第 7 章　Linux 终端常用命令

Linux 是一种类 UNIX 操作系统。在 UNIX 的发展早期，类 UNIX 操作系统没有图形界面，只有类 DOS 那样的命令行工作模式。后来随着 GUI 的发展，为个方便用户使用才在类 UNIX 系统上开发了 X-Window 系统。X-Window 系统同广泛使用的 Windows 系统功能类似，用户可以使用简单的窗口式界面同系统进行交互。

虽然图形用户界面操作简单直观，但是命令行操作方式仍然沿用至今，并且依然是 Linux 系统管理员进行系统配置与管理的首选方式。这主要因为：

- 命令行模式下，系统具有较高的执行效率，且稳定性高。管理员不必等待 Linux 处理 GUI 程序而直接可以得到结果。
- 命令行模式不需要启动图形工作环境，可以节省大量的系统资源。
- 命令行工具比较通用，拥有比 GUI 更多的命令行选项。
- 由于没有 GUI 开销，命令行工具更容易在远程计算机上运行。
- GUI 是另一个软件层，带来方便的同时也容易引入新的错误。
- GUI 工具仅有限地返回出错信息。

因此，掌握 Linux 系统中常用的终端命令对 Linux 系统管理员来说十分重要。Linux 系统中比较常用的终端命令分为以下几类：信息显示命令、系统管理命令、软盘操作命令、压缩备份命令、联机帮助命令等，本章将做详细介绍。

7.1　Linux 的终端窗口

早期的 Linux 系统并没有现在 Linux 系统所具备的 X-Window 图形化管理界面，而只有命令行终端模式来实现人机交互。只是后来随着 Linux 系统影响力越来越大，使用的用户也越来越多，为了方便普通用户使用 Linux 系统，才设计并开发出了 X-Window 图形化界面。但是原来的命令行终端模式仍然保留，并且发挥着非常重要的作用。

可以采用 3 种方法进入命令行终端工作方式：

- 在图形界面下启动终端窗口进入命令行工作方式
- 在系统启动时直接进入命令行终端方式
- 使用远程登录方式

7.1.1　启动终端窗口

Red Hat Enterprise Linux 5 像以往的版本一样，在 X-Window 图形环境中仍然保留了命令行终端窗口。在桌面的【应用程序】菜单中选择【附件】，在打开的子菜单里可以看到【终端】选项，单击该选项即可打开命令行终端窗口。在 X-Window 下打开命令行终端窗口如图 7.1 所示。

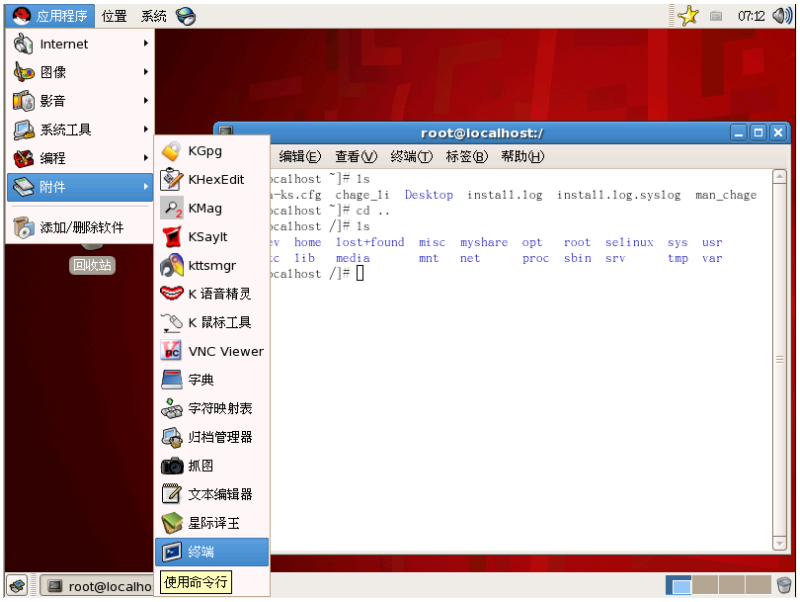

图 7.1 打开命令行终端窗口

打开命令行终端窗口后，会显示一个 Shell 提示符。若是根用户登录系统，提示符为“#”，若为普通用户登录系统则为“$”， 用户可以在提示符下输入带有选项和参数的字符命令，并能够在终端窗口里看到命令的运行结果。命令执行结束，系统会重新返回一个提示符，等待接收新的命令。

7.1.2　终端窗口的常规操作

在图 7.1 中所示的终端窗口中选择【文件】|【打开终端】或按快捷键 Shift+Ctrl+N 可以新建一个终端窗口。选择【文件】|【打开标签】或按快捷键 Shift+Ctrl+T 可以新建一个标签，如图 7.2 所示。

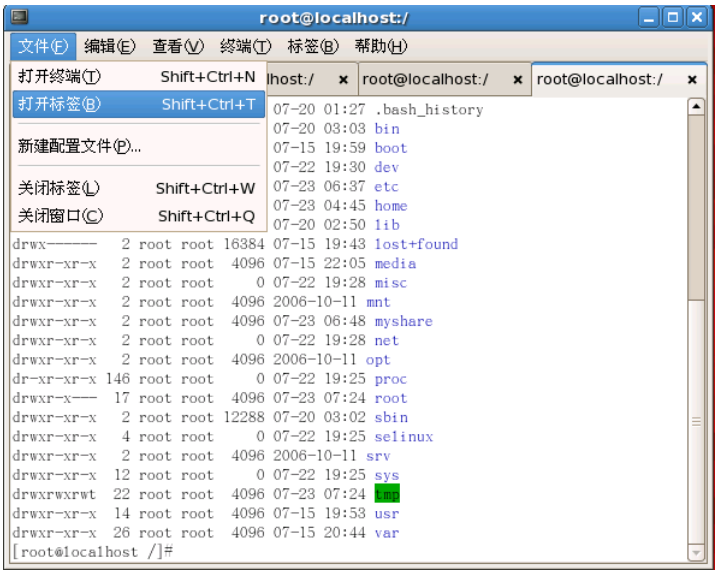

图 7.2 新建一个标签

终端窗口支持一些文本编辑器特征，窗口的右边是一个滚动条，用户可以通过它来查看以前输入的命令以及产生的结果。利用图 7.2 中【编辑】菜单的【复制】、【粘贴】功能，可以复制任何以前显示

过的文本，通过编辑后可以组成更复杂的命令行。复制粘贴命令还可以在不同标签、不同终端窗口中完成复制。

当编辑命令行时，新键入的字符会出现在光标所在的位置，可以使用左右方向键把光标在命令行上从一端移动到另一端，也可以按上下方向键在不同行之间移动。表 7-1 列出了可以用来移动光标的按键组合。

表 7-1 移动光标的快捷键

快捷键	功能	说明
Ctrl+f	字符前移	向前移动一个字符
Ctrl+b	字符后移	向后移动一个字符
Alt+f	单词前移	向前移动一个单词
Alt+b	单词后移	向后移∞一个单词
Ctrl+a	行起始	移动到当前行的行首
Ctrl+e	行结尾	移动到当前行的行尾
Ctrl+l	清屏	清屏并在屏幕的最上面开始一个新行

表 7-2 列出了编辑命令行输入的一些快捷方式。

表 7-2 编辑命令行输入的快捷方式

快捷键	功能	说明
Ctrl+d	删除当前位	删除当前的字符
Back space	删除前一位	删除前一个字符
Ctrl+t	调换字符	交换当前字符和前一个字符的位置
Alt+t	调换单词	交换当前单词和前一单词的位置
Alt+u	大写单词	把当前单词变成大写
Alt+l	小写单词	把当前单词变成小写
Alt+c	首字母大写单词	把当前单词变成首字母大写的单词
Ctrl+v	插入特殊字符	添加一个特殊字符。例如，要添加一个制表符，按Ctrl+v+Tab

表 7-3 列出了在命令行中实现剪切和粘贴的快捷方式。

表 7-3 命令行中剪切和粘贴的快捷方式

快捷键	功能	说明
Ctrl+k	剪切至行末	剪切文本直到行的末尾
Ctrl+u	剪切至行首	剪切文本直到行的起始
Ctrl+w	剪切前个单词	剪切光标前的单词
Alt+d	剪切下一个单词	剪切光标后的单词
Ctrl+y	粘贴新近文本	粘贴最近剪切的文本
Alt+y	粘贴先前的文本	回退到先前剪切的文体并粘贴它
Ctrl+c	删除整行	删除整行

在 Red Hat Enterprise Linux 中，命令行是区分大小写的。例如系统会认为“student”与“Student”是不同的两个名字。

7.1.2 命令行自动补全

为了简化打字工作，Bash Shell 提供了几种可以对输入不完整的值进行自动补全的方法。如果想要对输入进行自动补全，只需键入初始的几个字符，然后按<TAB>键，系统会自动匹配其余所需的输入。当有多种匹配，系统会进行提示，按“Esc+?”或是按两次<TAB>键，可以列出所有可能的匹配。自动

补全可以应用在下列四类输入工作中：

用环境变量名补全：如果输入的文本以“$”开始，Shell 就以当前 Shell 的一个环境变量补全文本。例如：

```
# echo   $P   <TAB><TAB>                    //按两次 TAB 键，列出了环境变量中所有第一个字母为“P”的可能匹配，
                                            //在显示所有的可能性之后，返回原来命令行，等待用户的选择
$PATH                                       $PRELINK_OPTS
$PIPESTATUS                                 $PROMPT_COMMAND
$PPID                                       $PS1
$PRELINK_FULL_TIME_INTERVAL          $PS2
$PRELINKING                                 $PS4
$PRELINK_NONRPM_CHECK_INTERVAL   $PWD
# echo   $PA   <TAB>                        //匹配了 PATH 环境变量
/usr/lib/qt-3.3/bin:/usr/kerberos/sbin:/usr/kerberos/bin:/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/usr/
X11R6/bin:/root/bin
```

如果仅输入“$”，则系统会按字母顺序列出所有的环境变量：

```
$ $    <TAB><TAB>                         //按两次 TAB 键，列出所有的环境变量
$_
$BASH
$BASH_ARGC
$BASH_ARGV
$BASH_COMMAND
$BASH_LINENO
$BASH_SOURCE
$BASH_SUBSHELL
$BASH_VERSINFO
$BASH_VERSION
$COLORS
$COLORTERM
...
```

1. 用用户名补全

如果输入的文本以波浪线“～”起始，Shell 会以用户名补全文本。例如：

```
# cd   ~s    <TAB><TAB>                    //按两次 TAB，列出所有以字母“s”开头的用户名
~sabayon      ~smmsp/      ~sshd/
~shutdown/   ~squid/       ~student1/   ~sync/
# cd   ~st   <Tab><Tab>                    //按两次 TAB，匹配了用户名 student1
#pwd
/student1
```

2. 用命令、别名或函数的名字补全

如果文本以常规字符起始，Shell 尝试利用命令、别名或函数名来补全文本。例如：

```
# ls
chage_li   man_chage   student1   student3   teacher1
# cd   t   <TAB>          //按一次 TAB，匹配了 teacher1 目录名
#pwd
/home/teacher1
```

3. 用主机名补全

如果输入的文本以“@”符号起始，系统会利用/etc/hosts 文件中的主机名来补全文本。例如：

```
# mail   root@      <TAB><TAB>      //按两次 TAB，列出了所有可用的主机名
```

```
@::1                    @localhost6              @localhost.localdomain
@localhost              @localhost6.localdomain6
```

表 7-4 列出了自动补全的快捷方式。

表 7-4 自动补全快捷方式

快捷键	说明
Alt+~	以一个用户名补全此处的文本
Alt+$	以一个变量补全此处的文本
Alt+@	以一个主机名补全此处的文本
Ctrl+x+～	列出可能的用户名补全
CtrI+x+$	列出可能的环境变量补全
Ctrl+x+@	列出可能的主机名补全
Ctrl+x+!	列出可能的命令名补全

7.2　常用的信息显示命令

在 Linux 系统下包含了大约 2000 个经常用到的命令。虽然这些命令大部分可以通过图形界面的操作来完成，但命令行快速、高效的优点会为系统的使用、管理和维护提供了极大的便捷。

在 Red Hat Enterprise Linux 中，常用的信息显示命令大约有 20 多个。通过信息显示命令，系统管理员可以直接了解系统的工作状态。

7.2.1　pwd 命令

pwd 命令用于从屏幕上输出当前的工作目录，如下所示：

```
#pwd
/home/student1
```

7.2.2　stat 命令

stat 命令用于显示文件指定文件的相关信息，如下所示：

```
#stat   who.txt
  File: “who.txt”
  Size: 46                Blocks: 16          IO Block: 4096     一般文件
Device: fd00h/64768d      Inode: 390155       Links: 1
Access: (0644/-rw-r--r--)  Uid: (      0/      root)   Gid: (      0/      root)
Access: 2007-07-23 06:49:29.000000000 +0800
Modify: 2007-07-23 06:48:20.000000000 +0800
Change: 2007-07-23 06:48:20.000000000 +0800
```

7.2.3　uname 命令

uname 命令用于显示操作系统信息，如下所示：

```
# uname  -a
Linux localhost.localdomain 2.6.18-8.el5xen #1 SMP Fri Jan 26 14:42:21 EST 2007 i686 i686 i386 GNU/Linux
```

