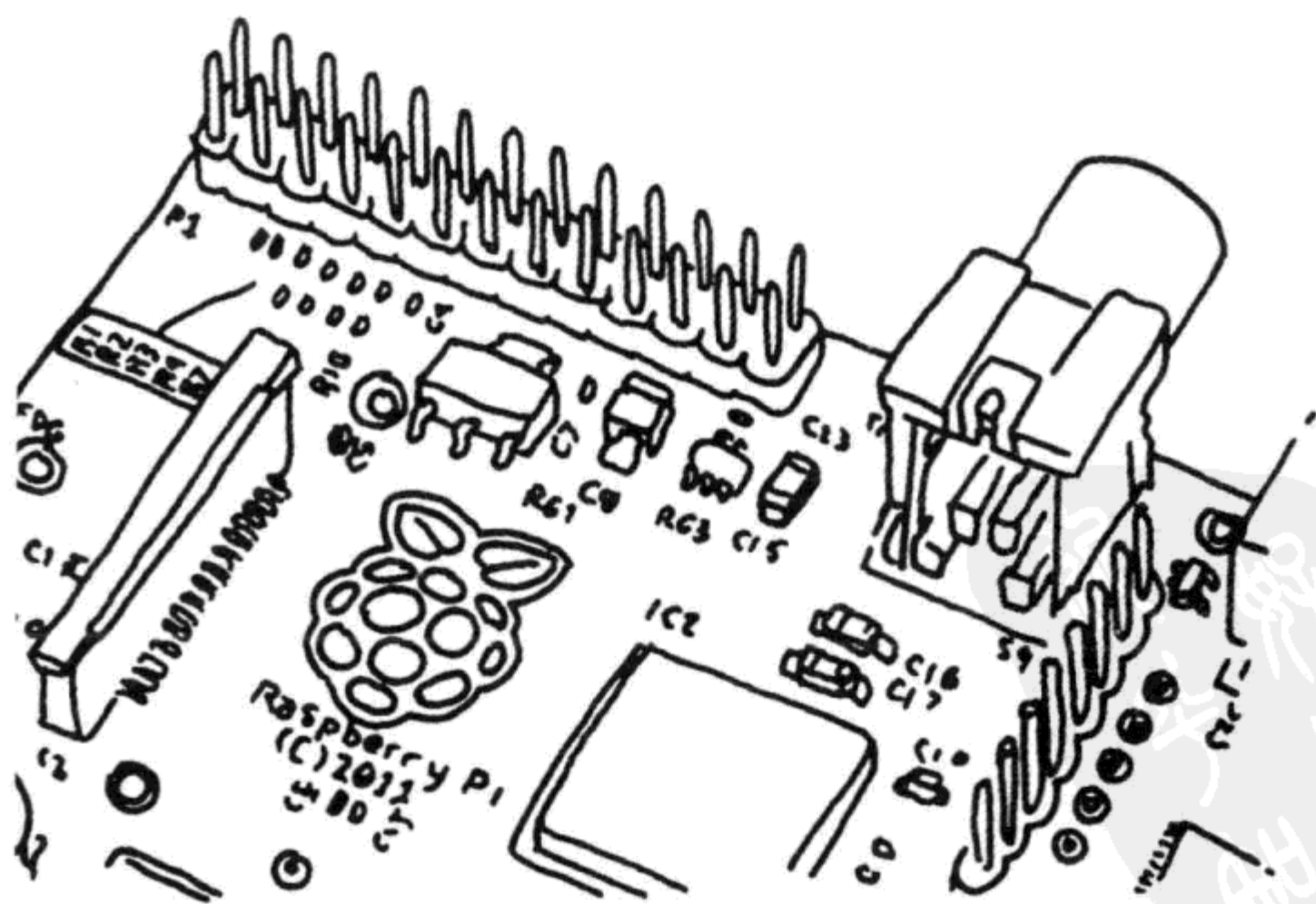


## 第 2 章

# 初识 Raspberry Pi 上的 Linux

Getting Around Linux on the  
Raspberry Pi





学习一点 Linux 的基本知识，可以帮助你从 Raspberry Pi 中获得更多的收获。本章的目标是带你快速领略 Linux 操作系统的基本知识，让你学会用基本的命令来操作文件系统、用命令或图形界面安装软件包，以及使用一些最常用的工具。

Raspbian 预装了名为“轻量级 X11 桌面环境”（Lightweight X11 Desktop Environment, LXDE）的图形化桌面环境（图 2.1）。这是一个基于 X Window 系统精简过的桌面系统，X Window 系统从 20 世纪 80 年代以来就是 UNIX 和 Linux 系统上最常用的图形界面。你在桌面或菜单中看到的部分软件是与 LXDE 绑定在一起的（如 Leafpad 文本编程器和 LX 终端）。

