

第 2 章

Linux Shell 命令操作

或许有很多读者有这样的疑问，目前 GNOME、KDE 等基于 X Window 的桌面环境已经相当完善了，轻点鼠标就可以完成所有的工作，还有必要学习晦涩难记的命令吗？答案是“有必要”。首先，读者是以计算机专业的角度来学习 Linux，因此并非以 Linux 为桌面办公或娱乐工具，目的不同则要求掌握的深度不同。再者，X Window 充其量只是建基于 Linux 内核之上的“一套软件”，并不能利用它完成所有的工作。最后，X Window 相当消耗资源，在以 Linux 为服务器或者以其为嵌入式平台的情况下，很可能并不开启其图形界面功能，因此只有命令行界面可用。所以，掌握好 Linux 的常用命令不仅是必要的，而且是学习 Linux 环境编程所必备的基本技能。本章主要介绍以下几方面的命令行知识。

- 获取帮助
- 通配符、引号、管道和输入输出重定向
- 基础操作
- 浏览及搜索文件系统
- 文件的复制、移动、链接和归档
- 阅读文本文件
- 编辑文本文件
- 文件内容操作命令
- 文件系统操作
- 用户管理
- 文件权限操作
- 进程相关命令
- 网络相关命令

2.1 获 取 帮 助

Linux 命令众多，对于 2.6 版以后的 Linux 内核，系统内置的命令已经超过 3000 多条。不仅命令数多，大部分命令的选项也多，以最常使用的 ls 命令为例，其选项数多达 60 余个。要记住如此众多的命令及其选项的用法是非常困难的。所幸的是，Linux 的联机文档非常丰富，几乎所有的命令都有相关的帮助文档。所以，学习命令的第一步就是：学习如何获取命令的帮助。

2.1.1 --help 选项

几乎所有的 Linux 命令都提供--help 选项以列出命令的用法、作用、选项的含义等信息。--help 选项是人们最常使用的获取命令帮助的方法。命令清单 2-1 是 Ubuntu12.04.1 中文环境下的 ls 命令的帮助信息。

命令清单 2-1

```
jianglinmei@ubuntu:~$ ls --help
用法: ls [选项]... [文件]...
List information about the FILEs (the current directory by default).
Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.
```

长选项必须使用的参数对于短选项时也是必须使用的。

-a, --all	不隐藏任何以. 开始的项目
-A, --almost-all	列出除. 及.. 以外的任何项目
--author	与-l 同时使用时列出每个文件的作者
-b, --escape	以八进制溢出序列表示不可打印的字符
--block-size=SIZE	scale sizes by SIZE before printing them. E.g., '--block-size=M' prints sizes in units of 1,048,576 bytes. See SIZE format below.
-B, --ignore-backups	do not list implied entries ending with ~
.....
.....	此处省略了大部分帮助选项信息
.....
-x	逐行列出项目而不是逐栏列出
-X	根据扩展名排序
-l	每行只列出一个文件
--help	显示此帮助信息并退出
--version	显示版本信息并退出

SIZE 可以是一个可选的整数，后面跟着以下单位中的一个：

KB 1000, K 1024, MB 1000*1000, M 1024*1024, 还有 G、T、P、E、Z、Y。

使用色彩来区分文件类型的功能已被禁用，默认设置和 --color=never 同时禁用了它。

使用 --color=auto 选项，ls 只在标准输出被连至终端时才生成颜色代码。

LS_COLORS 环境变量可改变此设置，可使用 dircolors 命令来设置。

退出状态：

- 0 正常
- 1 一般问题（例如，无法访问子文件夹）
- 2 严重问题（例如，无法使用命令行参数）

请向 bug-coreutils@gnu.org 报告 ls 的错误

GNU coreutils 的主页：<http://www.gnu.org/software/coreutils/>

GNU 软件一般性帮助：<http://www.gnu.org/gethelp/>

请向http://translationproject.org/team/zh_CN.html 报告 ls 的翻译错误

要获取完整文档，请运行：info coreutils 'ls invocation'

从清单可见，`--help` 首先列出了命令的用法（省略号表示前面的项可以重复出现多次）；然后说明了命令的作用（这里是英文，目前各 Linux 发行版的中文环境并不完善，所多地方中英文都会夹杂出现）；接下来是按字母顺序排列的各短、长选项的含义的说明；随后是对选项中的一些参数的补充说明；再后，指明了命令的退出状态值的含义，在 Shell 中，约定总是以状态 0 表示命令执行成功；最后是提示用户如何报告自己在使用该命令的过程中发现的问题的方法。

2.1.2 man

`man` 是 `manual` 的缩写，指的是 Linux 的系统手册。使用 `man` 命令可以显示系统手册页中的内容，每个手册页中的内容大多数都是对程序、命令、系统调用、C 标准库函数等的解释信息。

`man` 命令的基本用法是：

```
man [选项] [章节] 手册页.....
```

例如：

```
man -i 1 Ls
```

其中选项`-i` 指明查找手册页时忽略大小写，1 表示到手册页的第 1 章节查找（不指定章节则会查找所有章节），`Ls` 是要查找的手手册页。Linux 将每个手册页划分为 9 个章节，但并非每个章节都有内容，各章节有具体的含义，如表 2-1 所示。

表 2-1 man 的章节号及其含义

章节号	含 义	章节号	含 义
1	可执行程序或 Shell 命令	2	内核提供的系统调用
3	库函数	4	特殊文件（常见于 <code>/dev</code> ）
5	文件格式或约定（如： <code>/etc/passwd</code> ）	6	游戏
7	杂项	8	仅 root 能执行的系统管理命令
9	内核程序		

`man` 能查看的，都是系统内保存的这些命令的文档。这些文档一般都保存在`/usr/share/man` 目录或其子目录下，可以用 `man` 的`-w` 选项显示一个手册页的保存位置。如：

```
jianglinmei@ubuntu:~$ man -w ls
/usr/share/man/man1/ls.1.gz
```

使用 `man` 的`-f` 选项可以显示一个 `man` 手手册页有哪些章节。如：

```
jianglinmei@ubuntu:~$ man -f man
man (7)              - macros to format man pages
man (1)              - an interface to the on-line reference manuals
```

使用 `man` 的`-k` 选项可以在所有手册页中查找指定的关键字，并把含有该关键字的手手册页列举出来。如：

```
jianglinmei@ubuntu:~$ man -k sprintf
asprintf (3)        - print to allocated string
sprintf (3)          - formatted output conversion
vasprintf (3)        - print to allocated string
vsprintf (3)         - formatted output conversion
```

使用 man 命令查看手册页时，显示的是一篇文档，通过一些快捷键可以帮助用户在文档中浏览翻阅。表 2-2 列出了这些快捷键的作用。

表 2-2

查看手册页可使用的快捷键

快 捷 键	作 用	快 捷 键	作 用
空格/PgDn	向下翻页	b/PgUp	向上翻页
/<查找内容>	向下搜索并高亮查找内容	?<查找内容>	向上搜索并高亮查找内容
n	继续/和?的搜索	q	退出

2.1.3 info

在所有类 UNIX 操作系统中，都可以利用 man 来查看指令或相关档案的用法。但是，在 Linux 中，又额外提供了一种在线求助的方法，那就是 info 指令。

man 和 info 就像两个集合，它们有一个交集部分。基本上，info 指令的结果与 man 指令差不多。但与 man 相比，info 工具可显示更完整的最新的 GNU 工具信息。如果 man 页包含的某个工具的概要信息在 info 中也有介绍，那么 man 页中会有“请参考 info 页更详细内容”的字样。通常情况下，man 工具显示的非 GNU 工具的信息是唯一的，而 info 工具显示的非 GNU 工具的信息是 man 页内容的副本补充。

info 帮助，是以类似 html 文件格式组织，即支持回退，超链接等操作。表 2-3 列出了阅读 info 帮助可使用的快捷键及其作用。

表 2-3

查看 info 页可使用的快捷键

快 捷 键	作 用	快 捷 键	作 用
空格/PgDn	向下翻页	PgUp	向上翻页
tab	跳转到下一个超文本连接	回车	进入到光标下的超文本连接
b	跳到文档头	e	跳到文档尾
/<查找内容>	搜索并高亮查找内容	q	退出

2.2 通配符、引号、管道和输入输出重定向

2.2.1 通配符

通配符是用于匹配某种模式的特殊字符。Linux 下常用的通配符如表 2-4 所示。

表 2-4

Linux 下常用的通配符

通 配 符	匹 配
*	匹配 0 或多个任意字符
?	匹配且仅匹配一个任意字符
[abcde]	匹配方括号中列出的任意一个字符
[a-e]	匹配方括号中“-”两端字符之间的任意一个字符
[!abcde]	匹配方括号中未列出的任意一个字符

续表

通配符	匹配
[!a-e]	匹配不在方括号中“-”两端字符之间的任意一个字符
{debian,linux}	完整匹配花括号之间以逗号分隔的任意一个字符串

应当注意的是，文件名前的“.”和路径中的“/”必须显式匹配。如：

```
*file 不能匹配.profile  
/etc*.c 不能匹配/etc 目录下还后缀“.c”的文件
```

仅举几个通配符使用的示例如下：

```
jianglinmei@ubuntu:~$ ls *.txt  
10.txt 11.txt 1.txt 20.txt 21.txt 2.txt abc10.txt abc20.txt  
jianglinmei@ubuntu:~$ ls {20,abc}*.*txt  
20.txt abc10.txt abc20.txt  
jianglinmei@ubuntu:~$ ls [1-2]??.txt  
10.txt 11.txt 20.txt 21.txt
```

以上，第一条命令显示所有扩展名为 txt 的文件，第二条命令显示以 20 或 abc 开头的扩展名为 txt 的文件，最后一条命令显示以字符 1 或 2 开头后面有且仅有一个字符的扩展名为 txt 的文件。

2.2.2 转义字符

在 Linux Shell 中，当反斜线 (\) 后面的字符是\$、`、“”、\、换行符(、)、{、}、?、+、|、*、! 等特殊字符时，该反斜线作为**转义字符**使用，它的作用是指示 Shell 不要对其后的特殊字符进行特殊处理，仅当做普通字符。

```
jianglinmei@ubuntu:~$ ls *.txt  
10.txt 11.txt 1.txt 20.txt 21.txt 2.txt abc10.txt abc20.txt  
jianglinmei@ubuntu:~$ ls \*.txt  
ls: 无法访问*.txt: 没有那个文件或目录
```

2.2.3 引号

在 Linux Shell 中有三种引号，它们是：单引号、双引号和反（倒）引号。

单引号，单引号是一种强引用，由单引号括起来的任何字符都作为普通字符对待。如：

```
jianglinmei@ubuntu:~$ echo 'echo "directory is $HOME"'  
echo "directory is $HOME"  
jianglinmei@ubuntu:~$ ls *.txt  
10.txt 11.txt 1.txt 20.txt 21.txt 2.txt abc10.txt abc20.txt  
jianglinmei@ubuntu:~$ ls '*.*txt'  
ls: 无法访问*.txt: 没有那个文件或目录
```

反(倒)引号，反引号(“`”在键盘上与“~”处于同一个键)括起来的字符串被 Shell 解释为一条命令，Shell 会先执行反引号中的命令，并以它的标准输出结果取代整个反引号部分。如：

```
jianglinmei@ubuntu:~$ echo current directory is `pwd`  
current directory is /home/jianglinmei
```

反引号还可以嵌套使用。但应注意，嵌套使用时内层的反引号必须用反斜线 (\) 转义。如：

```
jianglinmei@ubuntu:~$ echo `echo current directory is \`pwd\``  
current directory is /home/jianglinmei
```

双引号，双引号是一种弱引用，由双引号括起来的字符（除“\$”、反引号“`”和反斜线“\”外）均作为普通字符对待。并且只有当“\”后面是“\$”、“`”、“””、“\”或换行符之一时，“\”才作为转义字符。如：

```
jianglinmei@ubuntu:~$ echo "current directory is `pwd`"  
current directory is /home/jianglinmei  
jianglinmei@ubuntu:~$ echo "home directory is $HOME"  
home directory is /home/jianglinmei  
jianglinmei@ubuntu:~$ echo "file*.?"  
file*.?  
jianglinmei@ubuntu:~$ echo "directory '$HOME'"  
directory '/home/jianglinmei'
```

