Linux 入门

本文翻译自“Introduction to Linux, A Hands on Guide”，作者 Machtelt Garrels。该书的版权信息如下：

第一章 什么是 LINUX

1.1. Linux 历史

1.1.1. UNIX

为了理解 Linux 的人气，我们要回到过去，大约 30 年前。

想象计算机有房子，甚至体育场馆那么大。这些计算机的尺寸已经是一个很大的问题了，但是另外一件事使得情况更加糟糕：每一台计算机都有自己的操作系统。软件总是为了某个目的而定制的，并且一个系统的软件不能在另外一个系统上面运行。你能在一个系统里工作并不意味着你就一定能在另外一个系统里工作。对于用户和系统管理员来说都是很困难的。

那个时候，计算机非常的昂贵，并且在购买了计算机后，为了让用户明白计算机是怎么工作的，还要花很多钱。每个计算能力的总成本非常的庞大。

那个时候技术还没有那么先进，所以人们在接下来的 10 年中，只能用这种体积巨大的计算机。1969 年，Bell 实验室的一群软件开发人员开始想办法解决软件的兼容性问题。他们开发了一个新的操作系统。这个操作系统有这样 3 个特点：
简单而精致
使用 C 语言而不是汇编语言
可以重复利用代码
他们把这个项目叫做“UNIX”。

可以重复利用代码是一个非常重要的特性。在那之前，所有的商业计算机系统都是用某个计算机的特定的代码来编写的。而 UNIX 只需要很少的那种特定的代码，也就是所谓的内核。这个内核是唯一的需要针对每个特定的系统而编写的，它是 UNIX 的基础。操作系统和所有其他的功能都是以这个内核为基础，用 C 语言来编写。C 语言是为了建立 UNIX 系统而设计的。使用这个新技术，开发一个能够在很多不同的硬件上面运行的操作系统就容易多了。

软加开发商马上就跟进了，因为他们可以毫不费力的卖出十倍的软件。这样出现了一个新的现象：想象一下，比如说不同厂商的计算机可以在同一个网络中相互通讯，或者不同计算机系统的用户不用额外的培训就能使用不同的计算机。UNIX 让用户能够使用不同的系统了。

UNIX 在接下来的几十年里一直在发展。更加多的事情变得可能，而且更加多的硬件和软件厂商为他们的产品提供了对 UNIX 的支持。

UNIX 一开始只用在非常大的机构里的大型机和小型机上（注意一台 PC 是微型计算机）。你要在大学，政府或者大的金融机构工作才能接触到 UNIX。

但是较小的计算机一直在被开发。到 80 年代末的时候，很多人都已经有了家用电脑。在那

个时候，已经有几个 UNIX 版本是为 PC 架构设计的，但是他们都不是免费的，而且非常非常的慢。所以大部分人在家用电脑上运行 MS DOS 或者 Windows 3.1。

1.1.2. Linus 和 Linux

到了 90 年代初，家用电脑终于能够运行一个完整的 UNIX 了。Linus Torvalds，一个在赫尔辛基大学学习计算机科学的年轻人，觉得应该有免费的教育版的 UNIX。于是他很快的就开始编写代码。

他开始询问问题，寻找能够帮助他把 UNIX 移植到 PC 上的答案和方法。下面是他在 1991 年发在 comp.os.minix 上面的一个帖子：

```
From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)
Newsgroups: comp.os.minix
Subject: Gcc-1.40 and a posix-question
Message-ID: <1991Jul3.100050.9886@klaava.Helsinki.FI>
Date: 3 Jul 91 10:00:50 GMT
Hello netlanders,
Due to a project I'm working on (in minix), I'm interested in the posix
standard definition. Could somebody please point me to a (preferably)
machine-readable format of the latest posix rules? Ftp-sites would be
nice.
```

从一开始，Linus 的目标就是开发一个和 UNIX 完全兼容的免费的操作系统。这就是为什么他询问 POSIX 标准。POSIX 现在还是 UNIX 标准。

在那个时代，即插即用还没有发明，但是很多人都希望有他们自己的 UNIX 系统，所以即使没有即插即用也只是一个小问题了。各种各样新硬件都有了新的驱动程序，并且新驱动程序出现的速度在不断的加快。几乎是一个新的硬件一出现，马上就有人把它买了后交给 Linux test（这个项目慢慢演变而成的名字），从而为越来越多的硬件发布更加多的免费代码。这些开发者并不只是针对他们的 PC；他们能找到的每件硬件对 Linux 都是有用的。

在那个时候，这些人被叫做书呆子或者怪物。但是这对他们来说无所谓，只要 Linux 支持的硬件名单越来越长。有了这些人，Linux 现在不但能很好的运行在新的 PC 上，而且还是那些旧的或者稀有的硬件的选择。没有 Linux，那些硬件是一无用处的。

Linus 的帖子发布的 2 年后，总共有了 12000 个 Linux 用户。这个在计算机爱好者中很流行的项目，在遵循 POSIX 标准的同时在稳定的增长。在接下来的几年中，UNIX 的所有功能都加了进来，演变成了目前的 Linux，一个成熟的操作系统。Linux 是一个完整的 UNIX 克隆，适合于工作站和中高档服务器。今天，软件和硬件市场中的重要商家都有他们自己的 Linux 开发团队；你甚至可以在你的本地代理商那里购买预安装的，有原厂支持的 Linux 系统，尽管还有很多的软硬件没有得到支持。

1.1.3. 当前 Linux 系统的应用

今天 Linux 已经进入到了桌面市场。Linux 开发者一开始的时候集中关注网络和服务功能，所以办公应用软件就成为了最后的障碍。我们不愿意承认微软在统治着这个市场，所以在过去的几年中，出现了很多替代应用程序来让 Linux 变成一个可以让人接受的工作站。这些替代产品提供了简单的用户界面以及和微软的办公应用程序兼容的软件，像文档处理，电子表格，演示文稿，等等。

在服务器方面，Linux 是众所周知的稳定和可靠的平台，为像著名的网上书店 Amazon，美国邮局，德国军队和其他许多的公司机构提供数据库和交易服务。Internet 供应商和服务商特别愿意把 Linux 用作防火墙，代理服务器和 Web 服务器。你总能在每个 UNIX 系统管理员的旁边找到一台 Linux，因为这是他们的使用起来很方便的管理站。Linux 集群曾经被用于“泰坦尼克号“，“怪物史瑞克”以及其他电影的拍摄中。在邮局里，他们是分捡邮件的神经中枢，在大型搜索引擎里，集群被用于 internet 搜索。这些只是世界各地 Linux 每天都在做的成千上万的重要工作中的几个例子。

值得一提的是，现代的 Linux 不仅能够运行在工作站和中高档服务器上，而且能在像 PDA，手机，嵌入式应用产品甚至实验性质的手表这样的“小玩意“上。这是 Linux 成为当今世界上唯一能被广泛的应用于不同的硬件的操作系统。

1.2. Linux 用户界面

1.2.1. Linux 难学吗？

Linux 是否难学取决于你问的人。有经验的 UNIX 用户会说不难，因为 Linux 是高端用户和程序员的理想操作系统，而且它曾经是，现在也是由这些人在开发的。

一个好的程序员希望的东西 Linux 都有：编译器，库，开发和调试工具。每个标准 Linux 版都有这些软件包。C 编译器是免费，不象很多 UNIX 发行版那样对这一工具收费。所有的文档和手册都很齐全，并且有很多例子帮助你能够马上开始做你的工作。它给人的感觉就好像 UNIX，所以从 UNIX 转换到 LINUX 是很自然的事情。

在 linux 的早期，只有专家才能使用它。那些掌握了 Linux 的人感到比其余的失败者要高人一等。他们经常告诉初学者去读手册。尽管每个系统都有手册，但是很难找到文档，即使找到了，里面的解释用了很多的技术术语。所以初学者很容易就放弃了。

Linux 社区开始意识到，如果 Linux 要成为操作系统市场中的重要一员，必须要做一些认真的改变，让 Linux 变得容易让人们接近。

1.2.2. 为菜鸟准备的 Linux

RedHat，SuSE 和 Mandriva 这样的公司冒了出来，他们提供适合大众消费的打包的 Linux 发行版。他们整合了大量的由社区开发的用户图形界面（GUI），使用户很容易管理程序和服务。作为一个今天的 Linux 用户，你仍然有各种办法来彻底的了解你的操作系统，但是这已经不是必须的了。

今天你可以通过图形界面登录，不用打一个字符就可以开始运行需要的应用程序，当然如果需要的话，你还是可以看到系统的代码。因为这样的结构，Linux 允许一个用户由浅如深的掌握这个系统：它同时适合新老用户。新用户没有必要去做一些困难的事情，而老用户没有必要一直用他们一开始接触 Linux 时候用的方法。

在服务领域的开发还在继续的同时，也正在为一般被认为不了解系统的桌面用户提供很好的东西。桌面应用程序的开发者正在非常努力的开发你曾经见过的最漂亮的桌面系统，或者把你的 Linux 机器看起来像你以前的 MS Windows 或者一个 Apple 工作站。最近的发展包括了 3D 加速的支持，USB 设备的支持，单点进行系统和软件包的更新，等等。Linux 有这些东西，并且把这些服务用普通用户能够理解的形式呈现出来。下面列举了一些非常好的例子；这些网站上有很多的截图，能够让你一窥 Linux 桌面的风采：
http://www.gnome.org/
http://kde.org/screenshots/
http://www.openoffice.org
http://www.mozilla.org

1.3. Linux 有前途吗？

1.3.1. 开源

开源软件背后的主意非常简单：当程序员可以读，发布和修改源代码，那么源代码就会成熟。人们可以适应它，修复它，调试它，而且他们可以以很快的速度做这些事情，这让那些在传统的公司里的软加开发者的进度看起来像蜗牛爬一样慢。这样的软件比那些用传统方法开发出来的软件更加的灵活和高质量，因为更多的人在更多不同的条件下测试了开源软件，而这是封闭环境下的软件开发人员办不到的。

开源倡议开始引起了商业世界的注意，并且逐渐的，商业供应商们开始明白了。尽管很多的学术和技术人员在 20 年前就明白了这是一条正确的路，商业供应商们需要 Internet 这样的应用才能让他们明白他们是可以从开源获利的。现在 Linux 早已过了被用作只有对有技术背景的人才有用的学术系统的阶段。

当前 Linux 不仅仅提供操作系统：已经有了一整套的基础设施来支持操作系统的开发，为操作系统而开发和测试程序，把所有需要的东西提供给用户，并且提供维护，更新和支持，和定制，等等。今天，Linux 已经准备好接受一个快速变化的世界的挑战。

1.3.2. 有十年的经验来为你提供服务

Linux 也许是最著名的开源倡议，但是有另外一个项目为 Linux 的流行作出了巨大的贡献。这个项目叫作 SAMBA,它的成就是对 MS Windows NT，OS/2，和 Linux 内置支持的，在 PC 机上用于文件和打印服务的 Server Message Block (SMB)/Common Internet File System (CIFS)协议做了逆向工程。现在几乎所有的系统都有适用的 SAMBA 软件包，这些软件包提供了在混合环境中使用 MS Windows 协议的交互连接方案：Windows 兼容的（向上支持到 WinXP）文件和打印服务。

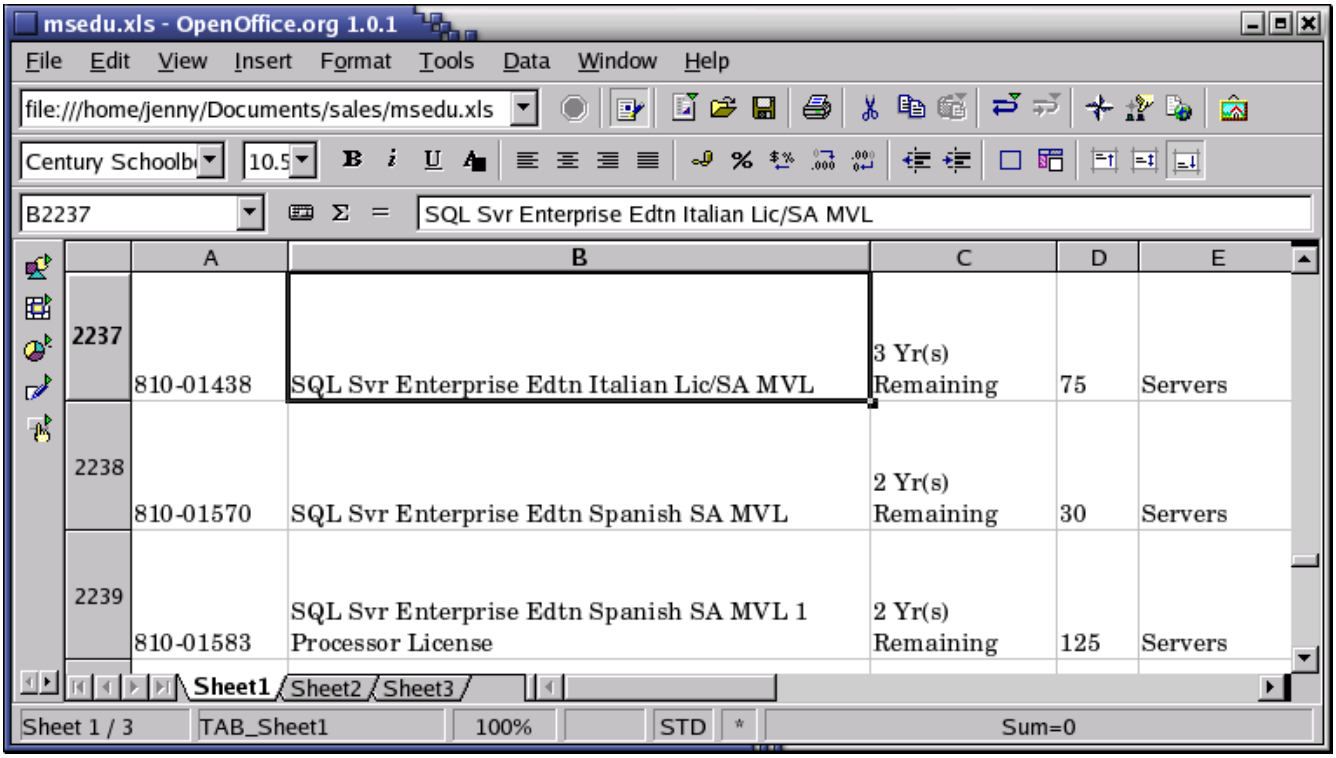
也许比 SAMBA 项目更加成功的是 Apache HTTP server 项目。这个服务器能够在 UNIX，Windows Nt 和许多其他的操作系统。它最初被叫作“A PAtCHy server”，是建立在一些现成的代码和补丁上的。它成熟的代码值得被冠以 Apache 这个美国土著部落的名字。这个部落的人民有优秀的战争战略和无穷无尽的忍耐力。Apache 比其他很多的 Web 服务器都要快很多，更加的稳定而且有更加多的功能。Apache 运行在每天有几百万访问者的网站上。尽管开发者没有提供什么官方的支持，Apache 用户社区可以为你提供所有问题的答案。现在有些第三方公司提供商业支持。

在办公应用程序方面，有 MS Office 办公套件的克隆，包括从部分到全部的 MS Windows 工作站上的应用程序。这些倡议让 Linux 被桌面系统市场接受起了很大的帮助，因为用户可以不用新的培训就能使用新的操作系统。随着桌面系统带来了普通用户的肯定，也同时带来了他们的变得越来越复杂和苛刻的特殊的要求。

开源社区主要由已经在里面超过 5 年以上的人组成。他们保证 Linux 在桌面市场和通用的 IT 应用中是重要的一员。拿薪水的雇员和志愿者一起努力使得 Linux 保持在市场中的位置。越来越多的用户，会带来越来越多的问题。开源社区保证答案会源源不断的来，并且以怀疑的眼光留意答案的质量，这样就使得 Linux 更加的可靠和容易接近。

本书不打算列举所有的 Linux 软件，因为有成千上万的软件包。在整个过程中，我们会提供给你最常用的软件包，这些软件包几乎都是免费的。为了让初学者减少害怕，这里是一个你最最想要的程序的截图。你可以看到他们不遗余力的让从 Windows 转过来的新用户感到自在。

图表 1-1. OpenOffice 与微软兼容的电子表格



1.4. Linux 的特性

1.4.1. Linux 的优点

Linux 的大量优点归根于他的起源，即扎根于 UNIX。当然这不包括第一个优点。

Linux 是免费的。

他们说就像是免费的啤酒。如果你不想花一分钱的话，你甚至连 CD 都不用买。Linux 可以全部免费从 Internet 网上下载下来。没有注册费，没有每个用户的费用，免费更新，并且如果你想要改变你的系统的行为的话你可以免费拿到源代码。

最重要的是，Linux 是自由的。

常用的许可证是 GNU Public License (GPL)。这个许可证允许任何人修改 Linux 并且最终重新发布修改后的发行版，只要修改后的源代码还是免费的。实际上，你可以免费拿到一个内核镜像，然后比如说增加对心灵传输机器的支持，或者对时间旅行机器的支持，然后卖掉你的新发行版，只要你的客户还是能拿到你的源代码。

Linux 可以被移植到任何硬件平台。

一个想要卖一种新电脑，或者不知道他的机器会运行什么样的操作系统（比如说你汽车或者洗衣机里面的 CPU）的供应商可以拿来一个 Linux 内核，修改后使它在他的硬件上工作，因为他可以免费拿到做这些事情所需要的文档。

Linux 是能长期运行的。

就象 UNIX，一个 Linux 系统是不需要重启就能长期运行的。这就是为什么很多任务被安排在夜间或者其他不忙的时候执行，使得在繁忙的时候有高可用性和对硬件更加合理的使用。这个特性使得 Linux 适合于那些人们无法或者没有时间来日夜控制他们的系统的环境中。

Linux 是安全的和多功能的。

Linux 的安全模式是基于 UNIX 的安全设计。UNIX 安全设计是坚固的，并且已经被证明是高质量的。但是 Linux 不仅仅适合作为抵抗 Internet 上敌人进攻的城堡，它还可以利用同样的高安全标准来适应其他的情况。你的开发机器和控制站会和你的防火墙一样的安全。

Linux 是可扩展的。

通过增加或者删除适当的软件包，Linux 可以运行在从只有 2MB 内存的掌上型电脑到有几百个节点的有千兆兆(PB)级存储空间的集群。你已经不再需要超级计算机了，因为你可以利用 Linux 系统提供给你的搭建模块来做更大的事情。如果你想做一些小事情，比如说给一个处理器建一个操作系统或者只是废物利用你的旧 486 机器，Linux 也可以做的很好。