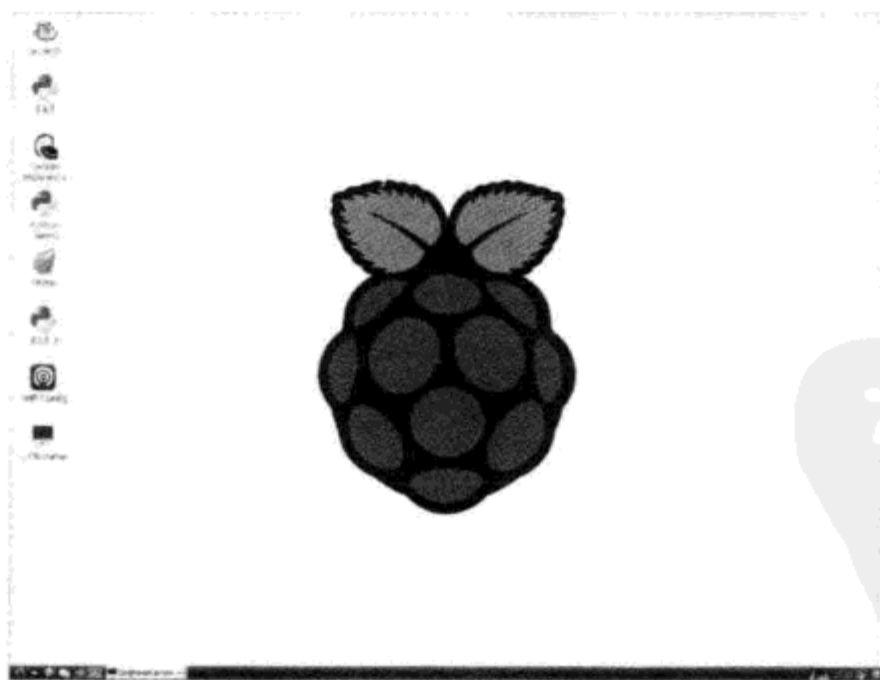


图2.1 图形化桌面

Openbox 窗口管理器在 LXDE 的基础上运行，它决定了窗口



和菜单的外观。如果你想调整桌面的外观，可以打开 Openbox 配置工具：点击屏幕左下角的桌面菜单，然后选择其他（Other）→ Openbox Configuration Manager。与 OS X 或 Windows 相比，在 Linux 下可以更容易自定义桌面环境或安装其他的窗口管理器。Raspberry Pi 的其他一些 Linux 发行版提供了不同的图形界面，有的适用于媒体中心，有的适用于电话系统，还有一些适用于网络防火墙。请参考 [http://elinux.org/RPi\\_Distributions](http://elinux.org/RPi_Distributions) 上的列表。

## 文件管理器 (File Manager)

如果不想用命令行来移动文件（稍后会介绍命令行的操作），从附件（Accessories）菜单中选择文件管理器（File Manager），然后就可以通过你所熟悉的文件图标和文件夹的形式来浏览文件系统了。

## 网络浏览器

默认的网络浏览器是 Midori (<http://twotoasts.de/index.php/midori/>)，这个浏览器只需占用很少的系统资源就可以运行。如今大家都已经熟悉了浏览器，便忽视了浏览器界面背后所发生的复杂的处理过程。由于 Raspbian 被设计成一个非常轻量级的操作系统，所以它自带的浏览器也缺少一些常见的功能：不支持 Flash 和 Java 插件（所以就看不 YouTube），也不支持基于 HTML5 的视频。后面我们会学习如何安装一些新的软件（如 Java）。可以在 Midori 窗口右上角的下拉菜单中找到常用的工具和菜单项（图 2.2）。你也可以安装一些其他的浏览器软件，如 NetSurf (<http://www.netsurf-browser.org/>) 或 Dillo (<http://www.dillo.org/>)。

## 视频与音频

截至本书写作时为止，多媒体播放器主要是 Omxplayer，它还



处在一个实验阶段。Omxplayer 是针对 Raspberry Pi 的处理器中所集成的图形处理单元（Graphics Processing Unit, GPU）而设计的，其他的自由软件（如 VLC 或 mPlayer）并不能与 Raspberry Pi 的 GPU 良好地配合工作。



图2.2 Web浏览器的下拉菜单



为了降低成本，Raspberry Pi 没有包含部分视频解码程序的许可证。如果你想观看 MPEG-2 格式（或微软的 VC-1 格式）录制的电视或 DVD，则需要从 Raspberry Pi 基金会的在线商店（<http://www.raspberrypi.com/>）购买一个许可证密钥。Raspberry Pi 自带了 H.264（MPEG-4）编解码器的许可证。

## 文本编辑器

Leafpad（<http://wiki.lxde.org/en/Leafpad>）是系统默认的编辑器，



可以在主菜单下找到。你也可以使用 Nano (<http://www.nano-editor.org/>)，这是一个文本界面的编辑器，非常易学易用。系统默认没有安装传统 UNIX 系统下的文本编辑器 Vim 和 Emacs，但它们都可以很方便地安装上（参考“安装新软件”）。

## 复制与粘贴

在应用程序之间复制、粘贴数据非常容易，不过也有一小部分古怪的软件不能完美地支持这个功能。如果鼠标上有中键，你可以像你所习惯的那样通过按住鼠标左键拖动来高亮选中一块文字实现复制，然后到目标窗口上点击鼠标中键把选中的文字粘贴过来。

