第 2 章　SHELL 命令

Shell 是 Linux 的一个特殊程序，是内核与用户的接口，它是命令语言、命令解释程序及程序设计语言的统称。Shell 是一个命令语言解释器，它拥有自己内建的 Shell 命令集，Shell 也能被系统中其他应用程序所调用。当用户成功登录 Linux 系统后,即开始了与 Shell 的对话交互过程，此时，不论何时键入一个命令，都被 Shell 解释执行。Linux 系统对 Shell 采用独立、自由、开放的方式，Shell 的种类有很多。C shell 用来交互；Bourne shell 用来编程;Korn shell 结合了 C shell 的交互式特性,融入 Bourne shell 的语法;大部分 Linux 系统的默认 Shell 类型为 bash(Bourne Again Shell），GNU 计划的一部分。本章主要介绍了 SEHLL 目录、文件命令，管道及重定向命令，用户权限管理命令以及进程管理等的使用。

2.1　Linux 目录类命令

2.1.1　Linux 系统的目录结构

Linux 文件系统继承了 UNIX 的特点，它采用了树型目录结构，通过目录将系统中所有的文件分级、分层的组织在一起，以根目录为起点，所有目录都从根目录里派生而来，

把设备视为文件，设备与文件使用统一的接口进行处理，树型结构的最上层是根目录，用 / 表示。Linux 安装以后，根文件下有许许多多的目录，无论哪个版本都有这些目录，不过不同的发行版本会有些小小的差异。表 2.1 描述了部分 Linux 系统目录结构中常见的目录。

表 2.1 Linux 系统目录结构

目录	描述
/home	包含 Linux 系统上各用户的主目录
/root	root 用户的主目录
/bin	二进制目录，存放许多用户级的 GNU
/sbin	系统二进制目录，存放许多 GNU 管理员级工具
/dev	设备目录，Linux 在这里创建设备节点
/lib	库目录，存放系统和应用程序的库文件
/tmp	临时目录，可以在该目录中创建和删除临时工作文件
/mnt	挂载目录，另一个可移动媒体设备的常用挂载点
/boot	启动目录，存放启动文件
/media	媒体目录，可移媒体设备的常用挂载点
/etc	系统配置文件目录
/proc	进程目录，存放现有硬件及当前进程的相关信息
/usr	用户二进制目录，大量用户级的 GNU 工具和数据文件存储在这里
/var	可变目录，用以存放经常变化的文件，比如日志文件

2.1.2　目录操作

常见目录操作包括目录创建、删除、进入、切换以及查看当前所在目录。

1．pwd　显示当前工作目录的绝对路径

命令格式：pwd

【 例 2-1 】显示用户当前工作目录路径。

```
[user@localhost ~]$ pwd
/home/user
```

注意

绝对路径：指从根目录（/）开始到当前目录（文件）的路径；
相对路径：指从当前目录到其下子目录（文件）的路径。

2．mkdir　创建目录

命令格式：　mkdir　[选项]　[目录]

mkdir 命令的选项说明如表 2.2 所示。

表 2.2　mkdir 选项说明

选项	说明
-p	若路径中的目录不存在，先创建目录
-v	每次创建目录都显示信息

【例 2-2】 创建空目录 test。

```
[user@localhost ~]$ mkdir　test　//当前目录下创建空目录 test
```

【例 2-3】 递归的创建目录。

```
[user@localhost ~]$ mkdir　-p　test1/test2
```

说明

mkdir 命令用于创建新的空目录，可以同时创建多个目录；较常用到的选项为“-p”，该命令用于创建嵌套的多层目录结构；若不使用“-p”选项，则只能在已经存在的目录中创建其他子目录

3．cd　更改工作目录路径

命令格式：　cd　[目录]

【例 2-4 】使用 cd 命令切换目录。

```
[user@localhost ~]$ cd　test
[user@localhost test]$ cd　..
[user@localhost ~]$ cd　/etc/yum
[user@localhost yum]$ cd　~
```

说明

几个特殊符号：

.　　代表当前所在目录
..　　代表当前目录位置的上一层目录
~　　代表家目录（home directory)，即 login 时所在的目录

4．ls　列出目录和文件信息

命令格式：　ls　[选项]　[目录] [文件]

ls 命令的选项说明如表 2.3 所示。

表 2.3　ls 选项说明

选项	说明
-l	以详细信息的形式展示出当前目录的文件
-a	显示当前目录中的全部文件（包括隐藏文件）
-d	查看目录属性
-t	按照创建时间顺序列出文件

-F	显示文件类型

【例 2-5】显示目录/var 下文件和子目录的简单信息。

```
[user@localhost ~]$ ls /var
account  cache  db  games  kerberos  local  log  nis  preserve  spool  tmp  yp
adm  crash  empty  gopher  lib  lock  mail  opt  run  target  www
```

【例 2-6】 显示当前目录下的所有文件和子目录的详细信息，包括隐藏文件。

```
[user@localhost ~]$ ls  -la
总用量 36
-rw-------.  1  user  user  36  12 月  4  201 8  .bash_history
drwxrwxr-x.  2  user  user   6  9 月  5  16:32  test
```

文件属性　引用次数　所有者　所属组　长度　修改时间　文件名

说明

文件属性的第一个字符表示文件类型，d 目录；- 文件；l 链接文件；b 表示为设备中可供存储的接口设备；c 表示为设备文件中的串行端口设备；r 可读；w 可写；x 可执行。

5．rmdir 删除空目录

命令格式： rmdir [选项] [目录]

rmdir 命令选项说明如表 2.4 所示。

表 2.4 rmdir 命令选项说明

选项	说明
-p	删除递归目录，当子目录删除后其父目录为空时，也一同被删除
-v	输出处理的目录详情

【 例 2-7 】 删除空目录 test。

```
[user@localhost ~]$ rmdir test
```

【 例 2-8 】 删除空目录 ~/test1/test2。

```
[user@localhost ~]$ rmdir -p test1/test2
```

2.2 文件操作命令

2.2.1 文件操作

1．touch 创建空文件、更改文件时间

命令格式： touch [选项] 文件

touch 命令的选项说明如表 2.5 所示。

表 2.5 touch 命令选项说明

选项	说明
-a	只更改访问时间（atime）
-m	更改文件的修改时间记录（mtime）
-c	假如目标文件不存在，则不会建立新的文件
-d<字符串>	使用指定字符串表示时间而非当前时间
-t<日期时间>	使用 MMDDhhmm 格式的时间而非当前时间

【例 2-9】 创建空文件 file1、file2、 file3 和 file4。

```
[user@localhost ~]$ touch file1
[user@localhost ~]$ touch file{2,3,4}
[user@localhost ~]$ ls -l file*
-rw-rw-r--. 1 user user 0 9 月      6 09:22 file1
-rw-rw-r--. 1 user user 0 9 月      6 09:22 file2
-rw-rw-r--. 1 user user 0 9 月      6 09:22 file3
-rw-rw-r--. 1 user user 0 9 月      6 09:22 file4
```

【例 2-10】 将文件 file1 的时间记录改为 3 月 12 日 19 时 30 分。

```
[user@localhost ~]$ ls -l file1
-rw-rw-r--. 1 user user 0 9 月      6 09:22 file1
//空文件 file1 的创建日期为 9 月 6 日 9 点 22 分
[user@localhost ~]$ touch -c -t 03121930 file1
[user@localhost ~]$ ls -l file1
-rw-rw-r--. 1 user user 0 3 月     12 19:30 file1
//将文件 file1 的 atime 和 mtime 时间更新为 3 月 12 日 19 时 30 分
```

 **说明**

时间格式 MMDDHHmm 指月（MM）日（DD）时（HH）分（mm），如果还需要加上年份，可以使用 201903121930，表示 2019 年 3 月 12 日 19 时 30 分。

2．cp 复制文件和目录

命令格式： cp [选项] [源文件|目录] [目标文件|目录]

cp 命令选项说明如表 2.6 所示。

表 2.6 cp 命令选项说明

选项	说明
-f	强制复制文件和目录，无论目的文件或目录是否已经存在
-r	递归复制目录下的子目录和文件
-i	若目标文件存在，则提示用户如何操作

【例 2-11】将/etc/passwd 文件复制到当前家目录的 test 文件夹中。

```
[user@localhost ~]$ cp     /etc/passwd      ~/test
```

这时在家目录的 test 中即有一个文件 passwd。

【例 2-12】生成一个 passwd_new 文件，要求内容与 passwd 文件内容相同。

```
[user@localhost ~]$ cd test
[user@localhost test]$ cp passwd passwd_new
```

【例 2-13】 将 ~/test 目录下的所有内容复制到当前家目录，命名为 testnew。

```
[user@localhost ~]$ cp   -r   test     test_new
```

3．mv 文件和目录改名、移动文件和目录路径

命令格式：mv [选项] [源文件|目录] [目标文件|目录]

mv 命令选项说明如表 2.7 所示。

表 2.7 mv 命令选项说明

选项	说明
-f	移动时自动替换已经存在的目标文件，不提示
-i	若目标文件存在，则提示用户如何操作

【例 2-14】 将~/test 目录下的所有后缀名为”*.png”的文件移动到~/pic 目录下。

```
[user@localhost ~]$ mv     ~/test/*.png     ~/pic
```

【例 2-15】把~/pic/1.png 文件改名为 ~/pic/life.png。

```
[user@localhost ~]$ cd  pic
[user@localhost pic]$ mv 1.png life.png
```

【例 2-16】把~/pic 目录名称更改为~/mypic。

```
[user@localhost ~]$ mv  ~/pic  ~/mypic
```

4．rm 删除文件或目录

命令格式：rm [选项] [文件|目录]

rm 命令选项说明如表 2.7 所示。

表 2.7 rm 命令选项说明

选项	说明
-f	强制删除文件，不给出提示信息
-r	递归删除目录及其子目录和文件
-i	删除前需要确认

【例 2-16】 删除文件 file1；删除文件 file2，删除前确认

```
[user@localhost ~]$ rm  file1
[user@localhost ~]$ rm  -I  file2
rm：是否删除普通空文件 "file2"？ y
```

