-2.fc7
Jul  5 15:31:27 lyrebird yum: Updated: NetworkManager.i386 0.6.5-7.fc7
Jul  5 15:31:27 lyrebird yum: Updated: NetworkManager-glib.i386 0.6.5-7.fc7
```

Просматривать журналы можно при помощи программы постраничного вывода, например, `less`, искать определенные записи (например, сообщения ядра от хоста lyrebird) можно при помощи команды `grep`, как показано в листинге 26.

Листинг 26. Просмотр журналов

```
[root@lyrebird ~]# less /var/log/messages
[root@lyrebird ~]# grep "lyrebird kernel" /var/log/messages | tail -n 9
Jul  5 15:26:46 lyrebird kernel: Bluetooth: HCI socket layer initialized
Jul  5 15:26:46 lyrebird kernel: Bluetooth: L2CAP ver 2.8
Jul  5 15:26:46 lyrebird kernel: Bluetooth: L2CAP socket layer initialized
Jul  5 15:26:46 lyrebird kernel: Bluetooth: RFCOMM socket layer initialized
Jul  5 15:26:46 lyrebird kernel: Bluetooth: RFCOMM TTY layer initialized
Jul  5 15:26:46 lyrebird kernel: Bluetooth: RFCOMM ver 1.8
Jul  5 15:26:46 lyrebird kernel: Bluetooth: HIDP (Human Interface Emulation) ver 1.2
Jul  5 15:26:59 lyrebird kernel: [drm] Initialized drm 1.1.0 20060810
Jul  5 15:26:59 lyrebird kernel: [drm] Initialized i915 1.6.0 20060119 on minor 0
```

Мониторинг журналов

Время от времени может возникать необходимость мониторинга системных журналов с целью поиска событий. Например, можно попробовать поймать редко случающееся событие в тот момент, когда оно произошло. В таком случае можно использовать команду `tail` с опцией `-f` *для отслеживания* содержимого системного журнала. В листинге 27 показан пример.

Листинг 27. Отслеживание обновлений в системном журнале

```
[root@lyrebird ~]# tail -n 1 -f /var/log/messages
Jul  6 15:16:26 lyrebird syslogd 1.4.2: restart.
Jul  6 15:16:26 lyrebird kernel: klogd 1.4.2, log source = /proc/kmsg started.
Jul  6 15:19:35 lyrebird yum: Updated: samba-common.i386 3.0.25b-2.fc7
Jul  6 15:19:35 lyrebird yum: Updated: procps.i386 3.2.7-14.fc7
Jul  6 15:19:36 lyrebird yum: Updated: samba-client.i386 3.0.25b-2.fc7
Jul  6 15:19:37 lyrebird yum: Updated: libsmbclient.i386 3.0.25b-2.fc7
Jul  6 15:19:46 lyrebird gconfd (ian-3267): Received signal 15, shutting down cleanly
Jul  6 15:19:46 lyrebird gconfd (ian-3267): Exiting
Jul  6 15:19:57 lyrebird yum: Updated: bluez-gnome.i386 0.8-1.fc7
```

Обнаружение в журналах сообщений о проблемах

Выявив проблему, вы захотите записать время, имя хоста и имя процесса, породившего проблему. Если сообщение позволяет точно идентифицировать проблему для ее решения, вы это делаете. Если нет, вам может понадобиться обновить syslog.conf, чтобы указать, какие дополнительные сообщения для соответствующей категории должны быть записаны в системный журнал. Например, у вас может возникнуть необходимость показать информационное сообщение вместо предупредительного или даже отладить уровень сообщения. Приложение может иметь дополнительные категории, которые можно использовать.

Наконец, если вам необходимо поместить в системный журнал пометки, которые помогут узнать, какие сообщения были зажурналированы и на какой стадии находится процесс отладки, можно воспользоваться запускаемой из терминального окна командой `logger` или скриптом shell, чтобы отправить сообщение, содержащее информацию о вашем выборе, демону syslog для журналирования в соответствии с правилами из syslog.conf.

Планирование задач

Из предыдущего раздела вы узнали о команде `logrotate` и увидели, что она должна запускаться через определенные промежутки времени. В следующих двух разделах, касающихся сервисов резервного копирования и сетевой службы времени, вы встретите в ту же необходимость регулярного запуска команд. Это только некоторые из множества задач администрирования, которые должны выполняться многократно и регулярно. Из этого раздела вы узнаете о средствах, используемых для автоматизации периодического планирования задач, а также о средствах, используемых для запуска задач в какое-то определенное время.

Запуск задач через равные промежутки времени

Запуском задач через равные промежутки времени управляет *cron*, состоящий из демона `crond` и набора таблиц, описывающих, какая работа должна быть выполнена и с какой периодичностью. Демон просыпается каждую минуту и проверяет crontab'ы, чтобы определить, что необходимо сделать. Пользователи управляют crontab'ами при помощи программы с`rontab`. Демон `crond` обычно запускается процессом init в момент запуска системы.

Для простоты давайте предположим, что вы хотите регулярно запускать команду, показанную в листинге 28. Фактически эта команда только выдает сообщение о дате и времени, но она иллюстрирует приемы использования `crontab` для настройки заданий для cron, и из ее вывода узнаем, когда запускался cron. Для настройки записей в crontab необходима строка с escape-метасимволами shell, поэтому лучше сделать это при помощи простых команд и параметров, так что в этом примере команда `echo` будет запущена скриптом /home/ian/mycrontab.sh, которому не требуются параметры. Это избавит от необходимости использовать escape-символы.

Листинг 28. Пример несложной команды

```
[ian@lyrebird ~]$ cat mycrontest.sh
#!/bin/bash
 echo "It is now $(date +%T) on $(date +%A)"
[ian@lyrebird ~]$ ./mycrontest.sh
It is now 18:37:42 on Friday
```

Создание crontab

Для создания crontab используется команда `crontab` с опцией `-e` ("edit"). Откроется редактор `vi`, если только вы не задали другой редактор в переменной окружения `EDITOR` или `VISUAL`.

Каждая запись в crontab состоит из шести полей:

1. Минута
2. Час
3. День месяца
4. Месяц года
5. День недели

6. Строка, которая должна быть запущена на исполнение при помощи `sh`

Значения для минут и часов колеблются в диапазоне 0-59 и 0-12 соответственно, для дня месяца и месяца — в диапазоне 1-31 и 1-12 соответственно. День недели может обозначаться в диапазоне 0-6, причем 0 означает воскресенье. День недели также может быть задан как sun, mon, tue и т.д. Шестое поле, в которое входит все, что располагается после пятого поля, — строка, которая передается `sh`. Символ процента (%) используется для обозначения новой строки, поэтому если вы хотите использовать % или другой специальный символ, перед ним надо поставить обратный слеш (\). Строка до первого % передается shell''у, а все строки после % передаются на стандартный ввод.

Некоторые связанные со временем поля могут определяться отдельным значением, диапазоном значений, например, 0-10 или sun-wed, или разделенным запятыми списком отдельных значений и диапазонов. В листинге 29 показана в некоторой степени искусственно созданная запись в crontab для нашего примера.

Листинг 29. Несложный пример crontab

```
0,20,40 22-23 * 7 fri-sat /home/ian/mycrontest.sh
```

В этом примере наша команда исполняется каждую 0-ю, 20-ю и 40-ю минуту (каждые 20 минут) часа между 10 часами вечера и полуночью по пятницам и субботам в течение июля. Подробности о других способах определения времени см. в страницах руководства man для crontab(5).

Как насчет вывода?

Вы можете заинтересоваться, что происходит с выводом команды. Большинство команд, предназначенных для использования совместно с cron, записывают вывод в журнал при помощи функции syslog, о которой вы узнали из предыдущего раздела. Однако любой вывод, направленный на стандартный вывод, будет отправлен пользователю по почте. В листинге 30 показан вывод, который вы могли бы получить от команды из нашего примера.

Листинг 30. Вывод cron, отправленный по почте

```
From ian@lyrebird.raleigh.ibm.com  Fri Jul  6 23:00:02 2007
Date: Fri, 6 Jul 2007 23:00:01 -0400
From: root@lyrebird.raleigh.ibm.com (Cron Daemon)
To: ian@lyrebird.raleigh.ibm.com
Subject: Cron <ian@lyrebird> /home/ian/mycrontest.sh
Content-Type: text/plain; charset=UTF-8
Auto-Submitted: auto-generated
X-Cron-Env: <SHELL=/bin/sh>
X-Cron-Env: <HOME=/home/ian>
X-Cron-Env: <PATH=/usr/bin:/bin>
X-Cron-Env: <LOGNAME=ian>
X-Cron-Env: <USER=ian>

