第四章探究操作系统

Now that we know how to move around the file system, it’s time for a guided tour of our Linux system. Before we start however, we’re going to learn some more commands that will be useful along the way:

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。然而在开始之前，我们先学习一些对研究 Linux 系统有帮助的命令。

* ls – List directory contents
* file – Determine file type
* less – View file contents
* ls — 列出目录内容
* file — 确定文件类型
* less — 浏览文件内容

**ls 乐趣**

The ls command is probably the most used command, and for good reason. With it, we can see directory contents and determine a variety of important file and directory attributes. As we have seen, we can simply type ls to see a list of files and subdirectories contained in the current working directory:

ls 可能是用户最常使用的命令了，这自有它的道理。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。正如我们已经见到的，只要简单的输入 ls 就能看到在当前目录下所有文件和子目录的列表。

[me@linuxbox ~]$ ls

Desktop Documents Music Pictures Publica Templates Videos

Besides the current working directory, we can specify the directory to list, like so:

除了当前工作目录以外，也可以指定别的目录，就像这样：

me@linuxbox ~]$ ls /usr

bin games kerberos libexec sbin src

etc include lib local share tmp

Or even specify multiple directories. In this example we will list both the user’s home directory (symbolized by the “~” character) and the /usr directory:

甚至可以列出多个指定目录的内容。在这个例子中，将会列出用户家目录（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

We can also change the format of the output to reveal more detail:

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

By adding “-l” to the command, we changed the output to the long format.

使用 ls 命令的“-l”选项，则结果以长模式输出。

**选项和参数**

This brings us to a very important point about how most commands work. Commands are often followed by one or more options that modify their behavior, and further, by one or more arguments, the items upon which the command acts. So most commands look kind of like this:

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

Most commands use options consisting of a single character preceded by a dash, for example, “-l”, but many commands, including those from the GNU Project, also support long options, consisting of a word preceded by two dashes. Also, many commands allow multiple short options to be strung together. In this example, the ls command is given two options, the “l” option to produce long format output, and the “t” option to sort the result by the file’s modification time.

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

We’ll add the long option “–reverse” to reverse the order of the sort:

加上长选项 “–reverse”，则结果会以相反的顺序输出：

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

The ls command has a large number of possible options. The most common are listed in the Table 4-1.

ls 命令有大量的选项。表4-1列出了最常使用的选项。

|  |  |  |
| --- | --- | --- |
| *Table 4-1: Common ls Options* | | |
| Option | Long Option | Description |
| -a | --all | List all files, even those with names that begin with a period, which are normally not listed(i.e.,hidden). |
| -d | --directory | Ordinaryly,if a directory is specified, ls will list the contents of the directory, not the directory itself. Use this option in conjunction with the -l option to see details about the directory rather than its contents. |
| -F | --classify | This option will append an indicator character to the end of each listed name. For example, a '/' if the name is a directory. |
| -h | --human-readable | In long format listings, display file sizes in human readable format rather than in bytes. |
| -l |  | Display results in long format. |
| -r | --reverse | Display the results in reverse order. Normally, ls display its results in ascending alphabetical order. |
| -S |  | Sort results by file size. |
| -t |  | Sort by modification time. |

|  |  |  |
| --- | --- | --- |
| *表 4-1: 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默认会被隐藏的隐藏文件。 |
| -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 | 当以长格式列出时，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**

As we saw before, the “-l” option causes ls to display its results in long format. This format contains a great deal of useful information. Here is the Examples directory from an Ubuntu system:

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：

-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

-rw-r--r-- 1 root root 159744 2007-04-03 11:05 oo-derivatives.doc

-rw-r--r-- 1 root root 27837 2007-04-03 11:05 oo-maxwell.odt

-rw-r--r-- 1 root root 98816 2007-04-03 11:05 oo-trig.xls

-rw-r--r-- 1 root root 453764 2007-04-03 11:05 oo-welcome.odt

-rw-r--r-- 1 root root 358374 2007-04-03 11:05 ubuntu Sax.ogg

Let’s look at the different fields from one of the files and examine their meanings:

