

Ubuntu Linux CLI Guide

Ubuntu Linux命令行简明教程

Magi Su
magisu@163.com

November 29, 2006

前言

这篇文章试图说明几个问题：

1. Linux并不很难，功能强大，并且很有趣。
2. Windows虽然更好学，但是没有Linux好玩。
3. 如何从Windows跳槽到Linux。
4. 一些常用命令的使用方法。
5. 该文章的作者写作风格更接近美式调侃型而非中式教条型。
6. 作者推荐你使用fish，因为它更方便。然而考虑到大部分人用的都是bash，这里也只好拿bash说事了。
7. 关于网络设置部分，已经对中国科学技术大学寝室电话上网部分进行了代码优化。

我写作这本小册子的动力在于不耐烦于Panda同学每天翻来覆去的电话召唤。所以如果发现这本小册子有益的话，请你们先感谢Panda同学。

其实，我认为所有Linux的教程都没什么大用处，包括这本小册子。对一个人入门还可能有点帮助，但是真正需要提高的话，阅读man里面的文献才是王道。

为了让你反复查阅，从而对系统的一些东西印象深刻，我在小册子中添加了大量的交叉参考。你如果懒得前后翻书，就可以不看，前提是你得知道我在说什么。

Magi Su
University of Science and Technology of China
0403 CHP

Contents

1	序言	1
1.1	版权	1
1.2	约定	1
1.2.1	字体和记号	1
1.2.2	称谓	2
1.3	免责声明	2
1.4	作者使用的系统	2
2	一些对于Linux的误解	3
3	新的观念	5
3.1	文件系统	5
3.1.1	文件	5
3.1.2	目录系统	6
3.1.3	分区和盘符	7
3.1.4	挂载	7
3.1.5	文件系统格式	8
3.1.6	文件系统的自检	8
3.1.7	磁盘整理	8
3.1.8	磁盘缓冲	8
3.2	用户管理	8
3.2.1	用户系统	8
3.2.2	用户权限	9
3.3	开机和关机	9
3.3.1	开机	9
3.4	多终端概念	10
3.5	命令行界面	10
3.5.1	铺垫一下	10
3.5.2	字母大小写	11
3.5.3	多种命令行界面	11
3.6	Shell的概念	11
3.6.1	fish	11
3.6.2	bash	12
3.6.3	ksh和csh	12
3.6.4	sh	12
3.6.5	busybox	12
3.7	多任务系统	12
3.7.1	前台和后台	12

3.7.2	进程标志	12
3.7.3	进程优先级	13
3.7.4	进程状态	13
3.8	图形界面	13
3.8.1	简单介绍	13
3.8.2	进一步讨论	14
3.9	内码	14
3.10	设备-文件映射	14
3.11	配置文件	14
4	常用命令	17
4.1	公共部分	17
4.1.1	参数	17
4.1.2	文件名	17
4.1.3	多重文件输入	18
4.2	目录控制	18
4.2.1	cd	18
4.2.2	mkdir	18
4.2.3	rmdir	19
4.2.4	pwd	19
4.3	文件管理	19
4.3.1	ls	19
4.3.2	dir	21
4.3.3	cat	21
4.3.4	head和tail	22
4.3.5	mv	22
4.3.6	cp	23
4.3.7	rm	24
4.3.8	find	24
4.3.9	ln	26
4.4	压缩解压	27
4.4.1	gzip/gunzip	27
4.4.2	bzip2/bunzip2	28
4.4.3	tar	28
4.4.4	zip/unzip	30
4.4.5	rar/unrar	32
4.4.6	7z	32
4.4.7	其他压缩工具	33
4.4.8	压缩软件的比拼	33
4.5	文件比较	33
4.5.1	cmp	33
4.5.2	comm	34
4.5.3	diff	35
4.6	重定向符号及其应用	35
4.6.1	重定向符号	35
4.7	管道符号 之后的常用命令	36
4.7.1	more	36
4.7.2	less	36
4.7.3	grep	36
4.8	Shell增强命令	36

4.8.1	history	36
4.9	辅助命令	36
4.9.1	whereis	36
4.9.2	which	37
4.9.3	who	37
4.9.4	man	37
4.9.5	fortune	38
4.10	程序运行控制	38
4.10.1	&符号	38
4.10.2	CTRL-C和CTRL-Z	38
4.10.3	ps	38
4.10.4	kill	40
4.10.5	nice	40
5	编辑器	43
5.1	一些古怪的概念	43
5.2	nano	43
5.2.1	系统结构	44
5.2.2	使用方法	44
5.2.3	设置文件	46
5.3	vim	47
5.3.1	著名教程	47
5.3.2	基本概念	48
5.3.3	基本用法	48
5.3.4	进一步的应用	48
5.3.5	进二步的应用	48
5.3.6	进三步的应用	48
6	管理命令	49
6.1	账号控制	49
6.1.1	su	49
6.1.2	sudo	49
6.1.3	passwd	50
6.1.4	useradd	50
6.1.5	userdel	50
6.1.6	chmod	51
6.2	文件系统命令	51
6.2.1	mount	51
6.2.2	umount	52
6.2.3	ntfs-3g	53
6.2.4	fsck	53
6.2.5	df	53
6.2.6	cdisk	54
6.2.7	mkfs	57
6.3	其他命令	57
6.3.1	shutdown	57

7	Ubuntu包管理系统	59
7.1	基本概念	59
7.1.1	Ubuntu下的软件	59
7.1.2	deb包的概念	59
7.1.3	源的概念	60
7.1.4	本地数据库	60
7.2	apt-***软件套装和aptitude	60
7.2.1	apt-get	60
7.2.2	apt-cdrom	61
7.2.3	apt-key	61
7.2.4	aptitude	62
7.2.5	安装/卸载软件包综述	71
7.3	其他包管理命令	72
7.3.1	dpkg	72
7.3.2	alien	72
8	系统核心配置文件	73
8.1	/etc/passwd	73
8.2	/etc/fstab	73
8.3	/etc/apt/sources.list	74
9	GNU编译环境	77
9.1	GNU编译器概述	77
9.1.1	gxx	77
9.2	调试器	78
9.2.1	gdb	78
9.2.2	gdb的前端	78
10	命令行下的Internet	79
10.1	ADSL上网设置	79
10.1.1	pppoeconf	79
10.1.2	pon	80
10.1.3	poff	80
10.1.4	plog	80
10.2	命令行下的网上“冲浪”	81
10.2.1	ping	81
10.2.2	wget	82
10.2.3	w3m	82
A	Grep学习笔记	i
A.1	版权声明	i
A.2	目录	i
A.3	正文	i
A.3.1	grep简介	i
A.3.2	用法	ii
A.3.3	参数	ii
A.3.4	grep正则表达式元字符集(基本集)	iii
A.3.5	用于egrep和grep-E的元字符扩展集	iii
A.3.6	POSIX字符类	iv
A.3.7	实例	iv

<i>CONTENTS</i>	VII
B 两个系统下的软件对比	v
C 如何获得帮助	xi
C.1 本地帮助	xi
C.2 网上求助	xi
C.2.1 在何处求助	xi
C.2.2 提问的方法	xi
D Ubuntu的历史	xiii
D.1 Ubuntu的简介	xiii
D.2 Ubuntu历史	xiv
D.3 Ubuntu动物进化图	xiv
E 更新列表	xvii
E.1 更新说明	xvii
E.2 感谢	xvii
E.3 特殊感谢	xvii

Chapter 1

序言

天下有大勇者
猝然临之而不惊
无故加之而不怒
此其所挟持者甚大
而其志甚远也
——苏轼 留侯论

本章是一些废话，如果你时间很紧张，可以不看。

1.1 版权

如所有自由软件爱好者所喜爱的方式，这篇文章在保证作者的名字不被修改或者删除的前提下可以进行任意的传播。如果修改，请给作者发一封邮件解释你所修改的内容，修改者应当在文章开头加上自己的名字，但请放在原作者之后。

因为没有找到文章的GNU/GPL宣言，看样子只好这样声明一下了。

1.2 约定

1.2.1 字体和记号

字体

如果你能看懂 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 记号我会非常开心，因为我就不用讲字体了。然而似乎不是很现实。

正文使用的是宋体字，其中如果有斜体成分那就是应该注意的词语了——当然，也有可能是书名。每一章的开头会有一小段引用的名言，用黑体字表示。正文之间会穿插一些仿宋体字，其作用相当于胡说八道，或者是一些注解。如果我想向你提醒某些内容，我会使用黑体字标记“注意”或者“推荐阅读”等字样。

对于英文字体，如果是屏幕中显示的字体，我会用正体，如果是输入的字体我会使用斜体。

记号

文章中所有xxxx代表一些随系统的不同而不同的字符，括号[]包含的内容是可以变化的参数，有时是可有可无的参数¹，正如大部分帮助文件中的约定一样。如果是[文件名]，在不明确指明的前提下是可以使用通配符*和?的。

1.2.2 称谓

Linux, BSD都属于类UNIX操作系统，我会在今后混用Linux和UNIX单词，虽然UNIX是商业产品而Linux是自由软件。至于BSD，没有用过，虽然它血统高贵，是名校Berkley的作品。如果你希望能够构建服务器等系统，我个人推荐BSD，因为相比自由软件开发人员，我更信赖Berkley的技术人员。

对于好的东西我们要极力赞扬，不管它是谁家的。
如果你是Ubuntu的Fans，特别想打倒我这句话，我只好说：打倒吧打倒吧。

文章中所有M\$记号代表我们亲爱的微软公司Microsoft。如果提到Windows命令，我的意思也可能是Windows下的DOS命令。

1.3 免责声明

文章的作者的水平不是很高，对文章的读者可能有一定误导作用。作者对此表示歉意——你明白我想说什么：既然你无偿得到了这篇文章，那么作者对一切不良后果概不负责。尤其是我会在文章之中教唆你输入一些可能危及整个系统的命令，请你自己详细甄别。如果你仔细看了这些命令前面的内容，我相信你有这个能力。

Windows我已经有近一个月没有碰过了，删都删了，所以如果里面提到Windows的操作有误的话，请指明。谢谢。

1.4 作者使用的系统

我使用的是Ubuntu 6.10，图形终端是KDE。虽然大部分Ubuntu用户都可能用的是标准配件Gnome，然而我仍然选择KDE，因为它是C++开发的。

如果你发现我使用的软件你不太明白是什么，请参阅B。

¹好比选做题一样

Chapter 2

一些对于Linux的误解

你要变革梨子
就要亲口尝一尝梨子的滋味。
——毛泽东

本章文字介绍一些常见的对于Linux的误解。

1. Linux很难学会，也不好

难学会是正确的，因为它功能太强大了。做一个类比：铁锹比推土机好用——事实上推土机还有学习班，铁锹恐怕连教用法的人都没有——但是推土机更强大。

想想，对于光盘系统，`mount` 6.2.1指令-o参数后面可以跟15个辅助选项，如果想要全部学会恐怕难于上青天¹。然而常用的却不是很多。

如果你用习惯了Linux，你就会发现它还是非常好用的。图形界面已经几乎胜过Windows的效果，而命令行界面同样可以用的很顺手——我希望你看完这本书，以及我推荐的其他参考书目，这样你会习惯于Linux命令行操作。

2. Linux只在少数人之中流行

事实上，国外许多大学都采用Linux/BSD²系统作为标准的系统，因为这些系统是免费的，并且功能强大，更重要的是它上面的大量应用软件都是免费和开源的，后者意味着只要你懂得编程，就能够自己定制这个软件。

3. 不同版本的Linux不一样

但是它们使用的内核几乎是相同的，一些常用的Shell命令也是一样的。

常见的Linux版本有这些：

- Red Hat Linux
- Fedora Core³

¹如果我告诉你`mount`的英文文档有1432页的话，你会不会更恐慌？

²BSD是一个和Linux类似的操作系统，事实上二者都属于类UNIX操作系统。

³RedHat和FedoraCore虽然是同一个公司出品的，但是FedoraCore是面向桌面用户的，而RedHat是面向企业用户的。有这样的一个俏皮说法，就是FedoraCore是（RedHat公司让大家免费使用的（以便于发现错误的（RedHat企业版的试用品）））。鉴于你可能没有受过GRE长难句训练，请按括号断句。

- **Mandrake/Mandriva Linux**
- **MagicLinux⁴**
- **Debian Linux⁵**
- **Ubuntu Linux⁶**

它们虽然由不同的公司（或者团体）开发，但是在具体操作上几乎是相同的。

当然，不同公司的产品存在一些差别。例如RedHat和FedoraCore的软件包是rpm格式的，而Debian和Ubuntu则是deb格式。

4. Linux图形界面< (= / >) 字符界面

自古以来就有GUI（Graphics User Interface图形用户接口）和CLI（Command Line Interface命令行接口）哪个更强之争。我个人倾向于CLI功能更加强大。然而，方便起见，图形界面也是必不可少的。我现在写作这个小册子就是用的Kile。而我平时写程序是用的GVim，都是GUI的产品。然而如果是要复制文件之类，我宁可用Konsole而不是Konqueror或者Dolphin。

5. Linux不如Windows

这是微软公司的一个巨大谎言。从下面几方面讨论：

稳定性 根据HP公司的报告，Linux系统的死机次数是Windows系统的死机次数的十分之一。

安全性 由于严格的系统权限机制，Linux下几乎没有病毒和流氓软件⁷。而根据Kaspersky公司的报告，Windows下的病毒已经达到20万之多。

易用性 在几年前你可以用这个攻击Linux，但是现在这个观点似乎说不通了。图形界面下的软件几乎同样好用。当然你有怀疑一切，反对一切，打到一切的权利，不过我的意见是：你用用就知道了。

功能 Linux下的软件功能非常强劲，可以参考LAMP网站组合（Linux + Apache + MySQL + PHP），free and more powerful。如果比较命令行软件，Windows下的软件更加软弱。此外，大部分商业软件都是先出Unix版本的，而不是Windows版本。

不要相信微软公司对Linux的评价。想想你的死敌是如何评价你的。

6. Linux和Windows水火不容

似乎曾经有过这么一段时间。不过现在随着NTFS-3g和ext2win的开发，两个系统已经可以共享文件了。相比之下，Linux对Microsoft Windows的文件系统支持更好。

⁴中国本土产品，对中文的支持很漂亮，但是是由几个业余爱好者维护的，我个人不推荐。

⁵Ubuntu就是Debian的一个衍生物。

⁶你现在正在用的Linux

⁷并不是完全没有，数量似乎在200左右，并且不流行。

Chapter 3

新的观念

旧的不去
新的不来
——中国古代谚语

本章介绍关于Linux系统的一些新的观念，不同于Windows的观念，如果你熟悉Windows的使用，我想你可以很快看懂。如果你连Windows都不熟悉，那么我希望你能够参考一些其他Linux书籍，我没有自信这一章能够从零开始。

3.1 文件系统

3.1.1 文件

文件系统里面当然是以文件为主的。文件的观念现在连只会打网游的小P孩都知道了，这里不再赘述。但是有必要谈到链接这个概念。链接是为了满足一个相同的文件可能需要放在不同的目录之中而产生的。

隐藏文件

Linux通过设置文件权限来隐藏文件，避免不必要的人看到。这个参见3.2.2。

当然，有些文件，特别是软件的配置文件往往不希望在用ls命令列表文件的时候看到，因为很多，平时又不用。这时可以让文件名前加一个.（小数点）记号，这样就会在ls后不加参数时隐藏这些文件。

链接

有时候一个相同的文件，特别是很大的文件，没必要在许多目录中都同时存在，这时可以在一个目录之中放下这个文件，而在其他目录中只放一个链接。

链接分两种，硬链接（hard link）和符号链接（symbolic link）。硬链接实际上是为文件建一个别名，链接文件和原文件实际上是同一个文件；而符号链接建立的是一个指向，即链接文件内的内容是指向原文件的指针，它们是两个文件。硬链接只能用于文件，不能用于目录；符号链接可以为目录建链接。硬链接不能跨文件系统操作。¹

¹本段文字摘自<http://www.chinaunix.net/bbsjh/3/2648.html>

符号链接可以指向一个不存在的文件。

关于文件系统，请参阅3.1，:-)。有关链接的命令，详见4.3.9。我的一些有助理解链接的实验，可以参阅4.3.6中的例子。

3.1.2 目录系统

似乎应该先讲下分区（partition），但是Linux下的分区是和目录系统离不开的。所以我会等到3.1.3再讨论。Linux下的目录和Windows下的目录概念基本一样，你可以理解成为装文件的一个箱子。其中，最上层的目录是根目录/，好比Windows下最上层的是“我的电脑”²。

以下是Linux的目录树列表：

/	根目录，就好比一棵树的主干一样。
/bin	binary的缩写，系统核心应用程序的目录。例如cp, mv等 [†] 。
/sbin	似乎是system binary的缩写，系统核心应用程序的目录。例如fsck等 ^{††} 。
/home	用户的温馨的家所在的位置。每个用户的主目录都在这里（当然，前提是如果有的话）。在登录之后，你可以用~符号来指定你的家目录。
/usr	user的缩写，应用程序和库所在目录。
/usr/bin	应用程序。
/usr/games	游戏。
/usr/lib	library的缩写，库文件。
/usr/lib64	64位库文件。
/usr/fonts	字体文件。
/usr/share	大量的程序文件，库文件，程序文档，设置文件等等。里面doc目录是文档中心。
/usr/X11R6	图形系统文件。
/usr/include	软件开发人员用的头文件。
/usr/sbin	应用程序。
/usr/local	用户在本地添加的应用程序。
/usr/src	source的缩写，源代码。
/cdrom	到/media/cdrom的一个链接。
/etc	一些系统程序的配置文件，例如文件系统挂载配置文件就是/etc/fstab。
/dev	device的缩写，设备映射文件。
/lib	库文件。
/media	通常分区挂载在这个目录下的子目录里面。
/mnt	mount的缩写，以前分区常挂载在这里，现在可谓今不如昔了。
/opt	option的缩写，附加软件的安装位置。
/proc	process的缩写，系统正在运行的应用程序的映射文件。
/root	超级用户所在的目录，小朋友不可以随便偷看的啊:-)。
/srv	server的缩写，服务器系统所在目录。
/sys	system的缩写，系统状态映射文件所在目录。
/tmp	temporary的缩写，临时文件所在目录。
/var	variable的缩写，经常发生改变的的系统文件所在目录。
/lost+found	文件系统崩溃之后，使用fsck命令修复，则命令产生的恢复的文件放到这里。

[†]参见4.3 ^{††}参见6.2

通过这张乱七八糟的表我认为你可以基本了解Linux的目录大体结构了。

²似乎Windows的文件系统层次不清，我是这么理解的，因为我的文档居然可以放在我的电脑平起平坐的位置上。

3.1.3 分区和盘符

从最底层开始，考虑以下分区表，在这里我们不探讨Linux的ext3和Windows的NTFS的区别，假定它们都是所谓“抽象的分区”，同样，不考虑扩展分区和主分区之间的区别。



在Windows下，它们被依次命名为C, D, E, F。而在Linux下，它们则被依次命名为/dev/hda1, /dev/hda2, /dev/hda3, /dev/hda4。对/dev/的详细讨论见3.10。

注意 Linux下标记1, 2, 3, 4的顺序和实际分区的顺序可能不同，正如上边四个分区在Windows下同样可以被标记为D, E, F, C一样。
命名的法则是前两个字母：

hd 硬盘/光驱等

sd USB盘，移动硬盘等

第三个字母指示硬件中盘的序号，例如a, b等。第四个数字指示传统意义上的分区。注意数字并不代表真实分区顺序。第一个分区也许并不是序号1。

例如你的第一块硬盘里面的第一个分区可能是/dev/hda1。而你的USB盘可能是/dev/sda1，等等。一般来说，光驱是/dev/hdc。

3.1.4 挂载

Linux下的分区和Windows下的分区同样需要挂载³。所不同的是，Windows下挂载就是赋予盘符，而Linux下挂载则是把磁盘的内容放在某一个目录下。请试着输入以下命令，如果你现在能进入命令行的话。

```
xxxx@xxxx:/home/xxxx$mount
```

你会看到类似下面的内容：

```
/dev/hda5 on / type ext3 (rw,errors=remount-ro)
...
/dev/hda2 on /boot type ext3 (rw)
/dev/hda9 on /usr type ext3 (rw)
/dev/hda6 on /home type ext3 (rw)
/dev/hda12 on /multimedia type ext3 (rw)
/dev/hda1 on /datay type ext3 (rw)
/dev/hda10 on /dataz type ext3 (rw)
```

这个表的意思是说，/dev/hda5挂载到了/目录下。就是说，/dev/hda5分区的东西在/目录下得到显示。而/dev/hda5中有boot这个目录，它里面的内容是得不到显示的，因为/dev/hda2里面的内容被挂载到了这个目录里面，/boot目录里面是/dev/hda2的内容。关于mount命令，参见6.2.1。

³你可能争辩说，Windows下的分区从来没有挂载一说。请到控制面板/管理工具/计算机管理/磁盘管理中自己找答案。

3.1.5 文件系统格式

Windows用户可能知道他（她）⁴的计算机里面有NTFS格式的分区和FAT格式的分区。

Linux也有自己的文件系统格式，这就是ext3⁵，此外还有ReiserFS。对于两个文件系统的效率问题，据USTC BBS上声称ReiserFS会更快一点。此外，Linux下的虚拟内存也是一个分区，Linux swap，对应Windows下的Pagefile.sys。