2.2.4 管道

管道是 Linux 中很重要的一种通信方式，其作用是把一个程序的输出直接连接到另一个程序的输入。在 Linux Shell 中使用“|”符号表示管道，用以连接两个命令，格式如：命令 1 | 命令 2。“|”符号的作用是把左边命令 1 的输出作为右边命令 2 的输入。

管道在 Linux 命令操作中的使用非常方便，功能强大。仅举一例如下。

```
jianglinmei@ubuntu:~$ ls -l | wc -l  
17
```

示例中的这一条命令由两个命令组成，“|”左边的 ls -l 和右边的 wc -l。左边命令的输出是一行一个文件的文件信息列表，右边命令 wc -l 的作用是输入统计文本的行数并输出，在此其输入即 ls -l 的输出结果。所以整条命令的作用就是统计目录中文件的个数（包括.和..），本例中显示了当前用户家目录下的文件数为 17。

2.2.5 输入、输出重定向

Linux Shell 在启动时会自动打开三个标准文件，标准输入文件、标准输出文件和标准错误输出文件，分别对应文件描述符 0、1 和 2。标准输入文件通常对应终端的键盘，标准输出文件和标准错误输出文件通常都对应终端的屏幕。

通常，一个命令进程从标准输入文件中获取输入数据，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。

直接使用标准输入、输出存在一些不方便的情况，如输入数据不可重用，用户第二次想使用相同的数据时必须重新输入，又如输出到终端屏幕上的信息只能看不能改也不能保存等。使用输入、输出重定向可以解决这些问题。

输入重定向是指把命令（或可执行程序）的标准输入重定向到指定的文件中。也就是说，输入可以不来自键盘，而来自一个指定的文件。输入重定向主要用于改变一个命令的输入源，特别是改变那些需要大量输入的输入源。

输入重定向的一般形式为：命令 < 文件名。例如：

```
jianglinmei@ubuntu:~$ wc -l < /etc/passwd  
38
```

/etc/passwd 文件是 Linux 系统保存用户账户的文件，该文件中一行文本表示一个用户的信息。示例中以/etc/passwd 文件作为 wc -l 的输入，即统计/etc/passwd 文件中文本的行数，亦即统计系统中的用户数。

另一种输入重定向称为 here 文档，它告诉 Shell 当前命令的标准输入来自命令行。here 文档的重定向操作符使用 “<<”。它将一对分隔符（处于 “<<” 和换行符之间的任何字符串）之间的正文作为命令的输入。例如：

```
jianglinmei@ubuntu:~$ wc -l << delim
this is first line
second line
third line
delim
3
```

示例中，here 文档的分隔符是 “delim”，两个 “delim” 之间的所有文本即为 wc -l 命令的输入内容，输出结果 3 即为两个 “delim” 之间的所有文本的行数。

输出重定向是指把命令（或可执行程序）的标准输出或标准错误输出重新定向到指定文件中。这样，该命令的输出就不显示在屏幕上，而是写入到指定文件中。

输出重定向的一般形式为：命令 > 文件名。例如：

```
jianglinmei@ubuntu:~$ ls -l abc*.txt > dir.out
jianglinmei@ubuntu:~$ cat dir.out
abc10.txt
abc20.txt
```

示例中，cat 命令的作用是显示文件的内容。应当注意，如果 “>” 符号右边的文件已经存在，那么这个文件将会被覆盖。在这种情况下，可以使用输出附加定向符 “>>”。“>>” 符号的用法和 “>>” 一样。只是当文件存在时，不会覆盖文件而是将新的内容追加到文件末尾。例如：

```
jianglinmei@ubuntu:~$ ls -l 2*.txt >> dir.out
jianglinmei@ubuntu:~$ cat dir.out
abc10.txt
abc20.txt
20.txt
21.txt
2.txt
```

和程序的标准输出重定向一样，程序的错误输出也可以重新定向。使用符号 “2>”（或追加符号 “2>>”）表示对错误输出设备重定向。例如：

```
jianglinmei@ubuntu:~$ ls *.txt 2> err.txt
jianglinmei@ubuntu:~$ cat err.txt
ls: command not found
```

示例中 ls 命令并不存在，但是因为错误输出被重定向到了 err.txt 文件，所以屏幕上没有任何结果，而在 err.txt 文件中记录了错误信息：ls: command not found。

另外，还可以使用另一个输出重定向操作符 “&>” 将标准输出和错误输出同时重定向到同一文件中。例如：

```
jianglinmei@ubuntu:~$ ls ./notexistedfile &> err.txt
jianglinmei@ubuntu:~$ cat err.txt
```

```
ls: 无法访问./notexistedfile: 没有那个文件或目录
```

使用操作符“2>&1”（注意 2>&1 之间不能有空格）可以将标准错误输出关联到标准输出，例如：

```
jianglinmei@ubuntu:~$ ls ./notexistedfile > err.txt 2>&1
jianglinmei@ubuntu:~$ cat err.txt
ls: 无法访问./notexistedfile: 没有那个文件或目录
```

示例中，首先将标准错误输出重定向到文件 err.txt，然后将标准错误输出关联到标准输出，最终的作用即将标准输出和错误输出同时重定向到文件 err.txt。

2.3 基础操作

2.3.1 sudo

Linux 的系统权限管理非常严格。从系统安全的角度考虑，即使是系统管理员，一般也不建议以 root 身份登录系统，有的 Linux 发行版更是在安装时就默认设置 root 用户不能登录。但是，普通用户通常不具有特权命令（如/usr/sbin 目录下的命令）的使用权限，作为系统管理员如果要执行特权命令就需要使用 sudo 命令。

sudo 命令的作用是：以其他用户（默认是超级用户 root）的身份去执行另一个命令。其一般用法很简单，把 sudo 写在任何要执行的其他命令之前即可，第一次使用 sudo 命令时，系统会要求用户输入自己的密码。例如：

```
jianglinmei@ubuntu:~$ sudo cat /etc/passwd
[sudo] password for jianglinmei:
```

当然，并非任意用户都能使用 sudo。要给予用户使用 sudo 的权限，必须在 sudo 的配置文件（默认是/etc/sudoers）中进行权限设置。

2.3.2 路径和当前工作目录

Linux 使用“路径”来标识一个文件或目录在整个文件系统中的位置。路径有绝对路径（也叫全路径）和相对路径之分。书写时，路径由目录名和文件名组成，目录名和目录名之间以及目录和文件名之间由“/”分隔，如：/home/jianglinmei/file.txt。

在本书第 1 章介绍过，Linux 采用的是多级目录树型层次结构，树型结构最上层是根目录，用“/”表示，其他的所有目录都是从根目录出发而生成的。**绝对路径**是文件或目录相对于根目录的路径，由文件系统的整棵目录树上从根目录到该文件或目录之间的所有节点组成。绝对路径总是以“/”开头。

用户在使用 Shell 时，总是工作在某一个目录下，比如初登录时一般处在自己的家目录下，家目录可用一个特殊的符号“~”表示。用户当前操作所处的目录即称为**当前工作目录**。使用 pwd 命令可以显示当前工作目录，如：

```
jianglinmei@ubuntu:~$ pwd
/home/jianglinmei
```

使用 cd 命令可以更改当前工作目录到其他目录，如：

```
jianglinmei@ubuntu:~$ cd /etc
jianglinmei@ubuntu:/etc$
```

cd 命令执行后，不会有任何输出信息。但是，可以注意到命令提示信息中当前路径部分改变了。上例中的“~”变成了“/etc”。

相对路径即相对于当前工作目录的路径。相对路径是以`.`或`..`开始的，其中`.`一般可省略。`.`表示当前目录，“`..`”表示上级目录。如以下更改当前目录的操作（注意命令行提示信息中当前目录的变化）：

```
jianglinmei@ubuntu:/usr/local/sbin$ cd ../bin/
jianglinmei@ubuntu:/usr/local/bin$ cd ../..
jianglinmei@ubuntu:/usr$ cd bin
jianglinmei@ubuntu:/usr/bin$ 
jianglinmei@ubuntu:/usr/bin$ cd ./X11/
jianglinmei@ubuntu:/usr/bin/X11$
```

2.3.3 创建和删除文件

使用 touch 命令可以将文件的访问时间和修改时间更改为当前时间。默认情况下，如果文件不存在的话，则将会创建一个空文件。例如：

```
jianglinmei@ubuntu:~$ ls blank.txt
ls: 无法访问 blank.txt: 没有那个文件或目录
jianglinmei@ubuntu:~$ touch blank.txt
jianglinmei@ubuntu:~$ ls blank.txt
blank.txt
```

在 Linux 中，删除文件的命令是 rm。例如：

```
jianglinmei@ubuntu:~$ rm blank.txt
jianglinmei@ubuntu:~$ ls blank.txt
ls: 无法访问 blank.txt: 没有那个文件或目录
```

rm 命令有两个常用的选项，一个是“`-f`”，另一个是“`-r`”。“`-f`”选项的作用是强制删除文件，忽略不存在的文件，也不会给出提示信息。“`-r`”选项的作用是递归删除目录及其内容，使用该选项必须特别小心，因为它会把指定目录下的所有文件以及子目录全都删除掉。例如：

```
jianglinmei@ubuntu:~$ ls
10.txt 1.txt 21.txt abc20.txt 模板 图片 下载 桌面
11.txt 20.txt 2.txt 公共的 视频 文档 音乐
jianglinmei@ubuntu:~$ rm -rf 音乐
jianglinmei@ubuntu:~$ ls
10.txt 1.txt 21.txt abc20.txt 模板 图片 下载
11.txt 20.txt 2.txt 公共的 视频 文档 桌面
```

rm 命令的另一个选项“`-i`”常用于使用通配符删除多个文件时。使用该选项，Shell 会要求用户逐一确认每个文件是否确定要删除，输入“y”并回车表示确定删除，其他字符表示不删除。例如：

```
jianglinmei@ubuntu:~$ rm -i abc*.txt
```

```
rm: 是否删除普通空文件 "abc10.txt"? y
rm: 是否删除普通文件 "abc20.txt"? n
jianglinmei@ubuntu:~$ ls abc*.txt
abc20.txt
```

2.3.4 创建和删除目录

在 Linux Shell 下，使用 `mkdir` 命令创建目录。该命令的用法较为简单，直接在命令后接目录名（可以是绝路路径也可以是相对路径）即可创建指定的目录，例如：

```
jianglinmei@ubuntu:~$ mkdir firstdir
jianglinmei@ubuntu:~$ ls
10.txt 1.txt 21.txt abc10.txt firstdir 模板 图片 下载 桌面
11.txt 20.txt 2.txt abc20.txt 公共的 视频 文档 音乐
jianglinmei@ubuntu:~$ ls /tmp
orbit-gdm pulse-2L9K88eMlGn7 pulse-PKdhtXMmr18n
jianglinmei@ubuntu:~$ mkdir /tmp/tmpdir
jianglinmei@ubuntu:~$ ls /tmp
orbit-gdm pulse-2L9K88eMlGn7 pulse-PKdhtXMmr18n tmpdir
```

默认情况下，如果要创建的目录的父目录不存在，则 `mkdir` 命令会报错，目录创建失败。但可以使用选项 “`-p`” 指示如果要创建的目录的父目录不存在，一并创建其父目录。例如：

```
jianglinmei@ubuntu:~$ mkdir /tmp/theother/somedir
mkdir: 无法创建目录"/tmp/theother/somedir": 没有那个文件或目录
jianglinmei@ubuntu:~$ mkdir -p /tmp/theother/somedir
jianglinmei@ubuntu:~$ ls /tmp/theother/
somedir
```

使用 `rmdir` 命令可以删除目录。但是要注意，该命令只能删除空目录，即不含任何文件的目录。例如：

```
jianglinmei@ubuntu:~$ ls
10.txt 1.txt 21.txt abc10.txt firstdir 模板 图片 下载 桌面
11.txt 20.txt 2.txt abc20.txt 公共的 视频 文档 音乐
jianglinmei@ubuntu:~$ ls firstdir/
notempty.txt
jianglinmei@ubuntu:~$ rmdir firstdir/
rmdir: 删除 "firstdir/" 失败: 目录非空
jianglinmei@ubuntu:~$ rm firstdir/notempty.txt
jianglinmei@ubuntu:~$ ls firstdir/
jianglinmei@ubuntu:~$ rmdir firstdir/
jianglinmei@ubuntu:~$ ls
10.txt 1.txt 21.txt abc10.txt 公共的 视频 文档 音乐
11.txt 20.txt 2.txt abc20.txt 模板 图片 下载 桌面
```