选一个文件，来看一下各个输出字段的含义：

|  |  |
| --- | --- |
| *Table 4-2: ls Long Listing Fields* | |
| Field | Meaning |
| -rw-r--r-- | Access rights to the file. The first character indicates the type of file. Among the different types, a leading dash means a regular file, while a “d” indicates a directory. The next three characters are the access rights for the file's owner, the next three are for members of the file's group, and the final three are for everyone else. The full meaning of this is discussed in Chapter 10 – Permissions. |
| 1 | File's number of hard links. See the discussion of links later in this chapter. |
| root | The user name of the file's owner. |
| root | The name of the group which owns the file. |
| 32059 | Size of the file in bytes. |
| 2007-04-03 11:05 | Date and time of the file's last modification. |
| oo-cd-cover.odf | Name of the file. |

|  |  |
| --- | --- |
| *表 4-2: 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 | 文件属主的用户名。 |
| 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。 |

**确定文件类型**

As we explore the system it will be useful to know what files contain. To do this we will use the file command to determine a file’s type. As we discussed earlier, filenames in Linux are not required to reflect a file’s contents. While a filename like “picture.jpg” would normally be expected to contain a JPEG compressed image, it is not required to in Linux. We can invoke the file command this way:

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 系统中，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

When invoked, the file command will print a brief description of the file’s contents. For example:

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

There are many kinds of files. In fact, one of the common ideas in Unix-like operating systems such as Linux is that “everything is a file.” As we proceed with our lessons, we will see just how true that statement is.

有许多种类型的文件。事实上，在类 Unix 操作系统中比如说 Linux 中，有个普遍的观念就是“一切皆文件”。 随着课程的进行，我们将会明白这句话是多么的正确。

While many of the files on your system are familiar, for example MP3 and JPEG, there are many kinds that are a little less obvious and a few that are quite strange.

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。

**用 less 浏览文件内容**

The less command is a program to view text files. Throughout our Linux system, there are many files that contain human-readable text. The less program provides a convenient way to examine them.

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。

What Is “Text”

什么是“文本”

There are many ways to represent information on a computer. All methods involve defining a relationship between the information and some numbers that will be used to represent it. Computers, after all, only understand numbers and all data is converted to numeric representation.

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 用来代表信息。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来表示。

Some of these representation systems are very complex (such as compressed video files), while others are rather simple. One of the earliest and simplest is called ASCII text. ASCII (pronounced “As-Key”) is short for American Standard Code for Information Interchange. This is a simple encoding scheme that was first used on Teletype machines to map keyboard characters to numbers.

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。

Text is a simple one-to-one mapping of characters to numbers. It is very compact. Fifty characters of text translates to fifty bytes of data. It is important to understand that text only contains a simple mapping of characters to numbers. It is not the same as a word processor document such as one created by Microsoft Word or OpenOffice.org Writer. Those files, in contrast to simple ASCII text, contain many non-text elements that are used to describe its structure and formatting. Plain ASCII text files contain only the characters themselves and a few rudimentary control codes like tabs, carriage returns and line feeds. Throughout a Linux system, many files are stored in text format and there are many Linux tools that work with text files. Even Windows recognizes the importance of this format. The well-known NOTEPAD.EXE program is an editor for plain ASCII text files.

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。

Why would we want to examine text files? Because many of the files that contain system settings (called configuration files) are stored in this format, and being able to read them gives us insight about how the system works. In addition, many of the actual programs that the system uses (called scripts) are stored in this format. In later chapters, we will learn how to edit text files in order to modify systems settings and write our own scripts, but for now we will just look at their contents.

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。

The less command is used like this:

less 命令是这样使用的：

less filename

Once started, the less program allows you to scroll forward and backward through a text file. For example, to examine the file that defines all the system’s user accounts, enter the following command:

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

Once the less program starts, we may view the contents of the file. If the file is longer than one page, we can scroll up and down. To exit less, press the “q” key. The table below lists the most common keyboard commands used by less.

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。

|  |  |
| --- | --- |
| *Table 4-3: less Commands* | |
| Command | Action |
| Page UP or b | Scroll back one page |
| Page Down or space | Scroll forward one page |
| UP Arrow | Scroll Up one line |
| Down Arrow | Scrow Down one line |
| G | Move to the end of the text file |
| 1G or g | Move to the beginning of the text file |
| /charaters | Search forward for the next occurrence of characters |
| n | Search forward for the next occurrence of the previous search |
| h | Display help screen |
| q | Quit less |

|  |  |
| --- | --- |
| *表 4-3: less 命令* | |
| 命令 | 行为 |
| Page UP or b | 向上翻滚一页 |
| Page Down or space | 向下翻滚一页 |
| UP Arrow | 向上翻滚一行 |
| Down Arrow | 向下翻滚一行 |
| G | 移动到最后一行 |
| 1G or g | 移动到开头一行 |
| /charaters | 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）**

The less program was designed as an improved replacement of an earlier Unix program called more. The name “less” is a play on the phrase “less is more”—a motto of modernist architects and designers.