7.2.4　hostname 命令

hostname 命令用于显示当前本地主机的名称，如下所示：

```
# hostname
localhost.localdomain
```

7.2.5　dmesg

dmesg 命令用于显示系统最后一次启动时的内核内部缓存信息，如下所示：

```
＃dmesg
Linux version 2.6.18-8.el5xen (brewbuilder@ls20-bc2-14.build.redhat.com) (gcc version 4.1.1 20070105 (Red
Hat 4.1.1-52)) #1 SMP Fri Jan 26 14:42:21 EST 2007
BIOS-provided physical RAM map:
Xen: 0000000000000000 - 000000000ffc7000 (usable)
0MB HIGHMEM available.          //0MB 高端内存可用
255MB LOWMEM available.         //255MB 低端内存可用
Using x86 segment limits to approximate NX protection
On node 0 totalpages: 65479
DMA zone: 65479 pages, LIFO batch:15
... ...
device peth0 entered promiscuous mode       //peth0 进入混杂模式
xenbr0: port 2(peth0) entering learning state
xenbr0: topology change detected, propagating
xenbr0: port 2(peth0) entering forwarding state
eth0: no IPv6 routers present                    //eth0 没有发现 ipv6 路由器
```

7.2.6　free 命令

free 命令用于显示当前内存和交换分区的使用情况，如下所示：

```
# free
                 total       used       free     shared    buffers     cached
Mem:           252444     248712       3732          0       1516      59636
-/+ buffers/cache:     187560      64884
Swap:          589816      52528     537288
```

7.2.7　locale 命令

locale 命令用于显示当前系统的语言设置，如下所示：

```
# locale
LANG=zh_CN.UTF-8
LC_CTYPE="zh_CN.UTF-8"
LC_NUMERIC="zh_CN.UTF-8"
LC_TIME="zh_CN.UTF-8"
...  ...
```

7.2.8　cat /etc/issue 命令

“cat　/etc/issue”显示当前系统的发行版本，如下所示：

```
# cat　/etc/issue
Red Hat Enterprise Linux Server release 5 (Tikanga)
Kernel \r on an \m
```

7.2.9　lastb 命令

lastb 是 last bad 的缩写，显示不成功登录的用户信息。系统将记录登录出错信息并存放在/var/log/btmp 文件中，lastb 命令会读取并显示该文件的内容，例如：

```
# lastb
student1 :0                                  Sun Jul 22 18:35 - 18:35    (00:00)
teacher1 :0                                  Sun Jul 22 18:35 - 18:35    (00:00)
ymh          :0                                Fri Jul 20 01:33 - 01:33    (00:00)
btmp begins Fri Jul 20 01:33:18 2007
```

7.2.10　date 命令

date 命令用于显示系统当前的日期、时间。配合参数“-s”可以对系统的日期、时间重新设定，如下所示：

```
# date
2007 年 07 月 23 日 星期一 10:41:48 CST
# date -s 2007-07-24
2007 年 07 月 24 日 星期二 00:00:00 CST
# date -s 12:59:00
2007 年 07 月 24 日 星期二 12:59:00 CST
# date
2007 年 07 月 24 日 星期二 12:59:05 CST
```

7.2.11　cal 命令

cal 命令用于显示本月的月历。带参数“-y”可以显示全年的年历，如下所示：

```
#cal
      七月    2007
    日 一 二 三 四 五 六
                    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
      31
```

按缩略日期表示形式（显示距 1 月 1 日的天数），显示 2007 年 8 月的日历，如下所示：

```
$ cal   -j 8   2007
          八月    2007
    日   一   二   三   四   五   六
```

```
        213 214 215 216 217 218
    219 220 221 222 223 224 225
    226 227 228 229 230 231 232
    233 234 235 236 237 238 239
    240 241 242 243
```

7.2.12　time 命令

time 命令计算执行一个进程所需要的时间。包括实际 CPU 时间、用户时间、系统时间。例如查看执行命令 ls 所需的时间，如下所示：

```
# time   ls
cal_txt         finger3_txt   groups_txt    id_txt      newgrp_txt   suple_txt~   w.txt
finger2_txt   finger_txt    groups_txt2   last_txt   suple_txt    who.txt
real      0m0.156s
user      0m0.008s
sys       0m0.096s
```

7.2.13　clock 命令

clock 命令用于显示时钟，如下所示：

```
#clock
2007 年 07 月 24 日 星期二 21 时 05 分 12 秒   -0.516871 seconds
```

7.2.14　cat /proc/cpuinfo 命令

“cat　/proc/cpuinfo”命令用于显示 CPU 的相关信息，如下所示：

```
cat   cpuinfo
processor         : 0
vendor_id         : GenuineIntel      //intel 处理器
cpu family        : 15
model             : 2
model name        : Mobile Intel(R) Celeron(R) CPU 2.00GHz     //intel 赛扬 2.0
stepping          : 8
cpu MHz           : 1995.839          //主频为 1995.839
cache size        : 256 KB            // 缓存 256KB
fdiv_bug          : no
hlt_bug           : no
f00f_bug          : no
coma_bug          : no
fpu               : yes               //支持浮点运算
fpu_exception     : yes
cpuid level       : 2
wp                : yes
flags             : fpu tsc msr pae mce cx8 apic mtrr mca cmov pat pse36 clflush dts acpi mmx fxsr sse sse2 ss
up
bogomips          : 5085.78
```

注意：/proc 目录下的文件不是真正意义上的文件，关闭系统后，该目录中的文件不再存在。该目录文件通常用来帮助查看内核运行信息。

7.2.15　cat /proc/interrupts 命令

“cat　/proc/interrupts”显示系统中正在使用的中断号（IRQ），如下所示：

```
# cat   /proc/interrupts
              CPU0
  1:        10372          Phys-irq   i8042
  6:            5          Phys-irq   floppy
  7:            0          Phys-irq   parport0
  8:            2          Phys-irq   rtc
 ...   ...
```

7.2.16　cat /proc/filesystems 命令

“cat　/proc/filesystems”命令用于显示当前使用的文件系统类型，如下所示：

```
# cat   /proc/filesystems
nodev     sysfs
nodev     rootfs
nodev     bdev
nodev     proc
...   ...
nodev     ramfs
          iso9660              //光盘所使用的文件系统
nodev     mqueue
nodev     selinuxfs
          ext3
nodev     rpc_pipefs
nodev     autofs
```

7.2.17　lsmod 命令

lsmod 命令用于显示目前已经加载的内核模块，如下所示：

```
# lsmod
Module                  Size   Used by
bridge                 53725   0
netloop                10817   0
netbk                  78017   0 [permanent]
blktap                385125   2 [permanent]
blkbk                  21089   0 [permanent]
...   ...
gameport               18889   1 snd_ens1371      //设置游戏端口
...   ...
mptscsih               26177   1 mptspi
mptbase                53089   2 mptspi,mptscsih
scsi_transport_spi     26177   1 mptspi
```

```
sd_mod                  22977  3
scsi_mod               130893  5 sg,mptspi,mptscsih,scsi_transport_spi,sd_mod   //scsi 设置模块
ext3                   123081  2
jbd                     56553  1 ext3
ehci_hcd                33229  0
ohci_hcd                23645  0
uhci_hcd                25677  0
```

7.2.18　set 命令

set 命令用于显示当前用户的环境设置，如下所示。

```
# set
BASH=/bin/bash                    //bash 使用/bin/bash
BASH_ARGC=()
BASH_ARGV=()
...  ...
XAUTHORITY=/tmp/.gdmYD66VT
XMODIFIERS=@im=SCIM
_=lsmod
consoletype=pty
qt_prefix=/usr/lib/qt-3.3         //qt 的默认路径为/usr/lib/qt-3.3
```

7.2.19　rulevel 命令

rulevel 命令用于输出前一个和当前的运行级别，例如：

```
# runlevel
N  5
```

输出“N　5”表示没有前一个运行级别，且当前的运行级别是 5。系统运行级别相关说明见表 7-5。

表 7-5 运行级别说明

运行级别	说明
0	关闭系统
1	单用户模式
2	多用户模式，但不支持NFS
3	完全的多用户模式
4	保留
5	图形用户界面
6	重新启动

系统的初始运行级别设置存放在文件/etc/inittab 里，不要把系统初始运行级别设置为 0 或 6,否则系统无法正常启动。

7.2.20　sysctl -a 命令

“sysctl　-a”命令显示 Red Hat Enterprise Linux 5 系统中所有可以设置的内核参数，如下所示：

```
# sysctl  -a
sunrpc.max_resvport = 1023
sunrpc.min_resvport = 665
```

```
sunrpc.tcp_slot_table_entries = 16
sunrpc.udp_slot_table_entries = 16
...  ...
```

7.2.21　uptime 命令

uptime 命令用于显示系统自上次启动到现在总的运行时间，如下所示：

```
# uptime
14:37:39 up 16:56,  4 users,  load average: 0.82, 0.18, 0.06
```

7.2.22　ps 命令

ps 命令用于监测进程的工作情况。进程是正在运行的程序，一直处于动态变化过程中，而 ps 命令所显示的进程工作状态是瞬时间的。如果试图连续查看某一进程的工作情况，必须连续使用该命令或者换用能够动态显示进程状态的 top 命令。ps 的命令格式如下：

```
ps [-e][-f][-h][-l][-w][-a][-r][-x][-u]
```

- -e：显示所有进程。
- -u：显示用户的 UID。
- -A：显示所有进程，等同于“-e”
- -w：宽格式输出
- -l：长格式输出
- -h：不显示标题
- -f：全格式

例如查看当前所有进程，命令行如下：

```
# ps  -e
  PID TTY          TIME CMD
    1 ?        00:00:04 init
    2 ?        00:00:00 migration/0
    3 ?        00:00:00 ksoftirqd/0
    4 ?        00:00:00 watchdog/0
  ...  ...
```

例如显示所有不是以 root 身份运行的进程，命令行如下：

```
# ps  -U root  -u root  -N
  PID TTY          TIME CMD
 1844 ?        00:00:00 portmap
 1932 ?        00:00:16 dbus-daemon
 2161 ?        00:00:00 sendmail
 2219 ?        00:00:00 xfs
 2286 ?        00:00:01 avahi-daemon
 2287 ?        00:00:00 avahi-daemon
 2298 ?        00:00:14 hald
 2305 ?        00:00:00 hald-addon-acpi
 2308 ?        00:00:01 hald-addon-keyb
```

例如根据用户指定的排列方式进行显示，命令行如下：

```
# ps  -eo  pid,tt,user,fname,f,tmout,wchan
  PID TT       USER     COMMAND        F       TMOUT     WCHAN
```

```
1 ?          root       init            4       -          -
2 ?          root       migratio        1        -         migration_thread
3 ?          root       ksoftirq        1       -          ksoftirqd
4 ?          root       watchdog        5       -          watchdog
5 ?          root       events/0        5       -          worker_thread
6 ?          root       khelper         1       -          worker_thread
…   …
```

例如显示所有进程状态和进程的基本信息，命令行如下：

```
# ps   -aux
Warning: bad syntax, perhaps a bogus '-'? See /usr/share/doc/procps-3.2.7/FAQ
USER        PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root          1  0.0  0.2   2044   560 ?        Ss   Jul23   0:04 init [5]
root          2  0.0  0.0      0     0 ?        S    Jul23   0:00 [migration/0]
root          3  0.0  0.0      0     0 ?        SN   Jul23   0:00 [ksoftirqd/0]
...   ...
root       9703  0.0  0.3   4472   964 pts/4    R+   14:47   0:00 ps -aux
```

7.2.23　top 命令

top 命令动态显示当前系统中消耗资源最多的进程，如图 7.3 所示。top 命令与 ps 命令基本作用相同，都是用来显示当前的进程及其状态。但 top 能够实现动态显示，可以不断刷新当前状态，即 top 提供了对系统处理器状态的实时监测。如果在前台执行，该命令会占据整个前台，直到用户按“q”键退出。

默认情况下该命令按照处理器占用时间、占用内存大小和进程的执行时间对进程进行排序，还可以根据用户需要，在定制文件中对其进行设定。

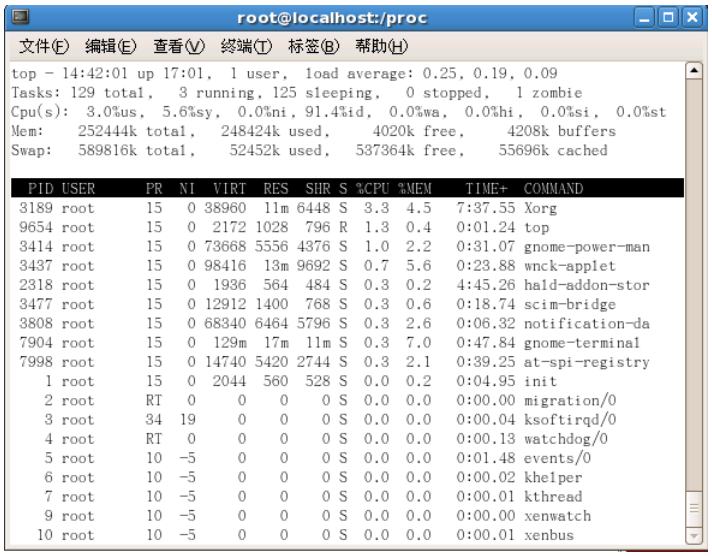

图 7.3 top 命令动态显示当前进程

7.2.24　pstree 命令

pstree 命令将所有进程以树状图的方式进行显示。树状图默认会以 init 进程为根，如果指定了选项 pid，则只显示指定进程为根的树状图。pstree 的命令格式如下：

```
pstree [-a] [-c] [-h|-Hpid] [-l] [-n] [-p] [-u] [-Z] [-A|-G|-U]
        [pid|user]
```

- -a：显示进程完整的指令及参数
- -p：显示进程的进程号
- -h：高亮显示当前进程及其父进程

例如以树状图方式显示进程及进程号，命令行如下：

```
#pstree -p
init(1)─┬─/usr/bin/sealer(3454)
        ├─acpid(2065)
        ├─at-spi-registry(7998)
        ├─atd(2255)
        ├─auditd(1785)─┬─python(1787)
        │              └─{auditd}(1786)
        ├─automount(2046)─┬─{automount}(2047)
        │                 ├─{automount}(2048)
        │                 ├─{automount}(2051)
        │                 └─{automount}(2054)
        ├─avahi-daemon(2286)───avahi-daemon(2287)
...   ...
```

