

# *The Linux Command Line(中文版)*

一本绝对值得一读的 好书





<b>引言</b>	<b>iii</b>
为什么使用命令行 . . . . .	iv
这本书讲什么 . . . . .	iv
谁应该读这本书 . . . . .	iv
这本书的内容 . . . . .	v
怎样阅读这本书 . . . . .	v
拓展阅读 . . . . .	vi
 <b>I 学习 Shell</b>	 <b>1</b>
<b>1 什么是 shell</b>	<b>2</b>
1.1 终端仿真器 . . . . .	2
1.2 第一次按键 . . . . .	2
1.3 试试运行一些简单命令 . . . . .	4
1.4 结束终端会话 . . . . .	4
1.5 拓展阅读 . . . . .	5
 <b>2 文件系统中跳转</b>	 <b>6</b>
2.1 理解文件系统树 . . . . .	6
2.2 当前工作目录 . . . . .	7
2.3 列出目录内容 . . . . .	7
2.4 更改当前工作目录 . . . . .	7
 <b>3 探究操作系统</b>	 <b>10</b>
3.1 ls 乐趣 . . . . .	10
3.2 确定文件类型 . . . . .	13
3.3 用 less 浏览文件内容 . . . . .	13
3.4 旅行指南 . . . . .	14
3.5 符号链接 . . . . .	17
3.6 硬链接 . . . . .	17
3.7 拓展阅读 . . . . .	17
 <b>4 操作文件和目录</b>	 <b>18</b>

4.1	通配符 . . . . .	19
4.2	mkdir - 创建文件夹 . . . . .	20
4.3	cp - 复制文件和目录 . . . . .	21



我想给大家讲个故事。

故事内容不是 Linus Torvalds 在 1991 年怎样写了 Linux 内核的第一个版本，因为这些内容你可以在许多 Linux 书籍中读到。我也不想告诉你，更早之前，Richard Stallman 是如何开始 GNU 项目，设计了一个免费的类似 Unix 的操作系统。那也是一个很有意义的故事，但大多数 Linux 书籍也讲到了它。

我想告诉大家一个你如何才能夺回计算机管理权的故事。

在 20 世纪 70 年代末，我刚开始和计算机打交道时，正进行着一场革命，那时的我还是一名大学生。微处理器的发明，让你我这样的普通老百姓也有可能真正拥有一台计算机。今天，人们很难想象，只有大企业 and 强大的政府才能够拥有计算机的世界，是怎样的一个世界。让我说，你想不出多少来。

今天，世界已经截然不同了。计算机遍布各个领域，从小手表到大型数据中心，及大小介于它们之间的每件东西。除了随处可见的计算机之外，我们还有一个无处不在的连接所有计算机的网络。这已经开创了一个人们可以自我营造和自由创作的奇妙的新时代，但在过去的二三十年里，一些事情仍然在发生着改变。一个大公司不断地把它的管理权强加到世界上绝大多数的计算机上，并且决定你对计算机的操作权力。幸运地是，来自世界各地的人们，正积极努力地做些事情来改变这种境况。通过编写自己的软件，他们一直在为维护电脑的管理权而战斗着。他们建设着 Linux。

一提到 Linux，许多人都会说到“自由”，但我不认为他们都知道“自由”的真正涵义。“自由”是一种权力，它决定你的计算机能做什么，同时，只有知道计算机正在做什么你才能够拥有这种“自由”。“自由”是指一台没有任何秘密的计算机，你可以从它那里了解一切，只要你用心的去寻找。



## 为什么使用命令行

你是否注意到，在电影中一个“超级黑客”坐在电脑前，从不摸一下鼠标，就能够在 30 秒内侵入到超安全的军事计算机中。这是因为电影制片人意识到，作为人类，我们应该本能地知道让计算机圆满完成工作的唯一途径，是用键盘来操纵计算机。

现在，大多数的计算机用户只是熟悉图形用户界面（GUI），并且产品供应商和此领域的学者会灌输给用户这样的思想，命令行界面（CLI）是过去使用的一种很恐怖的东西。这很不幸，因为一个友好的命令行界面，是用来和计算机进行交流沟通的，正像人类社会使用文字互通信息一样。据说，“图形用户界面让简单的任务更容易完成，而命令行界面使完成复杂的任务成为可能”，到现在这句话仍然很正确。

因为 Linux 是以 Unix 家族的操作系统为模型写成的，所以它分享了 Unix 丰富的命令行工具。Unix 在 20 世纪 80 年代初显赫一时（虽然，开发它在更早之前），结果，在普遍地使用图形界面之前，开发了一种广泛的命令行界面。事实上，一个主要的原因，Linux 开发者优先采用命令行界面而不是其他的系统，比如说 Windows NT，是因为其强大的命令行界面，可以使“完成复杂的任务成为可能”。

## 这本书讲什么

这本书对 Linux 命令行进行了综述。不像一些书籍仅仅涉及一个程序，比如像 shell 程序，bash。从更广泛的意义上来说，这本书将试着教授你如何与命令行界面友好相处。它是怎样工作的？它能做什么？使用它的最好方法是什么？

**这不是一本关于 Linux 系统管理的书。**然而任何一个关于命令行的重要讨论，都一定会牵涉到系统管理方面的内容，这本书仅仅提到一点儿管理方面的问题。这本书为读者准备好了其它的学习内容，帮助你为使用命令行打下坚实的基础，这可是完成任何一个系统管理任务所必需的至关重要的工具。

**这本书是围绕 Linux 而写的。**许多书籍，为了扩大自身的影响力，会包含一些其它平台的知识，比如一般的 Unix, MacOS X 等。这样做，书本中针对普遍话题的内容就减少了。另一方面，这本书只研究了当代 Linux 发行版。虽然，对于使用其它类似于 Unix 系统的用户来说，书中 95% 的内容是有用的，但这本书主要面向的对象是现代 Linux 命令行用户。

## 谁应该读这本书

这本书是为已经从其它平台移民到 Linux 系统的新手而写的。最有可能，你是使用某个 Windows 版本的高手。或许是老板让你去管理一个 Linux 服务器，或许你只是一个桌面用户，厌倦了系统出现的各种安全防御问题，而想要体验一下 Linux。很好，这里欢迎你们！

不过一般来说，对于 Linux 的启蒙教育，没有捷径可言。学习命令行富于挑战性，而且很费气力。这并不是说 Linux 命令行很难学，而是它的知识量很大，不容易掌握。Linux 操作系统，差不多有数以千计的命令可供用户操作。由此可见，要给自己提个醒，命令行可不是轻轻松松就能学好的。