2.3.5 查看用户、日期和输出简单信息

在 Linux 中使用 `who` 命令可以查看所有正在使用的用户的用户名、所用终端、登录时间等信息，使用 `whoami` 命令可以查看当前用户的信息。例如：

```
jianglinmei@ubuntu:~$ who
```

```
jianglinmei pts/1      2012-09-17 09:35 (10.8.18.212)
jianglinmei tty7      2012-09-17 09:40
jianglinmei pts/2      2012-09-17 09:41 (:0)
cliff   pts/4      2012-09-17 09:45 (10.8.18.212)
jianglinmei@ubuntu:~$ whoami
jianglinmei
```

在 Linux 中与日期相关的命令主要有两个。date 命令用于在屏幕上显示或设置系统的日期和时间，不带参数时将直接显示系统当前时间。例如：

```
jianglinmei@ubuntu:~$ date
2012 年 09 月 17 日 星期一 09:50:00 CST
```

使用 cal 命令可以显示公元 1 ~ 9999 年中任意一年或一个月的日历。不带参数时，显示当前月份的日历；只带一个参数时，该参数被解释为年份；带两个参数时第一个参数表示月份，第二个参数表示年份。例如：

```
jianglinmei@ubuntu:~$ cal 11 2012
November 2012
Su Mo Tu We Th Fr Sa
      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
```

在 Linux Shell 中输出简单信息的命令是 echo，前面已经有相关的示例。echo 命令将命令行中的参数显示到标准输出上。echo 命令常用于 Shell 脚本（详见第 3 章），用为一种输出提示信息的手段。如果要原样输出参数，则应将它的参数用引号括起来，否则 echo 将参数拆分为单词，然后各单词以一个空格分隔输出。echo 有两个常用的选项 “-e” 和 “-n”。

“-e” 作用为：保留 “\” 的转义作用，常用于换行符 “\n”。例如：

```
jianglinmei@ubuntu:~$ echo "first line\nsecond line"
first line\nsecond line
jianglinmei@ubuntu:~$ echo -e "first line\nsecond line"
first line
second line
```

“-n” 作用为：输出后不换行。例如：

```
jianglinmei@ubuntu:~$ echo -n "Please input password: "
Please input password: jianglinmei@ubuntu:~$
```

此处，因 “-n”的作用，echo 的输出没有换行，下一条命令提示信息紧接显示。

最后，使用 clear 命令可以将屏幕上的内容清空，只留一行命令提示符。

2.3.6 命令历史和名称补全

bash 为每个用户维护一个命令历史文件，即用户家目录下的 “`~/.bash_history`”。用户每执行一条命令，该命令就会自动加入到该历史文件中。使用命令历史机制，用户可以方便地调用或修改并执行以前使用过的命令。

最简单的调用历史命令的方法是使用键盘的上键头 “↑” 和下键头 “↓” 键。“↑” 键向后翻

阅历史命令，“↓”向前翻阅历史命令，当所需的历史命令显示在命令行上时，用户可以直接按回车执行，也可以对其编辑后按回车执行。

使用 history 命令可以显示历史命令列表。例如：

```
jianglinmei@ubuntu:~$ history
.....
2006  ls
2007  vi /home/jianglinmei/.bash_history
2008  history
```

历史命令列表每行前面的数字表示相应命令行在命令历史表中的序号，称为历史事件号，越后执行的命令的事件号越大。

history 命令的历史列表往往很长，可以在命令后给出一个数值参数 n，则仅显示最后 n 条命令。例如：

```
jianglinmei@ubuntu:~$ history 2
2008  history
2009  history 2
```

使用特殊字符“!”可以执行历史命令，常用的格式有两种：!n 和!-n（n 为一个整数）。前者，表示执行第 n 条（历史事件号为 n）命令；后者表示执行倒数第 n 条历史命令。例如：

```
jianglinmei@ubuntu:~$ !2006
ls
10.txt 1.txt 21.txt abc20.txt 模板 图片 下载
11.txt 20.txt 2.txt 公共的 视频 文档 桌面
```

显示结果的第一行“ls”为所执行的命令。

bash 为用户输入命令的方便，提供了**名称补全**的功能。这一功能是通过 Tab（制表符）键完成的。当用户在命令行上输入文件名的前部任意个字符并按下 Tab 键后，如果系统能唯一确定是哪个文件，则会自动补全相应的名称。如果系统找到多个文件，会把文件名补全到这些文件名相同部分的最后一个字符。如果系统无法确定相应的名称，会响铃提示，用户可连续按两次 Tab 键，系统会列出备选的文件列表。

2.3.7 ls 命令

ls 命令是 Linux 中使用最为频繁的命令之一。ls 命令的作用是显示文件指定目录（未指定时默认为当前工作目录）下的文件信息。

- 一般格式

```
ls [选项]... [文件]...
```

- 说明

如果参数是目录文件，则列出该目录下的文件信息；如果参数是文件，则列出有关该文件属性的一些信息。默认情况下，按字母顺序排列各输出条目。如果没有给出参数，则显示当前工作目录下的文件信息。

- 常用选项

-a, --all	显示所有文件（包括以“.”开头的隐藏文件）
-----------	-----------------------

-t	按文件的修改时间排序
-d, --directory	当遇到目录时列出目录本身而非目录内的文件
-F, --classify	加上文件类型的指示符号(“*/=@ ”中的一个),各指示符号的含义:“*”表示可执行文件,“/”表示目录,“=”表示socket文件,“@”表示链接文件,“ ”表示管道文件
-i, --inode	在每个文件的最前面显示inode号
-l	使用较长格式列出文件信息。该选项显示结果的第一列是与文件类型和文件访问权限相关的信息(分别参考本书第1.3.2.3小节和1.3.2.4小节),第二栏显示的数字是目录下的文件数或是文件的硬连接数
-h, --human-readable	以易于阅读的格式输出文件大小(单位为K、M、G),应配合-l选项一起使用

● 示例

(1) 按文件最后修改时间的顺序以长列表格式列出当前目录下扩展名为txt的文件,并标出文件的属性。

```
jianglinmei@ubuntu:~$ ls -ltF *.txt
-rw-r--r-- 1 jianglinmei jianglinmei 18 2012-09-17 21:21 abc20.txt
-rw-r--r-- 1 jianglinmei jianglinmei 16 2012-09-17 21:21 2.txt
-rw-r--r-- 1 jianglinmei jianglinmei 14 2012-09-17 21:21 1.txt
-rw-r--r-- 1 jianglinmei jianglinmei 0 2012-09-17 21:14 10.txt
-rw-r--r-- 1 jianglinmei jianglinmei 0 2012-09-17 21:14 11.txt
-rw-r--r-- 1 jianglinmei jianglinmei 0 2012-09-17 21:14 20.txt
-rw-r--r-- 1 jianglinmei jianglinmei 0 2012-09-17 21:14 21.txt
-rw-r--r-- 1 jianglinmei jianglinmei 0 2012-09-17 21:14 abc10.txt
```

(2) 以长列表格式列出当前目录下的扩展名为txt的文件,并标出文件的inode号。

```
jianglinmei@ubuntu:~$ ls -li *.txt
262303 -rw-r--r-- 1 jianglinmei jianglinmei 0 2012-09-17 21:14 10.txt
277720 -rw-r--r-- 1 jianglinmei jianglinmei 0 2012-09-17 21:14 11.txt
277745 -rw-r--r-- 1 jianglinmei jianglinmei 14 2012-09-17 21:21 1.txt
277725 -rw-r--r-- 1 jianglinmei jianglinmei 0 2012-09-17 21:14 20.txt
277729 -rw-r--r-- 1 jianglinmei jianglinmei 0 2012-09-17 21:14 21.txt
277743 -rw-r--r-- 1 jianglinmei jianglinmei 16 2012-09-17 21:21 2.txt
277737 -rw-r--r-- 1 jianglinmei jianglinmei 0 2012-09-17 21:14 abc10.txt
277730 -rw-r--r-- 1 jianglinmei jianglinmei 18 2012-09-17 21:21 abc20.txt
```

2.3.8 别名

bash提供了一个内部命令(没有对应的可执行文件的命令)alias,使用该命令可以为复杂的命令取一个简短或有意义的别名,使用别名的效果和使用命令本身的效果完全一样。其一般用法如下。

```
alias 名称=值
```

大多数Linux发行版都会在某个启动脚本文件中设置别名“ll”,用以简化ls -l命令的使用。例如,在Ubuntu12.04.1版本的“~/.bashrc”文件中有如下设置。

```
alias ll='ls -alF'
```

因此,在Ubuntu12.04.1中,用ll命令可以以长格式显示所有文件(包括“.”开头的文件)并加上文件类型的指示符号。

2.4 浏览及搜索文件系统

2.4.1 find

find 命令用于在文件系统的目录树中搜索文件，其功能强大，用法也较为复杂。

- 一般格式

```
find [路径] ... [表达式选项]... [动作表达式]
```

- 说明

默认路径为当前目录。表达式之间可由操作符连接起来，可用操作符包括（优先级递减）：(EXPR)、!EXPR、-not EXPR、EXPR1 -a EXPR2、EXPR1 -and EXPR2、EXPR1 -o EXPR2、EXPR1 -or EXPR2 和 EXPR1, EXPR2。未指定操作符时默认使用 -and。注意：使用圆括号时，因其为一个 Shell 特殊符号，应在圆括号前加一个“\”进行转义。

常用的动作表达式有：-print、-exec 和-ok。

-print	将查找到的文件输出到标准输出，此为默认的动作表达式
-exec	使用方法一般为“-exec command {} \;”，作用是对查找到的文件依次执行 command 操作。注意在“{} \;”是一个固定格式，{} 和 \; 之间要有一个空格
-ok	用法与-exec 基本相同，区别是在对文件执行操作时，会要求用户先确认

- 常用表达式选项

-name filename	查找名为 filename 的文件。如果 filename 含通配符，应将整个 filename 放在一对双引号内
-user username	查找文件属主是 username 的文件
-group groupname	查找文件属组是 groupname 的文件
-mtime -n +n	按文件更改时间查找文件，-n 指 n 天以内，+n 指 n 天以前
-atime -n +n	按文件访问时间查找文件
-ctime -n +n	按文件创建时间查找文件
-nogroup	查无有效属组的文件，即文件的属组在 /etc/groups 中不存在
-nouser	查无有效属主的文件，即文件的属主在 /etc/passwd 中不存在
-newer filename	查找修改时间比 filename 文件更新的文件
-type [bdcplfs]	查找指定类型的文件(块设备、目录、字符设备、管道、符号链接、普通文件或 socket 文件)
-size [+/-]n[ckMG]	查长度为 n 块(每块 512 字节)、n 字节、nK 字节、nM 字节或 nG 字节的文件。“+”表示比指定的大小更大，“-”比指定的大小更小
-depth	使查找在进入子目录前先行查找完本目录
-maxdepth levels	指定最大查找层数
-mount	查文件时不跨越文件系统 mount 点
-follow	如果遇到符号链接文件，就跟踪链接所指的文件

- 示例

(1) 在当前目录及其子目录下查找扩展名为 txt 的文件并以长列表格式显示。

```
jianglinmei@ubuntu:~$ find . -name "*.txt" -exec ls -l {} \;
-rw-r--r-- 1 jianglinmei jianglinmei 0 2012-09-17 21:14 ./20.txt
-rw-r--r-- 1 jianglinmei jianglinmei 0 2012-09-18 16:11 ./subdir/one.txt
-rw-r--r-- 1 jianglinmei jianglinmei 0 2012-09-17 21:14 ./10.txt
-rw-r--r-- 1 jianglinmei jianglinmei 18 2012-09-17 21:21 ./abc20.txt
-rw-r--r-- 1 jianglinmei jianglinmei 14 2012-09-17 21:21 ./1.txt
-rw-r--r-- 1 jianglinmei jianglinmei 0 2012-09-17 21:14 ./11.txt
-rw-r--r-- 1 jianglinmei jianglinmei 16 2012-09-17 21:21 ./2.txt
-rw-r--r-- 1 jianglinmei jianglinmei 0 2012-09-17 21:14 ./21.txt
```