## Shell 程序

很多任务需要你切换到命令行模式下，通过运行命令去实现。LX 终端（LXTerminal）程序提供了在图形界面下访问命令行终端的能力。Raspbian 上默认的 Shell 程序是 Linux 上最常见的 Bourne Again Shell（bash，<http://www.gnu.org/software/bash/manual/bashref.html>），也可以使用另一个名叫 Dash（[http://en.wikipedia.org/wiki/Debian\\_Almquist\\_shell](http://en.wikipedia.org/wiki/Debian_Almquist_shell)）的 Shell 程序。你可以通过程序菜单来切换 Shell 程序，也可以通过运行 `chsh` 命令来切换。

## 使用命令行

为了让学习使用命令行看上去更有意思一些，你可以把操作命令行想象成玩文字冒险游戏，游戏场景发生在文件与文件系统的迷宫中。如果你觉得这个想象一点也不吸引你，也没关系，本节中所介绍的命令和概念都是标准 Linux 系统所使用的，非常值得去学习。



在开始前，先打开一个 LX 终端（LXTerminal）（图 2.3）。在操作终端时，有两个基本技巧非常有用：自动补全与命令历史。通常情况下，你只需输入命令或文件名的前几个字符，然后按 Tab 键，终端就会尝试以当前目录中的文件或常用目录中的程序（Shell 通常会在 `/bin` 和 `/usr/bin` 目录搜索可执行程序）来补全你输入的字符串。如果你按键盘上的上下箭头键，可以调出以前执行过的命令。如果你不小心在命令中打错了字，可以用这个方法把错误的命令找出来修改后再重新执行。

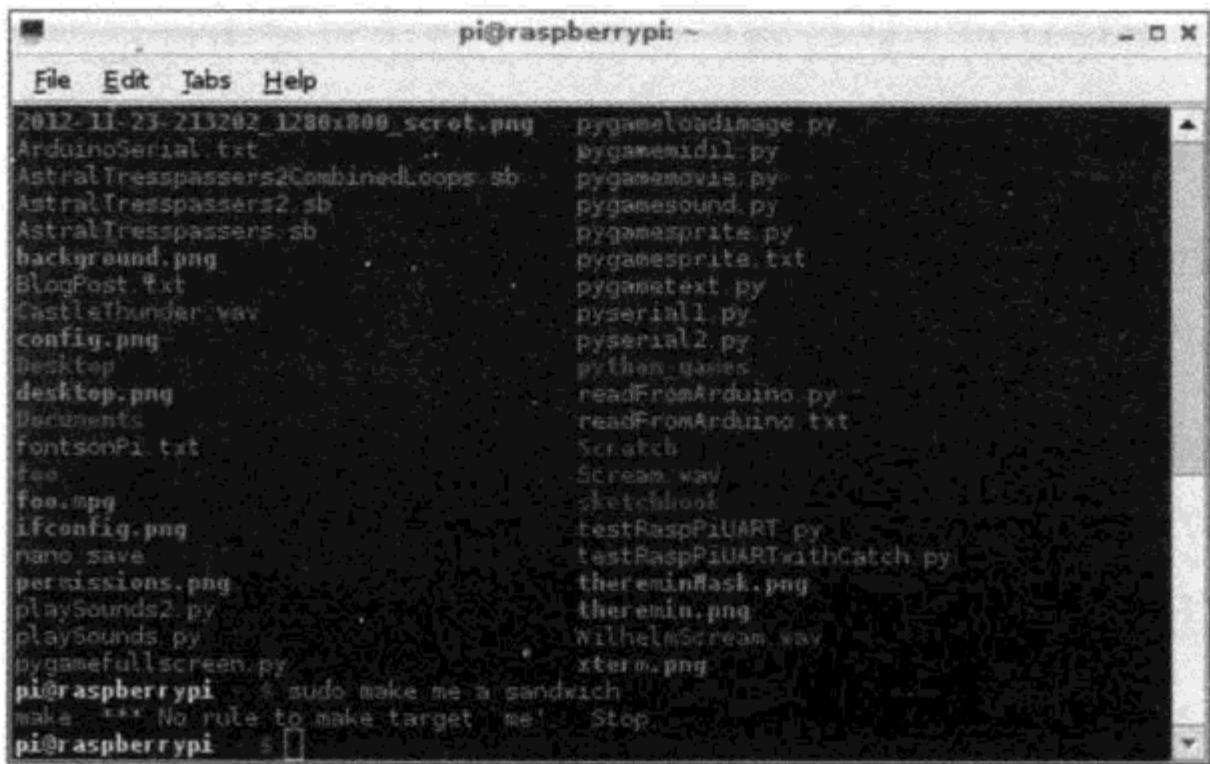


图2.3 通过LX终端（LXTerminal）访问命令行（Shell）

## 文件与文件系统

表 2.1 列出了文件系统中的重要目录，它们中的大部分都是按照 Linux 标准来设置的，标准决定了一些文件应该放在哪个目录中；其中有一部分则是 Raspberry Pi 所特有的。通过访问 `/sys` 目录中的文件，你可以操作 Raspberry Pi 上的所有硬件。