3.1.6 文件系统的自检

使用一段时间Ubuntu之后，你会发现开机速度明显减慢，但是再次开机之后又恢复正常。这是因为你长期使用一个文件系统之后系统会用fsck命令（参见6.2.4）给它做一个体检。如果开机中途滚动条停止前进，请输入CTRL-ALT-F8来检查是否是文件系统出了问题。

3.1.7 磁盘整理

忘记这个概念吧。根据官方文献，Linux下对于500GB以下的分区进行整理是没有任何意义的。事实上它所用的ext文件系统是“越用越高效”的文件系统。

3.1.8 磁盘缓冲

Linux中，并不是所有关于文件复制移动修改的命令都会马上在磁盘上生效，而是暂时先放在由内存形成的磁盘缓冲中。这样，如果再对操作进行马上修改，就不必再三读写磁盘。

常启动的软件系统也会放到磁盘缓冲中，这样再次启动的时候读盘时间会大大缩短，例如OpenOffice（B）。

当然，这也带来了一些麻烦，例如插拔USB盘的时候必须先用umount（6.2.2）来将缓存写入磁盘，才能拔下，否则会造成数据丢失——可是你在Windows下也得这样。

3.2 用户管理

3.2.1 用户系统

在一个好的操作系统中，必须考虑到多个人用同一台计算机而不会相互干扰的问题（这个对于有着多个终端供很多人同时使用的超级计算机来说尤为重要），Linux提供一个良好的权限管理系统，用来管理计算机的使用人员。大体而言，Linux有一个称作“超级用户”的账号，对应root，来对系统进行完全的支配和管理工作。此外还有一些有着相对较小的特权的系统管理账号，例如bin等。这些人都属于管理员组。剩下的是普通用户，只有管理自己的目录的权限。它们是普通用户组成员。特别的，Ubuntu Linux的安装过程中会产生一个特别的账号，应当称作系统的第一个用户，它在平时的权限和其他普通用户完全相同，但是一旦通过sudo命令，输入自己的密码，就可以拥有超级用户的无上权力。这个账户也属于超级用户组。

⁴我支持两性平等，但是习惯写“他”了。要不，以后我用“它”代替“他”和“她”？

⁵ext4即将诞生。

Ubuntu系统⁶不提倡用户经常使用root账号，相反，你应该习惯使用自己的普通用户账号，仅在进行特殊操作的时候才使用su或者sudo命令，以免造成大破坏。参见6.1.1。

不同的人最好不要用相同的账号。否则如果你的MM看到了你偷偷珍藏的前MM照片，她会怎么想？反正我没想过，因为我只有前MM。

以上是一个简单的说明，真正的Linux系统的账号系统是非常复杂的。如果你有兴趣，可以借助/etc/passwd文件窥其一斑，详见8.1。

如果想了解账号管理的命令，参见6.1.4。

注意 在bash下，超级用户的命令提示符是#而普通用户的命令提示符是\$。

3.2.2 用户权限

Linux的用户权限管理是基于文件和目录系统的权限控制进行的。大抵上一个Linux文件或目录具有一个三位数的权限属性，类似Windows的隐藏文件，系统文件等属性，但是二者性质完全不同。

每个文件或目录的三位数权限属性列表如下：

分类	root			所有者 [†]			所在组		
二进制									
数值	4	2	1	4	2	1	4	2	1
代表 ^{††}	r	w	x	r	w	x	r	w	x

[†]就是创建文件或者目录的用户

^{††}其中r代表读，w代表写/修改，x代表执行。

用三个数字代表文件属性，每一位数字上边三栏中的三大类属性。数字的值是下面相应拥有的属性的数值的和。例如，一个文件只能由超级用户读写执行，而普通用户只能读和执行，那么属性对应的数字就是7(4+2+1) 5(4+1) 5(4+1) =755。

此外，为了区分文件和目录，特别的，目录拥有一个d属性，即是directory的缩写。

另外还需要提到一点，就是关于s权限，是指这个文件只能由指定的账户组来进行运行的。一般这个账户组是属于系统管理员账户组。例如sudo命令(6.1.2)就只能由系统管理员组成员运行，拥有s属性。具体的说，就是只有第一个账户可以运行，超级用户是不需要sudo这个命令的。

参考 如果希望了解更多权限的资料，请查阅相关文献，例如man chmod。

3.3 开机和关机

3.3.1 开机

从打开电源到登录

用手按住开机按钮……

说笑而已，其实我想谈的是Linux的开机系统GRUB。这是一个功能非常强大的引导系统。

开机的时候你可能看到以下内容，如果你选择的是双系统的话：

Ubuntu, kernel 2.6.17-10-generic

⁶更广泛的说，所有类Unix系统。

```

Ubuntu, kernel 2.6.17-10-generic (Recovery mode)
Ubuntu, memtest86+
Other operating systems:
Microsoft Windows XP7

```

第一个选项是Linux的正常启动选项。第二个则是单用户系统维护选项，用于系统管理人员，一般用户没有必要也不推荐使用。第三个是用来检测计算机内存的工具，我的建议是没事别玩它。第四个是你的亲爱的M\$ Windows。

如果你升级过Linux内核，选项会更多。不过无论如何，一般来说如果要进入Linux，选择第一项就可以了。

登录

启动完毕之后一般是进入图形登录界面。不过我这里要讨论的是字符界面的登录。要想退出图形界面登录，转向字符界面登录，可以在启动之后任何时候按下CTRL-ALT-F<x> ($x \in \mathcal{N}$ 且 $1 \leq x \leq 6$)键。只要你没有在那个tty（参见3.4）登录，都会显示下面界面：

```

Ubuntu 6.10 xxxxxx tty*
xxxxxx login:

```

输入你的用户名和密码，就进入了Linux会话之中。有些像某些科幻电影的效果。

注销

在命令行界面中，输入logout或者按下CTRL-D快捷键，则从当前会话中注销。这样会重新出现登录时的界面。

关机

Linux有着一些内存缓冲技术，所以关机的时候需要输入特别的命令。如果是在图形界面中，则直接从菜单中关机。字符界面下，请参阅6.3.1

3.4 多终端概念

在登录的时候有没有注意到tty这个词？它指的就是终端。一般而言，你可以一次登录字符六个终端进行同时操作，每个终端都是相对平行的工作的，在终端之间通过CTRL-ALT-F<x> ($x \in \mathcal{N}$ 且 $1 \leq x \leq 6$)切换。好比你的计算机变出六个显示器和键盘，分别以tty1到tty6标记。这是一件很方便的事情。比如你可以开一个终端用vi写程序，另一个终端用man查阅相关资料。

图形终端也可以有很多个，默认是CTRL-ALT-F7键代表标准图形终端。

3.5 命令行界面

3.5.1 铺垫一下

我们开始上无机化学之前，按以往的惯例还是要先吹一把的。

——中国科技大学化学系 张祖德教授

⁷对于我来说，M\$的Windows已经成为过时的东西。我现在正在完全使用Linux工作。

Linux的字符界面比Windows更加强大。虽然CLI不是很受欢迎（特别是Windows Users），然而，为了降低你的抵触心理，这里就介绍几个小Tip让你快乐的接受Linux。

- 输入命令的前一个或者几个字母，按一下TAB键，命令的整个形式就会出来。如果有多个选项，再按一次就会列举出来。
- 同样的技巧可以用在输入长目录和长文件名中。例如audacious_i386.deb你只需要输入auda+TAB就可以了。
- 如果你用过Windows的批处理的话，不妨和Linux的ShellCode比较一下，你就会明白批处理文件（.bat）和脚本文件（.sh）简直不可同日而语。
- Linux的字符界面里你可以听音乐（mpg123），可以上网（w3m），可以和别人用ICQ聊天。
- Linux可以让你一次输入多条命令，也可以调整命令在前台或者后台运行。

3.5.2 字母大小写

Linux是一个精密的系统，一个体现是它区分字符的大小写。当然，另一种解释是软件开发人员懒得对大小写字母进行简单的变换导致了大小写表示不同意义这个结果。

举一个例子：如果你输入ls命令，系统会列出当前目录的文件，但是你输入Ls, LS, lS的话，它八成会报错，除非你有上述三个程序——然而我认为这三个程序你全都没有。

3.5.3 多种命令行界面

命令行界面是可以“超级变！变！变！”的。可供选择的有sh, csh, bash, fish等等。我个人推荐fish，然而bash是最普及的。有关这些内容参见3.6。

3.6 Shell的概念

Shell是一个接口，把你输入的命令解释给计算机听，一个不恰当的比喻是：好比一个游戏机的手柄把你的手指按键解释给游戏机一样，你可以随便换，凭手感好就足够了。如果你是计算机界的老油条⁸的话你应该记得Norton Commander代替command.com的故事。Shell就像Norton Commander和command.com。

3.6.1 fish

全称Friendly Interactive SHell，很棒的终端，我个人推荐。如果想要获得它，输入：

```
xxxx@xxxx:/home/xxxx$aptitude install fish  
但是fish有些其他SHell有的命令不支持，需要注意。  
有关aptitude的相关资料见7.2.4
```

⁸虽然我认为计算机界的老油条不会也不屑看我的这个小册子。

3.6.2 bash

这是Ubuntu Linux的标准装备，全称Bourne-Again SHell。

3.6.3 ksh和csh

很久以前用RedHat的时候用过，现在不太记得了。分别是Korn SHell和C SHell。其中csh似乎语法很接近C语言。

3.6.4 sh

最基本的SHell。

3.6.5 busybox

这是Ubuntu Linux在系统出现毛病时（例如无法引导upstart）使用的SHell。它集成了一些简单的命令，可以用来应急处理一些故障。如果你装过WindowsXP或者2000，应该知道故障恢复控制台一说。busybox就是Ubuntu的故障恢复控制台。

3.7 多任务系统

如同Windows，Linux也是一个多任务系统。换句话说，它有一个调度机制，使得你可以在感觉不到切换的情况下，一会干干这个，一会干干那个，就像一次同时在干很多事情的样子。如果你想要了解如何在命令行进行多任务工作，参见4.10。

3.7.1 前台和后台

前台 在前台的程序接受用户操作的信号，例如键盘的指令。

后台 后台程序默默无闻的在背后干活，不接受用户的操作信号，当然，这是相对的。

例如，你在输入命令的时候，你的SHell（3.6）就在前台，而如果你输入了一个命令，例如：

```
xxxx@xxxx:/home/xxxx$cp a b
```

有一段时间你就不能再输入命令让SHell运行了，这时cp（4.3.6）在前台，SHell在后台。

3.7.2 进程标志

所有的进程都有一个数字记号，称作PID。你在操作进程的时候，对进程操作和对PID操作是同样的结果，好比给进程起一个别名。

正如《唐伯虎点秋香》中周星星的编号是9???号一样。

3.7.3 进程优先级

进程有高低尊卑之分，体现在优先级上。一般而言，单位时间内，优先级高的进程能够多运行一会。优先级用NICE表示，值在 $[+20, -20]$ 之间。越小的值代表越高的优先级。如果你想修改进程优先级，参考4.10.5

如果你不理解优先级的概念，那么这样设想：你的女（男）朋友和你的老妈（爸）同时落水，你先救哪个，拯救哪个的进程的进程优先级就相对较高。

3.7.4 进程状态

进程状态分这些内容⁹：

D 不可中断的睡眠。（往往是因为输入输出导致）

如果你对化学反应动力学有研究，那么你应该明白一个反应可以分为几个基元反应，基元反应中最慢的那一个反应就是决定反应速率反应。文件输入输出（参见3.10）往往是最慢的决定性反应，如果它不能够反应出产物来，那么后面的反应就无法进行，整个反应就处在D状态。

R 正在运行中。

S 可中断的睡眠，正在等待信号。

和110的警察一样，如果没有人报警他们就打盹，有人报警他们就活跃在打击犯罪的第一线上。

T 停止了，停下来不干了。

Z 僵尸进程，终止了，但是还没有被完全解决掉。遇到这种生化危机的东西，我们坚决要用kill命令干掉它，详见4.10.4。

3.8 图形界面

3.8.1 简单介绍

Linux不像Windows，它可以有多种图形界面，就好像终结者如果换一层皮的话，就可以乔装成另一个人，而不是我们英俊魁梧的（前？）加州州长。其中GNOME和KDE是两大主要干将，还有一个副将叫XFCE，其他例如9WM和FVWM因为不是Ubuntu的标准装备，也不如上面三位流行，这里就不说了。

一般认为GNOME比较朴素，KDE比较华丽，而XFCE属于最轻量级的图形接口。我的意思是说如果你对自己的电脑没有十足的信心的话就不要装KDE了，如果你觉得自己的机器是老古董，就用XFCE吧。不过Linux的作者Linus Trovald推荐KDE（:-)），因为它的可调节程度比较高。

所有图形接口似乎都可以支持换肤之类的功能，类似XP的更换主题。如果你觉得自己的机子够牛，可以加装Beryl套件，它的绚丽效果胜过Windows Vista。

本小册子基本上不会讨论图形界面，因为我认为图形界面只要会玩Windows的话就可以很快搞定。

⁹译自ps的man，有删改。

3.8.2 进一步讨论

所有图形界面都是基于X图形界面的，换句话说，还是终结者的比喻，机器体是X，外边的皮是KDE，GNOME或者XFCE。现在的X应该是有X11R7版本了。

现在允许直接在图形界面登录了。所以下面的命令也没有了人气：

```
xxxx@xxxx:/home/xxxx$startx
```

3.9 内码

内码问题相信你在Windows下就遇到过，例如有些网站或者邮件打开来是乱码，这时候系统无法正确识别网站或者邮件的内码，于是就显示出了乱码，你修改一下内码就可以正确显示它们了。

世界各国人民有着各种不同的文字和语言，虽然经济全球化了，然而文化可能不能全球化。为了显示不同文字，就有了内码的问题。现在国际流行的似乎是UTF-8格式的可以显示各国文字的字符编码，而中国有一套GBK的显示中文的编码。这样，如果用GBK的解码方式显示UTF-8文字就会出现乱码，正如如果你用听汉语的方法听英语，你会觉得bus和yes这两个词都有点不吉利¹⁰。

据说有一个著名军阀曾经对治下的学生说：“你们都会各个国家的英语……”

Linux的核心是ANSI（美国国家标准）的，不支持中文的输入之类。只有在外边加上多语言包才能显示各国的语言。你会使用图形界面，所以你会使用中文作为标准语言。这样在CLI下就无法正确显示文字，因为Ubuntu 6.10下没有字符模式下的中文平台（zhcon已经不能用了）。你可以输入下面的命令：

```
xxxx@xxxx:/home/xxxx$export LANG=en
```

强制在字符终端下以英文为标准语言来保证显示。

如果你是在图形界面下的终端，那么就不必这样了。

顺便说一句，如果你有什么文档在图形界面下打开了是乱码的话，调整以下内码可能能够解决这个问题。

3.10 设备-文件映射

Linux下为了方便软件开发人员，对于很多东西都是以文件的形式来保存状态的。硬件映射的文件一般存放在/dev/下。例如，你的显示器就是/dev/tty?文件，你的硬盘分区就是/dev/hda?文件。系统的状态也是以文件的形式保存的。比如系统正在运行（调度系统掌控）的程序的状态就储存在/proc/下。

这个概念了解一下就可以了。

3.11 配置文件

Linux下面没有冗杂的罗嗦的扯淡的糊涂的混乱的昏聩的繁复的注册表。如果你不认为注册表是冗杂的罗嗦的扯淡的糊涂的混乱的昏聩的繁复的，那么我可以负责的告诉你注册表就是像上边这句话一样冗杂罗嗦扯淡糊涂混乱昏聩繁

¹⁰如果我用Chinglish解释，就是“爸死”和“爷死”。

复。如果不信的话，请问你在Windows下有没有用过“优化大师”¹¹，用过的话你就会明白注册表还需要“优化”。Linux系统不需要鲁大师的优化。

在Linux下，所有程序的配置都是保存在它自己的配置文件或者配置目录之中。有些目录或者文件以“.”字母开头，从而在ls命令（4.3.1）中一般看不到它们，这些文件一般是在用户的温馨的家目录中。有些文件是在/etc/目录之中，它们一般不隐藏，因为普通用户无法修改它们。

对于一些系统核心配置文件，参见8。其他配置文件会在相应的章节解释。

¹¹我不怕打官司，我要明确的说“优化大师”有两个问题：其一，“优化”的结果往往是“劣化”，系统会越来越混乱，甚至不能使用；其二，“优化大师”的作者更过分的是，他无非就是糊弄了几本注册表的书籍，然后搞了个漂亮的Skin，就开始推销他的伟大的软件来搞人民币，作为程序员来说写出这种软件来还要那么多钞票，这简直太过分了。如果你真的要使用系统优化软件的话，我推荐Iolo公司的System Mechanics。

Chapter 4

常用命令

4.1 公共部分

4.1.1 参数

几乎所有命令都有以下几个参数：

- **-help** 简明帮助，语言不定。

- **-version** 显示命令的版本。

参数分以-开头和以--开头两种，其中以-开头的是单字母参数，如果后面没有“参数的参数”的话可以叠加，例如：

```
xxxx@xxxx:/home/xxxx$ls -lR
```

等价于

```
xxxx@xxxx:/home/xxxx$ls -l -R
```

而

```
xxxx@xxxx:/home/xxxx#mount /dev/hdc /media/cdrom -t iso9660 -o loop
```

则不能叠加成：

```
xxxx@xxxx:/home/xxxx#mount /dev/hdc /media/cdrom -t iso9660 loop
```

特别注意参数分大小写。

本小册子只会列举常用的参数，如果需要更深入的了解，请使用man命令或- help参数。

4.1.2 文件名

Linux支持长文件名，所以在文件名中有空格的时候你应该用引号包含整个路径，或者在空格前加上\符号

```
xxxx@xxxx:/home/xxxx$cd "Quantum Mechanics"
```

```
xxxx@xxxx:/home/xxxx$cd "Quantum\ Mechanics"
```

~符号代表的是你的用户所在目录¹。

*号和?号作为通配符。

Linux不屑抄袭M\$的东西。所以如果你发现某些东西和M\$ Windows的命令相近，那么请记住，这是Windows抄袭UNIX的。

¹温馨的家

4.1.3 多重文件输入

对于很多命令，之后的输入文件名可以跟多个，程序将自动依次处理这些文件。例如：

```
xxxx@xxxx:/home/xxxx$ls -l OldChina inet
```

将依次列举这两个文件的有关消息。

4.2 目录控制

本节中，除了pwd命令之外，其他命令都和Windows的命令有着对应。

4.2.1 cd

change directory的缩写，改变路径，不支持-help参数。

用法

```
cd [目录]
```

提示

.代表当前目录，..代表上一层目录²。记得如果目录太长可以先输入目录名称的前几个字母再按TAB补完。

特别要注意的是在Linux下命令cd..是不被接受的，你必须使用cd..³，这个和M\$ Windows不同。你也许会觉得这样不够方便，我也不会替Linux辩解，等到你用多了就明白这样做的好处了。

如果不加参数，那么你将到你温馨的家目录中去。

4.2.2 mkdir

很奇怪的是cd是缩写，并且只能用缩写，而mkdir，也就是make directory的缩写，不能继续用首字母缩写。

用法

```
mkdir [参数] [目录名称]
```

参数

-p 你可以通过这个命令建立一个很深的子目录，它会自动建立相应的父目录结构。见例子。

例子

```
xxxx@xxxx:/home/xxxx$mkdir -p "/home/abc/def/ghi/jkl/the corresponding directory  
will be built automatically"
```

²这个又是M\$抄袭UNIX来的

³注意空格

提示

你可以一次建立很多目录，用空格相隔开。注意如果要建立带空格的目录，需要用引号或者\符号哦！

4.2.3 rmdir

很奇怪的事情同上一节。

参数

类似mkdir。参见4.2.2

提示

只能删除空目录。

4.2.4 pwd

显示当前所在目录。不支持 `-help`。

用法

`pwd`

提示

一用便知。不过，现在的Shell往往会在命令行显示你当前所在目录，例如bash，而fish会显示你当前所在目录的所有上层目录的首字母和当前目录的全称。所以这个命令可能没用。

4.3 文件管理

4.3.1 ls

ls命令代表的是list，列举目录和文件，如同dir。注意不同属性的文件和目录的颜色是不同的。

用法

`ls [参数] [文件名]`

参数

`[文件名]` 列举这个文件名的文件，可以使用通配符*和?。⁴

-a 列举隐藏文件，即是以.开头的文件，详见3.1.1。

-d 在用通配符寻找文件的时候可能会找到相应的目录，这时ls会显示目录里的内容。当然，这些资料都很有用，然而它们并不是我们需要的东西。所以就使用-d命令表明，要的只是目录的名称。

⁴你是想说“Windows (DOS) 也有这些通配符”吗？让你失望了，这两个通配符应该是UNIX首创的，按我的印象。即便不是UNIX首创的，也绝不是M\$的作品。

-h 可以让显示文件的大小的方法更“人性化”（这是ls自己说的）。也就是说，会用K, M, G等记号让你更好的读出文件的具体大小。

-l 列举文件的权限，大小等详细资料。

-R 递归列举一个目录下的所有子目录中的文件

-X 以文件的扩展名排列，便于找出同一类的文件。

例子

下面这个例子显示我的用户目录的内容，当然，有关个人隐私的资料我都删除了：

```
xxxx@xxxx:/home/xxxx$ls -l
```

```
总用量 4842
drwxr-xr-x 2 magi magi    1024 2006-10-05 17:14 Articles
drwxr-xr-x 5 magi magi    1024 2006-11-13 22:06 Books
-rw-r--r-- 1 magi magi  466055 2006-11-08 21:17 Borrow.pdf
drwx----- 2 magi magi    1024 2006-11-12 09:44 Desktop
-rw-r--r-- 1 magi magi    796 2006-09-21 21:16 Diary
drwxr-xr-x 7 magi magi    1024 2006-08-28 11:11 Downloads
drwxr-xr-x 6 magi magi    1024 2006-11-15 19:09 Homework
-rw-r--r-- 1 magi magi   8281 2006-11-11 14:00 New Microsoft Word 文
档.odt
drwxr-xr-x 4 magi magi    1024 2006-10-23 22:58 Pictures
drwxr-xr-x 3 magi magi    1024 2006-11-14 11:55 Projects
drwxr-xr-x 8 magi magi    1024 2006-11-17 17:53 Ubuntu CLI Guide
-rw-r--r-- 1 magi magi 1265362 2006-07-06 22:16 全世界人民一定胜利.rm
-rw-r--r-- 1 magi magi   49152 2006-11-15 23:16 样品制备.doc
```

最左边一排是关于文件权限的，请参见3.2.2。第二列是文件所属的组，然后是文件的所有者，接着是文件大小（以字节计算），然后是最后修改的日期和时间，当然，最后一列是文件名，总不能少了这一列吧。

然后是一个-a参数的例子：

```
xxxx@xxxx:/home/xxxx$ls -a
```

```
.                .gtkrc-2.0
..               Homework
4_0001.ppt       .hwdb
4_0005.doc       .hxplayerrc
.4_0005.doc.autosave.odt .ICEauthority
4_0005.odt       .icons
4_0006.doc       .java
4_0008.doc       jobrequest
4-1-A.doc        .kbfx.html
.adobe           .kchmviewer
.aptitude        .kde
Articles         .kderc
```

不是完全的列表，只取了前几行。因为这个列表很长。

注意 用户的个人目录里面有大量的关于应用程序设置的目录/文件。例如.kde就是KDE桌面平台的设置文件所在目录。其他参数请自行尝试。记住，随意ls不会对系统有任何害处。

参考 强烈推荐你使用man阅读这个命令的相关文献。

4.3.2 dir

这个命令大约是用来兼容Windows用户的，我想。因为我保存了dir和ls两个命令的相应帮助，然后用diff检查它们的区别，发现唯一的区别就是这两个命令的名字不同……

4.3.3 cat

cat的作用是输出一个文件，类似DOS里面的Type。

用法

cat [参数] [文件名] ...