7.2.25　history 命令

history 命令保留了最近执行的命令，默认可以保留 500 个。历史清单从 0 开始编号，直到保留的最大值。快速重新执行已经执行过的命令可以使用“！<命令事件号>”，例如：

```
#history
…   …
281   cd student1
282   ls
283   ls -al
284   cd ..
285   history
# !282
ls
anaconda-ks.cfg   chage_li   Desktop   install.log   install.log.syslog   man_chage
```

如果要检查某用户在系统上都运行了什么命令，可以以根用户身份登录系统，检查该用户主目录下的“.bash_history”文件。该文件中记录了用户所使用的命令和历史信息。例如：

```
#cd   /home/teacher1
# cat   .bash_history
rm   /etc/test
mkdir   homework
ls   -al
more
…   …
```

7.2.26　mesg 命令

mesg 命令用于设置是否允许其他用户用 write 命令给自己发送信息。如果允许别人发送信息，使用

命令：

```
mesg  y
```

如果不允许别人发送，则使用

```
mesg  n
```

根用户默认不允许其他用户给它发消息；普通用户默认允许。

7.3　常用的系统管理命令

在 Red Hat Enterprise Linux 中，经常用到的系统管理命令大约有 20 多个。系统管理员可以通过系统管理命令对系统的软硬件资源进行控制。

7.3.1　mkbootdisk 命令

mkbootdisk 命令用于创建系统启动盘。mkbootdisk 命令格式为：

```
mkbootdisk [--version] [--noprompt] [--verbose]
                    [--device devicefile] [--size size]
                    [--kernelargs <args>] [--iso] kernel
```

- -device devicefile：在设备文件上创建启动镜像。如果设备文件未指定，默认为/ev/fd0。
- -noprompt：正常情况下，mkbootdisk 会提示用户插入软盘，等待用户确认，然后开始制作启动盘。采用该参数后，mkbootdisk 不再进行提示。
- -iso：制作 iso 启动镜像文件。
- -version：显示 mkbootdisk 版本。
- -size：定制启动盘的容量，默认为 1.44Mb。

7.3.2　kill 命令

kill 命令用于杀死指定的进程。该命令向指定的进程发送终止运行的信号。进程收到终止指令后将自动结束，并处理好结束前的相关工作。其命令的格式为：

```
kill  [pid]
```

杀死进程前必须知道该进程的进程号，可以使用“ps | grep”命令查找相应的进程号，例如杀死进程“tt”：

```
# ps  |  grep  tt
10995 pts/4      00:00:00 tt
# kill 10995
```

对于一些没有能力自动结束的进程，可以使用参数“-9”强行结束。由于强行退出属于非正常结束，有可能造成数据的丢失，所以使用该选项一定要谨慎。例如杀死正在运行的 vi 进程。由于 vi 要求存盘后退出，直接使用 kill 命令无法正常结束，必须配合参数“-9”，例如：

```
# ps  |  grep  vi
10997 pts/4      00:00:00 vi
# kill  10997
# ps  |  grep  vi
10997 pts/4      00:00:00 vi
# kill  -9  10997
```

```
# ps  |  grep  vi
[1]+  已杀死                    vi
```

7.3.3　killall 命令

killall 命令使用进程名称来终止进程的运行。如果系统中有多个相同名称的进程，这些进程将全部被结束。例如：

```
# ps  |  grep  tt
10995 pts/4      00:00:00 tt
10995 pts/4      00:00:00 tt
#killall  tt
```

参数“-9”用于强制结束指定名称的所有进程。用该命令结束的进程属于非正常退出。例如：

```
# ps  |  grep  man
11135 pts/4      00:00:00 man
11180 pts/4      00:00:00 man
# killall  -9  man
[1]-  已杀死                    man ls
[2]+  已杀死                    man kill
# ps  |  grep man
#
```

7.3.4　alias 和 unalias 命令

alias 用于设定别名，可以用一个自定义的字符串来代替一个完整的命令行，从而减少打字工作量。unalias 用于取消指定的别名。不带任何参数直接执行 alias 命令，显示已经设定的别名。例如：

```
# alias                                  //显示已经设定的别名
alias cp='cp -i'
alias l.='ls -d .* --color=tty'
alias ll='ls -l --color=tty'
alias ls='ls --color=tty'
alias mv='mv -i'
alias rm='rm -i'
alias which='alias | /usr/bin/which --tty-only --read-alias --show-dot --show-tilde'
# alias list='ls /myshare'              //设定新的别名：用 list 字符串代替 “ls /myshare ”命令
# alias                                  //查看是否设定成功
alias cp='cp -i'
alias l.='ls -d .* --color=tty'
alias list='ls /myshare'                //已经设定
alias ll='ls -l --color=tty'
alias ls='ls --color=tty'
alias mv='mv -i'
alias rm='rm -i'
alias which='alias | /usr/bin/which --tty-only --read-alias --show-dot --show-tilde'
# list                                   //执行 list 命令，显示/myshare 下的目录
cal_txt         finger3_txt   groups_txt    id_txt      newgrp_txt   suple_txt~   w.txt
finger2_txt   finger_txt    groups_txt2   last_txt   suple_txt    who.txt
```

例如，不带参数的 rm 命令用于直接删除文件。通常为了防止误删除，都会采用参数“-i”。命令“rm

-i”在删除每个文件时都会进行询问，增强了安全性。为了简化该命令，可以为“rm　-i”指定别名，重载 rm 命令：

```
#rm　text1
#alias　rm='rm –i'
# rm　text2
rm：是否删除 一般空文件 “text2”？y
```

7.3.5　clear 命令

clear 命令用于清屏，并把光标移到左上角，命令行如下所示：

```
#clear
```

7.3.6　reboot 命令

使用 reboot 命令可以重新引导系统。该命令不会自动将内存中的数据写回硬盘，可能造成数据丢失。命令行如下所示 ：

```
#reboot
```

7.3.7　shutdown 命令

执行 shutdown 命令会把内存中的数据写回硬盘并关闭系统。配合其他参数可以实现系统的重启、关闭。例如把数据写回硬盘后立即重新启动，命令行如下所示：

```
shutdown　-r　now
```

例如，立即关闭系统，并在关闭前进行数据同步，命令行如下所示：

```
shutdown　-h　now
```

7.3.8　&命令

&命令指定程序在后台执行。有时用户需要执行的程序要花很多时间（例如从打印机输出），如果在前台运行，用户将无法进行其他工作。为此利用&将程序放到后台运行。例如在后到执行 gedit 程序，命令行如下所示：

```
#gedit &
```

7.3.9　jobs 命令

job 命令用于显示目前正在后台执行的任务列表，如下所示：

```
#vi　test　&　　//在后台运行 vi test
[1]111200
# man　ls　&　　//在后台运行 man ls
[2] 11428
# jobs　　//查看当前在后台执行的命令
[1]-　Stopped　　vi test
[2]+　Stopped　　man ls
```

7.3.10　fg 命令

fg 命令用于将用户在后台运行的程序移到前台，如下所示：

```
#vi   test   &          //在后台运行 vi test
[1]111200
# man   ls   &          //在后台运行 man ls
[2] 11428
# jobs                 //查看当前在后台执行的命令
[1]-   Stopped                    vi test
[2]+   Stopped                    man ls
# fg   2               //将 man ls 移到前台执行
man ls
# jobs                 //查看仍在后台运行的命令
[1]+   Stopped                    vi test
```

7.3.11　exit 命令

exit 命令用于退出并关闭命令行终端。

7.3.12　halt 命令

halt 命令用于关闭系统。

7.3.13　sync 命令

sync 命令用于将内存中的数据写回磁盘。通常在软盘、U 盘退出系统之前使用该命令，以确保内存中的数据已写入磁盘，避免不同步现象发生。事实上，命令“shutdonw -r now”与 命令“sync”＋“reboot”等效；命令“shutdown　-h　now ”与“sync”+“halt”等效。

7.3.14　mknod 命令

mknod 命令可以用来建立块设备或字符设备文件。mknod 的命令格式为：

```
mknod [OPTION]... NAME TYPE [MAJOR MINOR]
```

其中设备类型为 c 表示字符设备，设备类型为 b 表示块设备。在 Red Hat Enterprise Linux 5 下所有设备都放在/dev 目录下，通过 ls 命令可以查看设备文件特征：

```
#ls   -al   /dev
…   …
brw-rw----   1 root floppy     2,      0   07-25 05:40 fd0              //软盘
…   …
brw-rw----   1 root disk      22,      0   07-25 05:40 hdc              //IDE 硬盘
…   …
brw-r-----    1 root disk       8,      1   07-25 05:41 sda1             //SCSI 硬盘
…   …
crw-rw----   1 root root       4,      0   07-25 05:39 tty0            //字符终端 0
crw-------    1 root root       4,      1   07-25 05:44 tty1            //字符终端 1
```

```
…　…
```

其中第一个字段表示设备文件的访问权限，第一个字段的第一个字符表示设备类型；第五个字段是设备的“主设备号”，第六个字段是“辅设备号”。

通常 PC 机有两个 IDE 接口，每个 IDE 接口可以接两个 IDE 设备。其中 IDE 接口 1 的主设备为“/dev/hda”，从设备为“/dev/hdb”；IDE 接口 2 的主设备为“/dev/hdc”，从设备为“/dev/hdd”。对于 SCSI 硬盘，“/dev/sda”表示第一块 SCSI 设备，“/dev/sdb”表示第二块 SCSI 设备。

在 Linux 系统中每个物理硬盘可以建立 64 个分区，但在缺省情下，只能用 fdisk 程序分出 16 个分区。如果需要大于 16 个的分区，可以使用 mknod 进行创建。例如，建立 SCSI 硬盘/dev/sda 的第 17、18 个分区的设备文件，即/dev/sda17 和/dev/sda18，设备类型是 b，命令行如下：

```
#mknod   /dev/sda17   b   8   17
#mknod   /dev/sda18   b   8   18
…　…
```

7.3.15　chattr 命令

chattr 命令是 ext3（或 ext2）文件系统特有的安全机制，用于设置一个文件的 immutable 属性，即 i 属性。只有根用户有权为文件设置该属性。当 i 属性被设置后，任何用户（包括根用户）都将无法对该文件进行修改、删除或重命名，除非根用户通过命令清除该文件的 i 属性。通常利用 chattr 命令可以将密码文件设为不可更改，以防被恶意破坏。例如：

```
# chattr   +i   /etc/passwd
# tail   -1   /etc/passwd
student3:x:505:506::/home/student3:/bin/bash
# echo   "student4:x:551:551::/home/student4:/bin/bash"   >> /etc/passwd      //试图向 passwd 文件中注入数据
bash: /etc/passwd:  权限不够
```

可以看到文件/etc/passwd 设置 i 属性后，即使是根用户试图修改该文件，也会显示“权限不够”。清除该属性需使用选项“-i”，命令行如下：

```
# chattr   -i   /etc/passwd
# echo   "student4:x:551:551::/home/student4:/bin/bash"   >> /etc/passwd
#tail   -1   /etc/passwd
student4:x:551:551::/home/student4:/bin/bash                  //修改成功
#
```

i 属性不会在 ls 输出中显示，可以使用 lsattr 命令显示 i 属性：

```
# lsattr /etc/passwd
----i-------- /etc/passwd
```

7.3.16　echo 命令

echo 命令用于将命令行中的字符串显示在屏幕上。例如：

```
$ echo   city dalian
city dalian
```

7.3.17　wc 命令

任何一个文本文件都由行、单词和字符组成，使用 wc 命令可以对文本文件的这些基本信息进行统

计。wc 命令格式为：

```
wc [-l][-w][-c]
```

- -l：显示文件的行数。
- -w：显示文件中包含的单词数。
- -c：显示文件中包含的字符数。

例如查看文本文件 test 的基本信息：

```
# cat   test
This is my first document.
I am a good writer.
# wc   test
 2 10 47 test
```

可以看出 test 文件包含 2 行、10 个单词、47 个字符。

7.4　软盘操作命令集 mtools

在 Red Hat Enterprise Linux 中为了方便用户使用软盘，专门提供了 mtools 工具集。mtools 工具集由一组以字母“m”开头的命令组成，其中绝大多数命令格式与对应的 DOS 命令兼容。用户只需要在相应的 DOS 命令前加“m”，就可以像在 DOS 中一样操作软盘，无需再对软盘进行挂载和卸载。mtools 命令集只能对 FAT 格式的软盘进行操作，包括的命令如下所示：

```
# mtools
Supported commands:
mattrib, mbadblocks, mcat, mcd, mclasserase, mcopy, mdel, mdeltree
mdir, mdoctorfat, mdu, mformat, minfo, mlabel, mmd, mmount
mpartition, mrd, mread, mmove, mren, mshowfat, mtoolstest, mtype
mwrite, mzip
```

表 7-6 列出了 mtools 工具集中的命令说明。

表 7-6 mtools工具集及说明

mtools命令	对应的DOS命令	说明
mdir	dir	列目录
mcopy	copy	复制文件
mdel	del	删除文件
mmd	md	创建目录
mformat	format	磁盘格式化
mtype	type	显示文件内容
mrd	rd	删除目录，只能删除空目录，不能删除文件
mmove	move	移动文件
mren	ren	重命名
mattrib	attrib	显示属性
mcomp	comp	对两个文件进行比较
mcd	cd	改变当前所在目录
mbadblock	chkdsk	软盘检测，并在fat表中对坏块进行标识
mshowfat	showfat	显示软盘中分配给指定文件的簇
mtoolstest	toolstest	测试mtools设置
mlable	lable	设置卷标
mdeltree	deltree	删除目录树，连同目录下的子目录及文件一并删除

mmount		挂载软盘
mdu		显示文件目录所占空间
mtools		显示mtools命令集
mzip		改变Zip/Jaz驱动器保护模式，弹出磁盘
mcheck		用于检测指定软盘中的文件是否正确