表2.1 Raspbian文件系统中部分重要的目录

目 录	描 述
/	
/bin	所有用户都可以运行的程序和命令
/boot	系统启动时所需要的文件
/dev	表示你系统上的各种设备的特殊文件
/etc	配置文件
/etc/init.d	启动各种服务的脚本
/etc/X11	X11 配置文件
/home	各用户的主目录
/home/pi	pi 用户的主目录
/lib	内核模块与驱动程序
/media	可移动存储的挂载点
/proc	包含操作系统和各个运行中的进程相关信息的虚拟目录
/sbin	用于系统管理与维护的程序
/sys	Raspberry Pi 上的一个特殊的目录，可用于操作设备
/tmp	用于让各个程序创建临时文件
/usr	所有用户都可以使用的程序和数据
/usr/bin	包含了操作系统中大部分的程序
/usr/games	游戏都在这里
/usr/lib	用于支持常用程序的库文件
/usr/local	存放一些这台机器专用的软件
/usr/sbin	更多系统管理程序
/usr/share	一些在多个程序间共享的数据，如图标与字体
/usr/src	Linux 是开放源代码的，这里就包含了它的源代码
/var	系统日志或脱机缓存文件
/var/backups	一些系统关键文件的备份
/var/cache	存放应用程序（如 apt-get 和网页浏览器）缓存文件的目录
/var/log	所有的系统及服务程序的日志
/var/mail	如果配置了电子邮件功能，所有用户的邮件都保存在这里
/var/spool	待处理的数据（如接收到的电子邮件、打印任务）



在命令提示符前面，可以看到你当前所处的工作目录。在Linux中，你的主目录会被缩写为波浪线（~）。刚刚打开 LX 终端（LXTerminal）时，它会把当前的工作目录切换到你的主目录下，命令提示符会显示为

```
pi@raspberrypi ~ $
```

有关命令提示符的详细解释：

```
pi@❶ raspberrypi❷ ~❸ $❹
```

- ❶ 你的用户名：pi，后面加上@符号。
- ❷ 你的主机名（默认的主机名是 raspberrypi）。
- ❸ Shell 的当前工作目录。默认是你的主目录（~）。
- ❹ 这是命令提示符。你输入的一切内容都会出现在它的右侧，按回车键执行你所输入的命令。

使用 cd（change directory，改变目录）命令可以在文件系统的各个目录间切换，下面的两条命令对于 pi 用户来说具有相同的效果（把当前工作目录切换到主目录）：

```
cd /home/pi/  
cd ~
```

如果传给 cd 命令的路径以斜线开头，则表示这是一个绝对路径；否则这个路径被认为是相对于当前工作目录位置的相对路径。可以用 . 和 .. 分别表示当前目录和上一层目录。例如，把当前目录切换到系统根目录：

```
pi@raspberrypi ~ $ cd ..  
pi@raspberrypi /home $ cd ..
```





你也可以直接用 `/` 切换到根目录：

```
pi@raspberrypi ~ $ cd /
```

切换完目录后，可以用 `ls` 命令列出当前目录下的文件：

```
pi@raspberrypi / $ ls
bin  dev  home  lost+found  mnt  proc  run   selinux  sys  usr
boot  etc  lib   media      opt  root  sbin  srv      tmp  var
```

大部分命令都允许添加一些参数或开关来改变默认的行为。例如，`ls` 命令的 `-l` 参数可以让显示的文件列表更为详尽，显示出文件的大小、修改时间和权限：

```
pi@raspberrypi ~ $ ls -l
total 8
drwxr-xr-x 2 pi pi 4096 Oct 12 14:26 Desktop
drwxrwxr-x 2 pi pi 4096 Jul 20 14:07 python_games
```

用 `-a` 参数可以列出所有文件，包含隐藏文件：

```
pi@raspberrypi ~ $ ls -la
total 80
drwxr-xr-x 11 pi pi 4096 Oct 12 14:26 .
drwxr-xr-x  3 root root 4096 Sep 18 07:48 ..
-rw-----  1 pi pi  25  Sep 18 09:22 .bash_history
-rw-r--r--  1 pi pi  220 Sep 18 07:48 .bash_logout
-rw-r--r--  1 pi pi 3243 Sep 18 07:48 .bashrc
drwxr-xr-x  6 pi pi 4096 Sep 19 01:19 .cache
drwxr-xr-x  9 pi pi 4096 Oct 12 12:57 .config
drwx-----  3 pi pi 4096 Sep 18 09:24 .dbus
drwxr-xr-x  2 pi pi 4096 Oct 12 14:26 Desktop
-rw-r--r--  1 pi pi  36  Sep 18 09:35 .dmrc
drwx-----  2 pi pi 4096 Sep 18 09:24 .gvfs
drwxr-xr-x  2 pi pi 4096 Oct 12 12:53 .idlerc
-rw-----  1 pi pi  35  Sep 18 12:11 .lessht
drwx-----  3 pi pi 4096 Sep 19 01:19 .local
```



```
-rw-r--r--  1 pi  pi  675   Sep 18 07:48 .profile
drwxrwxr-x  2 pi  pi 4096   Jul 20 14:07 python_games
drwx-----  4 pi  pi 4096   Oct 12 12:57 .thumbnails
-rw-----  1 pi  pi   56   Sep 18 09:35 .Xauthority
-rw-----  1 pi  pi  300   Oct 12 12:57 .xsession-errors
-rw-----  1 pi  pi 1391   Sep 18 09:35 .xsession-errors.old
```