参数

- b 只对非空行显示行号。
- E 在每一行结束时显示\$这个字符。
- n 对所有行显示行号。

提示

你可以一次输入很多文件名称，用空格隔开。这样这些文件会被依次输出。此外，为了能够逐屏阅读输出，可以使用more命令，详见4.7.1。

例子

下面一个例子显示Ubuntu包管理中的源列表，并显示出行号。要了解Ubuntu中的包管理，详见8.3

```
xxxx@xxxx:/home/xxxx$cat -n /etc/apt/sources.list

1 #CN99 Ubuntu Apt-Source
2 deb http://ubuntu.cn99.com/ubuntu/ edgy main
  restricted universe multiverse
3 deb http://ubuntu.cn99.com/ubuntu/ edgy-updates
  main restricted universe multiverse
4 deb http://ubuntu.cn99.com/ubuntu/ edgy-security
  main restricted universe multiverse
5 deb http://ubuntu.cn99.com/ubuntu/ edgy-backports
  main restricted universe multiverse
6 deb http://ubuntu.cn99.com/ubuntu-cn/ edgy main
  restricted universe multiverse
7 deb http://ubuntu.cn99.com/ubuntu/ edgy-proposed
  main restricted universe multiverse
8
```

```
9 #USTC Ubuntu Apt-Source
10 # deb http://debian.ustc.edu.cn/ubuntu/ edgy main
restricted universe multiverse
11 # deb http://debian.ustc.edu.cn/ubuntu/ edgy-backports
restricted universe multiverse
12 # deb http://debian.ustc.edu.cn/ubuntu/ edgy-proposed
main restricted universe multiverse
13 # deb http://debian.ustc.edu.cn/ubuntu/ edgy-security
main restricted universe multiverse
14 # deb http://debian.ustc.edu.cn/ubuntu/ edgy-updates
main restricted universe multiverse
15 # deb-src http://debian.ustc.edu.cn/ubuntu/ edgy
main restricted universe multiverse
```

同样，由于这个文件很长，只截取了前一段。

注意 中国科技大学校内Ubuntu源暂时未向外公布，这个任务我就做了。希望校内Linux User Group不要找我麻烦。

4.3.4 head和tail

head依次显示文件的头十行，而tail依次显示文件的后十行。这样你就可以方便的选择你要的文件而不致于陷入垃圾信息的海洋之中。

用法

```
head [参数] [文件名] ...
tail [参数] [文件名] ...
```

参数

-c N 显示前（后）N字节的内容。

-n N 显示前（后）N行的内容。

提示

同4.3.3

4.3.5 mv

mv的意思就是move，移动一个文件。

用法

```
mv [参数] [源文件] [目的位置+文件名]
    如果不给目的文件名，则默认不改名。
```

参数

-b 如果目的文件存在（就是有重名），那么就创建一个备份。

-f 强行移动，在目的文件存在的时候覆盖这个文件。默认选项。

- i 和-f恰恰相反⁵，一旦要覆盖，先问你是否同意。
- u 更新移动，只有在源文件比目的文件最后修改日期晚或者目的文件不存在的时候才移动。

提示

目的位置可以只给一个目录，不用给出文件名了，默认将使用源文件名称。

Linux下似乎没有更改文件名的工具，那么把一个文件改名是否就可以用mv命令呢？

4.3.6 cp

cp是copy的缩写，复制的意思。

用法

cp [参数] [源文件]

参数

上边mv的所有介绍的参数对cp同样适合。

- H 复制文件时，如果碰到链接，就连同链接所指向的原始文件一起复制。
- l 并不复制文件，只是建立相应的硬链接。
- L 跟从符号链接一同复制。意思就是如果复制的时候符号链接断开（被链接的文件不在当前目录下⁶的情况下修复这个链接。
- p 复制的时候连同所有属性都复制，包括最后修改时间等。一般复制的时候目的文件的最后修改时间是复制时的时间。
- P 和-L相反，如果符号链接断开则强行复制。
- r -R 复制整个目录树
- s 只是建立相应的符号链接。关于链接参见3.1.1。

提示

比较cp和Windows中的XCopy。

例子

借助这个例子来理解链接这个概念。设我的目录下有ChemPhys.txt这个文件。现在有一个符号链接文件CP和一个硬链接CPS都针对这个文件。

现在用下面命令复制符号链接CP到test子目录下：

```
xxxx@xxxx:/home/xxxx$cp -L CP test
```

这时候到test目录下，可以看到CP这个符号链接仍然有效（通过ls的颜色指示看出）。用cat命令显示内容，也证实这一点。

删掉这个符号链接，再用下面命令复制符号链接CP到test子目录下：

⁵正合太极阴阳相生的道理，呵呵！

⁶这是我对符号链接断开的理解，如果有错误请不吝赐教！

```
xxxx@xxxx:/home/xxxx$cp -P CP test
```

因为本目录下没有ChemPhys.txt，所以这个CP是一个断开的符号链接，无法用cat命令显示它指向文件的内容。但是如果将ChemPhys.txt也复制到test子目录下，这个CP就不再无效了，而且如果是test/ChemPhys.txt文件有变动，则显示的是变动后的内容。

所以，符号链接应当是一个存有链接目的文件的相对路径（path）的一个文件，访问这个文件时自动通过相对路径访问相应文件。如果相对路径无效，则这个符号链接也无效了。

再看硬链接。如下命令：

```
xxxx@xxxx:/home/xxxx$cp -l CPS test
```

复制产生相应的硬链接。ChemPhys.txt，CPS，test/CPS代表同一个文件。

这里借助cp命令对链接进行了一下讨论，到此为止。

注意 如果你没有特别需要，不必阅读mv和cp的man文档。

4.3.7 rm

rm是remove的缩写，移去。

用法

```
rm [参数] [文件名/目录名]
```

参数

-i 每次删除的时候都予以提示。建议如果你不是很狗胆包天的话，在超级用户账号中总是使用这个并非默认开启的参数。

-r 删除整个目录，连同旗下一堆子目录：用一个GRE单词来讲就是decimate。

提示

如果你是一个敢于冒险并且愿意自己承担责任的人，试试下面的命令：

```
xxxx@xxxx:/home/xxxx$sudo rm -R /*
我不会为此承担责任的……
```

4.3.8 find

查找文件，功能非常强大，从它一共有1178行的辅助文档（80字符算一行）来看的确是这样的。

用法

```
find [参数] [文件名] [表达式] [命令]
```

参数

以下是一般参数：

-P 不跟从符号链接。

-L 跟从符号链接。

以下是可用的表达式：⁷

-name 查找文件名，含通配符*和?的文件名要用引号括起来。

-perm 000 文件属性，用3位数字显示，参见3.2.2。

-atime n n天之前访问过的文件

-mtime n n天之间修改过的文件

-ctime n 文件的状态在n前之间修改过

-newer filename 如文件的最后修改日期较filename新则为真

-user username 如文件的所有者为username则为真

-nouser 文件属主在/etc/passwd文件中不存在

-group grouname 文件组

-a 表达and算符

-o 表达or算符

! 表达not算符

各个表达式可以用\(\和\)括起来表示优先级。

执行的命令有：

-exec command 执行command，如找到相应的文件，command必须以“\;”记号结尾，此外在运行命令时将被代替为查找到的文件的路径和文件名。

-ok command 与exec相类似，但在执行每个命令之间要求用户确认

-print 打印找到的文件的名称

命令之后需要加上\;记号。

如果你不清楚and、or、not算符，请自行寻找相关书籍阅读。谭浩强的C语言教程可以做一个参考，虽然我觉得谭浩强的书不怎么样。当然，你要骂我，随你。

提示

find命令可以通过表达式来搜索相应的文件，这是Linux最高级的命令之一，就好比量子力学里面的Dirac算符一样厉害。

感谢Planck、Einstein、Schödinger、Heisenberg、Dirac、Feynmann、Fermi等人发明了量子力学，让我成功的在大学里面首次挂科。

默认将查找当前目录以及旗下所有子目录的内容，一网打尽不留余地。

⁷本段文字从<http://cpss.zz.ha.cn/bshell/UNIXcommands.htm>取得。有删改。

例子

下面的命令查找当前目录及子目录下所有名为cplusplus或者make*的文件，而文件的所有者为Panda。若有这样的文件，则显示它们的内容。

```
xxxx@xxxx:/home/xxxx$find . \ \( -name cplusplus -o -name "make*" \) -a -user Panda \) -exec cat \;
```

参考 find命令功能强大，建议阅读相应的man文件，虽然任务比较艰巨，但是和量子力学，特别是张永德教授手下的量子力学来说，还是简单了许多。我宁可翻译这个find文档，也不想再选修这样难度级别的课程了……

4.3.9 ln

ln命令就是用来建立链接的，上边我花了很大精力来啰里八嗦的和你解释链接，这里终于碰到建立链接的工具了。

我一直在想，是否要将ln命令提前，然而考虑到我花这么多精力来搞清楚链接是什么，读者必然花很大精力来做相同的理解。己所不欲，勿施于人。我建议你不必看这一段了……

如果你想详细了解一个关于链接的实验，见4.3.6里面的实验。

用法

ln有四种用法，正如“回”字有四种写法一样⁸。

```
ln [参数] [目标] [链接名]
创建指向指定目标的链接，链接名可选。
ln [参数] [目标]
在当前目录创建和目标名称一样的链接。
ln [参数] [目标] 目录
ln [参数] -t 目录 [目标]
在指定目录中分别创建指向每一个目标的链接。
```

参数

- f 强制删除任何已经存在的目标文件。
- s 创建符号链接。
- t 在指定目录创建链接。
- v 链接前先显示每个文件的名称。⁹

提示

通过链接可以使得一些较大的文件只在文件系统中保存一个，其他要求这些文件的地方用链接即可。可以节约磁盘空间。两种链接的说明见3.1.1。

很有趣的命令。

⁸鲁迅先生 孔乙己

⁹自用法第二行至此，所有文本摘自ln -help帮助文档。有删改。

4.4 压缩解压

Linux下支援（用下台湾说法）各种压缩解压软件，你可以使用Ark（KDE图形环境下的工具）进行操作。当然我们这里不谈这些，正如Cpper不谈Java一样。下面讨论如何在命令行下处理一些常见的压缩包。

4.4.1 gzip/gunzip

这两个命令似乎和zip文件格式有关，然而我想告诉你的是它使用的不是ZIP算法，而是Lempel-Ziv算法。有关这个算法的一个简单介绍，见Sahni先生写的数据结构教材。

用法

```
gzip [参数] [文件名]
gunzip [文件名]
```

参数

有关gzip的参数：

- l 显示压缩文件的内容。就是解压文件到屏幕中。
- r 压缩时包含子目录中的内容。
- v 显示正在压缩的文件的名称和压缩比等资料。
- t 检验压缩文件的完整性。
- 1 最快压缩，压缩比很低。
- 9 最好压缩，压缩比很高。但是速度很慢。

速度和效率永远是一对矛盾体。

有关gunzip的参数：没有。只要在后面加上.gz后缀的文件的文件名，这个命令将解压它。

提示

压缩过后的文件名将是原始文件名后面附加.gz后缀，对每个文件形成一个压缩包而不是形成一个包罗万象的压缩包（如果你对此感到不满，参见4.4.3）。一旦用gzip压缩，原始文件将被自动删除。同样，一旦用gunzip解压缩，压缩文件将被自动删除。

一般认为用LZ算法压缩文本效率比较高，压缩其他东西效率一般。

试试使用diff（4.5.3）命令检查gzip和gunzip两个命令，你会发现一些有趣的东西。

例子

压缩一个文件：

```
xxxx@xxxx:/home/xxxx$gzip -9v OldChina
```

解压缩：

```
xxxx@xxxx:/home/xxxx$gunzip OldChina.gz
```

4.4.2 bzip2/bunzip2

如果你仔细看了gzip/gunzip里面的提示，我想你会发现gzip和gunzip是内容相同的文件。同样的，bzip2和bunzip2也是相同的文件。bzip2和gzip的区别是：bzip is a block-sorting file compressor。看不懂的查StarDict。如果不会装StarDict，上网找教程去……

既然它们是相同的，那么用法也是一样的。不过bzip2默认压缩文件而bunzip2默认解压文件。

用法

-k gzip系列压缩 / 解压完源文件后会删掉它，bzip2同样也会，不过加上这个参数就不会了。

-t 测试压缩包。

-v 显示正在处理的文件。再加上一个v的话会显示更多细节。

-fast 最快压缩。

-best 最慢压缩。你知道我说的是压缩比¹⁰最大的意思。我的意思是压缩之后的文件最小。

提示

bzip2没有包含子目录的参数。所以你应该先用tar来把一坨文件打包之后再压缩。默认的文件后缀是.bz2。

4.4.3 tar

很久很久以前，人类只有磁带机作为备份工具。为了充分使用磁带机的功能，一位聪明的程序员发明了tar这个工具，使得很多个文件可以被合并成为一个大文件，也可以将这个大文件解开。这个默默无闻的程序员赢得了所有用户的称赞，但是我不知道有没有哪个国家的公主下嫁给他……

tar这个工具用来压缩解压.tar的文件。事实上说是压缩解压，这是不确切的。tar只是将这些文件头尾相接，中间加上一些数据指示文件的属性，以便将它们分开。tar就是这样的一个工具，好比小学教师喊口令：排队！于是小学生们排成一排，看起来整齐了，但是结果是他们的体积并没有缩小。如果你不相信，可以用tar将一些文本文件做成tar包，然后cat这个tar包看看都能看出来什么。

其实最早发明tar工具是用来备份文件的。所以tar命令中用来辅助备份文件的参数很多。当然现在的用户没有钞票来买磁带机，也没有必要了——他们都在用刻录机。但是这些备份命令还保留着，作为历史的纪念……

用法

tar [参数] [文件名]

¹⁰就是压缩后的文件大小和原始文件大小之间的比值

参数

以下为主参数，每个tar命令之后必须跟一个。

- A 将文件增加到tar包里面。
- c 新建tar包。
- d 比较tar包和文件系统里面的对应文件。
- delete 删除tar包中的内容。
- t 列举tar包中的内容。
- r 在tar包的末尾添加文件。
- u 比较tar包和文件系统中的对应文件，将修改日期比较新的文件替换tar包中的旧文件。
- x 将文件从tar包中解压。

以下为辅助参数，但是并不意味他们不重要：

- f 指定操作的tar文件名称，如果不指定的话自动到屏幕上输出——这当然不是我们所愿意的事情。
- h 不包含链接文件，而是加入它们指向的真实文件。
- j 使用bzip2压缩文件之后再加入tar包。
- k 不对已经存在的文件进行改动。
- v 显示正在处理的文件的名称。再加上一个v的话会显示更多细节。
- W 处理文件时，校验tar文件。
- z 使用gzip / gunzip处理tar文件。就是说，tar包将被压缩 / 解压。在碰到.tar.gz文件时就可以用z命令来处理。

提示

tar这个命令在Linux世界中非常流行，甚至超过了RAR。请学会使用，否则——就算了。但是你碰到的.tar.gz文件远远比.rar文件多得多。这时你得先用gunzip再用tar来对付它们。

你可能会奇怪为什么tar没有“包含子目录”这一个选项（如果你不奇怪，说明你没有好好学习），答案是：这个选项是默认的。

例子

我的家目录中有这些文件：

```
xxxx@xxxx:/home/xxxx$ls -l
```

```
-rw-r--r--  1 magi magi   8910 2006-11-22 10:35 BabyArmy
-rw-r--r--  1 magi magi   4717 2006-11-26 21:53 inet
-rw-r--r--  1 magi magi 23875 2006-11-29 21:32 OldChina
-rw-r--r--  1 magi magi   4487 2006-11-26 22:02 selfctrl
-rw-r--r--  1 magi magi    427 2006-11-16 23:42 decode
```

当然我不可能有这么点东西，不过举例子足够了。

我现在要把它们变成.tar包：

```
xxxx@xxxx:/home/xxxx$tar -cvf ABC.tar BabyArmy inet OldChina de-
code
```

当然，如果我真的只有这些东西的话，就可以用*来代替这些文件名称了，此外记得TAB键可以用来补全文件名，因此要输入的字母并不多。我得到的是：

```
BabyArmy
inet
OldChina
decode
```

tar告诉我，它已经处理好了这些东西了。我忽然发现selfctrl文件没有加上……

```
xxxx@xxxx:/home/xxxx$tar -rvf ABC.tar selfctrl
```

于是，这个tar文件中有这些内容：

```
xxxx@xxxx:/home/xxxx$tar -lf ABC.tar
```

```
BabyArmy
inet
OldChina
decode
selfctrl
```

我现在对BabyArmy有所改动……

```
xxxx@xxxx:/home/xxxx$tar -uf ABC.tar
```

于是被改动的BabyArmy文件在ABC.tar中得到了更新。

然后，因为OldChina里面批评了我们以前的伟大“副统帅”林彪同志，我害怕被当作现行反革命抓起来：

```
xxxx@xxxx:/home/xxxx$tar -delete -f ABC.tar OldChina
```

现在我用gzip压缩这个tar包：

```
xxxx@xxxx:/home/xxxx$gzip ABC.tar
```

于是我得到了一个很常见的.tar.gz包。我现在要把文件解开：

```
xxxx@xxxx:/home/xxxx$gunzip ABC.tar.gz
```

```
xxxx@xxxx:/home/xxxx$tar -xvf ABC.tar
```

都学会了吗？无论如何，本次猴戏到此结束……

4.4.4 zip/unzip

想起ZIP的作者，我就感到无比的痛苦。这位天才的程序员发明了ZIP压缩算法并且将之免费发放。然而很多年前他自杀了。我不知道他当时怎么想的。我很怀念天才。

这两个命令是用来压缩 / 解压.zip文件的工具。它们不是Linux标准配件。所以我也不会像上面那么罗嗦的讲解——虽然我很尊敬ZIP算法的发明人。

用法

```
zip [参数] [ZIP文件名] [原始文件]
unzip [参数] [ZIP文件名] [要解压的文件名称]
```


参数

zip的参数:

- u 更新文件。参见4.4.3。
- d 删除zip包中的文件。
- r 包含子目录中的文件。
- m 移动文件到包中。zip不像gzip，压缩之后会把原始文件删除。你必须指定-m才能这样。
- 1 快速压缩。
- 9 慢速压缩。你知道我的implication。
- F/-FF 努力 / 更努力的修复一个破损的zip包。
- v 显示压缩过程的细节。
- T 测试压缩包。

unzip的参数:

- l 不解压文件，只显示文件列表。
- t 测试压缩包。
- d **directory** 解压到directory目录中。
- x **file** 解压除了file以外的其他文件。

例子

我这里有一些汇编语言的作业：我从来不喜欢汇编语言。

```
xxxx@xxxx:/home/xxxx$zip -r Asm.zip Asm
```

```
adding: Asm/ (stored 0%)
adding: Asm/exam_10.htm (deflated 71%)
adding: Asm/exam_11.htm (deflated 77%)
adding: Asm/exam_12.htm (deflated 68%)
adding: Asm/exam_13.htm (deflated 69%)
adding: Asm/exam_1.html (deflated 88%)
adding: Asm/exam_2.html (deflated 80%)
adding: Asm/exam_3.htm (deflated 85%)
adding: Asm/exam_3.html (deflated 85%)
adding: Asm/exam_4.html (deflated 83%)
adding: Asm/exam_5.html (deflated 81%)
adding: Asm/exam_7.html (deflated 73%)
adding: Asm/exam_8.htm (deflated 75%)
adding: Asm/exam_9.htm (deflated 75%)
```

然后，我来解压它们:

```
xxxx@xxxx:/home/xxxx$unzip Asm.zip
```

4.4.5 rar/unrar

其实你要是见多了压缩软件，就会发现它们除了压缩比不同之外用法基本上相同。

用法

```
rar [命令] [参数] [压缩文件名称] [文件名称]
unrar [命令] [参数] [压缩文件名称] [文件名]
```

参数

rar的参数：

- a** 增加文件到包中。
- d** 删除包中指定的文件。
- e** 解压到一个目录中，不考虑目录树结构。
- m** 移动文件到压缩包中。
- r** 修复压缩包。
- t** 测试压缩包。
- u** 更新压缩包中的文件。见4.4.3中内容。
- x** 按照目录树结构解压。

提示

unrar的功能是rar的一个子集，就不讨论了。

4.4.6 7z

据说是一个很厉害的压缩软件，我也这么认为，它整体表现（除了名声）是最好的。

用法

```
7z [命令] [参数] [压缩文件名称] [文件名称]
```

参数

命令是不能缺少的，有且仅有一个：

- a** 将文件添加到压缩包中。
- d** 删除压缩包中的文件。
- e** 解压缩，不考虑包中保存的文件目录，简单的说就是把所有文件解压到同一个目录中。
- l** 列举压缩包中的文件列表。

t 测试压缩包的完整性。

u 更新压缩包中的文件，参见4.4.3中-u参数。

x 连同目录结构一起解压缩。

参数则是可以变化的：

-x 表明排除某些文件不予操作。

-l 不跟随链接，而是保存真实文件。

-o 指定输出文件的目录。

提示

默认的是包含所有子目录。不要加上-r参数——这是软件作者特地提醒的。

4.4.7 其他压缩工具

不觉得很无聊吗？做个列表吧，当年咱上小学的时候还拿ARJ当个宝呢，现在知道的人有几个！

无聊的原因是这些工具用法都是一样的，正如女朋友撒娇的方式 不管叫什么，什么性格 都差不多，不就是一哭二闹三上吊嘛。

- arj
- lha
- zoo
- lzx

4.4.8 压缩软件的比拼

这是一件很无聊的事。不过为了某些人的特殊爱好，我在此郑重宣布：总体而言最好的压缩软件是7zip。不过用的人不多。

4.5 文件比较

4.5.1 cmp

比较两个文件，按字节比较。这个工具对一般用户作用也许不大。

也许不大的意思就是也许大，这是简答的形式逻辑推论。如果有兴趣请参阅北京大学哲学系吴家国先生写的普通逻辑原理，金岳霖先生的就算了。

用法

cmp [参数] [文件1] [文件2] [-i [跳过1] [跳过2]]

参数

-b 输出不同的字节。

-i 跳过两个文件开头的几个字节，由[跳过1]决定。如果有[跳过2]存在，那么就对第一个文件跳过[跳过1]字节，对第二个文件跳过[跳过2]字节

-l 将所有的不同都输出。

-n bytes 只比较前bytes那么多的内容，按字节计算。

提示

默认状态下，只要碰到有不同，就停止比较。

例子

下面的例子是胡扯淡的比较：

```
xxxx@xxxx:/home/xxxx$cmp -l -n100 /bin/ls /bin/rmdir
```

```
26 233 212
33 320 154
34 53 62
35 1 0
45 10 7
49 33 32
51 32 31
69 0 340
70 1 0
73 0 340
74 1 0
89 64 24
93 64 24
97 64 24
```

得到的结果中，第一列是有区别的地方的位置，按照从比较的位置开头开始的顺序计算字节数而定。第二列是第一个文件的那个位置的字节，第三列是第二个文件的那个位置的字节

4.5.2 comm

本节摘自comm - -help的内容，有删改：

按行比较已排过序的[文件1]和[文件2]。，如果不附带选项，程序会生成三列输出。第一列包含[文件1]特有的行文，第二列包含[文件2]特有的行文，而第三列包含两个文件共有的行文。

用法

```
comm [参数] [文件1] [文件2]
```

参数

- 1 不输出[文件1]特有的行文。
- 2 不输出[文件2]特有的行文。
- 3 不输出两个文件共有的行文。

4.5.3 diff

比较两个文本文件的大小。这个似乎在比较程序结果的时候很有用的。

但是我业余写了五年多程序，还是从来没有用过。

用法

comm [参数] [文件1] [文件2]

参数

- i 忽略大小写的不同。
- E 忽略TAB制表符导致的不同。
- b 忽略大于一个的连续空格导致的不同。
- w 比较的时候，空格的统统不要的干活。
- d 尽量找相同的地方，寻找一个最小的区别的集合。就是说，尽量找两个文件的交集。
- r 比较两个目录，并且连子目录中的文件都比较。特别适合于比较不同版本的源代码。

例子

一用便知，不再赘述。

4.6 重定向符号及其应用

4.6.1 重定向符号

在这里介绍一下重定向符号吧。在Linux中>, >>, <, |几个符号具有特殊意义。

>, >>代表将一个命令输出的一部分字符¹¹写入一个文件，例如

```
xxxx@xxxx:/home/xxxx$ls>dirlist
```

就是将一个目录的东西输出到dirlist这个文件中。

而<命令代表将一个文件的内容作为一个命令的输入。|称作管道符号¹² (pipe)，代表将一个命令的输出当做另一个命令的输入进行处理。一个常见的例子是这样的：

```
xxxx@xxxx:/home/xxxx$ls -l | grep aptitude
```

¹¹事实上，是输出到stdio的字符，对于stderr之类的则不算。这个学过C的人应该知道我说的是什么意思。当然，如果你没学过，那么就记住我的话的意思是，并不是所有输出字符都会被输出

¹²它的确像一个管道

4.7 管道符号|之后的常用命令

希望有高人能够协助补充这一节的内容。

4.7.1 more

如果输出满一页的话，可以让你在按一个键之前看完这一页再继续下一页。

用法

```
command|more
```

4.7.2 less

似乎和more差不多，但是试试看它是否有着more than more的功能。

事实上less可以当做一个简单的文件浏览器，代替cat的作用。而且它的键位和vim非常接近。

4.7.3 grep

这个命令本来该好好讲的，但是我找到了一篇比我写的更好的讲义，见A。

4.8 Shell增强命令

以下的命令都在sh、bash、fish下测试过。

4.8.1 history

history命令正如其名，显示你输入命令的历史。fish不支持此命令。

你可以用这个命令检查你的男朋友是否偷偷动过你的计算机，如果他使用的是终端的话。

用法

```
history
```

4.9 辅助命令

4.9.1 whereis

同样是搜索命令，whereis仅仅搜索系统命令，系统命令的man文档以及系统命令的源代码。

用法

```
whereis [参数] [命令]
```

参数

-b 只搜索可执行文件。

-m 只搜索命令的man文档。

-s 只搜索命令的源代码。

提示

你可以通过这个命令了解系统命令的分布，当然，前提是你想要了解系统命令的分布。

例子

下面的命令搜索ls命令的位置：

```
xxxx@xxxx:/home/xxxx$whereis -b ls
```

```
ls: /bin/ls
```

4.9.2 which

有了whereis，which就没有什么意思了，不说也罢。

用法

which [命令名]

4.9.3 who

显示有多少个用户在使用这个系统。这个命令对一般用户没有什么用处。所以只是提一下。

用法

who

4.9.4 man

man好比Linux大百科全书——当然，这本书我还没听说过有。你对Linux的命令有什么疑问都可以查阅它。美中不足的是，它似乎是英文的。当然，听说有中文版问世，听说而已，没有亲见¹³。

用法

man [命令]

例子

我想学习一下ls的使用，怎么办捏？

```
xxxx@xxxx:/home/xxxx$man ls
```

事实上我就是用上面这个命令来写我的ls部分的。所以我再次重复我在前言里面说的话，man是王道，其他都是扯淡。

¹³鲁迅先生 论雷峰塔的倒掉

4.9.5 fortune

看名字似乎是一个很有意思的命令，那你何不自己运行一下呢？

用法

fortune

4.10 程序运行控制

4.10.1 &符号

在命令后面加上&符号，表示这个程序在后台（3.7）执行。

例子

下面演示一个后台复制：

```
xxxx@xxxx:/home/xxxx$ cp xxxx.xx xxxx.xx&
在复制的过程中，你不必等待，可以干点别的什么。
```

4.10.2 CTRL-C和CTRL-Z

CTRL-C 中断当前程序的运行，回到Shell之中

CTRL-Z 暂停当前程序的运行，回到Shell之中

4.10.3 ps

不要误会，这和Sony公司的PlayStation没有任何关系。ps应该是processes的缩写。列出所有进程的列表。

用法

ps [参数]

参数

-A 代表显示所有进程。

-F 完整输出。

-H 代表显示系统进程并且显示是那些程序运行了那些程序。

-eo pattern 代表显示pattern中指定的输出，pattern里面是一些记号，标记出你感兴趣的资料，每一列用逗号隔开。

下面是pattern的一些可用参数：

%cpu CPU占用率，数值越大的话拖系统后腿的可能性越大。列首用“%CPU”标志。

%mem 内存占用率，同样的数值越大的话拖系统后腿的可能性越大，因为抢占系统资源比较多。列首用“%MEM”标志。

就好比一个家里谁管钱的话，就最能折腾家里。所以女性同胞请注意，不给老公足够的零花钱是让老公给家里丢面子的重要表现。

egroup 启动进程的账户有效所在组。列首用“EGROUP”标志。

euser 启动进程的账户。列首用“EUSER”标志。

args 显示进程名称的同时显示启动进程时给进程赋予的参数。列首用“COMMAND”标志。

pid 显示进程的PID。参见3.7。

comm 进程的原始文件名称。列首用“COMMAND”标志。

nice 进程的优先级，越小的话优先级越高。列首用“NI”标志。参见3.7。

tty 进程在哪个终端运行，参见3.4。列首用“TTY”标志。

s 进程的状态。参见3.7。

提示

例子

下面命令列举当前用户的所有进程：

```
xxxx@xxxx:/home/xxxx$ps
```

PID	TTY	TIME	CMD
8342	pts/4	00:00:00	bash
8359	pts/4	00:00:00	sh
8859	pts/4	00:00:00	ps

PID和TTY的概念参见3.7和3.4。然后下一个TIME代表CPU时间，表明这个进程总共CPU花了多少时间运行它。最后一个是进程的命令名称。

下面命令显示所有进程的PID，进程名、CPU占有率和状态：

```
xxxx@xxxx:/home/xxxx$ps -eo pid,comm,%cpu,s
```

PID	COMMAND	%CPU	S
.....			
9487	konsole	0.0	R
9489	bash	0.0	S
9531	bash	0.0	S
10077	getty	0.0	S
10772	kdesud	0.0	S
10787	gconfd-2	0.0	S
10791	gvim	0.0	S
11348	gvim	0.0	S
12015	kio_file	0.0	S
12499	ksnapshot	0.0	S
13656	dhclient3	0.0	S
13723	kile	2.0	S
13759	bash	0.0	S
13976	syslogd	0.0	S

```
14510 man          0.0 S
14518 pager        0.0 S
14649 stardict     0.4 S
14652 bonobo-activati 0.0 S
14673 fish         0.2 R
14676 fishd       0.0 S
15091 ps            0.0 R
```

下面的命令查找僵尸进程，并且演示管道符号（4.6.1）：

```
xxxx@xxxx:/home/xxxx$ps -eo pid,comm,s | grep Z
```

参考 ps的功能十分强大，建议阅读man文档。特别的，ps有GNU，BSD，Unix三种方式。这里只介绍Unix方式的参数。

4.10.4 kill

很残暴的一个命令，谋杀一个进程。如果你不知道这个进程是什么，我的建议是别碰它，如果你用超级用户权限野蛮杀戮，你可能干掉系统的膾炙大穴，于是系统完蛋了，你后悔不已终于出家为僧/尼。注意kill不支持-`help`和-`version`。

用法

```
kill [进程PID]
```

参数

无

提示

可以用它来干掉“麻木的”，就是没有反应的进程。新建一个终端并且用ps和kill达到你的目的。当然，一般用户只能管好自己的事，只有超级用户可以多管闲事。

其实kill命令的功能远比提到的内容强大，但是对于一般用户只要知道这些就足够了。

4.10.5 nice

如果你的系统反应迟缓，那很有可能是你的系统的后台程序在不知道鼓捣什么。为了保证前台你和MM语音聊QQ聊的痛快，后台正在计算的波函数就只好靠边站了。这时你就用上了nice命令调整它的优先级，参见3.7

用法

```
nice [优先级] [命令]
```

例子

以下这个命令就是启动QQ和波函数程序，并且让QQ有很高优先级，计算波函数程序有很低优先级的一个例子：

```
xxxx@xxxx:/home/xxxx$nice -10 eva  
xxxx@xxxx:/home/xxxx$nice 10 wavefunction
```

有关eva见B。

Chapter 5

编辑器

工欲善其事，必先利其器

本章介绍Linux下的两个编辑器：nano和vim。二者之间的关系对于CS的玩家来说比较好理解，一个属于手枪范畴，一个属于冲锋枪 / 散弹枪 / 机枪范畴。虽然你最常用的是主武器，但是手枪并非毫无用处，有时候还是必须的。

5.1 一些古怪的概念

Linux下面的的编辑器里面有一些概念是很诡异的。这里需要澄清一下，免得引起混乱。你也许可能认为这个东西在Windows里面叫这个，在Linux里面怎么叫那个了。这个问题是一个伪问题。

如果你阅读过《庄子》外篇中《天下》一文，你应该记得惠施十辩，再假设你忘记了外篇作者佚名同志的浓厚的讽刺色彩，你应该明白，东西叫什么是人为规定的，牙刷之所以叫牙刷是因为大家都这么叫。编辑区域之所以叫编辑区域是因为MS\$的人都这么叫它... ..

缓冲区（**Buffer**）就是存放东西的地方，例如存放你所编辑的文件的地方是文件缓冲区，存放你所输入命令的地方是命令缓冲区.....

注意Buffer和Cache这两个词都是指“缓冲区”，然而含义并不相同。

命令 Linux的许多编辑器不带菜单，你需要输入快捷命令来进行相应的操作。

就想到了这些.....

5.2 nano

nano是一个轻量级的编辑器，小巧玲珑而又不失优雅。下面来介绍一下它的基本使用方法——注意，意思是说有些东西不会写，如果你想知道，自己看帮助吧。¹

¹以下文字部分翻译于nano的帮助文档。

注意 nano的英文意义是“纳米”，大约就是说这是一个很小的编辑器。但是它的功能却不是随体积的减小而降低。

5.2.1 系统结构

nano分三个部分，第一行属于状态栏，显示nano的版本和正在编辑的文件（对于新文件，显示New Buffer），如果文件被修改，则在右边显示Modified。

中间部分是编辑部分，显示你正在编辑的缓冲区。最下面一行是文件的属性，显示你光标所在方位等数据。最下面两行是命令提示，提示你可用的常见命令，这个是可以屏蔽的。参见5.2.2部分。

懂得一点洋文的都不用看这些内容吧，不过你已经看了……对不起。

5.2.2 使用方法

移动

方向键 光标逐个行 / 列的移动。

CTRL+F 前进一个字母。

CTRL+B 后退一个字母。

CTRL+P 上移一行。

CTRL+N 下移一行。

我不明白的是，有光标键的存在，又不是vim类型的三态编辑器（5.3.2），有必要定义CTRL组合代替光标吗？

CTRL+Space 前进一个单词²。

ALT+Space 后退一个单词。

CTRL+A / Home 移动到一行的起始位置。

CTRL+E / End 移动到一行的末尾。

CTRL-Y / F7 移动到上一页。

CTRL-V / F8 移动到下一页³。

ALT-9 移动到一段的起始位置。

ALT-0 移动到一段的结束位置。

ALT- 移动到文件的开始位置。

ALT-/ 移动到文件的最后位置。

ALT-] 移动到相符的括号中。（这个在编程的时候很有用。）

ALT-G 转到指定的行列。

²如果在图形终端下前进一个单词，那么很可能起作用的是中文输入法的变化。

³注意“上一页”，“下一页”在帮助文件中显示的是“previous screen”和“next screen”。

保存文件

CTRL-O / F3 保存文件。

CTRL-X / F2 退出nano（会提问是否保存文件）。

搜索

CTRL-W 搜寻一个字符串，在屏幕下方出现搜索的功能条，输入你要搜索的内容即可（允许使用正则表达式。要了解正则表达式，请参见A）。

ALT-W 继续上一次的搜索。

ALT-R 替换。

块操作

ALT-A 开始选定区域，选定区域的状态将在按下下列命令之后结束，或者你也可以再次按下ALT-A取消选定区域。

CTRL-K / F9 从光标开始剪切当前行，保存到剪切缓冲区中。如果有选定区域，则剪切选定区域的内容。

CTRL-T 从当前位置剪切到文件最后，保存到剪切缓冲区中。

ALT-6 从光标开始复制当前行到剪切缓冲区中。如果有选定区域，则剪切选定区域内容。

CTRL-U / F10 从剪切缓冲区中粘贴当前行。

ALT-} 增加选定区域的缩进。

ALT-{ 减少选定区域的缩进。

开关操作

这些开关也可以通过/etc/nanorc的编辑（5.2.3）来进行设置。

ALT-X 屏幕最下方的帮助信息显示开关。新手保留，老油条扔掉——老油条不会看我的文档……

ALT-O 屏幕第二行是否用于编辑的开关。

???真不明白这个是什么东西。nano里面有很多诡异的开关。

ALT-S 平滑卷动开关。

ALT-Y 语法颜色显示开关，编程用。见5.2.3。

ALT-H 灵巧Home键开关。按我的理解，打开这个开关可以让按下Home键之后光标定位到一行的第一个非空格上。

ALT-I 自动缩进开关，编程的时候可以让你少花精力来调整缩进。

ALT-B 指定是否备份文件，在保存之前备份原始文件。如果你在修改系统核心配置程序（8）文件时，建议你还是备份一下吧。

ALT-M 鼠标操作开关。

多文件编辑

nano可以一次编辑多个文件，对于这种情况，只需要在nano命令后面跟随多个文件名，nano会同时打开它们。这个叫做“多个缓冲区”。例如：

```
xxxx@xxxx:/home/xxxx$ nano file1 file2 file3
```

在多个缓冲区之间切换的命令如下：

ALT-> 打开下一个文件（缓冲区）。

ALT-< 打开上一个文件（缓冲区）。

其他的操作

下面列举一些看似“不重要”的操作：

CTRL-G / F1 获得帮助⁴。

ALT-D 清算字数。这个对于小学生写作文来说很有帮助。

CTRL-L 重新刷新屏幕。如果屏幕混乱就按下这个——我还没碰到要这样的情况。

CTRL-J 调整本段文档。在调整之后可以按下CTRL-U取消调整。这时CTRL-U原来的用法失效。

ALT-J 调整全部文档。在调整之后可以按下CTRL-U取消调整。这时CTRL-U原来的用法失效。

CTRL-R 插入一个新文件。参见5.2.3中关于multibuffer的内容。

5.2.3 设置文件

在root权限下输入下面的命令来设置nano的配置文件：

```
xxxx@xxxx:/home/xxxx# nano /etc/nanorc
```

你可以看到一些洋文的说明。我知道有的人看不懂所以我解释一下。看得懂请跳过本节，否则浪费您的时间本人概不负责。

nanorc中带#开头的文段是注释行，意思就是向你解释一些东西，通常是下面的文段的内容。所有注释行对于nano来说是没有任何意义的（对于你则不同）。其他的都是一些解释脚本，用于向nano指示一些语句。主要包含有两种（至少官方版本的nanorc里面只有这两种）。

set 设置某些开关。如果加上了，表明参数打开。

unset 关闭某些开关。

include 和C / C++的include差不多，导入一些文件的内容。官方版本nano会导入一些语法加亮信息，用于软件开发人员方便阅读源代码。

⁴还记得我说过话吗？什么教程都是扯淡（including *this），帮助文档才是王道。

set/unset

对于set / unset中可选内容，这里选择一些常用的解释一下（常用与否我说了算）：

autoindent 自动缩进。

backup 是否备份文件。

casesensitive 搜索时是否分大小写。

morespace 设置屏幕第二行是否用于编辑。

multibuffer 设置是否打开多重缓冲区。如果打开，则插入文件时将把文件放在新的缓冲区之中。

nohelp 开始的时候不显示最下方两行帮助。

smarthome 设置是否打开灵巧Home键。参见5.2.2。

tabsize 设置按下Tab键跳格数。

对于这些内容，只需要在文件中加入set [内容]或者unset [内容]即可。

include

include命令在官方nano中是用来引入编程时的语法加亮功能的。这些语法加亮文件保存在/usr/share/nano中。如果你要拿nano写程序（虽然我并不推荐，毕竟我们有Emacs和我最favorite的vim可以使用），可以考虑用include “文件名”打开。