例如列出软盘的目录，命令行为：

```
#mdir  a:
Volume in drive A has no label
Volume Serial Number is 3D91-2134
Directory for A:/
passwd   bak 223 08-26-2007 21:55
shadow        255 08-26-2007 21:55
    2files 478 bytes
  1 460 459 bytes free
```

例如将上例中的 shadow 文件进行复制，复制到新创建的目录 private 下，命令行为：

```
#mmd  a: private              //创建目录 private
#mcopy  a:shadow  a:private/shadow              //复制
```

例如删除软盘中 passwd.bak 文件，命令行为：

```
#mdel  a: passwd.bak
```

7.5　Linux 与 DOS 常用命令比较

Linux 环境中的命令与 DOS 下的命令具有许多相似性，有些则完全相同。表 7-7 例出了一些常用命令在 MS-DOS 和 Linux 中的的差异，并分别给出了简单的示例。

表 7-7 Linux与DOS常用命令比较

命令类型	MS-DOS	Linux	Linux示例	MS-DOS示例
复制文件	copy	cp	cp /home/teacher1/file1　/tmp	copy c:\teacher1\file1　d:\tmp
转移文件	move	mv	mv file1.doc　/home/teacher	move file1.doc　c:\teacher1\
删除文件	del	rm	rm /home/teacher1/example.doc	del c:\teacher1\example.doc
编辑文件	edit	gedit	gedit　/home/ example.doc	edit c:\example.doc
比较文件内容	fc	diff	diff　filel　file2	fc file1　file2
在文件中找字串	find	grep	grep ‘passwd’　file1.doc	find “username”　file1.doc
格式化软盘	format a:	mkfs	mkfs /dev/fd0	formant a:
列举文件	dir	ls	ls	dir
清除屏幕	cls	clear	clear	cls
退出shell	exit	exit	exit	exit
显示或设置日期	date	date	date	date
显示命令帮助	/?	man	man　command1	command1　/?
创建目录	mkdir或md	mkdir	mkdir directory1	mkdir directory1
查看文件	more	less	less file1.txt	more file1.txt
显示时间	time	date	date	time
显示内存情况	mem	free	free	mem
重新命令文件	ren	mv	mv filel.txt　file2.txt	ren file1.txt　file2.txt
显示当前位置	chdir	pwd	pwd	chdir

7.6　使用 mount 命令挂载外设

软盘、光盘、USB 存储器（包括 U 盘、移动硬盘等）是 Linux 系统重要的外部设备。在 Linux 中外部设备是被当作文件来使用的，Red Hat Enterprise Linux 系统一般会自动对其进行识别并挂载。当自动识别失败或是用户需要手工设定挂载参数时，可以使用 mount 命令来协助完成。

mount 命令实际上是用于挂载指定的文件系统，其命令格式如下：

```
mount [选项] [设备名] [挂载点]
```

- -r：表示挂载的文件系统为只读
- -a：表示挂载/etc/fstab 文件中列出的所有文件系统
- -A：卸载当前已被挂载的所有文件系统
- -h：显示帮助信息
- -t： 挂载的文件系统类型。

其中[设备名]指需要挂载的设备名称，[挂载点]是一个已存在的空目录，被挂载设备的根将指定在该目录位置。挂载点可以指定目录树中的任何位置，但必须是空目录。选项“-t”指定了挂载的文件系统类型，mount 命令可以挂载的文件系统类型及说明见表 7-8。

表 7-8 mount命令可挂载的文件系统类型说明

可挂载的文件系统类型	说明
auto	自动检测文件系统并进行挂载
ext2	Linux中使用最多的文件系统
ext3	ext2基础上的日志文件系统
iso9660	CDROM标准文件系统
hpfs	OS/2所使用的高性能文件系统，但在Linux系统中只能作为只读文件系统使用
minix	Linux的早期版本所采用的文件系统，该文件系统分区不能超过64M，一般只用与软盘或RAMDISK
msdos	MSDOS所使用的文件系统，不支持长文件名
nfs	网络文件系统
ntfs	Windows NT所使用的文件系统
vfat	扩展的MSDOS文件系统，支持长文件名，被Windows95/98/NT/2000所采用

如果不带任何选项使用 mount 命令，默认会显示目前已挂载的文件系统，例如：

```
# mount
/dev/mapper/VolGroup00-LogVol00 on / type ext3 (rw)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
devpts on /dev/pts type devpts (rw,gid=5,mode=620)
/dev/sda1 on /boot type ext3 (rw)
tmpfs on /dev/shm type tmpfs (rw)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw)
```

使用 umount 命令可以卸载已挂装的文件系统，但不能卸载根分区。umount 命令格式为：

```
umount [-hV]
umount -a [-dflnrv] [-t vfstype] [-O options]
umount [-dflnrv] dir | device [...]
```

- -V：显示版本信息并退出。
- -h：显示帮助信息并退出。

- -v：Verbose 模式。
- -n：卸载时的信息不写入/etc/mtab 文件。
- -r：如果卸载失败，尝试以只读方式挂载.。
- -a：所有在 /etc/mtab 文件中描述的设备均被卸载。
- （对于 2.7 以上版本，proc 文件系统不会被卸载。
- -t：指定卸载的文件系统类型，umount 命令会只对指定类型的文件系统进行卸载。
- -f：强制卸载。

7.6.1　软磁盘的挂载

使用 mount 命令可以对软盘进行挂载，并指定挂载的目录。例如将软盘挂载到/mnt/floppy 目录下，如果/mnt/floppy 目录不存在，则需要提前创建，命令行为：

```
#mkdir   /mnt/floppy
#mount   /dev/fd0   /mnt/floppy
```

使用 umount 命令可以卸载软磁盘，命令行为：

```
#umount   /dev/fd0
```

也可用挂载点作为参数进行卸载，命令行为：

```
#umount   /mnt/floppy
```

7.6.2　CDROM 的挂载

CDROM 的挂载与软盘的挂载方法类似，命令格式为：

```
mount   -t   iso9660   /dev/cdrom  光盘挂载点
```

其中 iso9660 是光盘的标准文件系统类型。例如在/mnt/cdrom 目录下挂载光驱，如果/mnt/cdrom 目录不存在，则需要提前创建，命令行为：

```
#mkdir   /mnt/cdrom
#mount   -t   iso9660   /dev/cdrom   /mnt/cdrom
```

通常光盘被挂载后，如果不进行卸载，光驱是无法打开的。必须卸载已经安装的光盘文件系统后，才可以顺利取出光盘。卸载光驱时使用 umount 命令，命令格式为：

```
umount   [光盘挂载点或光盘设备文件名]
```

例如对上例中挂载的光驱进行卸载，命令行为：

```
umount   /dev/cdrom
```

也可以利用挂载点作为参数进行卸载，命令行为：

```
umount   /mnt/cdrom
```

在 Linux 中，系统还提供了专门用于弹出或关闭光盘驱动器的命令 eject。ejetc 命令会先调用 umount 命令对光驱进行卸载，然后弹出光盘。eject 命令格式如下：

```
eject   [-t] [-n] [光盘挂载点或光盘设备文件名]
```

其中选项“-t”用于关闭指定的光盘驱动器，不会将光驱弹出。直接使用 eject 命令而不指定任何参数，将弹出默认的设备。可以使用选项“-n”查看系统默认的弹出设备，例如：

```
#eject   -n
eject: device is `/dev/hdc`
```

7.6.3　USB 存储设备的挂载

随着硬件成本的不断降低，USB 存储设备由于具有存取速度快、稳定性高、即插即用、携带方便等优点越来越得到广泛的应用和发展。无论是在 Windows 系统还是在 Linux 系统中均已提供了对该类设备的支持。通常使用的 USB 存储设备包括 U 盘和移动硬盘，一般不需要安装驱动程序即可以在 Linux 系统中使用。但由于 USB 存储设备的标准还不是很统一，Linux 不能保证对所有的 USB 存储设备都能正确识别。

在 Linux 中，USB 存储设备通常是作为 SCSI 设备来使用的。SCSI 设备文件名以“sd”开头，后接设备序号（按字母排序）及分区号（按数字排序），例如第一个 SCSI 设备上的第一个主分区表示为“/dev/sda1”，第三个 SCSI 设备的第二个主分区表示为“/dev/sdc2”。

对于 U 盘，如果 U 盘中没有进行分区则使用 SCSI 设备的设备名直接进行挂载，如果 U 盘中存在分区则使用带分区的 SCSI 设备名进行挂载，例如将 U 盘挂载到/mnt/flash 目录下，命令行为：

```
#mkdir   /mnt/flash
# mount   -t   vfat   /dev/sdb   /mnt/flash
```

卸载时使用 umount 命令，命令行为：

```
#umount   /dev/sdb
```

对于 USB 移动硬盘，在使用 mount 命令进行挂载时不能对整个硬盘设备进行挂载，必须指定相应的分区，例如将移动硬盘第一个分区挂载到/mnt/disk 目录下，命令行为：

```
#mkdir   /mnt/disk
# mount   -t   vfat   /dev/sdc1   /mnt/disk
```

卸载可以使用命令 umount，如下所示：

```
#umount   /dev/sdc1
```

7.7　Linux 备份与压缩的策略

没有任何系统是绝对可靠的，防止数据丢失最切实可行的方法是定期进行数据备份。在 Linux 系统中，对系统目录的备份是一种有效的保护手段，但并不是所有目录都需要备份，表 7-8 列出了系统目录备份的一种常规建议。

表 7-8 备份系统目录

1.必须备份或者需要频繁备份的目录	
/etc	系统配置文件存放的目录
/home	用户的目录空间
/root	根用户的主目录
/var	日志、邮件、打印机队列等存放的目录
/var/www	www服务器使用的目录
/var/ftp	ftp服务器使用的目录
2.不需要经常备份或只在全系统备份时才需要备份的目录	
/bin	存放操作系统中的可执行程序（主要是一些命令）的目录
/boot	存放引导文件、内核文件的目录
/dev	存放设备文件的目录（包括硬盘、软盘等）
/lib	存放库文件、pam和内核模块
/usr	存放用户使用的系统命令和应用程序等信息的目录
/tmp	临时目录
/opt	第三方和FHS软件存放目录

/lost+found	文件系统中补损坏或未正常链接的文件存放目录
/sbin	存放系统启动时所需要执行的程序
3.不需要备份或很少备份的目录	
/mnt	文件系统的挂载点
/proc	存放内核、操作系统的动态信息

由于现在的应用程序及文件普遍越来越大，为了节省磁盘空间、减少网络传输代价，在备份过程中一般采用了压缩技术。通常压缩与备份是同步进行的，常用的命令包括 tar、zip、gzip 等。

7.8　打包程序 tar

tar 在各 UNIX 版本中受到了广泛的支持，有着非常久远的历史。tar 是 tape archive 的缩写，最早是与磁带机联系在一起的，用于将系统中需要备份的数据打包归档到磁带中，在需要时再把备份的数据从磁带中恢复回来。随着计算机硬盘容量不断增大，CD-R 及移动磁盘的广泛使用，tar 命令已不仅仅局限在磁带机，更多地应用在磁盘备份中。tar 命令本身只进行打包不进行压缩，主要功能是将多个文件或目录打包在一个文件里，便于传输和保存。为了减少备份文件的大小，节省存储空间，tar 命令经常和许多压缩选项配合使用。tar 的一般格式为：

```
tar [选项]　备份后的文件名.tar　备份的文件或目录
```

- c 或-c：创建新的备份文件。
- v 或-v：Verbose 模式，即显示命令执行时的信息。
- f 或-f：　指定压缩的文件格式。
- x 或-x：对文件进行恢复。
- Z 或-Z：指定压缩为.Z 格式。
- z 或-z：指定压缩为.gz 格式。
- t 或-t：　查询包中内容。

tar 命令选项一共有 70 多个，上面是常用的几个选项，各选项可以配合在一起使用。

7.8.1　打包和解包的常规操作

例如对当前目录下的所有文件和目录进行打包，生成 example.tar 备份文件：

```
# ls
directory1   file1   file2   file3
# tar   cvf   example.tar *
directory1/
directory1/file4
file1
file2
file3
# ls
directory1   example.tar   file1   file2   file3
```

可以看到已经生成了 example.tar 打包文件，要解开此文件只需将选项“c”改为“x”即可：

```
# tar   xvf   example.tar
directory1/
```

```
directory1/file4
file1
file2
file3
```

在打包的时候如果使用了绝对路径，恢复时 tar 会自动将文件或目录恢复到原来的路径下，如果路径不存在就重新创建。为了避免这种情况发生，打包时应尽可能进到子目录下，然后再执行 tar 命令，例如：

```
$cd /myshare
$tar cvf   /dev/sda1 *
```

tar 命令不支持分卷，不具有磁盘修复功能，但可以用 tar 将目录打包成一个文件，例如：

```
# ls   directory1/
file4
# tar cvf directory1.tar directory1/
directory1/
directory1/file4
# ls
directory1   directory1.tar
```

7.8.2　查看 tar 包中的内容

可以使用选项“-tf”查看包中的内容，如下所示：

```
# tar   -tf   directory1.tar
directory1/
directory1/file4
```