It is now 23:00:01 on Friday
```

Где мой crontab?

Crontab, созданный вами при помощи команды `crontab`, хранится в /etc/spool/cron под именем пользователя, создавшего его. Так показанный выше crontab хранится в /etc/spool/cron/ian. Так что вы не удивитесь, узнав, что команда `crontab`, подобно рассмотренной ранее команде `passwd`, является suid-программой, которая запускается с полномочиями пользователя root.

/etc/crontab

Помимо пользовательских файлов crontab в /var/spool/cron, `cron` также проверяет /etc/crontab и файлы в каталоге /etc/cron.d. Эти системные crontab'ы имеют дополнительное поле между пятым полем для времени (день) и командой. Это дополнительное поле определяет пользователя, для которого будет запущена команда, обычно это root. /etc/crontab может выглядеть как в примере из листинга 31.

Листинг 31. /etc/crontab

```
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
HOME=/

# run-parts
01 * * * * root run-parts /etc/cron.hourly
02 4 * * * root run-parts /etc/cron.daily
22 4 * * 0 root run-parts /etc/cron.weekly
42 4 1 * * root run-parts /etc/cron.monthly
```

В этом примере реальная работа выполняется при помощи команды `run-parts`, которая запускает скрипты из /etc/cron.hourly, /etc/cron.daily и т.д.; /etc/crontab просто контролирует время выполнения повторяющихся заданий. Обратите внимание, что все команды здесь запущены от имени пользователя root. Заметьте также, что crontab может содержать присваивания значений переменным shell, которые будут установлены перед запуском команды.

Anacron

Функция cron хорошо работает в системах, работающих постоянно. В системах, которые могут быть отключены в течение долгого времени, например, в лэптопах, для управления планированием задач, обычно выполняемых ежедневно, еженедельно или ежемесячно, может использоваться команда *anacron* ("anachronistic cron", "анахроничный cron"). Anacron не используется для выполнения задач каждый час.

Anacron хранит метки времени файлов в /var/spool/anacron, чтобы записывать время выполнения заданий. При запуске anacron проверяет, прошло ли необходимое количество дней с тех пор, как задача была выполнена в последний раз, и при необходимости запускает ее. Таблица заданий для anacron хранится в /etc/anacrontab, формат которого немного отличается от формата /etc/crontab. Как и /etc/crontab, /etc/anacrontab может содержать настройки окружения. Каждое задание имеет четыре поля.

1. период
2. задержка
3. идентификатор задачи

4. команда

Период — число дней, но он может быть также определен как @monthly для гарантии, что задача будет выполняться только один раз в месяц независимо от числа дней в месяце. Задержка — число минут ожидания после того как задача должна быть запущена, но до того как она действительно начнет выполняться. Эта возможность может использоваться для предотвращения запуска слишком большого количества задач при первом старте системы. Идентификатор задачи может содержать любой отличный от пробела символ, за исключением слешей (/).

И /etc/crontab, и /etc/anacrontab обновляются путем непосредственного редактирования. Для обновления этих файлов или файлов в каталоге /etc/cron.d команда `crontab` не используется.

Запуск задач в определенное время

Иногда может быть необходимо запустить задание только один раз, а не делать это регулярно. Для этого используется команда `at`. Команды, которые должны быть запущены, берутся из файла, задаваемого при помощи опции `-f`, или со стандартного ввода, если опция `-f` не используется. Опция `-m` посылает пользователю письмо, даже если команда не имеет стандартного вывода. Опция `-v` показывает, в какое время будет запущена задача. Время также показано в выводе. В листинге 32 показаны примеры запуска использовавшегося ранее скрипта `mycrontest.sh`. В листинге 33 показаны выводы, которые возвращаются пользователю после запуска задания. Обратите внимание, что вывод немного более компактен по сравнению с соответствующим выводом задачи cron.

Листинг 32. Использование команды at

```
[ian@lyrebird ~]$ at -f mycrontest.sh -v 10:25
Sat Jul  7 10:25:00 2007

job 5 at Sat Jul  7 10:25:00 2007
```

Листинг 33. Вывод задачи, запущенной at

```
From ian@lyrebird.raleigh.ibm.com  Sat Jul  7 10:25:00 2007
Date: Sat, 7 Jul 2007 10:25:00 -0400
From: Ian Shields <ian@lyrebird.raleigh.ibm.com>
Subject: Output from your job        5
To: ian@lyrebird.raleigh.ibm.com

It is now 10:25:00 on Saturday
```

Спецификации времени могут быть довольно сложными. В листинге 34 показано несколько примеров. Ознакомьтесь со страницами руководства man для `at`, или с файлом /usr/share/doc/at/timespec, или с файлом /usr/share/doc/at-3.1.10/timespec, где 3.1.10 — версия пакета at.

Листинг 34. Варианты задания времени в команде at

```
[ian@lyrebird ~]$ at -f mycrontest.sh  10pm tomorrow
job 14 at Sun Jul  8 22:00:00 2007
[ian@lyrebird ~]$ at -f mycrontest.sh 2:00 tuesday
job 15 at Tue Jul 10 02:00:00 2007
[ian@lyrebird ~]$ at -f mycrontest.sh 2:00 july 11
job 16 at Wed Jul 11 02:00:00 2007
[ian@lyrebird ~]$ at -f mycrontest.sh 2:00 next week
job 17 at Sat Jul 14 02:00:00 2007
```

Команда `at` также имеет опцию `-q`. Увеличение очереди увеличивает для задачи значение nice. Существует также команда `batch`, похожая на команду `at`, за исключением того, что она запускает задачи только при достаточно низкой системной нагрузке. Подробности об этих командах см. в страницах руководств man.

Управление запланированными задачами

Просмотр запланированных задач

Задачами cron и at можно управлять. Как показано в листинге 35, для просмотра crontab используется команда `crontab` с опцией `-l`, для просмотра задач, поставленных в очередь командой `at`, используется команда `atq`.

Листинг 35. Просмотр запланированных задач

```
[ian@lyrebird ~]$ crontab -l
0,20,40 22-23 * 7 fri-sat /home/ian/mycrontest.sh
[ian@lyrebird ~]$ atq
16      Wed Jul 11 02:00:00 2007 a ian
17      Sat Jul 14 02:00:00 2007 a ian
14      Sun Jul  8 22:00:00 2007 a ian
15      Tue Jul 10 02:00:00 2007 a ian
```

Чтобы посмотреть, какая команда запланирована для исполнения при помощи `at`, используется команда `at` с опцией `-c` и номером задания. Вы заметите, что большая часть окружения, которая была активна во время запуска команды `at`, сохранена для запланированного задания. В листинге 36 показана часть вывода для задания 15.

Листинг 36. Использование at -c в сочетании с номером задания

```
#!/bin/sh
# atrun uid=500 gid=500
# mail ian 0
umask 2
HOSTNAME=lyrebird.raleigh.ibm.com; export HOSTNAME
SHELL=/bin/bash; export SHELL
HISTSIZE=1000; export HISTSIZE
SSH_CLIENT=9.67.219.151\ 3210\ 22; export SSH_CLIENT
SSH_TTY=/dev/pts/5; export SSH_TTY
USER=ian; export USER
 ...
HOME=/home/ian; export HOME
LOGNAME=ian; export LOGNAME
 ...
cd /home/ian || {
         echo 'Execution directory inaccessible' >&2
         exit 1
}
${SHELL:-/bin/sh} << `(dd if=/dev/urandom count=200 bs=1 \
   2>/dev/null|LC_ALL=C tr -d -c '[:alnum:]')`

#!/bin/bash
 echo "It is now $(date +%T) on $(date +%A)"
```

Обратите внимание, что содержимое нашего скрипта было скопировано в виде встроенного документа, который будет выполнен shell'ом, установленным в переменной `SHELL`, или /bin/sh, если переменная `SHELL` не установлена. Чтобы узнать о встроенных документах, обратитесь к Учебнику для экзамена LPI 101, Тема 103: Команды GNU и UNIX.

Удаление запланированных задач

Удалить запланированные задачи можно при помощи команды `cron` с опцией `-r`, как показано в листинге 37.

Листинг 37. Просмотр и удаление заданий cron

