

Linux 新手管理员指南

LINUX NEWBIE ADMINISTRATOR GUIDE

Shadowfax Linux Team
幻影快递 Linux 小组
ShadowfaxLinux@hotmail.com

Rev 2.00
Last Update: May 22, 2003

声明

本书《Linux 新手管理员指南》是英文版 LINUX NEWBIE ADMINISTRATOR GUIDE 的完整中文译本

中文版权所有：Linux 幻影快递小组（Shadowfax Linux Team, SLT）的所有成员及小组负责人苏淮河

英文版版权所有：Peter, Stan Klimas

我们非常欢迎您的反馈意见和批评指正。限于非商业用途，本书可以免费下载，复制和传阅。

对于商业用途的出版和印刷，敬请 email 联系本书的总体负责人 suhuaihe@hotmail.com。使用本介质进行任何商业用途的目的将追究其法律责任。

Disclaimer

This book, 《Linux 新手管理员指南》 is a full Chinese translation version of the book "Linux Newbie Administrator Guide" by Peter, Stan Klimas.

Chinese language version Copyright (c) by Shadowfax Linux Team, and the team leader Huaihe Su.

English language version Copyright ? by Peter, Stan Klimas.

Your feedback, comments, corrections, and improvements are appreciated. Send them to suhuaihe@hotmail.com.

This materials maybe distributed only subject to the non-commercial purposes, please contact email suhuaihe@hotmail.com for any ommercial publishing purposes.

目 录

第 0 部分: 为什么需要 Linux	1
0.1 为什么需要 Linux?	1
0.2 Linux 对我会有用吗?	2
0.3 Linux 对新手来说太难了	2
0.4 Linux 有什么好处?	2
0.5 Linux 和 Unix 有什么不同?	5
0.6 Linux 和微软的 Windows 有什么不同?	5
0.7 我不相信什么免费软件	6
0.8 天底下根本没有免费的午餐	7
0.9 我需要高的系统安全性。购买商业软件, 如果出了问题可以告他们	7
0.10 我需要标准。大的软件公司提供标准	7
0.11 我需要用微软的 Windows 才能读写 Word 文档	8
0.12 微软的 Windows 的普及性能够保证长久的使用	9
0.13 但是 Linux 可能会被“肢解”成很多不同的系统	9
0.14 Linux 只是一个小帮派	9
0.15 Linux 总体拥有费用(TCO)太高	10
0.16 Linux 只是一个理想的梦, 统治这个世界由来已久的是商业	10
0.17 Linux 太滥了	11
第 1 部分: 安装 Linux 之前	12
1.1 我应该选择哪个 Linux 发行版本?	12
1.2 LINUX 对硬件有什么要求?	14
1.3 我的硬件支持 Linux 吗?	15
1.4 怎样从网上下载 Linux?	16
1.5 怎样获得 Linux 安装光盘?	16
1.6 我有 RedHat 安装光盘, 但没有安装软盘, 应怎么办?	17
1.7 安装前需要阅读哪些资料?	18
1.8 我能在一部计算机上同时安装 Windows 和 Linux 吗?	18
1.9 应该怎样给硬盘分区?	19
1.10 我的硬盘被 WINDOWS 分区占满, 我能缩小 Windows 分区而不用重装系统吗?	22
1.11 怎样开始安装?	23

1.12	安装 Linux 困难吗?	23
1.13	我应该安装哪些软件包?	23
1.14	我应该选择哪一个图形用户界面, KDE 还是 GNOME?	24
1.15	我已经完成安装了。我该如何第一次登录?	24
1.16	我怎样才能把我的 Linux 搞崩溃?	25
1.17	我可以总是在图形用户界面下使用 Linux 吗?	26
1.18	我该如何升级 Linux?	26
第 2 部分: Linux 的资源, 帮助和相关链接		28
2.1	哪里有 Linux 的资料?	28
2.2	有没有 HELP 命令?	29
2.3	有没有术语字典?	30
2.4	网上搜索	31
2.5	新闻组	31
2.6	有没有因特网上的 Linux 的链接?	32
2.7	源代码--终极资源	33
第 3 部分: Linux 的基本操作		34
3.1	基础知识	35
3.1.1	文件名	35
3.1.2	使用不同的目录	37
3.1.3	执行一个程序	40
3.1.4	改变路径 (PATH)	41
3.1.5	关闭计算机	42
3.1.6	处理挂起的程序	44
3.1.7	命令选项	46
3.2	用户, 密码, 文件访问权限和安全性	47
3.2.1	主目录, 根目录和添加新用户	47
3.2.2	密码安全性	48
3.2.3	忘记了超级用户密码怎么办	49
3.2.4	忘记了普通用户密码怎么办	51
3.2.5	关闭或删除一个用户	52
3.2.6	文件的所有权和访问权	53
3.2.7	设置替代用户标识(SUID)	56
3.3	用“&”, “at”, “batch”和“cron”进行作业调度	57

3.3.1	怎样在后台执行命令	57
3.3.2	怎样在某一特定时间执行命令(使用 at ” 和 “ batch ”)	58
3.3.3	怎样设置 “ cron ”	59
3.4	Shell	60
3.4.1	什么是 Shell? 需要一个不同的 Shell 吗?	60
3.4.2	怎样定制自己的 Shell	61
3.4.3	设置文本终端的颜色	62
3.4.4	在文本模式或 X 模式控制台下,如何打印符号	64
3.4.5	如何编写简单的 Shell 脚本(Script)文件	65
3.4.6	引号的含义	66
3.4.7	输入输出重定向	66
3.4.8	Shell 的特殊字符(metacharacters)	68
3.5	如何安装软件包(Package)及 rpm 软件包管理器	70
3.5.1	如何安装从 Internet 下载的程序?	70
第 4 部分: Linux 管理员新手常见问题		73
4.1	关于启动 (LILO 和 GRUB)	74
4.1.0	LILO 和 GRUB	74
4.1.1	Linux 不能发现我的所有内存	75
4.1.2	LILO 仅显示 LI (或 LIL) 并死机	76
4.1.3	我如何能让操作系统变为由 LILO 缺省引导?	77
4.1.4	在启动时 LILO 提示符在屏幕上停留时间太短 (或太长)	78
4.1.5	卸载 Linux.....	79
4.2	访问我的驱动器	79
4.2.1	我的驱动器在哪里?	79
4.2.2	我如何访问我的 CDROM?	80
4.2.3	如何挂载软盘, zip 磁盘驱动器, dos 分区, 或网络驱动器?	81
4.2.4	如何通过 Samba 挂载远程 MS Windows 的文件系统?	83
4.2.5	有快捷方式访问在 DOS/Windows 软盘上的文件吗?	84
4.2.6	作为超级用户我可以挂载. 普通用户能做到吗?	85
4.2.7	挂载命令太长, 我如何用别名简化?	86
4.2.8	我能自动挂载吗?	87
4.2.9	我如何得到已识别的在并行端口(外部)的 zip 磁盘驱动器?	88
4.2.10	我能设置 32 位硬盘的 I/O?	89

4.2.11	我所打开的文件数已达到限量(错误信息)	90
4.2.12	我加了一个新硬盘, 开始使用时要做什么?	90
4.2.13	交换分区 SWAP	93
4.3	使用 X-Windows 工作	95
4.3.1	怎样切换文本和图形控制台?	95
4.3.2	我怎样在 X-server 上安装显示卡、监视器以及鼠标?	96
4.3.3	我能有一个 GUI 登陆提示符吗?	97
4.3.4	我怎么安装 KDE? (例如, 在 RedHat5.2)	97
4.3.5	我怎么将默认桌面改变成 KDE (如 Gnome 以及其他类型的 KDE)?	99
4.3.6	我能够同时运行多个 Linux 阶段作业吗?	100
4.3.7	在不关闭我的 Linux 阶段作业的前提下可以开启第二个 GUI 登录提示符吗	100
4.3.8	如何遥控 X-Windows?	101
4.3.9	我如何从 MS Windows 分区安装 TrueType 字体?	103
4.3.10	如何进行拷贝粘贴操作?	104
4.3.11	我如何使用 VNC 来显示和控制一个遥控桌面?	105
4.4	基本配置	107
4.4.1	如何设置我的声卡?	107
4.4.2	如何设置我的打印机?	108
4.4.3	Word Perfect 8 没有适合我的打印机的驱动程序	109
4.4.4	配置文件在哪里?	109
4.4.5	这些设备文件都是什么?	110
4.5	网络	115
4.5.1	有必要建立家庭网络吗?	115
4.5.2	怎样建立我的家庭网络?	115
4.5.3	设置 PPP 拨号 GUI 的问题	117
4.5.4	如何不通过 modem 用我的联网电脑浏览互联网?	120
4.5.5	怎样使用 Samba?	121
4.5.6	Sendmail	123
4.5.7	简单的 WEB 服务器 (Apache 阿帕奇)	124
4.5.8	简单的 FTP 服务器	125
4.5.9	如何通过电话线外部远程访问我的计算机?	126
4.5.10	家庭计算机是否会遭黑客攻击吗?	127
第 5 部分: Linux 的快捷键和主要命令		131

5.1	Linux 基本的键盘输入快捷键和一些常用命令.....	132
5.2	帮助命令.....	137
5.3	系统信息.....	138
5.4	基本操作.....	143
5.5	文件管理.....	144
5.6	查看和编辑文件.....	146
5.7	查找文件.....	147
5.8	X-windows 基础.....	148
5.9	网络应用程序.....	148
5.10	文件压缩和解压缩.....	151
5.11	进程控制.....	152
5.12	一些管理命令.....	154
5.13	硬盘和软盘实用工具.....	160
5.14	用户帐户和文件权限管理.....	162
5.15	程序安装.....	165
5.16	访问磁盘驱动器/分区.....	166
5.17	网络管理工具.....	167
5.18	与音乐有关的命令.....	170
5.19	与图形相关的命令.....	172
5.20	几个小游戏.....	174
第6部分: Linux 基本的应用程序.....		176
6.1	字处理程序.....	177
6.1.1	StartOffice/Open Office 软件.....	177
6.1.2	abiword 字处理程序.....	180
6.1.3	kword.....	180
6.1.4	klyx, lyx 和 latex.....	181
6.1.5	WordNet(字典/词典/同义词/反义词查询).....	184
6.2	电子表格.....	184
6.2.1	oocalc.....	184
6.2.2	gnumeric.....	185
6.2.3	kspread.....	185
6.3	数据库.....	185
6.4	CAD.....	186

6.5	网络浏览器 : Mozilla, Konqueror and Lynx	186
6.6	刻录 CD-R: cdrrecord 和 cdparanoia	187
6.7	使用 gnuplot 自动画图	194
第 7 部分: Linux 一些高级应用		196
7.1	LINUX 高级文本处理工具	196
7.2	LINUX 下的简单编程	210
7.3	数学工具	221
7.4	杂项	224
许可证、鸣谢及其它		226
中文译本后记		228
SHADOWFAXLINUX 小组电子邮件列表		229

第 0 部分：为什么需要 Linux

Linux 新手管理员指南(中文版)

LINUX NEWBIE ADMINISTRATOR GUIDE (Simplified Chinese Version)

中文版本 1.0 由幻影快递 Linux 小组(SLT-Shadowfax Linux Team) 翻译整理

Linux Newbie Administrator 英文原版版权所有：Peter and Stan Klimas.

<http://sunsite.dk/linux-newbie>

中文版本版权所有：幻影快递 Linux 小组 ShadowfaxLinux@hotmail.com

本章内容：

- 0.1 为什么需要 Linux ?
 - 0.2 Linux 对我会有用吗？
 - 0.3 Linux 对新手来说太难了
 - 0.4 Linux 有什么好处？
 - 0.5 Linux 和 Unix 有什么不同？
 - 0.6 Linux 和微软的 Windows 有什么不同？
 - 0.7 我不相信什么免费软件
 - 0.8 天底下根本没有免费的午餐
 - 0.9 我需要高的系统安全性。购买商业软件，如果出了问题可以告他们
 - 0.10 我需要标准。大的软件公司提供标准
 - 0.11 我需要用微软的 Windows 才能读写 Word 文档
 - 0.12 微软的 Windows 的普及性能够保证长久的使用
 - 0.13 但是 Linux 可能会被“肢解”成很多不同的系统
 - 0.14 Linux 只是一个小帮派
 - 0.15 Linux 总体拥有费用(TCO)太高
 - 0.16 Linux 只是一个理想的梦，统治这个世界由来已久的是商业
 - 0.17 Linux 太滥了
-

0.1 为什么需要 Linux ?

如果你真的喜欢计算机，Linux 绝对是你梦想中的操作系统。它比现在所有其他的操作系统更加有意思。然而，Linux 的真正革命在于它是开放性的软件。我们在科技和技术方面的发展为今日的免费信息和相互检查提供了可能性。

举个例子，如果你外出，你会乘坐那些是使用秘密技术和没有监督检查的方案制造出来的飞机吗？那些除了制造商以外再也没有人能够了解内部结构的飞机？你如何信任一个可能隐含秘密代码的程序？只有代码开放的 Linux 系统才能够真正符合运行关键任务的要求。它的安全性和性能是由基于每个人都可以阅读，批评和改进的全面解决方案，也是当今科学计算领域的集中体现。

制造马靴和精致的玻璃，测量时间等技术曾经是被严密保护的商业秘密。500 年前印刷出版技术的出现使得科学和技术的知识共享开始得到普及。在印刷业发展的早期，那些敢于共享的人会以“泄漏商业秘密”的罪名被起诉。Linux 相对于计算机时代，就像当年古滕贝格 (Johannes Gutenberg, 1400 - 1468, “西方版”的活字印刷术发明人) 对于印刷的作用。希望这次不会再有什么起诉-☺。Linux 的确让那些自称信息拥有者想让时间倒退五百年的人感到极不舒服。

0.2 Linux 对我会有用吗？

只有你自己可以回答这个问题。Linux 是一个成熟，强壮，安全和应用广泛的类似于 Unix 的操作系统。Linux 的强壮和应用广泛来自于一定的代价 - 为了安装和维护 Linux 可能要求你必须是一个计算机的行家。Linux 在操作系统和应用程序正确安装完毕后，相对来说很方便使用。举个例子，如果你设置一个基于图形用户界面的用户，并且将用户所需要的图标和菜单放在图形桌面上，你的母亲也可以很容易就能够使用 Linux。而且 Linux 很安全，不管她怎么努力尝试都不会毁坏这个系统 除非使用锤子☺

Linux 和微软的 Windows 截然不同，所有不要以为你玩转了 Windows，Linux 对你就不是什么问题了。你可能需要学习。另一方面，如果你从 Unix 转过来，Linux 对你来说应该很简单。如果你不喜欢计算机，或者不喜欢享受折腾计算机带来的乐趣，那么 Linux 管理员的工作可能不适合你。如果你不知道你的计算机硬件，Linux 的安装对你来说是个巨大的挑战。

0.3 Linux 对新手来说太难了

可能是这样的。但问题是：你是不是真的想要学习它？本书的作者里没有一个是计算机科学背景的，但是我们天天在使用它而且我们喜欢它。

0.4 Linux 有什么好处？

Linux 可以给你：

- 一个新潮，非常稳定，多用户，多任务的环境。Linux 基于非昂贵硬件，而且软件是免费的（或者近于免费）。Linux 是一个功能齐全而且强健的平台 不要错认为它是一个“穷人”才用的操作系统。开启包装盒后的 Linux 的功能相当于微软的 Windows NT 加上 5,500 美元的其他软件的总和，而且 Linux 更稳定，运行相似的任务比 Windows 要求更少的硬件资源。
- 标准的平台。Linux 非常标准 它基本上是一个与 POSIX 兼容的 Unix 系统（没错，Linux 是 Unix 最好的衍生品。Unix 不再用来和 Linux 产生任何关联是因为 Unix 是一个注册商标）。Linux 包含了所有标准的 Unix 工具和应用程序。
- 无法超越的计算能力，可移动性和适应性。1999 年 4 月，一台集群的 Linux 机器在性能测试上击败了 Gray 超级计算机。出于价格考虑，Linux 大部分运行在便宜的 Intel 的芯片上，但是它在其他很多硬件平台上（从小玩具到大型机）也运行良好。一个 Linux 发行商 (Debian) 声称：“Linux 是一个无所不在的操作系统”。Linux 几乎可以通过量身定做去适应任何计算任务的需要。

- 先进的图形用户界面。Linux 使用一个标准，网络透明的 X-Window 系统，并且自带窗口管理程序 (KDE 或者 GNOME)。
- 几十个异常出色而且免费的桌面应用程序。这些软件包括网络浏览器，电子邮件程序，字处理，电子表格，图形和线性图形程序，文件管理器，音频播放机，CD 刻录程序，游戏等等。
- 成千上万个免费的工具和应用小程序。“小的就是好的”在 Linux 哲学里表现得淋漓尽致。这些应用小程序协调工作用于实现非常复杂的计算任务。
- 几百上千个专业程序由全世界的无数研究人员编制，覆盖了天文，信息技术，化学，物理，工程，语言，生物等各个学科领域。在很多专业领域，Linux 更像是“唯一”的操作系统（试着看看你搞天文学的朋友的计算机用的是什么操作系统）。的确这一类的软件不是非常好使用，但是，如果你需要超强的计算能力，Linux 是现在在这个领域里最好的软件。不相信？看看以下链接：<http://sal.kachinatech.com/Z/2/index.shtml>
- 很多顶级的商业软件包括大型数据库（比如 Oracle，Sybase，当然 不包含微软）。这些软件很多（或者几乎）都提供给开发者和个人免费使用。
- 一个真正杰出的学习系统。如果你是家长看到你的儿子或者女儿在折腾 Linux，你真应该感到高兴，他们正在学习一些可以长久保值的東西。如果你是一个教师，你应该考虑在学校里安装一个 Linux。如果你更侧重于教育而不是培训，你应该选择 Linux。好的大学更多的是用 Linux 来作为教学的主要内容。举个例子，在 Linux 系统下，你可以让你的学生立刻学习如何编译一个电子表格应用程序的代码，这个程序和微软的 Excel 具有同样先进的功能。Linux 让你始终站在科学技术的前沿，比如信息技术，项目管理，质量控制，方法学等等。很多教师不在日常教学里使用 Linux，是因为他们本身缺少计算机教育背景。
- 出色的网络功能已经内置的系统。你可能会想你不需要网络？如果你试图实现家庭联网，没有网络功能根本不行。如何把两台或者更多的计算机联在一起，让它们共享硬盘，CDROM，声卡，Modem，打印机？如何只用一个因特网连接就能够让两台或者更多的计算机同时浏览网络？如何和你的儿子在家里打网络游戏？甚至，通过联网使用在奔腾服务器上的网络资源，你的老掉牙的 386 带 Windows 3.1 的机器可以再发挥相当的作用。支持这些功能所需的软件都在标准的 Linux 软件包里，绝对免费，只是按照你的要求设置一下就行了。它们并非“二流共享软件”，绝对是那些在因特网上运用广泛的软件（比如 Apache，几乎占据 50% 的网络服务器的份额，而 Sendmail 几乎占据涉及 70% 的所有电子邮件的总和）。充分享受家庭联网的快乐，是我之所有很喜欢 Linux 的主要原因。
- 可以连接到微软，NOVLE 和 APPLE 的网络。还照样可以读写微软 MS DOS 和 MS Windows 的其他文件格式。包括那些存放在微软 Windows 操作系统下的硬盘分区的文件。
- 很多顶级的系统开发平台，友好的编程语言及编程工具都免费包含在操作系统里面。如果你有特别需要必须阅读操作系统的源代码，也是免费的。Linux 自带的 C 语言编译器的兼容性和适应性可以说比现在所有其他不同的平台要好得多。Perl，Python，Guide，Tcl，Ruby，功能强大的 shell 脚本，甚至连汇编程序的编译器也包含在标准的 Linux 里面。
- 病毒入侵，计算机的“后门”，软件提供商的“特别要求”，强迫性的软件升级，专有的文件格式，软件使用许可证和市场策略，产品注册登记，高得惊人的软件价格，等等。觉得这些东西怎么样？Linux 没有病毒，因为它是一个极其安全的操作系统可以有效的防止病毒发作和大规模的传播。余下的那些，因为 Linux 遵循源代码开放和非商业化的原则，它封装了最佳的计算方法，可靠的代码验证，安全性，适应性，实用性和极佳的性能。

- Linux 操作系统平台保证它的“永久性”。既然 Linux 不是属于哪一家公司私人拥有，它就不可能因为公司的倒闭就消失。Linux 的通用公共许可证 (General Public License, GPL) 将保证，只要还有 Linux 的用户，他们就能够提供对操作系统的开发和维护。
- 一个技术快速更新的平台。这个由 Linux 的新式的开放软件开发模型所决定：“基于前一开发者的成果”和“对等代码检查”（相比于一般的封闭式软件开发模型的“一切从头开始”和“没有人能够读我的代码”）。即使现在这些 Linux 迷们都见上帝去了，Linux 的开发还是会和刚开初的那样继续下去。开放源代码的软件开发模型有它自己的特性：开发的过程从速度上来讲感觉好像很慢，但是整个开发过程展得很开，这样，一些程序的安全漏洞在这个过程中就被解决掉了，因为对于一个程序来说，实现相似的功能可能有几种不同的选择。

如果你还想了解更多关于通用公共许可证 (GPL)，请查阅著名的 GNU 资料库：

<http://www.gnu.org/copyleft/gpl.html>

<http://www.gnu.org/gnu/linux-and-gnu.html>

<http://www.gnu.org/philosophy/categories.html#TheGNUsystem>

简单说来，GNU 通用公共许可证允许任何人：

- 免费使用软件，没有任何限制
- 复制，分发和销售没有修改过的软件的源代码或者二进制格式文件
- 复制，分发和销售修改过的软件，只要程序的源代码还包含在通用公共许可证的许可范围以内
- 销售软件的服务支持

这个许可证“不允许”做的是：对软件做了修改，但是仅提供二进制的软件格式而不包含程序源代码。简而言之，通用开放许可证就是要尽量禁止有人盗窃别人的软件，然后把它们合成到自己的封闭性商业软件里。当然，如果你从程序作者那里获得许可，也可以把通用开放许可证软件合并到自己专有的商业软件里。被发现违反通用开放许可证的个人，也会被禁止使用通用开放许可证下的软件。

哪一个 Linux 发行商的许可证，可能是最重要的决定因素。它设计用来保证信息自由的长久性。另外还有一些重要的“开放源代码”的项目，比如科学和法律（不是开玩笑的）。Linux 的方式说起来也不是什么特别的新鲜事，简单的说就是科学的方法在软件开放上的应用：你免费得到信息，你加入你的意见和想法，最后，你又让其他人免费得到。不幸的是，某些大公司和他们的律师们很拼命的想去改变这种状况，把我们向后拉，最好拉回到大黑暗时代，这样信息就可以变成“专有”而不是“共有”。后果就是，你经常在报纸上看到一些著名的和 Linux 有关联的人被卷入到一些乱七八糟的纷争里面。

也许你想知道 Linux 的真正价值？以下是一些在亚马孙网站(www.amazon.com)的软件价格。所有的软件价格按美元计算，调查的日期是 2001 年 2 月 3 日，带打折价。几乎所有 Linux 对等的软件都包含在 Linux 的光盘里，而且没有最大用户数的限制。还有，Linux 在硬件的开销上绝对的便宜，因为几乎所有的 Linux 服务都可以在单一的服务器上运行。

Microsoft Windows 2000 Server (5-client)--\$848.99;

Microsoft Exchange 2000 Server (5-client)--\$1,279.99;

Microsoft Outlook 2000 (1-client)--\$94.99;

Systems Management Server 2.0 (10-Cals)--\$994.99;

Proxy Server 2.0--\$886.99;

Microsoft SQL Server 2000 Standard Edition (5-client)--\$1,229.99;

Microsoft SQL Server 2000 Standard Edition (1-user License)--\$4,443.99;

Microsoft BackOffice Small Business Server 4.5 NT (Add-On 5-CAL)--\$264.99;

Windows NT Server Prod Upgrade From BackOffice SBS Small Bus Server (25-client)--\$558.99;
Microsoft Windows 2000 Advanced Server Upgrade (25-client)--\$3,121.99;
Microsoft FrontPage 2000--\$129.99;
Microsoft Internet Security and Acceleration Server --\$664.99;
Site Server Commerce 3.0 (25-client)--\$4,092.99;
Visual C++ 6.0 Professional Edition with Plus Pack--\$525.99;
Microsoft Visual Basic Enterprise 6.0 with Plus Pack--\$1,128.99;
Microsoft Visual Sourcesafe 6.0 CD--\$469.99;
Microsoft Office 2000 Standard (1-client)--\$384.99;
Adobe Photoshop 6.0--\$551.99;
Microsoft Plus Game Pack--\$19.99.

在英文里，“免费”(free)有两种完全不同的解释，有些时候会导致对 Linux 产生一些概念上的误解。这两个意思分别来自于拉丁语的形容词“自由”(liber)和副词“免费的”(gratis)，这两个不同的意思经常用这样的例子来解释：“自由言论”和“免费啤酒”。大部分的 Linux 软件同时包含这两层意思，但是，只有第一层的意思是 Linux 的真正根本所在。

0.5 Linux 和 Unix 有什么不同？

比 Unix 更聪明的命令行？基本上没有。尽管这个方面一直在改变中（变得更好也可能变得更糟）。Linux 比起其他任何的商用 Unix 有更大的市场需求。精巧的图形用户界面也没有什么明显的不同。Linux 和其他的 Unix，都使用标准的 X-Windows 系统。

最主要的不同点：

- Linux 是免费的，但是其他的“Unix 们”都极其昂贵。对于应用程序也一样。很多非常优秀的应用程序在 Linux 上都可以免费得到。即使是你如果购买一些商业软件，在 Linux 平台上的价格也远远比在 Unix 上要便宜得多。
- Linux 可以在很多不同的硬件平台上运行，其中大众化的 Intel-X86 芯片和 IBM 兼容的个人计算机是占据了主导地位。而典型的 Unix 都是和提供商的专有硬件捆绑在一起的，这些硬件的价格更是远远高于一般的 PC 的价格。
- 使用 Linux，你是计算机的真正主宰。而对于 Unix，你只是一个“许可用户”(L-User)，有些 Unix 管理员戏称之为“失败者”(Loser)。
- 在 80 年代末 90 年代初，Linux 感觉上更象是 Dos/Win，但是现在它变得更加稳定牢固和功能齐全。而一个典型的 Unix 帐号，感觉上更加象六七十年代的大型机。
- 某些“Unix”们在一些特定的领域里可能会更成熟（举个例子：安全方面，一些工程应用，最尖端的硬件支持等）。对于那些只是想用来运行他们的服务器和工作站的一般用户，Linux 相对来说比较合适。

0.6 Linux 和微软的 Windows 有什么不同？

一旦 Linux 安装完毕，会有比 Windows 更精巧的鼠标点击？可以说几乎没有。Linux 的安装真的不是一个不小的挑战。一般来说，对于 Windows 来说，当你购买计算机的时候已经被预先安装在你的计算机里了。

最主要的不同点：

- Linux 免费，微软的 Windows 必须付钱。应用程序也一样。
- Linux 的文件格式是免费的。而对于 Windows，一般的做法是，你自己的“数据”被锁定在按软件提供商所规定的秘密文件格式里，如果你想要访问你的“数据”，必须付一定的钱给你的软件提供商用于购买他们在“一定时间内有效”的“工具”才能打开文件。如果政客们把我们的公共资料锁在他们自己的专有格式里，是不是太混帐了？一个典型的例子：“我们从微软得到三年的软件使用期来处理我们的健康记录，但是这些记录我们得保留 100 年”。详细请参见 http://news.bbc.co.uk/hi/english/health/newsid_1694000/1694372.stm
- 在 Linux 下面，看起来你不太会触犯一些许可证协议 所有软件都是你的。在 Windows 下，你看起来好像已经触犯了无数的许可证协议，很有可能会被认为是一个“计算机盗贼”，当然，除非你已经有了一个很高明的律师站在你的后面（别担心，其实没有人站在你的背后 ☺）。
- 微软的 Windows 试图让自己变成“较低价格 - 普通应用 - 平均水平”的操作系统（可能更好也可能更糟）。Linux 则是针对那些喜欢灵巧精悍和功能丰富的计算机用户。
- 微软的 Windows 是基于 DOS，而 Linux 是基于 Unix。微软的 Windows 图形用户界面是基于微软专有的“市场驱动”的内部标准。而 Linux 的通信用户界面是基于工业标准网络透明的 X-Windows。
- Linux 在很多方面都把 Windows 比了下去，比如网络功能特征，开发平台，数据处理能力，以及科研工作站。微软的 Windows 有更鲜锐的外观，一些运行良好的通用商业应用程序，还有很多给小孩子的游戏（其实这些也不算什么好游戏 - Linux 的游戏倾向于更加精巧复杂）。
- Linux 远比你想象的要功能丰富得多。听说过因特网有这样的说法吗？“两个伟大的产品来自于加利福尼亚大学。我不认为这只是个偶然现象”。

0.7 我不相信什么免费软件

那么你相信因特网吗？因特网和 Linux 有着几乎相同的想法和根源。你可能还会记得几年前，关于因特网是否能够免费的争论喧嚣尘上，无休无止。最后，得到的压倒性的结论是“不可能”，因为“谁来付钱呢？”

其实事实上也很简单。彼此协作和良好愿望可以在同时获益：你得到并不一定意味着我要失去什么。现在因特网不仅运行良好，而且一直在快速增长。Linux 也一样。

以下是一位 IBM 的高级执行总裁的观点：“我们对 Linux 如此感兴趣的主要原因，是因为我们深信：Linux 对程序应用的贡献，会象今日因特网对网络的贡献一样。”

(http://linuxtoday.com/news_story.php3?ltsn=2000-08-17-001-04-PS-EL)。2002年5月，IBM 刚刚投资了10亿美元，就为了能够让 Linux 在他们自己的所有硬件平台上运行（大型机，工作站，个人计算机，笔记本电脑等）。

0.8 天底下根本没有免费的午餐

使此成真的经济学范例的基础建立在资源稀缺性之上。只有当我们全都使软件成为独有的隐秘的时候软件资源才会稀缺。但Linux不是这样的！当我把我的软件转让给你的时候，它可能会对我造成一种机会成本，但即使我把软件转让给你，我还是能够保有它。这就是历史中的物质利益和免费午餐之争 (Brett Bazant <bbazant@shaw.wave.ca> (<http://linuxtoday.com/cgi-bin/showtb.pl?tbsn=12450&sn=5418>))

0.9 我需要高的系统安全性。购买商业软件，如果出了问题可以告他们

不要试图把诉讼作为一个手段。全世界每天那么多微软的 Windows NT 机器出各种各样的问题，还没有听说过法庭作出什么判决。仔细阅读一下你的 Windows 用户许可证协议，看看能不能找出对于软件无法正常工作有什么保证没有？试图去打官司只会浪费你的银子。

Linux 也没有提供什么保证，尽管它比目前任何版本的 Windows 都要安全得多。如果你真得对安全性非常在乎，你可以使用一些公司提供的高安全性的程序，这些产品必须依赖于有效的源代码来设置和测试它们的安全特性（比如，德国的 Kryptotom 公司提供高安全性的防火墙）。微软 Windows 的“安全黑箱”已经被多次证明是一种很幼稚的做法。

“安全性的考虑是你应该使用Linux和其他开放产品的原因，而不是Windows NT。Windows NT的安全根本不在你的控制范围以内。即使有什么坏事情发生，你也根本不知道该去告谁。为什么人们至今还是相信这样的神话，说Windows NT应该会比Linux会有“更好”的安全性，这实在让我感到很奇怪。”（以上由一个德国的研发工程师David Kastrup发布于因特网周刊 "Internet Week," <http://www.techweb.com/se/directlink.cgi?INW19990329S0050>）.

0.10 我需要标准。大的软件公司提供标准

也许这是人们最期望从大公司那里得到的了，但是事实却远不是这么一回事。一些大公司一度非常热衷于发明一些只有在他们自己的服务中心才能够拆解的东西。这些东西最终变成标准了吗？没有。它们没有变成标准的原因是因为没有公众利益的卷入和参与，或者说因为是专利产品导致公众无法参与。幸运的是，我们现在有了开放和免费的标准。一个“专有的标准”简直是荒唐的产物，很难相信有良好教育的人们会相信它。

一个计算机领域的例子，“标准”的微软 Word 文件格式在过去几年已经修改过无数次了。这种现象持续发生，很可能是因为这是生财有道的绝好办法：一旦其他公司已经成功“逆向解析”了 Word 的文件格式，微软马上做对应的修改。甚至对于一些“子格式”（你听说过“快速存盘”的文件格式吗？）也一样。而且这是在绝对封闭的条件下进行，微软根本不会为此公布任何的信息。用户如何能够从长期使用中获益？微软能够保证 Word 6.0 的文件格式在 2020 年的 Office 里还能够继续有效吗？

“微软的标准不仅自家专有而且一意孤行，在 Office 97 的文件格式和旧版本的 Office 文件格式之间秘密的不兼容性，即使在开放标准的 XML 下面也带着自家的扩展格式要求 Internet Explorer 5 或者微软的 Active Server 才能够阅读，诸如此类的事，是这个公司如何控制市场的一个再明显不过的标志”。

“ 微软的垄断没有保证你现在的 Office 能够和以前或者以后的 Office 互相兼容。不管多少微软的辩护者一直坚称微软的垄断有利于产生一个工业生产的标准。 ”

为了引起混淆，公司的做法一般不去“ 标准化 ”文件格式而是“ 标准化 ”那些用来生成这些文件的应用程序。这就像“ 标准化 ”生产东西的工厂但是不“ 标准化 ”所生产的东西。如果“ 文件生产工厂 ”不断的变换文件的格式标准来促进它们的销售，会发生什么事情？

“ 我们需要标准，开放的文件标准有利于不同的平台之间可以方便的交流资料。用于处理这些文档的字处理程序不应该成为一个主要的问题 ”。其实有几种文本资料的文件格式比微软的字处理程序更标准：ASCII，XML（带非私有的格式表），HTML，SGML，Latex，TEX，PostScript，pdf，dvi...这些文件格式在 Linux 下都得到很好的支持。微软的 Word 文档在 Linux 上的 OpenOffice 应用程序下也可以很方便地读写，可以满足你的需要。更主要的差别是，OpenOffice 文档使用的是 XML 的开放标准，这个标准是由独立的组织控制，而不是某个公司。

Linux 的最基本的特性，就是基于真实，公共资料和免费标准，因为“ 开放源代码 ”使得全部的相关设计资料对所有人都有效，不管他们是不是竞争者。我认为开放标准是 Linux 的真正动力源泉。当然，也有一些人觉得承担不起对另外一个商业实体所提供的算法和数据的信任，一直重复的说这是“ 不可信任的 ”等等，只好由他去吧。

Linux 社区有一种很强烈的预感，在“ 计算标准 ”方面，将来可能会有一个和大型软件提供商竞争的问题。这个包括他们的标准对于我们的“ 静态数据 ”，还有在我们计算机代码后面所绑定的知识。我们能不能请得起某个人来替我们做这样的决定：需要多长时间，以什么样的代价我们可以完成我们的工作？这个问题被忽略了，甚至使那些真正关心这个问题的人感到压力很大。

因为，Linux 必须是一个完整，全面的解决方案。

0.11 我需要用微软的 Windows 才能读写 Word 文档

在大公司的环境里，你可能没有多少选择的余地 他们习惯关起门来，为自己制作的一些不可转移的表格，模板，Visual Basic 设计的页面等等而沾沾自喜。

在相对小的环境里，你可以用在 Linux，Windows，Mac，Solaris 等不同平台上运行的 OpenOffice.org 的程序来制作文档，这些文档带有不依赖于任何平台的文件级的兼容性。OpenOffice 软件可以免费下载和安装且没有任何限制，所有也没有人应该抱怨文件的格式，当然一些试图取得控制权的家伙可能会☹。这里需要特别说明的是，它可以导入和导出微软的 Word 和 Excel 那些相对复杂的文件格式。然而，它的原有文件格式基本上来说更好，因为不是专有的文件格式。在功能上做比较，它几乎可以做微软 Office 可以做的所有事情，附加上一些自己的特有功能。在使用的难易程度比较，对于那些有微软 Office 使用经验的人，可能的会说“ 难上 50 % ”，有的会说“ 容易 20% ”，就看你是问谁了。非常复杂的文件你可以做成 PDF 的格式，而对于一般的文件，用 OpenOffice 可以非常轻松地对付。

所以看来，你可能不再需要微软的 Office 了。赶快到这里去下载给 Linux 和 Windows 的 OpenOffice 的软件吧！<http://www.openoffice.org/>

0.12 微软的 Windows 的普及性能够保证长久的使用

这个看起来好像是事实。“任天堂”可能也能保证“长久的使用”。但是，我需要计算机的“计算能力”；所以我应该选择一台带有强大功能操作系统的计算机，不是那些便宜好用性能一般，设计给“所有人”使用的计算机。

因为系统是基于开放源代码，Linux 不可能因为某公司的倒闭而失去支持，在这方面 Linux 有其非常积极的一面。这个被无数项目选择出来的标准不可能消失，而且这些项目很多是用来执行关键任务的。去看看国际空间站(International Space Station)，在那里 Linux 是他们的操作系统。

还有，从来不要低估 Linux 社区的能力。至少对于那些计算机的菁英们来说，Linux 才是一个能够“久呆”的地方。有些“Linux 人”甚至不想把 Linux 变得太普及，因为他们担心这可能会使天才辈出的 Linux 论坛降低级别。

0.13 但是 Linux 可能会被“肢解”成很多不同的系统

这是属于那些擅长于市场策略的人散播的典型的争论，总是对他们的竞争对手的产品表示“忧虑，不确定和怀疑”。

“肢解”在这个段落里意味“拆解一个计算机程序”，接着用之产生并行的程序的“子版本”，接着“分解”整个 Linux。

可以说，在几个主要的 Linux 发行商那里很少看见有害于 Linux 的“肢解”软件的事情。的确是有些“拆解”的事情发生，但是都是有好处的。对于 Linux 来说，没有什么人工设置的障碍来阻止“拆解软件”，也没有什么人工设置的障碍来阻止将来把那些最好的部分重新拼装起来。

“拆解软件”带来的好处，用以前德国哲学家黑格尔对方言研究得到的理论来说，是一种“精神现象”，黑格尔的结论是：“...分歧在一个事物的内部开始出现，看起来好像是一种不幸，为何不试着解释一下它的好处呢”

0.14 Linux 只是一个小帮派

Linux 团体曾经多次被一些有名的计算机杂志的专栏作家们冠以“宗教狂热分子”的帽子，他们在杂志上发布很多针对 Linux 的不公平的言论，而后收到不计其数的用户反馈。没错，Linux 团体数量庞大，教育良好，有强烈的愿望发表自己的观点。因为很多计算机专栏作家知道 Linux 没有多少钱可以赚，因为用户可以不用花很多钱在他们需要的计算和附加的商业广告上，而这些计算机专栏作家则希望更多一些。

让我们真实面对它吧，你们这些销售人员往往假称自己是专栏作家，而计算机出版社其实几乎没有非常实在的东西。在 1999 年 1 月 1 日之前出版的东西有多少是和 Linux 有关联的？在那个时候，Linux 至少也是一门很值得注意的技术。当然是！但是你们偷偷摸摸的把你们的读者锁在黑暗里，然后卖给他们那些所谓的“专栏文章”来赚取大量的钱财。现在，Linux 已经在主流媒体里（非计算机类）浮出表面，于是你们就开始写一些旨在误导读者的文章说“是啊，但是它(Linux)可能/也许不能/也许会.....”，诸如此类的话，说穿了就是要用“忧虑，不确定，怀疑”的策略来

把它杀死。然后再加上“微软已经……”之类的话，继续写一些垃圾来歌颂那些广告吹了无数遍却从来还没有真正实现过的东西，把那个速度奇慢，摇摇晃晃，标价惊人，架构先天不足的计算平台标榜为未来的“天堂”。它的“伟大成就”已经引起强烈而广泛的争论：它就是通过拒绝互操作性和杀死已经出台的标准（通过“拥抱”然后“私自扩展”它）赚取了“闻所未闻”的巨大财富。你在为谁服务？肯定不是你的读者。

0.15 Linux 总体拥有费用(TCO)太高

** 总体拥有费用(TCO)是 Total Cost of The Ownership 简称，其涵义为保有一台计算机而花费的全部费用，例如硬件购买费用，软件购买安装费用，对使用者的教育训练费用，管理维护费用等从计算机买入到废弃为止的全部费用。

没有人能够真正计算出一个通用软件的“总体拥有费用”。所有，一个好的律师和会计可以提供足够证据来证明他们为了软件总共付出了多少，这也是他们一般的做法。

就以微软的 Windows 的“总体拥有费用”为例，让我来给你做一个简单的估计。把微软的所有 Windows 软件制作人员的财富加在一起，加上一般的 Windows 程序员，策划人员，支持和培训人员，IT 的管理人员等等的财富。现在，再加上那些由于软件公司不断推出他们的“新功能”软件来达到今日垄断地位而已经确切受损的客户所失去的财富。把这个数值除以年数（时间段由你来选择），和微软 Windows 的用户数量（只计算那些一般来说会购买软件的国家用户数）。这就是微软 Windows 的“总体拥有费用”。不管你怎么计算，每人每年的费用应该都在几千美元以上。你没有付那么多？不，你肯定付了，只是你看不到而已。没错，发达国家每年浪费数十亿美元在软件上面。

Linux 要花多少钱？很难说。尽管用户数很少，但是你还是很难把它计算到超过每人每年 10 个美元。

我的个人观点，总体拥有费用不是我们看到的这么简单。我从花出去的钱里得到什么？也许应该计算一下“总体拥有价值”(TVO-Total Value of Ownership)，然后减去“总体拥有费用”得到“总体拥有净收益”。

我猜，会计人员谈论软件“总体拥有费用”只会强调说“是因为商业的需要”，而对价值和利益却只字不提。一般的商业软件其实是没有真正可以衡量的价值的，而只是为了满足当今商业社会的需求。好了，Linux 满足我的计算任务要求，而且几乎不要付什么费用。个人的愉悦和学习的价值则更加美妙。

0.16 Linux 只是一个理想的梦，统治这个世界由来已久的是商业

把 Linux 想象成一个“合作产物”，商业团体和个人协同工作来解决一般计算的需要或者问题。在劳动力和费用都比较便宜，再雇佣一个技术指导，大家协作起来工作让 Linux 符合他们的计算需要。解决方案全部是属于他们自己的，所以可以长久保存。而且花费肯定不会太大，因为他们可以充分利用 Linux 系统下的很多既有成果。

Linux 可以说是由这样一些“设计来符合他们自己需要”的行动，通过松散“合作”后的最终产物。所有说，Linux 本身就是一种商业，但是没有销售软件的必要——它只是付出非常合理的软件价格来符合你的需要，解决你的问题。而且这完全是你自己的东西，软件许可证永远不会过期，你也从不会失去程序的源代码。

0.17 Linux 太滥了

那就别用 Linux 好了。你用 Linux 不是帮什么人的忙。GNU/Linux 是免费而且功能强大的软件，但仅仅是给那些喜欢它或者需要它的人。有不少其他的软件供你选择，也许它们会更加符合你的需要。尽管大部分 Linux 人乐于看到 Linux 的快速增长和新用户的不断加入，但是还是有些人不是很喜欢因为他们担心“拥挤的人群和商业总有一天会沉溺我们一手创建的黑客天堂”。所有，你使用 Linux 不是在给什么人面子。

在这一章最后，再简单地总结一下 Linux 的优势和弱势：Linux 属于它的拥趸们（你的所有权来自于你在这个“Linux 拥趸俱乐部”“的免费登记），毫无疑问 Linux 功能强大丰富，配置灵活，出色的适应性随时符合你的需要（在配置时可能会有点复杂），很低的硬件费用，网络功能齐全，要求有一个计算机学历背景的管理员，一些基本的桌面软件还落在一些其他平台的商业软件产品后面（比如说，在 Windows 上的字处理和电子表格软件），很多出色的最终用户软件作为操作系统的“标准安装”免费得到，图形用户界面已经非常不错了，当然还没有完全象 MAC 和微软的 Windows 所提供的那么精致，Linux 高度标准化（UNIX, POSIX），开放的文件格式可以长久使用，成千上万的程序可以从网上自由下载（尽管可用性和质量可能会有所差别）。

总的说来，一句话：Linux 真的会让你感觉很爽！

第 1 部分: 安装 Linux 之前

Linux 新手管理员指南(中文版)

LINUX NEWBIE ADMINISTRATOR GUIDE (Simplified Chinese Version)

中文版本 1.0 由幻影快递 Linux 小组(SLT-Shadowfax Linux Team) 翻译整理

Linux Newbie Administrator 英文原版版权所有: Peter and Stan Klimas.

<http://sunsite.dk/Linux-newbie>

中文版本版权所有: 幻影快递 Linux 小组 ShadowfaxLinux@hotmail.com

本章内容:

- 1.1 我应该选择哪个 Linux 发行版本?
 - 1.2 LINUX 对硬件有什么要求?
 - 1.3 我的硬件支持 Linux 吗?
 - 1.4 怎样从网上下载 Linux?
 - 1.5 怎样获得 Linux 安装光盘?
 - 1.6 我有 RedHat 安装光盘,但没有安装软盘,应怎么办?
 - 1.7 安装前需要阅读哪些资料?
 - 1.8 我能在一部计算机上同时安装 Windows 和 Linux 吗?
 - 1.9 应该怎样给硬盘分区?
 - 1.10 我的硬盘被 WINDOWS 分区占满,我能缩小 Windows 分区而不用重装系统吗?
 - 1.11 怎样开始安装?
 - 1.12 安装 Linux 困难吗?
 - 1.13 我应该安装哪些软件包?
 - 1.14 我应该选择哪一个图形用户界面, KDE 还是 GNOME?
 - 1.15 我已经完成安装了。我该如何第一次登录?
 - 1.16 我怎样才能把我的 Linux 搞崩溃?
 - 1.17 我可以总是在图形用户界面下使用 Linux 吗?
 - 1.18 我该如何升级 Linux?
-

1.1 我应该选择哪个 Linux 发行版本?

不同发行版本间的区别是比较小的: 安装程序, 精选的工具/应用程序包, 硬盘管理(所有发行版本中, Linux 的大部分内容都在硬盘的相同位置)。 无论你选择哪个发行版本, 你最终都得到本质上相同的 Linux。

基于以下原因, 我们最常使用 RedHat Linux(也称为 RedHat 或 RH):

1. RedHat 很流行(无论是对新手还是专业测试者)。

2. RedHat 是一个针对大多数用户的发行版本。
3. RedHat 的安装程序相对比较简单。
4. RedHat 对 Linux 的扩充都是“公开源代码”的软件(这意味着 RedHat 发行版本 CD 中所包含的所有软件以及所有 RedHat 开发的软件都遵守“通用公开软件许可协议”, GPL(General Public License), 所以任何人可以合法的拷贝, 转赠, 再开发等)
5. 如果你不需要商业化的支持, 你可以花很少的钱得到 RedHat, 甚至免费得到。这是因为 RedHat 是“公开源代码”软件。

简而言之, 作为新手, 除非你需要其他的东西或有特殊的要求, 否则你最好使用 RedHat。RedHat 最新的版本是 2002 年十月发行的 8.0(名称为 RedHat 8.0)。之前的版本是 7.3, 7.2, 7.1, 7.0. 等等。如果从零售商那里购买, 请注明你所需要的版本号, 因为很多零售商为了清理库存, 会卖旧版本给你(不光是 Linux 零售商会这样)。本文的作者和 RedHat(或其他 Linux 发行者)没有任何联系。

我们向新手推荐 RedHat, 并不否认其他 Linux 发行者在某些领域提供比 RedHat 更好的功能。我们相信多个 Linux 版本并存对我们是有好处的, 因为他能帮助我们更好的理解 Linux。

我们曾试用过 Debian(译者注: 另一种 Linux 发行版本) 而且很喜欢它。它几乎和 RedHat 一样容易使用。但是, 它似乎没有 RedHat 这样通用(因此, 我们为新手选择了 RedHat)。Debian 最大的好处是, 它是 100% 的非商业化(由黑客自愿者开发, 维护, 是真正的 Linux 方式), 而且可能是最完全贴近 Linux 标准(可能这些标准也是他们制定的)。Debian 的另外一个好处是, 它发行了无数的 CD, 包含了几千种工具和应用程序---比其他的发行版本要容易的多。所有这些工具/应用程序都被他们很好的组织, 打包(便于安装)而且测试兼容性。这就使得 Debian 看起来安全, 但有一点保守, 过时。综上所述, 我们毫不反对向新手推荐 Debian 作为一个一般用途的 Linux 发行版本。

Corel 公司自己的 Linux 正在进行 beta 测试。他们的目的是开发一个较好的 Linux 平台来运行他们的办公应用程序: WordPerfect, WordProcessor, QuatroPro 电子表格, Corel presentation, Paradox 数据库, Coreldraw 艺术家软件包, 等等。Corel Linux 是基于 Debian, 因此, 如果你将来打算用 Corel 软件, 或许你现在应选择 Debian(而不是 RedHat)。

Slackware 似乎在“去芜存菁”的黑客中比较流行。他们更喜欢贴近操作系统而且会很频繁的升级操作系统的内核。这些是道听途说, 因为我们不用它。SuSe 在欧洲很流行, 它是具有德国风格的一般用途的 Linux 发行版本, 它安装简单且声誉良好。另外一个著名的, 且备受尊敬的是 Caldera, 其设计目标是针对公司用户。它的安装程序可能是最简单的了。Mandrake 是 RedHat 7.2 的克隆, 区别是它比 RedHat 升级的频率更快(为了不至于生出更多版本, RedHat 倾向于较少升级)。最近 Mandrake 看起来很流行。许多国家也有他们自己本地化的 Linux(中文版, 韩国语版, 日文版)。他们一般都带有该国文字的文档, 以便本国用户阅读。

此外还有“特殊目的”的版本, 象“实时版”Linux(多用于自动控制和机器人工业), “最小化”版本(可能会用于下一代的 CD/MP3 播放器, 或者基于软盘运行的 Linux), 并行运算和集群系统 Linux(如果你想做自己的天气预报系统, 也许它会很有用☺), 等等。这里的区别比较大, 但这些都不是很一般意义上的“一般用途”。作为新手, 你可能不会想从这些中的一个入手, 尽管你可能会去冒这个险(他们当然显示了 Linux 的力量和生存能力, 因为 Linux 不仅能在玩具上运行, 同时运行在集群系统上的 Linux 还是世界上运行最快的计算机系统)。

当然你所选择的版本还必须支持你的硬件。这就意味着如果你的 CPU 是 Intel 386,486, 奔腾, cyrix, K6 或类似的 CPU, 你就必须选择“Intel”或“I386”(除非你自己想编译 Linux 源代码, 这对新手来说比较困难☹)。这是因为 Linux 还有基于其他平台的版本(PowerPC, Alpha, 或许更多)。不要试图在你的计算机上装这些, 行不通。

简而言之, 尽管新手可能会被不同的 Linux 版本搞的眼花缭乱, 但要知道, 这些版本的存在是有理由的。这应该被认为是 Linux 的一个优势而不是缺点。

本指南主要针对 Intel 平台的 RedHat5.2, 6.0, 6.1。许多问题的答案应该同样适用于其他平台的其他版本, 但是, 我们没有试过。

1.2 LINUX 对硬件有什么要求?

“out-of-box”Linux 能够在一台 386SX, 8M 内存的计算机上运行。但这样的低端计算机只适合运行字符界面的应用程序(非 X-Window)。一台 16M 内存, 600M 硬盘(没分区)的 486 计算机就可以运行 X-Window 了。我的 64M 内存, 主频 90 的 586 运行 Linux 简直要飞起来了。我的 64M 内存, 奔腾 2-233 的计算机, 即使同时运行多个大型应用程序, 也能得到即时响应。我的 8M 内存, 1G 硬盘, 486-33 计算机在图形方式下运行稍微有点内存不足, 但在我家的网络环境中还是很有用的(486 计算机如果能 有 16M 内存就可以运行的很好)。我那 9M 内存, 120M 硬盘, 386SX-20 的老 Toshiba 便携计算机, 运行一些 Windows 3.11 下的老程序, 当连到家里的 Linux 网络时, 依然很有用。我们曾试过在这台机上运行 Debian, 它在字符方式下工作的挺好。

如果你愿意跳过一些额外的环节, 你甚至只要 4M 内存就可以安装并运行 Linux, 但这样做对于家用的一般用途的 Linux 来说可能有些不值得。我建议, 用 16M 内存, 如果可能, 用 32M 或 64M。因为你同时运行几个大型的图形界面程序时, 较多的内存会有较好的性能。要知道, 现在内存很便宜。

网络功能是 Linux 的一个闪光点, 所以应考虑买一个 10M 的以太网卡。它不很贵, 但却能将你家里的两台或更多的计算机连起来。也许你可以寻找一些 Windows 认为太老的网卡—你可以用很低的价钱买到, 但却能在 Linux 下很好的工作。如果只连两台计算机, 一条以太网交叉网线足够了, 如果想连多于两台的计算机, 你需要一个集线器(HUB)和普通网线(非交叉网线, 如果你有更多的网卡, 你可以考虑在一台计算机上装两块网卡, 通过交叉网线直接相连, 这样就不必用 HUB 了)。10base-T 系统使用 RJ45 接头, 所有的计算机都连接到同一个设备上(称为集线器)。集线器有一个额外的接口(称为上连接口), 你可以通过它连接到外部网络。

你也可以用同轴电缆建立一个较老类型的网络。这种情况不需要集线器, 因为这种网络技术比较老, 你几乎可以用免费得到的零件和网卡来建立:

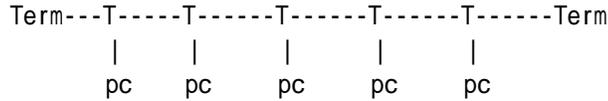
(为节省空间, 下面邮件经过编辑)

From: Edwards@brunel.ac.uk
Subject: 对 Linux 指南的一个建议

你好。好多老式的网卡都有一个同轴电缆接头, 当一些公司将网络升级到 UTP 网络后, 你可以向它们要一些这样的网卡。为每一部计算机加一个 T 形头, 并且在同轴电缆的两个尽头各增加一个 50 欧姆的终结头, 你就建成了一个可以接入很多台电脑的家庭网络。最重要的是, 不用使用昂贵的集线器(或者, 便宜的集线器容易出问题)。同轴电缆还有其他的优点: 它不容易损坏, 而且对其他设备的抗干扰能力较强。

缺点:对于细同轴电缆,每个网段不能超过 186 米,只能接入 30 台计算机,只能固定在 10M。但这对一般家用的网络已经足够了。此外如果连接其中一台机的网线有问题,整个网络都停止工作。除非某个人拔下某段网线,否则这种情况应该不会经常发生。你可以拔下某台机的 T 型头而不会影响整个网络。

简单图示如下, T 代表 T 型头, Term 代表终结头:



同轴电缆的连接头有点类似于电视/录像机的连接头,不同的是它有一个挂钩,用来将连接头和 T 型头紧密的连接起来。

1.3 我的硬件支持 Linux 吗?

Linux 不是支持任何计算机硬件的,而是支持比较老的,或者比较标准的,或比较流行硬件。这个原则适用于以下计算机部件:SCSI 卡,CDROM,可写光盘和可重复写光盘(CD-R 和 CD-RW),显卡,打印机,鼠标,调制解调器(MODEM),网卡,扫描仪,活动硬盘等。

最著名的例外是所谓的“软调制解调器”(即基于软件的调制解调器)。请尽量避免用它,因为它虽然比“硬调制解调器”便宜一点,但却是有缺陷的(有些功能是由计算机的 CPU 处理的,而不是由调制解调器处理的)。而且目前不大可能在 Linux 下使用“软调制解调器”(更多信息,请参阅:<http://www.idir.net/~gromitkc/winmodem.html>)。外置调制解调器肯定不会是“软调制解调器”。所以如果你没把握判断一个调制解调器卡是“软调制解调器”还是“硬调制解调器”,就买一个外置调制解调器(它虽然贵一点,但却不占用你计算机内部电源,很容易从一台计算机换到另一部计算机,容易看到计算机的状态)。此外在选购调制解调器时还应该注意:老的外置调制解调器使用 Rockwell 协议,效果不是很好,而 Linux 对新一点的 USB 调制解调器(2001 年三月后出现)支持的也不是很好。

另外一个有可能出问题的地方是显卡。如果你有一个最新型号 3D 显卡,或者比较少见的显卡,你最好到 <http://www.Xfree86.org> 去检查兼容性。

Linux 很好的支持所有型号的活动硬盘。

尽管制造商声称并口扫描仪是充分兼容的,但我并不指望 Linux 支持并口扫描仪(非 SCSI 口)。

因此,总的来说,如果你的计算机用的都是标准部件,运行 Linux 应该没有任何问题。要使用 Linux,你不需要花很多钱,尽量在你现有的计算机上安装 Linux。如果你想检查 Linux 是否支持你的比较新的部件或不太流行的部件,请到以下网址:

<http://hardware.redhat.com/hcl/genpage2.cgi>

和

<http://metalab.unc.edu/LDP/HOWTO/Hardware-HOWTO.html>

每次购买新硬件时,我都会到以上网址检查 Linux 兼容性。你也可以问厂家或零售商,但他们不一定能准确告诉你,因为现在很多公司的销售人员或技术支持不是很懂。如果购买一台全新的计算机,我

会考虑选购一台已经预装了 Linux 的系统,许多厂家(尤其是大厂家)提供这样的系统,也有很多不提供。但大的硬件生产厂家一般都会提供的。

如果你象我一样喜欢冒险,就不用理会上面的告诫,硬件兼容的可能性会是 90%。

如果你现有的 Linux 版本不支持某个硬件,不要放弃,有可能是以下原因:

- (1) Linux 是支持的,不过你不知道怎么配置(解决办法,先放几个星期,不要浪费时间。等你对 Linux 了解多点后,也许你会配置了)。
- (2) 你必须经过比较复杂的设置,才能使它工作(例如,一些你不知道的命令,或重新编译内核。注意,这些东西并不象想象的那么难)。
- (3) 新的 Linux 版本(或其他版本)或许支持。
- (4)或许厂家或有人已经升级了该硬件的软件,放在 internet 的某个地方,你需要去寻找,然后试着安装。
- (5)或许下个月 Linux 就支持了—因为 Linux 的开发速度的确很快。

1.4 怎样从网上下载 Linux?

请帮你自己个忙,尽量不要下载 Linux,而是订购一个安装光盘。Linux 的确可以完全从网上下载,但作为一个很大而且很复杂的操作系统,下载可能要花很长时间,而且下载过程可能会出错。

如果你的确是高速上网(绝对不是 56K modem,有可能是宽带网),而且你不是新手,下载 Linux 毕竟是你的一个选择。从 <http://www.linuxiso.org> 选择一个你喜欢的 Linux 版本的 ISO 映象文件,然后刻成光盘。

1.5 怎样获得 Linux 安装光盘?

有很多种可能的方式。

- (1) 从 Linuxmall 里(<http://www.linuxmall.com>)购买 RedHat 或 Mandrake 光盘。或者从 Cheapbytes(<http://cart.cheapbytes.com/cgi-bin/cart/>)。上次我看的时候,非官方的 RedHat 只要 2.99 美元,两张 CD。他们会只寄给你光盘,没有手册,没有支持,也没有启动软盘。但价钱是不错的,而且手册和用来制作启动软盘的工具都在第一张光盘里。我从 Cheapbytes 买了几次,都是很快就到货,而且质量不错。而且我的信用卡也没有被乱扣钱(声明,作者和 Cheapbytes 没关系)。
- (2) 从相同的地方购买盒装的 RedHat 或 Mandrake,或者从你喜欢的任何零售商那里购买。价钱可能最低都要 40 美元,你会得到手册,安装时电话或 email 支持(60 天内?),启动软盘,此外还有商业应用软件作为俯送产品,可能还会有其他的東西。如果你的时间的确很宝贵,你还可以花更多的钱买更多的服务。
- (3) 从你朋友那里复制。这是合法的,因为 Linux 是免费的。如果你有一套 Linux 光盘,不要不舍得借给你的邻居。

(4) 查询你附近的图书馆,或书店,或者 <http://www.amazon.com>。有些 Linux 工具书会在附送的光盘里带有完整的 Linux。对初学者,这是一个比较好的方法,因为你可以参考工具书。对于 Linux 数不清的工具,我需要一本工具书在身边。但缺点是这些书都有点过时。或许应该考虑官方的 Linux 和手册?

(5) 参加一次当地的 Linux 交流会。本地的 Linux 高手会免费为你安装 Linux(带着你的计算机),而且你还有可能免费得到一张 Linux 的光盘(也许你应该带张空白光盘去)。搜索 Linux 新闻组,看看什么时候有交流会在你附近举行。这也是一个认识其他 Linux 爱好者的好机会。

1.6 我有 RedHat 安装光盘,但没有安装软盘,应怎么办?

如果你的计算机能从光驱启动(许多早期的计算机做不到),你就不需要启动软盘了。检查一下计算机的 BIOS,你可以在那里设置启动顺序(缺省的设置是先软盘,再硬盘)。尽管我的计算机设置为先从光驱启动,但依然不能从光驱启动,所以有时 BIOS 并不能真正的反映你的计算机到底是从那里启动的。如果你能够从光驱启动,只要将 RedHat 安装光盘放进光驱,重启计算机,就可以进入 RedHat Linux 安装程序了。

如果你不知道怎样进入 BIOS,请阅读以下信息。一般地,在计算机启动时,按下正确的键(通常屏幕上有提示),就可进入 BIOS。一般是键。以下是流行的 BIOS 所用的键:

宏基笔记本:	在加电自检后按<F2>
AMI(American Megatrends):	在加电自检后按
Awards:	在加电自检后按或<Ctrl><Alt><Esc>
Compaq:	在启动时右下角出现方块后按<F10>
Dell:	<Ctrl><Alt><Enter>
DTK:	在加电自检后按<Esc>
IBM Aptiva 535:	加电后右上角出现带波浪线的方块后按<F1>
IBM PS/2:	先按<Ctrl><Alt>,在右上角看到光标后按
<Ctrl><Alt><Ins>	
Mr.BIOS:	在加电自检后按<Ctrl><Alt><S>
Packard Bell:	有些型号在加电自检后按<F1> 或者 <F2 >
Phoenix:	<Ctrl><Alt><Esc>,<Ctrl><Alt><S> 或者 <Ctrl><Alt><Enter>

如果你的计算机不能从光驱启动,你需要在 DOS 或微软 Windows 的 DOS 模式(仅仅用一个 DOS 窗口是不够的,你必须关机,切换到 MS-DOS 模式)下制作启动软盘。

一张好的软盘很重要(甚至不能有一个坏道)。制作安装软盘的程序并不检查软盘是否正确的写入。不要指望 DOS 的格式化程序帮你判断软盘是否坏了。如果我是你的话,我会同时制作两到三张软盘(你可能到那时才会相信有多少软盘有可能是坏的)。以我为例,只有第三张软盘是好的。如果你制作的软盘不能启动,就另外再做一张,应该就可以了。

以下是制作软盘的命令:

```
F:\dosutils\rawrite.exe -f F:\images\boot.img -d a: -n
```

制作附加软盘的命令是:

```
F:\dosutils\rawrite.exe -f F:\images\supp.img -d a: -n
```

这是假设你的光驱在 DOS 下是“F:”盘,你的软盘是“A:”盘。如果你的计算机设置和此不同,请调整参数。

上面命令的意思是,运行工具“rawwrite”,指定输入文件(`boot.img`),输出文件(软盘 A:),并且跳过“插入软盘”的提示。也许你会发现不用任何参数运行 `rawwrite` 更加简单,因为它会提示你给出输入文件名和输出文件名。

在 Linux 下,你可以在 `mount` 上 RedHat 光盘后,输入以下命令(你必须是 `root` 用户):

```
cd /mnt/cdrom/images/  
dd if=boot.img of=/dev/fd0
```

[‘`dd`’命令用来拷贝文件。上面命令指定输入文件(“`if`”)为‘`boot.img`’,输出文件(“`of`”)为一个软盘(/`dev/fd0`),如果你想写到第二个软驱,用/`dev/fd1`]

1.7 安装前需要阅读哪些资料?

对 Unix 有一些认识将会对你很有帮助。买一本好的 Linux 参考书,或者找出你以前的 Unix 书。几乎所有 Unix 的概念均适用于 Linux,而且几乎所有的 Unix 命令均能在 Linux 下运行,我发现 MS-Windows 的手册没什么用的(点这里,点这里,然后看屏幕的拷贝)。但 Unix 的手册却很棒(让你了解系统,而且对将来有益)。

也许你要了解你的硬件信息,如:你有多少个硬盘?每个多大?每个硬盘有多少个分区,每个分区是什么类型,多大?你的 DOS,Windows 分区在那里?那个是 DOS/Windows 启动分区(如果你打算让你的计算机双启动的话)?你的鼠标是什么型号的?你的显卡是什么型号的?你的显示器是什么型号的?等等。

到 BIOS 设置中查看你的硬盘的型号和硬盘映象信息(译者注:BIOS 不得不将磁头数,柱面数,重新影象,以支持大硬盘)。在 DOS 下运行 `fdisk` 来查看你的硬盘分区表,并打印它。注意观察系统启动信息,从而了解你的显卡和显存信息。启动到 Windows,到控制面板—设备,记下你的声卡,调制解调器,网卡的类型和设置(名称,型号,中断好,I/O 地址等)。查看鼠标下面的标签,从而确认你的鼠标的类型(下次买鼠标时,购买一个 3 键的 Logitech 或类似的—Linux 很好的支持他们)。阅读你显示器的手册,看你显示器支持的最大刷新频率(水平和垂直)。千万不要超过你显示器支持的最大刷新频率。因为这样会损坏你的显示器。

你也许想浏览 RedHat 或 Mandrake 的手册,如果你没有书面手册,在你的 CD 中有 html 格式的手册,你可以用任何浏览器阅读。点击这里 [here](#) 了解如何阅读这些手册,和你 CD 中一些其他的资料。

1.8 我能在一部计算机上同时安装 Windows 和 Linux 吗?

是的,没问题。很多 Linux 爱好者都使用双启动选项。只要你在你硬盘的一个分区上安装 Windows,并且在另外一个分区上安装 Linux。Linux 带有一个简单的启动管理器,称为 LILO(更复杂的一个是 GRUB),会让你在启动的时候选择启动那个操作系统。你应该先安装 Windows,然后在安装 Windows。否则 Windows 安装程序会让你无法再启动到 Linux。如果你要重新安装 Windows,请准备一张启动软盘,因为 Windows 将会重写 MBR,结果是你无法进入 Linux,所以你必须要用软盘启动到 Linux,重新运行 LILO,配置你的系统支持双启动。

在 Linux 下,你可以读写 Windows 分区的数据,所以数据在 Windows 和 Linux 间的转移没有任何问题。这样你就可以在 Linux 下使用你 Windows 下现有的一些资源,如声音文件,背景,照片,字体等(首先要确认这样做有没有违反版权。如果你使用的是微软租给你的软件,你可能会遇到问题。我很自豪的说,在我的 Linux 计算机里,没有使用任何盗版软件)[译者注:微软已经不使用“卖软件”这个名词,而是使用“出租软件”]

1.9 应该怎样给硬盘分区?