使用 `mv` 命令可以改变文件的名字。使用 `touch` 命令可以用于创建一个新的空文件。

```
pi@raspberrypi ~ $ touch foo
pi@raspberrypi ~ $ ls
foo Desktop python_games
pi@raspberrypi ~ $ mv foo baz
pi@raspberrypi ~ $ ls
baz Desktop python_games
```

删除文件用 `rm` 命令。要删除空目录，可以使用 `rmdir` 命令。如果要删除非空目录，使用 `rm -r` 命令。传给 `rm` 命令的 `-r` 参数意味着要求 `rm` 程序递归进入所有的子目录删除所有的文件。

如果想了解一个命令所包含的所有参数，可以使用 `man` 命令（或在命令后面添加 `--help` 参数）：

```
pi@raspberrypi ~ $ man curl
pi@raspberrypi ~ $ rm --help
```

要创建新的目录，使用 `mkdir` 命令。要把一个目录下的所有文件打包在一起，用 `tar` 命令——这个名字来源于“磁带存档”（tape archives）的缩写。你常常能看到很多文件与源代码以 `tar` 包的形式发布，并且它们常常还会用 `gzip` 进行一次压缩。练习一下下面的操作：

```
pi@raspberrypi ~ $ mkdir myDir
pi@raspberrypi ~ $ cd myDir
```



```
pi@raspberrypi ~ $ touch foo bar baz
pi@raspberrypi ~ $ cd ..
pi@raspberrypi ~ $ tar -cf myDir.tar myDir
pi@raspberrypi ~ $ gzip myDir.tar
```

这时,你就把`myDir`目录打包成了一个名为`myDir.tar.gz`的文件,这个文件可以很方便地通过网络或电子邮件传递。

## 更多 Linux 命令

Linux (与 UNIX) 如此成功的原因之一在于它的设计哲学:通过一系列可以组装在一起的简单小模块来构建一个非常复杂的系统。所以,你需要学习一些有关管道和重定向的基础知识,它们是实现这个设计哲学的基础。

管道是把两个程序联系起来的一种方式,通过管道,可以把其中一个程序的输出作为另一个程序的输入。所有的 Linux 程序都可以从标准输入 (通常表示为 `stdin`) 读入数据,向标准输出 (表示为 `stdout`) 输出数据并把错误信息抛向标准错误输出 (表示为 `stderr`)。通过管道可以把一个程序的 `stdout` 与另一个程序的 `stdin` 相连 (图 2.4)。管道的操作号是 `|`, 如:

```
pi@raspberrypi ~ $ ls -la | less
```

(按 Q 键退出 `less` 程序)

还可以尝试一下通过管道传输更多的数据:

```
pi@raspberrypi ~ $ sudo cat /boot/kernel.img | aplay
```