【例 2-17】 删除当前家目录下的目录 mypic

```
[user@localhost ~]$ rm  -r  mypic
```

注意

rm –rf *.*表示强制删除当前目录下的所有文件，管理员权限慎用，会删除根目录下的全部文件及其相关目录。

5．find 用于查找文件或目录

命令格式：find [查找范围] [选项] [查找条件]

如果用户没有指定查找范围，则 find 命令从当前目录开始搜索查找。常见的选项参数如表 2.8 所示。

表 2.7 find 命令常用参数选项

选项	说明
-name〈文件名〉	匹配文件的名称
-user〈用户名〉	匹配文件的所有者
-perm〈权限〉	匹配文件的权限
-type〈文件类型〉	匹配文件类型查找
-size n[ckMG]	匹配文件的大小（+50k 查找超过 50k 的文件,而-50k 则代表查找小于 50k 的文件）
--exec {} \;	后面可接对搜索到结果进一步处理的命令

【 例 2-18 】搜索在/etc/中所有以 host 开头的文件。

其中的”host*”表示所有以 host 开头的文件。

```
[root@localhost ~]# find /etc -name "host*" -type f
/etc/host.conf
/etc/hosts
/etc/hosts.allow
```

【 例 2-19 】搜索当前目录中文件属主具有读写权限，并且文件所属组及其他用户具有读权限的文件。

```
[root@localhost ~]# find . -perm 644    -type f -exec ls -l {} \;
-rw-r--r--. 1 root root 18 12 月  29 2013 ./.bash_logout
-rw-r--r--. 1 root root 176 12 月  29 2013 ./.bash_profile
-rw-r--r--. 1 root root 176 12 月  29 2013 ./.bashrc
-rw-r--r--. 1 root root 100 12 月  29 2013 ./.cshrc
```

【 例 2-20 】 在/etc 目录下查找大于 1MB 小于 10MB 的文件。

```
[root@localhost ~]# find /etc -size +1M -a -size    -10M
/etc/gconf/schemas/ekiga.schemas
/etc/selinux/targeted/contexts/files/file_contexts.bin
/etc/selinux/targeted/policy/policy.29
/etc/udev/hwdb.bin
/etc/brltty/zh-tw.ctb
```

【 例 2-21 】查找/var/log 目录中更改时间在 7 天以前的普通文件，并删除。

```
[root@localhost ~]# find /var/log -type f -mtime +7 -exec rm {} \;
```

【 例 2-22 】查找找出所有用户 susa 拥有的文件，并且把它们拷贝到/root/finder 目录中。

两种解决办法：

（1）[root@localhost ~]# mkdir   /root/finder

[root@localhost ~]# find / -user susa -type f，然后挨个复制到指定目录中。

（2）[root@localhost ~]# find / -user susa -type f -exec cp '{}' /root/finder \;

6. 查找文件或目录 which/whereis

（1）which

which 的命令格式如下：

```
which  命令
```

主要用来搜索二进制文件、可执行文件或者 shell 命令行的位置。

例如“which find”，返回信息如下：

```
[user@localhost ~]$ which find
/usr/bin/find
```

（2）whereis

whereis 的命令格式如下：

```
whereis  命令
```

主要用来搜索二进制文件的位置、源代码的位置和 man 帮助文件位置。

例如“whereis find”，返回信息如下：

```
[user@localhost ~]$ whereis find
find: /usr/bin/find /usr/share/man/man1/find.1.gz /usr/share/man/man1p/find.1p.gz
```

2.2.2 文件内容操作命令

1．cat 显示本文文件、连接文件内容

命令格式：cat [选项] [文件]

cat 命令的选项说明如表 2.8 所示。

表 2.8 cat 命令的选项说明

选项	说明
-n	对输出的所有行编号
-b	对非空输出行编号

【例 2-18】 显示文件/etc/inittab 文件的内容。

```
[user@localhost ~]$ cat  /etc/inittab
```

【例 2-19】 假设有 file1、file2、file3 三个文本文件，将这三个文本文件连接起来输出到文本文件 file4 中。

```
[user@localhost ~]$ cat file1
aaaaaa
[user@localhost ~]$ cat file2
bbbbbb
[user@localhost ~]$ cat file3
cccccc
[user@localhost ~]$ cat file1 file2 file3>file4
[user@localhost ~]$ cat file4
aaaaaa
bbbbbb
cccccc
```

2．more 分页显示文本文件

命令格式：more [选项] [文件]

more 命令根据窗口的大小进行分页显示，还可以提示文件的显示百分比。more 命令的选项说明如表 2.9 所示。

表 2.9 more 命令选项说明

选项	说明
+n	从第 n 行开始显示文件内容，n 代表数字
-n	一次显示的行数，n 表示数字

【 例 2-20 】分屏显示/etc/passwd 文件的内容。

```
[user@localhost ~]$ more  /etc/passwd
```

说明

查看一个内容较大的文件时，使用 more 命令。利用 Ctrl+f(或空格键)是向下显示一屏；Ctrl+b 是向上滚动一屏；Enter 键向下滚动 1 行；q 键退出 more 命令。

3．head 显示指定文件的前若干行

命令格式：head -n 行数值 [文件]

默认缺省显示文件的前 10 行内容。

例如，head -5 /etc/passwd 表示显示/etc/passwd 文件的前 5 行。

4．tail 查看文件末尾数据

命令格式：tail -n 行数值 [文件]

默认缺省显示文件的后 10 行内容

例如，tail -5 /etc/passwd 表示显示/etc/passwd 文件的后 5 行。

5．grep 在文件中搜索与字符串匹配的行输出

命令格式：grep [选项] 查找条件 源文件

grep 命令的常见选项说明如表 2.10 所示。

表 2.10 grep 命令选项说明

选项	说明
-i	查找时忽略大小写
-v	反转查找，输出与查找条件不相符的行

查找条件设置：要查找的字符串以双引号括起来，“^……”表示以……开头，“……$”

表示以……结尾

“^$”表示空行。

【 例 2-21 】搜索在/etc/passwd 中”/sbin/nologin”出现的行，找出系统中不允许登陆的用户。

```
[root@localhost ~]# grep /sbin/nologin /etc/passwd
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
```

2.2.3 文件链接与文件压缩

1. ln 文件链接

Linux 系统中，内核为每一个新创建的文件分配一个 inode 号，文件属性保存在 inode，访问文件时，inode 被复制到内存里，从而实现文件的快速访问。

ln 原理

文件放在外存，文件信息形成 FCB，FCB 的集合构成目录。访问一个文件时需要把目录调入内存，然后按名检索目录，目录占用内存空间问题产生按名检索时，名字不符的话其他信息并不需要读取，所以 FCB 中许多信息不需要全调入内存。减小 FCB：将文件的详细信息放入索引结点，FCB 中记录文件名和 inode 地址。目录小了，调入内存占空就少，检索也快了。

Linux 包括两种链接：

（1）硬链接（Hard Link）建立硬链接时，链接文件和被链接文件必须位于同一个文件系统中，并且不能建立指向目录的硬链接。默认情况下，ln 产生硬链接。两个文件具有相同的 inode。

（2）符号链接（Symbolic Link）。而对符号链接，则不存在这个问题。符号连接等价于建立了快捷方式。符号连接可以用来建立在不同的文件系统之上，并且可以对目录建立符号连接.ln 命令加参数 －s 产生符号链接。

链接文件命令是 ln 命令，该命令在文件之间创建链接。这种操作际上是给系统中已有的某个文件指定另外一个可用于访问的名称。

```
命令格式：ln  [选项]  源文件 [目标]
```

说明

链接的对象可以是文件，也可以是目录。不加参数默认创建的是一个硬链接，目录不能创建硬链接；-s 创建一个软链接（符号链接）。

【 例 2-22 】 硬链接文件的使用。

```
[user@localhost ~]$ ln  file  file_h
```

创建文件 file 的硬链接文件 file_h

```
[user@localhost ~]$ ls  -l  file file_h
-rw-rw-r--. 2 user user 12 9 月  29 17:37 file
-rw-rw-r--. 2 user user 12 9 月  29 17:37 file_h
```

查看源文件 file 和硬链接文件 file_h 的文件属性，可以看到两个文件的大小还有其他属性都一样，链接数由原来的 1 变成 2。

```
[user@localhost ~]$ stat  file
```

```
  文件："file"
  大小：12          块：8          IO 块：4096   普通文件
设备：fd00h/64768d   Inode：3647368     硬链接：2
权限：(0664/-rw-rw-r--)  Uid：( 1000/     user)   Gid：( 1000/     user)
环境：unconfined_u:object_r:user_home_t:s0
最近访问：2019-09-29 17:37:12.668085187 +0800
最近更改：2019-09-29 17:37:03.547084752 +0800
最近改动：2019-10-05 16:13:14.831050726 +0800
创建时间：-
[user@localhost ~]$ stat  file_h
  文件："file_h"
  大小：12          块：8          IO 块：4096   普通文件
设备：fd00h/64768d   Inode：3647368     硬链接：2
权限：(0664/-rw-rw-r--)  Uid：( 1000/     user)   Gid：( 1000/     user)
环境：unconfined_u:object_r:user_home_t:s0
最近访问：2019-09-29 17:37:12.668085187 +0800
最近更改：2019-09-29 17:37:03.547084752 +0800
最近改动：2019-10-05 16:13:14.831050726 +0800
创建时间：-
```

通过 stat 命令查看文件 file 和 file_h，两个文件的 inode 结点是都是 3647368，说明两个文件是同一个索引结点。

```
[user@localhost ~]$ rm file
[user@localhost ~]$ ls -l file file_h
ls: 无法访问 file: 没有那个文件或目录
-rw-rw-r--. 1 user user 12 9 月  29 17:37 file_h
```