另一方面，学习 Linux 命令行会让你受益匪浅，给你极大的回报。如果你认为，现在你已经是一个高手了。别急，其实你还不知道什么才是真正的高手。不像其他一些计算机技能，一段时间之后可能就被淘汰了，命令行知识却不会落伍，你今天所学到的，在十年以后，都会有用处。命令行通过了时间的检验。

如果你没有编程经验，也不要担心，我会带你入门。

## 这本书的内容

这些材料是经过精心安排的，很像一位老师坐在你身旁，耐心地指导你。许多作者用系统化的方式讲解这些材料，虽然从一个作者的角度考虑很有道理，但对于 Linux 新手来说，他们可能会感到非常困惑。

另一个目的，是想让读者熟悉 Unix 的思维方式，这种思维方式不同于 Windows 的。在学习过程中，我们会帮助你理解为什么某些命令会按照它们的方式工作，以及它们是怎样实现那样的工作方式的。Linux 不仅是一款软件，也是 Unix 文化的一小部分，它有自己的语言和历史渊源。同时，我也许会说过激的话。

这本书共分为五部分，每一部分讲述了不同方面的命令行知识。除了第一部分，也就是你正在阅读的这一部分，这本书还包括：

- 第二部分—学习 shell 开始探究命令行基本语言，包括命令组成结构，文件系统浏览，编写命令行，查找命令帮助文档。
- 第三部分—配置文件及环境讲述了如何编写配置文件，通过配置文件，用命令行来操控计算机。
- 第四部分—常见任务及主要工具探究了许多命令行经常执行的普通任务。类似于 Unix 的操作系统，例如 Linux，包括许多经典的命令程序，这些程序可以用来对数据进行强大的操作。
- 第五部分—编写 Shell 脚本介绍了 shell 编程，一个无可否认的基本技能，能够自动化许多常见的计算任务，很容易学。通过学习 shell 编程，你会逐渐熟悉一些关于编程语言方面的概念，这些概念也适用于其他的编程语言。

## 怎样阅读这本书

从头到尾的阅读。它并不是一本技术参考手册，实际上它更像一本故事书，有开头，过程，结尾。

### 前提条件

为了使用这本书，你需要安装 Linux 操作系统。你可以通过两种方式，来完成安装。

1. 在一台（不是很新）的电脑上安装 Linux。你选择哪个 Linux 发行版安装，是无关紧要的事。虽然大多数人一开始选择安装 Ubuntu, Fedora, 或者 OpenSUSE。如果你拿不定主意，那就先试试 Ubuntu。由于主机硬件配置不同，安装 Linux 时，你可能不费吹灰之力就装上了，也可能费了九牛二虎之力还装不上。所以我建议，一台使用了几年的台式机，至少要有 256M 的内存，6G 的硬盘可用空间。尽可能避免使用笔记本电脑和无线网络，在 Linux 环境下，它们经常不能工作。
2. 使用“Live CD.”许多 Linux 发行版都自带一个比较酷的功能，你可以直接从系统安装盘 CDROM 中运行 Linux，而不必安装 Linux。开机进入 BIOS 设置界面，更改引导项，设置为“从 CDROM 启动”。

不管你怎样安装 Linux，为了练习书中介绍的知识，你需要有超级用户（管理员）权限。

当你在自己的电脑上安装了 Linux 系统之后，就开始一边阅读本书，一边练习吧。本书大部分内容都可以自己动手练习，坐下来，敲入命令，体验一下吧。

## 为什么我不叫它“GNU/Linux”

在某些领域，把 Linux 操作系统称为“GNU/Linux 操作系统。”是比较明智的做法。但“Linux”的问题在于，没有一个完全正确的方式能为它命名，因为它是由许许多多，分布在世界各地的贡献者们，合作开发而成的。从技术层面讲，Linux 只是操作系统的内核名字，没别的含义。当然内核非常重要，有了内核，操作系统才能运行起来，但它并不能构成一个完整的操作系统。

Richard Stallman 是一个天才的哲学家，自由软件运动创始人，自由软件基金会创办者，他创建了 GNU 工程，编写了第一版 GNU C 编译器 (gcc)，创立了 GNU 通用公共协议 (the GPL) 等等。他坚持把 Linux 称为“GNU/Linux”，为的是准确地反映 GNU 工程对 Linux 操作系统的贡献。然而，GNU 项目早于 Linux 内核，而 GNU 项目的贡献得到了极高的赞誉，再把 GNU 用在 Linux 名字里，这对其他每个为 Linux 的发展做出重大贡献的程序员来说，就不公平了。

在目前流行的用法中，“Linux”指的是内核以及在一个典型的 Linux 发行版中所包含的所有免费及开源软件；也就是说，整个 Linux 生态系统，不只有 GNU 项目软件。在操作系统商界，好像喜欢使用单个词的名字，比如说 DOS, Windows, MacOS, Solaris, Irix, AIX. 所以我选择用流行的命名规则。然而，如果你喜欢用“GNU/Linux”，当你读这本书时，可以搜索并代替“Linux”。我不介意。

## 拓展阅读

- Wikipedia 网站上有些介绍本章提到的名人的文章，以下是链接地址：  
[http://en.wikipedia.org/wiki/Linux\\_Torvalds](http://en.wikipedia.org/wiki/Linux_Torvalds)  
[http://en.wikipedia.org/wiki/Richard\\_Stallman](http://en.wikipedia.org/wiki/Richard_Stallman)
- 介绍自由软件基金会及 GNU 项目的网站和文章：  
[http://en.wikipedia.org/wiki/Free\\_Software\\_Foundation](http://en.wikipedia.org/wiki/Free_Software_Foundation)  
<http://www.fsf.org>  
<http://www.gnu.org>
- Richard Stallman 用了大量的文字来叙述“GNU/Linux”的命名问题，可以浏览以下网页：  
<http://www.gnu.org/gnu/why-gnu-linux.html>  
<http://www.gnu.org/gnu/why-gnu-linux.html>  
<http://www.gnu.org/gnu/gnu-linux-faq.html#tools>



## Part I

# 学习 Shell

# 什么是 shell



一说到命令行，我们真正指的是 shell。shell 就是一个程序，它接受从键盘输入的命令，然后把命令传递给操作系统去执行。几乎所有的 Linux 发行版都提供一个名为 bash 的程序，bash 是 shell 的一种，来自于 GNU 项目。“bash”是“Bourne Again SHell”的首字母缩写，所指的是这样一个事实，bash 是 sh 的增强版，sh 是最初 Unix 的 shell 程序，由 Steve Bourne 写成。