(2) 仅查找当前目录(不含子目录)下扩展名为 txt 的, 字节数小于 10 的文件并以长列表格式显示。

```
jianglinmei@ubuntu:~$ find . -maxdepth 1 -name "*.txt" -size -16c -exec ls -l {} \;
-rw-r--r-- 1 jianglinmei jianglinmei 0 2012-09-17 21:14 ./20.txt
-rw-r--r-- 1 jianglinmei jianglinmei 0 2012-09-17 21:14 ./10.txt
-rw-r--r-- 1 jianglinmei jianglinmei 14 2012-09-17 21:21 ./1.txt
-rw-r--r-- 1 jianglinmei jianglinmei 0 2012-09-17 21:14 ./11.txt
-rw-r--r-- 1 jianglinmei jianglinmei 0 2012-09-17 21:14 ./21.txt
```

(3) 在当前目录及其子目录下查找扩展名为 txt 的, 字节数为 0 或大于 16 的文件并以长列表格式显示。

```
jianglinmei@ubuntu:~$ find . \(-size 0 -o -size +16c \) -name "*.txt" | xargs ls -l
-rw-r--r-- 1 jianglinmei jianglinmei 0 2012-09-17 21:14 ./10.txt
-rw-r--r-- 1 jianglinmei jianglinmei 0 2012-09-17 21:14 ./11.txt
-rw-r--r-- 1 jianglinmei jianglinmei 0 2012-09-17 21:14 ./20.txt
-rw-r--r-- 1 jianglinmei jianglinmei 0 2012-09-17 21:14 ./21.txt
-rw-r--r-- 1 jianglinmei jianglinmei 18 2012-09-17 21:21 ./abc20.txt
-rw-r--r-- 1 jianglinmei jianglinmei 0 2012-09-18 16:11 ./subdir/one.txt
```

此例将 find 的查找结果通过管道送给 xargs 命令去执行。xargs 命令用于“获取标准输入的文本，并将该文本作为一个命令来执行”。xargs 命令常用于管道的右边以对前一个命令的输出作进一步处理。

(4) 删除当前目录及其子目录下所有扩展名为 txt 的文件。

```
jianglinmei@ubuntu:~$ find . -name "*.txt" | xargs rm -f
jianglinmei@ubuntu:~$ ls *.txt
ls: 无法访问*.txt: 没有那个文件或目录
```

2.4.2 which

which 命令的作用是在环境变量\$PATH 设置的目录里查找符合条件的文件，一般是可执行文件。当有多个同名的命令文件存在时，通常使用 which 命令来查看当前生效的命令是哪个目录下的命令。例如：

```
jianglinmei@ubuntu:~$ which ls
/bin/ls
```

2.4.3 whereis

whereis 根据文件名搜索二进制文件、手册页文件或源代码文件。当未指定选项时，会将搜索

到的这三类文件都显示出来。例如：

```
jianglinmei@ubuntu:~$ whereis printf
printf: /usr/bin/printf /usr/include/printf.h /usr/share/man/man3/printf.3.gz
/usr/share/man/man1/printf.1.gz
```

也可以使用下列选项指定只搜索某类文件。

- b 只找二进制文件
- m 只找手册页文件
- s 只找源文件

例如：

```
jianglinmei@ubuntu:~$ whereis -m printf
printf: /usr/share/man/man3/printf.3.gz /usr/share/man/man1/printf.1.gz
```

2.4.4 locate

locate 命令在保存文档和目录名称的系统数据库内查找文件名符合指定模式的文件。

- 一般格式

`locate [选项]... 模式`

- 说明

Linux 系统自动创建并维护一个保存文档和目录名称的系统数据库（默认的数据库为 `/var/lib/mlocate/mlocate.db`）。数据库中的条目一般每天自动更新一次，也可以由管理员调用 `updatedb` 命令手动更新。与遍历文件系统进行搜索的 `find` 命令相比，`locate` 命令的优点是速度快，缺点是搜索不到最新变动过的文件。模式中可以使用通配符。

- 常用选项

- `-e, --existing` 只显示当前存在的文件条目。
- `-i, --ignore-case` 匹配模式时忽略大小写区别。
- `-l LIMIT` 限制仅显示 LIMIT 条结果。

- 示例

(1) 查找`/usr/share` 目录及其子目录下含有字符串“`sprintf`”的文件。

```
jianglinmei@ubuntu:~$ locate /usr/share/*sprintf*
/usr/share/info/autosprintf.info.gz
/usr/share/man/man3/asprintf.3.gz
/usr/share/man/man3/sprintf.3.gz
/usr/share/man/man3/vasprintf.3.gz
/usr/share/man/man3/vsprintf.3.gz
```

(2) 忽略大小写查找含有字符串“`mysql`”的文件，仅显示前 4 条结果。

```
jianglinmei@ubuntu:~$ locate -i -l 4 mysql
/etc/apparmor.d/abstractions/mysql
/etc/bash_completion.d/mysqladmin
/usr/lib/pymodules/python2.7/rdflib/store/MySQL.py
/usr/lib/pymodules/python2.7/rdflib/store/MySQL.pyc
```

2.5 阅读文本文件

阅读文本文件的方法有多种，可以用带编辑功能的、图形界面的文字处理软件，也可以用专用于文本阅读的 linux 命令，本节介绍后者。

2.5.1 cat

cat 命令是最简单的文本阅读命令。cat 命令将文件或标准输入连接合并输出到标准输出。cat 命令有两项功能，一是显示文件内容，二是连接合并文件内容。例如：

```
jianglinmei@ubuntu:~$ ls
1.txt 2.txt subdir 公共的 模板 视频 图片 文档 下载 桌面
jianglinmei@ubuntu:~$ cat 1.txt
this is 1.txt
jianglinmei@ubuntu:~$ cat 2.txt
this is 2.txt
jianglinmei@ubuntu:~$ cat 1.txt 2.txt
this is 1.txt
this is 2.txt
```

示例中最后一条命令即将两个文件的内容串连起来一起输出。cat 命令有一个较常用的选项“-n”，其作用是在每行前面显示行号。cat 命令经常和输出重定向一起使用，以将多个文件内容连接起来创建一个新文件。例如：

```
jianglinmei@ubuntu:~$ cat -n 1.txt 2.txt > 3.txt
jianglinmei@ubuntu:~$ cat 3.txt
1 this is 1.txt
2 this is 2.txt
```

cat 命令一个缺点是，当文件较大时，文本在屏幕上一闪而过，用户往往无法看清显示的内容。因此一般使用 more、less 等支持分页显示的命令来查看文本文件。

因 cat 命令不带文件名参数时，接受标准输入作为其输出内容，故可结合 cat 和输出重定向符来创建简单的文本文件。例如：

```
jianglinmei@ubuntu:~$ cat > create_by_cat.txt
this is a file created by "cat"
which is a very useful linux command.
```

注意，cat 可接受多行输入，要结束输入应按“Ctrl + D”组合键。

2.5.2 more 和 less

more 命令分页显示文件内容，每次一屏。more 支持翻页操作，翻到最后一页后会自动退出到 Shell 命令行。基本操作按键如下。

空格	向前翻页
b	向后翻页
q	退出

more 命令的一个常用选项是“+num”，其作用是从第 num 行开始显示。例如：

```
jianglinmei@ubuntu:~$ more +103 /usr/share/vim/vim73/doc/usr_01.txt
```

less 命令和 **more** 命令基本一样，也用来分页显示文件内容。**less** 除支持翻页操作外，还支持一部分基于 **vi** 编辑器的操作命令，如按行滚屏和搜索等。常用操作按键如下。

空格	向前翻页
b	向后翻页
k 或 “↑”	向上滚动一行
j 或 “↓”	向下滚动一行
/<关键字>	搜索
n	继续前一次搜索
q	退出

2.5.3 head 和 tail

head 命令在屏幕上显示指定文件的开头若干行，默认 10 行，可以用选项“-num”指定显示几行。例如：

```
jianglinmei@ubuntu:~$ head -3 /usr/share/vim/vim73/doc/usr_01.txt
*usr_01.txt*  For Vim version 7.3. Last change: 2008 May 07
```

```
VIM USER MANUAL - by Bram Moolenaar
```

tail 命令和 **head** 命令刚好相反，用于在屏幕上显示指定文件的末尾若干行，默认 10 行，可以用选项“-num”指定显示几行。例如：

```
jianglinmei@ubuntu:~$ tail -3 /usr/share/vim/vim73/doc/usr_01.txt
Next chapter: |usr_02.txt| The first steps in Vim
```

```
Copyright: see |manual-copyright| vim:tw=78:ts=8:ft=help:norl:
```

2.6 编辑文本文件

2.6.1 vi

vi 是“Visual Interface”的简称，它汇集了行编辑和全屏幕编辑的特点，成为类 UNIX 系统中最常用的编辑器。在 Linux 中，常用的是 **vi** 的改良版 **vim** (VI Improved)。Linux 下的 **vi** 命令实际是一个 **vim** 命令别名或到 **vim** 命令文件的符号连接。

vi 和一般文本编辑器不同的是，它是一种多模式编辑器，在不同模式下相同的按键所起的作用是不同的。**vi** 共有三种模式：命令模式、输入模式和末行模式。通过特定的按键可以在三种模式中转换，转换方式如图 2-1 所示。

在 Shell 中输入 **vi** 或 **vim** 命令均可打开 **vim** 编辑器，进入命令模式。在命令模式下输入“a”、“i”、“o”和“s”等字符进入到输入模式。在输入模式输入的任何字符（除 Esc 外）均作为文件的文本内容。在输入模式按 Esc 键可回到命令模式。从命令模式进入末行模式的方法是：输入“：“。

末行模式因光标停在屏幕最下方也即屏幕的最后一行而得名，在末行模式输入的任何字符均出现在末行上。在末行模式输入的一般是 vi 所能解释执行的“末行命令”，在用户按下 Esc 键或回车键后可回到命令模式，但若在末行输入“q”、“x”、“wq”或“q!”等命令再回车，则会退出 vim 编辑器回到 Shell 命令行。

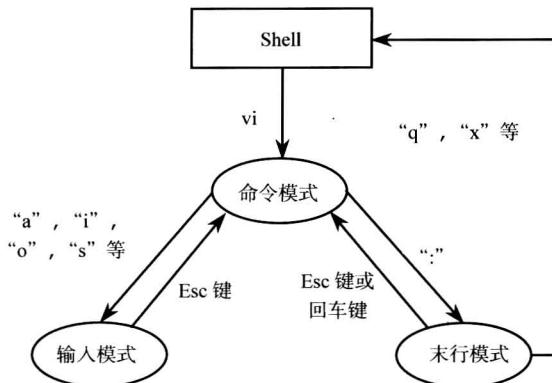


图 2-1 vi 模式转换图

在末行的几个退出命令中，“`q`” 表示正常退出；“`x`” 和 “`wq`” 表示保存后退出；“`q!`” 表示强制退出，常用于对文件进行了更改，要退出而不想保存的情况。

应当注意：掌握模式转换方法是进行 vi 操作的基础，使用 vi 需要频繁地在三种模式之间切换，这也是 vi 初学者最为头痛的地方，读者应反复对此进行练习方能熟练。

vim 中常用的命令如表 2-5 所示。

表 2-5

vim 常用命令列表

移动光标（命令模式）	
[NUM] <code>h</code> 或 左箭头键(<code>←</code>)	光标向左移动一个字符（注：[NUM]为一个整数，是可选部分，表示命令重复次数。如， <code>10h</code> 表示向左移动 10 个字符，下同）
[NUM] <code>j</code> 或 下箭头键(<code>↓</code>)	光标向下移动一个字符
[NUM] <code>k</code> 或 上箭头键(<code>↑</code>)	光标向上移动一个字符
[NUM] <code>l</code> 或 右箭头键(<code>→</code>)	光标向右移动一个字符
[<code>Ctrl</code>] + [<code>f</code>] 或 [<code>PgDn</code>]键	屏幕向下滚动一页
[<code>Ctrl</code>] + [<code>b</code>] 或 [<code>PgUp</code>]键	屏幕向上滚动一页
[<code>Ctrl</code>] + [<code>d</code>]	屏幕『向下』移动半页
[<code>Ctrl</code>] + [<code>u</code>]	屏幕『向上』移动半页
数字 0 或 [<code>Home</code>]键	移动到行头
\$ 或 [<code>End</code>]键	移动到行尾
^	移动到本行第一个非空白字符
H	光标移动到本屏首行的行首
M	光标移动到本屏中间行的行首
L	光标移动到本屏末行的行首
G	移动到文件末行

