Chinalinuxpub.com初学版精华

KingArthur

September 17, 2003

Powered by LaTeX

Contents

1 关于

这篇文档是由www.chinalinuxpub.com论坛初学版精华区整理而成。相关文字版权属原作者所有。由使用此文档所产生的一切问题由您自己承担责任。

如果您发现这里的某些文字出自您的手笔,而您又不希望它在这里出现的话,请到论坛告诉我,或者向daizi@vip.sina.com发送邮件。我会立刻删除相关内容。如果您发现这篇文档中有什么错误的话,也可以到论坛告诉我,或者向daizi@vip.sina.com发送邮件。

2 小技巧及初学FAQ

2.1 查找系统man中所有有关某条命令的man手册

man -K string

例如

man -K httpd

如果闲输出的太多,可以修改/etc/man.config去掉man的压缩包所在的目录的路径

2.2 如何同时启动多个x

Linux里的X–Windows以其独特的面貌和强大的功能吸引了很多原先对linux不感兴趣的人，特别是KDE和GNOME，功能强大不说，而且自带了很多很棒的软件，界面非常友好，很适合于初学者。下面告诉大家一个同时启动6个X的小技巧：

在~/.bashrc中加入以下几行：

```
alias X='startx -- -bpp 32 -quiet&'
alias X1='startx -- :1 -bpp 32 -quiet&'
alias X2='startx -- :2 -bpp 32 -quiet&'
alias X3='startx -- :3 -bpp 32 -quiet&'
alias X4='startx -- :4 -bpp 32 -quiet&'
alias X5='startx -- :5 -bpp 32 -quiet&'
```

其中32是显示器的色彩深度,你应该根据自己的实际情况设置。之后运行

```
# bash
```

使改变生效，以后只要依次运行X,X1,X2,X3,X4,X5就可以启动6个X-Windows了。

2.3 文本日历

显示本月日历

```
[root@ns1 mail]# cal
```

显示某年的日历

```
[root@ns1 mail]#cal 2002
```

显示3个月的日历

```
[root@ns1 root]# cal -3
```

把周一作为一周的第一天

```
[root@ns1 root]# cal -m
```

显示Julian日历(日子是从1月1日开始累加的)

```
[root@ns1 root]# cal -j
```

2.4 启动postgresql的tcp/ip连接支持

装了rpm的postgresql之后启动

```
/etc/init.d/postgresql start
```

是不能启动postgresql的tcp/ip连接支持的,所以打开/etc/init.d/postgresql这个文件把

```
su -l postgres -s /bin/sh -c "/usr/bin/pg_ctl -D $PGDATA -
p /usr/bin/postmaster start > /dev/null
2>&1" < /dev/null
```

改为:

```
su -l postgres -s /bin/sh -c "/usr/bin/pg_ctl -o '-o -F -i'
-w -D $PGDATA -p /usr/bin/postmaster start > /dev/null
2>&1" < /dev/null
```

这样就可以启动数据库的tcp/ip链接了

2.5 如何将man转存为文本文件

以ls的man为例

```
man ls |col -b >ls.txt
```

将info变成文本,以make为例

```
info make -o make.txt -s
```

2.6 如何在文本模式下发送2进制文件

首先检查系统有没有uuencode和uudecode。如果没有从光盘上装

```
rpm -ivh sharutils-x.xx.x-x.rpm
```

假设要发送的文件是vpopmail-5.2.1.tar.gz执行

```
uuencode -m vpopmail-5.2.1.tar.gz vpopmail.tar.gz>encodefile
```

说明：

uuenode是编码命令，－m是使用mime64编码，vpopmail-5.2.1.tar.gz是要编码的文件，vpopmail.tar.gz是如果解码后得到的文件名，encodefile是编码后的文件名。执行上述命令之后就可以通过mail命令发送编码后的文件了

```
mail chenlf@chinalinuxpub.com<encodefile
```

好了，现在我来接收邮件。在控制台上输入mail命令：

```
mail
Mail version 8.1 6/6/93. Type ? for help.
"/var/spool/mail/chenlf": 2 messages 2 new
>N 1 chenlf@ns1.catv.net Mon Jun 10 16:44 17/363
N 2 root@ns2.catv.net Mon Jun 10 16:45 6091/371145
& 2
Message 2:
From root@ns2.catv.net Mon Jun 10 16:45:28 2002
Date: Mon, 10 Jun 2002 16:44:51 +0800
From: root <root@ns2.catv.net>
To: chenlf@chinalinuxpub.com

begin-base64 644 vpopmai.tar.gz
H4sIABr15TwAA+w9a2PbNpL7NfwVqNPbWIlFPSzbiR2n9SuxE7/OcuLNtdmU
EiGLMUWqfFhWt7u//eYBgKRE2U7iTa+3VndjiQQGg5nBYDAYDC6H4XDgeH51
yW7ajdpf/h2fer1VX1lagr/1+spyq/BXff5SX2mtNBZXmovN5l/qjWZrqfEX
sfRvwWbik8aJEwnxl7ifDofXlLvp/Z/0c1nk/8uN/777NuqNen251ZrB/+XF
pcUG8r/ZbC0vL9ZXoPwi/O8von73qEx//sP5bwHHxanT8aUIe2IrDBIZJLFl
7QVJFFovpZOkkYxFL4yEFhVLCKhk1W2xG45E1wnEnohlIsJAiksvSlLHF24I
JQORhKIjRdKXYhh5Ayca6xcAD8DQm4HT7XuB/EGcSXgbPErEyAkSrNp3LqVw
grGoyaRbGzpxPHJFGssotq0Gtw6l9gTgJbixode9EOlQDMaTmEjE/AerydVc
rAY4jJzIFY7vC3wL2DgJvJIxIjFwkm6fWkfw1KoAIti/EgkWc3A6YRp05ReB
aeXAQH34GoXOwAvOVUnoEnwRYRqJeJAMgczRpYzEyEv6YQoUH8oACltLtjjD
Rr1YOCJ2BkPgJop1IuJu5A0TYh9xIdQwfrCWTdt9pMKvaZg4j5jT3PgojC5+
sFZswM0LAJzvSyhGXQSCOmLoO9DtEOAicBCD2qUT1agAg44BSd+1niIEzVPs
..................
..................
..................
& s 2 encodefile
"encode" [New file]
& q
```

然后进行解码

```
uudecode encodefile
ls
```

```
encodefile vpopmai.tar.gz
tar zxvf vpopmail.tar.gz
```

OK了

2.7 将man page转成HTML格式

使用man2html这个指令，就可以将man page转成HTML格式了。用法是:

```
man2html filename > htmlfile.html
```

2.8 rpm -qf和rpm -pql

假设你想知道/bin/ls这个文件属于哪个rpm包你可以使用

```
rpm -qf /bin/ls
fileutils-4.1-10
```

所以ls是属于fileutils-4.1-10这个rpm包了

例如想知道fileutils-4.1-10.i386.rpm 有那些文件,分别安装到系统哪个目录,可以执行

```
rpm -pql fileutils-4.1-10.i386.rpm
/bin/chgrp
/bin/chmod
/bin/chown
/bin/cp
/bin/dd
/bin/df
/bin/ln
/bin/ls
/bin/mkdir
/bin/mknod
/bin/mv
/bin/rm
/bin/rmdir
/bin/sync
/bin/touch
/etc/DIR_COLORS
/etc/profile.d
/etc/profile.d/colorls.csh
/etc/profile.d/colorls.sh
/usr/bin/dir
/usr/bin/dircolors
/usr/bin/du
```

```
/usr/bin/install
/usr/bin/mkfifo
/usr/bin/shred
/usr/bin/vdir
/usr/share/doc/fileutils-4.1
.......
.......
```

上述命令仅对rpm包安装的软件有效

2.9 在文本方式下自动mount光区和软盘

linux下使用广盘软盘都必须要先mount很不方便,大家可以使用autofs来自动mount自己的广盘和软盘. 具体步骤如下: 1.修改/etc/auto.misc

```
# $Id: auto.misc,v 1.2 1997/10/06 21:52:04 hpa Exp $
# This is an automounter map and it has the following format
# key [ -mount-options-separated-by-comma ] location
# Details may be found in the autofs(5) manpage

cd -fstype=iso9660,ro,nosuid,nodev :/dev/cdrom

# the following entries are samples to pique your imagination
#linux -ro,soft,intr ftp.example.org:/pub/linux
#boot -fstype=ext2 :/dev/hda1
#floppy -fstype=auto :/dev/fd0
#floppy -fstype=ext2 :/dev/fd0
#e2floppy -fstype=ext2 :/dev/fd0
#jaz -fstype=ext2 :/dev/sdc1
#removable -fstype=ext2 :/dev/hdd
```

我们只需要修改里面的cd和floppy部分:

```
# $Id: auto.misc,v 1.2 1997/10/06 21:52:04 hpa Exp $
# This is an automounter map and it has the following format
# key [ -mount-options-separated-by-comma ] location
# Details may be found in the autofs(5) manpage

cdrom -fstype=iso9660,ro,nosuid,nodev :/dev/cdrom

# the following entries are samples to pique your imagination
#linux -ro,soft,intr ftp.example.org:/pub/linux
#boot -fstype=ext2 :/dev/hda1
floppy -fstype=auto :/dev/fd0
floppy -fstype=ext2 :/dev/fd0
e2floppy -fstype=ext2 :/dev/fd0
```

```
#jaz -fstype=ext2 :/dev/sdc1
#removable -fstype=ext2 :/dev/hdd
```

然后修改/etc/auto.master 原来的文件是

```
# $Id: auto.master,v 1.2 1997/10/06 21:52:03 hpa Exp $
# Sample auto.master file
# Format of this file:
# mountpoint map options
# For details of the format look at autofs(8).
#/misc /etc/auto.misc --timeout=60
```

去掉最后一行前面的注释

```
# $Id: auto.master,v 1.2 1997/10/06 21:52:03 hpa Exp $
# Sample auto.master file
# Format of this file:
# mountpoint map options
# For details of the format look at autofs(8).
/misc /etc/auto.misc --timeout=60
```

其中60是说如果没有任何操作则在60妙后自动umount 好了,重新起动autofs

```
chkconfig --levels 345 autofs on
/etc/init.d/autofs restart
```

测试: 放张广盘到光区然后执行

```
cd /misc/cdrom
ls
```

就可以看到广盘里的内容了如果是软盘则执行

```
cd /misc/floppy
ls
```

2.10 用xvidtune调整你的显示器

大家会发现装了linux之后在windows下用的好好的显示器有时进到linux的xwindows里后就歪掉了，调整好之后回到windows后windows的桌面也外调了，来回启动系统每次都要调整很麻烦的，这里介绍一个办法一劳永逸

进入linux启动x在xterm里执行xvidtune,会弹出这个软件的窗口，点Auto然后点Left,Right等按钮调整你的显示器到最佳的位置，然后点界面上的Show按钮会得到类似这样的输出：

```
"1152x864" 121.50 1152 1232 1360 1568 864 865 868 911 +hsync
 -vsync
```

然后退出这个软件，修改你的/etc/X11/XF86Config-4文件在

```
Section "Monitor"
Identifier "AS 786T"
VendorName "Unknown"
ModelName "Unknown"
HorizSync 30 - 87
VertRefresh 50 - 160
Option "dpms"
EndSection
```

里加上刚才的输出，我的是：

```
Section "Monitor"
Identifier "AS 786T"
VendorName "Unknown"
ModelName "Unknown"
HorizSync 30 - 87
VertRefresh 50 - 160
Modeline "1152x864" 121.50 1152 1232 1360 1568 864 865 868
911 +hsync -vsync[1]

Option "dpms"
EndSection
```

保存然后重起试试看吧

2.11 将man转换为文本文件

以ls为例

```
$ man ls |col -b> ls.txt
```

然后就可以用任何文本浏览器打开ls.txt查看里面的内容了

2.12 根据文件扩展名鉴别文件类型

对于新手来说，对linux的文件扩展名可能会很困惑，文件扩展名是文件名的最后，位于最后一个“.”号的后面的那部分(在文件sneakers.txt里，"txt"是这个文件的扩展名).

下面是文件扩展名的简要列表和他们的说明:

.bz2 bzip2压缩文件

.gz gzip压缩文件

.tar tar打包文件

[1]注意:这里实际上和上一行(Modeline.....)是同一行,由于排版的原因在这里折成两行

.tbz tar打包并用bzip压缩的文件

.tgz tar打包并用gzip压缩的文件

.au audio文件

.gif GIF图象文件

.html/.htm HTML文件

.jpg JPEG图象文件

.pdf 文档的电子图象; PDF代表Portable Document Format

.png PNG图象(Portable Network Graphic的缩写)

.ps PostScript文件; 打印格式文件

.txt 纯文本文件

.wav audio文件

.xpm 图象文件

.conf 配置文件

.lock lock文件; 用来判断一个程序或者设备是否在被使用

.rpm Red Hat Package Manager文件，用来安装程序，软件和脚本文件

.c c源程序代码文件

.cpp C++源程序代码文件

.h C或者C++程序的头文件

.o 程序目标文件

.pl Perl脚本

.so 类库文件

.tcl TCL脚本

对于那些不常用或者没有扩展名的文件，如何判断它的类型呢？我们可以使用file命令，例如：一个名为saturday的文件没有扩展名。使用file命令，我们可以知道这个文件的文件类型：

```
$ file saturday
saturday : ASCII English text
```

在这个例子中。命令file saturday显示出这个文件是一个ASCII文本文件.任何文本文件都可以用诸如cat,more或者less以及文本编辑器来查看或者编辑

更多帮助请参考

```
man file
```

2.13 给一个文件标记上行号

使用nl命令。例如:

```
[root@ns2 root]# nl /etc/passwd
1 root:x:0:0:root:/root:/bin/bash
2 bin:x:1:1:bin:/bin:/sbin/nologin
3 daemon:x:2:2:daemon:/sbin:/sbin/nologin
4 adm:x:3:4:adm:/var/adm:/sbin/nologin
5 lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
6 sync:x:5:0:sync:/sbin:/bin/sync
7 shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
8 halt:x:7:0:halt:/sbin:/sbin/halt
9 mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
10 news:x:9:13:news:/var/spool/news:
11 uucp:x:10:14:uucp:/var/spool/uucp:/sbin/nologin
...
...
```

给输出的行号后加个点例如1. 2.

```
[root@ns2 root]# nl /etc/passwd -s '. '
1. root:x:0:0:root:/root:/bin/bash
2. bin:x:1:1:bin:/bin:/sbin/nologin
3. daemon:x:2:2:daemon:/sbin:/sbin/nologin
4. adm:x:3:4:adm:/var/adm:/sbin/nologin
5. lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
6. sync:x:5:0:sync:/sbin:/bin/sync
7. shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
8. halt:x:7:0:halt:/sbin:/sbin/halt
9. mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
10. news:x:9:13:news:/var/spool/news:
11. uucp:x:10:14:uucp:/var/spool/uucp:/sbin/nologin
...
...
```

2.14 将一个文件中的所有大写字母换成小写

```
cat filename |tr "[A-Z]" "[a-z]">filename2
```

例如把passwd文件全部转换为大写:

```
cat /etc/passwd|tr "[a-z]" "[A-Z]">passwd
```

2.15 计算一个文件有几行

```
wc -l filename
```

例如

```
wc -l /etc/passwd
41 /etc/passwd
```

2.16 开机默认进入图形界面

修改/etc/inittab文件，其中有一行id:3:initdefault，意思是说开机默认进入运行级别3（多用户的文本界面），把它改成id:5:initdefault，既开机默认进入运行级别5（多用户的图形界面）。这样就行了。

2.17 重装windows后丢失grub起动菜单

重装windows后，windows重写了你的mbr，覆盖掉了grub。解决方法很简单：用你的linux第一张安装盘引导进入linx rescue模式（在boot:提示符下输入linux rescue），执行下面两条命令就可以了

```
chroot /mnt/sysimage #改变你的根目录
grub-install /dev/hda #安装grub到mbr
```

2.18 在linux中使用windows分区

先说一点背景知识。linux支持很多种文件系统，包括windows的fat32和ntfs。对fat32的支持已经很好，可以直接使用，而对ntfs的支持还不是太好，只能读，而写是极危险的，并且对ntfs的支持不是默认的，也就是说你想要使用ntfs的话，需要重新编译内核。鉴于重编内核对于新手的复杂性，这里只讲解使用fat32分区的方法下面给出上述问题的两种解决方案：

1.在安装系统（linux），进行到分区选择挂载点时，你可以建立几个挂载点，如/mnt/c，/mnt/d等，然后选择你的windows fat32分区，把它们分别挂载到前面建立的挂载点即可。（注意，正如前面所说，在这里你不能把一个ntfs分区挂载到一个挂载点，应为ntfs不是默认支持的。）这样你装好系统后就能直接使用你的windows fat32分区了。例如，你把windows的c盘（linux中的/dev/hda1）挂载到/mnt/c，那么你就能在/mnt/c目录中找到你的c盘中的全部数据。

2. 如果你在安装系统时没有像方案1所说的那样挂载上你的fat32分区，没关系，仍然能够很方便的解决这个问题。首先，用一个文本编辑器（如vi）打开/etc/fstab，在文件的最后加入类似如下的几行

```
/dev/hda1 /mnt/c vfat default 0 0
```

你所要做的修改就是，把/dev/hda1改成你要挂载的fat32分区在linux中的设备号，把/mnt/c改成相应的挂载点即可。注意，挂载点就是一个目录，这个目录要事先建立。举一个例子，我有三个fat32分区，在windows中

是c,d,e盘，在linux中的设备号分别为/dev/hda1,/dev/hda5,/dev/hda6。那么我就要先建立3个挂载点，如/mnt/c,/mnt/d,/mnt/e，然后在/etc/fstab中加上这么几行：

```
/dev/hda1 /mnt/c vfat default 0 0
/dev/hda5 /mnt/d vfat default 0 0
/dev/hda6 /mnt/e vfat default 0 0
```

保存一下退出编辑器。这样以后你重启机器后就能直接使用c,d,e这三个fat32格式的windows分区了

2.19 在linux中如何安装软件

linux中的软件安装比windows中略微麻烦一些。你获得的软件包大体可分为两类，一类是像windows中一样已经做好的二进制安装文件，主要有.bin和redhat的.rpm两种两种格式。对于.rpm包，一般在x中直接双击即可安装；对于.bin文件，一般先要用chmod o+x xxxx.bin设置它的可执行属性，然后直接双击或在命令行中输入./xxxx.bin即可安装。这一类软件包的安装和windows一样简单。而另一类，也是unix系统下使用最广泛的软件发布方法，则是提供打包的程序源代码，文件常以.tar.gz结尾。要安装以这种方式发布的软件，你首先要把获得的源代码解包，然后根据源代码所附的文档的说明来编译，安装。对新手来说，这一类是非常头疼的。所以，我建议新手需要安装什么软件时，尽量去找.rpm包或.bin包，等过一段时间，对系统比较熟悉后再去试着找源代码安装。但请注意，可以说所有的软件都有第二类发布，但不一定都有第一类发布，所以学会通过源代码包来安装程序是极其重要的基本技能，千万不要把它扔掉

2.20 rh8.0/rh9.0中的XMMS不能播放MP3

是因为rh公司怕别人告他侵权，所以在rh8.0中去掉了XMMS对MP3的支持，8.0以前的版本都是好使的。在8.0中要解决也很简单，装一个插件就行了。此插件可以到www.chinalinuxpub.com论坛下载

2.21 访问windows分区时发现中文全变成问号

在 2.18 节中我们讲解了通过编辑/etc/fstab实现在linux中访问windows的fat32分区。同样，我们可以通过进一步修改/etc/fstab来实现中文文件名显示。只要把/dev/hda1 /mnt/c vfat default 0 0中的default全改为iocharset=cp936就行了

2.22 如何卸载linux

很简单。直接找个分区软件把linux分区格掉就行了。注意，如果你的机器上还装有windows，而且引导装载程序用的是grub的话，你还需要在格式化linux之前fdisk /mbr，清除掉MBR中的grub

2.23 安装软件包是提示缺少.so文件

.so文件就像windows中的.dll文件一样，是库文件。一个程序的正常安装和运行需要特定的库文件的支持。所以你需要去找到包含这个.so的包装上。去www.rpmfind.net用你缺的那个.so文件的名字为关键字搜一搜，你会找到所需要的.rpm包

2.24 linux下如何使用QQ

你需要下载对应于你gaim版本的QQ插件.可以到www.chinalinuxpub.com论坛或者linuxfans下载

2.25 文本控制台出现乱码

这是因为你安装了中文支持的缘故。解决的方法是安装一个zhcon(一个快速地外挂式CJK(中文/日文/韩文)的多内码平台)。关于zhcon的更进一步的消息，大家可以到他的官方主页zhcon.gnuchina.org查看。安装和使用请参考这个连接`http://hepg.sdu.edu.cn/Service/tips/zhcon_manual.html`

2.26 改变grub默认启动的系统

这需要修改/boot/grub/grub.conf。举一个例子你就明白了。假设你的/boot/grub/grub.conf是这样子的：

```
default=0
timeout=10
splashimage=(hd0,7)/grub/splash.xpm.gz
title Red Hat Linux (2.4.18-14)
root (hd0,7)
kernel /vmlinuz-2.4.18-14 ro root=LABEL=/
initrd /initrd-2.4.18-14.img
title DOS
rootnoverify (hd0,0)
chainloader +1
```

那么你的grub会默认启动Red Hat Linux (2.4.18-14)这个系统，把default=0改成default=1，那么grub会默认启动DOS这个系统。注意，这里的要点是：你想默认启动第n个title所指的系统，那么default应该是等于n-1

2.27 定制VI

在每个用户的目录下,都有一个vi 的配置文件`".vimrc"`(没有的话可以自己创建). 用户可以编辑它,使这些设置在每次启动vi 时,都有效. 例如,加入如下设置行:

```
set nu #显示行号
set nonu# 不显示行号
set ic #查找时不考虑大小写
set noic# 查找时考虑大小写
set smartindent #自动缩进
```

2.28 grub使用小技巧

我的gurb.conf文件是这样的:

```
default=1
timeout=10
splashimage=(hd0,4)/grub/my.xpm.gz
title Red Hat Linux (2.4.20-8.ipx)
        root (hd0,4)
        kernel /vmlinuz-2.4.20-8.ipx ro root=LABEL=/
        initrd /initrd-2.4.20-8.ipx.img
title Windows
        rootnoverify (hd0,0)
        chainloader +1
```

1. 给grub加口令

   使用grub-md5-crypt程序可以把一个字符串用md5加密来供grub使用.在上述文件的第三行(splashimage=(hd0,4)/grub/my.xpm.gz)后新加一行:

   ```
   password --md5 $1$PtzMx/$IB2QhN7SyZWH3WTXmp2P50
   ```

   其中`$1$PtzMx/$IB2QhN7SyZWH3WTXmp2P50`是123 经grub-md5-crypt加密后得到的加密口令.这样,如果想要在gurb起动菜单使用e命令和c命令,必需首先使用p命令来输入口令,这里即为123 .

2. 给起动项加锁

   让我们继续,在"title Windows"这行后面新加一行:

   ```
   lock
   ```

   这样在grub的起动画面想要选择title Windows 这一项时会提示你输入口令,即上面设定的123 .

3. 不同的起动项使用不同的口令

   如果想不同的起动项使用不同的口令,如title Windows 项使用口令tcp , title Red Hat Linux (2.4.20-8.ipx) 项使用口令ipx ,怎么办?好办,在title Red Hat Linux (2.4.20-8.ipx) 这一行后新加一行:

   ```
   password --md5 $1$Y13Nx/$JBRWp5p5AquDAQnE8nlnN1
   ```

这里的1Y13Nx/$JBRWp5p5AquDAQnE8nlnN1 是ipx 经grub-md5-crypt加密后得到的加密口令对于title Windows 这一项也可以按同样的方法完成,但是注意首先要把开始加的lock 这行删除.