## 1.1 终端仿真器

当使用图形用户界面时，我们需要另一个叫做终端仿真器的程序，去和 shell 交互。浏览一下我们的桌面菜单，我们可能会找到一个。虽然在菜单里它可能都被简单地称为“terminal”，但是 KDE 用的是 konsole 程序，而 GNOME 则使用 gnome-terminal。还有其他一些终端仿真器可供 Linux 使用，但基本上，它们都是为了完成同样的事情，让我们能访问 shell。也许，你会喜欢上这个或那个终端，由于它所附加的一系列花俏功能。

## 1.2 第一次按键

好，开始吧。启动终端仿真器！一旦它运行起来，我们应该能够看到一行类似下面文字的输出：

```
[me@linuxbox ~]$
```

这叫做 shell 提示符，当 shell 准备好了去接受输入时，它就会出现。然而，它可能会以各种各样的面孔显示，这则取决于不同的 Linux 发行版，它通常包括你的用户名主机名，紧接着当前工作目录（关于它的知识有点多）和一个美元符号。

如果提示符的最后一个字符是“#”，而不是“\$”，那么这个终端会话就有超级用户权限。这意味着，我们或者是以根用户的身份登录，或者是我们选择的终端仿真器提供超级用户（管理员）权限。

假定到目前为止，所有事情都进行顺利，那我们试着打字吧。在提示符下敲入一些乱七八糟的无用数据，如下所示：

```
[me@linuxbox ~]$ kaekfjaeifj
```

因为这个命令没有任何意义，所以 shell 会提示错误信息，并让我们再试一下：

```
bash: kaekfjaeifj: command not found
[me@linuxbox ~]$
```

## 命令历史

如果按下上箭头按键，我们会看到刚才输入的命令“kaekfjaeifj”重新出现在提示符之后。这就叫做命令历史。许多 Linux 发行版默认保存最后输入的 500 个命令。按下下箭头按键，先前输入的命令就消失了。

## 移动光标

可借助上箭头按键，来回想起上次输入的命令。现在试着使用左右箭头按键。看一下怎样把光标定位到命令行的任意位置？通过使用箭头按键，使编辑命令变得轻松些。

### 关于鼠标和光标

虽然，shell 是和键盘打交道的，但你也可以在终端仿真器里使用鼠标。X 窗口系统（使 GUI 工作的底层引擎）内建了一种机制，支持快速拷贝和粘贴技巧。如果你想高亮一些文本，可以按下鼠标左键，沿着文本拖动鼠标（或者双击一个单词），那么这些高亮的文本就被拷贝到了一个由 X 管理的缓冲区里面。然后按下鼠标中键，这些文本就被粘贴到光标所在的位置。试试看。

注意：不要受诱惑在一个终端窗口里，使用 Ctrl-c 和 Ctrl-v 快捷键，来执行拷贝和粘贴操作。它们不起作用。对于 shell 来说，这些控制代码有着不同的含义，它们被赋值，早于 Microsoft Windows 许多年。

你的图形桌面环境（像 KDE 或 GNOME），努力想和 Windows 一样，可能会把它的聚焦策略设置成“单击聚焦”。这意味着，为了让窗口聚焦（变得活跃）你需要单击它。这与“聚焦跟随着鼠标”的传统 X 行为相反，传统 X 行为是指只要把鼠标移动到一个窗口的上方，这个窗口就成为活动窗口。这个窗口不会成为前端窗口，直到你单击它，但它能接受输入。设置聚焦策略为“聚焦跟随着鼠标”，可以使拷贝和粘贴技巧更有益。尝试一下。给它一个机会，我想你会喜欢上它的。在窗口管理器的配置程序中，你会找到这个设置。

### 1.3 试试运行一些简单命令

现在，我们学习了怎样输入命令，那我们执行一些简单的命令吧。第一个命令是 `date`。这个命令显示系统当前时间和日期。

```
[me@linuxbox ~]$ date
Thu Oct 25 13:51:54 EDT 2007
```

一个相关联的命令，`cal`，它默认显示当前月份的日历。

```
[me@linuxbox ~]$ cal
October 2007
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 31
```

查看磁盘剩余空间的数量，输入 `df`：

```
[me@linuxbox ~]$ df
Filesystem            1K-blocks      Used Available Use% Mounted on
/dev/sda2              15115452    5012392   9949716   34% /
/dev/sda5              59631908   26545424   30008432   47% /home
/dev/sda1               147764      17370    122765   13% /boot
tmpfs                  256856         0     256856    0% /dev/shm
```

同样地，显示空闲内存的数量，输入命令 `free`。

```
[me@linuxbox ~]$ free
total        used        free      shared    buffers     cached
Mem:         2059676    846456    1213220         0
44028        360568
-/+ buffers/cache:    441860    1617816
Swap:        1042428         0     1042428
```

### 1.4 结束终端会话

我们可以终止一个终端会话，通过关闭终端仿真器窗口，或者是在 `shell` 提示符下输入 `exit` 命令：

```
[me@linuxbox ~]$ exit
```

## 幕后控制台

即使，终端仿真器没有运行，几个终端会话仍然在后台运行着。它们叫做虚拟终端或者是虚拟控制台。在大多数 Linux 发行版中，这些终端会话都可以访问，按下 Ctrl-Alt-F1 到 Ctrl-Alt-F6 访问不同的虚拟终端。当一个会话被访问的时候，它会显示登录提示框，我们需要输入用户名和密码。从一个虚拟控制台转换到另一个，按下 Alt 和 F1-F6。返回图形桌面，按下 Alt-F7。

## 1.5 拓展阅读

- 想了解更多关于 Steve Bourne 的故事，Bourne Shell 之父，读一下这篇文章：  
[http://en.wikipedia.org/wiki/Steve\\_Bourne](http://en.wikipedia.org/wiki/Steve_Bourne)
- 这是一篇关于在计算机领域里，shells 概念的文章：  
[http://en.wikipedia.org/wiki/Shell\\_\(computing\)](http://en.wikipedia.org/wiki/Shell_(computing))



