

## 第5章 使用Linux操作系统

---

如果你需要下列问题的一个快速解决方案	请查阅节号
设置日期	5.2.1
设置时间	5.2.2
列出目录的内容清单	5.2.3
切换目录路径	5.2.4
确定自己所在的位置	5.2.5
解读文件和目录列表清单	5.2.6
改变文件和目录的存取权限	5.2.7
改变文件和目录的所有者	5.2.8
给文件和目录改名	5.2.9
建立目录	5.2.10
建立链接	5.2.11
删除目录	5.2.12
拷贝文件和目录	5.2.13
移动文件和目录	5.2.14
删除文件	5.2.15
识别二进制文件	5.2.16
使用vi文本编辑器程序	5.2.17
使用pico文本编辑器程序	5.2.18
查看文本文件，不使用文本编辑器程序	5.2.19
把命令结果输出到文本文件中	5.2.20
建立假名 ( alias )	5.2.21
使用find命令确定文件存放位置	5.2.22
通过搜索locate数据库来查找文件	5.2.23
在path语句说明的路径中查找文件	5.2.24
在path语句中添加新路径	5.2.25
在文件内容中查找文本	5.2.26
寻求帮助	5.2.27
在后台运行命令程序	5.2.28
确定当前运行的命令	5.2.29
把命令程序调到后台去	5.2.30
把后台命令调回前台来	5.2.31
取消正在执行的命令	5.2.32
清除屏幕	5.2.33
一次运行多个命令	5.2.34
退出登录	5.2.35
重新引导启动机器	5.2.36
系统关机	5.2.37

---

### 5.1 概述

Linux的程序员们已经做了大量的工作，使现在的Linux操作系统比它最初的版本更加面

向图形用户界面(GUI)。这个事实应该说是解放了那些习惯于使用鼠标类点击输入设备和那些喜欢在其机器上有鲜艳的色彩和背景的人们。但是,你还是需要命令行的界面中进行许多的工作。事实上,是Linux操作系统的强大功能照亮了命令行操作方式,使用户几乎可以对其系统中发生的任何事件都有一定的控制。这一点对那些专注于提供网络或者因特网服务功能的服务器机器来说,尤其正确。因为 GUI只会降低这些机器的速度,所以学习掌握在这类机器上使用命令行工具的方法是非常必要的。但是请记住,你不光可以选择这两者之一进行操作,还可以在两者之间随时切换。

### 5.1.1 Linux操作系统与DOS和NT的比较

许多人认为Linux操作系统和MS-DOS差不多,因为两种操作系统都提供了基于命令行的操作方式,但实际上二者有一定的差别。Linux操作系统是一个多任务网络操作系统,要求用户按照各自的身份登录上机,而不是把他们一视同仁;分配给某个用户的安全级别是在建立他或她的帐户时就设置好了的。

而另一方面的DOS只不过是一个单任务系统,也就是说它一次只能完成一件事情。同时,它也不是为网络操作开发的,因此无法管理多个用户的帐户;它把任何坐在机器前面的人都当作是同一个人。从微软公司的操作系统产品线方面来看,Linux操作系统比较类似于Windows NT——而UNIX比这两者都要早得多了。

从较为初级的角度看,改用Linux操作系统后需要习惯的差异之一就是在涉及到目录的时候,Linux使用正斜杠(/)而不是DOS或者Windows中的反斜杠(\)。举例来说,Linux操作系统中的一个合法目录的写法是/usr/bin。另外,Linux操作系统在理解你输入的字母时非常挑剔,也就是说输入一个大写字母和输入一个小写字母其含义将是完全不同的两件事。因此输入字母“Vi”与输入字母“vi”是不一样的。这与大多数人们熟悉的Windows和DOS形成了强烈的对比,因为它们对大小写都是不敏感的。

### 5.1.2 文件存取权限和所有权限

除了那些熟悉Windows NT存取控制清单(Access Control List)的NT用户和那些设置过文件共享功能的Macintosh用户以外,普通Macintosh和Windows用户需要逐步适应的一个新的概念就是文件和目录都有所有者和相关的存取权限。每一个目录和文件都有一个所有者、一个分组和一系列存取权限。存取权限本身又分为对应着所有者、分组和全体用户的几个组成部分。

注意 Macintosh机器上文件共享功能的设置结构实际上是继承自UNIX操作系统的存取权限的概念,因此那些熟悉Mac文件安全性设置的人们比较容易理解Linux操作系统中的存取权限概念。