好了,现在重起机器试一下吧

2.29 让Mozilla支持flash

到http://www.chinalinuxpub.com/vbbforum/attachment.php?s=&postid=120855下载插件包,解包后进入目录执行./flashplayer-installer按提示操作即可

注:

1. 对redhat8.0来说,mozilla的installation path是/usr/lib/mozilla-1.0.1,而9.0应该是/usr/lib/mozilla-1.2.1
2. 包中的readem文件中有安装说明

2.30 如何使用md5sum

md5sum是校验文件完整性的工具,在网上下载的iso文件一般都带有带的md5sum文件. 假设你的iso文件为/tmp/my.iso,md5sum文件为/tmp/md5sum，那么：

```
cd /tmp
md5sum -c md5sum
```

如果输出是:

```
my.iso:ok
```

那么你的iso文件就没有问题，否则就要重下了

2.31 引导进入单用户模式

单用户模式的优越性之一是你不必使用引导软盘或引导光盘；不过，它仍旧给你提供了把文件系统挂载为只读模式或干脆不挂载这两种选择。

在单用户模式中，你的计算机引导入运行级别1。你的本地文件系统被挂载，但是你的网络不会被激活。你有一个可用的系统维护shell。和救援模式不同，单用户模式会自动试图挂载你的文件系统；如果你的文件系统无法被成功挂载，不要使用单用户模式。如果你的系统上的运行级别1 的配置被损坏，你就不能使用单用户模式。

如果你的系统引导了，但是在引导后却不允许你登录，你可以试着使用单用户模式。

如果你使用的是GRUB，使用以下步骤来引导入单用户模式:

1. 如果你配置了GRUB 口令，键入p 并输入口令。

2. 选择带有你想引导的内核版本的Red Hat Linux，然后键入e 来编辑。你会看到用于所选卷标的配置文件中的一个项目列表。

3. 选择起首为kernel 的行，然后键入e 来编辑那一行。

4. 转到行尾，然后键入single（按[空格]键，然后键入single）。按[Enter]来退出编辑模式。

5. 回到了GRUB 屏幕后，键入b 来引导入单用户模式。

如果你使用的是LILO，在LILO 引导提示（如果你使用的是图形化LILO，你必须按[Ctrl]-[x] 来退出图形化屏幕后再进入boot: 提示）后键入：

linux single

2.32 inux下如何解压rar包

到http://www.chinalinuxpub.com/vbbforum/attachment.php?s=&postid=126614下载rar for linux。解包后进入目录直接make install就可以用了(rar,unrar....)。File roller也能解rar包了

2.33 以另一个用户身份使用GUI应用程序

当我们使用su 命令时，您可能已经注意到了，我们只运行了那些在终端窗口中显示输出的命令。根据Linux 发行版，您可能必须要采取一些额外的步骤以便能运行GUI 应用程序。Linux 上的GUI 应用程序使用X 窗口系统，它旨在允许多个用户使用视窗化的应用程序通过网络访问一台计算机。对于只有单个用户的Linux 系统来说，适宜采取以下步骤，但是要记住X 窗口系统的网络继承性，那样就不会无意中暴露您的系统并且允许任意的网络用户在您的系统上打开窗口。

X 显示器是通过这种格式“主机名:显示器号.屏幕号”的名字来知晓的。对于运行在诸如PC 这样的工作站上的Linux 来说，通常只有一个带单一屏幕的显示器。在这种情况下，显示器名称可能会，也往往会被省略掉，因此显示器就是所谓的:0.0。我们假设您阅读本文时正在使用图形界面登录，那么您的启动程序应该已经为您设置了DISPLAY 环境变量。但是，当您使用su 切换到另一个用户时，并不会设置该变量。清单5 显示了如何使用echo 命令来显示DISPLAY 环境变量，接着显示了如何切换到另一个用户并设法启动xclock 应用程序，开始使用空白的DISPLAY 环境变量，然后使用被设置成:0.0 的值。

设法启动xclock

```
ian3@echidna:~> whoami
ian3
ian3@echidna:~> echo $DISPLAY
:0.0
ian3@echidna:~> su - db2inst1
```

```
Password:
db2inst1@echidna:~> echo $DISPLAY

db2inst1@echidna:~> xclock
Error: Can't open display:
db2inst1@echidna:~> DISPLAY=:0.0
db2inst1@echidna:~> export DISPLAY
db2inst1@echidna:~> echo $DISPLAY
:0.0
db2inst1@echidna:~> xclock
Xlib: connection to ":0.0" refused by server
Xlib: Client is not authorized to connect to Server
Error: Can't open display: :0.0
db2inst1@echidna:~>
```

让我们了解一下这里正在发生什么。在这个例子中，用户ian3 登录到系统，将其DISPLAY 环境设置成如我们期望的:0.0。当ian3 切换到用户db2inst1 时，并未设置DISPLAY 环境变量，启动xclock 的尝试失败，因为该应用程序不能打开显示器。

因此替代用户db2inst1 设置了DISPLAY 环境变量并将其导出，这样它就可以用于可能从这个终端窗口启动的其它shell 了。请注意，设置和导出环境变量没有使用前导的fl$w3O\text{œ}/\hat{}TK\hat{}\Omega T$",fl=°$\Omega Cxclock E$,"$\Omega$"

第二次失败的原因在于X 的客户机 / 服务器本质。虽然db2inst1 是在该系统仅有的显示器上的窗口中运行，但该显示器实际上还是归最初登录的用户（在本例中是ian3）所有。对ian3 而言，这个问题最简单的解决方案是：使用xhost 命令使该系统上的其他用户能使用这个显示器。在桌面上打开另一个终端窗口并输入这个命令：

```
xhost +local:
```

请注意尾部的冒号（:）。这将允许同一系统上的其他用户连接到X 服务器并打开窗口。在给出的示例中，运行db2inst1 的窗口现在可以启动xclock 或其它X 应用程序。

有关使用xhost 的更多详细信息，可以使用info xhost 或man xhost 命令来查看联机手册页。如果您对X 连接的安全性感兴趣，那么请从Xsecure 手册页开始入手。

2.34 在文字模式下设置刷新率

在X的终端下执行:

```
xrandr -s 1024x768 -r 75
```

其中：1024x768是分辨率，75是刷新频率。是马上生效的。

2.35 使用chattr给文件加上写保护

许多用户会无意删除重要的文件。用上chattr 命令后，我们就能保护自己重要的文件了。chattr 命令改变文件属性为 "i" ，我们就能写保护象/etc/fstab, /etc/passwd, /etc/group等重要文件了。

例如：

```
$ chattr +i filename
```

这个例子演示了保护 "filename" 使之不能被修改。去掉保护只要用chattr -i filename 就可以了。

chattr还有许多其他用法,希望大家能自己man一下看看

2.36 pidof命令

对于在 'ps aux' 中寻找程序的PID 感到厌倦了吗？命令pidof 查找指定程序的PID，并且用普通文本格式输出。加上-x 参数，包含shell 脚本的pid。

例如：

```
$ pidof -x bash
1138 984 597
```

如果要用一条命令杀掉一个守护进程或者脚本的话，可以用以下的命令：

```
$ kill -9 $(pidof -x pico)
```

2.37 制作启动盘

输入下面的命令然后按提示操作即可

```
mkbootdisk --device /dev/fd0 2.4.x
```

这里2.4.x是你当前所用的内核的完整版本号

2.38 用命令弹出/收回光驱托盘

弹出:eject
收回:eject -t
挺好玩的,呵呵:)

2.39 使用ISO文件

使用下面的命令:

```
mount -o loop path_of_iso path_of_mount_dir
```

2.40 在RedHat中配置ADSL

两种方法,任选一种:

1. redhat-config-network—-¿新建—¿xDSL连接—-¿填写帐号信息
2. adsl-setup—-¿输入adsl的用户名—-¿使用的网络接口(只有一张网卡的话就是eth0了,直接回车即可)—-¿选择no,直接回车—-¿输入DNS地址—-¿输入adal帐户的口令,不回显—¿是否允许普通用户控制此连接,选yes吧—¿选择防火墙等级—-¿检察无误后输入y然后回车结束配置.

Type '/sbin/ifup ppp0' to bring up your xDSL link and '/sbin/ifdown ppp0' to bring it down. Type '/sbin/adsl-status /etc/sysconfig/network-scripts/ifcfg-ppp0' to see the link status.

如果仍然无法上网,请仔细检查:

1. DNS是否设好?
2. 网关是否设好?
3. 是否有硬件问题?

2.41 文本下支持图片显示的浏览器

redhat8,9带了一个很好的浏览器,如果你的console开启了帧缓冲的话(framebuffer)输入:

```
w3m http://www.chinalinuxpub.com
```

开启帧缓冲只需要在/etc/grub.conf里加上vga=xxx例如

```
kernel /vmlinuz-2.4.20-18.7smp ro root=/dev/sda2 vga=791
```

791表示1024x768的分辨率有关分辨率的调节查看

```
/usr/src/linux-2.4/Documentation/vesafb.txt
```

下面是分辨率的对照表

	640x480	800x600	1024x768	1280x1024
256	0x301	0x303	0x305	0x307
32k	0x310	0x313	0x316	0x319
64k	0x311	0x314	0x317	0x31A
16M	0x312	0x315	0x318	0x31B

可以看出,791其实就是0x317的10进制表示形式

就可以得到显示网页图片的文本浏览器了,当然要支持中文你还需要装个zhcon

2.42 利用Nautilus中右键的脚本子菜单

在~/.gnome2/nautilus-scripts目录下建立一个脚本文件test.sh,内容如下:

```
#!/bin/bash
dir=‘dirname $NAUTILUS_SCRIPT_SELECTED_FILE_PATHS‘
cd $dir
tar zxvf $NAUTILUS_SCRIPT_SELECTED_FILE_PATHS
```

然后给它可执行权限

```
chmod +x test.sh
```

在Nautilus中用右键点击一个.tar.gz文件,选择脚本–¿test.sh,然后,你发现了什么? 呵呵,这个文件被解包了很方便吧:)

上面只是一个很简单的例子.实际上~/.gnome2/nautilus-scripts目录下的所有可执行文件都将显示在脚本菜单中。从该菜单中选择某脚本将会执行那个脚本。利用这一点,你可以给自己带来很多方便.

书写这种脚本时注意:

1. 当从本地文件夹中执行时，选中的文件名将会被传递给脚本；当从远程文件夹中执行时(如显示万维网或ftp 内容的文件夹)，脚本将不会使用任何参数。在各种情况下，下列脚本可以使用的环境变量会被Nautilus 设置：
   - NAUTILUS_SCRIPT_SELECTED_FILE_PATHS：用新行分开的所选文件的路径(除非是本地文件)
   - NAUTILUS_SCRIPT_SELECTED_URIS：用新行分开的所选文件的URI
   - NAUTILUS_SCRIPT_CURRENT_URI：当前位置的URI
   - NAUTILUS_SCRIPT_WINDOW_GEOMETRY：当前窗口的位置和大小
2. 这种脚本的默认工作目录是 /,而不是你在Nautilus中选中的文件所在目录

最后再贡献一个打包的脚本:

```
#!/bin/bash
dir=‘dirname $NAUTILUS_SCRIPT_SELECTED_FILE_PATHS‘
file=‘basename $NAUTILUS_SCRIPT_SELECTED_FILE_PATHS‘
cd $dir
tar czvf $file.tar.gz $file
```

2.43 如何安装xxxxxx.src.rpm这样的包

这种包是源代码rpm包,如果直接用rpm -ivh来安装的话,会在/usr/src/redhat/SOURCES目录下找到一个tar.gz打包的源代码包.也就是说需要你自己手工解包编译安装.但是可以直接用:

```
rpmbuild --rebuild xxxxx.src.rpm
```

来直接把源代码rpm包编译成普通的二进制rpm包.执行上述命令后,可以到/usr/src/redhat/RPMS/i386目录下找到可用的二进制rpm包.

2.44 快速进入某些目录

键入cd 可进入用户的home目录. 键入cd – 可进入上一个进入的目录.

2.45 XMMS播放列表不能显示中文

打开XMMS--->按ctrl+p--->选择fonts--->把两个勾都打上--->在两个文本框里都填上:
-adobe-helvetica-bold-r-*-*-14-*,-isas-fangsong ti-medium-r-normal-*-*-160-*-*-c-*-gb2312.1980-0
点击确定

2.46 任何共享上网?

假设网络拓扑结构如下:

```
internet<=====>eth0{localhost}eth1<=====>LAN[192.168.0.0/24]
```

那么localhost上需要执行:

```
echo 1 > /proc/sys/net/ipv4/ip_forward
iptables -t nat -A POSTROUTING -o eth0 -s 192.168.0.0/24 -j
 MSQUERADE
```

然后LAN上的机器把网关指向eth1,再设置好DNS就行了

2.47 删除文件名以连字符- 开始的文件

如,某个文件为-foo .如何删除呢?执行下面的命令:

```
rm -- -foo
```

或者

```
rm ./-foo
```

2.48 如何制作iso映象

使用下面的命令:

```
mkisofs -o test.iso -Jrv -V test_disk /home/carla/
```

在上面的示例中：

-o 为新的.iso 映像文件命名（test.iso）

-J 为了与Windows 兼容而使用Joliet 命名记录

-r 为了与UNIX/Linux 兼容而使用Rock Ridge 命名约定，它使所有文件都公共可读

-v 设置详细模式，以便在创建映像时获得运行注释

-V 提供了卷标识（test_disk）；该标识就是出现在Windows 资源管理器中的盘名

/home/carla 列表中的最后一项是选择要打包到.iso 中的文件（都在/home/carla/中）

2.49 让你的程序在你logout后仍然继续执行而不中断

```
nohop your_prog&
logout
```

2.50 使用head 和tail 以块方式读取文本流

假定您想只处理文件的一部分，譬如头几行或后几行，那您该怎么做呢？请使用head（它将头10 行发送至标准输出）或tail（它将后10 行发送至标准输出）。

您可以通过使用-n 选项改变这些命令发送至其标准输出的行数（当然，输出结果将随XF86Config 文件的内容而不同）：

将XF86Config 中选定行数的内容发送至标准输出

```
$ head -n 4 /etc/X11/XF86Config
# File generated by anaconda.
# ***************************************
# Refer to the XF86Config(4/5) man page for details about
# the format of this file.
$ tail -n 4 /etc/X11/XF86Config
Modes       "1600x1200"
ViewPort    0 0
EndSubsection
EndSection
```

如果您想让head 或tail 以字节而不是以行为单位，那该怎么办呢？您可以用-c 选项代替-n 选项。因此，要显示前200 个字符，请使用head -c 200 file，或者使用tail -c 200 file 来显示后200 个字符。如果数字后面跟有b（表示块（block）），那么这个数字将被乘以512。类似地，跟有k（表示千字节（kilobyte））表示用1024 去乘给定的数字，而跟有m（表示兆字节（megabyte））表示用1048576 字节去乘给定的数字。

请记住，head file1 file2 file3 和cat file1 file2 file3 — head 之间有重大差别。前者将打印每个文件指定行数的内容，不同文件的内容之间用头信息隔开，头信息以==¿ 后跟文件名开头。后者将打印由cat 命令后所列文件组成的输入流中指定行数的内容，但将把输入流作为单个文件对待。可以使用-q（表示静默（quiet））选项关闭文件名头信息。与-q 相反的是-v（表示详列（verbose））。

假如您要处理的文件在处理期间一直在发生变化（比如，当您让head 或tail 读取来自正在被另一个命令写入的文件的数据时，就是这种情况），请使用-f 选项让tail 持续读取来自指定文件的数据并将这些数据发送至tail 自己的标准输出中。通过管道发送数据时该选项会被忽略。因此，cat file — tail -f 将不会得到所期望的结果，但tail -f file 则可以。

（如果tail 正在读取的文件不止一个，那么各行内容之间将用标准头信息隔开，以指明它们来自哪个文件，标准头信息以==¿ 开头。）

这个选项用于监视系统日志再合适不过，譬如，在单独的终端窗口（或单独的控制台）中执行的tail -f /var/log/access.log 将持续打印每次点击后新添加的Apache 访问日志条目，一直到您用Ctrl-C 停止它为止。

通过组合使用head 和tail，可以从文件的中间部分读取给定长度的一块数据！下面说明如何做到：假定您想从文件开头算起第1000 字节处开始读取一块789 字节的数据。可以使用cat file — head -c 1788 — tail -c 789 来解决这一问题。

2.51 使用tac反向输出文件内容

```
[root@Ash root]# cat .bashrc
# .bashrc
# User specific aliases and functions
# Source global definitions
if [ -f /etc/bashrc ]; then
        . /etc/bashrc
fi
[root@Ash root]# tac .bashrc
fi
        . /etc/bashrc
if [ -f /etc/bashrc ]; then
# Source global definitions
# User specific aliases and functions
# .bashrc
```

注意到了吗?cat反过来写就是tac:)

2.52 字符串截断

```
$myVAR="foodforthought"
$echo ${myVAR##*fo}
rthought
$echo ${myVAR#*fo}
odforthought
```

简单的说,${myVAR##*fo}的值是这么得到的:找到从字符串foodforthought (即$myVAR的值) 开始处开始、且匹配通配符 *fo 的最长子字符串，然后将其从字符串foodforthought 的开始处截去.

${myVAR#*fo}和${myVAR##*fo}类似,但他是除去最短的匹配.

使用 %% 和使用 ## 类似,但他是从字符串末尾除去匹配的字符.同样, % 和 # 类似.

还可以使用:${myVAR:m:n}.他的值是从$myVAR的第m个字符开始的连续n个字符.

2.53 特殊的管道操作命令tee

tee命令把标准输入写入指定的文件,同时拷贝一份发送到标准输出.这样,我们就可以在把一个程序的输出保存到一个文件的同时在屏幕上实时观察输出:

```
make isntall | tee foo
ls -l | tee bar | grep ^foobar
```

2.54 读出文件前n行的3种方法

第一种方法

```
head -5 filename
```

第二种方法

```
awk 'NR==1,NR==5 {print $0}' filename
```

第三种方法

```
sed -n '1,5p' filename
```

2.55 vi中的拷贝

常用的copy和paste命令都是在命令模式（按Esc进入）下使用的。举几个例子：

- 拷贝当前行:yy

- 拷贝当前行到第15行:y15G
- 拷贝当前行和下一行:yj
- 拷贝当前行和下两行:y2j

实际上,j命令是把光标向上移动一行,15G是把光标移到第15行
如果想要拷贝一段不规则的文本（即不是完整的几行），可以在命令模式下按v键进入可视模式，使用方向键选择你想拷贝的那一段文本，然后按y键，这段内容就拷贝下来了。
拷贝完成后,把光标移到想要粘贴的地方，按下p键即可粘贴命令注意大小写喔。

2.56 修改grub的开机背景

很简单,只要作一个640x480的图片,并且将图片的色彩改为14bit即可。可以用gimp来完成:
gimp–¿大开文件–¿鼠标右键–¿图像–¿模式–¿索引–¿颜色数改为14
图像–¿缩放图像为640x480
然后另存为xpm,假设文件名为grub.xpm。执行gzip grub.xpm然后复制到/boot/grub/
修改grub.conf:

splashimage=(hd0,6)/boot/grub/grub.xpm.gz

然后重启看看吧

2.57 了解系统状况

uname –显示系统信息，-a显示系统完整鉴定信息，包括主机名，核心版本等
hostname –显示主机名，也可以临时修改主机名。如确实要更换主机名，编辑/etc/sysconfig/network
last –列出最近的用户登录，包括用户名，来自，登录时间
lastlog –列出每一个用户的最近登录情况
free –显示内存使用状况
top –显示监视器，q退出，gtop为图形界面下的一个优化版本。

2.58 如何将中文默认系统改为英文的

修改/etc/sysconfig/i18n为

```
LANG="en_US.iso885915"
SUPPORTED="en_US.iso885915:en_US:en"
SYSFONT="lat0-sun16"
SYSFONTACM="iso15"
```

退出重新登陆.如果是gnome不需要修改就是英文,如果是kde需要在控制中心里选择语言为默认C

2.59 如何才能允许root远程登陆

在别的机器上telnet redhat主机时，root不允许登陆,怎么修改才能允许呢？实际上,只需注释掉/etc/pam.d/login中的

auth required /lib/security/pam$_{s}ecuretty.so$ 一行即可。

2.60 使用dd制作大体积文件

使用dd的bs和count选项即可:

```
dd if=/dev/zero bs=1M count=100000|bzip2>foo.bz2
```

这样就制作了一个100000M的文件,并用bzip2进行了压缩,最后得到的文件是foo.bz2.体积大概只有几十个字节.但解压后得到的就是100000M的惊人大小了.当然,愿意的话你也可以直接用of选项得到输出文件,不过,小心你的硬盘哦:)

2.61 如何在LINUX下把网卡设为10M

变为10M

```
#mii-tool --advertise=10baseT-FD eth0
```

用ifconfig命令也可以，不明白的可以去看一下man页

2.62 为什么是./configure而不是configure?

这涉及到环境变量$PATH。$PATH是一组路径，你输入一个命令，系统会在$PATH记录的路径中寻找这个命令文件，找到的话就执行，找不到就返回一个错误：command not found。

echo $PATH

你会看到你的$PATH变量，假如是这样：/bin:/sbin:/usr/local/bin。那么，你输入ls，系统按顺序在/bin,/sbin,/usr/local/bin三个目录中寻找，结果他在/bin中找到了，于是他就执行。你输入ifconfig，系统首先在/bin中寻找，结果没找到，于是就继续在/sbin里找，结果找到了，于是就执行。依次类推。

又假如，你的当前目录是/home/usr，/home/usr下有个可执行文件hello.sh，你要想执行他，怎么办呢你直接hello.sh是不行的，因为你的$PATH变量中没有包含当前目录./（一般都不会包含，为了安全方面考虑），所以只能用绝对路径/home/usr/hello.sh或相对路径./hello.sh来执行他

2.63 取消^M字符

你FTP一些DOS文件到unix下时，你经常会看见每行文件后面有个讨厌的^M 字符，这是因为UNIX和WINDOWS系统对文本文件行结尾的处理不同造成的。假设foo文件中有^M，有几个简单的方法可以取消它：

1. “vi”打开此文件，在Command mode下敲入：

   ```
   :%s/^V^M//g
   ```

2. 在UNIX SHELL下敲入：

   ```
   sed 's/^V^M//g' foo > foo.new
   ```

3. `$ tr -d "\015" < foo > foo.unix`

3 Linux下软件的安装与卸载

在Windows下安装软件时，只需运行软件的安装程序（setup、install等）或者用zip等解压缩软件解开即可安装，运行反安装程序（uninstall、unware、“卸载”等）就能将软件清除干净，完全图形化的操作界面，简单到只要用鼠标一直点击“下一步”就可以了。而Linux好象就不一样了，很多的初学者都抱怨在Linux下安装和卸载软件非常地困难，没有像使用Windows时那么直观。其实在Linux下安装和卸载软件也非常简单，同样也有安装向导或解压安装的方式，不相同的只不过是除了二进制形式的软件分发外，还有许许多多以源代码形式分发的软件包，下面就来详细地讲一讲这些软件的安装与卸载：

一、二进制分发软件包的安装与卸载

Linux软件的二进制分发是指事先已经编译好二进制形式的软件包的发布形式，其优点是安装使用容易，缺点则是缺乏灵活性，如果该软件包是为特定的硬件/操作系统平台编译的，那它就不能在另外的平台或环境下正确执行。

1. *.rpm形式的二进制软件包
   安装：rpm -ivh *.rpm
   卸载：rpm -e packgename
   说明：RPM（RedHat Packge Manager）是RedHat公司出的软件包管理器，使用它可以很容易地对rpm形式的软件包进行安装、升级、卸载、验证、查询等操作，安装简单，而卸载时也可以将软件安装在多处目录中的文件删除干净，因此推荐初学者尽可能使用rpm形式的软件包。rpm的参数中-i是安装，-v是校验，-h是用散列符显示安装进度，*.rpm是软件包的文件名（这里的*.rpm特指*.src.rpm以外的以rpm为后缀的文件）；参数-e是删除软件包，packgename是软件包名，与软件包的文件名有所区别，它往往是文件名中位于版本

号前面的字符串，例如apache-3.1.12-i386.rpm和apache-devel-3.1.12-i386.rpm是软件包文件名，它们的软件包名称分别是apache和apache-devel。更多的rpm参数请自行参看手册页：man rpm。如果你不喜欢在字符界面下安装或卸载这些软件包，完全可以在X-Window下使用图形界面的软件包管理程序，如glint、xrpm这样的图形接口，或者是KDE的kpackge等，这样对软件包的安装、升级、卸载、验证和查询就可以通过点击鼠标来轻松完成。

2. *.tar.gz/*.tgz、*.bz2形式的二进制软件包
   安装：tar zxvf *.tar.gz 或tar yxvf *.bz2
   卸载：手动删除
   说明：*.tar.gz/*.bz2形式的二进制软件包是用tar工具来打包、用gzip/bzip2压缩的，安装时直接解包即可。对于解压后只有单一目录的软件，卸载时用命令“rm -rf 软件目录名”；如果解压后文件分散在多处目录中，则必须一一手动删除（稍麻烦），想知道解压时向系统中安装了哪些文件，可以用命令“tar ztvf *.tar.gz”/“tar ytvf *.bz2”获取清单。tar的参数z是调用gzip解压，x是解包，v是校验，f是显示结果，y是调用bzip2解压，t是列出包的文件清单。更多的参数请参看手册页：man tar。如果你更喜欢图形界面的操作，可以在X-Window下使用KDE的ArK压缩档案管理工具。

3. 提供安装程序的软件包这类软件包已经提供了安装脚本或二进制的安装向导程序（setup、install、install.sh等），只需运行它就可以完成软件的安装；而卸载时也相应地提供了反安装的脚本或程序。例如SUN公司的StarOffice办公软件套件就使用名为setup的安装程序，而且在软件安装后提供反安装的功能，目前这种类型的软件包还比较少，因其安装与卸载的方式与Windows软件一样，所以就无需多讲了。

二、源代码分发软件包的安装与卸载

Linux软件的源代码分发是指提供了该软件所有程序源代码的发布形式，需要用户自己编译成可执行的二进制代码并进行安装，其优点是配置灵活，可以随意去掉或保留某些功能/模块，适应多种硬件/操作系统平台及编译环境，缺点是难度较大，一般不适合初学者使用。

1. *.src.rpm形式的源代码软件包
   安装：

   ```
   rpm -rebuild *.src.rpm
   cd /usr/src/dist/RPMS
   rpm -ivh *.rpm
   ```

   卸载：rpm -e packgename
   说明：rpm –rebuild *.src.rpm命令将源代码编译并在/usr/src/dist/RPMS下生成二进制的rpm包，然后再安装该二进制包即可。packgename如前所述。

2. *.tar.gz/*.tgz、*.bz2形式的源代码软件包
   安装：tar zxvf *.tar.gz 或tar yxvf *.bz2 先解压,然后进入解压后的目录：

   ./configure 配置
   make 编译
   make install 安装

   卸载：make uninstall 或手动删除
   说明：建议解压后先阅读说明文件，可以了解安装有哪些需求，有必要时还需改动编译配置。有些软件包的源代码在编译安装后可以用make install命令来进行卸载，如果不提供此功能，则软件的卸载必须手动删除。由于软件可能将文件分散地安装在系统的多个目录中，往往很难把它删除干净，那你应该在编译前进行配置，指定软件将要安装到目标路径：./configure –prefix=目录名，这样可以使用“rm -rf 软件目录名”命令来进行干净彻底的卸载。与其它安装方式相比，需要用户自己编译安装是最难的，它适合于使用Linux已有一定经验的人，一般不推荐初学者使用。

关于Linux下软件的安装与卸载lanche已经讲了这么多，但可能还会有人问怎么知道一个tar.gz/bz2包是二进制文件包呢还是源代码包？如果你用过压缩工具就会明白，压缩包未必就是软件，它也可能是备份的许多图片，也可能是打包在一起的普通资料，要分辨它到底是什么最好的办法就是查看包里的文件清单，使用命令tar ztvf *.tar.gz / tar ytvf *.bz2或者在X-Window下使用图形化的ArK压缩档案管理工具都可以，源代码包里的文件往往会含有种种源代码文件，头文件*.h、c代码源文件*.c、C++代码源文件*.cc/*.cpp等；而二进制包里的文件则会有可执行文件（与软件同名的往往是主执行文件），标志是其所在路径含有名为bin的目录（仅有少数例外）。原来这么简单呀，还不快点自己试试！

4 linux下软件的安装和卸载又一篇

4.1 了解Linux应用软件安装包

通常Linux应用软件的安装包有三种：

1. tar包，如software-1.2.3-1.tar.gz。它是使用UNIX系统的打包工具tar打包的。

2. rpm包，如software-1.2.3-1.i386.rpm。它是Redhat Linux提供的一种包封装格式。

3. dpkg包，如software-1.2.3-1.deb。它是Debain Linux提供的一种包封装格式。

而且，大多数Linux应用软件包的命名也有一定的规律，它遵循：

名称-版本-修正版-类型

例如：

1. software-1.2.3-1.tar.gz 意味着：

   软件名称：software

   版本号：1.2.3

   修正版本：1

   类型：tar.gz，说明是一个tar包。

2. sfotware-1.2.3-1.i386.rpm

   软件名称：software

   版本号：1.2.3

   修正版本：1

   可用平台：i386，适用于Intel 80x86平台。

   类型：rpm，说明是一个rpm包。

   注：由于rpm格式的通常是已编译的程序，所以需指明平台。在后面会详细说明。

而software-1.2.3-1.deb就不用再说了吧！大家自己练习一下。

4.2 了解包里的内容

一个Linux应用程序的软件包中可以包含两种不同的内容：

1. 一种就是可执行文件，也就是解开包后就可以直接运行的。在Windows中所有的软件包都是这种类型。安装完这个程序后，你就可以使用，但你看不到源程序。而且下载时要注意这个软件是否是你所使用的平台，否则将无法正常安装。

2. 另一种则是源程序，也就解开包后，你还需要使用编译器将其编译成为可执行文件。这在Windows系统中是几乎没有的，因为Windows的思想是不开放源程序的。

通常，用tar打包的，都是源程序；而用rpm、dpkg打包的则常是可执行程序。一般来说，自己动手编译源程序能够更具灵活性，但也容易遇到各种问题和困难。而相对来说，下载那些可执行程序包，反而是更容易完成软件的安装，当然那样灵活性就差多了。所以一般一个软件总会提供多种打包格式的安装程序的。你可以根据自己的情况来选择。

4.3 使用tar打包的应用软件

1. 安装：整个安装过程可以分为以下几步：

   (a) 取得应用软件：通过下载、购买光盘的方法获得；

   (b) 解压缩文件：一般tar包，都会再做一次压缩，如gzip、bz2等，所以你需要先解压。如果是最常见的gz格式，则可以执行：“tar -xvzf 软件包名”，就可以一步完成解压与解包工作。如果不是，则先用解压软件，再执行“tar -xvf 解压后的tar包”进行解包；

   (c) 阅读附带的INSTALL文件、README文件；

   (d) 执行“./configure”命令为编译做好准备；

   (e) 执行“make”命令进行软件编译；

   (f) 执行“make install”完成安装；

   (g) ）执行“make clean”删除安装时产生的临时文件。

   好了，到此大功告成。我们就可以运行应用程序了。但这时，有的读者就会问，我怎么执行呢？这也是一个Linux特色的问题。其实，一般来说，Linux的应用软件的可执行文件会存放在/usr/local/bin目录下！不过这并不是“放四海皆准”的真理，最可靠的还是看这个软件的INSTALL和README文件，一般都会有说明。

2. 卸载：通常软件的开发者很少考虑到如何卸载自己的软件，而tar又仅是完成打包的工作，所以并没有提供良好的卸载方法。

   那么是不是说就不能够卸载呢！其实也不是，有两个软件能够解决这个问题，那就是Kinstall和Kife，它们是tar包安装、卸载的黄金搭档。还有就是使用命令whereis,找出软件所有的安装目录,全部删除.

