

第 3 章 Linux 常用命令

shell 是用户与操作系统内核之间的接口，具有极其重要的地位。在 shell 提示下，Ubuntu Linux 许多功能要比在图形化用户界面（GUI）下完成得更快。可以在 GUI 中打开文件管理器、定位目录，然后从中创建、删除或修改文件，而在 shell 提示下，只需使用几个命令就可以完成上述工作。shell 的提示界面如图 3-1 所示。本章主要介绍管理文件和目录的命令、有关磁盘空间的命令、文件备份和压缩命令、有关关机和查看系统信息的命令、管理使用者和设立权限的命令等。

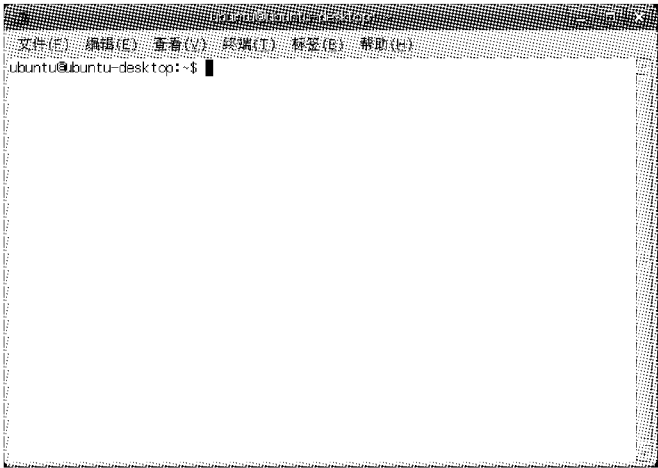


图 3-1 shell 提示界面

3.1 管理文件和目录的命令

几种经常使用的管理文件和目录的命令如表 3-1 所示。

表 3-1 管理文件和目录的命令

命 令	功 能	命 令	功 能
pwd	判定当前目录	ls	查看目录下的内容
cd	改变所在目录	cat	显示文件的内容

续表

命 令	功 能	命 令	功 能
grep	在文件中查找某字符	cp	复制文件
touch	创建文件		

3.1.1 pwd 命令

pwd 命令的英文解释为 print working directory(打印工作目录)。当输入 pwd 时，就是要求 Linux 系统显示当前位置。系统便会在 shell 提示窗口中输出当前目录名。当系统对信息请求做出响应时，这个响应被称作“标准输出(standard output)”，它可以输出到 shell 提示下，也可以被重导向导入其他程序或其他输出设备，如打印机。

pwd 命令用来判定当前目录：在目录中上下查看时会很容易迷失方向或者忘记当前目录全名。按照默认设置，Ubuntu Linux 中的 bash 提示只显示当前目录名，而不是整个路径。要判定当前目录在文件系统内的确切位置，请在 shell 提示下输入命令命令 pwd。就会看到类似以下的输出：

```
/home/user
```

以上例子表明，当前用户是在 user 的目录下，而这个目录又是在 /home 目录下。

3.1.2 cd 命令

cd 命令用来改变所在目录。只要知道所在位置即当前目录以及它与想转换到的位置之间的关系，就可以使用 cd 命令改变所在目录。只使用这个命令总是返回到当前的主目录；要转换到其他目录中，需要一个路径名。还可以使用绝对或相对的路径名。绝对路径从 /(指代根)开始，然后循序到所需的目录；相对路径从当前目录开始，当前目录可以是任何地方。下面的树形图显示了 cd 的运行方式。

```
/
/test1
/test1/test2
/test1/ test2/ test3
```