文件的存取权限是使用长格式化文件列表命令(ls -l)列文件清单时出现在最左边的第一数据段。这个数据段又可以被分解为10个标志位。

#### 1. 类型标志位

存取权限数据段的第一位,或者第一个字母表示的是这个

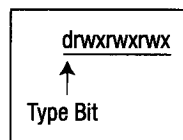


图5-1 类型位在存取权限数据段中的位置

项目的类型，如图 5-1 所示。许多不同的项目会用到这个标志位：

- -——表示该项目是一个普通文件
- d——表示该项目是一个目录
- l——表示该项目是一个符号链接（请参考 5.2.11 节中的内容）
- s——表示该项目是一个可执行文件，不论谁都可以像它的所有者那样调用它运行
- c——表示该项目是一个字符设备驱动程序
- b——表示该项目是一个块设备驱动程序

## 2. 存取权限三位组

存取权限的其余部分每三位一组，分为三组，如图 5-2 所示。

类型位后面的第 1 个三位组定义了其所有者的存取权限。第 2 个三位组定义了其所在分组的存取权限；第 3 个三位组则说明了系统上任何人的存取权限。从上面的介绍可以看出，认真设置这些标志位是非常重要的。

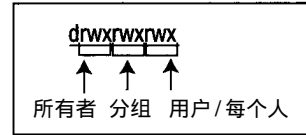


图5-2 三个存取权限三位组：  
所有者、分组和其他人（全部用户）

## 3. 解读三位组

解读各个三位组的方法是完全一样的。每一个三位组是由 3 个不同的数据位组成的：

- 读标志位（r）
- 写标志位（w）
- 执行标志位（x）

这些标志位永远都按照同一个顺序显示为 rwx 的形式。区别某个标志位是处于允许（on）还是禁止（off）的状态就要看该标志位是显示为一个字母还是一个连字符（-）。举例来说，在 rw-存取权限数据段表示读标志位在允许状态、写标志位在允许状态、但是执行标志位就是禁止状态；所以这个三位组就表示具有读和写的权限，但是没有可执行权限。

## 5.2 快速解决方案

### 5.2.1 设置日期

请使用 date 命令的 “date -d mm/dd/yyyy” 格式设置日期。比如，输入命令 “date -d 01/19/2010”。

窍门 date 命令还有许多其他的格式参数，查询详细资料请输入 “man date” 命令。

### 5.2.2 设置时间

请使用 date 命令的 “date -s hh:mm:ss” 格式设置时间。比如，输入命令 “date -s 14:02:00”。

窍门 time 命令还有许多其他的格式参数，查询详细资料请输入 “man date” 命令。

### 5.2.3 列出目录的内容清单

ls 命令列出某个目录中的内容。它在功能上类似于 DOS 的 dir 命令。图 5-3 给出了使用 ls 命

令列出的/root目录包含的内容。

```
[root@samurai /root]# ls -ls
total 16
  2 figure-5.1.gif    2 lsresults    1 nsmail
 10 figure-5.2.gif    1 mail
[root@samurai /root]#
```

图5-3 使用ls命令列出的root目录中的文件

这个命令有许多的命令行参数。其中最常用的一些列出在表 5-1中。最常使用的组合之一是ls -la, 这个命令格式将列出某个目录中包括隐藏文件在内的全部内容。隐藏文件是那些在长格式清单列表中文件名以句号 (.) 开头的那些文件。

表5-1 ls命令常用的命令行参数

参 数	使用 方法
l	每一行只显示一个文件或者目录
a	列出全部的文件, 包括那些以句号 (.) 开头而平常隐藏起来的文件
c	列出文件上次修改使用的日期
l	以长格式列出目录中的内容, 其中包括下面的信息: 存取权限、文件的链接个数或者目录中文件的个数、所有者、分组、以字节计算的文件长度、建立日期和文件名

窍门 有时候一个目录中的文件太多了, 因此在一个屏幕上没有办法把它们都显示出来。比如/usr/bin目录就是这种情况, 这时需要给ls命令加上“| more”选项。按照“ls /usr/bin | more”的格式输入命令。用户将看到屏幕以每次一页的方式显示usr/bin目录中的内容。

## 5.2.4 切换目录路径

请使用cd命令切换目录路径。有两种使用cd命令的不同方法。一个方法是直接切换目录路径; 另外一个相对直接切换目录路径。

### 1. 使用cd命令直接切换目录路径

直接切换到某个目录地址是一个很简单的操作过程。使用cd /path的格式就可以。举例来说, 如果想从目录/var/spool/中直接切换到目录/usr/bin中去, 直接输入命令“# cd /usr/bin”即可。

窍门 输入“# cd ~”命令将使用户回到他们的用户目录中去。在缺省情况下, 用户tom有一个名称为/home/tom的用户目录。以tom的身份登录进入系统的时候, 输入“# cd ~”命令与输入“# cd /home/tom”命令的效果是完全一样的。而以根用户身份登录进入系统的时候, 输入同样的内容就会把你带到/root目录中去, 也就是根用户的用户目录。