4.4 使用rpm打包的应用软件

rpm可谓是Redhat公司的一大贡献，它使Linux的软件安装工作变得更加简单容易。

1. 安装：我只需简单的一句话，就可以说完。执行：

   rpm -ivh rpm软件包名

rpm参数	参数说明
-i	安装软件
-t	测试安装，不是真的安装
-p	显示安装进度
-f	忽略任何错误
-U	升级安装
-v	检测套件是否正确安装

这些参数可以同时采用。更多的内容可以参考RPM的命令帮助。

2. 卸载：先查询:rpm -ql 软件名,确认后执行：rpm －e 软件名

   不过要注意的是，后面使用的是软件名，而不是软件包名。例如，要安装software-1.2.3-1.i386.rpm这个包时，应执行：

   rpm －ivh software-1.2.3-1.i386.rpm

   而当卸载时，则应执行：

   rpm －e software。

另外，在Linux中还提供了象GnoRPM、kpackage等图形化的RPM工具，使得整个过程会更加简单。这些软件的具体应用，笔者会另行文介绍。

4.5 使用deb打包的应用程序

这是Debian Linux提供的一个包管理器，它与RPM十分类似。但由于RPM出现得更早，所以在各种版本的Linux都常见到。而debian的包管理器dpkg则只出现在Debina Linux中，其它Linux版本一般都没有。我们在此就简单地说明一下：

安装

dpkg －i deb软件包名

如：dpkg －i software-1.2.3-1.deb

卸载

dpkg －e 软件名

如：dpkg －e software

5 Linux软件安装通用思路

在Linux系统中，软件安装程序比较纷繁复杂，不过最常见的有两种：

1. 一种是软件的源代码，您需要自己动手编译它。这种软件安装包通常是用gzip压缩过的tar包（后缀为.tar.gz）。
2. 另一种是软件的可执行程序，你只要安装它就可以了。这种软件安装包通常被是一个RPM包（Redhat Linux Packet Manager，就是Redhat的包管理器），后缀是.rpm。

当然，也有用rpm格式打包的源代码，用gzip压缩过的可执行程序包。只要您理解了以下的思路，这两种形式的安装包也不在话下了。

下面，我们就分成两个部分来说明软件安装思路：

5.1 搞定.tar.gz

1. 首先，使用tar -xzvf来解开这个包，如：

   ```
   tar -xzvf apache-1.3.6.tar.gz
   ```

   这样就会在当前目录中创建了一个新目录(目录名与.tat.gz包的文件名类似），用来存放解压了的内容。如本例中就是apache-1.3.6

2. 进入这个目录，再用ls命令查看一下所包含的文件，如：

   ```
   #cd apache-1.3.6
   #ls
   ```

   你观察一下这个目录中包含了以下哪一个文件：configure、Makefile还是Imake。

   (a) 如果是configure文件,就执行：

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

   (b) 如果是Makefile文件,就执行：

   ```
   #make
   #make install
   ```

   (c) 如果是Imake文件,就执行：

   ```
   #xmkmf
   #make
   #make install
   ```

3. 如果没有出现什么错误提示的话，就搞定了。至于软件安装到什么地方，通常会在安装时出现。否则就只能查阅一下README，或者问问我，:-)

如果遇到错误提示，也别急，通常是十分简单的问题：

1. 没有安装C或C++编译器；

   确诊方法：执行命令gcc（C++则为g++），提示找不到这个命令。
   解决方法：将Linux安装光盘mount上来，然后进入RPMS目录，执行命令：

   ```
   #rpm -ivh gcc*
   ```
   （哈哈，我们用到了第二种安装方式）

2. 没有安装make工具；

   确诊方法：执行命令make，提示找不到这个命令。解决方法：将Linux安装光盘mount上来，然后进入RPMS目录，执行命令：

   ```
   #rpm -ivh make*
   ```

3. 没有安装autoconf工具；

   确诊方法：执行命令make，提示找不到这个命令。解决方法：将Linux安装光盘mount上来，然后进入RPMS目录，执行命令：

   ```
   #rpm -ivh autoconf*
   ```

4. 缺少某些链接库；

   确诊方法：在make时，提示需要某些文件。解决方法：安装包含这个文件的包，这就需要积累了。

5.2 搞定.rpm

RPM是Red Hat公司随Redhat Linux推出了一个软件包管理器，通过它能够更加轻松容易地实现软件的安装。

- 安装软件：执行rpm -ivh rpm包名，如：

  ```
  #rpm -ivh apache-1.3.6.i386.rpm
  ```

- 升级软件：执行rpm -Uvh rpm包名。
- 反安装：执行rpm -e rpm包名。
- 查询软件包的详细信息：执行rpm -qpi rpm包名
- 查询某个文件是属于那个rpm包的：执行rpm -qf rpm包名
- 查该软件包会向系统里面写入哪些文件：执行rpm -qpl rpm包名

补充一点，rpm包分两种，一种是binary型，其文件名称一般为*.i386.rpm，这种包装上后程序也就装好了。还有一种是source类型，文件名一般是*.src.rpm，这是打过包的源文件，此rpm包装上后还需编译

6 linux下源代码版本的软件安装常识

本文写给刚踏入linux门槛的朋友，希望通过阅读此文，能给你们提高一点安装linux软件的知识。有写的不好地方还请指正。

在Linux下，安装源码形式的软件时，这种软件通常都是tar.gz 形式的包，假设为file-name-version.tar.gz 。

第一步，解包。

在你软件包所在目录下，执行tar 命令，如下：

```
tar zxvf file-name-version.tar.gz
```

解包后在同目录下生成file-name-version目录。

第二步，进行预处理（大多软件都需要这一部）。

进入file-name-version，你会发现有configure这个文件，这个文件用来收集当前的系统信息，和软件编译时需要的参数。

这一步如果能正确执行，会重新生成Makefile文件。大多情况下，不同软件执行config ure命令时都有许多相同参数，

我这里要强调的是--prefix参数，他指定你软件安装后的目录。

第三步，编译。

执行make命令，这是操作系统带的命令，默认的，它会去当前目录下的Makefile文件中读取编译参数。

如果这一步执行完毕，那所需要的可执行文件已经编译成功了，但是现在还在file-name-version目录下，所以需要把他们拷贝到由--prefix指定的目录，这就是下一步的工作了。

第四步，安装。

执行make install 命令。 这一步是make 命令去执行Makefile 文件中install 参数中指定的操作，执行完这一步后有用就拷贝到了指定目标目录。

如果没有指定--prefix参数，那目标目录就是/usr/local/file-name。这时的目录已经没有version后缀了。

有的软件在执行完这步后还可以执行make test 命令，用来测试软件是否成功安装。

第五步，善后。

到此为止，软件已经成功安装了，但是在安装过程中生成了许多临时文件，虽说不多，也不大，但留着也没什么用，

用如下方法，可清除这些文件。还是在file-name-version目录下，执行:

```
make clean
```

就可以清除掉那些文件，当然前提是Makefile文件中得有clean这个参数。

同样的，有的软件的Makefile 中还有uninstall 参数。那这时你执行

```
make uninstall
```

命令你猜会发生什么事呢，当然，所有安装到目标目录的文件全都被删除了，这个参数就是反安装了。

7 shell 程序怎样解释命令

Unix 的神秘之处就在于它的那些稀奇古怪的命令。比如在两个斜杠之间是否需要加一个点，或者连字符后是否需要加引号。要真正理解这些命令的话，就必须知道这些命令行是怎样被解释的。

Shell 程序(sh, ksh, csh, 或者其他变种) 读入并分析命令行，所以，命令行其实是对shell 程序的输入参数而已。

我们先来看一下读入的顺序：

1. 命令历史替换(除了Bourne shell)
2. 分开单词，包括特殊字符
3. 更新命令历史(除了Bourne shell)
4. 解释单引号和双引号
5. 别名替换(除了Bourne shell)
6. 输入输出重定向(< > 和 |)
7. 变量替换(以$开头的变量)
8. 命令替换(命令s inside back quotes)
9. 文件名扩展(file name wild cards)

你可以看到Bourne shell 没有命令历史和别名替换的步骤，因为Bourne shell 不包括这些功能。

1. 命令历史替换如果你在Korn shell (ksh), C shell (csh), 或者其他类似的shell 设置了命令历史。命令行就会在执行之前存入一个历史文件，你以后可以用

   history 查看这些命令。命令前面有一个数字：

```
13 ls *.txt
14 cd $HOME
15 ls *.log
```

在Korn shell 里面，你可以用r 加上数字来取回以前用过的命令。例如用r 13 就可以重复以上的ls *.txt 命令。

在C shell 里面，使用感叹号并且没有空格来替换r : !13。

处理命令行时，shell 首先检查命令替换，在历史文件里面寻找，取回每个命令，然后写入新的命令行。当然有关历史的处理不止简单的这些步骤。我们会在以后的文章中单独讨论。

2. 分开单词接下来的步骤就是把单词和特殊字符分开了。一个单词被shell 程序认为是命令的一部分。以下的命令用长格式列出当前目录并寻找包含mjb 的行。

```
ls -l|grep mjb
```

这个命令的单词包括：`ls`, `-l`, `|`, `grep` 以及`mjb`。单词也可以包含在引号里面。在以下的命令中，一个长格式列表搜索创建于`"Sep 07"`的文件。

```
ls -l|grep "Sep 07"
```

在这里，`"Sep 07"` 作为一个单词被放入引号。

3. 更新历史清单一旦单词被确定，这个命令就写道历史文件的结尾。

4. 解释单引号和双引号单引号里面的变量，不被处理，双引号里面的会被展开。可以输入以下的命令来检验：

```
echo $PATH
echo '$PATH'
echo "$PATH"
```

第一行显示`$PATH` 变量的值，第二行显示单词`$PATH`，第三行显示 $PATH 变量的值。

5. 别名替换每个命令的第一个单词会和别名列表比较。下面的例子中，ls 和grep 在别名清单中被检查，作别名替换。别名就是用一个命令来替换另外一个命令的方法。例如：以下的命令:

```
alias ll 'ls -l'
```

建立一个新的命令ll，当你输入

```
ll *.txt
```

时，就像输入ls -l *.txt 。

6. 管道和重定向到了这里，shell 程序“看”完了命令行里的所有单词。对于|，>，<，>>，和其他的重定向命令，找到一个，就建立管道或者重定向。

7. 变量扩展现在命令行内的变量扩展，所有包含在双引号内的变量被替换成对应的值。

8. 命令替换命令替换查找反引号，执行括在其中的命令以及参数，然后使用运行结果作为参数传递给要执行的命令，例如：

```
ls -l `ls -p|grep /`|more
```

命令ls -p 产生带给目录加注反斜杠的目录列表，如：

```
ls -p

STARTUP
file.txt
file2.txt
mystuff/
xdir/
```

加在后面的grep / 命令只选择那些包含反斜杠的项：

```
ls -p|grep /

mystuff/
xdir/
```

把整个命令用反引号引进后，生成的结果作为参数传递给ls -l

```
ls -l `ls -p|grep /`|more
```

这等同于：

```
ls -l mystuff/ xdir/|more
```

9. 通配符命令处理其寻找文件名的通配符并扩展他们。通配符包括：* 和? 以及中括号，例如：

```
ls -l [abc]*
```

显示那些以a 或b 或c 结尾的文件或者目录。

10. 执行最后命令被执行，这就完成了整个的流程。

需要指出的是：在上述第八步中，当处理反引号中的命令时，先把它分离出来，作为一个单独的命令按照以上 1 — 9 步来处理，对于用分号隔开的命令也一样。

8 使用adsl-setup配置ADSL

执行/sbin/adsl-setup 后,程序会提示你进行以下的工作:

```
1.LOGIN NAME

Enter your Login Name (default root): 这里输入ISP给的帐号

2.INTERFACE

Enter the Ethernet interface connected to the ADSL modem
For Solaris, this is likely to be something like /dev/hme0.
For Linux, it will be ethX, where 'X' is a number.
(default eth0): 选择网卡（自行选择）

3.Do you want the link to come up on demand, or stay up con-
tinuously?If you want it to come up on demand, enter the idle t-
ime in seconds after which the link should be dropped. If you w-
ant the link to stay up permanently, enter 'no' (two letters, l-
ower case.)NOTE: Demand-activated links do not interact well with
dynamic IPaddresses. You may have some problems with demand-
activated links.Enter the demand value (default no):
选择默认

4.DNS

Please enter the IP address of your ISP's primary DNS server.
If your ISP claims that 'the server will provide dynamic DNS
addresses',enter 'server' (all lower-case) here.
If you just press enter, I will assume you know what you are
doing and not modify your DNS setup.
Enter the DNS information here: 输入习惯使用的DNS

5.PASSWORD

Please enter your Password: ISP给的密码
```

6.USERCTRL

Please enter 'yes' (two letters, lower-case.) if you want to allow normal user to start or stop DSL connection (default yes):
自行选择

7.FIREWALLING

Please choose the firewall rules to use. Note that these rules are very basic. You are strongly encouraged to use a more sophisticated firewall setup; however, these will provide basic security. If you are running any servers on your machine, you must choose 'NONE' and set up firewalling yourself. Otherwise, the firewall rules will deny access to all standard servers like Web, e-mail, ftp, etc. If youare using SSH, the rules will block outgoing SSH connections which allocate a privileged source port.

The firewall choices are:
0 - NONE: This script will not set any firewall rules. You are responsible for ensuring the security of your machine. You are STRONGLY recommended to use some kind of firewall rules.
1 - STANDALONE: Appropriate for a basic stand-alone web-surfing workstation
2 - MASQUERADE: Appropriate for a machine acting as an Internet gateway for a LAN
Choose a type of firewall (0-2): 防火墙可以选1

8.Start this connection at boot time

Do you want to start this connection at boot time?
Please enter no or yes (default no):
是否启动时就连接?

9.Ethernet Interface: eth0
User name: root
Activate-on-demand: No
DNS: Do not adjust
Firewalling: STANDALONE
User Control: yes
Accept these settings and adjust configuration files (y/n)?

是否接受配置。如果Y就可以了，就起用了，N一切从头开始。

拨号使用/sbin/adsl-start 停止使用/sbin/adsl-stop 查看状态/sbin/adsl-status

9 各种压缩文件的解压（安装方法）

文件扩展名	解压（安装方法）
.a	ar xv file.a
.Z	uncompress file.Z
.gz	gunzip file.gz
.bz2	bunzip2 file.bz2
.tar.Z	tar xvZf file.tar.Z 或者 compress -dc file.tar.Z — tar xvf -
.tar.gz/.tgz	tar xvzf file.tar.gz 或者 gzip -dc file.tar.gz — tar xvf -
.tar.bz2	tar xvIf file.tar.bz2 或者 bzip2 -dc file.tar.bz2 — xvf -
.cpio.gz/.cgz	gzip -dc file.cgz — cpio -div
.cpio/cpio	cpio -div file.cpio 或者 cpio -divc file.cpio
.rpm/install	rpm -i file.rpm
.rpm/extract	rpm2cpio file.rpm — cpio -div
.deb/install	dpkg -i file.deb
.deb/exrtact	dpkg-deb –fsys-tarfile file.deb — tar xvf - 或者 ar p file.deb data.tar.gz — tar xvzf -
.zip	unzip file.zip
.rar	rar e file.rar

10 利用Samba共享Linux和Windows资源

10.1 Samba 概述

Samba是一组用于实现各种操作系统相互通信的功能广泛的程序。使用Samba要求各台计算机用网络连接在一起。如果没有网络，Samba的用途是十分有限的。

在大规模的网络中，管理员利用Samba把大量的Unix系统和大量的Windows以及其他类型的工作站连接在一起。有许多优秀的书籍专门解释关于Samba的各种复杂问题，不过这里只涉及基础知识－－但这已经足以让我们在一个小型的网络内连接一个Linux系统和一个Windows系统，安装一个Linux系统上的共享打印机。

Samba使用Microsoft的Server Message Block，或者说是SMB协议。SMB最初出现于80年代，后来Microsoft采用并扩展了它。SMB运行在另一个协议即NetBIOS之上，而NetBIOS又运行在其他诸如IPX或TCP/IP之类协议

之上。我们并不需要详细了解这些协议，只需要知道Microsoft使用SMB让我们在网络上共享文件。

Microsoft从来没有公开过SMB协议的细节，这给开发基于SMB的系统带来了困难。然而，一组由Andrew Tridgell领导的天才对该协议实施了反向工程，并在此基础上创建了早期的Samba。现在，Samba是源代码开放的，它由遍及世界各地的志愿者维护和改进。

只要你用过连接局域网的Windows PC就会知道，Windows允许我们“映射”或“共享”网络驱动器，这是通过在Windows资源管理器（即Windows中的文件管理工具）中选择“工具/映射网络驱动器”实现的。只要网络允许进行映射，我们就可以利用这个工具映射另一个系统中的驱动器，使它看起来就象本地文件系统的一部分。或者说，我们可以让另一台机器的D:驱动器成为本地系统的G:驱动器。

Samba的功能远远不止这一点，它能够让Windows PC共享Linux系统中的目录，也能够实现其他资源的共享，比如打印机。

大多数流行的Linux发行中都包含Samba。如果你的系统中没有安装Samba，最简单的方法就是下载一个编译好的Samba软件包。当然，最好的下载网站应该是Samba的官方网站：http://www.samba.org 。

10.2 配置Samba

安装好Samba之后还要配置它，“声明”某些可以被其他系统访问的资源。具体声明哪些资源在Samba的配置文件/etc/smb.conf中指定。在大规模的网络中，smb.conf文件可能相当复杂，但对于小型网络，只要为每个需要共享的目录或资源加入几行内容到smb.conf文件就可以了。

配置Samba可以用文本编辑器手工编辑smb.conf，但最简单的方法是使用SWAT。SWAT是一个Web界面的配置工具。本文后面会介绍使用SWAT配置Samba的方法。

smb.conf的man文档总长度超过75页，相当复杂。好在配置一个连接2到3台机器的Samba并不需要我们融会贯通全部75页文档，实际上只需要了解本文所介绍的基础知识即可完成这个任务。

不过必须明白Samba涉及到许多严肃的安全问题，这一点非常重要。当我们通过Samba声明某个目录共享时，网络上的其它系统就可以直接访问该目录。在大型网络里，或者网络与Internet连接时，最终有权访问该目录的机器数量可能远远超过设置Samba时所设想的数量。

如果你认为有必要关注系统的安全问题，我建议你读几本有关Samba的专著。本文提供的例子非常简单，有关安全方面的考虑很少，它适用于通常不能从Internet访问的小型家庭网络。

smb.conf文件会在安装Samba的时候自动安装到/etc/smb.conf下，因此，你应该通过修改smb.conf文件配置Samba，而不是从头开始创建smb.conf。默认文件包含了相当丰富的说明，指导用户如何启用、禁止或改变各种选项。在这个文件中，说明内容的前面由;或者符号引导，这两个符号告诉Samba在读取配置信息的时候忽略这些行。

smb.conf文件按“节”组织，每个节开头是一个加括号的关键字。配置网络时必须编辑[global]节，为每一个需要声明为共享的目录和资源加

入对应的条目。[global]节里定义的是一组全局选项，访问Samba共享资源的Windows组和机器的IP地址都在这里指定。

例如，在我的配置文件中，[global]节的内容如下：

```
[global]
workgroup = my_workgroup_name
hosts allow = 172.16.127.
security = user
```

workgroup行指定了Windows工作组的名字，hosts allow行指定了允许访问本地共享资源的机器（所有这些机器的IP地址都以172.16.127开头）。security行用于指定基本的访问安全级别。再次提醒一下，如果共享资源位于大型网络之中或直接连接到Internet，一定要特别小心，务必仔细地阅读与安全有关的Samba文档。

[global]节影响所有Linux下声明为共享的目录，但用户设置具体到每一个共享目录。，例如，下面是摘自我的smb.conf的一部分内容：

```
[hal]
comment = "Hal's home directory"
path = /home/hal
browseable = yes
writable = yes
available = yes
public = yes
only user = no
create mode = 0750
```

节的名字（[hal]）公布给Windows客户机，如果需要的话，Windows客户机还可从comment行得到更多有关该共享目录的说明。path是共享目录的路径。[hal]节的其他几行内容指定了客户机可以浏览目录、在目录中写入内容，以及赋予该目录和在该目录下创建的文件的各种许可。这里仅列出了可能用到的一部分参数，更详细的说明可以在man文档找到。

Linux系统的打印机也可以被Samba客户共享。下面是我的smb.conf文件中有关打印机共享的声明：

```
[ljet]
comment = "Laserjet"
path = /var/spool/lpd/lp
printer = lp
public = yes
printable = yes
print command = lpr -r -h -P %p %s
```

设置共享打印机时可能要把打印假脱机目录（/var/spool/lpd/ lp）设置成可以被所有用户访问，这可以用chmod a+rwx /var/ spool/lpd/lp命令实现。如果在配置打印共享时出现问题的话，不妨试一下这个命令。

编辑好smb.conf文件后可以用testparm命令检验它。testparm命令读取smb.conf文件并显示它找到的所有错误。显示错误的时候，按一下Enter键可以显示已设置的所有参数。下面显示的是testparm在我的系统上的输出（由于长度原因，[global]节已被截掉一部分）。

```
Load smb config files from /etc/smb.conf
Loaded services file OK.
Press enter to see a dump of your service definitions
# Global parameters
workgroup = YAMANOTE
netbios name =
netbios aliases =
server String = Samba Server
interfaces =
bind interfaces only = No
security = USER
encrypt passwords = No
...
[homes]
comment = /Home
path = /home
read only = No
create mask = 0750
guest ok = Yes
[hal]
comment = Hal Home
path = /home/hal
read only = No
create mask = 0750
guest ok = Yes
[ljet]
path =/var/spool/lpd/lp
read only = No
guest ok =Yes
print ok = Yes
print command =lpr -r -h -P %p %s
printer name=lp
```

启动Samba Daemon

Samba通信由一组后台运行的daemon程序处理，这些程序可以配置成在Linux系统启动的时候自动启动。daemon是对在后台持续运行的程序的称呼。如果Samba daemon没有在系统启动的时候自动启动，你可以用linuxconf之类的工具把它设置成自动启动。另外也可以用下面的命令手工启动Samba daemon，注意这些命令必须用root身份运行：

```
# /usr/sbin/smbd -D
# /usr/sbin/nmbd -D
```

Samba daemon启动后读入smb.conf文件并公布在该文件中设置成共享的目录和服务。缺省情况下，Samba会每隔60秒重新检查smb.conf文件，对smb.conf文件的改动会在Samba daemon下一次检查文件的时候生效。

10.3 用SWAT配置Samba

利用SWAT（Samba Web Administration Tool）可以在Web浏览器中管理Samba。

- 第一步：配置SWAT SWAT提供了一种编辑Samba配置文件smb.conf的简单方法，但使用SWAT之前先必须进行配置，编辑/etc/services和/etc/inetd.conf文件。把下面这行内容加入/etc/services文件：

  ```
  swat 901/tcp
  ```

  把下面这行内容加入/etc/inetd.conf文件：

  ```
  swat stream tcp nowait,400 root /usr/sbin/swat swat
  ```

  这里假定SWAT安装到了/usr/sbin目录。如果SWAT安装在其他目录，则必须修改上述内容使其反映出实际安装位置。

- 第二步：启动并登录到SWAT 修改好上述几个文件之后，启动Web浏览器并在地址栏输入下面这个URL：

  ```
  http://localhost:901/
  ```