在安装 Linux 前,你可能真很想知道什么是硬盘的分区。你关心这个是因为你不想删除你的 Windows 分区,你想让它们处于不同的分区,实现双启动。这就意味着,Windows 在一个分区,而 Linux 在另外一个分区。通常你不会将 Linux 安装在你 Windows 分区的剩余空间上。将 Linux 安装在 Windows 分区上是可能的,但我们不推荐它。

如果你计划双启动(Linux 和 Windows 在同一部计算机上),首先使用 DOS/Windows 工具 FDISK 制作 Windows 分区,留一些空间给 Linux。Linux 安装程序会在安装 Linux 时找到未使用空间,那时你可以给 Linux 分区,并格式化它。

将 Windows 分区设为主分区,并且设为可启动,安装,配置,在安装 Linux 前测试你的 Windows 系统。如果你只打算运行 Linux,你只需要一块没用过的硬盘就行了。

你可以只给 Linux 一个分区(如果双启动你需要再给 Windows 一个分区)。但是给 Linux 多个分区会更好一点,这样你就可以将用户数据和操作系统分离开来。这样,如果什么地方出错,或者你不得不格式化硬盘,或重装操作系统,你就不会丢失用户的数据(如果在安装时你跳过“重新格式化”,你可以完全重装操作系统而不会丢失保存在/home 目录下的数据。前提是/home 必须在它自己的分区上)。

安装 Linux 时,安装程序会问要求你给剩余的硬盘空间进行分区。有很多种可能的选择,取决于你的硬盘空间,要求和口味。我喜欢这样为 Linux 分区(以 2G 的硬盘为例):

安装点(mount point)	类型(type)	大小(size)
/	ext2	300 MB
/usr	ext2	1200 MB
/home	ext2	380 MB
swap	swap	120 MB

在上面的例子中,我给 root 分区 300M 空间,用来安装 Linux 操作系统。给安装点为/usr 的分区 1200M 空间,用来存放用户的程序(不是 Linux 发行版本带的程序,而是稍后我自己装的程序,例如 StarOffice 等)。我给安装点为/home 的分区 380M 空间,用来存放用户的设置和数据。我给操作系统 120M 空间用于虚拟内存(在硬盘上虚拟的内存空间,也称为交换区)。如果你的内核版本低于 2.2(RedHat5.2 以前的版本),你的交换区不能大于 127M。确定交换区大小的原则是:交换区应该是内存大小 2 倍。如果你需要更多的交换区空间(例如你的内存很大,或是你需要运行数据结构很大的用户程序),你可以将交换区设的更大一点,或是稍后在增加一个交换区文件。

2G 的空间对喜欢运行很多程序的用户来说足够了(这是因为 Linux 比 Windows 下相同的程序要小的原因),但是,如果你想要安装 Linux 光碟上的所有东西,2G 显然是不够的。我的经验是,不管硬盘有多大,最后总是装的满满的,而我总是后悔硬盘不够大 ☺。

如果我的硬盘确实没有足够的空间,我可能会考虑两个分区的安装方式(硬盘空间小于 650M):

安装点(mount point)	类型(type)	大小(size)
/	ext2	600 MB
swap	swap	50 MB

在这个例子中,我给 600M 空间用于安装基本操作系统,用户数据。给 50M 空间用于虚拟内存。50M 的交换区空间对于中等规模任务的系统来说足够了。600M 的空间意味着你必须要有选择的安装一些程序,否则你会面临用尽磁盘空间的危险。安装 RedHat 所带的应用程序时按<F1>,它会给你一个关于该程序的简单描述,然后你就知道你到底是否真的需要它(如果漏掉某些程序,不要担心,你可以稍后在安装)。如果你的选择比较合理,你会很容易的完成 RedHat 的安装,而且还有 200M 的剩余空间(假如你给 root 分区 600M 空间)。请注意将 root 分区和/home 分区绑在一起,虽然能为你省一些空间,但却不是最安全的方案。

你也可以用比上面的例子更少的磁盘空间完成 Linux 的安装,但这会要求你更加慎重的选择应该安装什么程序。

对于更大的硬盘,我推荐如下分区方式(假如你准备给 Linux 15G 空间):

安装点(mount point)	类型(type)	大小(size)
/	ext2	800 MB
/usr	ext2	5000 MB
/usr/local	ext2	3000 MB
/home	ext2	5200 MB
swap	swap	1000 MB

请注意安装点可以在另外一块硬盘上,Linux 将所有的硬盘空间整理为唯一的一棵目录树。

在老型号的计算机(486?)上安装 Linux 时应注意以下问题。许多老的 BIOS 不支持启动分区在第一块硬盘的 1024 柱面之外。为了克服这个限制,你只要将第一个分区设为启动分区,并且小于 1023 柱面(这样做的结果是,/ 分区只能小于 512M,但这已经足够了)。只要 Linux 能够启动,BIOS 的限制就不起作用了,因为 Linux 将接手硬件的管理,它能够存取位于 1024 柱面之外的分区。

在 Linux 中,你的硬盘是按如下规则命名的:

hda	—	第一个 IDE 硬盘(意思为"硬盘 A",例如连在第一个硬盘控制器上的主硬盘)
hdb	—	第二个 IDE 硬盘(意思为"硬盘 B",例如连在第一个硬盘控制器上的辅硬盘)
hdc	—	第三个 IDE 硬盘(意思为"硬盘 C",例如连在第二个硬盘控制器上的主硬盘)
hdd	—	第四个 IDE 硬盘(意思为"硬盘 D",例如连在第二个硬盘控制器上的辅硬盘)

数字代表着物理硬盘上的分区。

"hda1" 意为第一个硬盘上的第一个分区

"hda2" 意为第一个硬盘上的第二个分区

"hda3" 意为第一个硬盘上的第三个分区

(如果你在第一个硬盘上有多于一个的分区,依此类推)

"hdb1" 意为第二个硬盘上的第一个分区

(如果你将光驱连在第一个硬盘控制器的辅设备,就只是 hdb)。

"hdc1" 意为第三个硬盘上的第一个分区,等等。S

CSI 使用相似的命名规则,但却以 sd 开头,然后跟着用来表示 SCSI 接口的字母和用来表示 SCSI 设备的数字。例如:“sda4”意为第一个 SCSI 控制器上的第四个设备。如果你有一个外置的压缩硬盘连到计算机的并口上,它将会显示为“sda4”(压缩硬盘以 SCSI 仿真方式工作)。

Linux 安装程序为你列出的分区,将包括你计算机上已有的任何分区,例如,我的硬盘上有如下 Windows 分区:

mount point	type	size	comment
[no mount]	vfat	1200 MB	[“Win C drive, hda1”]
/mnt/dos_hdd2	vfat	1600 MB	[“Win D drive, hdd2”]

如果你想要支持双启动的话,在 Linux 安装过程中不要删除这些 Windows 分区。如果你删除这些分区,就会删掉 Windows 系统。如果不是很肯定,就在安装 Linux 前先备份 Windows 系统。msdos, fat, vfat, ntfs 通常是 DOS/windows 系统使用的文件系统。

作为一个快速参考,这里是关于 Linux 分区类型的一个简单小结。我从 Linux 手册拷贝过来的。有下划线的文件系统则是你最有可能使用的文件系统。其他文件系统(没列出来)作为附加项也很容易得到。

minix 是 Minix 操作系统使用的文件系统,是 Linux 最先使用的文件系统。它有许多缺点:分区只能小于 64MB,只支持短文件名,单一的时间戳,等等。在软盘和动态磁盘领域仍然有用。

ext 是对 Minix 的扩展。已完全被 ext2 取代,内核最终将不会支持它。

ext2 是 Linux 使用的,性能很好的文件系统,用于固定文件系统和可活动文件系统。它是作为 ext 文件系统的扩展而设计的。ext2 在 Linux 所支持的文件系统中,提供最好的性能(在速度和 CPU 使用方面),简短的说,ext2 是 Linux 的主要文件系统。

ext3 是对 ext2 增加日志功能后的扩展。它向前,向后兼容 ext2。意为 ext2 不用丢失数据和格式化就可以转换为 ext3,ext3 也可以转换为 ext2 而不用丢失数据(只要重新安装该分区就行了)。我从 2001 年 10 月就使用 ext3,它简单且稳定。在 RedHat 7.2 和 Mandrake 8.0,中作为一个选项。强烈推荐这种文件系统。

xiafs 作为 Minix 文件系统的一个稳定,安全的扩展,提供基本的最常用的特点。目前已经停止开发和维护了。不太常使用。

msdos 是 DOS, Windows 和一些 os/2 计算机使用的文件系统。文件名不能超过 8 个字符,然后跟一个 3 个字符的后缀。

umsdos 是 Linux 使用的扩展的 DOS 文件系统。增加了长文件名支持,UID/GID,POSIX 权限,特殊文件等而依然保持和 DOS 文件系统兼容。

vfat 是 Windows 95, Windows NT 使用的扩展的 DOS 文件系统。增加长文件名支持。

proc 是一种假的文件系统,用于和内核数据结构接口,它不占用磁盘空间,参考 man proc。

iso9660 是一种针对 ISO9660 标准的 CD-ROM 文件系统。自动支持两种扩展(看下面)。

High Sierra 是 ISO9660 标准的前身。Linux 自动支持它。

Rock Ridge 是 Rock Ridge interchange Protocol 协议的一个产品,用于进一步向 Unix 系统描述 ISO9660 文件系统下的文件。另外提供诸如长文件名,UID/GID,POSIX 权限,等信息。Linux 自动支持它。

hpfs 是 OS/2 使用的高性能的文件系统。在 Linux 系统中是只读的,因为缺乏相应的文档资料。

sysv 是 SystemV/Coherent 文件系统在 Linux 上的实现。它实现了所有的 Xenix, SystemV/386, 和 Coherent 文件系统。

nfs 是用于存取远方计算机硬盘的文件系统。

smb 是支持 SMB 协议的网络文件系统,Windows 用它来实现工作组共享。要使用 SMB 文件系统,你必须有一个特殊的安装程序,你可以在 ksmbfs 软件包中找到它,也可以在这里找到:

<ftp://sunsite.unc.edu/pub/Linux/system/Filesystems/smbfs>

[标准的 linux 命令"smbmount" 也能做到]

ncpfs 是 Novell NetWare 使用的,支持 NCP 协议的网络文件系统。

Devpts 是一个虚拟的文件系统,一般安装在/dev/pts。为了得到一个虚拟终端,进程打开/dev/ptmx,然后就可使用虚拟终端。

fat 不是一个单独的文件系统,而是 msdos, umsdos 和 vfat 文件系统的常用部分。

UFS 是一个广泛使用于各种操作系统的文件系统

swap 是一种特殊的分区,用于在内存和硬盘间交换数据的文件系统。

raiserfs 是 Linux 内核 2.4.1 以后(2001 年 1 月)支持的,一种全新的日志文件系统。

hfs (=hierarchical files system)苹果文件系统。现在仍然是 beta 版本。建议不要用于重要数据,除非只读。

ntfs--Windows NT 文件系统。在 Linux 下依然是实验阶段。建议生产环境不要使用,除非是只读(截至 2001 年 8 月)。

1.10 我的硬盘被 WINDOWS 分区占满, 我能缩小 Windows 分区而不用重装系统吗?

这是可能的。我的 Redhat 光碟里有个应用程序:fips.exe 就是做这个的。到你 RedHat 光碟的 \dosutils\fipsdocs\ 目录下寻找文档资料。如果我是你,我会在行动前先备份我的分区。也有商业软件可以改变分区大小而不用丢失数据。

我个人比较偏爱先在一个干净的分区安装 Windows,预留一些硬盘空间,不格式化,以便随后安装 Linux。Windows 和 Linux 在我硬盘中各占 50%。Linux 程序虽然比较小,但却提供微软系统要花几

千美元才能买到的功能。例如, telnet 服务器, ftp 服务器, http 服务器, 邮件服务器,几个数据库, 变程语言, 图像处理程序等。

1.11 怎样开始安装?

将安装软盘插入软驱, 或者将安装光盘插入光驱, 然后重新启动计算机。如果你能从光驱启动, 只要将安装光盘插入光驱, 然后重新启动计算机就行了。

你也可以在 DOS 下(或 Windows 的真 DOS 下, 不时 DOS 窗口)开始安装, 只要运行 EZSTART.BAT 就可以了。这个程序在我从 cheapbytes 买的 RedHat 光碟里有。

大多数安装程序会让你选择字符模式或图形模式。如果你的内存不够, 你只能使用字符模式。

1.12 安装 Linux 困难吗?

对我来说不难。其实对于大多数新手, 过程也很直观, 而且一点也不痛苦。基于你的硬件和安装选择, 一般要花上 1 到 2.2 小时。 [对于带有很有限的内存的运行较慢的系统, 安装时间预料会更长或可能出现问题--在我的 486-33 MHz 带 8 MB 内存的系统中安装 RH6.0 要花一整个晚上, 系统出现了一次暂停, 花了 5 分钟却什么也没做, 但安装正常]。从以期的安装版本中升级要花更长时间, 并有出现更多问题的趋向。

可是, 一些新手报告说安装过程对他们是一场 "恶梦" (硬件问题? 没有经验?)。如果你碰到问题, 我的建议是安装基本的系统, 不要安装那些可能你的机器上有的但需要最高安装技巧的硬件如你的奇特的显卡或其它复杂的硬件。在你理解更多关于这些硬件如何在你的系统中工作之后你可以增加/配置这些。甚至重新安装对于新手也是一种选择(似乎 Linux 高手认为重新安装是很没有面子的事)。似乎很多新手安装时出现问题, 原因是他们指定了太高的显示器分辨率(这可能不被支持或仅支持特别地调整)。而且为了支持某个设备而暂停整个安装是不明智的 可以在以后增加/调整这些特别的支持。

1.13 我应该安装哪些软件包?

Linux 所有软件都包含在软件包里。例如, 我的 Linux Mandrake 7.0 CD 安装盘包含了 1002 软件包。Mandrake 7.2 含有 2 张 CD—我安装 Mandrake 7.2 将 1123 软件包放到硬盘上。提醒你, 这不是 Linux 中所有可用的软件—只是由将 Mandrake 软件分类放在一起的人所做的选择。Mandrake 比 RedHat 包含了更多的软件。

不论哪一种版本, 该 CD 除了包含基本的操作系统(内核, 库文件, 命令行配置和维护工具的选择等等), 还带有适当的配置和监测工具的网络 "客户-服务器" 作出选择的软件包, 一些终端用户文本模式的应用程序, 基本的 X-window 系统, 至少一个 GUI 桌面(可能有几个)和许多基本的 GUI 应用程序。

该安装程序(RedHa 或者 Mandrake)会问你要安装哪些软件包。如果你选择 "工作站安装", 那么应从你的安装中忽略那些通常在服务器端被找到的软件包(例如, 可能略过 Apache 网络服务器)。如果你选择 "服务器安装", 那么典型地不要安装终端用户桌面应用程序(例如, GIMP 图形软件

被忽略)。你也能选择安装“全部”，这是我在家庭计算机安装 Linux 软件包时喜欢的选项。最后，你可以按你所选的软件包进行安装——读相关资料。

选择那些名字听起来较有趣的软件包或程序安装确实是很糟糕的想法——一些软件包有很不一般的名字，我从不去猜它们是做什么的。忽略了某一重要的软件包(如库文件)你的系统功能会被削弱。当你坚持运行包含在这个 CD 里的一些名字很酷而且重要的软件(版本“0.1”)时，你可能会觉得失望。总之，你可能会被你定制的 Linux 所展现的功能(或缺少的功能)所困扰。作为新手，有时要相信由你的版本的创建者所选择的默认选项。

所以，为了我安装后的最终“产品”，除非我想安装每一个软件，否则我不会选择安装那个诱人的选项“熟练人员安装—手动安装软件包”。可是对于初学者，我喜欢安全的“最多缺省安装项”，在你的 CD 中可调用该选项。

如果你没装某一软件包，以后你发现你需要它——不要惊慌。安装它很容易。读相关资料。

1.14 我应该选择哪一个图形用户界面，KDE 还是 GNOME?

如果磁盘空间允许，二者均可。你可以在以后决定你更喜欢 KDE，GNOME 或其他桌面，但无论你作何种选择，你确实要安装 KDE 和 GNOME 库文件。一旦你安装了这些库文件，KDE 程序能在 GNOME 下运行，反之亦然，取决于哪些库文件有更好的应用。就磁盘空间的大小而言，桌面程序仅仅是 KDE 和 GNOME 系统的很小的部分，所以你忽略了安装桌面你并不会节省多少空间，试着安装“库文件”。GNOME 和 KDE 是一套很好的程序和工具，所以完全安装这两个桌面程序确实是值得的。我从未听说这二个会互相干扰。在日常工作使用中，我使用 KDE 桌面，因为感觉它比 GNOME 更可靠。如果你喜欢更前卫，更酷的桌面，选用 GNOME，但如果有时不能完全正常工作，不要抱怨。

我也安装了其它的“窗口管理器”选项，它们几乎不占用什么空间(一些确实很小)，但在一些情况下它们很有用。只要安装了 KDE 和 GNOME 库文件，你就能在这些选项下运行 KDE 或 GNOME 程序。

KDE 确实功能更强大。但是在旧的硬件里(如 133 MHz Pentium)，我更喜欢 GNOME 而不是 KDE。其它的窗口管理器比 KDE 或 GNOME 更简洁。所以在适当的硬件下，我会选择其他的窗口管理器。

1.15 我已经完成安装了。我该如何第一次登录?

超级用户。

“root”是初始安装后存在的唯一的帐户(更新版本的安装程序确实提示你在安装期间创建普通用户帐户)。文本模式登录的例子如下

```
my_machine_name login: root
Password: my_password
```

在上面的例子中，我在登录提示中输入“root”。之后，我输入在初始安装时所选择的密码。当我输入时(为了安全起见)，密码不会出现在屏幕上。在我登录后，我发现我处于一个文本模式的终端。

如果你安装了 GUI 登录界面，登录步骤很相似，但是发生在 X-window 界面上。(如果偶尔碰到输入有问题，可能你要记住把你的鼠标停在正确的对话框上。X-window 登录界面实现了“根据鼠标定位”规则)。在成功登录后，我的默认的 GUI 桌面就出现了。

“root”是具有在系统之上的绝对权力的特殊帐户，由系统管理员使用。你以后的确需要建立至少一个以上的“普通用户”帐户来做日常工作(非系统管理员)。读相关资料学会如何操作。

1.16 我怎样才能把我的 Linux 搞崩溃？

作为 root 用户，你可以做任何事，包括加速系统崩溃。如果你真的想崩溃系统，试验一下：

```
cp /dev/zero /dev/mem
```

作为 root 用户，你甚至可以只要用一条简单的命令就可以删除你所有的文件（千万别尝试！）

```
rm -fr /
```

这并不是说 Linux 很容易崩溃，而是系统管理员(root)对系统有无限的权力。对应微软的 Windows 也差不多，你把 c:\Windows 或者 c:\Windows\System 目录里的一些文件删除，就能使整个系统瘫痪。Linux 更加稳定的最重要的不同点是：普通用户和系统管理员用户是完全分开的。普通用户只能操作属于他们自己的文件。类似的，不管程序出来什么大差错，用户运行的程序只能影响他们自己拥有的区域。“管理员”和“普通用户”的帐号截然分开让系统变得复杂，同时使 Linux 变成真正的多用户操作系统。这和微软 Windows 的旧版本的做法完全不同。在微软最新版本的 Windows 里，微软也已经开始逐步走向 Unix 的做法。就像一个老的说法：“那些不知道 UNIX 的人注定要去重新发明它”。

结论：不要使用 root 用户来作日常工作。加一个普通用户帐号作为日常管理工作使用，用这个帐号在 Linux 上做实验，这里显示然后实现：

```
adduser johnbrown  
passwd johnbrown
```

[输入一个长的精心选择的密码，在中间带数字]
[再次输入相同的密码]

```
exit
```

然后以 johnbrown 的用户重新登录

root 帐号只用来管理和系统设置。我从来不会用 root 帐号运行任何我不是非常了解的程序。至少，不会在准备用来做实际工作的计算机上使用。我是说真的。

我的学习过程如下：

1. 安装 Linux
2. 胡搞，做试验（root 或者非 root,做一些有趣的事情，设置乱七八糟的东西。我想要预知系统会如何动作 只有这样我才能够真正了解我在做的事情）

3. 删除系统
4. 除非我没有时间 (时间好像总是不够用), 否则回到第二步, 继续

Linux 一旦在可靠的硬件上安装完毕 命令行模式下 Linux 出奇的稳定 - - 可以运行好几个月甚至几年! 作为一个新手, 由 root 用户的胡乱操作或者很滥的硬件导致的问题, 你会发觉很多稀奇古怪的事情来打发你的时间。

1.17 我可以总是在图形用户界面下使用 Linux 吗?

在 Linux 下只使用图形用户界面我认为是可能的, 但是我觉得对于管理来说, 就不是那么有效了。Linux 下的 GUI 已经做得很好, 但是对比于命令行下的可以使用的功能, 还是显得有些苍白。我几乎每天使用 Linux 和 Windows 的图形用户界面, 我发现图形用户界面对于程序启动和其他日常工作非常有用。但是对于系统管理和自动化, 图形用户界面还是不过灵活。命令行接口 (CLI, Command Line Interface) 对我来说有更丰富的功能, 而且, 使用命令行接口让我对于的系统的工作原理有更多的实际了解。

引用两个典型的说法: “使用图形用户界面相对来说比较简单, 因为它限制只能执行有限的几个基本操作, 这样学习起来也更容易上手。但是, 如果这些基本操作不能符号你的要求, 那你可能会有点问题”。 “用户界面友好的系统的典型特征, 就是它能够符号用户的需要。但是, 大部分用户界面友好的系统都不是最简捷有效的”。 (引用来源:

<http://linux.oreillynet.com/pub/a/linux/2001/11/15/learnunixos.html>)

我发现图形用户界面和命令行的混合使用可能是最理想的。如果你希望只使用图形用户界面而不想学习任何命令行参数, 你可以考虑推迟安装你的 Linux 系统, 一直到图形用户界面工具比现在做得更好一些, 或者, 现在你已经有其他的人能够帮你管理你的计算机。

如果你有计算机超级天才的男朋友或者女朋友, 然后他 (或者她) 可能只要运行十几个程序, 你就可以有一个专门为她 (或者他) 定制的非常好用的图形用户界面, 所有图标, 菜单和按钮都符合要求。呵呵, 通过这样的特别的鼓励方法, 可能可以让她 (或者她) 很快就接受 Linux!

从这本指南里, 对于做资料来说使用命令行肯定要简捷得多。图标和菜单一般都是可以定制的, 所以你在你的计算机上看到的图标和菜单和我在我计算机上看到的肯定很不一样。还有, 设置一个图标或者命令, 我必须知道我在图标后面的真正的命令行的位置。简而言之, 对于刚开始阶段, 即使在图形用户界面下, 理解命令行对于系统设置来说也是必不可少的。我们在这本书里没有包含任何的屏幕截图, 是因为我们自己发现对于学习计算机, 屏幕截图没有什么实际的用处。

1.18 我该如何升级 Linux ?

对于一次完整的系统升级 (比如从 RedHat 7.0 到 RedHat 7.2) :

“对于升级 Linux 系统最省时有效的原则就是: 从零开始。做一次 “干净” 的系统安装, 然后再把数据加上。是的, 我们就是在说把硬盘重新分区, 然后从头开始安装。” (引用 URL 来源

<http://www.northernjourney.com/opensource/newbies/newb025.html>)

全盘考虑你的硬盘分区方案，会有助于你未来的系统升级（你的数据文件可以放在你的 home 目录下）。

以下是在系统升级时，我给自己做的一份清单供参考：

1. 以 root 登录
2. 将/etc 目录放到一个安全的数据不会被破坏的地方（比如 zip 盘）。当安装出现问题的时候，我可以参考以前的设置。
3. 复制所有在/home 目录上的重要数据。这是为了避免更糟糕的事情发生，比如，有一次我格式化了错误的分区
4. 使用 df (diskfree)命令查找和记下你文件系统的所有加载点
5. 举个例子，我的 home 目录加载在一个分区名字叫做/dev/hda9 的另外一块硬盘上，而 /usr/local 则加载在/dev/hda8 上。
6. 重新安装 Linux，但是保留和不要重新格式化那些你用来保存数据的分区，比如/home 和 /usr/local，这些目录经常有非常重要的数据。
7. 重新启动系统，看新系统是否能够正常工作。
8. 对于系统里的每个用户，根据旧的用户名和用户标识号，创建一个用户登录名。举例：

```
ls /home/maria/* -l    ( 查找用户 maria 的 home 目录得到她的用户标识，假设是 503 )
useradd maria -u 503  ( 创建一个用户标识为 503 的帐号 “maria” )
```

从每个用户的主目录里，删除一些会导致问题的设置文件比如：“./kde”，“.ICE”，等等。这也是一个用来删除垃圾文件的好机会，要删除这些文件的原因是：比如 KDE 的新版本用新的设置文件（在第一次使用时会自动产生）应该会运行得更好。举例：

```
cd /home/maria
rm -fr .kde
```

注意要确保在每个用户的主目录下的文件属于对应的用户。举个例子，我可以这样做：

```
cd /home/maria
chown -R maria *
chmod -R maria *
```

更多的信息参见第二部分，Linux 的资源，帮助和一些相关连接。

第 2 部分: Linux 的资源, 帮助和相关链接

Linux 新手管理员指南(中文版)

LINUX NEWBIE ADMINISTRATOR GUIDE (Simplified Chinese Version)

中文版本 1.0 由幻影快递 Linux 小组(SLT-Shadowfax Linux Team) 翻译整理

Linux Newbie Administrator 英文原版版权所有: Peter and Stan Klimas.

<http://sunsite.dk/linux-newbie>

中文版本版权所有: 幻影快递 Linux 小组 ShadowfaxLinux@hotmail.com

本章内容:

- 2.1 哪里有Linux的资料?
 - 2.2 有没有HELP命令?
 - 2.3 有没有术语字典?
 - 2.4 网上搜索
 - 2.5 新闻组
 - 2.6 有没有因特网上的Linux的链接?
 - 2.7 源代码--终极资源
-

2.1 哪里有 Linux 的资料?

这个指南不够用怎么办?

RedHat Linux的CD里包含了许多资料.它们一部分是HTML格式一部分是纯文本格式.你在安装之前可以在DOS或MS WINDOWS之下阅读所有的资料.例如,软拷贝的RedHat手册可以利用MS WINDOWS上的HTML浏览器(NETSCAPE或IE)阅读.只要打开D:\doc\rhmanual\manual\index.html就可以了(这里假设你的CD-ROM的驱动器号在WINDOWS下是D)。

你也可以看\doc\LDP 目录下的Linux文档计划手册.例如,你可以用你喜欢的浏览器打开doc\LDP\sag\sag.html 阅读Linux系统管理员指南。

你还可以看\doc\HOWTO 目录下的HOWTO文档,\doc\HOWTO\mini目录下的 迷你HOWTO 和\doc\FAQ 目录下的FAQ(常见问题)中的各种问题.以下的命令可以在DOS下阅读Linux-FAQ的纯文本文档。

```
D:
cd \doc\FAQ\txt
edit Linux-FAQ
```

在Linux里,你可以用以下命令读CD上的文档

```
lynx /mnt/cdrom/doc/rhmanual/manual/index.html
```

这里启动的是lynx,一个简单的文本模式的HTML浏览器。要注意的是CD 装载后才能在Linux下使用,以上的例子假设装载点是目录/mnt/cdrom/。也可用Linux下的NETSCAPE,STAROFFICE和其他的HTML浏览器读RedHat的手册或其他HTML格式的文档。在Linux下阅读CD上纯文本格式的文档,可以用象下面的命令。

```
cd /mnt/cdrom/doc/FAQ/txt/  
less Linux-FAQ
```

(less 命令让你卷页面阅读文本文件的内容)

不管你选择安装了哪些文档,Linux安装后文档总是位于/usr/doc/ 或 /usr/share/doc。如果你没有安装文档,现在就考虑安装吧,这样做可能是值得的。例如/usr/doc/LDP 目录下包括Linux文档计划手册。通过下面的命令可以浏览Linux系统管理员指南。

```
cd /usr/share/doc/LDP/sag  
lynx sag.html
```

同时还可以看/usr/share/doc/HOWTO目录下的HOWTO文档 usr/share/doc/HOWTO/mini 目录下的 迷你HOWTO 和有的时候文档可能在/usr/doc。

要找更多更新的文档,<http://www.ibiblio.org/mdw/index.html>。

2.2 有没有 HELP 命令 ?

多数的Linux命令可以带“--help”选项。例如,以下的命令可以给你简明的Linux cp (copy)帮助:

```
cp --help | less
```

更多的命令行的帮助要用所谓的参考手册页 man topic.例如:

```
man cp
```

会显示参考手册页中的cp (copy)命令。参考手册页是Linux下的标准“HELP”系统,含有很多的细节,非常技术化的信息,不过一般新手要花点力气才能看懂。

man命令用一个叫less的简单程序卷动文本页面。用箭头键卷页,按“q”键退出。事实上less能做的不只这些。运行less时按“h”键可以获得帮助,或通过以下命令了解更多的有关less的用法。

```
man less
```

还有一个叫info的命令 info topic.例如:

```
info cp
```

会显示“cp”(copy)命令的帮助文件。通常 info包含的信息和man相似,但都更新一点。不幸的是,info的浏览功能不是很直观,所以我man用的更多。还有一个 pinfo,info 的替代品(可能是因为它比info容易用的缘故吧)。

如果你忘记了想要用的命令的准确名字的话,可以试试apropos.例如,要想得到一组和“copy”有关的命令,我会在命令行运行:

```
apropos copy
```

whatis命令和apropos相似但只做关键字的比较,而apropos会对整个数据库搜索(关键字及其描述)。whatis 输出结果都比较短(可能也更相关)。

在一些使用菜单的程序中,例如在设置你系统服务用的 ntsysv(或setup,或 linuxconf),可以按F1键来了解某个服务具体是做什么的。

bash内置的命令列表可以通过在命令行敲入help获得。所有bash里的内置命令的帮助可以通过输入象以下的命令获得:

```
help cd
```

bash是标准的命令行“shell”,也就是在Linux里等同于DOS的命令行处理器“COMMAND.COM”。

KDE环境里有一个图形界面的“帮助浏览器”。它可以通过点击“Kpanel”(系统条,通常位于屏幕底部的)上相应的图标启动。这个浏览器可以用于浏览KDE专用的帮助和系统参考手册页。Gnome桌面有一个类似的帮助系统。

如果你想了解你CD上rpm 格式的包,你可以用图形界面的kpackage(在X终端上输入kpackage)浏览这些包,显示他们包含的信息,在你需要的时候安装它们(安装要由root进行)。旧版本和kpackage对应的是,glint(RH5.2),gnorpm(RH6.0),他们慢而且也不方便。

2.3 有没有术语字典?

这是一个相当大而全的:“新骇客字典”又叫“术语文件”

<http://www.catb.org/~esr/jargon/jargon.html>

它不仅是很好的资源,而且是很有趣的读物。推荐使用。

让有趣的更有趣,这里还有一个链接“女孩的(电脑)怪男孩指南”:

<http://www.lairgauche.com/geekguy.html>

如果你是另外一个性别的话你要看“男孩的(电脑)怪女孩指南”。

这是一个相当完整的计算机相关的词汇简称和缩写表:

http://www.geocities.com/ikind_babel/babel/babel.html

2.4 网上搜索

目前web上最好的搜索引擎是google，你会对它能找到的东西感到惊讶。Google速度非常快，因为它运行在Linux上，绝非儿戏。到<http://www.google.com/>上试试吧。做一个试验，在搜索栏里输入你的名字和你选的另外一个词。Google可以找到几乎所有网上和Linux（或其他的词）相关的东西。输入几个关键词就可以找到你要的Linux文档。

2.5 新闻组

这可能是个可怕的地方 - - 世界上最古怪的东西似乎都会在新闻组里出现。一般我不理会那些愚蠢或带挑衅的贴子和邮件，对于心怀恶意的，我会例外的通知系统管理员邮件的出处。在我在新闻组发帖后受到的广告，我一般都在阅读前删除，我知道我不是唯一这么做的人，所以如果你想让别人看你的邮件的话，请把标题栏写的清楚一些，特别是当你的邮件地址包括字符串“aol”的时候。

尽管有这么多不足，新闻组仍然是一个你查找所需信息的最有效的地方。

在上新闻组之前，我强烈推荐Google的新闻组存档 (<http://groups.google.com/>，过去叫DejaNews)。这是一个巨大的新闻组帖子存档，你还可以用很好的搜索工具搜索。这样，你通常无须翻阅无数的垃圾就可以找到你问题的答案，又无须承受发一个“已经是本周问了第十遍的问题”后的愤怒。你可能会对存档里的已有的信息感到惊讶。

这里有几个Linux新闻组，他们看上去比其他新闻组好很多（可能是因为他们是由Linux专家管理的比较好的缘故吧）。下面就是个短的列表。

news:comp.os.linux.announce

(有管理员的管理的，管理员对上去的帖子前都要检查。对所有的贴包括页脚在内都严格检查。)

news:comp.os.linux.setup

news:comp.os.linux.hardware

news:comp.os.linux.security

news:comp.os.linux.misc

(综合)

news:comp.os.linux.advocacy

(讨论Linux的优缺点，也可和其他的操作系统比较。这是很好的新闻组，如果你喜欢卷入无休止的争论的话)。

news:alt.linux.sux

(你可以在这里读写所有有关Linux的观点。)

news:comp.os.linux.networking

news:comp.os.linux.x

(X-windows)

news:comp.os.unix

(UNIX 总新闻组)

请注意新闻组礼节（网络礼节），如果违反你会被驱逐或被怒火所包围。要点如下：

- 不要发与新闻组话题无关的贴
- 不要在多个新闻组同时发帖 (一稿多投)。
- 用ASCII码, 不要加附件, 图象, HTML代码等等。
- 不要发广告 (特别是商业广告)
- 不要全用大写字母 - 这样谁都不理你。
- 使用简单描述性的标题 (用 “ HELP ” 不行) 并简单扼要的说明你的问题。这包括版本 (例如: RedHat -- 译者) 和版本号, 硬件型号 (USR 56K 不行, 要列出具体型号)

使用新闻组 (或叫 usenet), 你必须设定你的新闻服务器访问设置。最简单的是用 mozilla (edit -> preferences -> mail -> newsgroups) 输入新闻服务器 (你的互联网接入服务商, ISP, 会给你提供服务器的名字) 然后加入你相应的新闻组到你的本地信箱里。如果你不知道你新闻服务器的名字, 可以试试这个 “ news. 我的ISP名. 域名 ” 也可能是 “ 我的ISP名. 域名 ”

读新闻组我喜欢用 knode (在 X 终端里输入)。安装学习新闻组对我来说当然是值得的。

2.6 有没有因特网上的 Linux 的连接 ?

网上致力于 Linux 的网站肯定是数以千计。这里是一些我喜欢的 Linux 链接, 没有特别的排序。如果你想要其他的东西的话, 你应该能在这些页面里找到有用的链接。

http://sunsite.dk/linux-newbie/	这本书的主站。加入收藏夹。
http://www.linuxdoc.org/docs.html	Linux 文档计划 -- 包括许多 FAQ, HOWTO, 迷你 HOWTO, 指南的主页。不断更新中。
http://www.kalug.lug.net/linux-admin-FAQ/	Linux 管理员 FAQ (不是给新手的)。
http://members.aol.com/~swear/pedia/learning-linux.html	Gary 百科全书 -- 学习 Linux。加入收藏夹。
http://jgo.local.net/LinuxGuide/	Josh 主页。学习 Linux 的好资源。
http://www.control-escape.com/	这个网站看上去适合新手。
http://www.linuxninja.com/linux-admin/	轻松学会 Linux 管理 (LAME)。推荐。
http://metalab.unc.edu/mdw/index.html#guide	许多 Linux 文档。加入收藏夹。
http://www.frankenlinux.com	又一个对新手提供帮助的网站。
http://www.easyfeed.com/~jgo/LinuxGuide/	还是一个新手指南。
http://www.slashdot.org/	电脑怪才, 骇客, 高手们的论坛。
http://www.freshmeat.org/	关于 Linux 软件最新版本。
http://linuxtoday.com/	Linux 新闻 - 极好的每日一读。加入收藏夹。
http://www.llp.fu-berlin.de/	“ Linux 试验计划 ” 数据采集和其他科学研究方面的有趣资料。
http://www.linuxberg.com/	Linuxberg (Linux 大补膏? - 译者)。大门户网站。他们那里什么都有。我很喜欢他们 Linux 软件评分的做法, 我也只下载被评为 5 个企鹅的包。;-) 加入收藏夹。
http://counter.li.org/linuxcounter_eng.html	Linux 计数器。给自己注册做一个 Linux 份子。
http://www.ap.univie.ac.at/users/havlik/Album/Linux-Counter/	看看 Dennis Havlik 的给人深刻印象的 Linux 发展和地理分布图。
http://www.cl.cam.ac.uk/users/iwj10/linux-faq/index.html	Linux FAQ。
ftp://sunsite.unc.edu/pub/Linux/	Sunsite 的无数 Linux 软件存档。加入收藏夹。
http://stommel.tamu.edu/~baum/linuxlist/linuxlist/linuxlist.html	Linux 应用程序。
http://www.boutell.com/lsm/	Linux 应用程序。

http://www.linuxlinks.com/Software/	Linux应用程序。
http://directorysearch.mozilla.org/Computers/Operating_Systems/Linux/	巨大的新门户网站 (的比YAHOO还大) , 包括了许多非常适合Linux新手的链接。
http://dir.yahoo.com/.../Unix/Linux/	Yahoo的Linux入口。看上去是精心组织的, 他们拒绝加入这本指南!
http://www.debian.org/	Debian Linux 网站。
http://hardware.redhat.com/hcl/genpage2.cgi	Linux 硬件兼容表。
http://metalab.unc.edu/mdw/links.html	许多有用的Linux链接。
http://www.cse.unsw.edu.au/~conradp/linux/	一些很好的链接。
http://www.linuxstart.com/documentation/	更多的Linux文档链接。
http://www.linuxlinks.com/	更更多Linux链接。
http://www.gnu.org/	GNU主站。(GNU不是Unix。这是一个循环定义)
http://www.redhat.com/	Red Hat网站。他们太忙, 一般不要去打扰。
http://www.cs.Helsinki.FI/u/torvalds/	Linus Torvalds (他写了Linux的内核 - 译者) 的主页。

2.7 源代码--终极资源

Linux的终极参考资源是源代码。如果你已经安装了 (和标准发行版本一起发行, 如果你有足够的硬盘空间我们真诚推荐您安装) , 它在 `/usr/src/linux` (内核源代码) 和 `/usr/src/RPM/sources` (为了RPM包源代码的平衡)。新手怎么用源代码? 因为它包括了最细节的注释和文档。例如, 在这个指南的后面, 我们会讲如何通过 `/proc` 文件系统读/设置一些内核参数。你可以这样看到包括所有用到的参数的完整的文档:

```
less /usr/src/linux/Documentation/proc.txt
```

为了安装内核源代码, 我在安装时会选择适当的rpm包。为了安装我用的版本中其他的包, 我会把 "Source CD" 放入CD驱动器, 做类似以下操作 (以root身份, 用RedHat CD)

[从CD向硬盘安装gnumeric电子报表 (类似EXCEL的软件-译者) 的源程序]

```
su
cd /mnt/cdrom/SRPMS/
rpm -ivh gnume<Tab>
[unzip我安装的源代码]
cd /usr/src/RPM/SOUR<Tab>
tar -xvzf gnumer<Tab>
```

[查看gnumeric里统计函数的代码]

```
cd gnumeric/src/functions
less fn-stat.c
```

这是真实的关于一个特定的电子报表软件函数如何工作的终极参考, 我不是说着玩的。

第 3 部分: Linux 的基本操作

Linux 新手管理员指南(中文版)

LINUX NEWBIE ADMINISTRATOR GUIDE (Simplified Chinese Version)

中文版本 1.0 由幻影快递 Linux 小组(SLT-Shadowfax Linux Team) 翻译整理

Linux Newbie Administrator 英文原版版权所有: Peter and Stan Klimas.

<http://sunsite.dk/linux-newbie>

中文版本版权所有: 幻影快递 Linux 小组 ShadowfaxLinux@hotmail.com

本章内容:

- 3.1 基础知识
 - 3.1.1 文件名
 - 3.1.2 使用不同的目录
 - 3.1.3 执行一个程序
 - 3.1.4 改变路径 (PATH)
 - 3.1.5 关闭计算机
 - 3.1.6 处理挂起的程序
 - 3.1.7 命令选项
 - 3.2 用户, 密码, 文件访问权限和安全性
 - 3.2.1 主目录, 根目录和添加新用户
 - 3.2.2 密码安全性
 - 3.2.3 忘记了超级用户密码怎么办
 - 3.2.4 忘记了普通用户密码怎么办
 - 3.2.5 关闭或删除一个用户
 - 3.2.6 文件的所有权和访问权
 - 3.2.7 设置替代用户标识(SUID)
 - 3.3 用“&”, “at”, “batch”和“cron”进行作业调度
 - 3.3.1 怎样在后台执行命令
 - 3.3.2 怎样在某一特定时间执行命令(使用“at”和“batch”)
 - 3.3.3 怎样设置“cron”
 - 3.4 Shell
 - 3.4.1 什么是Shell? 需要一个不同的Shell吗?
 - 3.4.2 怎样定制自己的Shell
 - 3.4.3 设置文本终端的颜色
 - 3.4.4 在文本模式或 X模式控制台下, 如何打印符号
 - 3.4.5 如何编写简单的Shell脚本(Script)文件
 - 3.4.6 引号的含义
 - 3.4.7 输入输出重定向
 - 3.4.8 Shell的特殊字符(metacharacters)
 - 3.5 如何安装软件包(Package)及 rpm 软件包管理器
 - 3.5.1 如何安装从Internet下载的程序?
-

3.1 基础知识

3.1.1 文件名

Linux 系统区分英文字符的大小写。比如, myfile, Myfile 和 myFILE表示的是三个不同的文件。同样, 用户密码和登录名也需要区分大小写(这里沿用了Unix 和 C 语言的命名规则)。命名目录和命名文件的规则是相同的。除非有特别的原因, 否则用户创建的文件和目录名要使用小写字符。大多数的 Linux 命令也使用小写字符。

Linux 系统下的文件名长度最多可到256个字符。通常情况下, 文件名的字符包括: 字母、数字、"."(点)、"_"(下划线)和"-"(连字符)。Linux 允许在文件名中使用除上述符号之外的其它符号, 但并不建议用户这样做。有些转意字符(即该字符被系统借用, 表示某种特殊含义)在Linux 的命令解释器(shell)中有特殊的含义(shell 类似于 DOS下的命令处理器 COMMAND.COM)。这样的转意字符有: "?"(问号), "*" (星号), " "(空格), "\$"(货币符), "&", 扩号等等。在文件名中应尽量避免使用这些字符。文件名中可以有" "(空格), 但建议用户用"_"(下划线)来替代。"/" 既可代表目录树的根也可作为路径名中的分隔符(类似DOS下的"\"), 因此"/"不能出现在文件名中。

和DOS 一样, "." 和 ".." 分别用来表示"当前目录"和"父目录", 因此它们也不能作为文件名。

正如DOS 中一样, 我们不能将文件命名为 . 或者 .. (点或者双点)——它们分别代表当前目录和当前目录的父目录(上一级目录)。这跟DOS 中是完全一致的。

下面列出了一些转意字符及其意义:

* = 用于匹配任意长度大于等于零的字符序列。以 . (点)开头的文件名除外。

? = 用于匹配任意单个字符

[abc1] = 用于匹配枚举集合中任意单个字符。比例中, 集合由 'a', 'b', 'c', '1' 组成。

[a-z] = 用于匹配任意小写字母

[A-F] = 用于匹配任意大写字母

[0-9] = 用于匹配任意个位数字

[a-zA-Z0-9] = 用于匹配任意字母(大小写)或者数字。

\ (反斜线)也是一个特殊的字符。它屏蔽后继特殊字母的特殊含义(转意), 使该字符仅取其符号所代表的字面意义。

例: 下面的命令会列出当前目录中除了文件名以 . 开头的文件之外的任何文件。

```
ls *
```

与此等价的命令是 ls 或者 dir (不需要后跟 "*")。文件名以 . 开头的文件不被列出是因为 "." 是文件名的首字符, 它并不与 "*" 匹配。我们可以把文件名以 . 开头的文件等价地理解成DOS 中隐藏文件。使用 ls -a (带 all 选项的 ls 命令)或者 ls .* 命令来查看这些"点"文件。"点"文件在用户的 home 目录中很常见。它们通常是用户级的配置文件。

以下命令可列出当前目录中文件名含 . 的任何文件(文件名以点开头的除外)。

```
ls *.*
```

以下命令可列出当前目录中文件名含两个点的任何文件（文件名以点开头的除外）。

```
ls *.*.*
```

请注意 Linux 中没有与 DOS 中文件扩展名相应的概念，但是你可以使用扩展名。例如，一个文件可以命名为 `my_text.txt.zip`。另有一些 DOS 中的命名特色在 Linux 中是完全找不到的（比方说 `Micro~1.doc`）。

以下的命令会找出扩展名为“htm”后再跟任意一个字符的文件：

```
locate *.htm?
```

以下命令可列出当前目录中所有文件名以 a,b 或者任意大写字母开头的文件：

```
ls [abA-Z]*
```

以下命令可列出当前目录中所有文件名以 a 开头,以 n 结尾的文件：

```
ls a*n
```

命令行自动完成是一个有用的功能——使用 [Tab] 键可以极大的节约时间，使处理复杂的长文件名变得轻松。例如，使用自动完成的功能，可以极大的减化输入以下这个长文件名的工作。

```
dir Eurosong\ 2000\ Olson\ Brothers\ -\ Fly\ on\ the\ wings\ of\ love\ \ (denmark\).mp3
```

我只要输入

```
dir Eu<Tab>
```

如果没有其他文件的文件名以 Eu 开头，输入 <Tab> 之后，系统会自动输入文件名的其余部分。如果还有其他文件的文件名也以 Eu 开头，我们可以查看一下可能的选择（系统会列出），进一步输入文件名中的下一两个字符，从而使文件的选择唯一化，系统的自动写成功能就可以启动完成整个长文件名的输入了。文件名中的反斜线作用在于提示系统其后继的字符“空格”仅取其字面意义，即，文件名中包含的确实是空格。

文件名中的有“怪”字符怎么办？大部分问题可以通过自动完成功能解决。另外，处理文件名中含有转意字符的文件，我们可以使用成对的 ‘ ’（单引号）。由于这些特殊的转意字符被单引号引用，命令解释器不会试图去解释其特殊意义。例如，给一个名为 `my file*` 的文件更名（文件名中含空格和*），我们可以使用：

```
mv 'my file*' filename_without_weird_characters.txt
```

请注意我使用了一对 ‘ ’ 号（单引号）。使用成对的 “ ”（双引号）引用通常没有使用 ‘ ’ 有效。如果你使用 “ ” 号（双引号），部份转意字符可能会被命令解释器加以解释（这就意味着它们可能被转意，而不是保持字面意义）。

跟从 UNIX 的传统，在 Linux 上，创建文件时，我们可以在文件名中可以使用几乎任何字符，包括不可打印字符（控制字符）。但是这情况非常少见。如果你在遇到文件名使用了这些字符，你可能

会感觉很异样。我通常会在适当的位置使用一个转意字符*对这种文件进行重命名。我会先使用 ls 命令确认我要重命名的文件的文件名，然后对这个文件进行更名（使用 mv 命令）：

```
ls -l myfile*y.html
mv myfile*y.html myfile.html
```

(此例中我假定非标准字符位于字母 e 和 y 之间)

这里举一个关于在文件名中使用非常规字符的极端的例子：创建一个文件名以短横线 - 开头的文件，再等它删除。用常规的删除命令似乎没有办法（因为短横线通常是用来引导命令选项的）。例如：

```
dir > -junk
```

以上命令将创建一个以-junk 为名的文件（跟 DOS 类似，符号 > 将 dir 命令的输出重定向到一个名为“-junk”的文件。常规的删除文件命令无法把这个文件删除。我们需要使用：

```
rm ./-junk
```

文件名前的 ./ 表示当前目录。在此其目的是避免文件名把的首字符 - 被命令解释器当作 rm 命令的选项引导。事实上没有必要在文件名中使用这些特殊字符。遵从传统的命名规则，不会过多地使事情复杂化。

除了使用自动完成，单引号，双引号之外，我们也可以利用 \（反斜线）对文件名中包含了非常规字符的文件进行操作。反斜线的作用在于屏蔽其后继字符的特殊意义。例如，我可以使用如下命令创建一个以 *?[为文件名的文件：

```
touch \*\?\[
```

(touch 命令创建一个空文件。如果同名文件已存在，则改变该文件最后更新的日期/时间)。

3.1.2 使用不同的目录

Linux的文件系统目录树庞大而复杂。如果你非常熟悉它的话，会极大地提高你应用Linux的技巧。简单地说，典型的Linux包含五大文件系统目录。根据你自己系统的需要和大小，这些文件系统目录能安装在单个或多个物理硬盘的分区中。（如果需要的话，单个文件系统目录也可以安装在不同的设备中。）

“ / ” 根目录

包含基本的操作系统和维护工具。这个目录的文件可以启动系统，并且需要的话，可以完成紧急维护和修复的功能。

“ /usr ” 目录

包含所有的命令、程序库、文档和其它文件。这些文件在正常操作中不会被改变的。这个目录也包含你的Linux发行版本的主要的应用程序，譬如，Netscape。

“ /var ” 目录

包含在正常操作中被改变的文件：假脱机文件、记录文件、加锁文件、临时文件和页格式化文件等。

“ /home ” 目录

包含用户的文件：参数设置文件、个性化文件、文档、数据、EMAIL、缓存数据等。这个目录在系统省级时应该保留。

“ /proc ” 目录

整个包含虚幻的文件。它们实际上并不存在磁盘上，也不占用任何空间。（用 `ls -l` 可以显示它们的大小）当查看这些文件时，实际上是在访问存在内存中的信息，这些信息用于访问系统。

“ / ” 根目录部分有以下子目录：

`/bin` 系统启动时需要的执行文件（二进制），这些文件可以被普通用户使用。

`/sbin` 系统执行文件（二进制），这些文件不打算被普通用户使用。（普通用户仍然可以使用它们，但要指定目录。）

`/etc` 操作系统的配置文件目录。

`/root` 系统管理员（也叫超级用户或根用户）的Home目录。

`/dev` 设备文件目录。LINUX下设备被当成文件，这样一来硬件被抽象化，便于读写、网络共享以及需要临时装载到文件系统中。正常情况下，设备会有一个独立的子目录。这些设备的内容会出现在独立的子目录下。LINUX没有所谓的驱动符。

`/lib` 根文件系统目录下程序和核心模块的共享库。

`/boot` 用于自举加载程序（LILO或GRUB）的文件。当计算机启动时（如果有多个操作系统，有可能允许你选择启动哪一个操作系统），这些文件首先被装载。这个目录也会包含LINUX核（压缩文件 `vmlinuz`），但LINUX核也可以存在别处，只要配置LILO并且LILO知道LINUX核在哪儿。

`/opt` 可选的应用程序，譬如，REDHAT 5.2下的KDE（REDHAT 6.0下，KDE放在其它的X-WINDOWS应用程序中，主执行程序在`/usr/bin`目录下）

`/tmp` 临时文件。该目录会被自动清理干净。

`/lost+found` 在文件系统修复时恢复的文件。

“ /usr ” 目录下比较重要的部分有：

`/usr/X11R6` X-WINDOWS系统（version 11, release 6）

`/usr/X11` 同`/usr/X11R6`（`/usr/X11R6`的符号连接）

`/usr/X11R6/bin` 大量的小X-WINDOWS应用程序（也可能是一些在其它子目录下大执行文件的符号连接）。

/usr/doc LINUX的文档资料 (在更新的系统中, 这个目录移到/usr/share/doc)。

/usr/share 独立与你计算机结构的数据, 譬如, 字典中的词。

/usr/bin和/usr/sbin 类似与 “ / ” 根目录下对应的目录 (/bin和/sbin), 但不用于基本的启动 (譬如, 在紧急维护中)。大多数命令在这个目录下。

/usr/local 本地管理员安装的应用程序 (也可能每个应用程序有单独的子目录)。在 “ main ” 安装后, 这个目录可能是空的。这个目录下的内容在重装或升级操作系统后应该存在。

/usr/local/bin 可能是用户安装的小的应用程序, 和一些在/usr/local目录下大应用程序的符号连接。

需要理解的是: 所有目录出现在单一的文件目录树下, 即使这些目录在不同的分区, 物理驱动器 (包括软驱), 或者甚至分布在网络上。所以, 在LINUX下没有类似在DOS下的驱动符。在DOS或WINDOWS下的驱动符, 在LINUX下以一个装载位置下的子目录的形式出现。

大多数的LINUX发行版本, 目录系统很完善并且标准 (Linux Standard Base发布了微小的区别)。LINUX目录系统和典型的商业UNIX目录系统完全相似。

总结来说:

- 用户应该将文件存在/home/user_login_name目录下 (及其子目录下)。
- 本地管理员大多数情况下将额外的软件安装在/usr/local目录下并符号连接在/usr/local/bin下的主执行程序。
- 系统的所有设置在/etc目录下。
- 不要修改根目录 (“ / ”) 或/usr目录下的任何内容, 除非真的清楚要做什么。这些目录最好和LINUX发布时保持一致。
- 大多数工具和应用程序安装在目录: /bin, /usr/sbin, /sbin, /usr/x11/bin, /usr/local/bin。
- 所有的文件在单一的目录树下。没有所谓的 “ 驱动符 ”。

更多关于/proc目录 (仅为真正好奇的读者)

/proc “ 虚拟 ” 文件系统目录是一个实时的居留内存的文件系统, 用于监视操作系统内核和在计算机上运行的进程的状态。/proc目录是完全虚拟的, 即它不写入任何特殊的磁盘或其它永久的媒体, 它仅居留于计算机的内存, 并且它是不断地更新以反映系统的变化。/proc目录的大小总是0, 最后的日期总是当前日期。手工改变/proc目录下文件的内容可能会改变系统设置。许多LINUX使用工具利用/proc目录作为信息资源, 如, dmesg, ps, top等。

/proc目录的内容:

/proc/cpuinfo 关于处理器的信息, 如类型、厂家、型号和性能等。

/proc/devices 当前运行内核所配置的所有设备清单。

/proc/dma 当前正在使用的DMA通道。

`/proc/filesystems` 当前运行内核所配置的文件系统。

`/proc/interrupts` 正在使用的中断，和曾经有多少个中断。

`/proc/ioports` 当前正在使用的 I/O 端口。

举例，使用下面的命令能读出系统的 CPU 信息。

```
cat /proc/cpuinfo
```

3.1.3 执行一个程序

在命令行输入可执行程序的文件名但是程序不能运行？有三种可能性：

第一种可能性：没有输入正确的文件名。注意：Linux 文件名是区分大小写的！举个例子，输入“Pico”或者“PIC0”并不能启动 pico 编辑器程序。

第二种可能性：程序不在缺省路径 PATH 下？在 Linux 和其他的 Unix 下，可执行程序必须要在你的 PATH 下面你才能够运行。如果程序所在的目录不在缺省路径下，你可以进入到程序所在的目录：

```
cd the_program_directory      (the_program_directory 是程序所在的目录名)
./program_name                (program_name 是程序的文件名)
```

或者你可以输入程序完整的路径名如下：

```
/the_program_directory/./program_name
```

注意，必须在程序名前面加上点“.”和斜杠“/”才能够正确运行。这是 Linux 的另外一个安全特性，可以使“特洛伊木马程序”更难于入侵。点“.”指的是当前目录，斜杠“/”是文件和目录之间的间隔符（DOS 用的是反斜杠“\”）。

可以用一下命令来检查一下缺省路径：

```
echo $PATH
```

如果忘记了可执行程序的文件放在哪一个目录下面了，可以用以下命令查找：

```
find / -name "netscape"
```

以上命令从根目录“/”开始查找文件名叫“netscape”的程序。用这个命令可能速度更快一些：

```
locate netscape
```

(locate 命令运行速度更快是因为它依赖于系统预先建立的文件数据库。这个数据库由一个在背景运行的程序 cron 来建立。因为 cron 一般安排在夜间运行，所有如果你经常在夜晚关机，就不要依赖 locate 可以找到你要的文件，也不要巴望 locate 可以找到刚刚才安装的软件名称)

请注意，PATH 对于超级用户 root 和普通用户一般来说是不相同的。超级用户 root 的 PATH 包括 /sbin 和 /usr/sbin，而普通用户一般不包括。所以，普通用户不能执行在 /sbin 下面的文件，除非

他们指定程序的全路径名 (目录加文件名) 。还有, 如果你是使用 `su` 命令从普通用户变成超级用户 `root` 的, 因为继承了普通用户的缺省路径 `PATH`, 要执行 `/sbin` 目录下的文件, 还是要指定程序的全路径名。

如果你知道程序的名字但是不知道程序的全路径名, 可以用以下命令:

```
which netscape
```

就可以找到所有叫做 `netscape` 的文件的全路径名

第三种可能: 程序可能是不可执行的。如果是这个原因, 把它改成可以执行的 (必须是超级用户 `root` 或者文件的所有者)

```
chmod a+x my_file
```

这个命令把 “ `my_file` ” 变成是所有人都可以执行的。检查是否可行:

```
ls -l my_file
```

请注意, 在 Linux 和 Unix 下, 修改文件的扩展名 (比如 `.exe` 或者 `.bat`) 并不能使文件变成可以运行。文件能够被执行需要 “ 可执行访问模式 ”, 而不是象 DOS 下的 “ 文件属性 ”。

3.1.4 改变路径 (PATH)

一般来说, 你不需要去修改缺省路径 `PATH`。

`PATH` 是当你要求执行程序时, 文件搜索的目录列表。你可以用以下命令检查你的 `PATH`

```
echo $PATH
```

比如说, 在我的系统里, 显示用户 `yogin` 的 `PATH` 为:

```
/opt/kde/bin:/usr/local/bin:/bin:/usr/bin:/usr/X11R6/bin:/home/yogin/bin
```

“ : ” 是间隔符, 以上的 `PATH` 代表以下的目录表:

```
/opt/kde/bin
/usr/local/bin
/bin
/usr/bin
/usr/X11R6/bin
/home/yogin/bin
```

以下是在我系统上的超级用户 `root` 运行 “ `echo $PATH` ” 的结果:

```
/opt/kde/bin:/sbin:/bin:/usr/sbin:/usr/bin:/usr/X11R6/bin:/root/bin
```

你可以修改文件/etc/profile里以“PATH=”打头的行，这样所有用户的缺省路径都将被修改，我一般用pico编辑器：

```
pico -w /etc/profile
```

(可选项 -w 用于关闭文件的自动换行)

你必须重新登录，才能验证修改的结果。仅修改个别用户的缺省路径，可以编辑文件：

```
/home/user_login_name/.bash_profile
```

注意文件名前面的“.”。如果文件名的第一个字符是“.”，通常这个文件是不可见的，必须用ls -a的命令显示。

如果你真的想要当前目录在你的缺省路径PATH里，在你的PATH里加上“.”。

/etc/.bash_profile 文件看起来可能是这样的

```
PATH="$PATH:$HOME/bin:."
export PATH
```

这个命令从环境变量里得到 PATH (对于所有用户，PATH 在/etc/profile 文件里)，然后加上用户的\$HOME 目录(比如，用户 susan 的\$HOME 目录一般就是/home/susan 目录)，再加上“.”指的是当前目录。修改完 PATH 或者其他环境变量后，一般要使用 export 命令。这样才能使所有环境变量在修改之后，在设置脚本之外的环境都是可视的。

3.1.5 关闭计算机

在关闭计算机之前最好把所有的运行程序都关闭。从 GUI 的主菜单 (GNOME 或者 KDE) 选择“Logout”。接着，在 logout 的窗口里，选择“Shutdown”。

你也可以采用另外一种办法，在字符终端，按住<CTRL><ALT>(三个键同时按住)，等待系统关机完成后，在机器重新启动的时候关掉计算机电源。如果你在 X-Windows 里，先按<CTRL><ALT><F1>(三个键同时按住)切换到字符终端，然后再按<CTRL><ALT>。

从来不要在未正常关机的情况下关闭计算机的电源，否则下一次重新启动可能会有硬盘出错的信息。一般来说，由于不正常关机导致的硬盘错误会被系统自动修复，但是在某些场合下也可能导致严重的问题，你可能要手工修复损坏的文件或者重新安装系统！

如果你希望按住<CTRL><ALT>只是关闭计算机而不是重新启动，你可以编辑文件/etc/inittab。这个文件看起来如下：

```
# Trap CTRL-ALT-DELETE
ca:ctrlaltdel:/sbin/shutdown -t3 -r now
```

以 root 用户登陆，把文件修改如下：

```
# Trap CTRL-ALT-DELETE
ca:ctrlaltdel:/sbin/shutdown -t3 -h now
```

“ # ”打头的行只是注释行，只是给人看的，对计算机不起影响
“ t3 ”告诉 shutdown 程序在关闭所有的进程之前等待“三秒钟”
“ r ”选项指的是 reboot（重新启动）， “ h ”选项指的是 halt（关机）

如果你是 root 用户，也可以直接使用 shutdown 命令关机。这个命令可以在用于关闭本地和远程的计算机。但是最常用的是当本地计算机的键盘不起作用（由于系统挂住等原因），无法按 <CTRL><ALT><DELETE>时从远程登陆后使用 shutdown 命令关机。
telnet name_of_machine_with_no_operatable_keyboard

[以一般用户登陆到远程计算机 name_of_machine_with_no_operatable_keyboard]

```
su root
```

[登陆到 root 用户]

```
/sbin/shutdown -rn now
```

[关闭计算机]

这个命令跳过标准的关机程序，所以关机的速度很快，当系统变得很不稳定的时候特别有用。（选项 -n 在重新启动之前强行中止所有的进程）

注意，出于安全的考虑，你不能在远端或者通过因特网使用 root 用户直接登陆。你必须先 login 到一般用户，然后执行 su root 命令，输入正确的秘密后进入到超级用户 root。

你也可以使用 shutdown 把关机的时间安排得往后一点

```
/sbin/shutdown -r 23:59
```

将在午夜 0 点之前的一分钟重新启动计算机。 或者

```
/sbin/shutdown -r +1
```

将在你命令输入后的一分钟重新启动计算机

如果 shutdown 命令对你来说太长太难记了，也可以使用以下命令：

```
reboot      ( 重新启动 )  
halt        ( 关机 )
```

另外一个比较偏门的办法是把计算机切换到 runlevel 0（关机状态）或者 runlevel 6（重新启动状态）。命令如下：

```
init 0      ( 关机 )  
init 6      ( 重新启动 )
```

关于不同的 runlevel 的描述可以在文件 /etc/inittab 里找到。

3.1.6 处理挂起的程序

有错误的程序在 Linux 下照样会引起系统挂起没有响应。但是一个崩溃的程序一般不会影响操作系统本身，所有你不需要经常因为某个程序崩溃而重新启动系统。很多 Linux 的服务器运行超过一年也不需要重新启动。按我们的经验，一个不太稳定的操作系统一般是由硬件或者不正确的系统配置引起的。比如，我们多次碰到奔腾处理器的过热问题(CPU 风扇转的不如所要求的速度快或者干脆停了，CPU 芯片与散热片之间由于有脏东西导致接触不太紧密)，质量不好的内存条，速度不匹配的内存条，错误的 BIOS 设置（你最好把一些先进功能的设置去掉，让 Linux 来负责这些设置）。“signal 11”是一个典型的与硬件有关的出错信息，特别是当你执行一些要求复杂运算的任务：Linux 安装，内核的编译等等。如果你的奔腾芯片有过热的现象（特别是早期的一些奔腾芯片），这里有一些小诀窍让它“冷却”：清洗芯片上的散热片，更换 CPU 风扇，把计算机用机壳装起来（一些计算机玩家喜欢“裸机”），在计算机内多安装一个风扇，不要“超频”，减少一些不必要的任务，等等，还有，使用空白屏幕替代很复杂的屏幕保护。

非真正的系统挂机

有些程序运行时看起来好像是系统挂住了，其实是在等待用户的输入。比较经常碰到的是一些程序的命令行需要输入对应的文件名作为选项，但是用户只输入程序名却没有给需要的文件名，程序就假设用户即将在标准输入终端上输入文件名，于是处于等待状态。比如命令：

```
cat
```

输入回车后，看起来好像没有响应，其实程序在等待你的键盘输入。按住<CTRL>D(代表“文件结束”)就可以看到真正的结果了。

再举另外一个例子：我在用户新闻组上看到很多关于使用 tar 命令解压一个下载的文件时，引起系统挂死的问题。比如：

```
tar -zxv my_tar_file (命令错误!)
```

其实这也是在等待用户输入。因为命令没有带选项“-f 文件名”，所以文件 my_tar_file 并不被认为是一个文件名。正确的命令应该是：

```
tar -zxvf my_tar_file
```

要注意的是文件名必须紧跟在选项-f 后面，否则照样出错，比如：

```
tar -zxfv my_tar_file (命令错误!)
```

出错的程序都可以被中止

一个在前台运行的文本程序可以在任何时候使用<CTRL>C 中止，但是这对比较大的应用程序不起作用，因为这些应用程序为避免用户的误操作，通常把<CTRL>C 的功能屏蔽。当程序被挂起时，你可以使用<CTRL>Z 将程序送到后台运行（不是百分百可行），或者切换到不同的终端（使用<CTRL><ALT><F2>），然后，使用相同的用户名登录到系统。一旦你重新回到系统里，找到你需要中止的程序：

```
ps
```

这个命令代表“列出当前状态”(print status),可以显示当前系统中由该用户运行的进程列表。在 ps 的输出中,找到挂起的程序的 PID 号,然后执行命令:

```
kill 123
```

该命令将中止 PID 号码为 123 的进程。

作为一般用户,我只能中止那些由我运行的进程。而作为超级用户可以中止任何进程。想要看系统中所有正在运行的进程的完整列表,用命令:

```
ps -axu | more
```

这命令将列出正在运行的所有进程(选项 a),包括那些没有控制终端的进程(选项 x),包括每个进程所属的用户名(选项 u)。因为显示的内容可能要超过一个屏幕,所以用管道命令“|”加上“more”让换屏幕之间有个停顿。

kill 命令另有一个快捷的方法 killall,举例:

```
killall netscape
```

这个命令将中止名称是 netscape 的所有进程

```
killall pppd
```

这个命令将中止 ppp daemon 进程,取消所有的拨入连接

X-windows 的程序没有控制终端,可以很容易用 xkill 命令中止(在 X 终端里输入命令):

```
xkill
```

你可以看到光标变成一个“宣判死刑”的标志,把光标移到你要关闭的窗口,点击鼠标左键,窗口立刻关闭,该程序被中止。

对于上述命令,另外一个快捷的办法是按住<CTRL><ALT><ESC>,等待光标变成一个“宣判死刑”的标志指到对应的窗口。点击鼠标后,窗口就会消失,程序退出。

如果 X-Windows 系统崩溃实在无法恢复,强行中止 X-Server 最快的办法是按<CTRL><ALT><BACKSPACE>。然后,运行 ps -aux 找到任何可能还在运行的 X-Windows 程序,逐一使用 kill 命令中止这些进程。如果你不这样做,其他任何一个有问题的 X-Windows 程序可能又会引起整个 X-Windows 系统崩溃。如果你有在后台运行的程序,操作系统会要求你退出,接着发送信息“ There are stopped jobs”。你只要重复运行命令 logout (或者 exit),后台运行的程序将会被自动退出,然后你也会退出系统。

核心文件 (core files)

当一个程序崩溃时,它经常导出一个“核心文件”到用户的主目录下。一个“核心文件”通常是一个内存映像文件,这个文件带有一些给程序错误解析时要用到的信息。如果你只是普通用户不想解析任何错误程序,可以把“核心文件”删除。

```
rm core
```

或者放在那里不管，下一次有程序出错的时候，该文件会被自动覆盖。你也可以使用以下命令关闭“核心文件导出”的功能：

```
ulimit -c 0
```

看看是否生效使用命令：

```
ulimit -a
```

(该命令显示“用户限制”，选项“-a”代表“所有”)。如果想要关闭所有用户的“核心文件导出”功能，编辑文件/etc/profile (要求超级用户 root)，调整你要的设置。

如果你想知道“核心文件”有什么用处，可以在“核心文件”所在的目录使用命令：

```
gdb -c core
```

这个命令将启动 GNU 的程序调试工具，显示产生“核心文件”的程序名，程序中止时的信号等信息。输入 quit 退出程序调试工具。如果想知道更多的信号，用命令：

```
cat /usr/include/bits/signum.h | more
```

3.1.7 命令选项

大部分命令带很多选项，大部分选项在字符前面带有“-”，举个例子：

```
dir -l
```

显示当前目录下的文件（以长文件名格式显示，不带选项为缺省时按短文件名格式显示），

```
dir -l -a
```

或者

```
dir -la
```

以长文件名格式(选项 - l)显示所有文件包括隐含文件(选项-a)。

一般情况下，选项只带一个字符。这是为了遵循旧的 Unix 的传统。当然也有一些新的选项不止一个字符：

```
dir --help
```

这里，如果选项超过一个字符，选项前面就必须使用两个“-”作为引导。以上命令显示关于命令 dir 的一些简单帮助信息。如果显示的内容超过一个屏幕，可以使用管道命令：

```
dir -help | more
```