```
[ian@lyrebird ~]$ crontab -l
0,20,40 22-23 * 7 fri-sat /home/ian/mycrontest.sh
[ian@lyrebird ~]$ crontab -r
[ian@lyrebird ~]$ crontab -l
no crontab for ian
```

Для удаления системных заданий cron или anacron отредактируйте /etc/crontab и /etc/anacrontab или отредактируйте или удалите файлы в каталоге /etc/cron.d.

При помощи команды `atrm` с указанием номера задания можно удалить одно или более заданий, запланированных при помощи команды `at`. Несколько заданий должны быть разделены пробелами. В листинге 38 показан пример.

Листинг 38. Просмотр и удаление заданий при помощи atq и atrm

```
[ian@lyrebird ~]$ atq
16      Wed Jul 11 02:00:00 2007 a ian
17      Sat Jul 14 02:00:00 2007 a ian
14      Sun Jul  8 22:00:00 2007 a ian
15      Tue Jul 10 02:00:00 2007 a ian
[ian@lyrebird ~]$ atrm 16 14 15
[ian@lyrebird ~]$ atq
17      Sat Jul 14 02:00:00 2007 a ian
```

Настройка доступа пользователя к планированию задач

Чтобы не только root, но и все другие пользователи имели возможность воспользоваться `crontab` и функцией cron, они должны быть перечислены в файле /etc/cron.allow, если он существует. Если файл /etc/cron.allow не существует, но существует файл /etc/cron.deny, пользователи, перечисленные в нем, не могут использовать `crontab` и функцию cron. Если ни один из этих файлов не существует, использовать эту команду может только суперпользователь. Если файл /etc/cron.deny пуст, это означает, что все пользователи могут использовать функцию cron. По умолчанию /etc/cron.deny пуст.

Для функции at подобный смысл имеют соответствующие файлы /etc/at.allow и /etc/at.deny.

Резервное копирование данных

Планирование стратегии резервного копирования

Наличие хорошей резервной копии — необходимая часть системного администрирования, но принятие решения о том, что, когда и как должно копироваться, может вызвать затруднения. В сфере бизнеса обычно бывают крайне важны базы данных, такие как заказы клиентов или опись имущества, и часто используются специализированные средства резервного копирования и восстановления данных, описание которых выходит за рамки этого учебного пособия. С другой стороны, некоторые файлы являются временными, и нет необходимости в их резервном копировании. В этом разделе мы фокусируем внимание на системных файлах и пользовательских данных и обсуждаем некоторые принципы, методы и средства резервного копирования таких данных.

Существует три основных метода резервного копирования:

1. *Полное* резервное копирование — обычно создание резервной копии всей файловой системы, каталога или группы связанных файлов. Создание такой копии занимает длительное время и обычно осуществляется в сочетании с одним из следующих двух методов.
2. *Дифференциальное* или *кумулятивное* резервное копирование — резервное копирование всех данных, которые изменились после создания последней полной резервной копии. Для восстановления требуется полная резервная копия плюс самая последняя дифференциальная резервная копия.
3. *Инкрементальное* резервное копирование — резервное копирование только тех данных, которая изменилось после создания последней инкрементальной копии. Для восстановления требуется полная резервная копия плюс все инкрементальные копии (по порядку), созданные после последнего полного резервного копирования.

Что подлежит резервному копированию

Решая, что копировать, следует принять во внимание, насколько меняются данные. Это поможет определить, как часто должны создаваться их резервные копии. Подобным образом, резервные копии наиболее важных данных должны создаваться чаще, чем копии менее важных данных. Вашу операционную систему, вероятно, можно будет относительно легко восстановить, особенно если вы используете общий образ для нескольких систем, хотя важнее было бы создать копии файлов настроек для каждой системы.

Для отдела разработчиков может быть достаточно хранить резервные копии репозиториев, таких как репозитории CVS, в то время как личные песочницы (sandbox'ы) программистов могут быть менее важными. В зависимости от того, насколько важна для вашей деятельности электронная почта, может быть достаточно иметь нечасто создаваемые резервные копии почты или может быть необходимо иметь возможность восстановить почту на самую последнюю возможную дату. У вас может возникнуть желание хранить резервные копии файлов системного cron, но вы не станете волноваться о запланированных задачах отдельных пользователей.

Filesystem Hierarchy Standard предоставляет классификацию данных, которые могут помочь при выборе объектов и методов резервного копирования. Подробнее см. в Учебнике для экзамена LPI 101, Тема 104: Устройства, файловые системы Linux и стандарт Filesystem Hierarchy Standard.

Как только вы решили, что подлежит резервному копированию, необходимо решить, как часто следует делать полную копию и делать ли дифференциальные или инкрементальные резервные копии между созданием полных резервных копий. После принятия этих решений следующие советы помогут в выборе соответствующих инструментов.

Автоматизация резервного копирования

Из предыдущего раздела вы узнали, как планировать задачи, а также о функции cron, которая идеально поможет автоматизировать планирование резервного копирования. Однако резервные копии часто записываются на сменные носители, в основном, на пленку, поэтому, вероятно, будет необходимым вмешательство оператора. Чтобы гарантировать, что процесс резервного копирования происходит автоматически и обладает воспроизводимостью, насколько это возможно, необходимо создать и использовать соответствующие скрипты.

Создание дампов и восстановление содержимого сырых устройств

Один из способов создания полной резервной копии файловой системы — создать образ разделов, на которых она расположена. *Сырое устройство*, например, /dev/hda1 или /dev/sda2, может быть открыто и прочитано как последовательный файл. Точно так же оно может быть записано с резервной копии как последовательный файл. Это не требует со стороны средства резервного копирования знаний относительно расположения файловой системы, но необходимо, чтобы восстановление было сделано в такое место, которое имеет по крайней мере такой размер, как оригинал. Некоторые средства для управления сырыми устройствами *готовы к работе с файловой системой*, что означает, что они понимают одну или более файловых систем Linux. Эти утилиты могут создать дамп сырого устройства, но не могут создать дамп неиспользованной части раздела. Они могут требовать или не требовать для восстановления наличия раздела такого же или большего размера. Команда `dd` — пример утилиты первого типа, а команда `dump` — пример утилиты второго типа, который характерен для файловых систем типа ext2 и ext3.

Команда dd

Самая простая форма использования команды `dd` — копирование входного файла в выходной файл, где любой файл может быть сырым устройством. Для резервного копирования сырого устройства, такого как /dev/hda1 или /dev/sda2, входной файл будет сырым устройством. В идеале для уверенности, что в ходе резервного копирования данные не будут изменены, файловая система не должна быть смонтирована на устройстве или смонтирована только для чтения. В листинге 39 показан пример.

Листинг 39. Резервное копирование разделов при помощи dd

```
[root@lyrebird ~]# dd if=/dev/sda3 of=backup-1
2040255+0 records in
2040255+0 records out
1044610560 bytes (1.0 GB) copied, 49.3103 s, 21.2 MB/s
```

Параметры `if` и `of` определяют входной и выходной файлы соответственно. В этом примере входной файл — сырое устройство dev/sda3, а выходной файл — файл backup-1 в домашнем каталоге пользователя root. Чтобы создать дамп файла для записи его на пленку или на дискету, следует указать что-то типа `of=/dev/fd0` или `of=/dev/st0`.

Обратите внимание, что было скопировано 1,044,610,560 байт данных, и выходной файл имеет очень большой размер, несмотря на то, что фактически используется только около 3% этого конкретного раздела. Вы, наверно, захотите сжать данные, если только вы не используете при копировании на ленту аппаратное сжатие. В листинге 40 показан способ достичь этого, а также вывод команд `ls` и `df`, которые показывают размеры файлов и процент использования файловой системы на /dev/sda3.

Листинг 40. Резервное копирование с сжатием при помощи dd