删除源文件 file，硬链接文件的链接数变为 1，其他的对硬链接文件没有影响。

【 例 2-23 】软链接文件的使用。

```
[user@localhost ~]$ ln -s  file  file_s
[user@localhost ~]$ ls  -l  file  file_s
-rw-rw-r--. 1 user user 6 10 月  5 16:28  file
lrwxrwxrwx. 1 user user 4 10 月  5 16:29  file_s -> file
```

创建文件 file 的软连接文件 file_s, file_s -> file 表示 file 是源文件，file_s 的文件类型是 l（链接），链接数是 1。

```
[user@localhost ~]$ stat file
  文件："file"
  大小：6           块：8          IO 块：4096   普通文件
设备：fd00h/64768d   Inode：3953177     硬链接：1
权限：(0664/-rw-rw-r--)  Uid：( 1000/     user)   Gid：( 1000/     user)
环境：unconfined_u:object_r:user_home_t:s0
最近访问：2019-10-05 16:28:45.736095115 +0800
最近更改：2019-10-05 16:28:45.736095115 +0800
最近改动：2019-10-05 16:28:45.736095115 +0800
创建时间：-
[user@localhost ~]$ stat file_s
```

```
  文件："file_s" -> "file"
  大小：4          块：0          IO 块：4096    符号链接
设备：fd00h/64768d    Inode：3953178      硬链接：1
权限：(0777/lrwxrwxrwx)  Uid：( 1000/      user)    Gid：( 1000/      user)
环境：unconfined_u:object_r:user_home_t:s0
最近访问：2019-10-05 16:29:15.357096527 +0800
最近更改：2019-10-05 16:29:00.986095842 +0800
最近改动：2019-10-05 16:29:00.986095842 +0800
创建时间：-
```

使用 stat 命令查看 file 和 file_s 发现两个文件的 inode 结点是不一样的，不是同一个文件，一个是普通文件，一个是符合链接。

```
[user@localhost ~]$ rm file
[user@localhost ~]$ ls -l file file_s
ls: 无法访问 file: 没有那个文件或目录
lrwxrwxrwx. 1 user user 4 10 月  5 16:29 file_s -> file
[user@localhost ~]$ cat file_s
cat: file_s: 没有那个文件或目录
```

删除源文件 file，链接文件还是存在。由于源文件已经删除，所以链接文件的内容也不能查看了，软连接类似于快捷方式。

说明

硬链接创建后，链接文件和目标文件没有主次之分，文件系统中它们都和同 inode 相关联，具有相同的 inode 号，不允许跨文件系统，不允许对目录创建硬链接。

软链接应用也很广泛，在文件系统中多处共享同一个较大文件时，使用软链接可以避免创建多个副本；维护动态库的版本时，使用软链接，在升级库文件后，只需修改软链接的源文件，而使用该库的程序则不需要修改。

2. tar 文件打包、压缩

tar 是 Linux 中最常用的备份工具，也成为打包工具，可以进行文件的压缩和解压缩。Linux 中常用的压缩文件格式为*.tar、*.tar.gz、*.tar.bz2，其中.tar 表示使用 tar 命令打包数据但是没有压缩；.tar.gz 表示使用 tar 命令打包文件并进行 gzip 压缩；.tar.bz2 表示使用 tar 命令打包文件并进行 bzip2 压缩。

命令格式：tar [参数] 打包文件名 文件

tar 命令的命令选项说明如表 2.11 所示。

表 2.11 tar 命令选项说明

选项	说明
-c	产生.tar 打包文件
-v	列出打包解包的详细过程
-f	指定打包文件的名称
-z	以.gz 的格式压缩或解压打包文件
-j	以.bz2 的格式压缩或解压打包文件
-x	从打包文件中还原文件
-C	解压到指定目录
-r	将文件追加到打包文档末尾

tar 常用命令如下：

```
打包：
tar   -czvf   [存放路径]打包文件名.tar.gz   源文件或目录
tar   -cjvf   [存放路径]打包文件名.tar.bz2   源文件或目录
tar   cJvf    [存放路径]打包文件名.tar.xz   源文件或目录
解包：
tar   -xzvf   [存放路径]打包文件名.tar.gz    [-C   解压目录]
tar   -xjvf   [存放路径]打包文件名.tar.bz2   [-C   解压目录]
tar   xJvf   [存放路径]打包文件名.tar.xz    [-C   解压目录]
```

【 例 2-24 】将当前目录下的 file.c 文件添加到文件 myfile.tar 中。并且通过查看 myfile.tar 文件中的文件内容，来验证 file.c 文件是否被成功添加到 myfile.tar 打包文件中。

```
[user@localhost ~]$ tar   rvf   myfile.tar   file.c
```

【 例 2-25 】将当前用户目录下的目录文件 myfile 包括的所有文件用 tar 命令打包，然后用 gzip 命令压缩，打包压缩后的文件名为 myfile.tar.gz。

```
[user@localhost ~]$ tar -zcvf myfile.tar.gz myfile/
```

【 例 2-26 】查看 myfile.tar.gz 压缩文件的内容。

```
[user@localhost ~]$ tar -ztvf myfile.tar.gz
```

【 例 2-27 】将文件打包然后用 bzip2 命令压缩。将当前用户目录/home/user 下的目录文件 myfile 包括的所有文件用 tar 命令打包，并且用 bzip2 命令压缩，打包压缩后的文件名为 myfile.tar.bz2。

```
[user@localhost ~]$tar -jcvf myfile.tar.bz2 myfile/
```

【 例 2-28 】用 tar 命令解压经过 bzip2 压缩的 tar 文件。由于使用 bzip2 压缩，所以要加上 j 这个参数。将压缩文件 myfile.tar.bz2 解压到当前目录下。

```
[user@localhost ~]$tar -jxvf myfile.tar.bz2
```

2.3 管道及重定向

2.3.1 输入输出流控制

在 Linux 系统中，执行一个 shell 命令行时通常会自动打开三个标准文件：标准输入文件（stdin）、 标准输出文件（stdout）和标准错误输出文件（stderr）。进程将从标准输入文件中得到输入数据，将正常输出数据输出到标准输出文件，而将错误信息送到标准错误文件中。

1. 输入重定向

输入重定向主要用于改变一个命令的输入源，指输入可以不来自标准输入（键盘），而来自一个指定的文件。输入重定向的符号为“<”，一般的命令格式为“命令<文件名”

【例 2-29 】通过输入重定向将文件 file 的内容作为输入让 cat 命令执行。

```
[user@localhost ~]$ cat<file
hello world
// cat 从 file 获得输入数据
```

2. 输出重定向

输出重定向指将命令的执行结果输出到指定的文件中，而不是直接显示在屏幕上，重定向符号如表 2.12 所示。

表 2.12 输出重定向符号含义

类型	操作符	用途

重定向标准输出	>	将命令的执行结果输出到指定的文件中，而不是直接显示在屏幕上
	>>	将命令执行的结果追加输出到指定文件
重定向标准错误	2>	清空指定文件的内容，并将标准错误信息保存到该文件中
	2>>	将标准错误信息追加输出到指定的文件中
重定向标准输出和标准错误	&>	将标准输出、标准错误的内容全部保存到指定的文件中，而不是直接显示在屏幕上

【 例 2-30 】输出重定向实例

```
将标准输出重定向到文件
[user@localhost ~]$ ls /etc/ > etcdir
将标准输出重定向追加到文件
[user@localhost ~]$ ls /etc/sysconfig/ >> etcdir
将错误输出重定向到文件
[user@localhost ~]$ nocmd 2> errfile          //nocmd 表示不存在的指令
将标准输出和错误输出重定向到文件
[user@localhost ~]$ls afile bfile &> errfile
```

注意

①shell 遇到”>”操作符，会判断右边文件是否存在，如果存在就先删除，并且创建新文件。不存在直接创建。无论左边命令执行是否成，右边文件都会变为空。

②“>>”操作符，判断右边文件，如果不存在，先创建，以添加方式打开文件。

2.3.2 管道操作

将一个程序或命令的输出作为另一个程序或命令的输入，有两种方法，一种是通过一个临时文件将两个命令或程序结合在一起;另一种是 Linux 所提供的管道功能。这种方法比前一种方法更好。

管道可以把一系列命令连接起来，这意味着第一个命令的输出会作为第二个命令的输入通过管道传给第二个命令，第二个命令的输出又会作为第三个命令的输入，以此类推。显示在屏幕上的是管道行中最后一个命令的输出（如果命令行中未使用输出重定向）。

管道命令：由若干个指令组成，每个指令用“|”隔开，仅在前一个指令正确才会执行后面的指令。管道使用的命令格式如下：

```
command1|command2|command3|command4
```

举例：ls –la|more，其中 ls –la 显示当前目录下的详细信息，然后作为 more 分屏命令的输入，表示分屏显示当前目录下的详细信息。

cut 命令经常与管道结合在一起，进行日志的分析或将字符串进行分解。cut 的语法格式如下：

```
cut  －d  “分割字符”  [-cf]  fields
```

参数说明如表 2.13 所示。

表 2.13 cut 参数说明

选项	说明
-d	后面接的是分隔字符，默认是空格符

-c	后面接的是第几个字符
-f	后面接的是第几个区块

【 例 2-31 】取出 PATH 变量的第三部分路径

```
[user@localhost ~]$ echo  $PATH|cut  -d:  -f3
/usr/bin
```

【 例 2-32 】解析命令 cat /etc/passwd | grep /bin/bash | wc -l

这条命令使用了两个管道，利用第一个管道将 cat 命令（显示 passwd 文件的内容）的输出送给 grep 命令，grep 命令找出含有 “/bin/bash”的所有行；第二个管道将 grep 的输入送给 wc 命令，wc 命令统计出输入中的行数。这个命令的功能在于找出系统中有多少个用户使用 bash。

```
[user@localhost ~]$ cat /etc/passwd | grep /bin/bash | wc -l
3
```