7.8.3　打包链接文件

对于链接文件，tar 命令只打包链接，不会对源文件打包。如果需要对源文件打包，必须使用选项“-h”。例如当前目录下有 file1 和 file1_ln 两个文件，file1_ln 是 file1 文件的链接文件：

```
# ls   -al
drwxr-xr-x 3 root root   4096 08-08 20:46 .
drwxr-xr-x 5 root root   4096 08-08 20:34 ..
-rw-r--r-- 1 root root      0 08-08 20:16 file1
lrwxrwxrwx 1 root root      5 08-08 20:46 file1_ln -> file1
# tar   cvf   test1.tar *
file1
file1_ln
# tar   tvf   test1.tar
-rw-r--r-- root/root             0 2007-08-08 20:16:34 file1
lrwxrwxrwx root/root             0 2007-08-08 20:46:04 file1_ln -> file1
```

由于没有使用“-h”选项，上例中只对链接本身进行了打包，下面使用“-h”选项，对链接源文件进行打包：

```
# tar   hcvf   test1.tar *
file1
file1_ln
# tar   tvf   test1.tar
-rw-r--r-- root/root             0 2007-08-08 20:16:34 file1
```

```
-rw-r--r-- root/root          0 2007-08-08 20:16:34 file1_ln
```

7.8.4　向包中添加新文件

对于打包后的文件，如果要在包中增加新的文件，只需使用“-r”选项，无需重新打包全部文件。例如在上例生成的 test1.tar 中加入新的文件 file2，命令行如下：

```
# tar  rvf  test1.tar  file2
file2
# tar  tvf  test1.tar
-rw-r--r-- root/root          0 2007-08-08 20:16:34 file1
-rw-r--r-- root/root          0 2007-08-08 20:16:34 file1_ln
-rw-r--r-- root/root          0 2007-08-08 20:16:39 file2
```

7.8.5　生成.tar.gz 压缩包

在 Linux 系统中，gzip 程序可以用来实现压缩，生成“.gz”为后缀的压缩包。在使用 tar 命令进行打包的同时，配合使用“z”选项，也可以同步生成“.gz”的压缩包。例如将当前目录下的所有文件和目录打包并压缩，保存为 tmp.tar.gz 文件，命令行为：

```
# ls
directory1  file1       file2     file3
# tar  zcvf  tmp.tar.gz  *      //对当前目录进行压缩
directory1/
directory1/file4
file1
file2
file3
# ls              //可以看到已生成 tmp.tar.gz 压缩包
directory1    file1       file2    file3      tmp.tar.gz
```

使用“ztf”选项可以查看压缩包中的内容，命令如下：

```
# tar  ztf  tmp.tar.gz
directory1/
directory1/file4
file1
file2
file3
```

使用“zxvf”选项可以对后缀为“.tar.gz”的压缩包解压缩，命令行如下：

```
# tar  zxvf  tmp.tar.gz
directory1/
directory1/file4
file1
file2
file3
```

在上例中也可以对 tmp.tar.gz 文件先用 gzip 命令解压，然后再使用 tar 命令解包，同样可以完成压缩包的展开。

7.9　压缩程序 gzip 与 gunzip

在 Linux 中一种非常流行的压缩格式是“.gz”，该格式的压缩文件是由 gzip 程序生成，由 gunzip 程序来解压缩。gzip 具有较高的压缩率，但只能逐个生成压缩文件，无法将多个文件压缩并打包成一个文件，所以 gzip 经常和 tar 命令配合使用。即先用 tar 对多个文件打包，然后再用 gzip 进行压缩，通常会生成“.tar.gz”或“.tgz”后缀的文件。gzip 的命令格式如下：

```
gzip [-cdfhlLnNrtvV19] [-S suffix] [文件名 ...]
```

gzip 的选项比较多，选项说明见表 7-9。

表 7-9 gzip选项说明

选项	说明
-c --stdout	输出到标准输出设备上，原文件内容不变
-d --decompress	解压缩
-f --force	对输出文件强制写覆盖并对链接文件进行压缩f
-h --help	显示帮助信息
-l --list	列出压缩包的内容
-L --license	显示软件许可权
-n --no-name	不保存或恢复原文件的文件名和时间戳
-N --name	保存或恢复原文件的文件名和时间戳
-q --quiet	禁止警告
-r --recursive	对目录进行递归操作
-S .suf　--suffix .suf	使用自定义的压缩文件后缀名
-t --test	检测压缩包的完整性
-v --verbose	verbose 模式
-V --version	显示版本号
-1 --fast	快速压缩
-9 --best	最佳压缩（压缩率最高）

可以直接使用 gzip 命令对文件进行压缩，但不可以对目录进行压缩。压缩后的文件以原文件名为主文件名，以“.gz”为后缀，同时会自动删除原文件。

7.9.1　常规压缩与解压缩操作

例如当前目录下有 file1、file2 文件，对 file1 文件进行压缩，命令行为：

```
# ls
  file1   file2
# gzip   file1
# ls
  file1.gz   file2
```

也可以同时对多个文件进行压缩，例如：

```
# ls
file1.gz   file2   file3
# gzip   file2   file3
# ls
file1.gz   file2.gz   file3.gz
```

解压使用 gunzip 命令，解压后原压缩文件会自动删除，例如对 file1.gz 文件解压缩：

```
# gunzip   file1.gz
```

```
# ls
file1   file2.gz   file3.gz
```

也可以对多个文件一起解压缩，例如对 file2.gz、file3.gz 解压缩：

```
# gunzip   file2.gz   file3.gz
# ls
file1   file2   file3
```

使用选项“-d”，也可以使用 gzip 命令完成对文件的解压缩，例如：

```
# ls
file1.gz   file2.gz   file3.gz
#gzip   -d   *.gz
# ls
file1   file2   file3
```

7.9.2　查看.gz 压缩包中的内容

gzip 使用选项“-l”可以在未解压状态下查看压缩文件的基本信息，包括已压缩字节、未压缩字节、压缩率、压缩前文件名，例如：

```
# ls
file1.gz   file2.gz   file3.gz
# gzip   -l   *
gzip: directory1 is a directory -- ignored
         compressed        uncompressed   ratio uncompressed_name
                 32                   6 -33.3% file1
                 26                   0     0.0% file2
                 26                   0     0.0% file3
                 84                   6 -900.0% (totals)
```

7.9.3　自定义压缩包后缀

使用 gzip 压缩文件后，默认生成的后缀为“.gz”，但配合使用选项“-S”，可以自定义压缩文件的后缀名。例如对当前目录下的 file1 文件进行压缩，且自定义后缀名为“.teacher”，命令行为：

```
# ls
file1   file2   file3
# gzip   -S   .teacher   file1
# ls
file1.teacher   file2   file3
```

在解压时也必须说明自定义的后缀名：

```
# gunzip   -S   .teacher   file1.teacher
# ls
file1   file2   file3
```

也可以使用 gzip 命令解压：

```
#gzip   -S   .teacher   -d   file1.teacher
```

7.10　压缩程序 zip 与 unzip

zip 是 Linux 下另一个广泛使用的压缩程序。使用 zip 压缩后的文件名后缀是“.zip”， 解压缩时使用 unzip 命令。

7.10.1　使用 zip 生成压缩文件

zip 格式的文件在 Windows 系统中也被广泛使用，Windows 下著名的 winzip 程序就是用来专门处理 zip 文件的工具。zip 的命令格式如下：

```
zip [-选项] [-t mmddyyyy] [-n 后缀列表] [zip 文件] [源文件...]
```

如果没有指定源文件和 zip 文件 ，系统默认对标准输入进行压缩，并从标准输出导出压缩。zip 的选项说明见表 7-10。zip 文件如果没有指定后缀，则默认后缀为“.zip”。

表 7-10 zip的选项说明

选项	说明
-f	仅对变化的文件进行更新
-u	只有文件被改动或有新文件时才进行升级
-d	删除zip文件中的条目
-r	子目录递归
-l	将LF转换为CR LF
-9	最佳压缩（压缩率最高）
-q	静默模式
-v	verbose 模式，打印版本信息
-z	给zip压缩文件加注释
-@	从标准输入读入文件名
-x	不包括列出的文件
-i	包括列出的文件
-j	不包括子目录
-A	调整为自展开的exe文件
-T	检测文件的完整性
-y	对于链接文件仅保存符号链接
-e	加密
-k	用8.3格式转换zip文件中的文件名，并且文件名改为大写
-m	创建压缩文件后，自动删除源文件
-n	直接放入压缩包，不进行压缩

1. 常规压缩操作

例如对当前目录下的文件 file1、file2、file3 进行压缩，压缩后生成 file.zip 文件，命令行如下所示：

```
# ls
directory1   file1   file2   file3
# zip file.zip file1 file2 file3
  adding: file1 (deflated 74%)
  adding: file2 (deflated 40%)
  adding: file3 (deflated 39%)
# ls
directory1   file1   file2   file3   file.zip
```

在对目录进行压缩时，如果目录中存在子目录，则 zip 默认只将目录名放入压缩包，并不进入子目

录进行压缩，例如：

```
# ls
directory1   file1   file2   file3
# zip   filezip.zip   *
   adding: directory1/ (stored 0%)
   adding: file1 (deflated 74%)
   adding: file2 (deflated 40%)
   adding: file3 (deflated 39%)
# ls
directory1   file1   file2   file3   filezip.zip
# ls -l
drwxr-xr-x 2 root root 4096 08-08 20:17 directory1
-rw-r--r-- 1 root root     73 08-09 02:33 file1
-rw-r--r-- 1 root root     72 08-09 02:33 file2
-rw-r--r-- 1 root root     77 08-09 02:34 file3
-rw-r--r-- 1 root root    623 08-09 02:48 filezip.zip
```

2. 连同子目录一并压缩

如果希望连同子目录中的文件一并进行压缩，需使用选项“-r”。接上例，对当前目录中的所有文件、子目录及子目录下的文件一并进行压缩，如下所示：

```
# ls
directory1   file1   file2   file3
# zip   -r   filezip.zip *
   adding: directory1/ (stored 0%)
   adding: directory1/file4 (deflated 22%)
   adding: file1 (deflated 74%)
   adding: file2 (deflated 40%)
   adding: file3 (deflated 39%)
# ls   -l
drwxr-xr-x 2 root root 4096 08-08 20:17 directory1
-rw-r--r-- 1 root root     73 08-09 02:33 file1
-rw-r--r-- 1 root root     72 08-09 02:33 file2
-rw-r--r-- 1 root root     77 08-09 02:34 file3
-rw-r--r-- 1 root root    800 08-09 02:49 filezip.zip
```

3. 对指定目录中的文件进行压缩

如果只需要对指定目录中的文件进行压缩，忽略子目录，需要使用选项“-j”，命令行如下所示：

```
# ls
directory1   file1   file2   file3
# zip   -j   filezip.zip *
   adding: file1 (deflated 74%)
   adding: file2 (deflated 40%)
   adding: file3 (deflated 39%)
# ls   -l
drwxr-xr-x 2 root root 4096 08-08 20:17 directory1
-rw-r--r-- 1 root root     73 08-09 02:33 file1
-rw-r--r-- 1 root root     72 08-09 02:33 file2
-rw-r--r-- 1 root root     77 08-09 02:34 file3
-rw-r--r-- 1 root root    491 08-09 02:49 filezip.zip
```

4. 生成兼容压缩包

有时 zip 压缩文件需要在其他平台上解压缩，为了保持和其他平台的兼容性，最好使用比较通用的 8.3 命名格式来重新对文件命名，然后再进行压缩。可以使用选项“-k”完成转换。选项“-k”除了将文件名改为 8.3 格式外，还会将文件名调整为大写。该选项只影响压缩包内的文件名，对于源文件名不做更改。例如：

```
# ls
directory1   file000000000003.document   file1   file2
# zip   -k   file3.zip *
   adding: DIRECTOR/ (stored 0%)
   adding: FILE0000.DOC (deflated 39%)
   adding: FILE1 (deflated 74%)
   adding: FILE2 (deflated 40%)
# ls
directory1   file000000000003.document   file1   file2   file3.zip
```

注意：8.3 格式是文件名的常用命名方法之一。其中文件名分为主文件名和扩展文件名两部分，主文件名最长 8 字符，扩展文件名最长 3 字符。通常也把文件的扩展文件名称为文件的后缀。

5. 只打包而不进行压缩

对于 JPG，GIF，RMVB，MPEG 等格式的媒体文件，由于其自身已经采用了一定的压缩技术，如果再使用 zip 进行压缩，效果不大而且浪费时间。可以使用“-n”选项，对该类文件只打包，不进行压缩。文件类型之间需用冒号分隔。例如对当前目录下的文件进行压缩，忽略对“.JPG”、“.GIF”格式文件的压缩，命令行如下所示：

```
# ls
directory1   DSC03191.JPG   DSC03192.JPG   DSC06479.JPG   DSC09.GIF   file1
# zip   -n   .JPG : .GIF   files.zip *
          zip warning: name not matched: .GIF
          zip warning: name not matched: files.zip
   adding: directory1/ (stored 0%)
   adding: DSC03191.JPG (stored 0%)          //直接打包不压缩
   adding: DSC03192.JPG (stored 0%)          //直接打包不压缩
   adding: DSC06479.JPG (stored 0%)          //直接打包不压缩
   adding: DSC09.GIF (deflated 74%)          //直接打包不压缩
   adding: file1 (deflated 40%)
```

6. 从标准输入读入压缩文件名

选项“-@”允许从标准输入设备读入文件名。如果需要压缩的文件很多，无法在一行列出所有文件名，可以使用该选项，输入完毕按 Ctrl+d 结束。例如：

```
# zip   -@   filezip.zip
file1
file2
file3
   adding: file1 (deflated 74%)
   adding: file2 (deflated 40%)
   adding: file3 (deflated 39%)
```