注意 双引号是必须的。

5.3 vim

我为什么要讨论一个被很多顶尖高手讨论的昏天黑地气壮山河的工具呢？vim的强大功能已经不是三两个人赞扬过的了。而我如果写一个教程，似乎非常有可能是班门弄斧的事情。

5.3.1 著名教程

如果希望了解vim，请参阅<http://www.truth.sk/vim/vimbook-OPL.pdf>，这是vim的官方使用教材。

如果你希望快速掌握基本的编辑技能，可以在终端输入：

```
xxxx@xxxx:/home/xxxx$vimtutor
```

如果你看到的是乱码，记得3.9里面谈到的技巧。

此外，vim的帮助文件已经完全被翻译成为了中文。你可以到www处下载到这些帮助文档。想要安装的（当然你必须安装）话，解压（参见4.4部分）它们到一个指定的目录，并且运行下列命令：

```
xxxx@xxxx:/home/xxxx$./vimcdoc.sh -i
```

即可。这时这些帮助文档保存在~/vim/doc中。

5.3.2 基本概念

5.3.3 基本用法

5.3.4 进一步的应用

5.3.5 进二步的应用

5.3.6 进三步的应用

等有空再说。

Chapter 6

管理命令

肉食者鄙
未能远谋
——左传

本章介绍基本上只有超级用户才能使用的一些命令。当然，你如果是系统建立的第一个用户，那么你也可以用sudo命令执行它们。

6.1 账号控制

6.1.1 su

su是super的意思，代表超级用户。事实上用它你可以改变自己登录的账号。例如你也可以用su命令从root变成普通用户。

用法

su [参数][用户名称]

参数

-C command 登入之后运行命令command。

-s SHell 选择登入之后用的SHell。

提示

如果不给出用户名称，则默认的是root。回车之后会要求你输入密码。之后你的用户登录就会改变——前提是你的密码是正确的。如果你是超级用户，你可以不用密码就变成任何用户——意思就是，掌门永远可以收拾任何门内弟子。

6.1.2 sudo

参照上面的内容，似乎sudo应该是super do的意思。如果你这么想，我会认为你有一定比较基础的类比能力——我的意思是你想的对。sudo是所谓“第一个普通用户”用来升级成超级用户的工具。不过效力只局限在一个命令之内。

用法

`sudo [命令]`

提示

回车之后有时会让你输入密码，一般你输入过一次之后一段时间不会再让你输入。此外，你要输入的密码不是超级用户的密码，而是你的密码。

6.1.3 passwd

`passwd`是password的缩写，我不明白的是：两个字母，有必要缩写吗？普通用户可以更改自己的密码，超级用户可以更改任何人的密码。

用法

`passwd [用户名]`

提示

你必须先输入正确的密码才能更改密码。

6.1.4 useradd

增加一个用户。只有超级用户可以使用这个命令。

用法

`useradd [参数] [用户名]`

参数

-d directory 选择用户的温馨的家，记得不能给别人空头支票，请建好家之后再让别人进去。

-m directory 既然没有准备温馨的家，那就现搞一个出来。

-s SHell 确认用户使用的标准Shell。请用whereis找到bash，fish或者sh的完整路径填入。话不能只说个结尾吧？

回车之后有时会让你输入密码，一般你输入过一次之后一段时间不会再让你输入。

例子

建议用这个作为标准的新建用户命令，其中Panda是用户名称：

```
xxxx@xxxx:/home/xxxx#useradd -m -s/bin/bash Panda
```

6.1.5 userdel

干掉一个用户。同样的，只有超级用户可以使用这个命令。

用法

userdel [参数] [用户名]

参数

-f 土匪行径，即便他/她家有别人的东西，也要一起干掉。

-r 把这个用户在/home下的目录一并干掉，让他从系统中蒸发。

提示

三思而后行。

建议不要因为miff而删除你女/男友的用户。如果有必要，建议不要加上-r参数。

6.1.6 chmod

chmod是Linux中很罗嗦的命令之一，和权限有关的东西都很罗嗦。见3.2。普通用户只能管好自己，只有无所不能的超级用户才可以横扫千军。

用法

chmod [参数] [文件名]

参数

参数是三位的权限数字，见3.2.2。

-r 连同子目录中的文件一起设定权限。

例子

```
xxxx@xxxx:/home/xxxx#chmod 755 a.out
```

意思就是将a.out设置成root $7 = 4 + 2 + 1 = r + w + x$ ，所在组和所有者 $5 = 4 + 1 = r + x$ 。

6.2 文件系统命令

本节中提到的命令除非特别指出，只有超级用户有权执行。

6.2.1 mount

mount似乎是一个很伟大的程序。

用法

mount [文件系统记号] [挂载目录] [-t 文件系统格式] [-o 附加参数] [-a]

参数

[文件系统记号] Linux用来指示分区的一组记号。见前述3.1。

[挂载目录] 你要把这个分区挂载在的地方。一般放在/media的相应目录下。这个目录必须事先存在。

-t 文件系统格式 指明这个分区的文件系统格式。一般不必要，mount会自动识别。

-o 附加参数 指明一些其他参数。例如挂载光驱时必须加上loop参数。

-a 挂载/etc/fstab上已经存在的文件系统。具体挂载什么，请自行查阅/etc/fstab文件。参见8.2

-o 后面的参数

loop 挂载光驱和ISO文件需要这种命令。作用未知:-)。

iocharset=set 有时候挂接的文件系统显示文件名不正常，不用我说，内码问题，参见3.9

提示

对于ISO虚拟光驱文件同样可以用mount挂载。只需将/dev/xxx?改成文件名就行了，甚至可以使用通配符。注意要加上-o loop指示。

如果挂载目录里面本来有文件，则它们将不能被访问，除非解除挂载。

对于NTFS格式文件，建议使用ntfs-3g命令挂载。见6.2.3

例子

挂载光驱：

```
xxxx@xxxx:/home/xxxx#mount /dev/hdc /cdrom -o loop
```

挂载硬盘：

```
xxxx@xxxx:/home/xxxx#mount /dev/hda3 /media/hda3
```

不加参数的mount命令显示挂载的文件系统的情况。见3.1

6.2.2 umount

有挂载就有卸载。umount正是这个卸载命令。

必须提到的是，U盘，移动硬盘等在mount之后要拔下必须使用umount，以便将还在内存中的缓冲区数据（3.1）写入。

用法

```
umount [挂载目录] [-a]
```

参数

[挂载目录] 已经挂载文件系统的目录。

-a 卸载所有已经挂载的文件系统（当心！）。

6.2.3 ntfs-3g

这是专门为了操作Windows NTFS文件系统写的挂载程序。

用法

ntfs-3g [文件系统记号] [挂载目录]

参数

见6.2.1

提示

用mount挂载时不能解决写NTFS分区问题，这时ntfs-3g就派上了用场。

如果你的系统里面没有ntfs-3g这个命令，试试：

```
xxxx@xxxx:/home/xxxx# aptitude install ntfs-3g
```

关于aptitude命令参见7.2.4。

6.2.4 fsck

这是用来检查文件系统的命令。似乎是除了WindowsNT/2000/XP的NTFS文件格式不能检查之外其他文件系统（不管是不是Linux的）都可以检查。

用法

fsck [/dev/xxx?]

参数

-p 自动修复

提示

没事别玩它。如果你真的发现文件系统出现什么故障，试试看。

我小时候经常对自己家的东西乱用fsck命令，多次导致了家庭经济的重大损失。所以我建议你不要乱用fsck，除非真的毛病多多。

6.2.5 df

你可以猜测以下df是什么的缩写，d当然是磁盘的意思，disk，f是什么呢？free。所以df命令就是diskfree的缩写，显示磁盘剩余空间的工具。这个命令每个人都可以执行，关心国家大事嘛。

用法

df [参数]

参数

-h 用G, M, K等记号方便阅读。毕竟长长的数字比较难记, *e*总比*ln2*好记, *ln2*总比2.7182818...好记。

-T 显示文件系统的格式。

提示

常看看你的文件系统还剩下多少空间, 如果你硬盘给Linux分的不多的话。我自己当然40G全给Linux了。

有一句话叫做“对敌人的同情就是对自己的残忍”。我的理解就是“给Windows多空间就是摧残Linux的发展。”

6.2.6 cfdisk

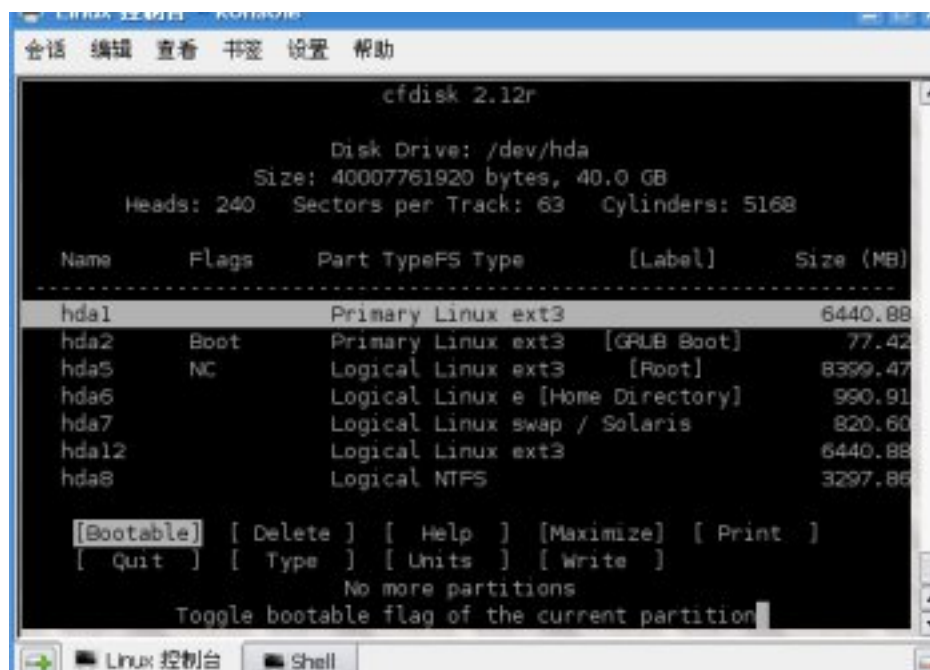
这是一个强力的磁盘分区程序。对于强力的工具, 需要强力的介绍。只有超级用户可以使用。如下图:

注意 仅供高级用户或者不怕自己计算机出毛病甚至系统崩溃的人使用。

用法

cfdisk

图解



从上到下介绍：

第一行（`cfdisk 2.12r`）是软件名称和版本的介绍。

第三行（`Disk Drive: /dev/hda`）是现在正在操作的硬盘的指示。

第四行（`Size: 40007761920 bytes, 40.0 GB`）是对磁盘大小的说明

第五行是一些硬件数据，一般用户不必了解。

下面的列表是磁盘分区表。表格格式第一列是分区列表，按照磁盘上真实分区顺序排列，由此也可以证实以前说过的，Linux的hda?顺序和真实分区顺序有出入。第二列是分区的特征。第三列是分区类型，分主分区和逻辑分区两种。主分区可以引导，最多只有四个，逻辑分区不能引导，可以有很多个¹。第四列是分区类型²。第五列是分区结构。第六列是分区的卷标，就是分区的名字。最后一列是分区的大小。

下方命令依次为：

Bootable 设置分区是否能够启动。一般在Windows分区和/boot目录所挂载分区设置为可以启动。

Delete 删除选中的分区（小心！）。

Help 帮助文件，显示命令列表。

Maximize 最大化分区使用。³

Print 打印分区表到屏幕或者文件中。

Quit 退出，不保存分区表。

Type 修改分区文件系统类型。

Units 改变分区大小显示模式，建议如果不是高手不要改动，以免看不懂。

Write 将分区表写入磁盘，这样你干的坏事就得逞了。

你可以通过按上下箭头来选择一个分区，左右键来选择下面的命令。
键盘上的命令：

b 同Bootable选项。

d 同Delete选项。

g 软件作者说叫你别乱按。

h 同Help选项。

m 同Maximize。⁴

n 新建分区。

p 同Print选项。

q 不保存分区表，退出程序。

t 同Type选项。

¹现在有了GRUB，逻辑分区也可以引导了

²你可能怀疑我是否真实的仅仅使用Linux，因为你注意到了hda8是NTFS分区，我的答案是，那个分区是有意保留下来，保存了我喜欢的一些游戏。

³我不明其意。求高人指点。

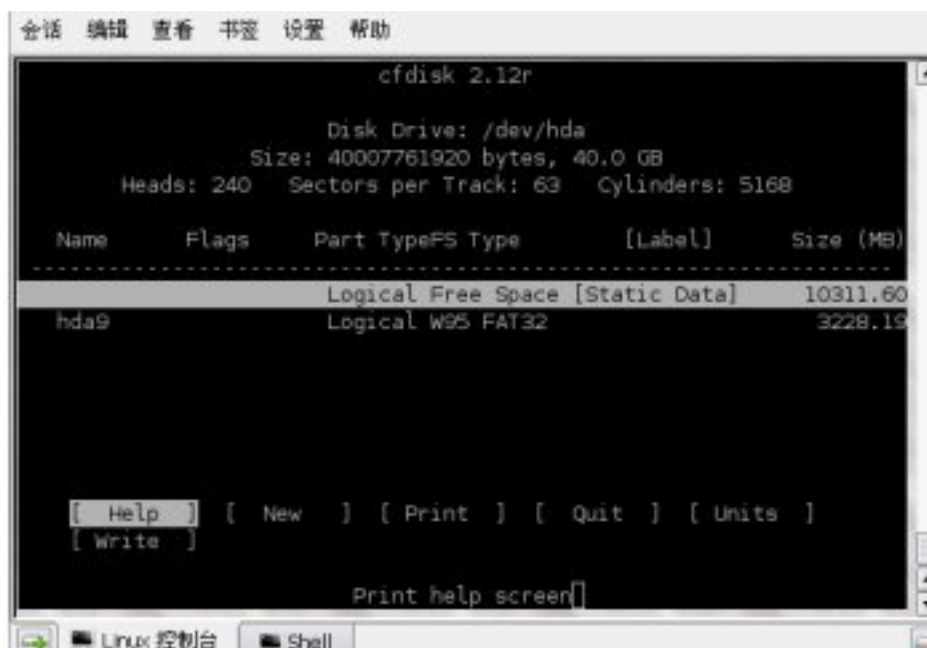
⁴我同样不明其意。

u 同Units选项。

W 同Write选项。

软件作者提醒你注意，所有其他选项都可以大小写任意，只有W选项必须大写输入。这是为了防止你误按此键，顿入歧途。

按d键删除一些分区，则出现下面的变化：被删除的分区前两列小时，第四列显示Free Space，如图：



新建一个分区，此时光标必须移动到有Free Space字样的地方，否则不能新建分区。好比你不能在已经建好楼房的地方再建楼，否则就是违章建筑，是和党和人民过不去的行为。在这时，下方的命令变为一下几个内容：

Help 同原来Help。

New 新建分区。

Print 同原来Print。

Quit 同原来Quit。

Write 同原来Write。

这时系统提示输入分区大小，用MB为单位。你可以参考光标最右边的空闲空间大小来新建分区。选定分区大小之后，系统会提示三个选项，用左右光标选择：

Beginning 在空闲空间开始处建立分区。

End 在空闲空间结束处建立分区。

Cancel 不建了，走人。

选定一项后，就可以达到你的罪恶目的。注意实际分区大小和输入的大小会有一些出入。

建好分区之后，按t键，系统会显示一些分区的格式代号。对于Linux用户，当然使用Linux的分区，选择83。如果你需要交换分区的话，选择82。

建立或删除分区操作结束之后，选择Write完成任务，或者选Quit收队不干。

总之，如果你在Windows下了解基本的分区知识，在这里分区是比较简单的一件事情。注意，这里的分区不会附带格式化功能，要格式化相应分区，请用mkfs命令。参见6.2.7。

6.2.7 mkfs

建立一个指定的文件系统。只有超级用户可以使用。

注意 小孩子不要乱玩。

用法

```
mkfs [-t fstype] [-L label] [分区映像文件]
```

参数

-t fstype fstype为文件系统的类型：

ext3 Linux的标准文件系统格式。也是默认格式。

reiserfs 另一种似乎更快的文件格式，不是很推荐。

-L label 分区卷标，就是它的名字。

提示

小心使用，这个和Windows里面的Format类似。

例子

下面的例子在/dev/hda1建立一个名为ROOT的ext3文件系统：

```
xxxx@xxxx:/home/xxxx$mkfs -text3 -LROOT /dev/hda1
```

6.3 其他命令

6.3.1 shutdown

关机命令。只有超级用户可以使用。

用法

```
shutdown [参数] [时间]
```

参数

-r 重新启动

-h 关机

-n 快速关机（不推荐）

[时间] 决定等待多久再关闭，如果要马上关闭，试试now。

提示

如果不加任何参数，则系统进入单用户维护状态。没事别玩它。

此外，reboot等价于shutdown -r now，halt等价于shutdown -h now。

Chapter 7

Ubuntu包管理系统

所以我们要运用脑髓，放出眼光，自己来拿！
——鲁迅先生 拿来主义

由于包系统Debian和Ubuntu共享，所以这里Debian和Ubuntu的使用也就稍微含糊一些。

7.1 基本概念

7.1.1 Ubuntu下的软件

Ubuntu Linux下有丰富多彩的软件，可以说Windows下有的Ubuntu Linux下都有参见B。而取得这些软件的方法则不是传统模式，所有这些软件都可以从一个（或者数个）Ubuntu Linux的软件包服务器（中国国内的服务器详见8.3）上下载。

传统模式：到街边小贩处猥琐的问道：“有碟吗？”于是小贩说：“要什么碟？软件碟DVD碟黄碟都有。”于是你拐弯抹角到一个小小的胡同，进入黑暗的地下室，然后肮脏的交易开始了——当然，这是说的武汉市武昌区电脑城的故事，其时打击盗版运动正热火朝天。在中国科技大学附近，买盗版碟是见得太阳的。

7.1.2 deb包的概念

你应该知道WinRAR这些东西，不错，就是压缩软件。因为原始软件体积过于庞大，所以需要对他们进行压缩，以便于网络传送。同时你也应该知道“安装”（install）这个概念，就是把软件送到一个特定的位置，并且配置好它的周边环境，让它能够正确的运行。

deb包就是一个压缩起来的可以安装的软件包。它是Debian公司¹的Linux使用的压缩包格式。

deb包的优点，参见3.5中尊敬的张教授的一句话：

¹Debian的命名是这样的，一个名叫Ivan的男人制作了他自己的Linux，而他的太太名字叫Debria，两个人一合计，开公司吧，名字就叫Deb(ria)i(v)an==Debian。

- 便于获得。你可以通过aptitude (7.2.4) 从源 (7.1.3) 下载它们，也可以从其他网站上下载。
- 便于安装和卸载。Linux下的软件以往是可以容易的安装却很难卸载，因为你不知道在什么地方安装了什么，所以你就不知道在什么地方删除哪些文件。而通过deb包你就可以方便的安装卸载。参见7.3.1
- 便于管理。Linux下的软件是有相互依赖关系 (dependencies) 的，有时候有的软件被删除了，依赖它的软件 (可能是你需要的) 也就同时没有用了，同时你也不知道你的系统里有那些软件是没有被任何其他软件依赖而又没有用的。参见7.2.1和7.2.4
- 和其他公司的软件包兼容。你可以使用alien命令 (7.3.2) 将RedHat/FedoraCore的rpm软件包转换成deb包然后使用。
- 数量巨大，据我所了解Ubuntu中文源里面的包个数在10000以上，真正软件的个数应该在1000到2000左右。
- 最后，据统计²，Debian和Ubuntu的用户远较其他版本的Linux的用户多。这意味着大部分软件都应该有deb包以保证用户的需要。

想要知道自己系统里面有哪些软件，以及可以下载哪些软件，参阅7.2.4

7.1.3 源的概念

Ubuntu的软件包都是从一些网站上下载的，这些网站称作“源” (source)。不同的源中下载的东西可能不一样。例如Ubuntu英文官方源上就没有中文官方源里面的一些中文软件。有的专属源只能用于下载某些特定的软件。想要知道自己系统使用的源和国内的一些常用源，见8.3。

你学的东西都是从书上得来的，不同版本和作者的书的内容不同，不同的书的内容不同。

7.1.4 本地数据库

为了解决软件依赖关系，Ubuntu建立了本地数据库，记录软件包之间的依赖关系。所以每当你对软件包进行操作的时候包系统都会读取软件包数据库。

7.2 apt-***软件套装和aptitude

本节命令需要超级用户权限执行。建议你看完全全部内容，特别是最后的综述之后再下手办事。

7.2.1 apt-get

apt的意思是advanced package tools，高级包工具。get就是get，而不是你想像中的“god enlighten them”。它的作用就是安装和卸载包。当然，似乎新立得和Adept更有市场。不过这些是图形界面的东西了，我觉得介绍的必要不是很大。

顺便说一句，APT的作者声称“本APT有着超级牛力”。

²出处已忘记

用法

apt-get [参数] [包的名称]

参数

install 安装新的包。这些包将被下载，并且保存到硬盘之中再安装。这样如果你卸载它们之后又要安装的话，系统会从硬盘缓冲存储区中取出而不是重新下载。

remove 卸载软件包。安装到系统的部分将被删除，而配置文件会保留下来便于下次重新安装的时候能迅速恢复到原来状态。

autoremove 删除系统没有用到的软件包。

clean 删除硬盘里面的缓冲存储。

autoclean 删除缓冲存储里面的旧版本包。

source 下载源代码，不过这些东西对一般用户似乎没有用。

update 下载更新软件包列表信息，下载来源是/etc/apt/sources.list，有关这个文件的信息，参见8.3。

upgrade 升级系统。

dist-upgrade 发布版升级，将整个系统升级到一个新的版本，例如从Dapper到Edgy，要了解Ubuntu的发布信息，参见D。

check 检查系统完整性。

提示

想要知道自己需要什么样的包，还是得看aptitude，见下。

7.2.2 apt-cdrom

有时候你得到了一张系统光盘，希望能够从光盘上优先复制软件，这里你就要用到apt-cdrom命令。

用法

apt-cdrom [参数]

参数

add 将光盘添加到sources.list之中。参见8.3。

7.2.3 apt-key

在某些源下载软件需要一个密钥³。这时你需要下载这个密钥并且安装到apt系统里面。所以apt-key就派上了用场。

³我不明白为什么！

用法

apt-key [参数]

参数

add filename 添加一个名为filename的key文件。

del keyid 删除一个标志为keyid的key。

update 更新key系统。

list 列举所有已经安装的key。

提示

这个命令用到的不多，所以就不多介绍了。

7.2.4 aptitude

本节内容部分来自aptitude - -help，有删改。

aptitude是一个更好的apt-get。它有有界面操作和无界面操作两种。有界面操作将在图解一节讨论。

顺便说一句，APTITUDE的作者声称“这个aptitude没有超级牛力”。

用法

aptitude [参数] ...