2.4 用户权限管理

2.4.1 认识系统中的用户和组

1. Linux 的用户类型

在 Linux 操作系统中，用户是分角色的，角色不同，用户权限和所完成的任务也不相同。用户角色是通过 UID（用户 ID）来识别的，每个用户都具有不同的 UID。Linux 用户类型分为 3 类：超级用户、普通用户和系统用户。

（1）超级用户：又称为 root 用户，UID 为 0，拥有计算机系统的最高权限。所有系统的设置和修改都只有超级用户才能执行。

（2）普通用户：是在安装后由超级用户创建，普通用户的权限相当有限，只能操作其拥有权限的文件和目录，只能管理自己启动的进程。

（3）系统用户：正常运行系统时的账户，每个进程运行在系统里都有一个系统用户，系统用户不能登陆，比如 bin、damon、mail 等。

2. 用户账户配置文件

Linux 中的用户账户涉及的配置文件有用户账户文件/etc/passwd、用户密码文件/etc/shadow。

（1） 用户账户文件/etc/passwd

该文件建议初学者通过 cat、head 等命令进行查看，不要随便在文件中进行更改。该文件记录了每个用户的必要信息，每一行对应一个用户的信息，每行的字段之间用”：”分割，共分为 7 个字段：

```
用户名：用户密码：UID：GID：用户信息说明：用户主目录：登陆 Shell
```

①账号名称：在 Linux 系统中用唯一的字符串区分不同的用户。

②密码：出于安全考虑，此处为密码占位符 x，字段加密后存放在/etc/shadow 这个文件中。

③UID：用户识别码 0-65535，用于唯一标识 Linux 系统的用户，用户名和 UID 都可以用于标识用户。

④GID：表示用户所属的组。

⑤用户信息说明：用途较小，主要解释账号的意义。

⑥用户主目录（用户主目录的绝对路径）：用户登陆 Shell 将该目录作为用户的工作目录。超级用户的工作目录为/root；每个用户都有自己的主目录，默认一般在/home 下建立与用户名一致的目录。

⑦登陆 shell：当前用户登陆系统时运行的程序名称，通常是/bin/bash， /sbin/nologin 的

表示用户不能登陆。

（2）用户密码文件/etc/shadow

/etc/shadow 文件又称为影子文件，包括用户及被加密的的密码以及用户账户的有效期等信息，该文件只有超级用户才有权限读取。

/etc/shadow 文件由 9 个字段组成：

用户帐号的名称：加密的密码字串信息：上次修改密码的时间(距离 1970.1.1)：密码的最短有效天数，默认值为 0：密码的最长有效天数，默认值为 99999：提前多少天警告用户口令将过期，默认值为 7：在密码过期之后多少天禁用此用户：帐号失效时间，默认值为空：保留字段（未使用）

3. 组账户

具有某种共同特征的用户集合就是组账户，通过组可以方便的地对多个用户进行访问权限的管理。Linux 的组账户可以分为两类：

（1）基本组(私有组、主要组)

当创建一个新的用户账户时,如果没有指定该用户属于哪一个组群,那么 Linux 就会创建一个和该用户同名的组群,这个组群就是私有组群,在这个私有组群中只包含这个用户。

（2）附加组（公共组、次要组）

可以包含多个用户账户，一个用户账户只能属于一个基本组,可以属于多个附加组。

4. 组账户配置文件

组账户中最常用的配置文件是/etc/group，用于保存组帐号基本信息，可以显示用户归属哪个组或哪几个组。/etc/group 文件内的最后一个字段中列出属于该组的用户成员（一般不包括基本组对应的用户帐号），多个成员之间以逗号“,”分隔。

比如把一个普通用户加入到 root 组群，那么这个用户就可以浏览 root 用户主目录的文件；如果 root 用户把把某个文件的读写执行权限放开，root 组群的所有用户就可以修改次文件;如果是可执行的文件，root 组群的用户也可以执行。

/etc/group 文件的字段含义如下：

用户组名：用户组密码：用户组标识符：组内用户列表

（1） 用户组名：用户组名跟用户名一样，不可重复。

（2） 用户组密码：该字段存放是用户组加密后的密码，但是该字段一般很少使用，Linux 系统的用户组都没有密码。

（3） 用户组标识符：简称 GID，用户唯一标识一个用户组。

（4） 组内用户列表：属于这个组的所有用户的列表。

2.4.2 管理用户及组

Linux 提供了一系列命令管理系统中的用户，包括用户的添加、删除、修改和用户组的添加、删除。常用的命令有 useradd、userdel、usermod、passwd 等。

1. 添加用户

添加新用户时使用 useradd 命令，命令格式如下：

useradd [选项] 用户名

useradd 命令的常见选项如表 2.14 所示。

表 2.14 useradd 命令常见选项

选项	说明
-u	指定用户 ID
-G	指定用户的附属组（补充组）
-s	指定用户登入后所使用的 shell
-d	设置登录时使用的主目录

-M	不建立用户主目录

2. 设置用户密码

设置用户密码的命令是 passwd，命令格式如下：

```
passwd  [选项] 用户名.
```

passwd 命令的常见选项如表 2.15 所示。

表 2.15 passwd 命令常见选项

选项	说明
-u	指定用户 ID
-G	指定用户的附属组（补充组）
-s	指定用户登入后所使用的 shell
-d	设置登录时使用的主目录
-M	不建立用户主目录

【 例 2-33 】创建一个名为 susa 的用户，用户 ID 为 4000，密码是 default123。

```
[root@localhost ~]# useradd  -u 4000 susa
[root@localhost ~]# cat  /etc/passwd|grep  susa
susa:x:4000:4000::/home/susa:/bin/bash
//查看/etc/passwd 文件，显示已经创建成功用户 susa
[root@localhost ~]# cat /etc/shadow|grep susa
susa:!!:18168:0:99999:7:::
//!!表示密码还没有设置启用
[root@localhost ~]# passwd  susa
更改用户 susa 的密码 。
新的 密码：
重新输入新的 密码：
passwd：所有的身份验证令牌已经成功更新。
//设置密码，密码不回显
```

【 例 2-34 】创建用户 newuser1,并设置该用户主目录为/home/www

```
[root@localhost ~]#  useradd  -d  /home/www  newuser1
[root@localhost ~]# cat /etc/passwd|grep newuser1
newuser1:x:1001:1001::/home/www:/bin/bash
```

【 例 2-35 】新增 newuser2 用户，不为用户建立并初始化宿主目录，用户不允许登陆到系统的 shell

```
[root@localhost ~]#  useradd  -M  -s /sbin/nologin newuser2
[root@localhost ~]# cat /etc/passwd|grep newuser2
newuser2:x:1002:1002::/home/newuser2:/sbin/nologin
```

3. 删除用户

若一个用户账户不再使用，可以使用 usedel 命令，将该用户从系统中删除。userdel 命令可以删除指定用户及该用户相关的文件和信息，命令格式如下：

```
userdel  [选项] 用户名
```

userdel 命令的常见选项如表 2.16 所示。

表 2.16 userdel 命令常见选项

选项	说明
-r	同时删除用户及主目录

【 例 2-36 】删除账号 newuser2，并删除相关文件。

[root@localhost ~]# userdel -r newuser2

4. 修改用户账户

修改用户账户信息即修改账号的属性，比如用户 Id、主目录、用户组、登录 Shell 等，修改用户账号信息的命令是 usermod,usermod 的命令格式如下：

usermod [选项]... 用户名

usermod 命令的常见选项如表 2.17 所示。

表 2.17 usermod 命令常见选项

选项	说明
-s	修改用户登入后所使用的 shell
-u	修改用户 ID
-g	修改用户所属基本组
-G	修改用户所属附加组

5. 添加用户组

每个用户都有一个用户组。如果在创建用户时未指定用户组，系统会以用户账户名作为该用户的用户组，并将该用户的用户组同步到/etc/group 中。以【例 2-33】中创建的 susa 用户为例，查看/etc/group 文件中包含 susa 的行，输出结果如下：

[root@localhost ~]# cat /etc/group|grep susa
susa:x:4000:

默认情况下新建用户的用户组与用户名相同，创建用户的同时被创建，也可以单独创建用户组，创建用户组的命令为 groupadd，命令格式如下：

groupadd [-g GID] 组账户名

-g 参数表示指定新建用户组的组 id。

举例：如果把一个用户加入到 root 组群，那么这个用户就可以浏览 root 用户主目录的文件；

如果 root 用户把把某个文件的读写执行权限放开，root 组群的所有用户就可以修改次文件；

如果是可执行的文件，root 组群的用户也可以执行。

如图 2.1 所示，将普通用户 susa 加入 root 组，可以浏览 root 用户的家目录，而没有加入 root 组的普通用户 user 则没有浏览 root 用户家目录的权限。

```
[root@localhost ~]# useradd -G root susa
[root@localhost ~]# echo "default"|passwd --stdin susa
更改用户 susa 的密码 。
passwd：所有的身份验证令牌已经成功更新。
[root@localhost ~]# cat /etc/passwd|grep susa
susa:x:1001:1001::/home/susa:/bin/bash
[root@localhost ~]# cat /etc/group|grep susa
root:x:0:susa
susa:x:1001:
[root@localhost ~]# su susa
[susa@localhost root]$ cd
[susa@localhost ~]$ ls /root
initial-setup-ks.cfg  test  模板  图片  下载  桌面
shdir                 公共  视频  文档  音乐
[susa@localhost ~]$ su user
密码：
[user@localhost susa]$ ls /root
ls: 无法打开目录/root: 权限不够
```

图 2.1　用户权限示例

【 例 2-37 】创建一个用户组 manager，创建一个名为 harry 的用户，其属于 manager 组，这个组是该用户的从属组。

```
[root@localhost ~]# groupadd manager
[root@localhost ~]# useradd -G manager harry
[root@localhost ~]# cat /etc/group|grep manager
manager:x:4001:harry
```