续表

移动光标（命令模式）	
[NUM]G	移动到第 NUM 行
gg	移动到文件首行，相当于 1G
[NUM]<Enter>键	光标向下移动 NUM 行
w	向前移动到下一个单词的第一个字符
b	向后移动到上一个单词的第一个字符
查找与替换（命令模式）	
/word	从光标处往下查找 word
?word	从光标处往上查找 word
n	重复上一次查找
N	以相反的方向重复上一次查找
查找与替换（末行模式）	
:n1,n2s/word1/word2/g	n1 与 n2 为数字。将第 n1 行与 n2 行之间的 word1 替换为 word2。可用 “\$” 代替 n2 表示文件末行。 例如：“:20,30s/If/if/g”
:n1,n2s/word1/word2/gc	同上。区别在于末尾的“c”，它表示在替换前要求用户确认，用户输入“y”则替换，否则不替换
删除、复制与粘贴	
[NUM]x 或 [Delete]键	删除光标处或光标前的下一个字符
[NUM]X 或 [Backspace]键	删除光标后的上一个字符
[NUM]dd	删除光标所在行或从光标开始往下的 NUM 行
dgg	删除文件第一行到光标所在行的所有行
dG	删除光标所在到文件最后一行的所有行
d\$	删除到行尾
d0	删除到行首
[NUM]yy	复制光标所在行或从光标开始往下的 NUM 行
ygg	复制文件第一行到光标所在行的所有行
yG	复制光标所在到文件最后一行的所有行
y\$	复制到行尾
y0	复制到行首
p	视已复制的内容为字符串还是行，将其粘贴到光标前（下一位置）或下一行
P	视已复制的内容为字符串还是行，将其粘贴到光标后（上一位置）或上一行
J	连接光标所在行与下一行
u	撤销上一个编辑内容
[Ctrl]+r	恢复上一次撤销的内容
点号 “.”	重做上一个命令
末行其他常用命令	
:w	保存文件

续表

末行其他常用命令

:w!	强制保存，能否保存成功与用户对文件的访问权限有关，一般用于只读文件
:w [filename]	另存为别的文件
:r [filename]	读入另一个文件，将其内容添加到光标所在行的下一行
:n1,n2 w [filename]	将第 n1 到 n2 行的内容保存到 filename 文件
:! command	执行 Shell 命令 “command”
:set nu	显示行号
:set nonu	取消显示行号
:set ic	设置搜索时忽略大小写
:set noic	取消搜索时忽略大小写

2.6.2 gedit

尽管 vi 编辑器的功能强大，但因其指令众多，要耐心地反复练习才能熟练。对于初学 Linux 的读者，可用支持鼠标操作的具有图形界面的文本编辑器 gedit 来替代。打开 gedit 的命令为 gedit [文件名]，文件名为可选，若未指定文件名则会打开一个空白文档。

gedit 的界面如图 2-2 所示。gedit 的操作非常简单，可以拖动鼠标选择文本，用快捷键[ctrl+c]进行复制，用快捷键[ctrl+v]进行粘贴。各种操作和 Windows 系列操作系统下的“记事本”非常相似，在此不作过多介绍。



图 2-2 文本编辑器 gedit

2.7 文件内容操作命令

2.7.1 grep

grep 也是 Linux Shell 下最常用的命令之一。grep 用于在文件中或标准输入中查找某项内容，可使用正则表达式进行匹配查找，功能强大。

● 一般格式

```
grep [选项]... PATTERN [文件]...
```

● 说明

在 Linux 下可用的正则表达式有三类，分别如下。

1. 基本正则表达式 (Basic Regular Expression 简称 BREs)
2. 扩展正则表达式 (Extended Regular Expression 简称 EREs)
3. Perl 正则表达式 (Perl Regular Expression 简称 PREGs)

默认的 “PATTERN” 是一个基本正则表达式，要用扩展正则表达式应添加 “-E” 选项，要用 Perl 正则表达式则应添加 “-P” 选项，不用正则表达式 (即查找普通字符串) 则应添加 “-F” 选项。

当使用基本正则表达式时，必须在字符 “?+,|,{},(,)” 符号前加上转义字符 (\) 以屏蔽掉它们在 Shell 中的特殊含义。

应当注意，书写 grep 命令时，先写 “匹配模式”，再写 “文件名”。

利用 grep 可在标准输入中查找模式的功能，常将其放在管道符号的右边以对前一命令的输出时行过滤。例如，以下命令显示当前目录下的所有目录文件 (模式 “^d” 表示以字母 d 开头)。

```
jianglinmei@ubuntu:~$ ls -l | grep ^d
drwxr-xr-x 2 jianglinmei jianglinmei 4096 2012-09-18 17:37 subdir
drwxr-xr-x 2 jianglinmei jianglinmei 4096 2012-09-17 19:30 公共的
drwxr-xr-x 2 jianglinmei jianglinmei 4096 2012-09-17 19:30 模板
drwxr-xr-x 2 jianglinmei jianglinmei 4096 2012-09-17 19:30 视频
drwxr-xr-x 2 jianglinmei jianglinmei 4096 2012-09-17 19:30 图片
drwxr-xr-x 2 jianglinmei jianglinmei 4096 2012-09-17 19:30 文档
drwxr-xr-x 2 jianglinmei jianglinmei 4096 2012-09-17 19:30 下载
drwxr-xr-x 2 jianglinmei jianglinmei 4096 2012-09-17 19:31 桌面
```

● 常用选项

-E, --extended-regexp	PATTERN 是一个可扩展的正则表达式
-F, --fixed-strings	PATTERN 是一组由断行符分隔的定长字符串
-G, --basic-regexp	PATTERN 是一个基本正则表达式
-P, --perl-regexp	PATTERN 是一个 Perl 正则表达式
-f, --file=FILE	从 FILE 中取得 PATTERN
-i, --ignore-case	忽略大小写
-w, --word-regexp	强制 PATTERN 仅完全匹配单词
-x, --line-regexp	强制 PATTERN 仅完全匹配一行
-v, --invert-match	匹配与 PATTERN 相反的模式
-m, --max-count=NUM	查找到 NUM 次匹配后即停止查找
-n, --line-number	输出行号
-R, -r, --recursive	递归到子目录中查找
--include=FILE_PATTERN	仅在文件名与 FILE_PATTERN 匹配的文件中查找
--exclude=FILE_PATTERN	不查找文件名与 FILE_PATTERN 匹配的文件

● 示例

(1) 在文件/usr/share/vim/vim73/doc/usr_01.txt 中查找含有字符串“manual”(忽略大小写)的行，并在输出结果中显示查找到的行的行号。

```
jianglinmei@ubuntu:~$ grep -i -n manual /usr/share/vim/vim73/doc/usr_01.txt
3:          VIM USER MANUAL - by Bram Moolenaar
5:          About the manuals
.....此处省略了若干行结果
185:Copyright: see |manual-copyright| vim:tw=78:ts=8:ft=help:norl:
```

(2) 使用扩展正则表达式在文件/usr/share/vim/vim73/doc/usr_01.txt 中查找以数字开头的行。

```
jianglinmei@ubuntu:~$ grep -E ^[0-9]+ /usr/share/vim/vim73/doc/usr_01.txt
1. The User manual
2. The Reference manual
1. Copy the tutor file. You can do this with Vim (it knows where to find it):
2. Edit the copied file with Vim:
3. Delete the copied file when you are finished with it:
```

(3) 在文件/usr/share/vim/vim73/doc/usr_01.txt 中查找含由两个“|”的行。

```
jianglinmei@ubuntu:~$ grep \|.*\| /usr/share/vim/vim73/doc/usr_01.txt
|01.1| Two manuals
|01.2| Vim installed
.....此处省略了若干行结果
Copyright: see |manual-copyright| vim:tw=78:ts=8:ft=help:norl:
```

2.7.2 sort

sort 命令用以对文本文件的各行进行排序。文本的顺序由系统所使用的字符集决定，对英文字符一般使用 ASCII 码值进行比较。

- 一般格式

```
sort [选项]... [文件]...
```

- 说明

如果不指定文件，则排序内容来自标准输入。排序的比较操作是依据从每一行中提取的一个或多个字段来进行的。默认情况下，以空白字符分隔每个字段。

- 常用选项

-b	忽略前导的空白区域
-f, --ignore-case	忽略字母大小写
-n, --numeric-sort	比较数值而非字符串值
-r, --reverse	逆序输出排序结果
-c, --check	检查输入是否已排序
-k 字段 1[, 字段 2]	比较从字段 1 开始到字段 2 之间的内容，字段 2 省略时默认比较到行尾
-o, --output=文件	将结果写入到文件而非标准输出(与输出重定向相比，该选项允许写入被排序的文件)
-t 分隔符	使用指定的字符代替空白字符作为字段分隔符
-u, --unique	在输出结果中去除重复行

- 示例

以 tosort.txt 文件为比较文件，其内容如下。

```
jianglinmei@ubuntu:~$ cat tosort.txt
banana 3, 10kg
apple 4, 7kg
pear 9, 1kg
orange 7, 3kg
pear 9, 2kg
grape 2, 8kg
```

(1) 以默认方式排序，去除重复行。

```
jianglinmei@ubuntu:~$ sort -u tosort.txt
apple 4, 7kg
banana 3, 10kg
grape 2, 8kg
orange 7, 3kg
pear 9, 1kg
```

(2) 以默认的空白字符为分隔符，对第二个字段进行从大到小排序。

```
jianglinmei@ubuntu:~$ sort -k 2 -r tosort.txt
pear 9, 1kg
pear 9, 1kg
orange 7, 3kg
apple 4, 7kg
banana 3, 10kg
grape 2, 8kg
```

(3) 以“,”为分隔符，对第二个字段进行数值排序。

```
jianglinmei@ubuntu:~$ sort -t , -k 2 -n tosort.txt
pear 9, 1kg
pear 9, 1kg
orange 7, 3kg
apple 4, 7kg
grape 2, 8kg
banana 3, 10kg
```

2.7.3 diff

diff 命令比较两个文本文件，找出它们的不同之处。

- 一般格式

```
diff [选项]... 文件1 文件2.
```

- 说明

diff 命令逐行比较两个文件，列出两者的不同之处。如果两个文件完全一样，则命令不显示任何输出。

- 常用选项

-i --ignore-case	忽略大小写的区别
-b --ignore-space-change	忽略由空格数不同造成的差异
-y --side-by-side	以两列并排的方式显示
-W NUM --width=NUM	每行显示最多 NUM (默认 130) 个字符

● 示例

以 tosort.txt 文件和 tosort2.txt 为比较文件，其内容如下。

```
jianglinmei@ubuntu:~$ cat -n tosort.txt
 1 banana 3, 10kg
 2 apple 4, 7kg
 3 pear 9, 1kg
 4 orange 7, 3kg
 5 pear 9, 1kg
 6 grape 2, 8kg
jianglinmei@ubuntu:~$ cat -n tosort2.txt
 1 banana 3, 10kg
 2 apple 4, 7g
 3 pear 9, 1kg
 4 orange 7, 3kg
 5 pear 9, 1g
 6 grape 2, 8kg
 7 peach 6, 6kg
 8 melon 8, 8g
```

(1) 以普通格式输出。

```
jianglinmei@ubuntu:~$ diff tosort.txt tosort2.txt
2c2
< apple 4, 7kg
---
> apple 4, 7g
5c5
< pear 9, 1kg
---
> pear 9, 1g
6a7,8
> peach 6, 6kg
> melon 8, 8g
```

输出结果中，“2c2”、“5c5” 表示两者第 2 行和第 5 行有差异，“<” 引导的为文件 1 中相应行的内容，“>” 引导的为文件 2 相应行的内容。每个差异之间以 “---” 行分隔。“6a7,8” 表示文件 1 的第 6 行之后，文件 2 多了 7、8 两行。

(2) 以并排格式输出，并指定每行最多 40 字符。

```
jianglinmei@ubuntu:~$ diff -y -W 40 tosort.txt tosort2.txt
banana 3, 10kg      banana 3, 10kg
apple 4, 7kg      |  apple 4, 7g
pear 9, 1kg       |  pear 9, 1kg
orange 7, 3kg      orange 7, 3kg
pear 9, 1kg       |  pear 9, 1g
grape 2, 8kg       grape 2, 8kg
>    peach 6, 6kg
>    melon 8, 8g
```