  出现提示时，输入root用户名和密码。通过SWAT主页可以访问所有联机Samba说明文档。

- 第三步：设置全局参数在SWAT中点击“Globals”按钮可以打开设置smb.conf全局参数的页面，在这里可以设置Windows工作组和网络上机器的IP地址。

- 第四步：用SWAT创建共享资源用SWAT创建共享资源时，先点击“Shares”按钮，在“Create Share”按钮的右边为共享资源输入一个名字，点击“Create Share”按钮，然后再为该共享资源指定其他的参数。

10.4 配置Windows

用Windows NT来访问Linux服务器上的共享驱动器。在Linux上配置并测试了Samba之后，我重新启动了NT，NT自动找到了Linux系统上的共享驱动器。

如果你使用的是Windows 95或98，你还要设置用户密码。首先从控制面板打开“密码”，在“用户配置文件”下启用“用户可以定制自己的参数”，然后在“设置密码”下面为自己设置一个用户名字和密码，注意要确保这里设置的内容和访问Linux系统时所使用的名字和密码一样。当你打开Windows桌面上的“网上邻居”时，应该已经可以访问共享驱动器了。

如果你在使用Samba的时候遇到了问题，它们很可能与安全设置和密码有关。请仔细阅读一下Samba的文档，包括HOWTO和FAQ。

在Windows中使用Linux打印机

如果你在Linux机器上把打印机指定为共享的Samba资源，你就可以从Windows系统访问这个打印机。不过在使用这个打印机之前，你先要把它加入到Windows的可用打印机列表中。

添加打印机的方法很简单。首先从Windows控制面板选择“打印机”，然后选择“添加打印机”。选中“网络打印机”并点击“下一步”，在网络的Samba服务器上找到待添加的打印机，双击这个打印机的图标，按照提示完成接下来的步骤。

完成后，Linux打印机就会出现在Windows的可用打印机列表中。

11 使用U盘,V盘，移动硬盘的终极方法

我被这个问题困扰了很久，研究了N种方法，现在我就吧我的方法告诉大家，一定能够搞定。我看了很多文章，都是写得很不全。看来只有自己研究才是呀。实践得真知！！！

1. 用lsmod看看`usb-storage,usb-uhci,sd_mod,scsi_mod`是否加载
2. 如果你是自己编译的内核，这些都编译进去了，就不需要加载了
3. 用dmesg—grep usb看看是否加载了usb相关的东东
4. 试着用fdisk -l /dev/sda (sdb/sdc/sdd)看看和哪个东东连接上了，如果你用的SCSI硬盘，那么一般就是sdb如果你用的IDE硬盘一般就是sda
5. 如果出现了类似于fdisk -l /dev/hdc(你的硬盘）的提示，那么恭喜你。你可以用：mount -t vfat /dev/sda1 /mnt/usb来挂接usb盘了。
6. 这是重点了，如果你还是没有用成功，那么。照着下面的方式加载模块

```
modprobe usb-storage
modprobe usb-uhci
modprobe sd_mod
modprobe scsi_mod
```

然后再用fdisk看看。如果还是不行那么再加载一个模块

```
modprobe ub-ohci
```

这个非常重要。我研究了很久才发现有个这个模块可以用

那么这样一定就可以用了。如果还是不行，检查你得盘是否没有插上或是usb损坏了！！如果还是不行，发贴子问！！如果还是不行，那么找点基础的书来看看：）呵呵

谢谢大家，希望对你使用usb有帮助！

12 Linux初学者Patch使用指南

12.1 简介

本文的目的是向Linux新手介绍一种无价的资源，Larry Wall的patch程序。patch是用来查找文件之间差异的GNU diff命令的一个接口；diff有很多选项，但是该命令最常用的用途是用来生成一个文件，该文件中列出了内容发生改变的行，显示两个原始文件、修改过的行以及由于内容没有变化而忽略掉的行。patch典型地用于把一个目录下的源代码文件更新到新的版本，从而就避免了下载整个新的源代码档案的必要。下载一个有效的patch仅仅需要下载发生变化的那些代码行就可以了。

patch最初源自十年前，那时网络带宽的限制促进了patch的发展，然而和当时的很多Unix工具一样，直到现在，patch还在广泛应用。在Dr. Dobb之旅的2月份的程序员杂志中，Larry Wall对早期的patch做了一些很有趣的说明：

> DDJ:顺便问一下，patch和diff哪个出现的早？
> LW:从很长一段时间来说，diff出现地比较早。我想diff大约比patch早10年出现，一回想起来，我就纳闷为什么没有人早些想到使用patch呢？

但是我想我知道这中间的原因。这很大程度上是心理因素使然。当开发出diff时，程序员增加了一个e选项，我想就是这个选项的原因，该选项后来滋生为一个ed脚本，因此大家都会对自己说，"嗯，如果我想自动使用diff，那么我就使用这个选项。"因此从来都没有人编写一个计算机程序来获取其它格式的输出并使用这些结果。或者是那些设计diff的人员，或者是那些使用diff格式而受益的人员太沉迷其中了，因为你可以对那些已经修改过的内容使用diff操作并让这些内容正常工作都是很容易的。

现在回想起来，这个问题是显而易见的。但是平心而论，与其说这是一个天才的灵感的闪现，还不如说这是自信心的体现。我开发出rn的第一个

版本，然后继续为它编写补丁，这整个事情就是一团乱麻。你不可能强制用户使用补丁，因为他们可以手工完成这些工作。因此，他们就会省略一些自己认为不必要的工作，他们把新的修改加诸于原来的程序之上，因此而使得程序混乱。我编写补丁，这样就没有人找借口说这很难了。

我不清楚是否事实就是如此，但是多年以来，我一直对别人讲patch对于计算机文化的影响比rn和Perl的影响都要大。现在Internet的速度比原来有大幅度的提高，把整个发行版本分散到世界各地也变得更加简单，似乎只有在开发者之间才需要传送补丁。我已经很多年没有传送Perl的patch工具包了。我认为虽然patch整体上的重要性在逐渐降低，但是仍然是开发者交流思想的一种方法。但是就那一段时间而言，patch真正在相当大的程度上都影响了软件的开发方式。

Larry Wall针对patch对于计算机业界总体上重要性正在降低的评价可能是正确的，但是在自由软件世界中，patch仍然是一种必不可少的工具。无处不在的patch使得新手和非程序员能够简单地参与软件的alpha测试和beta测试，这对于整个计算机业界是十分有益的。

在我留意到在Linux内核邮件列表中会周期性的出现这样一件事情时就产生了写这篇文章的念头。大约每三个月就会有人张贴要求把Linux内核源代码的发行版本独立出来的文章，据说这是因为有些人只对i386的代码和IDE的磁盘驱动感兴趣，他们并不想为每个内核发行版本都下载Alpha、Sparc等等的文件和众多的SCSI驱动程序。这篇文章后面紧跟的是一些耐心的回复文章（有些文章则并没有耐心），大部分文章都是在讨论原来的使用有关patch来更新内核源代码。接着Linus Torvalds就会再次声明自己没有兴趣投身于这种把内核源程序切割成小块的繁杂的劳动，但是如果有人愿意，他们可以自由地开展这项独立的工程。到现在为止都没有志愿者出现。我并不想谴责那些内核黑客不能耐心等待，却把生活变得复杂；我猜想直接使用内核工作可能比检查整个内核的发行版本方案要更加有趣、更富有挑战性。下载11M的内核源程序包可是件非常耗费时间的事情（对于那些按照时间上网的人来说，这是很昂贵的），但是内核patch才只有几十K大小，很少会超过1M。我的硬盘上的2.1.99开发内核源程序经过patch的升级，已经升级到了2.1.119版本，我怀疑如果我紧随内核的发展而不断升级，那么也许我就要完整地下载每一个发行版本了。

12.2 使用patch

patch附带有一个很好的帮助，其中罗列了很多选项，但是99%的时间只要两个选项就能满足我们的需要：

```
patch -p1 < [patchfile]
patch -R < [patchfile] (used to undo a patch)
```

-p1选项代表patchfile中文件名左边目录的层数，顶层目录在不同的机器上有所不同。要使用这个选项，就要把你的patch放在要被打补丁的目录下，然后在这个目录中运行`path -p1 < [patchfile]`。来自Linux内核patch的一个简短的引用可以这样实现：

```
diff -u --recursive --new-file v2.1.118/linux/mm/swapfile-
.c linux/mm/swapfile.c--- v2.1.118/linux/mm/swapfile.c W-
ed Aug 26 11:37:45 1998 +++ linux/mm/swapfile.c Wed
Aug 26 16:01:57 1998 @@ -489,7 +489,7 @@
int swap_header_version;
nt lock_map_size = PAGE_SIZE;
int nr_good_pages = 0; - char tmp_lock_map = 0; + unsigned
long tmp_lock_map = 0;
```

应用来自本段中使用-p1开关拷贝的patch可以有效地减短patch定位的路径；patch会查找当前目录下一个名为/mm的子目录，接着应该会在这儿发现swapfile.c文件，然后等待打补丁。在这个过程中，以破折号（“-”号，译者注）开始的行会被一个以加号（“+”号，译者注）开始的行代替。一个典型的patch会包含对多个文件的更新，每个部分中都由对两个版本的文件运行diff -u命令的输出结果组成。

patch在操作时把自己的输出结果显示在屏幕上，但是这种输出通常都滚屏太快，来不及观看。原来准备patch的文件名为*.orig，新的patch文件会覆盖这个初始文件名。

12.2.1 打补丁的问题

使用不同版本的patch问题来源可能不同，所有的版本在网络上都是可用的。Larry Wall近年来已经不再做很多工作来更新patch了，这可能是由于他最后发行的一个版本在大部分情况下都能正常运行。最近几年以来，一直是GNU项目的FSF程序员发行新版本的patch。他们首先修订有问题的patch，但是我最近一直使用没有问题的2.5版本（这是Debian2.0的发行版本号）。过去，我的2.1版本也一直运行的很好。当前的GNU patch的版本可以从GNU FTP站点上获取，然而大部分人都只使用他们Linux发行版中所提供的版本。

让我们假定你已经对一个目录下的源程序文件进行了patch修补工作，但是patch并没有清晰地发挥作用。这可能会偶然发生，在打补丁的过程中会显示错误信息，其中带有行号，说明哪一个文件出现了问题。有时错误是很明显的，例如缺少了分号，这种错误可以不费多大力气就能改正。另外一种可能是从patch部分删除了产生问题的部分，但是这样根据所涉及到的文件的不同可能会正常工作，也可能不能正常工作了。

另外一种常见的错位为：假设你有一个未使用tar打包的内核源程序文件，在/linux/arch/下浏览各个子目录时你会发现各种机器体系结构子目录，例如alpah、sparc等等。如果你和大多数Linux用户一样，使用的是Intel的处理器（或者是Intel系列），你可以决定删除这些目录，这些目录对于编译你特殊的内核并不需要，只是白白占用了磁盘空间。一段时间之后发行了一个新的内核patch，此时试图进行patch操作，当它发现不能找到自己打补丁需要的Alpha或者PPC文件，就会停顿下来。幸运的是patch在这些地方允许用户参与，它会询问”Skip this patch?”回答”y”，patch就可以按照正确的路径继续执行。也许你需要回答这个问题很多次，因此允许自己不需要的目录保留在磁盘上是一种很好的方法。

12.2.2 给内核打补丁的技巧

很多Linux用户使用patch都主要是给内核源程序打补丁，因此有一些技巧可以使用。可能最简单的方法是使用shell脚本给内核打补丁，这可以在内核源程序树中的/scripts子目录中找到。这种方便的、编写良好的脚本是由Nick Holloway在1995年编写的；两年以后，Adam Sulmicki增加了多种压缩格式的支持，包括*.bz、*.bz2、compress、gzip和无格式文本（也就是已经解压的patch）。这个脚本假定在你使用新版本的patch时，你的内核源程序是在/usr/src/linux目录中。这些缺省值可以通过这种格式的命令行开关覆盖：patch-kernel [sourcedir [patchdir]]。如果任何一部分的patch失败，对内核打补丁的过程都会失败，但是如果patch清晰地起作用，它就会调用find，这会删除所有的patch留下的*.orig文件。

如果你准备查看命令的输出，或者可能你希望保留*.orig文件直到你确定打过补丁的源程序编译已经通过，按照我的经验，直接运行patch（正如前面介绍的一样，patch位于内核源程序的最高目录）是很可靠的。为了避免对patch进行解压，在使用之前，可以使用这样一个技巧：

```
gzip -cd patchXX.gz — patch -p1
```

或者

```
bzip2 -dc patchXX.bz2 — patch -p1
```

在使用patch之后，可以使用find程序来检测被拒绝的文件：

```
find . -name *.rej
```

第一次使用这个命令，语法可能有些不清楚。点号（“.”）说明find应该查找当前目录并递规查找当前目录之下的所有子目录。记住，点号前后都应该有一个空格。通配符”*”号前面的反斜线把星号转义出来，以免shell会搞混，星号是有其它意义的。如果find找到了任何的*.rej文件，它就会把文件名打印到屏幕上。如果没有任何输出find就退出了，那么就差不多能确定patch正确发挥作用了。

find的另外一个工作是删除*.orig文件：

```
find . -name *.orig -print0 — xargs -0r rm -f
```

这个命令敲起来相当麻烦，可以使用一个新的shell别名来代替这个命令。在你的~/.bashrc文件中类似这样的一行：

```
alias findorig find . -name *.orig -print0 — xargs -0r rm -f
```

可以允许你只输入findorig就可以调用前面的命令。如果别名命令的定义中包含空格，那么就必须使用单引号。为了不用先退出再重新登陆就可以使用一个新的别名，可以在命令行中敲如~/.bashrc。

12.3 附加内容和结束语

在撰写本文时，我刚好把自己的机器从2.1版本使用patch升级到了2.5版本。这两个版本都是来自现在的FSF/GNU维护人员。马上我就注意到2.5版本默认的输出已经改变了，屏幕上显示的信息变少了。原来在patch检测进行修补的行号时显示的Larry Wall的"...hmm"不见了。2.5版本的输出只剩下诸如"patching file [filename]"之类的信息了，而没有早期版本显示的那么多信息了。无可否认，信息滚屏太快，根本无法阅读，但是输出可以重定向到一个文件中供以后使用。这种变化不会影响程序的功能，但是减少了人为的成分。在我看来，使用诸如原来的"...hmm"信息和源代码中的注释一样，对于提醒用户程序是显示执行的工作的结果是很有价值的，这就像人在呼吸一样，而不应该使用一些毫无结果的位集合。通过对patch命令行增加–verbose开关可以恢复原来的显示内容，但是我相信很多用户既不会注意到这个选项，也不会不辞辛劳地输入这个选项。2.1和2.5版本的另外一个不同是除非patch给出了-b选项，否则不能创建*.orig备份文件。

对于那些对软件和内核"前沿"bug报告测试和提供测试报告不感兴趣的人来说，patch并不是必须的，但是通常大部分Linux世界里有趣的开发都是属于这个范畴的。获得patch的使用方式并不困难，这种努力可以得到充分的回报。

(译者注：由于水平有限，错误疏漏之处在所难免，如果你认为本文翻译的有问题，可以对其进行修改，请将修改后的文章给译者（xjtufr@263.net）发送一份。)

13 linux关机命令小解

在linux下一些常用的关机/重启命令有shutdown、halt、reboot、及init，它们都可以达到重启系统的目的，但每个命令的内部工作过程是不同的，通过本文的介绍，希望你可以更加灵活的运用各种关机命令。

13.1 shutdown

shutdown命令安全地将系统关机。有些用户会使用直接断掉电源的方式来关闭linux，这是十分危险的。因为linux与windows不同，其后台运行着许多进程，所以强制关机可能会导致进程的数据丢失，使系统处于不稳定的状态，甚至在有的系统中会损坏硬件设备。

而在系统关机前使用shutdown命令，系统管理员会通知所有登录的用户系统将要关闭。并且login指令会被冻结，即新的用户不能再登录。直接关机或者延迟一定的时间才关机都是可能的，还可能重启。这是由所有进程(process〕都会收到系统所送达的信号〔signal〕决定的。这让像vi之类的程序有时间储存目前正在编辑的文档，而像处理邮件〔mail〕和新闻〔news〕的程序则可以正常地离开等等。

shutdown执行它的工作是送信号〔signal〕给init程序，要求它改变runlevel。Runlevel 0被用来停机〔halt〕，runlevel 6是用来重新激活〔reboot〕

系统，而runlevel 1则是被用来让系统进入管理工作可以进行的状态；这是预设的，假定没有-h也没有-r参数给shutdown。要想了解在停机〔halt〕或者重新开机〔reboot〕过程中做了哪些动作，你可以在这个文件/etc/inittab里看到这些runlevels相关的资料。

shutdown 参数说明:

-t 在改变到其它runlevel之前，告诉init多久以后关机。

-r 重启计算器。

-k 并不真正关机，只是送警告信号给每位登录者〔login〕。

-h 关机后关闭电源〔halt〕。

-n 不用init，而是自己来关机。不鼓励使用这个选项，而且该选项所产生的后果往往不总是你所预期得到的。

-c cancel current process取消目前正在执行的关机程序。所以这个选项当然没有时间参数，但是可以输入一个用来解释的讯息，而这信息将会送到每位使用者。

-f 在重启计算器〔reboot〕时忽略fsck。

-F 在重启计算器〔reboot〕时强迫fsck。

-time 设定关机〔shutdown〕前的时间。

13.2 halt—最简单的关机命令

其实halt就是调用shutdown -h。halt执行时，杀死应用进程，执行sync系统调用，文件系统写操作完成后就会停止内核。

参数说明:

-n 防止sync系统调用，它用在用fsck修补根分区之后，以阻止内核用老版本的超级块〔superblock〕覆盖修补过的超级块。

-w 并不是真正的重启或关机，只是写wtmp〔/var/log/wtmp〕纪录。

-d 不写wtmp纪录〔已包含在选项[-n]中〕。

-f 没有调用shutdown而强制关机或重启。

-i 关机〔或重启〕前，关掉所有的网络接口。

-p 该选项为缺省选项。就是关机时调用poweroff。

13.3 reboot

reboot的工作过程差不多跟halt一样，不过它是引发主机重启，而halt是关机。它的参数与halt相差不多。

13.4 init

init是所有进程的祖先，它的进程号始终为1，所以发送TERM信号给init会终止所有的用户进程、守护进程等。shutdown 就是使用这种机制。init定义了8个运行级别(runlevel)，init 0为关机，init 1为重启。关于init可以长篇大论，这里就不再叙述。另外还有telinit命令可以改变init的运行级别，比如，telinit -iS可使系统进入单用户模式，并且得不到使用shutdown时的信息和等待时间。

14 守候进程名字功能对照表

- amd：自动安装NFS（网络文件系统）守侯进程
- apmd:高级电源管理
- Arpwatch：记录日志并构建一个在LAN接口上看到的以太网地址和IP地址对数据库
- Autofs：自动安装管理进程automount，与NFS相关，依赖于NIS
- Bootparamd：引导参数服务器，为LAN上的无盘工作站提供引导所需的相关信息
- crond：Linux下的计划任务
- Dhcpd：启动一个DHCP（动态IP地址分配）服务器
- Gated：网关路由守候进程，使用动态的OSPF路由选择协议
- Httpd：WEB服务器
- Inetd：支持多种网络服务的核心守候程序
- Innd：Usenet新闻服务器
- Linuxconf：允许使用本地WEB服务器作为用户接口来配置机器
- Lpd：打印服务器
- Mars-nwe：mars-nwe文件和用于Novell的打印服务器
- Mcserv：Midnight命令文件服务器
- named：DNS服务器

- netfs：安装NFS、Samba和NetWare网络文件系统
- network：激活已配置网络接口的脚本程序
- nfs：打开NFS服务
- nscd：nscd(Name Switch Cache daemon)服务器，用于NIS的一个支持服务，它高速缓存用户口令和组成成员关系
- portmap：RPC portmap管理器，与inetd类似，它管理基于RPC服务的连接
- postgresql：一种SQL数据库服务器
- routed：路由守候进程，使用动态RIP路由选择协议
- rstatd：一个为LAN上的其它机器收集和提供系统信息的守候程序
- ruserd：远程用户定位服务，这是一个基于RPC的服务，它提供关于当前记录到LAN上一个机器日志中的用户信息
- rwalld：激活rpc.rwall服务进程，这是一项基于RPC的服务，允许用户给每个注册到LAN机器上的其他终端写消息
- rwhod：激活rwhod服务进程，它支持LAN的rwho和ruptime服务
- sendmail：邮件服务器sendmail
- smb：Samba文件共享/打印服务
- snmpd：本地简单网络管理候进程
- squid：激活代理服务器squid
- syslog：一个让系统引导时起动syslog和klogd系统日志守候进程的脚本
- xfs：X Window字型服务器，为本地和远程X服务器提供字型集
- xntpd：网络时间服务器
- ypbind：为NIS（网络信息系统）客户机激活ypbind服务进程
- yppasswdd：NIS口令服务器
- ypserv：NIS主服务器
- gpm：管鼠标的
- identd：AUTH服务，在提供用户信息方面与finger类似

15 设置RH为PPOE拨号服务器

操作系统	DEBIAN2.2升级到3.0
内核	2.4.18
硬件环境	赛扬3 1.1G，内存128M，硬盘40G，主板芯片组694T
显卡	ATI RAGE IIC(4M）
PPPOE服务器	RP-PPPOE 3.4.1
RADIUS服务器	FREERADIUS 0.8
数据库	MYSQL 3.23.56

PPPOE服务器认证的建立分三种认证方式，第一种方式为用RP-PPPOE自带的基于文本认证方式，第二种方式为建立在RADIUS认证服务器上的文本认证方式，第三种方式是基于MYSQL的数据库认证方式

15.1 编译内核

要建立PPPOE服务器，除了内核要支持PPP以外还需要内核支持PPPOE，不过在2.4.18里需要打开内核的不成熟代码才可以选择，内核的配置如下：

```
code maturity level options－－>
[*] prompt for development and/or incomplete code/drivers
networking options－－>
[*] packet socket
[*] packet socket:mmapped io
network device support－－>
[*] ppp (point-to-point protocol) support
[*] ppp multilink support (experimental)
[*] ppp filtering
[*] ppp support for async serial ports
[*] ppp support for sync tty ports
[*] ppp deflate compression
[*] ppp bsd-compress compression
[*] ppp over Ethernet (experimental)
character devices－－>
[*] non-standard serial port support
[*] hdlc line discipline support
```

编辑/etc/modules.conf(redhat好象是conf.modules)，如果没有就加入以下几行：

```
alias char-major-108 ppp_generic
alias /dev/ppp ppp_generic
alias tty-ldisc-3 ppp_async
alias tty-ldisc-13 n_hdlc
alias tty-ldisc-14 ppp_synctty
alias ppp-compress-21 bsd_comp
```

```
alias ppp-compress-24 ppp_deflate
alias ppp-compress-26 ppp_deflate
```

完成以后就可以下一步，建立PPP拨号服务器了

15.2 建立PPP服务器

在建立拨号服务器之前，应确保你在内核中打开了IP转发功能：

```
echo “1”>/proc/sys/net/ipv4/ip_forward
```

编译PPP服务器很容易，就是按以下几步就可以了：

1. 使用configure
2. 使用make编译PPPD，这里有几个参数比较重要，要支持windows的客户端，应该在编译时加上选项USE_MS_DNS=1,如果你的系统shadow的话，你应该使用make HAS－SHADOW=1表示支持shadow密码
3. 编译完成后，将生成pppd、pppdump、chat和pppstats这几个文件，使用make install安装这些文件
4. 修改pppd的访问权限，允许由root启动服务进程

```
chmod u+s /usr/sbin/pppd
```

15.3 建立PPPOE服务器

```
cd /rp-pppoe-3.4/src
./configure
make
make install
```

启动PPPOE进程：

```
pppoe-server －L 10.0.0.1 －R 10.0.0.2 －N 64 －k -u
-L: 指定PPPOE服务器的IP地址
-R:  指定PPPOE拨入服务器分配给客户端的IP地址段
-N:  允许客户端同时拨入的数量（默认是64 最大是65534）
-k: 使用内核方式（不过好象无法使用）
```

修改/etc/ppp/options，查看有没有以下几行，没有就加进去:

```
local
repaire-pap
login
auth
defaultroute
```

```
hide-password
ipcp-accept-local
ipcp-accept-remote
10.0.0.1:10.0.0.255
netmask 255.255.255.0
ms-dns 10.0.0.1
```

修改/etc/ppp/pppoe-server-options，将所有的行都注释掉
添加用户到/etc/ppp/pap-secrets中,例如添加用户luo，密码为123456，允许从任何位置拨入:

```
#client server secret ip addresses
luo * “123456” *
```

设置windows的用户名：luo 密码：123456，如果连接成功就可以进行下一步

15.4 建立RADIUS认证服务器

```
cd /freeradius-0.81
./configure
make
make install
```

编辑/usr/local/etc/raddb/clients，输入你的NAS（网络认证服务器）的IP地址和serect，本例中NAS为本机，serect为linux，看上去如下：

```
localhost linux
```

编辑/usr/local/etc/raddb/clients.conf，加入以下几行：

```
client 127.0.0.1{
secret = linux
shortname = localhost
}
nastype = other
```

编辑/usr/local/etc/raddb/naslist，加入：

```
localhost local
```

编辑/usr/local/etc/raddb/users，加入用户：luo、密码：123456，显示如下：

```
luo Auth-Type:=local, User-Password==“123456”
Service-Type:= Framed-User,
Framed-Protocol = PPP
Framed-IP-Address := 10.0.0.2
Framed-IP-Netmask := 255.255.255.0
```

启动RADIUS服务、测试帐号
启动调试模式

```
RADIUS -X
```

进行帐号测试

```
Radtest luo 123456 localhost 0 linux
```

如果能看到应答，说明RADIUS设置成功了
要使PPPD进程能够使用RADIUS进行认证，需要加入RADIUS认证的插件

```
cp radius.so /usr/lib
```

编辑/etc/ppp/options，加入以下到文件：

```
plugin /usr/lib/radius.so
radius-servers localhost:1812/1813
radius-auth-key linux
radius-ip-pool 10.0.0.2:10.0.0.255
```

运行客户端进行登陆，如果能通过认证则成功了

15.5 建立MYSQL服务器

```
tar zxvf mysql-3.23.56.tar.gz
cd mysql-3.23.56
./configure
make
make install
cd script
./mysql_install_db
/usr/local/bin/safe_mysqld&
```