Linux 操作系统和大部分的 Linux 程序有非常短的调试周期。

因为有成千的人在开发和测试 Linux，一般来说能够很快的发现错误并且找到人来改正这些错误。有的时候错误被发现后几个小时就能被改正。

1.4.2. Linux 的缺点

有太多的发行版了。

就像罗马人说的，“Quot capites, tot rationes” （人越多，意见越多）。乍看起来，Linux 发行版的数量可能会让你感到害怕或者让你觉得很荒谬，但是这也说明每个人都能找到他或她所需要的。你没有必要是专家才能找到合适的发行版。

如果你问的话，一般 Linux 用户会说他用的发行版是最好的。所以你应该选择那个发行版呢？其实没有必要担心太多，因为每个发行版基本上都包括了基本的软件包。在这个基础上，一些第三方软件使得 TurboLinux 更加适合于中小企业，RedHat 适合于服务器，而 SuSE 适合于工作站。但是，区别很可能只是表面的。最好的策略是测试一下几个发行版；很可惜的是，不是每个人都有时间这样做。幸运的是，有足够多的建议告诉你怎样选择你的 Linux。利用关键字“choosing your distribution”在 google 上做个很快的搜索，就能给你几十个很好的建议

的连接。Installation HOWTO （ http://www.tldp.org/HOWTO/Installation-HOWTO/ ）也讨论了怎样选择你需要的发行版。

Linux 对初学者不是很友善，并且让他们感到困惑。

必须承认 Linux，至少是它的核心系统，不如 MS Windows 友善，而且肯定比 MacOS 要难用。但是。。。随着 Linux 的流行，已经做了很多的努力让 Linux 变得容易使用，特别是对初学者。每天都有更多的信息被发布来填补给不同水平用户的文档的空白，比如说本书。

开源产品值得信任吗？

免费的产品能是可靠的吗？Linux 用户可以选择是否使用 Linux，和那些使用专有软件而没有这样的自由的用户比，他们很大的优势。经过长时间的测试，大部分的 Linux 用户得出结论，Linux 不但很好，而且在很多情况下比传统的方案更好更快。如果 Linux 不好的话，它早就不存在了，不会有现在的普及和几百万的用户。现在用户可以影响他们的系统，并且和社区分享他们的想法和意见。这使得系统每天都在改进。这确实是一个永远也做不完的项目，但是在一个不断变化的环境中，Linux 同时也是一个不断争取完美的项目。

1.5. Linux 版本

1.5.1. Linux 和 GNU

尽管有大量的 Linux 实现，如果仅仅是因为每个 Linux 机器是按照你的需要而把建造模块放在一起的一个盒子，那么你会发现不同发行版之间有很多相似的地方。安装一个系统只是一个长期关系的开端。就在你认为你有了一个很好的运行的机器的时候，Linux 会刺激你的想象力和创造力。你越意识到这个系统能给你的威力，你越想重新定义它的极限。

按照发行版，你的硬件和个人喜好，Linux 也许看起来不一样，但是作为所有的图形和其他的界面的基础的那部分是一样的。Linux 系统是基于 GNU（Gnu's Not UNIX）工具，这些工具提供了一套标准的方法来处理和使用系统。所有的 GNU 工具都是开源的，所以它们可以被安装在任何的系统上。大多数的发行版提供预编译的最常用工具的软件包，比如 RedHat 的 RPM 软件包和 Debian 的 Debian 软件包（也叫 deb 或者 dpkg），这样即使你不是个程序员你也能在你的系统上安装一个软件包了。然后，如果你喜欢自己动手，那么你会更加喜欢 Linux 的，因为大多数的发行版提供一套完整的开发工具来让你从源代码安装一个新软件。这种设置也允许你安装那些没有以适合你的系统的预编译形式存在的软件。

常用 GNU 软件：

Bash：GNU 命令外壳

GCC：GNU C 编译器

GDB：GNU 调试器

Coreutils：一套基本的 UNIX 风格的实用工具，比如说 ls, cat 和 chmod

Findutils：查找文件

Fontutils：转换字体格式或者建立新的字体

The Gimp：GNU 图像处理软件

Gnome：GNU 桌面环境

Emacs：一个非常强大的编辑器

Ghostscript 和 Ghostview：PostScript 文件的解释程序和图形界面

GNU Photo：和数字照相机互动的软件

Octave：一门主要是做数字计算和图形处理的编程语言

GNU SQL：关系数据库系统

Radius：一个远程验证和账户服务器

。。。

还有很多商业应用程序可以在 Linux 上运行。要了解他们更多的信息的话，就要看他们的文档了。在本书中，我们只讨论免费软件，他们大部分采用 GNU 许可证。

为了安装缺少的或者新的软件包，你需要一种软件管理格式。最常见的格式包括 RPM 和 dpkg。RPM 是 RedHat 软件包管理器。它被用于多种 Linux 系统中，尽管名字听起来不象。Dpkg 是 Debian 软件包管理系统，它使用一种叫 apt-get 的界面，这种界面也能管理 RPM 软件包。Novell Ximian Red Carpet 是带图形界面的 RPM 的第三方实现。另外的第三方软件供应商也许有他们自己的安装过程，有时候会模仿像 MS Windows 和其他平台上面的 InstallShield。当你慢慢的深入 Linux 的时候，你很可能会接触一个或多个这种程序。

1.5.2. GNU/Linux

Linux 内核（你系统的骨骼，参见 3.2.3.1）不属于 GNU 项目，但是使用相同的许可证。绝大部分的使用程序和开发工具（你系统的肌肉）是从 GNU 项目里拿来的，他们不是 Linux 独有的。因为任何可用的系统都必须包括内核和至少一个最精简的实用程序集，一些人认为这样的一个系统应该被称作 GNU/Linux 系统。

为了最大程度的独立于不同的发行版，在本书中我们就讨论这样的 Linux。如果我们不是在讨论 GNU/Linux 的时候，我们会表明特定的发行版，版本号和程序名。

1.5.3. 我应该安装哪个发行版

在安装前，硬件是你最重要的因素。既然每个 Linux 发行版包括了基本的软件包，而且能够按照任何不同的需求而建立起来（因为他们都使用 Linux 内核），你只要考虑这个发行版是否能在你的硬件上运行。比如说，LinuxPPC 能够在 Apple 和其他的 PowerPC 上面运行，但是不能运行在普通的基于 x86 的 PC 上。LinuxPPC 确实能在新的 Mac 上运行，但是你不能把它用于一些使用旧 BUS 技术的电脑。另外一个难以处理的情况是 Sun 硬件，它可能是旧的 SPARC CPU 或者新的 UltraSparc，这两者需要不同的 Linux。

有些 Linux 发行版是为一些特定的处理器优化过的，比如说为 Athlon CPU，但是他们也能很好的运行在标准的 486，586 和 686 因特尔处理器上。有时候为特殊 CPU 制作的发行版不是那么可靠，因为较少的人测试过。

大多数的 Linux 发行版提供一套准对通用 PC 的程序和为基于 Intel x86 CPU 优化后的内核。这些发行版经过很好的测试，并且被定期维护。它们集中精力于稳定的服务器的实现和简单的安装和更新过程。这样的例子包括 Debian，Ubuntu，Fedora，SuSE 和 Mandriva。他们是最主流的 Linux 系统，一般认为对初学者来说他们是很容易上手的，但是也不妨碍专业人员最大限度的利用他们的 Linux 机器。Linux 同时也很好的运行在笔记本和中档的服务器上。新硬件的驱动程序只有经过了广泛的测试才会被包括在系统里，这样就增加了系统的稳定性。

一个系统的标准桌面系统也许是 Gnome，但是另外一个系统也许是 KDE。一般情况下，Gnome 和 KDE 都可以运行于所有主流的 Linux 发行版。另外的窗口和桌面管理器适合更加高级的用户。

标准的安装过程允许用户在不同的基本配置之间选择，比如作为工作站的话，就会安装每天要用的软件包和开发工具，或者为服务器安装的话，就会选择不同的网络服务。专家级的用户在

初始安装过程中可以安装他们需要的软件包的任何一个组合。

本指南的目标是适合所有的 Linux 发行版。但是为了你自己方便，强烈建议初学者挑选一个主流的发行版，它默认状态下就支持常见的硬件和程序。

下面的是一些非常适合新手的选择：

Fedora Core

Debian

SuSE Linux

Mandriva (以前叫 MandrakeSoft)

Knoppix：一个在你的 CD-ROM 上运行的操作系统，你没有必要安装任何东西

ISO 镜像可以从 LinuxISO.org 下载。主要的发行版可以从正规的计算机商店里购买。

第 2 章. 快速入门

为了最有效的使用这个指南文档，我们将马上开始一个连接到 Linux 系统并进行基本实验操作的实践章节。

我们将讨论:

- 系统连接
- 系统断开连接
- 文本和图形模式
- 更改账户密码
- 通过文件系统导航
- 识别文件类型
- 查看文本文件
- 查找帮助信息

2.1. 登录，激活用户界面和登出

2.1.1. 简介

为了能在 Linux 系统直接工作，你将需要提供一个用户名和密码。你总是需要身份验证才能访问系统。正如我们已经在第 1 章练习中提到，大多数基于 PC 的 Linux 系统有两种基本的系统运行模式：一是快速，清醒的文本控制台模式，它看起来像带有鼠标的 DOS，多任务和多用户功能，或在图形模式，它看起来更好，但会消耗更多的系统资源。.

2.1.2. 图形模式

这是现今大多数台式计算机的默认模式。当你最初被要求输入你的用户名，然后在一个新窗口，需要输入你的密码，你就知道你将使用图形模式连接到该系统。

要登录时，请确保鼠标指针在登录窗口，提供您的用户名和密码给系统验证，然后点击确定或按回车键。


小心使用 root 帐号!

人们普遍认为使用 root 用户，即系统管理员帐户，来连接（图形化的）系统是一个坏主意。因为使用图形需要包括运行许多额外的程序，在使用 root 用户下涉及很多额外的权限。为了让所有的风险尽可能低，应该使用一个普通用户帐户进行图形的连接。但是，有足够的风险使得要记住这个一般性的建议，所有的 root 帐号的使用：只作为 root 用户登录时需要额外的特权。

在输入您的用户名/密码组合后，它可能需要一些时间才启动图形环境，这取决于您的计算机 CPU 速度，您使用的软件和您的个人设置。

若要继续，您需要打开一个终端窗口或简化的 xterm（X 代表支持图形环境的基础软件名称）。这个程序可以在应用程序->工具，系统工具或互联网菜单这些地方找到，这取决于您所使用的窗口管理器类型。可能有图标作为一个快捷方式以获得 xterm 窗口，而且桌面上点击鼠标右键，背景通常会提示一个包含终端窗口的应用程序菜单。

在浏览菜单时，你会发现很多事情可以在不通过键盘输入指令下完成。对于大多数用户而言，传统'n'点击来处理计算机任务的方法也可行。但是，本指南是为未来的需要更改系统核心的网络和系统管理员准备的。他们需要一个比用鼠标更强大的工具来处理所有任务。这个工具就是 shell，当在图形模式下，我们通过启动一个终端窗口中打开我们的 shell。

该终端窗口是您的系统控制面板。几乎所有后面涉及到的任务都可以使用这个简单但又强大的文本工具完成。一个终端窗口在打开后应始终显示一个命令提示。该终端显示标准提示，它显示用户的登录名和当前工作目录，由（～）代表：

图 2-1. 终端窗口

另一种常用的提示形式是:

```
[user@host dir]
```

在上面的例子中, user 代表你的登录名，hosts 代表工作所在的主机名，而 dir 指示你当前所在的文件系统路径。

稍后我们将详细的讨论提示和它的行为。现在，只需知道，提示可以显示各种信息，但它们不是你提交给系统的命令的一部分。

要从使用图形模式的系统断开，您需要关闭所有终端窗口和其他应用程序。在此之后，击中注销图标或菜单中找到退出按钮。关闭一切都没有必要的，系统能为你做这个，但当你再次连接时会话管理可以在你的屏幕上显示当前打开的应用程序，这需要较长时间并且不是我们预期的效果。然而，这种行为是可配置的。

当你看到登入界面时，输入用户名和密码，这样退出系统就成功了。

Gnome 或 KDE?

我们已经提过 Gnome 和 KDE 桌面几次了。这是两个最流行的桌面管理方式，虽然还有很多很多其他的方式。无论你选择的桌面工作是哪种- 只要你知道如何打开一个终端窗口就可行了。不过，我们会继续应用 Gnome 和 KDE 作为实现某些任务的最流行的方法。

2.1.3. Text mode

You know you're in text mode when the whole screen is black, showing (in most cases white) characters. A text mode login screen typically shows some information about the machine you are working on, the name of the machine and a prompt waiting for you to log in 当整个屏幕为黑色，显示（在大多数情况下）字符时，你知道你是处于文本模式下了。一个典型的文本模式登录屏幕显示一些关于你工作机器的资料，机器的名称和提示等你登录：

```
RedHat Linux Release 8.0 (Psyche)


blast login: _
```

文本模式登录与图形登录不同，你必须先提供您的用户名，然后按下回车键，因为有在屏幕上你不能用鼠标点击任何按钮。然后你输密码，输入后再按回车键。你不会看到任何迹象表明你键入的字符，甚至没有一个星号，你不会看到光标移动。但是，这在 Linux 上是正常的，是出于安全因素而设计的。

当系统认为你是有效用户，你可能会得到一些更多的信息，称为日期信息，它可以是任何内容。此外，显示 cookie 是 UNIX 系统中流行的做法，它包含一些一些明智或不明智的（这取决于你）想法。在此之后，你会得到一个 shell，使用图形模式下同样的提示。

 别以 root 用户登录

在文本模式类似的：只在做绝对需要管理员权限的安装和配置时才以 root 用户登录，如添加用户，安装软件，并进行网络和其他系统配置。一旦你完成后，立即退出这个特别帐号并以一个非特权用户来继续其他的工作。另外，有些系统如 Ubuntu，迫使你使用 sudo，这样你不需要直接访问管理员帐户。.

输入 logout 命令并按下回车键进行退出操作。当你看到登录窗口的时候，就表明你退出成功了。

 电源按钮

尽管 Linux 不应应用了不恰当的程序而导致的系统被阻塞而关闭，但按下电源按钮就相当于在新的系统上启动这些程序。然而，断电停机过程中没有运行完阻塞进程可能会造成严重的系统损坏！如果你想更安全些，就需要在退出图形界面下的系统时始终使用关闭选项，或者在登录屏幕上（如你必须提供你的用户名和密码）寻找一个关闭按钮。.

到现在为止，我们知道如何登录和退出系统了，这就为我们开始使用命令做好了准备。

2.2. 基础知识

2.2.1. 命令

这是快速指南，用于方便入门；我们将在后面的章节对它们进行详细的描述。

Table 2-1. 命令快速入门

命令	含义
ls	显示当前工作目录下的文件，类似 DOS 的 dir 命令
cd directory	更改工作目录
passwd	更改当前用户的密码
file filename	显示文件的文件名和文件类型
cat textfile	在显示屏上显示文件的内容
pwd	显示当前目录的文件系统路径
exit or logout	退出会话
man command	从 man 中获得命令信息
info command	从 info 中获得命令信息
apropos string	search the whatis database for strings

2.2.2. 一般说明

在图形或字符界面下的终端窗口，你在提示符之后输入这些命令，然后输入回车键。

命令可以发出单独的指令，如 ls。当你指定一个选项可得到不同的命令行为，通常使用一个破折号（前面 - ），例如 ls - l。相同的选项在另一个命令可能得到不同的含义。 GNU 程序需要长的选项，由两个破折号指示(--)，类似于前面说明的 ls --all。有一些命令没有选择。

命令的参数指定了这个命令需要起作用的对象。一个例子如：ls /etc，其中的目录/ etc 是 ls 命令的参数。这表明，你希望看到该目录的内容，而不是默认的当前目录的内容，想获得当前目录内容只要输入 ls 之后键入 Enter 就能获得。有些命令需要参数，有时参数是可选的。.

你能从在线帮助中找出一个命令是否带选项和参数，以及哪些是合法的，请参考 Section 2.3 。

在 Linux 中，如在 UNIX 中，目录是使用斜线分开的，如在网络所使用的地址(URLs)。我们将在后面深入讨论的目录结构。

符号 . 和 .. 在目录意义上有特殊的意义。我们会在练习中尽量了解，并在下一章进行更多的了解。

尽量避免登录或使用系统管理员帐户，root。除了做你的正常工作，大部分任务，包括系统检查，收集信息等，可以使用没有任何特殊权限的普通用户帐户。如果需要，例如当创建一个新用户或安装新软件时，获得 root 访问的首选方法是通过切换用户的 ID，请参阅 Section 3.2.1 中的例子。

本书几乎所有可以执行的命令不需要系统管理员权限。在大多数情况下，当以非特权用户发出一个命令或启动一个程序时，如果系统需要 root 权限时，会发出警告或提示你输入 root 密码。一旦完成后，立即现有离开应用程序或会话以使你拥有 root 权限。

读文档应成为你的第二习惯。特别是在开始的时候，重要的是要阅读系统文档，基本命

令，HOWTOs 手册等。由于文件数量是如此巨大，所有不可能包括所有相关的文件。这本书将引导你去阅读每个讨论主题所涉及的最合适文档，以激励阅读 man 手册的习惯。

2.2.3. 使用 Bash 特性

Bash 的几个特殊的组合键让你比在使用了 GNU 的 shell 做事情更容易，它几乎是任何 Linux 系统默认的 shell，参见第 3.2.3.2。下面是最常用的功能列表，强烈建议你作出的一种习惯去使用它们，以从一开始使用 Linux 就能获得最优体验。

Table 2-2. Bash 的主要组合键

Key or key combination	Function
Ctrl+A	Move cursor to the beginning of the command line.
Ctrl+C	End a running program and return the prompt, see Chapter 4.
Ctrl+D	Log out of the current shell session, equal to typing exit or logout.
Ctrl+E	Move cursor to the end of the command line.
Ctrl+H	Generate backspace character.
Ctrl+L	Clear this terminal.
Ctrl+R	Search command history, see Section 3.3.3.4.
Ctrl+Z	Suspend a program, see Chapter 4.
ArrowLeft and ArrowRight	Move the cursor one place to the left or right on the command line, so that you can insert characters at other places than just at the beginning and the end.
ArrowUp and ArrowDown	Browse history. Go to the line that you want to repeat, edit details if necessary, and press Enter to save time.
Shift+PageUp and Shift+PageDown	Browse terminal buffer (to see text that has "scrolled off" the screen).
Tab	Command or filename completion; when multiple choices are possible, the system will either signal with an audio or visual bell, or, if too many choices are possible, ask you if you want to see them all.
Tab Tab	Shows file or command completion possibilities.