### 2. 使用cd命令相对切换目录路径

根据当前所处的位置和想去的地方，你可以通过几种不同的方法相对地切换目录路径。如果只是想移动到目录结构的上一级去，可以输入“`cd ..`”命令。举例来说，这个命令可以把你从目录`/var/spool`带到目录`/var`中去。更进一步地说，输入“`cd ../..`”命令将把你从目录`/var/spool`带到根目录“`/`”中去。因为在根目录上没有更高一级的目录结构了，所以在目录`/var/spool`中输入“`cd ../../..`”命令还是把你从目录`/var/spool`带到根目录而已。“`..`”是一个指向自身的目录索引。

如果你想从一个目录切换到同一个父目录的另外一个目录中去，比如从`/etc/rc.d`到`/etc/skel`，你可以使用相对路径格式，输入命令“`cd ../skel`”。

### 5.2.5 确定自己所在的位置

`pwd`命令将显示用户所在位置的完整路径。

### 5.2.6 解读文件和目录清单列表

在任何一个目录中输入“`ls -l`”命令都会列出一个长格式的文件和目录清单。举例如下：

```
-r w - r - r - - 1 linda user          33191 Aug 15 17:35 text
```

从左到右，这一行数据的意义可以解读如下：

- 1) 这个项目是一个文件而不是目录，因为第一个字母不是 `d`。
- 2) 这个项目是一个文件原本而不是符号链接，因为第一个字母不是 `l`。
- 3) 这个文件的存取权限是“`-r w - r - r - -`”。
- 4) 这个文件指有一个版本，没有任何链接。
- 5) 这个文件的所有者是 `linda`。
- 6) 所有者愿意与之共享存取权限的分组是 `user` 组。
- 7) 这个文件的长度是 33191 个字节。
- 8) 这个文件最后一次的修改日期是 8 月 15 日的下午 5:35 分。
- 9) 这个文件的名称是 `text`。

### 5.2.7 改变文件和目录的存取权限

有两种方法可以用来改变某个项目的存取权限，两种方法都需要使用 `chmod` 命令的“`chmod perm file`”格式。在这个时刻最关键的是你必须了解存取权限是如何表示的。如果你对此还没有把握，请重新阅读本章前面关于这个问题的讨论。

使用通配符，你可以一次对多个项目进行修改。

#### 1. 使用字母修改

使用字母来修改一个文件的存取权限看起来是很直截了当的，但是当需要进行多重修改的时候，情况就会变得有一些复杂。这个方法需要使用加号（`+`）或者减号（`-`）从每一个存取权限三位组中一次性地对存取权限进行加减。三位组中的字母“`o`”代表所有者（`owner`），“`g`”代表分组（`group`），“`u`”代表用户（`user`）——也就是其他所有的人。

举例来说，名为“`text`”的文件在最初的时候其存取权限可能是完全开放的（即存取权限字符串是 `rwrxrwxrwx`）。在大多数情况下，这是一个很差的主意。因为它只是一个文本文件，

所以应该先输入“`chmod -x text`”命令把可执行权限去掉。选择这个存取权限字符串应该是“`rw-rw-rw-`”。也许你并不想让其他人——包括普通用户和同组成员等等——能够对这个文件的内容进行修改。那么输入“`chmod u-w g-w text`”命令把存取权限字符串修改为“`rw-r--r--`”。

## 2. 使用数字修改

使用数字来修改一个文件的存取权限看起来是比较复杂，但是只要了解了如何计算需要的值，就会发现其实这个操作还是很简单的。设置值是针对每一个三位组分别计算的。

请按照下面的方法为某个项目的存取权限计算数字值：

1) 先把存取权限分成三组。比如说，一个名称为“`files`”的目录的完全开放的存取权限是“`drwxrwxrwx`”。去掉字母`d`，把其余的部分分为3个三位组：`rwx`、`rwx`和`rwx`。

2) 每一个字母都有它各自的数值：

- 一个字母`r`是4
- 一个字母`w`是2
- 一个字母`x`是1

根据这个对应关系把每个三位组的值写出来。如果某一标志位上是字符“`-`”(连字号)，把它算做0。在上面的例子中，所有的三位组的值都是一样的“`421`”。

3) 把这些数字加在一起就得到每个三位组的存取权限数值。比如：“ $4 + 2 + 1 = 7$ ”。

4) 全部3个三位组的值都是一样的，因此这个文件的存取权限值就是“`777`”

如果想把某个项目的存取权限修改为另外的一个数值，先计算出新设置的数值。比如说，如果想把前面例子中的“`files`”目录的存取权限设置为“`664`”( `rw-r--r--` )，请输入“`chmod 644 files`”命令。