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 falls into the class of programs called “pagers,” programs that allow the easy viewing of long text documents in a page by page manner. Whereas the more program could only page forward, the less program allows paging both forward and backward and has many other features as well.

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。

**旅行指南**

The file system layout on your Linux system is much like that found on other Unix-like systems. The design is actually specified in a published standard called the Linux Filesystem Hierarchy Standard. Not all Linux distributions conform to the standard exactly but most come pretty close.

Linux 系统中，文件系统布局与类 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。

Next, we are going to wander around the file system ourselves to see what makes our Linux system tick. This will give you a chance to practice your navigation skills. One of the things we will discover is that many of the interesting files are in plain human- readable text. As we go about our tour, try the following:

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 into a given directory
2. List the directory contents with ls -l
3. If you see an interesting file, determine its contents with file
4. If it looks like it might be text, try viewing it with less
5. cd 到给定目录
6. 列出目录内容 ls -l
7. 如果看到一个有趣的文件，用 file 命令确定文件内容
8. 如果文件看起来像文本，试着用 less 命令浏览它

Remember the copy and paste trick! If you are using a mouse, you can double click on a filename to copy it and middle click to paste it into commands.

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。

As we wander around, don’t be afraid to look at stuff. Regular users are largely prohibited from messing things up. That’s the system administrators job! If a command complains about something, just move on to something else. Spend some time looking around. The system is ours to explore. Remember, in Linux, there are no secrets! Table 4-4 lists just a few of the directories we can explore. Feel free to try more!

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4-4仅仅列出了一些我们可以浏览的目录。闲暇时试试看！

|  |  |
| --- | --- |
| *Table 4-4: Directories Found On Linux Systems* | |
| Drectory | Comments |
| / | The root directory.Where everything begins. |
| /bin | Contains binaries (programs) that must be present for the system to boot and run. |
| /boot | Contains the linux kernel, intial RAM disk image (for drivers needed at boot time), and the boot loader.  Interesting files:   * /boot/grub/grub.conf or menu.lst, which are used to configure the boot loader. * /boot/vmlinuz,the linux kernel. |
| /dev | This is a special directory which contains device nodes. "Everything is a file" also applies to devices. Here is where the kernel maintains a list of all the devices it understands. |
| /etc | The /etc directory contains all of the system-wide configuration files. It also contains a collection of shell scripts which start each of the system services at boot time. Everything in this directory should be readable text.  Interesting files:While everything in /etc is interesting, here are some of my all-time favorites:   * /etc/crontab, a file that defines when automated jobs will run. * /etc/fstab, a table of storage devices and their associated mount points. * /etc/passwd, a list of the user accounts. |
| /home | In normal configurations, each user is given a directory in /home. Ordinary users can only write files in their home directories. This limitation protects the system from errant user activity. |
| /lib | Contains shared library files used by the core system programs. These are similar to DLLs in Windows. |
| /lost+found | Each formatted partition or device using a Linux file system, such as ext3, will have this directory. It is used in the case of a partial recovery from a file system corruption event. Unless something really bad has happened to your system, this directory will remain empty. |
| /media | On modern Linux systems the /media directory will contain the mount points for removable media such USB drives, CD-ROMs, etc. that are mounted automatically at insertion. |
| /mnt | On older Linux systems, the /mnt directory contains mount points for removable devices that have been mounted manually. |
| /opt | The /opt directory is used to install “optional” software. This is mainly used to hold commercial software products that may be installed on your system. |
| /proc | The /proc directory is special. It's not a real file system in the sense of files stored on your hard drive. Rather, it is a virtual file system maintained by the Linux kernel. The “files” it contains are peepholes into the kernel itself. The files are readable and will give you a picture of how the kernel sees your computer. |
| /root | This is the home directory for the root account. |
| /sbin | This directory contains “system” binaries. These are programs that perform vital system tasks that are generally reserved for the superuser. |
| /tmp | The /tmp directory is intended for storage of temporary, transient files created by various programs. Some configurations cause this directory to be emptied each time the system is rebooted. |
| /usr | The /usr directory tree is likely the largest one on a Linux system. It contains all the programs and support files used by regular users. |
| /usr/bin | /usr/bin contains the executable programs installed by your Linux distribution. It is not uncommon for this directory to hold thousands of programs. |
| /usr/lib | The shared libraries for the programs in /usr/bin. |
| /usr/local | The /usr/local tree is where programs that are not included with your distribution but are intended for system- wide use are installed. Programs compiled from source code are normally installed in /usr/local/bin. On a newly installed Linux system, this tree exists, but it will be empty until the system administrator puts something in it. |
| /usr/sbin | Contains more system administration programs. |
| /usr/share | /usr/share contains all the shared data used by programs in /usr/bin. This includes things like default configuration files, icons, screen backgrounds, sound files, etc. |
| /usr/share/doc | Most packages installed on the system will include some kind of documentation. In /usr/share/doc, we will find documentation files organized by package. |
| /var | With the exception of /tmp and /home, the directories we have looked at so far remain relatively static, that is, their contents don't change. The /var directory tree is where data that is likely to change is stored. Various databases, spool files, user mail, etc. are located here. |
| /var/log | /var/log contains log files, records of various system activity. These are very important and should be monitored from time to time. The most useful one is /var/log/messages. Note that for security reasons on some systems, you must be the superuser to view log files. |

