

附录 常用shell命令

本附录中列举了一些有用的 shell命令。这里并没有完全列出每个命令的各种选项，不过对于理解该命令是足够了。

这些命令的其他一些例子散布于本书的各个部分。

basename

格式：

```
basename path
```

basename命令能够从路径中分离出文件名。通常用于 shell脚本中，请看下面的例子：

```
$ basename /home/dave/myscript
myscript
```

```
echo "Usage: `basename $0` give me a file "
exit 1
...
```

如果上面的语句是脚本myscript中的一部分，那么它的输出应为：

```
myscript: give me a file
```

其中，\$0是一个包含当前脚本全路径的特殊变量。

cat

格式：

```
cat options files
```

选项：

-v：显示控制字符。

cat是最常用的文本文件显示命令。

```
$ cat myfile
```

上面的命令用于显示myfile文件。

```
$ cat myfile myfile2 >>hold_file
```

上面的命令把两个文件(myfile和myfile2)合并到hold_file中。

```
cat dt1 | while read line
do
    echo $line
done
```

在脚本中cat命令还可以用于读入文件。

compress

格式：

```
compress options files
```

选项：

-v：显示压缩结果。

compress命令可以用来压缩文件。压缩后的文件名具有 '.Z' 后缀。还可以使用该命令解压文件。

```
$ compress myfile
$ ls myfile*
myfile.Z
```

cp

格式：

```
cp options file1 file2
```

选项：

-i：在覆盖文件之前提示用户，由用户确认。

-p：保留权限模式和更改时间。

-r：拷贝相应的目录及其子目录。

要将文件myfile拷贝到myfile1.bak，使用：

```
$ cp myfile1 myfile1.bak
```

要将文件get.prd从/usr/local/sybin目录拷贝到/usr/local/bin目录，使用：

```
$ pwd
usr/local/sybin
$ cp get.prd ../bin
```

要将/logs目录下的所有文件及子目录拷贝到/hold/logs目录中，使用：

```
$ cp -r /logs /hold/logs
```

diff

格式：

```
diff options file1 file2
```

选项：

-c：按照标准格式输出(见下面的例子)。

-I：忽略大小写。

我们使用comm命令中的例子，diff命令将显示两个文件中不一致的行。

```
$ diff file1 file2
2,3c2,3
< The game
< Boys in company C
---
> The games
> The boys in company C
```

diff命令显示出两个文件中的第2行和第3行，它们的第3列不一致。

dircmp

格式：

```
dircmp options directory1 directory2
```

选项：

-s：不显示相同的文件。

dircmp命令与diff命令十分相似——它比较并显示两个目录中的不同。

dirname

格式：

```
dirname pathname
```

该目录正好和basename相反，它返回路径部分：

```
$ dirname /home/dave/myfile
/home/dave
```

du

格式：

```
du options directory
```

选项：

-a：显示每个文件的大小，不仅是整个目录所占用的空间。

-s：只显示总计。

du显示的磁盘空间占用是以512字节的块来表示的。它主要用于显示目录所占用的空间。

```
$ pwd
/var
$ du -s
14929 .
```

在本例中，/var目录所占用的空间为14929块(每块512字节)。

file

格式：

```
file filename
```

该命令用来确定文件的类型。

```
$ file core
core: ELF 32-bit LSB core file of 'awk' (signal 6), Intel 80386,
version 1
```

```
$ file data.f
data.f: ASCII text
```

```
$ file month_end.sh
month_end.sh: Bourne shell script text
```

fuser

格式：

```
fuser options file
```

选项：

-k：杀死所有访问该文件或文件系统的进程。

-u：显示访问该文件或文件系统的所有进程。

fuser命令可以显示访问某个文件或文件系统的所有进程。在有些系统上 -u和-m选项可以互换。还可以在if语句中使用fuser命令。

要列出设备/dev/hda5上的所有活动进程，使用：

```
$ fuser -m /dev/hda5
/dev/hda5: 1 1r 1c 1e 37 37r 37c 37e 144
144r 144c 144e 158 158r 158c 158e 167r 167c
167e 178 17 8r 178c 178e 189 189r 189c
```

