

Стандартные инструменты систем Unix

В данной главе описаны утилиты, предназначенные для **поиска файлов** или **установления путей к файлам**, **сжатия файлов**, а также другие стандартные инструменты, которые не обсуждались ранее. Несмотря на то, что рассматриваемые инструменты технически не являются **фильтрами**, они могут использоваться в рамках **конвейеров**.

Утилита `find`

Утилита **`find`** может выполнять полезную работу в начале конвейера в случае возникновения необходимости в поиске файлов. Ниже представлено несколько примеров ее использования. Вы также можете рассмотреть возможность добавления конструкции для пере-

направления стандартного потока ошибок **2>/dev/null** в строки команд для того, чтобы ваш экран не заполнялся сообщениями об ошибках.

Команда для поиска всех файлов в директории **/etc** и сохранения списка имен найденных файлов в файле **etcfiles.txt**:

```
find /etc > etcfiles.txt
```

Команда для поиска файлов во всей файловой системе и сохранения списка имен найденных файлов в файле **allfiles.txt**:

```
find / > allfiles.txt
```

Команда для поиска файлов с расширением **.conf** в текущей директории (и всех поддиректориях):

```
find . -name "*.conf"
```

Команда для поиска обычных файлов с расширением **.conf** и явным указанием типа этих файлов (без директорий, именованных каналов и других файлов специальных типов):

```
find . -type f -name "*.conf"
```

Команда для поиска файлов специального типа, представляющих директории, с расширением **.bak**:

```
find /data -type d -name "*.bak"
```

Команда для поиска файлов, созданных позднее файла **file42.txt**:

```
find . -newer file42.txt
```

Утилита **find** также может исполнять произвольные команды для обработки каждого из найденных файлов. В данном примере мы осуществляем поиск файлов с расширением ***.odf** и копируем их в директорию **/backup/**.

```
find /data -name "*.odf" -exec cp {} /backup/ \;
```

Кроме того, утилита **find** может запрашивать подтверждение перед исполнением заданной команды для обработки каждого из найденных файлов. В данном примере файлы с расширением ***.odf** будут удаляться в том случае, если вы подтвердите необходимость исполнения этой операции для каждого из найденных файлов.

```
find /data -name "*.odf" -ok rm {} \;
```

Утилита **locate**

Утилита **locate** имеет значительное отличие от утилиты **find**, заключающееся в том, что она использует данные индексирования

файловой системы для установления путей к файлам. Несмотря на то, что данный алгоритм поиска гораздо быстрее алгоритма с обходом всех директорий файловой системы, в случае его использования данные о файловой системе в подавляющем большинстве случаев являются устаревшими. В том случае, если данные индексирования файловой системы еще не собраны, вам придется осуществить их сбор путем выполнения команды **updatedb** (в дистрибутиве Red Hat Enterprise Linux для выполнения описанной операции понадобятся привилегии пользователя root).

```
[paul@RHEL4b ~]$ locate Samba
locate: не удалось выполнить stat ()
"/var/lib/mlocate/mlocate.db": Нет такого файла или каталога
[paul@RHEL4b ~]$ updatedb
updatedb: не удалось открыть временный файл для
"/var/lib/mlocate/mlocate.db"
[paul@RHEL4b ~]$ su -
Password:
[root@RHEL4b ~]# updatedb
[root@RHEL4b ~]#
```

В большинстве дистрибутивов Linux для исполнения команды **updatedb** один раз в день используется планировщик задач.

Утилита **date**

Утилита **date** может использоваться для вывода информации о дате, времени, часовом поясе, а также дополнительной информации.

```
paul@rhel55 ~$ date
Sat Apr 17 12:44:30 CEST 2010
```

Форматирование строки с информацией о дате может быть изменено в соответствии с вашими предпочтениями. Обратитесь к странице руководства для получения информации о других параметрах форматирования.

```
paul@rhel55 ~$ date +%A %d-%m-%Y
Суббота 17-04-2010
```

Во всех системах Unix для подсчета времени используется количество секунд, прошедших с 1969 года (первой секундой является первая секунда первого дня января 1970 года). Для вывода времени Unix в секундах может использоваться команда **date +%s**.

```
paul@rhel55 ~$ date +%s  
1271501080
```

Когда же значение этого счетчика достигнет двух миллиардов секунд?

```
paul@rhel55 ~$ date -d '1970-01-01 + 2000000000 seconds'  
Ср май 18 03:33:20 MSK 2033
```

Утилита **cal**

Утилита **cal** выводит календарь для текущего месяца, в котором выделен текущий день.

```
Апрель 2010  
Пн Вт Ср Чт Пт Сб Вс  
          1  2  3  
 4  5  6  7  8  9 10  
11 12 13 14 15 16 17  
18 19 20 21 22 23 24  
25 26 27 28 29 30
```