参数

install 安装/升级软件包。

remove 卸载软件包。

purge 卸载软件包并删除其配置文件。

hold 将软件包置于保持状态。即是：如果有新的版本，在操作过程中也不进行自动升级。

unhold 取消对一个软件包的保持命令。

update 下载新/可升级软件包列表。

upgrade 执行一次安全的升级，只升级可以升级的应用程序软件包，不改动系统核心部分。

dist-upgrade 执行升级，可能会安装和卸载软件包，改动系统核心部分。

search 按名称搜索软件包。

show 显示一个软件包的详细信息。

clean 删除已下载的软件包文件。

autoclean 删除旧的已下载软件包文件。

changelog 查看一个软件包的变更日志。

download 下载软件包的.deb文件。不安装。

reinstall 下载并(可能)重新安装一个现在已经安装了的软件包。

提示

如果你碰到下面的内容:

```
E: 无法打开锁文件 /var/lib/dpkg/lock - open (13 Permission denied)
E: Unable to lock the administration directory (/var/lib/dpkg/), are you root?
```

检查下自己是否是root, 如果不是, 想想什么命令能帮你摆脱困境? 参见6.1.2。

例子

下面的命令安装fvwm, 这是一个简单的图形界面。

```
xxxx@xxxx:/home/xxxx# aptitude install fvwm
```

```
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
Reading state information... 完成
正在读取扩展状态文件
正在初始化软件包状态... 完成
创建标签数据库... 完成
```

aptitude会处理好依赖关系, 下载相应的软件包。

下列软件包将被自动安装:

```
fvwm-icons librpplay3 libstroke0
```

下列“新”软件包将被安装。

```
fvwm fvwm-icons librpplay3 libstroke0
```

0 个软件包被升级, 新安装4 个, 0 个将被删除。

需要获取 3444kB 的存档。解包后将使用 9699kB。

您要继续吗? [Y/n/?]

选择y键安装fvwm。系统会提示下面内容:

警告: 您将安装以下软件包的不可信版本!

不可信的软件可能会危害您的系统安全。

只有当您非常清楚地了解这就是您所要执行的操作时, 才应该进行安装操作。

```
fvwm-icons libstroke0
```

您想忽略这条警告信息并继续执行吗?

要继续, 请键入“是”; 要中止, 请键入“否”:

不可信版本是因为没有从官方源下载（我使用的是cn99的源，见8.3）。一般不必理睬，输入“是”。`aptitude`将要调用`wget`（参见10.2.2）对软件包进行下载。如下：

```
正在编辑扩展状态信息... 完成
读取: 1 http://ubuntu.cn99.com edgy/universe librplay3 3.3.2-11 [43.5kB]
读取: 2 http://ubuntu.cn99.com edgy/universe libstroke0 0.5.1-5 [17.9kB]
读取: 3 http://ubuntu.cn99.com edgy/universe fvwm 1:2.5.16-2 [2998kB]
读取: 4 http://ubuntu.cn99.com edgy/universe fvwm-icons 2001.08.13-3 [384kB]
已下载 3444kB, 耗时 16s (213kB/s)
```

然后调用`dpkg`（参见7.3.1）进行安装。

```
正在预设定软件包 ...
选中了曾被取消选择的软件包 librplay3。
(正在读取数据库 ... 系统当前总共安装有 136615 个文件和目录。)
正在解压缩 librplay3 (从 .../librplay3_3.3.2-11_i386.deb) ...
选中了曾被取消选择的软件包 libstroke0。
正在解压缩 libstroke0 (从 .../libstroke0_0.5.1-5_i386.deb) ...
选中了曾被取消选择的软件包 fvwm。
正在解压缩 fvwm (从 .../fvwm_1%3a2.5.16-2_i386.deb) ...
选中了曾被取消选择的软件包 fvwm-icons。
正在解压缩 fvwm-icons (从 .../fvwm-icons_2001.08.13-3_all.deb) ...
正在设置 librplay3 (3.3.2-11) ...

正在设置 libstroke0 (0.5.1-5) ...

正在设置 fvwm (2.5.16-2) ...

正在设置 fvwm-icons (2001.08.13-3) ...
```

至此设定结束。⁴

下面的命令删除`fvwm`这个包，把它和它所产生的配置文件一起干掉：

```
xxxx@xxxx:/home/xxxx#aptitude purge fvwm
```

开始显示的是初始化内容——都是废话：

```
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
Reading state information... 完成
正在读取扩展状态文件
正在初始化软件包状态... 完成
创建标签数据库... 完成
```

然后才是真正的主题：

下列软件包未被使用，将被“删除”：

```
fvwm-icons librplay3 libstroke0
```

下列软件包将被“删除”：

```
fvwm{p}
```

0 个软件包被升级，新安装0 个， 4 个将被删除， 同时 1 个将不升级。

需要获取 0B 的存档。解包后将释放 9699kB。

您要继续吗？[Y/n/?]

⁴若要问我为什么选择`fvwm`作为例子，参见参考文献[2]。

不被用到的软件包也会被删除，以保证一个“干净”的系统。按Y键继续：

```
正在编辑扩展状态信息... 完成
(正在读取数据库 ... 系统当前总共安装有 137225 个文件和目录。)
正在删除 fvwm ...
正在清除 fvwm 的配置文件 ...
(正在读取数据库 ... 系统当前总共安装有 136974 个文件和目录。)
正在删除 fvwm-icons ...
正在删除 librplay3 ...
正在删除 libstroke0 ...
```

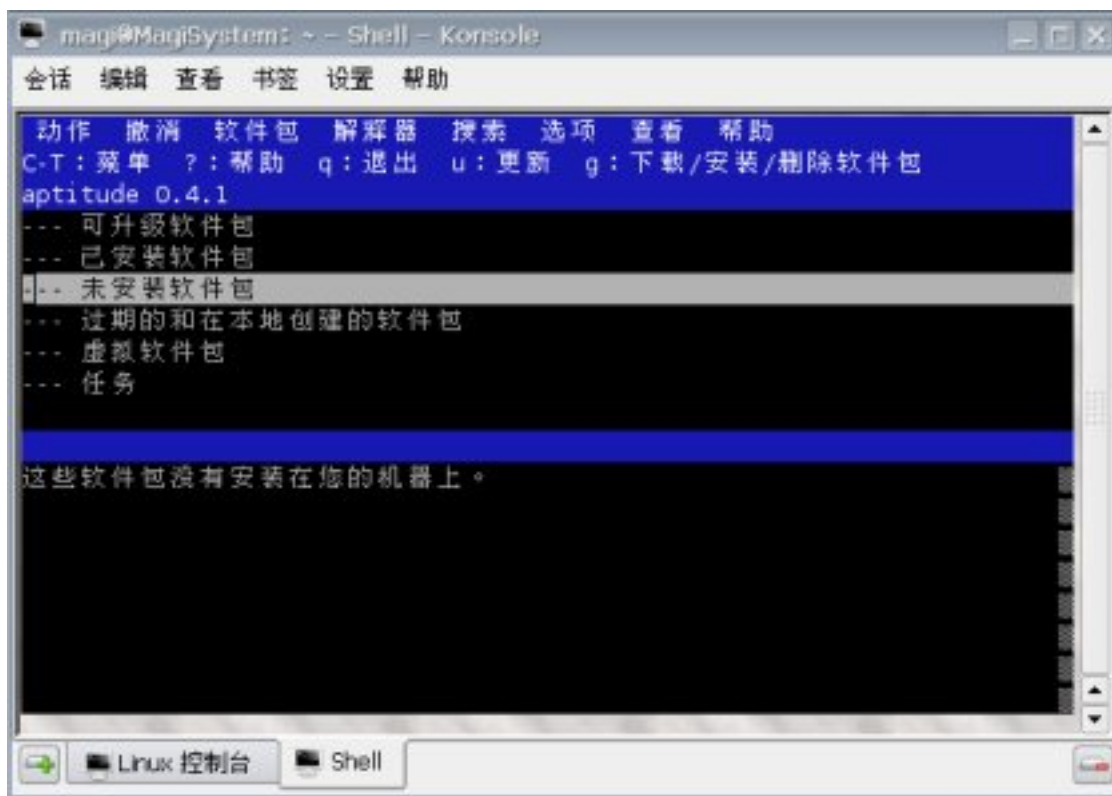
于是除去了fvwm这个心腹大患。

图解

aptitude还有一个图形化界面，十分方便好用，汉化也很完善。请在普通用户模式下输入aptitude，让我们来玩玩这个东西：

```
xxxx@xxxx:/home/xxxx$aptitude
```

显示下面的界面：

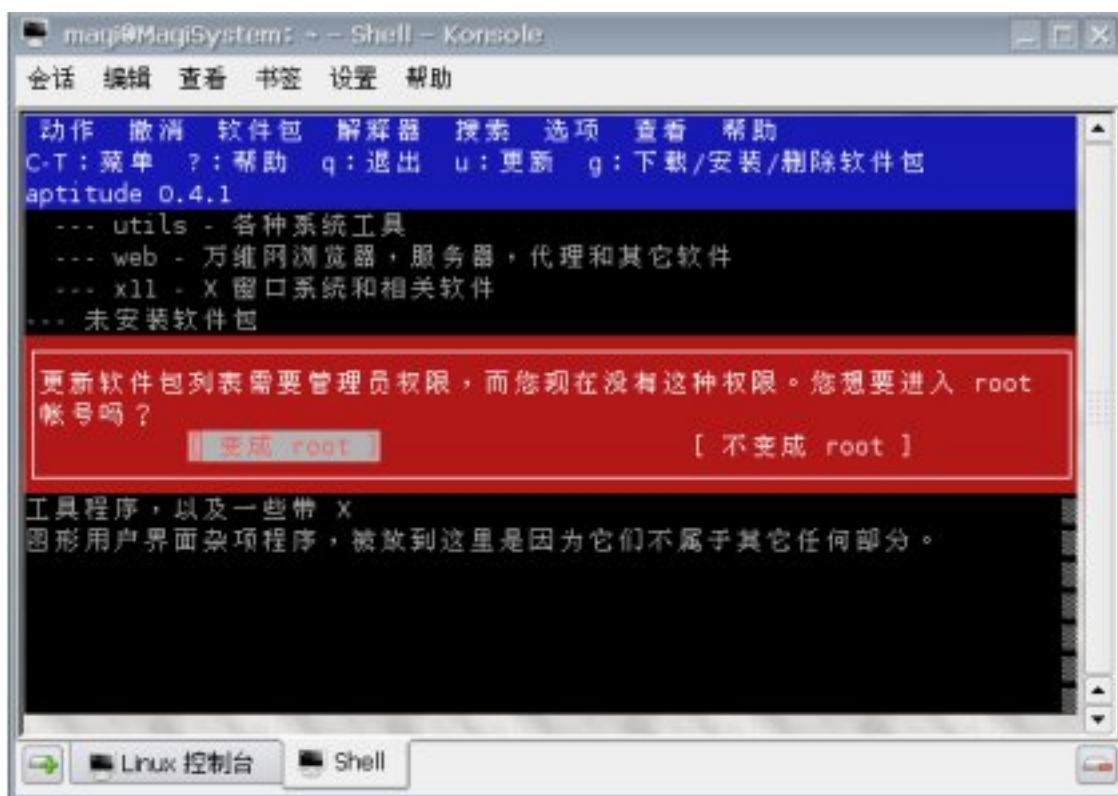


你可以用上下箭头移动光标，来选择你感兴趣的条目。按空格键展开/收回软件树。下边部分显示的是你选定的目录或者软件包的描述。

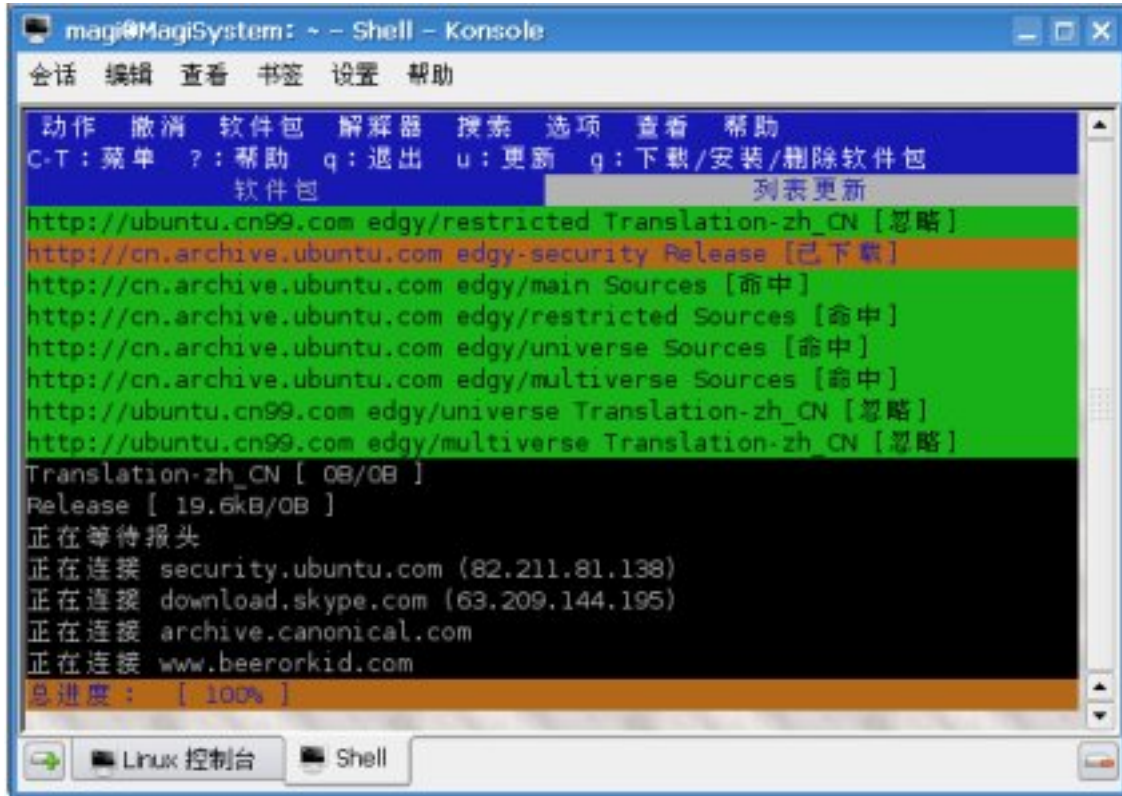
软件包左边显示的是软件包的状态和将要进行的动作。将要被安装的条目用绿色背景，将被删除的条目用粉红色背景，破坏的软件包用红色背景。

- v 虚拟，并不真实存在这个软件包，只是出于安装方便目的而存在。
- B 损坏，依赖关系无法满足。这个软件包可能不能正常运行。按B键可以搜索这样的软件包。屏幕下方会显示它需要满足何种依赖关系才能继续运行。
- i 安装，已经安装或者将要安装。
- c 卸载，但是保存配置文件。和remove命令相同。
- p 扫荡干净，和purge相同。
- d 删除。
- r 重新安装。
- u 升级，安装新版软件，带来更多功能（和麻烦）。

按F10或者CTRL-T选择菜单。菜单项右边是这个命令的快捷键。
选择更新软件包列表的时候会有以下界面：



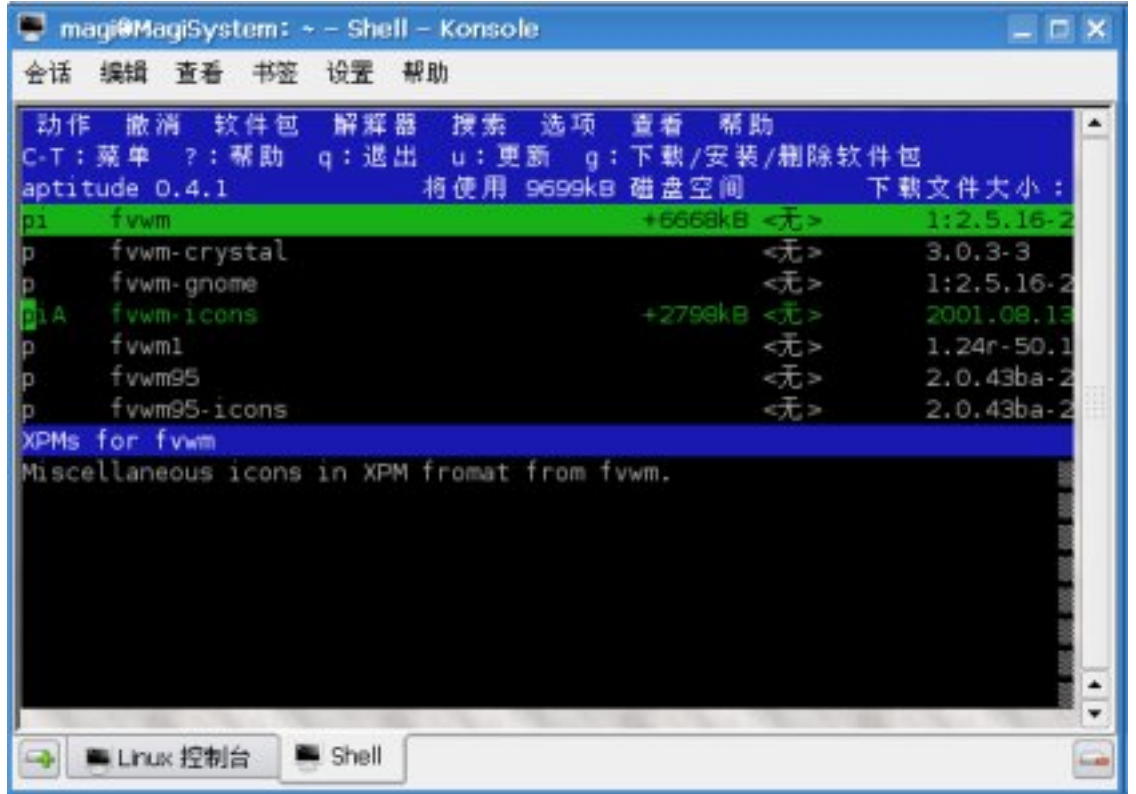
它会聪明的告诉你该干什么，输入你的密码，升级到root权限，继续：



The screenshot shows a terminal window titled "magi@MagiSystem: ~ - Shell - Konsole". The terminal displays the output of an APT update command. The output is as follows:

```
动作 撤消 软件包 解释器 搜索 选项 查看 帮助
C-T: 菜单 ? : 帮助 q : 退出 u : 更新 g : 下载/安装/删除软件包
软件包 列表更新
http://ubuntu.cn99.com edgy/restricted Translation-zh_CN [忽略]
http://cn.archive.ubuntu.com edgy-security Release [已下载]
http://cn.archive.ubuntu.com edgy/main Sources [命中]
http://cn.archive.ubuntu.com edgy/restricted Sources [命中]
http://cn.archive.ubuntu.com edgy/universe Sources [命中]
http://cn.archive.ubuntu.com edgy/multiverse Sources [命中]
http://ubuntu.cn99.com edgy/universe Translation-zh_CN [忽略]
http://ubuntu.cn99.com edgy/multiverse Translation-zh_CN [忽略]
Translation-zh_CN [ 0B/0B ]
Release [ 19.6kB/0B ]
正在等待报头
正在连接 security.ubuntu.com (82.211.81.138)
正在连接 download.skype.com (63.209.144.195)
正在连接 archive.canonical.com
正在连接 www.beerorkid.com
总进度: [ 100% ]
```