上述表的最后两个命令可能需要一些额外的解释。例如，如果你想变更目录到目录 directory_with_a_very_long_name，你不会输入很长名称，不会。你只要在命令行键入 cd 目录，然后按下 Tab 键，如果没有其他文件也具有相同的前三个字符，然后 shell 就自动为你显示完整的文件名。当然，如果没有其他文件以”d”开头，那么你可能只需键入 cd d 和 Tab。如果超过一个文件具有相同的开始字符时，这个 shell 将告知你，你可以在间隔很短的情况下

两次键入 Tab，Shell 就会显示你所需要的选择：

```
your_prompt> cd st
starthere stuff stuffit
```

在上面的例子中，如果你在输入前两个字符后键入”a”，并再次点击 Tab，没有其他的可选性后，Shell 自动完成目录的名称，你就不必键入字符串“rthere”：

```
your_prompt> cd starthere
```

当然，你还需要键入 Enter 使 Shell 接受你的命令。

同一个例子，如果你输入“U”，然后按下 Tab，shell 将自动为您添加“ff”，但随后它不做选择了，因为后面有多个可能的选择。如果你输入制表符 Tab 键再次，您会看到的选择，如果你键入一个或多个字符，使系统明确的选择你所需要的，并再次键入 Tab，或在你输入选择的完整个文件名称后键入 Enter，外壳程序匹配完整的文件名称并更改至该目录 - 如果它确实是一个目录名称。.

这对命令的参数是文件名的情况同样适用。

类似命令名称匹配。输入 ls 然后键入 Tab 两次，这将列出您的路径中的所有以 ls 开头的命令（见第 3.2.1）：

```
your_prompt> ls
```

```
ls          lsdev    lspci       lsraid       lsw
lsattr      lsmod    lspgpot     lss16toppm
lsb_release lsof     lspnp       lsusb
```

2.3. 获取帮助信息

2.3.1. 警告

GNU / Linux 越来越涉及主动性。通常这一系统有几种方法来实现这一目标。一种常见的获得帮助的方法是找到一个知道的人，不过耐心的和爱好和平的 Linux 社区成员希望，你要先去尝试本节提到的一个或多个方法，然后才请求他们的帮助，如果你能不遵循这个基本规则，那么他们的表达方式可能就比较苛刻了。

2.3.2. The man pages

很多用户一开始都害怕使用 man 手册，因为他们是所有文件的来源。但他们却是很有条理的，你会从下面的 man man 的例子看出来。

通常是在图形模式下的终端窗口下读取 man 手册的，或者如果你喜欢也可以只在文本模式下。在提示之后输入如下的命令，然后在键入 Enter:

```
yourname@yourcomp ~> man man
```

man 文档在你键入 Enter 后将显示在你的屏幕上：

```
man(1) man(1)

NAME
 man - format and display the on-line manual pages
 manpath - determine user's search path for man pages

SYNOPSIS
 man [-acdfFhkKtwW] [--path] [-m system] [-p string] [-C config_file]
 [-M pathlist] [-P pager] [-S section_list] [section] name ...

DESCRIPTION
 man formats and displays the on-line manual pages. If you specify
 section, man only looks in that section of the manual.
name is normally the name of the manual page, which is typically the
name of a command, function, or file. However, if name contains a
slash (/) then man interprets it as a file specification, so that you
 can do man ./foo.5 or even man /cd/foo/bar.1.gz.
```

使用空格键来浏览下一个页面。你可以使用 B 键回到以前的页面。当你到达终点，man 通常会退出，你会重新获得命令提示符。输入 q 如果你想到达结束前离开，或者在用户到达尾页后 man 不自动退出。

 页面阅读器

操作 man 页面的组合键取决于你的所使用的发行版本的页面浏览器。大多数发行版本使用 less 命令来查看网页和左右滚动。参阅第 3.3.4.2 页面浏览器的更多信息。

每一个 man 页面都包括几个标准段落，就像我们从 man man 的例子中看到的一样：

第一行包含的你正在阅读的命令名称，以及本手册页所在的段 ID 号。Man 页面根据章节排序。命令可能有多个 man 页面，例如用户手册页部分，系统管理手册页部分，以及程序员节手册页。

显示的该命令名称和简短描述，可以用于构建 man 页面的索引。你可以查找这个索引中使用 apropos 命令。

该命令的大纲规定了此命令拥有的所有选项和技术符号/或参数。你可以把选择想像成执行这条命令的一种方式。所带的参数就是命令的执行对象。一些命令没有选择或没有参数。可选的选项和参数放在“[”“]”之间，以表明它们是可选的。

接着给出了对这个命令的更长的详细描述。

列出选项和它们的描述。选项通常可以组合在一起。如果不是这样的话，本节将会告诉你。

环境变量介绍了影响该命令的 shell 变量（不是所有的命令都有环境变量）。

有时候还提供这个命令的具体章节。

对其他 man 页面的参考链接提供在 “SEE ALSO”一节。括号中的数字是能查到到这个命令的 man 页面段落。有经验的用户经常使用 / 命令附上搜索字符串，并键入 Enter 来切换到“SEE ALSO”部分。

通常也有有关已知错误（异常信息）的信息，并提交报告您可能会发现的新错误的地方。可能还会提供作者和版权的信息。

Some commands have multiple man pages. For instance, the passwd command has a man page in section 1 and another in section 5. By default, the man page with the lowest number is shown. If you want to see another section than the default, specify it after the man command: 一些命令有多个 man 页面。例如，passwd 命令有一个 man 页面在第 1 部分并且在第 5 部分也有另一个 man 页面。默认情况下，只显示最低数字的 man 页面。如果你想看到另一个部分，在 man 命令后指定该部分：

man 5passwd

如果你想看到 man 的所有页面，一页接一页的，就需要在 man 命令后附上-a 参数：

man -apasswd

使用这种方式，当你到达第一个 man 页面的末尾后并再次键入 SPACE，下一部分的 man 页面就会显示出来。

2.3.3. More info

2.3.3.1. Info 页面

除了 man 页面，您可以使用 info 命令查阅该命令的信息页面。这些信息页面通常含有较多的最新信息，是比较容易使用。一些命令的 man 页面就是参考信息页面的。

在终端上输入 info info：

```
File: info.info, Node: Top, Next: Getting Started, Up: (dir)


Info: An Introduction
**********************


 Info is a program, which you are using now, for reading
documentation of computer programs. The GNU Project distributes most
of its on-line manuals in the Info format, so you need a program called
"Info reader" to read the manuals. One of such programs you are using
now.


 If you are new to Info and want to learn how to use it, type the
command `h' now. It brings you to a programmed instruction sequence.

 To learn advanced Info commands, type `n' twice. This brings you to
`Info for Experts', skipping over the `Getting Started' chapter.
```

```
* Menu:

* Getting Started:: Getting started using an Info reader.

* Advanced Info:: Advanced commands within Info.

* Creating an Info File:: How to make your own Info file.
--zz-Info: (info.info.gz)Top, 24 lines --Top-----------------------------
Welcome to Info version 4.2. Type C-h for help, m for menu item.
```

使用箭头键来浏览文本上下文，移动光标到起始字符为星号并且包含你想要的信息的关键字的一行，然后按下回车。使用 P 和 N 键移动到上一个或下一个主题。空格键会让你翻到下一页，无论是否启动一个新主题或另一命令的信息页。使用 Q 退出。该 info 程序还有很多其他的信息资料。

2.3.3.2. whatis 和 apropos 命令

可使用 whatis 命令来得到一个简短的命令解释，如下面的例子：

```
[your_prompt] whatis ls
ls (1) - list directory contents
```

它列出了一个命令的简单信息，而且在 man 页面的第一部分也包含了这样的一个信息页。如果你不知道从哪里开始入手并且不知道开始阅读哪个 man 页，apropos 可以让你获得更多的信息。比如说你不知道如何启动一个浏览器，那么你可以输入以下命令：

```
another prompt> apropos browser
Galeon [galeon](1) - gecko-based GNOME web browser
lynx (1) - a general purpose distributed information browser


for the World Wide Web
ncftp (1) - Browser program for the File Transfer Protocol
opera (1) - a graphical web browser
pilot (1) - simple file system browser in the style of the


Pine Composer
pinfo (1) - curses based lynx-style info browser
pinfo [pman] (1) - curses based lynx-style info browser
viewres (1x) - graphical class browser for Xt
```

按 Enter 后你会看到，你的机器上有很多与浏览器相关的东西：不仅是网络浏览器，而且文件和 FTP 浏览器，以及文件浏览器。如果您还安装开发包，你也有可能得到编写浏览器相关

程序所附带的 man 页说明。一般来说，在 man 页命令说明在第一页，所以标为“（1）”，适合作为用户进行尝试。发出上述 apropos 命令的用户可能因此尝试启动 galeon，lynx 或 opera 命令，因为这些显然都与浏览万维网有关。

2.3.3.3. --help 选项

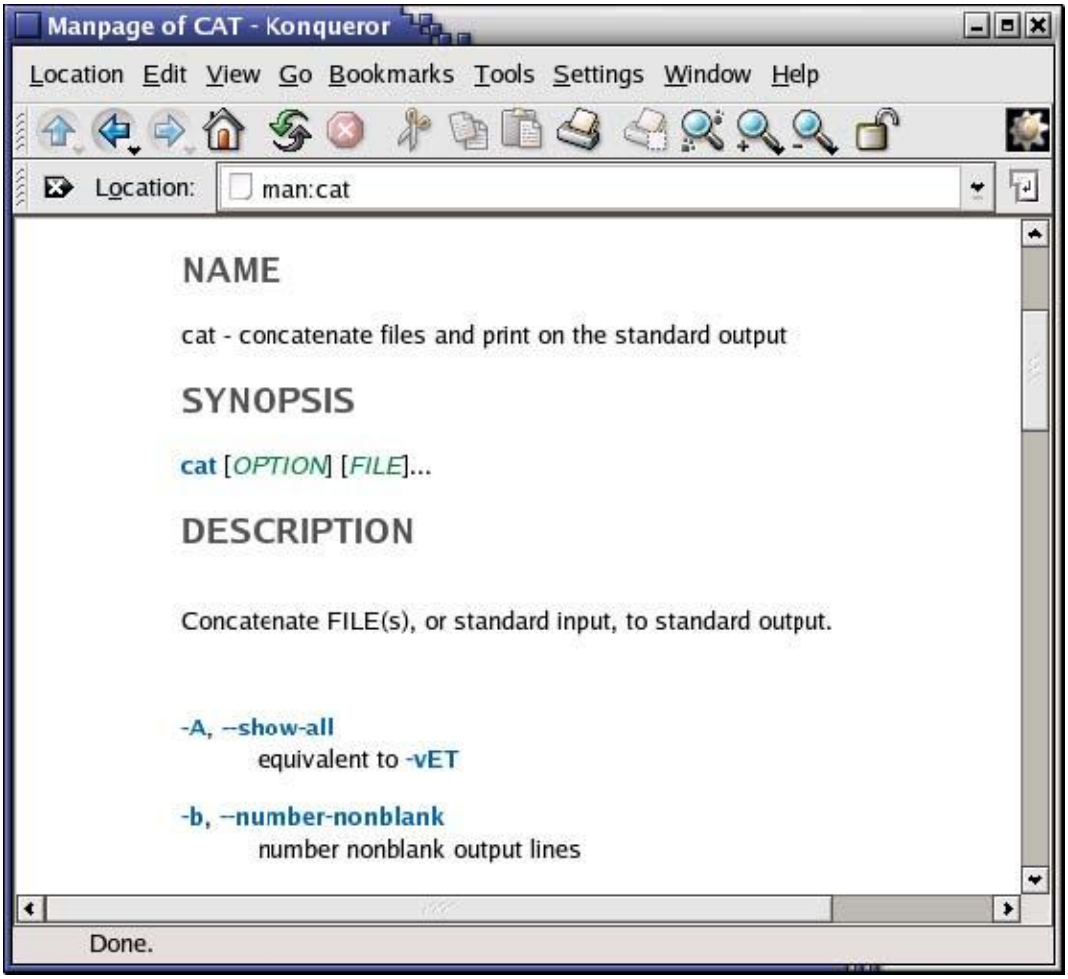
大部分的 GNU 支持—help 选项，该选项给出了如何使用这个命令的简短说明以及这个命令的一系列选项。 下面就是 cat 命令带有—help 选项后的输出结果：

```
userprompt@host: cat --help
Usage: cat [OPTION] [FILE]...
Concatenate FILE(s), or standard input, to standard output.


 -A, --show-all equivalent to -vET
 -b, --number-nonblank number nonblank output lines
 -e equivalent to -vE
 -E, --show-ends display $ at end of each line
 -n, --number number all output lines
 -s, --squeeze-blank never more than one single blank line
 -t equivalent to -vT
 -T, --show-tabs display TAB characters as ^I
 -u (ignored)
 -v, --show-nonprinting use ^ and M- notation,


except for LFD and TAB


 --help display this help and exit


 --version output version information and exit


With no FILE, or when FILE is -, read standard input.


Report bugs to <bug-textutils@gnu.org>.
```

2.3.3.4. 图形帮助

如果你喜欢的图形用户界面的话也不用绝望。 Konqueror，默认的 KDE 文件管理器，提供丰富多彩进入 man 和 info 页的方式。你可能想在地址栏尝试”info:info”，你会得到一个关于 info 命令的可浏览的 info 页。同样的，输入”man：ls”将向你输出 ls 命令的 man 页。你甚至可以完成命令的名称：你会在 man 页里看到所有由“ls”开头的命令下拉菜单。在地址栏位置输入"info:/dir"就会显示 info 页，以工具排列顺序显示。优秀的帮助内容，包括 Konqueror 的手册。从启动菜单或通过在终端窗口输入命令的 konqueror，在键入 Enter，就能见到下面的截图了。

Figure 2-2. Konqueror as help browser



Gnome 的帮助浏览器也具有非常好的用户友好性。你可以启动它使用以下几种方法，从 Gnome 的菜单中选择应用->帮助来启动，通过点击桌面上的 liftguard 图标或在一个终端窗口输入命令 gnome -help 来启动。该系统文件和 man 页可以方便的通过使用简单的接口进行浏览。

Nautilus 文件管理器提供了对 man 和 info 页搜索的索引，所以很容易浏览并被关联。Nautilus 可以从命令行，或单击您的主目录图标，或从 Gnome 的菜单启动。

GUIs 对系统文件最大的好处是，所有的信息是完全相通的，所以你可以单击“请参阅”部分和所有其他人的网页链接，浏览和获取，从而不必在浏览和获取知识上中断几个小时。

2.3.3.5. 特殊命令

一些命令没有单独的文档，那是由于其他命令的一部分。cd, exit, logout 和 pwd 就是这些特殊命令。 它们是你的 shell 程序的一部分，所以被叫做 shell 内嵌命令。请参考你使用的 shell 的 man 和 info 手册来来获得这些信息。用户开始使用 Linux 系统一般都是使用 Bash Shell。参考 Section 3.2.3.2 以获得更多的 shell 信息。

如果你更改了原始的系统配置，可能 man 手册还在，但是却由于你的 shell 环境的变更而变得不可见。这种情况下，你需要检查 MANPATH 环境变量。如何做将会在 Section 7.2.1.2 描述。

一些程序或包只有在/usr/share/doc 有一些简介或参考信息。具体情况请参见 Section 3.3.4。

最坏的情况，你可能偶然的从你的系统中移除了这些文档(希望只是偶然，因为这是非常不好的主意)。在这种情况下，首先要确定在搜索工具搜索下找不到任何有用的信息，搜索工具参考 Section 3.3.3。如果是这样的话，你需要重新安装与这些文档相关的命令的手册包，参见 Section 7.5。

2.4. 总结

Linux 系统通常运行在文本模式或图形模式。由于现今的 CPU 和 RAM 再也不是昂贵的资源，使得每个 Linux 用户都使用图形模式而且很多人确实是这么做的。但是这并不是说你不需要了解文本模式：我们将在本课程中使用文本模式环境，使用终端窗口的方式。

Linux 鼓励用户自主的获取知识并具有独立性。不可避免的，你需要阅读大量的文档以达到这样的目标；那也是你所看到的，我们在本书中为每个命令，工具和问题参考了其他的文档。你阅读越多的文档，你就更容易并更快的查阅相关手册。尽快的让阅读文档成为你的一种习惯。当你不知道一个问题的答案时，参考相关文档应该成为你的习惯。

我们已经学习了一些命令：

Table 2-3. New commands in chapter 2: Basics

Command	Meaning
apropos	Search information about a command or subject.
cat	Show content of one or more files.
cd	Change into another directory.
exit	Leave a shell session.
file	Get information about the content of a file.
info	Read Info pages about a command.
logout	Leave a shell session.
ls	List directory content.
man	Read manual pages of a command.
passwd	Change your password.
pwd	Display the current working directory.

2.5. 练习

我们所学的大部分内容是制造错误并查看是如何出错的。这些练习的作用是让你看到一些错误信息。做这些练习的顺序很重要。

不要忘记在命令行使用 Bash 特性：尽量少输入字符来完成这些练习！

2.5.1. 连接和断开

判定你是工作于文本模式还是图形模式

我正在文本/图形模式下工作(删除不使用的东西)

使用你安装时设定的用户名和密码登录系统

退出系统

再次登陆，使用一个不存在的用户名 ->发生了什么?

2.5.2. 密码

使用你的用户名和密码再次登陆。

• 更改你的密码为 P6p3.aa! 并单击 Enter 键。

->发生了什么?

• 再试一次，这次输入非常简单的密码，比如 123 或 aaa.

->发生了什么?

• 再试一次，这次不输入密码仅仅是单击 Enter 键

-->发生了什么?

使用命令 psswd 替代 passwd

-->发生了什么?

新密码

如果你的密码不更改会原来的密码，新密码将是"P6p3.aa!"。在这次练习之后还原你的密码。

注意某些系统不允许循环使用密码，比如在一定时间内或一定密码变更次数内恢复原来的密码。

2.5.3. 目录

这些练习会帮助你对这部分内容更熟悉。

• 输入命令 cd blah

->发生了什么?

• 输入命令 cd .. 记住在"cd" 和 ".."之间有空格! 使用 pwd 命令

->发生了什么?

• 使用 s 命令列出目录的内容

->你将会看到什么?
->你觉得这些是什么?
-> 使用 pwd 命令检查一下

• 输入 cd 命令
->发生了什么?

• 重复第 2 步的动作 2 次
->发生了什么?

• 显示目录的内容

输入命令 cd root
->发生了什么?
->你进入了哪个目录?

• 重复第 4 步

你知道另一种可能的方法来查看你所在的目录吗?