6. 删除用户组

若要删除已存在的用户组，可使用 groupdel 命令，命令格式如下：

```
groupdel　组帐号名
```

【 例 2-38 】删除用户组 manager。

```
[root@localhost ~]# groupdel manager
```

2.4.3　su 与 sudo

为了安全起见，建议使用 Linux 操作系统时，尽量以普通用户身份来操作，等需要系统设置时变换为 root 管理员身份。一般变换身份的方式由两种：(1)用 su 直接将身份变为 root，但是这个命令需要 root 的密码，一般用户需要 root 管理员的密码可以从普通用户变换为管理员用户。(2) 当多人同时管理一台主机时，为了 root 密码的保密性，可以使用 sudo 命令进行用户身份的切换。

1. su

使用 su 命令可以在任意用户之间进行切换。使用 su 命令的一般格式如下：

```
su　[选项]　username
```

usermod 命令的常见选项如表 2.18 所示。

表 2.18　su 命令常见选项

选项	说明
-l	切换用户的同时切换到对应用户的工作目录，环境变量也会随之改变
-	改变登录 shell

【 例 2-39 】由当前普通用户 user 变换为 root 管理员用户，但不改变为 root 用户的环境。

```
[user@localhost ~]$ su
密码：                          //root 用户密码
[root@localhost user]# env
PATH=/usr/local/bin:/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/home/user/.local/bin:/home/user/bin
MAIL=/var/spool/mail/user
PWD=/home/user              //部分环境变量
```

【 例 2-40 】由当前普通用户 user 变换为 root 管理员用户，并切换到 root 用户的环境。

```
[user@localhost ~]$ su -
密码：
MAIL=/var/spool/mail/root
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/root/bin
PWD=/root                           //对比【例 2-39】中的环境变量
```

2. sudo

当用户执行 sudo 时，系统会查找/etc/sudoers 文件，判断该用户执行 sudo 的权限。确认用户具有 sudo 的权限后，让用户“输入用户自己的密码”来确认。如果密码输入成功，执行 sudo 后续的命令，root 执行 sudo 时，不需要输入密码。

用户通过 sudo 命令输入自己的密码可以临时获得 root 的部分权限，可以不必知道 root 的密码，保证系统的安全性。

sudo 命令的格式如下：

```
sudo    -u   [用户名或 uid ] [命令]
```

其中，-u 后面可以接用户账户名称或者 UID，来切换到相应的用户。

默认情况下，普通用户是无法使用 sudo 的，需要编辑设置/etc/sudoers 的配置文件。使用 visudo 去编辑/etc/sudoers，其中 visudo 需要 root 身份执行。

```
## Sudoers allows particular users to run various commands as
## the root user, without needing the root password.
##
## Examples are provided at the bottom of the file for collections
## of related commands, which can then be delegated out to particular
## users or groups.
##
## This file must be edited with the 'visudo' command.
## Allow root to run any commands anywhere
root      ALL=(ALL)          ALL
user      ALL=(ALL)          ALL   //将 user 设置成完全可用。
```

 **说明**

使用 visudo 命令后，通过 vi 打开了/etc/sudoers 配置文件。在配置文件中增加一行“user ALL=(ALL)　　ALL”，使 user 用户可以使用 sudo 命令，增加完毕，保存退出，visudo 会自动检查/etc/sudoers 内部的语法，避免用户输入错误的信息。

【例 2-41】以普通用户的身份浏览管理员用户的家目录。

```
[user@localhost ~]$ ls /root
ls: 无法打开目录/root: 权限不够
[user@localhost ~]$ sudo ls /root
[sudo] password for user:                    //输入 user 用户的密码即可。
anaconda-ks.cfg   file2  file4   公共   视频   文档   音乐
file1        file3     user     模板   图片
```

2.4.4　文件与目录权限设置

1．文件和目录权限简介

用户对文件到底拥有何种权限，需要由访问权限和归属（所有权）共同决定。文件和目录的访问权限有读取（r）、写入（w）和执行（x）。读取指允许查看文件内容、显示目录列表；写入指允许修改文件内容，允许在目录中新建、移动、删除文件或子目录；执行指允许运行程序、切换目录。文件的所有者包括属主（文件的所有者）、属组（同组用户）和其他人。属主指拥有该文件或目录的用户帐号；属组指拥有该文件或目录的组帐号。

 **注意**

例如某目录的权限为[d][r w x][r - -][r - -]，非 root 的其他人是否可以进入该目录？回答是否定的，因为在 Linux 中，文件是否能执行，通过是否具有 x 属性来决定执行，与文件名没有关系；x 与目录的关系相当重要，如果在该目录下不能执行命令，也就无法进入。

2. chmod 设置基本权限

只有系统管理员 root 用户和文件/目录的所有者才可以更改目录和文件的权限，使用 chmod 命令可以更改文件或目录的权限，一般有字符法和数字法两种。

（1）字符法

chmod 的命令格式如下：

chmod [ugoa] [+-=] [rwx] [文件或目录]

其中，u、g、o、a 分别表示属主、属组、其他用户、所有用户，+、-、= 分别表示增加、去除、设置权限。

举例：

```
$chmod o+w my.txt          //my.txt 文件的其他人增加读权限
$chmod ug+w,o-w my.txt     //my.txt 文件的属主和属组增加写权限，其他人去除写权限
$chmod +w my.txt           //my.txt 文件的所有用户增加写权限
```

（2） 数字法

chmod 的命令格式如下：

chmod nnn [文件或目录]

nnn 为 3 位八进制数，与字符法表示的对应关系如表 2.19 所示。

表 2.19 权限字符数字对照表

权限项	读	写	执行	读	写	执行	读	写	执行
字符表示	r	w	x	r	w	x	r	w	x
数字表示	4	2	1	4	2	1	4	2	1
权限分配	文件所有者			文件所属组			其他用户		

如数字 777 表示为 111 111 111 即 rwxrwxrwx；664 表示为 110 110 100 即 rw-rw-r—。

【 例 2-42 】重新设置 mymkdir 文件的权限，为属主用户添加执行权限，去除其他用户的读取权限

```
[root@localhost ~]# chmod u+x,o-r mymkdir
[root@localhost ~]# ls -l mymkdir
-rwxr----- 1 root root 29588 05-12 06:19 mymkdir
```

【 例 2-43 】重新设置 mymkdir 文件的访问权限，恢复为“rwxr-xr-x”

```
[root@localhost ~]# chmod 755 mymkdir
[root@localhost ~]# ls -l mymkdir
-rwxr-xr-x 1 root root 29588 05-12 06:19 mymkdir
```

【 例 2-44 】使用 chmod 命令设置文件权限

查看文件权限如下所示：

```
[root@localhost ~]$ ls  -l  afile
[root@localhost ~]-rw-rw-r--  1  user class1  0  Apr  3 16:52  afile
```

1）增加文件属主 user 的执行权限（x）

```
[root@localhost ~]$ chmod  u+x  afile
```

2）去除文件属组 class1 的写权限（w）

[root@localhost ~]$ chmod g-w afile

3）设置属主权限为读写，属组其他用户的文件权限为读

[root@localhost ~]$ chmod 644 afile

3. chown 设置文件/目录的归属

chown 命令可以更改文件和目录的所属主和所属组。默认情况下，文件或目录的所有者为创建该文件的用户或文件被创建时通过命令指定的用户，需要的时候可以对文件的所有者和所属组进行修改。

chmod 的命令格式如下：

chown [选项] 属主.属组 文件或目录

chown [选项] 属主:属组 文件或目录

常见选项-R 表示递归修改指定目录下所有文件、子目录的归属

【 例 2-45 】chown 命令实例

```
[user@localhost ~]$ ls -l file
-rw-rw-r--. 1 user user 0 10 月  7 09:31 file //查看 user 用户下家目录中的文件 file 的属主和属组，默认是创建用户 user。
[user@localhost ~]$ chown :susa file
chown: 正在更改"file" 的所属组: 不允许的操作
//普通用户无法修改文件的属组和属主
[user@localhost ~]$ su - root
密码：
//切换到管理员用户 root 下
[root@localhost ~]# chown :susa /home/user/file
//将 file 文件的属组修改为 susa 组，由于用户切换，文件 file 使用绝对路径
[root@localhost ~]# ls -l /home/user/file
-rw-rw-r--. 1 user susa 0 10 月  7 09:31 /home/user/file
//查看文件的属组已经修改为 susa 组
[root@localhost ~]# chown root.root /home/user/file
//将文件的属主属组修改为 root
[root@localhost ~]# exit
登出
//退回到普通用户 user
[user@localhost ~]$ ls -l file
-rw-rw-r--. 1 root root 0 10 月  7 09:31 file
//查看文件 file 的属主和属组修改为 root
[user@localhost ~]$ echo "hello">file
bash: file: 权限不够
//由于 file 文件的属主和属组发生变换，user 用户无法对 file 文件进行写入
//思考 user 用户如何可以对 file 文件进行写入？
```

2.4.5 综合实例操作

【 例 2-46 】规划一个用户与组群：有程序开发员 5 人，项目管理员 2 人，分别取名为：prg01-prg05，mgr01,mgr02，并分别从属于组 program 与 manage，现按下列要求规划：

