

# *The Linux Command Line(中文版)*

一本值得阅读的 好书





引言 ii

    为什么使用命令行 . . . . . iii

    这本书讲什么 . . . . . iii

    谁应该读这本书 . . . . . iii

    这本书的内容 . . . . . iv

    怎样阅读这本书 . . . . . iv

    拓展阅读 . . . . . v

I 学习 Shell 1

1 什么是 shell 2

    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

2 文件系统中跳转 6

    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



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，只有大企业和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，我也许会说些过激的话。

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

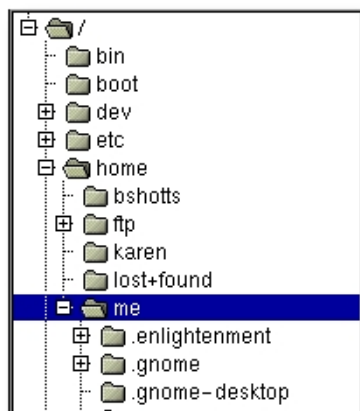


Figure 2.1: File system tree as shown by a graphical file manager

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。

Table 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。