要杀死设备/dev/hda5上的所有进程，使用：

```
$ fuser -k /dev/hda5
```

要查看doc_part文件是否被打开，有哪些进程在使用，可用：

```
$ fuser -m /root/doc_part
/root/dt: 1 1r 1c 1e 37 37r 37c 37e 144
144r 144c 144e 158 158r 158c 158e 167r 167c
167e 178 178r 178c 178e 189 189r 189c 189e 201
201r 201c 201e 212 212r 212c 212e 223 223r
```

有些系统上的fuser命令能够在列表中显示用户登录ID。如果你的系统不具有这样的功能，可以按照fuser命令输出中末尾含有‘e’的数字在ps -ef或ps xa命令的输出中用grep命令查找相应的用户登录ID。

head

格式：

```
head -number files
```

head命令可以显示相应文件的前10行。如果希望指定显示的行数，可以使用-number选项。

例如：

```
$ head -1 myfile
```

只显示文件的第一行，而

```
$ head -30 logfile |more
```

则显示logfile文件的前30行。

logname

格式：

```
logname
```

该命令可以显示当前所使用的登录用户名：

```
$ logname
dave
```

mkdir

格式：

```
mkdir options directory
```

选项：

-m：在创建目录时按照该选项的值设置访问权限。

```
$ mkdir HOLD_AREA
$ ls -l HOLD*
-rw-rw-r-- 1 dave admin 3463 Dec 3 1998 HOLD_AREA
```

上述命令创建了一个名为 HOLD_AREA 的目录。

more

格式：

```
more options files
```

该命令和page及pg命令的功能相似，都能够分屏显示文件内容。

选项：

-c：不滚屏，而是通过覆盖来换页。

-d：在分页处显示提示。

-n：每屏显示n行。

```
$ more /etc/passwd
```

上面的命令显示passwd文件

```
$ cat logfile |more
```

上面的命令显示logfile文件。

nl

格式：

```
nl options file
```

选项：

-l：行号每次增加n；缺省为1。

-p：在新的一页不重新计数。

nl命令可用于在文件中列行号，在打印源代码或列日志文件时很有用。

```
$ nl myscript
```

上面的命令将列出myscript文件的行号。

```
$ nl myscript >hold_file
```

则将上面命令的输出重定向到hold_file文件中。

```
$ nl myscript | lpr
```

将上面命令的结果重定向到打印机。

printf

格式：

```
printf format arguments
```

该命令有点类似于awk命令的printf函数，它将格式化文本送至标准输出。

其中，格式符format包含三种类型的项，这里我们只讨论格式符：

```
%[- +]m.nx
```

其中横杠-为从行首算起的起始位置。一般说来 m表示域的宽度而n表示域的最大宽度。

‘%’后面可跟下列格式字符：

s：字符串。

c：字符。

d：数字。

x：16进制数。

o：10进制数。

printf命令本身并不会产生换行符，必须使用转义字符来实现这样的功能。下面是最常用的转义字符：

\a：响铃。

\b：退格。

\r：回车。

\f：换页。

\n：换行。

\t：跳格。

```
$ printf "Howzat!\n"
Howzat!
```

上面的命令输出了一个字符串，使用\n来换行。

```
$ printf "\x2B\n"
+
```

上面的命令把16进制值转换为ASCII字符+。

```
$ printf "%-10sStand-by\n"
Stand-by
```

上面的命令从左起第10个字符的位置开始显示字符串。

pwd

格式：

```
pwd
```

显示当前的工作目录，可以用：

```
$ pwd
/var/spool
```

```
$ WHERE_AM_I='pwd'
$ echo $WHERE_AM_I
/var/spool
```

在上面的脚本中，使用了命令置换来获得当前目录。

rm

格式：

```
rm options files
```

选项：

-i：在删除文件之前给出提示(安全模式)。

-r：删除目录。

rm命令能够删除文件或目录。

```
$ rm myfile
$ rm -r /var/spool/tmp
```

上面的第二条命令能够删除 /var/spool/tmp目录下的所有文件及子目录。

rmdir

格式：

```
rmdir options directory
```

选项：

-p：如果相应的目录为空目录，则删除该目录。

```
$ rmdir /var/spool/tmp/lp_HP
```

上面的命令将删除 /var/spool/tmp目录下的lp_HP目录。

script

格式：

```
script option file
```

-a：将输出附加在文件末尾。

可以使用script命令记录当前会话。只要在命令行键入该命令即可。该命令在你退出当前会话时结束。它可以将你的输入记录下来并附加到一个文件末尾。

```
$ script mylogin
```