2.5.4. Files

• 更改目录到/ 然后至 etc。使用 ls 命令；如果输出比你的窗口还长，则拉长窗口，或者试试 Shift+PageUp 和 Shift+PageDown.

文件 inittab 包含了列表中第一个问题答案。使用 file 命令查看它的属性。

-> inittab 的文件类型是 .....

• 使用命令 cat inittab 并读取该文件的内容

-> 你的计算机的默认模式是什么?

• 使用 cd 命令返回到你的 home 目录。

• 输入命令 file

-> 这条命令能查找到文件"."的意义吗?

• 你能使用 cat 命令查看到"."问吗?

• 用—help 选项显示 cat 命令的帮助。使用计算输出行数的选项来计算 /etc/passwd 文件列出的 用户个数

2.5.5. 获取帮助

- 查看 man intro
- 查看 man ls
- 查看 info passwd
- 输入 apropos pwd 命令
- 对 cd 命令试试 man 或 info

-> 你如何查找更多的关于 cd 命令的信息?

- 读 ls --help 并试验一下.

第 3 章. 文件和文件系统

通过第 2 章的学习，我们准备更详细的讨论 Linux 系统的文件和目录。很多 Linux 用户觉得 Linux 系统难用是由于他们缺少哪种数据存储在哪些地方的整体认识。我们将会尽量说明文件系统的文件组织架构。

我们将列举最重要的文件和目录，使用不同的方法来查看这些文件的内容，并学习如何创建，移动和删除文件和目录。

在完成本章的练习之后，你将能够了解：

- 描述 Linux 文件系统的设计
- 显示和设置路径
- 描述最重要的文件，包括 Kernel 和 Shell
- 查找隐藏文件
- 创建，移动和删除文件和目录
- 显示文件的内容
- 了解并使用不同的链接类型
- 找出文件属性并更改文件权限

3.1. Linux 文件系统概览

3.1.1. 文件

3.1.1.1. 概述

UNIX 系统的一个简单描述，同样也适用于 Linux 系统，就是：

"在 UNIX 上，一切都是文件；如果哪个东西不是文件，那么它就是一个进程。"

这个描述是正确的，即使有一些特殊的文件可能不仅仅是普通文件(比如，指定的管道和套接字)，但是为了简单起见，任务一切都是文件是可以接受的广泛含义。Linux 系统里，类似于 UNIX 系统，文件和目录是没有区别的，因为目录只是包含了其他文件名的一种文件。程序，服务，文本，图像等等都是文件。输入和输出设备，以及其他的所有设备，根据系统定义都被认

为是文件。

为了有序的管理这些文件，人们总是把它们想象成位于硬盘上的一种类似于树形结构，就像我们所知的 MS-DOS 一样。较大的分支包含了更多的分支，末端分支包含树形结构的叶子节点或普通文件。现在开始我们使用这种树形图像，但是我们在后面发现为什么这并不是完全正确的图像。

3.1.1.2. 文件分类

大部分文件叫做规则文件； 它们包含普通数据，比如文本文件，可执行文件或是程序文件，输入到程序里或从程序里输出等。

当你在说 Linux 系统遇到的任何东西是文件时，注意有几个例外。

目录：包含一系列其他文件的文件。

特殊文件：用于输入和输出的一种机制。大部分特殊文件都位于 /dev 目录下，我们将在后面讨论它们。

链接 s：它使得文件或目录在系统文件树的不同部分可见。我们将详细讨论链接。

(域)套接字：一种特殊的文件，类似于 TCP/IP 套接字，提供了受文件系统访问控制保护的进程间网络保护。

指定管道：类似于套接字，并形成一种进程间通信的方式，不需要使用网络套接字语义。

Ls 命令的-l 选项使用输出行的第一个字符显示出文件的类型：

```
jaime:~/Documents> ls -l
total 80
-rw-rw-r--1 jaime jaime 31744 Feb 21 17:56 intro Linux.doc
-rw-rw-r--1 jaime jaime 41472 Feb 21 17:56 Linux.doc
drwxrwxr-x 2 jaime jaime 4096 Feb 25 11:50 course
```

下面这个表总体描述了文件类型的特征：

Table 3-1. 文件类型

符号	含义
-	普通文件
d	目录
l	链接
c	特殊文件
s	套接字
p	管道
b	块设备

为了不一直使用长长的清单来查看文件类型，很多系统不是仅仅使用 ls，而是使用 ls -F 命令，它能在文件名后面添加"/=*|@"以显示文件的类型。为了方便刚入门的用户，-F 和-颜色选择通常组合起来使用，参见 Section 3.3.1.1。我们将在整篇文档中使用 ls -F 以获得更好的可读性。

作为用户，你只需要直接处理平面文件，可执行文件，目录和链接文件。特殊类型的文件是为了系统处理你特殊的需求的，并且通常由系统管理员和编程人员来操作。

现在，在我们学习重要文件和目录之前，需要先了解一下分区的作用。

3.1.2. 关于分区

3.1.2.1.为什么要分区?

很多人对分区是什么的认识比较含糊，因为每个操作系统都有创建和删除分区的能力。Liunx 在同一个物理磁盘上使用多个分区看起来可能有些奇怪，即使使用的是标准安装，所以需要加以解释一下。

不同分区的其中一个目标是在灾难中获得更高的数据安全性。通过将硬盘分区，数据能够聚集和分离。当事故发生的时候，只有事故分区的数据会丢失，而其他分区的数据则很可能会幸存下来。

当 Linux 没有使用基于日志的文件系统并且遇到了断电情况，这将导致灾难发生。基于安全和健壮性的原因来分区，这样系统的一部分损坏就不会导致整台计算机的损坏。这就是现今使用分区最主要的原因。一个简单的例子：一个用户创建了一个脚本，一个程序或是一个 web 应用来填满整个硬盘。如果硬盘只包括一个大的分区，整个系统在硬盘满了之后将会停止运作。如果这个用户将数据存储在别的分区，那么这个只会影响该数据分区，而系统分区和其他分区则会继续保持正常运作。

请记住使用基于日志的文件系统只能在断电和突然与存储设备断开的情况下提供数据保护。它并不会提供文件系统的坏块和逻辑错误保护。出现这种错误情况，你应该使用 RAID(冗余磁盘阵列)解决方案。

3.1.2.2. 分区设计和类型

Linux 系统有两类主要的分区：

数据分区：正常的 Linux 系统数据，包括根分区 ，它包含了启动和运行系统的所有数据。

交换分区：计算机物理内存的扩展，位于磁盘的额外内存空间。

大部分的系统包含一个根分区，一个或多个数据分区，一个或多个交换分区。在混合环境中的系统可能还包含其他系统数据的分区，比如使用 FAT 或是 VFAT 文件系统来存储 MS Windows 数据的分区。

大部分的 Linux 系统在安装时使用 fdisk 来设置分区的类型。可能你从第 1 章的练习中已经注意到，这经常自动发生。然后，在某些偶然情况下，你可能没有这么幸运。在这种情况下，你将需要手动的选择分区类型并手动进行磁盘分区。标准的 Linux 分区可以有 82 个交换分

区和 83 个数据分区，这些分区可以是基于日志的(ext3) 或常规的(ext2, 老的文件系统)。fdisk 工具设计有内含的帮助，你大可忘记这些具体值。

除了这两种类型，Linux 还支持大量的其他类型的文件系统，比如 Reiser 文件系统，JFS，NFS，FATxx 以及很多其他的(私有的)操作系统的文件系统。

标准的根分区(以一个单斜杠标示，/)大约有 100-500MB 大小，包含了系统配置文件，大部分的基本命令和系统程序，系统链接库，一些临时空间和管理员帐户的 home 目录。一个标准类型的安装需要 250MB 的根分区。

交换空间(用 swap 说明) 只供系统自己使用，在正常操作中被隐藏起来。交换分区类似于 UNIX 系统的交换分区，保证了不管发生什么系统都可以持续的运作。在 Linux 上，由于有了这块额外的内存，你不会看到类似于内存溢出，请关闭一些应用再重试这样气人的消息。交换分区或叫虚拟内存已经在 UNIX 系统以外的很多操作系统被采用很长时间了。

使用磁盘上的内存天然的比计算机上的真实物理内存要来得慢，但是有这个额外的内存却能极大的缓和系统内存使用。我们将在讨论 Chapter 4 时对交换分区进一步的了解。

Linux 系统极大的依赖于位于磁盘上的两倍于物理内存大小的交换分区。当安装一个系统的时候，你必须做到如何处理这种事情。比如有一个系统的 RAM 有 512 MB 大小：

第一种可能：一个有 1 GB 大小的交换分区

第二种可能：两个 512 MB 大小的交换分区

第三种可能：有两块磁盘，每块磁盘上有一个 512MB 大小的交换分区

最后一种选择在高 I/O 负载的情况下会得到最好的性能。

根据特定需求阅读相应软件的文档。一些应用软件，比如数据库，可能就需要更多的交换空间。其他的，比如手持系统，由于缺少磁盘可能根本就没有交换空间。交换空间可能还依赖于你内核的版本。

内核在很多版本中是位于独立的分区的，这是因为它是你的系统最重要的文件。如果是这样的话，你可以发现你还有一个/boot 分区，里面存放着你的内核文件和相关数据文件。

其余的磁盘通常被分作数据分区，所有的非系统核心数据可能位于一个分区，比如你在配置一个标准的工作站的安装。当非系统核心数据被分别存放在不同的分区时，通常按照如下的设置模式：

一个分区用于用户程序(/usr)

一个分区包含用户的个人数据信息(/home)

一个分区存储打印和邮件请求这个的临时数据 (/var)

一个分区用于存放第三方和额外的软件(/opt)

当分区被设置好之后，你只能去增加它了。变更已有分区的大小和属性是可能的但并不推荐。

磁盘划分为分区取决于系统管理员。在大型系统中，他或她可能使用恰当的软件来跨多个磁盘设置为一个分区。很多发行版本针对工作站(一般用户)和普通服务器的用途安装标准的分

区，但是也可以客户化分区。在安装过程中你可以使用系统特定的工具或 fdisk 定义你自己的分区设计方案，系统特定工具通常有图形界面，而 fdisk 则是一个创建分区并设置其属性的文本工具。

工作站或客户化的安装主要的用途是一个人和相同一个人使用。选择的安装软件反映了这种用途并打包成一个通用的用户包，比如友好的桌面主题，开发工具，E-mail 的客户程序，多媒体软件，网络和其他服务。所有东西放在同一个大分区中，交换分区设置为 RAM 的两倍大小，这样一般的工作站就完成了，它为个人使用数据提供了最大的磁盘空间，但是不利的地方是当发生问题的时候会缺少数据的一致性。

在服务器上，系统数据应该和用户数据分离。提供服务的程序与这些服务涉及的数据存储于不同的地方。在这种类型的系统上需要创建不同的分区：

- 一个分区存放启动机器的所有比需的数据。
- 一个分区存放配置信息和服务程序。
- 一个或多个分区存放服务器数据，比如数据库表，用户邮件和 ftp 归档等。
- 一个分区存放用户程序和应用。
- 一个或多个分区存放用户特定的文件(home 目录)
- 一个或多个交换分区(虚拟内存)

服务器通常拥有更多的内存，因此也就有更多的交换空间。某些服务器进程，比如数据库进程，可能需要比平常更多的交换空间；查看指定的文档以了解详细信息。为了获得更好的性能，交换空间通常被分成不同的交换分区。

3.1.2.3. 挂载点

所有的分区都是通过挂载点挂载到系统上的。挂载点指定了在文件系统上存放一个特定数据集的地方。通常，所有的分区都是通过根分区连接的。根分区，使用斜线表示(/)，目录就创建在它上面。这些空目录将是挂载到它上面的分区的挂载点。例如：给定含有以下目录的一个分区：

videos/cd-images/pictures/

我们要将这个分区挂载到文件系统上一个叫/opt/media 的目录上。为了实现这个目标，系统管理员必须确认目录/opt/media 存在。最好，它是一个空目录。这个步骤如何实现将在这章的后面部分解释。然后，使用 mount 命令，管理员就可以将分区挂载到系统上了。当你查看先前的空目录/opt/media ，你就会看到它包含挂载分区(磁盘或磁盘分区，CD, DVD, 闪存卡，USB 或其他存储设备)的文件和目录。

当系统启动时，所有的分区都将被挂载到系统上，就像文件/etc/fstab 所描述的那样。一些分区没有默认的挂载到系统上，如果它们没有经常连接进系统，比如你的数码相机的存储卡。如果配置正确，设备在系统发现连接后就会被挂载上，或者它也能由用户手动挂载。你在挂载和卸载设备的时候并不需要管理员帐号。在 Section 9.3.就有一个示例。

在系统运行时，分区和挂载点的信息可以通过使用 df 命令(代表 disk full or disk free)显示出来。在 Linux 上，df 命令属于 GNU 版本所有，并且它支持 -h 或 human readable 选项，这极大提高了用户的可读性。 注意到商业 UNIX 机器通常有它们自己版本的 df 和很多其他

命令。它们的功能通常相同，可是 GNU 版本的命令通常有更多更好的特性。

df 命令只显示活动的非交换分区的分区信息。这些分区可以包含其他网络系统的分区，就像下面这个例子，其 home 目录挂载到了网络上的一个文件系统，这种情况在企业环境里经常能够遇到。

```
freddy:~>
df-h
Filesystem    Size     Used        Use% Mounted on
                       Avail
/dev/hda8     496M     183M  288M  39% /
/dev/hda1     124M     8.4M  109M  8% /boot
/dev/hda5     19G      15G   2.7G  85% /opt
/dev/hda6     7.0G     5.4G  1.2G  81% /usr
```



```
/dev/hda7     3.7G     2.7G  867M  77% /var
fs1:/home     8.9G     3.7G  4.7G  44%
                                   /.automount/fs1/root/home
```

3.1.3. 更多的文件系统形式

3.1.3.1. 图形化

为了方便，Linux 文件系统经常被想象成树状结构。在标准的 Linux 系统上，你将发现文件系统的设计与图 3-1 所示的类似。

这是 RedHat Linux 系统的一种文件系统形式。根据系统管理，操作系统和 UNIX 机器的任务，这个结构可以多种多样，而且目录也可以随意的删除和增加。它们的名字甚至并非必须的，它们只是一个约定。

文件系统的树形结构以斜线(/)作为起点。这个目录，包含了所有的底层目录和文件，也被叫做根目录或文件系统的”根”。

在根目录下一层的目录通常在其前面加上斜线，以指示它们所处的位置同时预防与其他同名目录的冲突。当启动一个系统后，去查看根目录的内容是一种好的习惯。让我们来看看你能看到什么：

```
emmy:~> cd /
emmy:/> ls
bin/ dev/ home/ lib/ misc/ opt/ root/ tmp/ var/
boot/ etc/ initrd/ lost+found/ mnt/ proc/ sbin/ usr/
```

图 3-1. Linux 文件系统形式

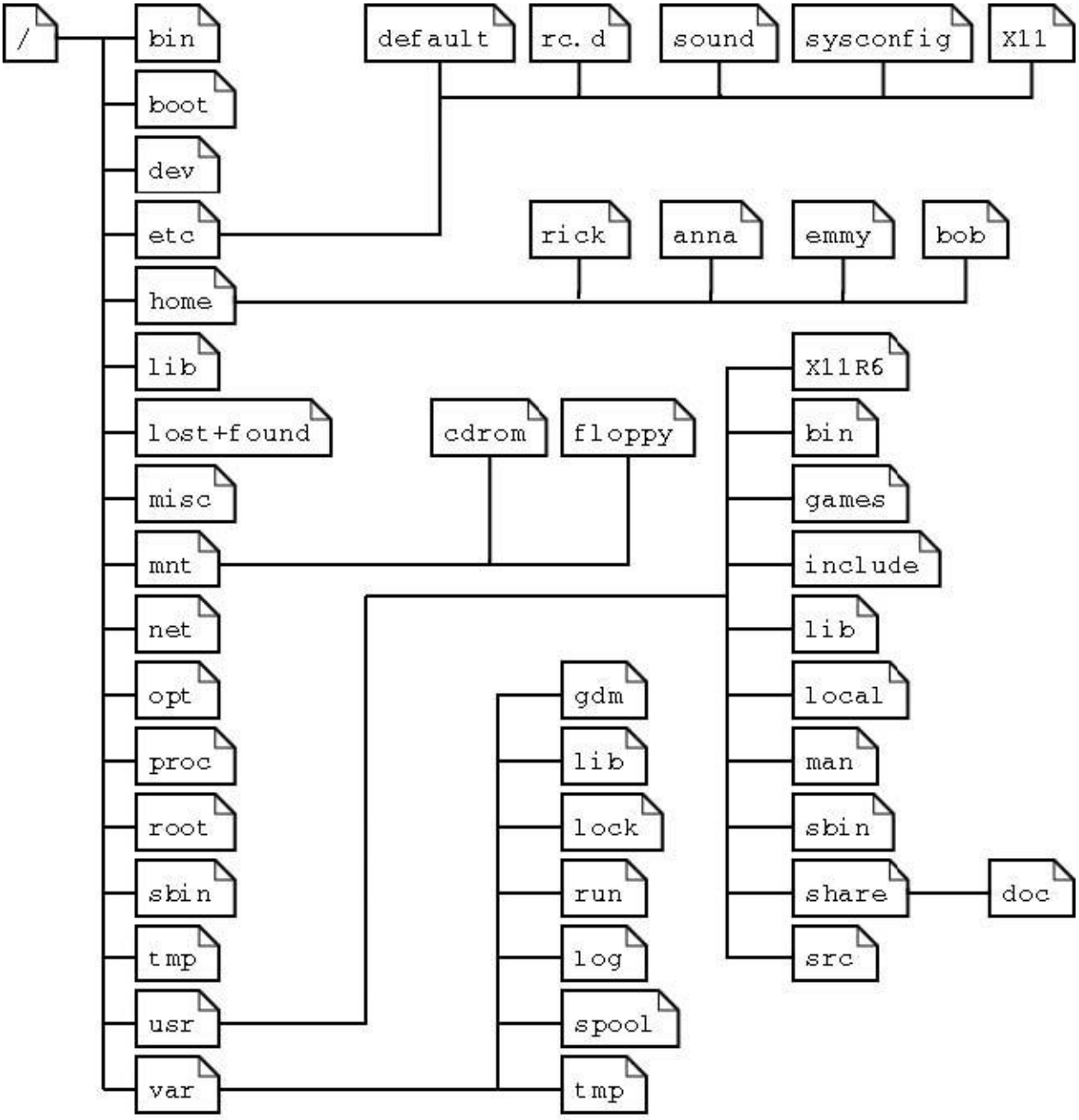


Table 3-2. 根目录的子目录

目录	内容
/bin	Common programs, shared by the system, the system administrator and the users
/boot	The startup files and the kernel, vmlinuz. In some recent distributions also grub data. Grub isthe GRand Unified Boot loader and is an attempt to get rid of the many different boot-loaders we know today.
/dev	Contains references to all the CPU peripheral hardware, which are represented as files with special properties.