# 文件系统中跳转



我们需要学习的第一件事（除了打字之外）是如何在 Linux 文件系统中跳转。在这一章节中，我们将介绍以下命令：

- `pwd` — 打印出当前工作目录名
- `cd` — 更改目录
- `ls` — 列出目录内容

## 2.1 理解文件系统树

正如 Windows，一个类似于 Unix 的操作系统，比如说 Linux，以分层目录结构来组织所有文件。这就意味着所有文件组成了一棵树型目录（有时候在其它系统中叫做文件夹），这个目录树可能包含文件和其它的目录。文件系统的第一级目录称为根目录。根目录包含文件和子目录，子目录包含更多的文件和子目录，依此类推。

注意不同于 Windows，Windows 每个存储设备都有一个独自的文件系统，类似于 Unix 的操作系统，比如说 Linux，总是有一个单一的文件系统树，不管有多少个磁盘或者存储设备连接到计算机上。根据系统管理员的兴致，存储设备连接到（或着更精确些，是挂载到）目录树的各个节点上。系统管理员负责维护系统安全。

## 2.2 当前工作目录

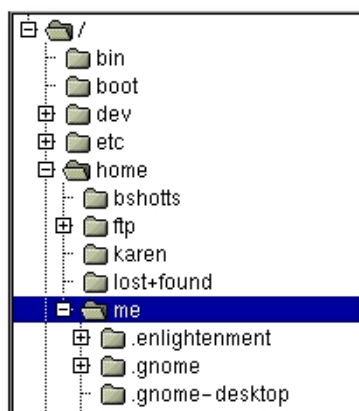


图 2.1: File system tree as shown by a graphical file manager

大多数人都可能熟悉图形文件管理器，它描述了文件系统树的结构，正如图 1 所示。注意通常，这是一棵倒置的树，也就是说，树根在最上面，而各个枝干在下面展开。

然而，命令行没有图片，所以我们需要考虑用不同的方法，在文件系统树中跳转。

把文件系统想象成一个迷宫形状，就像一棵倒立的大树，我们站在迷宫的中间位置。在任意时刻，我们处于一个目录里面，我们能看到这个目录包含的所有文件，以及通往上面目录（父目录）的路径，和下面的各个子目录。我们所在的目录则称为当前工作目录。我们使用 `pwd`（打印工作目录）命令，来显示当前工作目录。

```
[me@linuxbox ~]$ pwd
/home/me
```

当我们首次登录系统后，（或者启动终端仿真器会话后），当前工作目录设置成主目录。每个用户都有他自己的主目录，当用户以普通用户的身份操控系统时，主目录是唯一允许用户编写文件的地方。

## 2.3 列出目录内容

列出一个目录包含的文件及子目录，使用 `ls` 命令。

```
[me@linuxbox ~]$ ls
Desktop Documents Music Pictures Public Templates Videos
```

实际上，用 `ls` 命令可以列出任一个目录的内容，而不只是当前工作目录的内容。`ls` 命令还能完成许多有趣的事情。在下一章节，我们将介绍更多关于 `ls` 的知识。

## 2.4 更改当前工作目录

要更改工作目录（此刻，我们站在树形迷宫里面），我们用 `cd` 命令。输入 `cd`，然后输入你想要的工作目录的路径名，就能实现愿望。路径名就是沿着目录树的分支到达想要的目录，期间所经过的路线。

路径名可通过两种方式来指定，一个是绝对路径，另一个是相对路径。首先处理绝对路径。

## 绝对路径

绝对路径开始于根目录，紧跟着目录树的一个个分支，一直到达期望的目录或文件。例如，你的系统中有一个目录，大多数系统程序都安装在这个目录下。这个目录的路径名是 `/usr/bin`。它意味着从根目录（用开头的“/”表示）开始，有一个叫“usr”的目录包含了目录“bin”。

```
[me@linuxbox ~]$ cd /usr/bin
[me@linuxbox bin]$ pwd
/usr/bin
[me@linuxbox bin]$ ls
... Listing of many, many files ...
```

我们把工作目录转到 `/usr/bin` 目录下，里面装满了文件。注意 shell 提示符是怎样改变的。为了方便，通常设置提示符自动显示工作目录名。

## 相对路径

绝对路径从根目录开始，直到它的目的地，而相对路径开始于工作目录。一对特殊符号来表示相对位置，在文件系统树中。这对特殊符号是“.”（点）和“..”（点点）。

符号“.”指的是工作目录，“..”指的是工作目录的父目录。下面的例子说明怎样使用它。再次更改工作目录到 `/usr/bin`：

```
[me@linuxbox ~]$ cd /usr/bin
[me@linuxbox bin]$ pwd
/usr/bin
```

好的，比方说更改工作目录到 `/usr/bin` 的父目录 `/usr`。可以通过两种方法来实现。或者使用绝对路径名：

```
[me@linuxbox bin]$ cd /usr
[me@linuxbox usr]$ pwd
/usr
```

或者，使用相对路径：

```
[me@linuxbox bin]$ cd ..
[me@linuxbox usr]$ pwd
/usr
```

两种不同的方法，一样的结果。我们应该选哪一个呢？输入量最少的那个。

同样地，从目录 `/usr/` 到 `/usr/bin` 也有两种途径。或者使用绝对路径：

```
[me@linuxbox usr]$ cd /usr/bin
[me@linuxbox bin]$ pwd
/usr/bin
```

或者，用相对路径：

```
[me@linuxbox usr]$ cd ./bin
[me@linuxbox bin]$ pwd
/usr/bin
```

有一件很重要的事，我必须指出来。在几乎所有的情况下，你可以省略“./”。它是隐含地。输入：

```
[me@linuxbox usr]$ cd bin
```

实现相同的效果，如果不指定一个文件的目录，那它的工作目录会被假定为当前工作目录。

有用的快捷键

在表 2-1 中，列举出了一些快速改变当前工作目录的有效方法。

表 2.1: cd 快捷键

快捷键	运行结果
cd	更改工作目录到主目录。
cd -	更改工作目录到先前的工作目录。
cd user_name	更改工作目录到用户主目录。例如, cd bob 会更改工作目录到用户“bob”的主目录。

关于文件名的重要规则

- 1. 以“.”字符开头的文件名是隐藏文件。这仅表示，ls 命令不能列出它们，除非使用 ls -a 命令。当你创建帐号后，几个配置帐号的隐藏文件被放置在你的主目录下。稍后，我们会仔细研究一些隐藏文件，来定制你的系统环境。另外，一些应用程序也会把它们配置文件以隐藏文件的形式放在你的主目录下。
- 2. 文件名和命令名是大小写敏感的。文件名“File1”和“file1”是指两个不同的文件名。
- 3. Linux 没有“文件扩展名”的概念，不像其它一些系统。可以用你喜欢的任何名字来给文件起名。文件内容或用途由其它方法来决定。虽然类似 Unix 的操作系统，不用文件扩展名来决定文件的内容或用途，但是应用程序会。
- 4. 虽然 Linux 支持长文件名，文件名可能包含空格，标点符号，但标点符号仅限使用“.”，“-”，下划线。最重要的是，不要在文件名中使用空格。如果你想表示词与词间的空格，用下划线字符来代替。过些时候，你会感激自己这样做。