3.2 用户，密码，文件访问权限和安全性

3.2.1 主目录，根目录和添加新用户

对于一般用户来说，主目录(home directory)是硬盘上唯一可以原来写东西的地方。一般的路径名是/home/login_user_name。

主目录用于存储各种用户文件：设置文件，程序配置文件，用户文档，数据，netscape 的缓存，邮件等等。作为一个普通用户，你可以在主目录下建立新的目录安排你自己的目录结构。其他用户无法阅读你的文件或者写数据到你的主目录，除非你给他们适当的权限。

除了自己的主目录以内的文件，一般用户也可以看到，阅读和执行很多系统里的文件，但是一般来说他们不能修改和删除这些文件。

root 用户（也叫“超级用户”）是一个特殊的系统管理帐号，可以修改系统里的任何文件。经常使用 root 用户作为缺省的登录用户不是什么好习惯。你的误操作将有可能导致严重后果。建议设置一个一般用户给自己作为日常使用的帐号，只在必要使用 root 用户的权限时才登陆到 root 用户模式下。一般来说，root 用户是 Linux 初始化安装后的唯一一个用户。

使用 root 用户建立一个普通用户：

```
adduser joe
passwd joe
[输入用户秘密]
[再次输入用户秘密确认]
```

在上述的例子中，要求超级用户 root。以上命令在系统里产生一个 joe 的用户，接着设置该用户的密码。现在，可以告诉 joe 他的用户名和初始密码，然后他可以登录到系统，然后修改他的密码。还要注意的，用户名和密码都是大小写敏感。

超级用户可以修改任何人的密码，尽管他/她无法阅读该用户的密码。用户密码采用单向加密算法，加密后仅仅在系统上储存加密后的密码，对于旧的系统一般存在/etc/passwd 文件里，新的系统一般存在/etc/shadow 文件里，未经加密的密码文件从来不被储存在系统里。当用户登录的时候，系统把用户的输入密码使用相同的加密算法得到的结果再和密码文件（/etc/passwd 或者 /etc/shadow）里的结果相比较。

超级用户和一般用户的分开使得 Linux 系统更加安全。甚至让病毒在 Linux 系统下很难有所作为（因为用户运行的程序只能把数据写到他/她自己的有读写权限的目录里，不会感染整个操作系统的核心部分）。

一般建议用户在第一次登录到系统的时候立刻修改自己的密码：

```
passwd
Changing password for joe
(current) UNIX password: [输入旧的密码]
New password: [输入新的密码]
```

```
Retype new password:      [重复输入新的密码]
passwd: all authentication tokens updated successfully.
```

实际上,当你输入密码时,出于安全原因,键盘输入并不会显示在屏幕上。如果你是第一次修改密码,看不到键盘输入的结果可能会有点不适应。

在 Linux 里,同一个密码可以用来:

- 登录到文本终端
- 登录到图形用户界面 (GNOME 或者 KDE)
- 取消锁定的文本终端
- 取消密码保护的图形用户界面下的屏幕保护 (GNOME 或者 KDE)

3.2.2 密码安全性

一个“脆弱”(指容易被破解)的密码通常是安全问题的根源。即使在家里,完全有可能当你在浏览因特网的同时,黑客已经攻破你的计算机,正在修改和删除你计算机上的文件,或者通过你的计算机在本地警察局的网络里做一些令人头痛的事情。所以,即使在家里也要保证用户密码的安全。一旦有人登录到你的计算机,即使是普通用户,他也有可能找到其他方法获得超级用户的密码。其实就看你管理计算机系统的能力和黑客破解能力的较量了。

以下是一些不好的密码:

- 根本没有密码
- 密码就是“password”
- 密码和登录用户名相同
- 你的名字,女儿的名字,儿子的名字,妻子的名字,丈夫的名字……或者任何一个亲人的名字。人的名字其实非常的有限 只要查一下一些类似“如何给婴儿取名”之类的书就可以了,不要以为你是印度来的就没有人知道你的名字。
- 你的姓或者其他人的姓。姓的数量比名字更加有限!
- 你家小狗的昵称,老婆的昵称等等。昵称的数量比姓更少!
- 你喜欢的体育俱乐部的名称,节日名称,牙膏的名称等等。避免使用非常出名的足球队的名称,也不要任何摇滚乐队的名字
- 你的生日,社会保险号码,等等。
- 公司,部门,小组的名称
- 密码写在你的记事本上或者计算机壳子上
- 你在因特网联机商店使用的密码,电子邮件邮箱的密码等
- 任何在字典里可以找到的单词。英文字典其实没有包含你想象中那么多的单词(10万个单词?10万个单词的文件小于1MB!)。一个水平很一般的黑客很容易就加密所有的字典词汇然后逐一和你的密码比较。一个不可否认的事实,因特网上的确已经存在用来制造“字典攻击”的现成工具。不信?试着找一个工具来破解你的密码看看有多容易!
- 任何其他词汇,姓,宠物或者成语,不管是哪一种语言。对于一个有经验的黑客来说,如果他已经覆盖了一种语言,稍微加工一下就可以很容易覆盖其他常用的语言。有多少常用的语言?40个?黑客只需要把预先做好的几个文件加到破解文件清单里就行了。
- 任何以上的密码在前面或者后面加上一两个数字或者字符

一个好的密码最少要6个字符,有些专家甚至建议最少要10个字符,包含字符(最好大小写混合),数字和特殊字符(比如?,*,\$,%,#),还要定期做修改(建议8到16周之间)。

不巧的是，越不好破解的密码往往越难以记住。为了解决这个问题，我花了十分钟发明了我自己的“密码公式”。比如说，我总是以“@”字符作为密码的开头和结尾，然后使用两个单词并且用“!”把它们连接起来，每一个单词的最后一个字符大写，比如“@whitE!housE@”。看起来象一个不太好记的密码，但是只要我记得自己的“密码规则”，其实可以很容易记住这个密码。当然，如果你是一个记忆的超级天才，你可以使用 `mkpasswd` 并且试着记住它 ☺。

系统管理员可以使用 Linux 自带的工具来设置密码的规则（密码最小长度，是否要求特殊字符，密码过期的期限等），以 root 用户运行：

`linuxconf`

在菜单“user account”-“policies”-“password & account policies”设置。在以下场合里，一般的用户不能设置一个密码：

- 当密码长度太短
- 密码是一个字典里的单词
- 密码中不包含数字或者特殊字符

但是对于 root 用户，可以设置任何密码而只接收系统的警告信息。

还要保证任何包含你的密码的文件（比如 `/root/.kde/share/config/kpprc`）有正确，安全的权限设置，这样才能保证你的密码不会被其他人看到。举个例子，你可能要运行命令：

```
chmod 600 kpprc
```

如果你使用电话线上网，每周只用几个小时，你的密码太脆弱可能不会有什么大问题。但是如果你使用 cable modem 宽带上网，或者如果大部分时间你都呆在因特网上，你最好重新考虑一下你的系统安全。

有些计算机半文盲使用一些脆弱得让人惊讶的密码，根据 CNN 的相关报道，“大概有 50% 的计算机用户基于家庭成员的名字，配偶的名字或者宠物的名字作为密码。大概 30% 的计算机用户基于流行音乐歌手的名字或者体育明星的名字作为密码”。

http://www.cnn.com/2002/TECH/ptech/03/13/dangerous_passwords/index.html

特别要请注意的是：在以上“基础密码”后面加上一两个数字并不会使密码安全提高多少。

3.2.3 忘记了超级用户密码怎么办

尽管我从来不会忘记自己的密码，但是我还是会仔细研究一下这个专题，以防将来有一天突然看到我的母亲在阅读我的 ICQ 聊天记录时，可能会派上用场。

第一种办法：

最容易解决“忘记密码问题”的方法是将 Linux 重新启动到单用户模式，可以在“lilo”的命令提示符下输入：

```
linux single
```

这种办法将使你变成 root 超级用户而不需要输入任何密码。现在作为 root 用户，你可以使用以下命令重新设置密码（不需要输入旧的密码）

```
passwd
```

你可能觉得这实在是太不安全了！那是因为，如果有人可以物理上“访问”你的计算机硬件，没有计算机系统是安全的。但是，我并不喜欢在我的计算机上有这个“linux single”的漏洞存在，所以我在文件/etc/lilo.conf 里（在“image=”段落的尾部）加上以下内容就可以把这个漏洞拔除：

```
password= "my_password"
restricted
```

这个设置使得 Linux 启动时，在 lilo 的命令提示符下，当用户输入 linux 带任何参数时必须输入正确的密码。如果用户不是使用命令行的启动模式而使用正常的启动模式，没有密码也能够启动系统。为了让修改生效，必须重新运行 lilo 命令。因为我设置的密码在 lilo.conf 文件里没有加密，我还必须把文件/etc/lilo.conf 改成只有 root 用户可以读写：

```
chmod 600 /etc/lilo.conf
```

第二种办法

另外一个解决“超级用户密码丢失”的办法是使用 Linux 启动盘或者安装 CD 来启动你的计算机。然后找到你计算机硬盘上的 root 分区，使用 mount 命令挂载该分区，接着修改文件/etc/shadow。因为当我从软盘启动 Linux 时，我可以不需要输入任何密码就成为 root 用户。在密码文件里，把 root 用户的加密密码删除，所以 root 的密码将是空的。

用户帐号的信息储存在以下纯文本文件里：

```
/etc/passwd
/etc/shadow
```

文件/etc/passwd 包含了我计算机上的所有用户，并且很容易阅读。文件每行包含一个帐号信息，总共包含六个“：”间隔符号(这意味着七个字段)。举例如下

/etc/passwd 文件例子

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:99:99:Nobody:./sbin/nologin
apache:x:48:48:Apache:/var/www:/sbin/nologin
```

1	root	登录用户名
2	x	字符 x
3	0	用户标识
4	0	该用户的优先用户组标识
5	root	注释（比如用户的全名）
6	/root	用户的主目录(HOME DIRECTORY)
7	/bin/bash	用户的 shell 的名字

真正最重要的用户信息其实储存在文件 `/etc/shadow` 文件里。这个文件相对安全因为只有 `root` 用户才能阅读。在这个文件里，每行包含九个字段：

1	登录用户名
2	加密后的密码
3	从 1970 年 1 月 1 日到上一次修改密码的天数
4	距离下一次修改密码的天数
5	密码定期作修改的天数
6	密码即将过期时提前通知用户的天数
7	密码过期到用户被取消的天数
8	密码被取消时距离 1970 年 1 月 1 日的天数
9	预留字段

一些旧的 Unix 版本不包含 `/etc/shadow` 文件，而是把密码存放在 `/etc/passwd` 每行中的第二个字段。这个字段在新的系统中只显示一个“x”字符。

举个例子，我的 `/etc/shadow` 文件中的 `root` 用户看起来是这样的：

```
root:$1$BuPbmLaz$1G7.evIChyqaEI0TIzP0F.:11071:0:99999:7:-1:-1:134540356
```

当密码被消除后，看起来是这样的：

```
root::11071:0:99999:7:-1:-1:134540356
```

现在，`root` 用户就没有密码了，所以我就可以重新启动计算机，在 `login` 登录提示下，输入“`root`”，对于密码提示只需要按回车键就可了（没有密码）。登录成功后，可以立刻使用以下命令修改 `root` 密码：

```
passwd
```

有个例外，尽管在 `/etc/shadow` 删除了密码，对于 Debian 的发行版本并不允许你按“无密码”状态登录，在这种情况下，你需要做的是用其他用户（密码已经）的加密后的密码取代 `root` 的密码，然后使用该密码重新登录。

想给远程用户设置用户帐号，用电子邮件发送加密密码也是一个安全的办法，“我正在帮你在我的计算机上设置 `ftp` 帐号，请把你的加密密码用电子邮件发送给我”。等你收到加密密码后，把它插入到 `/etc/shadow` 文件里。这样，用户就可以登录了，因为只有她知道她的密码，其他人不知道。

为了让我的计算机没有“软盘访问”功能有点难度，我喜欢在没有软驱的情况下运行计算机。不幸的是，Linux 光盘现在是可以自启动的。我在 BIOS 里设置启动设备的顺序以保证系统从硬盘启动而不是软驱或者光驱，然后给 BIOS 设置加上密码保护，这样就没有其他人能够修改 BIOS 设置。但是，我还是很担心 BIOS 的密码很容易被破解，或者有人打开计算机盖把 BIOS 的电池放电来重新设置 BIOS。还可能有人把我的硬盘拿走然后在别的计算机上阅读☺。所以，我正在考虑在我的计算机上安装一个现在已经有效的 Linux“加密文件系统”，但是以其考虑这么多麻烦问题，我觉得还是干脆把计算机锁在屋里才是一个真正的好办法。这些听起来头很大吧？事实大概也是这样的。这里我只是把一些计算机的安全问题稍微罗列了一下，即使在 Linux 下，如果黑客有潜在可能可以直接接触你的计算机硬件，Linux 也是不安全的。

3.2.4 忘记了普通用户密码怎么办

如果是普通用户（非 root 用户）忘记密码，这个不是什么大问题，因为 root 可以修改任何人的密码。举个例子（要求 root 用户）

```
passwd barbara
```

按回车键后，将提示输入用户 barbara 的新密码（root 用户不需要知道旧的密码）。如果一个普通用户想要修改自己的密码，该用户会被要求输入旧的密码（这是一个安全特性，以避免当你离开你的座位有人试图修改你的密码）。

3.2.5 关闭或删除一个用户

用户帐号可以被临时关闭或者永久删除

临时关闭（锁定）一个用户帐号，并不需要修改该用户的密码。只需要在 /etc/shadow 文件里属于该用户的行的第二个字段（密码）前面加上星号“*”就可以了。星号“*”指的是该用户不允许登录。当你想要把该用户恢复正常，只需要把星号“*”去掉就可以了，用户就可以恢复正常。

以下是一个在 /etc/shadow 关闭用户 peter 的例子：

```
peter:*$1$narMEFn6$fhA1pu0U422HiSL5aggLI/:11193:0:99999:7:-1:-1:134539228
```

我也可以使用以下命令来关闭用户帐号：

```
passwd peter -l
```

使用以下命令重新释放该用户：

```
passwd peter -u
```

对于永久性（不可恢复）的删除一个用户帐号，我一般这样做：

- 以 root 登录
- 把我的用户标识改成要删除的那个用户，检查是否有新的重要的电子邮件：

```
su doomed_user_login_name
```

```
mail
```

```
logout
```

- 删除用户和用户组：

```
userdel doomed_user_login_name
```

```
groupdel doomed_user_login_name
```

- 然后把该用户从所属的其他用户组里面删除：

```
usermod -G doomed_user_login_name doomed_user_login_name
```

- 强制删除该用户的主目录和主目录下的所有文件和子目录：

```
rm -fr /home/doomed_user_login_name
```

3.2.6 文件的所有权和访问权

Linux (还有其他 Unix) 是一个安全, 多用户的操作系统, 同时, 这也产生了文件访问权限的复杂性。没有设置好文件的访问权限可能会导致一些莫名其妙的问题。充分了解文件的访问权限是管理任何多用户操作系统 (Linux, Unix, Windows NT) 的最重要环节。

我的建议是: 学习 Linux (或者其他 Unix) 的文件访问权限的内容, 你从来不会为此感到后悔。

文件所有者

每个文件 (或者目录) 从属于一个文件所有者 (一般是一个用户名) 和一个用户组。文件所有者一般来说就是生成 (或者拷贝) 这个文件的用户。用户组经常包含一个用户 - 文件所有者。用户组通常有一个名字来标识该用户, 但是也不是必须的。一个文件只能被文件所有者删除, 或者是文件所属的用户组里的其他用户, 或者是 root 用户。对于其他用户, 如果被赋予适当的权限, 也有可能修改或者删除该文件。文件所属的用户和用户组可以通过命令 `ls -l` (长文件名显示格式) 来显示:

```
ls -l junk
```

屏幕输出如下:

```
-rwx----- 1 yogin inca 27 Apr 24 14:12 junk
```

该文件属于所有者 yogin 和用户组 inca

文件的从属可以通过命令 `chown` (修改文件所有者) 和 `chgrp` (修改用户组), 一般来说需要 root 用户:

```
chown peter junk
chgrp peter junk
ls -l junk
```

执行以上三条命令后, 命令 `ls -l` 输出如下:

```
-rwx----- 1 peter peter 27 Apr 25 20:27 junk
```

当你以 root 用户身份为其他用户删除或者拷贝文件时, 修改文件的从属可能是经常要做的事情, 在做完文件的整理工作后, 把文件的拥有者改成对应的用户。

文件的权限

文件的拥有者可以把文件的访问属性设成三种不同的模式: 读(r), 写(w)和运行(x)和三个不同的用户级别: 文件所有者(u), 所属的用户组(g), 系统里的其他用户(o)。你可以检查当前的文件访问权限:

```
ls -l filename
```

如果文件对于三种不同的用户都提供三种文件访问模式，输出结果看起来应该是：

```
-rwxrwxrwx
```

跳过第一个字符“-”（第一个字符显示文件的类型，“-”表示普通文件，“d”表示目录文件，“l”表示链接文件，“c”表示字符设备，“b”表示块设备，“p”表示命名管道比如FIFO文件（First In First Out，先进先出），“f”表示堆栈文件比如LIFO文件（Last In First Out，后进先出）。

第一个字符之后的第一个三位字符组表示对于文件拥有者对该文件的权限，第二个三位字符组表示文件用户组对该文件的权限，第三个三位字符组表示系统其他用户对该文件的权限。如果没有权限，一般显示“-”字符。

以下是一个显示一个属于root的文件用户权限：该文件的拥有者root拥有所有权限，但是用户组和其他用户只能阅读和执行。

```
drwxr-xr-x 2 root root 21504 Apr 24 19:27 dev
```

第一个字符d显示该文件是一个目录。

你可以使用chmod命令来修改属于你的文件的访问权限。举个例子，以下命令将把文件junk给所有用户增加“只读”权限。

```
Chmod a+r junk
```

在以上的命令，除了用“a”表示所有用户(all)，我还可以用“u”表示用户(user)，“g”表示用户组(group)，“o”表示其他用户(other users)。除了加号“+”增加权限，我还可以使用减号“-”删除权限，等于号“=”设置权限。除了“r”表示只读权限(read)，我还可以用“w”表示写权限(write)，“x”表示执行权限(execute)。

第二个例子，以下命令将删除其他用户对junk文件的执行权限：

```
chmod o-x junk
```

除了字符，也可以使用数字来设置权限。想知道是然后工作的，看以下例子：

```
execute=1  
write=2  
read=4
```

对于指定的三种不同用户级别总结如下：

- | | |
|--------------------------|-----------|
| 0 = 没有任何权限（不能读，不能写，不能执行） | （常见） |
| 1 = 只能执行 | （看起来不太正常） |
| 2 = 只能写 | （看起来不太正常） |
| 3 = 只能写和执行 | （看起来不太正常） |
| 4 = 只读 | （常见） |
| 5 = 只读和执行 | （常见） |
| 6 = 读和写 | （常见） |

7 = 读, 写和执行 (常见)

要给三个不同的用户级别设置访问权限, 只需要把三个数字粘在一起就可以了。举例:

```
chmod 770 junk
```

将给文件拥有者和所属用户组所有权限 (读, 写和执行), 而对于其他用户没有任何权限。

```
chmod 666 junk
```

将给所有用户 (文件拥有者, 所属用户组, 其他用户) 读写权限, 但是没有执行权限。请注意这个 666 权限设置里很经常用到, 有人认为这是整个 Linux (或者其他 Unix 里) 的精髓所在。

```
chmod 411 junk
```

将给文件拥有者以只读权限, 对于所属用户组和其他用户只有执行权限。这个看起来好像没什么实际用处, 但是对于北美的 Linux 用户可能会感到很有趣, 因为 411 电话号码是他们用来获得电话号码查号帮助的。如果对于权限设置, 你还能想出什么好主意, 别忘了给我来电子邮件哦 (可能是 007?) !

文件访问权限的数字表示法叫做 “ 八位组 ” 因为是基于八进制的 (我们的日常计数系统是基于十进制)。八进制有八个数值从 0 到 7, 最大数字是 7。对应的, 十进制有十个数字从 0 到 9, 最大数字是 9。八进制表示法对于二进制的文件权限表示法的确非常方便, 每一个标志都可以通过设置成 0 或者 1 来表示 “ 允许 ” 或者 “ 不允许 ”, 如以下的例子:

用户级别:	文件拥有者	用户组	其他用户
权限设置例子	rwX	rw-	r--
缺省权限	---	--X-	wX
权限的二进制表示法	111	110	100
权限的八进制表示法	7	6	4

目录权限

目录的访问权限和一般文件的访问权限是不同的。对于一般文件:

```
r    =允许读文件内容
w    =允许修改文件内容
x    =允许执行该文件
```

对于目录而言:

```
r    =允许列出该目录下的文件和子目录
w    =允许生成和删除该目录下的文件
x    =允许访问该目录
```

使用 umask 设置缺省文件属性

当一个文件生成时, 系统给以文件缺省的文件权限。在我的系统里, 缺省权限是:

```
-rw-r--r--
```

这意味着由该用户生成的文件能被该用户读和写，而用户组和其他用户只能读。还有，在我的 Redhat 系统里，用户不能读取其他用户的主目录，因为用户主目录的缺省权限是

```
drwx-----
```

我可以使⤵以下命令检查我刚生成的文件的缺省权限：

```
umask -S
```

(可选项 - S 代表 “ 符号 ” 告诉 umask 按容易阅读的格式显示文件权限，而不是缺省的数字格式)
我可以修改新生成文件的缺省权限

```
umask u=rw,g=,o=
```

对于新生成的文件，以上命令将给文件拥有者以读和写的权限，而用户组和其他用户将没有任何访问权限。

在 umask 命令里使用数值来设置文件的缺省属性更加麻烦。因为数值显示的是从用户那里去除掉的权限 (刚好和 chmod 相反)，比如：

```
umask 000
```

对于新生成的文件，你将给所有人所有的权限。下一个例子给文件拥有者以读和写的权限，而其他用户没有任何权限：

```
umask 177
```

为了让设置对系统永久有效，在文件 /etc/profile 里修改对应的行。

3.2.7 设置替代用户标识(SUID)

我的 MP3 播放器播放音乐的时候断断续续，听起来好像受到干扰似的。这是因为程序没有得到足够的处理器 “ 能量 ” (它对处理器的要求有点大)。这可能是你的计算机实在太老掉牙了，或者是你可能同时运行太多的 “ CPU 饥渴型 ” 的程序。这种情况下，可以把播放器进程以较高的的任务优先级运行 (程序的优先级可以通过 nice 来设置，了解更多情况使用命令 man nice 或者 info nice)。或者也可以用 root 用户来运行程序，root 用户运行的程序的优先级通常给比一般用户的要高。

如果这个解决了你的 “ 音乐干扰 ” 问题，使用 suid 修改可执行文件，这样其他用户运行该程序的时候，系统能够给以该用户和文件拥有者 (一般是 root 用户) 同样的 “ 有效用户标识 ”，举例：

```
chmod a+s /usr/bin/xmms
```

这将给 xmms 程序动了一点小手脚，文件显示时：

```
ls -l /usr/bin/xmms
```

在我的计算机上输出结果如下：

```
-rwsr-sr-x 1 root root 908k Feb 22 2000 /usr/bin/xmms
```

第一个 s 指示用户替代标识(suid)已经被设置。第二个 s 指示替代组标识(sgid)已经被设置。这样，每一个运行该程序的用户将给予和程序拥有者同样有效的用户标识，和用户所属组同样有效的组标识，在这个例子里，指的是用户 root 和用户组 root。

设置替代用户标识 suid 可能会在你的计算机上产生一个安全漏洞。在一个封闭的家庭网络里，看起来不会构成什么很大的问题，因为所有的来源都可以很容易追踪得到。但是，即使在家里，我绝对不会在任何不能确定来源的程序上设定替代用户标识，即使有些安装程序建议我这么做。还有，在你的系统里的可执行文件里设置太多的替代用户标识肯定不是什么好主意，这违背了 Unix 系统的安全理念。的确，有些程序要求你设置替代用户标识才能正常运行，举个例子比如 kppp (KDE 图形用户界面下的最常用的 modem 拨号程序)。这是因为他们要求直接访问硬件，一般来说只有 root 用户才允许这么做。

如果你的系统一直有与性能相关的问题，或者一些“实时硬件”（比如 CD 刻录机）程序经常崩溃，可以试着关闭一些后台服务程序。在 Redhat Linux 下以 root 用户运行 setup，然后关闭那些你不需要的服务程序。更快捷的办法，你可以转换到命令行窗口，把整个图形用户界面关闭。这样，你的计算机的性能肯定大大好于以前的任何时候。

对于那些需要或者喜欢 Linux 作为“唯一”的操作系统（工作站，服务器，办公室桌面电脑，游戏机，多媒体等等），有专门两个 Linux 的内核补丁：“低延迟补丁”和“抢先多任务补丁”，专门用来有效解决由于系统过载导致的“延迟”问题。

3.3 用“&”，“at”，“batch”和“cron”进行作业调度

3.3.1 怎样在后台执行命令

在命令的最后加上&。举例，以下命令将在 X 窗口下后台启动 icq 客户端程序，所以当 icq 运行时，我的 X 窗口不会被挂住。

```
!icq&
```

进程识别号，任务编号 (job_number)，显示在屏幕上，在运行其他相关命令的时候你可以使用。相关的命令有：

```
fg job_number
```

fg=foreground，把后台进程恢复到前台运行，如果该任务已经被停止，将重新启动任务

```
bg job_number
```

bg=background，把前台的进程送到后台运行，如果该任务已经被停止，将重新启动任务。跟命令行后面加&具有相同效果

```
<CTRL><Z>
```

把当前正在前台运行的进程送到后台并且停止该进程

```
jobs
```

列出所有的激活的任务进程

```
kill process_ID
```

中止进程识别号为 process_ID 的进程，可以使用 ps 命令找到你要中止的进程的识别号

为了让一个后台运行的进程在你退出系统后继续运行，可以使用命令 nohup(=no hungup), 举例：

```
nohup make &
```

该命令可能在编译一个很大的程序。

3.3.2 怎样在某一特定时间执行命令(使用 at ” 和 “ batch ”)

at 命令可以让你指定特定的日期和时间来运行某个程序。举个例子，我可以在每天早上七点钟开始在我的 CDROM 上播放音乐：

```
at 7:00  
cdplay <CTRL><D>
```

在以上例子，我在命令行里输入 “ at 7:00 ” 接着按回车。接着，at 命令显示一个 “ at> ” 的提示符，然后我输入 “ cdplay ” 接着按回车，最后，我同时按住<CTRL>和<D>结束输入。如果我接着按回车，另外一个 “ at> ” 提示符出现，我可以接着输入那些我希望在早上 7 点钟运行的程序。然后在最后输入<CTRL><D>。<CTRL><D>就象是向当前的输入送一个文件结束符合，告诉输入已经完毕。不要连续按<CTRL><D>两次，否则你就自动退出系统了。

你可以使用以下命令列出当前在任务调度表里的任务：

```
at -l
```

你可以看到那些还在任务调度表里还在等待的程序的清单。

如果你改变主意，绝对把某个程序从任务调度表里删除。举例：

```
atrm 8
```

该命令将从任务调度表里删除命令编号为 8 的程序
我也可以安排一个程序在稍晚一点的时间运行，比如：

```
at 23:55 12/31/00  
startx
```

该命令将在新千夜晚启动我的 X-Windows 系统 (2000 年 12 月 31 日，午夜前 5 分钟) 。

如果你不能运行 `at` 命令，查看一下是否任务调度服务程序(`atd`)已经被加载（可以使用 `root` 用户运行 `ntsysv` 加载）。如果 `at` 命令对 `root` 用户有效但是对于普通用户无效，检查一下是否文件 `/etc/at.deny` 是否存在，同时，确认文件 `/etc/at.allow` 不存在。这是所有用户可以运行 `at` 命令的缺省设置。如果你想要只有指定的用户才能够运行 `at` 命令，可以生成文件 `/etc/at.allow` 并且把用户的名单加到文件里。

对于 `at` 命令其他的可选项，运行

```
man at
```

当系统载荷很低的时候，如果你希望在后台运行一个“处理器饥渴型”的任务，你可以选择使用 `batch` 命令。举例，我可以运行 `setiathome`（一个数据分析程序用来帮助寻找外部智能生命，SETI）：

```
batch
```

```
at>setiathome <CTRL><D>
```

在以上例子里，我在命令行输入 `batch` 然后回车，在“`at>`”命令提示符下，我输入想要在后台运行的程序名称。程序试图立刻运行，但是其实会等到系统载荷小于 0.8 的时候才会开始运行。你可以通过检查虚拟文件 `/proc/loadavg` 来检查系统载荷：

```
cat /proc/loadavg
```

当批命令完成后，输出结果将会以电子邮件的方式送到我的信箱。

3.3.3 怎样设置“cron”

`cron`（一个 Linux 的后台进程，经常在夜间运行）在 Redhat 系统里是缺省配置。所有你不需要做任何事情，除非你想在系统里加入一些自己的任务，或者想修改一些任务的运行时间表。

请注意，要使你系统里的一些功能长期运行正常，`cron` 可能是不可缺少的。其他和 `cron` 关联的比如：

- 重建 `locate` 命令需要使用的数据库文件
- 清除 `/tmp` 目录
- 重建命令手册
- “翻转”记录文件，比如，丢弃旧的记录，重新命名中间过程记录，产生新的记录等等
- 运行其他任务。比如把你最近拷贝的字体加入到系统里面

所以，总是把你的 Linux 在夜晚关机可能不是什么好的主意。这样可能导致 `cron` 根本没有机会来运行任何任务。如果你倾向于在夜晚关机，最好调整 `cron` 在一些其他的时间运行。

想了解 `cron` 什么时候被唤醒开始运行它的任务，可以查看 `/etc/crontab`，举例：

```
cat /etc/crontab
```

你会看到内容大概如下：

```
# run-parts
01 * * * * root run-parts /etc/cron.hourly
02 4 * * * root run-parts /etc/cron.daily
22 4 * * 0 root run-parts /etc/cron.weekly
42 4 1 * * root run-parts /etc/cron.monthly
```

你可以看到有四种不同的 cron 任务：每小时，每天，每周和每月。你可以修改修改它，或者加入你自己的任务。以下介绍它是如何工作的：

每列的顺序表示如下：

分钟（0 - 59），小时（0 - 23），日期（1 - 31），月份（1 - 12），星期（从 0 - 6 分别表示星期天到星期六）。“*”表示任何有效的数值。

在以上例子里，“每小时”的任务在时钟的“第一分钟”运行，这相当于每小时发生一次。“每日”的任务发生在每天时钟的 4 点 02 分，刚好每天一次。“每周”任务发生在星期天早晨 4 点 02 分。“每月”的任务发生在每月第一天的 4 点 42 分。该任务的运行脚本名称放在每行的最后。

如果你想要让早上 4 点的任务在中午执行，只需要把 4 改成 12 就可以了。Cron 每分钟被唤醒一次检查是否/etc/crontab 已经被修改，所以当你修改完文件后不需要重新启动 cron 程序。

如果你想要增加一个任务到 cron 里，把运行任务的脚本（或者脚本的连接）放在对应的目录里：
/etc/cron.hourly，/etc/cron.daily，/etc/cron.weekly，/etc/cron.monthly。

以下是/etc/crontab 的一个例子，该任务每周运行三次（星期一，星期三，星期五）：

```
02 4 * * 1,3,5 root run-parts/etc/cron.weekly
```

以下是一个新闻组上的例子，显示如何自动用电子邮件发送一个日记文件

回复：help in crontab

发信：Dean Thompson <Dean.Thompson@csse.monash.edu.au> 日期：2001-03-03 16:35

Newsgroups: comp.os.linux.admin,comp.os.linux.networking,comp.os.linux.security

> 我该如何在/etc/crontab 文件里设置，让每天的工作记录文件自动发送到信箱 abc@abc.com < /var/log ?

你可以试试看以下的例子

```
0 0 * * * (/bin/mail abc@abc.com < /var/log/messages) > /dev/null 2>&1
```

3.4 Shell

3.4.1 什么是 Shell? 需要一个不同的 Shell 吗?

shell 是计算机用来解释你输入的命令然后决定进行何种处理的程序。shell 也可以在非交互模式下调用。举个例子，可以把一些要输入的命令预先存放在文本文件里（叫做 shell 脚本）然后执行该文件。可以把 shell 看作是 DOS 下面的 command.com（命令行解释器），而 shell 脚本就象是 DOS 的批处理文件 (*.bat)。和 DOS 相比较，shell 的功能要先进和丰富得多。

在 Linux 系统下有几个不同的 shell (如果你全部安装) :

bash	"Bourne Again" shell
sh	Bourne shell, 在很多UNIX下是标准的shell
csh	C shell, 和C编程语言语法相近, 在大部分UNIX下有效
pdksh	public domain Korn shell
tcsh	微型 C shell, 在小的系统里经常使用
sash	单机版本的shell, 当文件库无效的时候可以使用
ash, zsh等	可能还有更多其他版本

Linux系统下缺省的shell是bash (你的计算机上应该也是), 这是一个非常出色和标准的shell, 象我这样的Linux新手实在想不出有什么理由要修改它。bash不仅对UNIX下应用广泛的Bourne shell后向兼容, 而且合并了其他shell的一些非常有用的特性。从一个新手的角度来看, 出于历史原因, 一些既存的shell脚本可能要求特定的shell程序才能够运行, 为了保证shell脚本充分的后向兼容性, Linux保留了不同类型的shell。一些shell可能很有用, 比如, 如果你准备给一些小型“嵌入式设备”写程序, 那么你可能需要tcsh (微型C shell)。

你可以检查你现在正在运行的shell :

```
echo $SHELL
```

如果你想要试一下其他类型的shell, 举个例子 :

```
tcsh
```

该命令将启动微型C shell。结束使用, 输入 :

```
exit
```

这将回到你原来的shell (在第一个shell下输入exit会把你退出系统)。你可以通过显示环境变量“shell level” (shell层次) 看看你当前环境下的shell的堆栈 :

```
echo $SHLVL
```

在以上的命令里, “\$”表示“扩展shell环境变量”, “SHLVL”是“变量名”, “echo”是屏幕输出命令。

每个用户的缺省shell在文件/etc/passwd最后一个字段指定。如果你真的想要修改它, (要求root用户) 把你想要的shell替换文件里的“/bin/bash”。

3.4.2 怎样定制自己的 Shell

在我的计算机上, 命令提示符看起来是这样的 :

```
[stan@marie stan]$ _
```

在这里，“stan”是用户名，“marie”是计算机名，第二个“stan”是我当前的工作目录，而“_”代表光标。

提示符通过环境变量PS1来设置。可以使用以下命令显示当前设置：

```
echo $PS1
```

提示符的系统范围（对于系统里的所有用户有效）的设定在文件/etc/bashrc里，可能包含以下的行：

```
PS1="[ \u@ \h \W] \ $ "
```

要定制提示符，我可以编辑文件/etc/bashrc（root用户）然后在两个引号之内输入任何你想要的文本。以下是一些我可能会使用的特别代码：

\u	-	当前用户的用户名（=\$LOGNAME）
\h	-	运行shell的计算机名（hostname）
\H	-	完整的计算机名
\W	-	当前工作目录名
\w	-	当前工作命令全路径名
\\$	-	对于普通用户显示\$，对于超级用户显示#
\!	-	当前命令的历史编号
\#	-	当前命令编号（在当前的shell下运行的命令）
\d	-	当前日期
\t	-	当前时间（24小时制）
\T	-	当前时间（12小时制），bash 2.0有效
\@	-	当前时间（AM/PM格式），bash 2.0有效
\s	-	shell的类型
\a	-	告警声音
\j	-	用户的任务数
\n	-	换行
\\	-	反斜杠
\[-	不可打印字符开始标志
\]	-	不可打印字符结束标志
\nnn	-	ASCII字符对应的八进制数值
\$(date)	-	date（输出当前日期）命令输出或者其他的命令

以下是一个关于如何加彩色的例子。更详细的信息参考后面一节。

```
PS1="\[\033[1;32m\][ \u@ \h \W] \ $\[\033[0m\] "
```

这里还存在有“第二层”输出提示符，在环境变量里是PS2。当需要附加输入时，shell使用“第二层”输出提示，在我的系统上“第二层”输出提示符是“>”。我不是很在意“第二层”输出提示符，尽管我可以修改PS1相同的办法修改PS2。除此之外，还有PS3，PS4，但是非常少见。

3.4.3 设置文本终端的颜色

文本终端的颜色可以使用“ANSI 非常规字符序列”来生成。举例：

```
echo -e "\033[44;37;5m ME \033[0m COOL"
```

以上命令设置背景成为蓝色，前景白色，闪烁光标，输出字符“ME”，然后重新设置屏幕到缺省设置，输出字符“COOL”。“e”是命令echo的一个可选项，它用于激活特殊字符的解析器。

“\033”引导非常规字符序列。“m”意味着设置属性然后结束非常规字符序列，这个例子里真正有效的字符是“44;37;5”和“0”。

修改“44;37;5”可以生成不同颜色的组合，数值和编码的前后顺序没有关系。可以选择的编码如下所示：

编码	颜色/动作
0	重新设置属性到缺省设置
1	设置粗体
2	设置一半亮度（模拟彩色显示器的颜色）
4	设置下划线（模拟彩色显示器的颜色）
5	设置闪烁
7	设置反向图象
22	设置一般密度
24	关闭下划线
25	关闭闪烁
27	关闭反向图象
30	设置黑色前景
31	设置红色前景
32	设置绿色前景
33	设置棕色前景
34	设置蓝色前景
35	设置紫色前景
36	设置青色前景
37	设置白色前景
38	在缺省的前景颜色上设置下划线
39	在缺省的前景颜色上关闭下划线
40	设置黑色背景
41	设置红色背景
42	设置绿色背景
43	设置棕色背景
44	设置蓝色背景
45	设置紫色背景
46	设置青色背景
47	设置白色背景
49	设置缺省黑色背景

其他有趣的代码还有：

\033[2J	清除屏幕
\033[0q	关闭所有的键盘指示灯
\033[1q	设置“滚动锁定”指示灯 (Scroll Lock)
\033[2q	设置“数值锁定”指示灯 (Num Lock)
\033[3q	设置“大写锁定”指示灯 (Caps Lock)
\033[15:40H	把关闭移动到第15行，40列
\007	发蜂鸣生beep

想要了解更多，使用命令 `man console_codes`

3.4.4 在文本模式或 X 模式控制台下,如何打印符号

以下描述的是如何快速访问PC扩展字符集 (codes 128-255) 的办法，它在PC世界里非常通用：微软的Windows，DOS（如果你安装了ANSI的驱动程序），任何文本模式的Linux应用程序（包含shell命令行）。

它是这样工作的。首先确认<Num Lock>是打开的，接着按住<ALT>键。当<ALT>被按住的时候，在小的数字键盘上输入：0181。现在，释放<ALT>接着希腊字母mu" μ 就出现了。以下是一个完整的清单：

```

128 ? 147 ? 166 ! 185 ± 204 ì 223 ß 242 ò
129 ? 148 ? 167 $ 186 ° 205 í 224 à 243 ó
130 , 149 * 168 " 187 » 206 î 225 á 244 ô
131 f 150 - 169 © 188 ¤ 207 ï 226 â 245 õ
132 ? 151 - 170 ¢ 189 ¤ 208 ð 227 ã 246 ö
133 ? 152 ~ 171 « 190 ¤ 209 ñ 228 ä 247 ÷
134 ? 153 ? 172 ¬ 191 ¿ 210 ò 229 å 248 ø
135 ? 154 s 173 -
192 À 211 Ó 230 æ 249 ù
136 ^ 155 > 174 ® 193 Á 212 Ô 231 ç 250 ú
137 ? 156 ? 175 ¯ 194 Â 213 Õ 232 è 251 û
138 s 157 ? 176 ° 195 Ã 214 Ö 233 é 252 ü
139 < 158 ? 177 ± 196 Ä 215 × 234 ê 253 ý
140 ? 159 Y 178 ¢ 197 Å 216 Ø 235 ë 254 þ
141 ? 160 179 * 198 Æ 217 Ù 236 ì 255 ÿ
142 ? 161 ; 180 ´ 199 Ç 218 Ú 237 í
143 ? 162 ¢ 181 µ 200 È 219 Û 238 î
144 ? 163 £ 182 ¶ 201 É 220 Ü 239 ï
145 ? 164 ¢ 183 · 202 Ê 221 Ý 240 ð
146 ? 165 ₣ 184 , 203 Ë 222 Þ 241 ñ

```

现在，如果我真的想要有点与众不同，我可以给一个文件命名：`µ · ° C ± b3`。微软Windows，DOS和Unicode的字符集和上面显示的有一些小差别，但是有用的“核心部分”是一样的。如果你想要了解详细的ANSI的信息，参考：<http://www.hclrss.demon.co.uk/demos/ansi.html>。Linux使用Unicode标准。

在X - Windows下，以上的组合键不能正常工作，但是我可以使用：

```
kchselect&
```

或者

```
gcharmap&
```

选择一个Unicode然后复制到我的应用程序里。不是所有的Unicode现在都有效，但是大部分是有效的。在以下链接，你还可以找到一些通用的Unicode编码（数值）和它们对应的html符号。

http://www.hclrss.demon.co.uk/demos/ent4_frame.html.

3.4.5 如何编写简单的 Shell 脚本(Script)文件

新建一个文本文件包含所需要的脚本。举例，我会使用pico编辑器写一个脚本用来运行程序tar，带上必要的可选项可以用来解压从因特网下载下来的*.tar的文件（我好像总是记不住tar的所有参赛）。我决定把我的脚本名称叫做“untar”：

```
pico untar
```

因为在我的当前工作目录里untar文件不存在，所有pico文本编辑器自动创建这个文件，现在，我输入以下内容：

```
#!/bin/bash
echo this is the script file $0
echo untarring the file $1
# this calls tar with options -xvzf (extract,
# verbose, filter through gzip, input filename)
tar -xvzf $1
```

我使用<CTRL>O保存这个文件，然后<CTRL>X退出。

脚本的第一行，以“#!”开始是特别的提示 - 它告诉shell应该用哪一个程序来解释我的脚本。在这个例子里，我使用bash shell /bin/bash。第一行必须使用“#!”开头，否则脚本不会运行（系统认为是一个文本文件）。其他以“#”开始的行是注释行只是给作者和读者使用的，计算机将跳过这些行。

在以上脚本里，参数\$0，\$1，\$2...是传递到脚本里面的参数。举个例子，如果我运行我的脚本名“myscript”带七个参数如下：

```
myscript a b c d e f g
```

那么，参数\$0就是myscript，\$1就是a，\$2就是b，\$3就是c，依此类推。

脚本的第二行和第三行，echo命令输出所有在它后面同一行上的文本，然后扩展在脚本里对应的参数\$0和\$1。第四行和第五行是我写的注释文本，提醒我在这个脚本里要做的的事情。只有最后一行是真正起作用的。

一旦脚本已经写好，我把文件属性改成对文件拥有者是“可执行”的：

```
chmod u+x untar
```

然后我的脚本就可以这样运行了：

```
./untar my_tar.tar.gz
```

Linux脚本的确非常丰富，灵活，功能强劲，还可能有点复杂。然而，对于日常任务编写一些简单脚本，它并不需要什么特别高深的知识。你可以把一些要用到的命令放在一起，一个接一个，输入到文件里。我使用脚本很频繁是因为我太懒了，不想一次一次的输入相同的命令。

一个最简单的方法，可以把一组命令放在文本文件里然后使用source命令传递给shell让它直接运行：

```
source my_file
```

这个时候就不需要在文本第一行加上“#!”的标志了。

3.4.6 引号的含义

一般来说，以下字符对于 shell 有特殊的含义：

```
\ ' " ` < > [ ] ? | ; # $ ^ & * ( ) = <Space> <Tab> <Newline>
```

这里有四种不同的符号：反斜杠(\)，单引号(‘)，双引号(“)，反向单引号(`)。

- 反斜杠(\)表示：关闭后面字符的特殊含义
- 单引号(‘)表示：关闭在两个单引号之间所有字符的特殊含义
- 双引号(“)表示：关闭在两个双引号之间所有字符的特殊含义除了\$ ` \
- 反向单引号(`)表示：告诉 shell 首先运行两个反向单引号之间的命令，然后把得到的结果再传递给两个单引号之外的命令。同样的功能也可以通过“\$command”命令来实现，而且可能会更方便。

举个例子，我可以创建一个奇怪的目录名叫做“*”通过使用“\”或者“'”符号：

```
mkdir \*
mkdir ' * '
```

这屏蔽了“*”对于 shell 的特别含义。如果没有“\”，“*”意味着当前目录下所有文件。

3.4.7 输入输出重定向

有三个最重要的输入输出流：标准输入(stdin),标准输出(stdout),标准错误(stderr)。它们对于控制台(“控制台”指的是键盘用于输入，屏幕用于输出)来说是缺省的，但是它们可以被重定向。

重定向标准输出，可以使用“>”符号，举例：

```
dir my_dir > filelisting.txt
```

将把 dir 命令的标准屏幕输出重定向到文本文件 filelisting.txt 文件里，所以屏幕上没有任何输出。这个文件可以用来编辑(比如使用 pico 文本编辑器)或者合并到其他的文件里。

重定向标准错误，可以使用结构“2>”，举例：

```
dir my_dir 2> errorlisting.txt
```

以上命令将送标准输出到屏幕上，如果没有错误信息，将没有任何信息写到 errorlisting.txt 文件里。如果出错，则没有什么东西输出到屏幕，而文件 errorlisting.txt 将包含错误信息。错误信息有可能是这样的：

```
dir: my_dir: Permission denied
```

最后,我也可以把标准输出和标准错误一起输出到同一个文件里,

```
dir my_dir > file_and_error_listing.txt 2>&1
```

以上命令先重定向标准输出到文本文件里,然后再重定向标准错误到和标准输出同样的位置。它如何实现可能看起来有点古怪,但是是可行的。

在以上的例子里,如果重定向的文件已经存在,该文件会被覆盖。如果你要追加到该文件的末尾,可以使用“>>”符号,以上的例子就变成:

```
dir my_dir >> filelisting.txt
dir my_dir 2>> errorlisting.txt
dir my_dir >> file_and_error_listing.txt 2 > &1
```

如果你对“2>”感到很迷惑,这里有一个简单的办法可以帮你理解,标准流有标准的解析器:“0”代表标准输入,“1”代表标准输出,“2”代表标准错误。

```
dir my_dir > file.txt
```

是以下命令的简写方式:

```
dir my_dir 1 > file.txt
```

那么以下命令就是用来输出标准错误:

```
dir my_dir 2 > file.txt
```

还有,你还可以使用符号“|”(管道命令)把一个命令的标准输出送到另外一个命令的标准输入。在以下这个标准的例子里,dir命令的标准输出通过管道输入到命令more里(输出满屏的时候自动暂停):

```
dir | more
```

你还可以使用“tee”命令把标准输出同时写到文件和屏幕,

```
dir | tee filelisting.txt
```

tee是“T型连接器”的模拟音,在管道中的主要的用途是分流。

这个部分这样都用来讲述标准输出重定向,对于标准输入重定向不像标准输出重定向那么有用,但是它可以使用以下方式实现:

```
cat < my_file
```

还有一种叫做“直接插入式”的标准输出,可以通过“<<”来实现。不要管它了,看起来对我没有什么实际用处。不过,如果你真的想知道,这里有一个例子(这里“>”式第二个提示符)

```
cat << my_marker
> my_line_from_the_keyboard
> another_line_from_the_keyboard
> my_marker
```

除了重定向到常规文件和“过滤器”之外（如以上的例子所示），你还可以重定向到设备和其他特殊文件。看下面这些例子。

重定向到设备文件的例子。以下命令将显示文件列表到第四个文本终端：

```
dir > /dev/tty4
```

以下是一个重定向到一个特殊的 FIFO(先进先出)文件的例子。该命令送信息“you are lucky”到叫做“lucky”的 ICQ 用户 UIN 77777777（假定你已经用你的 ICQ 程序连接到 ICQF 服务器上）

```
echo message 77777777 "you are lucky" < ~/.licq/licq_fifo
```

以上的例子能够工作是因为在你 licq 目录下的文件“licq_fifo”是一个特别的 FIFO 序列文件。以上这个例子，对比于在图形用户界面下的 ICQ 程序有什么特别有用的地方吗？举个例子，你可以写一个短的脚本带上多个信息给你的那些 ICQ 伙伴们：

```
#!/bin/bash
echo Messaging UIN: $1 Message: $2 Times: $3
# The next command puts puts your licq in the status "on-line, invisible".
echo 'status *online' > ~/.licq/licq_fifo
c=0
while [ $c -le $3]
do
echo message $1 $2 > ~/.licq/licq_fifo
c=`expr $c + 1`
echo $c " "
done
echo 'status offline' > ~/.licq/licq_fifo
echo "all done"
```

这个例子利用了 licq 通信模型（FIFO 文件）和简单的文件重定向功能，给你一个关于如何“自动化”licq 的主意。

3.4.8 Shell 的特殊字符(metacharacters)

一般来说，这些字符对于 shell 有特别的含义：

```
\ ' " ` < > | ; <Space> <Tab> <Newline> ( ) [ ] ? # $ ^ & * =
```

以下是这些字符的含义：

\ ‘ “ 和 ‘ 主要用来注释，前面已经描述过 (参见 3.4.6)。

< 和 > 主要用来输入和输出重定向

| 是管道命令，管道左边的标准输出是管道右边的标准输入

； 用于间隔在同一命令行上的几个命令

<Space> 和 <Tab> 间用于分开命令的字符和单词

<NewLine> 完成一条命令或者一组命令

() 用于封装需要使用不同的 shell 启动的命令，比如 (dir)

{ } 用于封装要用当前 shell 启动的一组命令，比如 { dir }，需要空格间隔

& 使当前命令在后台运行 (有它自己独立的进程)，所以下一条命令不需要等待前一条命令结束才能开始。

* 当搜索文件时，它匹配除了以 “.” 开头的文件

? 当搜索文件时，它匹配任何单个字符

[] 当搜索文件时，它匹配任何在[]里面的单个字符

&& 是用于连接两个命令的“与操作”，

command1 && command2，只有当 command1 退出状态为 0 时 (没有错误)，command2 才会被执行。比如，cat file1 && cat file2 只有当 file1 正常显示时，file2 才能被显示。

|| 是用于连接两个命令的“或操作”

command1 || command2，只有当 command1 退出状态非 0 时 (有错误)，command2 才会被显示。比如：cat file1 || cat file2 只有当显示 file1 出错时，file2 才能被显示

= 指定值给变量

举例，命令 me=blahblah 设定值 “blahblah” 给变量 “me”，我可以输出变量名：

```
echo $me
```

\$ 预处理扩展变量名

变量可以使用 “=” 来设定值，也可以通过预先变量设定来设置

\$0 被执行的 shell 脚本的名称

\$# 按位置对应的命令输入参数，\$1 第一个参数，\$2 第二个参数，\$3 第三个参数... 直到\$9

\$* 扩展所有的位置参数给命令

`$@` 扩展所有的位置参数给命令, 但是当“`$@`”使用时, 参数个别标注

3.5 如何安装软件包(Package)及 rpm 软件包管理器

3.5.1 如何安装从 Internet 下载的程序?

答案基于你所下载的是哪一种软件包. 如果你下载的程序是 Red Hat 二进制形式软件包*.rpm (如果假定一个选择, 我选择这种格式), 你能避免很多安装上的问题.

RedHat 二进制形式程序包的安装

如果我想安装的程序是 RedHat 二进制形式软件包(*.rpm), 我能使用命令行或 GUI 工具. 我喜欢用命令行因为它运行快而且不出错. Red Hat 软件包管理器安装程序被称作 rpm . 首先我要阅读软件包内容的信息(选项):

```
rpm -qpi my_new_file.rpm
```

该行查询(模式"q", 必须在破折号后第一个字母)尚未安装的软件包(选项"p")以便显示该软件包所包含的信息(选项"i"). 目如果我想安装该程序, 我(作为超级用户)运行:

```
rpm -ihv my_new_file.rpm
```

上面的命令安装了该组件. 运行 rpm 来安装该组件(模式"i", 必须是在破折号后第一个字母), 此时屏幕上显示比平常更多的信息(选项"h"=显示"hashes"来展现解包过程, 选项"v"=详细). 软件包内容放到了它们所在的目录下(rpm 知道它们的位置). 安装后, 准备运行, 我得知可执行文件的名称和位置. 如果找不到可执行文件, 下面一行列出了软件包所包含的所有文件和它们所在的目录:

```
rpm -qpl my_new_file.rpm
```

该行查询(模式"q")尚未安装的软件包(选项"p")以便显示该软件包所包含的所有文件清单(选项"l").

调用 rpm 的 GUI 前端界面是: gnopro (旧版本, 导致 RH6.0 有些混淆, 但新版本已改进), kpackage (用于更新的版本), 老的 glint (很慢, 用于 RH5.2).

Troubleshooting. rpm 被认为是智能软件包管理器. 如果安装失败, 我可以读错误信息, 并能知道下一步所要做的事:

(1) 安装失败是由于我有相同软件包的早期版本且版本冲突. 解决方法: 不安装, 仅需升级该软件包.

```
rpm -Uvh my_new_file.rpm
```

(2) 安装失败是因为须首先安装另一个软件包. 我得找到这个缺少的软件包并安装它, 然后在尝试安装该套件. 在下面特殊的例子里, 我可以选择忽略缺少的附件(我真的知道我正在做的会导致软件发生故障):

```
rpm -ivh --nodeps my_new_file.rpm
```

或甚至可能是:

```
rpm -ivh --nodeps --force my_new_file.rpm
```

从 TARBALL 源代码中安装

如果我从网上所下载的是一个已压缩的 tarball (*.tar.gz or *.tgz)形式的 Linux 源代码,安装步骤比二进制的 rpm 更长更棘手.我作为超级用户来安装该程序.

首先,我改变我当前的工作目录到/usr/local :

```
cd /usr/local
```

其次, 我将我从网上下载的 tarball 解压:

```
tar -xvzf /home/the_dir_where_the_tarball_is/my_tarball.tar.gz
```

这选取了 (选项 "x") *.tar.gz (or *.tgz) tarball 的内容, 解压它 (选项 "z"),同时给我比平常更多的信息 (选项"v" = 详细). 请注意选项"f"是指"file", 所以字母"f"后面要紧跟着文件名. Tarball 的内容被释放到在我的当前目录下的由 tar 创建的子目录里. 典型的例/usr/local/. 该 tarball 知道新的子目录的名字.

如果 tarball 未被压缩(比如, *.tar), 我可以用:

```
tar -xvf /home/the_dir_where_the_tarball_is/my_tarball.tar
```

第三, 我要知道新目录的名称, 然后用 cd 进入:

```
dir  
cd the_new_program_subdir
```

因为有些目录的名字很长, 我使用 autocompletion 选项来节省键盘输入时间—我只要输入头几个字母再按下<TAB> .

第四, 通过执行下面的三条命令能编译大多数程序:

```
./configure  
make  
make install
```

上面的命令要花点时间完成(1分钟?0.5小时?). 如果上面命令的任意一个运行失败, 可以读 README 或 INSTALL 或任何该程序所提供的信息. 一些程序要求环境的定制(如路径的定义)或额外库文件的安装, 或其它. 有时这是痛苦的事. 很简单的程序可能不需要 "./configure" 或/和 "make install" 步骤, 在此情况下"make"将单独工作.

第五, 如果运行正常, 我可以发现我刚刚编译的新的可执行文件. 当运行以下命令时, 该可执行文件的名字显示绿色:

```
ls --color
```

现在, 我能运行这个可执行文件, 例如:

```
./the_executable
```

一些程序自动将可执行文件安装到/usr/local/bin 目录下, 所以我可以试着:

```
/usr/local/bin/the_executable
```

第六, 如果我想更频繁运行这个程序, 我从/usr/local/bin 目录创建一个到该可执行文件的软连接:

```
cd /usr/local/bin  
ln -s /usr/local/the_new_program_subdir/the_executable .
```

这种方式, 使可执行文件(实际是到该文件的软连接)在我设置的路径中. 只需简单输入其文件名即可运行(不需输入到该文件的全路径). 一些程序安装了可执行文件(或一个连接)在"bin" 目录下, 在此情况下你能略过最后一步。

从 RPM 软件包源代码安装

也有作为"rpm 源代码" 的软件包. 正如在本章开始部分所描述, 它们要求用"rpm" 工具对 *.rpm 软件包进行安装. 但是因为 "rpm" 安装了源代码(典型地使用 C 语言写的源代码), 我需要通过执行相同的"./configure ; make ; make install" 顺序编译源代码, 正如从 tarballs 源代码中安装一样 (参见前面的答案).

第 4 部分: Linux 管理员新手常见问题

Linux 新手管理员指南(中文版)

LINUX NEWBIE ADMINISTRATOR GUIDE (Simplified Chinese Version)

中文版本 1.0 由幻影快递 Linux 小组(SLT-Shadowfax Linux Team) 翻译整理

Linux Newbie Administrator 英文原版版权所有: Peter and Stan Klimas.

<http://sunsite.dk/linux-newbie>

中文版本版权所有: 幻影快递 Linux 小组 ShadowfaxLinux@hotmail.com

本章内容:

- 4.1 关于启动 (LILO 和 GRUB)
 - 4.1.0 LILO 和 GRUB
 - 4.1.1 Linux 不能发现我的所有内存
 - 4.1.2 LILO 仅显示 LI (或 LIL) 并死机
 - 4.1.3 我如何能让操作系统变为由 LILO 缺省引导?
 - 4.1.4 在启动时 LILO 提示符在屏幕上停留时间太短 (或太长)
 - 4.1.5 卸载 Linux
- 4.2 访问我的驱动器
 - 4.2.1 我的驱动器在哪里?
 - 4.2.2 我如何访问我的 CDROM?
 - 4.2.3 如何挂载软盘, zip 磁盘驱动器, dos 分区, 或网络驱动器?
 - 4.2.4 如何通过 Samba 挂载远程 MS Windows 的文件系统?
 - 4.2.5 有快捷方式访问在 DOS/Windows 软盘上的文件吗?
 - 4.2.6 作为超级用户我可以挂载. 普通用户能做到吗?
 - 4.2.7 挂载命令太长, 我如何用别名简化?
 - 4.2.8 我能自动挂载吗?
 - 4.2.9 我如何得到已识别的在并行端口 (外部) 的 zip 磁盘驱动器?
 - 4.2.10 我能设置 32 位硬盘的 I/O?
 - 4.2.11 我所打开的文件数已达到限量 (错误信息)
 - 4.2.12 我加了一个新硬盘, 开始使用时要做什么?
 - 4.2.13 交换分区 SWAP
- 4.3 使用 X-Windows 工作
 - 4.3.1 怎样切换文本和图形控制台?
 - 4.3.2 我怎样在 X-server 上安装显示卡、监视器以及鼠标?
 - 4.3.3 我能有一个 GUI 登陆提示符吗?
 - 4.3.4 我怎么安装 KDE? (例如, 在 RedHat5.2)
 - 4.3.5 我怎么将默认桌面改变成 KDE (如 Gnome 以及其他类型的 KDE)?
 - 4.3.6 我能够同时运行多个 Linux 阶段作业吗?
 - 4.3.7 在不关闭我的 Linux 阶段作业的前提下可以开启第二个 GUI 登录提示符吗?
 - 4.3.8 如何遥控 X-Windows?
 - 4.3.9 我如何从 MS Windows 分区安装 TrueType 字体?
 - 4.3.10 如何进行拷贝粘贴操作?
 - 4.3.11 我如何使用 VNC 来显示和控制一个遥控桌面?

4.4 : 基本配置

- 4.4.1 如何设置我的声卡 ?
 - 4.4.2 如何设置我的打印机 ?
 - 4.4.3 Word Perfect 8 没有适合我的打印机的驱动程序
 - 4.4.4 配置文件在哪里 ?
 - 4.4.5 这些设备文件都是什么 ?
- #### 4.4 网络
- 4.5.1 有必要建立家庭网络吗 ?
 - 4.5.2 怎样建立我的家庭网络 ?
 - 4.5.3 设置 PPP 拨号 GUI 的问题
 - 4.5.4 如何不通过 modem 用我的联网电脑浏览互联网 ?
 - 4.5.5 怎样使用 Samba ?
 - 4.5.6 Sendmail
 - 4.5.7 简单的 WEB 服务器 (Apache 阿帕奇)
 - 4.5.8 简单的 FTP 服务器
 - 4.5.9 如何通过电话线外部远程访问我的计算机 ?
 - 4.5.10 家庭计算机是否会遭黑客攻击吗 ?
-

4.1 关于启动 (LILO 和 GRUB)

4.1.0 LILO 和 GRUB

新的 GRUB 和旧的 LILO 二者都是系统引导载入程序。它们使你在系统引导时选择不同的操作系统成为可能。大多数 (所有 ?) 在这一部分所描述的系统引导问题或许能在安装最近的 Linux 内核和最新的 GRUB 引导装入程序时得到解决。GRUB 比 LILO 更好是因为 LILO 依赖硬盘的绝对地址来找到引导映像而 GRUB 能理解文件系统并寻找包含那个引导映像的文件。在安装时, 当系统给与选择时, 我们推荐使用 GRUB。

GRUB 主要配置文件是 `/boot/grub/menu.lst` (或 `grub.conf`, 在我的系统中是符号连接)。这里是一些在这个文件里发现的那些项目的注释。

行首带有#号的是注释。

```
timeout 5
```

#除非按下某键, 否则 5 秒后以上设置开始导入缺省的操作系统。

```
default 0
```

以上设置使菜单列表的第一项成为缺省的操作系统。 我能使用 "default 3" 让菜单的第四项成为缺省的操作系统。

```
title linux
kernel (hd0,2)/boot/vmlinuz root=/dev/hda3 mem=64M hdc=ide-scsi
```

以上两行定义了引导时间菜单项并设置了该项的引导动作。

第一行命名该项为 "linux"。

```
# 第二行指明内核位于第一个物理硬盘 (hd0), 第三个分区 (2), 引导映像为文件 /boot/vmlinix
# 而且在第二行, 下列选项传到内核:
# root=/dev/hda3 (即, 在第一个硬盘(hda)上有三个分区, 系统引导文件在第三个区上 hda3
# mem=64M (即, 强迫使用 64 兆的物理内存)
# hdc=ide-scsi (在我 CD ROM 中使用 SCSI 模拟器因为它是一个 CD 刻录机).
```

```
initrd /boot/initrd-2.4.17-custom.img
```

```
# 当模块载入到"initial ram disk" (initrd)时, 定义包含了在引导时间内所需的模块的文件。
# 我使用 mkinitrd /boot/initrd-2.4.7-10custom.img 2.4.7-custom 来重新创建我的[新映像], (因为我重新编译了内核)。
```

使用 `info grub` 来查看 GRUB 文档。

4.1.1 Linux 不能发现我的所有内存

如果你有多于 64 兆的物理内存, 版本号 2.0.36 或更低的 Linux 内核将缺省地使用仅仅最初的 64 兆。为了查看 Linux 在你的系统占用了多少内存, 输入:

```
cat /proc/meminfo
或
free
```

你能使用下面命令查看你的 Linux 内核版本号:

```
uname -a
```

最近流行的带有“内存问题”的内核版本为 2.0.36, 是出现在 RedHat 5.2。我的 RedHat 6.0 带有版本号为 2.2.5-15 的内核不再有这个“内存问题”。

在 RedHat 5.2 中, 为了能让多于 64 兆的内存得到识别, 你必须编辑 (作为超级用户) 文件 `/etc/lilo.conf`, 并在你的第一个 `image=` 语句之前加一行, 像:

```
append="mem=80M"
```

如果你的内存不是 80 兆, 调整上面这一行。要使得在 `/etc/lilo.conf` 文件里的任何改变生效, 你必须*重新运行程序:

```
lilo
```

(观察是否运行没有错误) 并重新启动。重启后你能使用这两个命令之一来检查你的调整是否有效:

```
cat /proc/meminfo
free
```

出于测试的目的或者如果程序运行有问题, 在 LIL0 提示符下指定内存大小的选择是有用的:

```
[在启动时在 LIL0 提示符下输入] linux "mem=16M"
```

偶然，我听到一个建议。如果启用你的所有内存发生问题或机器锁住，你可以跳过上层的几兆字节。例如仅仅启用你的 80MB 中的 78MB。很明显这是对于一些使用主存的大部分上层空间的 SCSI 控制器的例子。取其所值。

偶尔在一些系统中，Linux 仅能识别 16 兆字节的内存。通常这与在 BIOS 设置里启用了 "memory hole at 15-16 MB" 设置有关。（解决办法是禁用这 BIOS 设置）。总之在你的 BIOS 设置禁用所有的“高级”特性可能是个好主意。（例如，BIOS 病毒监测似乎是问题的常见来源）。

带有不同时间脉冲的内存条也能导致内存识别的问题或系统崩溃。（解决办法是替换内存条使所有的内存条时间脉冲一致）。

4.1.2 LILO 仅显示 LI (或 LIL) 并死机

以下引自我的那本很不错的手册，由 SAMS 出版社出版，Kamran Hussain, Timothy Parker 等人撰写的 "Red Hat Linux Unleashed"：

“当 LILO 装载时，显示单词 LILO。在每执行某一特定的步骤之前或之后，显示一个字母。如果 LILO 在某个步骤失败，使用目前已显示出的字母来判定问题所在。 [...]

LI [...] 这或者是由于硬盘几何参数不匹配或者由于 /etc/lilo/boot.b 被移动后没有重新运行映射 (map) 安装程序而引起。

LIL [...] 典型原因是介质出错或硬盘几何参数不匹配。”

这个几何参数是指用于你的 BIOS 的硬盘配置的扇区/磁头/柱面的数量。希望这对你有帮助。

拥有一本 Linux 手册或至少有一本普通的 UNIX 手册是个好主意。Windows 手册是毫无用处的而 Linux 手册是很有用的。"Red Hat Linux Unleashed" 是一本很好的手册但我相信有很多其他的手册也一样好用。

像上面的 LILO 错误出现时，你能够使用一个 Linux 或 DOS 引导盘来启动你的机器。基于错误的不同，似乎有几种常见的可能来纠正这样的 LILO 错误。

1. 如果 LILO 崩溃（这似乎不常见），你能移开并重新安装它。在 Linux 下你能通过运行以下命令来移开 LILO：

```
lilo -u /dev/had
```

或，在 DOS 下：

```
FDISK/MBR
```

这重写了 LILO 所在的硬盘主引导记录 (MBR)，并用“干净的”[无病毒]DOS 信息来替换它。在清除了 LILO 后如果你重新启动你的计算机，你无法使用 Linux（如果这种情况发生，你能从软盘启动然后在 DOS MBR 的顶部重装 LILO）

为了重装 LILO，只要简单地重新运行命令 lilo（作为超级用户）。

2. 指定选项

linear

在你的/etc/lilo.conf 文件的顶部。对于大的硬盘(>8 GB) 这个选项特别有用。具体细节请看 man lilo.conf。这个 linear 选项是安全的—它不会影响一个正常工作的系统，所以你总能在任何情况下指定该选项。

你也可能想通过用 BIOS 访问你的硬盘的方式来做这件事。例如在你的 BIOS 里通过开/关 LBA (“线性或大块存取”) 模式来判断这是否有帮助。

你可以试着指定以下选项(用更大的硬盘或更新的 BIOS 来克服 1024 个柱面的限制是由帮助的)而不是“linear”选项：

lba32

这是新的选项所以它不会与由 RH6.1 或更低版本的 Linux 所提供的 LILO 一起工作。如果遇到这类的“LI”问题使用最新的 Mandrake 或 RedHat--硬盘越来越大，BIOS 制造商也设置了新的窍门来解决。

3. 观察你的 BIOS 设置，判定那指定的硬盘几何参数与你的硬盘如何不匹配。

在 Linux 下你能使用以下命令来显示硬盘几何参数(例如，你的第一个 IDE 硬盘)