如果当前是在 test3 目录之下，想转换到 test1，就需要移到目录树的上一层。
若执行命令：

```
cd test1
```

系统会给出一个错误消息：没有那个文件或目录。说明该目录不存在。这是因为在 test3 之下并没有 test1 目录。

要向上移到 test1,输入:

```
cd /test1
```

这是一个绝对路径的例子。它告诉 Linux 从目录树的顶端(/)开始向下一直转换到 test1 为止。如果一个路径的第一个字符是 /,那么这个路径就是绝对路径,否则,就是相对路径。使用绝对路径允许转换到从 / 开始的目录,它要求知道完整的路径。而使用相对路径允许转换到相对于目前所在目录的目录中。如果要改换到当前目录下的子目录中,使用相对路径就会很方便。命令 `cd ..` 告诉系统向上移到当前所在目录的直接上级目录中去。要向上移两级目录,请输入 `cd ../../..` 命令。

关于绝对路径和相对路径的知识,可以用下面的例子来测试。在主目录下,输入相对路径:

```
cd ../../etc/X11
```

上面的例子中,当执行该命令后,应该是在目录 X11 中,其中包括许多与 X 窗口系统相关的配置文件和目录。

下面简单介绍最后一个 `cd` 命令的执行过程:

- 首先向上移动一级,转到登录目录的父目录中去(可能是 /home);
- 然后,再向上移动到该目录的父目录中去(根目录或 / 目录);
- 再次,向下移动到 `etc` 目录中;
- 最后,移到 X11 目录中。

相反地,使用一个绝对路径会更快地到达 `/etc/X11` 目录中。例如: 输入命令

```
cd /etc/X11
```

绝对路径以根目录(/)开始,然后向下移到指定的目录中。

必须注意,在标明要访问的目录或文件的相对路径之前,一定要确保知道自己的当前工作目录。但是,如果标明的是到另一个目录或文件的绝对路径,就不必担心在文件系统中的位置了。如果不能肯定所在的当前目录,输入 `pwd` 命令,当前的工作目录就会在屏幕上显示出来,可以用它来作为使用相对路径名转换目录的向导。

3.1.3 ls 命令

`ls` 命令用来查看目录的内容,英文含义是 list,是管理文件时最常用的一个命令。因为某些文件是隐藏文件(又称“点文件”),所以 `ls` 命令本身不会显示目录中的所有文件。但是 `ls` 命令有许多可用的选项,在 `ls` 命令后指定附加的选项就能看到它们。表 3-2 列举了几种重要且常用的选项。

表 3-2 ls 命令各选项的含义

选项	含 义
-a	列举目录中的全部文件,包括隐藏文件
-l	列举目录内容的细节,包括权限(模式)、所有者、组群、大小、创建日期、文件是否是到系统其他地方的链接,以及链接的指向
-f	文件类型(File type)。在每一个列举项目之后添加一个符号。这些符号包括: / 表明是一个目录;@ 表明是到其他文件的符号链接;* 表明是一个可执行文件
-r	逆向(reverse)。从后向前地列举目录中的内容
-R	递归(recursive)。该选项递归地列举所有目录(在当前目录之下)的内容
-s	大小(size)。按文件大小排序

下面通过几个例子来分析这些选项的作用。

输入命令：

12-12

系统将返回以点起首的文件,如图 3-2 所示。

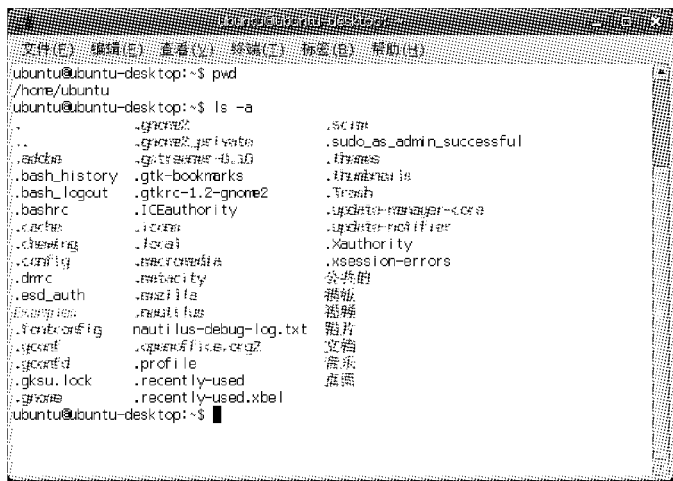


图 3-2 带有-a 选项的 ls 命令

隐藏文件多数是配置文件。它们给程序、窗口管理器、shell 等设置首选项。它们被隐藏的目的在于防止用户对其进行无意识的篡改。当用户在目录中搜寻某项事物时，一般不是在寻找这些配置文件，因而在 shell 下查看目录内容时把它们隐藏起来可以避免屏幕的拥挤。