安装RADIUS数据库

```
cd freeradius-0.81/src/modules/rlm_sql/drivers/rlm_sql_my-
sql
mysql -u root -p 密码 radius<db_mysql.sql
```

编辑/usr/local/etc/raddb/radius.conf使其支持SQL，修改后如下：

```
authorize{
preprocess
chap
mschap
suffix
sql
```

```
}
accounting{
......
sql
......
}
```

修改sql.conf，如下所示：

```
server="loclahost" login="root" password="mysql的root的
密码"
```

加入组帐号

```
mysql -u root -p 密码 radius
insert into radgroupreply (groupname,attribute,op,value)
values ('user','Auth-Tyep',':=','local');
insert into radgroupreply (groupname,attribute,op,value)
values ('user','Service-Type',':=','FramedUser');
insert into radgroupreply (groupname,attribute,op,value)
values('user','Framed-IP-Netmask',':=','255.255
.255.0');
#设定拨入用户的掩码，本例只有PPPOE服务，所以可以与options中
设定相同
```

加入用户帐号：

```
insert into radcheck (username,attribute,op,value) values
('luo','User-Password',':=','123456');
```

将用户帐号加入组帐号

```
insert into usergroup(username,groupname) values('luo',
'user');
```

如果提示无法找到rlm_sql_mysql文件，则:

```
cp /usr/local/lib /usr/lib
```

测试：

```
radtest luo 123456 localhost 0 linux
```

见到应答就OK了！

16 RPM 大全

RPM 有五种基本的操作方式(不包括创建软件包): 安装, 卸载, 升级, 查询,和验证。下面我们就来逐一的讲解吧。

16.1 安装RPM包

RPM 软件包通常具有类似foo-1.0-1.i386.rpm 的文件名。其中包括软件包的名称(foo)，版本号(1.0)，发行号(1)，和硬件平台(i386)。安装一个软件包只需简单的键入以下命令：

```
$ rpm -ivh foo-1.0-1.i386.rpm
foo ##################################
```

RPM安装完毕后会打印出软件包的名字(并不一定与文件名相同)，而后打印一连串的号以表示安装进度。虽然软件包的安装被设计的尽量简单易行，但还是可能会发生如下的错误：

1. 软件包已被安装
   如果您的软件包已被安装, 将会出现以下信息：

   ```
   $ rpm -ivh foo-1.0-1.i386.rpm
   foo package foo-1.0-1 is already installed
   error: foo-1.0-1.i386.rpm cannot be installed
   ```

   如果您仍旧要安装该软件包，可以在命令行上使用`--replacepkgs` 选项，RPM将忽略该错误信息强行安装。

2. 文件冲突
   如果您要安装的软件包中有一个文件已在安装其它软件包时安装，会出现以下错误信息：

   ```
   # rpm -ivh foo-1.0-1.i386.rpm
   foo /usr/bin/foo conflicts with file from bar-1.0-1
   error: foo-1.0-1.i386.rpm cannot be installed
   ```

   要想让RPM 忽略该错误信息，请使用`--replacefiles` 命令行选项。

3. 未解决依赖关系
   RPM软件包可能依赖于其它软件包，也就是说要求在安装了特定的软件包之后才能安装该软件包。如果在您安装某个软件包时存在这种未解决的依赖关系。会产生以下信息：

   ```
   $ rpm -ivh bar-1.0-1.i386.rpm
   failed dependencies: foo is needed by bar-1.0-1
   ```

   您必须安装完所依赖的软件包，才能解决这个问题。如果您想强制安装(这是个坏主意，因为安装后的软件包未必能正常运行)，请使用`--nodeps` 命令行选项。

16.2 卸载RPM包

卸载软件包就象安装软件包时一样简单：

```
$ rpm -e foo
```

注意这里使用软件包的名字name “foo”，而不是软件包文件的名字file “foo-1.0-1.i386.rpm”。

如果其它软件包依赖于您要卸载的软件包，卸载时则会产生错误信息。如：

```
$ rpm -e foo
removing these packages would break dependencies:foo is n-
eeded by bar-1.0-1
```

若让RPM忽略这个错误继续卸载(这可不是一个好主意，因为依赖于该软件包的程序可能无法运行)，请使用`--nodeps` 命令行选项。

16.3 升级RPM包

升级软件包和安装软件包十分类似：.

```
$ rpm -Uvh foo-2.0-1.i386.rpm
foo ##################################
```

RPM将自动卸载已安装的老板本的foo 软件包，您不会看到有关信息。事实上您可能总是使用-U 来安装软件包，因为即便以往未安装过该软件包，也能正常运行。因为RPM 执行智能化的软件包升级，自动处理配置文件，您将会看到如下信息：

```
saving /etc/foo.conf as /etc/foo.conf.rpmsave
```

这表示您对配置文件的修改不一定能向上兼容。因此，RPM 会先备份老文件再安装新文件。您应当尽快解决这两个配置文件的不同之处，以使系统能持续正常运行。

因为升级实际包括软件包的卸载与安装两个过程，所以您可能会碰到由这两个操作引起的错误。另一个你可能碰到的问题是：当您使用旧版本的软件包来升级新版本的软件时，RPM会产生以下错误信息：

```
$ rpm -Uvh foo-1.0-1.i386.rpm
foo package foo-2.0-1 (which is newer) is already installed
error: foo-1.0-1.i386.rpm cannot be installed
```

如果你确有需要将该软件包”降级，加入`--oldpackage` 命令选项就可以了。

16.4 查询已安装的软件包

使用命令rpm -q来查询已安装软件包的数据库。简单的使用命令rpm -q foo 会打印出foo软件包的包名，版本号，和发行号：

```
$ rpm -q foo
foo-2.0-1
```

除了指定软件包名以外，您还可以使用以下选项来指明要查询哪些软件包的信息。这些选项被称之为“软件包指定选项“。

- -a 查询所有已安装的软件包
- -f 将查询包含有文件. 的软件包
- -p 查询软件包文件名为的软件包
  还可以指定查询软件包时所显示的信息。它们被称作信息选择选项：
- -i 显示软件包信息，如描述, 发行号, 尺寸, 构建日期, 安装日期, 平台, 以及其它一些各类信息。
- -l 显示软件包中的文件列表。
- -s 显示软件包中所有文件的状态。
- -d 显示被标注为文档的文件列表(man 手册, info 手册, README's, etc)。
- -c 显示被标注为配置文件的文件列表。这些是您要在安装完毕以后加以定制的文件(sendmail.cf, passwd, inittab, etc)。

对于那些要显示文件列表的文件, 您可以增加-v 命令行选项以获得如同ls -l 格式的输出。验证软件包 验证软件包是通过比较已安装的文件和软件包中的原始文件信息来进行的。验证主要是比较文件的尺寸, MD5 校验码, 文件权限, 类型, 属主和用户组等。

rpm -V命令用来验证一个软件包。您可以使用任何包选择选项来查询您要验证的软件包。命令rpm -V foo 将用来验证foo软件包。又如：

- 验证包含特定文件的软件包：

  ```
  rpm -Vf /bin/vi
  ```

- 验证所有已安装的软件包：

  ```
  rpm -Va
  ```

- 根据一个RPM包来验证：

  ```
  rpm -Vp foo-1.0-1.i386.rpm
  ```

如果您担心你的RPM数据库已被破坏，就可以使用这种方式。

如果一切均校验正常将不会产生任何输出。如果有不一致的地方，就会显示出来。输出格式是8位长字符串, “c” 用以指配置文件, 接着是文件名. 8位字符的每一个用以表示文件与RPM数据库中一种属性的比较结果。“.” (点) 表示测试通过。.下面的字符表示对RPM软件包进行的某种测试失败：

显示字符	错误源
5	MD5 校验码
S	文件尺寸
L	符号连接
T	文件修改日期
D	设备
U	用户
G	用户组
M	模式e(包括权限和文件类型)

如果有错误信息输出, 您应当认真加以考虑，是通过删除还是重新安装来解决出现的问题。

16.5 教你一招

RPM不仅是安装/卸载程序的工具，它还是系统维护和诊断的一把好手。看过下面几个例子你就会领教它的厉害了。

- 如果您误删了一些文件, 但您不能肯定到底都删除了那些文件，怎么办？您可以键入：

  ```
  rpm -Va
  ```

  rpm会在屏幕上显示出文件删除的情况。若你发现一些文件丢失了或已被损坏, 您就可以重新安装或先卸载再安装该软件包。

- 如果您碰到了一个自己不认识的文件，要想查处它属于哪个软件包，您可以输入以下命令

  ```
  rpm -qf /usr/X11R6/bin/xjewel
  ```

  输出的结果会是：

  ```
  xjewel-1.6-1
  ```

- 如果发生综合以上两个例子的情况，如文件/usr/bin/paste出了问题。您想知道哪个软件包中包含该文件，您这时可以简单的键入：

  ```
  rpm -Vf /usr/bin/paste
  ```

- 如果您想了解正在使用的程序的详细信息, 您可以键入如下命令来获得软件包中关于该程序的文档信息：

```
rpm -qdf /usr/bin/ispell
```

输出结果为:

```
/usr/man/man4/ispell.4
/usr/man/man4/english.4
/usr/man/man1/unsq.1
/usr/man/man1/tryaffix.1
/usr/man/man1/sq.1
/usr/man/man1/munchlist.1
/usr/man/man1/ispell.1
/usr/man/man1/findaffix.1
/usr/man/man1/buildhash.1
/usr/info/ispell.info.gz
/usr/doc/ispell-3.1.18-1/README
```

- 您发现了一个新的koules RPM，但您不知道它是做什么的，您可以键入如下命令：

```
rpm -qip koules-1.2-2.i386.rpm
```

- 现在您想了解koules的RPM包在系统里安装了哪些文件，您可以键入：

```
rpm -qlp koules-1.2-2.i386.rpm
```

输出结果为：

```
/usr/man/man6/koules.6
/usr/lib/games/kouleslib/start.raw
/usr/lib/games/kouleslib/end.raw
/usr/lib/games/kouleslib/destroy2.raw
/usr/lib/games/kouleslib/destroy1.raw
/usr/lib/games/kouleslib/creator2.raw
/usr/lib/games/kouleslib/creator1.raw
/usr/lib/games/kouleslib/colize.raw
/usr/lib/games/kouleslib
/usr/games/koules
```

以上只是几个常见例子。随着您进一步的使用RPM，您会发现它的各种功能选项组合可以实现更为强大的RPM包管理功能。

17 RPM命令手册

17.1 安装

命令格式：

```
rpm -i [options] file1.rpm [file2.rpm ... fileN.rpm]
```

或者

```
rpm --install [options] file1.rpm [file2.rpm ... fileN.rpm]
```

参数：

file1.rpm [file2.rpm ... fileN.rpm] ： 将要安装的RPM包的文件名

详细选项：

- -h (or `--hash`) 安装时输出hash记号(``#'')
- --test 只对安装进行测试，并不实际安装。
- --percent 以百分比的形式输出安装的进度。
- --excludedocs 不安装软件包中的文档文件
- --includedocs 安装文档
- --replacepkgs 强制重新安装已经安装的软件包
- --replacefiles 替换属于其它软件包的文件
- --force 忽略软件包及文件的冲突
- --noscripts 不运行预安装和后安装脚本
- --prefix `<path>` 将软件包安装到由`<path>` 指定的路径下
- --ignorearch 不校验软件包的结构
- --ignoreos 不检查软件包运行的操作系统
- --nodeps 不检查依赖性关系
- --ftpproxy `<host>` 用`<host>` 作为FTP代理
- --ftpport `<port>` 指定FTP的端口号为`<port>`

通用选项

- -v 显示附加信息
- -vv 显示调试信息
- --root `<path>` 让RPM将`<path>`指定的路径做为"根目录"，这样预安装程序和后安装程序都会安装到这个目录下
- --rcfile `<rcfile>` 设置`rpmrc`文件为 `<rcfile>`
- --dbpath `<path>` 设置RPM 资料库存所在的路径为`<path>`

17.2 删除

命令格式：

```
rpm -e [options] file1.rpm [file2.rpm ... fileN.rpm]
```

或者

```
rpm --erase [options] file1.rpm [file2.rpm ... fileN.rpm]
```

参数
file1.rpm [file2.rpm ... fileN.rpm] ： 要删除的软件包
详细选项

- --test 只执行删除的测试
- --noscripts 不运行预安装和后安装脚本程序
- --nodeps 不检查依赖性

通用选项

- -vv 显示调试信息
- --root <path> 让RPM将<path>指定的路径做为"根目录"，这样预安装程序和后安装程序都会安装到这个目录下
- --rcfile <rcfile> 设置rpmrc文件为 <rcfile>
- --dbpath <path> 设置RPM 资料库存所在的路径为 <path>

17.3 升级

命令格式

```
rpm -U [options] file1.rpm [file2.rpm ... fileN.rpm]
```

或者

```
rpm --upgrade [options] file1.rpm [file2.rpm ... fileN.rpm]
```

参数
file1.rpm [file2.rpm ... fileN.rpm] : 软件包的名字
详细选项

- -h (or –hash) 安装时输出hash记号(‘‘#’’)
- --oldpackage 允许"升级"到一个老版本
- --test 只进行升级测试
- --excludedocs 不安装软件包中的文档文件

- --includedocs 安装文档
- --replacepkgs 强制重新安装已经安装的软件包
- --replacefiles 替换属于其它软件包的文件
- --force 忽略软件包及文件的冲突
- --percent 以百分比的形式输出安装的进度。
- --noscripts 不运行预安装和后安装脚本
- --prefix <path> 将软件包安装到由 <path> 指定的路径下
- --ignorearch 不校验软件包的结构
- --ignoreos 不检查软件包运行的操作系统
- --nodeps 不检查依赖性关系
- --ftpproxy <host> 用 <host> 作为FTP代理
- --ftpport <port> 指定FTP的端口号为 <port>

通用选项

- -v 显示附加信息
- -vv 显示调试信息
- --root <path> 让RPM将<path>指定的路径做为"根目录"，这样预安装程序和后安装程序都会安装到这个目录下
- --rcfile <rcfile> 设置rpmrc文件为 <rcfile>
- --dbpath <path> 设置RPM 资料库存所在的路径为 <path>

17.4 查询

命令格式：

```
rpm -q options
```

或者

```
rpm \verb+--+query options
```

参数：
pkg1 ... pkgN ： 查询已安装的软件包
详细选项

- -p <file>(or ‘‘-’’) 查询软件包的文件

- -f <file> 查询<file>属于哪个软件包
- -a 查询所有安装的软件包
- --whatprovides <x> 查询提供了 <x>功能的软件包
- -g <group> 查询属于<group> 组的软件包
- --whatrequires <x> 查询所有需要 <x> 功能的软件包

信息选项

- <null> 显示软件包的全部标识
- -i 显示软件包的概要信息
- -l 显示软件包中的文件列表
- -c 显示配置文件列表
- -d 显示文档文件列表
- -s 显示软件包中文件列表并显示每个文件的状态
- --scripts 显示安装、卸载、校验脚本
- --queryformat (or –qf) 以用户指定的方式显示查询信息
- --dump 显示每个文件的所有已校验信息
- --provides 显示软件包提供的功能
- --requires (or -R) 显示软件包所需的功能

通用选项

- -v 显示附加信息
- -vv 显示调试信息
- --root <path> 让RPM将<path>指定的路径做为"根目录"，这样预安装程序和后安装程序都会安装到这个目录下
- --rcfile <rcfile> 设置rpmrc文件为 <rcfile>
- --dbpath <path> 设置RPM 资料库存所在的路径为 <path>

17.5 校验已安装的软件包

命令格式：

```
rpm -V ( or --verify, or -y) options
```

或者

```
rpm --verify options
```

或者

```
rpm -y options
```

参数
pkg1 ... pkgN：将要校验的软件包名
软件包选项

- -p <file> Verify against package file <file>
- -f <file> 校验<file>所属的软件包
- -a Verify 校验所有的软件包
- -g <group> 校验所有属于组 <group> 的软件包

详细选项

- --noscripts 不运行校验脚本
- --nodeps 不校验依赖性
- --nofiles 不校验文件属性

通用选项

- -v 显示附加信息
- -vv 显示调试信息
- --root <path> 让RPM将<path>指定的路径做为"根目录"，这样预安装程序和后安装程序都会安装到这个目录下
- --rcfile <rcfile> 设置rpmrc文件为 <rcfile>
- --dbpath <path> 设置RPM 资料库存所在的路径为 <path>

17.6 校验软件包中的文件

语法：

```
rpm -K [options] file1.rpm [file2.rpm ... fileN.rpm]
\end{quote}
```

或者

```
\begin{quote}
\begin{verbatim}
rpm --checksig [options] file1.rpm [file2.rpm ... fileN.rpm]
```

参数：

file1.rpm [file2.rpm ... fileN.rpm] 软件包的文件名

Checksig–详细选项

- --nopgp 不校验PGP签名

通用选项

- -v 显示附加信息
- -vv 显示调试信息
- --rcfile <rcfile> 设置rpmrc文件为 <rcfile>

17.7 其它RPM选项

- --rebuilddb 重建RPM资料库
- --nitdb 创建一个新的RPM资料库
- --quiet 尽可能的减少输出
- --help 显示帮助文件
- --version 显示RPM的当前版本

18 Conexant蕊片全向QL8410 USB ADSL猫在linux下驱动成功方案一例

首先在安驱动前一定要先了解一些内容：

1. 先要了解清楚你的ISP商给你的一些参数情况，比如你的ISP商给你的封装方式是PPPOE，PPPOA，是ATM LLCSNAP 上的路由IP（RFC1483），还是向我一样是ATM LLCSNAP 上的桥接IP（RFC1-483），还有就是你的IP，子网掩码，网关，DNS（当然PPPOE，PP-POA是没有这些参数的，这两种方式应该有用户名和密码，不过我不是用得这两种方式所以这方面我了解的不多，而ATM LLCSNAP 上

的路由IP （RFC1483），ATM LLCSNAP 上的桥接IP（RFC1483）才有这些参数），再有就是你的VCI，VPI，这两个参数在这个驱动里是很重要的，以上这些参数我是在我的win下的驱动里找到的，我想大家应该也能在win下的驱动里找到，找不到就问问IPS商应该能得到满意的回答的；

2. 是选择linux系统，我用的是RedHat9（内核版本为2.4.20-8），在这个系统下，用我这种封装协议是不用设置内核的（我的是ATM LLC-SNAP 上的桥接IP （RFC1483））别得系统可能需要设置内核；

3. 是要在win下下载一个RPM包，linux-atm-2.4.1-1.i386.rpm 。查的方法是rpm -q libusb-0.1.6-3，上面的linux-atm-2.4.1-1.i386.rpm包也可以查你的系统是否已经安装了）；

   linux-atm-2.4.1-1.i386.rpm包下载地址如下：

   ```
   ftp://ftp.chg.ru/.5/sourceforge/accessrunner/linux-at-
   m-2.4.1-1.i386.rpm
   ftp://ftp.chg.ru/.5/sourceforge/linux-atm/linux
   -atm-2.4.1-1.i386.rpm
   ```

4. 如果你的linux没有安装这三个包，在X下你直接双击就可以安装了，或在终端下用rpm -Uvh 文件名，来安装；

5. 就是要设置linux下的内核了，（如果和我一样用的是ATM LLC-SNAP 上的桥接IP （RFC1483）封装协议就不用设置内核了直接跳到第7步就行了），内核设置先进入/usr/src下，建立一个名为linux文件夹，然后把这个文件夹与内核文件夹（在RedHat9下名为linux-2.4.20-8）联接起来，也可以在RedHat的GNOME下直接建立文件夹联接（点右键就能看到了）内核设置先在终端下进入到/usr/src/linux下，执行make xconfig(或make menuconfig)，设置内容如下：

   ```
   Code maturity level options
   [*] Prompt for development and/or incomplete code/driv-
   ers
   Loadable module support
   [*] Enable loadable module support
   [*] Set version information on all module symbols
   [*] Kernel module loader
   Networking options
   <*> Packet socket
   <*> Unix domain sockets
   [*] TCP/IP networking
   [*] Asynchronous Transfer Mode (ATM)
   USB support
   <*> Support for USB
   ```

```
[*] Preliminary USB device filesystem
<M> UHCI
<M> OHCI
```

如果使用的是RFC1483/2684 路由协议还要加上以下项目

```
Networking options
[*] Asynchronous Transfer Mode (ATM)
[*] Classical IP over ATM
```

如果是PPPOE协议要加上以下项目

```
Networking options
[M] RFC1483/2684 Bridged protocols
Network Device Support
<M > PPP (point-to-point protocol) support
<M> PPP Deflate compression
<M> PPP over Ethernet (EXPERIMENTAL)
```

如果是PPPEA协议要加上以下项目

```
Network Device Support
<M > PPP (point-to-point protocol) support
<M> PPP Deflate compression
<M> PPP over ATM
```

设置这一项

```
Processor type and features
[ ] Symmetric multi-processing support
```

如果可以你可以把上面这些都设置上不用分什么PPPOE、PPPOA

6. 设置完之后就要在终端上执行make dep命令了，不用编绎内核！！！我可是在这里搞了很长时间，最后才发现不用编绎内核，只用make dep建立文件联接就行了

7. 然后进入/usr/src/linux/configs目录复制kernel-2.4.20-i686.config文件到/usr/src/linux下并改名为config（这一步很关键！我以前一直不能安装驱动成功就是因为这个文件没有复制到linux目录下。）

8. 现在就可以把已经下好的驱动解压到/usr目录下了解压后的文件夹名为cxacru，进入到/usr/cxacru目录执行make就可以安装了，

9. 安装完后要进入/etc目录打开cxacru文件设置参数了，内容如下：

```
# Config file for Conexant AccessRunner
# Driver mode （这一项可以不用改）
DRIVER_MODE=1 # 1 = normal, 2 = debug, 3 = normal+max
speed (without ask adsl status), 4 = debug+max speed
(without ask adsl status)
# Protocol （这一项可以选择你使用的协议类型，跟我一样的
协议用3）
PROTOCOL_MODE=1 # 1 = RFC1483/2684 routed, 2 = PPP o-
ver ATM (pppoa), 3 = RFC1483/2684 bridged, 4 = PPP o-
ver Ethernet (pppoe)
# Paths （这一项不用改）
BINARY_PATH="/usr/sbin"
ATM_PATH=""
# ATM （这里改成你的VPI，VCI）
VPI=8
VCI=32
（下面这一项如果你用的是RFC1483/1684 routed/bridged 加
上你的参数就行了）
# Specific for RFC1483/2684 routed/bridged
IP_ADDRESS=
NETMASK=255.255.255.0
GATEWAY=
```

10. 如果你用有DNS那就进入/etc/resolv.conf 加入如下参数

    nameserver 194.224.52.37 （地址改成你的就行了，如果是多地址就再加上一行nameserver 地址）

11. 到这一步要是跟我一样用桥接，有IP，子网掩码，网关，DNS的就可以在终端下用cxstart.sh命令驱动猫了。（以下的内容是我的一些联想，我因为不是使用这些协议，所以我没有做过试验，驱动上的西文说明写着这些）

12. 如果是用PPPOE、PPPOA协议的就还要在第10步的基础上再设置一些东西，使用PPPOE的进入/etc/ppp/options 设置内容如下：

    ```
    lock
    defaultroute
    noipdefault
    noauth
    holdoff 4
    maxfail 25
    persist
    usepeerdns
    ```

```
asyncmap 0
name adslppp@telefonicanetpa
user adslppp@telefonicanetpa
lcp-echo-interval 2
lcp-echo-failure 7
plugin /usr/lib/pppd/plugins/pppoe.so
nas0
```

进入/etc/ppp/chap-secrets加入如下内容：

```
adslppp@telefonicanetpa * adslppp
```

进入/etc/ppp/pap-secrets加入如下内容：

```
adslppp@telefonicanetpa * adslppp
```

都设好后就可以用cxstart.sh来启动了，用cxstop.sh来关闭驱动。

上面就是我的安装过程，望各位兄弟指证，如果跟我用的系统、协议和猫一样的兄弟应该可以在第11步就成功了，使用其它猫，和协议的兄弟当做参考吧，希望这篇心得能够对广大使用USB接口的ADSL猫的兄弟们起到作用，我在这里也祝大家能成功，我想只要坚持，成功总会到来的！！！

19 uperblock corrupt后的修复工作

在linux文件系统里,superblock扮演着相当重要的角色,它记录了inode表和空闲块表在硬盘上的存放位置等重要信息,由于它很重要,所以系统自动将superblock进行冗余保存,在恢复严重瘫痪的文件系统时可能要对它进行操作.

前几天我就遭遇乐superblock corrupt而被它"折磨"到现在才解脱,我想把这一次经验教训贴出来,希望以后对朋友们处理这样的"灾难"有所帮助.

出现superblock corrupt的原因基本是非正常关机(断电)引起的,也有例外情况,我听说有人在用samba时shutdown就出现了superblock corrupt,当然有的也可能是假象,比如你的硬盘是scsi硬盘,系统崩溃后用boot/rescue缺省软盘启动,那么你e2fsck任何一个分区都会得到superblock corrupt的提示,这并不是e2fsck比较笨,因为scsi硬盘没有驱动和superblock corrupt对于他来讲都是一样的–找不到inode头.

当你的系统出现superblock corrupt而无法启动时,不要着急慢慢来.

1. 用应急盘启动,先看fdisk的结果.如果你的分区表看起来正常,那么恢复的可能性就比较大了,如果出现cannot open /dev/sda2的提示,那么想一想你的scsi卡启动没有,如果没有,那么你可以试着用小红帽的安装光盘启动,记住,仅仅是看分区表,千万不要写它.然后把分区情况详细记录下来.

2. 试着e2fsck /dev/hda2,(先不要加-p -y 之类的参数,)用手动进行修复,同时也可以了解具体是文件系统的那些地方损坏了,如果你的运气好,e2fsck过去了,那么恭喜你,/dev/hda2已经基本修复,当然修复的可能是99.9%,也可能是99%,这就看文件系统的损坏程度乐,不过现在可以说你的数据已经都找回来了.剩下的事就是mount上把数据备份出来以防万一.

3. 如果e2fsck没过去(确保你的硬盘已经正确驱动了),也不要着急,因为superblock在硬盘中有很多地方有备份,现在你最好把硬盘卸下来挂到另一个好的linux系统上,当然同样要保证硬盘被正确驱动乐.先用e2fsck /dev/hda2,如果结果和前面一样就用e2fsck -b xxx -f /dev/hda2, xxx是硬盘上superblock的备份块,xxx=n*8192+1,n=1,2,3...一般来讲,如果系统瘫痪的真正原因是superblock损坏，这种办法就应该可以恢复你的数据了。如果执行后的结果还是不能通过,那么往下一步.