```
hdparm -g /dev/hda
```

你能很容易在网上使用 Google 寻找你的硬盘型号来找到推荐的厂商给出的硬盘几何参数。

4. 放置 LILO 在另一分区(不同的硬盘)并使用 fdisk，是该分区成为可引导盘(若你的系统支持从另一个盘启动)。否则交换你的硬盘，让由老的 BIOS 更好地支持的那个盘成为在你的第一个 IDE 接口(DOS 驱动盘“C”)上的第一个硬盘

5. 可能你更喜欢用“loadlin”而不是“lilo”。在 DOS 下,从你的 CDRom 察看程序 /dosutils/loadlin.exe。它从 DOS 引导 Linux。

6. 使用 GRUB 代替 LILO。Mandrake 7.2 包含了 GRUB 作为缺省地引导载入程序。

4.1.3 我如何能让操作系统变为由 LILO 缺省引导？

这可以在 lilo 配置文件/etc/lilo.conf 里设置。我的设置(lilo 版本为 0.21.5.1-4MDK)看起来像这样：

```
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
vga=normal
default=linux
keytable=/boot/us.klt
```

```
lba32
prompt
timeout=50
message=/boot/message
menu-scheme=wb:bw:wb:bw
image=/boot/vmlinuz
    label=linux
    root=/dev/hda3
    append=" mem=96M"
    read-only
image=/boot/vmlinuz
    label=failsafe
    root=/dev/hda3
    append=" mem=96M failsafe"
    read-only
other=/dev/hda1
    label=windows
    table=/dev/hda
other=/dev/fd0
    label=floppy
unsafe
```

这四个 "label=" 值定义了引导选择的名称。在文件的顶部由选项 "default=" 值所指定的是操作系统缺省引导项。没有这个 "default" 值, 出现在文件 /etc/lilo.conf 里的第一个标签值是缺省引导项。

在对 /etc/lilo.conf 文件作了任何改变之后, 别忘了重新运行命令:

```
lilo
```

也可以在 GUI (图形用户界面) 下配置 lilo。例如, 试着在一个 X 终端, 作为超级用户, 输入:

```
klilo &
```

4.1.4 在启动时 LIL0 提示符在屏幕上停留时间太短 (或太长)

增加或调整这一行:

```
delay=100
```

在你的 /etc/lilo.conf 文件第一个 "image=" 或 "append=" 语句之前的右边。(lilo 的更新的版本使用 "timeout" 选项来替代。) 这个数字是以十分之一秒 (0.1 秒) 为单位的延迟时间, 所以以上例子延迟时间是十秒。在对 /etc/lilo.conf 文件作了任何改变之后, 别忘了重新运行 lilo, 否则你的改变不起作用。

4.1.5 卸载 Linux

如果你真的想“卸载”Linux，你可以运行以下两个命令（在 DOS 或 MS Windows 下）：

```
LOCK C:
FDISK/MBR
```

这摆脱了 LIL0—它重写了你的 LIL0 所处的第一个硬盘的主引导记录。“lock”命令允许“raw”写到盘上，在更新的 DOS 版本，正常情况下作为反病毒措施这是不允许的。采用 FDISK/MBR 的问题是对于任何的成功或失败都不会返回记录报告。所以最好与“lock”命令一起使用。之后你能使用 DOS “FDISK”移开 Linux 分区，重新分配硬盘空间。

显然，MS FDISK 并不总是能够处理删除 Linux 分区的问题。此时，我可以使用 linux fdisk。最简单的方法可能是从 Linux 安装软盘/CD 中引导，在安装时使用 Linux 分区工具在它弹出时删除这个分区。之后我中断安装，Linux 消失。

如果你还有问题，这里是对于 MBR 置零的最终解决方案(下面文本来自：

<http://www.linuxgazette.com/issue63/okopnik.html>，编辑如下)：

```
* * * * *
```

注意：以下建议将完全清除包含了你的所有分区信息的主引导记录 (MBR)。不要这么做，除非你知道这确实是你想要的结果 - 这将使你的硬盘处于不可引导状态，与刚出厂时的效果一样，也就是，没有数据，要求分区和格式化。

```
* * * * *
```

基于 Linux 的解决方案 如果你能引导 Linux - 假设通过引导软盘 - 你只需调用 "dd":

```
dd if=/dev/zero of=/dev/hda bs=512 count=1
```

这让 MBR 全部置零。很明显，你得在根分区做这件事。

基于 DOS 的解决方案 用带有 "debug" 的 DOS 软盘引导；运行

```
debug
```

在 '-' 提示符下，用 0 "块填充" 一个 512-字节的内存块。

```
f 9000:0 200 0
```

用 'a' 命令启动汇编模式，并输入以下代码：

```
mov dx,9000
```

```
mov es,dx
```

```
xor bx,bx
```

```
mov cx,0001
```

```
mov dx,0080
```

```
mov ax,0301
```

```
int 13
```

```
int 20
```

按下 <Enter> 退出汇编模式，深呼吸 - 按下 "g" 执行，然后按 "q" 退出 "debug"。你的硬盘现在处于原始状态，并可再用于分区和安装。

4.2 访问我的驱动器

4.2.1 我的驱动器在哪里？

Linux 显示所有的目录都在一个目录树下，而与他们位于哪一个驱动器/硬件无关。通常，这种解决方案比传统的 DOS/Windows 模式要好—它完全从基本的硬件中抽象化了文件系统。如果你得不

断地重新安排，扩大你的硬件或增加网络资源你就喜欢这种方式。但是对于那些习惯于使用 DOS 方法来处理驱动器的用户，这变得有点复杂。

很简单，不要去寻找驱动器的字母。在 Linux 下没有这样的字母。在你的单一的 Linux 文件系统（目录树）下你的磁盘内容是作为子目录出现的。缺省地，可移动介质的内容是不会自动出现在这些子目录里——你必须“挂载”你的驱动器。具体参见下一个问题的回答。在你移开那个介质之前你也应先卸下那个驱动器。

在 Linux 下你能访问（读和写）各种驱动器和文件系统。这包括了本地 Linux 分区，DOS 和 MS Windows 分区（在硬盘或软盘），ZIP 和 Jazz 磁盘驱动器，和 CDROM 光盘。它也支持很多不常用的文件系统类型。这意味着你能使用针对 Windows 界面的 Netscape 来下载 Linux 软件，保存下载软件到你的 MS Windows 硬盘分区，然后启动 Linux，从你的硬盘的 Windows 分区复制这个下载的软件到 Linux 分区，最后在 Linux 下安装这个软件。

4.2.2 我如何访问我的 CDROM ？

挂载它。挂载你的 CD 的所有目录和文件到你的 Linux 目录树，所以无须驱动器的字母，你能很容易访问他们。作为超级用户，你能用以下命令挂载 CDROM：

```
mount -t auto /dev/cdrom /mnt/cdrom
```

如果这条命令工作，你的 CD 的内容出现在目录/mnt/cdrom

可能这条命令不会立刻产生作用——你可以定制它。这里解释了它是如何工作的。

这条命令告诉操作系统要挂载能自动监测文件系统类型("-t auto")的一个文件系统。设备是 /dev/cdrom。安装点（是指需挂载的文件系统安装时所处的目录）是/mnt/cdrom。这个目录必须存在并且为空。如果不存在，用以下命令创建它：

```
mkdir /mnt/cdrom
```

若挂载命令失败，可能在你的系统中不存在设备/dev/cdrom？试试：

```
ls -l /dev/cdrom
```

/dev/cdrom 仅仅是方便的软连接，连到映射在你的硬件上的真正的“设备”。在一个 IDE 系统中，你的真正的 CDROM 可能位于/dev/hdb 上。所以试着在上面的 mount 命令用/dev/hdb 来代替 /dev/cdrom：

```
mount -t auto /dev/hdb /mnt/cdrom
```

如果这失败了，你能试着用/dev/hdc 或 /dev/hdd，如果你的 CD 是位于第二个 IDE 接口上的一个 IDE CDROM。如果你的 CDROM 都不是这些，可能你没有 IDE 而是 SCSI CDROM，那么试试 /dev/sda, /dev/sr0, 等。

可能存在的驱动器的简要清单如下：

```
hda -- 在第一个 IDE 接口的主驱动器 (总是位于第一个硬盘)
hdb -- 在第一个 IDE 接口的从驱动器 (你必须有至少二个硬盘)
```

```
hdc -- 在第二个 IDE 接口的主驱动器 (如果你的计算机有二个 IDE 接口, 大多数新的计算机都有)
hdd -- 在第二个 IDE 接口的从驱动器 (若有一个)
sda -- 第一个 SCSI 驱动器
sdb -- 第二个 SCSI 驱动器 ("sdc"是第三个 SCSI 驱动器, 等等. 在一个系统中能有很多 scsi 驱动器)
sr0 -- 第一个 SCSI CD 驱动器 (有时称为 scd0)
sr1 -- 第二个 SCSI CD 驱动器 (有时称为 scd1), (sr2 是第三个 SCSI CD 驱动器, 等等. 在一个系统中能有很多 scsi CD 驱动器).
```

总之有一个设备 `/dev/cdrom` 总是好的。因为一些程序假定它存在。如果它在你的系统中不存在，你可以用以下命令来创建它作为软连接，例如：

```
ln -s /dev/hdb /dev/cdrom
```

(以上假设你的 `cdrom` 是 `/dev/hdb` 驱动器)

如果你不能挂载是因为 "the device is already mounted or directory busy" (该设备已挂上或目录忙)，可能挂载点 `/mnt/cdrom` 是你的当前工作目录。你得改变当前工作目录到其他地方以便能挂载它，例如，改变当前目录到根目录，可使用这个命令：

```
cd /
```

为了卸下已挂载的 CD，退出目录 `/mnt/cdrom` 并作为超级用户输入：

```
umount /mnt/cdrom
```

如果你的 CDROM 没有卸载，它可以拒绝弹出这个介质。而且，如果前一个 CD 没被卸载，当你挂载下一个时可能会遇到问题。如果你因为 "the device is busy" (设备忙碌) 而不能卸载，可能 `/mnt/cdrom` (或任何在该目录下的子目录) 是你的当前目录？你必须改变你的当前目录到任何其他地方而不是挂载点，以便卸载该设备。

4.2.3 如何挂载软盘，zip 磁盘驱动器，dos 分区，或网络驱动器？

和 CDROM 大致一样--- 如果你不读这一节，看前面的回答。

软盘

我能挂载我的软盘 (作为超级用户)，用命令：

```
mount -t auto /dev/fd0 /mnt/floppy
```

此外，保证目录 `/mnt/floppy` 必须存在并且为空。而且，`/mnt/floppy/` 不能是你的当前目录。

在成功挂载后，来自软盘的文件出现在目录 `/mnt/floppy/`。所有的用户能够读这些文件，但只有在根目录能够改变/删除这些文件。如果你想让用户能将文件写到软盘请更深入地阅读相关帮助。

为了卸载软盘，(你必须在弹出盘时作这一工作)使用：

```
umount /mnt/floppy
```

如果你因为 "the device is busy" (设备忙碌) 而不能卸载, 可能 /mnt/floppy/ 是你的当前目录。输入以下命令退出 (例如) :

```
cd
```

该命令改变你当前目录到你的主目录。

Zip 磁盘驱动器

我用以下命令挂载并行端口外设 zip 磁盘驱动器 (scsi 模拟器) :

```
mount -t vfat /dev/sda4 /mnt/zipdrive
```

这里使用了 "-t vfat" 因为在 vfat 文件系统里预格式化了 zip 磁盘, 该文件系统是一种由长文件名支持的 MS Windows 文件系统。没有卸载它你不能弹出磁盘。此外, 目录必须存在并且为空。而且, 不能是你的当前工作目录 (见前面的解答)。

我能够使用以下命令挂载内部 IDE zip 磁盘驱动器:

```
mount -t vfat /dev/hdd4 /mnt/zipdrive
```

在我的系统中, 这是在第二个 IDE 接口的第二个驱动器上, 所以 "hdd"—如果在你的系统里需要的话用 "hdb" 或 "hdc" 替换它。

来自 Alvaro Reguly <alvaro@reguly.net> 的提示. "我有一台 ATAPI Zip 磁盘驱动器 (作为 ATAPI 软盘验证过) 所以为了使它和 Debian 以及 kernel 2.4.3 一起工作我得把我的 BIOS 设置从 "Autodetect" 转到 "None" (当然只针对 Zip), 并挂载它。使用命令

```
mount -t vfat /dev/hdb  
(不带有末尾的 4!)"
```

除了 USB, Linux (1999 四月) 支持所有的 zip 磁盘驱动器 (内部 SCSI 和 IDE, 外部 SCSI 和并行端口)。如果在你的系统中不能自动装载 zip 磁盘驱动器的驱动程序, 详见本章关于如何手动装载 zip 磁盘驱动器的模块 (驱动程序)。

DOS/Windows 分区

在我的计算机里带有 Linux 和 MS Windows 二个操作系统, 我使用双启动系统。在使用以下命令挂载 DOS/Windows 分区后我能访问在此分区的文件:

```
mount -t vfat /dev/hda1 /mnt/dosdrive
```

此外, 基于你的 DOS 文件系统位于哪个分区, 你可以定制这条命令。"hda1" 意为第一个 IDE 硬盘 (hd a), 第一个分区 (1); "hda2" 是第一个 IDE 硬盘, 第二个分区; "hda3"--第一个 IDE 硬盘, 第三个分区; "hdb1"--第二个 IDE 硬盘, 第一个分区 (或仅仅 "hdb" 如果它是 CDROM 作为从驱动器安装在你的第一个 IDE 接口)。"hdc" 是第三个 IDE 驱动器, hdd 是第四个 IDE 驱动器。

SCSI 接口有相似的名字, 开始是字母"sd", 接着是显示 SCSI 接口的字母, 然后跟着显示 SCSI 设备 id 的数字。例如, sda4 是指 "第一个 SCSI 接口, id 号是 4"。

为了挂载后所有的用户能读和写, 你可以试着用命令:

```
mount -t vfat -o user,rw,exec,umask=000 /dev/hda1 /mnt/dosdrive
```

这条命令使用了选项 (-o user,rw,exec,umask=000), 给与每个人在你的 DOS /dev/hda1 分区里的所有文件全部的权限。(你应当问你自己 这对你的系统是否安全)。如果用户还不能写到 DOS 分区, 可能在你的挂载点的权限需要设置。这条命令(由根执行)将在挂载点/mnt/dosdrive 设置权限以便所有的用户都得到读, 写和执行的权力:

```
chmod a=rwx /mnt/dosdrive
```

网络文件系统 (NFS)

这主要是针对那些需直接访问位于另一 Linux 计算机的文件的系统。为了挂在远程文件系统如 NFS, 首先检查是否 NFS 服务已被激活(使用程序 setup)。NFS 也要求来自其它计算机的权限。为了在服务器上设定权限, 作为超级用户, 运行:

```
netconf
```

并在“输出文件系统”菜单下调整设置。

如果你更喜欢手动设定, 在文件/etc/exports 里设置权限。我的/etc/exports 的设置看起来像:

```
/usr hacker(ro) mars(ro)
/home hacker(rw) mars(rw)
/mnt hacker(rw) mars(rw)
```

这给与了二台名为 hacker 和 mars 的机器以权限来挂载目录/usr/ (只读), /home 和/mnt (读写)。

如果你适当设置了 NFS, 你现在能够使用下面的命令来挂载网络目录:

```
mount -t nfs mars:/home /mnt/mars_home
```

这挂载了一台称为"mars" 的机器的目录/home/ 的内容进入目录/mnt/mars_home (必须存在且为空)。

很多的操作系统可识别 NFS, 但 MS Windows 不会。所以 MS Windows 的远程共享需另外处理, 详见下一问题的解答。

4.2.4 如何通过 Samba 挂载远程 MS Windows 的文件系统?

通过 Samba 协议, 一个远程 MS Windows 文件系统能被挂载到 Linux 文件系统。

必须安装 Samba , 若未安装, 到这里下载 :

http://linux-newbie.sunsite.dk/lnag_drives.html#samba_setup#samba_setup

作为超级用户, 输入命令如下 :

```
smbmount //mars/windows /mnt/mars_windows -c marie
```

这挂载了来自名为 mars 的 MS Windows 机器的被称作 windows 的 MS Windows 资源。客户机的挂载点是 /mnt/mars_windows/ 。选项 "-c" 指定了 samba 服务器是名为 marie 的机器(这并不是必需的, 但它在我的系统里)。

要使上面的命令工作, 对于在 MS Windows 机器上将目录或驱动器作为资源共享, 必须赋予它们权限。要设置权限, 在 MS Windows 机器上, 使用"控制面板—网络" 激活文件共享, 然后启动“资源管理器”, 在要共享的目录或驱动器上点击鼠标右键, 点属性, 切换到"sharing", 给自己一个权限, 给这个资源一个名字

为了卸载一个 MS Windows 目录, 使用 :

```
smbumount /mnt/mars_windows
```

如果有问题, 请看 :

```
man smbmount
```

4.2.5 有快捷方式访问在 DOS/Windows 软盘上的文件吗?

使用"mtools", 无须挂载。例如, 我能使用 mdir 命令, 快速检查我的 DOS 软盘根目录的内容 :

```
mdir a:\
```

我也能使用 mcopy 将在软盘根目录下的文件"autoexec.bat" 复制到我的 Linux 的当前目录 :

```
mcopy a:\autoexec.bat
```

你必须是超级用户, 才能将文件写到软盘。

输入 "mtools"来查看丰富的 mtools 工具集合所支持的命令, 这些命令和最普遍的 DOS 命令相似(例如: mformat, mtype, mren, mmove, mdel, mrd, mattrib, ...), 如果你在使用中有什么问题, 使用手册, 例如:

```
man mtype
```

将看到在 DOS 分区如何显示一个文本文件的内容。

为了访问并非标有 a: 或 b: 的 DOS 驱动器, 你需配置 mtools 以显示与其它 DOS “驱动器字母” 相关的驱动器。这很容易—你只要编辑和改变文件 /etc/mtools.conf。典型地, 我使用 pico 来做这个工作(必须是超级用户):

```
pico /etc/mtools.conf
```

例如, 我的 /etc/mtools 文件里包含了象这样的一行:

```
drive c: file="/dev/hda1"
```

这一行指示 mtools 将 "/dev/hda1" 分区称为 "c:"。在 /etc/mtools.conf 里的设置仅要求去掉注释(在行的开头删去 "#") 并调整适当的入口。

4.2.6 作为超级用户我可以挂载。普通用户能做到吗?

你可以作为超级用户编辑这个文件 /etc/fstab, 给予普通用户权限来挂载一个特别的驱动器。例如, 我能用 pico 文本编辑器来做:

```
pico -w /etc/fstab
```

选项 ‘-w’ 关闭了对长文本行的限制。

这里是我的文件 /etc/fstab 的内容:

```
/dev/hda2 / ext2 defaults 1 1
/dev/hdc3 /home ext2 defaults 1 2
/dev/hdc2 /usr ext2 defaults 1 2
/dev/hdc4 swap swap defaults 0 0
/dev/fd0 /mnt/floppy auto noauto,users,rw 0 0
/dev/cdrom /mnt/cdrom auto noauto,user,ro 0 0
/dev/sda4 /mnt/zipdrive vfat noauto,user,rw,exec 0 0
/dev/hda1 /mnt/dosdrive vfat noauto,user,rw 0 0
none /proc proc defaults 0 0
hacker:/mnt/cdrom /mnt/hacker_cdrom nfs noauto,user,ro 0 0
hacker:/mnt/floppy /mnt/hacker_floppy nfs noauto,user,rw 0 0
hacker:/home /mnt/hacker_home nfs noauto,user,rw 0 0
hacker:/usr /mnt/hacker_usr nfs noauto,user,rw 0 0
```

每一行包含了六个由空格分割的域(这意味着每行有六个由空格分开的输入点)。

第一个是设备的名字。

第二个是挂载点(指你的 Linux 系统中一个存在的目录指向挂载的资源)。

第三个是文件系统类型。对于可能包含几种类型文件系统的可移动介质, 我使用 "auto" 选项, 让 Linux 检测当前展现于此的是哪一种文件系统。(检测文件系统的顺序是由

/etc/filesystems 文件的内容决定。你可能想证实是否在 "msdos" 之前它指定了 "vfat" 还是缩略了 DOS 的长文件名)。

第四个域包含了选项: "auto" = 系统启动时挂载文件系统; "rw" = 可读写; "ro" = 只读, "user" = 用户有可挂载该文件系统的权限(也可以使用 "users" 允许一个用户挂载而另一用户卸载— 否则只有挂载该文件系统的用户才能卸载), "exec" 允许该文件系统的程序运行。

第五个域的数字指定是否在系统备份时也备份该文件系统,

第六个域的数字决定是否在引导时检查文件系统的完整性。在我的 /etc/fstab 里 hacker 是在我的家庭网络的另一台计算机(叫作"hacker") 的文件系统。在此作为如何挂载网络资源的例子。用 man fstab 来获得更多信息。

例如, 如果普通用户(非超级用户)拥有挂载 cdrom 的权限(指定 "user" 选项), 他们可以使用一个命令来挂载它:

```
mount /mnt/cdrom
```

超级用户用于挂载操作的命令对于普通用户不会有效, 因为在 /etc/fstab 文件里普通用户的权限通过某些选项而受到限制。所以他/她不能同时指定设备和挂载点。

对于一个能够对磁碟进行写操作或执行某一程序能力的普通用户, 必须在 "挂载点" 目录下赋予他/她适当的权限。例如, 在目录 /mnt/floppy 下给予所有用户全部的权限(读, 写, 执行)。

```
chmod a+rwx /mnt/floppy
```

现在(对于软盘, 在文件 /etc/fstab 里选定了 "rw" 选项) 用户能够对软盘进行写操作。如果在 /etc/fstab 文件里启用了 "exec" 选项, 用户也能从软盘中运行程序。

请注意, DOS vfat 文件系统并不知道 Linux 所给予的文件权限。在挂载时 Linux 管理这些权限是通过给予已挂载的文件系统缺省的文件权限: 挂载该文件系统的用户是所有文件的拥有者, 并被赋予对该文件系统写操作的权利(如果在 fstab 指定了 "rw")。但是其他用户只能读。如果你想改变这种设置, 你可以使用 "umask=" 选项, 在你的 /etc/fstab 文件里的相对应的文本行看起来象下面的这个例子:

```
/dev/sda4 /mnt/zipdrive vfat noauto,users,rw,exec,umask=000 0 0
```

在你的 zip 磁盘驱动器, 每个用户都被完全地赋予所有的权限(挂载, 卸载, 读, 写, 执行)。

总之, 这个 /etc/fstab 文件保存了你挂载文件系统时的默认值和允许用户访问的类型。你应当定制该文件来简化你的系统的挂载操作。Linux 默认的挂载配置出于安全的目的而给予了必要的限制, 当你在家安装 Linux 时可以删除一些限制。

4.2.7 挂载命令太长, 我如何用别名简化?

一个别名是一个更复杂或更常用命令的缩写。为了创建别名,我作为超级用户,编辑/etc/bashrc 文件。在系统中别名对于所有用户都是有效的(为了创建用户指定的别名,编辑在用户主目录里的 .bashrc 文件)我的/etc/bashrc 文件的相关部分如下:

```
alias cdrom="mount -v /mnt/cdrom"
alias ucdrom="umount -v /mnt/cdrom"
alias dosdrive="mount -v /mnt/dosdrive"
alias udosdrive="umount -v /mnt/dosdrive"
alias zipdrive="mount -v /mnt/zipdrive"
alias uzipdrive="umount -v /mnt/zipdrive"
alias floppy="mount -v /mnt/floppy"
alias ufloppy="umount -v /mnt/floppy"
```

"-v" 选项代表 "verbose", 即: 告诉 Linux 在挂载时告诉我更多信息。为了让别名生效,用户要重新登录。现在用户可以使用这一简单命令挂载软盘:

```
floppy
```

他/她可使用下面的命令卸载:

```
ufloppy
```

4.2.8 我能自动挂载吗?

是的,当你访问一个文件系统时你能够自动挂载它,而当你停止使用时卸载。这和你在 MS Windows 下所做的很相似。但是,如果你在 DOS 或 Windows 广泛使用可移动介质,你一定已经注意到了自动挂载并非完全安全。

在 Linux 下挂载有二种方式,分别称为 "超级挂载" ("supermount")和 "自动挂载" ("automount")。

超级挂载. Mandrake 7.2 给你使用"supermount" 的选项作为设置选项。所以对于"supermount" 的最简单的方法是安装最新的 Mandrake 并选择该选项。在运行 Mandrake 的计算机里我的 /etc/fstab 文件包含了下列各行:

```
/dev/hda3 / ext2 defaults 1 1
none /dev/pts devpts mode=0620 0 0
/dev/hda4 /home ext2 defaults 1 2
/mnt/cdrom /mnt/cdrom supermount fs=iso9660,dev=/dev/cdrom 0 0
/mnt/floppy /mnt/floppy supermount fs=vfat,dev=/dev/fd0 0 0
/mnt/zip /mnt/zip supermount fs=vfat,dev=/dev/zip 0 0
none /proc proc defaults 0 0
/dev/hdb2 /usr ext2 defaults 1 2
/dev/hdb5 swap swap defaults 0 0
```

在以上例子中,你可能注意到了我选择超级挂载入 3 个文件系统: cdrom, floppy 和 zipdrive。我能手动编辑文件/etc/fstab (比如,使用 pico) 或使用 supermount 命令来定制超级挂载以满足我的需要。

自动挂载。为了配置"自动挂载",我先运行 ntsysv 程序(在作为超级用户条件下)并保证启用自动挂载服务("autofs")。

然后, 我通过编辑/etc/auto.master 和 /etc/auto.misc 文件来配置自动挂载。 例如 (作为超级用户):

```
pico /etc/auto.master
```

我的 /etc/auto.master 文件就象:

```
/misc /etc/auto.misc --timeout 1
```

这假定在目录/misc (必须存在且为空)里要挂载我的需自动挂载的设备。 在我停止使用这些设备后一秒 (例如, 在我退出该目录后), 我的自动挂载设备将自动卸载。这是很短的时间—你可以选择更长一些。详细的配置文件是/etc/auto.misc 。这是我的文件内容:

```
kernel -ro,soft,intr ftp.kernel.org:/pub/linux
cdrom -fstype=auto,ro :/dev/cdrom
floppy -fstype=auto,rw :/dev/fd0
zipdrive -fstype=vfat,rw :/dev/sda4
dosdrive -fstype=vfat,ro :/dev/hda1
hacker_cdrom -fstype=nfs,ro hacker:/mnt/cdrom
hacker_floppy -fstype=nfs,rw hacker:/mnt/floppy
hacker_usr -fstype=nfs,ro hacker:/usr
```

每行包含了3个由空格分隔的域。第一个域是"关键点", 它是(在 /misc 之下的)要挂载的设备所处的子目录的名字。 这个目录不应当存在, 当我使用 ls 命令时该目录不可见,但我能用 "cd"命令进入该目录, 然后我能挂载该设备。 不要问我为什么这样和如何在 GUI 里使用这种的自动挂载。我不知道。 在我的 auto.misc 文件里与 hacker 相关的是在我的家庭网络的另一台计算机里的 cdrom 和软盘。

我自动挂载到/misc 目录 (不是 /mnt), 因而我也能不使用自动挂载而用手动挂载文件系统到/mnt 目录。

4.2.9 我如何得到已识别的在并行端口(外部)的 zip 磁盘驱动器?

RedHat 6.0 and 6.1

在我升级到 RedHat 6.0 时,zip 磁盘驱动器 (zip100 磁盘驱动器)安装后不工作。(安装程序被告知不能发现 zip 磁盘驱动器)。 所以,在安装完成后,我(作为超级用户)发出以下命令, 将并行端口 zip 磁盘驱动器的模块插入到内核:

```
/sbin/insmod parport
/sbin/insmod ppa
```

为了让这二行在每次启动后自动执行, 我将它们加入到/etc/rc.d/rc.local 文件的末尾(该文件就象 DOS 里的 autoexec.bat 文件)。

如果做这样的设置后还不工作,你也可以编辑/etc/conf.modules 文件。我的文件里包含了这样一行:

```
alias parport_lowlevel parport_pc
```

并且没有提及"ppa"模块的行。

对于更新版本的 Zip250 磁盘驱动器, 我从/etc/rc.d/rc.local 文件中执行以下二行:

```
/sbin/insmod parport  
/sbin/insmod imm
```

4.2.10 我能设置 32 位硬盘的 I/O?

更新版的 Linux (如 Mandrake 7.0) 能自动开启硬盘优化程序: 32-位输入/输出(I/O)和直接内存存取(DMA)。 这里是如何手动开启硬盘优化程序的文档。(基于由 Edward Choh 所著的文档, 参见 <http://hardwarezone.community.com.sg/main.htm>.)

这个程序能让我的机器很好地工作, 但要警告你它可能损伤你的硬盘的内容。 所以如果你真的是一个新手, 不要做这件事。 现在不要感到厌烦或心虚—我不能保证它能很好地为你工作。

开启 32-位 I/O 和 DMA 要由超级用户来做。我是在单一用户模式内做的(这样如果有些错误发生, 我重新启动可使文件系统的损伤减到最小。)。 我确实不会在当前还在运行很多系统或 X-windows 时在其中的某一系统中做这样的事, 并且会对任何宝贵数据进行备份。

要使你的计算机能在单一用户模式下导入, (在导入过程中)在 lilo 提示符下我输入:

```
linux single
```

假设, 我想在我的第一个 IDE 硬盘-- "hda"启用 32 位 I/O。 首先, 我测试该硬盘当前的性能并记录该结果:

```
hdparm -t /dev/hda
```

现在我显示当前 I/O 和 DMA 的设置:

```
hdparm -c /dev/hda
```

[我的系统显示 0, 意味着 32 位 I/O 被关闭, 默认使用 16 位存取]

```
hdparm -d /dev/hda
```

[我的系统再次显示 0, 意味着硬盘 DMA 存取被关闭]

现在我开启 32 位 I/O 和 DMA:

```
hdparm -c 1 /dev/hda  
hdparm -d 1 /dev/hda
```

现在我再测试该硬盘的性能并与原先的做一比较:

```
hdparm -t /dev/hda
```

如果正常工作且性能提高, 我就能够 ” 提交 ” 这个新的设置, 所以这些设置能使一个软件再生。

```
hdparm -k 1 /dev/hda
```

为了让你在每次启动机器时都能使这新设置产生作用, 你可以在 `/etc/rc.d/rc.local` 文件的末尾增加一行(该文件与 DOS 的 `AUTOEXEC.BAT` 类似):

```
hdparm -c 1 -d 1 -k 1 /dev/had
```

如果某些工作不正常, 或性能并未真正提高, 只要我不执行最后那个操作, 我在任何时候启动总是旧的设置发生作用。

在我的家庭网络我实现四个硬盘之间的转换。在 3 个新硬盘之间转换是成功的: 性能提高了 30-300% , 至少一个硬盘 ” 感觉 ” 比以前更 快。 有一个硬盘(很古怪)在性能测试时总让计算机死机, 我每次都得重新启动机器(不损伤数据)。

4.2.11 我所打开的文件数已达到限量(错误信息)

你能通过 `/proc` 文件系统增加该限量。 这个文件系统完全是虚拟的—它只是用于看或设置 Linux 内核的某一部分的 ” 窗口 ” 。 为了能读到我的系统中已同时打开的文件的最大数量, 我使用以下命令:

```
cat /proc/sys/fs/file-max
```

在我的系统中(Mandrake 7.2), 该限量为 8192。为了增加它, 我(作为超级用户)使用:

```
echo 16000 > /proc/sys/fs/file-max
```

你也可以在相关的内核变量中增加该限量:

```
echo 30000 > /proc/sys/fs/inode-max
```

为了使这一改变永久保存, 把上一行加到你的启动脚本文件 `/etc/rc.d/rc.local` 的末尾。你可以用以下命令行来更深入了解 `/proc` Linux 内核界面, 它所包含的变量的意义和推荐值。(如果你已安装了 Linux 内核源代码, 即使对新手也是很大的资源):

```
less /usr/src/linux/Documentation/proc.txt
```

或 (在 RedHat 8.0 下)

```
man proc
```

4.2.12 我加了一个新硬盘, 开始使用时要做什么?

0. 计划新硬盘文件系统的布局

1. 给新硬盘分区
2. 将新分区格式化
3. 测试新的空间
4. 复制旧分区的数据到新的分区(选项)
5. 编辑 /etc/fstab
6. 重新启动
7. 清除旧数据(选项)

这里是我的方案的长版本。

0. 计划新硬盘文件系统的布局

你喜欢在哪里使用这个新的空间? 输入:

```
df
```

概括显示每一个存在的已挂载分区的闲置/已用空间的使用情况。 输入:

```
du
```

发现所选目录的大小。

例如, 我考虑在以下挂载点之一使用新硬盘空间。

```
/usr/local  
/home  
/home/share/downloads  
/usr/local/mp3s  
/usr/local/dos_data
```

以下解释了原因。

/usr/local 被认为可存放 Linux 的任何升级。最好在一个独立分区放置该目录, 因为即使我重新格式化其它分区也不影响在/usr/local 目录下所储存的内容。 我确信它是 "ext2" 类型或可能是 "reiserfs" 类型。

/home 包含了用户数据。确实地, 它是最值得关心的数据。很明显, 该目录被认为可存放 Linux 的任何升级。哇, 明显地, 让它在独立分区。 通常其类型是 "ext2" 或 "reiserfs"。
/usr/local/mp3s 是非标准 Linux 目录。我可以在此保存我的 MP3 (音乐) 文件。 该目录会变得很大。

/usr/local/dos_data。另一个非标准 目录。如果我使用双重启动系统, 我会考虑做一个额外的 "DOS FAT32" 类型或相似类型的分区以便于在 MS Windows 和 Linux 之间透明地共享文件 (二种途径)。 我将配置所有基于 Windows 的程序来使用这个 "驱动器" 使之成为存放由用户产生的所有文件的分区。 在该位置, 我还建了 "mp3s", "cds" 和其它这样的目录。这个方法的严重缺陷--在重新安装时 MS Windows 可能使该分区变得杂乱无章。

1. 给新硬盘分区。

例如, 如果我的新硬盘是在第二个 IDE 接口的从驱动器(可能是第四个 IDE 驱动器), 我能使用:

```
cfdisk /dev/hdd
```

或更老的(且标准的)工具:

```
fdisk /dev/hdd
```

如果你的驱动器不是 "hdd", 按实际驱动器调整以上命令。

hda -- 第一个 IDE 接口的主驱动器 (整个磁盘)

hdb -- 第一个 IDE 接口的从驱动器

hdc -- 第二个 IDE 接口的主驱动器

hdd -- 第二个 IDE 接口的从驱动器

sda -- 第一个 SCSI 驱动器(整个磁盘)

sdb -- 第二个 SCSI 驱动器(整个磁盘)

...

sdp -- 第十六个 SCSI 驱动器(整个磁盘)

对于其它磁盘, 参考 /usr/src/Linux/Documentation/devices.txt.

绝大多数情况下, 我让我的 Linux 分区是 ext2 类型("Linux").

分区有些技巧—如果你从未做过, 读 man fdisk 和 man cfdisk. 删除一个分区及其中的所有数据很容易. 确信你知道你正使用哪一个磁盘.

直到我写入新的分区布局, fdisk 或 cfdisk 才会对硬盘做出改变. 所以如果我出错, 这个新布局不会写入就退出了. 仅当所有步骤完全结束, 新布局才写入.

2. 格式化每个分区. 例如, 为了格式化第一个分区, 同时检查损伤的区块(-c), 我输入:

```
mkfs -c -t ext2 /dev/hdd1
```

3. 测试新分区. 手动挂载新分区(挂载目录必须存在且为空). 复制一些文件到每个分区. 随机查看/编辑一组文件. 然后全部删除.

4. 复制数据. 选项 - 仅当你想把旧分区的数据移到新分区. 进入单用户模式 (init 1). 手动挂载新分区. 复制旧分区的数据到新分区. 小心旧数据. 你可能不想在出错时失去, 所以我还不想删除它—我重新命名了顶层目录. 例如:

```
cp -R /usr/local/ /mnt/hdd1/  
mv /usr/local/ /usr/local.old.backup_of_2001-04-21
```

5. 编辑 /etc/fstab 文件. 修改该文件以反映你的新文件系统的布局. 很可能, 给新分区插入挂载点或改变任何旧的挂载点. 例如, 如果从 /usr/local 所在分区移开该目录, 我需要增加下面一行:

```
/dev/hdd1 /usr/local ext2 defaults 1 2
```

6. 重新启动并测试. 重新启动是为了卸载旧的, 挂载新的挂载点. 例如:

```
umount /usr/local  
mount -a
```

但是重新启动可能会对新布局进行更严格的测试.

7. 清除旧数据. 几天后, 当我确信每一部分都能正常工作, 我删除旧的文件(如:

/usr/local.old.backup_of_2001-04-21 目录)。

4.2.13 交换分区 SWAP

交换分区是计算机物理内存的扩展。很可能, 你在 RedHat 初始设置时创建了一个交换分区。你能使用以下命令行证实你的系统中可用交换空间的大小:

```
cat /proc/meminfo
```

通常推荐一台计算机应当有: 至少 4 MB 交换空间, 对于一个仅运行命令行的系统需至少 32 MB 总内存(物理+交换), 对于一个运行 X-windows 的系统至少 64 MB 总内存(物理+交换), 并且在系统中交换空间至少是物理内存的 1.5 倍。

如果这太复杂, 你可以让交换空间是你的物理内存的 2 倍, 但不少于 64 MB。
如果你需要改变你的交换分区大小, 这里是一些基本方法。

4.2.13.1 交换分区

你能够拥有几个交换分区。[旧的 Linux 内核限制了每个交换分区的大小, 大致可达到 124 MB, 但 Linux kernels 2.2.x 以上版本取消了该限制] 这里是创建和启用一个交换分区的步骤:

使用 fdisk 创建一个适当大小的分区 (分区类型 82, "Linux swap")。

格式化分区, 检查损坏的区块, 例如:

```
mkswap -c /dev/hda4
```

你要用你自己的分区名字代替/dev/hda4。 因为我没有指定分区大小, 系统自动监测。
启用交换分区, 例如:

```
swapon /dev/hda4
```

为了让该交换分区在启动时能自动启用, 你得在/etc/fstab 文件中包含适当的入口, 例如
/etc/fstab, for example:

```
/dev/hda4 swap swap defaults 0 0
```

如果你需要禁用该交换分区, 你(作为超级用户)使用:

```
swapoff /dev/hda4
```

4.2.13.2 交换文件

交换文件通常比交换分区慢,所以这不是永久推荐的交换方法。可是,如果你临时需要更多的交换空间,创建一个交换文件能被快速安装。你能有最多可达 8 个的交换文件,每个文件最大可达 16 MB。这里是产生交换文件的步骤:

- 创建一个指明了大小的交换文件:

```
dd if=/dev/zero of=/swapfile bs=1024 count=8192
```

这物理上创建了交换文件 /swapfile,块大小 1024 字节,文件包含 8192 个块,总大小大约 8 MB。 [dd 命令复制了文件。在以上例子中,输入文件 (if)是 /dev/zero,输出文件 (of)是 /swapfile。你不能使用 cp (copy)命令来复制文件,因为交换文件在硬盘上是物理连续的]。

- 用下面的命令设置文件:

```
mkswap /swapfile 8192
```

- 发出下面命令把缓存区的内容强制写入硬盘:

```
sync
```

- 启用交换文件用:

```
swapon /swapfile
```

当你用完交换文件,你关闭并移开它

```
swapoff /swapfile  
rm /swapfile
```

你也可以参看由 Linus Torvalds 本人写的资料:

```
man mkswap
```

4.3 使用 X-Windows 工作

4.3.1 怎样切换文本和图形控制台？

任何时候，按下组合键<CTRL><ALT><F1>系统将切换到第一文本控制台，<CTRL><ALT><F2>系统切换到第二文本控制台，<CTRL><ALT><F3>将会使系统切换到第三文本控制台，等等，一直到<CTRL><ALT><F6>，总共可以切换六个文本控制台。当有图形用户界面（以下简称 GUI，Graphical User Interface）控制台运行时，按下组合键<CTRL><ALT><F7>将会切换到第一 GUI 控制台，<CTRL><ALT><F8>会切换到第二 GUI 控制台，如此类推，直到<CTRL><ALT><F11>，共计五个 GUI 控制台。第十二个控制台即可以作为第六 GUI 控制台（RedHat 6.1 版本中）亦可以作为显示内核信息的界面（Mandrake 7.0 版本中）。一般情况下，没有或者只有一个 GUI 控制台运行。

<CTRL><ALT><F1>表示：“按下<CTRL>键和<ALT>键，然后按下<F1>键；释放<F1>键，然后再释放<CTRL>键和<ALT>键。”

因此，当使用一台 Linux 系统的计算机时，可以有多个控制台同时运行，我们可以通过以上描述的热键在多个控制台之间切换。使用 Linux 系统，必须对每一个控制台登录，- 可以作为同一个用户多次登录，也可以作为不同用户登录 - ，每一个登陆的对话都是独立的，它们之间应该是互不影响的（除非你两次作为同一个用户登录）。第一到第六个控制台是基于文本方式的，我们称之为 command - line terminal（行命令界面）或者称为 tty1, tty2, ... tty6（惯用语，tty 来源于 teletypewriter）。第七个及以后的控制台都是图形界面（GUI）控制台，所有的 GUI 控制台都是本地控制台，- 所以基于 Linux 系统的计算机都是真正意义上的多用户多任务系统。

我们可以通过网络可以远程联络 Linux 计算机。当计算机联网时，我们可以在服务器端远程运行一个程序，将显示结果发送到本地终端控制台上，此时本地的终端可以是基于 Linux 系统的控制台，也可以是基于非 Linux 系统的控制台。当一个程序不能运行于 MS Windows 服务器，而能运行于 Linux 服务器时，这种本地 / 远程运行方式对我们来说是十分有利的。

一个全屏文本控制台可以自我结束，而在图形（GUI）模式下我们也可以根据需要创建一个伪终止，我们称之为 pts0, pts1, . . .。

在一个文本终端上我们可以使用以下命令来为自己的终端命名：

```
tty
```

我可以使用以下的命令来决定我的计算机上正在本地使用哪个阶段操作。

```
uname -a
```

uname 这个命令的名字起源于“UNIX name”。这个命令将显示出操作系统的名字，服务器的名字，操作系统核心的版本号，以及内核的编辑时间。

可以在文本方式下输入“exit”来关闭任何文本终端。

Linux 的 GUI 控制台和你使用过的其他 GUI 界面非常相似，例如，MS Windows，但是，他们还有很多的的不同点。我很喜欢用多“桌面”方式来管理应用程序。KDE 默认提供 4 个桌面（这些桌面可以被自定义为 1 到 16）。我运行很多应用程序，从来不用关闭它们，所以应用这种管理方式我可

以在桌面 2 中保持一个因特网连接应用 (如 Netscape, knode newsreader, 和 licq), 在桌面 3 中运行 abiword, staroffice 和一个文本编辑程序, 在桌面 4 中运行游戏以及征服者和它的帮助文档, 而让桌面 1 来运行一些偶然的事务。在同一个桌面中, 使用者可以通过点击 “ K-panel ” 的标签条或者使用 <CTRL><TAB> 组合键来切换不同的桌面。我们也可以通过点击应用程序窗口, 或者在 “ 应用程序平台 ” 中点击相应的图标, 亦或者使用组合键 <ALT><TAB> 来切换不同的应用程序。使用组合键 <CTRL><ESC> 将会显示一份当前的主机上运行的应用程序的列表。

4.3.2 我怎样在 X-server 上安装显示卡、监视器以及鼠标 ?

一般来说这些设备应该在你初始化安装时进行设置, 除非你跳过了这些步骤。如果现在安装这些设备, 也许你要尝试使用以下这些文本模式配置器中的一个来进行启动配置。

```
XConfigurator  
XF86setup  
Xconf
```

在 RedHat 中, 你也可以运行 setup, 并且从那里访问 XConfigurator。

在 Linux 下安装配置 X - Windows, 你也许需要了解你的计算机硬件。你也许需要找到你的监视器的手册, 查找你的监视器支持的最大同步扫描频率 (场扫描和行扫描)。计算机启动时将会提供给你的显卡类型及显存容量的信息。同时执行这些命令将会提供给你一些十分有用的信息。

```
lspci  
superprobe
```

阅读鼠标底座的标签, 你将会得到鼠标类型的信息。下次你购买鼠标时, 请购买三键 “ Linux - ready ” (Linux 兼容) Logitech (罗技) 或者近似的鼠标 (所有的三键鼠标都适用于 Linux)。

一个标准的鼠标总会给我们带来方便 - - 我从来不会购买一个模样奇怪的鼠标, 因为那意味着一个专用的驱动或者一个痛苦的安装或者设置过程。

当在测试 X - Windows 时, 如果屏幕出现空白, 奇怪的线或者明显的不可操作现象, 使用组合键 <CTRL><ALT><BkSpace> 快速关闭计算机, 重新检查你的监视器的同步频率。运行在过高的同步频率下会损坏你的监视器。

如果你真的有某些问题, 请首先安装一个标准的 VGA 模式 (640X480 像素, 16 位或者 256 位颜色)。你可以在明白你的系统如何工作后, 再做细调, 或者可以在 Linux 下寻找某些有效的配置工具的帮助。

在安装配置好 Linux 后, 你可以使用以下命令来手动启动 Linux。

```
Startx&
```

这个 “ & ” 将会使你的命令在后台中运行, 所以你的文本终端不会被阻断。你也可以使用 :

```
init 5
```

这个命令将会是你的系统切换到第五运行状态下，即“图形界面运行状态”。若要在系统重新启动时自动启动（用户选择），请阅读下几段内容。

4.3.3 我能有一个 GUI 登陆提示符吗？

需要在系统启动时自动启动 X - sever 并且显示一个图形登录提示符，你只需要修改 / etc / inittab 文件中的一个字符。这个文件对某些设置规定如下：

```
id : 3 : initdefault :
```

改为

```
id : 5 : initdefault :
```

这个设定将引导 linux 内核的第 5 运行模式，我们称之为 X - Windows。不同的运行模式定义在文件 / etc / inittab 解释：

- 0 - 挂起（不要将此项作为默认设定）
- 1 - 单用户模式
- 2 - 多用户模式，不带 NFS（与选项 3 相同，不支持网络）
- 3 - 标准的多用户模式
- 4 - 未定义
- 5 - X11
- 6 - 重新启动（不要将此项作为默认设定）

你可以使用行命令来更改启动方式，例如，这个命令（已经作为根命令执行）：

```
init 6
```

这个命令将重新启动你的计算机，当然下一个命令将会使你的计算机切换到单用户模式：

```
init 1
```

为了知道我当前处于什么运行模式，我使用命令 runlevel。

为了细调我的 Linux 登录屏幕的外观，我使用命令（在 Linux 下）：

```
kcontrol &
```

选择“系统” - “登陆管理器”。我喜欢的登陆屏幕是一个模拟的钟表，大字体，上一词的登录者名称已经输入，光标处于输入登录密码的对话框中。

4.3.4 我怎么安装 KDE？（例如，在 RedHat5.2）

这部分只是针对那些使用老版本感兴趣的用戶。

由于 RedHat5.2 发行时, KDE 使用的库文件的授权问题还没有解决, 所以 RedHat5.2 版没有默认安装 KDE。另外, KDE 是一个非常好的 GUI 界面, 现在这些授权问题已经解决, RedHat6.0 已经集成了 KDE (和精心设计的 Gnome 一起)。当然, KDE 的代码已经存在于 RedHat5.2 的光盘中, 你只是需要自己安装。(如果你没有安装光盘, 你可以从 KDE 的网站上下载 KDE 代码。)

首先执行以下命令来检查你的 X - sever 工作情况:

```
xinit
```

[X - sever 是 X - Windows 的精简, 没有“Windows 管理器”。你可以在这里执行你的 X - Windows 程序, 只要在 X - terminal 窗口输入程序名即可(在程序名前面输入 ./ 或者全路径)。但是在这里你不能进行诸如调整窗口的大小和添加光标等功能。]

你可以在 X - terminal 窗口中输入 exit 命令来关闭 X - sever, 或者按组合键 <CTRL><ALT><BkSpace> 来关闭 X - Windows 服务。(后一种方法会将你的机器挂起 - 在这种情况下不要重新启动计算机)。

如果你的 X - sever 无法工作, 请看下一个解决方法。

现在, 从根开始登录, 放入 RH5.2 光盘:

```
mount /mnt/cdrom
```

跳到以下目录:

```
cd /mnt/cdrom/kde/distribUtion/RedHat/i386/binar/
```

在将如长路径或者文件名时使用自动完成(按 TAB 键)是非常有用的。 阅读 README 文件:

```
cat readme-2rh51-rpms | more
```

使用“RedHat 包管理器”来安装一些必要的文件包:

```
rpm -iv packagename
```

这个包文件是以 .rpm 作为后缀。首先安装 qt 库文件, 然后是 KDE 支持文件, 再是 KDE 库文件, 最后是 KDE 基本文件。如果你的安装顺序错误, 从属关系检查将会出现错误, 安装包被停止安装 (rpm 将会发布一个错误信息)。没有关系, 重新安装需要的文件包, 并持续装下去。在安装好 KDE 基本文件后, 你也许要为 KDE 安装其他的文件包(应用程序, 管理, 网络, 游戏, 图形, 多媒体) - 他们都不很大, 所以你可以确认全部安装。最后确认你已经安装了全部的组件, 输入:

```
rpm -ivh *.rpm
```

vh 操作提供了一些额外的消息。你将会得到“文件包已经安装”的消息, 如果还有更多的消息你可以滚动浏览, 你可以使用组合键<Shift><pgup>来上翻。如果你不喜欢这种行命令形式的 rpm 包管理器, 你可以使用 GUI 前台来安装相同的包, 称之为 glint (仅 RH5.2 有效)。在 X - Windows 终端输入“glint”即可。

现在，在你的系统中 KDE 已经成为你的默认 X - Windows 管理器。在用户目录中将会建立一个.Xclients 文件：

```
pico .Xclients
```

这个文件里只包含一行内容：

```
/opt/kde/bin/startkde
```

输入此行然后保存文件（调整此行内容是必需的，以保证 startkde 文件的正确定位）。现在，是此文件对所有用户有效（可以执行）。

```
chmod a+x .Xclients
```

若许可被改变请检查：

```
ls -l .Xclients
```

[以一个点为开始的文件将不会被一般的 Ls 命令显示，就如 Dos 下的隐含文件，你必须使用他的名字或者 ls -a]

如果你生成的文件是根文件而不是用户文件，那么为正确的用户改变文件的拥有者和文件组：

```
chown user_name .Xclients
```

```
chgrp user_name .Xclients
```

现在，输入 startx 将会启动以 KDE 作为窗口管理器的 X - sever。

4.3.5 我怎么将默认桌面改变成 KDE（如 Gnome 以及其他类型的 KDE）？

在我的用户目录里，我使用我最喜爱的 pico 编辑器建立（如果存在，编辑）以下文件，.xsession：

```
pico .xsession
```

[请注意文件名前面的点，文件名前带点的文件是“不可视的”。] 在我的 RedHat6.2 系统中，这个文件包含一行：

```
exec .xsession
```

KDE 的确像是我们度身定做的，尽管对某些老的硬件来讲是一个很重的负担或者是过载。这是在我的 RedHat 或者 Mandrake 安装光盘中的视窗管理器列表。

startkde	（在某些系统中运行 KDE，命令也许就是 KDE）
gnome - session	（运行 Gnome）
xfce	（运行 Xfce，这是我最爱的“轻量级”桌面）
afterstep	（运行 afterstep）
anotherlevel	（运行 anotherlevel）

```
fvwm2          (运行 fvwm2)
fvwm           (运行 fvwm)
```

当然，这种交互的视窗管理器只会在你安装到你的系统中后才会运行。以上的视窗管理器在 RH/Mandrake 光盘中都是可用的，只要你决定安装它们。我只会使用 KDE，尽管其他的管理器更小或更快，Gnome 是一个著名的 X - Windows 软件，它被称为比 KDE 更先进更漂亮，但是他还是有非常多的问题，所以除非你不介意时常碰到一些小问题，我不建议你使用它。RH6.x 包括了所有的主要的 X - Windows 系统，即 Gnome 和 KDE。

4.3.6 我能够同时运行多个 Linux 阶段作业吗？

当然可以。当你在系统中运行第一个 `startx` 命令时，此命令在默认的第 0 号显示区中打开了第一个 X-Session。第二个 X-Session 必须在不同的显示区打开。例如，此命令将会在 1 号显示区中打开第二个 X-Session：

```
startx -- :1
```

你最多可以拥有 6 个打开的 X-Session，使用 `<CTRL><ALT><F7>` 切换回第 0 号显示区。

`<CTRL><ALT><F8>` 切换到第二屏，等等，直到 `<CTRL><ALT><F12>`。如果需要了解更多的信息请 `try man`。

类似的情况，你也可以打开一个不带窗口管理器空的 X-server，下面这个命令将会在第二显示区打开一个此类程序。

```
xinit - :2
```

在这个基本的 X-display 之上，我可以的命令运行不同的窗口管理器（这样就会有二个完全不同的窗口管理器在同时运行）。在 X-terminal 上输入以下这些命令完成这个操作（阅读先前的答案可以获得更多的细节）：

```
startkde
gnome-session
xfce
afterstep
anotherlevel
fvwm2
fvwm
```

4.3.7 在不关闭我的 Linux 阶段作业的前提下可以开启第二个 GUI 登录提示符吗

在 RedHat 系统中，在不同的本地控制台中激活多个当前的 GUI 登录（RedHat 中使用 `gdm` 程序作为图形登录界面），我需要编辑以下文件：`/etc/X11/gdm.gdm.conf`。通过在文件的底部输入以下的内容，我激活了 4 个终端热键从 `<CTRL><ALT><F7>` 到 `<CTRL><ALT><F10>`。

```
[servers]
```

```
3=/usr/bin/X11/X vt10
2=/usr/bin/X11/X vt9
1=/usr/bin/X11/X vt8
0=/usr/bin/X11/X vt7
```

拥有四个 GUI 界面，使我在一台计算机上除了运行 KDE 和 GNOME 之外还可以同时拥有两个空闲的 GUI 界面，所以我的姐姐即使没有我的屏幕保护密码也可以登录计算机。

在 Mandrake 中(Mandrake 使用 KDE 作为缺省得图形登录界面)，在不同的本地控制台上激活多个当前的 GUI 登录界面，我必须修改以下两个文件：`/etc/X11/xdm.Xservers` 在这个文件中添加以下内容：

```
:0 local /usr/X11R6/bin/X : 0
:1 local /usr/X11R6/bin/X : 1
:1 local /usr/X11R6/bin/X : 1
```

另外一个文件是`/etc/X11/xdm/xdm-config` 在这个文件中将所有针对显示 0 到显示 1 和显示 2 设置添加进去：

```
DisplayManager._0.authorize: true
DisplayManager._1.authorize: true
DisplayManager._2.authorize: true
DisplayManager._0.setup: /etc/X11/xdm/Xsetup_0
DisplayManager._0.startup: /etc/X11/xdm/GiveConsole
DisplayManager._0.reset: /etc/X11/xdm/TakeConsole
DisplayManager._1.setup: /etc/X11/xdm/Xsetup_0
DisplayManager._1.startup: /etc/X11/xdm/GiveConsole
DisplayManager._1.reset: /etc/X11/xdm/TakeConsole
DisplayManager._2.setup: /etc/X11/xdm/Xsetup_0
DisplayManager._2.startup: /etc/X11/xdm/GiveConsole
DisplayManager._2.reset: /etc/X11/xdm/TakeConsole
```

以上操作可以让我在一台计算机上同时运行 Gnome 和 KDE 并且仍有三个空闲的 GUI 终端。

4.3.8 如何遥控 X-Windows ?

- 在本地计算机上启动 X-server，例如：

```
xinit
```

- 由 X-terminal 授权给远程计算机，允许它在你的本地计算机屏幕上输出显示内容。

```
xhost name_of_the_remote_server
```

在确保安全的环境下（比如在家中），我甚至可以使用以下命令给所有的服务器授权在我的屏幕上输出信息（注意，当连接到互联网时请勿使用此命令）。

```
xhost +
```

- 登陆远程服务器

在远程服务器上打开一个 X-program 将信息直接显示输出在你的本地的机器上，例如，你可以打开一个窗口管理器：

```
startkde -display local_machine_name : 0.0 &
```

符号“&”使命令在后台运行，所以你的远程登录窗口对你来说仍然是有效的，0.0 表示“display 0 server 0”。由于你的 Linux 系统计算机可以运行多个 X-session 此操作著名当前打开的是第一个显示区的第一屏幕。

如果你的当前终端中的环境变量由 DISPLAY 命令给出正确的定位，则可以不必指定“-display”操作。例子中的命令格式是我的系统中的默认值，并不是所有的系统都是如此，你可以使用以下命令，检查你的 DISPLAY 变量设置

```
echo $DISPLAY
```

- 在结束远程的 X-session 后，可以使用以下命令来恢复对 X-server 的控制：

```
xhost -name_of_the_remote_server
```

或者

```
xhost -
```

例如，下面一系列指令将在远程计算机上运行 Netscape，我们称之为“marie”，

```
startx
xhost marie
telenet marie
[登录]
netscape -display hacker : 0.0 &
[执行你的操作]
[退出]
xhost -marie
```

从原则上讲，你可以在网络上的任何一台计算机运行一个程序，将结果显示在另外一台计算机上（不一定是你正在操作的计算机）。

我在一台地配置的计算机上（486-33 8M 内存）使用 X-windowsing 运行了很多大型的应用程序（如 kde, Word Perfect8, 以及 Netscape 等），而在正常情况下这些程序在这台计算机上是不能运行的。同样在没有设置 nfs mount 的远程计算机上进行文件操作也很方便快捷。

X-windows 是专为网络上远程运行程序设计的操作系统，遥控 X-windowing 是一个非常有用的工具，若想获得更多的使用经验，请多作尝试。

你甚至可以在一个远程的 Linux（或者任何 Unix）系统中运行一个程序，然后重定向显示在一个本地的 MS 操作系统的计算机上，前提条件是在 MS Windows 中安装了 X-windowing。

若想得到更多的应用请查询：http://www.linuxworld.com/linuxworld/lw-2000-09/lw-090legacy_1.html

4.3.9 我如何从 MS Windows 分区安装 TrueType 字体？

有些应用程序使用 TrueType 字体，但是并没有附带 TrueType 字体的库文件（或者作为选购产品）。这样你可能需要安装你自己的 TrueType 字体。下面介绍一下我是如何手动安装 TrueType 字体的。Mandrake 包括了一个 GUI 工具传送 MS Windows 字体到 Linux，以下操作将会非常容易实现这个目标。

0. 从 K-menu (KDE 菜单) 中选择 “system” - “Font Manager” (或者相应的程序) 记录下你已经安装的字体。

1. 使用行命令，检查 “free type” 字体服务程序是否安装：

```
rpm -q freetype
```

这个命令使用 rpm 包管理器检查 “freetype” 程序包是否安装。如果此包已经安装，跳到下一步。如果没有安装，那么就要从你的发行版 CD 中安装它。我在安装 RedHat 时采用全部安装已经将 “free type” 安装到我的计算机中。

2. 以根用户身份登录，建立一个目录，用来存放 TrueType 字体：

```
cd /usr/X11R6/Lib/X11/fonts
mkdir TrueType
```

这个目录是由系统配置文件/etc/X11/XF86Config 指定，所以请确认你的安装目录与上述内容一致，如果你想在一个不同的目录下安装此文件，你需要编辑/etc/X11/XF86Config 文件，来调整相应的设置。在我的 “默认安装” RedHat 配置中包含以下的一行内容：

```
FontPath “/usr/X11R6/X11/fonts/TrueType”
```

3. 以根用户身份登录，将 *.ttf 文件由原始位置拷贝到你刚才建立的 TrueType Font 目录，我是从我的 MS Windows 分区中取得的 TrueType 字体，也许你需要一个不同的文件源位置：

```
cd /usr/X11R6/lib/fonts/TrueType
cp /mnt/dos_hdaL/windows/fonts/my_private_fonts/*.ttf
```

当然在拷贝任何字体之前，请确认这些字库没有妨害到你的许可授权。

4. 以根用户身份登录，运行以下的命令：

```
cd /usr/X11R6/lib/X11/fonts/TrueType
ttmkfdir > fonts.dir
cp fonts.dir fonts.scaLe
```

5. 关闭所有的 X-windows 应用程序，然后退出 X-windows

6. 以根用户身份登录。重新启动 X-font 服务程序 (或者重新启动计算机) :

```
/etc/rc.d/init.d/xfs stop
/etc/rc.d/init.d/xfs start
```

7. 重新登录你的 KDE 然后从 K 菜单中选择 “ System ” - “ Font Manager ” 来确认字体安装是否正确。

这部分的内容可以参考 : <http://www.computerbits.com/archive/20001000/linux0010.htm>

4.3.10 如何进行拷贝粘贴操作 ?

在 X-windows 下, 使用鼠标选中要被复制的文本, 切换到复制的目的地址。点击鼠标中键执行粘贴操作。此方法既快捷又好用。注意在切换过程中请不要取消对选中文本的选择。如果你的鼠标只有两个按键, 则同时按下两个键或者只按下有边的按键即可 (这取决于你的设置)。这个是标准的 X-windows 复制粘贴操作。

许多 GUI 应用程序 (不是所有的) 也支持 Mac/MS Windows 风格的复制粘贴操作, 选中被复制的文本, 使用菜单选项 “ 编辑 ” 中的 “ 复制 ” (或者从下拉菜单中选择, 亦可在鼠标右键的菜单中选择), 切换到复制的目的地址, 使用菜单选项 “ 编辑 ” 中的 “ 粘贴 ”。如此操作, 对那些使用同类工具的应用程序来说是很方便的。但是, 并不是所有的应用程序都可以这样操作 (诸如 GNOME 和 KDE 的应用程序之类)。

最后一个方法是键盘快捷键方式。我可以 <CTRL><C> 复制选中的文本使用 <CTRL><v> 进行粘贴操作。文本的选中也可以不使用鼠标, 使用组合键 <CTRL><Shift><RightArrow> 亦可完成。

你也可以使用复制粘贴历史, 运行 klipper 程序 (在 X-terminal 中, KDE)

这两个复制的方法是两个完全独立的方法, 所以它们之间应该是没有影响的。

为了将一个窗口的内容或者把整个屏幕存储成一个图形文件, 我使用 knapsnot 这个软件。或者使用 <ALT><PrintScreen> 组合键对当前屏幕进行复制并把内容存储在剪贴板中。组合键 <CTRL><ALT><PrintScreen> 会将整个桌面都复制下来存储在剪贴板中。

若要复制文本控制台的内容 (在 GUI 控制台之外), 我可以在 X-terminal 使用以下命令 (可能要以根用户身份登录) :

```
cat /dev/vcs1
```

然后就可以在 X 终端中使用鼠标操作我想复制和粘贴的内容了。若要在其他的终端进行复制粘贴操作可以调整 “ vcs1 ” 中的数值。

如果你运行了 gpm 后台程序, 那么基于文本的控制台就可以支持鼠标了。通过输入 gpm 来检测鼠标。如果鼠标已经经过了正确的设置, 它将会运行得很好。 (设置鼠标需要运行 moUseconfiG 程序。) 若要在系统启动时自动运行 gpm 并使其驻留在内存中, 选择 gpm 后台程序在其中使用 nt sysv 工具。

gpm 是使用方法类似于 GUI 复制粘贴操作：选中被复制的文本，将光标移动到复制的目的地址，然后点击鼠标中键（或者是双键亦或者是单击右键）完成粘贴操作。

4.3.11 我如何使用 VNC 来显示和控制一个遥控桌面？

(VNC= Virtual Networking Computing 虚拟网络计算机)。一个十分有用的应用程序。

VNC 是一个多平台的工具，它允许我们通过标准的网络连接来显示一个遥控的图形桌面。例如，我可以通过 VNC 在 MS Windows PC 上遥控楼下的 Linux 服务器的 X-windows 系统，或者使用其他的途径。VNC 甚至可以运行在一个 56K 的调制解调器接入的网络，这种接入方法只能用来作娱乐使用或者在紧急情况下（相对于正常的网络连接这种接入实在是太慢了）。

最近，Mandrake 和 RedHat 将在正式发行的光盘上包括 VNC 服务。至于你所下载的 MS Windows 系统是否支持这项服务请在以下网址下载更多的信息和详情：

<http://www.uk.research.att.com/vnc/>

在 Linux 中，VNC 包括以下四各命令：vncserver，vncviewer，vncpasswd，和 vncconnect。大多数情况下我只需要其中的两个命令：vncserver 和 vncviewer。下面对这些命令详细描述：

vncserver

此服务程序必须在主（或遥控）计算机上运行。你只能作为使用者（不需要根用户身份）使用此项服务。

vncviewer

本地应用程序，用于远程接入运行 vncserver 的计算机并显示其环境。你需要知道远程计算机的 IP 地址和 vncserver 设定的密码。

vncpasswd

vncserver 的密码设置工具。vncserver 服务程序没有设置密码将不能运行（好习惯）。如果你没有设置，运行 vncserver 时它会提示你输入一个密码。所以，一般我不会单独运行这个命令来设置密码。

vncconnect

告诉 vncserver 连接到远程一个运行 vncviewer 的计算机的 IP 和端口号。这样我就可以避免给其他人一个接入的密码。

Xvnc

一个 vnc “主控”程序，一般来书不需要直接运行。（vncserver 和 vncviewer 实际上是 Xvnc 的脚本）

查找所有可用的选项，运行：

Xvnc --help

出于安全的考虑，一般不建议直接以超级用户帐号运行 vncserver 程序。如果你需要超级用户的环境，请以一般用户登录后再使用 su 命令登录到超级用户帐号。

两个典型的例子如下：

例子 1：在 Windows 计算机上，我可以显示在远端的 Linux 服务器上的 X 窗口环境。按如下步骤：

[在 Windows 的机器上，启动 DOS 终端输入以下命令]

```
telnet my_linux_server_name
```

[在 Linux 服务器上用你的用户帐号登录，然后输入命令]

```
vncserver
```

[输入一个比较好的密码当系统提示输入密码时，我的密码是 “357+Simon&Garfunkel”]

[再次输入相同密码确认]

[注意观察屏幕输出信息，看 vnc 服务在哪一个屏幕窗口运行，我的是 “4”]

[在 Windows 机器上，从 “开始” -> “程序” -> “VNC” -> 运行 “vnc viewer”]

[在显示窗口里，输入远端 Linux 机器的 ip 地址和对应的屏幕窗口号，如下所示]

```
my_linux_server_ip_address:4
```

[在接下来出现的密码输入框里，输入密码]

```
357+Simon&Garfunkel
```

[你应该可以看到一个 X-Windows 的窗口显示在 Windows 的桌面上]

[就像平时你在 X-Windows 的环境下执行你的操作]

[完成之后，切换到你的 DOS 环境下的 telnet 会话进程，输入以下两条命令]

```
vncserver - kill :4
```

```
logout
```

例子 2：在 Linux 的 X 窗口环境下，我可以显示和远程控制 MS-Windows 的计算机屏幕。但愿没有人正在使用那一台计算机，否则他会看到我在移动他的鼠标光标。

按如下步骤：

[走到 Windows 的计算机上，因为你可能没有办法用 telnet 来访问它]

[从 “开始” 菜单，选择 “程序” -> “VNC” -> 运行 “WIN VNC (App 模式)”]

[从 “SystemTray”，用鼠标右击 vnc 的图标，选择 “Properties”]

[在对话框出现后，输入密码。把对应的屏幕号码设为 “auto”]

[走回你的 Linux 服务器，开启桌面]

[启动 X 终端输入命令]

```
vncviewer ms_windows_server_name_or_ip
```

[当提示出现时，输入密码]

[现在，你应该可以看到 MS-Windows 的窗口出现在你的 Linux 桌面上了]

[做你日常的工作]

[结束的时候，右击 System Tray 的 VNC 图标，选择 “关闭 VNC”]

4.4 : 基本配置

4.4.1 如何设置我的声卡？

试着运行 (以 root 用户)

```
sndconfig
```

除非你有一块非常特殊的声卡，这个程序会工作的很好。在设置过程的最后，是 Linus 的声音告诉你他怎样读 “ Linux ” (在 RedHat 中， “ sndconfig ” 也可以通过 “ 设置 ” 功能来调用——输入 setup)。

你可能想用命令行的播放器来试一试你的声卡和光驱。放一张音乐 CD 到你的光驱，然后输入：