运行这个命令前,建议先把音量调小一些。这个命令把系统内



核对应的 0 和 1 的数据传递给音乐播放器，当成音乐来播放。这就是“内核之声”。

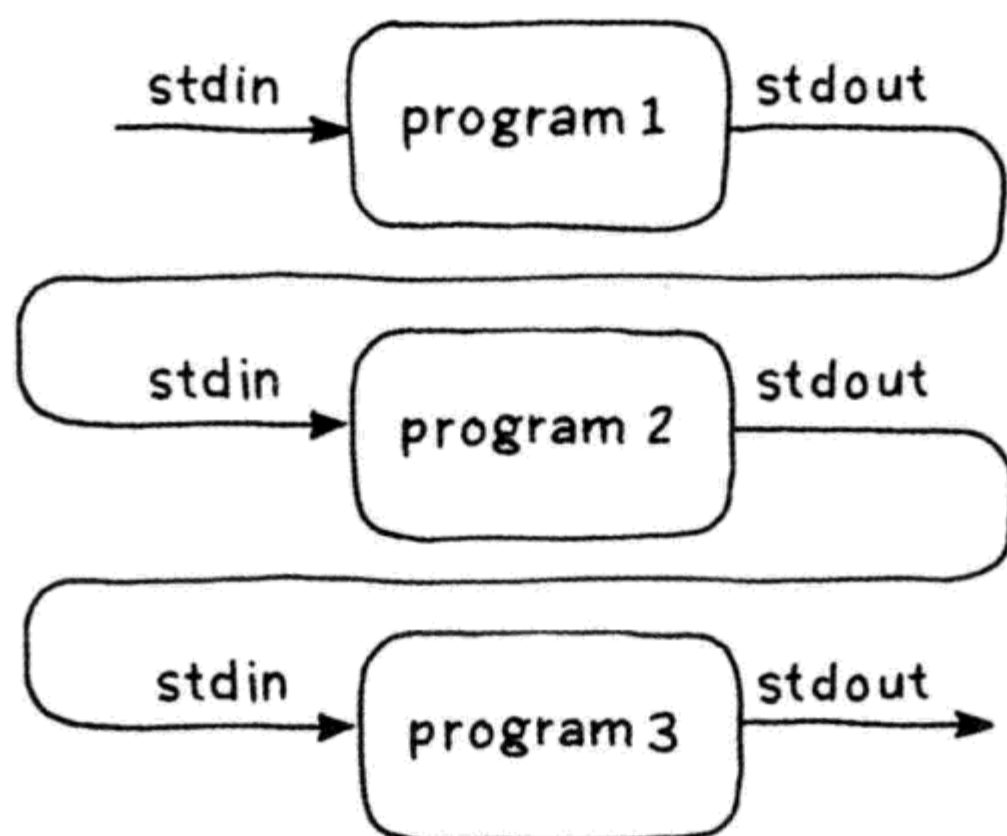


图2.4 管道可以用于把小程序连接在一起形成一个更大的任务

本书后面章节中的命令还会用到重定向，也就是指把程序的标准输出发送到一个文件中去。在后面你会看到，Linux 中很多的东西都可以被当成文件来使用（如 Pi 的通用输入输出口），所以重定向是一个非常有用的功能。使用 > 操作符来实现重定向：

```
pi@raspberrypi ~ $ ls > directoryListing.txt
```

## 特殊的控制键

除了先前提到的自动补全（Tab 键）和调用历史命令（光标向上键）的控制键，Shell 中还提供了其他一些有用的控制键。



### Control-C

中断当前正在运行的程序。在有些交互式应用程序（如文本编辑器）中可能无效。

### Control-D

退出 Shell。必须在命令提示符后紧跟着输入才有效（在 \$ 提示符后不要输入 Control-D 以外的其他任何字符）。

### Control-A

把光标移动到行首位置。

### Control-E

把光标移动到行末位置。

除此之外，还有很多可以使用的快捷键，不过上面列出的是日常操作中最为常用的。

需要在屏幕上显示一个文件的内容，并且要分屏显示时，可以使用 `less` 命令：

```
pi@raspberrypi ~ $ ls > flob.txt
pi@raspberrypi ~ $ less flob.txt
```

如果想把一个文件的内容一次性全部输出到标准输出上，可以用 `cat` 命令（concatenate 的缩写）。在需要把一个文件的内容传到另一个程序中或者把它重定向到其他地方时，这个命令非常有用。

例如，用 `cat` 命令可以非常方便地实现把一个文件复制为另一个文件的功能（第二个 `cat` 命令中，把两个文件连接成一个新的文件）。

```
pi@raspberrypi ~ $ ls > wibble.txt
pi@raspberrypi ~ $ cat wibble.txt > wobble.txt
pi@raspberrypi ~ $ cat wibble.txt wobble.txt > wubble.txt
```





用 **tail** 命令可以查看一个文件的最后几行内容（如一个日志文件中的最后几条），用 **head** 命令则可以查看前几行。如果要在一个或多个文件中搜索一个特定的字符串，可以用强大的 **grep** 命令：

```
pi@raspberrypi ~ $ grep Puzzle */*
```

由于 **grep** 命令能支持正则表达式丰富的语意，这帮助它成了一个非常强大的工具。正则表达式不太容易让人理解，这也是很多人认为 Linux 对初学者来说很难用的一个重要原因。

## 进 程

Pi 上运行的每一个程序都是一个独立的进程，所以任何时候总是有很多个进程同时在运行。当你刚把 Pi 启动完成时，系统会运行大约 75 个进程，每个进程负责一项任务或提供一个服务。可以用 **top** 命令查看正在运行的进程，与此同时也可以看到 CPU 和内存的使用情况。**top** 命令能让你看到占用资源最多的进程，**ps** 命令可以用来列出所有的进程和它们对应的进程 ID（PID）：

```
pi@raspberrypi ~ $ ps -aux | less
```

如果要中断一个出错或失去响应的进程，可以用 **ps** 命令找出这个进程的 ID，然后用 **kill** 命令来中断它的运行：

```
pi@raspberrypi ~ $ kill 5689
```

对于一些系统进程，你是没有权限去中断它的运行的，但可以通过稍后介绍的 **sudo** 命令来获得需要的权限。





## sudo 与权限

Linux 是一个多用户操作系统，一条最基本的原则就是每个用户可以在他们自己的目录下创建、删除或修改属于他们自己的文件。**root** 用户（管理员）可以修改文件系统中的任何文件，为了规避错误地对文件进行操作，日常使用时应该尽量避免使用 **root** 用户登录。



如果以 **pi** 用户登录，你很难对系统造成什么致命性的破坏。但以管理员身份操作则可能对系统造成严重的破坏，这也许是不小心的错误操作引起的，也有可能是故意的。所以，在使用 **sudo** 时应该特别小心，尤其是移动或删除文件时，更应慎之又慎。当然，如果你确实已经把系统破坏了，也还是可以通过参考附录 A 中的方法重新在 SD 卡上安装一个新的系统。