（1）每个开发员拥有自己的帐户，用户名：prg01-prg05，密码：prg01-05 ；

（2）每个开发员从属于 program 组，并共享两个子目录：program 与 source，开发者和同组成员拥有所有权限；

（3）每个管理员拥有自己的帐户，用户名 mgr01-mgr02，密码：mgr01-mgr02 ；
（4）每个管理员从属于 manage 组，并共享两个子目录：project 与 document，管理员和同组成员拥有所有权限；
（5）开辟一个公共子目录/home/public，让它被所有的用户共享，而且拥有所有权限，但不能被非属主删除。

```
[root@localhost ~]#groupadd  program
[root@localhost ~]#groupadd  manage
  //创建两个组 program 与 manage
[root@localhost ~]#useradd  -G  program  prg 01
[root@localhost ~]#echo “prg01”|passwd  --stdin  prg01
//创建开发员用户 prg01，从属于组 program，echo “prg01”|passwd  --stdin  prg01 显示的将给 prg01 用户设置密码，同样的方法创建用户 prg02-prg04
[root@localhost ~]#useradd  -G  manage  -n  mgr01
[root@localhost ~]#echo “mgr01”|passwd  --stdin  mgr01
//创建管理员用户 mgr01，设置密码为 mgr01,同样的方法创建管理员用户 mgr02
[root@localhost ~]#mkdir  /home/program
[root@localhost ~]#mkdir  /home/source
[root@localhost ~]#mkdir  /home/project
[root@localhost ~]#mkdir  /home/document
//创建四个共享目录
[root@localhost ~]#chmod  770  /home/program
[root@localhost ~]#chown  :program  /home/program
[root@localhost ~]#chmod  770  /home/source
[root@localhost ~]#chown  :program  /home/source
//开发者和同组成员拥有所有权限
[root@localhost ~]#chmod  770  /home/project
[root@localhost ~]#chown  :manage  /home/project
[root@localhost ~]#chmod  770  /home/document
[root@localhost ~]#chown  :manage  /home/document
//管理员和同组成员拥有所有权限
[root@localhost ~]#mkdir  /home/public
[root@localhost ~]#chmod  o+rwxt  /home/public
//开辟一个公共子目录/home/public，让它被所有的用户共享，而且拥有所有权限，但不能被非属主删除，也可使用命令#chmod  777  /home/public 、 #chmod  o+t /home/public 或 #chmod 1777  /home/public
//其中 t 为粘滞位，粘滞位主要用途：为公共目录（例如，权限为 777 的）设置，权限字符为“t”， 用户不能删除该目录中其他用户的文件。
```

【 例 2-46 】根据以下要求完成对文件/目录权限的设置：
（1）添加组 group，添加用户 aa、bb 并加入 group 组；
（2）新建文件/abc.txt；
（3）设置用户 aa 对文件拥有读、写和执行权限；
（4）设置组 group 内成员对文件拥有读和写权限
（5）设置除属主和属组外其他人对文件没有任何权限
（6）新建目录/abc

（7）设置用户 bb 对目录拥有读、写执行权限
（8）设置组 group 内成员对目录拥有读和执行权限
（9）设置除属主和属组外其他人对目录没有任何权限

```
[root@localhost ~]# groupadd group
[root@localhost ~]# useradd -G group aa
[root@localhost ~]# useradd -G group bb
[root@localhost ~]# touch /abc.txt
[root@localhost ~]# ls -l /abc.txt
-rw-r--r--. 1 root root 0 10 月    7 10:34 /abc.txt
 [root@localhost ~]# chown aa:group /abc.txt
 [root@localhost ~]# chmod 760 /abc.txt
[root@localhost ~]# ls -l /abc.txt
-rwxrw----. 1 aa group 0 10 月    7 10:34 /abc.txt
[root@localhost ~]# su aa
[aa@localhost root]$ echo "hello">/abc.txt
//切换到属主用户 aa 进行写入的测试
[root@localhost ~]# su bb
 [bb@localhost root]$ echo "world">/abc.txt
[bb@localhost root]$ cat /abc.txt
World
//切换到同组用户 bb 进行写入的测试
[root@localhost ~]# mkdir /abc
[root@localhost ~]# ls -ld /abc
drwxr-xr-x. 2 root root 6 10 月    7 10:57 /abc
[root@localhost ~]# chown bb:group /abc
[root@localhost ~]# chmod 650 /abc
 [root@localhost ~]# ls -ld /abc
drw-r-x---. 2 bb group 6 10 月    7 10:57 /abc
[root@localhost ~]# su bb
[bb@localhost root]$ cd /abc
bash: cd: /abc:  权限不够
//思考为什么权限不够?
[bb@localhost root]$ exit
exit
[root@localhost ~]# su aa
[aa@localhost root]$ cd   /abc
//思考为什么 aa 用户可以进入/abc?
```

2.5　进程管理

程序是保存在硬盘、光盘等介质中的可执行代码和数据，是静态保存的代码。进程是在 CPU 及内存中运行的程序代码，是动态执行的代码；进程之间存在亲缘关系，每个进程可以创建一个或多个进程，例如提供 Web 服务的 httpd 程序，当有大量用户同时访问 web 页面时，httpd 程序可能会创建多个进程来提供服务。

系统在管理进程时，按照进程的相关属性对进程进行管理，常见的进程属性包括：

（1） 进程标识（PID）：每个进程在创建时会分配一个唯一的 PID。

（2） 父进程标识（PPID）：通过父进程标识和进程标识可以判别进程之间的亲缘关系。

（3） 进程的状态：进程状态包括：运行态 running（运行或准备运行）、等待态 sleeping（包括可中断和不可中断状态）、停止态 stopped、僵死态 zombie。

（4）进行执行的优先级、进程连接的终端以及进程占用的内存、CPU 等资源。

2.5.1 进程查看

1. ps 指令

ps 指令一条基本且非常强大的进程查看命令，可以确定有哪些进程信息：运行的状态、是否结束、有没有僵死、哪些占用了过多的资源等，也可以监控后台进程的工作情况。ps 的命令格式如下

```
ps  [选项]
```

当 ps 后无参数选项，显示当前终端的系统进程，ps 命令中各选项的含义如表 2.20 所示。

表 2.20 ps 命令的部分选项参数

选项	说明
-a	显示当前终端上的所有进程，包括其他用户的进程信息
-e	显示系统中所有进程，包括其他用户进程和系统进程的信息
-l	以长格式显示进程的信息
-u	显示面向用户的格式（包括用户名、CPU 及内存使用情况等信息）
-x	显示后台进程的信息
-f	显示进程的所有信息

【 例 2-47 】查看当前进程的详细信息。

```
[user@localhost ~]$ ps -l
F S   UID     PID    PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
0 S  1000    4705    4701  0  80   0 - 29088 wait   pts/0    00:00:00 bash
0 R  1000   39784    4705  0  80   0 - 34343 -      pts/0    00:00:00 ps
```

信息说明：S ：进程状态（R 表示进行状态；S 表示休眠状态；T 表示暂停或终止状态机；Z 表示僵死状态）；UID：进程启动者的用户 ID；C：进程最近使用 CPU 的估算；PRI：进程的优先级；TIME：进程启动以后占用 CPU 的总时间；CMD：启动该进程的命令名称；TTY：进程所在终端的终端号，其中图形界面的虚拟终端表示为 pts/0，字符界面的终端号为 tty2-tty6，“？”表示进程不占用终端。

【 例 2-48 】查看 httpd 进程是否正在运行。

```
[user@localhost ~]$ ps -ef|grep httpd
root      40045      1  0 22:14 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
apache    40055  40045  0 22:14 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
apache    40056  40045  0 22:14 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
apache    40057  40045  0 22:14 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
apache    40058  40045  0 22:14 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
apache    40059  40045  0 22:14 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
apache    40060  40045  0 22:14 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
user      40196   4705  0 22:15 pts/0    00:00:00 grep --color=auto httpd
```

//系统内有多个 httpd 进程正在运行，通过 PID 和 PPID 可以看出这些 httpd 进程之间的

亲缘关系。

【 例 2-49 】显示所有用户正在运行的进程，包括无终端的进程

```
[user@localhost ~]$ ps -aux|more
USER        PID  %CPU  %MEM      VSZ      RSS      TTY        STAT  START
TIME COMMAND
root          1   0.0   0.1      126080   6812     ?          Ss    12:29      0:11
/usr/lib/systemd/systemd --switched-root --system --deserialize 21
root          2   0.0   0.0      0        0        ?          S     12:29      0:00 [kthreadd]
root          3   0.0   0.0      0        0        ?          S     12:29      0:00 [ksoftirqd/0]
root          7   0.0   0.0      0        0        ?          S     12:29      0:00 [migration/0]
root          8   0.0   0.0      0        0        ?          S     12:29      0:00 [rcu_bh]
root          9   0.0   0.0      0        0        ?          S     12:29      0:00 [rcuob/0]
root          10  0.0   0.0      0        0        ?          S     12:29      0:00 [rcuob/1]
root          11  0.0   0.0      0        0        ?          S     12:29      0:00 [rcuob/2]
```

信息说明如下：

USER：用户名。

%CPU：占用 CPU 时间与总时间的百分比。

%MEM：占用内存与系统内存总量的百分比。

VSZ：进程占用的虚拟内存空间，单位 KB。

RSS：进程占用的内存空间，单位 KB。

TTY：该 进程是在那个终端机上面运作，若与终端机无关，则显示 ?，另外， tty1-tty6 是本机上面的登入者程序，若为 pts/0 等等的，则表示为由网络连接进主机的程序。