```
[root@lyrebird ~]# dd if=/dev/sda3 | gzip > backup-2
2040255+0 records in
2040255+0 records out
1044610560 bytes (1.0 GB) copied, 117.723 s, 8.9 MB/s
[root@lyrebird ~]# ls -l backup-[12]
-rw-r--r-- 1 root root 1044610560 2007-07-08 15:17 backup-1
-rw-r--r-- 1 root root  266932272 2007-07-08 15:56 backup-2
[root@lyrebird ~]# df -h /dev/sda3
Filesystem            Size  Used Avail Use% Mounted on
/dev/sda3             972M   28M  944M   3% /grubfile
```

Сжатие при помощи gzip уменьшило размер файла примерно до 20% от его полного размера. Однако неиспользованные блоки могут содержать какие-то данные, поэтому даже сжатая резервная копия может быть значительно больше общего размера содержащихся на разделе данных.

Если разделить размер на количество записей, обработанных `dd`, вы увидите, что `dd` записывает данные блоками размером 512 байт. При копировании на сырое выводное устройство, например, ленту, это может серьезно понизить производительность, поэтому `dd` может читать или записывать данные гораздо более крупными блоками. Укажите опцию `obs` для изменения размера вывода или опцию `ibs` для определения размера выводного блока. Также можно определить только `bs`, чтобы установить одинаковый размер блока для ввода и вывода.

Если для хранения резервной копии необходимо сделать запись на несколько лент или сменных накопителей, копию следует разбить на более мелкие части, например, при помощи утилиты `split`.

Если необходимо пропустить блоки, например, ярлыки дисков или лент, это можно сделать при помощи `dd`. Примеры см. в страницах руководства man.

Помимо простого копирования данных, команда `dd` может преобразовывать данные, например, между ASCII и EBCDIC, между порядками "от старшего к младшему" (big-endian) и "от младшего к старшему" (little-endian) или между записями данных переменной длины и записями данных фиксированной длины. Очевидно, эти преобразования могут быть полезны при копировании реальных файлов, а не сырых устройств. Подробности также см. в страницах руководства man.

Команда dump

Команда `dump` может использоваться для полного, дифференциального или инкрементального резервного копирования на системах ext2 или ext3. В листинге 41 показан пример.

Листинг 41. Резервное копирование с сжатием при помощи dump

```
[root@lyrebird ~]# dump -0 -f backup-4 -j -u /dev/sda3
  DUMP: Date of this level 0 dump: Sun Jul  8 16:47:47 2007
  DUMP: Dumping /dev/sda3 (/grubfile) to backup-4
  DUMP: Label: GRUB
  DUMP: Writing 10 Kilobyte records
  DUMP: Compressing output at compression level 2 (bzlib)
  DUMP: mapping (Pass I) [regular files]
  DUMP: mapping (Pass II) [directories]
  DUMP: estimated 12285 blocks.
  DUMP: Volume 1 started with block 1 at: Sun Jul  8 16:47:48 2007
  DUMP: dumping (Pass III) [directories]
  DUMP: dumping (Pass IV) [regular files]
  DUMP: Closing backup-4
  DUMP: Volume 1 completed at: Sun Jul  8 16:47:57 2007
  DUMP: Volume 1 took 0:00:09
  DUMP: Volume 1 transfer rate: 819 kB/s
  DUMP: Volume 1 12260kB uncompressed, 7377kB compressed, 1.662:1
  DUMP: 12260 blocks (11.97MB) on 1 volume(s)
  DUMP: finished in 9 seconds, throughput 1362 kBytes/sec
  DUMP: Date of this level 0 dump: Sun Jul  8 16:47:47 2007
  DUMP: Date this dump completed:  Sun Jul  8 16:47:57 2007
  DUMP: Average transfer rate: 819 kB/s
  DUMP: Wrote 12260kB uncompressed, 7377kB compressed, 1.662:1
  DUMP: DUMP IS DONE
[root@lyrebird ~]# ls -l backup-[2-4]
-rw-r--r-- 1 root root 266932272 2007-07-08 15:56 backup-2
-rw-r--r-- 1 root root 266932272 2007-07-08 15:44 backup-3
-rw-r--r-- 1 root root   7554939 2007-07-08 16:47 backup-4
```

В этом примере `-0` определяет *уровень дампа*, выражающийся целым числом, исторически сложилось, что используется значение от 0 до 9, где 0 обозначает полный дамп. Опция `-f` определяет выходной файл, который может быть сырым устройством. Укажите `-`, чтобы направить вывод на стандартный вывод. Опция `-j` определяет уровень сжатия по умолчанию, равный 2, с использованием сжатия bzlib. Если вы предпочитаете сжатие zlib, используйте опцию `-z`. Опция `-u` указывает, что запись информации о дампе, обычно это /etc/dumpdates, должна быть обновлена. Все параметры, стоящие после опций, — файл или список файлов, причем файл также может быть сырым устройством, как в этом примере. Обратите внимание, насколько резервная копия меньше в случае, если программа резервного копирования осведомлена о структуре файловой системы и может не сохранять неиспользуемые блоки устройства.

Если выводом является такое устройство как лента, когда его объем будет полностью использован, команда dump запросит другой том. Также можно предусмотреть несколько имен файлов, разделенных запятыми. Например, если вам нужно, чтобы автоматически был создан дамп, которому требуется две ленты, вы можете вставить ленты в /dev/st0 и /dev/st1, запланировать команду dump, указав обе ленты в качестве вывода, и отправиться домой спать.

Если определить уровень дампа выше 0, будет создан инкрементальный дамп из всех новых файлов и файлов, изменившихся с момента создания последнего дампа уровня меньшего, чем данный. Поэтому дамп уровня 1 будет дифференциальным, даже если одновременно был получен дамп уровня 2 или выше. В листинге 42 показан результат обновления метки времени существующего файла на /dev/sda3 и создания нового файла, а затем сделан дамп уровня 2. После этого создан другой новый файл и сделан дамп уровня 1. Также показана информация

из /etc/dumpdates. Для краткости часть вывода второго дампа опущена.

Листинг 42. Резервное копирование с сжатием при помощи dump

```
[root@lyrebird ~]# dump -2 -f backup-5 -j -u /dev/sda3
  DUMP: Date of this level 2 dump: Sun Jul  8 16:55:46 2007
  DUMP: Date of last level 0 dump: Sun Jul  8 16:47:47 2007
  DUMP: Dumping /dev/sda3 (/grubfile) to backup-5
  DUMP: Label: GRUB
  DUMP: Writing 10 Kilobyte records
  DUMP: Compressing output at compression level 2 (bzlib)
  DUMP: mapping (Pass I) [regular files]
  DUMP: mapping (Pass II) [directories]
  DUMP: estimated 91 blocks.
  DUMP: Volume 1 started with block 1 at: Sun Jul  8 16:55:47 2007
  DUMP: dumping (Pass III) [directories]
  DUMP: dumping (Pass IV) [regular files]
  DUMP: Closing backup-5
  DUMP: Volume 1 completed at: Sun Jul  8 16:55:47 2007
  DUMP: 90 blocks (0.09MB) on 1 volume(s)
  DUMP: finished in less than a second
  DUMP: Date of this level 2 dump: Sun Jul  8 16:55:46 2007
  DUMP: Date this dump completed:  Sun Jul  8 16:55:47 2007
  DUMP: Average transfer rate: 0 kB/s
  DUMP: Wrote 90kB uncompressed, 15kB compressed, 6.000:1
  DUMP: DUMP IS DONE
[root@lyrebird ~]# echo "This data is even newer" >/grubfile/newerfile
[root@lyrebird ~]# dump -1 -f backup-6 -j -u -A backup-6-toc /dev/sda3
  DUMP: Date of this level 1 dump: Sun Jul  8 17:08:18 2007
  DUMP: Date of last level 0 dump: Sun Jul  8 16:47:47 2007
  DUMP: Dumping /dev/sda3 (/grubfile) to backup-6
    ...
  DUMP: Wrote 100kB uncompressed, 16kB compressed, 6.250:1
  DUMP: Archiving dump to backup-6-toc
  DUMP: DUMP IS DONE
[root@lyrebird ~]# ls -l backup-[4-6]
-rw-r--r-- 1 root root 7554939 2007-07-08 16:47 backup-4
-rw-r--r-- 1 root root   16198 2007-07-08 16:55 backup-5
-rw-r--r-- 1 root root   16560 2007-07-08 17:08 backup-6
[root@lyrebird ~]# cat /etc/dumpdates
/dev/sda3 0 Sun Jul  8 16:47:47 2007 -0400
/dev/sda3 2 Sun Jul  8 16:55:46 2007 -0400
/dev/sda3 1 Sun Jul  8 17:08:18 2007 -0400
```

Обратите внимание, что backup-6 на самом деле больше, чем backup 5. Дамп уровня 1 иллюстрирует использование опции `-A` для создания таблицы содержимого, которая может использоваться, чтобы определить, находится ли файл в архиве, без необходимости действительно монтировать архив. Это особенно полезно при использовании лент или других сменных архивных накопителей. Вы снова увидите эти примеры позже в этом разделе, когда мы будем обсуждать восстановление данных.

Команда `dump` может создавать файлы или подкаталоги дампа, но не может обновить /etc/dumpdates, и поддерживается только уровень дампа 0, то есть полный дамп.

Листинг 43 иллюстрирует процесс формирования и записи на дискету дампа каталога /usr/include/bits и его содержимого при помощи команды `dump`. В этом случае дамп не помещается на одну дискету, поэтому требуется новый том. Запрос и ответ выделены жирным шрифтом.

Листинг 43. Резервное копирование каталога в несколько томов при помощи команды dump