/etc	Most important system configuration files are in /etc, this directory contains data similar to those in the Control Panel in Windows
/home	Home directories of the common users.
/initrd	(on some distributions) Information for booting. Do not remove!
/lib	Library files, includes files for all kinds of programs needed by the system and the users.
/lost+found	Every partition has a lost+found in its upper directory. Files that were saved during failures are here.
/misc	For miscellaneous purposes.
/mnt	Standard mount point for external file systems, e.g. a CD-ROM or a digital camera.
/net	Standard mount point for entire remote file systems
/opt	Typically contains extra and third party software.
/proc	A virtual file system containing information about system resources. More information about the meaning of the files in procis obtained by entering the command man procin a terminal window. The file proc.txtdiscusses the virtual file system in detail.
/root	The administrative user's home directory. Mind the difference between /, the root directory and /root, the home directory of the root user.
/sbin	Programs for use by the system and the system administrator.
/tmp	Temporary space for use by the system, cleaned upon reboot, so don't use this for saving any work!
/usr	Programs, libraries, documentation etc. for all user-related programs.
/var	Storage for all variable files and temporary files created by users, such as log files, the mail queue, the print spooler area, space for temporary storage of files downloaded from the Internet, or to keep an image of a CD before burning it.

你如何来发现一个目录是位于哪个分区上的呢？使用 df 命令后接点(.)选项来显示现在的目录所处的分区，并告知这个分区已经使用的空间数量：

```
sandra:/lib> df -h .
Filesystem Size Used Avail Use% Mounted on
/dev/hda7 980M 163M 767M 18% /
```

一般来说，根目录下的所有目录都是在根分区上的，除非在 df 命令显示的列表下该目录有一个单独的入口。

更多信息请参考 man hier。

3.1.3.2. 实际的文件系统

对大部分的用户和平常的系统管理任务来说，认为文件和目录是按照树形机构来组织的已经足够了。但是，计算机，并不能理解树或树形结构的含义。

每一个分区都有它自己的文件系统。设想所有的这些文件系统统一在一起，你可以把整个

系统想象成一个树形结构，但是事实上它并非如此简单。在文件系统里，一个文件表示成一个inode，它是一种序列号包含了构成文件实际数据的信息：这个文件属于谁，以及在磁盘上的存储位置等。

每一个分区都有其自己的一系列 inodes；在有多个分区的整个文件系统上，有相同 inode 号的文件也能并存。

每一个 inode 均描述了磁盘上的一个数据结构，该结构存储文件的属性，包括文件数据的物理位置。当一个磁盘初始化以存储数据时，通常在系统安装过程中或是当添加额外磁盘到现有系统上时，每个分区上一个确定数目的 inode 就会被创建。Inode 的数目将是磁盘可同时容纳文件(目录文件，特殊文件，链接文件等)的最大数目。我们通常设计一个 inode 管理存储上的 2-8Kb 大小。

在文件刚创建的时候，它会得到一个空闲的 inode。其中包含了如下的相关信息：

文件所有者和组所有者

文件类型

文件的权限

创建，最后一次读和更改的日期和时间

Inode 信息更改的日期和时间

文件的链接引用数目

文件大小

文件数据的实际存放地址

Inode 唯一不包含的信息是文件名和所在目录。这些目录存储于特定的目录文件中。比较文件名和 inode 号后，系统就能构建一个用户理解的树形结构了。用户可以使用 ls 加-i 选项来显示 inode 号。Inodes 在磁盘上有它们自己的存储空间。

3.2. 文件系统的定位

3.2.1. 路径

当你需要系统执行一条命令的时候，你不需要对这个命令指定完整的路径名。例如，我们知道 ls 命令在/bin 目录下(用 which -a ls 查看)，我们并不需要输入/bin/ls 来显示当前目录下的所有内容。

PATH 环境变量会关联这些命令到具体目录上。它能列举出可执行文件所在的系统目录，因此节省了用户大量的记住并输入命令所在具体目录的精力。所有 PATH 变量自然的包含了大量的含有 bin 的目录，就像如下用户验证的那样。Echo 命令可以用来显示 PATH 变量的内容：

```
rogier:> echo $PATH /opt/local/bin:/usr/X11R6/bin:/usr/bin:/usr/sbin/:/bin
```

在这个例子中，当需要制定程序时就会在目录/opt/local/bin, /usr/X11R6/bin, /usr/bin, /usr/sbinand /bin 中进行搜索。只要有合适的程序被找到，搜索就会停止，即使

没有搜索全部的目录。这可能导致奇怪的现象。在下面的第一个例子中，用户知道有一个程序 sendsms 可以用于发送 SMS 信息，而且系统中的另一个用户也可以使用它，但是他却不能用这个命令。区别就在于 PATH 变量的配置上：

```
[jenny@blob jenny]$ sendsms
bash: sendsms: command not found
[jenny@blob jenny]$ echo $PATH
/bin:/usr/bin:/usr/bin/X11:/usr/X11R6/bin:/home/jenny/bin
[jenny@blob jenny]$ su - tony
Password:
tony:~>which sendsms
sendsms is /usr/local/bin/sendsms


tony:~>echo $PATH
/home/tony/bin.Linux:/home/tony/bin:/usr/local/bin:/usr/local/sbin:\
/usr/X11R6/bin:/usr/bin:/usr/sbin:/bin:/sbin
```

注意 su (变换用户)工具的使用方法，它允许你使用另一个用户的 shell 环境变量，只要你知道该用户的登录密码。

一个反斜线表示继续到下一行的内容，而不需要 Enter 来分隔行。

在下面的这个例子中，一个用户想要使用 wc(字数统计)命令来检查文件的行数，但是没有任何反应，他必须使用 Ctrl+C 组合键来终止该程序：

```
jumper:~> wc -l test


(Ctrl-C)
jumper:~> which wc
wc is hashed (/home/jumper/bin/wc)


jumper:~> echo $PATH
/home/jumper/bin:/usr/local/bin:/usr/local/sbin:/usr/X11R6/bin:\
/usr/bin:/usr/sbin:/bin:/sbin
```

which 命令显示在用户的 home 目录下有一个 bin 目录，该目录包含了 wc 程序。因为这个 wc 程序在该用户的 home 目录下首先被搜索到，因此这个”自身的”程序就被执行了，但由于输入参数它并不理解，导致我们必须停止它。要解决这个问题有几种方法(通常在 UNIX/Linux 中要解决一个问题都有几种方法)：一种方案是重命名该用户的 wc 程序，另一种方案是给出这个命令的完整路径名，该路径名可以使用带-a 选项的 which 命令得到。

如果用户频繁使用其他目录的程序，他可以更改 path 变量将自己的目录放在最后：

```
jumper:~> export PATH=/usr/local/bin:/usr/local/sbin:/usr/X11R6/bin:\
/usr/bin:/usr/sbin:/bin:/sbin:/home/jumper/bin
```

 变更没有永久生效!

注意当在 shell 里使用 export 命令设置变量时，这个变更是临时的并只对当前会话生效(直到你退出会话)。新开一个会话，即使当前的会话依旧运行，在新会话的 path 变量也不会反映已经做的更改。我们将在 Section 7.2 看到我们如何使这种类型的变更永久的保存，使用的方法是在一个 shell 配置文件上添加这些参数。

3.2.2. 绝对路径和相对路径

路径，就是你在文件系统的树形结构里查到到一个给定文件的途径，能从树的主干(/或 root 目录)起始。这种情况下，路径起始于斜线/，被称为绝对路径，因为那不会产生错误：系统上只有一个文件在那里。

在另一种情况，路径不是以斜线/开始并且可能会产生冲突，比如前面的例子~/bin/wc(在用户的 home 目录下)和 bin/wc 在/usr 目录下就有可能冲突。不以斜线/开头的路径经常叫做相对路径。

在相对路径下，我们也用. 和.. 来表示当前目录和父目录。几个实际的例子：

• 当你在编译源代码的时候，安装文档经常指示你运行./configure 命令，表示在当前目录运行 configure 程序(新代码产生的地方)，而不是系统其他目录的 configure 程序。

• 在 HTML 文件上，经常使用相对路径来使得页面移动到别的地方更容易：

```
<img alt="Garden with trees" src="../images/garden.jpg">
```

再一次注意两者的区别：

```
theo:~> ls /mp3
ls: /mp3: No such file or directory
theo:~>ls mp3/
oriental/ pop/ sixties/
```

3.2.3. 至关重要的文件和目录

3.2.3.1. 内核

内核是系统的核心。它与底层的硬件和外围设备通信。内核还保证进程和后台进程(系统进程)在正确的时间启动和停止。内核还有很多其他重要的任务，因为众多以致在这一领域有一个特殊的内核开发邮件列表，在这个邮件列表里会共享大量的信息。详细讨论内核可能会使我们偏离主题。从现在开始，我们应该知道内核文件是 Linux 系统最重要的文件。

3.2.3.2. The shell

3.2.3.2.1. 什么是 shell?

当我在寻找一种恰当的方式来解释 shell 概念的时候，我发现这比我预期的难。有各种各样的定义，从简单的比对”shell 犹如车的方向盘”，到 Bash 手册里面对 shell 的模糊定义”bash 是兼容 sh 的命令解释器”，后者更模糊的解释”shell 管理系统与用户的交换”。Shell 更比这些描述的做得更多。

Shell 能比做与计算机通信的一种方式，一种语言。很多用户知道其他语言，如桌面系统

的点击类型的语言。但是这种语言计算机主导着会话，而用户被动的接受任务。让编程者在 GUI 格式下包含命令的所有选项和可能的使用方式很困难。因此，GUIs 比命令或后端指令的功能来的弱。

另一方面，shell 是与计算机交流的一种高级方式，因为它允许双向通信并启动发起会话。会话双方的地位是平等的，因此新想法可以被方便的测试。Shell 允许用户以一种灵活的方式管理系统。一个额外有用的性质是 shell 允许任务自动运行。

3.2.3.2.2. Shell 类型

就像人们懂得不同的语言和方言，计算机也懂得不同的 shell 类型：

sh or Bourne Shell: 是一种最初的 shell，现在仍然用在 UNIX 系统和 UNIX 相关的环境上。这是最基本的 shell，特性比较小的一种小程序。在 POSIX 兼容的模式下，bash 将仿效这种 shell。

bash or Bourne Again SHell: 这是标准的 GNUshell 类型，直观并且灵活。可能是新手最有用的，同时也是高级用户的一种强大的工具。在 Linux 上，bash 是普通用户标准的 shell 类型。它是 Bourne shell 所谓的超集，有一系列的附件和插件。这说明 Bourne Again SHell 与 Bourne shell 是兼容的：能在 sh 使用的命令，也能在 bash 中运用。然后，反过来就不一定成立了。本书的所有例子和练习都是使用 bash。

csh or C Shell: 这种 shell 的语义类似于 C 编程语言。有时程序员会使用这种 shell。

tcsh or Turbo C Shell: C shell 的一个超集，增强了用户的友好型和速度。

ksh or the Korn shell: 有时被具有 UNIX 使用背景的用户所赞赏。它是 Bourne shell 的超集，对初学者其配置可能是一个梦魇。

文件/etc/shells 给出了 Linux 已知的 shell 类型：

```
mia:~> cat /etc/shells
/bin/bash
/bin/sh
/bin/tcsh
/bin/csh
```


假的 Bourne shell

注意/bin/sh 经常链接到 Bash，当这个调用的话可以使用 Bourne shell 兼容的方式运行。你的默认 shell 在文件/etc/passwd 里面设置，就像用户 mia 这样：

```
mia:L2NofqdlPrHwE:504:504:Mia Maya:/home/mia:/bin/bash
```

为了从一种 shell 类型切换到另一个类型，只需要在终端窗口键入该 shell 的名称。系统就会自动查找 PATH 变量设置目录名称，由于 shell 是一个可执行文件(程序)，当前 shell 启动并执行它。由于每种 shell 都有它特定的外观，所以将会呈现一个新的提示：

```
mia:~> tcsh
```

[mia@post21 ~]$

3.2.3.2.3. 我使用的是哪一种 shell?

如果你不知道你使用的是哪种 shell，可以检查/etc/passwd 文件下你对应用户名的那一行，或者输入如下命令：

echo $SHELL

3.2.3.3. 你的 home 目录

你的 home 目录是你登录系统后默认到达的目录。很多情况下它是/home 的子目录，尽管这可能会不同。你的 home 目录有可能位于远程文件服务器的磁盘上；这种情况下你的 home 目录可能表示为 /nethome/your_user_name。另外某些情况，系统管理员可能采用比较难懂的设置，你的 home 目录可能位于/disk6/HU/07/jgillard。

不管你的 home 目录的路径如何，你并不需要过份在意它。正确的 home 目录路径存储在 HOME 环境变量上，以防止某些程序需要它。使用 echo 命令你就能正确的获得这个变量的值：

orlando:~> echo $HOME
/nethome/orlando

你能在你的 home 目录下做任何事情。你能在你想要的尽可能多的文件在里面，尽管数据和文件的总量受到限制，这些限制来自硬件和分区的大小，有时候还可能是系统管理员应用了空间额度限制。当磁盘空间很昂贵的时候设置磁盘空间使用是一种通用的方式。现在，空间限制在大型环境下是不推荐使用的。使用 quota 命令你能自己查看如果你被限制了磁盘使用空间：

pierre@lamaison:/> quota -v
Diskquotas for user pierre (uid 501): none

万一额度被设置了，你可以查看被限制的分区和它们的特定限制参数列表。在宽限期内超额使用空间可能会被接受，该期限只有很少或根本没有限制条件。使用 info quota 或 man quota 命令可以获得详细的信息。

 配额不够?

如果你的系统找不到 quota，那么文件系统就没有被应用了限制。 你的 home 目录由波浪线 (~)表示，它是对/path_to_home/user_name 的简写。在 HOME 变量里也存储了相同的路径，因此你不需要进行其他任何操作就能启用它。一个简单的例子：

从/var/music/albums/arno/2001 目录 切换到你的 home 目录下的 images 目录使用如下命令：

rom:/var/music/albums/arno/2001> cd ~/images

rom:~/images> pwd

/home/rom/images

在这章后面我们将讨论管理文件和目录的命令，以保持你的 home 目录规整。

3.2.4. 最重要的配置文件

我们在前面提到了，大部分的配置文件存储在/etc 目录下。 这些文件的内容可以使用 cat 命令查看，它将把文本文件的内容发送到你的标准输出(通常是你的终端)。具体语法如下：

cat file1file2... fileN

在这一部分我们将给出最常用配置文件的概览。这当然不是一个完整的列表了。增加额外的包将会在/etc 目录下相应的增加额外的配置文件。当阅读配置文件的时候，你会发现这些文件注释良好并不需加以额外的说明。一些文件也有包含额外文档的 man 手册，比如 man group

Table 3-3. 最常用的配置文件

File	Information/service
aliases	Mail aliases file for use with the Sendmail and Postfix mail server. Running a mail server on each and every system has long been common use in the UNIX world, and almost every Linux distribution still comes with a Sendmail package. In this file local user names are matched with real names as they occur in E-mail addresses, or with other local addresses.
apache	Config files for the Apache web server.
bashrc	The system-wide configuration file for the Bourne Again SHell. Defines functions and aliases for all users. Other shells may have their own system-wide config files, like cshrc.
crontaband the cron.* directories	Configuration of tasks that need to be executed periodically - backups, updates of the system databases, cleaning of the system, rotating logs etc.
default	Default options for certain commands, such as useradd.
filesystems	Known file systems: ext3, vfat, iso9660 etc.
fstab	Lists partitions and their mount points.
ftp*	Configuration of the ftp-server: who can connect, what parts of the system are accessible etc.
group	Configuration file for user groups. Use the shadow utilities groupadd, groupmod and groupdel to edit this file. Edit manually only if you really know what you are doing.

hosts	A list of machines that can be contacted using the network, but without the need for a domain name service. This has nothing to do with the system's network configuration, which is done in /etc/sysconfig.
inittab	Information for booting: mode, number of text consoles etc.
issue	Information about the distribution (release version and/or kernel info).
ld.so.conf	Locations of library files.
lilo.conf, silo.conf, aboot.confetc.	Boot information for the LInux LOader, the system for booting that is now gradually being replaced with GRUB.
logrotate.*	Rotation of the logs, a system preventing the collection of huge amounts of log files.
mail	Directory containing instructions for the behavior of the mail server.
modules.conf	Configuration of modules that enable special features (drivers).
motd	Message Of The Day: Shown to everyone who connects to the system (in text mode), may be used by the system admin to announce system services/maintenance etc.
mtab	Currently mounted file systems. It is advised to never edit this file.
nsswitch.conf	Order in which to contact the name resolvers when a process demands resolving of a host name.
pam.d	Configuration of authentication modules.
passwd	Lists local users. Use the shadow utilities useradd, usermod and userdel to edit this file. Edit manually only when you really know what you are doing.
printcap	Outdated but still frequently used printer configuration file. Don't edit this manually unless you really know what you are doing.
profile	System wide configuration of the shell environment: variables, default properties of new files, limitation of resources etc.

rc*	Directories defining active services for each run level.
resolv.conf	Order in which to contact DNS servers (Domain Name Servers only).
sendmail.cf	Main config file for the Sendmail server.
services	Connections accepted by this machine (open ports).
sndconfigor sound	Configuration of the sound card and sound events.
ssh	Directory containing the config files for secure shell client and server.
sysconfig	Directory containing the system configuration files: mouse, keyboard, network, desktop, system clock, power management etc. (specific to RedHat)
X11	Settings for the graphical server, X. RedHat uses XFree, which is reflected in the name of the main configuration file, XFree86Config. Also contains the general directions for the window managers available on the system, for example gdm, fvwm, twm, etc.
xinetd.*or inetd.conf	Configuration files for Internet services that are run from the system's (extended) Internet services daemon (servers that don't run an independent daemon).