```
cdplay
```

如果这样不工作，可能是你没有 /dev/cdrom？检查你是否能以 root 身份 mount 一张数据 CD，然后创建 /dev/cdrom 设备并将它链接到正确的设备 (很可能是 /dev/hdb)。例如 (以 root 用户)：

```
ln -s /dev/hdb /dev/cdrom
```

如果 cdplay 对 root 用户工作，但对其他用户不工作，你可能需要给予所有用户对设备文件 /dev/cdrom 的读/写许可，检查目录 /mnt/cdrom 的许可，或者想前面解释的那样修改 /etc/fstab 文件。比如：chmod 666 /dev/cdrom (/dev 目录是所有设备以文件形式出现的地方)

播放第三支歌，输入：

```
cdplay play 3
```

你可以用命令 cdp 使用 cdplay 的简单的命令行界面，但可能你更倾向于可在 X-Windows 下使用的界面 (例如：从 KDE 的 “ K ” 菜单，选择：多媒体光驱播放器)。

按下光驱上的按钮或者发出下面任何一条命令都可以停止播放音乐：

```
eject  
cdplay stop
```

故障处理 如果你的声卡出现问题，手工配置是另一个选择。这里是我设置一块 16 位兼容声霸卡的过程，由于检测有误的缘故它总是在半速状态工作 (很慢而且音量很小)。需要阅读的文档是 /usr/src/linux-xxx/Documentation/sound (希望你安装了核心源码，这样才会有文档)。文件 /etc/modules.conf 需要修改，关键的一行 (修改后) 是：

```
options sb esstype=1688 io=0x220 irq=5 dma=0 dma16=5 mpu_io=0x330
```

我要做的改动是插入 “ esstype= ” 选项。修改完毕后最好冷启动来测试你的设置 (用 shutdown 命令停机，然后重置电源)。

4.4.2 如何设置我的打印机？

打开一个 X 终端，然后输入（作为 root 用后，否则你会被提示输入 root 密码）：

```
printtool &
```

这个程序会完成全部的打印机设置，你只需要填入你的打印机类型信息以及它接在哪里。

指定正确的打印机端口是最重要的一步。如果你不知道是哪个，可以试着按如下方法配置：对于 RedHat 5.2：lp1(第一个并行口)，lp2(第二个并行口)，或 lp3(第三个并行口)；对于 RedHat 6.0 及以上版本：lp0(第一个并行口)，lp1(第二个并行口)，或者 lp3(第三个并行口)。从 RedHat 5.2 升级到 6.0 后，由于并行口名字改变导致打印机停止工作。我不得不重新运行 printtool 来调整打印端口。端口好的改变是为了与其它设备的编号总是从 0 开始的情况一致。

试着向打印端口发送一张 ASCII 文本测试页，只有当测试页打印正常的时候才算大功告成。如果你要配置的是一台远程打印机，确信你的机器对远程打印机有访问权限。访问权限是在连接那台远程打印机的机器中的/etc/hosts.lpd(安全程度较高)或/etc/hosts.equiv(安全程度较低)中设置的。这些文件只是简单地列出能够使用本地打印机的远程机器，一个机器名一行。我的文件看起来是这样的：

```
hacker  
mars
```

我的系统中本来没有/etc/hosts.lpd，所以我创建了一个。

欲了解你机器中的打印机信息，可以察看文件/etc/printcap：

```
cd /etc/  
cat printcap
```

这里是我的/etc/printcap 文件中某些代码的含义：

: 域分隔符（分隔文件中的条目）
\ (出现在行尾) 下一行继续
lp 打印机名。“lp”是你机器中默认的打印机名，默认情况下，多个打印机被顺序地命名为 lp0, lp1 等等（或任何你喜欢的名字）。但是不要将打印机名和打印机连接的并行口设备名混淆。

sd=/var/spool/lpd/lp 我的脱机目录(sd)。
mx#0 打印任务的最大块数。“0”表示没有限制。
sh 我希望忽略打印标题页。标题页在你的打印任务前显示你的名字（如果你在家打印意味着浪费纸张）。
Rm=mars 远程机器名，我的系统里是“mars”（我的打印机连在另一台计算机上）。
Rp=lp 或者 lp=/dev/lp0 远程打印机名，打印机在远程计算机上的名字（在我的家庭网络里，打印机“lp”连接在计算机“mars”上）。或者本地机器的设备名字。“/dev/lp0”在 RedHat 6.x 中是第一个并行口（在 RedHat 5.2 中曾经是/dev/lp1，并口号改变了）。
lf=/var/spool/lpd/lp/filter 输入过滤器文件(if)。你的打印任务在被送到打印机前，将会被这个“过滤器”格式化。

Sf 禁止换页 (sf)。通常情况下,当打印任务结束时发送换页指令(设为禁止换页,如果你的打印机总是在每个打印任务的最后多打印一张空白页)。

打印机是通过命令 lpc 来控制的(以 root 身份)。用“?”察看选项。这个程序以它的怪异而声名狼藉,所以不要轻易地被它吓住。察看打印队列用命令 lpq,清除打印任务用 lprm,这两个命令适用于所有用户(不仅仅是 root)。你能够从命令行用 lpr 命令打印。在 KDE 环境下,你能用“K-按钮”——“Utilities”——“Print Queue”控制打印队列。

大多数打印机在 Linux 下能工作得很好,但有的打印机可能不会发挥全部功能,这是由于厂商提供的信息或驱动程序不足造成。因此,购买打印机前,你可能要参考 Linux 打印机数据库:<http://www.linuxprinting.org/database.html>。简而言之,最好选择这里列出的打印机类型(<http://www.linuxprinting.org/suggested.html>):对于经济性彩色打印:一台 Epson Stylus,比如:Stylus C80(较好)或 Stylus C60(较便宜)。比较而言,惠普喷墨打印为次选。请注意,喷墨打印机一般情况下进行黑白打印时表现不是很好。

而且,喷墨打印机是为个人用户设计的,并不适合处理大量的打印。当然它们能提供优质的彩色输出,特别是在优质纸张上。孩子们喜欢喷墨打印机。

- 低端激光打印:Lexmark 或者 Brother 打印机。很多惠普激光打印机也能工作得很好,但是当你选择惠普打印机的时候一定要仔细挑选,因为它们提供的支持非常有限。廉价的激光打印机总是黑白的,但提供高质量的文本输出。如果你选择一台支持“Postscript”的打印机会避免很多麻烦。

4.4.3 Word Perfect 8 没有适合我的打印机的驱动程序

如果你已经在 KDE 里通过 printtool 安装了打印机,有一个驱动程序并且工作正常,那么设置 Word Perfect 使用“passthru postscript”驱动程序。

4.4.4 配置文件在哪里?

系统级的配置存放在/etc目录中。用户级的配置存放在用户的主目录/home/user_login_name。这里是我常用的系统级配置文件列表:

SHELL 默认文件

/etc/bashrc - bash shell 的系统级默认功能和别名

/etc/profile - bash shell 的系统级默认值,包括系统级的环境变量

/etc/passwd - 含有用户的密码和其他信息。Root 用户能够直接修改,但建议用配置工具修改,例如 passwd 命令。一个损坏的/etc/passwd 很容易令一个 Linux 系统不可用。

/etc/shadow - 存有 passwd 文件的“shadow”信息。比如:不应被所有人看到的信息。

/etc/group - 类似/etc/passwd 文件,但是关于用户组的。

/etc/crontab - 设置“cron”,意为定期地执行命令(以小时、天、星期、年等单位)。

/etc/initab - 系统启动时运行不同的程序和进程。

/etc/issue - 和登录提示一起出现的信息。常常被 rc.local 脚本覆盖。

/etc/issue.net - 与上面相同,但是在通过网络登录时使用。

/etc/motd - “每日消息(Message of the day)”文件,用户登录后显示。

/etc/rc.d/rc.local - 系统启动时最后执行的脚本。我把定制我的本地机器的命令放在此文件的末尾。它的功能类似 DOS 的“autoexec.bat”。

网络配置

/etc/hosts - 含有一个主机名和固定 ip 地址列表
/etc/hosts.allow - 允许使用网络服务的主机名
/etc/hosts.deny - 禁止使用网络服务的主机名
/etc/resolv.conf - 设置了本地机器使用的域名服务器列表
/etc/inetd.conf - 守护进程 inetd 的配置文件, 说明了你的机器提供哪些 TCP/IP 服务。
/etc/exports - 指明了哪些文件系统能想那些主机提供网络文件系统 (NFS)。Man exports 包含如何为远程用户设置此文件的信息。

硬件配置

/etc/conf.modules - 配置 linux 的核心模块。模块类似 MS Windows 或 DOS 下的设备驱动程序。
/etc/fstab - 还有分区和文件系统信息。系统用来 mount 目录树上不同的文件和分区。
/etc/mtab - 显示当前以被 mount 的设备和分区, 以及它们的状态。
/etc/lilo.conf - lilo 启动管理程序的配置文件。
/boot/grub/grub.conf - grub 启动管理程序的配置文件。
/etc/printcap - 打印机设置
/etc/termcap - ASCII 数据库, 定义了不同控制台、终端和打印机的功能和字符特性。你一般不去改变它们。
/etc/X11/XF86Config - X 配置文件。Xfree 4.xx 版本的配置文件是/etc/X11/XF86Config-4 (如果它不存在, 系统会试用 XF86Config)。

4.4.5 这些设备文件都是什么?

设备以文件的形式出现在目录/dev中。它们可读、可写, 如果你有权限这么做的话。文件列表能揭示设备的一些重要细节:

```
ls -l /dev/ttyS3
```

我的系统给出如下的输出:

```
crwxr-xr-x 1 root tty 4, 67 Mar 13 22:59 ttyS3
```

最左边的“c”表明这是一个字符设备。如果是“b”则意味着“块设备”, “p”=先入先出设备 (FIFO), “u”=非缓冲字符设备, “d”=目录, “l”=符号链接。数字“4, 67”的意思是主设备为4, 次设备号为67。为了使你系统中所有用户能使用某些设备, 你需要设置正确的权限。

例如:

```
ls -l /dev/usb/scanner0  
chmod 666 /dev/usb/scanner0
```

这里有一个常见设备的清单:

/dev/ttyS0 - 第一个串行口。鼠标通常接在这里。
/dev/ttyS1 - 第二个串行口。你的调制解调器很可能接在这里。
/dev/ttyS2 和/dev/ttyS3 是第三和第四个串行口 (一般不存在, 但你的内置调制解调器可以被设置成使用其中之一)。

/dev/modem - 串口调制解调器。通常是一个指向/dev/ttyS1、/dev/ttyS2 或/dev/ttyS3 的符号链接，具体取决于你的调制解调器接在那个串行端口。

/dev/mouse - 鼠标。通常是一个指向/dev/ttyS0 或相似设备（见上）的符号链接，具体取决于你的鼠标接在哪个串行端口。

/dev/lp0 - 接在第一个并行口的打印机。通常你的打印机接在这里。

/dev/lp1 - 接在第二个并行口的打印机（一般不存在）。

/dev/fd0 - 第一个软盘驱动器（一般都有）。

/dev/fd0H1440 - 第一个软盘驱动器的高密度模式驱动程序。一般情况下，这个驱动程序（或其他有相似名字的设备驱动程序）在以特定密度格式化一张软盘的时候被调用。Slackware 还有支持将一张 3.5 英寸软盘格式化为最大 1.7MB 的驱动程序。默认情况下，RedHat 和 Mandrake 没有这些驱动程序。

/dev/fd1 - 第二个软盘驱动器。

/dev/hda - 第一个 IDE 硬盘（整个硬盘）。大多数 IBM 兼容 PC 的硬盘是 IDE 设备。

/dev/hdb - 第二个 IDE 硬盘（整个硬盘）。很多计算机中，IDE 光驱接在这里。

/dev/hdc - 第三个 IDE 硬盘（整个硬盘）。很多计算机中，IDE 光驱接在这里。

/dev/cdrom - 指向相应设备的链接，通常是/dev/hdc 或/dev/hdb (CDROM) 或/dev/scd0 (CD-R/RW)。

/dev/hda1 - 第一个 IDE 硬盘的第一个分区。/dev/hda2 是第一个 IDE 硬盘的第二个分区。不难猜测，/dev/hdd8 将会是第四个 IDE 硬盘的第八分区。

/dev/tty1 - 第一个字符终端。/dev/tty2 是第二个字符终端，以此类推。

/dev/dsp - 数字音频，例如声卡。“dsp”是“digital signal processing (数字信号处理)”的缩写。

/dev/sndstat - 用 cat /dev/sndstat 来更多地了解你的声卡状态。

/dev/null - 当你不想要输出的时候就送到这里。

/dev/random - 用来读伪随机数。用 cat /dev/random 来在屏幕上显示一对乱七八糟的字符。还有一个/dev/urandom 设备用来生成低质量的随机数序列。

/dev/sda - 第一个 SCSI 硬盘（整个硬盘）。在一台家用计算机中，不太可能有 SCSI 设备（昂贵）。

/dev/sdb - 第二个 SCSI 硬盘（“sdc”是第三个 SCSI 硬盘，以此类推。一个系统中可以有多个 SCSI 硬盘）。

/dev/sda1 - 第一个 SCSI 硬盘的第一个分区。

/dev/sr0 - 第一个 SCSI 光驱（有时叫做/dev/scd0）。如果你有一个 ATAPI CD Writer，也可能在这里。

/dev/sr1 - 第二个 SCSI 光驱（有时叫做/dev/scd1）。（/dev/sr2 是第三个 SCSI 光驱，以此类推。系统中可以有多个 SCSI 光驱）

/dev/usb/scanner0 - 一个 usb 扫描仪。试用 less
/usr/src/linux/Documentation/usb/scanner.txt 了解如何从头开始配置扫描仪。欲了解更多：

```
less /usr/src/linux/Documentation/devices.txt  
man MAKEDEV
```

象/usr/src/linux/Documentation/devices.txt 解释的那样，我可能需要创建一些指向设备文件本身的符号链接来配置我的系统。这只是一个简单的实例，并不意味着通用的建议。当然，如果这些链接存在，它们应该有以下的：

```
/dev/mouse 当前鼠标端口***  
/dev/tape 当前磁带设备
```

```
/dev/cdrom 当前光驱设备***
/dev/cdwriter 当前 CD-writer 设备 ( 但是我的 RedHat 是 /dev/cdrecorder )
/dev/scanner 当前扫描仪设备
/dev/modem 当前拨出 ( 调制解调器 ) 端口
/dev/root 当前 root 文件系统
/dev/swap 当前 swap 设备
```

***表示我的 Mandrake 系统中确实存在的符号链接。例如, 如果鼠标有什么问题, 我会做下面的事情 (以 root) :

```
ls -l /dev/mouse
[检查鼠标设备是否存在且指向哪里]
ln -s /dev/ttyS0 /dev/mouse
[创建一个符号链接/dev/mouse 指向第一个串行端口]
```

对于 SCSI (及 ATAPI) 设备, /dev/tape 和 /dev/cdrom 应该指向“熟”设备 (/dev/st*和/dev/sr*, 相应地), 而 /dev/cdwriter 和 /dev/scanner 应该指向相应的普通 SCSI 设备 (/dev/sg*)。非临时套接字和管道可能存在于 /dev。常见的有:

```
/dev/printer socket lpd local socket
/dev/log socket syslog local socket
/dev/gpmdata socket gpm mouse multiplexer
```

若干 Linux 守护进程

守护进程是定期被唤醒、检查你的系统, 然后执行某些特定功能的驻留程序。它们一般不接受任何输入, 通常也不产生任何输出。你的 Linux 系统可能以设置了运行相当多的守护进程。其中大多数可以选择 (或不选) 通过以 root 身份运行程序 ntsysv (RedHat), 然后选择相应项目。

每个守护进程的简短描述可以在 ntsysv 中按<F1>得到。如果你需要的守护进程没有在 Ntsysv 中列出来, 你得放入 RedHat/Mandrake 的安装 CD, 安装相应的软件。Ntsysv 的另一个选择可以是 tksysv (以 root 身份从 X 终端输入), 它可能更灵活, 但也可能更复杂 (它能让你在每一个运行级别上设置守护进程)。另外, 单一但更强大、灵活、难以使用的工具是 /sbin/chkconfig。

这里有一个包含简短描述的常见守护进程列表:

anacron – 检查由于系统关机导致未被执行的“cron”任务, 并且执行它们。对于你有计划好的 cron 任务但却不能每次运行的情况特别有用——anacron 将在启动时检查它们。
amd – automount 守护进程 (自动 mount 可移动介质)。
apmd – 高级电源管理 BIOS 守护进程。适用于支持 apm 的机器, 尤其是膝上型笔记本电脑。
arpwatch – 保持观察以太网/ip 地址对。
atd – 运行由“at”排队的任务。
autofs – 控制 automount 守护进程的操作 (对 amd 的竞争)。
bootparamd – 向无盘工作站提供必要启动信息的服务器进程。
crond – 自动任务调度程序。管理那些定期但不经常执行的任务, 例如循环日志文件, 清理 /tmp 目录, 等等。
cupsd – 通用 UNIX 打印系统 (CUPS) 守护进程。CUPS 是一个先进的假脱机打印系统, 允许一台打印机在整个网络的打印选项及自动功能设置在一台服务器上。它是 Mandrake Linux 的默认打印系统。

dhcpcd - 实现动态主机配置协议 (DHCP) 和 Internet Bootstrap 协议 (BOOTP)。
gated - 取代 routed 和 egppup 并能处理多种路由协议的路由守护进程。
gpm - 非常有用的鼠标服务, 支持运行在 Linux 字符终端的应用程序。
httpd - Apache Web Server 守护进程。
inetd - 监听网络服务请求连结, 特别是拨入服务。这个进程可以自动加载或卸载其他守护进程 (ftppd, telnetd, 等等), 从而节省系统资源。新的系统中以被 xinetd 代替。
isdnlinux - 服务用户的 ISDN 卡。
kerneld - 自动加载和卸载核心模块。
klogd - 解释和显示日志信息的进程。核心消息取决于消息的等级。可能的等级有 (从 /usr/include/linux/kernel.h 复制而来) :

KERN_EMERG "<0>" 系统不可用

KERN_ALERT "<1>" 必须采取措施

KERN_CRIT "<2>" 严重状态

KERN_ERR "<3>" 错误状态

KERN_WARNING "<4>" 警告状态

KERN_NOTICE "<5>" 正常但重要的情况

KERN_INFO "<6>" 消息

KERN_DEBUG "<7>" 调试级别信息

消息通常会被记录到 /var/log/kernel 目录下相应的文件中。

kudzu - 启动过程中检测并配置新添会改变的设备。

keytable - 装入选定的键盘布局。

linuxconf - 配置工具。如果你希望 linuxconf 在启动中执行一些维护系统配置的任务, 它的自动执行部分将会运行。

lpd - 打印进程。

mcserv - Midnight Commander 网络文件系统的服务器程序。它向运行 Midnight 文件系统 (当前, 只有 Midnight Commander 文件管理器) 的客户提供对主机文件系统的访问。如果是 Root 用户身份运行, 此程序将会尝试获得一个保留端口。否则, 它会使用 9876 端口。如果系统有 Portmapper 运行, 那么这个端口将会被 portmapper 记录, 然后客户会自动连结到正确的端口。如果系统没有 portmapper 在运行, 那么必须用参数 -p 手工指定一个 (见下)。

named - 互联网域名服务器 (DNS) 守护进程。

netfs - 网络文件系统 mounter。用来在启动时 mount nfs, smb, 和 ncp 共享。

network - 通过调用 /etc/sysconfig/network-scripts 中的脚本程序在启动过程中激活所有网卡。

nfsd - 输出 nfs 共享, 当远程系统请求时。

ndlock - 启动和停止 nfs 文件加锁服务。

numlock - init 运行级别变化时锁住 numlock 键。

pcmcia - 膝上型笔记本电脑中 pcmcia 卡的一般服务。

portmap - 远程系统调用所需。通常, 你需要它来运行网络。

postfix - 邮件发送代理 sendmail 的替代品。现在是桌面 Mandrake (RedHat 使用 sendmail) 桌面安装的默认值。

random - 保存和恢复“熵”池从而生成更高质量的随机数。

routed - 管理路由表的守护进程。

rstatd - 核心统计服务器。

rusersd, rwalld - 识别用户和远程用户的“wall”消息服务。

rwhod - 维护 rwho(1) 和 ruptime(1) 所用数据库的服务器进程。它依赖于向网络发送广播信息的能力。

sendmail - 邮件发送代理。RedHat 的默认邮件发送程序。

smbd - SAMBA (或 smb) 进程, 面向你的网络中 MS Windows 计算机的网络连接服务 (硬盘、打印机共享, 等等)。

squid - 带缓存的 http 代理。代理将来自客户的请求转发到外部世界, 然后再将结果返回。你可能会用这个代理, 如果你希望用你的 Linux 机器做为你的网络中其他计算机访问 Internet 的网关。另一个实现办法 (在家里可能更安全) 是配置 masquarading。

syslogd - 管理系统活动日志。配置文件为 /etc/syslog.conf。

smtpd - 简单邮件传输协议, 为交换电子邮件设计。一些支持 SMTP 的守护进程包括 sendmail, smtpd, rsmtpd, qmail, zmail, 等等。

usb - 管理 USB 设备的进程。

xfs - X 字体服务器。

xntpd - 为 NIS 域发现服务器并把信息保存在绑定文件中。

ypbind - NIS 绑定服务器。位于网络信息服务域 (NIS Domain) 中的机器需要。

4.5 网络

4.5.1 有必要建立家庭网络吗？

这是一个很好的想法。有了家庭网络，你放在壁橱里早被淘汰的计算机照样可以运行得很出色。家庭网络的好处包括，可以共享硬盘，zip 盘，CDROMs，调制解调器，打印机，甚至声卡，远程运行程序（文本或图形模式），通过一条电话线连接让所有计算机同时浏览网络。如果你一旦失去对你的计算机的控制，还可以远程关闭它，减少出问题的可能性——详细解答参考“关机”。

4.5.2 怎样建立我的家庭网络？

硬件

所有的硬件必须正确安装。网卡应该在 RedHat 系统初始安装时已经安装好了。如果要在以后添加新卡的话，系统在重新引导时会自动检测出网卡并进行配置。如果系统在引导时不能正常检测和配置新卡，你可能现在就要自己手动安装了。如果你安装了不止一块网卡，那么其他网卡也都需要手工配置，因为 Linux 只能检测出一块网卡。

在 Linux 系统下，大多数网卡的驱动都是模块化的。所以用手工设置网卡时仅仅是添加适当的模块和参数。你不需要重新编译内核，也不需重新启动计算机（除非网卡比较特殊）。要了解需要什么样的模块和参数，你可能需要参考内核源码的文档：

```
less /usr/src/linux-2.4/Documentation/networking/net-modules.txt
```

必要的话，你可以使用命令：

```
modprobe -1 | more
```

列出内核使用的所有模块。

如果你需要更多的帮助，请参阅 Linux Network Administrator Guide（系统文件 `/user/share/doc/LDP/nag` 或访问：[//www.tldp.org/LDP/nag/](http://www.tldp.org/LDP/nag/)）。这个指南非常出色。

有很多种把模块加入内核的方法。对于网卡，最简单的就是以 root 的权限运行 `netconf` 命令，在那里指定模块的名字。另一种方法是开启一个 X-terminal，运行“su”（使你具有 root 的权限），然后运行 `/usr/bin/kernelcfg`。

大多数情况你不必为网卡模块指定参数（IRQ 和地址）--模块本身知道应该用什么样的参数。不过，在一台计算机上设置两个网卡时，要仔细考虑参数的设置。必须确认没有硬件冲突。经常出现的问题是：当你要把网卡设定成 IRQ5 或 IRQ3 时，IRQ5 却被声卡使用了，而 IRQ3 又与第二个串口（COM2,cua1,ttys1）冲突。检查文件 `/proc/interrupts`，`/proc/ioport` 和阅读引导信息可以对你有所帮助。

例如，我的 WD8013 网卡（与 SMC Elite 和 SMC Elite Plus 相同）使用 IRQ10（通过卡上跳线设置并设置在内核模块中），地址 0x300 和“WD”模块。我的“SMC EtherEZ”网卡（卡上无跳线）使用

IRQ9, 地址 0x240 和 “ SMC EtherEZ ” 模块。千万不要省掉地址中的 “ 0x ”, 它表示 “ 十六进制 ”, 而没有该引导符的数字, 将被认为是十进制。

加入模块以后你也许要用 `/proc/modules` 检测文件看模块是否确实加载了 (或以 `root` 权限运行 `lsmod`)。模块配置文件是 `/etc/modules.conf`, 如果你遇到困难 (我曾错误地加入文件), 则可用 `pico` 或其他工具手工编辑和调整该文件。

网络配置

安装网卡和连接网线之后, 以 `root` 权限运行: `netconf` 来设置网络。这个程序很有用处! `netconf` 也能在图形用户界面下面执行, 但是我没有试过。如果你需要更多地了解网络是如何运行的, 可以看看以前的话题。另外, 下面的资料也会有所帮助:

<http://ieee.uow.edu.au/~mjp16/wylug-netlinux/notes.html>.

它也包含了一些非常好的关于如何建立小型办公室网络的例子。就像我们的家庭网络一样, 这些网络只是在某些特定时间和外部网络相通。

在 `netconf` 中怎样填写和填写什么要由你的网络环境来决定。你肯定要填写 “ 基本主机信息 ” (启用第一个以太网卡, `eth0`, 填写本地计算机的名称, 代号和 IP 地址) 和 “ 其他主机信息 ” (家庭网络上的其他主机的名称, 代号和 IP 地址)。这些信息将存入文件: `/etc/hosts` 和 `/etc/sysconfig/network`, 所以或许要检测这些文件并进行适当的手工调整。

当设置网络时, 不要把 IP 地址 127.0.0.1 “ 环路测试地址 ” 搞坏。IP 地址 127.0.0.1 必须总是在配置文件里 -- 计算机通过这个 IP 地址来自己和自己对话。

如果你没有 IP 地址 (就是典型的没有对外界永久性连接的家庭网络的情况), 你可以创建一个。它是什么并不重要, 因为当通过 `modem` 和外界连接上时, 你将会被动态地分配一个 IP 地址 (一个机器可以有多个 IP 地址)。

但是你创造的 IP 必须在格式上是正确的。并且, 网络掩码必须和网络地址类型的 (种类 A, B, 或者 C) 相匹配。见前面提到的 “ 网络管理员指南 ” IP 地址一章。 (帮助文件应该在系统上 `/usr/doc/LPD/nag` 目录里。如果没有找到, 可以在 <http://metalab.unc.edu/mdw/index.html#guid> 下载。对家庭的网络, 你可以创造 C 类网络 (最高 254 台机器——最小的网络), IP 的第一个 3 位数字在 192 ~ 223 之间。最后的 3 数字识别 C 类网络上的机器, 必须在 1 和 254 之间 (不要使用 0, 它代表整个网络, 或 255 代表 “ 广播地址 ”)。中间的两组, 能是 0 ~ 255 的任何数字。这样, 223.223.223.1 是 223.223.223.0 (最后的零表示全部网络), 广播地址 223.223.223.255 的 C 类网络的第一台机器。C 类网络的掩码总是 255.255.255.0 (除非你把你的网络再分成更小的 “ 子网络 ”, 我们不在这里讨论它)。使用为 “ 私有网络 ” 保留的 IP 地址比重新创造一个更安全。对我而言, IP 192.168.1.1 工作得很好。使用这种方法, 你创造的 IP 地址保证不会干扰任何其他地址。典型地, 网络上的第一台机器期望被用来与外界连接 (因为它是第一个被联接上, 但没有一个标准)。所以, 我在所有的机器上都把 `GATEWAY` 设置为 `XXX.XXX.XXX.1`, 但我让机器 `XXX.XXX.XXX.1` 本身这个地址空着。尽管我让这个地址空着, `netconf` 插入 `gateway` 到 `/etc/sysconf` 文件中。曾有在启动时掉网卡期间出现烦人的错误信息。为了去掉它, 编辑 `/etc.sysconfig/network` 如下:

```
GATEWAYDEV=  
GATEWAY=
```

[如果你不在你的机器上的 ethernet 接口设置一个网关, 这台机器就不会连到外面。你通过调制解调器连接, 在第一台机器上的 PPP 接口将会被设置为默认网关, 例如使用 kppp]

计算机的名字完全自由 - 用户通常选自己喜欢和熟悉词汇。除非你域名登记为永久性的连接, 家庭的网络的域名可以随便命名。试着去创造一些可能别人没有用过的名字 - 这样一旦当你有永久性的连接, 它会使你的工作容易一些。

至于 DNS (域名服务器, 在 linux 里叫 named = name daemon), RedHat 5.2 和 6.0 安装时自带 “ caching-only ” 功能的域名服务器, 所以配置相对来说比较简单。另外, 你可以选择不使用本地 DNS - 如果你的本地 DNS 正在查 “ 外部的 ” 服务器却找不到它, 机器会被挂起几分钟如同死机。使用本地 DNS, “ 命名的 ” 服务必须能在运行设置时检查。设置缓冲 DNS, 在 netconf 中设置足够的空间。例如, 我给全部计算机上装满了表示第一台计算机的 “ nameserver 1 ” (我在机器 xxx.xxx.xxx.1 和在全部其他的机器上 xxx.xxx.xxx.1 地址进入了回路测试地址 127.0.0.1)。在我的网络中没有路由指向其他的网络和主机, 因为我没有其他的网络。

除了正确地安装好硬件, 然后在每个计算机的 netconf 填写必要的配置信息。为了使网络正常运行, 已经不需要在标准的 RedHat 上做更多的事情。

现在一个接一个地重新起动你家庭网络里的全部计算机 (这不是必须的, 但是也没有坏处), 注意看引导信息 (如果滚屏速度太快, 可以用 <Shift><PgUp> 向前翻页, 或者过后用命令 dmesg 来显示这些信息)。你的卡设置正确吗? 用命令 route (root 权限) 来看一看 eth0 接口是否运行。用 ping 命令来试验单个的机器之间的连接。试试远程联接到你的计算机, 看 loopback-only 接口是否工作, 使用命令:

```
telnet name_of_the_machine_you_are_sitting_at
```

成功登录后, 你可输入

```
exit
```

退出 telnet。最终, 你可以远程联接你网络上的所有计算机, 用命令:

```
telnet name_of_a_remote_machine
```

如果每一台机器都工作, eth0 网络接口就设置好了。

设置完 PPP 和连接到你的 ISP 之后, 会有另外一个网络接口 (PPP0) 并能远程登录到世界上的任意一台机器上。

4.5.3 设置 PPP 拨号 GUI 的问题

GUI 图形用户界面

如果我是你, 我将安装 KED 并使用 kppp 拨号软件。kppp 非常容易设置和运行。要设置你的 PPP 拨号, 运行 kppp (它在 “ KDE ” 菜单的 “ Internet ” 子菜单里, 你也可以在 xterm 输入 kppp 运行)。按 “ setup ” 键, 创建一个账户, 然后填写一些要求的信息: 比如电话号码, 授权协议, 你的 ISP 的域名和 ISP 的域名服务器 (DNS) 等。这些必要信息, 你的 ISP 应该提供给你了。

下一步，我必须检查/etc/modem 是否存在并且指向正确的端口。如果有必要，我还可以建立一个连接到/dev/ttyS1（需要 root 权限）：

```
ls -l /dev/modem ln -s /dev/ttyS1
```

如果你的 modem(调制解调器)在 (DOS 上是连接在 COM2 的口中，ttyS1 应该没有什么问题。类似的，ttyS0 用于 COM1，ttyS2 用于 COM3，ttyS3 用于 COM4。如果端口设置错误，modem 将不能正常拨号。对于为非标准的串行口设置，参见 setserial 的命令。

在旧版本的 kppp 软件，你可能碰到一个加锁(lock)出错的信息。解决办法是，确定编辑文件/etc/ppp/options 是空的（需要 root 权限）：

```
pico /etc/ppp/options
```

并且删除“lock”然后保存文件。近期的版本已经不存在这个问题。

RedHat 6.0 需要一个补充的步骤：设置“suid”（“继承用户 id”）以使“kppp”以根权限的有效用户 ID（它需要直接访问硬件）。如果没有它，kppp 提示“设置不对”，和“不能创建 lock 文件”。必须以 root 权限运行以下命令：

```
cd /usr/bin/chmod a+s kppp
```

常见问题的解决方法

如果你设置了正确的串口但是 modem 却不能正常拨号，modem 可能设置的不对（或许它是一个“winmodem”，扔掉它，买一个不好一点的 modem）。例如，我曾经运行 kppp 设置，自己手动编辑 modem 命令和输入“初始化字符串”ATZ1（代替默认的标准 Hayes 兼容的 ATZ 命令）。否则，我的那个 modem 不能正常拨号。

如果你的电话线需要脉冲式拨号（而不是通常的双音频拨号），你或许要把 modem 拨号指令由“ATDT”变成（="ATtention Dial Touchtone"）“ATDP”（="ATtention Dial Pulse"）。这些命令适用于任何贺氏兼容的 modem。如果你的 modem 拨号正确且也连接上了，但你的身份验证失败，或许你的 ISP 使用了不同的身份验证协议。打电话给他们，询问他们使用什么验证协议。或者试试“pap”，“terminal-based”或“chap”（在你建立 kppp 时）知道你发现那一个正常工作为止。

例如：我曾经碰到连接可靠性的问题（大概出错提示信息：“time-out for the pppd startup”，连接在开始几次就已经建立了）。后来，我改变“流控”选择，把“CRTSCTS”改为 XON/XOFF 这个问题就解决了。（大多数情况下，推荐的流控方法依然是 CRTSCTS。）

非正常断线有很多原因：

(1) 电话线路质量。例如：通过噪音很大的电话线。大多数的 modem 器会自行挂断电话。为了改变这个缺省设置，你或许需要把“S10=50”加到你的“modem initialization string”上。这个设置使得挂机之后 0.01 秒才清除 DTR 信号。（详细请参考你的 modem 手册，“US Robotics”或许需要“S25=200”）

(2) 如果有人给你打电话, 呼叫等待会中断你的连接。

(3) 旧电话线和肮脏或锈蚀的电话插头插座经常会出现问题。检查接头, 更换电话线。把电话线放到远离电话噪声的地方。

(4) 在距离较长和噪声较大的农村使用 modem 的高速。降低调制解调器速度 (或者搬到城市)。

(5) 很多 ISP 在侦测到你没有在激活状态时 (30 分钟?) 会自行中断你的连接。

如果你在设置 ppp 时总有问题, 你或许可以试试 minicom, 看看 modem 是否能够正常工作。Minicom 有点像 Linux 的 PROCOMM。如果你在 RedHat 初始设置时你选择安装了 minicom, 你的系统就会有这个软件。这是一个源自 comp.so.linux.help 新闻组的帖子, 它解释了如何使用 minicom 手工启动 PPP。

这是一个源自 comp.so.linux.help 新闻组的帖子, 它解释了如何使用 minicom 手工启动 PPP :

发自: mark <balthazaar@one.net.au>

主题:

回应: kppp 的 pppd 问题

BachuZ 写道:

>做一个实验, 试着使用 minicom 连接你的 ISP, 手工启动 ppp ... 能证明有毛病的 scripts.

> 怎么做?

很容易!! 如果你的 ISP 不允许用手动操作的登录你就难办了。所有我使用过的 ISP 都允许这这样。

1. 启动 minicom。
2. 拨打你的 ISP。
3. 登录。
4. 在你的 ISP 启动 PPP 之后, 用 ALT-Q 退出 minicom (或不管结果如何都不复位退出)。
5. 启动 pppd 例如: `pppd -d -detach /dev/modem 115200 &`

OK, PPP 将要运行。试着 ping 你的 ISP, 或者任何一个已知的 IP 地址。测试应该一切正常。

顺便说一下, 这些都在 PPP-HOWTO 的手册里。如果你能让 PPP 按这种方式运行, 接下来你可能会会有一个脚本的问题。如果 PPP 能正常运行, 你可能要重新配置 PPP。

祝你好运!

命令行方式

如果你要从命令行启动 PPP, 以 root 权限运行 netconf 来配置你的第一个 PPP 接口 (PPP0)。你要输入的信息大概和 kppp 差不多 (参考上面的描述): 正确的 modem 口上的设备名称, modem 初始化和拨号字符串, ISP 的电话号码, 认证协议 (在正确的地方输入登录名和 PASSWORD) 都要正确。在一些更老的版本中, netconf 没有地方供输入 ISP 的 DNS 服务器的 IP 地址, 所以, 我编辑

了文件 `/etc/ppp/pap-secrets` (我使用 PAP 认证协议), 且在 `netconf` 创建的设置行的最后加了两个 DNS IP 地址, 看起来像:

```
# added by linuxconf
```

```
my_login_name ppp0 my_password 111.111.111.111 222.222.222.222
```

当设置完成时, 我就可用如下命令启动 PPP0 接口 (以 root 权限, 除非我在 `netconf` 中制定普通用户可以启动接口):

```
ifup ppp0
```

或者可以用以下命令把 PPP0 中止

```
ifdown ppp0
```

命令行 `ppp` 的设置并不比运行 `kppp` 复杂多少, 而且连接更可靠。有许多命令脚本可以用来启动和中止 `ppp`。但是, 它们不容易设置和使用, 并且, 在很多新手那里经常会出现各种问题。异常的 `ppp` 连接可以用以下命令清除 (root 权限), 且无需重新启动系统:

```
killall pppd
```

如果这些关于 `ppp` 设置的最基本信息对你没有什么太大帮助, 这里有一些有用的链接供参考:

Roderick A. Anderson < raanders@altoplanos.net > wrote:我每次都用我的 linux5.x 拨号访问我的网页。它在 <http://home.altoplanos.net/~raanders/diald.html>

Bill Unruh < unruh@physics.ubc.ca > wrote:我正好想让你注意

<http://axion.physics.ubc.ca/ppp-linux.html> 那里有关于 `ppp` 设置的详细的指令。这是一个远程使用 PAP 或 CHAP 特殊的情况 (或许没有 ISP 告诉你这些信息)。Kppp 的确非常好用, 但是很多时候它也经常出问题。

4.5.4 如何不通过 modem 用我的联网电脑浏览互联网?

你网络上的一台计算机必须有安装 modem (或者其他的 Internet 连接如 DSL 和 CABLE 等)。然后设置 "IP 伪装", 这样, 所有从你的网络到你的 ISP 的请求看起来好像来自同一台电脑, ISP 就会让它们通过。

在 RedHat 5.2 上简单的 "IP 伪装" 仅是一个简单的命令 (在有 modem 的计算机上运行):

```
/sbin/ipfwadm -F -p m
```

这个命令设置你的 ip 防火墙默认策略为 "IP 伪装"。顺便要提一下的是, 它不是很安全, 但对于家庭用户来说应该没有问题。(潜在的 danger 是: 如果有人攻击你的电脑, 他可以使用匿名用户来隐含他的身份。这样, 不管黑客进行何种恶意攻击, 你都将成为替罪羊受到谴责。黑客甚至设置他的路由回到你的网络, 然后通过你来隐藏他的真实身份)。了解更多的信息, 请参考文件:

```
/usr/doc/HOWTO/mini/IP-Masquerade.
```

以下显示了一些更安全的设置：

```
ipfwadm -F -p deny
ipfwadm -F -a m -S 192.168.1.1/32 -D 0.0.0.0/0
ipfwadm -F -a m -S 192.168.1.3/32 -D 0.0.0.0/0
ipfwadm -F -a m -S 192.168.2.0/24 -D 0.0.0.0/0
```

这个设置默认策略是“拒绝”和明确两个伪装机器的 IP 192.168.1.1 and 192.168.1.3。它也伪装任一台网络 192.168.2.0 上的机器。数/32 表示点对点的网络（这意味着“machine-to-machine”），C 类网络选择/24 识别。-D 0.0.0.0/0 识别默认的路径是化装的机器连接 Internet。在 REDHAT 6.0 上命令的名字是 ipfwadm-wrapper（ipfwadm 代替），并且，我必须使用第二个，更安全（设置伪装为缺省策略在我的系统上不工作）。另外，，用新内核（2.0.34，和以后的版本）默认设置不工作必须打开使用：

```
echo "1" > /proc/sys/net/ipv4/ip_forward
```

实际上，ipfwadm-wrapper 是一种包装，因为它让我用旧的方法设置使用新的内码的防火墙策略。象这那样做可以更好一点：

```
ipchains -P forward DENY
ipchains -A forward -s 192.168.1.0/24 -j MASQ
```

如果你想在系统启动时就执行这个命令，且加到文件 /etc/rc.d/rc.local 的末尾。这文件是象 DOS 里的 AUTOEXEC.BAT。推荐阅读手册和其他文件，看看这些命令做什么，有什么其它的选择：

```
less /usr/doc/HOWTO/mini/IP-Masquerade
man ipfwadm
man ipchains
```

在 REDHAT 7.1 (KERNEL 2.4.x) 防火墙能被设置使用新的 iptables 命令。假如你不同时用 iptables，你能还使用老 ipchains。也可用 ntsysv 确定 ipchains 有效和 iptables 无效。

4.5.5 怎样使用 Samba？

Samba (smb) 是为 Linux-MSWindows 互连而设计的程序。它使一台 Linux 计算机伪装成一台 MS Windows NT 服务器，这样让你的基于 Windows3.1/95/98 计算机连接到网络。Samba 不仅仅替换 WinNT——它做得比 WinNT 更好！

我的家庭网络上的 3 台机器之中的 1 台是 Linux/Win95 双重启动。我这样配置我的网络，如果我引导 Win95，则另一台 Linux 扮演 Samba 服务器的角色。用这种方法 Win95 机器可以使用网络打印机，共享 Linux 目录，远程登录，或通过 Linux 的调制解调器浏览 Internet 等等。我也能从 Linux 计算机上存取 Win95 机器上的文件。

Samba 只能配置在一台 Linux 机器上。

首先，以 root 权限运行 setup，从菜单中选择"System Services"(RH6.0)或"ntsysv"(RH5.2)，并确定启动了"smb"服务。（按键 F1 可获得程序 setup 的有关帮助。万一你对不同服务的作用感到好

奇,也可以在帮助里找到答案。)如果 smb 不在列表中,可能是在安装 RedHat 时没有安装。将 RedHat CD 盘放进 CDRom,加载 CD,以 root 权限运行 glint(RH5.2)或 gnormp(RH6.0)安装程序,在系统中加入 Samba 服务。

接下来开始配置 Samba。配置相对来说比较简单,因为所有的 Samba 配置全都在一个注解清晰的文件/etc/smb.config 里完成。最少的设定包括指定工作组名。注意如果你没有填充"host allow"选项,所有的主机都是允许的,这对家庭网络是可行的。RedHat CD 提供的/etc/smb.conf 文件样本中的其他选项也可能需要调整以满足你的特别要求,但我什么也没有改动,我的 Windows 机器就可以连到我的 Samba 服务器上。

检查文件/etc/smb.conf 的选项:

```
domain master = yes
domain logons = yes
```

确定它们已启用。你可能需要他们。

在 RH6.0 系统中,你也可以用下面的命令对 samba 进行一些基本的设定(以 root 权限):

```
netconf
```

然后是对 MS Windows 95(或其他系统)进行网络配置。这一步和连接到 WinNT 服务器的配置完全一样。确定网络功能("client for Microsoft Network")已启用并填加了工作组名。当启动 MS Windows 时用你的 Linux 用户名和密码登入网络。点击“网络邻居”(Network Neighborhood)图标,如果你的 Linux 服务器已在列表中,说明已成功登录。进一步可以看到你在文件/etc/smb.conf 中选择要共享的目录和打印机列表。

如果看不到或不能正常使用公共目录,确定目录已创建并为所有用户设定了正确的读写权限。

MS Windows 95b 及以上版本(95c, 98 和新的 NTs)可能使用了密码加密技术。这会引起 Window 机器登录失败,这时你需要在文件/etc/smb.conf 启用选项:

```
encrypt passwords = yes
```

在 Linux 服务器端,你可以用以下命令启动,停止,重新启动和检查 Samba 的状态(以 root 权限):

```
samba start
samba status
samba restart
samba stop
```

修改文件/etc/smb.conf 之后需要重新启动 samba。

如果你已通过 Samba 与你的 Linux 计算机相连并且 Linux 机器已连在 Internet 上,你就可以用 Windows 版的 Netscape 游览网页了。为实现之,在带有调制解调器的 Linux 机器上必须已设定 IP 伪装(详见[这里](#))并且在文件/etc/smb.conf 中已启用了 Samba dns 名称解释选项:

```
dns proxy = yes
```

然后通知 Windows 启用 dns 服务器, 指定 Linux 服务器名和 IP 地址(在控制面板中选择 Network-TCP/IP)。

你也可以将远程的 Windows 目录加载到你的 Linux 文件系统中。具体操作见这里。

以上仅仅介绍了最基本的 Samba 设定。更多资料请参考：

```
man samba; man smb.conf; man smbclient; man smbmount
less /usr/doc/HOWTO/SMB-HOWTO (RH5.2 系统)
documentation in /usr/doc/samba-2.0.3/doc (RH6.0 系统)
http://www.sfu.ca/~yzhang/linux/samba/ (samba minihowto)
http://www.germanynet.de/teilnehmer/101,69082/samba.html
```

也可以研究文件/etc/smb.conf。

4.5.6 Sendmail

Sendmail 是 Internet 标准的邮件处理系统, 也是 RedHat Linux 缺省的邮件处理系统(Mandrake 使用 PostFix)。作为用户, 你不会直接使用 sendmail -- sendmail 运行在计算机的后台, 是管理所有用户邮件的主服务器引擎。在文本模式下, 你可选择 pine 或 elm 来阅读/发送邮件(选一个然后一直用它 -- 同时管理两个独立的邮箱会很不方便)。在 KDE 下, 你可从 K-menu 中选用"mail client"(kmail)。如果想通过 ppp 拨号与外界通信, 你可能更想用内建在 Netscape 中的 mailer 与远程的基于 Internet 服务提供商的邮箱直接联系(跳过本地计算机的邮件服务器)。

Sendmail 非常灵活且功能强大, 但如果要定制它以适应你的特别要求的话, 管理起来将会非常困难。幸运的是 Red Hat (5.2 或 6.x) 自带了封装的 sendmail (虽然有某些限制)。

在我的家庭系统(Red Hat 缺省安装)上, 可以无问题地发送邮件给在同一台机器的另一用户(例如用 pine)。一旦我通过调制解调器与我的 Internet 服务提供商相连([启用 IP 伪装](#)), 就可以用任何邮件软件发送邮件到本地或世界上任何一个地方。但当我没有连上 Internet 时, 所有邮件都会排在队列中并等待 Internet 连接, 即使邮件是发送给我的家庭网络中的另外一台计算机(小小的烦恼)。连上 Internet 后邮件即可发送出去。(这种情形之所以发生是因为 sendmail 在尝试寻找 DNS, 而我的系统却没有 DNS -- RedHat 缺省的 DNS 只是缓存。)如果确实想在 RH6.x 上避免这种情形, 你可以运行 netconf(以 root 权限)指明 sendmail 根本不使用 DNS(RH5.2 中的 Linuxconf 没有此项选择)。我自己使用 DNS。

在 RedHat 6.x 系统中, 你还可以调用命令 netconf 对 sendmail 的其他选项进行配置(以 root 权限)。在“邮件传递系统”——“中继主机”("mail deliver system"- "relay to hosts")列表中, 我加入了我的 Internet 服务提供商的名字和家庭网络上其他机器的名字。我还把我的家庭网络中的机器名字加在“按名字中继”("relay for by name")的列表中。

如果从家庭网络发送电子邮件(例如用 pine)到外部世界时, 必须了解因简单安装而带来的某些限制。例如在你创建了自己的 IP 地址并且你的域名还没有注册的情况下, 你将不可能收到回复的邮件。在发送邮件中给出“回复”地址"user_login_name@machine.domain"是不行的, 因为根据任何外部的 DNS 你的域名都不存在, 所以没有任何途径让邮件到达你手里。为解决这个问题, 你可以用 Netscape 中的邮件程序 mailer 与家庭网络的外部世界通信。Netscape 使用你在它的“编辑喜好”中的设定与你的基于 ISP 的邮箱(位于 ISP 已注册的服务器上)直接通讯从而跳过基于你没有注册的

家庭网络的电子邮件系统)。另一种可行的办法是在“回复到”("reply to")一栏中指定正确的地址。如果是在 Netscape 中,你可以设定 NetscapeMail 使用你本地的 Linux 计算机作为邮件服务器,这样的话你也可以从 NetscapeMail 发送邮件到家庭网络中的计算机(不仅仅发送到外部世界)。选项“回复到”("reply to")同样可以在 KDE"mail client"的设置里设定。在"pine"中,你可以在"setup-configure", "customized-hdrs"下输入下面类似的东西来指定“回复到地址”:

```
Reply-to: joe@joe_net.net
```

其中"joe@joe_net.net"是你的正确的回复电子邮件地址。

如果你的确希望在没有 Netscape 的帮助下从基于 ISP 的邮箱中取邮件到你的帐户,你可以考虑安装 fetchmail(参见 man fetchmail)。

4.5.7 简单的 WEB 服务器 (Apache 阿帕奇)

在 RedHat 6.0 系统中安装一简单的阿帕奇 web 服务器非常容易,我是在安装 RedHat 时已安装了阿帕奇 web 服务器的 rpm 软件包。如果你还没有安装,请把 RH CD 盘放入 CDROM 中,加载 CDROM,然后安装 apache-*.rpm 软件包(“*”是对应的软件版本号和平台)。如果已知要安装什么,做起来则非常简单(以 root 的权限):

```
cd /mnt/cdrom/RedHat/RPMS
rpm -ivh apache*.rpm
```

接下来,运行“setup”(以 root 权限)并确定已启用 httpd daemon(在"system services"下)。[守护程序(A daemon)是位于后台的程序,当需要的时候就会被激活。对于 httpd,如果有人访问你的 http 服务器,则会激活它的 httpd daemon(httpd 守护程序)。]

现在,每次启动计算机时,httpd 都会自动开启。也可以用下面的命令手动开启它(以 root 权限):

```
/etc/rc.d/init.d/httpd start
```

关闭时使用命令:

```
/etc/rc.d/init.d/httpd stop
```

系统中应该已有目录:/home/httpd(检查它是否存在),在它的下面是子目录 html—这是人们从 web 访问你的计算机时的“根目录”(“他们的根目录”意味着他们不可以访问你的系统中/home/httpd/html/之上的任何一层目录)。缺省情况下,这个目录中包含一些阿帕奇安装程序生成的 html 文件和手册。现在你应该可以从你的家庭网络中的另一台机器连结到你 web 服务器了。例如在 Netscape 的地址(“Location”)行输入:

```
http://my_http_server_name
```

就可以浏览阿帕奇的操作手册。

想把自己的东西放在 web 服务器上, 先把阿帕奇的样本文件移到其他地方, 然后把自己的 html 文件(要公开给大众的)复制或连接到/home/httpd/html/目录。(不要忘记包含文件 index.html, 当人们访问你的服务器时, 首先显示这个文件的内容。)

阿帕奇的配置和记录文件位于/etc/html/目录下, 你可以显示它们或对它进行定制。用下面这个简单的命令(以 root 权限)可以实时显示记录文件的内容:

```
tail -f /etc/httpd/logs/access_log
```

[命令 tail 通常显示一个文本文件的尾部。选项“-f”可使命令 tail 在记录文件随着时间增大时不断地显示文件的新尾部——很方便地监视记录文件并且可以知道谁登录了你的服务器。]

可用下面命令以图形模式下安装阿帕奇服务器(以 root 权限在一 X-window 窗口中。如果已从 RedHat CD 盘安装了“Comanche”):

```
Comanche
```

4.5.8 简单的 FTP 服务器

在旧的 RedHat Linux 系统中(RH<7.1)中, 安装一个 ftp 服务器再简单不过了——它的功能安全是封装的。因为在文件/etc/inetd.conf 中, ftp 被设定为一个标准的服务(像 telnet 和 gopher 一样)。以下是我的/etc/inetd.conf 文件中的有关内容:

```
ftp      stream  tcp     nowait  root    /usr/sbin/tcpd  in.ftpd  -l  -a
telnet   stream  tcp     nowait  root    /usr/sbin/tcpd  in.telnetd
gopher   stream  tcp     nowait  root    /usr/sbin/tcpd  gn
```

安装 ftp 服务器的第二步是在文件/etc/passwd 中定义 ftp 帐号:

```
ftp:*:14:50:FTP User:/home/ftp:
```

其中":"是字段分隔符。第一个字段是帐号(用户)名“ftp”, 第二个字段"*"是指禁用密码(任何人不可以用名字“ftp”登录), 用户标识是 14, 用户组标识是 50, "FTP User"是注解, 根目录是 /home/ftp, 最后一字段为空(对普通用户, 指定用户使用的 shell 名)。

因为 RedHat 已经帮我安装好了 ftp 服务器, 任何人都可以 ftp 我的计算机, 或者以一用户名登录(将提示输入密码并直接指向用户的根目录), 或者以匿名(anonymous)并给出电子邮件地址作为密码的方式登录。任何用户也可以在 Netscape 的地址("location")行输入:

```
ftp://my\_computer\_name
```

自动连接到 ftp 服务器(Netscape 负责发送匿名("anonymous")用户名和电子邮件地址作为密码)。

匿名("anonymous")ftp 用户被指定到/home/ftp 目录, 并作为他们的根目录(不可访问任何上层目录)。我把所有想提供的文件放在子目录/home/ftp/pub 下。

目录/home/ftp/bin 下存有远程用户可执行的命令。在我的系统中有: compress, cpio, gzip, ls, sh, tar, zcat; 权限全设为只可运行(111)。

目录/home/ftp/etc 下是所有令匿名用户正常工作的安装文件(edited passwd, group, ld.so.cache)。所有库文件放在目录/home/ftp/lib 下(我猜想这些库文件是运行那些匿名用户可执行的命令时要用到的。)

Red Hat 7.1 用 xinetd 代替了旧的 inetd, 并且缺省关闭了大部份的网络服务。如果你不能 telnet 到你自己的计算机或其他的网络服务被关闭, 可能要检查目录/etc/xinetd.d 中的文件并编辑名字为网络服务的文件, 以至文件中有: disable = no。这是基于安全因素的考——只选择必需的网络服务并启动它们。除非需要, 否则不要启动 ftp 服务—ftp 在过去有不少的安全隐患。如果启动了网络服务, 确保在文件/etc/hosts.allow 和/etc/host.deny 加入了适当的安全设定。

样本文件/etc/xinetd.d/tftp 显示服务 tftp 被关闭:

```
{
    disable = yes
    socket_type      = dgram
    protocol        = udp
    wait            = yes
    user            = root
    server          = /usr/sbin/in.tftpd
    server_args     = -s /tftpboot
}
```

你如果没有这个文件, 用命令 cat 检查你已安装了那些服务。如果是用命令 setup(以 root 权限)启用了启动程序, 那么大多数服务的守护程序(daemon)在系统启动时都会自动开启。

4.5.9 如何通过电话线外部远程访问我的计算机?

唯一的困难是由于 IP 地址是 Internet 服务提供商 (ISP) 从他们的 IP 地址池中动态分配给你的, 所以你每次连网时 IP 地址都不相同(除非你与你的 ISP 有特别的安排)。而要从外部世界 telnet, ftp, 或访问你的网页(由你的阿帕奇服务器提供), 对方必须知道你当前的 IP 地址。为找出你当前的 IP 地址, 可用以下“界面配置”命令(运行时不需要任何参数)显示你机器上所有活动的网络界面信息:

```
/sbin/ifconfig
```

在我的机器上运行这个命令显示三段信息: eth0(第一个以太网网络界面, 连结到我的家庭网络中的其他计算机), lo(环路测试专用界面, IP 地址为。必须出现在每一台机器上), 和 ppp0(第一个点对点协议界面)。ISP 分配给我的当前 IP 地址显示在 ppp0 标题下。(你的 Linux 机器可同时被指定多个 IP 地址, 所以如果你有一静态的用于家庭网络的 IP 地址, 它仍然是有效的但只有在家庭网络中是可见的。)

一旦知道了 IP 地址, 我就可能通过 ICQ 或电子邮件把它发送给我的朋友。他们就可以 telnet, ftp 或用其它方式连接到我的计算机(当然在我的机器上必须有他们的帐号)并且在我的 Linux 机器上运行程序, 也可以在器的地址(“location”)行输入 http://my_ip_address 来浏览我的家庭 web 网页, 等等。如果某个朋友的本地机器上有 Xwindows, 他/她甚至可以在我的服务器上运行 GUI 程序, 而把结果显示在他/她的计算机上。

我也可以写一个当我连上 Internet 时会自动通知我朋友脚本, 然后把脚本名加在 kppp-setup-account-edit-dial-"execute program upon connect"(连接时执行程序)中。下面的脚本是在我上

班的时候，会通知我家里有没有人上网。(我把脚本输入在一个文本文件中并用命令 `chmod o+x file_name` 把它变为可执行文件)：

```
#!/bin/bash
sleep 15
/sbin/ifconfig | mail -s notification my_email_address
```

脚本的第一行告诉计算机把这个文件解释成 bash shell 脚本。第二行让脚本等待 15 秒(只是为确定完全建立 ppp 连接之前不要发送电子邮件)。第三行执行命令 `ifconfig` 并将结果转给邮件程序 (mail)，然后邮件程序以主题“notification”将结果发送到我的电子邮件地址 (my_email_address)。

一种更灵活的远程访问家庭计算机的方法是把它配置成一台拨号的 ppp 服务器(相对于连接 ISP 时使用的拨号客户)。如果某个人知道怎么做，请打个电话给他/她吧。

总而言之，与严格控制计算机通信量的 MS Windows 3.x/95/98 不同，Linux 是面向网络的，这使的各种各样的网络连接(不管是连入还是连出 Linux 计算机)都非常容易。强大的网络功能一般被认为是 Linux 的强项，但从一个真正新手的角度看，也会是一个问题(参见下面的问题)。

4.5.10 家庭计算机是否会遭黑客攻击吗？

很不幸，这是完全可能的而且攻击家庭计算机的尝试是非常普通的。每当你连上你的 Internet 网络提供商 (ISP)，你就处在危险之中。如果你想知道为什么有这种可能性，可看上一个问题。明显地，如果你的 Internet 连接是永久性的(例如：电缆调制解调器)，危险性更大，反之，如果连接是短暂的(像典型的短时间的，通过电话调制解调器连接)，危险性就会少许多。

真正的危险是如果入侵者以任何帐号登录了你的机器，就可能找到(知道)“本地安全开发” (“local security exploit”)并获取 root 的权限。如果你是一个新管理员并且/或者你的机器不是真正面向安全的，这种可能非常大(你在家，不是吗？--你想，谁会担心机器的安全性！)。

为了保护你自己，永远不要让陌生人登录你的计算机。为所有帐号设定相对长的既有数字又有字母的密码。时不时地更换密码。加强所有用户密码策略的最好办法是运行(以 root 权限，在 RH6.0 下) `linuxconf`，在“密码和帐号策略” (“password and account policies”)中，把密码的最少长度改为 6 个或更多字符，最少的非字母字符的长度改为 1 个或 2 个，必须更换密码的天数改为 90 天或更少，并且预设一条密码过期的警告信息，例如：还有 7 天过期。查阅这里 [here\(FAQ2.htm#pass_security\)](#) 以获取更多有关差密码的资料。永远不要创建一个没有密码或密码特别愚蠢的帐号。不要习惯性地总是以 root 的权限在你的计算机上工作 -- 如果你运行一个已知有完全漏洞的程序，别人就会找到攻击你的途径。旧版本的 Linux 有一些已知的安全漏洞，所以如果你想让不信任的人登录你的计算机，或者你要运行一些服务器端的网络服务(例如：ftp 或 http 服务器)，使用更新版本的 Linux。

另一个很好的主意是时不时地筛选那些包含所有登录你的计算机的记录的文件：

```
/var/log/secure(最新的记录), /var/log/secure.1(旧的记录), /var/log/secure.2(更旧的记录), 等等。你可能还想浏览位于目录/var/log 下其它有用的记录文件，不定期地检查它们。最典型的“警告”信号是计算机的端口被扫描：同一个 IP 号不断的请求连接到你的系统的 telnet, ftp, finger 和其它端口—某个人尝试获取你的系统资料。
```

如果你从不远程连接到你家里的 Linux 机器，那么在家庭网络中限制使用连接到你机器上的服务器端的网络服务的权力是一个非常好的主意(文件/etc/inetd.conf 列出了所有的网络服务)。我们可以用两个文件来控制：/etc/hosts.allow 和/etc/hosts.deny。这两个控制访问的文件的工作方式如下：当收到外部的连接请求时，先扫描文件/etc/hosts.allow，如果连接请求的机器名相匹配，则授予访问的权限(与文件/etc/hosts.deny 中任何输入项没有关系)。否则，扫描文件/etc/hosts.deny，如果机器名相匹配，则关闭连接。如果两个文件中都没有匹配的机器名，就会授权给请求连接的机器。

B. Staehle(一位 Linux 调制解调器专家)写信给我建议不要安装任何网络服务。“如果没有正确配置网络服务，一些小的脚本程序就会破坏你的计算机。一个新手永远不要向外界开放网络服务(例如：ftp, telnet, www)。如果你一定要装，确保只让你可控制的系统建立连接。文件/etc/hosts.deny 应包含

```
ALL: ALL
```

并且文件/etc/hosts.allow 中应该只有

```
ALL: 127.0.0.1
```

只允许被定义的主机连接。不要使用主机名！” <Bill 的建议到此为止>。

我的/etc/hosts.deny 文件的确跟上面的建议一模一样(ALL: ALL)，但文件/etc/hosts.allow 中加了另外两台我信任的计算机，它们可连接到我所有的网络服务，还有另外一台可访问 telnet 和 ftp(所有的 IP 号都是假的)：

```
ALL: 127.0.0.1, 100.200.0.255, 100.200.69.1
in.telnetd, in.ftpd: 100.200.0.2
```

上面的样本中“ALL: ALL”代表“所有的服务：所有的主机”，也就是来自“任何主机”到“任何本地网络服务”的连接。

更多资料，可查阅“Linux 网络管理员指南”(“Network Administrator Guide”)，它在你的 RedHat(或其他版本)的 CD 盘。我打印出来并精装了这本书。

为了验证你的计算机对外界提供了哪些服务，可利用基于 web 的工具。去到 <http://scan.sygatetech.com/>，然后点击“scan now”(“现在扫描”)。

以下以另外一些可扫描你的系统的地址：

```
http://crypto.yashy.com/
http://davidovv2.homestead.com/freetoolsservices.html
http://privacy.net/
http://scan.sygatetech.com/
http://security1.norton.com/us/intro.asp
http://suicide.netfarmers.net/
http://trojanscanner.com/cgi-bin/nph-portscanner
http://www.doshelp.com/dostest.htm
http://www.dslreports.com/secureme/
http://www.dslreports.com
```

```
http://www.earthlink.net/freescan/  
http://www.grc.com  
http://www.hackerwhacker.com/  
http://www.nessus.org  
http://www.netcop.com/newscan/fullscan.html  
http://www.privacyscan.org  
http://www.sdesign.com/cgi-bin/fwtest.cgi  
http://www.sdesign.com/securitytest/index.html  
http://www.securityspace.com/  
http://www.vulnerabilities.org/nmapemail.html  
http://grc.com  
http://www.dsireports.com/scan  
http://www.dsireports.com/security/sec025.htm
```

为安全起见，另外一个好主意是不要发布你正在使用的操作系统及其版本号。我把文件/etc/issue和/etc/issue.net的内容：

```
Red Hat Linux release 6.2 (Zoot)  
Kernel 2.2.14-5.0 on an i586
```

换成类似下面的文字：

```
WARNING: THIS IS A PRIVATE NETWORK  
UNAUTHORIZED USE IS PROHIBITED AND ALL ACTIVITIES ARE LOGGED  
IBM S/390 LINUX
```

这样做把幽默和增加安全性结合在了一起(我希望)。

文件/etc/issue和/etc/issue.net的内容在系统每次重新启动时都会重写(当脚本/etc/rc.local在运行时)。所以为保证更改之后不会再变，我把这些文件改为对所有用户都是只读的(以root权限)：

```
chmod a=r /etc/issue*
```

替代上面的命令，我也可以编辑(以root权限)脚本/etc/rc.d/rc.local，用###注解掉其中的5行，有关的部分变为：

```
# This will overwrite /etc/issue at every boot. So, make any changes  
# want to make to /etc/issue here or you will lose them when you reboot  
### echo "" > /etc/issue  
### echo "$R" >> /etc/issue  
### echo "Kernel $(uname -r) on $a $SMP$(uname -m)" >> /etc/issue  
### cp -f /etc/issue /etc/issue.net  
### echo >> /etc/issue
```

还有一个好的安全手段是禁用ping。Ping是类似声纳的请求，当收到另一台计算机的请求时会将请求发回。对安装，查错，或探测你的机器是否在网上非常有用。但也可以通过不断地发送ping请求("ping of death")来探测或攻击你的机器。我使用IP伪装禁止我的机器对来自自己网络的pinging做出回应。我对取自网页

<http://www.securityfocus.com/focus/linux/articles/linux-securing2.html> 的下面的命令和注释进行了少少的修改：

```
ipchains -A input -p icmp --icmp-type echo-request -i ppp0 -j REJECT -l
      (1) (2)   (3)           (4)                   (5)   (6)   (7)
```

命令 ipchains 的标志解释如下：

1. (A)添加一新规则。
2. 规则作用于的链(chain)，样本中规则将作用于 ingress(输入)包。
3. (P)规则作用于的协议。样本中为 icmp。
4. ICMP 类型，样本中所有的 icmp 回应请求都会被封锁。“ICMP 回应”指 ping。
5. 界面名。样本中是第一个通过电话的连接，ppp0。
6. 目标，或如何处理有问题的包。
7. 在系统记录文件中记录所有匹配规则条件的包。

IP 伪装在本指南的 IP 伪装章节有详细的描述。

我采取的另外一项安全预防措施是：时不时地检查一下是否有人在我的系统上安装了“root 套件”。我用的是“chkrootkit”实用程序(很小，只有 25k，可从 <http://www.chkrootkit.org/> 下载)。下载 tarbal 后，进行以下操作：