```
[root@lyrebird ~]# dump -0 -f /dev/fd0 /usr/include/bits
  DUMP: Date of this level 0 dump: Mon Jul  9 16:03:23 2007
  DUMP: Dumping /dev/sdb9 (/ (dir usr/include/bits)) to /dev/fd0
  DUMP: Label: /
  DUMP: Writing 10 Kilobyte records
  DUMP: mapping (Pass I) [regular files]
  DUMP: mapping (Pass II) [directories]
  DUMP: estimated 2790 blocks.
  DUMP: Volume 1 started with block 1 at: Mon Jul  9 16:03:30 2007
  DUMP: dumping (Pass III) [directories]
  DUMP: End of tape detected
  DUMP: Closing /dev/fd0
  DUMP: Volume 1 completed at: Mon Jul  9 16:04:49 2007
  DUMP: Volume 1 1470 blocks (1.44MB)
  DUMP: Volume 1 took 0:01:19
  DUMP: Volume 1 transfer rate: 18 kB/s
  DUMP: Change Volumes: Mount volume #2
  DUMP: Is the new volume mounted and ready to go?: ("yes" or "no") y
  DUMP: Volume 2 started with block 1441 at: Mon Jul  9 16:05:10 2007
  DUMP: Volume 2 begins with blocks from inode 2
  DUMP: dumping (Pass IV) [regular files]
  DUMP: Closing /dev/fd0
  DUMP: Volume 2 completed at: Mon Jul  9 16:06:28 2007
  DUMP: Volume 2 1410 blocks (1.38MB)
  DUMP: Volume 2 took 0:01:18
  DUMP: Volume 2 transfer rate: 18 kB/s
  DUMP: 2850 blocks (2.78MB) on 2 volume(s)
  DUMP: finished in 109 seconds, throughput 26 kBytes/sec
  DUMP: Date of this level 0 dump: Mon Jul  9 16:03:23 2007
  DUMP: Date this dump completed:  Mon Jul  9 16:06:28 2007
  DUMP: Average transfer rate: 18 kB/s
  DUMP: DUMP IS DONE
```

Если резервная копия записывается на ленту, следует помнить, что обычно лента перематывается после каждого использования. Устройство с именем типа /dev/st0 или /dev/st1 перематывается автоматически. Соответствующие неперематываемые эквивалентные устройства — /dev/nst0 и /dev/nst1. В любом случае всегда можно воспользоваться командой `mt` для выполнения таких операций с магнитной лентой как проматывание файлов и записей, перемотка и запись отметок конца файла (EOF marks). Дополнительную информацию см. в страницах руководств man для `mt` и `st`.

Разумный выбор уровней дампов позволит минимизировать количество архивов, необходимых для восстановления к любому определенному уровню. Пример стратегии на основе головоломки Ханойская башня см. в страницах руководства man для `dump`.

Эти команды, как и команда `dd`, имеют большое количество опций, не описанных в этом кратком введении. Подробнее см. в страницах руководства man.

Частичное и ручное резервное копирование

До сих пор вы знакомились с инструментами, которые хорошо работают резервном копировании файловых систем целиком. Иногда бывает необходима резервная копия не всей

файловой системы, а отдельных файлов или подкаталогов. Например, может понадобиться создавать резервную копию большей части файловой системы еженедельно, а резервную копию файлов электронной почты — ежедневно. Для этих целей обычно используются другие две команды, `cpio` и `tar`. Обе они могут записывать архивы в файлы или на устройства, например, на ленты и дискеты, и обе могут восстанавливать данные из таких архивов. Сейчас из этих двух команд чаще используется `tar`, возможно потому, что она лучше работает с полными каталогами, и GNU-версия tar поддерживает сжатие и с помощью gzip, и с помощью bzip.

Использование cpio

Команда `cpio` работает для создания архива в режиме *copy-out*, для восстановления архива — в режиме *copy-in*, для копирования набора файлов из одного места в другое — в режиме *copy-pass*. В режиме copy-out используются опции `-o` или `--create`, в режиме copy-in — опции `-i` или `--extract` и в режиме copy-pass — опции `-p` или `--pass-through`. Вводом является список файлов, получаемый на стандартный ввод. Вывод происходит на стандартный вывод, или на устройство, или в определенный файл при помощи опции `-f` или `--file`.

В листинге 44 показано, как сгенерировать список файлов при помощи команды `find`. Обратите внимание, что команда `find` с опцией `-print0` используется для создания строк, оканчивающихся на ноль (null-terminate), для имен файлов и команда `cpio` с опцией `--null` — для чтения этого формата. Это позволяет правильно оперировать именами файлов, имеющими пробелы или символы перевода строки.

Листинг 44. Резервное копирование домашнего каталога при помощи cpio

```
[root@lyrebird ~]#  find ~ian -depth -print0 | cpio --null -o >backup-cpio-1
18855 blocks
```

Чтобы видеть список файлов по мере их архивирования, добавьте к `cpio` опцию `-v`.

Как и при использовании других команд, способных архивировать файлы, можно задать размер блока. Подробнее об этих и других опциях см. в страницах руководства man.

Использование tar

Команда `tar` (название происходит от *Tape ARchive*) создает архивный файл, или *tarfile*, или *tarball* из набора входных файлов или каталогов; также она восстанавливает файлы из таких архивов. Если в качестве ввода для `tar` используется каталог, все файлы и подкаталоги включаются автоматически, что делает `tar` очень удобным для архивирования поддеревьев структуры каталогов.

Как и для других команд, которые мы обсуждали, вывод может быть направлен в файл, на устройство, такое как лента или дискета, или на стандартный вывод. Местоположение вывода определяется при помощи опции `-f`. Другие наиболее часто используемые опции — это `-c` для создания архива, `-x` для разархивирования, `-v` для подробного вывода, содержащего список обрабатываемых файлов, `-z` для сжатия с использованием gzip и `-j` для сжатия с использованием bzip2. Большинство опций команды `tar` имеет короткую форму, при которой используется один дефис, и длинную форму, при которой используется пара дефисов. Описание длинных форм и других опций см. в страницах руководства man.

В листинге 45 показано, как создать резервную копию системных заданий cron при помощи

`tar`.

Листинг 45. Резервное копирование системных заданий cron при помощи tar