输出结果中，中间有 “|” 的行表示两者有差异，中间有 “>” 的行表示是文件 2 中多出的行。

2.7.4 wc

wc 命令用于统计文件的行数、字数和字节数。这里的“字”是指由空白字符（如空格、制表

符、换行符等)分隔的字符串。可以同时对多个文件进行统计。不指定任何选项的情况下，会同时统计行数、字数和字节数，输出结果的格式如下。

行数	字数	字节数	文件名
----	----	-----	-----

如果用选项指定了统计哪些项，则其他项的结果不会显示，但显示项的顺序不变。另外，如果未指定文件，则对标准输入进行统计，输出结果中将无“文件名”列。以上一小节所用的两个文件为例。

```
jianglinmei@ubuntu:~$ wc tosort.txt tosort2.txt
 6 18 79 tosort.txt
 8 24 102 tosort2.txt
14 42 181 总用量
```

可以用以下选项指明统计哪一项。

-c, --bytes	统计字节数
-l, --lines	统计行数
-w, --words	统计字数

例如：

```
jianglinmei@ubuntu:~$ wc -l -w tosort.txt tosort2.txt
 6 18 tosort.txt
 8 24 tosort2.txt
14 42 总用量
```

2.8 文件的复制、移动、链接和归档

2.8.1 cp

cp 命令将源文件复制至目标文件，或将多个源文件复制至目标目录。

- 一般格式

```
cp [选项]... 源文件 目标文件
或: cp [选项]... 源文件... 目录
或: cp [选项]... -t 目录 源文件...
或: cp [选项]... -r 源目录... 目录
或: cp [选项]... -a 源目录... 目录
```

- 说明

cp 可以将源文件(非目录文件)复制为目标文件，或将源文件(非目录文件)复制到其他目录下，或者使用“-r”选项将源目录递归地复制到其他目录下，或者使用“-a”选项将一个目录下的所有子目录或文件复制为其他目录(复制前该目录不存在)。

- 常用选项

-a, --archive	复制目录下的文件到另一目录并保留文件的属性
---------------	-----------------------

-d	保留符号链接
-f, --force	覆盖目标文件并且不提示
-i, --interactive	覆盖前询问
-n, --no-clobber	不覆盖已存在的文件
-p	保留文件的属性
--parents	复制前在目标目录创建源文件路径中的所有目录
-R, -r, --recursive	递归复制目录
-t,	指定目标目录 (--target-directory)
-u, --update	只在源文件比目标文件新，或目标文件不存在时才进行复制
-v, --verbose	显示详细的进行步骤
-x, --one-file-system	不跨越文件系统进行操作

● 示例

(1) 本当前目录下的所有扩展名为 txt 的文件复制到 subdir 目录。

```
jianglinmei@ubuntu:~$ cp *.txt subdir/
```

(2) 在当前目录下创建 recurse 目录，然后将 subdir 目录递归地复制到 recurse 目录。

```
jianglinmei@ubuntu:~$ mkdir recurse
jianglinmei@ubuntu:~$ cp -r subdir/ recurse/
jianglinmei@ubuntu:~$ ls recurse/
subdir
jianglinmei@ubuntu:~$ ls recurse/subdir/
1.txt 2.txt 3.txt create_by_cat.txt tosort2.txt tosort.txt
```

(3) 将 subdir 目录下的所有文件(含目录)复制到 archive 目录，并要求保留源文件属性 (即归档)。

```
jianglinmei@ubuntu:~$ cp -a subdir/ archive
jianglinmei@ubuntu:~$ ls archive/
1.txt 2.txt 3.txt create_by_cat.txt tosort2.txt tosort.txt
```

(4) 在当前目录下创建 other 目录，使用 “-t” 选项将 1.txt 和 3.txt 复制到该目录。

```
jianglinmei@ubuntu:~$ mkdir other
jianglinmei@ubuntu:~$ cp -t other/ 1.txt 3.txt
jianglinmei@ubuntu:~$ ls other
1.txt 3.txt
```

2.8.2 mv

mv 命令用于将源文件重命名为目标文件，或将源文件移动至指定目录。

● 一般格式

```
mv [选项]... 源文件 目标文件
mv [选项]... 源文件... 目录
mv [选项]... -t 目录 源文件...
```

● 说明

从 mv 命令的一般格式可见，mv 命令的用法与 cp 命令基本相同，只是可用选项更少，因此

使用上也更简单。

- 常用选项

<code>-i, --interactive</code>	覆盖前询问
<code>-f, --force</code>	覆盖前不询问

- 示例

(1) 将当前目录下的 1.txt 文件改名为 one.txt。

```
jianglinmei@ubuntu:~$ mv 1.txt one.txt
jianglinmei@ubuntu:~$ ls -l 1.txt one.txt
ls: 无法访问 1.txt: 没有那个文件或目录
-rw-r--r-- 1 jianglinmei jianglinmei 3218 2012-09-20 21:34 one.txt
```

(2) 将当前目录下的所有扩展名为 txt 的文件移动到 other 目录，覆盖已存在的文件。

```
jianglinmei@ubuntu:~$ mv -f *.txt other/
jianglinmei@ubuntu:~$ ls other/
1. txt 2.txt 3.txt create_by_cat.txt one.txt tosort2.txt tosort.txt
```

2.8.3 ln

ln 命令可创建硬连接或符号连接。

- 一般格式

`ln [选项]... 源文件 目标文件`

- 说明

连接的对象可以是文件也可以是目录。创建硬连接会使用文件的连接数增加。

- 常用选项

`-s, --symbolic` 创建符号连接而非硬连接

- 示例

(1) 在当前目录下为 other 目录下的 1.txt 创建一个硬连接文件 one.txt。

```
jianglinmei@ubuntu:~$ ln subdir/1.txt one.txt
jianglinmei@ubuntu:~$ ls -l one.txt
-rw-r--r-- 2 jianglinmei jianglinmei 3218 2012-09-21 09:30 one.txt
jianglinmei@ubuntu:~$ ls -l subdir/1.txt
-rw-r--r-- 2 jianglinmei jianglinmei 3218 2012-09-21 09:30 subdir/1.txt
```

从显示结果中可见，连接前后的两个文件的连接数都变成了 2。

(2) 在当前目录下为 other 目录下的 1.txt 创建一个符号连接文件 1.txt。

```
jianglinmei@ubuntu:~$ ln -s subdir/1.txt 1.txt
jianglinmei@ubuntu:~$ ls -l *.txt
lrwxrwxrwx 1 jianglinmei jianglinmei 12 2012-09-21 11:57 1.txt -> subdir/1.txt
-rw-r--r-- 2 jianglinmei jianglinmei 3218 2012-09-21 09:30 one.txt
```

2.8.4 tar

tar 命令是 Linux 下常用的一个文件打包与解包的工具，可将许多个文件一起保存至归档文件

中，以及从归档文件中还原所需的文件。

- 一般格式

```
tar [选项...] [FILE]...
```

- 说明

tar 除了保存文件至归档文件外，还可用于保存文件至磁带。关于磁带操作本书不予介绍，有兴趣的读者可参阅 tar 的手册页。tar 还可同时对其归档内容进行压缩或解压缩操作。

- 常用选项

<code>-c, --create</code>	创建一个新归档
<code>-t, --list</code>	列出归档内容
<code>-x, --extract, --get</code>	从归档中解出文件
<code>-k, --keep-old-files</code>	解压时不要替换存在的文件
<code>-f, --file=ARCHIVE</code>	使用归档文件或 ARCHIVE 设备
<code>-j, --bzip2</code>	通过 bzip2 压缩或解压归档
<code>-J, --xz</code>	通过 xz 压缩或解压归档
<code>-z, --gzip, --gunzip, --ungzip</code>	通过 gzip 压缩或解压归档
<code>-Z, --compress, --uncompress</code>	通过 compress 压缩或解压归档
<code>-v, --verbose</code>	详细列出处理的文件
<code>--exclude=PATTERN</code>	排除以 PATTERN 指定的文件

- 示例

(1) 为当前目录下的 other 目录创建一个归档文件 o.tar，然后查看归档内容。

```
jianglinmei@ubuntu:~$ tar -cf o.tar other/
jianglinmei@ubuntu:~$ tar -tf o.tar
other/
other/create_by_cat.txt
other/1.txt
other/one.txt
other/2.txt
other/tosort.txt
other/3.txt
other/tosort2.txt
```

(2) 为当前目录下的 other 目录创建一个归档并压缩的文件 o.tar.gz，然后将其解压并解包。

```
jianglinmei@ubuntu:~$ tar cvzf o.tar.gz other/
jianglinmei@ubuntu:~$ mkdir tar
jianglinmei@ubuntu:~$ tar -xvzf o.tar.gz -C tar
other/
other/create_by_cat.txt
other/1.txt
other/one.txt
other/2.txt
```

```
other/tosort.txt
other/3.txt
other/tosort2.txt
jianglinmei@ubuntu:~$ ls tar/
other
```

2.9 文件系统操作

2.9.1 挂载——mount

在 Linux 下，挂载文件系统的命令是 `mount`。挂载的作用是把一个文件系统连接到主目录树的一个目录节点（称为“挂载点”），从而使用户能够通过主目录树访问到该文件系统的数据。必须具有超级用户权限才能使用 `mount` 命令。

- 一般格式

```
mount [-t 文件系统类型] [-o 选项] [设备    挂载点目录]
```

- 说明

(1) 文件系统的类型通常不必指定，`mount` 会自动选择正确的类型。常用类型有以下几种。

<code>iso9660</code>	光盘或光盘镜像
<code>msdos</code>	DOS fat16 文件系统
<code>vfat</code>	Windows 9x fat32 文件系统
<code>ntfs</code>	Windows NT ntfs 文件系统
<code>smbfs</code>	Mount Windows 文件网络共享
<code>nfs</code>	UNIX(LINUX) 文件网络共享
<code>vboxsf</code>	VirtualBox 主机共享文件夹
<code>ext3 或 ext4</code>	Linux 文件系统

(2) “-o 选项” 主要用来描述设备的挂载方式，各选项之间以“,” 分隔，常用的选项有以下几种。

<code>loop</code>	用来把一个文件当成硬盘分区挂载上系统
<code>ro</code>	采用只读方式挂载
<code>rw</code>	采用读写方式挂载
<code>iocharset</code>	指定访问文件系统所用字符集

(3) 使用 “-a” 选项可以根据系统配置文件 “/etc/fstab”的内容挂载其中定义的所有文件系统。例如：

```
jianglinmei@ubuntu:~$ sudo mount -a
```

- 示例

先使用 `mkisofs` 命令将当前目录下的文件和子目录制成一个光盘映像文件，如下：

```
jianglinmei@ubuntu:~$ mkisofs -r -J -V mydisk -o test.iso .
```

(1) 创建/mnt/vcd 目录作为虚拟光盘的挂载点，并将 test.iso 挂载上去。

```
jianglinmei@ubuntu:~$ sudo mkdir /mnt/vcd
jianglinmei@ubuntu:~$ sudo mount -t iso9660 -o loop ./test.iso /mnt/vcd
jianglinmei@ubuntu:~$ ls /mnt/vcd
1.txt      h.txt      o.tar      other      subdir  test.iso  模板  图片  下载
archive   one.txt   o.tar.gz  recurse  tar      公共的  视频  文档  桌面
```

(2) 创建/mnt/win_d 目录，并挂载 ntfs 分区到该目录。

```
jianglinmei@ubuntu:~$ sudo mkdir /mnt/win_d
jianglinmei@ubuntu:~$ sudo mount -t ntfs -o iocharset=cp936 /dev/hda5 /mnt/win_d
```

注：本示例应在/dev/hda5 存在并且是一个 ntfs 分区时方能执行成功。

(3) 创建/mnt/vbox_share 目录，并将 VirtualBox 的共享文件夹 temp 挂载到该目录。

```
jianglinmei@ubuntu:~$ sudo mkdir /vbox_share
jianglinmei@ubuntu:~$ sudo mount -t vboxsf temp /mnt/vbox_share
```

在 VirtualBox 中，设置共享主机文件夹的方法如下：在 VirtualBox 主界面中单击工具栏的“设置”按钮，出现“设置”对话框，在该对话框的左侧选择“共享文件夹”，如图 2-3 所示。在该对话框的“共享文件夹列表”栏右侧有一个带“+”的文件夹图标，单击该图标，将弹出“添加共享文件夹”对话框，如图 2-4 所示。在该对话框中选择要共享的文件夹并指定共享名即可，这里的“共享名”将作为 mount 命令中的设备名使用。注意，使用共享主机文件夹，需要“VirtualBox 增强工具”的支持。