4. 利用dd命令.先dd if=/dev/hda2 of=/tmp/rescue conv=noerror(/tmp/-rescue是一个文件),把重要的数据拷出来,当然,这个盘要比你损坏的盘大一点,否则拷不下.另外,上面的dd命令在不同的境况下if和of应作相应的修改，写在这里只是一个例子，总之在用dd之前最好先看看man.刚才你已经看到你的分区表了,现在找一个和你的硬盘一样的硬盘,应该是一摸一样（大小，型号),在这块硬盘上按照坏盘上的分区表分区，分的区也应该是也是一模一样然后用dd命令把坏盘上superblock location后的东东全部拷到好盘的superblock location后，愿上帝保佑你，当你再次启动系统时就可以看到熟悉的数据了,有人用这种方法恢复了99%以上的数据,不过好在这种方法(包括前面的方法)都没有动那块坏盘上的数据,如果还是没有恢复,那没你还有最后一种选择.

5. 在手册页里称这种方法为last-ditch recovery method,就是说这是最后的恢复方法，只有当你已经尝试了其他的方法,都没有能恢复你的数据的情况下才用,因为这需要冒一定的风险.把你的硬盘挂在一台好的linux box上，运行：mke2fs -S /dev/hda2(如果你的数据在hda2里）这条命令只重建superblock，而不碰inode表，不过这仍有一定的风险。good luck to you all.

当时也有人建议我如果实在不行的话就重装系统（不动分区也不格式化），这也可能有效，但你也应该清楚这种方法就像mke2fs -S /dev/hd*一样是有风险的。

一点建议：

如果你的硬盘不是可以轻易就重做的，最好在建立一个新的系统后：

1. 拿出笔和纸,把你的分区信息详细记录下来.

2. 用mkbootdisk做好现在这个系统的启动盘并测试.特别是如果你用的硬盘是scsi的。

3. 在用mke2fs建立一个文件系统后将屏幕上的superblock所在位置记录下来。

4. 用crontab对重要数据进行备份。

ext2文件系统（包括其他的unix文件系统）是很强壮的，但你仍然应该小心的呵护他。

20 如何在Linux 环境中使用USB 存储器

如何在Linux 环境中使用USB 接口的存储设备？这是各大电脑论坛上出现得比较多的一个问题，同此可见这也是摆在许多电脑玩家面前的一道难题。本文就为您提供一套完美的解决方案，通过下面的方法，您仅可以在Linux 环境中方便地使用U 盘、USB 硬盘盒，而且还能将数码相机作为USB 大容量存储器。这里就以Red Hat Linux 7.2 为例，来一步一步地介绍如何在Linux 环境中使用USB 存储器。

1. 在X-Windows 环境中打开“控制中心”，展开“信息→USB 设备”控制台树，查看Linux 是否已经正确识别您的USB 存储器，如果没有看到USB 设备信息，那么您就应该要检查USB 端口是否已经在BIOS 中打开，或USB 存储器与电脑的物理连接有没有连接好了。图1 所示为笔者的FinePix 数码相机信息。

2. 在控制台状态下输入如下命令：

   ```
   fdisk -l /dev/sda
   ```

   注意：不要在sda 后面加通配符“*”或“?”，否则运行命令后返回的信息将不正常。如果您有多个USB 设备话，则设备名在Linux 下分别表示为sda、sdb、sdc 等。

   运行上面的命令后，笔者的电脑上返回如图2 所示信息，表示系统已经找到笔者的USB 设备，设备名称为/dev/sda1、可启动（Boot 下面的那个*表示是活动分区），容量8MB、文件系统为FAT12。

3. 在/mnt 目录下建立一个挂装USB 存储器的目录：

   ```
   mkdir /mnt/usb
   ```

4. 然后再运行装载设备命令，将USB 设备挂装到/mnt/usb 目录下：

   ```
   mount -t msdos /dev/sda1 /mnt/usb
   ```

   注意：如果在图2 中显示您的USB 设备的文件系统为FAT32，请使用mount -t vfat /dev/sda1 /mnt/usb 命令装载USB 设备。

5. 运行如下命令即可查看USB 存储器中的文件信息：

```
ls /mnt/usb
```

然后您就可以像对待硬盘一样对USB 存储器中的数据进行拷贝、删除等操作了。

如果您不习惯于在控制台下敲敲打打，还可以在KDE 桌面上建立一个指向USB 存储器的快捷方式，用鼠标完成所有的文件操作，就像在Windows 中一样简单，建立快捷方式的操作非常简单：用鼠标在桌面空白处单击鼠标，从弹出的快捷菜单中选择“新建/硬盘”，右键单击新创建的快捷方式，选择“属性”菜单命令，切换到“设备”选项卡，分别输入USB 存储器的设备名称、安装点和文件系统类型，如图3 所示。以后只需要单击该快捷方式即可方便地访问USB 存储器了。

21 关于多操作系统共存的研究报告

多操作系统共存引起普通电脑用户的关注还是上个世纪微软推出Windows 95的时候，为了与大量DOS下的应用程序兼容，许多用户不得不在DOS 6.22与Windows 95之间进行双系统引导。随着软件业飞速发展，这一现象逐渐过渡到Windows 98与Windows 2000、Windows XP甚至其他类型操作系统间的双引导，多引导。

多操作系统共存首先解决的是兼容性问题。基于某个操作系统平台设计的软件或硬件在另一个操作系统平台上通常无法运行，或者运行不正常，最有效最直接的解决办法就是将它们安装在各自所支持的操作系统上，搭建两个PC系统显然成本太高，多系统共存是最经济的办法，只需在不同的操作系统间切换即可达到目的；有时候一台电脑需要提供给多个用户使用，不同的用户需求也不同：有的用户仅仅为的是娱乐；有的用户需要进行图形处理，稳定性显然非常重要；搭建个人网站使得操作系统提供的网络服务显得特别重要。多系统共存为不同的用户提供所需的操作系统，它们之间互不干扰和平共处；更多的情况下大家只是为了体验新操作系统带来的新理念，原有的操作系统作为大本营不能够轻易放弃，那么多操作系统共存则是最佳的选择。

21.1 技术准备

不同类型的操作系统所采用的硬盘分区格式是不相同的，而且不兼容，为了大家更好地掌握多操作系统安装，首先让我们准备一下硬盘分区格式之间的基础知识。

21.1.1 常见的硬盘分区格式

1. FAT12

   它是一种相当“古老”的磁盘分区格式，与DOS同时问世，它采用12位文件分配表并因此而得名。FAT12能够管理的磁盘容量极为有限，目前除了软盘驱动器还在采用FAT12之外，基本上已经没有什么地方能找到它了。

2. FAT16

   这是MS-DOS和早期Windows 95操作系统最常使用的磁盘分区格式。它采用16位文件分配表，硬盘容量最大支持2GB，是目前所获支持最广泛的一种磁盘分区格式，几乎所有的操作系统都支持这一种格式，DOS、Windows系列，甚至独树一帜的Linux都支持这种分区格式。但是FAT16分区格式存在巨大的缺点：大容量磁盘利用效率低。在微软的DOS和Windows系统中，磁盘文件的分配以簇为单位，一个簇只分配给一个文件使用，不管这个文件占用整个簇容量的多少。这样，即使一个很小的文件也要占用一个簇，剩余的簇空间便全部闲置，造成磁盘空间的浪费。由于分区表容量的限制，FAT16创建的分区越大，磁盘上每个簇的容量也越大，造成的浪费也越大。所以为了解决这个问题，微软推出了一种全新的磁盘分区格式FAT32，并在Windows 95 OSR2及以后的Windows版本中提供支持。

3. FAT32

   顾名思义，这种格式采用32位的文件分配表，磁盘的管理能力大大增强，突破了FAT16 2GB的分区容量限制。由于现在的硬盘生产成本下降，其容量越来越大，运用FAT32的分区格式后，我们可以将一个大硬盘定义成一个分区而不必分为几个分区使用，大大方便了对磁盘的管理。FAT32推出时主流硬盘空间并不大，所以微软设计在一个不超过8GB的分区中，FAT32分区格式的每个簇容量都固定为4kB，与FAT16相比，大大减少磁盘的浪费，提高磁盘利用率。目前，支持这一磁盘分区格式的操作系统有Windows 95 OSR2/Windows 98/Windows 98 SE/Windows Me/Windows 2000/Windows XP，Linux Redhat部分版本也对FAT32提供有限支持，如果将Linux系统安装在FAT32分区下，必须使用软盘进行引导。但是这种分区格式也有它明显的缺点，首先是由于文件分配表的扩大，运行速度比采用FAT16格式分区的磁盘要慢，特别是在DOS 7.0下性能差距更明显。另外，由于早期DOS不支持这种分区格式，所以无法再使用早期DOS系统。

4. NTFS

   NTFS为Windows NT操作系统而生并随着Windows NT4跨入主力分区格式的行列，它的优点是安全性和稳定性极其出色，在使用中不易产生文件碎片。NTFS分区对用户权限作出了非常严格的限制，每个用户都只能按照系统赋予的权限进行操作，任何试图超越权限的操作都将被系统禁止，同时它还提供了容错结构日志，可以将用户的操作全部记录下来，从而保护了系统的安全。但是NTFS分区格式兼容性不好，特别是对使用广泛的Windows 98 SE/Windows Me系统，它们还需借助第三方软件才能操作NTFS分区。微软近期推出Windows XP并结束了Windows 9X系列的开发，Windows XP基于NT技术提供完善的NTFS分区格式支持，看来微软对推广NTFS分区格式有相当大的决心。NTFS分区也在不断升级中，目前Windows 2000所支持的为NTFS 5.0。

5. Ext和Swap

   Linux是近年来炒作最多、呼声最高的操作系统，版本繁多，支持的分区格式也不尽相同，但是它们的Native主分区和Swap交换分区都采用相同的格式——Ext和Swap。和NTFS分区格式相似，这两种分区格式的安全性与稳定性极佳，使用Linux操作系统死机的机会将大大减少。但是目前支持这类分区格式的操作系统只有Linux。Ext和NTFS类似也有多种版本，目前最新的Linux Redhat 7.2版本已经支持Ext3了。

21.1.2 不同分区格式的转换

由于操作系统的变更和升级，往往需要转换不同的分区格式。中国用户使用最广泛的是微软的DOS、Windows 95、Windows 98、Windows NT/2000/XP系列，分区格式也由早期的FAT16向FAT32与NTFS进化，由于种种原因，特别是多系统共存和系统升级，用户需要从一种分区格式向另一种分区格式迁移，而这3种分区格式间的相互转换频率最高。分区格式的转换不外乎两种途径，操作系统本身提供的转换工具和第三方磁盘工具，后者中最富盛名的是PowerQuest PartitonMagic分区魔术师（以下简称PQMagic）。我们创建了FAT16、FAT32、NTFS分区进行图解，分区情况如右图：

分区格式化是PQMagic提供的功能之一，它可将分区转变成FAT16/FAT32/NTFS甚至Linux的Ext2/Swap，但是它带有破坏性质，下面就不再论述了。

1. FAT16向FAT32及NTFS的转换

   FAT16分区格式是目前所有微软操作系统都支持的格式。Windows 95 OSR2以后的Windows版本均提供FAT16向FAT32转换的工具，Windows NT/2000/XP都有将FAT16转换为NTFS的系统工具，如Convert命令。PQMagic支持FAT16向FAT32的转换，在需要转换的分区上点击鼠标右键，选择“Convert”，弹出的对话框将列出你所能转换的分区格式，确定并选择，然后点击“OK”。Windows 98自带的Fdisk也提供FAT32支持，但是转换过程将破坏原有FAT16磁盘上的数据，在这里我们不推荐大家使用。Windows NT/2000/XP下，PQMagic（Windows版）会提供向NTFS的转换，但是这一功能在DOS和Windows 98下没有提供。

   Windows NT/2000/XP提供FAT16/FAT32向NTFS转换的命令Convert，格式为：Convert 盘符（加冒号）/FS：NTFS。

2. FAT32向FAT16及NTFS的转换

   PQMagic支持FAT32向FAT16的转换，而向NTFS分区转换则必须在Windows NT/2000/XP下进行，两者的操作与“FAT16向FAT32及NTFS的转换”中“FAT16转FAT32”、“FAT16转NTFS”对应操作相似，读者朋友可参照进行。

3. NTFS向FAT16及FAT32的转换

   微软似乎不打算让NTFS格式的使用者发生“倒退”，所以打算尝试NTFS的用户可要小心了，虽然PQMagic支持NTFS向FAT32的转换，但是考虑到NTFS分区格式的特殊性，其耗时也许会较长，不如备份数据进行格式化来得痛快。FAT16似乎气数已尽，我们只有通过NTFS转FAT32转FAT16才能无损转换，耗时耗电耗硬件。

21.2 多操作系统安装详解

21.2.1 正常安装多系统

1. Windows 98 SE、Windows 2000 Professional共存

   Windows 98 SE是玩家们非常偏爱的游戏平台，而追求稳定性和新功能的Windows用户则更加喜爱Windows 2000 Professional系统。对于中国的大量计算机用户来讲，分开专用的娱乐用机和办公用机并不现实，因此很多人需要在自己的计算机上同时安装Windows 2000 Professional和Windows 98 SE两个操作系统。虽然都是微软的产品，但是这两个系统并不是随便怎么安装都相安无事的。

   (a) 按照先Windows 98 SE，后Windows 2000 Professional的顺序安装

   我们在Windows 98 SE操作系统存在的情况下，分别在窗口状态下运行Winnt32.exe安装，在DOS（Windows 98所带DOS 7）状态中运行Winnt.exe安装。在安装选项中选择“全新安装”，安装过程都非常顺利。Windows 2000 Professional安装完成后，这样每次启动时，我们可以通过双系统引导菜单来选择使用Windows 98 SE或是Windows 2000 Professional操作系统。

   应该提醒用户注意的是，如果要将Windows 2000 Professional操作系统与原Windows 98 SE系统安装在同一个硬盘分区时，这个硬盘分区只能是FAT16或FAT32文件格式。如果Windows 98 SE的安装分区使用NTFS文件格式，会导致Windows 98 SE无法启动，而NTFS文件格式对于保证Windows 2000的稳定性是相当重要的，所以明智的做法是将Windows 98 SE和Windows 2000 Professional分别安装在两个不同的分区中，并且在c区，即引导分区保持FAT16或FAT32文件格式。同样，在使用不同安装顺序时，我们也发现选择硬盘分区的文件格式对于安装多操作系统是相当重要的。

   (b) 按照先Windows 2000 Professional，后Windows 98 SE的顺序安装

   我们首先使用Windows 2000 Professional操作系统的安装默认选项，即Windows 2000 Professional安装在硬盘C区，并且将

此分区转换为NTFS文件格式。我们发现，在这种条件下，根本无法完成Windows 98 SE的安装。这主要是因为Windows 98 SE没有提供对NTFS文件格式的支持，因此在安装过程进行到创立Windows 98 SE引导信息时，安装程序会因无法找到引导分区而报错。当然，此时引导分区是存在的，但是Windows 98 SE及其安装程序都不能识别。

我们将引导分区转换为FAT16或FAT32文件格式后，Windows 98 SE可以安装完成（默认选项安装）。在进行了多次安装后，我们发现按照这种顺序安装Windows 98 SE与Windows 2000 Professional，一般会造成不能引导系统，Windows 98 SE破坏了Windows 2000 Professional的引导信息。如果情况是简单的否定也罢了，奇怪的是有两次我们又能正常安装而且引导也很正常！造成这种现象的原因我们一直没有找到。在这种情况下，唯一的问题是由于Windows 2000 Professional和Windows 98 SE共用C：分区中某些同名目录以及其中的某些同名程序（如Program files目录以及Internet explorer等），因此安装Windows 98 SE后会造成这些程序被破坏，在两个操作系统中我们都无法调用它们。而这种现象在我们以先Windows 98 SE后Windows 2000 Professional的顺序安装时并没有出现，因此将Windows 98 SE安装到其他分区是最好的选择。

由于上面提到的引导问题，我们不推荐按照这种顺序安装。

2. Windows Me、Windows 2000 Professional共存

在Windows 98 SE后，微软推出了Windows 9X系列的“最终版”Windows Me，与其前辈相比，它的稳定性并没有大的提高，而且还存在一些兼容性的问题。但是作为Windows 9X系列最新的产品，Windows Me因为它的华丽界面，更接近Windows 2000 Professional的操作体验等原因，还是成为相当多用户的选择。而我们更感兴趣的是，作为Windows 2000 Professional之后推出的版本，在与Windows 2000 Professional组成多操作系统时，它与Windows 98 SE是否有什么不同。同样，我们使用两种安装顺序来构建多操作系统。

我们在安装了Windows Me的硬盘上安装Windows 2000 Professional操作系统，操作过程和使用Windows 98 SE时完全一样，甚至在安装选项中仍然有“从当前操作系统升级”的选项，从发布时间上说，这里的“升级”这两个字用得不太合适。

接下来的测试中，我们将上面的安装顺序反过来，先安装Windows 2000 Professional后安装Windows Me。因为Windows Me能够正确识别Windows 2000 Professional引导信息并加入自身的引导信息，所以两个操作系统都可以顺利地安装使用。

3. Windows 98 SE、Windows 2000 Professional、Redhat7.2、BeOS共存

随着Linux的崛起，越来越多的PC用户开始将目光投向它，但当前的实际情况是Linux还不可能替代Windows系统在桌面操作系统的

统治地位，对于那些既想体验Linux又不想破坏原有Windows系统的PC用户来说，多系统引导是唯一的办法。本小节主要讨论的正是Windows操作系统和Linux如何共存的问题。最后再向在家介绍一下新型多媒体操作系统BeOS的安装。

(a) 按照先Windows 98 SE，后Windows 2000 Professional、Redhat7.2至BeOS的顺序安装

如何对硬盘分区是正确安装Linux所必须的也是很关键的一步。首先我们应该了解Linux系统所专用的分区格式和Windows系统的是完全不同的，因此我们不能简单地按照以往安装Windows系统那样对硬盘进行分区。通常情况下，硬盘都会分为一个主分区（Primary Partition）并设置为激活状态（Active），同时我们还会分出扩展分区（Extended Partition）并在其上细分若干逻辑分区（Logical Partition），激活的主分区在Windows系统（包括早期的DOS系统）下作为引导分区。Windows系统与Linux系统共存按照这种分区方法显然是不行的，为了方便操作，我们以功能强大的PowerQuest PartitonMagic为例进行讲解。

步骤1 准备

启动DOS版PowerQuest PartitonMagic，我们可以使用拷贝有PQM5的DOS启动软盘进行这一步，当然，从引导速度来考虑推荐使用光盘启动。

步骤2 创建分区

因为Linux系统的特殊要求，我们需要单独为它创建分区，建议将它放在扩展分区以前。我们的做法是创建两个主分区：其中一个是Windows系统引导分区，我们需要激活它（Set Active），因为安装Windows 98 SE的缘故，我们选择FAT32格式，这也是Windows 2000 Professional所支持的；另一个留给Linux作为它的“/”根分区使用，其大小可以自己确定，我们建议不要低于800MB。如果要安装大量的软件，2G以上比较合适。分区的格式选择Linux Ext2，如果希望在Linux安装过程中再进行创建分区工作，可以将这个分区删除掉，但一定要保证在扩展分区前面有足够的空间留给Linux使用，注意PQMagic对这段未定义的空间会显示为“Unallocated”。Linux与Windows系统不同，它需要单独的交换分区“Swap”，它对创建位置没有要求，既可以创建在扩展分区前面，也可以在后面，大小与内存容量相同或略大皆可。随后就是扩展分区和逻辑分区的创建，大家对这两步操作都应该比较熟悉，不再赘述。

应该注意的是，虽然我们使用的Redhat7.2支持将“/”根分区放在逻辑分区上而且可以使用硬盘引导，但有些版本的Linux则不支持或需要软盘引导。这就是我们采取上述分区方式的原

因。

注：因为DOS刷新率实在太低，为了尽量提供清晰的图片给读者，我们是对Windows版的PowerQuest PartitonMagic进行的截图，它和DOS版的MPowerQuest PartitonMagic操作完全一样。

步骤3 Windows系统安装

退出PQMagic进行Windows 98 SE以及Windows 2000 Professional的安装。我们将Windows 98 SE安装在C盘，这也是绝大多数用户的选择。Windows 2000 Professional系统盘可以和Windows 98 SE系统盘安装在同一分区，但因为目录重叠问题建议将两系统盘区分开来。我们将Windows 2000 Professional安装在D盘，以保证两者都可正常使用。

步骤4 Redhat7.2的安装

Linux的安装我们选择了比较流行的发行版本Redhat7.2，两张光盘。相信绝大多数用户对Redhat安装还不甚了解，我们简单的向大家介绍一下它的安装过程。

早期Linux安装相当复杂，随着版本的深入以及各发行公司不断的包装，它的安装过程趋于简单化和人性化。我们选择的Redhat7.2是最新的发行版本，首先对BIOS进行设置采用光盘引导，插入Redhat7.2第一张光盘，光盘启动后即进入安装界面。第一个需要用户作出选择的是安装过程中使用的语言，有德、意、法、日、韩等15种，可惜没有中文，缺省状态是英文。接下来是鼠标、键盘的选择，没有特别情况的话一路点NEXT。欢迎画面以后进入安装类型选择的画面，用户可以选择工作站（Workstation）、服务器（Server）、膝上电脑（Laptop）和自定义（Custom），为了领略Redhat的风采我们选择了自定义。

接下来的操作非常重要，安装程序会让用户决定磁盘分区策略。为了清楚的介绍Linux分区知识，我们选择第二项选项然后点击NEXT。之前我们已经用PQMagic为它划定了分区，所以可以很清楚地看到PQMagic创建的Ext2分区，即hdc2，但是该分区尚未格式化，而且没有标出“Mount Point”载入点。点击“Edit”，弹出的对话框会有“Mount Piont”选择，务必选择“/”并点取“Format”，然后“OK”结束。

安装程序会给出警告，提示系统会格掉原有的数据，点击“OK”即可。

接下来是引导程序的选择，用过Linux的朋友一定对LILO不陌生，它是Linux常用的引导程序。现在我们又多了一个GRUB，既然是缺省选项，我们“NEXT”就是。下一步设置的是引导密码，注意不是系统密码，没有特殊要求还是不要设置为好，一旦忘记将会很痛苦。接着是对网络进行配置和设置防火墙。

然后是支持语言的选择，终于出现了中文，而且有简繁体选择。屏幕右上方的选项是系统缺省语言选择，当然是中文啦。时区的选择似乎Redhat做得比Windows更漂亮。

接着是设置Linux Root帐号的密码，Root相当于Windows NT/2000/XP的Administrator，与Windows系统不同的是Linux的Root帐号必须设置密码以确保安全。网络设置以后是安装组件的选择。想拥有XWindows和KDE吗？选上吧！如果想要一个服务器，请选上所有的组件，完全安装需要大约2G硬盘空间。接下来系统会检测电脑上的显卡，一般主流显卡都在支持之列。必须承认微软在兼容性以及获得的支持上具有极大的优势，各位可以看看右边的滚动条，所支持的显卡并不是很多。以后就是文件的拷贝了，接着是重启，然后……安装结束。整个过程仅需20分钟左右，是不是太快了点？呵呵。

步骤5 多系统引导
重启电脑后将见到GRUB多系统引导画面，DOS选项统指微软系列操作系统，选择它则可进入Windows 98SE/Windows 2000 Professional启动菜单。

步骤6 BeOS的安装
BeOS是由Be公司开发的一个免费的新型多媒体操作系统，最为突出的是它的多媒体处理能力，播放效果出众。Personal Edition个人版BeOS安装基于Windows系统，Windows 98 SE下只要双击安装程序即可，和普通应用程序的安装完全一样！双击桌面图标，即可退出Windows进入BeOS。

至此，我们实现了Windows 98 SE、Windows 2000 Professional、Redhat7.2和BeOS的多系统安装及引导。

(b) 按照先Redhat7.2，后Windows 98 SE、Windows 2000 Professional的顺序安装

经过多次尝试，我们认为按这种顺序安装将使3种操作系统的正常引导及使用变得相当麻烦。无论采取何种分区方式，Windows 98 SE安装程序都会破坏Redhat引导扇区并代之以自身的引导信息。Windows 98 SE与Windows 2000 Professional共存没有任何困难，而要使Redhat能恢复使用则必须重新安装，安装顺序与先Windows后Linux无异。还有一个解决办法是为Redhat创建启动软盘，Redhat安装的最后一步即是，如果不幸选择跳过，也只有重新安装。因此我们建议各位不要以这种顺序进行安装，复杂且没有必要。

4. Windows 98 SE、Windows 2000 Professional、Windows XP Professional共存

近期，微软推出了Windows XP，它不仅继承了Windows 98优异的游

戏性能，还有Windows 2000的稳定。好奇的朋友准备在自己的电脑上安装Windows XP好好体验一番，让我们看看这三者如何相处。

和上面的小节相比，现在讨论的共存问题则要简单许多，Windows 98 SE、Windows 2000 Professional和Windows XP Professional可以说是兄弟部队，安装起来并不费事。按照Windows 98 SE至Windows 2000 Professional再Windows XP Professional的顺序安装，系统将不会有任何问题，唯一应该注意的是把不同的操作系统安装到不同的分区以免“打架”。

比较有趣的现象出现在Windows 98 SE和Windows XP Professional的安装顺序上。以往要实现多个微软操作系统共存都是遵循先低后高（版本）的原则，以保证新版本引导程序的运行，而Windows 98 SE与Windows XP Professional共存无论是谁先装，都可以正常使用。因此，大家可以放心地体验Windows XP Professional，即使不满意，再装一个Windows 98 SE双系统共存即可。

21.2.2 多硬盘造成的影响

随着硬盘价格的不断下降，许多用户的电脑里都有了两块甚至多块硬盘，磁盘空间的增大，为安装多个操作系统创造了条件。然而依然有问题困扰着用户：增加硬盘后盘符发生了怎样的变化，多块硬盘对多个操作系统的安装有什么影响？让我们以2块硬盘为例简单了解一下盘符问题。

假设我们有硬盘DISK1和硬盘DISK2。在我们购买它们时候，商家往往会为我们进行初步的分区及格式化。如图所示，它们分别有3个分区：一个主分区并已激活，两个逻辑分区。那么它们的盘符排列又是什么样的呢？（见表1）