```
[root@lyrebird ~]# tar -czvf backup-tar-1 /etc/*crontab /etc/cron.d
tar: Removing leading `/' from member names
/etc/anacrontab
/etc/crontab
/etc/cron.d/
/etc/cron.d/sa-update
/etc/cron.d/smolt
```

В первой строке вывода указано, что `tar` удалит лидирующий слеш (\) из имен членов. Это позволяет восстановить файлы в какое-то другое место для проверки, прежде чем заменить системные файлы. Это хорошая идея, которая позволит при создании архивов избежать перемешивания абсолютных и относительных имен файлов, поскольку при восстановлении из архива все имена будут относительными.

Команда `tar` при помощи опции `-r` или `--append` может добавить в архив дополнительные файлы. Это может привести к тому, что в архиве будет несколько копий файла. В таком случае в ходе операции восстановления будет восстановлен только *последний файл*. Для выбора одного из нескольких файлов используется опция `--occurrence`. Если архив находится не на ленте, а на обычной файловой системе, для обновления архива используется опция `-u` или `--update`. Это работает подобно дополнению архива, за исключением того, что метки времени для файлов в архиве сравниваются с метками в файловой системе и добавляются только те файлы, которые изменились после создания заархивированной версии. Как было упомянуто, это не работает с архивами на лентах.

Как и другие изучаемые здесь команды, команда `tar` имеет множество опций, не описанных в этом кратком введении. Подробнее см. в страницах руководств man или info.

Целостность файла резервной копии

Целостность файла резервной копии чрезвычайно важна. Если резервная копия испорчена, нет смысла хранить ее. Хорошая стратегия резервного копирования также подразумевает проверку резервных копий.

Первый шаг к обеспечению целостности резервной копии — убедиться, что данные, для которых делается резервная копия, собраны правильно. Для того чтобы данные, для которых создается резервная копия, не изменились в процессе копирования, обычно бывает достаточно, чтобы система не была смонтирована или была смонтирована только для чтения. Если вам необходимо создать резервную копию файловых систем, каталогов или файлов, которые меняются в то время как создается резервная копия, следует убедиться, что не было сделано изменений в ходе резервного копирования. Если изменения были, необходимо избрать стратегию для их сбора, или повторив резервное копирование, или, возможно, заменив такие файлы внутри резервной копии. Разумеется, это затрагивает и процедуру восстановления.

Допустим, вы получили хорошие резервные копии, их необходимо периодически проверять. Один из способов состоит в том, чтобы восстановить резервную копию на запасной том и

убедиться, что результат совпадает с тем, что было скопировано. Это самое простое, что следует сделать, прежде чем позволить обновить файловые системы данными резервной копии. Если резервная копия сохранена на медиа-носитель, такой как CD или DVD, можно использовать команду `diff` как часть процедуры резервного копирования, чтобы убедиться в качестве резервной копии. Помните, что даже качественные резервные копии при хранении могут портиться, поэтому, даже если они проверялись во время резервного копирования, их следует периодически проверять. Хранение дайджестов используемых программ, таких как `md5sum` или `sha1sum` — также хороший способ проверки целостности файла с резервной копией.

Восстановление файловых систем из резервных копий

Резервное копирование файлов дает возможность восстановить их при необходимости. Иногда может возникнуть желание восстановить всю файловую систему, но гораздо чаще необходимо восстанавливать только определенные файлы или, возможно, набор каталогов. Почти всегда, прежде чем действительно сделать восстановленные файлы реальными, данные восстанавливаются в какое-то временное место и производится проверка, действительно ли восстановлено то, что нужно, и совместимы ли восстановленные данные с текущим состоянием системы.

Родственная проблема — необходимость убедиться, что нужные элементы находятся в определенной резервной копии, поскольку часто возникает необходимость получить доступ к версии файла, который был изменен, или, возможно, удален "когда-то на прошлой или позапрошлой неделе". Имея в виду все вышесказанное, давайте рассмотрим опции восстановления.

Восстановление dd-архива

Вспомните, что команда `dd` не распознает файловую систему, поэтому, чтобы узнать, что находится в дампе раздела, необходимо восстановить его. В листинге 46 показано, как раздел, из которого ранее в листинге 39 был создан дамп, восстановить на раздел /dev/sdc7, специально созданный на сменном USB-устройстве только с этой целью.

Листинг 46. Восстановление раздела при помощи dd

```
[root@lyrebird ~]# dd if=backup-1 of=/dev/sdc7
2040255+0 records in
2040255+0 records out
1044610560 bytes (1.0 GB) copied, 44.0084 s, 23.7 MB/s
```

Вспомните, что после того как была получена эта резервная копия, мы добавили к файловой системе на /dev/sda3 некоторые файлы. Вы увидите, что это действительно так, если подмонтируете недавно восстановленный раздел и сравните его с оригиналом, как показано в листинге. Обратите внимание, что файл, метка времени которого была обновлена при помощи `touch`, здесь не показан, как следовало бы ожидать.

Листинг 47. Сравнение восстановленного раздела с текущим state

```
[root@lyrebird ~]# mount /dev/sdc7 /mnt/temp-dd/
[root@lyrebird ~]# diff -rq /grubfile/ /mnt/temp-dd/
Only in /grubfile/: newerfile
Only in /grubfile/: newfile
```

Восстановление dump-архива при помощи restore

Вспомните, что в последний раз мы использовали `dump` для дифференциального резервного копирования и что мы создали таблицу содержимого. В листинге 48 показано, как, используя сам архив (backup-5) или таблицу содержимого (backup-6-toc), воспользоваться командой `restore` для проверки файлов из архива, созданного при помощи `dump`.

Листинг 48. Проверка содержимого архивов

```
[root@lyrebird ~]# restore -t -f backup-5
Dump tape is compressed.
Dump   date: Sun Jul  8 16:55:46 2007
Dumped from: Sun Jul  8 16:47:47 2007
Level 2 dump of /grubfile on lyrebird.raleigh.ibm.com:/dev/sda3
Label: GRUB
         2      .
    100481      ./ibshome
    100482      ./ibshome/index.html
        16      ./newfile
[root@lyrebird ~]# restore -t -A backup-6-toc
Dump   date: Sun Jul  8 17:08:18 2007
Dumped from: Sun Jul  8 16:47:47 2007
Level 1 dump of /grubfile on lyrebird.raleigh.ibm.com:/dev/sda3
Label: GRUB
Starting inode numbers by volume:
        Volume 1: 2
         2      .
    100481      ./ibshome
    100482      ./ibshome/index.html
        16      ./newfile
        17      ./newerfile
```

Команда `restore` также может при помощи опции `-C` сравнить содержимое архива с содержимым файловой системы. В листинге 49 мы обновили newerfile и затем сравнили резервную копию с файловой системой.

Листинг 49. Сравнение архива с файловой системой при помощи restore

```
[root@lyrebird ~]# echo "something different" >/grubfile/newerfile
[root@lyrebird ~]# restore -C -f backup-6
Dump tape is compressed.
Dump   date: Sun Jul  8 17:08:18 2007
Dumped from: Sun Jul  8 16:47:47 2007
Level 1 dump of /grubfile on lyrebird.raleigh.ibm.com:/dev/sda3
Label: GRUB
filesys = /grubfile
./newerfile: size has changed.
Some files were modified!  1 compare errors
```

Восстановление при помощи команды `restore` может производиться интерактивно или автоматически. В листинге 50 показано, как восстановить newerfile в домашний каталог пользователя root (так что при необходимости его можно проверить, прежде чем заменить обновленный файл), а затем заменить обновленный файл резервной копией. Этот пример иллюстрирует интерактивное восстановление.

Листинг 50. Восстановление файла при помощи restore

```
[root@lyrebird ~]# restore -i -f backup-6
Dump tape is compressed.
restore > ?
Available commands are:
        ls [arg] - list directory
        cd arg - change directory
        pwd - print current directory
        add [arg] - add `arg' to list of files to be extracted
        delete [arg] - delete `arg' from list of files to be extracted
        extract - extract requested files
        setmodes - set modes of requested directories
        quit - immediately exit program
        what - list dump header information
        verbose - toggle verbose flag (useful with ``ls'')
        prompt - toggle the prompt display
        help or `?' - print this list
If no `arg' is supplied, the current directory is used
restore > ls new*
newerfile
newfile
restore > add newerfile
restore > extract
You have not read any volumes yet.
Unless you know which volume your file(s) are on you should start
with the last volume and work towards the first.
Specify next volume # (none if no more volumes): 1
set owner/mode for '.'? [yn] y
restore > q
[root@lyrebird ~]# mv -f newerfile /grubfile
```

Восстановление архива cpio

Команда `cpio` в режиме copy-in (опция `-i` или `--extract`) может вывести список содержимого архива или восстановить избранные файлы. Использование опции `--absolute-filenames` при перечислении файлов уменьшит количество ненужных

сообщений, которые в противном случае выдаст `cpio`, поскольку эта опция отбросит все лидирующие символы / от каждого имени пути, имеющего / в начале. В листинге 51 показан частичный вывод листинга нашего предыдущего архива.

Листинг 51. Восстановление избранных файлов при помощи cpio

```
[root@lyrebird ~]# cpio  -id --list --absolute-filenames <backup-cpio-1
/home/ian/.gstreamer-0.10/registry.i686.xml
/home/ian/.gstreamer-0.10
/home/ian/.Trash/gnome-terminal.desktop
/home/ian/.Trash
/home/ian/.bash_profile
```

В листинге 52 показано, как восстановить все файлы, содержащие в имени пути или имени файла слово "samp". Вывод пропущен через команду `uniq`, чтобы уменьшить количество сообщений вида "Removing leading '/' ...". Для создания каталога следует использовать опцию `-d`; в противном случае все файлы будут созданы в текущем каталоге. Кроме того, `cpio` не заменит на файловой системе никакие более новые файлы на архивные копии, если только не использовалась опция `-u` или `--unconditional`.

Листинг 52. Восстановление избранных файлов при помощи cpio