既然我们已经知道了如何在文件系统中跳转，是时候开始 Linux 操作系统之旅了。在开始之前，我们先学习一些对研究 Linux 系统有帮助的命令。

- ls 一列出目录内容
- file 一确定文件类型
- less 一浏览文件内容

### 3.1 ls 乐趣

有充分的理由证明，ls 可能是用户最常使用的命令。通过它，我们可以知道目录的内容，以及各种各样重要文件和目录的属性。正如我们所知道的，只简单的输入 ls 就能看到在当前目录下所包含的文件和子目录列表。

```
[me@linuxbox ~]$ ls  
Desktop Documents Music Pictures Publica Templates Videos
```

除了当前工作目录以外，也可以列出指定目录的内容，就像这样：



```
me@linuxbox ~]$ ls /usr
bin  games  kerberos  libexec  sbin  src
etc  include lib      local   share  tmp
```

甚至可以列出多个指定目录的内容。在这个例子中，将会列出用户主目录（用字符“~”代表）和/usr 目录的内容：

```
[me@linuxbox ~]$ ls ~ /usr
/home/me:
Desktop  Documents  Music  Pictures  Public  Templates  Videos

/usr:
bin  games  kerberos  libexec  sbin  src
etc  include lib      local   share  tmp
```

我们也可以改变输出格式，来得到更多的细节：

```
[me@linuxbox ~]$ ls -l
total 56
drwxrwxr-x 2 me me 4096 2007-10-26 17:20 Desktop
drwxrwxr-x 2 me me 4096 2007-10-26 17:20 Documents
drwxrwxr-x 2 me me 4096 2007-10-26 17:20 Music
drwxrwxr-x 2 me me 4096 2007-10-26 17:20 Pictures
drwxrwxr-x 2 me me 4096 2007-10-26 17:20 Public
drwxrwxr-x 2 me me 4096 2007-10-26 17:20 Templates
drwxrwxr-x 2 me me 4096 2007-10-26 17:20 Videos
```

使用 ls 命令的“-l”选项，则结果以长模式输出。

## 选项和参数

我们将学习一个非常重要的知识点，大多数命令是如何工作的。命令名经常会带有一个或多个用来更正命令行为的选项，更进一步，选项后面会带有一个或多个参数，这些参数是命令作用的对象。所以大多数命令看起来像这样：

```
command -options arguments
```

大多数命令使用的选项，是由一个中划线加上一个字符组成，例如，“-l”，但是许多命令，包括来自于 GNU 项目的命令，也支持长选项，长选项由两个中划线加上一个字符组成。当然，许多命令也允许把多个短选项串在一起使用。下面这个例子，ls 命令有两个选项，“-l”选项产生长格式输出，“-t”选项按文件修改时间的先后来排序。

```
[me@linuxbox ~]$ ls -lt
```

加上长选项“-reverse”，则结果会以相反的顺序输出：

```
[me@linuxbox ~]$ ls -lt --reverse
```

ls 命令有大量的选项。表 3-1 列出了最常使用的选项。

表 3.1: ls 命令选项

选项	长选项	描述
-a	-all	列出所有文件，甚至包括文件名以圆点开头的隐藏文件。
-d	-directory	通常，如果指定了目录名，ls 命令会列出这个目录中的内容，而不是目录本身。把这个选项与 -l 选项结合使用，可以看到所指定目录的详细信息，而不是目录中的内容。
-F	-classify	这个选项会在每个所列出的名字后面加上一个指示符。例如，如果名字是目录名，则会加上一个 “/” 字符。
-h	-human-readable	以长格式列出。以人们可读的格式，而不是以字节数来显示文件的大小。
-l		以长格式显示结果。
-r	-reverse	以相反的顺序来显示结果。通常，ls 命令的输出结果按照字母升序排列。
-S		命令输出结果按照文件大小来排序。
-t		按照修改时间来排序。

深入研究长格式输出

正如我们先前知道的，“-l” 选项导致 ls 的输出结果以长格式输出。这种格式包含大量的有用信息。下面的例子目录来自于 Ubuntu 系统：(注：由于排版原因，删除了原文的的一部分.)

```
-rw-r--r-- 1 root root 3576296 2007-04-03 11:05 Experience ubuntu.
ogg
-rw-r--r-- 1 root root 1186219 2007-04-03 11:05 kubuntu-leaflet.png
-rw-r--r-- 1 root root 47584 2007-04-03 11:05 logo-Edubuntu.png
-rw-r--r-- 1 root root 44355 2007-04-03 11:05 logo-Kubuntu.png
-rw-r--r-- 1 root root 34391 2007-04-03 11:05 logo-Ubuntu.png
-rw-r--r-- 1 root root 32059 2007-04-03 11:05 oo-cd-cover.odf
```

选一个文件，来看一下各个输出字段的含义 (表 3-2):

表 3.2: ls 长格式列表的字段

字段	含义
-rw-r--r--	对于文件的访问权限。第一个字符指明文件类型。在不同类型之间，开头的 “-” 说明是一个普通文件，“d” 表明是一个目录。其后三个字符是文件所有者的访问权限，再其后的三个字符是文件所属组中成员的访问权限，最后三个字符是其他所有人的访问权限。这个字段的完整含义将在第十章讨论。
1	文件的硬链接数目。参考随后讨论的关于链接的内容。
root	文件所属用户组的名字。
32059	以字节数表示的文件大小。
2007-04-03 11:05	上次修改文件的时间和日期。
oo-cd-cover.odf	文件名。

## 3.2 确定文件类型

随着探究操作系统的进行，知道文件包含的内容是很有用的。我们将用 `file` 命令来确定文件的类型。我们之前讨论过，在 Linux 统中，并不要求文件名来反映文件的内容。然而，一个类似“`picture.jpg`”的文件名，我们会期望它包含 JPEG 压缩图像，但 Linux 却不这样要求它。可以这样调用 `file` 命令：

```
file filename
```