通过这个指南后面我们将会更多的了解这些文件并对其中的一部分进行详细的学习。

3.2.5. 最常用的设备

设备，即 PC 机中除了 CPU 之外其他几乎所有的外围设备，都会作为/dev 目录下的一个入口显示在系统上。这种 UNIX 方式的设备处理方法，其中一个优点就是不管是用户还是系统都不需要担心设备的规格问题。

Linux 或 UNIX 的新手一般承受不起他们需要学习的大量的新名字和新概念。这也是在这部分的介绍包含一系列常用设备的原因。

Table 3-4. 常用设备

Name	Device
cdrom	CD drive
console	Special entry for the currently used console.
cua*	Serial ports
dsp*	Devices for sampling and recording

fd*	Entries for most kinds of floppy drives, the default is /dev/fd0, a floppy drive for 1.44 MB floppies.
hd[a-t][1-16]	Standard support for IDE drives with maximum amount of partitions each.
ir*	Infrared devices
isdn*	Management of ISDN connections
js*	Joystick(s)
lp*	Printers
mem	Memory
midi*	midi player
mixer*and music	Idealized model of a mixer (combines or adds signals)
modem	Modem
mouse(also msmouse, logimouse, psmouse, input/mice, psaux)	All kinds of mouses
null	Bottomless garbage can
par*	Entries for parallel port support
pty*	Pseudo terminals
radio*	For Radio Amateurs (HAMs).
ram*	boot device
sd*	SCSI disks with their partitions
sequencer	For audio applications using the synthesizer features of the sound card (MIDI-device controller)
tty*	Virtual consoles simulating vt100 terminals.
usb*	USB card and scanner
video*	For use with a graphics card supporting video.

3.2.6. 最常用的变量文件

在/var 目录下你将发现一系列存储可变数据的目录(与不经常改变甚至从不变更的 ls 程序或系统配置文件不同形成对比)。所有更改频繁的文件，比如日志文件，邮件，锁文件，打印文件等都保存在/var 的子目录下。

出于安全考虑，这些文件与主要的系统文件是区分开存储的，因此我们可以持续的监视并根据需要设置严格的权限。很多这种类型的文件也比平常需要更多的权限，比如/var/tmp 就要求任何用户都能写入。很多用户活动将在这边进行，可能还有连接到你的系统的网络上匿名的用户。这也是/var 目录及其所有子目录通常位于一个单独分区的原因。采用这种方法，比如受到邮件攻击才不至于文件系统的其他分区，这些分区包含了很多更重要的数据，比如你的程序

和配置文件。

 /var/tmp 和 /tmp

目录/tmp 下的文件在正常的系统活动或系统重启时可以被不经意的删除。在一些系统(客户化的)目录/var/tmp 的行为也是不可预知的。然而，这种情况并不是默认的行为，我们还是建议使用 /var/tmp 目录来保持临时文件。当不确定的时候，跟你的系统管理员进行确认一下。如果你管理你自己的系统，你就可以肯定的保证这个地方是安全的，只要你没有故意的修改目录/var/tmp 的设置(修改通常需要 root 用户，普通用户无法操作).

UNIX 系统上的其中一个安全系统，通常在 Linux 系统上也得到实现，就是日志保持功能，该功能记录了用户的动作，进程，系统事件等。叫做 syslogdaemon 的后台进程的配置文件决定了哪些以及多久这些日志信息将被保存。默认的日志保存地方是/var/log 目录，包含了访问日志，服务器日志，系统消息等不同的日志文件。

在/var 目录下，我们可以找到服务器数据，这些数据与关键数据比如服务器程序和它的配置文件等区别开。 一个典型的 Linux 系统例子是/var/www，它包含了包含了 web 服务器提供实际的 HTML 页面，脚本和图像。FTP 服务器的 FTP 结构(通过远程客户端能下载数据)也是被存放在/var 的其中一个子目录里的。 由于这些数据可以公开访问并经常被匿名用户更改，所有跟保存敏感数据的分区或目录分开，单独保存在这边才是安全。

在大多数的工作站安装中，/var/spool 至少包含 at 和 cron 目录，其中包含了预定的任务。在正式的环境中，该目录通常还包括 lpd 目录，该目录包含了打印队列和打印机的配置文件，以及打印机的日志文件。

在服务器系统我们通常设有目录/var/spool/mail，包含本来用户接收到的邮件，分类为一个用户一个相应的文件，即用户的”收件箱”。 一个相关的目录是 mqueue，是未送出邮件信息的打印区域。这些部分在处理大量用户的邮件系统可能会非常繁忙。新闻服务器也会使用 /var/spool 目录区域，因为它要处理巨大的信息量。

目录/var/lib/rpmd 特定于基于 RPM (RedHat 包管理器) 的发行版本；这个地方是 RPM 包存放信息的地方。其他的包管理器一般将数据存储与/var 的某个地方。

3.3. 操作文件

3.3.1. 查看文件属性

3.3.1.1. ls 的更多信息

除了文件名，ls 命令还给出了很多其他的信息，比如我们已经讨论过的文件类型。它也能显示文件的权限，文件大小，inode 号，创建的日期和时间，所有者和文件的链接数。在 ls 命令上加上-a 选项，隐藏文件也能被显示出来。有许多文件以点作为文件名的开始字符。一个典型的例子是你的 home 目录下的配置文件。当你使用一种系统达到一定的时间，你将会注意到有几十种文件和目录被创建但却没有自动显示在目录索引上。几乎每个目录都包含一个名字叫点(.)的文件和两个点(..)的文件，这两个文件结合它们的 inode 号可以觉得目录在文件系统树形机构中的位置。

你应该认真的去阅读下 ls 的 info 手册，因为它是附带有很多选项的常用命令。选择可以组合使用，就像大部分 UNIX 命令和它们的选项。一个常用的组合是 ls -al；它显示了一系列

文件以及它们的属性，以及链接指向的地址。 ls -latr 显示了相同的文件名，但是这次这些文件名是根据更改时间反向排序的，因此最近更改的文件位于列表的底端。这里有几个例子：

```
krissie:~/mp3> ls
Albums/ Radio/ Singles/ gene/ index.html


krissie:~/mp3> ls -a
./ .thumbs Radio gene/
../ Albums/ Singles/ index.html


krissie:~/mp3> ls -l Radio/
total 8
drwxr-xr-x 2 krissie krissie 4096 Oct 30 1999 Carolina/
drwxr-xr-x 2 krissie krissie 4096 Sep 24 1999 Slashdot/


krissie:~/mp3> ls -ld Radio/
drwxr-xr-x 4 krissie krissie 4096 Oct 30 1999 Radio/


krissie:~/mp3> ls -ltr
total 20
drwxr-xr-x 4 krissie krissie 4096 Oct 30 1999 Radio/
-rw-r--r--1 krissie krissie 453 Jan 7 2001 index.html
drwxrwxr-x 30 krissie krissie 4096 Oct 20 17:32 Singles/
drwxr-xr-x 2 krissie krissie 4096 Dec 4 23:22 gene/
drwxrwxr-x 13 krissie krissie 4096 Dec 21 11:40 Albums/
```

在大部分 Linux 版本中 ls 命令默认是 color-ls 的别名。这个特性允许你只适用 ls 而不附带其他选项就能查看文件类型。为了达到这样的目的，每个文件类型都有属于它们自己的颜色。标准的颜色方案位于目录/etc/DIR_COLORS 下：

Table 3-5. Color-ls default color scheme

Color	File type
blue	directories
red	compressed archives
white	text files
pink	images
cyan	links
yellow	devices
green	executables

flashing red	broken links

更多的信息查看 man 手册。在早期相同的信息是以在每个非标准文件名前加个前缀实现的。不需要颜色标示的使用形式(像打印一个目录列表)和一般的易读性，这些方案至今仍然可以使用：

Table 3-6. Default suffix scheme for ls

Character	File type
nothing	regular file
/	directory
*	executable file
@	link
=	socket
|	named pipe

命令 ls 的完整功能和特性可以参考 info coreutils ls.

3.3.1.2. 更多工具

为了我们正在处理的数据类型，我们可以使用 file 命令。应用某些测试来确认文件系统下的文件属性，数字和语言测试，file 命令可以有根据的推测出文件的格式类型。一些例子如下：

```
mike:~> file me+tux.jpg
me+tux.jpg: JPEG image data, JFIF standard 1.01, resolution (DPI),
"28 Jun 1999", 144 x 144
mike:~> file 42.zip.gz
42.zip.gz: gzip compressed data, deflated, original filename,
`42.zip', last modified: Thu Nov 1 23:45:39 2001, os: Unix
mike:~> file vi.gif
vi.gif: GIF image data, version 89a, 88 x 31
mike:~> file slide1
slide1: HTML document text
mike:~> file template.xls
template.xls: Microsoft Office Document
mike:~> file abook.ps
```

```
abook.ps: PostScript document text conforming at level 2.0
mike:~> file /dev/log
/dev/log: socket
mike:~> file /dev/hda
/dev/hda: block special (3/0)
```

命令 file 有一系列的选项，其中-z 选项可以查看压缩文件。查看 info file 以获得更详细的描述。谨记 file 命令产生的结果不是绝对的，它只是一种猜测。换句话说，file 命令有可能会被欺骗了。

 为什么对文件类型和文件格式这么大惊小怪？

简单的说，我们将讨论几个命令行工具用于查看平面文本文件。这些工具当作用于错误类型的文件时会出错。最坏的情况，它们将摧毁你的终端窗口并且/或者发出大量的杂音。如果你遇到这种情况，只需关闭这个终端会话并开始一个新的会话。但是应尽量避免这种情况，因为这样经常会吵到其他人。

3.3.2. 创建和删除文件和目录

3.3.2.1 制造垃圾...

... 这并不难做到。现在很多服务器是连接到网络的，因此自然的文件就从这台机器拷贝到另一台机器了。特别是在图形环境下使用，创建文件极其容易而且经常不需要用户的许可。为了证明这个问题，以下是一个新用户目录的完整内容，在标准的 RedHat 系统上创建的：

```
[newuser@blob user]$ ls -al
total 32
drwx------3 user user 4096 Jan 16 13:32 .
drwxr-xr-x 6 root root 4096 Jan 16 13:32 ..
-rw-r--r--1 user user 24 Jan 16 13:32 .bash_logout
-rw-r--r--1 user user 191 Jan 16 13:32 .bash_profile
-rw-r--r--1 user user 124 Jan 16 13:32 .bashrc
drwxr-xr-x 3 user user 4096 Jan 16 13:32 .kde
-rw-r--r--1 user user 3511 Jan 16 13:32 .screenrc
-rw-------1 user user 61 Jan 16 13:32 .xauthDqztLr
```

第一眼看上去，”使用的”home 目录内容看起来并不是很糟糕：

```
olduser:~> ls
app-defaults/ crossover/ Fvwm@ mp3/ OpenOffice.org638/
articles/ Desktop/ GNUstep/ Nautilus/ staroffice6.0/
bin/ Desktop1/ images/ nqc/ training/
```

brol/ desktoptest/ Machines@ ns_imap/ webstart/

C/ Documents/ mail/ nsmail/ xml/

closed/ Emacs@ Mail/ office52/ Xrootenv.0

但是当所有的以点开始命名的文件和目录也包含进来，在这个目录下就会有 185 个条目了。这是由于在用户的 home 目录下，大部分的应用程序有它们自己的目录和/或文件，这些文件包含了用户特定的设置。通常的这些文件在你第一次启动应用程序的时候被创建。在某些情况下，要创建一个不存在的目录可能会通知你，但是大部分情况下这些都是自动完成的。

而且，新文件可以被连续的创建，因为用户想要保持文件，保存文件的不同版本，使用网络应用，以及下载文件并保存到它们本地的机器上。这个过程不会停止。因此每个人都需要一个方案以明确地保持对事态的概览。

在下一节，我们将讨论我们保持有序性的方法。我们只讨论可用于 shell 的文本工具，因为图形工具很直观而且跟著名的点击类型的 MS Windows 文件管理器看起来一样，包括图形帮助功能和你期望从这类应用中获得的功能。下表就是最流行的 GNU/Linux 文件管理器的概述。大部分文件管理器可以从你的桌面菜单中启动，或者点击你的 home 目录图标，或者从命令行，发出这些命令：

nautilus: Gnome 中默认的文件管理器，GNU 桌面。使用这个工具的优秀的文档可以从 http://www.gnome.org 获得。

konqueror: 经常用于 KDE 桌面的文件管理器。其操作手册位于 http://docs.kde.org.

mc: Midnight Commander， Norton Commander 之后的 UNIX 文件管理器。 所有的文档可以从 http://gnu.org/directory/ 或其镜像网站 http://www.ibiblio.org 找到。

这些应用程序通常非常值得去试试而且会给 Linux 的初学者留下很深的印象，因为这里有多种多样的方式：这些知识最流行的目录和文件管理工具，还有很多其他的类似项目已经在发展中了。现在让我们找出内在因素并看看这些图形工具是如何使用普通的 UNIX 命令的。

3.3.2.2. 工具聚

3.3.2.2.1. 创建目录

保存事务在同一个地方的一种方法是通过创建目录和子目录(或者文件夹和子文件夹)保存这些文件在一个默认的位置。这个可以通过 mkdir 命令实现：

```
richard:~> mkdir archive
richard:~> ls -ld archive
drwxrwxrwx 2 richard richard 4096 Jan 13 14:09 archive/
```

一步同时创建目录和子目录可以使用-P 选项来实现：

```
richard:~> cd archive
```

```
richard:~/archive> mkdir 1999 2000 2001
richard:~/archive> ls
1999/ 2000/ 2001/
richard:~/archive> mkdir 2001/reports/Restaurants-Michelin/
mkdir: cannot create directory `2001/reports/Restaurants-Michelin/':
No such file or directory
richard:~/archive> mkdir -p 2001/reports/Restaurants-Michelin/
richard:~/archive> ls 2001/reports/
Restaurants-Michelin/
```

当这个新文件需要创建文件默认的权限之外的其他权限时，新的访问权限可以再次被设置，还是使用命令 mkdir，更多请参考 Info 手册。我们将在学习下一部分文件安全时讨论文件的访问方式。

目录的名字需要遵循规则文件名称命名的规则。一个最重要的限制是你在同一个目录下不同有两个相同名称的文件(但是谨记 Linux，类似于 UNIX，是一种大小写敏感的操作系统)。事实上文件名的长度没有限制，但是一般保持小于 80 个字符，这样它才适合终端窗口的一行。你可以在文件名上使用任何字符，尽管实际使用上应排除对 shell 有特殊含义的字符。当你对此迷惑时，请查看 Appendix C。

3.3.2.2.2. 文件移动

现在我们已经合理的构建了我们的 home 目录，因此是时候使用 mv 命令清理无类别的文件了： `richard:~/archive> mv ../report[1-4].doc reports/Restaurants-Michelin/`

当重命名文件的时候也可以使用这个命令：

```
richard:~> ls To_Do
-rw-rw-r--1 richard richard 2534 Jan 15 12:39 To_Do

richard:~> mv To_Do done

richard:~> ls -l done
-rw-rw-r--1 richard richard 2534 Jan 15 12:39 done
```

必须清楚这样只是更改了文件名。所有的其他属性都是保持不变的。

关于 mv 命令的详细的语法和特性可以在 man 或 info 手册上查到。当你面临一个问题的时候使用这个文档应该是你的第一反应。涉及问题的答案很可能就在系统文档里面。每一个有经验的用户都似乎每天阅读 man 手册的，因此初学者更应该一直阅读它。一段时间以后，你将知

道那些最常用命令的最常用的选项，但是你还是应该以这些文档作为注意的信息来源。注意到 HOWTOs，FAQs，man 手册和其他来源的信息正被慢慢的合并到 info 手册中，它也是现在最新的在线文档了。

3.3.2.2.3. 文件拷贝

使用 cp 命令拷贝文件和目录。一个有用的用于递归拷贝的选项(拷贝目录下所有的文件和目录)，使用 cp 附加-R 选项。一般的语法如下：

```
cp [-R] fromfiletofile
```

作为一个例子，用户 newguy，想要用户 oldguy 相同的 Gnome 桌面设置。解决这个问题的一种方法是拷贝用户 oldguy 的设置到用户 newguy 的 home 目录下。

```
victor:~> cp -R ../oldguy/.gnome/ .
```

在涉及文件权限上这种方法会产生一些错误，但是所有的这些错误与用户 newguy 不需要的私人文件有关。我们将在下一节讨论如何来更改这些权限，如果确实需要的话。

3.3.2.2.4. 删除文件

使用 rm 命令来删除单一的文件，用 rmdir 命令移除整个空目录(使用 ls -a 来确认目录是否为空)。 rm 命令也有选项来删除一个非空目录以及该目录下的所有子目录，参考 info 手册以了解这些危险的选项。

 How empty can a directory be?