Вы можете выбрать любой месяц из прошлого или будущего.

```
paul@rhel55 ~$ cal 2 1970
```

```
Февраль 1970  
Пн Вт Ср Чт Пт Сб Вс  
 1  2  3  4  5  6  7  
 8  9 10 11 12 13 14  
15 16 17 18 19 20 21  
22 23 24 25 26 27 28
```

Утилита **sleep**

Утилита **sleep** иногда используется в сценариях для перехода в режим ожидания на заданный промежуток времени в секундах. В данном примере показана методика реализации пятисекундного периода ожидания с использованием утилиты **sleep**.

```
paul@rhel55 ~$ sleep 5  
paul@rhel55 ~$
```

Команда **time**

Команда **time** выводит информацию о том, сколько времени тратится на исполнение заданной команды. На исполнение команды **date** тратится совсем немного времени.

```
paul@rhel55 ~$ time date
Cy6 Apr 17 13:08:27 CEST 2010
```

```
real    0m0.014s
user    0m0.008s
sys     0m0.006s
```

Команда **sleep 5** выполняется в течение пяти секунд **реального** времени, но при этом на ее исполнение тратится совсем немного **процессорного времени**.

```
paul@rhel55 ~$ time sleep 5
```

```
real    0m5.018s
user    0m0.005s
sys     0m0.011s
```

А на исполнение команды **bzip2**, осуществляющей сжатие файла, тратится достаточно много **процессорного времени**.

```
paul@rhel55 ~$ time bzip2 text.txt
```

```
real    0m2.368s
user    0m0.847s
sys     0m0.539s
```

Утилиты **gzip** - **gunzip**

Пользователям всегда недостаточно дискового пространства, поэтому инструменты для сжатия данных всегда актуальны. Утилита **gzip** позволяет осуществить преобразования файлов, после которых они будут занимать меньше дискового пространства.

```
paul@rhel55 ~$ ls -lh text.txt
-rw-rw-r-- 1 paul paul 6.4M apr 17 13:11 text.txt
paul@rhel55 ~$ gzip text.txt
paul@rhel55 ~$ ls -lh text.txt.gz
```

```
-rw-rw-r-- 1 paul paul 760K apr 17 13:11 text.txt.gz
```

Вы можете получить оригинал вашего файла, воспользовавшись утилитой **gunzip**.

```
paul@rhel55 ~$ gunzip text.txt.gz
```

```
paul@rhel55 ~$ ls -lh text.txt
```

```
-rw-rw-r-- 1 paul paul 6.4M apr 17 13:11 text.txt
```

Утилиты **zcat** - **zmore**

Содержимое текстовых файлов, сжатых с помощью утилиты **gzip**, может быть просмотрено с помощью утилит **zcat** и **zmore**.

```
paul@rhel55 ~$ head -4 text.txt
```

```
/
```

```
/opt
```

```
/opt/VBoxGuestAdditions-3.1.6
```

```
/opt/VBoxGuestAdditions-3.1.6/routines.sh
```

```
paul@rhel55 ~$ gzip text.txt
```

```
paul@rhel55 ~$ zcat text.txt.gz | head -4
```

```
/
```

```
/opt
```

```
/opt/VBoxGuestAdditions-3.1.6
```

```
/opt/VBoxGuestAdditions-3.1.6/routines.sh
```

Утилиты **bzip2** - **bunzip2**

Файлы также могут сжиматься с помощью утилиты **bzip2**, которая, несмотря на немного большие по сравнению с утилитой **gzip** затраты времени, позволяет достичь лучших показателей сжатия данных.

```
paul@rhel55 ~$ bzip2 text.txt
```

```
paul@rhel55 ~$ ls -lh text.txt.bz2
```

```
-rw-rw-r-- 1 paul paul 569K apr 17 13:11 text.txt.bz2
```

Декомпрессия результирующих файлов может быть осуществлена с помощью утилиты **bunzip2**.

```
paul@rhel55 ~$ bunzip2 text.txt.bz2
```

```
paul@rhel55 ~$ ls -lh text.txt
```

```
-rw-rw-r-- 1 paul paul 6.4M apr 17 13:11 text.txt
```

Утилиты **bzcat** - **bzmore**

И, аналогичным образом, с помощью утилит **bzcat** и **bzmore** может осуществляться вывод содержимого файлов, сжатых с использованием утилиты **bzip2**.

```
paul@rhel55 ~$ bzip2 text.txt
```

```
paul@rhel55 ~$ bzcat text.txt.bz2 | head -4
```

```
/
```

```
/opt
```

```
/opt/VBoxGuestAdditions-3.1.6
```

```
/opt/VBoxGuestAdditions-3.1.6/routines.sh
```