当调用 `file` 命令后，`file` 命令会打印出文件内容的简单描述。例如：

```
[me@linuxbox ~]$ file picture.jpg
picture.jpg: JPEG image data, JFIF standard 1.01
```

有许多类型的文件。事实上，在类似于 Unix 操作系统中比如说 Linux，有个普遍的观念就是“任何东西都是一个文件”。随着课程的进行，我们将会明白这句话的真谛。

虽然系统中许多文件格式是熟悉的，例如 MP3 和 JPEG 文件，但也有一些文件格式比较含蓄，极少数文件相当陌生。

## 3.3 用 less 浏览文件内容

`less` 命令是一个用来浏览文本文件的程序。纵观 Linux 系统，有许多人类可读的文本文件。`less` 程序为我们检查文本文件提供了方便。

### 什么是“文本”

在计算机中，有许多方法可以表达信息。所有的方法都涉及到，在信息与一些数字之间确立一种关系，而这些数字可以用来表达信息。毕竟，计算机只能理解数字，这样所有的数据都被转换成数值表示法。

有些数值表达法非常复杂（例如压缩的视频文件），而其它的就相当简单。最早也是最简单的一种表达法，叫做 ASCII 文本。ASCII（发音是“*As-Key*”）是美国信息交换标准码的简称。这是一个简单的编码方法，它首先被用在电传打字机上，用来实现键盘字符到数字的映射。

文本是简单的字符与数字之间的一对一映射。它非常紧凑。五十个字符的文本翻译成五十个字节的数据。文本只是包含简单的字符到数字的映射，理解这点很重要。它和一些文字处理器文档不一样，比如说由微软和 OpenOffice.org 文档编辑器创建的文件。这些文件，和简单的 ASCII 文件形成鲜明对比，它们包含许多非文本元素，来描述它的结构和格式。普通的 ASCII 文件，只包含字符本身，和一些基本的控制符，像制表符，回车符及换行符。纵观 Linux 系统，许多文件以文本格式存储，也有许多 Linux 工具来处理文本文件。甚至 Windows 也承认这种文件格式的重要性。著名的 NOTEPAD.EXE 程序就是一个 ASCII 文本文件编辑器。

为什么我们要查看文本文件呢？因为许多包含系统设置的文件（叫做配置文件），是以文本格式存储的，阅读它们可以更深入的了解系统是如何工作的。另外，许多系统所用到的实际程序（叫做脚本）也是以这种格式存储的。在随后的章节里，我们将要学习怎样编辑文本文件，为的是修改系统设置，还要学习编写自己的脚本文件，但现在我们只是看看它们的内容而已。

`less` 命令是这样使用的：

```
less filename
```

一旦运行起来，less 程序允许你前后滚动文件。例如，要查看一个定义了系统中全部用户身份的文件，输入以下命令：

```
[me@linuxbox ~]$ less /etc/passwd
```

一旦 less 程序运行起来，我们就能浏览文件内容了。如果文件内容多于一页，那么我们可以上下滚动文件。按下“q”键，退出 less 程序。

下表列出了 less 程序最常使用的键盘命令。

表 3.3: less 命令

命令	行为
Page UP or b	向后翻滚一页
Page Down or space	向前翻动一页
UP Arrow	向前移动一行
Down Arrow	向后移动一行
G	移动到最后一行
1G or g	移动到开头一行
/characters	向前查找指定的字符串
n	向前查找下一个出现的字符串，这个字符串是之前所指定查找的
h	显示帮助屏幕
q	退出 less 程序

less 就是 more（禅语：色即是空）

less 程序是早期 Unix 程序 more 的改进版。“less”这个名字，对习语“less is more”开了个玩笑，这个习语是现代主义建筑师和设计者的座右铭。

less 属于“页面调度器”程序类，这些程序允许通过页方式，在一页中轻松地浏览长长的文本文档。然而 more 程序只能向前分页浏览，而 less 程序允许前后分页浏览，它还有很多其它的特性。

3.4 旅行指南

Linux 系统中，文件系统布局与类似 Unix 系统的文件布局很相似。实际上，一个已经发布的标准，叫做 Linux 文件系统层次标准，详细说明了这种设计模式。不是所有 Linux 发行版都根据这个标准，但大多数都是。

下一步，我们将在文件系统中游玩，来了解 Linux 系统的工作原理。这会给你一个温习跳转命令的机会。我们会发现很多有趣的文件都是普通的可读文本。将开始旅行，做做以下练习：

- 1. cd 到给定目录
- 2. 列出目录内容 ls -l
- 3. 如果看到一个有趣的文件，用 file 命令确定文件内容
- 4. 如果文件看起来像文本，试着用 less 命令浏览它

**记得复制和粘贴技巧！** 如果你正在使用鼠标，双击文件名，来复制它，然后按下鼠标中键，粘贴文件名到命令行中。

在系统中游玩时，不要害怕粘花惹草。普通用户是很难把东西弄乱的。那是系统管理员的工作！如果一个命令抱怨一些事情，不要管它，尽管去玩别的东西。花一些时间四处走走。系统是我们自己的，尽情地探究吧。记住在 Linux 中，没有秘密存在！表 3-4 仅仅列出了一些我们可以浏览的目录。闲暇时试试看！

表 3.4: Linux 系统中的目录

目录	注解
/	根目录，万物起源。
/bin	包含系统启动和运行所必须的二进制程序。
/boot	包含 Linux 内核，最初的 RMA 磁盘映像（系统启动时，由驱动程序所需），和启动加载程序。有趣的文件： <ul style="list-style-type: none"><li>• /boot/grub/grub.conf or menu.lst，被用来配置启动加载程序。</li><li>• /boot/vmlinuz，Linux 内核。</li></ul>
/dev	这是一个包含设备结点的特殊目录。“一切都是文件”，也使用于设备。在这个目录里，内核维护着它支持的设备。
/etc	这个目录包含所有系统层面的配置文件。它也包含一系列的 shell 脚本，在系统启动时，这些脚本会运行每个系统服务。这个目录中的任何文件应该是可读的文本文件。有意思的文件：虽然/etc 目录中的任何文件都有趣，但这里只列出了一些我一直喜欢的文件： <ul style="list-style-type: none"><li>• /etc/crontab，定义自动运行的任务。</li><li>• /etc/fstab，包含存储设备的列表，以及与他们相关的挂载点。</li><li>• /etc/passwd，包含用户帐号列表。</li></ul>
/home	在通常的配置环境下，系统会在/home 下，给每个用户分配一个目录。普通只能在他们自己的目录下创建文件。这个限制保护系统免受错误的用户活动破坏。
/lib	包含核心系统程序所需的库文件。这些文件与 Windows 中的动态链接库相似。
/lost+found	每个使用 Linux 文件系统的格式化分区或设备，例如 ext3 文件系统，都会有这个目录。当部分恢复一个损坏的文件系统时，会用到这个目录。除非文件系统真正的损坏了，那么这个目录会是个空目录。