7. 排除无需压缩的文件

如果某一目录下需要压缩的文件占绝大多数，只需在选择源文件时使用选项“-x”，将个别不需要压缩的文件排除即可，例如：

```
# ls
file1   file2   file3
# zip   filezip.zip   *    -x   file3   //跳过 file3
   adding: file1 (deflated 74%)
   adding: file2 (deflated 40%)
```

8. 对日期进行指定

如果需要压缩某一日期之后的文件，可以使用选项“-t”对日期进行设定。日期格式为 mmddyy，其中 mm 表示月份，dd 表示日，yy 表示年的后两位。例如在当前目录下，对 2007 年 8 月 9 日之后创建的文件进行压缩，可以使用命令：

```
#zip   -t   080907   files.zip   *
```

9. 最佳压缩

选项“-9”表示最佳压缩，即压缩率最高。zip 的压缩率共分 9 个等级，其中 1 级最低，即压缩速度快，但节省存储空间少；9 级最高，即压缩速度慢，但可节省较多存储空间。zip 默认的压缩等级为 6，下面是对不对压缩等级，压缩率的比较：

```
# zip   -9   test1.zip   file1
adding: file1 (deflated 40%)
# zip   -1   test2.zip   file1
adding: file2 (deflated 36%)
```

10. 压缩符号链接

对于符号链接文件，zip 在压缩前会自动寻找该链接指向的源文件，并对源文件进行压缩，压缩后链接关系不再存在，例如：

```
# ls -l
-rw-r--r-- 1 root root     25 08-09 04:07 file1
lrwxrwxrwx 1 root root      5 08-09 04:20 file1_ln -> file1
# zip   file1.zip   file1   file1_ln                          //对 file1 及 file1 的链接文件 file1_ln 进行压缩
   adding: file1 (deflated 40%)
   adding: file1_ln (deflated 40%)
# unzip file1.zip                                            //对 file1.zip 文件解压
Archive:   file1.zip
   inflating: file1
   inflating: file1_ln
# ls
file1   file1_ln   file1.zip
# ls -l
-rw-r--r-- 1 root root   25 08-09 04:07 file1
-rw-r--r-- 1 root root   25 08-09 04:07 file1_ln             //已不再是链接文件
-rw-r--r-- 1 root root 298 08-09 04:21 file1.zip
```

如果只希望对链接文件自身进行压缩，解压后仍然保持其链接属性，可以使用选项“-y”，例如：

```
# zip   -y   file1.zip   file1   file1_ln
   adding: file1 (deflated 40%)
   adding: file1_ln (stored 0%)
# unzip   file1.zip
Archive:   file1.zip
   inflating: file1
      linking: file1_ln                -> file1
finishing deferred symbolic links:                          //自动恢复链接关系
```

```
  file1_ln                    -> file1
# ls   -l
-rw-r--r-- 1 root root   25 08-09 04:07 file1
lrwxrwxrwx 1 root root    5 08-09 04:27 file1_ln -> file1          //解压后仍是链接文件
-rw-r--r-- 1 root root 288 08-09 04:25 file1.zip
```

11. 删除源文件

通常在用 zip 生成压缩文件的同时，源文件保持不变。使用选项“-m”，可以在生成压缩文件后，自动删除源文件，进一步节省存储空间。例如对当前目录下的 file1、file2 文件进行压缩，生成 file.zip 文件，同时删除 file1、file2 文件，命令行如下：

```
# ls
  file1   file2   file3
# zip –m file.zip file1 file2
  adding: file1 (deflated 74%)
  adding: file2 (deflated 40%)
# ls   -l
-rw-r--r-- 1 root root    77 08-09 02:34 file3
-rw-r--r-- 1 root root   800 08-09 02:49 file.zip        //文件 file1、file2 已删除
```

7.10.2　使用 unzip 进行解压

在 Linux 系统中，使用 unzip 命令来对 zip 压缩文件进行解压缩。unzip 命令格式为：

```
unzip [选项] zip 文件
```

- -Z：以 zipinfo 格式显示压缩文件内的信息，包括文件数目、已压缩和未压缩字节数、压缩比等。
- -l：以简略格式列出压缩文件的基本信息，包括文件名、修改日期时间、未压缩文件的大小等。如果存在注释，也一并显示。
- -L：如果压缩文件是从不区分大小写的文件系统中创建，例如 MS-DOS 操作系统，则将文件名全部改为小写，并添加“^”前缀。
- -t：通过 CRC 校验对 zip 压缩包进行检测。
- -x：用于排除解压缩包中的特定文件。

1. 常规解压缩操作

例如对当前目录下的 file.zip 压缩文件进行解压缩，命令行如下所示：

```
# unzip file1.zip                  //对 file1.zip 文件解压
Archive:   file1.zip
  inflating: file1
  inflating: file2
# ls
file1   file2   file1.zip
```

2. 排除无需解压的文件

如果希望排除压缩包中的某个特定文件，可以使用选项“-x”，例如：

```
# ls
file1.zip
# unzip file1.zip   -x   file2          //对 file1.zip 文件解压,但不对其中包含的 file2 进行解压
Archive:   file1.zip
  inflating: file1
# ls
```

```
file1   file1.zip
```

3. 以 zipinfo 格式查看压缩包内容

使用“-Z”选项，可以以 zipinfo 格式查看压缩包的内容，命令行如下：

```
# unzip   -Z   files.zip
Archive:   files.zip     11159281 bytes     8 files
drwxr-xr-x   2.3 unx            0 bx stor   8-Aug-07 20:17 directory1/
-rw-r--r--   2.3 unx   2130896 bx defN   2-Aug-07 05:58 DSC03191.JPG
-rw-r--r--   2.3 unx   2145453 bx defN   2-Aug-07 05:58 DSC03192.JPG
-rw-r--r--   2.3 unx   1305640 bx defN   2-Aug-07 05:58 DSC06479.JPG
-rw-r--r--   2.3 unx          73 tx defN   9-Aug-07 02:33 DSC09.GIF
-rw-r--r--   2.3 unx          25 tx defN   9-Aug-07 03:33 file1
drwxr-xr-x   2.3 unx            0 bx stor   9-Aug-07 04:26 test/
-rw-r--r--   2.3 unx   5582827 bx stor   9-Aug-07 03:33 :.zip
8 files, 11164914 bytes uncompressed, 11158237 bytes compressed:   0.1% //由于包含多幅图片，压缩率为 0.1%
```

4. 以简略格式查看压缩包内容

在上例中，如果使用“-l”选项，可以以简略格式查看压缩包的内容，命令行如下：

```
# unzip   -l   files.zip
Archive:   files.zip
  Length       Date     Time      Name
  --------      ----      ----      ----
         0   08-08-07 20:17     directory1/
   2130896   08-02-07 05:58     DSC03191.JPG
   2145453   08-02-07 05:58     DSC03192.JPG
   1305640   08-02-07 05:58     DSC06479.JPG
        73   08-09-07 02:33     DSC09.GIF
        25   08-09-07 03:33     file1
         0   08-09-07 04:26     test/
   5582827   08-09-07 03:33     :.zip
  --------                      -------
  11164914                       8 files
```

7.11　其他常用备份与压缩工具

7.11.1　压缩程序 bzip2 与 bunzip2

与 gzip 类似，bzip2 也是一种常用的压缩工具。bzip2 压缩后的文件一般具有后缀“.bz2”，可以使用 bunzip 对其解压缩。bzip2 不具有将多个文件或目录进行打包的功能，只能单纯地对文件进行压缩。在产生后缀为“.bz2”的压缩文件后，bzip2 默认将自动删除源文件。bzip2 命令格式如下：

```
bzip2 [选项] [源文件...]
```

bzip2 的选项说明如表 7-11 所示。

表 7-11 bzip2 选项说明

选项	说明
-h --help	显示帮助信息
-d --decompress	强制解压缩

-z --compress	强制压缩
-k --keep	保留源文件
-f --force	允许写覆盖
-t --test	检测压缩文件的完整性
-c --stdout	输出到标准输出设备
-q --quiet	静默模式
-v --verbose	verbose模式 （显示提示信息）
-L --license	显示软件许可协议
-V --version	显示软件版本
-s --small	使用较小的内存 （大约2500k）
-1 .. -9	设置压缩等级1～9
--fast	快速压缩（等级1）
--best	最佳压缩（等级9）

1. 常规压缩操作

例如对当前目录下的文件 file1、file2、file3 进行压缩，命令行为：

```
#ls
file1   file2   file3
# bzip2 *
# ls
file1.bz2   file2.bz2   file3.bz2
```

可以看到，在生成 file1.bz2、file2.bz2 和 file3.bz2 压缩文件后，源文件已经自动删除。在压缩过程中没有任何提示，如果希望看到压缩过程中的提示信息，可以使用选项“-v”，例如：

```
# ls
file1   file2   file3
# bzip2   -v   *
  file1:      0.681:1, 11.745 bits/byte, -46.81% saved, 47 in, 69 out.
  file2:      1.170:1,   6.839 bits/byte, 14.52% saved, 62 in, 53 out.
  file3:      1.033:1,   7.742 bits/byte,   3.23% saved, 62 in, 60 out.
# ls
file1.bz2   file2.bz2   file3.bz2
```

2. 压缩但不删除源文件

如果希望在生成压缩文件后，源文件不被删除，可以使用带选项“-k”的 bzip2 命令，如下所示：

```
#ls
file1   file2   file3
# bzip2   *
# ls
file1   file2   file3
file1.bz2   file2.bz2   file3.bz2
```

3. 解压缩操作

对 bzip2 文件解压缩可以使用 bunzip 命令。解压缩后，bzip2 文件也会自动被删除，如下所示：

```
# ls
file1.bz2   file2.bz2   file3.bz2
# bunzip2 *
# ls
file1   file2   file3
```

解压缩也可以使用带选项“-d”的 bzip2 命令，例如对当前目录下的 file1 文件进行压缩和解压缩，命令行为：

```
# bzip2   file1
# ls
file1.bz2
# bzip2   -d   file1.bz2
# ls
file1
```

4. 对损坏 bz2 文件进行恢复

如果 bzip2 文件有损坏，可以试着用 bzip2recover 命令进行恢复，例如对当前目录下的 file.bz2 文件进行修复，命令行为：

```
# ls
file.bz2
# bzip2recover   file.bz2
bzip2recover 1.0.3: extracts blocks from damaged .bz2 files.
bzip2recover: searching for block boundaries ...
    block 1 runs from 80 to 358
bzip2recover: splitting into blocks
    writing block 1 to `rec00001file.bz2' ...
bzip2recover: finished
# ls
file.bz2      rec00001file.bz2              //生成了恢复文件 rec00001file.bz2
# bunzip2 rec00001file.bz2
# ls
file.bz2   rec00001file
# cat rec00001file                         // rec00001file 中记录了 file 文件的内容片段
aaaaaaaaaaaaaaaaaaaaaaaa
bbbbbbbbbbbbbbbbbbbbbbbb
ccccccccccccccccccccccc
ddddddddddddddddddddddddd
eeeeeeeeeeeeeeeeeeeeeeee
fffffffffffffffffffff
```

可以看出，bzip2recover 命令实际是对需要修复的文件按块进行搜索，将每一块保存到一个文件名为“recoverXXXXX.bz2”的文件中。每一个这样的文件，记录了源文件内容的一些片段。

7.11.2　dump 与 restore 命令

dump 命令与 tar 类似，长久以来一直是 UNIX 系统中最为常用的备份工具之一。与 tar 不同，dump 支持分卷和增量备份（所谓增量备份是指仅备份最近一次备份以来修改过的文件，也称差异备份）。restore 命令用来恢复已备份的文件，可以从备份磁带或由 dump 生成的备份文件中恢复原文件。

1. dump 命令

当用 dump 命令制作一个文件系统的备份时，必须通过参数来指定 dump 的级别、备份的介质及需要备份的文件系统。dump 的命令格式为：

```
dump [选项] [参数]  文件系统
```

表 7-12 列出了 dump 命令常用的选项。

表 7-12 dump命令的选项说明

选项	说明

-t date	指定增量备份的基准时间。在这个时间之后修改或增加的任何文件将得到备份。该选项将使dump忽略/etc/dumpdates文件中的设定值。date的格式可以参考手册中的ctime。
-u	将备份的细节记录在/det/dumpdates文件中。
-s feet	指定磁带以英尺为单位长度。
-n	当dump操作需要用户参与，例如更换磁带，dump就向该用户组的所有成员发出提示信息。
-d density	设定磁带的密度，默认每英寸1600bit。
-f file	指定dump所写入的文件或设备。
-b kbperdump	指定每个dump记录的KB数。
-B records	指定每个卷包含的dump记录数，该记录数表示一盘磁带能装载的数据容量。
0-9	dump级别，分为0～9级，其中级别0表示备份所有文件，1～9表示增量备份。对于不为0的备份等级，表示仅对上次同等级或更低级别的备份后修改过的文件进行备份。
-W	列出需要备份的文件系统。
-w	列出需要备份的单个文件。

例如使用 dump 命令将第一个 IDE 硬盘上第一个文件系统（/dev/hda1）完全备份（level 0）到磁带机上，命令行为：

```
#dump   -ouf   /dev/tape   /dev/hda1
```