|  |  |
| --- | --- |
| *表 4-4: Linux 系统中的目录* | |
| 目录 | 评论 |
| / | 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 RAM 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：   * /boot/grub/grub.conf or menu.lst， 被用来配置启动加载程序。 * /boot/vmlinuz，Linux 内核。 |
| /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：   * /etc/crontab， 定义自动运行的任务。 * /etc/fstab，包含存储设备的列表，以及与他们相关的挂载点。 * /etc/passwd，包含用户帐号列表。 |
| /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。 |
| /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 帐户的家目录。 |
| /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，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。 |

**符号链接**

As we look around, we are likely to see a directory listing with an entry like this:

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

Notice how the first letter of the listing is “l” and the entry seems to have two filenames? This is a special kind of a file called a symbolic link (also known as a soft link or symlink.) In most Unix-like systems it is possible to have a file referenced by multiple names. While the value of this may not be obvious, it is really a useful feature.

注意看，为何这条信息第一个字符是“l”，并且有两个文件名呢？ 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 Unix” 系统中， 有可能一个文件被多个文件名所指向。虽然这种特性的意义并不明显，但它真地很有用。

Picture this scenario: a program requires the use of a shared resource of some kind contained in a file named “foo,” but “foo” has frequent version changes. It would be good to include the version number in the filename so the administrator or other interested party could see what version of “foo” is installed. This presents a problem. If we change the name of the shared resource, we have to track down every program that might use it and change it to look for a new resource name every time a new version of the resource is installed. That doesn’t sound like fun at all.

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。

Here is where symbolic links save the day. Let’s say we install version 2.6 of “foo,” which has the filename “foo-2.6” and then create a symbolic link simply called “foo” that points to “foo-2.6.” This means that when a program opens the file “foo”, it is actually opening the file “foo-2.6”. Now everybody is happy. The programs that rely on “foo” can find it and we can still see what actual version is installed. When it is time to upgrade to “foo-2.7,” we just add the file to our system, delete the symbolic link “foo” and create a new one that points to the new version. Not only does this solve the problem of the version upgrade, but it also allows us to keep both versions on our machine. Imagine that “foo-2.7” has a bug (damn those developers!) and we need to revert to the old version. Again, we just delete the symbolic link pointing to the new version and create a new symbolic link pointing to the old version.

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。

The directory listing above (from the /lib directory of a Fedora system) shows a symbolic link called “libc.so.6” that points to a shared library file called “libc-2.6.so.” This means that programs looking for “libc.so.6” will actually get the file “libc-2.6.so.” We will learn how to create symbolic links in the next chapter.

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”。 在下一章节，我们将学习如何建立符号链接。

**硬链接**

While we are on the subject of links, we need to mention that there is a second type of link called a hard link. Hard links also allow files to have multiple names, but they do it in a different way. We’ll talk more about the differences between symbolic and hard links in the next chapter.

讨论到链接问题，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。