

一个地方复制到另一个地方等。另外，用户也可以启动一个终端模拟程序xterm，它为用户提供一个到操作系统的基本命令行界面。下面一节有关于它的详细描述。

10.2.3 shell

尽管Linux系统具有图形用户界面，然而大多数程序员和高级用户都更愿意使用一个命令行界面，称作shell。通常这些用户在图形用户界面中启动一个或更多的shell窗口，然后就在这些shell窗口中工作。shell命令行界面使用起来更快速，功能更强大，扩展性更好，并且让用户不会遭受由于必须一直使用鼠标而引起的肢体重复性劳损（RSI）。接下来我们简要介绍一下bash shell（bash）。它是基于UNIX最原始的shell（Bourne shell）的，而且实际上它的名字也是Bourne Again shell的首字母缩写。经常使用的还有很多其他的shell（ksh，csh等），但是bash是大多数Linux系统的默认shell。

当shell被启动时，它初始化自己，然后在屏幕上输出一个提示符（prompt），通常是一个百分号或者美元符号，并等待用户输入命令行。

等用户输入一个命令行后，shell提取其中的第一个字，假定这个字是将要运行程序的程序名，搜索这个程序，如果找到了这个程序就运行它。然后，shell会将自己挂起直到该程序运行完毕，之后再尝试读入下一条命令。重要的是，shell也只是一个普通用户程序。它仅仅需要从键盘读取数据、向显示器输出数据和运行其他程序的能力。

命令中还可以包含参数，它们作为字符串传给所调用的程序。比如，下面的命令行

```
cp src dest
```

调用cp程序并包含两个参数，src和dest。这个程序将第一个参数解释为一个现存的文件名，然后创建该文件的一个副本，其名称为dest。

并不是所有的参数都是文件名。在命令行

```
head -20 file
```

中，第一个参数-20通知head程序输出file中的前20行，而不是默认的10行。负责控制一个命令的操作或者指定一个可选数值的参数称为标志（flag），习惯上由一个破折号标记。为了避免歧义，这个破折号是必要的，比如

```
head 20 file
```

是一个完全合法的命令，它告诉head程序输出文件名为20的文档的前10行，然后输出文件名为file的文档的前10行。大多数Linux命令接受多个标志和多个参数。

为了更容易地指定多个文件名，shell支持魔法字符，有时称为通配符。比如，一个星号可以匹配所有可能的字符串，因此

```
ls *.c
```

告诉ls列举出所有文件名以.c结束的文件。如果同时存在文件x.c，y.c，z.c，那么上述命令等价于下面的命令

```
ls x.c y.c z.c
```

另一个通配符是问号，负责匹配任意一个字符。一组在中括号中的字符可以表示其中的任意一个，因此

```
ls [ape]*
```

列举出所有以“a”，“p”或者“e”开头的文件。

像shell这样的程序不一定非要通过终端（键盘和显示器）进行输入输出。当它（或者任何其他程序）启动时，它自动获得了对标准输入（负责正常输入），标准输出（负责正常输出）和标准错误（负责输出错误信息）文件进行访问的能力。正常情况下，上述三个文件默认地都指向终端，因此标准的输出是从键盘输入的，而标准输出或者标准错误是输出到显示器的。许多Linux程序默认从标准输入进行输入并从标准输出进行输出。比如

```
sort
```

调用sort程序，其从终端读取数据（直到用户输入Ctrl-D表示文件结束），根据字母顺序将它们排序，然