接下页



表 3.4 (续)

目录	注解
/media	在现在的 Linux 系统中，/media 目录会包含可移除媒体设备的挂载点，例如 USB 驱动器，CD-ROMs 等等。这些设备连接到计算机之后，会自动地挂载到这个目录结点下。
/mnt	在早些的 Linux 系统中，/mnt 目录包含可移除设备的挂载点。
/opt	这个/opt 目录被用来安装“可选的”软件。这个主要用来存储可能安装在系统中的商业软件产品。
/proc	这个/proc 目录很特殊。从存储在硬盘上的文件的意义上说，它不是真正的文件系统。反而，它是一个由 Linux 内核维护的虚拟文件系统。它所包含的文件是内核的窥视孔。这些文件是可读的，它们会告诉你内核是怎样监管计算机的。
/root	root 帐户的主目录。
/sbin	这个目录包含“系统”二进制文件。它们是完成重大系统任务的程序，通常为超级用户保留。
/tmp	这个/tmp 目录，是用来存储由各种程序创建的临时文件的地方。一些配置，导致系统每次重新启动时，都会清空这个目录。
/usr	在 Linux 系统中，/usr 目录可能是最大的一个。它包含普通用户所需要的所有程序和文件。
/usr/bin	/usr/bin 目录包含系统安装的可执行程序。通常，这个目录会包含许多程序。
/usr/lib	包含由/usr/bin 目录中的程序所用的共享库。
/usr/local	这个/usr/local 目录，是非系统发行版自带，却打算让系统使用的程序的安装目录。通常，由源码编译的程序会安装在/usr/local/bin 目录下。新安装的 Linux 系统中，会存在这个目录，但却是空目录，直到系统管理员放些东西到它里面。
/usr/sbin	包含许多系统管理程序。
/usr/share	/usr/share 目录包含许多由/usr/bin 目录中的程序使用的共享数据。其中包括像默认的配置文件的图标，桌面背景，音频文件等等。
/usr/share/doc	大多数安装在系统中的软件包会包含一些文档。在/usr/share/doc 目录下，我们可以找到按照软件包分类的文档。
/var	除了/tmp 和/home 目录之外，相对来说，目前我们看到的目录是静态的，这是说，它们的内容不会改变。/var 目录是可能需要改动的文件存储的地方。各种数据库，假脱机文件，用户邮件等等，都驻扎在这里。
/var/log	这个/var/log 目录包含日志文件，各种系统活动的记录。这些文件非常重要，并且应该时时监测它们。其中最重要的一个文件是/var/log/messages。注意，为了系统安全，在一些系统中，你必须是超级用户才能查看这些日志文件。

## 3.5 符号链接

在我们到处查看时，我们可能会看到一个目录，列出像这样的一条信息：

```
lrwxrwxrwx 1 root root 11 2007-08-11 07:34 libc.so.6 -> libc-2.6.so
```

注意，这条信息第一个字符是“1”，并且看起来像有两个文件名？这是一个特殊文件，叫做符号链接（也称为软链接或者 symlink）。在大多数类似 Unix 系统中，有可能一个文件被多个文件名参考。虽然这种特性的意义并不明显，但它真地很有用。

描绘一下这样的情景：一个程序要求使用某个包含在名为“foo”文件中的共享资源，但是“foo”经常改变版本号。这样，在文件名中包含版本号，会是一个好主意，因此管理员或者其它相关方，会知道安装了哪个“foo”版本。这又会导致一个问题。如果我们更改了共享资源的名字，那么我们必须跟踪每个可能使用了这个共享资源的程序，当每次这个资源的新版本被安装后，都要让使用了它的程序去寻找新的资源名。这听起来很没趣。

这就是符号链接存在至今的原因。比方说，我们安装了文件“foo”的 2.6 版本，它的文件名是“foo-2.6”，然后创建了叫做“foo”的符号链接，这个符号链接指向“foo-2.6”。这意味着，当一个程序打开文件“foo”时，它实际上是打开文件“foo-2.6”。现在，每个人都很高兴。依赖于“foo”文件的程序能找到这个文件，并且我们能知道安装了哪个文件版本。当升级到“foo-2.7”版本的时候，仅添加这个文件到文件系统中，删除符号链接“foo”，创建一个指向新版本的符号链接。这不仅解决了版本升级问题，而且还允许在系统中保存两个不同的文件版本。假想“foo-2.7”有个错误（该死的开发者！），那我们得回到原来的版本。一样的操作，我们只需要删除指向新版本的符号链接，然后创建指向旧版本的符号链接就可以了。

在上面列出的目录（来自于 Fedora 的 /lib 目录）展示了一个叫做“libc.so.6”的符号链接，这个符号链接指向一个叫做“libc-2.6.so”的共享库文件。这意味着，寻找文件“libc.so.6”的程序，实际上得到是文件“libc-2.6.so”。在下一章，我们将学习如何建立符号链接。

## 3.6 硬链接

尽管在讨论链接问题，我们需要提一下，还有一种链接类型，叫做硬链接。硬链接同样允许文件有多个名字，但是硬链接以不同的方法来创建多个文件名。在下一章中，我们会谈到更多符号链接与硬链接之间的差异问题。

## 3.7 拓展阅读

- 完整的 Linux 文件系统层次标准可通过以下链接找到：  
<http://www.pathname.com/fhs/>
- 一篇关于 UNIX 和类 UNIX 系统目录结构的文章：  
[http://en.wikipedia.org/wiki/Unix\\_directory\\_structure](http://en.wikipedia.org/wiki/Unix_directory_structure)
- ASCII 文本格式的详细描述：  
<http://en.wikipedia.org/wiki/ASCII>

# 操作文件和目录



此时此刻，我们已经准备好了做些真正的工作！这一章节将会介绍以下命令：

- `cp` —复制文件和目录
- `mv` —移动/重命名文件和目录
- `mkdir` —创建目录
- `rm` —删除文件和目录
- `ln` —创建硬链接和符号链接