## 5.2.8 改变文件和目录的所有者

修改某个项目的所有者是一个很简单的操作。使用 `chown` 命令的格式“`chown owner item`”即可。使用通配符，你可以一次对多个项目进行修改。

## 5.2.9 给文件和目录改名

Linux操作系统可以接受的文件或者目录名的长度最多为 256个字符。除了字母和数字之外，可以用在文件名中又不引起问题的字符还有句号( `.` )、下划线( `_` )和连字号( `-` )。

### 5.2.10 建立目录

建立目录是一个很简单的操作。在 DOS 中，建立目录的命令是 `md`。Linux 操作系统中的对应命令是 `mkdir`。输入“`mkdir testbed`”命令就可以建立一个名为“`testbed`”的目录。你还可以使用 5.2.4 节介绍的直接或者相对目录路径进行操作。

### 5.2.11 建立链接

链接文件是在不制作另外一个拷贝的情况下建立文件之间联系的方法，这样它们就可以同时出现在两个不同的地方。有两种类型的链接：硬链接和软链接(符号链接)。两种链接都

是用ln命令建立的。

### 1. 硬链接

建立一个硬链接就是在节点（inode）级别建立一个指向某些文件的指针。节点是把文件系统的内容与它们的描述符对应起来的映射表的一部分。请使用“ln source link”的格式建立硬链接。因为硬链接在文件系统中指向完全一样的同一个位置，所以删除原始文件并不能使该链接失效。原始文件只有在全部相关硬链接都被删除之后才能被清除。

### 2. 符号链接

建立一个符号链接在被链接的位置生成一个该文件的拷贝，它的存取权限与原始文件的完全相同。请使用“ln -s source link”的格式建立符号链接。因为符号链接不指向同一个节点，因此删除了原始文件就会使全部的符号链接失效。

**窍门** 在发行版本的CD-ROM光盘上有一个非常有用的RPM包叫做“symlinks”。这个程序可以帮助用户跟踪查出所有的失效链接——即那些仍然保留在系统中但是指向的是那些已经不存在的程序的符号链接。

## 5.2.12 删除目录

有两种方法可以删除目录。一种是使用rmdir命令，它只能删除那些空目录。如果想使用rmdir命令来删除某个目录，这个目录中就不能包含有任何文件或目录。

一个更有效，但是也更潜在着危险的命令（特别是对根用户来说）就是“rm -r”。这个命令将删除目录和其中全部的内容，包括下一级的目录。举例来说，假设用户建立了一个/testbed目录，并且在其中也已经保存了一些测试脚本。这些脚本经过修修改改之后，用户把那些需要保留的保存到了其他的地方，这里就只剩下那些准备丢弃的了。现在只要输入“rm -r /testbed”命令就可以删除/testbed目录和它的全部内容。

**注意** 根用户在使用“rm -r”命令的时候有很大的危险性，因为他有删除任何东西的权限。有许多系统管理员在他或她的Linux或者UNIX经历中，都曾经因为使用这个命令的时候不小心而删除过全部的重要数据。这也是除非有绝对的必要，尽量不要以根用户身份进行工作的原因之一。

相关解决方案	请阅读节号
编写一个命令脚本程序	18.2.1

## 5.2.13 拷贝文件和目录

Linux操作系统使用cp命令拷贝文件。cp命令的格式是“cp original destination”；或者如果用户需要拷贝多个文件到同一个目的地，其命令格式为“cp original1 original2 original3 destination”。

**注意** 如果对成组的文件进行拷贝操作的话，那么文件的目的地必须是一个目录。

cp命令还接受表5-2中的通配符。比如上面的例子就可以简化为“cp original\* destination”。

表5-2 cp命令通常使用的通配符

通配符	含义
*	任意个数的未知字符
[]	任何包括在括号中字符范围内的字符
?	任何一个未知的字符

#### 5.2.14 移动文件和目录

对文件或者目录进行移动可以看作是先将文件进行拷贝，再删除原本这样一个操作的简化方式。这个任务是使用mv命令来完成的，其格式为“mv original destination”。mv命令与cp命令很相似，也可以接受表5-2中的通配符，也允许一次性把多个文件移动到某个目录中去。

**窍门** mv命令不能在不同的硬盘分区之间传递文件和目录。这时候只能先把文件和目录拷贝到目的地，然后再删除那些原始文件。

#### 5.2.15 删除文件

请使用rm命令的“rm file”格式删除文件。这个命令接受cp和mv命令使用的同样的通配符（如表5-2所示），还有“-r”参数（在5.2.12节讨论的）。使用这个命令的时候，前面对根用户的警告信息依然有效。

#### 5.2.16 识别二进制文件

Linux操作系统中的二进制文件通常在他们的文件名上都没有扩展名。输入命令“pico”将会运行一个文件名为pico的程序。在Windows中，输入“notepad”命令将会运行一个叫做notepad.exe的文件。