```
su [提供密码]
cd /usr/local
tar xvzf /home/my_name/chkrootkit.tar.gz
cd /usr/local/chkro<tab>
make
./chrootkit
```

最后一条命令实际上是在系统上寻找 rootkit。“Rootkit”是一隐藏的软件，一旦某个人获取了访问你系统的权限(像“root”一样)，他/她就可以安装这个软件来监听，监视，保护访问，等等。

第 5 部分: Linux 的快捷键和主要命令

Linux 新手管理员指南(中文版)

LINUX NEWBIE ADMINISTRATOR GUIDE (Simplified Chinese Version)

中文版本 1.0 由幻影快递 Linux 小组(SLT-ShadowFax Linux Team) 翻译整理

Linux Newbie Administrator 英文原版版权所有: Peter and Stan Klimas.

<http://sunsite.dk/linux-newbie>

中文版本版权所有: 幻影快递 Linux 小组 ShadowfaxLinux@hotmail.com

本章内容:

- 5.1 Linux 基本的键盘输入快捷键和一些常用命令
- 5.2 帮助命令
- 5.3 系统信息
- 5.4 基本操作
- 5.5 文件管理
- 5.6 查看和编辑文件
- 5.7 查找文件
- 5.8 X-windows 基础
- 5.9 网络应用程序
- 5.10 文件压缩和解压缩
- 5.11 进程控制
- 5.12 一些管理命令
- 5.13 硬盘和软盘实用工具
- 5.14 用户帐户和文件权限管理
- 5.15 程序安装
- 5.16 访问磁盘驱动器/分区
- 5.17 网络管理工具
- 5.18 与音乐有关的命令
- 5.19 与图形相关的命令
- 5.20 几个小游戏

简介:

这是包含在Linux软件包里 (RedHat 或者Mandrake), 我们比较经常使用和比较有用的一些命令集合。在你当前路径的命令提示行下按<TAB>键, 就可以看到有效的命令一览。在我的小型家用系统上, 在我的当前路径下有3786个可执行命令。这些命令的大部分可以在图形用户界面下使用鼠标的左键或者右键来访问, 全部都可以使用命令行来运行 (除非你没有安装对应的软件包, 在我的CD里都可以运行)。那些要求图形用户界面的程序必须在图形用户下才能运行, 比如, 你可以通过在KDE或者GNOME的终端输入命令来运行这些程序。一些更高级 (对新手没有用处?) 的命令工具可以

参考第七部分的相关内容。

以下给那些对Linux没有概念的新手：

1. Linux是大小写敏感的系统，举个例子，Mozilla, MOZILLA, mOzilla和 mozilla是四个不同的命令（但是只有第四个mozilla是真正有效的命令）。还有，my_file, my_file,和 my_FILE是三个不同的文件。用户的登录名和秘密也是大小写敏感的（这是因为UNIX系统和C语言的传统一向是大小写敏感所致）。
2. 文件名最多可以有256个字符，可以包含数字，点号“.”，下划线“_”，横杆“-”，加上其他一些不被建议使用的字符。
3. 文件名前面带“.”的文件在输入“ls”或者“dir”命令时一般不显示。可以把这些文件看作是隐含文件，当然你也可以使用命令“ls -a”来显示这些文件。
4. “/”对等于DOS下的“\”（根目录，意味着所有其他目录的父目录，或者是在目录之间和目录和文件之间的一个间隔符号）。举个例子，cd /usr/doc。
5. 在Linux系统下，所有的目录显示在单一目录树下（有别于DOS系统的驱动器标识）。这意味着所有的物理设备上的所有文件和目录都合并到单一的目录树下。
6. 在配置文件里，以#打头的行是注释行。在修改配置文件的时候尽量不要删除旧的设置可以把原来的设置加上“#”变成注释行，总是在修改地方对应地加入一些关于修改的注释，你会发觉在以后的管理中获益多多。
7. Linux是继承性的多用户操作系统。你的个人设定（和其他用户的个人设定）放在你的主目录下（一般是/home/your_user_login_name）。许多的配置文件的文件名都以“.”开头，这样用户一般看不到这些文件。
8. 整个系统范围的设定一般放在目录/etc下。
9. 和其他的多用户操作系统类似，在Linux下，文件和目录有自己的拥有者和访问权限。一般来说，你只被允许文件到你的主目录下（/home/your_user_login_name）。学习一些关于文件权限管理的相关知识，否则你肯定会觉得Linux实在很麻烦。
10. 命令参数选项一般由“-”引导，后面跟一个字符（或者“--”，当选项超过一个字符时）。这样，“-”有点象DOS下的“/”。举个例子，输入命令 rm --help。
11. 输入command&（命令名后面加&符号）在背景模式下运行一个命令。这个一般用来在X窗口下启动其他的程序。

5.1 Linux 基本的键盘输入快捷键和一些常用命令

<Ctrl><Alt><F1>

切换到第一个文本终端。在Linux下你可以有多达六个不同的终端。这个命令的意思是：“同时按

住<Ctrl>键和<Alt>键，然后按<F1>键，再释放所有的键”。

<Ctrl><Alt><Fn> (n=1..6)

切换到第n个文本终端。（你也可以使用不是很经常用到的命令chvt n 来实现，n指的是第n个文本终端）。在文本终端下（不是在X窗口），你也可以简单使用<ALT><Fn>来实现切换，不需要<CTRL>键。

打印出你正在使用的终端名称，如果你希望知道终端的名字，可以使用命令fgconsole。

<Ctrl><Alt><F7>

切换到第一个图形用户界面（一般来说X-window在第七个终端）

<Ctrl><Alt><Fn> (n=7到12)

切换到第n个图形用户界面。根据缺省，第一个X-Window在第7个终端运行，从第8到第12什么也没有，当然你可以逐个启动这些图形用户界面。

<Tab>

（在文本终端下）可以使用TAB自动完成命令，或者显示所有的可选项。这个快捷键真的非常好用，经常使用你会发觉它可以节约你很多的时间。

<ArrowUp>

（在文本终端或者X窗口下）滚动和编辑以前输入的命令。按<ENTER>执行一个历史命令

<ArrowDown>

回滚

<Shift><PgUp>

滚动终端输出。对于登录提示也起作用，所以你可以使用它回滚启动信息。你显卡的内存大小决定你可以回滚多少内容

<Shift><PgDown>

回滚终端输出

<Ctrl><Alt><+>

（在X窗口下）改变X服务器的屏幕解析率（如果你设置X服务器有多个不同的屏幕解析率）。比如对于我的标准SVGA卡和显示器，在文件/etc/X11/XF86Config有以下的设置行：（从缺省开始，到可以支持的最大虚拟屏幕解析率）

```
Modes "1024x768" "800x600" "640x480" "512x384" "480x300" "400x300" "1152x864"Z
```

当然，首先我必须设置我的X服务器，可以使用using Xconfigurator, xf86config, 也可以手工编辑文件：/etc/X11/XF86Config。XFdrake (Mandrake使用图形用户界面进行配置)。你也可以参考命令xvidtune和xvidgen。

<Ctrl><Alt><->

(在X窗口下)把X服务器的屏幕解析率修改到上一次的设置。

<Ctrl><Alt><Esc>

(在X窗口, KDE下)关闭我鼠标将要指向的窗口(鼠标的光标形状会有所改变)。同样的效果也可以使用命令xkill(在X终端上)来实现。当一个程序窗口被挂住的时候特别有用。

<Ctrl><Alt><BkSpc>

(在X窗口下)终止当前的 X窗口服务。如果X窗口不能正常退出时可以使用。

<Ctrl><Alt><Delete>

(适用于文本终端下)关机 and 重新启动。这是一个在文本终端下的正常关机命令，千万不要按计算机上的reset键来重新关机和重新启动！

<Ctrl>c

终止当前进程(对于一般的小型文本模式的应用程序)

<Ctrl>d

(在一个空白的命令行上输入)退出当前的终端。参加下一个命令。

<Ctrl>d

给当前的进程送文件结束符合。不要按两次否则你会把自己退出系统。

<Ctrl>s

停止终端传输

<Ctrl>q

从新开始终端传输。如果你的终端突然莫名其妙的停止响应，可以参考上一条命令。

<Ctrl>z

把当前进程送到后台处理。

exit

退出系统。我也可以使用logout命令达到同样效果。(如果你启动了第二个shell程序,比如,使用bash,这条命令将使你退出第二个shell回到第一个shell,但是不会退出系统。当然,可以再一次使用exit退出)

reset

恢复崩溃的终端(有些终端显示一些奇怪的字符)到缺省的设置。当你使用cat命令准备显示一个二进制文件时,你可能看不到你所输入的命令,尽管命令仍然照常工作。

<鼠标的中间键>

粘贴当前选择的文本。这是一个常规的Linux“复制 - 粘贴”操作。它是对被广泛采用的通信用户界面下“复制 - 粘贴”操作的一个非常有用的扩展(但是它不支持旧版本的NetScape,因为在旧版的NetScape只能使用MAC/Windows风格的“复制 - 粘贴”操作。如果你运行“setup”程序进行设置,你可以在文本终端下使用这个功能。而且它在大多数的对话框里也能很好的工作,真的非常方便!)如果你有一个“Linux-Ready”的鼠标(logitech或者其类似产品)或者你通过使用“模拟三键鼠标”,这个功能可以得到最大限度的发挥。一般来说,“第三键”的模拟是通过同时点击左右两个鼠标键来实现。

~

(水文符合) 我的主目录(一般来书是/home/我的登录名)。举例,命令~/my_dir将修改我的工作目录到我的主目录的子目录“my_dir”下。相对于输入“cd ~”,你也可以只输入“cd”,我把我的所有文件都放在自己的主目录下。

.

(点符号) 当前目录。举例: ./my_program 将试图执行当前目录下的“my_program”文件

..

(两个点)到当前的上一级目录(也叫“父目录”)。举个例子,命令“cd..”将修改我的当前工作目录到上一级目录。

一些KDE的快捷键(很有用,但不是非要不可的)

<Alt><Tab>

切换不同的程序窗口。往回切换,使用 <Alt><Shift><Tab>

<Ctrl><Tab>

切换不同的桌面。往回切换,使用 <Ctrl><Shift><Tab>

<Ctrl><Esc>

显示我系统里正在运行的进程列表。允许我终止那些由我启动的进程(或者送信号给这些进程)

<Alt><F1>

访问K菜单(对等于微软Windows的的“开始”菜单)。

<Alt><F12>

使用键盘上的箭号键模拟鼠标的操作

<Alt><鼠标左键>

拖曳并移动一个窗口。一般来说，我通过拖曳窗口的标题栏来移动一个窗口。有些时候，我用它把一个窗口移动到屏幕之外。使用这个功能，我可以把窗口移动到屏幕的任何位置。

<Alt><PrintScreen>

把当前屏幕的快照存到剪贴板

<Ctrl><Alt><PrintScreen>

把当前桌面的快照存到剪贴板

<Ctrl><Alt><I>

锁定桌面

<Ctrl><Alt><d>

切换隐藏/显示桌面的功能(当老板走过来的时候迅速藏起你的纸牌接龙游戏非常管用哦！)

<Alt><SysRq><command_key> (非必须掌握)

这是在Linux的内核(底层内核)上支持的一组组合键。它意味着这些组合键在大部分情况下都是有效的。组合键主要是针对开发人员的程序调试目的或者在紧急场合下才使用；你也可以使用其他的办法，安全第一。<SysRq>键在PC上指的是PrintScreen键。组合键的功能可以通过设置相对应的内核参数来激活或者取消，比如：`echo "1" > /proc/sys/kernel/sysrq`

<Alt><SysRq><k>

终止在当前虚拟终端上运行的所有进程(包括X窗口)。这个组合键被称为“安全访问键”(SAK)。

<Alt><SysRq><e>

发送 TERM 信号(结束信号)到除了init进程以外的所有运行进程，要求其退出

<Alt><SysRq><i>

发送 KILL (终止信号)到除了init进程以外的所有运行进程，这个组合键比上一个更加有效，但是也可能引起一些程序被异常终止。

<Alt><SysRq><l>

发送KILL命令到所有运行进程(包括init)，系统将不再能够使用

<Alt><SysRq><s>

在所有的挂载文件系统上运行紧急同步(缓存写)。这可以保护数据丢失

<Alt><SysRq><u>

以只读模式重新挂载所有已加载的文件系统。这和上面的同步组合键有相同的功能，但是另外有一个好处：如果操作顺利完成，下一次硬件重新启动时，fsck将不会重新检查所有的文件系统。

<Alt><SysRq><r> 关闭键盘的原始模式。当你的X窗口程序挂住没有响应时特别有用。使用这个组合键之后，你可能会接着使用<CTRL><ALT>程序启动。

<Alt><SysRq>

不进行同步和卸载就立刻重新启动系统。之后你可能会碰到一下错误信息。

<Alt><SysRq><o>

关机 (如果正确设置以后)

<Alt><SysRq><p>

导出当前的寄存器和标志到控制台

<Alt><SysRq><t>

导出当前任务和相关信息到控制台

<Alt><SysRq><m>

导出内存的内容到控制台

<Alt>SysRq><数字>

“数字”从0到9。设置控制台记录的级别，用于控制哪一些内核信息将被显示在控制台上。举例，“0”意味着只有象PANIC和OOPS的之类的紧急信息才被显示在控制台上。

<Alt><SysRq><h>

显示帮助信息。还有，任何其他不被支持的 <Alt><SysRq><key> 组合键将显示同样的帮助。

5.2 帮助命令

命令名 -help | more

显示一个简略的命令帮助 (对大部分命令有效)。举个例子，试着使用 “ cp -help | more ”。“ --help ” 和和DOS下的 “ /h ” 开关功能类似。当输出超过一个屏幕时，加上 “ more ” 是很有必要的。

man **命令名**

显示对应命令系统的帮助手册。输入 “ q ” 退出浏览器。如果你设置了高级选项，试着输入 “ man man ”。命令 “ info **命令名** ” 和命令 “ man **命令名** ” 功能相似，但是包含更多的最新信息。帮助手册对于新手可能有点难于读懂——因为它们最初是写来给UNIX程序员看的。使用 “ **命令名** -help ” 可以得到一个简略容易消化的命令帮助。有些程序自带README文件或者其他帮助信息文件——建议你看看目录 /usr/share/doc。在指定的部分显示命令帮助，可以使用这样的命令 “ man 3 exit ”，这个命令只显示exit命令帮助手册的 “ 第三部分 ”；或者使用命令 “ man -a exit ”，这个命令显示exit命令帮助手册的 “ 所有部分 ”。exit命令帮助的所有部分是：1 - 用户命令；2 - 系统调用；3 - 子调用；4 - 设备；5 - 文件格式；6 - 游戏；7 - 杂项；8 - 系统管理；9 - 新内容。打印完整的命令帮助，可以使用命令 “ man **命令名** | col -b | lpr ” (可选项 col -b 删除所有的退格键和一些难以阅读的特殊字符)。

info **命令名**

显示指定命令的帮助信息。info 命令是取代man 命令的一个好处是，它通常带有最近更新的系统资料。多使用 “ 空格键 ” 和 “ 退格键 ”，否则你可能会晕头转向。按 “ q ” 键退出。如果觉得用于翻阅的办法不太好用，你也可以试着使用pinfo命令，看你会不会更喜欢这个替代品。

apropos 命令名

对所输入的命令名给出一个帮助一览表。

whatis 命令名

给出匹配所输入命令名的简短清单。whatis命令有点象apropos 它们使用相同的数据库。不同的是，whatis搜索的是关键字，apropos搜索的是关键字的具体描述。

help 命令名

显示bash shell内置命令的简单信息。使用 help 命令不带任何参数将显示bash shell所有内置的命令。最短的bash shell内置命令应该包括:alias, bg, cd, echo, exit, export, help, history, jobs, kill, logout, pwd, set, source, ulimit, umask, unalias, unset.

kdehelp
kdehelpcenter

(在X-终端下有效, 使用在你的系统上可以工作的那一个命令). 使用图形的浏览器显示整个系统的帮助。一般来说, KDE帮助通过把对应的图标放在KDE控制板上来实现。对等的GNOME帮助系统可以使用gnome-help-browser命令。

5.3 系统信息

pwd

输出工作目录。举例, 在屏幕上显示我当前所在的目录

hostname

输出本地主机的名称(我正在使用的这一台机器)。使用netconf修改机器的名称(要求超级用户权限)

whoami

输出我的登录名称

id username

输出用户标识ID(UID)和其对应的用户组标识ID(GID), 有效的ID(如果不同于真正的用户ID)和所属的其他用户组

date

输出操作系统的当前日期, 时间和时区。如果要以ISO标准格式输出, 我必须使用命令“date -lseconds”。

我可以修改当前的日期和时间到2003-12-31 23:57, 使用命令: date 123123572003。或者使用两个命令(比较容易理解和记住):

```
date --set 2003-12-31
date --set 23:57:00
```

如果要重新设置硬件时钟 (BIOS) , 可以使用命令 `setclock` , 要求超级用户权限。

`time`

侦测一个进程所需要的时间加上其他处理的时间的总和。不要和上面提到的 `date` 命令混淆。比如, 我可以使用命令 “ `time ls` ” 来判断显示一个目录需要多长时间; 或者我可以使用命令 “ `time sleep 10` ” 来测试 `time` 命令 (睡眠10秒钟, 什么也不做)。

`clock`

(两个命令中的一个) 从计算机的硬件 (由电池供应的实时的硬件时钟) 获得日期/时间。你可以使用这个命令来设置硬件时钟, 但是 `setclock` 可能简单一些 (看前面的两个命令)。举例: “ `hwclock --systohc --utc` ” 命令把系统时钟设置到硬件时钟 (以UTC格式)。

`who`

显示登录在系统里的用户

`w`

显示登录在系统里的用户, 检查他们正在做什么以及他们的处理器使用状况。属于常用的安全方面的命令。

`rwho -a`

(=remote who, 远端的who命令)。显示网络里其他计算机的用户登录状况。这个命令要求 `rwho` 服务必须在远端机器上运行。如果没有, 以 `root` 用户运行 `setup` (RedHat 特有) 激活 “ `rwho` ”。

`finger 用户名`

显示对于一个用户的系统信息。可以试一下命令: `finger root`。任何人都可以使用 `finger` 命令访问任何一台连接到因特网提供 `finger` 服务的计算机。比如: `finger @finger.kernel.org`

`last`

显示最后一个登录到系统的用户。经常运行这个命令作为系统安全检测的手段之一是一个绝好的主意。

`lastb`

(“ =last bad ” , 最后一个坏的) 显示最后一个不成功的登录尝试。但是这个命令在我的系统上不能工作, 所以你可能要使用: `touch /var/log/btmp`

“ `/var/log/btmp` 在一般的系统安装里无效的主要原因是因为: 它是一个包含登录的出错信息的完全可读的文件。一个用户登录时最经常犯的错误的输入用户密码作为用户名, 这样, 文件

/var/log/btmp对计算机黑客来说简直是一个礼物”。(谢谢Bruce Richardson提供这个信息)。

如果要解决这个问题, 修改该文件的文件访问权限为只有root用户才可以使用命令“lastb”。命令: “chmod o-r /var/log/btmp”

history | more

显示当前用户在命令行模式下执行的最后(1000个)命令。参数“|more”在输出满屏时暂停。如果要检查其他用户在你系统上都运行了什么, 以root用户登录, 检查在该用户主目录下的文件“.bash_history”(没错, 该文件可以被修改和删除)。

uptime

显示自从上一次启动到现在机器运行的总时间。

ps

(= “print status” 或者 “process status”, 打印状态或者处理状态) 列出由当前用户运行的进程一览。

ps axu | more

列出当前运行的所有进程, 包括那些不是来自控制终端的进程, 在显示用户进程的同时显示用户名。

top

持续列出正在运行的所有进程, 按CPU的占用率排序(占用率高的排在最前面)。按<CTRL>C退出。

PID = 用户标识
USER = 启动或者拥有该进程的用户。
PRI = 进程的优先级别(值越大, 优先级别越低, 一般进程是0, 最高的级别是-20, 最低的级别是20。
NI = 优化级别(比如, 进程试图以预先给定的优先级别的数值来运行)。数值越高, 进程的优化级别也越高(优先级别越低)
SIZE = 进程加载到内存的代码 + 数据 + 堆栈的字节总数(以千字节计算)
RSS = 物理内存被占用的大小(以千字节计算)
SHARE = 和其他进程共享的内存(以千字节计算)
STAT = 进程的当前状态 S-睡眠, R-运行, T-停止或者被跟踪, D-不可中断的睡眠, Z-不死的进程
%CPU = CPU占用百分比(自从上一次屏幕更新以来)
%MEM = 共享物理内存的占用率
TIME = 进程占用的CPU时间(自从进程启动以来)
COMMAND=启动该任务使用的命令行(小心命令行里出现的密码等信息, 有权限运行top命令的用户都可以看见它们!)

gtop

ktop

(X终端) 在图形用户界面下的两个top功能。我比较习惯使用gtop(在gnome自带)。在KDE环境下, ktop在K菜单的“System”菜单下的“Task Manager”里。

```
uname -a
```

(= “ Unix Name ” 带可选项 “ all ”) 显示你本地计算机信息。也可以使用通信用户界面的guname (X终端下) 得到更好的信息显示。

```
XFree86 -version
```

显示本地计算机的X-Windows的版本。

```
cat /etc/issue
```

检查你的发行版本。你也可以把你自己的信息放到文件/etc/issue里 在用户登录的时候会显示。如果你想要显示更多的信息,还有一个比较常见的做法是把本地专有的登录信息内容放在文件/etc/motd里 (“ motd ” = “ message of the day ” ,当天信息)

```
free
```

内存的信息 (以千字节显示) 。“ 共享 ” 内存是指可以被多个进程共同享有的内存 (比如,可执行代码是 “ 共享 ” 的) 。“ 缓冲 ” 和 “ 缓存 ” 则是用来保留最近访问的文件和数据 当其他进程需要更多的内存时候这些内容可以被减缩。

```
df -h
```

(=disk free 硬盘剩余空间) 输出所有文件系统的硬盘信息 (以易读的模式, h-human readable, 人类可读)

```
du /-bh | more
```

(=disk usage, 硬盘使用状况)输出根目录 “ / ” 下每个子目录具体的硬盘使用状况

```
cat /proc/cpuinfo
```

CPU信息 它显示文件cpuinfo的内容。要注意的是文件在/proc目录下不是真正的文件 它们只是在观察内核信息时起连接的作用。

```
cat /proc/interrupts
```

显示正在使用的中断号。在配置一个新硬件的时候可能需要浏览一下。

```
cat /proc/version
```

Linux的版本和其他信息

```
cat /proc/filesystems
```

显示当前使用的文件系统的类型

```
cat /etc/printcap |more
```

显示打印机的设置

```
lsmod
```

(= "list modules". 显示模块。必须是root用户，如果不是，使用命令/sbin/lsmod执行)显示当前已经加载的内核模块。

```
set | more
```

显示当前的用户环境（全部显示）。一般来说东西太多了。

```
echo $PATH
```

显示环境变量“PATH”的内容。这个命令也可以用来显示其他的环境变量。使用set来察看所有的环境变量（和上一个命令功能相同）

```
dmesg | less
```

输出内核信息（也就是常说的内核内部缓存信息）。按“q”退出“less”。也可以使用“less /var/log/dmesg”来直接检查最近一次系统启动时“dmesg”输出到文件里的内容。

```
chage -l my_login_name
```

查看我的密码过期信息

```
quota
```

查看我的硬盘区间（硬盘使用的限制）

```
sysctl -a | more
```

显示所有可设置的Linux内核参数

```
runlevel
```

输出和上一个和当前的运行级别(runlevel)。输出“N5”意味着：“没有上一个运行级别”和“5是当前的运行级别”。要修改运行级别，使用“init”，举例：“init 1”把当前的运行级别切换到“单用户模式”。

运行级别是Linux的操作模式。运行级别可以使用init命令进行快捷的修改。举例，“init 3”将把模式切换到运行级别3，以下是运行级别的标准：

- 0 - 中断（不要把系统初始设置“initdefault”设成这个值）
- 1 - 单用户模式
- 2 - 多用户模式，不带NFS（如果你没有网络，和运行级别3相同）
- 3 - 全功能多用户模式
- 4 - 目前没有使用
- 5 - X11（图形用户界面）
- 6 - 重新启动（不要把系统初始设置“initdefault”设成这个值）

系统的初始运行级别设置在文件：`/etc/inittab`里

`sar`

查看抽取系统活动记录文件信息（`/var/log/sarxx`，其中xx指的是当前的日期）。`sar`可以抽取很多种系统统计信息，包括CPU的平均载荷，i/o的统计，当天的网络流量统计或者是几天以前的数据。

5.4 基本操作

`ls`

`dir`

列出当前目录的内容。`dir`命令是`ls`命令的一个别名，所以这两个命令在文件罗列时使用的颜色代码是绝对一致的：

- 深蓝色 = 目录
- 浅灰色 = 一般文件
- 绿色 = 可执行文件
- 紫色 = 图形文件
- 红色 = 压缩文件
- 浅蓝色 = 链接文件
- 黄色 = 设备文件
- 棕色 = FIFO文件（先进先出，命令管道）

`ls -al |more`

列出当前目录的内容，所有文件（包括那些以“点号”开头的隐含文件），并以“长格式”显示。把所有输出定向到管道`more`，在屏幕满屏时可以暂停。`ls`命令还有很多其他的非常有用的可选项。还有一些快捷键可以减少麻烦的命令输入。试试`ll`（=长格式罗列，相当于`ls -l`）。另外一个我经常使用的可选项是“`ls -ad`”（列出所有当前目录下的子目录但是不输出它们的内容）。

`cd 目录`

修改当前目录。只输入“`cd`”没有指定目录名将切换到你的主目录。“`cd -`”将切换到上一次你访问的目录，这个命令在两个不同的目录之间切换时非常方便。“`cd ..`”将退到当前目录的上一级目录（父目录）。

`./程序名`

在当前目录下运行一个可执行程序。当可执行文件不在我的PATH下时，“`./`”是必须的。一个在我的PATH下的可执行文件可以简单的输入程序名运行而不需要“`./`”。

`shutdown -h now`

（要求root用户）关机。一般用于在远程关闭计算机。在控制台上可以使用`<CTRL><ALT>`来关闭计算机（所有用户都可以）

`halt`

`reboot`

init 6

(三个命令都要求root用户)关机或者重新启动计算机。主要用于远程系统重新启动,比上面的命令要简单一些。如果计算机停止响应时非常有用,我一般通过网络从其他机器telnet到停止响应的计算机上然后重新启动它。在本地计算机上,我则一般使用<CTRL><ALT>来重新启动。

vlock

(旧版本的RedHat没有这个功能)锁住当前的文本终端。也可以使用vlock -a锁住所有的终端(可能不是什么好主意)。最好的做法一般是使用logout命令退出系统。你不能在图形用户界面的终端上使用vlock命令 因为窗口管理器自带一个屏幕保护程序和一个加锁的工具(在KDE下一个小的锁型标志,键盘快捷键<CTRL><ALT><L>)。

5.5 文件管理

cp source destination

复制源文件source 到目标文件destination。举例“cp /home/stan/existing_file_name .”将复制文件到我的当前工作目录。使用“-R”可选项(代表“recursive”)复制整个目录树的内容,比如:“cp -R my_existing_dir/ ~”将复制我当前工作目录的一个子目录到我的主目录里。

mcopy source destination

从DOS文件系统复制一个文件或者复制一个文件到DOS的文件系统里(不需要挂载DOS文件系统)。

举例:

“mcopy a:\autoexec.~/junk”。查看“man mtools”了解更多的相关命令,可以不需要挂载DOS文件系统就可以直接访问DOS文件,比如命令:mdir, mcd, mren, mmove, mmd, mrd, mformat ... 我们不是很经常使用mtool的命令是因为 通常可以把DOS/MS Windows的文件系统挂载到Linux里,然后用Linux的常规命令来访问这些文件。

mv source destination

移动文件和更改文件名。对于文件和目录的移动和改名都可以使用这个命令。

rename 字符串 更新字符串 文件名

很好用的文件名修改工具。举例:“rename .htm .html *.htm”把所有的以“htm”为扩展名的文件名里的“htm”全部换成“html”。

ln source destination

从叫做source的源文件创建一个叫做destination的目标文件的“硬链接”。这个文件看起来好像是一个源文件的复制文件,但是实际上只保留了一个文件的硬拷贝,只是两个(或者多个)目录的入口指到同一个文件上。任何对这个文件所作的修改在所有地方都可以看得见。当一个目录得入口删除后,其他得目录得入口保持不变。“硬链接”的局限性是:文件必须在同一个文件系统里,创建目录或者一些特殊文件的硬链接是不可能的。

ln -s source destination

从叫做source的源文件创建一个叫做destination的目标文件的“标志链接”（软链接）。“标志链接”只是指定到“真实文件”的访问路由。和“硬链接”相比较，它的好处是源文件和目标文件可以在不同的文件系统里。“标志链接”也有自己的不足之处：如果“真实文件”被移动或者删除，“链接”就被破坏 不知道路径指到什么地方；“软链接”可以产生“循环参照”（就象数据库或者电子表格，举例：a指向b，b指向c，c指向a）。简而言之，标志链接非常有用而且也很常用（比“硬链接”要常用得多），但是同时可能带来一些文件管理的复杂性。

rm files

删除文件。你必须是拥有这些文件才能够删除它们（或者是root用户）。在很多系统里，你将会被提问是否确认删除文件。如果你不想被提问，使用“-f”（= force,强制删除）参数。举例：rm -f * 将删除所有当前工作目录下的所有文件，而且没有任何确认提示。

mkdir directory

创建新目录。

rmdir directory

删除一个空目录

rm -r files

删除文件，目录和所有它们的子目录。作为root用户，要小心使用这个命令 如果你在顶级目录树上执行这条命令，你将删除系统里所有的文件，到目前为止在Linux里还没有“取消删除”的命令。如果你真的想要这么做（再仔细想想？），这里告诉你该怎么办：使用命令 “rm -rf /*”。

rm -rf files

（强制）删除文件，目录和所有它们的子目录如上所述，但是跳过确认信息提示。如果你是root用户，绝对要小心使用这个命令。

mc

激活“Midnight Commander”文件管理器（看起来有点象Linux下的Norton Commander）。根据一些恐龙级的计算机用户的反映，这是至今为止最好的文件管理器。

konqueror &

（在X终端下）激活KDE文件管理器。也许这是文件管理器的极品。比微软的文件浏览器要好得多。它集成了网页浏览，PDF文件阅读和其他功能。真的很酷！

xwc

（在X终端下）另外一个很出色的文件管理器，是X-Win-Commander的缩写。比Konqueror的速度快，但是功能上稍微逊色。

nautilus &

(在X终端下)一个真正很酷的文件管理器。比Konqueror要慢,但是提供了一些有用的功能。比如图标提供文件内容预览!它甚至可以预览声音文件!但是对CPU速度依赖较大,它在我的1.33GHZ的计算机上运行出色,但是我从来不在我的133MHZ的老机器上使用它。

5.6 查看和编辑文件

```
cat filename | more
```

这个命令用于查看一个被称为 "filename"的文本文件的内容,每次一页。 "|"符号是"pipe" 符号(在很多美式键盘里,它与"\\"共用一个键)。more 使文本内容的输出显示停止在每一屏.对于长文件,有时可以很方便地使用 head 和 tail 来显示文件的开始和结尾部分或用 less 使之能上下滚动.如果你恰巧使用 cat 显示一个二进制文件,你的终端随后显示古怪的字符,你能使用 reset 命令恢复它。

```
cat filename | less
less filename
```

(选用二条命令之一) 滚动一个文本文件的内容.退出按 q 。"less" 大致相当于 "more" , 该命令来自 DOS,但经常 "less"比 "more"更方便因为它可使我能使用滚动条上下滚动。

```
head filename
```

显示长文本文件的前 10 行。

```
tail filename
```

显示长文本文件的最后 10 行。使用 tail -f filename 让文件的末尾持续读取该文件的不断出现的数据。

```
pico filename
```

使用简单,标准的文本编辑器 pico 来编辑一个文本文件。使用 <Ctrl>x 推出。Linux 有很多文本编辑器,包括几个基于图形用户界面的编辑器。一个新的 pico 克隆版(GPLed) 是 nano。

```
pico -w filename
```

编辑文本文件,没有文本行长度的限制.对于编辑配置文件很方便,如。/etc/fstab。

```
kwrite
```

(在 X 终端)很不错,"高级文本编辑器"。支持垂直文本选择!

```
kate
```

```
kedit
```

```
gedit
```

(在 X 终端).简单但很好的文本编辑器(基于 GUI)。

```
gxdit
```

(在 X 终端)另一多用途,具特色的文本编辑器。它甚至能定时备份。

```
latte
```

(在 X 终端)"代码"编辑器,即,用于写程序的无格式文本编辑器。

```
nedit
```

(在 X 终端)另一种程序编辑器。很好。

```
bluefish
```

(在 X 终端)html 编辑器(源码的句法高亮标注,有很多工具和选项)。

```
ispell filename
```

一个 ASCII 文本文件的拼写检查命令。AbiWord, WordPerfect, StarOffice 和其它文本处理器都使用键入就检查的策略,所以除非需要,否则你不用担心简单的 ispell 命令。Linux 新版本(如, RH7.0)包含了功能增强的拼写检查模块,称为 aspell,但上面的命令仍然工作。

```
look thermo
```

在你的系统单词目录(/usr/share/dict/words)中查寻开始于"thermo"的单词.

```
wvHtml ms_word_document.doc > filename.html
```

转化 MS Word 文件到 html 文件.

5.7 查找文件

```
find / -name "filename"
```

在你的文件系统中从根目录 "/" 开始搜索被称为 "filename" 的文件. "filename" 文件名可以包含通配符 (*, ?).

find 命令功能强大. 它有很多选项让你以不同的方式搜索文件. 例如, 通过日期, 文件大小, 权限, 拥有者等等. 但一些搜索查询要花费几分钟编写. 查看 info find. 这里是一些使用 find 来完成一些有用的任务的复杂例子.

```
find $HOME -name core -exec rm -f {} \;
```

以上命令从你的 \$HOME 目录开始寻找, 发现名字是 "core" 的文件. 对于每一个这种被找到的文件, 执行 "rm -f" (强制删除文件). {} 代表了已发现的文件, "\ " 终止了该命令列表.

```
find /dev -user "peter" |more
```

以上命令显示由用户 "peter" 所拥有的所有设备的文件. 显示文件名是 find 命令的默认行为, 所以如果这就是我所要做的, 无须特别指定.

```
find /home/peter -nouser -exec ls -l {} \; -ok chown peter.peter {} \;
```

在 /home/peter 目录下找到非有效拥有者的文件. 以长格式列出文件. 然后将这些文件的所有权改为用户 "peter" 和 组 "peter". 要处理文件的所有权, 你可能必须是超级用户.

```
locate filename
```

发现包含字符串 "filename" 的文件名. 这比前面的命令更容易, 更快, 但基于数据库 (通常在夜间重建), 所以无法找到刚刚存到文件系统的文件. 为了强制立即更新数据库, 作为超级用户可以使用: updatedb&.

```
which executable_name
```

如果我在命令行键入可执行文件的名字, 显示该可执行文件的全路径. 例如, 这条命令:

```
which mozilla. 我的系统出现:
```

```
/usr/bin/mozilla
```

```
whereis command
```

显示 "command" 命令的二进制文件, 源文件和手册的位置.

```
rgrep -r 'celeste' . |more
grep -r 'celeste' . |more
```

(二条命令, 在你的系统中选用一条). 在当前目录和所有子目录中搜索含有字符串"celeste"所有文件(选项"-r"代表递归). 显示文件名和在文件中包含所寻找的字符串所在的行.

```
kfind &
```

(在 X 终端). 在一个 GUI 前端界面使用 find and grep. 很不错. 命令行末尾的& 使 find 运行在后台以便使 X 终端保持可用状态.

5.8 X-windows 基础

```
xinit &
```

启动基本的 X-windows 服务器(没有 windows 管理器). "&"使命令运行在后台.

```
startx &
```

启动 X-windows 服务器和默认 windows 管理器.就象在带有 Win3.1 的 DOS 下键入"win" .

```
startx -- :1 &
```

在显示进程 1 中启动另一个 X-windows(默认显示进程为 0). 你能够同时运行几个 GUI 终端.在这些界面之间使用<Ctrl><Alt><F7>, <Ctrl><Alt><F8>来转换.

```
xterm
```

(在 X 终端) 运行一个简单的 X-windows 终端. 键入 exit 将关闭它.对于 Xwindows 有其它更高级的"虚拟"终端.我喜欢普遍使用的 konsole 和 kvt (二者均来自 kde) 和 gnome-terminal (来自 gnome). 如果你需要看起来更奇特的终端, 试试 Eterm. 我还可以选择简单, 运行快的 rxvt.

```
startkde
gnome-session
xfce
afterstep
AnotherLevel
fvwm2
fvwm
```

(在 X 终端, 7 条不同的命令, 使用其中之一来启动你喜欢的 windows 管理器). 在 X 服务器上在一个 X 终端启动你喜欢的 windows 管理器.

5.9 网络应用程序

```
mozilla &
```

(在X终端下) 运行mozilla网页浏览器. 现在的版本是Mozilla 1.01 (2002年10月), 这个浏览器非常不错. Mozilla是Netscape的一个衍生品(在老的Linux系统上一般使用NetScape). 还有其他几个不错的替代品比如konqueror和galeon (在Linux的X终端上输入konqueror和galeon直接运行)

mozilla -display host: 0.0 &

(在X终端下)在当前的计算机上运行Mozilla如何把输出定向到叫做“host”的计算机的第0个显示器的第0个窗口屏幕。你当前的计算机必须有权限在“host”计算机显示内容(一般的做法是,在“host”计算机上的X终端输入命令“xhost [当前的计算机名]”。其他的X - Windows的程序也可以通过同样的办法来运行)。

lynx file.html

以纯文本方式来浏览html超文本文件。尽管lynx看起来不像基于图形用户界面的网页浏览器那么出色,但是它很小巧,几乎没有出差错的时候,也不需要什么特殊的设置,只要你的网络是正常的。

konqueror &

(在X终端下)集成了文件管理器和网页浏览器的功能。很不错,和Mozilla的很多功能可以一比高低。KDE自带的软件。

pine

一个很不错,老古董的文本邮件阅读工具。还有另外一个老古董是elm。你的Mozilla将会阅读你因特网上的邮件,pine则让你阅读“本地”的邮件。比如,你儿子的邮件或者在你网络里的另外一台计算机的cron进程发送给你的信息。命令mail也可以用来阅读和写邮件,但是没有那么方便使用。

mutt

一个非常基本但是及其有用的快速邮件阅读工具。

mail

给email使用的一个基本的操作系统工具。如果你只是要阅读邮件,可以参考上面的快速email阅读命令。如果你要在shell脚本上发送邮件,mail倒是一个很好的选择。

kmail &

(在x终端下)很好的图形用户界面的邮件程序。我使用kmail,因为它比Netscape的mail做得要出色的多。我可以拥有多个电子邮件帐号,然后分别从本地的smtp服务器和ISP的pop3服务器上下载邮件到同一个收件箱。简单易用,支持数字签名。

licq &

(在x终端下)一个ICQ的客户端程序。还有另外一个不错的选择是kxicq。就的Linux发行版本不带licq客户端程序,你可能需要自己下载安装。

knode &

(在x终端下)启动我喜欢的新闻组阅读器(usenet)。比Netscape内置的阅读器要好。

talk username1

和另外一个登录在系统里的用户交谈 (也可以使用 “ talk username1@machine_name ” 来和另外一台计算机上的用户交谈) 。要接受会谈请求, 输入命令 “ talk username2 ” 。如果你正在忙着而有人想要和你交谈, 你可以使用命令 “ mesg n ” 拒绝接受信息。你可以使用命令 who 或者 rwho 来查看有哪些用户当前登录在系统里。talk 是一个老古董的 UNIX 标准工具, 在某些场合下仍然非常有用。

telnet server

使用 telnet 协议连接到另外一台计算机。可以使用计算机名或者 IP 地址。你会要求输入用户登录名和密码 你必须远端计算机上有有效的用户帐号和密码。telnet 将让你连接到另外一台计算机上好像你就在使用那一台计算机的键盘 (几乎完全一样) 。telnet 不是非常安全 你输入的任何东西在网络上都是开放的文本模式, 包括你的密码! 一个在 “ 传输路径 ” 上稍微有点专业知识的系统管理员就可以阅读你所输入的内容。如果你要加密的传输, 使用 ssh (需要一些系统设置) 。

rlogin server

(= remote login, 远程登录) 。这将使用你在当前登录系统上使用的用户名和密码, 如果登录失败将会提醒你输入密码。

rsh server

(= remote server) 连接到远程计算机的另外一个办法。这将使用你在当前登录系统上使用的用户名和密码, 如果登录失败将会提醒你输入密码。

ssh servername -l username

(= secure shell, 安全的 shell) 使用安全连接远处登录到另外一台计算机。ssh 是安全的, 因为它使用一对 RSA “ 公共 - 私有 ” 钥匙压缩所有传送的数据。如果你没有指定用户名, 它假设你使用当前的用户名。

ftp server

ftp 到另外一台计算机 (还有一个 ncftp 带有一些附加的功能, 另外还有基于图形用户界面的 gftp) 。ftp 对于从一台计算机上复制文件到另外一台计算机上非常有用。可以试着用 “ anonymous ” (匿名) 用户如果你在远程计算机上没有有效帐号。建立连接之后, 使用 “ ? ” 查询有效的 ftp 命令一览。最基本的 ftp 命令有以下这些 :

ls	查看远程计算机上的文件
ASCII, binary	设置文件传输模式为 “ 文本模式 ” 或者 “ 二进制模式 ” , 选择正确的模式非常重要
get	从远程计算机上复制一个文件到本地计算机
mget	从远程计算机上复制多个文件到本地计算机
put	从本地计算机上复制一个文件到远端计算机
mput	从本地计算机上复制多个文件到远端计算机
bye	结束连接

如果你要使用脚本文件实现自动化, 你可以考虑 ncftpget 和 ncftpget, 举例:

```
ncftpget -u my_user_name -p password -a remote.host.domain.remote_dir *local.html
```

如果你的计算机在防火墙后面，ncftpput可能会有一些问题 - 你需要配置文件 /home/user_name/.ncftp/firewall。另外一个选择，你也可以使用lftp来达到同样目的：

```
lftp -e "mput -a *local.html" -u my_username, my_password  
ftp://remote.host.domain
```

如果要保留ftp目录的镜像，可以使用命令fmirror：

```
wget -m --no-parent http://sunsite.dk/linux-newbie
```

从网站上复制文件。以上这个例子使用可选项-m (mirror, 镜像) 从主站上获得完整的文件集合。可选项 "--no-parent" 限制搜索子目录里的文件。

minicom

minicom是一个专门给串口设计的“终端模拟器”。看起来和用起来都很象“procomm”或者“telix”。对于串口通信的故障诊断非常有用。

rx

使用Zmodem, Ymodem和Xmodem协议接收文件。Xmodem需要文件名。输入rx -help得到更多的信息。现在还有谁在使用这些协议？

5.10 文件压缩和解压缩

```
tar -zxvf filename.tar.gz
```

(=tape archiver, 磁带压缩) 解压一个你从因特网上下载的tarball文件 (*.tar.gz or *.tgz)。

```
tar -xvf filename.tar
```

拆解一个tarred (但是没有压缩) 的文件。

```
tar czvpf /var/backups/mybackup.tar.gz /home  
cd /; tar xzvpf /var/backups/mybackup.tar.gz */myfile.rtf'
```

作为root用户，创建一个/home目录的备份压缩文件。第二条命令显示如何从备份恢复一个文件。在tarball文件里将不包含“点”文件 (指的是以“.”开头的文件或者目录)。要把所有文件解包，可以使用：tar cvzf filename.tgz *.[a-zA-Z]*

```
gunzip filename.gz
```

解压一个压缩文件 (*.gz或者*.z)。如果你想把文件压缩成这种格式，可以使用gzip(还有zip或者compress命令)。注意一下gunzip有趣的发音听起来象“gun zip”(大炮压缩)。

```
zcat filename.gz | more
```

(=zip cat)显示一个压缩文件的内容。还有其他一些相关命令不需要预先解开压缩文件就可以使

用 : zless, zmore, zgrep...等等。

```
bunzip2 filename.bz2
```

(=big unzip) 解压一个由bzip2压缩工具压缩的文件 (*.bz2)。一般用于很大的文件。

```
unzip filename.zip
```

解压一个由DOS下的PKZIP压缩工具压缩的文件 (*.zip)

```
zip filename.zip filename1 filename2
```

把两个文件 “ filename1 ” 和 “ filename2 ” 压缩到一个压缩文件叫做 “ filename.zip ”。

```
unarj e filename.arj
```

解压一个使用arj压缩的文件 (*.arj)

```
lha e filename.lha
```

解压一个使用lha压缩的文件 (*.lha)

```
uudecode -o outputfile filename
```

解码一个使用uuencode的文件。uu-encoded在使用email传送非纯文本文件的时候应用非常广泛 (uuencoded把任何文件格式都变成文本文件)

```
cat filename | mimencode -o filename.mime
```

```
cat filename.mime | mimencode -u -o filename
```

(两个命令) 第一个命令把文件filename编码成基于因特网邮件7bit数据传送标准的文件 filename.mime。第二个命令把基于因特网邮件7bit数据传送标准的文件filename.mime解码成一般文件filename。

```
ar -x my_archive.a file1 file2
```

(=压缩存档)。从压缩文件my_archive.a解压出文件file1和file2。ar压缩工具更多的用于保存文件库记录。

```
ark &
```

(在X终端下)。一个图形用户界面的压缩工具。也许是你管理压缩文件所需要的方便工具。还有另外一个选择是gnozip。

5.11 进程控制

```
ps
```

(= “print status” or “process status”) 列表显示当前正在运行的进程的进程标识号 (PID)。用 `ps aux` 查看你的系统上下在运行的所有进程 (也包括其他用户的进程和没有终端控制的进程), 每个进程都显示其所有者的名字。使用 “top” 命令持续显示当前正在运行的进程列表。

`any_command &`

在后台运行任务命令 (符号 “&” 表示 “在后台运行前面的命令”)。任务号 (Job_number) (译者注: 和进程号) 将显示在屏幕上, 你可以使用任务号将该任务调到前台 (见下面)。当我从一个 X 终端启动一个 GUI 程序时, 我经常使用 “&”。

`jobs`

列表显示后台运行或是被停止了了的进程并显示它们的任务号。

`fg job_number`

将一个后台运行的或是被停止了了的进程调到前台运行

`bg job_number`

将一个进程调至后台运行, 这与在命令行用 “&” 启动后台进程是完全一样的。这将重新启动一个被停止了了的后台进程。一般可以用 `<Ctrl>z` 来停止当前的前台进程。如果你有被停止了或后台运行的任务, 你必须在命令行输入 `exit` 命令两次, 以从系统中注销。

`batch`

`at>updatedb<Ctrl>d`

在系统负荷低的时候执行任何命令 (经常需要长时间完成的任务)。你可以从系统中注销, 但进程将继续运行。当命令完成后, 一封包含输出信息的电子邮件将发送给你。在以上示例中, “at>” 是一个提示符, 将要执行的命令是 `updatedb`, 然后按 `<Ctrl><d>` 来结束命令的输入 (你可以输入很多命令提交执行, 用 `<Enter>` 来分隔它们)。

`at 17:00`

在指定时间执行命令。你将被提示输入要执行的命令, 直到你按 `<Ctrl>d`。相关的命令有 `atq` (显示由 `at` 启动的进程队列) 和 (从 “at 队列” 中删除一个进程)

`kill PID`

强制一个进程终止。首先要用 `ps` 确定终止的进程的进程标识号 (PID)。

`killall program_name`

通过程序名终止一个 (组) 程序。例如: `killall pppd` 将断开你的拨号网络。

`Nohup program_name`

(=no hangup)。运行程序 `program_name`, 并使它在你注销时不被中断。输出结果被重定向到你的用户的主目录下的 `nohup.out` 文件。在你确信不想运行一个交互程序的时候, 使用 `nohup`。

`Xkill`

(在 X 终端中) 用鼠标终止一个基于 GUI 的程序。(用你的鼠标指到你终止的进程的窗口并点击)

kpm

(在 X 终端中) KDE 进程管理程序

lpc

(由 root 用户执行) 检查和控制打印机。输入“?”可以查看命令列表。

lpq

显示打印机队列的内容。在你可从“K”菜单 - 实用程序(“K” menu-Utilities)中调用基于 GUI 的“打印机队列”(“Printer Queue”)

lprm job_number

从打印队列中删除一个打印任务。

nice program_name

执行一个程序并调整其优先级。因为在本例中没有指定优先级, 将从缺省优先级(通常是 0)增加 10(进程将运行较慢)。较小的值, 代表较高的优先级。优先级的取值范围为 -20 到 19。只有 root 用户能指定负值。用 top 可显示正在运行的进程的优先级。

renice -18 PID

(以 root 用户运行) 将一个正在运行的进程的优先级调整为 -18。普通用户只能调整他们自己的进程, 而且只能从当前值向上调(即使其运行变慢)。可以使用 renice +10 -u peter 来减少用户 peter 对 CPU 资源的占用, 这样当他运行高强度运算任务时, 使其他用户不受到损害。

<Ctrl>c, <Ctrl>z, <Ctrl>s 和 <Ctrl>q 也属于本章内容, 但它们在前面已经描述过了。简单地讲, 它们表示停止当前命令, 调度当前命令到后台运行, 停止数据传输, 恢复数据传输。

lsdf

列表显示已经打开的文件。如果你是 root 用户, 所有的打开的文件都将被显示。使用 lsdf /dev/tty1 你可以指定显示由第一控制台所有的进程打开的所有文件。要想只显示网络文件(对安全审计非常有用), 你可以使用 lsdf -l (以 root 用户执行)。

Watch -n 60 my_command

以 60 秒为间隔反复执行 my_command 命令(缺省间隔为 2 秒)

5.12 一些管理命令

su

(=substitute user id)采用超级用户(=root)的身份(将提示你输入口令)。输入“exit”返回你先前的注册。不要习惯于用身份在你的机器上工作。root 帐户是用于管理的, 当你需要访问管理帐户时, su 命令可以使你轻松进入。你也可以使用 su 命令去采用其他任何用户的身份, 例如: su barbara 将你变成“barbara”(除非你是超级用户, 否则需要口令)。

```
Alias ls= " ls -color=tty "
```

为命令产生一个别名，用增加一个关于颜色的格式。在这个例子中，别名也叫“ls”，只有当输出被指向一个终端（而不是文件）时，color 选项才被执行。如果你想让系统上的所有用户始终能使用这个别名，就把它写到/etc/bashrc 文件中。别名是定制你的系统的一种方便的方法。只输入“alias”可以查看你的帐户可使用的别名的列表。使用 unalias alias_name 可以删除一个别名。

```
cat /var/log/httpd/access_log
```

显示自从上次日志文件被“转动”（一般每天在 cron 程序运行时转动一次）后，谁连接过你的 http (apache) 服务器。前一次的日志文件为 access_log.1，再前一次的日志文件 access_log.2，依此类推。

```
cat /var/log/secure
```

（以 root 用户运行）检查重要系统日志。如果你使用 Internet 访问的话，定时检查的一个很好的主意。

```
ftpwho
```

（以 root 用户运行）检查当前谁连接在你的 ftp 服务器。

```
printtool
```

（在 X 终端上以 root 用户运行）打印机配置工具。设置文件存放在/etc/printcap 和 /var/spool/lpd 目录下。

```
setup
```

（以 root 用户运行）配置鼠标、声卡、键盘、和系统服务。许多发行套装都有自己特定的配置实用程序，setup 是 Redhat 的缺省配置程序。Mandrake 7.0 提供非常好用的 DrakConf。

```
linuxconf
```

（以 root 用户运行，在文本模式或是在 X 终端上）。在这里你可以访问和更改上百种网络设置。功能非常强大 一次不要改动太多东西，改动你所不清楚的项目时一定要小心。Redhat 网络配置工具 netconf 是 linuxconf 的一个子集，因此它比较简单而且很多时候容易使用。

```
mouseconf
```

（以 root 用户运行）。一个配置鼠标的简单工具（在完成初始安装后）。Mandrake 也有一个类似的工具 mousedrake。

```
kudzu
```

（以 root 用户运行）。自动检测和配置硬件。如果你的鼠标（或是其他串行设备）有问题，你可能想要停用 kudzu，这样它在系统启动过程中就不会运行（kudzu 弄乱了我的系统，使我的鼠标不能工作）。你可以在你需要时手动运行它。

timeconfig

(以 root 用户运行) 为你的系统设置时区。我的计算机的硬件时钟保存的时间是 UTC (协调世界时, 也称 GMT 或格林威治时间)。这样, 我避免了任何在转换时区时 (由夏时制、通过网络跨洲传送文件或旅行外出引起的) 可能发生的问题。时惯上在服务器上使用 UTC 时间来避免时间“倒流” (有时会引发一些问题)。文件的时间戳总是保持使用 UTC 时间, 并使用时区信息来显示当地时间。例如, 许多应用程序 (如编译器, 数据库) 是依靠比较文件的时间戳来区分新的文件与旧的文件。保证时区信息正确是很重要的。我将当地时间作为 BIOS 时间来避免发生问题的唯一原因是, 当我的从一台计算机上进行双重启动, 而其它操作系统 (MS Windows?) 不知道如何处理 UTC。然后, 我通过选择“Hardware clock set to GMT”来告诉 Linux 服务器, 这 Linux 样在需要 UTC 时, 可以计算出来。

setclock

(以 root 用户运行)。从当前 Linux 系统时间设置计算机硬件时钟。首先, 使用“date”命令设置 Linux 系统时间。例如, 我可以将日期和时间改为 2000 - 12 - 31 - 23 : 57, 使用命令:

```
date 123123572000
```

然后将时间写到硬件时钟, 使用命令:

```
setclock
```

dateconfig&

(在 X 终端上, 以 root 用户运行, 否则将提示你输入 root 用户口令)。一个非常棒的 GUI 实用工具, 用于设置操作系统时间、硬件时钟和时区信息, 告之 BIOS 使用 UTC 时间, 而不需要前面的两个命令。

Xvidtune

(在 X 终端上) 调整所有分辨率下的显示器设置, 以消除黑带, 上下左右调整显示, 等等。(首先使用显示器自身的调整按钮, 使得文本模式在屏幕上正确显示)。然后用 xvidtune 针对每种分辨率调整显示频率, 使其在屏幕上正确显示。为了使改动保持下去, 在屏幕上显示频率, 然后将它们传送到/etc/X11/XF86Config 设置文件中。在较新型的显示器上, 你可能更愿意使用显示器的内建设置来调整你的显示器, xvidtune 主要是给那些没有记忆设置参数的功能的老式显示器用的。

Kvideogen

(在 X 终端上) 为定义你的屏幕分辨率产生“模式行”。在产生设置文本 (“模式行”) 后, 你可以将其复制 - 粘贴到的设置文件/etc/X11/XF86Config 中 (或是/etc/X11/XF86Config-4, 如果你使用 X-sever 4.00 版本)。另见快捷键<Ctrl><Alt><+>。

```
SVGATextMode 80x25x9
```

```
SVGATextMode 80x29x9
```

(以 root 用户运行) 修改文本终端的文本分辨率。在示例中 (第二行) 我将文本屏幕改为 80 列 x29 行, 字符高度为 9 像素。第一行定义的是一个常用的分辨率, 这样第二行命令在系统上不能工作, 我能通过按两次<ArrowUP> (译者注: 上箭头) 和<Enter>来重新控制屏幕。可以使用哪些模式取决于你的显示卡和你的显示器的刷新频率 需要编辑文件/etc/TextConfig (以 root 用户运行), 去掉相应行前面的注释符, 使 SVGATextMode 知道系统支持哪些模式。

SuperProbe

(以 root 用户运行) 一个检测显示卡类型和内存数量的实用工具。

```
cat /var/log/Xfree86.0.log
```

一个对检测 X-window 设置过程中的错误非常有用的日志文件。文件名中的 “ 0 ” 表示 “ 显示方式 0 ” - 如果你需要 “ 显示方式 1 或 2 等 ” 的日志文件, 你需要相应地修改文件名。

lspci

显示主板信息和 PCI 扩展槽内都插了哪些卡。我的旧的计算机有 ISA (或 EISA) 插槽, 没有 PCI。

lsdev

显示硬件信息 (DMA, IRQ, IO 端口)

lsdf |more

显示系统上打开的文件。

kernelcfg

(在 X 终端上以 root 用户运行) 增加、删除核心模块的 GUI 工具。每个模块就象一个设备驱动程序 Linux 内核中支持一个设备或功能的一段程序。你在命令行使用 insmod 命令也可以完成同样的工作。

lsmod

(=list modules)显示当前加载的核心模块。一个模块就象一个设备驱动程序 - 它对一个硬件或一个特性提供操作系统核心支持。

modprobe -l|more

列出系统内核的所有可用模块。哪些模块可用是由如何编译 Linux 核心决定的。几乎每一种可能的模块特性都可以被编译到内核中, 以 “ 硬连接 ” (可能会快一点, 但不能删除) 或 “ 模块 ” (可能会慢一点, 但可以根据需要装载/删除), 或 “ 无 ” (根本不支持这个特性) 等方式。系统核心所支持的模块 (编译过的) 都以文件的形式存在 /lib/modules (及其子目录) 下, 当你丢失模块时, 你可以浏览这些目录, 可以找到线索。如果系统核心不支持你所需要的模块, 你可以启用该模块并重新编译系统核心 (这是因为原装的核心都只包括常用的模块, 没有预编译一些试验性的模块。还有, 如果你有一些该死的边缘硬件...)。

modprobe sb

装载 soundblaster (sb) 模块。使用前面的命令来查找已经加载的核心模块。

```
insmod parport
```

```
insmod ppa
```

(以 root 用户运行) 向内核中插入模块 (一个模块大致相当于一个 DOS 设备驱动程序)。平常, 我用 “modprobe” (见前面的命令) 来插入模块。这个例子展示如何为支持一个外置并口 100MB ZIP 驱动器来插入模块 (好像在 RH6.0 和 6.1 下, 外部 ZIP 驱动器以其他方式工作都会有问题)。对于 250MB 外置 ZIP, 可用 imm 模块来代替 ppa 模块。

```
rmmod module_name
```

(以 root 用户运行, 但不是必须) 从系统核删除 module_name 模块。

```
depmod -a
```

(以 root 用户运行) 为核心建立模块依赖关系表。除非你想/etc/modules 而不想重新启动系统, 否则不必要。

```
setserial /dev/cua0 port 0x03f8 irq 4
```

(以 root 用户运行) 将一个串行接口设置为非标准设置。这个例子显示第一个串口的标准设置 (cua0 或 ttyS0)。第二个串口 (cua1ttyS1) 的 PC 标准设置为: 端口地址 0x02f8, 中断 3。第三个串口 (cua2 或 ttyS2) 为: 0x03e8, 中断 4。第四个串口 (cua3 或 ttyS3) 为: 0x02e8, 中断 3,。如果你要它们在启动时即被设置, 将这些设置加到文件中。详见 man setserial。

```
Tunelp
```

(以 root 用户运行, 很少用到) 调整并行端口。

```
/sbin/chkconfig --level 123456 kudzu off
```

(以 root 用户运行) 用来检查/启用/禁用在各个运行级下自动启动的系统服务的工具。一般情况下, 如果我需要在当前运行级下启用/禁用一个服务, 我只用 Redhat ntsysv 实用程序, 但 chkconfig 给了我更多的灵活性。另外一个可用的工具是 (基于 X 界面)。这个例子展示如何禁用 kudzu 服务, 使它在任何运行级都不启动 (它弄乱了我的一台机器上的鼠标设置)。显示在所有运行级下启动/停止的服务, 我使用:

```
chkconfig --list|more
```

检查服务的当前状态, 我使用:

```
service --status --all
```

立即启动一个服务, 我可以使使用一些东西, 像 (启动一个 ftp 服务器):

```
service su-ftpd start
```

重新启动网络服务 (如: 在我修改了它的配置后), 我可以使:

```
service smb restart
```

```
symlinks -r -c ds /
```

(以 root 用户运行) 检查和修改系统上的符号链接。从根开始, 遍历所有的子目录 (选项 -r=“rescuer”) 修改绝对路径链接杂乱链接为相对链接, 删除空悬链接, 缩短超长链接 (选项 -cds)。如果你的文件系统覆盖几个不同的硬盘分区, 你需要为每个分区重新运行这个命令 (如: `symlinks -r -cds /usr`)。

```
cd /usr/src/linux-2.4.7-20
make xconfig
```

(在 X 终端上以 root 用户运行) 一个漂亮的 GUI 前端工具, 可在编译自定义内核的准备过程中配置内核选项。(例子中的目录名中包含我的版本号, 如果你的内核版本号不是本例中的, 你需要修改你的目录名。你需要 “Tk” 解释器来地, 并且安装了核心的源代码)。“make xconfig” 的其他实现方法是 “make config” (在文本模式下运行一个脚本, 并回答一些问题) 和 “make menuconfig” (运行一个基于文本菜单驱动的配置工具)。

更多信息; Less /usr/share/doc/HOWTO/Kernel-HOWTO

在使用 “ ” 完成新内核的配置选择后, 你要执行以下命令来编译新的内核;

`make clean`(这是一个可选的命令; 它清除旧的目标文件, 它可能会延长编译过程, 但在一些情况下它可以阻止一些问题的发生)

```
make dep
make bzImage
```

最后的命令需要一些时间来完成 (可能是 10 分钟或是 2 个小时, 这取决于你的硬件)。它产生文件 `arch/386/boot/bzImage`, 即是你的新内核。接下来:

```
make modules
make modules_install
```

现在新内核安装在 `/lib/modules/KernelName`

如果你想运行多个内核的时候, 不要改动模块目录的名字 - 内核必须能够找到与其匹配的模块。如果你想改变内核的名字, 你必须编辑主内核 `makefile` 文件 (如: `/usr/src/linux-2.2.14/Makefile`), 修改最开始的几行。我的文件 (缺省 TH 7.2) 是:

```
VERSION = 2
PATCHLEVEL = 4
SUBLEVEL = 7
EXTRAVERSION = -10custom
```

可以用来显示当前运行的内核的名字。我的是。

我“原装”的 RedHat 内核配置在 `/boot/config-2.4.28-24(RedHat 8.0)` 文件中, 同时一些附加的 “custom” 内核设置在目录 `/usr/src/linux-x.x.x./configs` 中。我可以从命令的对话框中加载它们中的任何一部分。

现在我能安装新内核。安装过程涉及到将新内核复制 (同时改名) 到目录中:

```
cp arch/386/boot/bzImage /boot/vmlinuz-2.4.7-10custom
cp System.map /boot/System.map-2.4.7-10custom
```

并修改 `/etc/lilo.conf` 或 `/boot/grub/grub.conf`, 这样在启动时我可以选择用哪一个内核 (新的或旧的) 引导。强烈建议保存旧内核做为一个引导的选项 (万一新内核拒绝引导)。

如果你使用 `initrd(initial ram disk)` 做两阶段引导, 你还需要生成一个包含在启动过程中内核用到的模块的映像:

```
mkinitrd /boot/initrd-2.4.7-10custom.img 2.4.7-custom
```

看一下如何给内核打补丁, 快速参考:

```
cd /usr/src/linux-2.4.7-10
patch -E -p1 < /home/download/the_patch_to_apply
```

阅读 `/usr/doc/HOWTO/Kernel_HOWTO` 也会有所帮助, `man depmod` 也行。配置、编译和安装一个新内核非常简单, 但它也能导致许多问题。编译内核也是检测你的硬件的一个好方法, 因为它涉及到相当大量的计算。如果你的硬件整合得不好的话, 你会收到 “signal 11” 错误 (请阅读 `/usr/doc/FAQ/txt/GCC_SIG11/FAQ`)。

`ldconfig`

(以 `root` 用户运行) 为动态链接库的加载程序 (“`ld`”) 重新产生联接和缓存区。当你在系统上安装一个新的动态链接库的后, 你也许想要运行 `ldconfig`。(每次当你引导计算机的时, 它都会重新运行一遍, 所以如果你重新启动, 就不必再人为运行一遍了)。

```
mknod /dev/fd0 b 2 0
```

(=`make node`, 以 `root` 用户运行) 手动创建一个设备文件。这个例子展示的是创建一个与你的第一个软驱设备相关联的设备文件, 当你意外地删除了该设备的时候, 它会十分有用。选项是: `b` = 块设备, `c` = 字符设备, `p` = FIFO 设备, `u` = 无缓冲字符设备。两个整数指定了主设备号和次设备号。平时我也不知道 `mknod` 需要哪些参数。要创建设备时, 我先阅读 `man MAKEDEV` 来找出设备的名字, 然后运行 `/dev/MAKEDEV` 脚本文件, 这个脚本文件可根据名字知道是何种设备 见下一个命令。如果提到的手册没有帮助的话, 我推荐包含内核源代码的最终文档:

```
less /usr/src/linux/Documentation/devices.txt
```

```
cd /dev
./MAKEDEV audio
```

(以 `root` 用户运行) 恢复 “`audio`” 设备, 我刚好把它搞得有点不灵了。也可见前面的命令。

5.13 硬盘和软盘实用工具

```
fdisk /dev/had
```

(= “`fixed disk`”, 以 `root` 用户运行) Linux 硬盘驱动器分区工具 (DOS 有一个同名的工具)。在上面的例子中, 我指定要对第一个 IDE 接口上的第一个硬盘进行分区, 因此是 “`hda`”。如果我是你, 我会在对任何分区使用之前备份所有重要数据。我不知道有谁喜欢用 `fdisk` (无论是 Linux, 还是 DOS 版本) 我更喜欢简单易用的 `cfdisk`, 见下一个命令。

```
cfdisk /dev/had
```

(以 `root` 用户运行) 基于菜单的硬盘驱动器分区工具。比更容易使用 (见前一个命令)。物理驱动器包括基本分区 (每个硬盘上最多 4 个), 和逻辑分区 (数量上没有限制)。一个基本分区是可

以引导的。逻辑分区必须包含在“扩展分区”中；扩展分区不能使用其本身，它们只是逻辑分区的一个容器。当对一个硬盘进行分区时，一般我会：(1) 创建一个基本分区，(2) 使这个基本分区可以引导，(3) 创建一个扩展分区，(4) 在扩展分区中创建逻辑分区。

```
sfdisk -l -x |more
```

(以 root 用户运行) 列出你的系统上的所有驱动器的分区表 (包括扩展分区)。

```
parted /dev/had
```

Linux (ext2) 和 DOS (FAT 和 FAT32) 硬盘分区的处理工具。它可以创建、删除、移动、复制、压缩和扩展分区。你在使用它之前应该备份你的数据并仔细阅读 info parted。

```
fdformat /dev/fd0H1440  
mkfs -c -t ext2 /dev/fd0
```

(=floppy disk format, 两个命令, 以 root 用户运行) 对第一个软驱 (dev/fd0) 中的高密度软盘 (1440KB) 执行低级格式化。然后建立一个文件系统 (-t ext2), 检查/标出坏块 (-c)。建立文件系统相当于高级格式化。你还可以将软盘格式化不同的密度 (非标准的); 试一下 ls /dev/fd0<Tab>。你能够使用 fdformat /dev/fd0 格式化成缺省密度 (一般是 1440K)。

```
badblocks /dev/fd01440 1440
```

(以 root 用户运行) 对一张高密度软盘进行坏块检查并将结果显示到屏幕上。参数“1440”指定了要检查 1440 块。这个命令不修改软盘。badblocks 也可对用来对硬盘表表进行检查, 但是在我进行全面读 - 写检查之前, 我必须先将文件系统卸下:

```
mount [查出哪个设备包含我要进行坏块检查的磁盘分区]  
umount /dev/hda8 [卸下选定的分区]  
badblocks -n /dev/hda8 [以非破坏读 - 写方式检查选定的分区, 这样我的数据就不会被删除了]  
mount /dev/hda8 [如果没有坏块信息显示出来, 将该分区安装回去]  
如果发现坏块, 它们能够被在硬盘上标记出来, 这用它们就不会再被使用了:  
e2fsck -c /dev/hda8
```

```
fdck -t ext2 /dev/hda2
```

(=file system check, 以 root 用户运行) 检查和修复文件系统, 例如在由于电源故障导致的“不干净”关机之后。上面的这个例子执行对分区 hda2 的检查, 文件系统类型为 ext2。你必须卸下分区或以“单用户模式”(在 LIL0 提示符处输入“linux single”或以用户使用 init 1 来进入单用户模式) 引导来执行这个命令。如果在文件系统检查过程中发现错误, 我选择缺省设置来修复。

```
Tune2fs -j /dev/hda2
```

(以 root 用户运行, 只用于支持 ext3 的内核—RH7.2) 调整文件系统的可调参数。上面的这个例子展示如何一个磁盘分区 (本例为 hda2) 增加日志, 成功地将文件系统转为 ext3 (日志) 文件系统。要完成这个转换, 你还须编辑 /etc/fstab 文件, 将文件系统类型由 ext2 改为 ext3, 否则你将遇到问题 ext2 不能安装一个非正常关闭的日志文件系统。要检查文件系统的类型, 使用 mount (无参数) 或 cat /etc/mstab。如果你需要关于的更多信息, 试一下:

<http://www.symonds.net/~rajesh/howto/ext3/ext3-6.html>

tune2fs 的其它参数可以让你添加一个卷标签，调整安装文件系统（被检查过）的数量（最大安装数），或激活定时文件系统检查（很少用到）。

```
dd if=/dev/fd0H1440 of=floppy_image
dd if=floppy_image of=/dev/fd0H1440
```

（两个命令，dd=“data duplicator”）在当前目录下创建一个软盘的映像文件，名为“floppy_image”。然后将 floppy_image（文件）复制到另一张软盘。功能像 DOS 中的“DISKCOPY”。

```
mkbootdisk -device /dev/fd0 2.4.2-3
```

制作一张应急引导软盘。在系统安装过程中你会被问到是否要制作一张引导盘。以上命令展示在安装后如何制作启动盘，使用第一个软盘驱动器（/dev/fd0）。你可以运行 `uname -a` 或 `ls /lib/modules` 来知道内核名字（命令中需要，这里是 2.4.2-3）。

5.14 用户帐户和文件权限管理