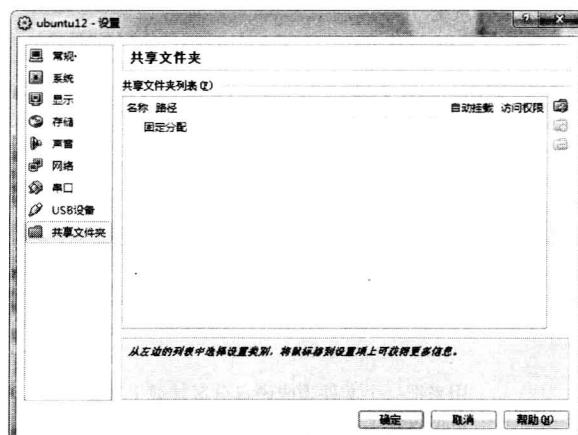


图 2-3 VirtualBox 设置共享文件夹

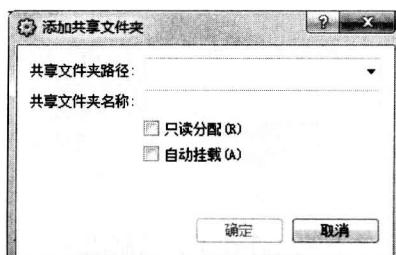


图 2-4 VirtualBox 添加共享文件夹

2.9.2 卸载——umount

卸载是挂载的反操作，即将文件系统从主目录树上脱离出来，卸载后相应的文件系统即无法访问到。卸载文件系统的命令是 umount。其用法较简单，一般格式为：

umount 挂载点

例如：

```
jianglinmei@ubuntu:~$ sudo umount /mnt/vcd
```

2.9.3 查看系统信息

- uname 命令

使用 uname 命令可以查看系统的一般信息。uname 常用选项如下所示。

-s, --kernel-name	输出内核名称
-n, --nodename	输出网络节点上的主机名
-r, --kernel-release	输出内核发行号
-v, --kernel-version	输出内核版本
-m, --machine	输出主机的硬件架构名称
-p, --processor	输出处理器类型或“unknown”
-i, --hardware-platform	输出硬件平台或“unknown”
-o, --operating-system	输出操作系统名称
-a, --all	以上面列出的次序输出所有信息。

例如：

```
jianglinmei@ubuntu:~$ uname -a
Linux jianglinmei-ubuntu 2.6.38-8-generic #42-Ubuntu SMP Mon Apr 11 03:31:50 UTC 2011
i686 athlon i386 GNU/Linux
```

- df 命令

使用 df 命令可以查看文件系统的信息，其常用选项如下所示。

-a, --all	包含虚拟文件系统
--total	显示总计信息
-h, --human-readable	以可读性较好的格式显示尺寸(例如：1K 234M 2G)
-t, --type=类型	只显示指定文件系统为指定类型的信息
-T, --print-type	显示文件系统类型

例如：

```
jianglinmei@ubuntu:~$ df -h
文件系统 容量 已用 可用 已用% 挂载点
/dev/sda1 16G 3.1G 12G 21% /
none 495M 640K 494M 1% /dev
none 501M 212K 501M 1% /dev/shm
none 501M 92K 501M 1% /var/run
none 501M 0 501M 0% /var/lock
```

2.10 用户管理

用户管理操作一般要超级用户才能执行，本节将较简略地介绍最常见的用户管理操作。

- 添加用户——useradd

useradd 的一般格式如下。

```
useradd [选项] 登录名 (即用户名)
```

常用选项有：

-b, --base-dir	指定家目录的父目录
-d, --home-dir	指定家目录
-g, --gid	指定用户的首组 ID
-G	批定用户的辅组 ID 列表
-m, --create-home	创建家目录
-N, --no-user-group	不创建与用户名同名的组
-p, --password	批定密码
-s, --shell	指定默认 Shell
-u, --uid	指定 UID

例如，以下命令将创建 alice 用户，同时为其在/opt 目录下创建家目录，指定其默认 Shell 为 bash。

```
sudo useradd -m -b /opt -s /bin/bash alice
```

- 删除用户——userdel

userdel 的一般格式为：

```
userdel [选项] 登录名 (即用户名)
```

常用选项有：

-f, --force	强制删除
-r, --remove	同时删除家目录和邮件池目录

例如：

```
jianglinmei@ubuntu:/opt/alice$ sudo userdel -f -r alice
```

- 添加组——groupadd

groupadd 的一般格式如下。

```
groupadd [选项] 组名
```

常用选项有：

-g, --gid	指定组 ID
-----------	--------

例如：

```
jianglinmei@ubuntu:~$ sudo groupadd commonuser
```

- 删除组——groupdel

groupdel 的一般格式如下。

```
groupdel 组名
```

例如：

```
jianglinmei@ubuntu:~$ sudo groupdel commonuser
```

- 将用户添加到组或从组中移除——gpasswd

gpasswd 命令的一般格式如下。

```
gpasswd [选项] 组名
```

常用选项有：

-a, --add USER	将 USER 加入组
-d, --delete USER	将 USER 从组中删除
-M, --members USER,...	设置组用户

例如：

- (1) 将用户 alice 加入 root 组。

```
jianglinmei@ubuntu:~$ sudo gpasswd -a alice root
正在将用户“alice”加入到“root”组中
```

- (2) 将用户 alice 从 root 组移除。

```
jianglinmei@ubuntu:~$ sudo gpasswd -d alice root
正在将用户“alice”从“root”组中删除
```

2.11 文件权限操作

- 更改文件属主——chown

chown 命令的一般格式如下。

```
chown [选项]... [所有者] [: [组]] 文件...
```

当指定“组”时，chown 会同时更改文件的所属组。常用选项有：

-h, --no-dereference	仅影响符号链接本身，而非符号链接所指的目的文件
-R, --recursive	递归处理所有的文件及子目录
-v, --verbose	为处理的所有文件显示诊断信息

例如，将 subdir 目录及其子目录和文件的所有者改为 alice，所属组也改为 alice，命令如下。

```
jianglinmei@ubuntu:~$ ls -ld subdir/
```

```
drwxr-xr-x 2 jianglinmei jianglinmei 4096 2012-09-21 09:30 subdir/
jianglinmei@ubuntu:~$ sudo chown -R alice:alice subdir/
jianglinmei@ubuntu:~$ ls -ld subdir/
drwxr-xr-x 2 alice alice 4096 2012-09-21 09:30 subdir/
```

- 更改文件属组——`chgrp`

`chgrp` 命令的一般格式如下。

```
chgrp [选项]... 用户组 文件....
```

常用选项有：

<code>-h, --no-dereference</code>	仅影响符号链接本身，而非符号链接所指的目的文件
<code>-R, --recursive</code>	递归处理所有的文件及子目录
<code>-v, --verbose</code>	为处理的所有文件显示诊断信息

例如，将 `subdir` 目录及其子目录和文件的所属组改为 `jianglinmei`，命令如下。

```
jianglinmei@ubuntu:~$ ls -ld subdir/
drwxr-xr-x 2 alice alice 4096 2012-09-21 09:30 subdir/
jianglinmei@ubuntu:~$ sudo chgrp jianglinmei subdir/
jianglinmei@ubuntu:~$ ls -ld subdir/
drwxr-xr-x 2 alice jianglinmei 4096 2012-09-21 09:30 subdir/
```

- 更改文件权限——`chmod`

`chmod` 命令的一般格式如下。

```
chmod [选项]... 模式[,模式]... 文件...
```

或：`chmod` [选项]... 八进制模式 文件...

`chmod` 命令总是会更改符号连接的目标文件的权限，而不是符号连接本身的权限。`chmod` 命令的权限模式可用字符串形式表示也可用八进制数形式来表示。使用字符串形式，其格式应符合正则表达式：“`[ugoa]*([-+=][[rwxXSt]*|[ugo]])+|`”。使用八进制形式时，模式应由三或四位八进制构成。使用三位八进制时，从最高到最低位依次表示文件所有者（属主）、文件所属组和其他用户的权限，每个八进制数的三个二进制位从高到低又分别表示有无读、写、执行权限（1 为有该权限，0 为没有该权限）；使用四位八进制时最高位表示特殊权限（参见本书 1.3.2.4 节）。

常用选项有：

<code>-R, --recursive</code>	递归处理所有的文件及子目录
<code>-v, --verbose</code>	为处理的所有文件显示诊断信息

例如：

(1) 为 `subdir` 目录及其子目录和文件的所属组添加“写”权限，命令如下。

```
jianglinmei@ubuntu:~$ sudo chmod -R g+w ./subdir/
```

(2) 将 `one.txt` 的权限设置为只有所有者可读，命令如下。

```
jianglinmei@ubuntu:~$ ls -l one.txt
-rw-rw-r-- 2 alice alice 3218 2012-09-21 09:30 one.txt
jianglinmei@ubuntu:~$ sudo chmod 600 one.txt
jianglinmei@ubuntu:~$ ls -l one.txt
```

```
-rw----- 2 alice alice 3218 2012-09-21 09:30 one.txt
```

2.12 进程相关命令

2.12.1 进程和作业

Linux 是一个多用户多任务的分时操作系统，也就说，同一时间有分属于多个用户的多个进程在运行。作为 Linux 系统的使用者，特别是管理员，需要时常关注进程的运行情况以更加有效地使用或管理系统。

用户在 Shell 中执行一条外部命令的时候，系统会为其启动一个或多个进程。进程可以在前台运行也可以在后台运行。前台运行的进程与用户可交互，后台运行的进程则无法接受用户的输入，其结果一般也不显示在标准输出设备。

Shell 分前后台来控制的不是进程而是作业（Job）或者进程组（Process Group）。一般来说，一次提交给 Shell 的一整条命令对应一个作业。一个前合作业可以由多个进程组成，一个后合作业也可以由多个进程组成，Shell 可以运行一个前合作业和任意多个后台作业，这称为作业控制。作业与进程组的区别在于：如果作业中的某个进程又创建了子进程，该子进程不属于作业，如果作业运行结束而这个子进程还没终止，该子进程自动变为后台进程组。

要在后台执行一条命令（一个作业），可以在命令后加上“&”符号。将命令放到后台执行时，屏幕上会显示一个作业号和一个进程号，例如：

```
jianglinmei@ubuntu:~$ vi &
[1] 2910
```

示例中，方括号中的 1 是作业号，2910 是进程号。使用 jobs 命令可以查看终端启动了哪些作业，例如：

```
jianglinmei@ubuntu:~$ jobs -l
[1]+ 2910 停止 (tty 输出)      vi
```

示例中显示有一条作业号为 1 的作业处于停止状态，jobs 的“-l”选项表示以长格式显示结果。使用命令“fg <JOBID>”可将后台运行的作业切换到前台运行。例如：

```
jianglinmei@ubuntu:~$ fg 1
```

执行该命令后，vi 编辑器恢复到正常运行的界面状态。使用“Ctrl + Z”组合键可暂停前台运行的作业。例如，在 vi 运行时，按下“Ctrl + Z”组合键，屏幕下方会显示：

```
[1]+ 已停止          vi
```

同时会退出 vi 回到 Shell 命令行。使用命令“bg <JOBID>”可将暂停的前台运行的作业切换到后台运行。例如：

```
jianglinmei@ubuntu:~$ bg 1
[1]+ vi &
```

```
[1]+ 已停止          vi
```

2.12.2 查看进程——ps

Linux 用户，特别是系统管理员，需要经常关注当前系统中运行的进程的状况。本书第 1 章介绍了使用图形界面“系统监视器”查看进程信息的方法，本节来关注如何使用 Shell 命令 ps 来查看系统中当前正在运行的进程的信息。

- 一般格式

```
ps [选项]
```

- 说明

使用 ps 命令可以查看进程的许多信息，比如：哪些进程正在运行，哪些进程被挂起，进程的 ID，进程运行了多久，进程使用的资源的情况等。ps 命令格式简单，但选项众多，而且选项前有无“-”是有区别的，无“-”的选项称 BSD 格式项，在此仅介绍该命令的常用方法。

- 常用选项

-e	显示所有进程
-f	全格式
-h	不显示标题
-l	长格式
a	显示终端上的所有进程，包括其他用户的进程
r	只显示正在运行的进程
x	显示没有控制终端的进程
u	面向用户格式