使用 dump 命令也可以直接实现对目录的备份。例如对当前目录下的 zip 子目录进行备份，备份文件名为 dumpzip，命令行为：

```
# dump   -f   dumpzip   zip
  DUMP: Date of this level   dump: Thu Aug   9 09:18:33 2007
  DUMP: Dumping /dev/mapper/VolGroup00-LogVol00 (/ (dir myshare/zip)) to dumpzip
  DUMP: Label: none
  DUMP: Writing 10 Kilobyte records
  DUMP: mapping (Pass I) [regular files]
  DUMP: mapping (Pass II) [directories]
  DUMP: estimated 522 blocks.
  DUMP: Volume 1 started with block 1 at: Thu Aug   9 09:18:34 2007
  DUMP: dumping (Pass III) [directories]
  DUMP: dumping (Pass IV) [regular files]
  DUMP: Closing dumpzip
  DUMP: Volume 1 completed at: Thu Aug   9 09:18:35 2007
  DUMP: Volume 1 560 blocks (0.55MB)
  DUMP: Volume 1 took 0:00:01
  DUMP: Volume 1 transfer rate: 560 kB/s
  DUMP: 560 blocks (0.55MB) on 1 volume(s)
  DUMP: finished in less than a second
  DUMP: Date of this level   dump: Thu Aug   9 09:18:33 2007
  DUMP: Date this dump completed:   Thu Aug   9 09:18:35 2007
  DUMP: Average transfer rate: 560 kB/s
  DUMP: DUMP IS DONE
```

dump 能够备份文件系统中的所有文件（全备份），也可以有选择地备份最近修改过的文件，具有增量备份功能。增量备份可以节省大量的时间和系统资源，dump 命令中的级别设置可以很好地完成增量备份。表 7-13 是 dump 等级设置的一个标准范例。

表 7-13 dump等级设置范例

星期	dump等级	星期	dump等级
星期日	等级0，全备份	星期一	等级9，增量式备份
星期二	等级8，增量式备份	星期三	等级7，增量式备份
星期四	等级6，增量式备份	星期五	等级5，增量式备份
星期六	等级4，增量式备份		

该时间表中，从星期日开始，每一次的增量备份都比前一次大，所有前一次增量备份中的文件以及自那以后修改过的文件都需要进行备份。整体运行的效里是从星期日开始，修改过的所有文件在每天的增量备份中都得到了备份。当出现磁盘损坏等需要恢复备份的情况，按该时间表方法可以节省大量时间和精力。例如，如果周三发生了磁盘损坏，需要恢复备份，只需恢复上周日全备份数据和本周二的增量备份数据即可。

2. restore 命令

restore 的命令格式与 dump 相似，restore 的选项如表 7-14 所示。

表 7-14 restore命令选项说明

选项	说明
-r	恢复整个dump备份。
-C	将dump备份的内容同硬盘上的文件进行比较，通常用来检查恢复是否成功。
-x filelist	仅从dump备份中恢复指定的文件或目录，其中filelist是待提取的文件或目录列表。
-T file	列出dump备份文件的内容。如果指定了文件或目录，则仅列出与之相关的内容。
-R	从一组备份磁带中的某一特定磁带开始恢复。
-i	以交互方式恢复备份。
-y	如果restore在恢复备份时遇到损坏的数据块，不进行提示继续恢复。
-v	verbose模式，每恢复一个文件都会显示相关信息。
-b blocksize	指定dump块大小。
-D filesystem	当使用-C选项时，指定被比较的文件系统的名字。
-T directory	指定restore使用的临时文件。
-N	不释放文件，仅打印文件名。
-m	根据i节点数释放，而不是根据文件名。

例如在当前目录下恢复备份文件 dumpzip 所做的备份，命令行如下：

```
# restore  -t  -f  dumpzip
Dump     date: Thu Aug  9 09:18:33 2007
Dumped from: the epoch
Level 0 dump of / (dir myshare/zip) on localhost.localdomain:/dev/mapper/VolGroup00-LogVol00
Label: none
             2          .
        390145          ./myshare
        390188          ./myshare/zip
        390174          ./myshare/zip/file2
        390189          ./myshare/zip/file3
        390192          ./myshare/zip/file1.bz2
        390212          ./myshare/zip/file.bz2
        390206          ./myshare/zip/rec00001file1
        390213          ./myshare/zip/rec00001file
```

注意：dump 和 restore 命令虽然多年来被广泛使用，但由于其对正在变化的文件系统（例如日志、电子邮件、数据库等）的处理上还存在可靠性和稳定性问题，以及 dump 与 restore 版本自身的兼容性问题，使得该命令大多仍然应用在磁带等老式介质的备份上。

7.11.3　cpio 命令

cpio 与 tar 程序相似，但 cpio 不仅可以实现分卷备份，还可以跳过磁带上的坏区。cpio 经常使用两个选项：-o 和-i。其中选项“-o”表示从标准输入设备读入文件名，归档后从标准输出设备输出；选项“-i”与之相反，用于恢复备份。cpio 命令经常和命名管道、输入输出重定向联合使用来实现备份。

例如对当前目录下所有的“.txt”文件进行备份，备份到 backup_txt 文件中，命令如下：

```
# ls
file1.txt   file2.txt   file3.txt   file4.doc
# ls   *.txt   |   cpio -o > backup_txt
1 block
# ls
backup_txt   file1.txt   file2.txt   file3.txt   file4.doc
```

恢复备份需要使用选项“-i”，例如对上例中创建的备份文件 backup_txt 进行恢复，命令如下：

```
# ls
backup_txt
# cpio   -i   <   backup_txt
1 block
# ls
backup_txt   file1.txt   file2.txt   file3.txt
```

7.11.4　dd 命令

dd 命令与 MS-DOS 下的 disk copy 程序功能类似，主要用于实现磁盘到磁盘的复制。dd 命令格式如下所示：

```
dd [if=文件系统 1] [of=文件系统 2]
```

dd 除了用于整盘复制，还可用来建立软盘的全盘镜像或从磁盘镜像建立软盘系统。例如需要将一张 1.44MB 软盘中的内容保存到硬盘上，由于软盘不是 Linux 文件系统，无法直接复制，可以使用 dd 命令建立磁盘镜像，命令行如下：

```
#dd   if=/dev/fd0   of=floppy_image
```

如果需要将磁盘镜像恢复到磁盘上，只需反转上例中的参数，如下所示：

```
#dd   if=floppy_image   of=/dev/fd0
```

7.11.5　rsync 命令

rsync 是 remote syncronize 的缩写，通常在系统之间创建文件和目录的同步副本。使用 rsync 可以在本地磁盘和远程系统之间备份文件和目录。由于其只对修改过的文件和目录从源位置复制到目的位置，因而可以有效节省带宽。命令格式为：

```
rsync [选项]  源...   目标...
```

- -a：归档模式。
- -v：verbose 模型。
- -e：登录远端服务器所使用的 Shell。

rsync 的选项很多，可以参考 rsync（1）。

例如将一台名为 studnet.example.com 主机上的/home/share 目录中的所有 doc 文件备份到主机 teacher.example.com 的/usr/local/share 目录下，应在 student 主机上执行下面的命令：

```
#rsync   -av   -e ssh    /home/share/*.doc   root@teacher.example.com:/usr/local/share
root@teacher.example.com's password:
building file list ...done
file1.doc
file2.doc
file3.doc
```

```
write 23432423 bytes    read    242 bytes 2534444.56 bytes/sec
total size is 23432165 speedup is 1.00
```

可以看到，rsync 命令通过 SSH 与 teacher.example.com 主机建立了连接，并将/home/share 目录下的所有文件备份到了 teacher 主机的/usr/local/share 目录下。

如果 studnet.example.com 主机中的/home/share 目录自上次运行 rsync 命令以来没有发生变化，则再次运行 rsync 命令，显示如下：

```
#rsync   -av   -e ssh     /home/share/*.doc   root@teacher.example.com:/usr/local/share
root@teacher.example.com's password:
building file list ...done
file1.doc
file2.doc
file3.doc
write 274 bytes    read    23 bytes 86.34 bytes/sec
total size is 23432165 speedup is 55778.89
```

可以看到，由于 studnet.example.com 主机上的/home/share 与 teacher.example.com 主机的/usr/local/share 已经同步，再次使用 rsync 命令，并没有复制任何文件。

如果向 studnet.example.com 主机上的/home/share 目录中添加文件 file4.doc，再次运行 rsync 命令，显示如下：

```
#cp   file4.doc   /home/share
#rsync   -av   -e ssh     /home/share/*.doc   root@teacher.example.com:/usr/local/share
root@teacher.example.com's password:
building file list ...done
file4.doc
write 2374 bytes    read    29 bytes 23486.34 bytes/sec
total size is 23434994 speedup is 78.89
```

可以看到，rsync 仅对新添加的文件 file4.doc 进行的复制。

7.11.6　使用 cp 命令制作软盘镜像

由于软盘易于损坏、不适合长期保存，因此一般需要将软盘中的内容制作成镜像文件备份在硬盘中，在需要使用时，再将镜像文件恢复到软盘中。在 Red Hat Enterprise Linux 系统中可以使用 cp 命令完成镜像文件的制作，命令格式为：

```
cp   /dev/fd0   镜像文件名
```

例如将软盘中的内容制作成镜像文件 myfloppy.img 保存在/home/backup 目录下，命令行为：

```
#cp   /dev/fd0   /home/backup/myfloppy.img
```

在需要使用软盘时，可以再次使用 cp 命令将镜像文件恢复到软盘中，命令行为：

```
#cp   /home/backup/myfloppy.img   /dev/fd0
```

在 Red Hat Enterprise Linux 中也可以直接将镜像文件挂载到文件系统中，从而可以像直接使用软盘一样读取其中的内容。可以使用 mount 命令对镜像文件进行挂载，例如将上例中的镜像文件挂载到/mnt/floppy_img 目录下，命令行为：

```
#mount   -o   loop   myfloppy.img   /mnt/floppy_img
```

如果/mnt/floppy_img 目录不存在，需要提前创建。

7.11.7　制作光盘镜像

ISO 文件是目前广泛使用的光盘镜像标准格式。为了方便传输和数据保存，通常可以将光盘中的内容制作成 ISO 文件存储到磁盘中，在需要使用时再将镜像文件刻录到光盘中。在 Windows 中可以使用 WinISO、Alcohol 等软件从光盘生成 ISO 文件，在 Linux 中与制作软盘镜像一样，可以直接使用 cp 命令。命令格式为：

```
cp　/dev/cdrom　镜像文件名
```

例如将光盘中的内容制成镜像文件，文件名为 mycdrom.iso，命令行为：

```
#cp　/dev/cdrom　mycdrom.iso
```

可以使用 cdrecord 命令将制作完成的镜像文件刻录到光盘，命令格式为：

```
cdrecord　-v　speed=刻录速度 dev=刻录机设备号　镜像文件名
```

其中刻录机设备号可以使用“cdrecord -scanbus”命令获得。例如将上例中的镜像文件刻录到光盘中，命令行为：

```
#cdrecord　-v　speed=8　dev=0,0　mycdrom.iso
```

在 Linux 中不仅可以将光盘制成镜像文件，还可以将系统中的任何文件或目录制作成镜像文件。这需要使用命令 mkisofs，命令格式如下：

```
mkisofs　-r　-o 镜像文件名　路径名
```

例如将/home/backup 目录下的所有内容制成镜像文件，文件名为 mybackup.iso，命令行为：

```
#mkisofs　-r　-o mybackup.iso　/home/backup
```

在 Linux 系统中，镜像文件也可以直接挂载到系统中，从而直接从镜像文件中读取数据，无需再该录到光盘。挂载镜像文件需要使用 mount 命令，格式如下：

```
mount　-o　loop 镜像文件名　挂载点
```

例如将上例中制作的 mybackup.iso 文件挂载到/mnt/backup 目录下，命令行为：

```
mount　-o　loop　mybackup.iso　/mnt/backup
```

7.12　常用联机帮助命令

在 Red Hat Enterprise Linux 5 系统中提供了丰富的联机帮助（包括手册和信息文档页），可以使用 man 和 info 等命令进行查询。

7.12.1　man 命令

man 命令用于查看 Linux 系统的手册。手册是 Linux 中广泛使用的联机帮助形式，其中不仅包括了常用的命令帮助说明，还包括配置文件、设备文件、协议、库函数等多种信息。man 命令的一般格式为：

```
man　[-acdfFhkKtwW]　[--path]　[-m system] [-p string] [-C config_file] [-M pathlist] [-P pager]
[-B browser] [-H htmlpager] [-S　section_list] [section] name ...
```

其中各选项说明如表 7-15 所示。

表 7-15 man命令选项说明

选项	说明
-a	man默认在显示第一个搜索到的手册之后会自动退出，如果使用该选项会强制man继续显示所有匹配的手册。
-d	不显示手册，只显示错误信息。

-c	强制对手册重新排版。
-K	对所有的联机手册搜寻指字的字符串。一般会比较慢，大约每搜索500页需要1分钟左右时间，如果指定章节会加快搜索。
-k	同apropos
-h	打印帮助信息并退出。
-f	同whatis
-t	对联机手册重新排版，并将结果输出到标准输出设备。
-w	不显示联机手册内容，只显示手册的位置。
-W	与-w相同，但每行打印一个文件名，不显示其他信息。
--path	同-w
-H	设定以文件方式显示HTML文件时所使用的命令，其设定值的优先级高于环境变量HTMLPAGER，man默认使用/bin/cat。
-m system	按照给定的system名，指定另一组联机手册。
-C config_file	设定使用的配置文件，系统默认是/etc/man.config，有关配置文件的详细说明可以查看man.config（5）。
-D	显示并打印debug信息。
-M pathlist	设定搜索手册的路径，多个路径之间用冒号分隔。如果搜索路径为空，则“-M”不起作用。如果没有指定路径则使用环境变量MANPATH的设定值；如果MANPATH没有指定，则使用/usr/lib/man.conf内的设定值
-P pager	设定使用何种page，其优先级高于环境变量MANPAGER和PAGE的设定值。man默认使用/usr/bin/less -is。
-S section_list	设定搜索的章节列表，用冒号分隔，其设定值的优先级高于环境变量MANSECT。
-B	设定使用HTML文件时的浏览器，其优先级高于环境变量BROWSER的设定值。man默认使用/usr/bin/less -is。