使用 `ls -a` 命令来查看所有的文件会显示大量的细节,但是通过添加更多的选项,可以看到更多的细节。如果想查看一个文件或目录的大小、创建时间等等,在 `ls -a` 命令后面添加 `long`(长)选项(`-l`)就可以了。这个命令显示了文件创建的日期、大小、所有者、权限等等。使用 `ls` 命令来查看某个目录内容时,不必位于该目录中。譬如,要在某个用户

的主目录中查看 /usr 目录中的内容,输入:

```
ls -al /usr
```

系统返回信息如图 3-3 所示。

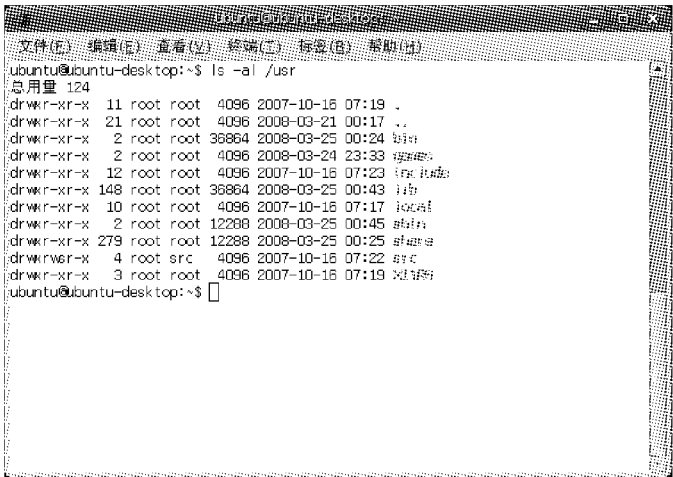


图 3-3 /usr 目录使用 ls -al 命令后的输出示例

3.1.4 cat 命令

cat 命令用来合并文件,它是 concatenate(连锁)的简写。cat 命令还可以在屏幕上显示整个文件的内容(例如,输入 cat filename.txt)。如果文件较长,它会在屏幕上快速地滚过。要防止这一效果的发生,可使用 cat filename.txt | ls 命令。同时使用管道(管道命令在后面的章节中会讲解)(|)和 ls 命令会一页一页地显示文件。然后,使用向上或向下的箭头键在页码中向前或向后移动。

3.1.4.1 使用重导向

重导向就是使 shell 改变它所认定的标准输出,或者改变标准输出的目标。要重导向标准输出,使用 > 符号。把 > 符号放在 cat 命令之后(或在任何写入标准输出的工具程序和应用程序之后),就会把输出重导向到跟在符号之后的文件中。