这五个命令属于最常使用的 Linux 命令之列。它们用来操作文件和目录。

现在，坦诚地说，用图形文件管理器来完成一些由这些命令执行的任务会更容易些。使用文件管理器，我们可以把文件从一个目录拖放到另一个目录，剪贴和粘贴文件，删除文件等等。那么，为什么还使用早期的命令行程序呢？

答案是命令行程序，功能强大灵活。虽然图形文件管理器能轻松地实现简单的文件操作，但是对于复杂的文件操作任务，则使用命令行程序比较容易完成。例如，怎样复制一个目录下的 HTML 文件到另一个目录，但这些 HTML 文件不存在于目标目录，或者是文件版本新于目标目录里的文件？要完成这个任务，使用文件管理器相当难，使用命令行相当容易：

```
cp -u *.html destination
```

### 4.1 通配符

在开始使用命令之前，我们需要介绍一个使命令行如此强大的 shell 特性。因为 shell 频繁地使用文件名，shell 提供了特殊字符来帮助你快速指定一组文件名。这些特殊字符叫做通配符。使用通配符（也以文件名代换著称）允许你依据字符类型来选择文件名。下表列出这些通配符以及它们所选择的对象：

表 4.1: 通配符

通配符	意义
*	匹配任意多个字符（包括零个或一个）
?	匹配任意一个字符（不包括零个）
[characters]	匹配任意一个属于字符集中的字符
[!characters]	匹配任意一个不是字符集中的字符
[:class:]	匹配任意一个属于指定字符类中的字符

表4.2列出了最常使用的字符类：

表 4.2: 通配符

字符类	意义
[:alnum:]	匹配任意一个字母或数字
[:alpha:]	匹配任意一个字母
[:digit:]	匹配任意一个数字
[:lower:]	匹配任意一个小写字母
[:upper:]	匹配任意一个大写字母

借助通配符，为文件名构建非常复杂的选择标准成为可能。下面是一些类型匹配的范例：

表 4.3: 通配符范例

模式	匹配对象
*	所有文件
g*	文件名以“g”开头的文件
b*.txt	以”b”开头，中间有零个或任意多个字符，并以”.txt”结尾的文件
Data???	以“Data”开头，其后紧接着 3 个字符的文件
[abc]*	文件名以”a”,”b”, 或”c”开头的文件
BACKUP.[0-9][0-9][0-9]	以”BACKUP.”开头，并紧接着 3 个数字的文件
[:upper:]*	以大写字母开头的文件
[![:digit:]]*	不以数字开头的文件

接下页



表 4.3 (续)

模式	匹配对象
*[[:lower:]]123]	文件名以小写字母结尾，或以“1”，“2”，或“3”结尾的文件

接受文件名作为参数的任何命令，都可以使用通配符，我们会在第八章更深入的谈到这个知识点。

字符范围

如果你用过别的类似 Unix 系统的操作环境，或者是读过这方面的书籍，你可能遇到过 [A-Z] 或 [a-z] 形式的字符范围表示法。这些都是传统的 Unix 表示法，并且在早期的 Linux 版本中仍有效。虽然它们仍然起作用，但是你必须小心地使用它们，因为它们不会产生你期望的输出结果，除非你合理地配置它们。从现在开始，你应该避免使用它们，并且用字符类来代替它们。

通配符在 GUI 中也有效

通配符非常重要，不仅因为它们经常用在命令行中，而且一些图形文件管理器也支持它们。

- 在 Nautilus (GNOME 文件管理器) 中，可以通过 Edit/Select 模式菜单项来选择文件。输入一个用通配符表示的文件选择模式后，那么当前所浏览的目录中，所匹配的文件名就会高亮显示。
- 在 Dolphin 和 Konqueror (KDE 文件管理器) 中，可以在地址栏中直接输入通配符。例如，如果你想查看目录 /usr/bin 中，所有以小写字母“u”开头的文件，在地址栏中敲入“/usr/bin/u\*”，则文件管理器会显示匹配的结果。

最初源于命令行界面中的想法，在图形界面中也适用。这就是使 Linux 桌面系统如此强大的众多原因中的一个。

4.2 mkdir - 创建文件夹

mkdir 命令是用来创建目录的。它这样工作：

```
mkdir directory ...
```

**注意表示法：**在描述一个命令时（如上所示），当有三个圆点跟在一个命令的参数后面，这意味着那个参数可以重复，就像这样：

```
mkdir dir1
```

会创建一个名为“dir1”的目录，而

```
mkdir dir1 dir2 dir3
```

会创建三个目录，名为“dir1”，“dir2”，“dir3”。



### 4.3 cp —复制文件和目录

cp 命令，复制文件或者目录。它有两种使用方法：

```
cp item1 item2
```

复制单个文件或目录” item1” 到文件或目录” item2”，和：

```
cp item... directory
```

复制多个项目（文件或目录）到一个目录下。

#### 有用的选项和实例

这里列举了 cp 命令一些有用的选项（短选项和等效的长选项）：

表 4.4: CP 选项

选项	意义
-a, -archive	复制文件和目录，以及它们的属性，包括所有权和权限。通常，复本具有用户所操作文件的默认属性。
-i, -interactive	在重写已存在文件之前，提示用户确认。如果这个选项不指定，cp 命令会默认重写文件。
-r, -recursive	递归地复制目录及目录中的内容。当复制目录时，需要这个选项（或者 -a 选项）。
-u, -update	当把文件从一个目录复制到另一个目录时，仅复制目标目录中不存在的文件，或者是文件内容新于目标目录中已经存在的文件。
-v, -verbose	显示翔实的命令操作信息

表 4.5: CP 实例

命令	运行结果
cp file1 file2	复制文件 file1 内容到文件 file2。如果 file2 已经存在，file2 的内容会被 file1 的内容重写。如果 file2 不存在，则会创建 file2。
cp -i file1 file2	这条命令和上面的命令一样，除了如果文件 file2 存在的话，在文件 file2 被重写之前，会提示用户确认信息。
cp file1 file2 dir1	复制文件 file1 和文件 file2 到目录 dir1。目录 dir1 必须存在。
cp dir1/* dir2	使用一个通配符，在目录 dir1 中的所有文件都被复制到目录 dir2 中。dir2 必须已经存在。

接下页

表 4.5 (续)

命令	运行结果
cp -r dir1 dir2	复制目录 dir1 中的内容到目录 dir2。如果目录 dir2 不存在，创建目录 dir2，操作完成后，目录 dir2 中的内容和 dir1 中的一样。如果目录 dir2 存在，则目录 dir1(和目录中的内容)将会被复制到 dir2 中。