```
[root@lyrebird ~]# cpio  -ivd "*samp*" < backup-cpio-1 2>&1 |uniq
cpio: Removing leading `/' from member names
home/ian/crontab.samp
cpio: Removing leading `/' from member names
home/ian/sample.file
cpio: Removing leading `/' from member names
18855 blocks
```

Восстановление архива tar

Команда `tar` также может сравнивать архивы с текущей файловой системой и восстанавливать файлы из архивов. Для выполнения сравнения используются опции `-d`, `--compare`, или `--diff`. Вывод покажет файлы, содержимое которых отличается, а также файлы, у которых отличаются метки времени. В листинге 53 показан расширенный вывод (использована опция `-v`), полученный в результате сравнения ранее созданного файла и файлов в /etc после того как с целью изменить метку времени был затронут файл /etc/crontab. Опция `--directory /` дает команде `tar` указание выполнить сравнение, начиная не с текущего, а с корневого каталога.

Листинг 53. Сравнение архивов и файлов при помощи tar

```
[root@lyrebird ~]# touch /etc/crontab
[root@lyrebird ~]# tar --diff -vf backup-tar-1 --directory /
etc/anacrontab
etc/crontab
etc/crontab: Mod time differs
etc/cron.d/
etc/cron.d/sa-update
etc/cron.d/smolt
```

В листинге 54 показано, как извлечь из текущего каталога только /etc/crontab и /etc/anacrontab.

Листинг 54. Извлечение файлов из архива при помощи tar

```
[root@lyrebird ~]# tar -xzvf backup-tar-1 "*tab"
etc/anacrontab
etc/crontab
```

Обратите внимание, что `tar`, в отличие от `cpio`, автоматически создает иерархию каталогов.

Системное время

Установка системной даты и времени

В системе Linux системное время является крайне важным. Ранее вы видели, что cron и anacron выполняют действия в определенное время, поэтому для корректной работы им необходимо точное время. Большинство средств для резервного копирования и восстановления, которые обсуждались в предыдущем разделе, наряду со средствами разработки, такими как `make`, также зависят от надежного измерения времени. Большинство компьютеров, собранных примерно с 1980 года, включают некий набор механизмов часов и большинство созданных, начиная с 1984 года, имеет стабильный механизм часов, который поддерживает ход часов, даже если компьютер выключен.

Если вы проводили установку системы Linux в графическом режиме, вы, вероятно, настроили время и выбрали нужный часовой пояс. Можно выбрать, использовать ли для настройки часов Network Time Protocol (NTP), а также выбрать или не выбирать возможность поддержки системных часов с использованием Coordinated Universal Time (UTC).

Независимо от того, живете ли вы в Нью Йорке, Бухаресте, Находке, Улан-Баторе, Бангкоке или Канберре, в Linux большая часть вычислений, связанных со временем, привязана к Coordinated Universal Time (UTC). Если вы запускаете только систему Linux, принято настраивать аппаратные часы в соответствии с UTC, но если вы загружаете также и другую операционную систему, такую как Windows, может понадобиться настроить аппаратные часы на местное время. Поскольку рассматривается Linux, это не имеет значения, за исключением случаев, когда внутри Linux оказываются два различных метода хранения записи о часовых поясах и, если они не совпадают, можно уладить несоответствие, например, при помощи нескольких дополнительных меток времени в файловых системах FAT. В листинге 55 показано, как использовать команду `date` для просмотра текущей даты и времени. Даже если аппаратные часы поддерживают время UTC, всегда отображается локальное время.

Листинг 55. Отображение текущей даты и времени

```
[root@lyrebird ~]# date;date -u
Mon Jul  9 22:40:01 EDT 2007
```

Команда `date` поддерживает большое разнообразие возможных форматов вывода, некоторые из которых вы уже видели в листинге 28. Если вы хотите больше узнать о различных форматах данных, обратитесь к странице man для команды `date`.

Если необходимо настроить дату, сделать это можно, передав дату и время в качестве аргумента. Требуемый формат сложился автоматически и в некоторой степени необычен даже для американцев и действительно необычен для остальной части мира. Необходимо указать как минимум месяц, день, час и минуту в формате MMDDhhmm, можно также добавить год в виде двух- или четырехзначного числа (CCYY или YY) и при желании точку и за ней двухзначное число для секунд. В листинге 56 показан пример, в котором системная дата изменена чуть больше чем на минуту.

Листинг 56. Настройка системной даты и времени

```
[root@lyrebird ~]# date; date 0709221407;date
Mon Jul  9 23:12:37 EDT 2007
Mon Jul  9 22:14:00 EDT 2007
Mon Jul  9 22:14:00 EDT 2007
```

Настройка часов BIOS на временную зону UTC

Система Linux, как и большинство других современных операционных систем, фактически имеет двое часов. Первые часы — аппаратные, иногда называемые Real Time Clock, сокращенно (RTC), или часы BIOS, обычно они связаны с колеблющимся кварцевым кристаллом, имеющим точность хода до нескольких секунд в день. Точность зависит от различных колебаний, например, окружающей температуры. Вторые часы — внутренние программные часы, которые идут непрерывно, в том числе и при перерывах в работе системы. Они подвержены отклонениям, связанным с большой системной нагрузкой и задержкой прерываний. Однако система обычно считывает показания аппаратных часов при загрузке и потом использует системные часы. Команда `date`, о которой вы только что узнали, устанавливает не аппаратные, а системные часы.

Если используется NTP, можно установить аппаратные часы в ходе первой инсталляции системы и больше никогда не беспокоиться о них. Если нет, эта часть учебного пособия покажет, как просмотреть и установить время на аппаратных часах.

Для просмотра текущих показаний аппаратных часов можно воспользоваться командой `hwclock`. В листинге 57 показаны текущие показания обоих часов, и системных, и аппаратных.

Листинг 57. Показания системных и аппаратных часов

```
[root@lyrebird ~]# date;hwclock
Mon Jul  9 22:16:11 EDT 2007
Mon 09 Jul 2007 11:14:49 PM EDT  -0.071616 seconds
```

Обратите внимание, что значения различаются. Можно синхронизировать аппаратные часы с системными при помощи команды `hwclock` с опцией `-w` или `--systohc` и синхронизировать системные часы с аппаратными при помощи команды `hwclock` с опцией `-s` или `--hctosys`, как показано в листинге 58.

Листинг 58. Настройка соответствия системных часов в аппаратным

```
[root@lyrebird ~]# date;hwclock;hwclock -s;date
Mon Jul  9 22:20:23 EDT 2007
Mon 09 Jul 2007 11:19:01 PM EDT  -0.414881 seconds
Mon Jul  9 23:19:02 EDT 2007
```

Можно указать опции `--utc` или `--localtime`, чтобы системные часы поддерживали UTC или местное время. Если значение не указано, оно берется из третьей строки файла /etc/adjtime.

Ядро Linux имеет режим, при котором каждые 11 минут системное время копируется в аппаратные часы. По умолчанию эта функция отключена, но она включается NTP. Запуск какой-либо команды, которая устанавливает время устаревшим способом, например, `hwclock --hctosys`, отключает ее, поэтому, если используется NTP, хорошей идеей будет просто позволить NTP выполнить эту работу. Чтобы узнать, как проверить, обновляются ли часы каждые 11 минут или нет, обратитесь к странице man команды `adjtimex`. Может возникнуть необходимость установить пакет adjtimex, если он не установлен по умолчанию.

Команда `hwclock` сохраняет изменения, сделанные в аппаратных часах, для того чтобы периодически компенсировать погрешность часов. Необходимые данные хранятся в /etc/adjtime, который является ASCII-файлом. Если NTP не используется, для компенсации отклонений часов можно использовать команду `adjtimex`. В противном случае аппаратные часы будут регулироваться приблизительно каждые 11 минут при помощи NTP. Кроме того, эта команда показывает, используют аппаратные часы местное время или время UTC, первое значение в /etc/adjtime показывает величину отклонения аппаратных часов за день (в секундах). В листинге 59 показаны два примера.

Листинг 59. /etc/adjtime показывает отклонение часов и какое время они показывают, локальное или UTC