选择要安装的包，按加号，则它被自动安装（升级），减号则代表卸载：



按g键，则会预览你要做什么：

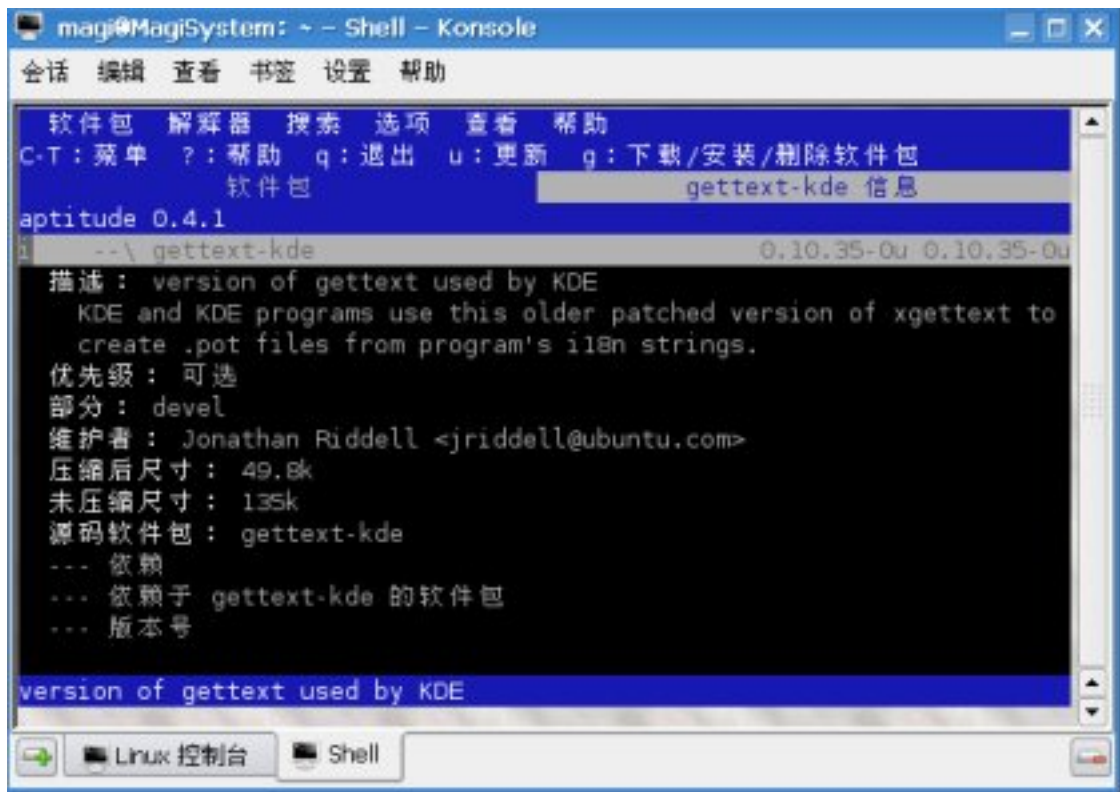


The screenshot shows a terminal window titled "magie@Magisystem: ~ - Shell - Konsole". The window displays the output of the aptitude command, showing a list of software packages to be installed. The packages are listed in a table format with columns for package name, size, and version. The packages are: fwm-icons (279kB), librplay3 (123kB), and libstroke0 (111kB). Below the table, there is a section for "被保持的软件包" (held packages) which includes gaim and libpng12-0. The terminal also shows a menu at the top and a status bar at the bottom.

```
aptitude 0.4.1 将使用 9699kB 磁盘空间 下载文件大小 :
--\ 被自动安装以满足依赖关系的软件包
piA fwm-icons +279kB <无> 2001.08.13
ciA librplay3 +123kB <无> 3.3.2-11
piA libstroke0 +111kB <无> 0.5.1-5
--\ 被保持的软件包
i gaim 2.0.0beta4 1:2.0.0+be
i libpng12-0 1.2.8rel-5 1.2.8rel-5

这些软件包被安装的原因是您选择安装的一个软件包要求它们。
如果您选择了一个软件包，其当前状态的描述将会出现在这片空间。
```


对一个特定的软件包按回车可以得到它的相应资料：



```

magi@MagiSystem: ~ - Shell - Konsole
全话 编辑 查看 书签 设置 帮助
软件包 解释器 搜索 选项 查看 帮助
C-T: 菜单 ? : 帮助 q : 退出 u : 更新 g : 下载/安装/删除软件包
      软件包
      gettext-kde 信息
aptitude 0.4.1
-- \ gettext-kde 0.10.35-0u 0.10.35-0u
描述: version of gettext used by KDE
      KDE and KDE programs use this older patched version of xgettext to
      create .pot files from program's i18n strings.
优先级: 可选
部分: devel
维护者: Jonathan Riddell <jriddell@ubuntu.com>
压缩后尺寸: 49.8k
未压缩尺寸: 135k
源码软件包: gettext-kde
--- 依赖
--- 依赖于 gettext-kde 的软件包
--- 版本号
version of gettext used by KDE
Linux 控制台 Shell

```

这些资料对一般用户来说，只有描述比较重要，可惜是English的……
按/键可以进行搜索。搜索是属于“增量搜索”的，即是：输入一个字母，就开始搜索相应条目。

aptitude的更进一步使用可以参考帮助：选择帮助菜单或者按?键。

参考 建议阅读man aptitude。虽然也有四五百行英文。

7.2.5 安装/卸载软件包综述

以下列举了一些关于安装卸载软件包的经验：

- 对软件包的操作必须由超级用户进行。
- 操作前先用aptitude update命令，保证本地数据库是最新的，同时你安装
的版本也是最新的。
- 操作过程中请尽量不要开动其他进程。
- 如果你缺少硬盘空间，试试aptitude autoclean
- 软件包的原则是“够用就行”，如果你希望稳定的使用一个软件包系
统。所以不要每次都用upgrade命令。当然，尝鲜也不是不可以。

7.3 其他包管理命令

7.3.1 dpkg

dpkg这个命令是包管理系统的核心。无论apt-get还是aptitude最终都是调用dpkg进行包管理的。不过既然有了aptitude这个工具，dpkg的用处就不是很大了。下面仅介绍一些常见的用法。

dpkg只有超级用户可用。

dpkg的- help已经汉化。这更给了我偷懒的权力。

用法示例

有些deb文件是自己下载的，例如Code::BlocksB是官方源没有的软件包，需要自己到官方主页下载。安装一些自己下载的deb文件不能使用aptitude等命令，这样做：

```
xxxx@xxxx:/home/xxxx# dpkg -i CB_20061107_rev3190_Ubuntu6.06.deb
```

有时候aptitude执行了一半被强行中断，为了修复包管理系统，用下面的命令：

```
xxxx@xxxx:/home/xxxx# dpkg --configure -a
```

对所有包进行重新检查设置。

7.3.2 alien

有些软件包只有rpm格式，需要转换成为deb格式进行安装。这时用到alien命令进行转换。

用法

alien [文件名]

提示

产生的deb文件要用dpkg安装，详见7.3.1。

Chapter 8

系统核心配置文件

紧密团结在以党中央为核心的团
体之中
——高中政治教科书

Linux里面没有注册表，一切系统配置都是由修改相应文件产生的。下面介绍一些系统配置文件的格式，以便于你修改。你可以用nano (5.2) 或者vim (5.3) 来修改相应的内容。

8.1 /etc/passwd

这是重中之重，好比宪法一般。下面请看一个特定的passwd文件（有删节）：

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
magi:x:1000:1000:Magi Su,,,:/home/magi:/bin/bash
```

从左到右依次：用户账号名称，密码¹，用户组编号，用户编号，用户名称（长），使用的Shell文件路径名称。

passwd文件没必要编辑，使用相应命令就可以了，参见6.1

8.2 /etc/fstab

fstab是挂载系统的列表，可以而且应该仔细说说。下面请看源文件（当然，你会认为这里有删节。不幸的是，你错了，这是我的系统的fstab完全版本）：

¹原来在这里可以看到被加密的密码，后来由于passwd文件经常被窃取，从而使得黑客能够用john the ripper等工具破解root密码，现在密码被保存在shadow文件之中。你可以自己检查passwd和shadow的权限，passwd普通用户可以读取而shadow只有root可以读取。

```

proc          /proc          proc    defaults      0      0
/dev/hda5     /              ext3    defaults,errors=remount-ro 0 1
/dev/hda2     /boot          ext3    defaults      0      2
/dev/hda9     /usr           ext3    defaults      0      2
/dev/hda6     /home         ext3    defaults      0      1
/dev/hda7     none          swap    sw            0      0
/dev/hda12    /multimedia   ext3    defaults      0      2
/dev/hda1     /datay        ext3    defaults

```

第一排想必你也记起来了，如果你不记得了，就看看3.10吧。

其他几排都是关于分区挂载的数据。最左边一列是分区，记得查阅3.1。第二列是挂载到的目录。第三列是文件系统的类型。第四列是挂载参数。最后两列分别是dump和pass，我不太明白具体含义。

如果你想要计算机在开机的时候自动挂载某些分区，不妨照葫芦画瓢，增加你自己的数据，例如：

```

/dev/hda8     /dataz        reiserfs defaults      0      0

```

想要了解文件系统，温习一下6.2

8.3 /etc/apt/sources.list

这里给出一个sources.list的段落。注意里面的源都是网络源，没有本地的光盘源。

```

#中国科技大学的源
deb http://debian.ustc.edu.cn/ubuntu/ edgy
main restricted universe multiverse
deb http://debian.ustc.edu.cn/ubuntu/ edgy-backports
restricted universe multiverse
deb http://debian.ustc.edu.cn/ubuntu/ edgy-proposed
main restricted universe multiverse
deb http://debian.ustc.edu.cn/ubuntu/ edgy-security
main restricted universe multiverse
deb http://debian.ustc.edu.cn/ubuntu/ edgy-updates
main restricted universe multiverse
deb-src http://debian.ustc.edu.cn/ubuntu/ edgy
main restricted universe multiverse

```

#号代表注释，只是方便阅读没有实际意义。

deb代表我们需要的是Debian包²。deb-src则是下载源代码的指示。之后是包的来源的网址（如果是光盘源则是光盘的名称），再后面是系统的标记，在这里是edgy。

之后main代表主要的软件包，restricted代表只有可执行文件的软件包（没有源代码），universe是不被官方支持的自由软件，multiverse是不被官方支持的非自由软件。当然，你真正需要的时候，谁管这么多！

以下列举一些常用的源的地址：

²记得我说过Ubuntu是Debian的衍生物？

```
#CN99源, 适合电信网通用用户
deb http://ubuntu.cn99.com/ubuntu/ edgy
main restricted universe multiverse
deb http://ubuntu.cn99.com/ubuntu/ edgy-updates
main restricted universe multiverse
deb http://ubuntu.cn99.com/ubuntu/ edgy-security
main restricted universe multiverse
deb http://ubuntu.cn99.com/ubuntu/ edgy-backports
main restricted universe multiverse
deb http://ubuntu.cn99.com/ubuntu-cn/ edgy
main restricted universe multiverse
deb http://ubuntu.cn99.com/ubuntu/ edgy-proposed
main restricted universe multiverse

#LupaWorld源 适合电信网通用用户
deb http://mirror.lupaworld.com/ubuntu/archive edgy
main restricted universe multiverse
deb http://mirror.lupaworld.com/ubuntu/archive edgy-security
main restricted universe multiverse
deb http://mirror.lupaworld.com/ubuntu/archive edgy-updates
main restricted universe multiverse
deb http://mirror.lupaworld.com/ubuntu/archive edgy-backport
main restricted universe multiverse
deb http://mirror.lupaworld.com/ubuntu/ubuntu-cn edgy
main restricted universe multiverse
deb http://mirror.lupaworld.com/ubuntu/backports
edgy-extras main restricted universe multiverse

#Ubuntu中国 官方源
deb http://cn.archive.ubuntu.com/ubuntu edgy
main restricted universe multiverse
deb http://cn.archive.ubuntu.com/ubuntu edgy-security
main restricted universe multiverse
deb http://cn.archive.ubuntu.com/ubuntu edgy-updates
main restricted universe multiverse
deb http://cn.archive.ubuntu.com/ubuntu edgy-backports
main restricted universe multiverse
deb-src http://cn.archive.ubuntu.com/ubuntu edgy
main restricted universe multiverse
deb-src http://cn.archive.ubuntu.com/ubuntu edgy-security
main restricted universe multiverse
deb-src http://cn.archive.ubuntu.com/ubuntu edgy-updates
main restricted universe multiverse
deb-src http://cn.archive.ubuntu.com/ubuntu edgy-backports
main restricted universe multiverse
deb-src http://security.ubuntu.com/ubuntu edgy-security
main restricted universe multiverse

#上海交通大学 适合教育网用户
deb http://ftp.sjtu.edu.cn/ubuntu/ edgy
```

```
main multiverse restricted universe
  deb http://ftp.sjtu.edu.cn/ubuntu/ edgy-backports
main multiverse restricted universe
  deb http://ftp.sjtu.edu.cn/ubuntu/ edgy-proposed
main multiverse restricted universe
  deb http://ftp.sjtu.edu.cn/ubuntu/ edgy-security
main multiverse restricted universe
  deb http://ftp.sjtu.edu.cn/ubuntu/ edgy-updates
main multiverse restricted universe
  deb http://ftp.sjtu.edu.cn/ubuntu-cn/ edgy
main multiverse restricted universe

#Skype 语音聊天软件
  deb http://download.skype.com/linux/repos/debian/
stable non-free

#一些商业软件的下载
  deb http://archive.canonical.com/ubuntu edgy-commercial main
```

如果你连包的概念都不清楚了，请参见7.1

Chapter 9

GNU编译环境

自己动手
丰衣足食
——毛泽东

本章并不深入探讨GNU编译环境的详细内容，仅仅讨论在科研计算编程中用到的一部分——甚至是一小部分内容。例如，make命令就不予讨论了。¹

9.1 GNU编译器概述

GNU编译器支持大量的源代码类型，常见的有C，C++，Fortran²之类。

所有GNU编译器（gcc，g++，gfortran）的原理都是将源程序先编译成某种中间语言，之后再对中间语言进行第二次编译，生成机器码。编译默认输出文件为源代码所在位置目录的./a.out。

本小册子只介绍基本的使用方式。

9.1.1 gxx

xx是可以使用cc，++和fortran替代的，分别代表C，C++，Fortran的编译器。

用法

gxx [参数] [源程序]

参数

-g 增加调试信息

¹毕竟非计算机专业的学生，特别是化学物理专业的学生没必要搞的这么深入，哪里有这么精力啊。

²如果你问我为什么提到Fortran，我的答复是中国科技大学的搞理论计算的一些教授使用Fortran。我写这一部分主要是为了母校特别是同系同学中那些被老板要求用Linux编程的同学服务，e.g. Panda同学。

9.2 调试器

每个程序都几乎不可能一开始写即是正确的，即便是最简单的程序也可能出错，而这些错误可能是语法错误，就是你的程序在某个地方有错误的语句，没有正确的加上分号或者是其他问题，也可能是逻辑错误，就是你的程序思路有问题。前者编译器可以几乎正确的指出这些问题。然而后者则需要调试器来解决。

9.2.1 gdb

`gdb`是Gnu DeBugger的缩写，GNU调试器。它是一个功能强大的调试工具，然而学习使用起来颇为困难。

如果想要阅读相关材料，请到<http://blog.csdn.net/haol/category/9197.aspx>阅读相应文章。

9.2.2 gdb的前端

为了解决`gdb`使用困难的问题，很多不同版本的`gdb`前端（相当于终结者的血肉外皮）光荣诞生了。

`cdb`

`cdb`是字符界面的`gdb`前端。

`ddd`

`ddd`是图形界面的`gdb`前端。

Chapter 10

命令行下的Internet

通过建立一条信息高速公路来刺激美国经济的主张成为民主党的竞选纲领之一，后来亦成为美国民主党政府的经济改革政策的重要组成部分。
——中国科学院

本章介绍命令行下的网络程序。

10.1 ADSL上网设置

10.1.1 pppoeconf

这个命令用来设置ADSL上网的账号和密码。只有超级用户可以使用。

使用方法

以下使用方法在中国科技大学寝室电话ADSL接口上通过。

如果你懂得英文的话使用方法非常简单，只要按部就班的选择就行了。但是为了照顾看不懂的同学（如果你是中科大校内学生我真的怀疑你是否是走后门进来的……）我就做一个解释。

第一屏是一个选择，询问是否列出了所有网络接口。一般直接选是就可以了。之后是一个网络设置过程，请稍微等待。如果出现NOT CONNECTED窗口，请检查是否接好了各个接口，这代表没有找到网络链接。否则出现OKAY TO MODIFY，则选择“是”继续设置，否则选择“否”。然后出现POPULAR OPTIONS选项，直接选是即可。

然后出现提示，依次输入上网账号的用户名和密码，注意密码不会隐藏。

信任你同寝室的哥们，但要小心外贼难防！

接着是USE PEER DNS选项，选择DNS（域名解析系统），这里直接选择“是”即可。之后是LIMITED MSS PROBLEM，直接选“是”即可。设置到此为止。

然后询问是否一开机就接通网络，随你选。之后选择是否立即接通网络，若选择是，则出现CONNECTION INITIATED，表明网络已经接通，否则程序自动退出。

10.1.2 pon

如果是ADSL设置完毕，只要输入pon dsl-provider即可打开链接，不必登入超级用户。

```
xxxx@xxxx:/home/xxxx$pon dsl-provider
```

接入无论成功与否，不会有显示。如果需要检查相应的内容，参见10.1.4

10.1.3 poff

关闭网络链接。事实上这个命令是用kill杀死相应的pon进程达到目的的。

用法

poff [参数]

参数

-r 断开网络之后重新接通。

-a 关闭所有pon打开的链接。

提示

如果你多次输入pon dsl-provider打开网络链接，则需要加入-a指令集体终止它们。

10.1.4 plog

检测pppoe链接的日志。事实上这个命令就是用tail (4.3.4) 来输出/var/log/ppp.log的最后几条内容。检查日志的时候只要观察最后一段时间内的日志就可以了。

一个异常的联网日志

以下是一个异常的日志：

```
Nov 18 15:23:53 localhost pppd[21126]:
Plugin rp-pppoe.so loaded.
Nov 18 15:23:53 localhost pppd[21128]:
pppd 2.4.4 started by root, uid 0
Nov 18 15:23:53 localhost pppd[21128]:
PPP session is 92
Nov 18 15:23:53 localhost pppd[21128]:
Using interface ppp0
Nov 18 15:23:53 localhost pppd[21128]:
Connect: ppp0 <--> eth0
Nov 18 15:23:53 localhost pppd[21128]:
PAP authentication failed
Nov 18 15:23:53 localhost pppd[21128]:
Connection terminated.
```

倒数第二行显示了错误的类型，即是PAP认证错误，一般是账号和密码没有对上。

另外一个异常的日志是：

```
Nov 18 15:32:18 localhost pppd[21408]:
Timeout waiting for PADO packets
Nov 18 15:32:18 localhost pppd[21408]:
Unable to complete PPPoE Discovery
Nov 18 15:32:18 localhost pppd[21408]:
Terminating on signal 15
Nov 18 15:32:18 localhost pppd[21408]:
Exit.
Nov 18 15:32:21 localhost pppd[21450]:
Plugin rp-pppoe.so loaded.
Nov 18 15:32:21 localhost pppd[21452]:
pppd 2.4.4 started by root, uid 0
```

这是我有意拔下网线所得到的日志，没有任何登录信息，只有pppd开始的信息。

一个正常的日志如下：