快速分辨某个文件是否是一个二进制文件（程序）的方法是在它的长格式文件清单列表的存取权限中是否有字母x。字母“x”表示该文件是可执行的，也就是一个程序。另外一个方法需要按照“file filename”的格式使用file命令。

#### 5.2.17 使用vi文本编辑器程序

vi编辑器是一个小巧但功能强大的程序。许多人在最初的时候认为它很难用，因为它的编辑命令都是用一两个字母来表示的；但是只要你习惯了使用vi，使用它进行编辑工作也会十分迅速。其原因在于这个编辑器只是一个非常短小的程序，因此它只需要使用很少的内存；同时其编辑命令很简短，又都可以通过键盘快速输入，所以基本上用不着在键盘和鼠标之间来回切换。

输入“vi file name”就可以编辑或者建立一个文件了。

vi编辑器有三种基本的模式：

- 插入模式——这个模式是通过在命令模式按下“i”键进入的；也可以通过表5-3中的命令经过一定的步骤之后进入到插入模式。
- 命令模式——这是用户输入单键命令的模式。在其他模式中按下ESC键就可以进入到命令模式。

表5-3 常见的能够进入编辑模式的vi命令

命 令	结 果
a	在当前光标位置后面处直接进入插入模式
A	在当前文本行之后进入插入模式
i	在当前光标位置前面处直接进入插入模式
I	在当前文本行之前进入插入模式
o	在光标下面插入一个新行并移动到其上，进入插入模式
O	在光标上面插入一个新行并移动到其上，进入插入模式

- 冒号模式——从命令模式输入一个冒号 (:) 就可以进入这个模式。在这里用户可以输入更加复杂的命令，通常是一些文件操作命令。在表 5-4中列出了一些常见的冒号模式命令。

除了执行编辑和插入操作的命令之外，还有表 5-5中列出的删除命令。

vi编辑器包括的内容比本书介绍的要丰富得多，因此市面上会有许多介绍 vi编辑器程序使用特色的书籍。如果你想多学习一些关于 vi编辑器的知识的话，可以访问下面给出的这些 Web 页：

- [www.vim.org](http://www.vim.org)
- [www.roxanne.org/faqs/vi\\_editor/index.html](http://www.roxanne.org/faqs/vi_editor/index.html)
- [www..genome.washington.edu/uwgc/unix/howtousevi.htm](http://www..genome.washington.edu/uwgc/unix/howtousevi.htm)
- [www.fisica.uson.mx/UnixDocs/unix4.vi.8-1-94.html](http://www.fisica.uson.mx/UnixDocs/unix4.vi.8-1-94.html)
- <http://physics.ucsc.edu/tutor/vi.html>

表5-4 常见冒号模式的vi命令

命 令	结 果
wq	保存并退出文件
w	保存文件但不退出
q	退出并且不保存文件
q!	退出并不保存所做的修改

表5-5 常见执行删除操作的vi命令。

命 令	结 果
dd	删除整个行
#dd	按照输入的数字删除几行，从当前行开始
D	删除光标位置后面全部内容
dw	删除光标所在位置的单词

### 5.2.18 使用pico文本编辑器程序

pico编辑器程序提供了一个易于使用的界面，它与 DOS中的edit命令很相似。图 5-4给出了 pico编辑器程序的画面。

输入 “ pico filename ” 就可以编辑或者建立一个文件。这个程序在屏幕的底部提供了一行

菜单，在其中列出了进行编辑时可以使用的操作选项。这些操作选项列在表 5-6中。

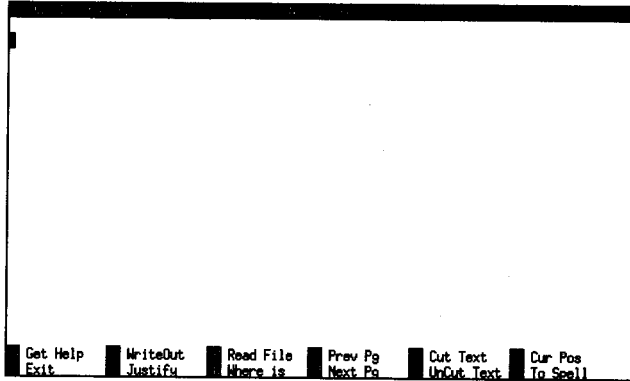


图5-4 pico编辑器

表5-6 pico编辑器程序常用的编辑操作

键 组 合	结 果
Ctrl + c	在文件中显示当前光标位置的数据
Ctrl + g	进入帮助系统
Ctrl + j	调整文本，使之均匀排列在一行的左右边界之间
Ctrl + k	剪切选定的文本块，并把它保存到缓冲区中去
Ctrl + o	保存到文件
Ctrl + r	打开文件
Ctrl + t	运行拼写检查
Ctrl + u	把缓冲区中的文本块粘贴到当前光标位置处
Ctrl + w	准备进行一次文本搜索
Ctrl + x	退出pico文本编辑器程序