例如,使用 cat 自身会在屏幕上输出用户所输入的内容,如同是在重复刚输入的行。图 3-4 显示了 cat 会重复输入的每一行。

要把 cat 的输出重导向到一个文件中,在 shell 提示下输入以下命令:

```
cat > snow.txt.
```

按 Enter 键转到一个空行,系统返回信息如图 3-5 所示。

然后使用 Ctrl+D 键来退出 cat。

注意:此处并没有重复显示。这是因为 cat 的标准输出已经被重导向了。重导向的

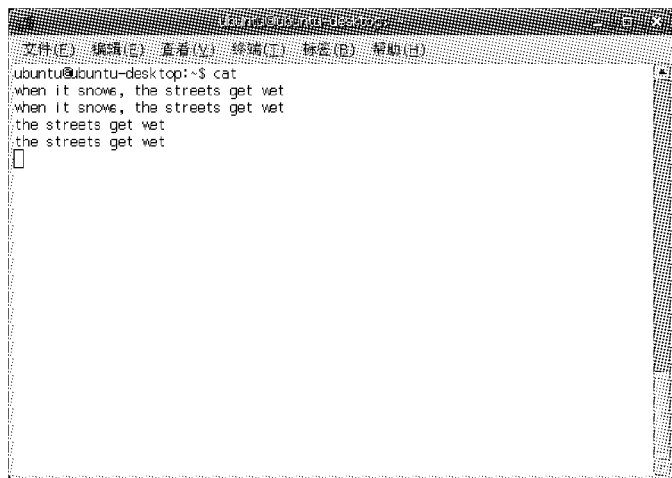


图 3-4 cat 命令

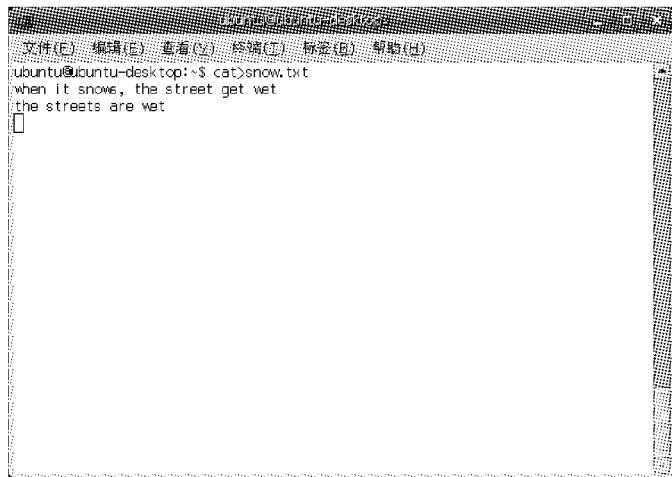


图 3-5 把输出重导向到一个文件

地方是刚刚制作的一个叫做 snow.txt 的新文件。

如前所示,可以使用 cat 来读取文件。在 shell 提示下,输入命令:

```
cat snow.txt
```

系统返回信息如图 3-6 所示。

再把输出重导向到另一个文件,暂且把它叫做 conclusion.txt。在这个例子中,输入命令 cat> conclusion,然后按 Enter 键,系统返回信息如图 3-7 所示。

现在,在一个空行上,使用 Ctrl+D 键来退出 cat。

下一步,使用 cat 把 conclusion.txt 和 snow.txt 连接起来,然后再把两个文件的输出重导向到一个叫做 weather.txt 的新文件。输入下面的命令:

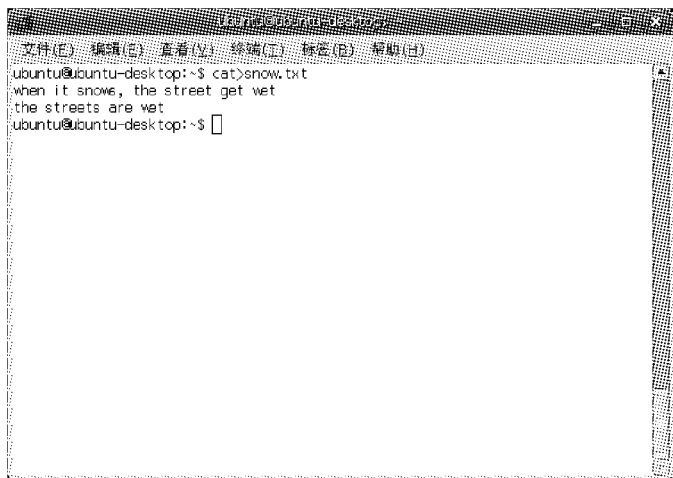


图 3-6 使用 cat 命令读取文件

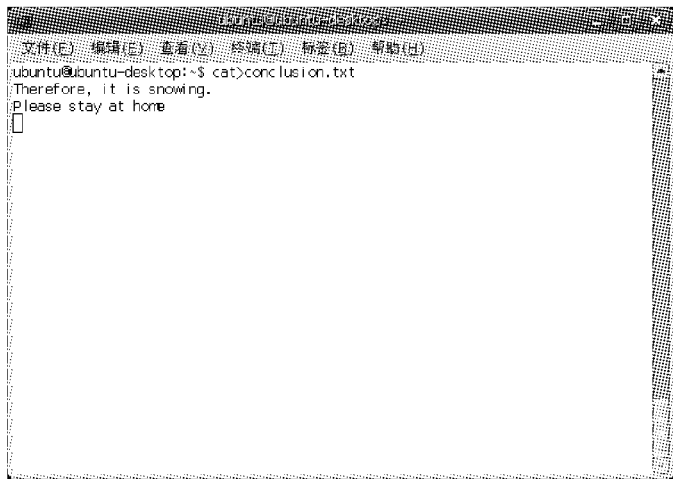


图 3-7 使用 cat 命令输出重导向

```
cat snow.txt conclusion.txt>weather.txt
```

使用命令 `cat weather.txt` 来检查一下。

系统返回信息如图 3-8 所示。

可以看到 `cat` 已经把 `conclusion.txt` 添加到 `snow.txt` 的结尾处。

还可以使用输出重导向来在一个现存文件之后添加新信息。这与使用 `>` 符号相似,使用 `shell` 把信息发送到标准输出之外的某个地方。然而,使用 `>>` 给文件添加信息,而不是完全替换文件内容。可以用通过下面的实例来加深理解:刚才已创建了两个文件(`snow.txt` 和 `conclusion.txt`),然后使用后补输出的符号把它们连接起来。把 `conclusion.txt` 的内容添加到 `snow.txt` 文件中去,因此输入命令:



图 3-8 连接文件和重导向输出

```
cat conclusion.txt >> snow.txt
```

现在,使用 `cat snow.txt` 的命令来检查一下。最终的输出在文件结尾处显示了 `onclusion.txt` 的内容,如图 3-9 所示。

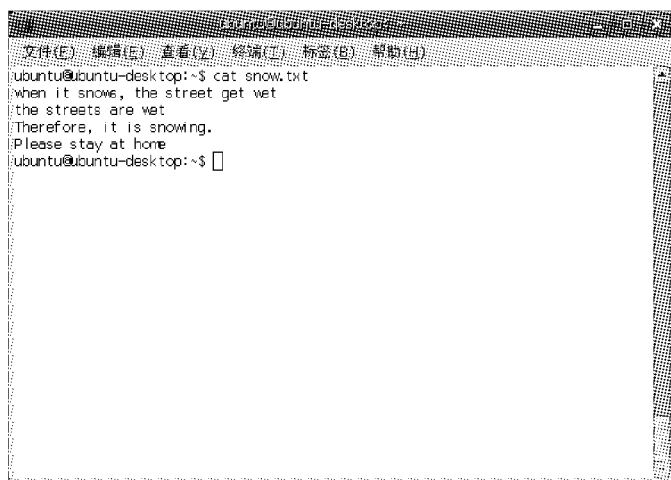


图 3-9 使用输出重导向来在一个现存文件之后添加新信息

输入的命令把文件 `conclusion.txt` 的输出后补到文件 `snow.txt` 中去。通过后补输出,利用了现存文件而不必创建新文件,从而节省了时间(以及一些磁盘群集)。现在比较一下文件 `snow.txt` 和文件 `weather.txt` 的结果,就会看到它们一模一样。要做比较,输入命令:

```
cat snow.txt; cat weather.txt
```