通常情况下目录 . (点)和 .. (双点)不可以删除，因为空目录也需要它们来判断目录在文件系统等级中的级别。在 Linux 系统，就像 UNIX 系统一样，没有垃圾回收站 - 至少 shell 没有，尽管可以有很多其他方案可作为垃圾回收站。因此一旦被删除，文件就永远移除了，没有办法重新找回来除非你有系统备份，或者你足够快且具备非常好的系统管理能力。为了防止初学者失误，可以使用-i 选项来启动 rm, cp 和 mv 命令的交换模式。这样系统就不会在请求后立即操作。相反的它需要额外的键入 Enter 来形成这种操作：

```
mary:~> rm -ri archive/
rm: descend into directory `archive'? y
rm: descend into directory `archive/reports'? y
rm: remove directory `archive/reports'? y
rm: descend into directory `archive/backup'? y
rm: remove `archive/backup/sysbup200112.tar'? y
rm: remove directory `archive/backup'? y
rm: remove directory `archive'? y
```

我们将在 Chapter 7 讨论如何把这个选项设置为默认的，这将讨论客户化你的 shell 环境。

3.3.3. 文件查找

3.3.3.1. 使用 shell 特性

在移动文件的例子中你就看到了 shell 如何一次性操作多个文件。在那个例子中，shell 通过在放括号"[" 和 "]"之中设置必要条件自动找出用户想要的。Shell 能够数字范围并大写或小写类似的字符。它还能使用星号替代你所想要的大量的字符，以及用问号替代一个字符。所有类型的替代可以同时使用；shell 能合理的处理它们。比如 Bash shell 可以方便的处理类似于这样的表达式 ls dirname/*/*/*[2-3]。

在其他类型的 shell，星号通常用于减少用户输入的字符数：用户可以用 cd dir*来替代输入 cd directory。然而使用 Bash，这个就没必要了，英文 GNU shell 有一个特性就文件名自动完全匹配。就是说你可以输入命令(所有的)或文件(当前目录下的)的最开始的几个字符并且如果没有冲突，shell 将找出你所需要的。比如在一个包含很多文件的目录下，你能只需要输入 ls A 并按两次 Tab 键就能查看是否有文件以 A 字母开头，而不用按 Enter。如果只有一个文件以"A"开头，这个文件就会马上作为 ls 参数的命令(或者其他任何 shell 命令)。

3.3.3.2. Which

一个很简单的查找文件的方法是使用 which 命令，它在用户路径的目录下查找要求的文件。当然，由于查找路径只包含有可执行文件的目录，因此 which 不支持二进制文件。Which 命令在解决”命令找不到”的问题时很有用。在下面的例子中，用户 tina 无法使用 acroread 程序，但是他的同事在同一个系统里却没有问题。这个问题类似于前面的 PATH 问题：tina 的同事告诉他能够看见所需程序位于/opt/acroread/bin 目录下，但是这个目录却不在他的当前路径下：

```
tina:~> which acroread
/usr/bin/which: no acroread in (/bin:/usr/bin:/usr/bin/X11)
```

如果提供这个命令的完整路径就可以解决这个问题，或者重新设置 PATH 变量的以引入该路径：

```
tina:~> export PATH=$PATH:/opt/acroread/bin


tina:~> echo $PATH
/bin:/usr/bin:/usr/bin/X11:/opt/acroread/bin
```

使用 which 命令也可以检查一个命令是否是另一个命令的别名：

```
gerrit:~> which -a ls
ls is aliased to `ls -F --color=auto'
ls is /bin/ls
```

如果你的系统不能这么用，也可以使用 alias 命令获得别名信息：

```
tille@www:~/mail$ alias ls
alias ls='ls --color'
```

3.3.3.3. 查找并定位

当查找不是位于默认路径下的其他路径时，这就是真实有用的工具了。这个 find 工具，

继承自 UNIX，功能非常的强大，因此也导致了语法有点复杂。然后，GNU find 解决了语法的问题。这个命令不仅允许你查找文件名，它也能把文件大小，最后更改的日期和其他文件属性作为查找参数。最常使用的是查找文件名：

```
find <path>-name<searchstring>
```

它的含义是："查找给定路径下的所有文件和子目录，打印出文件名包含了搜索字符串的文件名"（不包含内容中匹配的）。

find 的另一个应用是查找某种大小的文件，就像下面的例子，用户 peter 想要找出当前目录或其子目录下的所有大小超过 5M 的文件：

```
peter:~> find . -size +5000k
psychotic_chaos.mp3
```

如果你仔细查看 man 手册，你还能发现 find 在查找到的文件上进行其他操作。常用的例子是删除该文件。最好最初不用使用 - exec 选项以查找到正确的文件，之后使用该选项重新运行一遍以删除找到的文件。如下，我们查找以.tmp 结尾的文件：

```
peter:~> find . -name "*.tmp" -exec rm {} \;

peter:~>
```

最优化!

这个命令将会多次调用 rm 只要查找到到符合条件的文件。在最坏的情况下，这可能有几千上万次。它将是系统的极大负担。一个更好的方法是使用管道(|)并且使用带有 rm 参数的 xargs 工具。这种方式下，rm 仅在命令完成后才被调用，而不是每次找到文件。参考 Chapter 5 以了解更多关于 I/O 重定向消除每天工作量的情况。

再后来(根据 man 手册是在 1999 年，在 find 出现 20 年后)，locate 被发展出来。这个程序更容易使用，但是比 find 有更多的限制。因为它的输出是基于文件索引数据库，而且每天只更新一次。另一方面，在 locate 数据库里的搜索比 find 使用更少的资源，因此能立刻的显示结果。

现在大部分的 Linux 版本使用 slocate，安全的 locate，现代的 locate 以防止用户访问到他们没有权限访问的目录。Roots 的 home 目录就是这方面的一个例子，这些目录不能被公开访问。一个用户想查找某个懂得 C shell 人只需发出 locate .cshrc 命令，就能显示所有拥有 C shell 的客户化配置文件的用户。假设用户 root 和 jenny 都运行 C shell，那么只有 /home/jenny/.cshrc 文件将会被显示，而不会显示 root's 的 home 目录下的那个 C shell 配置文件。在大部分系统上，locate 引用了 slocate 程序：

```
billy:~> ls -l /usr/bin/locate
lrwxrwxrwx 1 root slocate 7 Oct 28 14:18 /usr/bin/locate -> slocate*
```

用户 tina 能使用 locate 来查找他想要的应用：

```
tina:~> locate acroread
```

/usr/share/icons/hicolor/16x16/apps/acroread.png
/usr/share/icons/hicolor/32x32/apps/acroread.png
/usr/share/icons/locolor/16x16/apps/acroread.png
/usr/share/icons/locolor/32x32/apps/acroread.png
/usr/local/bin/acroread
/usr/local/Acrobat4/Reader/intellinux/bin/acroread
/usr/local/Acrobat4/bin/acroread

没有包含 bin 子目录的目录不能含有该程序-它们不含有可执行文件。因此只剩下三种可能的路径了。在目录/usr/local/bin 里面的文件正是 tina 想要的：它是启动该程序的实际 shell 脚本的链接：

```
tina:~> file /usr/local/bin/acroread
/usr/local/bin/acroread: symbolic link to ../Acrobat4/bin/acroread
tina:~> file /usr/local/Acrobat4/bin/acroread
/usr/local/Acrobat4/bin/acroread: Bourne shell script text executable
tina:~> file /usr/local/Acrobat4/Reader/intellinux/bin/acroread
/usr/local/Acrobat4/Reader/intellinux/bin/acroread: ELF 32-bit LSB
executable, Intel 80386, version 1, dynamically linked (uses
shared libs), not stripped
```

为了保持路径尽可能的短，这样可以使系统在用户需要执行一条命令时不需要搜索太长时间，我们添加目录/usr/local/bin 到路径中，而不是其他只包含特定程序二进制文件的目录，/usr/local/bin 目录还包含了其他有用的程序。

同样的，find 和 locate 的完整描述可以在 info 手册查到。

3.3.3.4. grep 命令

3.3.3.4.1. 一般的行过滤

一个简单但很强大的程序，grep 可以用于过滤输入行并返回某种模式给输出。Grep 在字面上有几千种应该程序。在下面的例子中，用户 jerry 使用 grep 来查看它使用 find 做了什么：

```
jerry:~> grep -a find .bash_history
find . -name userinfo
man find
find ../ -name common.cfg
```

 查找历史

bash 的搜索功能有很有用，该功能使用 Ctrl+R 启动，比如我们想要再次检查 find 到底做什么：

```
thomas ~> ^R
```

```
(reverse-i-search)`find': find `/home/thomas` -name *.xml
```

在你的搜索提示下输入你的搜索字符串。你输入的字符越多搜索的限制就越多。它读取 shell 会话的命令历史记录(当你退出会话后它被写入你的 home 目录的.bash_history 文件中)。当最近发生的并含有你的搜索字符将会被显示。如果你想要查看包含相同字符串的以前的命令，需要再次键入 Ctrl+R。

更多关于 bash 请查看 info 手册。

所有的 UNIX 使用者有一个在线目录。Linux 同样也有。该目录包含一系列位于 words 文件中的知名的单词，位于目录/usr/share/dict。为了快速的确认一个单词是否正确，并不需要图形化的应用程序：

```
william:~> grep pinguin /usr/share/dict/words


william:~> grep penguin /usr/share/dict/words
penguin
penguins
```

 字典 vs. 文字列表

一些版本提供 dict 命令，该命令比简单的在文字列表中查找文字提供了更多的功能。

在我下面的那个 home 目录的所有者是谁，请看，这是他的电话号码!

```
lisa:~> grep gdbruyne /etc/passwd
gdbruyne:x:981:981:Guy Debruyne, tel 203234:/home/gdbruyne:/bin/bash
```

Arno 的邮件地址是什么？

```
serge:~/mail> grep -i arno *
sent-mail: To: <Arno.Hintjens@celeb.com>
sent-mail: On Mon, 24 Dec 2001, Arno.Hintjens@celeb.com wrote:
```

find 和 locate 经常和 grep 一起组合使用来定义一些复杂的查询。想获得更多的信息，请查看关于 I/O 重定向的 Chapter 5

3.3.3.4.2. 特殊字符

对 shell 有特殊含义的字符必须被转换。在 Bash 里转换字符使用反斜线，其他 shell 也类似；这样就能把字符的特殊含义去除了。Shell 里有很多特殊字符，其中最常见的是/, ., ? 和 *。在 info 手册和 shell 文档里面可以查找到这些特殊字符的完整列表。

例如，如果你要显示文件"*" 而不是目录下的所有文件，你必须这样来使用:

```
less \*
```

当文件名包含空格时也可这么处理：

```
 cat  This\ File
```

3.3.4. 查看文件内容的更多方式

3.3.4.1. 常用方式

除了直接将文件输出到标准输出的 cat 命令之外，还有其他的工具可以查看文件内容。

最简单的方式是使用图形界面来查看文件而不是使用命令行工具。在简介部分我们已经看过了一种办公软件 OpenOffice.org。其他的例子有 GIMP (在命令行输入 gimp 来启动)，它是 GNU 图形处理工具；使用 xpdf 来查看 pdf 文件；用 GhostView (gv)查看日记脚本文件；用 Mozilla/FireFox，links (一种文本模式的浏览器),Konqueror，Opera 很有很多其他的工具来查看 web 内容；用 XMMS，CDplay 以及其他工具来查看多媒体文件内容；用 AbiWord，Gnumeric，Koffice 等查看办公软件。Linux 有数千个应用程序，要全部列举它们需要几天的时间。

我们将集中精力与 shell 或文本模式的应用软件，它们是其他应用工具的基础。这些命令对文本环境下的文本工作良好。当有疑问的时候，首先需要使用 file 命令检查下文件的类型。所有，让我们看看有哪些有用的文本工具可以用于查看文件的内容。

 字体问题

比如我们现在正在讨论的纯文本工具，在作用于”无格式”的文本文件的时候通常会遇到字符编码的问题。特殊的字符，比如重音符号的字面字符，中文字符和其他语言的字符，它们使用与 en_US 编码不同的编码方式，就会以错误的方式显示或生成不可取的垃圾字符。这些问题将会在 Section 7.4 讨论。

3.3.4.2. "less 就是 more"

毫无疑问当你工作在 UNIX 环境下，迟早会听到某人说起这个短语的。一些 UNIX 的历史说明了这点：

首先使用的是 cat。输出以一种不可控制的流方式输出。

然后就是 pg，这个命令依然可以在旧的 UNIX 系统中找到。这个命令将文本一次一页输出到标准输出。

more 是 pg 的改进版本。这个命令现在还可以在每个 Linux 系统上使用。

less 是 more 的 GNU 版本，它又额外的特性以允许高亮搜索字符串，回滚等。它的语法很简单：

```
less  name_of_file
```

更多的信息请参考 info 手册。

3.3.4.3. head 和 tail 命令

这两个命令分别显示一个文件的最初 / 最后几行。要查看最后 10 行内容只需输入：

```
tony:~> tail -10 .bash_history
locate configure | grep bin
```

```
man bash
cd
xawtv &
grep usable /usr/share/dict/words
grep advisable /usr/share/dict/words
info quota
man quota
echo $PATH
frm
```

head 的原理与其类似。tail 命令便利的特性可以在文件发生变更的时候持续的显示最后 n 行记录。-f 选项常常被系统管理员用于检查 log 文件。更多的信息请参见系统文档。

3.3.5. 链接文件

3.3.5.1. 链接类型

在我们更多的了解了文件系统的文件和文件表示方法后，理解链接(或快捷键)是一件很容易的事情了。链接只是一种两个或更多个文件名匹配到相同数据的一种方式。有两种方式来实现这种目标：

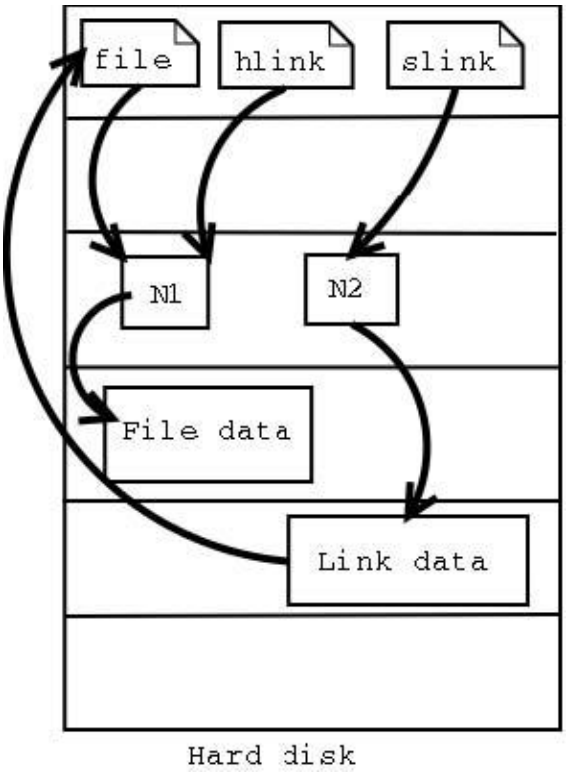
• 硬链接：用相同的 inode 关联两个或多个文件名。硬链接共享磁盘上的相同数据库，尽管在形式上它们表示为一个独立的文件。

这里就有一个不利因素了：硬链接不能跨分区，因为 inode 号在一个给定分区里是唯一的。

• 软链接或符号链接：一个指向其他文件的小文件。符号链接包含了目标文件的路径而不是磁盘的物理位置。由于这种方式方式没有使用 inode，所有软链接可以跨分区使用。

这两种链接表现类似，但是并不相同，表现在如下的示例：

Figure 3-2. 硬链接和软链接的机制



注意移除软链接的目标文件将无法继续使用这个链接。

每一个规则文件原则上是一个硬链接。硬链接不能跨分区，因为它们使用 inode 而且 inode 号在一个分区里具有唯一性。

可能我们还会议论第三种类型的链接，user-space 链接，它类似于 MS Windows 的快捷键。这种链接文件包含有只能被图形文件管理器解释的元数据。对内核和 shell 来说这些仅仅是普通文件。它们可能以.desktop 或 .lnk 为后缀。可以在~/.gnome-desktop 找到这样的例子：

```
[dupont@boulot .gnome-desktop]$ cat La\ Maison\ Dupont
[Desktop Entry]
Encoding=Legacy-Mixed
Name=La Maison Dupont
Type=X-nautilus-home
X-Nautilus-Icon=temp-home
URL=file:///home/dupont
```

下面是与 KDE 桌面相关的一个例子：

```
[lena@venus Desktop]$ cat camera
[Desktop Entry]
Dev=/dev/sda1
FSType=auto
```

```
Icon=memory
MountPoint=/mnt/camera
Type=FSDevice
X-KDE-Dynamic-Device=true
```

使用你的图形环境的特性来创建这种类型的链接非常容易。如果你需要帮助，你的系统文档是你首选的参考资料。

在下面这一节，我们将学习如何使用命令行来创建 UNIX 类型的符号链接。

3.3.5.2. 创建符号链接

符号链接对初学者来说特别有趣：它们显而易见而且不需要担心分区方面的问题。

创建链接的命令用 ln。为了创建符号链接，你需要使用 - s 选项：

```
ln  - s targetfile linkname
```

在下面的例子中，用户 freddy 在他 home 目录下的一个子目录里面创建了一个链接指向系统其他部分的一个目录：

```
freddy:~/music> ln -s /opt/mp3/Queen/  Queen

freddy:~/music> ls -l
lrwxrwxrwx 1 freddy freddy 17 Jan 22 11:07 Queen -> /opt/mp3/Queen
```

符号链接通常是非常小的文件，而硬链接跟原始文件有着相同的大小。

符号链接的使用非常广泛。它们通常可以用于节省空间，在安装时需要文件在另一个地点的情况下做文件拷贝，解决突发的需要在新环境下运行脚本的需求并能节省大量的工作量。系统管理员可能决定需要转移一个用户的 home 目录到一个新地点，比如 disk2，但是如果他想要所有的事情运转得跟原来一样，比如使用/etc/passwd 文件，最小的代价就是创建从/home 目录一个符号链接执行新的/disk2/home。

3.4. 文件安全

3.4.1.访问权限: Linux's first line of defense

Linux 的安全模式是基于 UNIX 系统使用的安全模式的，它遵循严格的已经被证明是健壮的 UNIX 安全模式(有时候甚至更强)。在 Linux 系统上，每个文件录属于一个用户和一个组用户。还有第三种类型的用户，这种用户不是文件的所有者也不是文件所有组的一员。类似于这样的用户，他们读，写和执行文件的权限可以被赋予或是拒绝。

我们已经能够使用附带 long 选项的 ls - l 命令来列举文件。这个命令会为这三种类型的用户显示文件权限属性；这些权限会表示为首个字符之后的九个字符，文件属性行的首个字符表示文件的类型。如下例子所示，代表文件权限的九个字符的前三个表示文件所有者对文件的访问权限。接下去的三个字符表示文件所有组的用户对该文件的访问权限，最后三个字符表示其他用户的访问权限。权限通常具有相同的顺序的：所有者，组和其他用户的读，写，执行权

限。几个例子如下：

```
marise:~> ls -l To_Do
-rw-rw-r--1 marise users 5 Jan 15 12:39 To_Do
marise:~> ls -l /bin/ls
-rwxr-xr-x 1 root root 45948 Aug 9 15:01 /bin/ls*
```

第一个文件是规则文件。用户 marise 或者属于组 users 的用户可以读和写(更改/移动/删除)文件，但是他们不能执行该文件。其他用户只允许读这个文件，而不能写或执行它.