DOS以及Windows 95/98完全按照这种方式记录盘符，不管操作系统是在添加硬盘以前安装的还是添加硬盘以后安装的，盘符都是这样进行排列。许多正在使用Windows 95/98系统的用户，在添加新的硬盘以后会发现以前使用正常的软件无法运行了，这就是典型的盘符错乱，是由于那个“第三者插足”的“D”盘造成的。而Windows 2000/XP略有不同，假使你已经安装了操作系统，再添加新的硬盘，系统给新硬盘提供的盘符是靠在最后的。如你已有“C：，D：，E：”，那么新硬盘的盘符则是“F：，G：，H”，这样就避免了盘符错乱。当然，光驱的盘符也会跟着向后靠，某些需要光盘的程序就需要重新安装或调整光驱盘符了。

对于空白或仅第一块硬盘安装有操作系统的两块硬盘来说，多个操作系统的安装并没有特别的影响，实际上对第二块硬盘的分区来说，它们和第一块硬盘的其他逻辑分区一样都被当作逻辑分区来使用。对于独立安装有操作系统的两块硬盘，我们还是应该注意以下一些问题：

1. BIOS的设置

   对于Windows 98及其以后的Windows版本如2000/XP，BIOS对硬盘的隐蔽功能已经没有效果了，即使将BIOS内的硬盘设置为NO INSTALLED，操作系统依然能够辨认出安装的硬盘，这样设置对Windows 98下的盘符有影响（见表2）。因为两块硬盘都安装有独立的操

作系统，在BIOS中改变两块硬盘启动顺序将进入各自的操作系统，这也给我们提供了另一种多系统共存的思路。但和上文讲到的多系统共存相比，需要不断地在BIOS内进行切换而且受限于BIOS功能。

2. 操作系统文件类型的选择

   NTFS分区在Windows 9X下不可见，而新的Windows 2000/XP都能识别，所以在安装过程中就要特别小心。Windows 2000/XP的安装有不少的提示，如安装路径的选择，原系统的备份（Windows 2000升级至Windows XP没有提供此功能），用户应该小心选择避免数据的丢失。升级安装系统应该考虑到原系统所支持的分区格式，即使安装程序给出升级的推荐，对于引导分区也不要轻易升级格式。

21.2.3 功能强大的第三方软件

目前，随着计算机存储技术的发展，硬盘的容量越来越大，新的电脑硬盘配置一般都超过了10GB，这样用户不仅可以在硬盘中存储更多的教学、游戏软件，还可以通过安装多个操作系统来使用更多的应用软件或更好地进行软件开发工作。

过去当我们想了解诸如Linux和BeOS这样的新操作系统时，通常需要把它们安装在不同的计算机上（或者使用多个硬盘），才能对它进行测试。由于各类操作系统的兼容性问题，我们不得不在测试完一个操作系统后去格式化硬盘甚至重新分区，以便安装其他的操作系统，这样做十分繁琐并且不利于学习和工作。由于大容量硬盘的采用，现在人们通常利用可以装载多个引导程序的工具软件，在同一台机器上安装并运行多个操作系统。

在这些系统引导工具软件中，System Commander，PowerQuest PartitionMagic中的引导管理以及Bootmanager Bootstar和XOSL是比较典型且常用的几个。作为系统共存测试的重要部分，我们对这几款软件进行了测试和比较。那么它们之间有什么不同，各自有什么特殊的功能或过人的能力？我们将在下文中一一讲解。

1. System Commander 7.02

   首先是System Commander，这大概也是最熟悉的多系统引导软件，目前已经发展到7.02版本。它以其优秀的功能、友好的界面和简便的操作，受到大量计算机用户的青睐。新版本的System Commander界面更加华丽，设置更为方便。总之，System Commander在完善内部的同时也没有忘记讨好用户，这是一款“秀外惠中”的软件。

   System Commander可将硬盘的主引导区记录(MBR- Master Boot Record) 换成自己的程序, 用户可以在一台电脑上安装各种操作系统，最高可支持上百个。它不仅支持所有使用FAT文件格式的操作系统, 如DOS、Windows 95/NT，还支持FAT32、OS/2的HPFS以及Windows NT的NTFS、Netware等所有IBM PC所支持的文件格式。对于使用多个硬盘的用户，System Commander可以方便地找出所有硬盘中的操作系统，甚至在用户替换硬盘后可以自动更新操作系统列表。

对于用户来说，System Commander最突出的特点就是引导界面的图形化和个性化，它甚至允许用户随意更改图标。看看我们使用的这些图标，是不是非常有趣？

关于System Commander的安装和使用，我们将在下文中做更加详细的介绍。

2. Bootmanager Bootstar

   Bootmanager Bootstar是一款硬盘分区、辅助安装多操作系统的工具，最新版本7.28。它可支持多达4个硬盘且每个硬盘可有4至15个分区，能将DOS、Windows各版本（95/98/Me/NT/2000）、Linux等系统并列安装到硬盘上，并在启动电脑时自由选择硬盘/磁盘方式启动。另外，程序还设计有独特的口令保护功能，可以让整个硬盘分区隐藏起来。

   Bootmanager Bootstar使用一种独特的方式配置启动选项，用户需要配置程序自己的Bootprofiles表。

   它是机器启动时由Bootmanager Bootstar先于系统启动显示给用户的操作表单，其中包括有哪些分区可见及由哪个分区作为根启动分区（磁盘）等信息，以便用户可以选择并实现其预先计划的效果。每一项内容的右击选单中都有设定项，包括激活命令、可见选项和口令设置等，用户可根据自己的需要进行安排。有些操作系统（如Windows NT/2000和Linux）需要分区入口的确切位置，这时我们就需要在“Positions In MBR”中设置相关选项。

3. XOSL 多操作系统引导管理工具XOSL（Extended Operating System Loader），是一个得到GNU General Public License（GPL）许可的完全免费的软件，最新版本为1.1.5。

   XOSL作为多系统引导工具软件的后起之秀，功能相当强大，支持的系统包括：BeOS、MS-DOS、FreeDOS、Linux（with Lilo）、Solaris、VxWorks 5.x、Windows 95/98/NT/2000等。它可以同时管理多达24个引导程序并对它们设置保护口令；如果我们不希望每次启动时都去选择引导程序，那么它也能自动引导到上次运行的操作系统。XOSL可以针对不同的系统或硬盘设置活动分区，也可以隐藏微软系统分区，甚至可以实现主、副硬盘的交换。

   XOSL可以将主引导程序记录设置在任何驱动器上，也可以在任何驱动器上引导DOS/Windows 9x系统。以往多系统引导软件的大忌就是共存于同一块硬盘，但是XOSL却可以与其它引导程序管理器共存而不会产生冲突。

   我们在使用中可以不用引导任何操作系统而直接进入XOSL的分区管理器。另外，XOSL提供对主引导记录的防病毒保护功能，这让我们的系统安全性有了明显的提高。

4. PowerQuest PartitionMagic中的引导工具

作为很常用的无损分区软件，PQMagic也提供了转换引导分区的功能，但是仅仅可以作为一种多系统共存的手段，如果作为系统引导的手段，则略显繁琐。要使用PQMagic制作多引导分区，我们必须在安装时选择自定义安装模式，并且再选择安装PartitionMagic For Dos/Windows 3.X。所谓PQMagic中的多系统引导，实际上仅仅是调用其中的“转换引导分区”功能。虽然它还无法和以上的软件相比，但是，对于已经使用了这款软件进行分区的用户来说，这种多系统引导方式却可以保证系统的安全性。

我们必须了解，某些引导软件之间存在着或多或少的兼容性问题。而对于接管了我们硬盘引导区的软件，它们的崩溃对于系统来说是致命的，一旦出现这种情况，很可能会造成用户不可弥补的损失。因此，这里强烈建议大家在使用时注意以下几点：

(a) 不要同时使用多款系统引导软件。在安装此类软件前，请首先确认已经卸载了其他的系统引导软件。

(b) 建议在安装此类软件时，按照其提示启动恢复盘。

(c) 建议在每次分区完成后，都进行硬盘分区表的备份工作。这样一旦分区或引导出现异常，我们可以使用分区表恢复功能来抢救系统，减少损失。

System Commander手把手

我们将以System Commander 7.02（以下简称SC）为例，一步一步教读者朋友如何安装多系统，在此之前我们先要作几点说明：

1. SC的操作系统支持列表：

   除了以上列表，SC号称支持所有基于PC的操作系统。

2. 现在一般能找到的SC 7.02都是Windows安装版，所以在安装前硬盘里至少要有一个Windows操作系统。

3. 如果是全新安装，建议用Fdisk进行分区，虽然SC自带的分区工具以及PQMagic之类第三方分区工具功能更强、速度更快，不过兼容性和稳定性不好，将来可能造成分区表被破坏，所以不建议使用。

4. 虽然SC可以识别任何它所支持的操作系统的分区格式，但SC自身只能被安装在FAT、FAT32或者NTFS分区之上。

5. 硬盘分区规划如下：

   以上规划只是为了单纯的多系统安装而定制的，读者可根据自己的情况来规划分区。但要注意：C区一定要分成“4”中要求的格式；硬盘中最多只能有4个主分区（由所有逻辑分区组成的扩展分区算一个主分区）；Fdisk只能划分一个主分区和一个扩展分区；如果没有本地安全性的要求，并且希望各个操作系统间可以共享数据，建议将Win2000及WinXP所在区分为FAT32格式，否则，应该分为NTFS格式。

6. 我们将遵循如下的安装顺序（顺序并不会影响最终的安装效果）：Windows 98、SC7.02、Windows Me、Windows 2000、Windows XP、Linux。

好了，已经罗嗦了半天了，让我们赶快开始吧！

21.3 系统破坏与修复

一般来说，按本文所介绍的步骤进行操作是不会有任何问题的。但实际上情况是很复杂的，会因为各种各样的原因引起硬盘分区表或引导区的破坏。下面我们就几种常见的情况说明一下修复的方法。

1. 非正常卸载SC造成的损坏。由于SC采用了特殊的技术独占主引导区，并且具有很高的优先级，所以当你不在需要SC时，必须按上文中所述的那样进行卸载，否则即使是格式化C区、重装系统甚至重新分区电脑都无法正常引导。如果由于你的疏忽已经造成了这样结果，也不必着急，有两种解决方法：用光盘或者软盘引导系统然后重装SC，再按要求卸载就可以了；如果你实在不想再用SC了，那么还有一个简单的办法，就是在DOS环境下运行命令“fdisk /mbr”，这个命令可以重写主引导记录，使其恢复正常。

2. 病毒破坏造成的损坏。这种情况一般用杀毒软件都可以解决，只要用杀毒软盘引导系统，再按杀毒软件的提示杀毒，并修复引导区或分区表即可，像现在国内比较流行的KV300、金山毒霸2002、瑞星2002都有修复硬盘的功能。

3. 多种第三方分区及磁盘扫描工具交叉使用造成的损坏，比如：SC的分区工具于PQmagic就有冲突存在，如果你一会儿用SC改变分区，一会儿又用PQmagic，就很容易造成分区表被破坏，虽然很多时候并不影响正常使用，但磁盘扫描工具总会报错又修不好，怎么办？也许你想起了大名鼎鼎的Norton Disk Doctor，用它来试试吧，结果它能发现错误并自动修复，一切看来都很顺利，可当你重启电脑后可怕的事情发生了，分区完全乱了套而且大量数据丢失，情况严重时就只能求助于数据恢复公司了，不过对于个人而言，修复的费用简直是天价，已经够买好几块硬盘了。由于这两类工具种类繁多，所以情况也很复杂，难以穷尽，仅以此实例说明其严重性。没有很好的解决办法，所以应尽量少用这类工具，如果要用就用一个，不要换着试。

其实对于以上三种情况，最好的办法还是防患于未然，在分区一切正常时就备份引导区及分区表，很多杀毒软件和磁盘分区工具都有此功能，下面以金山毒霸2002为例说明一下备份方法：在DOS环境下运行金山毒霸目录中的“KAVFIX.EXE”，出现如图所示界面，在“Tools”菜单栏中选择“Backup Boot Record”即可进行备份。备份的文件大小视硬盘大小而定，一般一张软盘就可以装下，备份文件最好存在软盘或第二块硬盘上，有条件的话刻在光盘上更好。如果就存在当前硬盘上，分区表被破坏时可能会造成该备份文件的丢失，这样备份就失去意义了。

21.4 系统启动盘的制作

系统启动盘也叫系统应急盘，它是一旦系统因感染病毒或相关文件损坏等原因而不能直接进入（启动）系统时所要用的软盘，这时如果没有这张小小的软盘，一切修复工作都无法进行，你也就只有望机兴叹、一筹莫展的份了。因此，平时准备一张系统启动盘非常有必要。

21.4.1 Windows 95/98/Me 系统启动盘的制作

Windows 95/98/Me这三个系统的启动盘制作起来比较简单，方法也大致相同，主要有以下3种：

1. 在操作系统安装过程的初期，系统将自动询问是否需要在此时创建启动盘，你只要选择需要在此时创建，插入软盘，按照屏幕提示操作就可以了。

2. 在运行系统后，打开“控制面板”，双击“添加/删除程序”，然后在出现的“添加/删除程序属性”的对话框里选择“启动盘”选项，然后插入一张空白的软盘，点击“创建启动盘”，系统就会自动制作启动盘了，这时制作出来的启动盘最完整。我们随后会详细介绍一下它的内容。

3. 在DOS下制作启动盘：

当Windows系统崩溃无法进入图形界面时，如果此时仍然能够进入DOS状态，那么就可以运行Windows\command目录下的Bootdisk.bat批处理文件，运行后按提示插入软盘后回车就行了。另外，也可以直接把Windows\Command\Ebd目录下的所有文件拷贝到软盘中，同样也是一张启动盘了。

这里我们以Windows 98启动盘为例，简要一下介绍Windows9x启动盘中的内容：

1. 通用光驱驱动程序（用此软盘启动后会出现含有三个项目的多重启动菜单，选第一项就可以加载通用光驱的驱动程序，这个驱动程序能支持大多数大多数的ATAPT、IDE和SCSI光驱）；

2. 虚拟磁盘的创建程序（用软盘启动时创建一个大小为2MB的虚拟磁盘，原来被压缩成CAB格式的诊断工具和虚拟光驱驱动程序就可以释放到虚拟磁盘上了）；

3. 其他一些重要的系统文件和工具：
   Autoexec.bat,系统启动时自动运行的批处理文件
   Command.com,内部命令解释程序
   Config.sys,用于载入设备驱动程序的配置文件
   Drvspace.bin,磁盘压缩驱动
   Ebd.cab,包含一些应用程序的压缩包
   Extract.exe,Cab格式压缩包的解压程序，直接执行Extract.exe可以查看它的具体命令行参数，这里用来在启动时释放出Ebd.cab中的程

序；也可以用它从Windows安装目录的压缩包中解出任何Windwows系统文件
Fdisk.exe,用于硬盘分区的命令文件
Himem.sys,管理扩展内存和高端内存
Io.sys,系统引导文件
Msdos.sys,启动选项文件(路径、多重启动等)
Oakcdrom.sys,通用的DOS光驱驱动程序（用法：在Config.sys中加入一行device=oakcdrom.sys /d:mscd001）
Ramdrive.sys,启动时建立虚拟盘
Ebd.cab中包含的部分文件：
Attrib.exe,设置文件属性
Chkdsk.exe,简单的磁盘检测工具
Debug.exe,Debug调试程序
Edit.com,在DOS下进行编辑的工具
EXT.exe,（Cab格式压缩包的解压程序，比直接使用Extract.exe命令简单很多）
Format.com,格式化命令
Mscdex.exe,DOS下的光驱启动文件（用法：在Autoexec.bat中加入mscdex.exe /d:mscd001）
Scandisk.exe,磁盘扫描程序
Sys.com,系统传送命令，可以将系统启动文件从软盘传输到硬盘，反之亦可，命令行为Sys A: C:（这里假设A:为启动软盘，C:为系统所在硬盘分区）

另外建议把Deltree.exe（删除目录的命令）、Discopy.com（磁盘复制的命令）、Mouse.com（DOS下的鼠标驱动）、Smartdrv.exe（设置磁盘高速缓冲区）和xcopy.exe/xcopy32.exe（高级文件拷贝程序）等几个常用的文件也拷贝到启动盘上，这样启动盘的内容就更加完善了。

21.4.2 Windows 2000系统启动盘的制作

把Windows 2000的光盘插入驱动器中，进入BootDisk目录，并运行其中的Makebt32（中文提示信息）或Makeboot（英文提示信息）命令，按屏幕提示操作即可制作出Windows 2000的启动盘。不过，采用这种常规方式制作Windows 2000启动盘一次就要4张软盘，这似乎过于繁琐了。其实如果只是为了启动系统，我们完全可以制作出一张简化版的Windows 2000启动盘：

1. 准备一张格式化过的软盘（格式化操作必须在Windows 2000下进行）。

2. 从Windows2000启动分区的根目录中将Ntldr、Ntdetect.com、Boot.ini、Io.sys和Bootsect.dos这几个文件拷贝到软盘中，这样一张简化版的Windows 2000启动盘就制作完成了。

21.4.3 Windows XP“启动盘”的制作

Windows XP虽然与Windows 2000同属于NT核心的系统，但Windows XP“启动盘”的制作则要简洁得多：放入一张空白软盘，在资源管理器中用右键点击软驱，在菜单中选择“格式化”，在弹出的对话框中勾选“制作MS-DOS启动盘”，然后点击“确定”即可。不过这张“启动盘”的内容也非常简单，仅仅包含基本的系统引导文件，只能启动到MS-DOS提示符状态，不能启动光驱，也无法访问NTFS文件系统。

21.4.4 Linux启动盘的制作

相对于Windows系统来说，Linux的启动盘制作起来稍微要复杂一些。Linux启动盘一般可分为如下两种：一种是只含内核（Kernel）的启动盘，通常也称为Boot盘；另一种除了内核以外，还包含了根文件系统（Root File System），有时也称作Boot/Root盘。后者可以说已经是一个完整的操作系统，由于它在制作过程中通常要对内核和根文件系统进行压缩，并且涉及到对内核进行重新编译的过程，这里我们就不作详细介绍了。

我们平常意义上的Linux启动盘一般指的是前者，这种盘只含有系统内核，并不包括根文件系统。它的内核中只指明了根文件系统所在的位置，在启动过程中需要根据该参数去安装根文件系统，也就是说它必须和硬盘上的根文件系统配合才能使用。

Linux发行的版本很多，这里我们以常见的RedHat Linux 7.2为例简要介绍一下Boot启动盘的制作方法：

1. 在安装RedHat Linux 7.2时，安装向导中有一步便是启动盘的制作，这时插入一张软盘按屏幕提示操作就可以了。

2. 在当前的Linux系统下制作一张启动软盘的方法是：

   (a) 以Root的身份进行登录；

   (b) 弄清当前Linux内核的版本，这里RedHat Linux 7.2的核心是2.4.7-10版，使用Mkbootdisk命令制作启动盘：mkbootdisk --device /dev/fd0 2.4.7-10（其中参数--device /dev/fd0表示软盘驱动器的设备号是/dev/fd0）。这样，系统就会将启动盘的镜像写入软盘。

21.5 结语

通过我们的介绍，相信大家对多操作系统共存已经有了一个较清晰的认识。多系统共存给用户带来的效益也正是我们所倡导的物尽所用各取所需。就多操作系统共存技术本身而言，要有针对不同用户群体而设计的操作系统的存在，它的话题就将继续下去，而这一时间将会相当漫长。

我们看到，借助强大而华丽的Windows XP系统，最近微软终于完成了Windows操作系统的统一，不过我们知道，一款操作系统的成功，并不

是仅仅依靠技术或市场的优势，最近北京市政府的软件采购就说明了这一点。我们相信，真正一统天下的操作系统现阶段不会也不可能出现。

虽然Windows系列几乎完全占领了PC的操作系统市场，但是，毕竟还存在着微软没有涉足的领域，还有微软的大旗没有插上的山丘，Linux、OS/2等软件还在苦苦地支撑等待着自己曙光的出现，操作系统市场的争斗仍然在继续，以后的日子，我们会一如既往地提供给大家最新的相关资讯。

22 使用find命令来查找文件

每一种操作系统都是由成千上万个不同种类的文件所组成的。其中有系统本身自带的文件，用户自己的文件，还有共享文件等等。我们有时候经常忘记某份文件放在硬盘中的哪个地方。在微软的WINDOWS操作系统中要查找一份文件是相当简单的事情，只要在桌面上点击“开始”－“搜索”中就能按照各种方式在本地硬盘上，局域网络，甚至在INTERNET上查找各种文件，文档。

可是使用Linux的用户就没有那么幸运了，在Linux上查找某个文件确实是一件比较麻烦的事情。毕竟在Linux中需要我们使用专用的“查找”命令来寻找在硬盘上的文件。Linux下的文件表达格式非常复杂，不象WINDOWS,DOS下都是统一的AAAAAAA.BBB格式那么方便查找，在WINDOWS中，只要知道要查找的文件的文件名或者后缀就非常容易查找到。Linux中查找文件的命令通常为“find”命令，“find”命令能帮助我们在使用,管理Linux的日常事务中方便的查找出我们需要的文件。对于Linux新手来说，“find”命令也是了解和学习Linux文件特点的方法。因为Linux发行版本繁多，版本升级很快，在Linux书籍上往往写明某个配置文件的所在位置，往往Linux新手按图索骥还是不能找到。比如说REDHAT Linux 7.O和REDHAT Linux 7.1中有些重要的配置文件所在的硬盘位置和文件目录就有了很大的改变，如果不学会使用“find”命令，那么在成千上万的Linux文件中要找到其中的一个配置文件是相当困难的，笔者在没有精通“find”命令之前就吃过这样的苦头。好，下面就详细为大家介绍强大的“find”命令的全部使用方法和用途。

22.1 通过文件名查找法：

这个方法说起来就和在WINDOWS下查找文件一样容易理解了。如果你把这个文件放在单个的文件夹里面，只要使用常见的“ls”命令就能方便的查找出来，那么使用“find”命令来查找它就不能给你留下深刻的印象，毕竟“find”命令的强大功能不止这个。如果知道了某个文件的文件名，而不知道这个文件放到哪个文件夹，甚至是层层套嵌的文件夹里。举例说明，假设你忘记了httpd.conf这个文件在系统的哪个目录下，甚至在系统的某个地方也不知道，则这是可以使用如下命令：

```
find / -name httpd.conf
```

这个命令语法看起来很容易就明白了，就是直接在find后面写上-name，表

明要求系统按照文件名查找，最后写上httpd.conf这个目标文件名即可。稍等一会系统会在计算机屏幕上显示出查找结果列表：

```
/etc/httpd/conf/httpd.conf
```

这就是httpd.conf这个文件在Linux系统中的完整路径。查找成功。

如果输入以上查找命令后系统并没有显示出结果，那么不要以为系统没有执行find/ -name httpd.conf命令，而可能是你的系统中没有安装Apache服务器，这时只要你安装了Apache Web服务器，然后再使用find / -name httpd.conf就能找到这个配置文件了。

22.2 无错误查找技巧：

在Linux系统中“find”命令是大多数系统用户都可以使用的命令，并不是ROOT系统管理员的专利。但是普通用户使用“find”命令时也有可能遇到这样的问题，那就是Linux系统中系统管理员ROOT可以把某些文件目录设置成禁止访问模式。这样普通用户就没有权限用“find”命令来查询这些目录或者文件。当普通用户使用“find”命令来查询这些文件目录是，往往会出现”Permissiondenied.”（禁止访问）字样。系统将无法查询到你想要的文件。为了避免这样的错误，我们可是使用转移错误提示的方法尝试着查找文件，输入

```
find / -name access_log 2>/dev/null
```

这个方法是把查找错误提示转移到特定的目录中去。系统执行这个命令后，遇到错误的信息就直接输送到stderrstream 2 中，access_log 2就是表明系统将把错误信息输送到stderrstream 2中，/dev/null是一个特殊的文件，表明空的或者错误的信息，这样查询到的错误信息将被转移了，不会再显示了。

在Linux系统查找文件也会遇到这样一个实际问题。如果我们在整个硬盘，这个系统中查找某个文件就要花费相当长的一段时间，特别是大型Linux系统和容量较大的硬盘，文件放在套嵌很深的目录中的时候。如果我们知道了这个文件存放在某个大的目录中，那么只要在这个目录中往下找就能节省很多时间了。使用find /etc -name httpd.conf 就可以解决这个问题。上面的命令就是表示在etc目录中查询httpd.conf这个文件。这里再说明一下“/ ”这个函数符号的含义，如果输入“find/ ”就是表示要求Linux系统在整个ROOT目录下查找文件，也就是在整个硬盘上查找文件，而“find/etc”就是只在etc目录下查找文件。因为“find/etc”表示只在etc目录下查找文件，所以查找的速度就相应要快很多了。

22.3 根据部分文件名查找方法：

这个方法和在WINDOWS中查找已知的文件名方法是一样的。不过在Linux中根据部分文件名查找文件的方法要比在WINDOWS中的同类查找方法要强大得多。例如我们知道某个文件包含有srm这3个字母，那么要找到系统中所有包含有这3个字母的文件是可以实现的，输入：

```
find /etc -name '*srm*'
```

这个命令表明了Linux系统将在/etc整个目录中查找所有的包含有srm这3个字母的文件，比如absrmyz，tibc.srm等等符合条件的文件都能显示出来。如果你还知道这个文件是由srm 这3个字母打头的，那么我们还可以省略最前面的星号，命令如下：

```
find/etc -name 'srm*'
```

这是只有像srmyz 这样的文件才被查找出来，象absrmyz或者absrm这样的文件都不符合要求，不被显示，这样查找文件的效率和可靠性就大大增强了。

22.4 根据文件的特征查询方法：

如果只知道某个文件的大小，修改日期等特征也可以使用“find”命令查找出来，这和WINDOWS系统中的”搜索”功能是基本相同的。在微软的”搜索”中WINDOWS中的”搜索助理”使得搜索文件和文件夹、打印机、用户以及网络中的其他计算机更加容易。它甚至使在Internet 上搜索更加容易。”搜索助理”还包括一个索引服务，该服务维护了计算机中所有文件的索引，使得搜索速度更快。使用”搜索助理”时，用户可以指定多个搜索标准。例如，用户可以按名称、类型及大小搜索文件和文件夹。用户甚至可以搜索包含特定文本的文件。如果用户正使用Active Directory，这时还可以搜索带有特定名称或位置的打印机。