STAT：该程序目前的状态

R ：该程序目前正在运作，或者是可被运作；

S ：该程序目前正在睡眠；

T ：该程序目前正在侦测或者是停止了；

Z ：该程序应该已经终止，但是其父程序却无法正常的终止他，造成 zombie (僵尸) 程序的状态

START：该进程被触发启动的时间。

TIME ：该进程实际使用 CPU 运作的时间。

COMMAND：产生该进程的命令。

2. top 指令

top 命令和 ps 命令的基本作用是相同的，显示系统当前的进程及其状态，但是 top 是一个动态显示过程，通过用户按键来不断刷新当前状态。如果在前台执行该命令，它将独占前台，直到用户按下“q”键终止该程序为止。

top 命令提供了实时的对系统处理器的状态监视，它可以显示系统中 CPU 最“敏感”的任务列表，该命令可以按 CPU 使用、内存使用和执行时间对任务进行排序，它的很多特性都可以通过交互式命令、或者在个人定制文件中进行设定。

top 命令格式如下：

```
top  [选项]
```

top 常见选项的含义如表 2.21 所示。

表 2.21　top 命令的部分选项参数

选项	说明
-u〈用户名|UID〉	仅监视指定用户的进程

-p〈进程 PID〉	仅监视指定进程 ID 的进程
-d〈间隔秒数〉	监控进程执行状态的间隔时间，单位是秒

【 例 2-50 】使用 top 命令动态显示进程信息

```
[user@localhost ~]$ top
top - 22:43:16 up 10:13,  2 users,  load average: 0.08, 0.04, 0.05
Tasks: 440 total,   2 running, 438 sleeping,   0 stopped,   0 zombie
%Cpu(s): 37.0 us,  5.3 sy,  0.0 ni, 57.7 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem :  3866920 total,  2661432 free,   670468 used,   535020 buff/cache
KiB Swap:  2359288 total,  2359288 free,        0 used.  2918392 avail Mem
  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
 4049 user      20   0 1548984 242200  48164 S  29.6   6.3   4:42.22 gnome-shell
 3067 root      20   0  224328  37136   9236 S  10.2   1.0   1:24.11 Xorg
 4701 user      20   0  574952  26776  14940 S   1.0   0.7   0:27.51 gnome-terminal
41998 user      20   0  146420   2392   1428 R   0.7   0.1   0:00.22 top
  138 root      20   0       0      0      0 S   0.3   0.0   0:03.88 rcuos/0
 4233 user      20   0  378504  19100  14816 S   0.3   0.5   1:00.35 vmtoolsd
    1 root      20   0  126080   6812   3932 S   0.0   0.2   0:11.83 systemd
    2 root      20   0       0      0      0 S   0.0   0.0   0:00.08 kthreadd
```

显示结果排序：缺省按照 cpu 使用情况排序；

m 键：按照内存排序；

t 键：运行时间进行排序；

u 键，键入用户名，查看某一用户的 CPU 使用情况；

k 键， 输入 PID，可终止某一进程；

q 键，退出 top。

2.5.2 进程管理命令

1. 进程的前后台

系统中每个进程都有一个进程号，用于系统识别和调度进程，启动一个进程主要有两个途径：手工启动和调度启动（调度任务见 2.6.2 计划任务）。用户输入命令，直接启动一个进程便是手工启动进程，手工启动又分为前台启动和后台启动。

（1）前台启动

前台就是指一个程序控制着标准输出和标准输入，当前台运行一个程序的时候，用户不能再执行其他程序。

（2）后台启动

后台就是指一个程序不从标准输入设备接受输入，一般也不将结果输出到标准输出设备上。

例如，ls -a >text & ，表示 shell 检测到命令后面有一个&，就生成一个子 shell 在后台运行这个程序，并立即显示提示符等待用户输入下一个命令。

特殊情况，cat f1 |grep file|wc －l，同时启动 3 个进程，它们都是当前 shell 的子程序，互称为兄弟进程。

2. 进程的前后台调度

（1）Ctrl+Z 组合键

将当前进程挂起，即调入后台并停止执行。

（2）jobs 命令

查看处于后台的任务列表。

（3）bg 命令

将前台作业切换到后台运行，若没有指定作业号，则将当前作业切换到后台。

（4）fg 命令

将处于后台的进程恢复到前台运行，需指定任务序号。

（5）Ctrl+C 组合键

中断正在执行的命令。

【 例 2-51 】进程前后台调度实例。

```
[user@localhost ~]$ man bg
// Ctrl+Z 将 man bg  挂起
[1]+  已停止                  man bg
[user@localhost ~]$ vi &     //vi 启动后台运行
[2] 23361
[user@localhost ~]$ jobs -l
[1]- 23326  停止                  man bg
[2]+ 23361  停止  (tty  输出)        vim
[user@localhost ~]$ bg 1   //将 1 号作业 man bg 从前台切换到后台
[1]- man bg &
[user@localhost ~]$ fg 1   //将 1 号作业 man bg  从后台恢复到  前台运行
man bg
```

3. kill 命令

终止一个进程可以使用 kill 命令，用于终止指定 PID 号的进程，kill 的命令格式是：

```
kill  [-9]  进程号
-9 选项用于强制终止
```

kill 指令一般和 ps 指令结合使用，如示例 2-x 所示。

【 例 2-52 】从当前终端切换到字符终端 2，并进行登录，查找终端 2 的进程 PID，通过 kill 命令结束，退出 tty2 的终端。

```
[root@localhost ~]# ps -ef|grep tty2
root        25349   25269   0 18:02 tty2        00:00:00 -bash
root        25569    8700   0 18:03 pts/0       00:00:00 grep --color=auto tty2
[root@localhost ~]# kill -9 25349
[root@localhost ~]# ps -ef|grep tty2
root        25575       1   0 18:04 tty2        00:00:00 /sbin/agetty --noclear tty2 linux
root        25660    8700   0 18:04 pts/0       00:00:00 grep --color=auto tty2
```

2.6 其他命令

2.6.1 日期和时间

1. cal 显示日历信息

使用 cal 命令可以显示计算机系统的日历，cal 的命令格式如下：

```
cal  [[[日]月]年]
```

【 例 2-53 】显示本月的日历。

```
[user@localhost ~]$ cal
```

【 例 2-54 】显示 2018 年的日历。

```
[user@localhost ~]$ cal 2018
```

【 例 2-55 】显示 2019 年 5 月份的日历。

```
[user@localhost ~]$ cal  9  2019
```

2. date 显示和设置系统日期和时间

使用 date 命令可以显示和设置计算机系统的日期和时间。只有超级用户有权限使用 date 命令设置日期和时间，普通用户只能使用 date 命令显示日期和时间。

date 的命令格式如下：

```
date [选项] [显示时间格式] （以+开头，后面接格式）
```

常用选项：-d<字符串> 表示指定字符串描述的时间。

【 例 2-56 】显示当前计算机系统的日期和时间。

```
[user@localhost ~]$ date
2019 年  10 月  07 日  星期一  20:13:33 CST
```

【 例 2-57 】按照指定的格式显示计算机系统的日期和时间。

```
[user@localhost ~]$ date +%Y%m%d
20191007
[user@localhost ~]$ date +%r%a%d%h%y
下午  08 时 15 分 45 秒一 0710 月 19
```

//更多格式信息可以通过 man date 帮助来进行查询。

2.6.2 计划任务

如果要在固定的时间上触发某个作业，就需要创建计划任务，计划任务分为一次性计划任务和周期性计划任务。

1. 一次性计划任务

一次性计划任务使用 at 命令，在指定的日期、时间点自动执行预先设置的一些命令操作，属于一次性计划任务

at 的命令格式如下所示：

```
at  [选项]  [HH:MM]  [yyyy-mm-dd]
```

选项：-l 查看用户计划任务

-d 删除用户计划任务

-c 查看 at 计划任务具体内容

使用 at 指定一次性计划任务前需要确保 atd 服务是开启的 service atd start（默认是打开的）。

说明

超级用户任何情况下都可以使用 at，其他用户，是否可以使用取决于两个文件 /etc/at.allow 和/etc/at.deny 文件。/etc/at.allow 如果存在，只有其中列出的用户可以使用；/etc/at.allow 不存在，检查/etc/at.deny，这个文件中的用户不能使用改命令；空的/etc/at.deny 意味着所有用户都可以使用该命令,默认状态为所有用户均可使用该命令。

【 例 2-58 】设置调度，要求在 2019 年 12 月 31 日 23 点 59 分向登陆系统的所有用户发送 happy new year!

```
at  23:59 12312018
at>who
at>wall happy new year!
at><EOT>
```

//输入 at 命令后，系统出现“at>”提示符，等待用户输入执行命令，输入完成后 Ctrl+D 结束输入，屏幕显示 at 调度的执行时间。

【 例 2-59 】设置调度，在当天的 21:00 将/var/log 压缩打包，然后关机。

```
# at 21:00    指定在当天 21 点 00 分执行的计划任务
at>tar  - cjf    log.tar.bz2 /var/log    计划任务内容
at>shutdown –h now
at><EOT>  多条命令输入完毕后按 ctrl+D 快捷键结束
```

```
//#at  - l      查看计划任务
//#at  - c 1  查看编号为 1 的计划任务的具体内容
//#at  - d 1  删除编号为 1 的计划任务
```

2. 周期性计划任务

crontab 命令按照预先设置的时间周期（分钟、小时、天……）重复执行用户指定的命令操作，属于周期性计划任务。主要设置文件包括全局配置文件（位于文件：/etc/crontab）和用户定义的文件（/var/spool/cron/用户名）

管理 cron 计划任务，执行下面的命令格式：

```
编辑计划任务：crontab    -e    [-u    用户名]
查看计划任务：crontab    -l    [-u    用户名]
删除计划任务：crontab    -r    [-u    用户名]（root 用户可以管理指定用户的计划任务，普通用户只能管理自己的计划任务）
```

/etc/crontab 文件是 cron 的默认配置文件，文件中的每一行都代表一项任务，它的格式如下：

分钟   小时   日期   月份   星期   命令

/etc/crontab 文件内容描述如表 2.22 所示。

表 2.22 /etc/crontab 文件内容

字段	说明
分钟	取值为从 0 到 59 之间的任意整数
小时	取值为从 0 到 23 之间的任意整数
日期	取值为从 1 到 31 之间的任意整数
月份	取值为从 1 到 12 之间的任意整数
星期	取值为从 0 到 7 之间的任意整数，0 或 7 代表星期日
命令	要执行的命令或程序脚本

时间数值的特殊表示方法：

```
*          表示该范围内的任意时间
,      表示间隔的多个不连续时间点
-          表示一个连续的时间范围
/          指定间隔的时间频率
```