Linux操作系统中有许多种的文本编辑器程序。下面给出一些工作在文本模式的编辑器程序供用户练习：

- joe
- jed
- jpico
- elvis
- vim
- ed
- cjoe

上面的清单中没有包括工作在 GUI模式的文本编辑器程序，它们也许更能够吸引用户。其中的一些是：

- xemacs
- cooedit
- asedit
- aswedit
- nedit
- xedit
- kedit

- gedit
- gnotepad+
- gxedit
- xjed

各种发行版本都包括有一些工作在文本模式或者 GUI模式下的文本编辑器程序。

### 5.2.19 查看文本文件，不使用文本编辑器程序

在需要对文本文件的内容进行编辑的情况下，文本编辑器程序即使不是必需的，也是有用的。但是，如果只是为了看看文本文件的内容，文本编辑器程序就有点大材小用了。

#### 1. more和less命令

两个最有用的和最相似的文件内容阅读程序就是 more命令和less命令。如果想使用这两个命令中的某个编辑文件，请输入“more filename”或者“less filename”命令。使用这两个程序之一阅读文本文件的好处是：如果文件的长度超过了一个屏幕，文本会停下来等待用户按下相应的键，然后再滚动显示一屏幕内容。

more命令和less命令之间只有很小的差异。more命令是两者中比较初级的那一个，但对大多数人来说也已经足够用了。通过more程序阅读文本文件时可以使用的命令列在表5-7中。使用这两个程序的好处之一是它们不需要把整个文件都读入内存后才开始显示其内容，因此它们对任何大小的文件运行得都一样快。

less命令在文件中移动方面提供了更多的选择。通过less程序阅读文本文件时可以使用的命令列在表5-8中。其中一些与more命令使用的完全一样，而另外一些则大不相同。

表5-7 使用more命令阅读文本文件时的常用命令

按 键	结 果
b	在文本中后退一行
回车键或者r键	在文本中前进一行
h	显示帮助信息
空格键	前进到文本的下一页
q	退出more阅读程序

表5-8 使用less命令阅读文本文件时的常用命令

按 键	结 果
b	在文本中后退一行
回车键	在文本中前进一行
f	屏幕内容更新一屏
h	显示帮助信息
空格键	前进到文本的下一页
q	退出less阅读程序
y	屏幕回退一行

#### 2. cat命令

cat命令的简单格式在Linux操作系统中的作用与type命令在DOS中的作用是一样的。输入“cat filename”命令会把文件的内容全部显示在屏幕上。虽然在阅读短文件内容时很有用，但是对长文件还是使用more或者less命令更好一些。

### 5.2.20 把命令结果输出到文本文件中

有些Linux命令会不断输出一屏又一屏的数据。在这种情况下，最好是把这些输出送到一个文件中去，这样用户就可以比较悠闲地阅读它们，或者今后再进行查询。这个过程叫做重定向（redirection）。

请使用大于号（>）的“command > file”格式把某个命令的输出重定向到某个文件中去。比如：“ls > listing”。

### 5.2.21 建立别名

经过一段时间之后，用户会发现自己经常需要输入一些复杂的命令，或者需要经常使用某些命令行的同一些参数。为了减少用户输入的字符量，他们可以建立一些别名（alias），也就是使用一个比较简短的命令来代替原来很长的命令内容。要想建立别名，需要编辑他们自己用户目录中的.bash\_profile文件。不管用户是使用哪个帐户登录进入系统的，都可以使用“vi ~/.bash\_profile”命令开始操作。

注意 这里介绍的操作只适用于bash shell，它是Linux操作系统缺省使用的shell。

在这个文件的末尾开始建立用户自己的别名语句。别名的定义格式是“alias name = 'command'”。举例来说，如果想把原来的“ls -la”命令定义为别名命令ls，需要加上的别名定义语句是：

```
alias ls = 'ls -la'
```

如果想使用新的别名，请先退出登录再重新登录进入原先的帐户。

注意 如果用户想在一台彩色显示器上获得一些花哨的效果，请使用下面的语句：

```
"alias 'ls -laF -colorize=yes'"
```

### 5.2.22 使用find命令确定文件存放位置

find命令是一个功能强大的操作命令，但有的时候不容易掌握。它的基本语法是：“find startingpoint flags filename”。这个命令最常见的使用方法是“find / -name file”，它从根目录开始查找文件名为file的文件或者包含file文件的目录。