```
Nov 18 15:30:03 localhost pppd[21311]:
Plugin rp-pppoe.so loaded.
Nov 18 15:30:03 localhost pppd[21313]:
pppd 2.4.4 started by root, uid 0
Nov 18 15:30:03 localhost pppd[21313]:
PPP session is 105
Nov 18 15:30:03 localhost pppd[21313]:
Using interface ppp0
Nov 18 15:30:03 localhost pppd[21313]:
Connect: ppp0 <--> eth0
```

10.2 命令行下的网上“冲浪”

我个人觉得网上“冲浪”这个词很愚蠢的有意思。浏览网页为什么要叫surf the Internet呢？

命令行上同样可以上网，不过有很多问题，例如图片无法浏览等等，而且很多图形接口下的工具都没有。所以我不推荐你这么做。不过有些命令只有命令行上有，或者命令行上的这些命令功能强大。所以用用也不错。

下文用到的URL你可以理解成为网址。

10.2.1 ping

检查远程网站能否接通。

实质就是发送一个数据包，然后检查对方是否会响应。

就是给别人拼命发短信，看对方回多少条。一般如果回的多，代表对方愿意理你。像我给我前女友发短信她就从来不回，这个叫做“ping不通”。

它有一个特别的作用，就是你可以选择apt源7.1来每个都ping一下，看看哪个的反应速度快，就选择它作为你的主要源，可以节省大量时间。

用法

ping [参数] [URL]

参数

-c 发送“短信”次数。

-i 发送的间隔时间，按秒计算。只有超级用户才能设置少于0.2秒的间隔时间。

例子

下面这个例子是联系www.csdn.net¹，发送10个包，间隔0.1秒：

```
xxxx@xxxx:/home/xxxx#ping -c10 -i0.1 www.csdn.net
```

```
PING www.csdn.net (211.100.21.179) 56(84) bytes of data.  
64 bytes from 211.100.21.179: icmp_seq=3 ttl=52 time=325 ms  
64 bytes from 211.100.21.179: icmp_seq=4 ttl=52 time=326 ms  
64 bytes from 211.100.21.179: icmp_seq=5 ttl=52 time=327 ms  
64 bytes from 211.100.21.179: icmp_seq=6 ttl=52 time=326 ms  
64 bytes from 211.100.21.179: icmp_seq=7 ttl=52 time=325 ms  
64 bytes from 211.100.21.179: icmp_seq=8 ttl=52 time=325 ms  
64 bytes from 211.100.21.179: icmp_seq=9 ttl=52 time=329 ms  
64 bytes from 211.100.21.179: icmp_seq=10 ttl=52 time=329 ms
```

```
--- www.csdn.net ping statistics ---  
10 packets transmitted, 8 received, 20% packet loss, time 104ms  
rtt min/avg/max/mdev = 325.876/327.170/329.938/1.627 ms, pipe 8
```

第一行是显示要ping的网站名称和它的IP地址（就是网络用户的一个四位的唯一的编号）。下面八行显示的是ping的结果，一共收到了八个包²。最左边的“方程”是包的顺序，中间ttl是一个“指纹”数据，可以不用关心。最后部分是发送/接受之间的时间差，越小表明速度越快。

10.2.2 wget

这是一个网络下载工具。

用法

```
wget [URL]
```

提示

详细的使用方法见man wget。

10.2.3 w3m

这是一个文本状态下的浏览器……自从zhcon不能用了，它的存在就没有意义了……

¹这是一个国内著名的程序员站点，高手如云，好比袁峰袁老大。

²看样子我的网络质量还不怎么样……

Appendix A

Grep学习笔记

师夷长技以自强

A.1 版权声明

整理: Jims of 肥肥世家 <yjnet@21cn.com>

Copyright ? 2004 本文遵从GPL协议, 欢迎转载、修改、散布。

第一次发布时间: 2004年7月16日

L^AT_EX转换¹: Magi Su @ University of Science and Technology of China

A.2 目录

1. grep简介
2. 用法
3. 参数
4. grep正则表达式元字符集（基本集）
5. 用于egrep和 grep -E的元字符扩展集
6. POSIX字符类
7. 实例

A.3 正文

A.3.1 grep简介

grep（Global search Regular Expression and Print out the line，全面搜索正则表达式并把行打印出来）是一种强大的文本搜索工具，它能使用正则表达式搜索文本，并把匹配的行打印出来。Unix的grep家族包括grep、egrep和fgrep。egrep和fgrep的

¹本人对原文格式进行了修改，以便用L^AT_EX排版。修改了一些小节的顺序，删除了有短形式的长的参数（保留这些参数的短形式）。添加的内容用仿宋字体标出。

命令只跟grep有很小不同。egrep是grep的扩展，支持更多的re元字符，fgrep就是fixed grep或fast grep，它们把所有的字母都看作单词，也就是说，正则表达式中的元字符表示回其自身的字面意义，不再特殊。linux使用GNU版本的grep。它功能更强，可以通过-G、-E、-F命令行选项来使用egrep和fgrep的功能。

grep的工作方式是这样的，它在一个或多个文件中搜索字符串模板。如：果模板包括空格，则必须被引用，模板后的所有字符串被看作文件名。搜索的结果被送到屏幕，不影响原文件内容。

grep可用于shell脚本，因为grep通过返回一个状态值来说明搜索的状态，如：果模板搜索成功，则返回0，如：果搜索不成功，则返回1，如：果搜索的文件不存在，则返回2。我们利用这些返回值就可进行一些自动化的文本处理工作。

A.3.2 用法

grep可以作为管道符号|之后的命令使用。也可以对文件使用：

```
grep [参数] [正则表达式模板] [文件]
```

A.3.3 参数

- ? 同时显示匹配行上下的?行，如：grep -2 pattern filename同时显示匹配行的上下2行。
- b 打印匹配行前面打印该行所在的块号码。
- c 只打印匹配的行数，不显示匹配的内容。
- f File 从文件中提取模板。空文件中包含0个模板，所以什么都不匹配。
- h 当搜索多个文件时，不显示匹配文件名前缀。
- i 忽略大小写差别。
- q 取消显示，只返回退出状态。0则表示找到了匹配的行。
- l 打印匹配模板的文件清单。
- L 打印不匹配模板的文件清单。
- n 在匹配的行前面打印行号。
- s 不显示关于不存在或者无法读取文件的错误信息。
- v 反检索，只显示不匹配的行。
- w 如：果被\ <和\ >引用，就把表达式做为一个单词搜索。
- V 显示软件版本信息。

A.3.4 grep正则表达式元字符集（基本集）

`^` 锚定行的开始 如：'`^grep`'匹配所有以grep开头的行。

`$` 锚定行的结束 如：'`grep$`'匹配所有以grep结尾的行。

`.` 匹配一个非换行符的字符 如：'`gr.p`'匹配gr后接一个任意字符，然后是p。

`*` 匹配零个或多个先前字符 如：'`*grep`'匹配所有有一个或多个空格后紧跟grep的行。`.*`一起用代表任意字符。

`[]` 匹配一个指定范围内的字符，如：'`[Gg]rep`'匹配Grep和grep。

`[^]` 匹配一个不在指定范围内的字符，如：'`[^A-FH-Z]rep`'匹配不包含A-R和T-Z的一个字母开头，紧跟rep的行。

`\ (.\)` 标记匹配字符，如：'`\ (love\)`'，love被标记为1。

`\ <` 锚定单词的开始，如：'`\ <grep`'匹配包含以grep开头的单词的行。

`\ >` 锚定单词的结束，如：'`grep\ >`'匹配包含以grep结尾的单词的行。

`x\ {m\ }` 重复字符x，m次，如：'`o\ 5\` '匹配包含5个o的行。

`x\ {m, \ }` 重复字符x，至少m次，如：'`o\ {5, \ }`'匹配至少有5个o的行。

`x\ {m, n\ }` 重复字符x，至少m次，不多于n次，如：'`o\ {5, 10\ }`'匹配5-10个o的行。

`\ w` 匹配文字和数字字符，也就是[A-Za-z0-9]，如：'`G\ w*p`'匹配以G后跟零个或多个文字或数字字符，然后是p。

`\ W` `\ w`的反置形式，匹配一个或多个非单词字符，如：点号句号等。

`\ b` 单词锁定符，如：'`\ bgrep\` '只匹配grep。

A.3.5 用于egrep和grep -E的元字符扩展集

`+` 匹配一个或多个先前的字符。如：'`[a-z]+able`'，匹配一个或多个小写字母后跟able的串，如：loveable, enable, disable等。

`?` 匹配零个或多个先前的字符。如：'`gr?p`'匹配gr后跟一个或没有字符，然后是p的行。

`a|b|c` 匹配a或b或c。如：'`grep—sed`'匹配grep或sed

`()` 分组符号，如：'`love(able—rs)ov+`'匹配loveable或lovers，匹配一个或多个ov。

`x{m}`, `x{m, }`, `x{m, n}` 作用同`x\ {m\ }`, `x\ {m, \ }`, `x\ {m, n\ }`

A.3.6 POSIX字符类

为了在不同国家的字符编码中保持一致，POSIX(The Portable Operating System Interface)增加了特殊的字符类，如：[: alnum:]是A-Za-z0-9的另一个写法。要把它们放到[]号内才能成为正则表达式，如：[A-Za-z0-9]或[: alnum:]。在linux下的grep除fgrep外，都支持POSIX的字符类。

- [: alnum:] 文字数字字符
- [: alpha:] 文字字符
- [: digit:] 数字字符
- [: graph:] 非空字符（非空格、控制字符）
- [: lower:] 小写字符
- [: cntrl:] 控制字符
- [: print:] 非空字符（包括空格）
- [: punct:] 标点符号
- [: space:] 所有空白字符（新行，空格，制表符）
- [: upper:] 大写字符
- [: xdigit:] 十六进制数字（0-9, a-f, A-F）

A.3.7 实例

要用好grep这个工具，其实就是要写好正则表达式，所以这里不对grep的所有功能进行实例讲解，只列几个例子，讲解一个正则表达式的写法。

通过管道过滤ls -l输出的内容，只显示以a开头的行：

```
xxxx@xxxx:/home/xxxx$ls -l | grep 'a'
```

显示所有以d开头的文件中包含test的行：

```
xxxx@xxxx:/home/xxxx$grep 'test' d*
```

显示在aa, bb, cc文件中匹配test的行：

```
xxxx@xxxx:/home/xxxx$grep 'test' aa bb cc
```

显示所有包含每个字符串至少有5个连续小写字母的字符串的行。

```
xxxx@xxxx:/home/xxxx$grep '[a-z]\{5\}' aa
```

如：果west被匹配，则es就被存储到内存中，并标记为1，然后搜索任意个字符（.*），这些字符后面紧跟着另外一个es（\ 1），找到就显示该行。如：果用egrep或grep -E，就不用“\ ”号进行转义，直接写成'w(es)t.*\ 1'就可以了。

```
xxxx@xxxx:/home/xxxx$grep 'w\ es\ )t.*\ 1' aa
```


Appendix B

两个系统下的软件对比

不怕不识货
就怕货比货

本附录提供一个Windows和Linux下功能相近或者相同的软件的对比，使得你可以方便的转移到Linux下去。如果你想要一个更加详尽的列表，可以参考[4]。为了便于Gnome和KDE的用户，我将把这两个图形界面下的专属软件分开列举。

其中，Linux项代表不属于任何桌面系统专属的软件。

- 桌面环境

Windows Windows

GNOME GNOME

KDE KDE

- 三维桌面

Windows Windows Vista

Linux Beryl

GNOME

KDE

- 文件管理

Windows 资源浏览器, Total Commander

GNOME 文件管理器

KDE Konqueror, Dolphin

- 纯文本处理

Windows Notepad, UltraEdit

Linux GVim

GNOME GEdit

KDE Kate, KHexEdit

- 文字处理
 - Windows** Microsoft Word
 - Linux** OpenOffice文字处理
 - GNOME**
 - KDE** KWord
- 电子表格
 - Windows** Microsoft Excel
 - Linux** OpenOffice电子表格
 - GNOME**
 - KDE** KSpreadSheet
- 幻灯片制作
 - Windows** Microsoft Powerpoint
 - Linux** OpenOffice幻灯片
 - GNOME**
 - KDE** KPresenter
- 网页浏览
 - Windows** Internet Explorer
 - Linux** Firefox, Opera
 - GNOME**
 - KDE** Konqueror
- FTP工具
 - Windows** CuteFTP, LeapFTP
 - GNOME** Gftp
 - KDE** KFTPGrabber
- 日程安排
 - Windows** Outlook
 - GNOME**
 - KDE** Korganizer
- 翻译工具
 - Windows** 金山词霸
 - Linux** StarDict
 - GNOME**
 - KDE**
- 地图

Windows Google Earth

Linux Google Earth for Linux

GNOME

KDE KGeography

- 星图

Windows

GNOME

KDE KStars

- 元素周期表

Windows

GNOME

KDE Kalzium (非常专业, 以我化学物理专业的眼光看来)

- 数学计算

Windows Mathematica, MatLab, Maple

Linux Mathematica, MatLab, Mathomatics, Maxiom

GNOME

KDE

- 数据处理

Windows Opera

Linux R, Octave

GNOME

KDE

- 下载工具

Windows 迅雷, Flashget

Linux DownThemAll!

GNOME

KDE

- 语音聊天

Windows Skype

Linux Skype

GNOME

KDE

- 图形处理

Windows Photoshop

Linux GIMP

GNOME

KDE Krita

- 网上聊天

Windows MSN, Yahoo, QQ

Linux LumaQQ

GNOME Gaim

KDE Konversation, Eva, Kopete

- 歌曲播放

Windows WinAMP, 千千静听

Linux Xmms, Audacious, BeepMediaPlayer

GNOME Rhythmbox

KDE amaroK

- 电影播放

Windows Windows Media Player, 暴风影音, RealPlayer

Linux MPlayer

GNOME Totem

KDE Kaffeine

- BT下载

Windows BitTorrent, BitSpirit, BitComet

Linux Bittoradio, Azureus

GNOME

KDE QTorrent, KTorrent

- 光盘刻录

Windows Nero

GNOME Gnomebaker

KDE K3b

- 邮件收发

Windows Outlook, Foxmail

Linux Thunderbird

GNOME Evolution

KDE Kmail

- 电驴下载

Windows eMule

Linux aMule

GNOME

KDE

- C++ Programming

Windows Visual C++

Linux Code::Blocks

GNOME Anjuta

KDE Kdevelop

- PDF浏览器

Windows Adobe Acrobat Reader

Linux Adobe Acrobat Reader for Linux, xpdf

GNOME PDF阅读器

KDE KGhostView, KPDF (对中文支持不好, 不推荐)

- 简单游戏

Windows Windows游戏

GNOME GNOME游戏

KDE KDE游戏

Appendix C

如何获得帮助

求人不如求自己。
——观音菩萨

C.1 本地帮助

在终端输入`man [命令]`即可获得关于[命令]的英文帮助，因为是这些软件的开发人员写的，所以非常详尽。

如果你要一个简明的帮助，`[命令] - -help`可以助你一臂之力。而且它有可能（记住，只是有可能）是中文的。

C.2 网上求助

C.2.1 在何处求助

Ubuntu的中文官方网站是www.ubuntu.com.cn，可以在这个网站的wiki版面阅读一些资料，或者在forum版面提问。

中国科技大学校内BBS同样有Linux版面，可以在上面提问，不过回答质量一般不如上面的网站高，因为上面的网站是官方的。

此外，在www.linuxsir.org，www.chinaunix.org等网站同样可以获得很多有用的消息。

C.2.2 提问的方法

不要以“我是新手……”或者“救命啊……”这种废话作为问题的开始，没有人同情一个只会哭泣的成年人的。如果谁用这种语气提问，我的答复只有以下两个：

1. 新来的？混两年再说话。
2. 救命？先拨110或者119或者120吧。

请这样提问:

我的 $Mathematica$ 在升级 $Edgy$ 之后无法打开帮助

最近将Ubuntu从6.06版本升级到6.10版本, 发现Mathematica在打开帮助文件的时候报以下错误:

```
Warning: locale not supported by Xlib, locale set to C
```

```
Warning: X locale modifiers not supported, using default
```

```
Warning: translation table syntax error: Unknown keysym name: osfActivate
```

```
Warning: ... found while parsing 'iKey;osfActivate:PrimitiveParentActivate()'
```

```
Warning: translation table syntax error: Unknown keysym name: osfCancel
```

```
Warning: ... found while parsing 'iKey;osfCancel:PrimitiveParentCancel()'
```

然后出现段错误, 我试着重新安装, 但是仍然出现相同的情况, 以前从来没有出现过这种问题。请问问题是怎么产生的, 如何解决。

提问前请先Google一下你的问题 (抵制Baidu)。并且多尝试一下能想到的解决方案, 在问题中讲一下你所做过的努力。

还有一点要注意: 礼貌永远不过分。因为别人没有义务回答你的问题。

Appendix D

Ubuntu的历史

忘记历史就意味着背叛
——列宁

D.1 Ubuntu的简介

以下内容摘自wiki.ubuntu.org.cn，有删改：

Linux 内核，其发音为 'lee-nucks'，是 Ubuntu 操作系统的核心。内核是操作系统的重要组成部分，它是硬件和软件之间进行通讯的桥梁。Linux 诞生于1991年，由芬兰学生 Linus Torvalds 发起。那时，它只能运行在 i386 系统上，实质上是个独立编写的 UNIX 内核之克隆，旨在充分利用当时全新的 i386 架构。现如今 Linux 几乎能运行在所有现代架构之上，这要归功于来自世界各地的人们所付出的大量开发工作。Linux内核不仅在技术上占有一席之地，还在意识上占有重要位置。有一整个社群的人相信自由软件的理念，并花费大量时间帮助开源技术，使之臻于完美。正是这个社群里的人们促使了Ubuntu的出现，促成了标准委员会，它引领着互联网的发展，促成了一些组织如Mozilla基金会的出现，它负责创建了Mozilla Firefox，还促成了无数其它软件项目，想必您一直以来就受益匪浅。开放源码的精神，这通常归因于Linux，正在深刻影响着各地的软件开发者和用户，他们驱使着各个社区朝着共同的目标前进。

GNU 项目，其发音为 “guh-noo”，于1984年发起，目标是开发一个完整的 UNIX 类的操作系统——GNU 系统，它全部由自由软件组成。使用 Linux 内核的 GNU 操作系统变种已被广泛使用；尽管这些系统常冠以 “Linux”，更准确的说它们应称为 GNU/Linux 系统。GNU 项目和自由软件的理念紧密相连，它也是衍生自 GNU 的、诸如 Ubuntu 等项目的关键所在。自由软件这一概念的说明参见 第1.2节 — 自由软件。

Ubuntu 是一个南非的民族观念，着眼于人们之间的忠诚和联系。该词来自于祖鲁语和科萨语。Ubuntu（发音“oo-BOON-too”-“乌班图”）被视为非洲人的传统理念，也是建立新南非共和国的基本原则之一，与非洲复兴的理想密切相关。Ubuntu 精神的大意是“人道待人”（对他人仁慈）。另一种翻译可以是：“天下共享的信念，连接起每个人”。“具有 ubuntu 精神的人心胸开阔，乐于助人，见贤思齐而不忌妒贤能，因为他/她拥有适度的自信，而这源自如下认识：自己乃是属于一个更大的整体，当他人受到伤害或死去时，当他人受到折磨或压迫时，这个整体就会消失。”——大主教 Desmond Tutu。作

为一个基于 GNU/Linux 的平台，Ubuntu 操作系统将 ubuntu 精神带到了软件世界。

Ubuntu 的版本号是根据我们发布一个版本的日期而定。版本号由该次发布的年份和月份组成，并未反映其实际版本。我们的首次发布(Warty Warthog)是在2004年10月，因此该版本为4.10。

D.2 Ubuntu历史

日期	版本
2004.10	4.10 Warty Warthog
2005.04	5.04 Hoary Hedgehog
2005.10	5.10 Breezy Badger
2006.06	6.06 Dapper Drake
2006.10	6.10 Edgy Eft
2007.04	7.04 Festy Fawn

D.3 Ubuntu动物进化图

以下内容查不到出处：



Warty Warthog



Hoary Hedgehog



Breezy Badger



Dapper Drake



Edgy Eft



Festy Fawn

Appendix E

更新列表

千里之行
始于足下

E.1 更新说明

11.17 完成初稿。

11.18 解决了一些排版问题。添加前台，后台概念，添加磁盘缓冲概念。对短参数叠写做了说明。修改了umount命令的说明。添加对ntfs-3g命令的补充。修改了一些病句。修改了权限说明。

11.26 开始nano和vim的工作。添加“配置文件”的概念。

11.29 完成nano的工作，添加压缩解压部分。

E.2 感谢

感谢bobyang对权限问题错误的更正。

E.3 特殊感谢

感谢久石让先生和宫崎骏先生的作品。特别是创作出《幽灵公主交响组曲》的久石让先生。¹

¹我知道在科技论文中随意感谢别人是不正确的，但是这是科普论文。

Bibliography

- [1] Linux的开发人员 *Linux的man手册*
- [2] 王垠 *完全用Linux工作*
- [3] 出处未知 *提问的智慧*
- [4] 出处未知 *Windows到Linux替代软件列表*
- [5] *nano*和*vi*的帮助文件
- [6] *vim-OPL*