这两个文件的内容都会被显示。首先是 snow.txt 文件,然后是 weather.txt 文件,如图 3-10 所示。

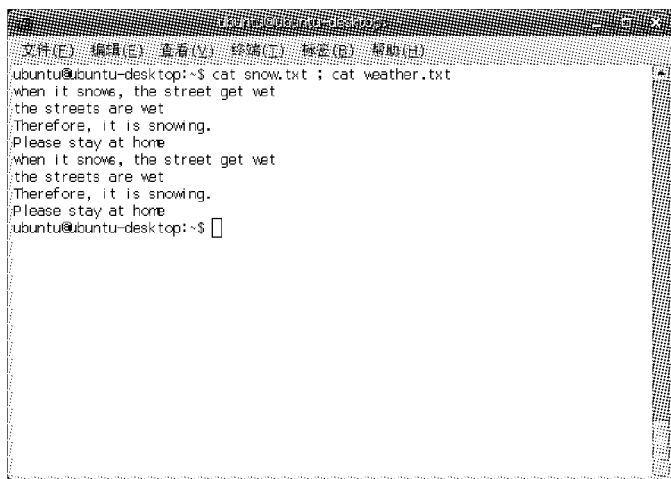


图 3-10 文件串联

3.1.4.2 重导向标准输入

不但可以重导向标准输出,还可以重导向标准输入。

使用重导向标准输入符号<是在告诉 shell 使用某个文件作为命令的输入。

使用一个已创建的文件来演示这一概念。输入:

```
cat <snow.txt
```

因为用来分隔 cat 命令和文件的是小于号(<),sneakers.txt 的输出就被 cat 命令读取,如图 3-11 所示。

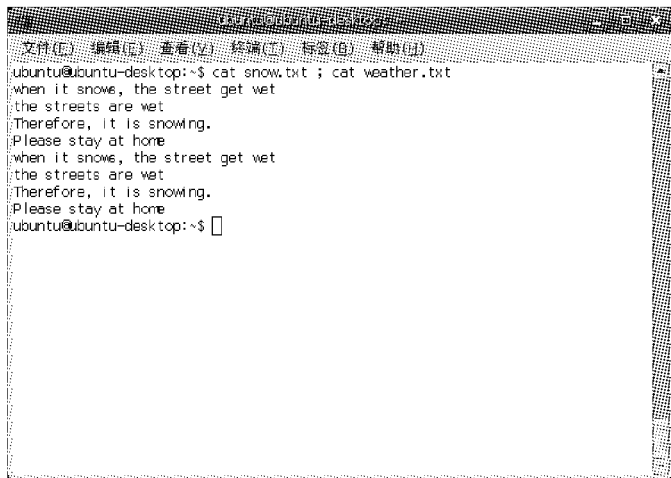


图 3-11 重导向标准输入

3.1.5 grep 命令

grep 命令最大的功能是在一堆文件中查找一个特定的字符串。例如找 money 这串字符(当然也可以查找中文字)。在写程序时这个命令变得非常有用,可以用它来查找某一函数,或者找某个特定的人名,或者相关的词组。

在文件 test.txt 中查找 money 这串字,可以输入命令:

```
grep money test.txt
```

就会看到文件中带有 money 的每一行。

除非另有说明,grep 查找是区分大小写的。这意味着查找 Money 与查找 money 不同。在 grep 的选项中有 -i 这一项,它会允许在查找文件时不区分大小写。

选项 -H 不仅输出带有 money 的每一行,而且还将这些匹配项的文件名一同输出。

3.1.6 touch 命令

touch 命令用来创建新文件,当然也可以通过应用程序(如文本编辑器)来创建。这两种方法都会创建一个空白的文件,可以在其中添加文本或数据。要使用 touch 命令来创建文件,在 shell 提示下输入命令:

```
touch <文件名>
```

把<文件名>替换成要创建的文件名。假如查看一下目录内容,就会看到该文件的大小为零,这是因为它是一个空文件。例如,在 shell 提示下输入命令 ls -l newfile 会返回以下输出:

```
-rw-r--r-- 1 ubuntu ubuntu 0 2008-03-25 22:08 newfile
```

3.1.7 cp 命令

cp 命令用来拷贝文件,要复制文件,输入命令:

```
cp <源><目标>
```

把<源>替换成想复制的文件,把<目标>替换成想保存复制文件的目录名。

因此,要把文件 test.txt 复制到/home/user 目录下输入命令:

```
cp test.txt /home/user
```

cp 命令的常用选项如表 3-3 所示。