```
[root@lyrebird ~]# cat /etc/adjtime
0.000990 1184019960 0.000000
1184019960
LOCAL
root@pinguino:~# cat /etc/adjtime
-0.003247 1182889954 0.000000
1182889954
LOCAL
```

Обратите внимание, что в обеих этих системах на аппаратных часах используется местное время, но отклонения часов отличаются — 0.000990 на lyrebird и -0.003247 на pinguino.

Настройка часового пояса

Часовой пояс — критерий того, насколько местное время отличается от UTC. Информация о доступных часовых поясах, которые можно настроить, хранится в /usr/share/zoneinfo. По традиции /tec/localtime является линком на один из файлов часового пояса из этого дерева каталогов, например, /usr/share/zoneinfo/Eire или /usr/share/zoneinfo/Australia/Hobart. В современных системах с большой долей вероятности это будет копия соответствующего файла часового пояса, поскольку когда в процессе загрузки нужна информация о местном часовом поясе, файловая система /usr/share не может быть смонтирована.

Подобным образом другой файл, /etc/timezone, традиционно является линком /etc/default/init и используется для установки переменной окружения для часового пояса TZ и нескольких связанных с местоположением переменных окружения. Файл в системе может существовать, а может нет. Если он существует, он может содержать просто имя текущего часового пояса. Информацию о часовом поясе можно также найти в /etc/sysconfig/clock. В листинге 60 показаны эти файлы, взятые из систем Ubuntu 7.04 и Fedora 7.

Листинг 60. Информация о часовых поясах из /etc

```
root@pinguino:~# cat /etc/timezone
America/New_York

[root@lyrebird ~]# cat /etc/sysconfig/clock
# The ZONE parameter is only evaluated by system-config-date.
# The timezone of the system is defined by the contents of /etc/localtime.
ZONE="America/New York"
UTC=false
ARC=false
```

В некоторых системах, например, Debian и Ubuntu, для установки часового пояса есть команда `tzconfig`. В других, например, в Fedora, для установки часового пояса и проверки, используют ли часы UTC или нет, используется команда `system-config-date`. Листинг 61 иллюстрирует использование команды `tzconfig` для просмотра текущего часового пояса.

Листинг 61. Просмотр настроек часового пояса при помощи tzconfig

```
root@pinguino:~# tzconfig
Your current time zone is set to America/New_York
Do you want to change that? [n]:
Your time zone will not be changed
```

Настройка Network Time Protocol

Network Time Protocol (*NTP*) — протокол для синхронизации часов компьютера по сети. Обычно проводится синхронизация с UTC.

NTP версии 3 — Internet draft standard, описанный в RFC 1305. Текущая версия NTP версии 4, находящаяся в стадии разработки, еще не описана в RFC. RFC 4330 описывает Simple NTP (SNTP) версии 4.

Синхронизация времени достигается путем отправки сообщения *серверам времени*. Время возвращается отрегулированным при помощи смещения на половину времени задержки при прохождении туда и обратно. Поэтому точность времени зависит от задержки сети и от того, насколько задержка одинакова для обоих направлений. Чем короче путь до сервера времени, тем, вероятно, более точным будет время.

В Интернете существует огромное количество компьютеров, поэтому серверы времени организованы в *страты*. Относительно маленький номер серверов страты 1 поддерживает очень точное время от источника, например, от атомных часов. Больший номер серверов страты 2 получает их время от сервисов страты 1 и делает его доступным для еще большего номера серверов страты 3 и так далее. Чтобы облегчить нагрузку на серверы времени, большое количество волонтеров отдает сервисы времени через pool.ntp.org. Циклические (round robin) DNS-серверы достигают баланса нагрузки на NTP, распределяя запросы к NTP-серверу между группой доступных серверов.

Информация о конфигурации NTP хранится в файле /etc/ntp.conf, так что можно отредактировать файл, сохранить его и затем перезапустить демон ntpd. В листинге 62 показан пример файла /etc/ntp.conf.

Листинг 62. Файл /etc/ntp.conf

```
[root@lyrebird ~]# cat /etc/ntp.conf
# Permit time synchronization with our time source, but do not
# permit the source to query or modify the service on this system.
restrict default kod nomodify notrap nopeer noquery

# Permit all access over the loopback interface.  This could
# be tightened as well, but to do so would effect some of
# the administrative functions.
restrict 127.0.0.1

# Hosts on local network are less restricted.
#restrict 192.168.1.0 mask 255.255.255.0 nomodify notrap

# Use public servers from the pool.ntp.org project.
# Please consider joining the pool (http://www.pool.ntp.org/join.html).

#broadcast 192.168.1.255 key 42         # broadcast server
#broadcastclient        # broadcast client
#broadcast 224.0.1.1 key 42             # multicast server
#multicastclient 224.0.1.1              # multicast client
#manycastserver 239.255.254.254         # manycast server
#manycastclient 239.255.254.254 key 42  # manycast client

# Undisciplined Local Clock. This is a fake driver intended for backup
# and when no outside source of synchronized time is available.
#server 127.127.1.0  # local clock
#fudge 127.127.1.0 stratum 10

# Drift file.  Put this in a directory which the daemon can write to.
# No symbolic links allowed, either, since the daemon updates the file
# by creating a temporary in the same directory and then rename()'ing
# it to the file.
driftfile /var/lib/ntp/drift

# Key file containing the keys and key identifiers used when operating
# with symmetric key cryptography.
keys /etc/ntp/keys

# Specify the key identifiers which are trusted.
#trustedkey 4 8 42

# Specify the key identifier to use with the ntpdc utility.
#requestkey 8

# Specify the key identifier to use with the ntpq utility.
#controlkey 8
server 0.us.pool.ntp.org
restrict 0.us.pool.ntp.org mask 255.255.255.255 nomodify notrap noquery
server 1.us.pool.ntp.org
restrict 1.us.pool.ntp.org mask 255.255.255.255 nomodify notrap noquery
server 2.us.pool.ntp.org
restrict 2.us.pool.ntp.org mask 255.255.255.255 nomodify notrap noquery
```

Если используются серверы времени pool.ntp.org, они могут размещаться в мире где угодно. Обычно вы будете получать лучшее время, накладывая на серверы ограничения, как в этом примере, где используется us.pool.ntp.org, в результате чего были выбраны только серверы, находящиеся в США.

Команды NTP

Для настройки системного времени с сервера времени NTP можно использовать команду `ntpdate`, как показано в листинге 63.

Листинг 63. Настройка системного времени с сервера времени NTP при помощи ntpdate

```
[root@lyrebird ~]# ntpdate 0.us.pool.ntp.org
10 Jul 10:27:39 ntpdate[15308]: adjust time server 66.199.242.154 offset -0.007271 sec
```

Поскольку серверы работают в циклическом режиме, в следующий раз при запуске этой команды вы, возможно, увидите другой сервер. В листинге 64 показаны первые несколько DNS-ответов для 0.us.pool.ntp.org несколькими моментами позже запуска описанной выше команды `ntpdate`.

Листинг 64. Циклический NTP-сервер pool

```
[root@lyrebird ~]# dig 0.pool.ntp.org +noall +answer | head -n 5
0.pool.ntp.org.  1062   IN      A       217.116.227.3
0.pool.ntp.org.  1062   IN      A       24.215.0.24
0.pool.ntp.org.  1062   IN      A       62.66.254.154
0.pool.ntp.org.  1062   IN      A       76.168.30.201
0.pool.ntp.org.  1062   IN      A       81.169.139.140
```

Команда `ntpdate` сейчас признана устаревшей, поскольку то же самое действие можно выполнить при помощи команды `ntpq` с опцией `-q`, как показано в листинге 65.

Листинг 65. Настройка системного времени при помощи ntpd -q

```
[root@lyrebird ~]# ntpd -q
ntpd: time slew -0.014406s
```

Обратите внимание, что команда `ntpd` использует информацию о сервере времени из конфигурационного файла /etc/ntp.conf. Подробнее о команде `ntpd` и ее опциях см. в страницах руководства man. Кроме того, следует осознавать, что если запущен демон ntpd, `ntpd -q` завершит его исполнение, оставив в /var/log/messages сообщение о неудаче.

Другая связанная команда — `ntpq`, которая позволяет запросить демон NTP. Подробности см. в страницах руководства man.