1. man 命令常规操作

例如使用 man 查看 clear 命令的帮助：

```
clear(1)                                                    clear(1)
NAME
       clear - clear the terminal screen      //清空终端屏幕命令-----clear
SYNOPSIS
       clear                  //语法
DESCRIPTION
       clear clears your screen if this is possible.  It looks in the environ-
       ment for the terminal type and then in the terminfo database to  figure
       out how to clear the screen.
       clear ignores any command-line parameters that may be present.
SEE ALSO
       tput(1), terminfo(5)              //也可以查看 man1 中的 tput 和 man5 中的 terminfo 命令
       This describes ncurses version 5.5 (patch 20060715).
```

也可以查看 man 命令自身的手册帮助信息：

```
#man man
man(1)                                                      man(1)
NAME
       man - format and display the on-line manual pages     //在线手册页-----man
SYNOPSIS                        //man 格式
       man  [-acdfFhkKtwW]  [--path]  [-m system] [-p string] [-C config_file]
       [-M pathlist] [-P pager] [-B browser] [-H htmlpager] [-S  section_list]
       [section] name ...
DESCRIPTION
       man formats and displays the on-line manual pages.  If you specify sec-
       tion, man only looks in that section of the manual.  name  is  normally
       the  name of the manual page, which is typically the name of a command,
```

```
        function, or file.  However, if name contains  a  slash  (/)  then  man
        interprets  it  as a file specification, so that you can do man ./foo.5
        or even man /cd/foo/bar.1.gz.
        See below for a description of where man  looks  for  the  manual  page
        files.
OPTIONS          //man 选项
        -C  config_file
```

可以看到，手册一般包括“NAME、DESCRIPTION、FILES、SEE ALSO”等几个部分，按“q”键可以退出 man 命令的交互界面。

2. 按章节查询

man 手册的存储位置定义在/etc/man.config 文件中，一般按类型存放在/usr/share/man 目录下，组成不同的章节：

```
# ls  -d  /usr/share/man/man?
/usr/share/man/man1  /usr/share/man/man5  /usr/share/man/man9
/usr/share/man/man2  /usr/share/man/man6  /usr/share/man/mann
/usr/share/man/man3  /usr/share/man/man7
/usr/share/man/man4  /usr/share/man/man8
```

其中每一个目录下都存放着对应类型的手册文件，手册文件大多为“.gz”格式的压缩文件，采用“手册名称.章节.gz”命令规则。例如查看第 2 章所包括的手册文件，命令如下所示：

```
#ls /usr/share/man/man2
accept.2.gz              ioprio_get.2.gz              sched_setparam.2.gz
access.2.gz              ioprio_set.2.gz              sched_setscheduler.2.gz
acct.2.gz                io_setup.2.gz                sched_yield.2.gz
add_key.2.gz             io_submit.2.gz               security.2.gz
...  ...
```

手册分为 man1~man9 共 9 个章节，对应 9 种类型，其说明如表 7-16 所示。

表 7-16 手册章节说明

章节	说明
man1	提供给普通用户使用的可执行命令说明。
man2	系统调用、内核函数的说明。
man3	子程序、库函数说明。
man4	系统设备手册，包括“/dev”目录中设备文件的参考说明。
man5	配置文件格式手册，包括“/etc”目录下各种配置文件的格式说明。
man6	游戏的说明手册。
man7	协议转换手册。
man8	系统管理工具手册，这些工具只有根用户才可以使用。
man9	Linux系统例程手册。

也可以使用“man N intro”命令查看某一章手册的说明信息，其中“N”取值 1～9，与手册章节相对应。例如查看第 4 章手册的说明信息如下：

```
#man  4  intro
INTRO(4)                  Linux Programmer's Manual                  INTRO(4)
NAME
        intro - Introduction to special files
DESCRIPTION
        This chapter describes special files.
FILES
        /dev/* — device files          //设备文件，包括/dev
```

```
AUTHORS
       Look  at  the header of the manual page for the author(s) and copyright
       conditions.  Note that these can be different from page to page!
SEE ALSO
       standards(7)
Linux                              1993-07-24                          INTRO(4)
(END)
```

如果在不同的章节中有相同的说明项，可以在使用 man 命令的同时指定手册章节。例如 passwd 命令在 man1 和 man5 中均有帮助说明，若查看 passwd 命令在手册第 5 章中的帮助说明，可以使用如下命令：

```
#man  5  passwd
PASSWD(5)                  Linux Programmer's Manual                  PASSWD(5)
NAME
       passwd - password file
DESCRIPTION                    //passwd（5）描述内容如下
       Passwd  is  a text file, that contains a list of the system's accounts,
       giving for each account some useful information like user ID, group ID,
       home  directory,  shell,  etc.  Often,  it also contains the encrypted
       passwords for each account.  It should  have  general  read  permission
       (many  utilities, like ls(1) use it to map user IDs to user names), but
       write access only for the superuser.
       In the good old days there was no great problem with this general  read
       permission.  Everybody  could  read  the  encrypted passwords, but the
       hardware was too slow to crack a well-chosen  password,  and  moreover,
       the  basic  assumption  used  to  be that of a friendly user-community.
       These days many people run some version of the shadow  password  suite,
       where /etc/passwd has asterisks (*) instead of encrypted passwords, and
       the encrypted passwords are in /etc/shadow which  is  readable  by  the
       superuser only.
       Regardless  of whether shadow passwords are used, many sysadmins use an
       asterisk in the encrypted password field to make sure  that  this  user
       can  not  authenticate  him-  or herself using a password. (But see the
       Notes below.)
       If you create a new login, first put an asterisk in the password field,
       then use passwd(1) to set it.   //在创建一个新的用户时，应在 password 字段中添加一个“*”符号，并
                                       //用 man1 中介绍的 passwd 命令设置密码
```

3. man 文件的输出

man 虽然具有强大的在线查询功能，但由于 man 文件的格式不是一般的文本文件，所以很难直接将帮助信息进行打印。通常是将 man 的执行结果通过输出重定向，导入到另一个文本文件中进行编辑打印。例如将 ls 命令的帮助信息输出到/ls_help 文件中，命令行为：

```
#man ls  >  /ls_help
```

7.12.2　info 命令

info 文档是 Linux 系统中提供的另一种格式的帮助信息，与手册相比有更强的交互性。可以使用 info 命令查看 texinfo 格式的帮助文档。

info 文档通常存放在/usr/share/info 目录中，如下所示：

```
#ls
a2ps.info.gz            emacs-16.gz          grub.info.gz
accounting.info.gz      emacs-17.gz          gzip.info.gz
ada-mode.gz             emacs-18.gz          history.info.gz
annotate.info.gz        emacs-19.gz          idlwave.gz
```

info 命令格式为：

```
info  [命令名]
```

例如查找 passwd 的帮助信息，命令行如下：

```
#info passwd
PASSWD(1)                    User utilities                    PASSWD(1)
NAME
        passwd - update a user's authentication tokens(s)
SYNOPSIS
        passwd [-k] [-l] [-u [-f]] [-d] [-n mindays] [-x maxdays] [-w warndays]
        [-i inactivedays] [-S] [--stdin] [username]
...  ...
```

info 命令支持文件的链接跳转，使用方向键在显示的帮助文档中选择需要进一步查看的文件名，并按回车，被选择的文件会自动打开。

除了 info 命令外，还可以使用 pinfo 命令查看 info 文档。pinfo 命令采用了 lynx 浏览器风格，使得查看文档的操作更加简单。pinfo 支持彩色显示链接文件、支持鼠标功能。例如直接使用 pinfo 命令查看文件列表，如图 7.4 所示。

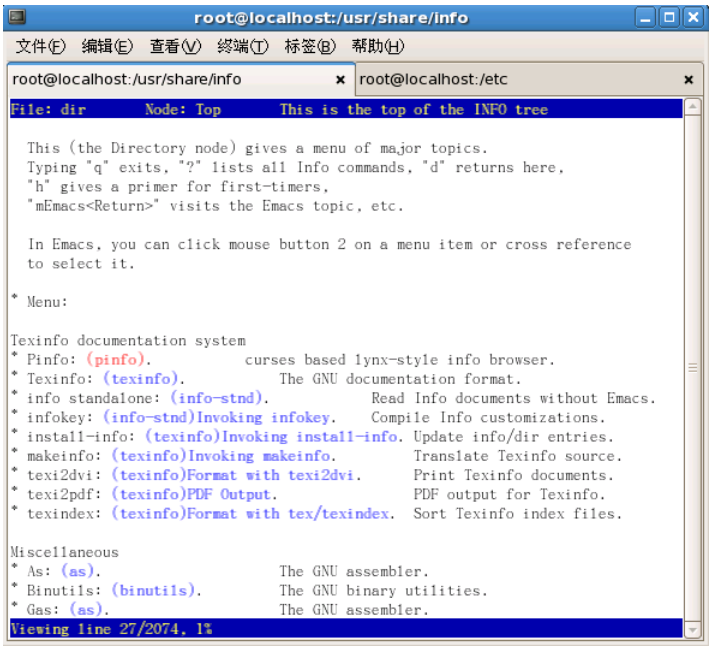

图 7.4　使用 pinfo 命令查看帮助

7.12.3　help 命令

Shell 命令众多，但没有独立的帮助文件。help 命令提供了对这些 Shell 命令的在线帮助支持。help 命令格式如下：

```
help [选项] [命令名]
```

选项：-s：只显示命令格式。例如显示命令 cd 的命令行格式：

```
# help  -s  cd
cd: cd [-L|-P] [dir]
```

如果查看命令 cd 的详细帮助信息，命令行为：

```
# help  cd
cd: cd [-L|-P] [dir]
    Change the current directory to DIR.  The variable $HOME is the
    default DIR.  The variable CDPATH defines the search path for
    the directory containing DIR.  Alternative directory names in CDPATH
    are separated by a colon (:).  A null directory name is the same as
    the current directory, i.e. `.'.  If DIR begins with a slash (/),
    then CDPATH is not used.  If the directory is not found, and the
    shell option `cdable_vars' is set, then try the word as a variable
    name.  If that variable has a value, then cd to the value of that
    variable.  The -P option says to use the physical directory structure
    instead of following symbolic links; the -L option forces symbolic links
    to be followed.
```

直接在待查询的命令后带选项“--help”，也可查询该命令的帮助信息。例如查询命令 mkdir 的帮助信息，命令行为：

```
# mkdir  --help
用法：mkdir [选项] 目录...
若目录不是已经存在则创建目录。
  -Z, --context=CONTEXT (SELinux) set security context to CONTEXT
长选项必须用的参数在使用短选项时也是必须的。
  -m, --mode=模式    设定权限<模式> (类似 chmod)，而不是 rwxrwxrwx 减 umask
  -p, --parents      需要时创建上层目录，如目录早已存在则不当作错误
  -v, --verbose      每次创建新目录都显示信息
      --help     显示此帮助信息并退出
      --version  输出版本信息并退出
请向 <bug-coreutils@gnu.org> 报告错误。
```

help 命令也可以查询自身的帮助信息，例如：

```
# help help
help: help [-s] [pattern ...]
    Display helpful information about builtin commands.  If PATTERN is
    specified, gives detailed help on all commands matching PATTERN,
    otherwise a list of the builtins is printed.  The -s option
    restricts the output for each builtin command matching PATTERN to
    a short usage synopsis.
```

7.12.4　其他相关命令

1. whereis 命令

whereis 命令顾名思义是用来查询文件的存储位置，通常用来查找一个命令的二进制文件、源文件或帮助文件在系统中的位置。whereis 命令的格式为：

```
whereis [选项] 命令名
```

- -b：只查找二进制文件
- -m：只查找帮助文件

❑ -s：只查找 source 文件

如果不带任何选项，则查找并显示所有相关的文件。例如查找命令 mkdir 相关的文件：

```
# whereis mkdir
mkdir: /bin/mkdir /usr/share/man/man1p/mkdir.1p.gz /usr/share/man/man3p/mkdir.3p.gz
/usr/share/man/man1/mkdir.1.gz /usr/share/man/man2/mkdir.2.gz
```

只查找与命令 mkdir 相关的二进制文件：

```
# whereis -b mkdir
mkdir: /bin/mkdir
```

只查找与命令 mkdir 相关的帮助文件：

```
# whereis -m mkdir
mkdir: /usr/share/man/man1p/mkdir.1p.gz /usr/share/man/man3p/mkdir.3p.gz /usr/share/man/man1/mkdir.1.gz
/usr/share/man/man2/mkdir.2.gz
```

2. whatis 命令

与 man 或 info 相比，whatis 命令可以提供更加简洁的帮助信息。whatis 命令对所输入的关键词在 whatis 数据库中进行查找，显示与之相关的信息。

在使用 whatis 命令前，应建立 whatis 数据库。该数据库只有系统管理员才能建立，建立的时间视系统软硬件性能而定。建立 whatis 数据库命令行为：

```
#makewhatis
```

whatis 数据库建立后可以使用 whatis 命令进行查询。例如查询 cd 命令的帮助信息：

```
# whatis cd
cd                    (1p)  - change the working directory
cd [builtins]          (1)  - bash built-in commands, see bash(1)
```

3. apropos 命令

apropos 命令对所输入的字符串在 whatis 数据库中进行搜索，找出并显示包含该字符串的所有数据。apropos 命令是基于字符串，而 whatis 命令是基于关键词，因而 apropos 命令通常会显示比 whatis 命令更多的信息。例如查询命令 cd 的帮助信息：

```
# apropos cd
/etc/nscd.conf [nscd]     (5)  - name service cache daemon configuration file
/usr/sbin/nscd [nscd]     (8)  - name service cache daemon
BN_gcd [BN_add]         (3ssl) - arithmetic operations on BIGNUMs
Encode::EBCDIC           (3pm) - EBCDIC Encodings
FcAtomicDeleteNew         (3)  - delete new file
...  ...
```