通过使用 **sudo** 或类似的工具可以使普通用户临时获得管理员权限，用来执行安装软件包之类的操作，这样就可以规避直接以 **root** 用户登录所带来的安全风险。在修改系统配置或与硬件打交道时，你常常会使用到 **sudo**。

系统中的每一个文件都属于某一个用户和某一个用户组，用 **chown** 和 **chgrp** 命令可以修改文件所属的用户或用户组。要执行这两个命令，你必须拥有 **root** 权限。

```
pi@raspberrypi ~ $ sudo chown pi garply.txt
pi@raspberrypi ~ $ sudo chgrp staff plugh.txt
```

每个文件都被设置了相关的权限，决定了一个文件是否可以被



读、写或执行。这些权限可以按文件的拥有者、文件所属的用户组或全部用户分别设置（图 2.5）。

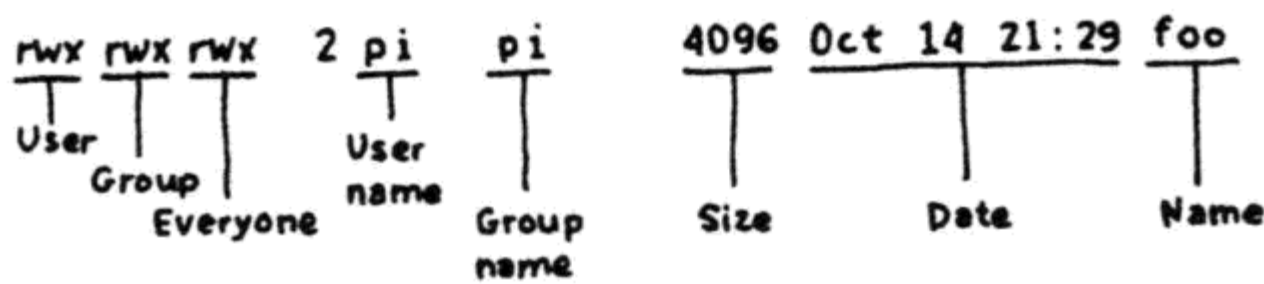


图2.5 对文件所有者、文件所属组和全部用户的权限设置

用 `chmod` 命令可以设置文件的权限。相关的选项如表 2.2 所示。

表2.2 `chmod`命令中使用的选项

u	用户
g	用户组
o	所属用户组以外的用户
a	所有用户
r	读权限
w	写权限
x	执行权限
+	添加权限
-	去除权限

下面是把这些选项组合在一起使用的一些例子：

```
chmod u+rwx,g-rwx,o-rwx wibble.txt❶
chmod g+wx wobble.txt❷
chmod -rw,+r wubble.txt❸
```

❶ 只允许文件所有者读、写和执行。



② 允许文件所属用户组的用户写和执行。

③ 该文件对所有用户都是只读。

密码是保护你的个人空间与文件的唯一安全措施，所以你应该设置较为复杂的密码，尤其是把 Pi 接入网络的情况下。可以用 `passwd` 命令来修改密码。

## 网 络

如果你的 Pi 已经接入网络，就会经常用到一些网络相关的 Linux 工具。如果你需要诊断网络连接的故障，可以用 `ifconfig` 命令，这个命令可以用于显示你所有的网络接口以及它们上面所绑定的 IP 地址（图 2.6）。

```
pi@raspberrypi ~ $ ifconfig
eth0      Link encap:Ethernet  HWaddr b8:27:eb:de:7a:b6
          inet addr:192.168.2.2  Bcast:192.168.2.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:7632 errors:0 dropped:0 overruns:0 frame:0
          TX packets:7050 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:558086 (545.0 KiB)  TX bytes:1680110 (1.6 MiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:14 errors:0 dropped:0 overruns:0 frame:0
          TX packets:14 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:1432 (1.3 KiB)  TX bytes:1432 (1.3 KiB)
```

图2.6 用`ifconfig`命令可以显示出网卡的所有信息

在诊断网络连接问题时，`ping` 命令是最基本的工具。你可以使用 `ping` 命令（像声呐那样）来检测 Internet 或局域网中两个 IP 地址之间是否可以相互连通。不过要注意，有些网站会屏蔽 `ping` 的检测流量，所以你也可能需要多 `ping` 几个网站才能确定网络是否确实存在问题。



```
ping yahoo.com  
ping altavista.com  
ping lycos.com  
ping netscape.com ❶
```

❶ 这个 ping 命令会失败。

Secure Shell (SSH) 可以用于安全地（会被加密）远程登录另一台电脑，前提是远程电脑上启动了 SSH 服务。Raspbian 中已经预装了 SSH 客户端程序。通过 SSH 远程访问 Raspberry Pi 就不需要在 Pi 上连接显示器和键盘了，这就是我们稍后会介绍的“无显示器运行”方式。

与 SSH 相关的还有 `sftp` 程序，可以用于两台电脑之间安全地传输文件。除此之外还有 `scp` 命令，可以通过局域网或 Internet 在两台电脑之间复制文件。这些工具都是基于安全套接字层（Secure Sockets Layer, SSL）协议来开发的，可以保证密码和传输过程的安全。这些都是标准的 Linux 工具。