这个命令比较复杂，有时候查找操作还要花费比较长的时间。表 5-9列出了一些经常使用的命令行参数和它们的使用方法。

表5-9 find命令的常用参数

参 数	说 明	格 式	示 例
atime	根据此参数说明的修改时间查找文件。加号（+）表示“在其后”，减号（-）表示“在其先”，没有加减号表示“准确时间”	find start -atime +days file_name、find start -atime - days file_name、或者 find start -atime days file_name	find /etc/rc.d - atime -5
ctime	查找在此参数指定的时间段内其目录清单列表中的数据发生改变的文件。加号（+）表示“在其后”，减号（-）表示“在其先”，没有加减号表示“准确时间”	find start -ctime +days file_name、find start -ctime - days file_name、或者 find start -ctime days file_name	find / -ctime 2

(续)

参 数	说 明	格 式	示 例
exec	查找符合 criteria 条件的文件，然后使用 command 命令来对它们进行处理。 { } (注意括号之间有空格) 表示把这些找到的文件放在什么位置；“ \ ” 符号用来结束这条命令	find start criteria -exec command { } \;	find /var/log -ctime -4 -exec grep 'http' { } \;
group	查找属于此参数说明的分组的那些文件	find start -group groupname	find / -group staff
iname	查找文件名是此参数说明的名称的那些文件	find start -iname file_name	find /home -iname data
mtime	查找在此参数指定的时间段内其内容发生改变的文件。加号 (+) 表示“在其后”，减号 (-) 表示“在其先”，没有加减号表示“准确时间”	find start -mtime +days file_name、find start -mtime -days file_name、或者 find start -mtime days file_name	find /var/log -mtime -10
name	查找文件名中包含此参数说明的字符串的那些文件	find start -name file name	find /mnt/cdrom -name '*.rpm'
print	输出满足查找条件的那些文件和目录的内容	find start criteria -print	find /home/bob -name 'Notes' -print   more
user	查找属于此参数说明的用户的那些文件	find start -user username	find / -user joy

窍门 find命令一个特别有用的使用方法是把它与xargs命令结合在一起使用。xargs命令允许用户把find命令找到的文件名们作为另外一个程序的输入。一个使用了这种组合方法的例子是：“find /home/bob -name 'Notes' | vi”，这个命令组合将在vi文本编辑器中把文件名中包含了'Notes'字样的全部文件都打开。

### 5.2.23 通过搜索locate数据库来查找文件

locate命令没有find命令功能那样强大，因为它是在一个包含了文件系统某个时刻全部内容的文件名数据库中去检索数据，而不是真的在文件系统中进行搜索。但是它要比find命令快得多，如果用户需要查找的文件是在该数据库更新之前建立的，这个命令就可以大显身手了。

使用这个命令的格式是“locate file”。

窍门 请输入“updatedb”命令更新数据库。这个数据库通常是作为一个标准的cron工作被自动更新的。

相关解决方案	请查询节号
设置备份工作在某个特定时刻运行	20.2.8

### 5.2.24 在path语句说明的路径中查找文件

whereis命令对某些任务来说是最快的查找工具。使用这个命令的格式是：“whereis file”。

whereis命令的局限性是它只在用户现有的path语句说明的路径中进行查找。如果某个文件没有存在于这些路径中，它就没有办法找到。请输入“echo \$PATH”命令查看现有路径。

### 5.2.25 在path语句中添加新路径

请按照下面的方法在bash shell (Linux操作系统的缺省shell) 中添加路径语句：

- 1) 使用“vi ~/.bash\_profile”之类的命令编辑~/.bash\_profile文件。
- 2) 找到以PATH开头的语句。
- 3) 在该语句后加上一个冒号(:)，再添上新的路径内容。
- 4) 退出再重新登录进入系统使新设置生效，也可以重新运行这个帐户的profile文件。举例来说，如果用户使用的是bash shell，请输入“source ~/.bash\_profile”命令。

### 5.2.26 在文件内容中查找文本

grep工具程序类是用来在文件内容中进行查找的。这个命令的格式是：“grep flags expression files”。在表5-10中列出了一些可供使用的命令行参数；expression是你想查找的文本，files就是准备在其中进行查找的那些文件。

比如说，我们准备在~/test文件中查找文字“red”，不考虑大小写。这个命令就是：“grep -i red ~/test”。

表5-10 grep命令中的常用参数。

参 数	说 明
-i	忽略大小写
-n	给出文字匹配处那一行的行号
-v	显示不匹配expression的那些行
-C	显示匹配处前后两行的内容

### 5.2.27 寻求帮助

Linux操作系统中的许多命令都带有扩展文档。获得帮助最快的方法是使用man (英文manual的缩写) 命令。它将显示该命令的使用手册页——如果这个命令有使用手册页的话。输入“man command”打开一个使用手册页；按下q键从中退出。