```
useradd user_name
passwd user_name
```

（以 root 用户运行）创建一个新帐户（你必须是 root 用户）。例如：`useradd Barbara`。在接下来的一步别忘了为新用户设置口令。用户的主目录（被创建的）是 `/home/user_name`。你也可以用一个等价的命令 `adduser user_name`

```
ls -l /home/peter
useradd peter -u 503 -g 503
```

（以 root 用户运行）创建一个用户来匹配一个已有目录（可能是以前安装中的）。如果用户 ID 和组 ID（每个文件显示出来的）以前都是 503，那么我用匹配的用户名，用户 ID（UID）和组 ID（GID）来创建用户。这样在系统升级后可以避免由于改变用户文件的所有权而引起的混乱。

```
userdel user_name
```

删除一个帐户（你必须是 root 用户）。用户主目录和未发送邮件必须单独处理（人工处理，因为你必须决定如何处理这些文件）。还有一个命令 `groupdel` 来删除组。

```
groupadd group_name
```

（以 root 用户运行）在你的系统上创建一个新组。在一个家用计算机上不是必须的，但对于有少量用户的家用计算机是十分方便的。

例如，我能够创建一个“friends”组，使用：

```
groupadd friends
```

然后编辑文件，将我的名字和朋友的名字加到组的列表中的一行，最后一行看起来像这样：

```
friends:x:502:stan,pete,marie
```

然后，我改变一个指定文件的权限，这样这个文件就属于我和组“friends”。

```
chgrp friends my_file
```

这样，这个组的成员就有访问这个文件的特殊权限，而其他人则没有，改变读和写权限的例子如下：

```
chmod g=rw o= my_file
```

另外一种做法是将写的权限赋予所有用户，即使是在家庭电脑上绝对是不安全的。

groups

列出当前用户所属的组。我还可以使用 `groups john` 来找出用户 `john` 所属的组。

```
usermod
```

```
groupmod
```

(以 `root` 用户运行) 两个命令行工具，用于修改用户帐户和组，而不用修改相关文件：`/etc/passwd`, `/etc/shadow`, `etc/group` `/etc/gshadow`。一般是不必要的。

```
userconf
```

(以 `root` 用户运行) 菜单驱动的用户配置工具 (口令政策、组信息更改、增加用户等)。它是 `linuxconf` 包的一部分，但可以单独运行。

```
passwd
```

修改当前帐户的口令。如果你是 `root` 用户，你能为任何用户修改口令，使用命令：`passwd user_name`

```
chfn
```

(= " change full name ") 修改你的信息 (全名，办公室房间号，电话号码，等等)。用 `finger` 命令运行于你的 `login_name`，便会显示这些信息。

```
change -M 100 login_name
```

(= " change age ") 将用户的口令有效期设为 100 天。

```
quota username
```

```
setquota username
```

```
quotaon /dev/had
```

```
quotaoff /dev/had
```

管理用户磁盘限额的一组命令。一般在家中电脑上不用。“Disk quota”意思是每个用户使用磁盘空间的限制。这些命令分别是显示用户限额，设置用户限额，在一个给定的文件系统 (本例是 `/dev/hda`) 上启用限额系统，关闭限额系统。我所见过的典型的发行套件的缺省设置：对所有用户没有限制，所有文件系统上的限额系统都关闭。

```
kuser
```

(以 `root` 用户运行，在 X 终端上) 使用一个 GUI 程序管理用户和组。好用且基本包括你平常管理用户帐户的命令。

```
chmod perm filename
```

(=change mode)修改你的文件的访问权限(除非你是 root 用户,你可以修改任何文件)。你能够以三种方式(读())、写()、执行())访问三个级别(所有者())、文件所有者同组成员()、系统上其他人())的用户的文件。检查当前访问权限,使用:

```
ls -l filename
```

如果这个文件可以为所有用户以所有方式访问,其权限显示为:rw-rw-rw-
第一个三位表示文件所有者的权限,第二个表示拥有这个文件的组的权限,第三个表示其他人的权限。没有权限显示为“-”。

当设置权限时,以下符号被用到:“u”(=用户或文件所有者),“g”(拥有文件的组),“o”(其他人),“a”(所有人,如所有者,组和其他人),“=”(设置权限为),“+”(增加权限),“-”(取消权限),“r”(读文件的权限),“w”(写文件的权限,代表修改文件的权限),“x”(执行文件的权限)。

例如,这个命令为所有的用户增加读文件的权限:

```
chmod a+r junk
```

这个命令将取消所有用户执行文件的权限:

```
chmod o-x junk
```

你也可以去[这里](#),有更多的信息。

你可以用命令 umask (参阅 man umask) 为你创建的新文件设置缺省权限。

```
Chown new_ownership filename
```

```
Chgrp new_groupname filename
```

改变文件的所有者和组。在你复制一个文件供其他人使用时,你应当使用以上两个命令。只有文件的所有者才能删除它。

```
lsattr files
```

列表文件属性。不常用到,因为很多最有趣的属性还没有实现。属性可以用命令来修改。这些属性是:A(=修改文件时不要更新时间),S(=同步更新),a(=此文件只能追加),c(=在内核级上压缩文件,尚未实现),i(=不变的文件),d(=不可转储),s(=安全删除)和u(=不可删除,尚未实现)。一个有趣的用法是使一个文件不可删除,即使 root 用户也不行(除非设置者清除该属性)。

```
Sudo /sbin/shutdown -h now
```

(作为一个普通用户,会被提示输入用户口令)运行“shutdown”命令(或另外一个系统管理员授权你可以运行的命令)。使用,系统管理员可能授权指定用户执行指定命令,而不用出示 root 用户口令。/etc/sudoers 文件必须被置为类似如下内容:

```
my_login_name my_host_computer_name=/sbin/shutdown
```

```
pwck
```

```
grpck
```

(以 root 用户运行, 两个命令) 验证口令文件和组文件的完整性。

```
pwconv
grpconv
```

(以 root 用户运行) 可能你不需要这些命令。它们将旧格式的口令和组文件转换成更安全的“shadow”文件。

5.15 程序安装

```
rpm -ivg package_name-version.platform.rpm
```

(以 root 用户运行) 安装一个软件包 (选项“i”, 必须是破折号 (dash) 后面的第一个字母), 采用交互方式 (选项“v” = verbose) 并以数值方式显示安全进程 (选项“h”)。rpm 代表“RedHat 软件包管理器”

```
rpm -Uvh package_name-version.platform.rpm
```

(以 root 用户运行) 升级 (选项“U”, 必须是破折号 (dash) 后面的第一个字母) 一个软件包, 采用交互方式 (选项“v”) 并以数值方式显示安全进程 (选项“h”)。

```
rpm -ivh -force -nodep package_name-version.platform.rpm
```

(以 root 用户运行) 安装一个软件包, 忽略任何可能的冲突和包依存关系问题。

```
rpm -e package_name
```

(以 root 用户运行) 卸载 (选项“e” = 删除 (erase)) 软件包 package_name。请注意在软件包名称最后没有“-version.platform.rpm” (软件包名称与用来安装它的*.rpm 文件名一样, 只是没有了破折号、版本、平台和“rpm”)。

```
rpm -qpi package_name-version.platform.rpm
```

查询 (选项“”, 必须是破折号 (dash) 后面的第一个字母) 还没有安装的软件包 (选项“p”), 显示软件包包含的信息 (选项“i”)。

```
rpm -qpl package_name-version.platform.rpm
```

查询 (选项“”, 必须是破折号 (dash) 后面的第一个字母) 还没有安装的软件包 (选项“p”), 显示软件包包含的全部文件的列表 (选项“l”)。

```
rpm -qf a_file
```

查找文件“a_file”所属的已安装的软件包。如果你偶然删除了一个文件, 现在需要找到正确的软件包来重新安装, 这个命令很有用。

```
rpm -qi package_name
```

查询已安装的软件包，显示其自身的信息。请注意，请注意在软件包名称最后没有 “ - version.platform.rpm ”

```
rpm -qai | more
```

查询并显示系统上安装的所有软件包的信息。在我的简单的系统上，我安装了大约 600 多个软件包，很显然，我必须花很多时间来阅读它们的信息。要查一下你的软件包的数量，试试：

```
rpm -qa | grep -c ' '
```

要找出一个指定的软件包，试：

```
rpm -qa | grep -l the_string_to_find
```

(选项 -i 使 grep 命令忽略了字符的大小写，所以无论是大写，还是小写，都会匹配的)

```
rpm -Va
```

检验 (选项 “V”) 我的系统上所有的 (选项 “a”) 软件包。它列出自安装后被改动过的所有文件。下面是一些关于输出的解释：

.	测试通过
C	这是一个配置文件
S	MD5 校验失败
S	文件大小不同
L	符号链接已经改变
T	文件修改时间已经改变
D	设备文件被修改
U	拥有文件的用户已经改变
G	拥有文件的组已经改变
M	文件模式 (权限和/或文件类型) 已经被修改

kpackage

gnorpm

glint

(在 X 终端上，如果你想能够安装软件包，以 root 用户运行，3 个命令) Redhat 软件包管理器 (rpm) 的 GUI 前端。RH5.2 中 glint 的现在看来太旧了。gnorpm 是 “正式的” GUI 软件包安装程序，老的版本非常慢并且很难懂，但是新的版本 (RH7.0 中的) 有了很大的改善。kpackage 是 “正式的” KDE 程序，且一直都非常好用。用它们中的任何一个都以查看系统上安装了哪些软件包，不有哪些 Redhat CD 上有但还没安装，显示软件包的信息，如果你想的话还可以安装它们 (安装必须由 root 用户来进行)。

5.16 访问磁盘驱动器/分区

mount

关于装载驱动器的细节见[这里](#)。以下的命令展示一些例子。

```
mount -t auto /dev/fd0 /mnt/floppy
```

(以 root 用户运行) 装载软驱。/mnt/floppy 目录必须存在，而且是空，并不能是你的当前目录。在/etc/fstab 中不必有设置，因为你是 root 用户而且在命令行提供了所有需要的信息。文件系统的类型将被自动检测。

```
mount -t auto /dev/cdrom /mnt/cdrom
```

(以 root 用户运行) 装载 CD。你可能需要创建或修改/dev/cdrom 文件, 它取决于你的 CDRom 在哪儿。/mnt/cdrom 目录必须存在, 而且是空, 并不能是你的当前目录。

```
mount /mnt/floppy
```

(以普通用户或 root 用户运行) 以用户的身份装载软驱。在/etc/fstab 中必须设置。/mnt/floppy 目录不能是你的当前目录

```
mount /mnt/cdrom
```

(以普通用户或 root 用户运行) 以用户的身份装载 CD。在/etc/fstab 中必须设置。/mnt/cdrom 目录不能是你的当前目录

```
umount /mnt/floppy
```

卸载软驱。目录/mnt/floppy 不能是你 (或其他任何人) 的当前工作目录。取决于你的设置, 你可能不能卸载一个别人装载的驱动器。

```
Mount /mnt/hda1 /mnt/dos_drive1
```

从本机硬盘装载一个 DOS (MS Windows) 分区。

5.17 网络管理工具

```
netconf
```

(以 root 用户运行) 一个非常好的菜单驱动的网络设置工具。

```
ping machine_name
```

检验你是否能和其他机器相连 (给出机器名或 IP), 当结束的时候按<Ctrl>C (如果不按<Ctrl>C, 命令将一直执行) 象所有的 Linux 命令一样, ping 有许多选项, 包括 “ping of death” 攻击, 看起来你可以 ping 许多服务器, 导致它们死机 试一下选项 -f 和 -s。

```
route -n
```

显示内核路由表。

```
host host_to_find  
nslookup host_to_find  
dig ip_to_find
```

(三个命令, 用任意一个) 在缺省域名服务器 (DNS) 上查询一个 Internet 名字 (或 IP 地址) host_to_find。用这个方法你可以检查你的 DNS 是否工作。如果你只知道一个主机的 IP 地址, 你还可以查出它的名字。

```
tracert host_to_trace
```

查看你的信息到达 host_to_trace 的线路。(可以是主机名字或 IP 地址)

```
mtr host_to_trace
```

(以 root 用户运行) 一个强大而好用的工具, 结合了旧的 ping 和 traceroute 的功能 (RH 7.0)。

```
nmblookup -A ip_address
```

查看一个联网的 MS Windows 主机 (用 NetBIOS 名字) 的状态。这个命令与 Windows 的 nbtstat 命令是等价的。

```
ipfwadm -F -p m
```

(RH 5.2 命令, RH6.0 见下一个命令) 为保护本机设置防火墙 IP 转发策略。(不是很安全, 但是简单)。目的: 你的家庭网络的所有机器在外界看起来是一台十分繁忙的机器, 而且如你被允许从所有的计算机上同时浏览 Internet。

```
echo 1 > /proc/sys/net/ipv4/ip__forward
```

```
ipfwadm-wrapper -F -p deny
```

```
ipfwadm-wrapper -F -a -m -S xxx.xxx.xxx.0/24 -D 0.0.0.0/0
```

(三个命令, RH 6.0)。与前一个命令相同, 用你的家庭网络分配给你的机器的“C”类地址替换那些 x。更多的详细信息见[这里](#)。

```
ipchains -P forward DENY
```

```
ipchains -A forward -s xxx.xxx.xxx.0/24 -d 0.0.0.0/0 -j MASQ
```

(两个命令, RH 7.0) 与前面的命令一个, 但是在 RH 7.0 下工作。

```
ipchains -L
```

列出所有的防火墙规则。用来检查你的防火墙设置是否工作。

```
iptables -L
```

Linux 内核 2.4.x 使用新的防火墙“iptables”(IP 列表)。上面的例子列出防火墙规则。

```
firewall-config
```

(以 root 用户运行, 在 X 终端上) 建立你的自定义防火墙的 GUI 工具。

```
ifconfig
```

(以 root 用户运行) 显示当前激活的网络接口 (ethernet, ppp, 等等)。你的第一块网卡显示为 eth0, 第二块网卡为 eth1, 等。通过调制解调器的第一个端口为 ppp0, 第二个端口为 ppp1, 等等。“lo”是“loopback only”接口, 它总是处于激活状态。使用选项 (详见 ifconfig --help) 来设置接口。

```
ifup interface_name
```

(以普通用户运行/sbin/ifup) 启动一个网络接口，例如：

```
ifup eth0
ifup ppp0
ifup ppp1
```

只有当在 ppp 设置中授予权限 (使用 netconfig) 时，用户才能启动或关闭 ppp 接口。要启动一个 ppp 接口 (拨号连接)，我一般使用 KDE “K” 菜单中的 kppp (或在 X 终端上输入 kppp)。

```
/etc/rc.d/init.d/network restart
```

使用正常初始化脚本重新启动网络 (与系统引导过程中的一样)。这对于你只是手工修改了网络设置是很有用的。Init.d 中列出的其他的服 务也可以相似方法停止、启动或重新启动 (用 stop,start 或 restart 选项调用脚本)。

```
ifdown interface_name
```

(以普通用户运行/sbin/ifdown)。关闭网络接口，如：ifdown ppp0。也可参见前面的命令。

显示关于你的网络状态的信息。

(以用户运行，如果你希望在 X 终端上使用基于的界面)。结合了 traceroute 和 ping 的功能的网络诊断工具。在 RH 7.0 版本中。

```
nmap ip_number
```

显示 ip_number 机器上的端口映射。对于建立你的网络设置的安全真的十分有用，因为你可以看到开放的端口。nmap 包括在 RH7.0 “Linux PowerTools” CD 上，是一个非常方便的 GUI 前端，“nmapfe”。nmap 还能做操作系统的“指纹”。一般情况下，人们 (和他们的 ISPs) 不喜欢他们的计算机端口被扫描 (他们视它为可能是攻击前的搜索)。所以他们一发觉就会抱怨，如果你在你自己的计算机上练习怎样使用 nmap，你的 ISP (一般都找他们抱怨) 可能很快就会找你的。我是怎么知道的？

```
ethereal
```

(在 X 终端上，以 root 用户运行) 网络分析工具 查看通过你的计算机的网络流量。包括在 RH7.0 “Linux PowerTools” CD 上，在某些场合使用 ethereal 可能是违反规定的，在工作场合非授权使用可能会导致被解雇。

```
tcpdump -i ppp0 -a -x
```

(以 root 用户运行) 以 ASCII 码和十六进制显示所有通过第一个电话拨号接口的网络流量。可能会显示很多。是一个相当原始的工具，但它很实用，可以用来建造你所需要的自定义的监听和日志工具。

5.18 与音乐有关的命令

```
cdplay play 1
```

播放音频 CD 的第一首曲目。使用 cdplay 播放整个 CD。当你听够了的时候，使用 cdplay stop。

```
eject
```

得到一个免费的咖啡杯托盘。:)))。 (弹出 CDROM 托架)。这个命令对是缺省的，但也能用来从指定的安装点或设备来弹出可移除的介质。如：我可以从一个并口 (外置) 的 zipdrive 弹出 zipdisk (以 root 用户运行)，使用：eject /dev/sda4

```
play my_file.wav
```

播放一个 wav 音频文件。

```
rec my_file.wav
```

从麦克风录制一个 wav 音频文件。

```
mpg123 myfile.mp3
```

播放一个 mp3 文件。

```
mpg123 -w my_file.wav my_file.mp3
```

从一个 mp3 音频文件生成一个 wave 音频文件。如果你想把一些 mp3 文件写到一个正常的音频 CD 上，这个命令非常有用。你首先要将 mp3 文件转为 *.wav 格式。不要惊讶转换速度非常之慢。解压 mp3 文件是需要高强度的 CPU 运算的。

```
xmms &
```

(在 X 终端上) 非常好的 GUI mp3 播放器。

```
freeamp &
```

(在 X 终端上) 另一个非常好的 GUI mp3 播放器。

```
lame input_file output_file
```

MP3 编码器。你可能需要自己下载和安装这个程序 (Linux 标准发行版本不提供这个程序，是因为有关 MP3 压缩技术的专利的无妨争议)。

```
knapster
```

(在 X 终端上) 启动程序下载 MP3 文件，这些文件是其他 napster 用户共享出以供下载的。你可以将你的 MP3 文件共享出来。当使用它时，感觉相当 cool。Gnutella 和 FreeNet 将很快取代它们 -> 那种感觉更 cool。

```
cdparanoia -B "1-"
```

(CD 抓轨工具) 读出一个音频 CD 的内容, 将其转存成当前目录的 wave 音频文件, 每个文件一首曲目。 " 1- " 选项代表 " 从第一首曲目到最后一首 "。 - B 选项强制将每首曲目放到一个单独的文件。

```
grip&
```

(在 X 终端上) 一个 GUI 界面的抓轨工具 (见前面一个命令)。

```
playmidi my_file.mid
```

播放一个 MIDI 文件。 `playmidi -r my_file.mid` 将在屏幕上显示文本模式的效果。

```
sox audio_file another_format_audio_file  
=Sound exchange man sox sox
```

```
kscd
```

(在 X 终端上) CD 播放器。

```
kmidi
```

(在 X 终端上) MIDI 播放器。

```
kmid
```

(在 X 终端上) MIDI/卡拉 OK 播放器。

```
kmix
```

(在 X 终端上) 声音混响器。

```
studio&
```

(在 X 终端上) 音效工作室 编辑声音文件, 增加效果, 等等。你可以从 RH 7.x 的 PowerTools CD 上得到。

```
extace&
```

(在 X 终端上) 声音可视化工具。

```
festival -tts my_file.txt
```

读文件 `my_file.txt` 的内容 (ASCII 码文本)。 " festival " 是一个语音合成器, 发行在 RedHat 7.0 " Linux PowerTools " CD 中。要从命令行读出一些东西, 你需要启动 " festival ", 然后在 " festival> " 提示符处, 输入适当的命令 (" scheme " 语言解释器), 如在这个例子中 (黑体代表提示符):

```
festival  
festival>(SayText "good dog, really good dog")
```

```
festival>(quit)
```

5.19 与图形相关的命令

```
kghostview my_file.ps
```

(在 X 终端上) 在屏幕上显示一个 postscript (或 pdf) 文件。我也能用看起来比较旧的 ghostview 或 gv 来达到同样的效果。我也能够从视窗中打印 postscript 文件。

```
xpdf my_file.pdf
```

(在 X 终端上) 查看一个 pdf 文件。要查看 pdf 文件, 我更愿意使用 Adobe Acrobat Reader 的 Linux 版本 (会快一些)。

```
acroread my_file.pdf
```

你可以从 <http://www.adobe.com/products/acrobat/readstep2.html> 下载它。

```
enscript my_file.txt -U 2
```

将一个文本文件转换为 postscript 格式并在缺省打印机上打印出来。我还能将输出发送到一个 postscript 文件。

```
enscript my_file.txt -U 2 -o my_file.ps
```

选项 -U 2 使 enscript 在一张纸当打印 2 页逻辑页面, 这样可以节省纸张, 并带来方便, 压缩打印输出。你还可以选择每页纸 4 页逻辑页, 一页纸上打印得越多, 就越难认。Enscript 真的很灵活, 详见 man enscript, 在许多格式选项中进行选择。

```
ps2pdf my_file.ps my_file.pdf
```

从一个 postscript 文件生成一个 pdf (Adobe portable document format) 文件。

```
mpage -2 my_file.ps > new_file.ps
```

打印 postscript 文件 my_file.ps, 在一个物理页上输出两个逻辑页。将输出结果保存在文件 new_file.ps 中。

```
ps2ps file.ps new_file.ps
```

```
psnup -nup 2 -pletter new_file.ps new_file2.ps
```

生成一个物理页上包含两个逻辑页的文件的另一种方法。首先, 我用 “postscript distiller” 将一个 postscript 文件简化 (代价是文件变得大一点)。然后, 我用 psnup 工具产生 new_file2.ps 文件, 它每一个物理页包含两个逻辑页。我也可以将四或八个逻辑页放到一个物理页上。

```
gimp
```

(在 X 终端上) 一个看起来不怎么样但功能非常强大的图像处理器。要学习一段时间才能使用, 但它对艺术家来说真是太好了, 几乎没有什么不能处理的。用你的鼠标右键调出本地菜单, 学习怎样使用图层。将你的文件保存为本地的 gimp 文件格式 *.xcf (保存图层, 供以后编辑), 只有将其平面化 (flatten) 并保存为 png 格式 (或其他什么格式), 你才可以使用。 “学会如何做出正确的选择是关键。”

`gphoto`

(在 X 终端上) 强大的照片编辑器和相机图像检测程序。

`kpaint`

(在 X 终端上) 简单的位图画图程序 (“ 画笔 ” 类型) 。

`xfig`

(在 X 终端上) 一个简单的画图程序。对于制作基本的草图或流程图非常有用。

`dia`

(在 X 终端上) 一个用预制组件画流程图的工具。

`display my_picture`

(在 X 终端上) 显示一幅图片，只能浏览。你还可以输入 `display &`，然后从菜单中选择文件来查看图像，旋转，改变颜色，采用特定效果，等等。 `Display` 是 `ImageMagick` 软件包的一部分，在一起的还有下面描述的几个实用工具。

`identify -verbose my_picture`

给出关于图像文件 `my_picture` 的描述：格式、类型、分类、大小（以像素为单位）、颜色数量、大小（以字节为单位）、等等。

`convert -geometry 60x80 mypicture.gif out_small_picture.gif`

将一个图片转换为 60x80 像素尺寸。下面有一个关于如何在不同图形文件格式进行转换的例子。

`animate -delay 6x5 pic1 pic2 pic3`

连续显示两个或多个图片。在示例中，图片文件名为 `pic1,pic2,pic3`，图片的间隔时间为 0.06 秒，整个展示过程在 5 秒中之内反复进行。

`combin pic1 pic2 combined_pic.miff`

将两个或多个图像加到另一个图像中。例如你可以将一个图标加到每个图像中。

`montage -geometry 800x600+0+0 my_picture montage.niff`

从 `my_picture` 文件生成一个平铺的图像上，尺寸为 800x600，边框为 0x0。输出结果送到 `montage.miff`。

`zgv my_picture`

在一个 vga 屏幕上显示一个图片以供浏览。

```
giftopnm my_file.giff > my_file.pnm  
pnmtopng my_file.pnm > my_file.png
```

将一个有版权的图形文件转换为一个原始的、方便的 pnm 文件。然后将 pnm 文件转换为 png 文件，png 文件是 Internet 图片的更新更好的标准（技术上更好，再加上没有被 giff 专利拥有者控告的危险）。

```
xwd -out my_cupture_screen_file.xwd
```

（在 X 终端上）捕捉 X-windows 屏幕上的内容，并保存到 X-windows “转存” 图像文件（*.xwd）。以后你可以使用转换工具将 xwd 文件转换为你喜欢的格式。在 KDE 环境下，你可以使用键盘快捷键<Alt><PrintScreen>或<Ctrl><Alt><PrintScreen>来复制当前窗口工将整个桌面送到剪贴板。

```
convert my_capture_screen_file.xwd my_capture_screen.jpg
```

将 X-windows 屏幕转存文件*.xwd 转换为*.jpg 格式文件。这个转换工具能够对很多文件格式进行转换。

```
import -display 192.5.100.10:0 -window root my_file.jpeg
```

从 192.5.100.10 服务器上显示端口 0 上运行的 X-windows 上捕捉 root 屏幕的内容。输出文件是 my_file.jpeg（可以根据所给出的文件扩展名改变文件格式）。为了能够捕捉屏幕的内容，你必须要有对屏幕写的权限（在 X - 终端上运行命令 xhost + 可以将这一权限赋予任何用户）。选项见 man important。

```
ksnapshot
```

（在 X 终端上）基于 GUI 的捕捉屏幕内容的工具。

```
xine Frankenstein.avi &
```

（在 X 终端上）。从文件“frankenstein.avi”看电影。看起来你电视。

5.20 几个小游戏

你的系统上很可能安装了许多小游戏。这个只是从我的标准 Linux 发行版 CD 上的一个例子。

```
kpat
```

（在 X 终端上）单人纸牌游戏。sol（快）和 pysol（慢但是？）是另外的两个选择。我喜欢：sol -variation=freecell&

```
xboing
```

（在 X 终端上）非常好的弹球游戏。

xboard

(在 X 终端上) 国际象棋。对于我来说很好玩。

koquest &

(在 X 终端上) 和你的儿子进行一场权力的角逐。非常好的板上游戏。

kmines

(在 X 终端上) 扫雷。

civserver

civclient

(在 X 终端上) (在 X 终端上) 为 FreeCivization 游戏启动一个服务器 (第一个命令)。然后, 当服务器运行正常后, 启动一个客户端 (第二个命令)。有些人也启动其他的客户端程序和你玩。FreeCiv 在 RH7.0 CD 上。

fgfs

“Flight Gear” 飞行模拟器。

第 6 部分: Linux 基本的应用程序

Linux 新手管理员指南(中文版)

LINUX NEWBIE ADMINISTRATOR GUIDE (Simplified Chinese Version)

中文版本 1.0 由幻影快递 Linux 小组(SLT-Shadowfax Linux Team) 翻译整理

Linux Newbie Administrator 英文原版版权所有: Peter and Stan Klimas.

<http://sunsite.dk/linux-newbie>

中文版本版权所有: 幻影快递 Linux 小组 ShadowfaxLinux@hotmail.com

本章内容:

- 6.1 字处理程序
 - 6.1.1 StartOffice/Open Office 软件
 - 6.1.2 abiword 字处理程序
 - 6.1.3 kword
 - 6.1.4 klyx, lyx 和 latex
 - 6.1.5 WordNet(字典/词典/同义词/反义词查询)
- 6.2 电子表格
 - 6.2.1 oocalc
 - 6.2.2 gnumeric
 - 6.2.3 kspread
- 6.3 数据库
- 6.4 CAD
- 6.5 网络浏览器: Mozilla, Konqueror and Lynx
- 6.6 刻录 CD-R: cdrecord 和 cdparanoia
- 6.7 使用 gnuplot 自动画图

简介:

本章只是覆盖那些我们经常使用或者喜欢使用的应用程序。Linux 有成千上万个应用程序, 如果你不是很满意我们介绍的, 到这里可以找到更多:

<http://www.linuxapps.com/>

<http://stommel.tamu.edu/~baum/linuxlist/linuxlist/linuxlist.html>

<http://www.boutell.com/lsm/>

<http://www.linuxlinks.com/Software/>

6.1 字处理程序

6.1.1 StartOffice/Open Office 软件

OpenOffice 是一个完整的 Office 组件：字处理程序，电子表格程序，演示程序，画图程序，图形模块，数学公式编辑器等等。它包含了专门给 Linux 使用的最好最丰富的字处理程序和电子表格程序，特别推荐大家使用。OpenOffice 一般包含在最新发行的 Linux 安装光盘上（2002 年 11 月份以后）。如果你需要最新版本，可以直接到这里免费下载：

<http://www.openoffice.org/>

文件很大，大概 70MB 左右，如果你使用 56K Modem 下载可能不太现实。当前版本（非开发者使用版本）是 OpenOffice 1.0.1(2002 年 10 月)。OpenOffice 是跨平台的：可以在 Linux，MS Windows，Solaris 和 Mac OS X 上运行，在不同平台之间带有文件级的兼容性。

OpenOffice 简史

StarOffice 曾经是一个商业应用程序（“Star”分部，德国）。后来该程序被 Sun Microsystem 公司购买，源代码也在通用公共许可证（GPL）范围内贡献给开放源代码社区（2000 年 8 月）。然后，它被很多和 Sun 公司有关联或者 Sun 雇佣的程序员迅速开发。开放源代码的版本叫做“OpenOffice.org”。Sun 公司有些时候会根据比较稳定的版本发行自己的叫做“StarOffice”。所以，StarOffice 和 OpenOffice 基本上可以说是指相同的产品，在 StarOffice 里有时会带上一一些小的附加功能。出于一些注册商标的考虑，OpenOffice 正式的官方名称叫 OpenOffice.org。

描述

OpenOffice 组件看起来非常象微软 Windows 下的 Office 组件。这个组件包括很多丰富功能，程序也比较大（可能要求一定的磁盘空间，内存和 CPU 的速度），很多内置的特性（可能需要一些特别的鼠标点击才能使用一些特性）。如果你的计算机内存小于 64M，OpenOffice 可能会有很多麻烦，不值得安装，内存更大一些会好得多。OpenOffice 非常稳定，尽管有些时候会显示一些奇怪的字符在我的屏幕上。它和微软的 Office 有文件级的兼容性：可以读写微软的 Word，Excel 和 PowerPoint 文件格式。原则上，OpenOffice 使用最基本的 XML 开放文件格式：文本和图形压缩在同一个文件里。当我解压文件的时候（`unzip my_file.sxv`），我可以完全解压原来的图形 - 这是微软的 Office 可能无法做到的（有些时候会对文件管理带来严重后果）。

OpenOffice 看起来不像 Linux 其他 Office 程序那么“性感”。但是，它是真正可以实际应用的程序，它的完美性还在快速提高。简而言之，我们非常推荐使用 OpenOffice/StarOffice 来符号你日常的办公需要。就功能上的比较，它几乎覆盖了你在微软的 Word 和 Excel 上可以实现的功能，还附加了一些在微软的 Office 一直缺少的功能。

总体来说，OpenOffice 是一个开放和根据标准文件格式设计的完美程序（只有那些没有远见或者死脑筋的决策人员才会坚持将他们的重要数据放在只能用一个特定公司发行的软件才能阅读的文件里）。这个文件格式可以在关键场合下使用，因为它可以使用第三方的软件软件根据来解读。

安装

OpenOffice 的安装可能稍微有点麻烦，过程大概如下：

- 确认你有足够的硬盘空间。使用 `df` 命令检查可用的硬盘空间：

```
df -h
```

这将显示一个可用硬盘空间的报告。你最少需要 350MB 的硬盘空间（安装完成后，大概有 100M 的空间可以重新释放）

- 解压下载的安装文件。我使用 root 用户在 /usr/local 的目录下执行“本地服务器”安装，但是你也可以选择你的主目录 /home/your_login_name 做个人安装：

```
cd /usr/local
tar -xvf StarOffice5.2.tgz
```

使用你下载文件的实际文件名替代 StarOffice5.2.tgz (StarOffice5.2.tgz 是我下载时使用的文件名)

- 作为 root 用户，在本地服务器，带 /net 选择项，运行安装程序：

```
cd /usr/local/OpenOffice641
./setup /net
```

[如果不带“/net”或者“-net”参数，OpenOffice 将运行个人安装（装在你自己的主目录下！），这样就只有一个用户可以运行它，而且你的主目录会被挤得满满的]

- 网络安装完成后，每个用户必须执行自己得安装，将一些个人得文件放到自己得主目录下。可以通过以下实现（按普通用户，不带 /net 参数）：

```
cd /usr/local/OpenOffice641
./setup
```

在命令行（X 终端）上运行任何 OpenOffice 的程序，我可以使用的：

ooffice&	（字处理）
oowriter&	（同上）
oocalc&	（电子表格）
oodraw&	（矢量画图程序）
oimpress&	（演示程序，类似于微软的 PowerPoint）
oosetup&	（安装程序）
oopadmin&	（打印机管理程序）
oomath&	（数学格式编辑器，一般不单独运行）

小技巧

OpenOffice 附带一个扩展，内容敏感的帮助程序（按 <F1>）。这里我们收集了一些在使用字处理程序时的快速参考（从 2002 年 11 月后有效）：

- 考虑使用“Styles”（风格）来格式化文件。除非你写一个非常简单的文件，避免“物理格式化”。按 <F11> 可以得到完整的 Style 列表。注意看菜单条（左手侧）的下拉框，有一个可以在文件里使用的段落风格列表。

- “风格”可以分成以下几个部分：

字符风格	(对一个字符或者一组字符有效)
段落风格	(段落的范围从一个<ENTER>到另外一个<ENTER>)
帧的风格	(一个帧一般代表一个文本框，一个图形或者一张图片)
页面风格	
罗列风格	(数字罗列或者标号罗列)

这里，最重要的是段落风格。我用它来格式章节的标题，表格的标题等等。要在段落里使用指定风格，可以把光标移到要修改的段落，然后双击在“Stylelist”（风格列表）的Style名称。要修改一个风格（或者创建一个新的风格），可以使用菜单“Format” - “Style” - “Catalog”。

- 如果要给章节的题目编号，可以在“Tools” - “Outline Numbering”下设置编号系统。
- 要插入一个标题（表格，图形，等等），我可以把光标移动到表格（或者图形）上，然后使用菜单“Insert” - “Caption”，我也可以通过对话框里的“Options”按钮来修改标题的编号级别。当然，我也可以等到当我点击标题出现对话框的时候再修改。
- 要修改所有的域（目录编号，标题，目录表等等），我按住<CTRL>接着按<F9>。这将选择整个文件然后更新整个文件。另外一个办法是，使用菜单“Tools” - “Update” - “Update All”
- 显示文件的整体轮廓，按<F5>
- 修改缺省的文件模板，创建一个“default”的文件再把它保存为“Default”（“File” - “Templates” - “Save”）
- 在你输入的同时检查拼写，可以通过“Tools” - “SpellCheck” - “AutoSpellCheck”激活拼写检查功能。要检查一个词的拼写，把光标移动到那里，按<F7>
- 对于输入自动完成功能，当选择项出现的时候选择所要的选项，按<ENTER>
- 表格单元可以被分割（如果你从微软的Word转过来，你可以再次使用表格单元分割）
- 按<F2>弹出一个“公式条”。在表格里，你可以用它来计算在其他表格单元里的值。状态栏显示的是表格单元的参照值。在表格之外，你可以使用“公式条”来实现一些实时的计算。
- 你可以通过在段落开始的地方输入“fn”然后按<F3>来插入一段预先做好格式和编号的“引用”文本
- 信件合并功能在以下的链接可以找到：

http://documentation.openoffice.org/HOW_T0/word_processing/writer2_EN.html

- 以下是一些有用的快捷键功能（大部分在 OpenOffice 下有效，不仅仅是字处理程序）

<CTRL>X 剪切	<F1> 帮助	<CTRL><SPACE> 不间断的空格
<CTRL>C 复制	<F2> 公式计算	<CTRL><SHIFT> 不间断的横杠-
<CTRL>V 粘贴	<F3> 自动文本输入	<CTRL><ENTER> 强制换页
<CTRL>A 选择所有	<F4> 数据源查看/关闭	<SHIFT><ENTER> 不改变段落的换行
<CTRL>F 查找	<SHIFT><F4> 在电子表格程序中切换绝对/相对参考值	<CTRL><SHIFT><ENTER> 强制换列（对于多列的文本有效）
<CTRL>Z 恢复 Undo	<F5> 导引开/关	<INSERT> 插入模式/覆盖模式转换
<CTRL><SHIFT>P 超级脚本	<F7> 拼写检查	<HOME> 到光标所在行的开始
<CTRL><SHIFT>B 子脚本	<F9> 更新域（或者重新计算电子表格）	<END> 到关闭所在行的末尾
	<F11> 风格列表开/关	<CTRL><HOME> 到文档的开始
	<F12> 标号开/关	<CTRL><END> 到文档的末尾

使用鼠标做选择：

<鼠标左键>： 选择文本，表格单元等
 <SHIFT><鼠标左键>： 扩展当前的选择

使用鼠标做拖曳：

<鼠标左键>： 拖曳和移动选择的文本
 <CTRL><鼠标左键>： 复制选择的文本到对应的位置

6.1.2 abiword 字处理程序

在 X 终端里输入 abiword 或者 Abiword 可以启动该程序，你可以在以下链接下载最新版本：

<http://www.abisource.com>

它是一个很小又非常不错的字处理程序。基本上满足一些基本的字处理需要。尽管功能可能还不够丰富，但是对我非常有用，举个例子，它和 StarOffice 一样支持即时拼写检查功能，但是不需要安装 StarOffice 那么多的程序。现在这个程序还在积极开放中，在 Linux 下和微软的 Windows 下都有效。

6.1.3 kword

kword 还在开发中，我们不建议在重要场合使用，因为程序有时会崩溃。然而，它应该是三个（在通用开放许可证下的）主要 Office 组件里看起来最酷的一个。它是以帧为基础的，象“framemaker”（说说而已，我没有用过 framemaker）一样非常容易使用，功能强大非常适合于桌面出版（当然比 abiword 功能更丰富）。

按其他语言运行 kword，我可以这样做（在 X 终端，运行 kword 的荷兰语言版本）：

```
exec sh -c "KDE_LANG=nl kword" &
```

以下是一些有用的“标准”快捷键。这些快捷键适用于任何 KDE 环境下的应用程序：

<ALT>	访问顶级菜单
<ALT><一个字符>	快速访问顶级菜单对应的项目。字符对应于顶级菜单中带下划线的字符。举例（英语菜单）： <ALT>F — File “文件” 菜单 <ALT>E -- Edit “编辑” 菜单 <ALT>V -- View “查看” 菜单 <ALT>I -- Insert “插入” 菜单 <ALT>O -- Format “格式” 菜单
<CTRL>X	剪切
<CTRL>C	复制
<CTRL>V	粘贴
<CTRL>A	选择所有

6.1.4 klyx, lyx 和 latex

（在 X-Windows 终端输入 lyx）。lyx 是 latex 的前端程序（WYSIWYG，所见即所得，在 X-Windows 底下运行）。[还有一个程序叫 klyx，是 lyx 对应 KDE 桌面的变种，不过现在已经不再更新了。]

latex 很多年以来一直是资料准备和排版输入的重要程序，在学术界应用非常广泛，因为它可以非常方便的处理数学公式，管理非常大的文件，等等。

好消息是，即使你不知道什么是 latex，你仍然可以方便地使用 lyx。尽管 lyx 的哲学观点有点不同于我们常见的字处理程序，lyx 仍然是一个非常好的字处理程序，这可能需要一下调整你现在的观念。Latex(lyx)的哲学观点是先输入你的文本，接着定义所要的“风格”，然后把格式化的工作留给排版程序。这意味着你不需要手工调整间距（词与词之间的间隔，段落间的空行，章节等等）。当你的输入工作完成后，你“编译”你的文本然后产生一个设备独立文件（Device Independent File，“*.dvi”）。dvi 文件可以使用 dvi 阅读器来阅读和打印。输出的结果一般来说非常出色，但是“文件生成”的过程对比于常用的字处理程序的确有点麻烦。

Latex 的强项是打印输出的质量非常出色。它可以用来处理冗长而复杂的资料（技术文档，数学，等等），几乎所有的外国字符都有效（包括一些非常罕见的标志），而且在很多不同的平台上兼容，文件格式也应用广泛。弱点是，使用起来相对复杂。

Lyx 是免费的，包含在你的 Mandrake 或者 RedHat 的安装光盘上供你免费尝试。你也可以到这里下载：

http://idirect.linuxberg.com/kdehtml/off_word.html

几乎所有的 Linux 程序你都可以从 Linuxberg.com 网站上或者其他的 Linux 软件仓库下载。

如果除了容易使用的 lyx，你还想试一试更直接，专业的 latex，这里是一些简单的入门介绍：

- 用你经常使用的纯文本编辑器创建一个 latex 文档，做拼写检查，等等。然后，把文本文件存为扩展名为 “ *.tex ” 的文件。可以参考我后面提供的例子。
- 激活 Latex，把这个文本文件 “ 编译 ” 成一个 “ *.dvi ” 的文件。

```
Latex my_latex_file.tex
```

- 通过运行 dvi 把上一条命令产生的文件打印输出到 postscript 的应用程序里，常用的办法是把输出送到 lpr 打印机上

```
dvips my_file.dvi
```

- 你也可以使用以下命令把输出存为 postscript 文件格式

```
dvips -o output_file.ps my_file.dvi
```

可选项 -o 用于引导输出文件

- 你也可以使用以下命令创建一个 pdf 文件

```
dvipdf my_file.dvi output_file.pdf
```

或者

```
ps2pdf my_file.ps my_file.pdf
```

- 你可以阅读以下任何文件(dvi, ps, pdf)，举例（在 X 终端上）

```
kdvi my_file.dvi&
```

或者

```
kghostview my_file.pdf&
```

下页是一个完整的 Latex 文件的例子：

```

% Any line starting with "%" is a comment.
% "\" (backslash) is a special Latex character which introduces a Latex
% command.
\documentclass[10pt]{article}
\begin{document}
% Three commands are present in every Latex document. Two of them are
% above and one at the very end of this sample document.
This is a simple document to try \LaTeX. Use your favorite plain text
editor to type in your text. See how the command \LaTeX produces the
\LaTeX logo. Here is the end of the first paragraph.
Here starts the second paragraph (use one or more empty lines in your
input file to introduce a new paragraph).
The document class of this sample is ``article'' and it is defined at the
very beginning of the document. Other popular classes are ``report'',
``book'' and ``letter''.
Please note that the double quote is hardly ever used, utilize
two ` to begin a quote and two ' to close it. This nicely formats the
opening and closing quotes.
Here are different typefaces:
{\rm This is also roman typeface. It is the default typeface.}
{\bf This is bold typeface. }
{\em This is emphasize (italic) typeface.}
{\sl This is slanted typeface, which is different from the italic.}
{\tt This is typewriter typeface.}
{\sf This is sans serif typeface.}
{\sc This is small caps style.}
You can itemize things:
\begin{itemize}
\item one
\item two
\item three
\end{itemize}
You can also enumerate things:
\begin{enumerate}
\item one
\item two
\item three
\end{enumerate}
Try some foreign letters and symbols:
\aa \AA \o \O \l \L \ss \ae \AE \oe \OE \pounds \copyright \dag \ddag \S
\P. There are also three dashes of different length: - -- ---.
Try some accents over the letter ``a'': \'{a} \`{a} \"{a} \^{a} \~{a}
\={a} \. {a} \b{a} \c{a} \d{a} \H{a} \t{a} \u{a} \v{a}. Other letters can
be accented in a similar way.
The pair of ``\$' marks a math context. Many special characters are
available only in the ``math'' context. For example, try the Greek
alphabet:
Small: $ \alpha \beta \gamma \delta \epsilon \varepsilon \zeta \eta
\theta \vartheta \iota \kappa \lambda \lambda \mu \nu \xi \omicron \pi \varpi
\rho \varrho \sigma \varsigma \tau \upsilon \phi \varphi
\chi \psi \omega $
Capital: $ A B \Gamma \Delta E Z H \Theta I K \Lambda M \Xi \Pi P
\Sigma T \Upsilon \Phi X \Psi \Omega $
Try some equations: $ x^{y+1} + \sqrt{p \times q} = z_{try\_subscripts} $
\begin{center}
$ \frac{x \times y}{x/2+1} = \frac{1}{3} $
\end{center}
\LaTeX math commands are very similar to those in the old ``Word Perfect''
equation editor.
Use the verbatim mode to print the 10 special symbols which normally have
special meaning in \LaTeX: \verb|%\${}_{#&^~|}. The special symbols must be
contained between any two identical characters which in the example above
is |. Most of these special symbols can also be printed by preceding the
character with a backslash: \% \$ \{ \} \_ \# \& \^..
% This command ends the document (this is the third one that *must* be
% present in every document).
\end{document}

```

6.1.5 WordNet(字典/词典/同义词/反义词查询)

我一般使用 WordNet 作为字典/词典（在文本终端输入 wn）。在 RedHat 7.0 的安装光盘里没有，所以我必须去下载和自己安装（大概 10MB 左右），很不错的软件。看看以下连接：

<http://www.cogsci.princeton.edu/~wn/>

在 RedHat8.0 里，WordNet 是预先安装好的，可以用一个前端的图形用户界面程序启动它：

```
ktheasurus&
```

6.2 电子表格

6.2.1 oocalc

我现在一般使用一个很不错的电子表格程序叫 calc，它是 OpenOffice.org 的一个部分。你也可以在你熟悉的桌面模式下通过点击以下的菜单来运行它：

“K” - “Office” - “OpenOffice.org” - “OpenOffice.org Calc”

或者在 X 终端上输入命令：

```
oocalc&
```

我是一个经常使用电子表格的用户，所以我经常留心一些比较成熟的电子表格程序。以我的观点，Linux 目前还缺乏一个比较好的电子表格程序，但是 oocalc 已经可以做大部分工作了。

在 oocalc 里，用户甚至可以定义他们自己的函数。举例，我可以在编辑器里（“Tools” - “Macros” - “Edit”）写以下函数：

```
REM ***** BASIC *****  
REM This function calculates an area of a donut with radii r1 and r2  
Function my_function(r1,r2)  
REM return the value using the variable called like the function  
my_function=abs(pi()*r1^2 - pi()*r2^2)  
End function
```

然后，我就可以在电子表格文件里使用这个函数了，比如：`=my_function(2,3)`

到现在为止（2002 年 11 月），oocalc 可以支持多达 256 个列（A-IV），32000 行（1-32000），256 个表格（表格 1 到表格 256，表格可以被重新命名）。所能支持的行和列的数量还在增加之中。它还有几百个内置的函数（覆盖和兼容你在微软的 EXCEL 可以找到的所有函数）。

用户界面有时候看起来不那么美观（比较微软的 EXCEL）。举例，把一系列数据加到图表里需要选择一定范围里的数据，然后使用鼠标把它们拖到图表里。还有，图表组件（对应 XY 二维数据）支持两个 Y 轴，两个 X 轴，多种线的类型，数据的位图，错误条显示等等。如果你下决心学习它，功能真的很强大。我的感觉：你的鼠标点击要很注意按照要求来做。

6.2.2 gnumeric

在 X 终端下很不错的电子表格程序，是 GNOME 的一部分，包含在标准的 RedHat 的安装盘里（RedHat6.0 或者更高的版本）。尽管还有不少地方需要完善，这个程序还是非常值得一用的。Gnumeric 还在积极的开发中，很有潜力在将来成为最出色的电子表格程序。它已经有很多内置的函数，但是，最头痛的事情是，打印功能有些时候很不稳定。

6.2.3 kspread

kspread 是另外一个强烈推荐电子表格程序。从 KDE2.0 之后一直是 KDE(Koffice)的一部分。不过还不推荐在重要场合使用。Gnumeric 和 kspread 的文件格式都是标准的 XML 文件（已经是标准，属于下一代的增强的 HTML），使用这个文件格式肯定是绝佳的新闻。如果你以前经常遇到微软 EXCEL 的文件格式不兼容问题。和 HTML 一样，XML 对人类是易读的。因为电子表格文件是压缩的，如果我要在终端上阅读该文件，我可以这样做：

```
zless my_gnumeric_file.gnumeric
```

以下是一个电子表格的文件尺寸比较（只包含一个公式，复制 10,000 次），可以看出对人类易读的文件格式并非一定得付出其他代价，而微软的 EXCEL 文件格式里显然充满了太多没用的垃圾：

文件格式	文件尺寸 (BYTES)
Gnumeric 1.09	27,136
Oocalc 1.01	57,756
Kspread	90,560
MS EXCEL	549,888
QuattroPro 9 for MS Windows	155,648

6.3 数据库

如果你是一个喜欢数据库的人，你会发觉 Linux 在这方面覆盖得很全面。

postgreSQL

一个功能强大的数据库，在 Mandrake 和 RedHat 的安装光盘上有（免费，在 BSD 许可证下无限制使用）

mySQL

参见 <http://www.mysql.com>。通用公开许可证下的数据库，比 PostgreSQL 简单和容易使用，功能非常全面。受到很多数据库开发人员的青睐。和 PostgreSQL 相似，mySQL 不仅仅是一个简单的个人数据库，所以，可能要花点力气去学习。

除了以上，还有一些商用数据库免费提供给个人使用，比如：

Sybase for Linux :

http://www.sybase.com:80/sqlserver/linux/aselinux_install.html

Interbase :

<http://www.interbase.com/downloads/products.html>

Oracle for Linux:

http://platforms.oracle.com/linux/index_lin.htm

关于Oracle for Linux的信息帮助, 参考:

<http://jordan.fortwayne.com/oracle/index.html>

6.4 CAD

QCAD (GPL):

<http://www.qcad.org>

(比较简单, 但是对于小的画图或者学习很有用)

OCTREE

http://www.octree.de/html/frames/eng/f_octree.htm

(对于非商业用途, 免费)

VariCAD

<http://www.variacad.com/>

(专有商用软件, 15天试用期版本可以免费下载, 非常好的机械设计软件)

VAR KON (LGPL):

<http://www.varkon.com/>

Microstation

<http://www.microstation.com/academic/products/linux.htm>

(专有商用软件, Microstation的专业版本, 非常出色, 比AutoCAD软件还要好)

还有一个软件叫做“LinuxCAD”, 不过, 看起来实在不怎么样 (不推荐使用)。

6.5 网络浏览器 : Mozilla, Konqueror and Lynx

所有最新发行的 Linux 版本 (2002 年 11 月以后) 都包含主要的网络浏览器 “ Mozilla ”, 在 X 终端上输入命令 :

```
mozilla&
```

这是一个绝对顶级的浏览器, 你已经不会再要求更多了 (除了速度可能要提高一些以外)。它是著名的 NetScape 浏览器的新生代。

其他的网络浏览器还有 : 基于 KDE 的 konqueror 和 galeon。我一般使用 galeon。

如果你的 ISP 的速度实在太慢，你可能会倾向于使用基于文本的网络浏览器：

lynx

不要期望它看起来能和其他花里胡哨的图形界面下的网络浏览器一样。它是基于文本模式的。它的好处是，稳定而且速度极快。最适合的场合：当你要快速浏览一个 HTML 文件的时候。

在一些老版本的 Linux 上，你可能还装有一个 4.xx 版本的 NetScape 浏览器（“一个真实可靠的浏览器”）。可以在 X 终端上使用以下命令运行它：

netscape&

如果要编辑 html 文件（包含这本指南），我一般使用 Mozilla(所见即所得，浏览或者代码浏览)，或者 netscape，或者 WebMaker。

6.6 刻录 CD-R:cdrecord 和 cdparanoia

特别声明：

复制任何版权保护的介质是非法行为。不要在任何非法场合下使用以下指令。

介绍

刻录光盘以前的确很麻烦，但是最近我只要使用以下命令（root 用户，在 X 终端上）：

xcdroast&

如果你可以运行这个程序，那么我也可以。你就不需要读更多了。

也许 xcdroast 在你的系统上不能正常工作，那么以下这些我以前经常使用的步骤可能对你有帮助。要特别注意的是，xcdroast 只是以下描述的这些命令的图形用户界面前端工具，如果这些命令在你的系统上不能正常工作，那么 xcdroast 也不能正常工作。我的所有抱怨是，我那个号称“12x8x32x” CDRW，没有商标的便宜货可以以最高 12x 的速度写 CDR 盘片，但是对于 CDRW 只能以 2x 的速度读写（而不是号称的 8x），而刻录音乐 CD 的时候居然只能达到 1x 的速度，不管我怎么配置都白费力气。

设置

- 所有的设置必须要使用 root 帐号。最新发行的版本（比如 RedHat8.0）可能根本不需要设置。
- 检查一下你的 boot 文件，看看是否你的 ide-scsi 驱动器的这个参数已经传递给内核程序“hdb=ide-scsi”。需要的话，在以下文件里：

/etc/lilo.conf 或者 /boot/grub/grub.conf（取决于你使用哪一个启动程序）

加入可选项“hdb=ide-scsi”，这样，文件中对应的行看起来如下：

```
append="hdb=ide-scsi"
```

(对于/etc/lilo.conf, 在linuxd的image那一个部分)

```
kernel /boot/vmlinuz-2.4.7-10 ro root=/dev/hda6 hdb=ide-scsi  
( 对应于/boot/grub/grub.conf )
```

如果你的CD刻录机不在“hdb”(第一个IDE接口的第二个驱动器), 你可能需要自己做一些调整。它让你的IDE-ATAPI CD刻录机在Linux系统里看起来象是一个scsi设备(它其实不是真的scsi设备, 而是IDE设备, 只不过假装成scsi设备)。当/etc/lilo.conf修改完成后, 运行一遍lilo。对于grub, 不需要重新运行。

- 追加loop设备如果这个设备文件不在你的/dev目录下, 这不是硬性规定, 但是如果你要刻录数据CD, 这个功能很不错。loop设备可以直接挂载一个CD映像文件(看起来好像已经是文件系统的一部分)来检查CD的内容。当我完成RedHat的安装后, loop设备并不在我的系统上。所以我使用以下办法来自己追加它:

```
cd /dev/  
./MAKEDEV loop
```

- 确认需要的程序模块都被加载到系统内核里

```
lsmod
```

如果有必要, 把以下这两行内容加入到文件 /etc/rc.d/rc.local, 这样所以要求的内核模块在计算机启动的时候都会被自动加载。

```
/sbin/insmod ide-scsi  
/sbin/insmod loop
```

这两个内核模块对于IDE驱动器模拟SCSI设备, 以及loop设备的支持是必须的。

- 检查, 创建和修改设备文件 /dev/cdrom, 确保它指向正确的设备, 大概如下:

```
ls -l /dev/cdrom  
ln -s /dev/scd0 /dev/cdrom (如果有必要)
```

为什么要这样做的原因是, /dev/cdrom可能指到其他的IDE设备上去了(比如, 很有可能是/dev/hdb), 既然你的CD刻录机已经在scsi设备的模拟模式下, 这个修改是必须的。

你可能想把/dev/cdwriter也指到/dev/cdrom上, 如下所示。但是如果只有一个CD驱动器就没有必要。

```
ln -s /dev/cdwriter /dev/cdrom
```

- 为了使/etc/lilo.conf的修改开始有效, 重新启动计算机。检查一下CD刻录机是否能够正常读取数据。
- 检查cdrecord程序是否已经被安装, 举例:

```
cdrecord -scanbus
```

如果没有安装,可以从你经常使用的Linux下载网站(比如<http://www.linuxberg.com/>)下载cdrecord程序。接着安装源代码,编译,安装程序,然后创建一个连接指到可执行程序上,这样启动程序的时候就简便多了。如果你可以找到一个*.rpm格式的文件,安装起来就更简单了。

```
cd /usr/local
tar -xvzf /the_path_to_which_you_downloaded/cdrecord-1.6.1.tar.gz
ls
cd cdrecord-1.6.1
make
make install
ls /opt/schily/bin/
ln -s /opt/schily/bin/* /usr/local/
```

cdrecord是一个用于刻录CD最实际好用的命令行工具。目前是有不少图形用户界面的CD刻录程序,但是如果最底层的cdrecord不能正常工作,这些工具都没有实际用处。我的建议是,在适当时候使用命令行。这会让你更好理解它是如何工作的,而且更加稳定和灵活。如果你需要制造CD的封面,安装和使用基于通信用户界面的前端程序当然更好一些,也使原来学习Windows的用户更方便上手。

- 看看你的CD刻录机是否已经被系统识别,如果是,应该在输入以下命令后显示出来:

```
cdrecord -scanbus
```

制作数据CD

先创建一个CD映像文件包含你要刻录的数据:

```
mkisofs -r -o cd_image input_data_directory
```

以上命令将产生一个ISO-9660格式的(ISO标准的9660类型文件系统)文件,该文件包含所有在input_data_directory目录下的所有文件,这时,这个文件只是在硬盘上的一个普通文件。这个文件就是我制作CD要用到的一个映像文件。可选项-o指示接下来的一个参数是输出文件的文件名。参赛-f激活ISO协议的“Rock Ridge”扩展,这样,文件的属性能够被存储下来,而且所有的文件设置成可读(所有用户可读,不仅仅是文件的拥有者)。文件名被缩减成8.3 DOS文件名(8个字符文件名,3个字符扩展名)格式(尽管Linux支持的ISO-9660“Rock Ridge”扩展,它也可以保留所有的文件的权限和全文件名),这是让这个文件系统可以跨平台(DOS, Windows, Linux, Unix等等)使用的办法,对用户来说也更方便一些。

输入数据目录input_data_directory下的文件可以从系统中的不同目录产生一个标志链接,然后再合并起来(因为不需要复制数据,可以节约硬盘空间)。如果你决定这样做,你必须使用参赛 -f 告诉命令mkisofs允许使用标志链接,

```
mkisofs -r -f -o cd_image input_data_directory_containing_symlinks
```

你可能想要通过把CD映像文件挂载到loop设备上来检查CD的内容,使用以下命令:

```
mount -t iso9660 /dev/loop0 cd_image /mnt/cdrom
```

[现在文件的内容应该出现再目录/mnt/cdrom下面]

```
cd /mnt/cdrom
```

[通过loop设备检查文件的内容]

- 如果以上一切正常，你可以开始刻录你的数据CD了

```
cdrecord -v speed=8 dev=0,0,0 -data cd_image
```

“ dev= ” 的第一个数字表示scsi的总线号（第一条总线0，第二条总线1，依此类推），第二个数字表示在该scsi总线上的设备号（从0到7），第三个数字表示scsi设备的lun号码（一般是0）。你必须知道确切的号码，前面两个数字可以通过运行“ cdrecord -scanbus ” 来获得，第三个数字总是“ 0 ”。确认你使用了正确的数字，否则你可能写到错误的设备上毁坏你的数据。

在刻录CD-R的时候，尽量避免运行一些CPU资源消耗很大的任务（比如在硬盘上删除或者创建一个很大的文件），否则可能会中途出错。象我那老版本的RedHat6.2，在我刻录CD的时候，不允许我运行任何任务。

类似的做法，我也可以把因特网上下载下来的CD ISO映像文件刻录到CD上。比如：

（Debian的FAQ里）刻录一个映像文件“ binary-i386-1.iso ” 的推荐做法是：

```
nice --18 cdrecord -eject -v speed=2 dev=0,6,0 -data -pad binary-i386-1.iso
```

这个命令指定以优先级别18（很高的任务级别）的运行CD刻录程序，可以减少发生错误的可能性。

制作音乐CD

- 在刻录CD之前，必须确认音轨文件是以下文件格式之一：*.cdr（我猜可能就是*.cdda.raw的格式），*.wav（音频文件）和*.au。
- sox应用程序可以在不同的声音文件进行互相转换（sox可以识别很多不同的声音文件格式）。举例，我可以把“*.wav”文件转换成“*.cdr”文件：

```
sox my_file.wav my_file.cdr
```

你不需要手工做文件格式转换 – cdrecord直接支持 *.wav 和 *.au 格式（它会自动把 *.wav 文件和 *.au 文件转换成 *.cdr 文件）。这个非常方便，因为音频文件一般来说比较大。

- 音乐CD不包含任何文件系统，按“原始数据”格式存储。这意味着你不能挂载一个音乐CD。还有，每个音轨都是单独存储的，看起来更像是CD上的不同分区。

- 如果你要把音乐CD上的内容写到硬盘上的对应文件里（典型的文件格式比如*.raw或者*.wav），你需要一个“CD刮写工具”。一个常见的程序是“cdparanoia”。

如果cdparanoia没有安装，你可以自己下载安装：

```
cd /usr/local
tar -xvzf /the_path_to_which_you_downloaded/cdparanoia-III-alpha9.6.src.tgz
ls
cd cdparanoia-III-alpha9.6
./configure
make
make install
```

从音乐CD上“刮写”第一个音轨，我可以使用的：

```
cdparanoia 1
```

该命令将在当前目录下生成一个叫做“cdda.wav”的音频文件。

如果要“刮写”第一和第二个音轨存为“原始数据”格式，使用命令：

```
cdparanoia -B -p "1-2"
```

可选项 -B 指示使用批处理模式（batch mode），这样每个音轨都会存放在不同的文件里（这可能是你真正想要的，否则所有音轨都被放到同一个输出文件里）。可选项 -p 指示输出使用“原始数据”格式。对应的文件名是：track1.cdda.raw, track2.cdda.raw。

以下命令按4x的速度“刮写”所有的音轨，每个音轨存为不同的*.wav文件：

```
cdparanoia -S 4 -B "1-"
```

注意你有足够的硬盘空间。如果你设置双系统启动，你也可以使用DOS上的分区。

- 如果要写音频文件到CD-R(W)上，我可以使用的：

```
cdrecord -v speed=8 dev=0,0,0 -pad -dao -audio track*
```

- 有些音乐CD在各个音轨之间没有间隔。要复制这种类型的CD，最好使用工具“cdrdao”。

```
cdrdao read-cd --device 0,0,0 --buffers 64 --driver generic-mmc-raw --read-raw
toc-file.toc
```

- 把以上刚生成的文件刻录到CD上，可以使用：

```
cdrdao write --device 0,0,0 --buffers 64 --driver generic-mmc --speed 12 toc-
file.toc
```

- 还要注意的CDRW(支持多次读写)盘片。有些家用音响不能播放CDRW的盘片。比如我家里的JVC播放机就不能播放用CDRW刻录的音乐。CDRW比较适合于存放数据而不适合存放音乐(除非我只想在计算机上播放这些盘片)。

创建混合模式的CD

创建混合模式的CD(通常指的是同时包括数据 and 音乐的CD, 比较常见的是游戏CD)不是什么很难的事情, 举例如下:

```
mount -t iso9660 /dev/cdrom /mnt/cdrom
```

(挂载混合模式CD的数据部分)

```
mkisofs -r -o cd_image /mnt/cdrom
```

(把CD上数据做成是ISO系统文件)

```
umount /mnt/cdrom
```

(卸载CD)

```
cdparanoia -B "2-"
```

(从音乐CD上“刮写”除了第一道音轨之外的所有音轨, 因为第一个音轨要用来存放数据)

```
cdrecord -v speed=2 dev=0,0,0 -data cd_image -audio track*
```

(把数据和音乐一个接一个的写到光盘上)

CD复制

绝大部分的CD都可以使用以上方法把数据或者音乐复制到硬盘上。但是有些CD却不行。

举个例子, 以下这些CD必须用特殊的办法来处理: 可自启动的CD(比如Linux的安装光盘), 要求标签的CD, 有数据错误的CD盘片等。对于数据CD, 我一般使用以下命令做一个备份:

```
dd if=/dev/cdrom of=cd_image  
cdrecord -v speed=2 dev=1,0,0 -data cd_image
```

第一条命令: dd复制所有输入文件(在这个例子里是/dev/cdrom)到硬盘上的一个输出文件上(在这个例子里, 文件存放在当前的目录下, 文件名是cd_image)。

第二条命令: cdrecord复制文件上面产生的文件“cd_image”到一张空白的CD上。

对于有数据错误的CD, 你可以尝试使用:

```
dd conv=noerror,notrunc if=/dev/cdrom of=cd_image
cdrecord -v speed=2 dev=1,0,0 -data cd_image
```

可选项 “ conv=noerror,notrunc ” 指定潜在的读盘错误将被忽略，出错的文件不要截断。对于音乐CD，我使用以下命令：

```
cdparanoia -B "1-"
```

(把音乐CD所有音轨刮写到当前目录下，对应的音频文件的名称是track01.cdda.wav, track02.cdda.wav...)