## 无显示器运行

如果你想在不用连接显示器、键盘和鼠标的情况下使用 Raspberry Pi：即所谓“无显示器运行”，是完全可行的。如果只是需要使用命令行终端，你只需把 Raspberry Pi 接上网线，就可以用 SSH 客户端程序去连接它了（默认的用户名为 `pi`，密码为 `raspberry`）。在 Mac 上，可以使用 Terminal 工具，Windows 和 Linux 上可以使用 PuTTY 工具（<http://www.chiark.greenend.org.uk/~sgtatham/putty/>），Linux 上也可以使用 `ssh` 命令。Raspberry Pi 上的 SSH 服务默认就是启用的（需要的话可以通过运行 `raspi-config` 命令把它设置为开机



默认不启动)。

另外一种通过网络使用 Raspberry Pi 的方式是使用 VNC (虚拟网络计算, Virtual Network Computing) 服务器。使用 VNC 的好处在于,你可以在你的笔记本或台式电脑上操作 Raspberry Pi 上的图形界面。这种方式可以很好地作为便携开发环境的解决方案。参考 Raspberry Pi Hub 网页 ([http://elinux.org/RPi\\_VNC\\_Server](http://elinux.org/RPi_VNC_Server)) 的介绍设置 TightVNC (<http://www.tightvnc.com/>), TightVNC 是一个轻量级的 VNC 服务器。

## /etc 目录

/etc 目录中存放了所有的系统全局配置文件和启动脚本,在你第一次启动 Raspberry Pi 并通过配置工具修改系统配置时,实际上就是修改了 /etc 目录下的文件。你需要通过 `sudo` 获取管理员权限才能修改 /etc 目录下的文件,所以如果看到某些教程上提到要修改配置文件,就需要通过 `sudo` 启动一个文本编辑器来修改这些文件:

```
pi@raspberrypi ~ $ sudo nano /etc/hosts
```

## 设置日期时间

一般的笔记本电脑或台式电脑会有额外的硬件和备用电池 (通常是一个纽扣电池) 用于记录当前的日期时间, Raspberry Pi 没有。但是 Raspbian 系统被配置为一旦接入网络就自动通过网络时间协议 (Network Time Protocol, NTP) 来同步系统日期和时间。

把系统时间设置为正确的值对于一些软件来说非常重要 (如第 7 章中提到的用 `cron` 服务来做一个定时台灯)。如果要手工设置日期时间,可以用 `date` 程序:



```
$ sudo date --set="Sun Nov 18 1:55:16 EDT 2012"①
```

## 安装新软件

强大的软件包管理是 Linux 完胜于其他操作系统的原因之一。包管理程序负责下载和安装新的软件，并自动解决软件之间的依赖关系。为了让系统做到模块化，Linux 下的很多软件包在安装时都会依赖其他的软件包。包管理程序让这种依赖关系很容易得到自动解决，并且包管理程序自身也非常可靠。

Raspbian 默认只安装了很少的一些软件，所以你很快就会遇到需要下载和安装软件的问题。在本书的示例中，我们使用命令行来安装软件，因为这是最灵活也是最便捷的方法。

`apt-get` 程序带上 `install` 参数是用于下载软件包的，`apt-get` 会自动下载安装你想要的软件所必需依赖的其他软件包，这样你就无须人工去到处寻找那些依赖包了。安装软件需要系统管理员权限，所以运行 `apt-get` 时总是需要带上 `sudo` 前缀（下面的命令用于安装 Emacs 编辑器）：

```
pi@raspberrypi ~ $ sudo apt-get install emacs
```

---

### 屏幕截图



在我们写这本书时，非常需要一个 Pi 上的屏幕截图工具。我们找到了一个名叫 `scrot`（是 `SCREenshOT` 的缩写）的工具。当然，你也可以安装 GNU Image Manipulation Program（GIMP）或 ImageMagick 来获取屏幕截图，不过

---

<sup>①</sup> 这条命令中的 EDT 是 Eastern Daylight Time 的缩写，在美国等国家表示使用东部夏令时。当系统时区设置为中国所对应的时区时，可以使用中国标准时间（China Standard Time）的缩写 CST 来替换它。——译者注





我们用 scrot 已经足够。安装 scrot 的命令：

```
sudo apt-get install scrot
```

---

## 进一步学习

通过很多渠道可以学习到更多的 Linux 相关知识，你可以尝试从以下地方开始。

### *Linux Pocket Guide*

( <http://shop.oreilly.com/product/0636920023029.do> )

非常有用的快速参考书。

### *Linux in a Nutshell*

( <http://shop.oreilly.com/product/9780596154493.do> )

比上一本要详细一些，但仍然是一本快速参考手册。

### *The Debian Wiki*

( <http://wiki.debian.org/FrontPage> )

Raspbian 是基于 Debian 开发的，所以 Debian Wiki 上的很多信息对 Raspbian 同样适用。

### *The Jargon File*

( <http://catb.org/jargon/> )

曾经以 *New Hacker's Dictionary* 的书名出版，里面收录了值得了解的一些定义和故事，属于 UNIX/Linux 的亚文化。