如果在某个使用手册页的开始部分说明这个文档将不再继续有效，需要另外使用texinfo程序阅读，请使用“info command”的格式阅读这个文件。

### 5.2.28 在后台运行命令程序

如果想在`不影响当前操作`的情况下运行某个命令，请使用“&”符号，其格式为“command &”。

### 5.2.29 确定当前运行的命令

有两种完全不同的方法可以用来查看当前机器上都运行着哪些东西。一个是列出当前正在运行的那些命令；另外一个列出运行着的那些进程。

#### 1. 确定有哪些命令正在运行

如果想查看后台有哪些命令正在运行，请使用jobs功能。输入这个命令后将列出正在后台运行的全部命令的清单——如果有的话。在这个清单中，最重要的信息通常是工作号，就

是那个最左边的数字。

#### 2. 确定有哪些进程正在运行

如果想查看有哪些进程正在运行，请使用 `ps` 命令。它列出的清单中只包括那些在输入这个命令的虚拟控制台上运行的进程。其中最重要的信息是最左边的数字，即 PID（英文进程号 Process ID 的缩写）。

### 5.2.30 把命令程序调到后台去

使用 `bg`（英文后台 background 的缩写）命令可以把当前正在屏幕上运行的命令调入到后台去，这样该终端就可以另做他用。请按照下面的方法把某个程序调入后台：

- 1) 输入 “ `jobs` ” 命令获取这个命令的工作号 `job`。
- 2) 按下键组合 “ `Alt + FX` ”，其中 X 是代表某个空闲终端的功能键的编号。
- 3) 登录进入刚才窗口中使用的同一个帐户。
- 4) 输入 “ `bg job` ” 把这个工作调入到后台。

窍门 这个操作通常有一个简单的完成方法，就是在显示该进程运行结果的终端上按下组合键 “ `Ctrl + Z` ”。

### 5.2.31 把后台命令调回前台来

如果想把一个正在后台运行的命令调回前台以便对它进行操作，首先要按照 “ 5.2.29 节的说明获取工作号，然后使用 `fg`（英文前台 foreground 的缩写）命令的 “ `fg job` ” 格式即可。

### 5.2.32 取消正在执行的命令

在 Linux 操作系统中可以使用好几种方法终止命令的执行。终止当前正在执行的某个命令最快的方法是按下组合键 “ `Ctrl + C` ”。这个方法只有在用户能够从某个虚拟控制台上控制这个程序的时候才奏效。

终止某个出错程序的另外一个办法是 `kill`（杀）掉它的进程。请按照下面的方法进行的操作：

- 1) 输入 “ `ps` ” 命令获取进程的 PID，这个命令要求你是运行这个程序的那个人或者是根用户。如果是根用户，请输入 “ `ps -aux` ” 命令查看所有的进程，不管是谁拥有它们的。
- 2) 在清单中找到这个出错进程。因为程序的名称列在清单的最右边，通常一眼就可以找到它。
- 3) 请记下 `ps` 命令输出清单中进程状态行最左边的 ID 数字。
- 4) 输入 “ `kill ID` ”，终止该进程。
- 5) 使用 `ps` 命令查看该进程是否仍在运行中。
- 6) 如果它没有出现在清单中，你就已经完成了操作。如果它还在运行中，请输入 “ `kill 9 ID` ”。这个命令就可以达到目的。这个命令强制性地终止了那个进程，并不进行善后工作，因此不到必要的时候最好别使用它。

### 5.2.33 清除屏幕

使用 `clear` 命令就可以清除屏幕。

### 5.2.34 一次执行多个命令

用户可以使用分号 (;) 分隔多个命令, 把它们输入在一行上。举例来说, 我们想把命令 “ls -la /etc/rc.d/init.d” 命令的输出送到文件 systeminits 中去, 然后再使用 more 命令检查这个文件, 需要输入下面的内容:

```
ls -la /etc/rc.d/init.d > systeminits ; more systeminits
```

另外一个一次运行多个命令的方法是使用命令脚本。编写命令脚本的内容将在第 18 章中详细讨论。

### 5.2.35 退出登录

使用 logout 命令就可以从某个帐户中退出登录。

### 5.2.36 重新引导启动机器

如果想重新引导 Linux 机器, 请输入 “reboot” 命令, 或者输入比较正规的 “shutdown -r now” 命令。如果机器发生了锁死现象, 不接受键盘的任何输入, 一般按下组合键 “Ctrl + Alt + Del” 能够解决问题。但是最好尽可能避免这样操作, 因为这个强制性的关机操作一般不进行善后工作。最好是检查一下 /etc/inittab 文件, 看看组合键 “Ctrl + Alt + Del” 是否是 “shutdown -h now” 命令的别名操作。如果是的话, 就可以使用这个组合键安全地关闭机器并切断电源。

### 5.2.37 系统关机

如果想关闭系统并切断电源, 请输入 “halt” 命令, 或者输入比较正规的 “shutdown -h now” 命令。