第二个例子是关于可执行文件的，区别在于：每个人可以执行这个程序，但是你需要 root 用户才能去更改它。

在 info 手册里有对如何使用 ls 命令来显示文件访问权限的详细资料，查看 What information is listed 这一节的内容。

为了便于使用这些命令，访问权限或模式以及用户组都有一个相关的编码。请看下表。

Table 3-7. 访问模式代码

代码	意义
0 or -	没有授权，不能使用该位置的访问权限。
4 or r	读权限被授予这个位置的用户。
2 or w	写权限被授予这个位置的用户。
1 or x	执行权限被授予这个位置的用户。

Table 3-8. 用户组的代码

代码	意义
u	用户权限
g	组用户权限
o	其他用户权限

这种简单的设计方式会被严格的执行，即使无网络安全保护也能提供非常高可的安全性。使用其他功能，安全配置可管理程序的用户访问，它能根据对文件基础的认识为文件服务，并能包含类似于 home 目录和系统配置文件这样的敏感数据。

你应该知道你的用户名是什么。如果你不知道，使用 id 命令系统可以显示出你的用户名，以及你所属的默认组和其他的成员组。

```
tilly:~> id

uid=504(tilly) gid=504(tilly) groups=504(tilly),100(users),2051(org) Your user name is also stored in the environment variable USER:
tilly:~> echo $USER

tilly
```

3.4.2. 工具集

3.4.2.1. chmod 命令

应用严格的文件权限，它有时候甚至是件麻烦事，其中一个通常的后果是需要根据各种各样的原因修改文件的访问权限。我们使用 chmod 命令来实现这种需求，最后 to chmod h 成为了一个非常流行的英语单词，意思是更改一个文件的访问权限。chmod 命令可以搭配字符或数字选择，只有你觉得好用。

下面的例子使用字符选项来解决一个初学者通常会遇到的权限问题：

```
asim:~> ./hello
bash: ./hello: bad interpreter: Permission denied
asim:~> cat hello
#!/bin/bash
echo "Hello, World"
asim:~> ls -l hello
-rw-rw-r-- 1 asim asim 32 Jan 15 16:29 hello
asim:~> chmod u+x hello
asim:~> ./hello
Hello, World
asim:~> ls -l hello
-rwxrw-r-- 1 asim asim 32 Jan 15 16:29 hello*
```

字符+ 和 - 选项被用于对相关的组赋予或收回权限。用逗号分隔的组合也是允许的。在

info 和 man 手册里包含了很多有用的例子。这里有另一个例子，运行如下命令使得先前的那个文件就变成了用户 asim 的私人独享的文件了：

```
asim:~> chmod u+rwx,go-rwx hello


asim:~> ls -l hello
-rwx------1 asim asim 32 Jan 15 16:29 hello*
```

这种类型的问题会遇到一个权限被拒绝的错误信息，它通常就是一个访问权限方面的问题了。还有，这些说法，”它昨天还能正常运行”，以及”当我使用 root 用户它能正常运行”，很可能就是由于文件权限引起的问题了。

当使用数字参数的 chmod 命令，该值需要根据每一组将访问权限代表的数字加起来。因此我们能得到 3 个数字号码，这些号码就是 chmod 所作用的符号值。下表列出了最常用的组合：如果你的 chmod 命令参数少于三个，缺少的这个字符将从左边开始算起用 0 替代。Linux 上通常还有第四个数字，在第一组三个数字之前，并代表了特殊的访问模式。任何涉及这些以及更多的信息请考考 info 手册。

Table 3-9. 使用 chmod 进行文件保护

命令	意义
chmod 400 file	保护一个文件以防止意外重写
chmod 500 directory	保护你自己以防止意外的移除，重命名目录或是从目录中转移文件。
chmod 600 file	只有做这个权限设置的用户才能修改此私有文件。
chmod 644 file	公共的可读文件，只有进行这个权限设置的用户才能修改它。
chmod 660 file	跟你在同一个组的用户可以修改该文件，其他用户不能做任何访问。
chmod 700 file	保护文件使得其他用户均无法访问该文件，但是该文件的所有者则拥有所有文件权限。
chmod 755 directory	其他用户可以读和执行这些文件，但是只有命令发起的用户才能更改该文件。
chmod 775 file	组内共享的标准文件。
chmod 777 file	任何用户都可以对文件进行任何操作。

3.4.2.2. 登录其他用户组

当你在命令行输入 id 命令，在你的目前登录的用户名和组名之后，你将得到你从属的所有组的列表。然后，在很多 Linux 系统，你一次只能登录到一个组中。默认的，这个活动的或主

要组就是你在文件/etc/passwd 中指派的组。在这个文件的第四列表示的是主要组的 id 号，该 id 号可以在文件/etc/group 查找到。一个例子如下：

```
asim:~> id
uid=501(asim) gid=501(asim) groups=100(users),501(asim),3400(web)
asim:~> grep asim /etc/passwd
asim:x:501:501:Asim El Baraka:/home/asim:/bin/bash
asim:~> grep 501 /etc/group
asim:x:501:
```

在文件/etc/passwd 的第四列包含了值"501"，表示以上例子的 asim 组。从文件 /etc/group 我们可以获得这个组 id 号对应的组名称。当开始连接系统时，用户就使用组 asim 来登录了。

 用户私有组的设计方式

为了获得更大的灵活性，大部分 Linux 系统遵循所谓的用户私有组的设计方式，就是把每个用户赋给它们的自己的组。这个组只包含这个特定用户，所有叫做”私有组”。通常的这个组名和用户登录的名称一样，这可能会引起一些混淆。

除了他自己的私有组之外，用户 asim 也能是组 users 和 web 的成员。由于这是用户的次级组，他将需要使用 newgrp 命令来登录这些次级组(首先使用 gpasswd 来设置组的密码)。在这个例子中，用户 asim 需要创建属于组 web 下的新的文件：

```
asim:/var/www/html> newgrp web
asim:/var/www/html> id
uid=501(asim) gid=3400(web) groups=100(users),501(asim),3400(web)
```

当 asim 现在创建新文件时，这些新文件就属于组 web 所有，而不是先前的 asim 组：

```
asim:/var/www/html> touch test

asim:/var/www/html> ls -l test
-rw-rw-r--1 asim web 0 Jun 10 15:38 test
```

登录到新的组会阻止你使用 chown (参见 Section 3.4.2.4) 或者可以请求系统管理员为你更改你的身份。

为获得更多关于 newgrp 的信息请参考 man 手册。

3.4.2.3. 文件权限模板

当新文件保存在某个地方，它首先应该服从标准的安全程序。文件没有设定权限不能在 Liunx 系统上存在。标准的文件权限是在新文件创建时根据模板来决定的。模板的值可以使用 umask 命令显示出来：

```
bert:~> umask
0002
```

跟为每个文件添加符号值，类似于 chmod 的方式不同， 为了计算一个新文件的权限，它们需要减去所有可能的访问权限。在上面的例子中，我们可以看到显示了 4 个值，但是只有 3 个权限类别： 用户，组，和其他。第一个 0 是特殊文件属性的一部分，这方面的内容我们将在 Section 3.4.2.4 和 Section 4.1.6 进行讨论。当你输入 umask 命令时也有可能你的系统并不会显示第一个 0，这样你只能看到 3 个数字号码来代表默认的文件权限。

每一个类似于 UNIX 的系统都有创建新文件的系统功能，该功能也被叫做每次用户使用程序就会创建新文件，例如，从网络上下载一个文件，保存一个新的文本文档等等。这种功能能创建新文件和新目录。当创建一个新目录的时候，会赋予所有用户全部的读，写和执行权限。当创建新文件时，会赋予所有用户读和写权限，但是执行权限不给赋予任何用户。这样，在权限模板还没有被应用之前，一个目录的权限是 777 或 rwxrwxrwx，一个文件的权限是 666 或 rw-rw-rw-。

使用这种功能创建文件或目录，权限的值根据默认的权限减去 umask 值所得的。因此，假如 mask 的值是(0)002 的话一个新建的目录默认权限为 775，一个文件默认权限为 664。这个可以从下面的实例中得到证明：

```
bert:~> mkdir newdir

bert:~> ls -ld newdir
drwxrwxr-x 2 bert bert 4096 Feb 28 13:45 newdir/

bert:~> touch newfile

bert:~> ls -l newfile
-rw-rw-r--1 bert bert 0 Feb 28 13:52 newfile
```

 文件与目录

目录默认有更多的权限：它通常有执行权限。如果你没有这个权限，它就服务被访问了。试一试，通过把目录权限更改为 644！如果你使用 newgrp 命令登录到别的组上，这个 mask 值保持不变。因此，如果这个值是 002 的话，你在新组里面创建的文件和目录也可以被改组其他用户访问，而不需要使用 chmod 来更改权限。

Root 用户通常有更严格的文件创建权限：

```
[root@estoban root]# umask
```

这些默认值在 shell 配置文件里其作用范围被设置成系统范围的，配置文件有/etc/bashrc 或/etc/profile。你能在你的 shell 配置文件里进行更改，请参考 Chapter 7 关于如何客户化你的 shell 环境。

3.4.2.4. 更改用户和组的所有权

当一个文件属于错误的用户或组时，可以通过 chown（更改所有者）和 chgrp（更改组）命令进行相应的解决。在一个文件文件需要被组里其他成员共享的环境下更改文件的所有权是一个很频繁的系统管理任务。两个命令都很灵活，使用—help 选项你可以查到它们的使用方法。

chown 命令可以更改文件的用户和组的所有权，而 chgrp 命令只能更改组的所有权。当然，系统会检查想更改该文件的用户是否对该文件有足够的权限。

为了只更改文件的用户所有权，使用如下语法：

```
chown  newuser file
```

如果你在用户名之后使用冒号(参加 info 手册)，也会把组更改成发出该命令的用户的优先组。在 Linux 系统，每个用户有他自己的组，因此这样的形式可以把文件变成私有的：

```
jacky:~> id
uid=1304(jacky) gid=(1304) groups=1304(jacky),2034(pproject)
jacky:~> ls -l my_report
-rw-rw-r-- 1 jacky project 29387 Jan 15 09:34 my_report
jacky:~> chown jacky: my_report
jacky:~> chmod o-r my_report
jacky:~> ls -l my_report
-rw-rw---- 1 jacky jacky 29387 Jan 15 09:34 my_report
```

如果 jacky 想要共享这个文件，不需要使用对每个用户赋予读权限，只需要似乎用 chgrp 命令进行更改就可以了：

```
jacky:~> ls -l report-20020115.xls
-rw-rw---- 1 jacky jacky 45635 Jan 15 09:35 report-20020115.xls
jacky:~> chgrp project report-20020115.xls
jacky:~> chmod o= report-20020115.xls
jacky:~> ls -l report-20020115.xls
-rw-rw---- 1 jacky project 45635 Jan 15 09:35 report-20020115.xls
```

采用这种方式，组 project 中的用户就可以对这个文件进行操作了。不在这个组下的用户跟它就没有什么关系无法操作它。

chown 和 chgrp 使用-R 选项，都可以用来实现所有权的递归处理。这种情况下，给定目录底下的所有文件和子目录都会属于该用户和/或该组。

 约束

在大部分系统上，一般是限制普通用户使用 chown 和 chgrp 命令的。如果你不是系统管理员，由于安全因素你就不能更改用户和组的所有权。如果这些命令的时候不被限制，有恶意的用户可以将这些文件的所有权赋予其他用户和组，并更改这些用户的用户环境，最终引起其他用户文件的损坏。

3.4.2.5. 特殊模式

为了系统管理方便，而不是一直被解决权限问题给阻碍了，可以对整个目录或单独的程序赋予特殊的访问权限。有三种特殊模式：

• Sticky bit 方式： 在执行一个任务后，命令被保存在系统内存中。最初这是节省内存的广泛使用的特性：大任务只有一次被装载进内存。但是现在内存比较便宜并且出现了更好的技术来处理它，因此它不再用户处理单个文件的优化了。当应用于整个目录时，sticky bit 却有不同的含义。在这种情况下，用户想更改一个目录下的文件，他必须是这个文件的所有者或者这个文件有恰当的使用权限。这个特性使用与类似/var/tmp 这样的目录，这种目录可以被任何人访问，但是它不适合用户去更改或删除别的用户的数据。sticky bit 特性在文件的权限字段末端用 t 来显示：

```
mark:~> ls -ld /var/tmp
drwxrwxrwt 19 root root 8192 Jan 16 10:37 /var/tmp/
```

sticky bit 方式可以使用命令 chmod o+t directory 来设置。这个著名的"t" 表示 UNIX 中保留正文入门的特性。

• SUID (设置用户 ID) 和 SGID (设置组 ID)：在用户或组权限字段由字符 s 表示。当在可执行文件被设置为这种模式时，该模式将作用于文件的用户和组的权限，进而代替了用户的命令设置，因此可以访问系统资源。我们将在 Chapter 4 详细讨论。

• SGID (设置组 ID)在一个目录上：在这种特殊情况下目录下创建的每个文件都与该目录拥有一致的组所有者(而通常的行为是文件是属于创建它的用户所有)。使用这种方式，当共享目录的时候，用户不需要担心文件的所有权：

```
mimi:~> ls -ld /opt/docs
drwxrws---4 root users 4096 Jul 25 2001 docs/

mimi:~> ls -l /opt/docs
-rw-rw----1 mimi users 345672 Aug 30 2001-Council.doc

This is the standard way of sharing files in UNIX.
```

 现有文件将保持不变!

在别的地方创建的文件移至 SGID 目录将保持它们的初始的用户和组所有者属性不点。这可能会有些迷惑。

3.5. 总结

在 UNIX 系统，同样在 Linux 系统，所有的实体在某种方式或其他方式都表现为带有恰当文件属性的文件。使用(预定的)路径运行用户和系统管理员查找，读取并操作文件。

我们已经迈向了通往专家的第一步：我们已经讨论了真实的和伪装的文件系统，并且我们知道了 Linux 文件的安全模式，以及几个系统默认的其他安全防范策略。

Shell 是与系统进行交互的最重要的工具。我们在本章学习了几个 shell 工具，请参考下表。

Table 3-10. 第 3 章的新命令: 文件和文件系统

命令	意义
bash	GNU shell program.
cat file(s)	Send content of file(s) to standard output.
cd directory	Enter directory. cd is a bash built-in command.
chgrp newgroup file(s)	Change the group ownership of file(s) to newgroup
chmod modefile(s)	Change access permissions on file(s)
chown newowner[:[newgroup]] file(s)	Change file owner and group ownership.
cp sourcefiletargetfile	Copy sourcefileto targetfile.
df file	Reports on used disk space on the partition containing file.
echo string	Display a line of text
export	Part of bash that announces variables and their values to the system.
file filename	Determine file type of filename.
find pathexpression	Find files in the file system hierarchy
grep PATTERNfile	Print lines in filecontaining the search pattern.
head file	Send the first part of fileto standard output
id	Prints real and effective user name and groups.
info command	Read documentation about command.
less file	View filewith a powerful viewer.
ln targetfilelinkname	Make a link with name linknameto targetfile.
locate searchstring	Print all accessible files matching the search pattern.
ls file(s)	Prints directory content.
man command	Format and display online (system) manual pages for command.

mkdir newdir	Make a new empty directory.
mv oldfilenewfile	Rename or move oldfile.
newgrp groupname	Log in to a new group.
pwd	Print the present or current working directory.

quota	Show disk usage and limits.
rm file	Removes files and directories.
rmdir file	Removes directories.
tail file	Print the last part of file.
umask [value]	Show or change new file creation mode.
wc file	Counts lines, words and characters in file.
which command	Shows the full path to command.

我们也强调一个事实，你应该读 man 手册。这个文件是你的必备文档，它包含了很多问题的答案。上面列表包含了你在日常管理中使用到的基本命令，但是它们能比我们在这讨论的功能做得更多。

最后但不是最终，简单看下文件权限的基本情况：

Table 3-11. 文件权限

Who\What	r(ead)	w(rite)	(e)x(ecute)
u(ser)	4	2	1
g(roup)	4	2	1
o(ther)	4	2	1

3.6. 练习

仅以你的常用用户 ID 登录系统。

3.6.1. 分区

你的 home 目录在哪个分区？

你的系统有多少个分区？

你安装的 Linux 系统总大小是多少？

3.6.2. 路径

显示你的搜索路径。

输出一个无意义的路径通过命令 export PATH=blah 并显示该目录的内容。你的 home 目录的路径是什么？另一个用户如何从它的 home 目录去访问你的 home 目录，使用相对路 径？

到/var 下面的 tmp 目录。

只使用一个命令共享目录/usr 目录。然后转至 doc 目录。你现在的工作目录是什么？

3.6.3. 遍历系统

将目录更改至/proc 目录。

你的系统运行的是哪种类型的 CPU(s) ？

现在使用的 RAM 是多少？

你有多少交换空间？

系统装载了哪种类型的驱动？

系统已经运行了多少个小时了？

你的系统能解析哪个文件系统？

转至/etc/rc.d | /etc/init.d | /etc/runlevels 并为你的运行级别选择恰当的目录。

什么服务应该在这种级别上运行？

哪个服务可以运行于图形模式而不能运行于文本模式？

更改到目录/etc

监控用户的日志在系统保留多长时间？

你运行的是哪个版本？

工作期间是否有任何的问题或消息？

你的系统有多少个用户？不要自己去数，让计算机帮你做！

有多少个组？

时区信息保存在哪个地方？

你的系统安装 HOWTOs 手册了吗？

变更目录到/usr/share/doc.

指出 N GNU coreutils 包的三个程序

你的系统安装的是哪个 bash 版本?

3.6.4. 操作文件

在你的 home 目录下创建新的目录。

你能把这个目录移动到与你的 home 目录同级别的地方吗？

从目录/usr/share/pixmaps 拷贝所有的 XPM 到这个新目录。XPM 是什么意思?

按字母反序排列文件。

更改目录到你的 home 目录。创建一个新目录并拷贝/etc 目录下的所有文件到这个新目录下。确认你们同时拷贝了/etc 的子目录的文件和目录(递归拷贝)

到新目录下，根据文件以大写和小写字母开头为依据创建两个目录，并分别将对应的文件放至两个目录下。用最少的命令来完成这个任务。

移除剩下的文件。

使用一个命令删除整个目录及其内容。

使用 grep 命令查找哪些脚本启动了图形模式的 Font Server。

Sendmail 服务程序位于哪个地方?

在你的 home 目录创建一个链接文件至指向目录/var/tmp。检查它可以正常使用。

在你的 home 目录下创建另一个链接文件执行刚才的创建的链接文件。检查它能正常使用。移除第一个链接文件并列出目录的内容。第二个链接文件发生了什么改变?

3.6.5. 文件权限

你能在/home 目录下更改文件的权限吗?

你的标准文件创建模式是什么?

更改目录/etc 的所有权为你自己的用户和组。t

更改文件~/.bashrcso 的权限使得只有你和你的主要组成员才能读它。

发出命令 locate root。你有没有注意到什么特别的?

创建一个链接文件指向目录/root。它可以被使用吗?

测试样题

1.删除空目录有哪一个命令_____

A. mkdir
B. rmdir
C. del
D. cp

2.在 linux 系统中，要列出当前目录下所有扩展名为.txt 的文件，需要使用以下哪个命令_____

A. ls *.txt
B. cat *.txt
C. find *
D. cp a.txt b.txt

3.改变用户所属组的命令是哪一个_____

A. chgrp
B. chmod
C. umask
D. cd

4.在 Linux 系统中，根据下列命令及其执行结果,能更改文件的属主为 st01 的命令是_____

A. chmod st01 text.text
B. chown st01 text.text
C. chmod :st01 text.text
D. chown :st01 text.text

5.一般来说，Linux 系统中_____目录用于存放系统启动时需要的文件

A. /home
B. /bin
C. /sbin
D. /boot

6.在 linux 系统中用户执行“pwd” 命令的结果是“/usr/share/doc”，那么执行命令“cd ../..”后再执行“pwd”命令的结果是_____

A. /usr/share/doc
B. /usr/share/
C. /usr
D. /