- 常用输出格式及其各列的含义

(1) 选项-lf 格式。

使用-lf 选项输出的各列为：

F S UID	PID PPID C PRI NI ADDR SZ WCHAN STIME TTY	TIME CMD
---------	---	----------

各列的具体含义如下。

F	进程旗标。0 表示一般进程，1 表示已创建但未执行，4 表示有超级用户权限，显示结果是这些值之和
S	进程状态（参见表 2-6）
PID	进程 ID
PPID	父进程 ID
C	占用的 CPU 百分比，仅整数部分
PRI	优先级
NI	Nice 值
ADDR	内核进程的内存地址。一般进程只显示“-”
SZ	占用内存大小
WCHAN	进程是否正在运行，“-”表示正在运行
STIME	启动进程的时间
TTY	终端
TIME	进程消耗 CPU 的时间，格式为：[DD-]hh:mm:ss
CMD	命令的名称和参数

(2) 选项 u 格式。

使用 u 选项输出的各列为：

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
------	-----	------	------	-----	-----	-----	------	-------	------	---------

各列的具体含义如下。

USER	进程所有者
PID	进程 ID
%CPU	占用的 CPU 百分比，“##.#”格式
%MEM	占用的内存百分比
VSZ (virtual memory size)	占用的虚拟内存大小，单位为 KB
RSS (resident set size)	占用的非交换的物理内存大小，单位为 KB
TTY	终端的次要装置号码
STAT	进程状态（参见表 2-6）
START	启动进程的时间
TIME	进程消耗 CPU 的时间，格式为：[DD-]hh:mm:ss
COMMAND	命令的名称和参数

表 2-6

进程状态标识及含义

标 识	含 义	标 识	含 义
D	不可中断的睡眠（通常为 IO 引起）	<	高优先级
R	运行或就绪	N	低优先级
S	可中断的睡眠（等待事件完成）	L	有分页被锁进内存
T	停止（因作业控制或调试跟踪）	s	会话的领导者进程（控制进程）
W	页面交换（2.6.xx 内核起无效）	l	多线程进程
Z	僵尸进程（因结束时父进程未对其清理引起）	+	在前台进程组

● 示例

(1) 显示本用户本次登录启动的进程信息。

```
jianglinmei@ubuntu:~$ ps -lf
F S  UID      PID  PPID C PRI  NI ADDR SZ WCHAN  STIME TTY          TIME CMD
0 S molin     5516  5515  0  75   0 - 13835 wait    23:15 pts/9    00:00:00 -bash
0 R molin     5802  5516  0  76   0 - 1367 -       23:44 pts/9    00:00:00 ps -lf
```

(2) 以长格式显示所有进程信息。

```
jianglinmei@ubuntu:~$ ps -elf
F S  UID      PID  PPID C PRI  NI ADDR SZ WCHAN  STIME TTY          TIME CMD
4 S root      1     0  0  80   0 -  880 poll_s 23:51 ?          00:00:00 /sbin
.....此处省略若干进程信息
5 R 1000     1582  1451  0  80   0 - 2526 ?      23:56 ?      00:00:00 sshd:
0 S 1000     1583  1582  7  80   0 - 2833 wait    23:56 pts/1    00:00:00 -bash
0 R 1000     1681  1583  0  80   0 - 1588 -      23:56 pts/1    00:00:00 ps -e
```

(3) 以 BSD 面向用户的格式有终端的进程的信息。

```
jianglinmei@ubuntu:~$ ps au
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
```

```

root      797  0.0  0.0  4612  864  tty4    Ss+  23:51  0:00 /sbin/getty -8 38400 tty4
root      806  0.0  0.0  4612  852  tty5    Ss+  23:51  0:00 /sbin/getty -8 38400 tty5
root      812  0.0  0.0  4612  856  tty2    Ss+  23:51  0:00 /sbin/getty -8 38400 tty2
root      813  0.0  0.0  4612  868  tty3    Ss+  23:51  0:00 /sbin/getty -8 38400 tty3
root      818  0.0  0.0  4612  868  tty6    Ss+  23:51  0:00 /sbin/getty -8 38400 tty6
root      1018 0.0  0.0  4612  864  tty1    Ss+  23:51  0:00 /sbin/getty -8 38400 tty1
root      1040 0.5  1.6  48492 15088  tty7    Ss+  23:51  0:02 /usr/bin/X :0 -auth
/var/run/lightdm/root/:0 -nolisten tcp vt7 -novtswitch
1000     1583 0.3  0.6  11332  6236  pts/1    Ss   23:56  0:00 -bash
1000     1682 0.0  0.1   6352  1168  pts/1    R+   23:58  0:00 ps au

```

2.12.3 结束进程——kill

kill 命令用于向进程发送信号，默认发送的信号是“TERM”，其作用是令进程终止。用“kill -l”选项可查看 kill 命令可以发送的信号有哪些。

kill 命令用于结束进程的格式是：kill <进程 ID>，因此 kill 命令通常要配合 ps 命令使用，以获知进程的 ID，然后向其发送信号。例如：

```

jianglinmei@ubuntu:~$ vi &
[1] 1953
jianglinmei@ubuntu:~$ ps -lf
F S UID      PID  PPID  C PRI  NI ADDR SZ WCHAN  STIME TTY          TIME CMD
0 S 1000      1583  1582  0 80   0 - 2833 wait    Sep20 pts/1    00:00:00 -bash
0 T 1000      1953  1583  0 80   0 - 1686 signal   00:15 pts/1    00:00:00 vi
0 R 1000      1957  1583  0 80   0 - 1588 -       00:15 pts/1    00:00:00 ps -lf
jianglinmei@ubuntu:~$ kill 1953
jianglinmei@ubuntu:~$ fg vi
vi
Vim: Caught deadly signal TERM
Vim: preserving files...
Vim: Finished.
已终止
jianglinmei@ubuntu:~$ ps -lf
F S UID      PID  PPID  C PRI  NI ADDR SZ WCHAN  STIME TTY          TIME CMD
0 S 1000      1583  1582  0 80   0 - 2854 wait    Sep20 pts/1    00:00:00 -bash
0 R 1000      1963  1583  0 80   0 - 1588 -       00:18 pts/1    00:00:00 ps -lf

```

示例中演示了 vi 进程从启动到被终止的过程，使用 ps 命令列出了这一过程中进程的变化。首先在后台启动 vi，然后向其发送 TERM 信号，当 vi 被切换到前台时捕获了 TERM 信号，导致进程终止。

2.13 网络相关命令

- ping

ping 是最常用的网络命令之一，用于诊断 tcp/ip 协议是否安装好，目标主机是否可达等。其一般格式为：

```
ping [选项]... 目标主机或 IP...
```

常用选项有：

-c 指定发送数据包的数量，默认为一直发送直到用户按下 Ctrl+C 组合键

例如：

```
jianglinmei@ubuntu:~$ ping -c4 www.hqu.edu.cn
PING www.hqu.edu.cn (210.34.240.107) 56(84) bytes of data.
64 bytes from www.hqu.edu.cn (210.34.240.107): icmp_req=1 ttl=60 time=2.61 ms
64 bytes from www.hqu.edu.cn (210.34.240.107): icmp_req=2 ttl=60 time=1.81 ms
64 bytes from www.hqu.edu.cn (210.34.240.107): icmp_req=3 ttl=60 time=1.94 ms
64 bytes from www.hqu.edu.cn (210.34.240.107): icmp_req=4 ttl=60 time=1.81 ms

--- www.hqu.edu.cn ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 1.814/2.048/2.612/0.330 ms
```

● ifconfig

ifconfig 命令可以用来查看 IP 地址，也可用于设置 IP 地址，其选项众多，功能强大。应注意，ifconfig 设置的 IP 只本次生效，在重启电脑后会丢失。

ifconfig 命令的常见用法如下。

(1) 查看当前活动网口的 IP。

```
jianglinmei@ubuntu:~$ ifconfig
eth1      Link encap:以太网 硬件地址 08:00:27:1a:1b:c4
          inet 地址:10.8.30.252 广播:10.8.30.255 掩码:255.255.255.0
          inet6 地址: fe80::a00:27ff:fea1:bc4/64 Scope:Link
            UP BROADCAST RUNNING MULTICAST MTU:1500 跳点数:1
            接收数据包:58186 错误:0 丢弃:0 过载:0 帧数:0
            发送数据包:34389 错误:0 丢弃:0 过载:0 载波:0
            碰撞:0 发送队列长度:1000
            接收字节:10643221 (10.6 MB)  发送字节:4573083 (4.5 MB)

lo       Link encap:本地环回
          inet 地址:127.0.0.1 掩码:255.0.0.0
          inet6 地址: ::1/128 Scope:Host
            UP LOOPBACK RUNNING MTU:16436 跳点数:1
            接收数据包:16 错误:0 丢弃:0 过载:0 帧数:0
            发送数据包:16 错误:0 丢弃:0 过载:0 载波:0
            碰撞:0 发送队列长度:0
            接收字节:1152 (1.1 KB)  发送字节:1152 (1.1 KB)
```

(2) 查看所有网口的 IP。

```
jianglinmei@ubuntu:~$ ifconfig -a
.....此处省略输出
```

(3) 设置 eth1 网口的 IP 地址和掩码。

```
jianglinmei@ubuntu:~$ sudo ifconfig eth1 192.168.0.100 netmask 255.255.255.0
```

(4) 禁用 eth1 网口。

```
jianglinmei@ubuntu:~$ sudo ifconfig eth1 down
```

(5) 激活 eth1 网口。

```
jianglinmei@ubuntu:~$ sudo ifconfig eth1 up
```

● netstat

netstat 可用于查看网络的状态，其选项众多，本节仅列举其若干常见用法如下。

(1) 查看本机路由信息。

```
jianglinmei@ubuntu:~$ netstat -r
```

内核 IP 路由表

Destination	Gateway	Genmask	Flags	MSS	Window	irtt	Iface
10.8.30.0	*	255.255.255.0	U	0	0	0	eth1
default	10.8.30.1	0.0.0.0	UG	0	0	0	eth1

(2) 查看本机 22 号 TCP 端口状态。

```
jianglinmei@ubuntu:~$ netstat -t -an | grep :22
tcp      0      0 0.0.0.0:22          0.0.0.0:*          LISTEN
tcp      0      116 10.8.30.252:22    10.8.18.212:14513  ESTABLISHED
tcp      0      0 10.8.30.252:22    10.8.18.212:15459  ESTABLISHED
tcp6     0      0 ::*:22            ::*:*              LISTEN
```

示例中，“-t” 选项表示仅查看 TCP 端口，亦可使用 “-u” 选项指定查看 UDP 端口。

2.14 小结

本章较为详细地介绍了 Linux Shell 的一些特殊字符和 50 余个最常用的命令的最常用方法。掌握这些基本命令的使用是熟悉 Linux 操作环境和编程环境的基础。在此基础上，结合 Linux 系统的手册页，读者应当能够逐渐掌握更多 Linux 命令的使用，继而熟练掌握 Linux 操作环境。

2.15 习题

- (1) 请简述有哪些常见的获取 Shell 命令帮助的方法，具体如何操作？
- (2) 什么是路径、绝对路径、相对路径和当前工作目录？
- (3) Linux Shell 默认会打开哪些标准设备文件？如何使用输入、输出重定向？如何使用管道？
- (4) 如何使用 ls 命令查看文件的硬链接数？
- (5) 如何在文件系统中查找名称中含有“alice”的文件，并将它的信息详细显示出来？
- (6) locate 和 find 命令有何区别？
- (7) vi 中，将光标定位到屏幕中间一行的命令是什么？将光标移动到本行第一个非空白字符的命令是什么？
- (8) 列出当前目录下所有符号连接文件的命令。
- (9) 如何对文本文件的每行按第二个字段的数值大小排序？

- (10) diff 命令的显示结果中“>”符号表示什么意思?
- (11) 如何统计一个文件的字数?
- (12) 如何创建符号连接文件?
- (13) 如何将一个目录及其子目录全部打包并压缩成一个“.gz”文件,然后对其解压和解包?
- (14) 如何挂载一个“.iso”光盘映像文件?
- (15) 如何查看本系统的 CPU 类型?
- (16) 创建 tom 用户并为其创建家目录同时指定其 Shell 为 bash 的命令怎么写?
- (17) 如何将用户加入到一个组?
- (18) 如何为文件的所属组添加可执行权限?
- (19) Linux 的进程有哪些状态,如何查看进程的状态?
- (20) 如何查看本机的默认网关是什么?