```
cdrecord -v speed=2 dev=1,0,0 -audio track*
```

(把所有的音频文件刻录到CD上，每个音轨之间的时间间隔是2秒)

如果要以最准确的方法复制CD，cdrecord手册的建议如下：

```
cdda2wav -v255 -D2,0 -B -Owav
cdrecord -v dev=2,0 -dao -useinfo *.wav
```

如果要复制一张混合模式的CD，使用：

```
dd if=/dev/cdrom of=cd_image
```

(当数据部分结束，音轨部分开始的时候，dd命令会有一个错误信息，这个错误是意料之中，不会造成什么问题)

```
cdparanoia -B "2-"
```

(“刮写”音乐CD上除了第一道音轨以外的所有音轨，因为第一道上面存放的数据内容)

```
cdrecord -v speed=2 dev=1,0,0 -data cd_image -audio track*
```

(再把所有的数据和音频文件一个接一个的写道CD上)

可多次复写的CD(CDRW, Re-writable CDs)

CDRW的处理和普通的CDR(write-once CD, 只能写一次)没有什么很大的区别。但是在你使用之前，必须把它变成“清洗”成空白的盘片：

```
cdrecord -v speed=2 dev=1,0,0 blank=fast
```

关于如何“清洗”CDRW盘片，使用命令：

```
cdrecord blank=help
```

举个例子，以下“清洗盘片”的命令在我的计算机上大概要运行半小时，但是如果没有什么敏感数据在上面，也不是非要这样做不可。

```
cdrecord dev=0,0,0 blank=disk
```

还有，要记得一些老的音响设备不能播放CD-R刻录的音乐。

简化一些冗长的命令

要简化在使用cdrecord (或者cdrdao) 冗长的命令，我可以把以下一行内容加入道文件 /etc/bashrc里：

```
alias cdrecord="cdrecord -v speed=2 dev=1,0,0"
```

重新登录后使得/etc/bashrc开始有效。然后，我就可以使用以下短得多的命令 (不需要老是指定CD刻录机的速度和设备的名字)：

```
cdrecord -audio track*
```

6.7 使用 gnuplot 自动画图

gnuplot 是一个可以根据数据或者数学公式自动产生图表的非常号的工具。对于交互的图形生成，我比较喜欢电子表格。如果你根据“模板”从大量的数据里生成可视化图形，在这个过程中需要不断进行调整，对于不那么流行的 gnuplot 可能会有点不那么顺手。gnuplot 有很多灵活的特性，但是需要时间去学习。设计一个复杂的图形模板有些时候要花掉我两小时的时间 (但是可以接受，因为模板可以在以后重复使用)。最好的办法是启动 gnuplot，然后在“gnuplot>”的提示符下，输入“help”。gnuplot 在 Linux 和 Windows 下都有效。

我的数据存放在文本文件 (ASCII, *.dat) 里。我的“图形模板”存放在gnuplot“命令”文件里 (*.gnu)。输出将会输出道图形文件里 (*.png)，该图形文件可以被打印或者输出道其他的字处理软件里。

以下是一个从命令文件“make_graphs.gnu”产生图形输出的例子，运行命令：

```
gnuplot make_graphs.gnu
```

显示输出结果，我可以输入 (在 X 终端)：

```
display my_graph.png
```

用于生成二维图形的文件 "make_graphs.gnu"例子 如下页所示：

```
# 以 # 开头的都是注释行
# 给图形打上日期和时间戳

set timestamp "%Y-%m-%dT%T%z"

# 以下设置图形的解析度 (缺省值100)

set samples 600

# 把输出存为一个 *.png的文件, 并设置为彩色
set output "my_plot.png"
set terminal png color

#interesting terminals: png, x11, postscript, postscript eps, hpgl

set title "My Graph"           # 设置图形的标题
set xlabel "Distance [m]"      # 设置x轴的第一个标题(底部)
set x2label "Distance [feet]"  # 设置x轴的第二个标题(顶部)
set ylabel "sin meters"        # 设置y轴的第一个标题(左侧)
set y2label "log feet"         # 设置y轴的第二个标题(右侧)
set xtics                      # 控制坐标轴上的刻度
set x2tics; set ytics; set y2tics # 同一行上的命令可以使用 ";" 来分隔
set mytics                     # 设置坐标轴上的最小值, 这里我把它加到y轴上
set xrange [0:15]              # 在x轴上的数值范围
set x2range [0:15.0/0.305]     # 把“米”制量度转化成“英尺”
set yrange [*:*]              # "*" 设置数值范围为自动
set y2range [*:*]             # 设置y轴的第二个数值范围(右侧)
set nologscale                 # 控制画图的类型, 缺省是线性
set nogrid                    # 或者 "set grid", 设置不画格子, 缺省是nogrid
set key outside                # 或者 "set nokey", 设置标注的位置

# 以下的行使用4个不同的系列生成图形:
plot sin(x) axes x1y1, log(x) axes x2y2, "data.dat" using 1:2, \
"data.dat" using 1:3
# 比较长的行可以使用 "\ " 来分隔
# 第三个系列使用数据文件里的第一和第二列数据
# 第四个图形从数据文件里采用第三列的数据和第一列的数据做对照
```

第 7 部分: Linux 一些高级应用

Linux 新手管理员指南(中文版)

LINUX NEWBIE ADMINISTRATOR GUIDE (Simplified Chinese Version)

中文版本 1.0 由幻影快递 Linux 小组(SLT-Shadowfax Linux Team) 翻译整理

Linux Newbie Administrator 英文原版版权所有: Peter and Stan Klimas.

<http://sunsite.dk/Linux-newbie>

中文版本版权所有: 幻影快递 Linux 小组 ShadowfaxLinux@hotmail.com

本章内容:

7.1 LINUX 高级文本处理工具:

banner, script, emacs, vi, nano, diff, patch, grep, regular_expressions, sed, (g)awk, cvs, sort 和 ascii, gpg(pgp), ...

7.2 LINUX 下的简单编程:

perl, python, tcl/tk/wish, c/c++, kdevelop, glade, ruby, g77, ...

7.3 数学工具:

dc, bc 和“普通”计算器, scilab, 命令行上的随机字符产生

7.4 其它:

wine, RAID disk tools, network traffic shaping

简介:

本章包含的命令对于新手来说都不是最重要的, 但是如果你花时间去学它们的话, 你会获得许多乐趣和创意。除此之外, 它们也很富有教育意义, 因为 LINUX 是一个很好的学习平台。LINUX 上有很多的免费程序, 在你的系统上有一些可能会找不到, 这取决于你所安装系统的版本。你总可以用 google 查到他们的网站然后把他们下载下来。

7.1 LINUX 高级文本处理工具

```
/usr/games/banner -w79 "Happy Birthday, Marie" > marie.txt
```

生成一个宽度为 79 个字符的 ASCII 码“banner”。结果输出到一个名为“marie.txt”的文件中。这是一个有趣的老式的工具。另一个用于 ASCII 码文本显示效果的工具是 figlet。例如: figlet “Funny!” 在我的终端上将显示如下: (我经常使用固定大小的字体来显示它)


```
cat * | grep my_word | more
```

将在当前目录下所有文件（文件名以园点开头的除外）里搜索，将文件里含有“my_word”字符串的行一行行地显示在屏幕上。

下面是一个完成同样功能，但更加简练的命令：

```
grep my_word * | more
```

模版是由“正则表达式”表式，正则表达式是一种强有力和标准的表示方法。

还有一种是“递归”的grep，叫做rgrep。它会搜索所有当前目录和当前目录下的所有子目录里的文件，检查是否有文本和模版my_word匹配，并打印出和模版匹配的文本所在相应行。

```
rgrep -r my_word . | more
```

正则表达式 (regexpr)

正则表达式用于在搜索和替换等操作中进行模式匹配。它们总是和一些实用工具（例如：grep, sed等）以及程序语言（例如：perl）一起使用。Shell命令dir，就使用了一点正则表达式（下面会提到具体的主要区别）。以下简短的归纳了几乎所有标准正则表达式的特点：

在正则表达式里，大部份的字符都是只和自己本身匹配。因此，如果要搜索“peter”这个字符串，我就会只用一个搜索字符串“peter”。除此之外还有一部份字符被称为“特殊字符”（“原字符”）。它们都有特别的含义。

正则表达式的特殊字符有：“\”（反斜杠），“.”（园点），“*”（星号），“[”（方括号），“^”（尖嘴箭头，只是用于字符串的开始），“\$”（美元符，只是用于字符串的结尾）。一个这样的字符通常就注定了模版字符串是一个特殊的字符串。

反斜杠，“\”是一个起“避开”作用的字符。例如，需要匹配一个特殊的字符，我们就把那个特殊字符放在反斜杠之后，以“避开”这个特殊字符的原有功能。因此，“\\”将作为搜索所有含有“\”的字符串的正则表达式，“\.”将搜索所有含有园点的，“*”将搜索所有含有星号的，“\[”将搜索所有含有方括号的，“\^”将搜索所有含有尖嘴箭头的，即使它在一行字符串的开头。“\\$”将搜索所有含有美元符号的，即使它在一行字符串的结尾。

反斜杠后跟一个正规字符也可能表示一个特殊含意。因此，符号\<和\>分别匹配一个文本字符串的开头前的空字符串和结尾后的空字符串。符号\b匹配一个文本字符串的两边的空串，\B就匹配一个文本字符串里的空串。

园点，“.”，匹配任何单个的字符[dir命令里，在这种情况下，用“?”来代替“.”]因此，“m.a”匹配“mpa”和“mea”，但不匹配“ma”或“mppa”。

任何字符串都可以被“.*”匹配（园点和星号）。[dir命令用“*”；来代替]。总的来说，任何一个后面跟有“*”的模版会和以这个模版开头后面跟有零个或多个字符的字符串匹配。因此，“m*”会和零个或多个“m”匹配。如果要匹配一个或多个m，我就会用“mm*”。

字符 “ * “ 是一个对字符进行重复操作的操作符。其它的进行重复操作的操作符不是象它那样经常使用。下面是所有可以进行重复操作的操作符：

- * 被操作的项会匹配 0 或多次。
- \+ 被操作的项会匹配 1 或多次。
- \? 被操作的项会匹配最多一次。
- \{n\} 被操作的项会匹配正好 n 次。
- \{n,\} 被操作的项会匹配 n 次或 n 次以上。
- \{n,m\} 被操作的项会匹配最少 n 次，最多 m 次。

尖箭嘴，” ^ “，表示一行的开始字符，因此，” ^a “就是匹配以字符 a 开头的行。

美元符，” \$ “，表示一行结尾的字符，因此，” a\$ “就是匹配以字符 a 结尾的行。

下面的命令在一个文件名为 “ myfile ” 的文件里搜索以 “ s ” 开头并以 “ n ” 结尾的行，然后把结果打印到标准输出（屏幕）上。

```
cat myfile | grep ' ^s.*n$ '
```

对于那些能使字符串变为匹配正则表达式的特殊字符，如果你想在字符串里表示，就需要在它的前面加上一个反斜杠就行了。

方括号，” [“ 表示一组字符。因此， [abD] 的意思就是 a 或 b 或 D。 [a-zA-C] 的意思就是 a 到 z 和 A 到 C 中的任何一个字符。

请注意一些在方括号表示的字符集里的特殊字符。在这个字符集里，这些特殊字符是 ” [“， ”] “， ” - “，和 ” ^ “，还有一些组合也是特殊的，象 ” [: “， ” [= “ 和 ” [. “。反斜杠不是特殊字符。

下面是一些有用的字符集类别（根据 POSIX 标准的定义）：

- [:upper:] = 大写字母
- [:lower:] = 小写字母
- [:alpha:] = 字母，包大小写
- [:digit:] = 0 到 9
- [:alnum:] = 字母加数字
- [:space:] = 空白，象 <SPACE> , <Tab> 和 <NewLine> 都是
- [:graph:] = 除了空格以外的可打印的图形字符
- [:print:] = 可打印的字符，包扩空格
- [:punct:] = 标点符号字符，就是图形字符减去字母和数字
- [:cntrl:] = 控制字符，不能打印的字符
- [:xdigit:] = 16 进制数字

下面的命令搜索 dir 命令的输出，并把那些含有大写字母后紧跟一个数字的字符串的行打印出来：

```
dir -l | grep ' [[:upper:]][[:digit:]] '
```

tr(=translation---转化)。 一个用于在文本文件中进行所有同类字符串或者用于 “ 挤压 ” 空格的过滤器。

例子：

```
cat my_file | tr 1 2 > new_file
```

这个命令读取文件 `my_file` 的内容，并把它们传到用于转化的工具 `tr`，`tr` 这个工具会把文件里所有字符“1”替换成字符“2”，结果会引导到文件名为 `new_file` 的文件中。

`sed`

(=stream editor ---- 流编辑器) 我爱用 `sed` 来过滤文本文件。通常用来匹配的模版都是放在一对斜杠//和引号之间的。

比如，如果要打印出含有字符串“1024”的行，我就可能会用：

```
cat filename | sed -n '/1024/p'
```

这里，`sed` 过滤掉 `cat` 命令的输出。选项“-n”告诉 `sed` 把那些不符和匹配条件的行都挡在输出结果的外面，对于那些符和条件的行，`sed` 对它们的作用是“p”=打印。

另外一个例子，这次是删除那些被选出来的行：

```
cat filename | sed '/.*o$/d' > newfile
```

在上面的例子里，所有以一个字符‘o’结尾的行将会被删除。我用了一个正则表达式来表示那些以一个字符‘o’结尾的行。输出（比如，所有以字符‘d’结尾的行）就会被传到文件名为 `newfile` 的文件中。

又是一个例子，这次是搜索和替换，我用了 `sed` 的‘s’这样一个作用，通常它是放在两个表达式的前面：

```
cat filename | sed 's/string_old/string_new/' > newfile
```

更简练的写法是：

```
sed 's/string_old/string_new/' filename > newfile
```

在一个 `html` 文件里插入一段从一个文本文件里得来的文本，我可能会用下面的 `script`：

```
sed '/text_which_is_a_placeholder_in_my_html_file/r text_file_to_insert.txt'
index_master_file.html > index.html1
```

`gawk`

(=GNU `awk`. `Awk` 这个命令是一个传统的 UNIX 工具。) 一个用于对文本文件进行处理的工具，在很多方面和 `sed` 很相似，但是功能更强大。`Perl` 可以做所有 `gawk` 可以做的事，甚至更多，因此，我基本上不太用 `gawk`。简单的任务我会用 `sed`，对更复杂的我就会用 `perl`。在有些情况下，`awk script` 比相同功能的 `perl` 程序短小很多，容易懂和维护，并且更快。

gawk 特别适用于处理那些文本格式的表格。一个表由记录组成（通常一行就是一个记录）。每条记录含有由界定符分隔开的域。经常用到的界定符有 空格（gawk 预定），逗号和冒号。所有的 gawk 命令的格式就是：

```
gawk 'pattern {action}' my_file.
```

在命令行里，你可以不要 pattern(模版)或者 action(作用)：预定的模版是“匹配任何东西”并且预定的作用是“打印输出行”。gawk 还可用来做过滤器（用来处理另一个命令的输出，就象我们下面的例子一样）。

把含有字符串“1024”的行打印出来，我会用：

```
cat filename |gawk '/1024/{print}'
```

就象在 sed 里一样，用来匹配的模版是放在一对“//”里的。

gawk 比 sed 强的地方就在于它能对域进行操作。\$1 是“第一个域”，\$2 是“第二个域”，等等。\$0 是“一整行”。下一个例子把含有字符串“1024”的行里的第 3 和第 2 个域的值读出来，并在每个域的打印出来的值的第一行加上“姓名”和“登记号”这样两个域的标签。打印的最终结果放入一个名为“newfile”的文件中：

```
cat filename | gawk '/1024/{print "Name: " $3 " ID : " $2}' > newfile
```

第三个例子里，gawk 将把那些第三个域的值等于“peter”或者含有字符串“marie”的行找到并打印出来。

```
Cat filename |gawk '$3 == "peter" || $3 - /marie/ '
```

为了进一步了解最后一个命令，我们把 gawk 里的逻辑运算符列出如下：

==	等于
!=	不等于
<	小于
>	大于
<=	小于等于
>=	大于等于
-	和正则表达式匹配
!-	不和正则表达式匹配
	逻辑或
&&	逻辑与
!	逻辑非

CVS

并发版本控制系统。试着用：info cvs 拿到更多的信息。在好几个程序员同时在一个相同的程序上工作时，cvs 对“源码库”的版本维护是很有用的。

cervisia

(在 X 终端里)。一个 GUI 前端界面的 cvs 版本管理系统。

file -z filename

确定文件名为 filename 的文件的类型。选项-z 还要告诉你文件是哪一种压缩文件，如果文件是一个压缩文件的话（不仅仅只是告诉你它是个压缩文件而已）。

为了确定文件的内容的类型，file 命令会在文件的内部去寻找一个特别的文件类型标志（“魔法数”）--- 而不是象在 MS windows 一样靠文件名的后缀来确认文件的类型。“魔法数”存在一个关于文件类型的数据库的文本文件/usr/share/magic 之中。

touch filename

把文件名为 filename 的文件的日期/时间戳改成当前的时间。如果文件不存在的话，就会产生一个新的文件。你可以把时间戳改成任意你想要的时间，

touch -t 200201311759.30(2002 年 1 月 31 日时间 17 : 59 : 30)

在 ext2 文件系统中每一个文件跟有三个时间值：

---最后一次访问文件的时间 (atime)
---最后一次修改文件的时间 (mtime)
---最后一次文件的 i 节点被改变的时间 (ctime)

touch 将只对文件的前两个时间特性进行修改，最后一个通常总是当前时间。它们都可以用下面提到的 stat 命令读出来（开下一个说明）

stat filename

打印出一个文件的一般信息（所谓 i 节点的内容）。

strings filename | more

显示出一个名为 filename 的二进制文件里含有的字符串。比如说，strings 作为检查一个未知的二进制文件的第一步是很有用的。

od

(=octal dump --- 八进制输出)。把文件内容以八进制数显示。当一个文件的输出中含有不能打印的字符是很有用的。比如：如果一个文件名含有不能显示的字符是很痛苦的。另外，它对看二进制文件也很有帮助。

例子：

dir | od -c |more

(我可能宁愿用：ls -b 来看含有不能显示的字符的文件名)。

```
cat my_file | od -c |more
od my_file | more
```

不同输出的比较：

显示一个二进制文件 (/bin/sh) 的首 16 个字符，用 ASCII 码或反斜杠 escapes (八进制)：

```
od -N 16 -c /bin/sh
```

输出：

```
0000000 177 E L F 001 001 001 \0 \0 \0 \0 \0 \0 \0 \0
```

同样的二进制用 ASCII 码显示：

```
od -N 16 -a /bin/sh
```

输出：

```
0000000 del E L F soh soh soh nul nul nul nul nul nul nul nul
```

同样的二进制用简练的十六进制显示：

```
od -N 16 -t x1 /bin/sh
```

输出：

```
0000000 7f 45 4c 46 01 01 01 00 00 00 00 00 00 00 00
```

同样的二进制用八进制输出：

```
od -N 16 /bin/sh
```

输出：

```
0000000 042577 043114 000401 000001 000000 000000 000000 000000
```

wc

(=word count, 文字计数器)打印出文件的行数，字数和字节数。

例子：

```
dir | wc
cat my_file | wc
wc myfile
```

cksum filename

计算文件名为 filename 的 CRC (= “循环冗余校验”) 来保证文件的完整性。

`md5sum filename`

计算文件名为 `filename` 的文件的 md5 校验和 (128 位) 来报证文件的完整性。

`mkpasswd -l 10`

产生一个长度为十个字符, 并很难猜测的随机密码。

`sort -f filename`

把文件里的行按 `ascii` 码的顺序排序。选项 `-f` 用来告诉 `sort` 不用考虑字符的大小写。下面是 `ascii` 字符集 (看 `man ascii`) :

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	00	NUL '\0'	32	20	SPACE	64	40	@	96	60	`
1	01	SOH	33	21	!	65	41	A	97	61	a
2	02	STX	34	22	"	66	42	B	98	62	b
3	03	ETX	35	23	#	67	43	C	99	63	c
4	04	EOT	36	24	\$	68	44	D	100	64	d
5	05	ENQ	37	25	%	69	45	E	101	65	e
6	06	ACK	38	26	&	70	46	F	102	66	f
7	07	BEL '\a'	39	27	'	71	47	G	103	67	g
8	08	BS '\b'	40	28	(72	48	H	104	68	h
9	09	HT '\t'	41	29)	73	49	I	105	69	i
10	0A	LF '\n'	42	2A	*	74	4A	J	106	6A	j
11	0B	VT '\v'	43	2B	+	75	4B	K	107	6B	k
12	0C	FF '\f'	44	2C	,	76	4C	L	108	6C	l
13	0D	CR '\r'	45	2D	-	77	4D	M	109	6D	m
14	0E	SO	46	2E	.	78	4E	N	110	6E	n
15	0F	SI	47	2F	/	79	4F	O	111	6F	o
16	10	DLE	48	30	0	80	50	P	112	70	p
17	11	DC1	49	31	1	81	51	Q	113	71	q
18	12	DC2	50	32	2	82	52	R	114	72	r
19	13	DC3	51	33	3	83	53	S	115	73	s
20	14	DC4	52	34	4	84	54	T	116	74	t
21	15	NAK	53	35	5	85	55	U	117	75	u
22	16	SYN	54	36	6	86	56	V	118	76	v
23	17	ETB	55	37	7	87	57	W	119	77	w
24	18	CAN	56	38	8	88	58	X	120	78	x
25	19	EM	57	39	9	89	59	Y	121	79	y
26	1A	SUB	58	3A	:	90	5A	Z	122	7A	z
27	1B	ESC	59	3B	;	91	5B	[123	7B	{
28	1C	FS	60	3C	<	92	5C	\ '\\'	124	7C	
29	1D	GS	61	3D	=	93	5D]	125	7D	}
30	1E	RS	62	3E	>	94	5E	^	126	7E	~
31	1F	US	63	3F	?	95	5F	_	127	7F	DEL

如果你对控制符有疑虑, 这里是一在控制台上的控制符的含意 (来源: `man console_codes`)。下面每行给出了控制符的助记符, ASCII 码的十进制表示, 在控制台上显示的组和键和一个简短的描述:

BEL (7, <Ctrl>G) 响铃 (=警报, beep 声).
BS (8, <Ctrl>H) 返回一列 (但不会超过该行的开始).
HT (9, <Ctrl>I) 水平 tab 移动, 去到下一个 tab 开始的地方, 如果在行内每有 tab, 就去到行尾。
LF (10, <Ctrl>J), VT (11, <Ctrl>K) and FF (12, <Ctrl>L) 所有这三个都给出一个换行。
CR (13, <Ctrl>M) 给出一个回车。
SO (14, <Ctrl>N) 在有 LF/NL(新行模式)和回车设定下, 击活 G1 字符集。
SI (15, <Ctrl>O) 击活 G0 字符集。
CAN (24, <Ctrl>X) and SUB (26, <Ctrl>Z) 中断转义序列 (escape sequences)。
ESC (27, <Ctrl>[) 开始转义序列。
DEL (127) 是乎略或删除。
CSI (155) 控制序列导入。

uniq

(=unique-----唯一的)去掉已排序的文件中的重复行。例如: `sort myfile | uniq`

`fold -w 30 -s my_file.txt > new_file.txt`

用空格将文本文件 `my_file.txt` 中的每行截成 30 个字符, 并输出到 `new_file.txt`。

`fmt -w 75 my_file.txt > new_file.txt`

格式化文件中的每行为 75 个字符。把超过 75 个字符的行截短, 把少于 75 个字符的行补齐, 但不会移去空行。

`nl myfile > myfile_lines_numbered`

给文件 `myfile` 的每行标上行号。把结果输出给文件 `myfile_lines_numbered`。

`indent -kr -i8 -ts8 -sob -l80 -ss -bs -psl "$@" *.c`

通过插入或删除空格来改变“C”源码的外观。上面例子里格式的选项是和 Linux 内核的源码的风格是一至的 (`script /usr/src/linux/scripts/Lindent`)。看 `man indent` 来了解各个选项的意义。存在的文件先被备份然后就被格式话好的文件给替代了。

`rev filename > filename1`

打印文件 `filename`, 每行字符以相反顺序显示。上面的例子里结果直接输入到文件 `filename1`。

`shred filename`

用垃圾字符重复并覆盖文件 `filename` 的内容, 使任何人都无法再去阅读文件的内容。

`paste file1 file2 > file3`

按行把两个或多个文件合并, 用<Tab>作默认分界符 (用选项 "d=" 来选择自己的分界符)。

如果文件 `file1` 的内容是:

1
2
3

并且文件 `file2` 的是:

a

```
b
c
d
```

那么结果 file3 里的内容就应该是：

```
1      a
2      b
3      c
        d
```

```
join file1 file2 > file3
```

把两个文本文件里有相同域的行连成一行。Join 和数据库的“表连接”操作是类似的，但它只是对两个文本文件进行操作。默认的分界符是空格。如果你想用其它的操作符，你可用 `man join` 去查看里面的选项。

如果 file1 里的内容是：

```
1      Barbara
2      Peter
3      Stan
4      Marie
```

并且 file2 的是：

```
2      Dog
4      Car
7      Cat
```

那么结果 file3 就应该是：

```
2      Peter Dog
4      Marie  Car
```

```
des -e plain_file encrypted_file
```

(= "Data Encryption Standard" 数据加密标准)加密文件 `plain_file`。你会被要求输入一个键值用于加密算法使用。加密后的结果存放到文件 `encrypted_file`”。解密就用：

```
des -d encrypted_file decrypted_file.
```

```
gpg
```

“Gnu Privacy Guard-----GNU 的私人警卫”——一个和 PGP (“Pretty Good Privacy”) 功能一样的免费软件。gpg 比 PGP 更安全并且没有使用有专利的算法。gpg 主要用于给你的 email 加签名并且检查其它信息的签名。你也可用它来给你的信息加密解密。<http://www.gnupg.org/>上有所有的详细信息，包括一个易读和详细的手册。

开始，我需要一对匙值：个人匙和公共匙。个人匙用于给我的信息签字。公共匙用于传给他人来确认我的信息的签名在他们收到我的信息后来确认我的信息的签字。（某人还可用公共匙来给信息加密这样这个信息就只能通过我的个人匙解密并阅读）。我用下面的命令来生成这一对匙值：

```
gpg -gen-key
```

我的匙会存放在目录~/ .gnupg 下 (它会被我用一个密码值加密在我生成匙值的过程中) 。在一个纯文本文件中显示我的公共匙的值 , 我用 :

```
gpg -armor -export my_email_address > public_key_stan.gpg
```

它会生成一个含有如下类似内容的文件 public_key_stan.gpg

```
.....BEGIN PGP PUBLIC KEY BLOCK.....  
Version: GnuPG v1.0.1 (GNU/Linux)  
Comment: For info see http://www.gnupg.org  
mQGIBDmznEYRBACoN438rxANaMfCy5bfj6KWM0/TR6x6HZ0gpmhGeuouM/SOR2IU  
/G30NdCuzHeFs93BhtY0ldzoEMtMyZHnvdhZC2bx/jhgaaMbEaSsXwRhVB0xVPYx  
rHbsgSULHYzRFF34MS3/Lse3QWfWxzA7l0lBxB7nLwZKZqaNONRFR42owCg60hV  
TDPEB2N0lIMyt12R4ZByFSsEAJ1tE7pb9b6TP7cw21vkljC+Bl2uSzn/B4vNICWK  
TTuZHVv0w0jFcbd8DB0/1tIZU0rIzLSqJyQDGiNn258+7LetQ+LKG/1YKbiAcosz  
4QirBuLIEF2M9GuXYCwZypE3Dwv+4YupvybR31CgLTJ8p4sKqC5n0eSr2oSrtHYZ  
yuJtA/9v2HcebOncfCNOK+cVRmcTB1FrI/Gh/vNCfeZyXaJxlqDfCU2vJHtBemiE  
AtcfZHB/iHy0DM68lFRJSAIFaa5um9iWHh5/vWCGZLqtpwZ7kyMw+2D6CFKWATsy  
wQA1g1VcGkNc14Crrd36qf60bl+b8pn2zDhwZtLsELsXyXkNhbQmU3RhbIBKIEts  
aW1hcyA8U3RhbktSaW1hc0B3ZWJoYXJ0Lm5ldD6lVgQTEQIAFgUC0afMRgQLCgQD  
AxUDAgMWAqECF4AAcGkQt+ZBooH8bHd2kwCghAt9aKIk0mRjv+g7YcRPotVtrwkA  
n1a4xEVEyaKgK0MaJnopf69K9+vouQENBDmznH4QBADgFpLP+tWZPnVYg47cn+9b  
XQRjd0tNsDE6BYH872/sR1oCrdH6k+gXFOiZxRZ3PEIK2/olo59kh5xa9aBxNdEC  
FuXJN0UeImh0FbDttVkslqVWYyfXnLz+wtcXg0Q0L0q8vY4lUtZw2WkV6EkM+/x8  
6UhA2XVaMJKbDRKFSVilbwADBQP+JCzLj5HDgpRvf+KM72nzSg7sp8Tki7nF9wNA  
PODK0SeQgl3dwXYyF6AVenIETE/3xRwoYQN1bxVZs0ex9vzqPrQC3dRONBljd74r  
kfXwUTl2fNqX4N9iuVCo2gCGbi5+gfeK1GhsWdsq0z40f+18k+XBdWmY8sCNioIT  
tnvm1QeIRgQYEQIABgUC0afMfgAKCRC35KigfXsd9SGAJ9/FWSkEfgbE/Yc46d8  
Ef1gYg3l1ACff3oLeAMeGG079gW6UGp9RJ6mRao=  
=X1k2  
.....END PGP PUBLIC KEY BLOCK.....
```

现在我可以把我的公共匙 email 给那些我希望能安全的交流的人。

他们可以把他的存放在他的 pgp 系统上 :

```
gpg -import public_key_stan.gpg
```

甚至可能更好 , 我可以把我的公共匙提交给一个公共匙服务器。我用下面的命令来寻找我就近的服务器 :

```
gpg -keyserver wwwkeys.pgp.net -send-keys  
linux\_naq@canada.com
```

“ wwwkeys.pgp.net ” 是我选的匙服务器 , 并且 linux_naq@canada.com 是我的 email 地址 , 用于在本地的匙圈对我的识别。我只需要提交给一个公共服务器即可 (其它会与这个服务器自动同步) 。

现在 , 我可以开始使用 gpg。如果要手工地给一个文件 my_message 签字 , 我会用 :

```
gpg -clearsign my_message
```

这个会产生一个文件 `my_message.asc` , 并含有如下内容 :

```
.....BEGIN PGP SIGNED MESSAGE.....  
Hash: SHA1  
  
Hello World!  
  
.....BEGIN PGP SIGNATURE.....  
Version: GnuPG v1.0.1 (GNU/Linux)  
Comment: For info see http://www.gnupg.org  
  
iD8DBQE5p9+3t+ZBooH8bHcRApn/AJ9kx9+pU3GJBuvJN9Bo3bW3ku/5PwCgquht  
mfrPrt7PQtdmGox72jkY0lo=  
=rtK0  
.....END PGP SIGNATURE.....
```

如果要确认一个签过名的文件, 我会用 :

```
gpg -verify my_message.asc
```

如果你的文件即使有很小的改变, 签名检查不会被通过。

手工签名是笨拙的, 但是, 举例来说, `kmail` 就能自动为我做电子签名。

“ docbook ” 工具集

Docbook 是近来文件保管的标准。Docbooks 工具集包含在 RH6.2 的 “ jade ” 软件包里, 它还包有如下转换工具: `db2ps`, `db2pdf`, `db2dvi`, `db2html`, `db2rtf`, 这些工具能把 docbook 文件转成: `postscript(*.ps)`, Adobe PDF (`*.pdf`), `DVI(*.dvi)`, `HTML(*.html)`和 `RTF(*.rtf)`格式。

“ 文件保管 ” 就是文件的格式能自动转成其他的有用格式。比如说, 有一个正在完成的文件, 在将来, 它可能考虑被作成一个报告的形式, 一个年报的形式, 一个报纸里的文章的形式, 一个网页, 还也可能是一本书的格式, 我 (作者) 现在还不能确定它。用 “ 固定的代码 ” (字体, 字体大小, 分页符, 行中置符, 等等。) 来设置文件的各式其实是浪费时间的---特别文件类型的各种各样的风格是和出版者相关的。解决的方法是用那些 “ 逻辑 ” 上的排版元素象文件标题, 章标题, 小章标题, 强调风格, 图片名, 字幕和表, 等等。这就是 “ docbook ” 做的事情—它是一种风格的描述 (用 `xml`, `html` 的一个优集, 并且与 `sgml` 有关) --因而也被称作风格单。在文件出版的时候, 文件的格式就会依照逻辑版面的定义实际显示出来。

当我们学习如何使用 docbook 的时候, 我们会对这一段里的内容进一步详细说明。

7.2 LINUX 下的简单编程

perl

强大和广泛使用的脚本语言，在大师们当中很流行。Perl 看上去很神秘，但如果你用它来完成一些简单的任务，它其实是相当直接的。对于简单的编程而言，Perl 就象一把瑞士军刀。Perl 的语法和 C 语言很象。在 MS-Windows 下 Perl 也有一个一流的解释器，所以你写的代码是可以跨平台的。Eric Reymond (著名的 Linux 大师) 是这样描述 perl 的：“Perl，显然是现代脚本语言里面的一个 800 磅的大猩猩。它大部分被系统管理员用于替代 shell 脚本语言，一部分这要归功于它对 UNIX 库函数和系统调用的包容，另外这也要归功于非常活跃的 Perl 社区开发的大量的 Perl 模块。在网上 85% 的“活”的内容的后面估计都用了它来作为 CGI 语言。Larry Wall，它的创建者，被任为是开放代码源社区的重要领导者之一，并且总是被任为是黑客崇拜者的罗马神殿里仅次于 Linus Torvalds 和 Richard Stallman 排名第三的人物。”

我怎样写一个简单的 perl 脚本？

我可能会用 pico (或其它任何我想选用的文本编辑器) 来输入一个简单的 perl 脚本：

```
pico try_perl
```

下面的脚本例子没有任何用处，只是用来说明 perl 的一些基本特征：

```
#!/usr/bin/perl -w
# a stupid example perl program
# the lines starting with # are comments except for the first line
# names of scalar variables start with $
$a=2;
$b=3;
# each instruction ends with a semicolon, like in "c"
print $a**$b,"\n";
$helloworld='Hello World';
print $helloworld,"\n";
system "ls";
```

第一行告诉 shell 如何执行我的文本文件。选项“-w”会让 perl 输出更多的警告信息等，这可能会对你调试你的脚本程序有用。下面三行(以#开头)是注释。再下面的几行大部分是用于自我解释的：我给两个变量(\$a和\$b)赋值，把\$a再\$b次方并把结果打印出来。字符“\n”打印一个新行，就和C语言一样。然后我再定义一个新的变量，把字符串“Hello World”赋给它，再然后，我把这个变量打印出来。最后，我运行一个本地操作系统的命令“ls”，它在Linux里用于显示当前目录的内容。这真是一个傻傻的脚本程序。

在存好这个文件后，我会把它变成可执行的：

```
chmod a+x try_perl
```

我现在输入如下命令来运行这个脚本程序：

```
./try_perl
```

下面是一个稍长一些作起来很简单，但又对我很有用的一个脚本程序。我有一个由某个数据获取系统生成的大型文本文件。我需要每隔一行就删除一行，这样我就能将这个文本文件填满我的 MS Excel 文件：

```
#!/usr/bin/perl -w
# Create a text file containing a selection of lines from an original file.
This is needed
# so that data for manual postprocessing are fewer.
#
# Prompt the user for the filename, and the selection of lines to preserve in
the output.
print STDOUT "Enter the filename: ";
chomp($infile=<STDIN>);
open(INFILE,"<$infile"); # open the file for reading.
print STDOUT "Enter the number of initial lines to preserve: ";
chomp($iskip=<STDIN>); # the first lines may contain column headings etc
print STDOUT "Enter the skip: ";
chomp($skip=<STDIN>);
#
# The name of the output file is created automatically on the basis of the
# input file and the selection of lines. It is always of type CSV, so preserve
is so.
$outfile=$infile.'.pro'.'. $iskip.'.'. $skip.'.csv'; #glue strings together using
the dot operator
open(OUTFILE,">$outfile"); # open file for writing.
#
# write the "initial" lines to the output file.
for($a=0;$a<$iskip;$a++) {
$line=<INFILE>;
print OUTFILE $line;
}
#
# do the rest of the file
$c=0;$w=0;$skip++;
while($line=<INFILE>){
$c++;
if(!($c%$skip)) { #use % for remainder of integer division
print OUTFILE $line;
$w++;
}
}
#
close(OUTFILE);
print STDOUT "Read Lines: ", $c+$iskip, " Wrote lines: ", $w+$iskip,"\n";
```

python

现代的和非常优雅的面向对象的解释器。它是强大的并且比 perl 有更好的可读性。在网上可以找到由创造者 G.van Rossum 写的很好和很大的手册。(可以试着去 <http://www.python.org/doc/> 浏览或去 <ftp://ftp.python.org> 去下载)。

我如何写一个简单的 Python 程序？

编辑一个含有你的 Python 程序的文本文件。我会用 kde 的 “kate” 编辑器来作 (在 X-Windows 下)

```
kate try_pathon.py &
```

在里面输入简单的 python 代码来看一下它的工作情况：

```
#!/usr/bin/env python
print 2+2
```

第一行 (以 “#!” 开头) 告诉 shell 如何执行这个文本文件 它必须在那里用于告诉 Linux 这个特别的文本文件是一个 Python 脚本。第二行是 Python 的一个简单表达式。

在存好这个文件后，我让它变为可执行的：

```
chmod a+x try_python.py
```

然后我就打下面命令来执行：

```
./try_python.py
```

Python 是很好和很现代的程序语言。如果你喜欢面向对象的程序设计，不妨一试。在因特网上有它的很多库和扩展。比如说：

科学计算的 python (<http://starship.python.net/crew/hinsen/scientific.html>)

数学计算的 python (<http://sourceforge.net/projects/numpy>)，在工程领域应用广泛。

下面是一个有一点长，但希望能自我解释的 python 代码。一个短的说明：python 的流程控制依靠代码的缩排 它使得程序更容易读，看上去更自然，但需要花一些时间去习惯它。

```
#!/usr/bin/env python
# All comments start with a the character "#"
# This program converts human years to dog years
# get the original age
age = input("Enter your age (in human years): ")
print # print a blank line
# check if the age is valid using a simple if statement
if age < 0:
print "A negative age is not possible."
elif age < 3 or age > 110:
print "Frankly, I don't believe you."
```

```
else:
    print "That's the same as a", age/7, "year old dog."
```

tcl

(发“tickle”这个音)很流行的脚本语言

一个简单的 tcl 程序？

```
#!/usr/bin/tclsh
puts stdout {Hello World!}
```

wish

(在 X 终端上输入)这是一个 Tk (tcl 在 X 窗口上的扩展)的前台语言。常常用于生成一个程序的前台。

我如何用 Tk 写一个图形界面的程序？

Tk 是简单但功能强大的 tcl 程序设计语言的图形界面扩展。例如，我用 pico 生成一个含有简单 tk 程序的文本文件：

```
pico tryTk
```

然后在里面输入简单的 tk 代码，看看它是如何工作的：

```
#!/usr/bin/wish
button .my_button .text "Hello World" .command exit
pack .my_button
```

第一行(以“#!”开头)告诉 shell 用哪个工具来执行这个文本文件。下面两行是简单的 tk 程序的例子。首先，我生成了一个按钮叫作“my_button”并且把它放在我的类层次的根部(在“my_button”前用园点表示)。对于按钮，我在上面显示文本“Hello World”并关联上一个退出程序的命令(在按下按钮的时候)。最后一行来调整我的程序的窗口大小使它正好能放下我的按钮。

在给文件存盘后，我把它变成可执行的：

```
chmod a+x tryTk
```

然后我用下面方法执行(需要在 X 终端下，因为它需要 X 窗口的运行)：

```
./tryTk
```

对于建立 GUI 的前端，Tk 是很流行的。

ruby

一个纯的面向对象的脚本语言。相对来说，它还比较新，但它正快速的流行起来，并有可能成为将来很受欢迎的语言。

为了写一个简单的 ruby 程序，我会用我喜欢的文本编辑器来编辑并在第一行放上如下一行：

```
#!/usr/bin/ruby
```

下面是一个简单的例子来帮助我理解一些 ruby 的基本概念：

```
#!/usr/bin/ruby
#This is a comment
a = Array.new
print "Please enter a few words (type EXIT to stop):\n"
i = 0
while enterWord = STDIN.gets
  enterWord.chop!
  if enterWord == "EXIT"
    break
  end
  a[i] = enterWord
  i += 1
end
#sort the array
for i in 0...a.length-1 do
  for j in i+1...a.length do
    if a[j] < a[i]
      tmp = a[i]
      a[i] = a[j]
      a[j] = tmp
    end
  end
end
#Output the results
print "You entered " + a.length.to_s + " entries.\n\n"
for i in 0...a.length do
  print "Entry " + (i+1).to_s + ": " + a[i] + "\n"
end
```

我把我的 ruby 脚本程序存为文件“myprogram”，我在命令行里输入下面来执行它：

```
./myprogram
```

gcc filename.c

GNU 的 C 编译器。如果你了解 C 的话，它是很直接明了的。网上有大量的手册。

我如何编译一个简单的 C 程序？

打开一个你喜欢的文本编辑器并输入你的源码。例如，我可能会用 pico:

```
pico hello.c
```

然后输入 Kerningham 和 Richie (“c” 的发明者) 的 C 程序的启萌例子 :

```
#include <stdio.h>
void main(void) {
printf("hello world\n");
}
```

把文件存好然后开启 GNU 的 C 编译器来编译文件 “hello.c” :

```
gcc hello.c
```

gcc 编译器会产生一个可执行的二进制文件 “a.out” , 可用如下方式运行 :

```
./a.out
```

```
g++ filename.C
```

GNU 的 C++ 编译器。大写字母 “C” 总是用来表示 C++ 源码。如果你需要一个 “集成的系统开发环境 (IDE)” , 你可以试者用一下 kdevelop.

我如何编译一个简单的 C++ 程序 ?

就象在 c 里面一样，我打开一个文本编辑器并且写入我的程序。例如，用 pico，我写入如下程序 :

```
//This is a comment (to the end of line, C++ style)
#include <stdio.h>
#include <math.h>
#include <iostream.h>
#include <stdlib.h>
//define a function
double wheeldrop (double dGap, double dDiameter) {
double dDrop, dRadius, dNotDrop;
dRadius = dDiameter * 0.5;
dDrop = dRadius . sqrt( (dRadius*dRadius).(0.25*dGap*dGap) );
return (dDrop);
} //end of the function
//The function main is the entry point to the program
void main(void) {
double dGap, dDiameter, dDrop, dRadius, dNotDrop; //variables
for (;;) { //infinite loop
cout << "Please enter gap between track segments and \n"
<< "diameter of train wheel in inches (.1 .1 to exit): ";
cin >> dGap >> dDiameter;
if ((dGap == .1) && (dDiameter == .1))
```

```
break;
else if (dGap < dDiameter) { //do calculations
dDrop = wheeldrop (dGap, dDiameter);
printf ("The wheel will drop %f inches.\n\n", dDrop);
}
else {
printf ("Error, your train is going to crash.\n Gap bigger then
wheel!\n\n");
}
}
}
```

我把文件存为“train.C”，然后打开 GNU 的 C++ 编译器来编译“train.C”并把它生成名为“traincalc”的执行文件：

```
g++ -o traincalc train.C
```

我用下面方式来执行：

```
./traincalc
```

kdevelop

(在 X 终端里输入)K 的集成开发环境。很值得去下载（如果你的软件不包含它的话）

glade

(在 X 终端里输入)用户界面的图形生成器。

“Glade 是由 Damon Chaplin 开发的一个界面生成器。它允许 Gnome/Gtk 图形用户界面的图形和交互的构造。从 Glade，生成的界面可以存为一个 xml 文件或是在 C 的 source tree 里的 C 代码。Glade 还允许定义处理器函数的定义用于附上界面上的各种各样的事件。例如，当一个菜单上的某一项被按下时，就会去调用一个函数（名）。”（选自：

http://linuxtoday.com/news_story.php3?tsn=2000.07.16.013.04.PS.GN）

Linux 下有哪些 C 函数可用于编程？

象我这样的新手，这样的函数太多了。我从研究目录/usr/include 及其子目录下的头文件（*.h）开始。为了查找一个含有给出函数（例如：cosh）的原型的头文件，我会做下面类似的事：

```
cd /usr/include
grep -H "cosh" *.h
```

还有很多有趣的库都不是跟着标准软件配制来的，例如，GNU 的关于科学计算的库（GSL）：

<http://sources.redhat.com/gsl>

或者去

<http://www.phy.duke.edu/~hsg/sci-computing.html>

as

nasm

ndisasm

(三条命令)。汇编，一个“原汇编”和一个反汇编。下面试着给你们这些新手看一下如何使用:-)

例如：ndisasm /bin/sh |more 在我的系统上（在这个例子里用/bin/shSHELL）从二进制文件产生了一个很长的“汇编助记符”的输出，但我是一点也看不明白的：-（如果你想了解更多的 nasm，你可能会看文件：/usr/share/doc/nasm-doc-0.98/html/nasmdoc0.html

这是一个对Linux2.2.17或更高版本的intel汇编的例子：

```
;; hello.asm: Copyright (C) 2001 by Brian Raiter, under the GNU
;; General Public License (version 2 or later). No warranty.
BITS 32
org 0x05936000
db 0x7F, "ELF"
dd 1
dd 0
dd $$
dw 2
dw 3
_start: inc eax ; 1 == exit syscall no.
mov dl, 13 ; set edx to length of message
cmp al, _start . $$
pusha ; save eax and ebx
xchg eax, ebx ; set ebx to 1 (stdout)
add eax, dword 4 ; 4 == write syscall no.
mov ecx, msg ; point ecx at message
int 0x80 ; eax = write(ebx, ecx, edx)
popa ; set eax to 1 and ebx to 0
int 0x80 ; exit(bl)
dw 0x20
dw 1
msg: db 'hello, world', 10
```

在把它存为一个文本文件 hello.asm 后，我可以用下面的命令编译并输出到“hello”然后让它成为可执行的：

```
nasm -f bin -o hello hello.asm && chmod +x hello
```

下面我执行它：

```
./hello
```

这个例子我是从 <http://www.muppetlabs.com/~breadbox/software/tiny/>借来的。

为何某些人要用汇编？在汇编下编译后，我的系统上这个可执行文件的大小是 56 字节。用“C”写的同样功能的程序编译后可执行文件的大小是 13.7K 个字节。

这里有一些简短的信息来帮助我理解上面的程序：

“ ; ” 加注释（只到行尾）。
 “ msg : ” 一个标号的例子（就象在 Fortran 里一样）
 org (= “ origin ” (来源)) 声明程序在内存里的开始地址（在被执行并调入内存时）。
 db,dd,dw 是汇编的“伪指令”用来在输出文件里查入初始值。
 “ \$ ” 求出含有表达式的行的起始位置，因此你就可以用“ JMP \$ ”来表示一个无限循环。“ \$\$ ” 求出当前节的起始位置。

在 80x86(“ Intel ”)里的 32 位通用寄存器有：EAX,EBX,ECX,EDX,ESI,EDI,EBP 和 ESP。（字母“ E ”表示扩展。它在名字的前面是因为它能用作 16 位的寄存器但我们只需把名字叫作：AX,BX,CX,SI,DI,BP 和 SP。同理，它们也可视作 8 位寄存器：AL,AH,BL,BH,CL,CH,DL,DH。这里，“ L ”和“ H ”分别表示“高”和“低”的字节）。

一些 80x86 通用处理器的指令记忆码的说明：

名字	语法	注解
NOP	NOP	不做任何事
MOV	mov destination, source	转移（考备，设置）数据
XCHG	XCHG operand1,operand2	交换值
CMP	CMP operand1,operand2	比较两个操作数
PUSH	PUSH source	推上栈(把值放上栈并把栈的指针增加)。
PUSHF	PUSHF	推上标记
PUSHA	PUSHA	推入所有通用寄存器
POP	POP destination	从栈里弹出（从栈里取值，并把栈的指针减少）。弹出是栈的反操作。
POPF	POPF	弹出标记
POPA	POPA	弹出所有通用寄存器
INC	INC operand	加一
DEC	DEC operand	减一
ADD	ADD Dest,Source	加
ADC	ADC Dest,Source	加并存放
SUB	SUB Dest,Source	减
INT	INT number	执行中断
CALL	CALL subroutine	调一个子例程
RET	RET	从当前或最内的子例程返回
JMP	JMP destination	跳到目的地址执行
JE	JE destination	如果相等就跳离
JNE	JNE destination	如果不相等就跳离
JZ	JZ destination	如果是零就跳离
JNZ	JNZ destination	如果非零就跳离
JP	JP destination	如果有奇偶校验就跳离
JNP	JNP destination	如果无奇偶校验就跳离
JPE	JPE destination	如果奇偶校验是偶就跳离
JPO	JPO destination	如果奇偶校验是奇就跳离
JCXZ	JCXZ destination	如果CX寄存器是零就跳离
JECXZ	JECXZ destination	如果 ECX 寄存器是零就跳离

guile

“ Scheme ” 程序语言的一个实现。Scheme 是 LISP 语言（在过去 40 年一直是人工智能的主裁语言）的一个现代的方言。

一个小的 guile 解释器的例子：

guile

```
(+ 1 1)
(define a 2)
(/ a 3)
(= a 7)
(display "hello\n")
(system "ls")
(exit)
```

第一个命令运行 guile 的解释器。后面的四个命令作加法，定义，除法和比较，用的是波兰表达式（操作符在操作数之前）。看在这页里的逆波兰表达式。最后的命令是退出解释器。

g77

GNU FORTRAN. 网上的手册在：<http://gcc.gnu.org/onlinedocs/g77/>. 如果你真的需要很了解 FORTRAN, 你可能需要到下面网址：<http://studbolt.physast.uga.edu/templon/fortran.html> 去找一个符合你的需要的 Linux 下的 FORTRAN 编译器。

一个 Fortran 代码的小例子，它打印出 1 到 20 的数的平方和立方。

```
PROGRAM TRY_FORTRAN
  INTEGER X
  PRINT 200, "X", "X^2", "X^3"
  DO X=1, 20
  PRINT 100, X, X**2, X**3
  END DO
  100 FORMAT (I20, I20, I20)
  200 FORMAT (A20, A20, A20)
END
```

为了编译这个文件，我运行 fortran 的编译器并使用选项来认识“自由格式”的源码（我不喜欢固定格式的源码）：

```
g77 -ffree-form try_fortran.f
```

现在我运行它（编译执行文件的默认文件名是 a.out）

```
./a.out
```

expect

“程序对话”的脚本语言。看 man expect

kylix

这是由 Borland(aka Inprise)提供的全新 (Feb 2001) 的商业软件。简短的说，它就是著名的面向对象的 Pascal (“Delphi”) 的 Linux 的移植。kylix 不会在你的 Linux 的 cd 上，你必须为它付钱，但是如果你想在 Linux, MS Windows 和 Web, 大量的预置件等之间做最好的快速应用程序开发的代码移植, kylix 几乎是最好的。在我看来, Delphi 明显比 MS Visual Basic 好很多。

```
javac
java
```

Java 语言的编译器和解释器。在 Linux 下，javac 和 java 实际上是用合适的选项调用 kaffe 的脚本（试一下 `cat /usr/bin/java`）

一个 java “单独”的小试验程序。我用我最爱的文本编辑器，例如 kate（在 X 终端下）输入如下 java 代码：

```
/* Comments are marked like in C++
 * A java class to display "Hello World"
 */
class HelloWorldApp {
public static void main(String[] args) {
System.out.println("Hello World."); // Print "Hello World." followed by a
newline
    }
}
```

我把它存入一个名为 `try_java.java` 文件中，现在我开始编译它：

```
javac try_java.java
```

编译产生了一个叫作 `HelloWorldApp.class` 的“字节码”文件（编译一半的文件，执行时须用解释器）。我输入下面命令运行：

```
java HelloWorldApp
```

如果想看如何把一个简单的 java applet 嵌入在网页里的例子，你可以去下面网站看一下：<http://java.sun.com/docs/books/tutorial/getStarted/cupojava/unix.html>，上面的例子就是从那里借过来的。

```
make
```

运行“make”这个工具是用来建立（编译，联接等）一个项目，这个项目在当前目录下的 `Makefile` 文件里有描述。

Make 用来让系统“更新”，当系统中的一个文件有改变，所有需要的对系统的更新操作都由 Make 完成。make 是“智能”的，它不会对那些不需要改变的文件去进行变化，通常这些文件的改变是由文件的时间来确认的。一般来说，它用于建立软件包（编译，联接……），make 还可以做其它的任务，例如，系统管理。Makefile 看上去象下面这样：

```
target : prerequisites
[Tab]commands
```

这里，目标通常都是一个文件（但也不绝对），prerequisites 是 target 文件所需要的文件。如果 target 不存在或者比任何一个 prerequisites 旧的话，“commands”就会执行。上面的第一行被叫做“归则”，第二行叫做“动作”。请注意任何一个动作行必须以 tab 字符开头。下面是一个例子，Makefile 生成一个名叫“edit”的执行文件：

```
my_program : main.o command.o
```

```
cc .o my_program main.o command.o
main.o : main.c defs.h
cc .c main.c
command.o : command.c defs.h command.h
cc .c command.c
clean :
rm my_program main.o command.o
```

为了用这个 Makefile 来生成一个执行文件叫作 “my_program”，我打：make。它会在后台工作来确认它所需要的文件，这样，首先，它会编译 “command.c” 成目标文件 “command.o”，然后再编译 “main.c” 成 “main.o”，最后它把 “main.o” 和 “command.o” 联上执行文件

“my_program”。另外还可用这个 makefile 去删除目录下的所有执行文件和目标文件。我们只要打：make clean 即可。由于目标 “clean” 不需要依赖其它前提，它不会在通常调用里执行，而是需要很清晰的调用。目标 “clean” 就是一个 “假” 目标的例子。

yes

生成一个含有 “yes” 并用不结束的字符串的输出（可按<Ctrl><c>终止或掉电）。

听上去象是一个很蠢的工具，但它可用于在命令行写一个很简单的程序。例如，下面这个用于确定在 100,000 个随机数里各个数字出现的频率的例子（所有的命令在一行上）：

```
yes | sed '100000q' | awk
'BEGIN{srand();u=6*log(10)}{printf"%e\n",rand()*exp(rand()*u)}'|
cut -c1 | sort | uniq -c
```

我不希望这个例子会把你吓着，它确实说明了过时的 UNIX 的命令行操作同样可以玩得很复杂（和功能很强），就象你所期待的一样。如果你想知道为什么数字出现的频率是各种各样的（直觉上来说如果数字是随机的，它应该是一个常数）就到我借来这个例子的网站上去看一下吧：

<http://www.newscientist.com/ns/19990731/letters4.html>

7.3 数学工具

dc

一个任意精度的命令行“逆波兰表达式(RPN)”计算器。

dc 来自现代数字计算机的主要概念-堆栈。计算机中的堆栈和厨房里的一叠盘子没有什么不同：最后进来的最先出去（即 LIFO，亦称后入先出）。它和先入先出（FIFO）的队列正好相反。

你只能对位于堆栈顶部的数值执行运算。两个最基本的运算是：压入（在堆栈顶端置入）和弹出（从堆栈顶端取回）。一元运算从堆栈中弹出一个数值（“一元”指“要求一个操作数”），二元运算从堆栈中弹出两个数值（“二元”指“要求二个操作数”），三元运算从堆栈中弹出两个数值（“三元”指“要求三个操作数”）。每种情况下，运算结果总是被压回堆栈的顶部。

逆波兰表达式计算器（普通手持式）在技术人员和学者中很常见。它从不要求输入圆扩号。

历史 无扩号逻辑是由波兰数学家 Jan Lukasiewicz (1878-1956)在第二次世界大战前发明的。起初，运算符位于操作数之前。但为便于计算机使用，运算符被置于操作数之后，这也就是“逆波兰表达式”中“逆”字的由来。

下面练习对堆栈操作：

```
dc [启动任意精度逆波兰表达式计算器]
1 [将数值“1”压入堆栈]
2 [向堆栈中压入另一个数值]
3 [向堆栈中压入另一个数值]
4 [向堆栈中压入另一个数值]
f [显示整个堆栈的内容；你应该看到 1 2 3 4]
p [显示堆栈顶部的数值但不改变堆栈本身；你应该看到 4]
+ [执行加法运算（二元运算），即将最上面的两个数值（4，3）弹出堆栈，然后在把运算结果（7）放入堆栈]
p [显示堆栈顶部的数值；例如，刚才加法运算的结果（7）。]
p [再次显示堆栈顶部的数值以证明堆栈的内容没有被刚才的显示操作(p)改变。]
* [执行乘法运算（二元运算）；即弹出堆栈中的最后两个数值，然后将运算结果（14）放入堆栈]
p [显示刚才乘法运算的结果（14）]
P [弹出堆栈中的最后一个数值（14）]
p [显示堆栈顶部的数值]
2000 [向堆栈中压入一个较大整数]
k [为堆栈顶部的数值（例如：2000）设置精度]
1 [向堆栈中压入另一个数值]
f [显示整个堆栈的内容]
701 [向堆栈中压入另一个数值]
/ [堆栈中的最后两个数值执行出发运算，例如：2000 除以 701]
p [显示最后除法运算的结果]
q [退出任意精度逆波兰表达式计算器]
```

请注意，逆波兰表达式从不需要扩号。可以阅读 dc 的联机帮助（man bc）了解更多功能。

bc

任意精度交互式计算器。输入“quit”退出 bc。在进行浮点除法运算前输入“scale=20”（或其他精度），否则将得到取整后的商值而不是一个浮点数。

kcalc&

xcalc&

（X 图形终端上）标准图形界面计算器。

```
e '2*3/4+sin(pi/2)'
```

这个“e”表达式求值程序不包含在 RedHat 7.x 的光盘中。不过，在众多的命令行计算其中我最喜欢这个。可以试着从这个地址下载：<http://www.softnet.tuc.gr/~apdim/projects/e/>

```
expr 1 + 1 / 3
```

执行整数运算。“expr”并不意味着一个计算器，大多数情况下在脚本程序中用做流程控制。上面的例子回返回结果“1”，因为在整数除法中 1/3 等于 0 是正确的。

octave

octave 是一种用于数字计算的高级交互语言，非常接近“matlab”，包含在任何完整的 Linux 发布中。

scilab&

(X 图形终端上) 一个强大完善的数字计算系统，在一定程度上与“matlab”相似，但有一个更加紧凑的图形界面。如果不是有非常复杂的运算要求就不要去试它了。这个软件包含在 RadHat 7.0 “powertools” 光盘中。主页：<http://www-rocq.inria.fr/scilab/>

下面的小例子示范了一些线性代数运算，我的输入用黑体表示：

```
-->a=[1 1 1;2 3 4]
a =
! 1. 1. 1. !
! 2. 3. 4. !

-->b=[1 1; 2 2;3 3]
b =
! 1. 1. !
! 2. 2. !
! 3. 3. !

-->c=a*b
c =
! 6. 6. !
! 20. 20. !

-->d=inv(c)
d =
1.0E+14 *
! 11.258999 - 3.3776997 !
! - 11.258999 3.3776997 !

-->
```

```
head -c 8 /dev/random
cat /dev/random | od
cat /dev/urandom | memencode
```

(三个命令) 例子显示了如何在 Linux 命令行通过读设备“random”或“urandom”生成随机字符。第一个命令通过读设备“random”产生大约 8 个字符，这样能产生高质量(难于预测)的随机数。这个过程会变得很慢，如果你的机器“随机信息”已经用尽的话(比如，当产生大量的随机字符)。这是的解决办法是等待或者用键盘任意输入、移动鼠标、切换终端、让你的硬盘读写等来生成等多随机噪音(“随机信息”)。第二个命令不断地产生随机字符，但以八进制显示。它只能用 Ctrl+c 来中断。第三个命令使用了设备“urandom”，但生成大量随机字符的时候比“random”快。当系统的随机信息少时，“urandom”的输出的随机程度可能会下降，但除了一些有特殊要求的应用程序外，仍能满足绝大多数情况的要求。输出被过滤成可打印的 MIMIE 格式(互联网上用于邮件的 7 位编码标准)。Linux 下的详细的随机数生成原理和实现描述源代码可以在你的 Linux 系统/usr/src/linux/drivers/char/random.c 文件中找到。

```
factor 1053385846622239345
```

查找任意整数的所有质因数（素数）。一个整数的因数是能够整除这个整数的数。例如：6 的因数有：1，2，3 和 6。

R

一种针对复杂的统计计算的编程语言/计算环境。输入“quit()”退出。

gnuplot

图表创建工具。非常适合非交互式（批处理）任务，但对交互式使用并不简单。你可以在 <http://www.duke.edu/~hpgavin/gnuplot.html> 找到很好 gnuplot 说明。

7.4 杂项

我怎样才能运行微软 Windows 程序（用“wine”）？

不要为了运行微软 Windows 程序而安装 Linux。如果需要 Windows 程序，你可能想把你的计算机配置成双重启动。

当然，Linux 下的“wine”库可以让你运行某些微软 Windows 程序，尽管是以牺牲速度为代价。在我的系统上（安装了 Wine），我能通过在 X Windows 的终端窗口中输入：

```
wine /mnt/dos_hda1/windows/sol.exe
```

运行微软的纸牌游戏。

其中，/mnt/dos_hda1 是可以并且已经挂接微软 Windows 所在硬盘分区的目录名。

如果你还没有安装 Wine，那就将 Mandrake 光盘放入光驱并让系统识别到它，然后以系统管理员的身份（root）执行：

```
cd /mnt/cdrom/Mandrake/RPMS/  
rpm -ihv wine-991212-1mdk.i586.rpm
```

Mandrake 软件包能够与 RedHat 兼容，所以你可以用它来安装 RedHat 缺少的软件包。

如果我有两个或更多的硬盘，能够使用 RAID 吗？

RAID=“冗余磁盘阵列（Redundant Array of Inexpensive Disks）”。RAID 能够用一块磁盘去镜像另一块磁盘以达到在发生磁盘损坏的情况下保护数据、维持系统正常运转的目的。Linux 包含了一套 RAID 工具供你定制一个符合你要求的 RAID 系统。它们包括：

```
mkraid - 初始化/升级 RAID 磁盘阵列  
raid0run - 启动老式（无超级块）RAID0/线性阵列[译者注：原文如此。意即通常所说无镜像]  
raidstart - 启动 RAID 设备
```

raidstop – 停止 RAID 设备
raidtab – RAID 设备配置文件

RAID 把两个或更多磁盘合成一个单独的逻辑设备。不同级别的 RAID 有：

RAID 0 (条状) 简单地把两个或多个磁盘合成一个单独的逻辑设备，没有任何冗余。经常和 RAID1 及 RAID5 一起使用。RAID0 + RAID1 被称作 RAID10, RAID0 + RAID5 被称作 RAID50。

RAID 1 (镜像) 结合两块磁盘，每一块磁盘存储相同的数据。

RAID 4 结合了三块或更多的磁盘，其中一块磁盘专门用作奇偶校验。任何一块磁盘损坏时，整个逻辑设备保持完好，但性能下降。由于性能的缘故，这种配置并不常用。

RAID 5 结合了三块或更多的磁盘，并将奇偶校验分布在所有的磁盘上。原理和 RAID4 相似但性能明显更好。

在 http://www.osfaq.com/vol1/linux_softraid.htm 能够找到更多的有关信息。

RedHat Linux 7.2 在系统初始安装过程中具备了轻而易举地创建软件级 RAID 的功能。这个过程的描述可以在 <http://www.redhat.com/docs/manuals/linux/RHL-7.2-Manual/custom-guide/software-raid.html> 找到。简而言之，在安装过程中的“硬盘分区”部分：

- (1) 创建“software RAID”（软件级别的 RAID）分区。你不能为每一个“RAID 类型”的分区指定一个 mount point。但你可以给出每个分区的大小，使它“能够扩展”或者使它成为主分区。
- (2) 按“Make RAID”（创建 RAID）按钮
- (3) 在出现的对话框中输入：这个 RAID 阵列的 mount point、分区类型、RAID 类型（RAID0、RAID1 和 RAID5 中之一）、RAID 成员分区（你在第一步中创建的）及备用分区数（RAID1 和 RAID5 需要）。（当软 RAID 出现磁盘损坏时，备用分区将自动被使用）在第一步中，你应该？？？
- (4) 按“OK”按钮，检查“Drive Summary”（磁盘信息）是否你的 RAID 阵列显示正确。

注：如果你在为 /boot 创建 RAID 分区，必须选择 RAID 1 而且必须使用最初的两块硬盘之一（IDE 优先，SCSI 次之）。如果你不是为 /boot 创建 RAID 分区，而是创建根 RAID 分区，这个分区必须是 RAID 1 而且必须是最初的两块硬盘之一（IDE 优先，SCSI 次之）。

使用 shapecfg 调整网络流量

在 <http://oreilly.linux.com/pub/a/linux/2000/08/24/LinuxAdmin.html> 能找到很有用的信息。尽管看起来我永远不需要在家里调整网络流量，但 shapecfg 仍然是一个有趣的练习。

参看附页：如何升级系统内核

许可证、鸣谢及其它

LNAG 许可证

《Linux初级系统管理员手册》(LNAG)(简称“本手册”)遵照“开放内容许可证(the Open Content License)”(<http://opencontent.org/openpub/>)标准并附带以下条件:

本手册在不断更新中。因此,为了向读者提供最佳服务,除非已经被原作者以外的其他作者修改,任何发布者应提供一个尽可能新的版本。例如:本手册的最新版本或者距离你通过网站服务、刻录光盘或打印发布本手册时的时间一年之内的版本。本手册最新版本可以在<http://sunsite.dk/linux-newbie>找到。

本许可证意味着任何可公开访问的镜像必须每年至少更新一次,如果存在新版本的话。如果你无意更新,请不要创建本手册的镜像。

本手册中所有的代码示例遵循通用公共许可证(GNU,<http://www.gnu.org/copyleft/gpl.htm>)发布。

可以通过电子邮件联系本手册的维护者Stan和Peter Klimas(linux_nag@canada.com)。

鸣谢

Linux初级管理员手册(LNAG)在丹麦奥尔堡(Aalborg)大学的SunSite服务器上免费发布(<http://sunsite.dk/linux-newbie>)。感谢Esben Haabendal Soerensen <bart@sunsite.auc.dk>。

感谢“linsup.com”(<http://linsup.com/>)提供免费空间,并及时更新澳大利亚的镜像(<http://linsup.com/newbie/>)。感谢Kenan Bektas (VP Engineering)提供免费空间并及时更新北美镜像(<http://dbstreams.ca/mirrors/linux-newbie/>)。还要感谢Andamooka将LNAG收藏与众多优秀的免费书籍中。我们需要想办法如何更新它。

主要的帮助和建议来自(按字母顺序排序):Alan W. Irwin <irwin@beluga.phys.uvic.ca>, Benjamin Smith <bens@saber.net>, Barbara Klimas <bklimas@magma.ca>, Ben McCosh <ben@albagroup.com>, Bill Staehle <staehle@netvalue.net>, Bill Unruh <unruh@physics.ubc.ca>, Brian Kelsay <ripcrd6@kcinter.net>, Ding-Hou Lee <dl105@columbia.edu>, Gary <swear@aa.net>, Greg Mizell <GMizell@peek-traffic.com>, Jaakko Alarto <pikkumyy@ikiliekki.org>, Jeff Greenlee <greenj@analogy.com>, jeff covey <jeff.covey@pobox.com>, Jo <rainbow@linuxfocus.org>, Juhapekka Tolvanen <juhtolv@st.jyu.fi>, Ken Foskey <waratah@zip.com.au>。 特别感谢波特兰Linux用户小组的校阅!

感谢更多的向我们表达看法、帮助纠正谬误、提出更好建议或更好修辞的朋友。我们永远感激任何反馈,迄今为止还没有收到读者愤怒的指责。我们感到抱歉不能回答令你烦恼的具体问题:((), 不能为单独的用户提供免费服务。同时抱歉我们不为微软 Windows 用户提供任何支持。

感谢数以千计的 GNU/Linux 开发人员。还有必要提到 Richard Stallman 和 Linus Torvalds 的名字吗?

其它

你可以看出本手册是由新手写给新手的，任何主题都不应该被视为权威解释——已经有足够多详尽的文档了，大部分在你的 Linux 系统的 /usr/doc 目录中，当然一般很难读懂。如果你发现了可能混淆或误导 Linux 新用户的错误（本手册仍在不断更新，当前的版本可能有一些错误），请立即通知我们。不要把你的身家性命依靠在这本手册上！

如果你也想做一些贡献（任何你感兴趣的 Linux 话题），我们会非常愿意接受，并乐于把你的名字包括进来，等等。唯一的条件是你的建议应该易于我们这样的新手理解。

如果你创建自己的镜像，请确保定时更新。本手册（LNAG）仍在不断完善，发现的错误之处会被定期更正。我们的许可证要求更新本手册（如果在 <http://sunsite.dk/linux-newbie> 有新的版本），否则请删除你的过时的镜像。

希望本手册能对你有所帮助。

Best Regards,

Stan and Peter Klimas
(linux_nag@canada.com)

中文译本后记

一封感谢信给华枫论坛 (www.chinasmile.net) 的朋友们

“骐驎一跃，不能十步，弩马十驾，功在不舍”

?? 引自 《荀子》 - 劝学篇

四个月前，在这里招募人员翻译一本 linux 的书，现在全书已经翻译完毕，并经过了第一次检查和修订。全书 236 页，打印完拿在手里很薄，份量却有一点沉。没有大功告成的喜悦，却有如体育课上坚持咬牙跑完全程后无尽疲惫中夹着些许宽慰。

因为人员是在这里募集的，这里再一下老 JOE 的这块地，向过去几个月里一起“战斗”的网友们的深切致谢：

特别感谢 Allen，在经历最困难的时候，即使学业繁忙，也一直和我坚守在最后的战壕里；在打扫残局的无尽疲惫里，没有你的一再伸手“救援”，离最后完成仍然有待时日；

特别感谢 WangLing，敏捷迅速，译文准确流畅无可挑剔，总是在完成自己的部分后一再主动请缨，是你给整个小组注入一种不知疲倦的前进的力量；

特别感谢 Sunny，工作，家庭两头忙，在受到“要实在太忙可以交给别人”的一再“诱惑”之下绝不“投降”，终于慢工出细活--非常感谢你的坚持和信任；

特别感谢 flx，第一个完成自己手头任务的人，是全队的“急先锋”和“一级劳动模范”；

特别感谢北美老牛 wendell，你的经验和建议使本书获益良多；

特别感谢 jeff, hummingbird, emerson 等网友的兢兢业业，尽职尽责，没有你们的不懈努力，这项目进行不到一半恐怕就散架了；

特别感谢 cstring, lisa, Zhang Guo Bin，你们的热情参与，是这本书能够顺利完成的。

此外，还有其他热情提供各种建议和帮助的朋友，在此一并致谢。

“做一件事，最难的是开始，最重要的是坚持。如果坚持，就会达到目的。”

和这里的诸位共勉。

SHADOWFAXLINUX 小组电子邮件列表

幻影快递小组邮件：

shadowfaxlinux@hotmail.com

所有小组成员的邮件：

Allen	xieyx97@yahoo.com
Cstring	harrysongca@yahoo.ca
Emerson	emerson7509@hotmail.com
flx	fzlx@yahoo.com
G Zhao	gzhao_2001@hotmail.com
Hummingbirds	wangzh_30@hotmail.com
Jeff Lee	jefflee416@sympatico.ca
Lisa	erinst@yahoo.com
Sunny Chen	chenpager@yahoo.com
trt	suhuaihe@hotmail.com
Wang Ling	wangling88@hotmail.com
Wendell	wendell_67@hotmail.com
Yao Ming Wei	yimingwei@hotmail.com
Zhang Guo Bing	guoping_zhang@hotmail.com