应用示例：

```
0    17    *    *    1-5        周一到周五每天 17:00
30    8    *    *    1,3,5      每周一、三、五的 8 点 30 分
0    8-18/2    *    *    *      8 点到 18 点之间每隔 2 小时
0    0    */3    *    *         每隔 3 天
```

【 例 2-60 】（root 用户）：每天早上 7:50 自动开启 sshd 服务，22 点 50 时关闭；

每隔 5 天清空一次 FTP 服务器公共目录/var/ftp/pub（如果目录不存在，可以安装 vsftpd，或者自己创建目录）；

每周六的 7:30 时，重新启动 httpd 服务；

每周一、三、五的 17:30 时，打包备份/etc/httpd 目录。

```
[root@localhost ~]# crontab -e
50 7 * * *        /usr/bin/systemctl    start    sshd
50 22 * * *        /usr/bin/systemctl    stop sshd
0 0 */5 * *        /usr/bin/rm    -rf    /var/ftp/pub/*
30 7 * * 6         /usr/bin/systemctl restart httpd
30 17 * * 1,3,5    /usr/bin/tar czvf httpdconf.tar.gz2 /etc/httpd
//执行命令需要绝对路径,可以通过 whereis 来查找绝对路径。
```

【 例 2-61 】（susa 用户）：每周日晚上 23:55 时将“/etc/passwd”文件的内容复制到宿主目录中，保存为 pwd.txt 文件。

```
[root@localhost ~]# crontab     -e     -u    susa
55 23 * * 7    /bin/cp     /etc/passwd     /home/jerry/pwd.txt
```

2.6.3 SHELL 的实用功能

1. tab 自动补齐命令

在 bash 命令提示符下输入命令、程序名或路径时，输入命令的几个开始字符，按 Tab 键可自动补齐命令、程序名或路径名。有时多个命令有相同的开头，按下两次 Tab 键，系统会列出全部命令供用户选择。

2. 命令历史记录

bash 支持历史命令功能，通过上下光标键来选择，history 命令可以查看执行的命令。

Shell 还可以执行指定序号的历史命令,使用方式!num。

当历史命令比较多时，可以结合 grep 命令和管道进行查找使用，如图 2.2 所示。

```
  506  history|grep ls
[root@localhost ~]# !443
ls -l /etc/vimrc
-rw-r--r--. 1 root root 1982 Jan 30  2014 /etc/vimrc
```

图 2.2 历史命令操作举例

3. 设置别名（alias）

（1）设置别名

设置别名的命令格式如下：

```
alias   [别名]=[需要定义别名的命令]
```

例如，以长格式的形式查看当前目录下的所有文件，包括隐藏文件，可以设置别名 alias ll=‘ls –al’。

注意

等号两端不能空格，字符串需要加引号。

（2）查看别名

查看当前用户的所有别名:$ alias

```
[user@localhost ~]$ alias
alias egrep='egrep --color=auto'
alias fgrep='fgrep --color=auto'
```

```
alias grep='grep --color=auto'
alias l.='ls -d .* --color=auto'
alias ll='ls -l --color=auto'
alias ls='ls --color=auto'
alias vi='vim'
alias which='alias | /usr/bin/which --tty-only --read-alias --show-dot --show-tilde'
//显示当前 user 用户下的所有别名
```

（3） 取消别名

取消别名:$ unalias 命令

（4） 设置别名永久生效

为命令取的别名在该次登录期间始终有效。若要别名在每次登录时都有效，需将 alias 命令写到初始化脚本文件（.bashrc）中。

举例：设置别名 rm=’rm –i’，使用 rm 删除命令删除时每次都有提示确认信息，并让别名设置永久生效。

```
具体的步骤如下：
[user@localhost ~]$ vim .bashrc
在脚本文件.bashrc 中增加别名 rm=’rm    -i’，保存退出.bashrc 脚本文件。
[user@localhost ~]$ rm file
rm：是否删除有写保护的普通空文件 "file"？y
//设置别名生效，每次删除文件都有提示确认信息
思考：如何撤销别名永久生效？
```

4. Linux 下获得系统帮助

Linux 系统配置了一些帮助文档，用户可以通过帮助命令查看相关命令的帮助信息，常用的帮助命令有 man、info 等命令。

（1） man

man 命令用于获取 Linux 系统的帮助文档，man 的命令格式如下：

```
man      命令/配置文件
```

使用 man 命令阅读手册页时，可以使用上、下箭头滚动文本，使用 Page Up 和 Page Down 键翻页，按 Q 或 q 键退出阅读环境、按“/”键后查找内容。

man 帮助文档分为 9 个章节，具体的 9 个章节如表 2.23 所示

图 2.23 man 手册的章节含义

代码	代表内容
1	普通的命令
2	内核调用的函数与工具
3	常见的函数与函数库
4	设备文件的说明
5	配置文件
6	游戏

7	惯例与协议
8	管理员可用的命令
9	内核相关的文件

【 例 2-62 】查看 tar 命令的帮助信息。

```
[user@localhost ~]$ man tar
```

man 命令不仅能够查看命令帮助，还能够查看配置文件帮助.

【 例 2-63 】查看配置文件 resolv.conf 的帮助信息。

```
[user@localhost ~]$ man resolv.conf
```

【 例 2-64 】查看系统调用 fork 的帮助信息。

```
[user@localhost ~]$ man    2    fork
或
[user@localhost ~]$ man      fork
```

（2） info

info 命令也可以获取命令的帮助。和 man 命令不同的是，info 命令的帮助信息更容易阅读。在 info 帮助信息中，如果标题的前面有"*"符号，则代表这是一个可以进入查看详细信息的子页面，只要按下回车键就可以进入。

Info 的命令格式如下：

```
info   [命令]
```

【 例 2-64 】使用 info，查看命令 ls 的帮助信息。

```
[user@localhost ~]$ info ls
```

ls 命令的 info 帮助信息中可以查看详细的子页面的标题,info 命令主要是靠快捷键来进行操作的，快捷键操作同 man 命令。

2.7 课后训练

1.完成下列文件和目录的基本操作：

（1）查看当前工作目录为用户的主目录。

（2）在当前家目录下创建一个新目录 mydir

（3）创建空文件 aa.c、 bb.c，并将其拷贝到新目录 mydir 中，然后将 bb.c 更名为 fork.c。

（4）创建 aa.c 的软链接 aa.soft。使用软链接文件 aa.soft 显示 aa.c 文件的内容(编辑 aa.c 文件，内容不限),查看文件 aa.soft aa.c 的文件类型。

（5）建立一个 bb.c 文件的链接文件 bb.hard，查看 bb.c 和 bb.hard 的索引结点是否是一个，要求删除 bb.c 后 bb.hard 还能使用

（6）使用至少 5 种显示文件/etc/passwd 内容的命令，体会各有什么不同。

（7）查找/etc 目录下以 http 开头的文件，保存结果到/tmp/fhttp 种。

（8）查找/etc 下以 shell 开头且与 shell 大小写无关的文件名及其匹配的行（不包括错误信息和父目录），并导出到/tmp/ishell.files。

（9）在文件/root/initial-setup-ks.cfg 中查找到所有包含字符串 boot 的行。将找出的行按照原文的先后顺序拷贝到/root/lines 文件中。

（10）搜索在目录/tmp/src 及其子目录下所有以 yy1 开头的 5 天以上未使用的文件，并将这些文件拷贝到/tmp/old 目录中。（目录和文件需要自己创建并修改访问时间）。

（11）用管道方式分页显示/var 目录下的 内容

2.完成下列文件打包与压缩命令

（1）归档当前家目录下的~/mydir,生成~/mydir.tar。

（2）查看~/mydir.tar 归档文件的内容。

（3）将归档文件~/mydir.tar 解包出来，到指定文件夹 test。

（4）将~/test 目录压缩成 ~/test.tar.gz。

（5）将~/mydir 目录压缩成~/mydir.tar.bz2。

（6）解压缩 test.tar.gz 到当前目录。

（7）解压缩 mydir.tar.bz2 到当前目录。

3.创建指定用户和组：

（1）增加 usergroup 组，GID 号为 6000。

（2）新增 user1 用户，UID 号为 6000，密码为空，并将其附属组加入 usergroup 组中。

（3）新增 user2 用户，密码为 password，将用户的附属组加入 root 组和 usergroup 组。用户的主目录为/user2 目录。

（4）新增 user3 用户，不为用户建立并初始化宿主目录，用户不允许登陆到系统的 shell。

4.设置用户的密码：

（1）设置 user1 用户的密码为 111111。

（2）锁住用户 user2 的密码，查看密码状态，然后解锁用户 user2 的密码。

5.新建目录/var/www/user1，并设置如下权限：

（1）将此目录的所有者设置为 user1，并设置读写执行权限。

（2）将此目录的组设置为 usergroup，并设置读执行权限。

（3）将其它用户的权限设置为只读。

6.使用文字设定法对/root/ab 文件设置权限，用户所有者为读取、写入和执行权限，同组用户为读取和写入权限，而其他用户没有任何权限。

7.使用数字设定法对/root/ab 文件设置权限，用户所有者只有读取和写入权限。

8.将/root/ab 文件的用户所有者更改为用户 user1。

9.设置别名 qstat='ps -Ao pid,tt,user'，设置为永久生效。

10.打开火狐浏览器后台运行，查看进程 id 号，将进程结束。

11.执行 vi a.txt，将进程挂起，然后切换到前台运行。

12.配置一个 cron 任务：用户 susa 必须配置一个定时执行任务，每天在本地时间 14:23 时执行命令/bin/echo hana。（如果 susa 用户不存在，需要自己创建。）

13.登记一个一次性运行命令 ps －aux 的作业，使其在 3:00am 执行（执行时间可以按需求修改）。

14. 按照描述完成下面描述的关键字。

描述	关键字
最基本的用户标识编号	
本地组信息的位置	
本地用户账户密码信息的位置	
/etc/passwd 的第七个字段	
GID	