将会启动script命令并将所有会话内容记录在 mylogin文件中。

shutdown

格式：

```
shutdown
```

该命令将关闭系统。很多系统供应商都有自己特定的命令变体。

```
$ shutdown now
```

上面的命令将会立即关机。

```
$ shutdown -g60 -I6 -y
```

上面的命令将会在60秒之后关机，然后重新启动系统。

sleep

格式：

```
sleep number
```

该命令使系统等待相应的秒数。例如：

```
$ sleep 10
```

意味着系统在10秒钟之内不进行任何操作。

strings

格式：

```
strings filename
```

该命令可以看二进制文件中所包含的文本。

touch

格式：

```
touch options filename
```

选项：

-t MMDDhhmm 创建一个具有相应月、日、时分时间戳的文件。

下面的命令能够以当前时间创建文件或更新已有文件的时间戳。

```
$ touch myfile
$ ls -l myfile
-rw-r--r-- 1 dave admin 0 Jun 30 09:59 myfile
```

上面的命令以当前时间创建了一个名为 myfile 的文件。

```
$ touch -t 06100930 myfile2
$ ls -l myfile2
-rw-r--r-- 1 dave admin 0 Jun 10 09:30 myfile2
```

上面的命令以时间戳6月10日上午9:30创建了一个名为 myfile2 的空文件。

tty

格式：

```
tty
```

可以使用 tty 来报告所连接的设备或终端。

```
$ tty
/dev/tty08
```

可以使用 tty -s 命令来确定脚本的标准输入。返回码为：

0：终端。

1：非终端。

uname

格式：

```
uname options
```

选项：

-a：显示所有信息。

-s：系统名。

-v：只显示操作系统版本或其发布日期。

要显示当前操作系统名及其他相关信息，可以用：

```
$ uname a
Linux bumper.honeysuckle.com 2.0.36 #1 Tue Oct...
```

uncompress

格式：

```
uncompress files
```

可以使用该命令来恢复压缩文件。

```
$ uncompress myfile
```

上面的命令解压缩先前压缩的 myfile 文件。注意，在解压缩时不必给出 .Z 后缀。

wait

格式：

```
wait process ID
```

该命令可以用来等待进程号为 process ID 的进程或所有的后台进程结束后，再执行当前脚本。

下面的命令等待进程号为 1299 的进程结束后再执行当前脚本：

```
$ wait 1299
```

下面的命令等待所有的后台进程结束后再执行当前脚本：

```
$ wait
```

WC

格式：

```
wc options files
```

选项：

-c：显示字符数。

-l：显示行数。

-w：显示单词数。

该命令能够统计文件中的字符数、单词数和行数。

```
$ who|wc
 1      6     46
$ who|wc -l
 1
```

在上面第一个例子中，who 命令的输出通过管道传递给 wc 命令，该命令显示出如下的几列：

行数、单词数、字符数

在上面的第二个例子中，wc 命令只显示文件中所包含的行数。

```
$ VAR="tapedrive"
echo $VAR | wc -c
10
```

上面的脚本显示出变量VAR中所包含的字符串的长度。

whereis

格式：

```
whereis command_name
```

whereis命令能够给出系统命令的二进制文件及其在线手册的路径。

```
$ whereis fuser
fuser: /usr/sbin/fuser /usr/man/man1/fuser.1
```

```
$ whereis sort
sort: /bin/sort /usr/man/man1/sort.1
```

注意，在下面的例子中，whereis命令没有显示出相应命令的二进制文件路径，因为它们在内建的shell脚本，但是该命令给出了其在线手册的路径。

```
$ whereis times
times: /usr/man/man2/times.2
```

```
$ whereis set
set: /usr/man/mann/set.n
```

who

格式：

```
who options
```

选项：

- a：显示所有的结果。
- r：显示当前的运行级别(在Linux系统中应当使用runlevel命令)。
- s：列出用户名及时间域。

whoami 显示执行该命令的用户名。这不是who命令的一个选项，可以单独应用。

who命令可以显示当前有哪些用户登录到系统上。要显示当前登录的用户，可以用：

```
$ who
root          console      Apr 22 13:27
pgd           pts/3        Jun 14 15:29
peter         pts/4        Jun 14 12:08
dave          pts/5        Jun 14 16:10
```

要显示自己的用户名，可以用：

```
$ whoami
dave
```