例如我们知道一个Linux文件大小为1,500 bytes，那么我们可是使用如下命令来查询

```
find / -size 1500c
```

字符c 表明这个要查找的文件的大小是以bytes为单位。如果我们连这个文件的具体大小都不知道，那么在Linux中还可以进行模糊查找方式来解决。例如我们输入find/ -size +10000000c 这个命令，则标明我们指定系统在根目录中查找出大于10000000字节的文件并显示出来。命令中的“＋”是表示要求系统只列出大于指定大小的文件，而使用“-”则表示要求系统列出小于指定大小的文件。下面的列表就是在Linux使用不同“find”命令后系统所要作出的查找动作，从中我们很容易看出在Linux中使用“find”命令的方式是很多的，“find”命令查找文件只要灵活应用，丝毫不必在WINDOWS中查找能力差。

```
find / -amin -10 # 查找在系统中最后10分钟访问的文件
find / -atime -2 # 查找在系统中最后48小时访问的文件
find / -empty # 查找在系统中为空的文件或者文件夹
find / -group cat # 查找在系统中属于 groupcat的文件
find / -mmin -5 # 查找在系统中最后5分钟里修改过的文件
find / -mtime -1 #查找在系统中最后24小时里修改过的文件
find / -nouser #查找在系统中属于作废用户的文件
find / -user fred #查找在系统中属于FRED这个用户的文件
```

下面的列表就是对find命令所可以指定文件的特征进行查找的部分条件。在这里并没有列举所有的查找条件，参考有关Linux有关书籍可以知道所有find命令的查找函数。

-amin n	查找系统中最后N分钟访问的文件
-atime n	查找系统中最后n*24小时访问的文件
-cmin n	查找系统中最后N分钟被改变状态的文件
-ctime n	查找系统中最后n*24小时被改变状态的文件
-empty	查找系统中空白的文件，或空白的文件目录，或目录中没有子目录的文件夹
-false	查找系统中总是错误的文件
-fstype type	查找系统中存在于指定文件系统的文件，例如：ext2
-gid n	查找系统中文件数字组ID 为n的文件
-group gname	查找系统中文件属于gnam文件组，并且指定组和ID的文件

Find命令的控制选项说明：

Find命令也提供给用户一些特有的选项来控制查找操作。下表就是我们总结出的最基本，最常用的find命令的控制选项及其用法。

选项	用途描述
-daystart	测试系统从今天开始24小时以内的文件，用法类似-amin
-depth	使用深度级别的查找过程方式,在某层指定目录中优先查找文件内容
-follow	遵循通配符链接方式查找;另外，也可忽略通配符链接方式查询
-help	显示命令摘要
-maxdepth levels	在某个层次的目录中按照递减方法查找
-mount	不在文件系统目录中查找，用法类似-xdev
-noleaf	禁止在非UNUX文件系统，MS-DOS系统，CD-ROM文件系统中进行最优化查找
-version	打印版本数字

使用-follow选项后，find命令则遵循通配符链接方式进行查找，除非你指定这个选项，否则一般情况下find命令将忽略通配符链接方式进行文件查找。

-maxdepth选项的作用就是限制find命令在目录中按照递减方式查找文件的时候搜索文件超过某个级别或者搜索过多的目录，这样导致查找速度变慢，查找花费的时间过多。例如，我们要在当前(.)目录技巧子目录中查找一个名叫fred的文件，我们可以使用如下命令

```
find . -maxdepth 2 -name fred
```

假如这个fred文件在./sub1/fred目录中，那么这个命令就会直接定位这个文件，查找很容易成功。假如，这个文件在./sub1/sub2/fred目录中，那么这

个命令就无法查找到。因为前面已经给find命令在目录中最大的查询目录级别为2，只能查找2层目录下的文件。这样做的目的就是为了让find命令更加精确的定位文件，如果你已经知道了某个文件大概所在的文件目录级数，那么加入-maxdepth n 就很快的能在指定目录中查找成功。

22.5 使用混合查找方式查找文件

find命令可以使用混合查找的方法，例如我们想在/tmp目录中查找大于100000000字节并且在48小时内修改的某个文件，我们可以使用-and 来把两个查找选项链接起来组合成一个混合的查找方式。

```
find /tmp -size +10000000c -and -mtime +2
```

学习过计算机语言的朋友都知道，在计算机语言里，使用and ,or 分别表示“与”和“或”的关系。在Linux系统的查找命令中一样通用。还有这样的例子，

```
find / -user fred -or -user george
```

我们可以解释为在/tmp目录中查找属于fred或者george这两个用户的文件。

在find命令中还可以使用“非”的关系来查找文件，如果我们要在/tmp目录中查找所有不属于panda的文件，使用一个简单的

```
find /tmp ! -user panda
```

命令就可以解决了。很简单。

22.6 查找并显示文件的方法

查找到某个文件是我们的目的，我们更想知道查找到的文件的详细信息和属性，如果我们采取现查找文件，在使用LS命令来查看文件信息是相当繁琐的，现在我们也可以把这两个命令结合起来使用。

```
find / -name "httpd.conf" -ls
```

系统查找到httpd.conf文件后立即在屏幕上显示httpd.conf文件信息。

```
12063 34 -rw-r--r-- 1 root root 33545 Dec 30 15:36 /etc/ht-
tpd/conf/httpd.conf
```

下面的表格就是一些常用的查找文件并显示文件信息的参数和使用方法

选项	用途描述
-exec command;	查找并执行命令
-fprint file	打印文件完整文件名
-fprint0 file	打印文件完整文件名包括空的文件
-fprintf file format	打印文件格式
-ok command;	给用户命令执行操作，根据用户的Y 确认输入执行
-printf format	打印文件格式
-ls	打印同种文件格式的文件

22.7 对find命令的补充

文件查找不仅限于使用find—尽管它很强大.文件查找的前提要查找者只知道关于文件的某方面的信息, 希望根据这些信息定位文件的位置:

1. 已知程序(全)名比如知道有个程序叫ifconfig, 也可以运行它, 但不知道它到底身居何处:

   which ifconfig

2. 只知道文件部分的名字, 或怀疑同名的程序有多个COPY:

   locate ifconfig

3. 已知文件位于当前目录下, 只知道文件名中含有某些字符:

   echo *abc*

4. 已知文件位于某个rpm包中, 想知道其位置

   rpm -ql MySQL

5. 已知文件含有某些字符串, 且位于某目录之下

   grep -lr text *

6. 作为root想知道某某用户最近访问过的文件
   查看其~/.bash_history文件, 或其它SHELL中存放命令历史列表的文件

7. 已知某文件被运行中的程序所使用

   lsof

23 LINUX经典问答

1. Q:在Linux 中怎样实现类似DOS 环境下的AUTOEXEC.BAT 批处理功能?

   A:将脚本添加到/etc/rc.d/rc.local 中即可。这是由Linux启动时的第一个进程init控制的。init进程还可以自动启动/etc/rc、/etc/rc.d、/etc/rc?.d目录下的许多脚本文件。

2. Q:Linux 中NE2000 兼容网卡的安装?

   A:目前市场上NE2000 兼容网卡比较多,如D-Link DE220p 等。如果Linux 不能检测到这些网卡，则可以用手动的方法安装。以D-Link DE220p 网卡为例，在Red hat Linux6.0 和Turbo Linux 下可以这样安装：首先检测出网卡的IO 端口号和irp 号，然后启动进入Linux，在/etc/rc.d/rc.sysinit 文件中加入语句(假设网卡的IO=0x240,irp=0x5)：

modprobe ne.o io=0x240 irq=5

然后启动Linux 即可。

对于Slackware Linux,可以编辑/etc/rc.d/rc.modules 文件，将NE2000网卡驱动前的注释符号“ ”去掉，设置网卡IO 号即可,即:

/sbin/modprobe ne io=0x240

3. Q:Slackware Linux3.4 的LILO 被破坏后的恢复?

   A:一台同时安装Slackware Linux3.4、Windows98、WindowsNT 的机器，在Windows98 重新安装后，主引导区的LILO 丢失，不能引导Linux，按如下步骤处理后，就可以修复LILO：首先用BOOT 盘引导机器，在BOOT: 提示符后输入mount root=/dev/hda4 回车，其中hda4 是Linux 的引导分区，应当根据你的具体情况而定；这样机器就会启动进入到Linux 状态，然后以root 用户登录，进入/sbin 目录，运行liloconfig 程序，选第6 项“Reinstall LILO using the existing lilo.conf”，这样就会恢复原来的LILO 设置。

4. Q:怎样将LILO 备份到软盘上?

   A:在/sbin 目录下运行命令lilo -b /dev/fd0 即可。备份的软盘可以用来启动Linux 系统，与硬盘启动没有区别。

5. Q:能不能从DOS 下启动Linux?

   A:能。例如Red Hat Linux6.0 的光盘上有一个程序loadlin，它就可以从DOS 下启动Linux。条件是你要有一个Linux 内核映像文件vmlinuz(可在Linux 安装光盘上找)，还要知道Linux 的启动分区。命令的执行格式为：

   loadlin vmlinuz root=/dev/hda4

   其中/dev/hda4 为Linux 的root 文件系统所在的硬盘分区

   可以用这种方法恢复主引导分区中遭到破坏的LILO 系统。

6. Q:Linux 能否实现用户登录运行脚本?

   A:可以。用户登录时，Bash 首先执行全局登录脚本(由root 建立)/etc/profile，然后在用户起始目录下依次寻找.bash_profile、.bash_login、.profile 三个文件，执行最先找到的一个。可以用这种办法像Netware 一样为不同的用户定制运行环境。此外，用户退出登录时还可以运行.bash_logout 脚本。

7. Q:在Linux 中给命令指定别名(alias)：

   A:如果命令或命令序列太长，或不符合用户的习惯，那么为它指定一个别名是不错的办法。虽然可以为命令建立“链接”解决长文件名的问题，但对于带命令行参数的命令，链接就无能为力了。而指定别名则可以解决此类所有问题。只要举一些例子就可以了：

alias l='ls -l' ;用l 代替ls -l 命令(Xenix 下就有类似的l 命令) alias cd..='cd ..' ;用cd.. 代替cd .. 命令(对在DOS 下使用惯了cd.. 的人帮助很大) alias md='mkdir' ;用md 代替mkdir 命令(对在DOS 下…) alias c:='mount /dev/hda1 /mnt/c cd /mnt/c' ;用c: 命令代替命令序列：安装DOS分区，再进入

8. Q:一台机器为Win98 与Linux 双启动，默认启动OS 为Linux,现想改变默认启动为Win98，怎么办?

   A:简单。编辑lilo.conf 文件，在其文件的第四行后加上default=dos，存盘退出即可。

9. Q:系统的关闭：

   A:#shutdown －r +10 (十分钟后系统自动重启)
   # shutdown －r 13:00 (13:00整,系统自动重启)
   # shutdown －r now (系统立即重启)
   # reboot (系统立即重启)
   # haltsys (关闭系统)
   # shutdown －h (关闭系统)

10. Q:使用虚拟控制台：

    A:一般新安装的Linux 有7个虚拟控制台,热键为: CTRL+ALT+F1，CTRL+ALT+F2，ALT+F3，...例如:登录后按CTRL+Alt+F2 键,这时又可以看到login: 提示符, 这个就是第二个虚拟控制台。

11. Q:强行退出X-windows

    A:当工作在X-windows 界面时，想迅速切换到字符界面,只需使用热键Ctrl+Alt+Backspace。

12. Q:好的口令应满足什么条件?

    A:通常，好的口令应当易于记忆但却不易被猜中，它们应当：

    - 是大小写的混合;
    - 可拼读;
    - 既含字母，也含非字母的字符;
    - 6 到8 个字符长度;
    - 易于键入。

13. Q:ping 命令能干些什么?

    A:它能告诉你现在哪些机器可用。ping 可能是最基本的UNIX 网络命令，它仅仅向你提供的地址发送一个小包，然后侦听这台机器是否有“回答”。你可使用机器的Internet 地址，如192.78.222.81，或者也可使用机器名：ping therehost。该名字可以是局部的主机(host)文件中的名字或由域名服务器(DNS)解析的名字，但在任何情况下都应是网络可以用来找出特定机器的名称。如果机器对ping 没有响应，它也不会对其他什么有反应，因为在网络上“看”不见它。在这种情况下，除非问题非常简单和明显，诸如机器被别人关掉了等，否则，去问问系统管理员。不管在哪种情况下都要涉及到系统管理员，因为重启UNIX 系统并不像DOS 机器一样简单。

14. Q:Linux 下软驱的使用:

    A:如果是Linux 的ext2 文件系统，用如下命令:

    ```
    #mount -t ext2 /dev/fd0 /mnt
    ```

    DOS 格式的软盘则用命令:

    ```
    #mount -t msdos /dev/fd0 /mnt
    ```

    然后就可以在/mnt 里访问软盘的内容了。注意在取出软盘之前要先卸掉软盘上的文件系统：

    ```
    #umount /mnt
    ```

    否则会导致信息丢失。
    在软盘上建立文件系统可用如下命令:

    ```
    #mke2fs /dev/fd0 1440
    ```

15. Q:我的linux 为什么普通用户不能登陆?

    A:看看是不是有/etc/nologin 文件，删掉它再试试(另：root 通常是不能telnet 登录的，这很正常)。产生该问题的原因：系统在shutdown 时会产生这个文件，如果shutdown 过程意外终止,这个文件就没有被正常删除，这时候普通用户不能登录了。另外系统管理员在维护系统的时候也可以生成这个文件来阻止用户登录。

16. Q:如何设定每个帐号同一时间内允许的连接数?

    A:在RedHat 中的设置：首先在/etc/pam.d/login 中加上

    ```
    session required /lib/security/pam_limits.so
    ```

    然后在/etc/security/limits.conf 加入要限制的用户名或用户组，例如：

```
@student hard nproc 20
edward hard maxlogins 2
```

无须重新启动机器，设置完毕即可使用。

17. Q:怎么样做到限时登录Linux?

    A:一个简单的限时登录方法:写三个shell 程序，调用at 和系统维护功能:

    (a) 在指定的时间执行该shell，在/etc下生成一名为nologin的文件，如：

    ```
    vi /sbin/login.denied
    echo " Login Denied " > /etc/nologin
    chmod 700 login.denied
    ```

    (b) 在指定的时间执行该shell，删除/etc/下的nologin文件，如：

    ```
    vi /sbin/login.allowed
    if [ -f /etc/nologin ]; then
    rm /etc/nologin
    fi
    chmod 700 login.allowed
    ```

    (c) 编写一个限制时间的shell，如：

    ```
    vi /sbin/security
    if [ -f /sbin/login.denied ]; then
    at -f /sbin/login.denid 22:00
    fi
    if [ -f /sbin/login.allowed ]; then
    at -f /sbin/login.allowed 8:00
    if
    ```

    此种设置的功能是：从晚上10:00 到第二天早上8:00 禁止非root用户登录，显示为系统维护状态。

    另外，还需对root 用户的登录终端进行限制，最好设置在console，在RedHat 5.0 下在/etc/security/access.conf中配置

    ```
    -:root:ALL EXCEPT console
    ```

    就可以了。

18. Q:限制root 登录的终端:

    A:如果只允许root 在tty1 终端登录，则修改/etc/securetty 文件，将其它的终端注释掉，只留tty1:

```
tty1
# tty2
# tty3
# tty4
# tty5
# tty6
```

这样/bin/login 程序读取/etc/securetty 文件时就会知道root 只允许在tty1 登录。这样可以减小黑客以root 登录的可能性。

19. Q:阻止Linux 对ping 的反应:

    A:在/etc/rc.d/rc.local 文件中增加如下一行:

    ```
    echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_all
    ```

20. Q:防止IP 欺骗攻击:

    A:编辑host.conf 文件并增加如下几行:

    ```
    order bind,hosts
    multi off
    nospoof on
    ```

21. Q:删除硬盘主引导区的LILO：

    A:用Fdisk 程序，`C:> Fdisk /mbr`。然后用分区大师软件或Fdisk 程序可以删除Linux 分区。

22. Q:Linux FTP 服务器中将用户限制在自己目录下的方法：

    A: 第一步：创建一个ftp guest 组，用groupadd 命令，也可以用ftp 服务器创建的ftp 组(less /etc/group 看看是否存在，如果装了ftp server的话都有这个组存在)；

    第二步：在ftp 组中添加ftp 用户；

    第三步：修改/etc/ftpaccess 文件，加入guestgroup 的定义：guestgroup ftp; 或者用guestuser 直接指定ftp 用户名，如：guestuser ftpuser；

    第四步：向这个用户的$HOME 目录下拷贝必要的文件，主要是与ls 有关的。一般的ls 要Lib 支持，你得重新编译ls，或把lib 目录copy 过去，或者拷贝ftp server 带的ls, 比如把/home/ftp/ 下的bin，etc，lib 三个目录拷贝到这个用户的根目录下, 并修改权限(chown username.ftp *)。

    注意：现在Linux 下带的ftp 无须拷贝这些文件，至少Redhat6.1 是不需要拷贝的，所以就可以省去第四步。

24 XF86Conifg文件详解

24.1 摘要

X-Window系统给我们提供了许多配置工具，但最终也是生成了一个配置文件：XF86Config，它存放在/etc/X11目录下。如果你能够直接读懂它，那么对其做一些小的调整，那将是十分容易的事。本文将分析一下这个配置文件。

它分成了许多个小节，每个小节分别配置一部分。

24.2 File配置小节

该小节用来设置X Window系统所用的字体路径，下面是一个配置实例：

```
Section "Files"
RgbPath "/usr/X11R6/lib/X11/rgb"
FontPath "/usr/X11R6/lib/X11/fonts/TrueType"
FontPath "unix/:-1"
EndSection
```

其中Section “Files”表示Files小节开始，EndSection表示这个小节结束。其间有几个配置项：

- RgbPath：设置RGB色彩数据库的路径。这个配置项的内容是在安装的时候就写好的，肯定没错，别改它。
- FontPath：用来设置字体的路径。而象Redhat Linux会采用X Font Server（字体服务器）来管理所有的字体，这时，你就会看到字体路径就象：”unix/:-1”

24.3 Server Flag配置小节

该小节用来设置X Server的各种选项，包括一些功能键的设置，下面是一个配置实例：

```
Section "ServerFlags"
# NoTrapSignals
# DontZap
# DontZoom
# DisableVidModeExtension
# AllowNonLocalXvidtune
# DisableModInDev
# AllowNonLocalModInDev
EndSection
```

Section ”ServerFlags”表示Server Flags小节的开始，EndSection表示该小节的结束，其中有以下几个配置项：

- NoTrapSignals：不捕捉信号，在错误发生时直接dump产生Core文件。若启用这项设置将可能导致控制台不稳定，不过对于高手而言，dump产生的core文件有利于纠错。建议不启用。
- DontZap：取消使用Ctrl+Alt+Backspace组合热键退出X Window系统的功能。默认是注释掉它，也就是说可以使用Ctrl+Alt+Backspace组合热键退出X Window系统。建议不启用。
- DontZoom：取消使用Ctrl+Alt++和Ctrl+Alt+-组合键切换显示模式的功能。默认是注释掉它的，也就是说可以使用这两个组合键进行显示模式的切换。建议不启用。
- DisableVidModeExtension：禁示使用xvidtune程序调整画面。
- AllowNonLoaclXvidtune：允许使用非本地端的xvidtune程序
- DisableModInDev：关闭动态变更输入设备的设置
- AllowNonLocalModInDev：允许非本地端变更键盘和鼠标的设置

注：xvidtune程序是X windows中提供的一个应用程序，它用来调整画面显示大小和位置的程序。

24.4 键盘配置小节

该小节用来设置各式键盘，在此可以指定键盘的传输协议、语系、信号、字符对照表，下面是一个配置实例：

```
Section "Keyboard"
Protocol "Standard"
AutoRepeat 500 5

# ServerNumLock

LeftAlt Meta
RightAlt Meta
ScrollLock Compose
RightCtl Control

# XkbDisable
# XkbModel "pc102"
# XkbModel "microsoft"
#
#
# XkbOptions "ctrl:swapcaps"
XkbRules "xfree86"
XkbModel "pc101"
```

```
XkbLayout "en_US"
EndSection
```

Section "Keyboard"表示Keyboard小节的开始，EndSection表示该小节的结束，其中有以下几个配置项：

- Protocol：用来配置键盘所使用的传输协议。可选值为：
  - Standard：标准传输协议
  - Xqueue：使用X队列传输协议

  默认值就是Standard，一般无需修改，大家的大都是标准的键盘。
- AutoRepeat：用来配置对按住某键不放的处理，格式为：

  AutoRepeat 毫秒数次数

  如：AutoRepeat 500 5，表示当按住某键500毫秒（0.5秒）后，开始自动送出该按键信号，每秒5次。
- ServerNumLock：让X server处理NumLock信号，效果等于关闭键盘右方数字键的功能。
- 以下一组配置项是用来设置键盘上Alt、Ctrl、Shift、ScrollLock等键的功能定义：

  LeftAlt Meta
  RightAlt Meta
  ScrollLock Compose
  RightCtl Control

  左边是键名，LeftAlt就是左边的ALT键，RightAlt就是右边的ALT键…，而右边则是功能定义，包括：
  - Compose：等于一般Scroll Lock按键的功能；
  - Control：等于一般Ctrl按键的功能；
  - Meta：等于一般Alt按键的功能；
  - ModeLock：等于一般Caps Lock按键的功能
  - ModeShift：等于一般Shift按键的功能。

  接下来则是关于XKB的配置：

- XkbDisable：关闭键盘扩展属性，也即不指定其类型及对应的语言
- XkbCompat：设置键盘兼容性，缺省值是“default”，包含有“Basic”的设置，按串口的“mousekeys”、“accessx”、“misc”、“iso9995”与“japan”，相关文件配置放在/usr/X11R6/lib/X11/xkb/compat目录下。
- XkbGeometry：指定键盘结构，不同的键盘有不同的硬件结构，除非是使用Amiga、Atari、Macintosh等专用键盘，否则应使用缺省值“pc”，不设置也行。它的相关选项在/usr/X11R6/lib/X11/xkb/geometry目录下。
- XkbKeycodes：指定键盘按键送出的信息，缺省值为“xfree86”，其余的可用设置位于/usr/X11R6/lib/X11/xkb/keycodes目录下。
- XkbKeymap：指定键盘配置类型设置文件，可用的文件位于/usr/X11R6/lib/X11/xkb目录里的keymap子目录。如果设置了本选项相当于同时设置了XkbKeycodes、XkbTypes、XkbCompact、XkbSymbols以及XkbGeometry。
- XkbLayout：设置键盘输出语系，使用缺省值“us”即采用美式英文。可选值为：

  de：德文fr：法文it：意大利文jp：日文ru：俄文

- XkbModel：设置键盘的型号，常见的是“pc101”、“pc102”。若采用具有微软标志的键盘的话，请改为“pc104”。
- XkbOptions：设置键盘选项，通常并不需要任何选项设置，除非您想要交换按键的定义。如“ctrl:swapcaps”就可以将Ctrl与Caps Lock键互换。
- XkbRules：指定X window采用的键盘规范文件，除非有特殊需要（使用SGI的键盘），否则应使用缺省值：“xfree86”，其它的可选值在/usr/X11R6/lib/X11/xkb/rules目录下。
- XkbSymbols：定义键盘各按键所对应的字码，各国键盘革些字码不尽相同。缺省使用“us”，即ASCII，字码表对照文件放在/usr/X11R6/lib/X11/xkb/symbol目录下，用户可以修改。
- XkbTypes：设置键盘的种类，缺省值为“default”，在/usr/X11R6/lib/X11/xkb/types目录下给出了所有的选项。
- XkbVariant：设置键盘变量，通常无须设置。

24.5 鼠标配置小节

该小节用来设置诸如鼠标、触摸屏等标准输入设备。下面是一个配置实例：

```
Section "Pointer"
Protocol "PS/2"
Device "/dev/mouse"

# Protocol "Xqueue"

# BaudRate 9600
# SampleRate 150

Emulate3Buttons
Emulate3Timeout 50

# ChordMiddle

EndSection
```

与前面一样，Section "Pointer"代表该小节的开始，EndSection代表该小节结束。在该配置小节中，共有以下配置项：

- Protocol：设置鼠标使用的传输协议，可使用的协议有：
  - Auto：让X window自己检测，但可能不准确
  - BusMouse：总线型鼠标，早期的串口鼠标
  - GlidePoint：使用ALPS串口版本的GlidePoint触摸屏及其兼容产品
  - GlidePointPS/2：使用ALPS PS2版本的GlidePoint触摸屏及其兼容产品
  - IMPS/2：使用微软PS/2版本的IntelliMouse及其兼容产品
  - IntelliMouse：使用微软串口版本的IntelliMouse及其兼容产品
  - Logitech：使用旧版Logitech串口鼠标，新版的已改为“Microsoft”协议
  - Microsoft：在1992年起的5-8年间使用的串口鼠标
  - MMHitTab：使用旧版Logitech串口的HitTablet手机板
  - MMSeries：使用旧版Logitech串口的MouseMan鼠标

  - MouseMan：使用中期的Logitech串口的MouseMan鼠标
  - MouseManPlusPS/2：使用LogitechPS/2版本的天貂及其兼容产品
  - MouseSystem：采用MouseSystem传输协议的串口鼠标可以使用该选项
  - NetMosuePS/2：使用Genius PS/2版本的NetMouse网络鼠标及其兼容产品
  - NetScrollPS/2：使用Genius PS/2版本的NetScroll滚轮鼠标及其兼容产品
  - OSMouse：使用由操作系统控制的鼠标，而非让X Window控制鼠标
  - PS/2：现在最广泛使用的PS/2接口的鼠标
  - SysMouse：让FreeBSD能够使用外围设备代号为/dev/sysmouse的鼠标设备
  - ThinkingMouse：使用Kensington串口版本的Thinking鼠标ThinkingMousePS/2：使用PS/2版本的Thinking鼠标及其兼容产品
  - Xqueue：假如你在键盘设备段让键盘使用X队列传输协议，则这里也要

- Device：用来设置连接鼠标的外围设备代码，通常都是/dev/mouse
- BaudRate：用来设置波特率，只对某些Logitech鼠标有效，当使用AceCad绘图板时，需要将其设置为9600
- SampleRate：用来设置采样率，只对某些Logitech鼠标有效
- Emulate3Buttons：将双键鼠标模拟成为三键鼠标
- Emulate3Timeout：设置模拟三键超时时间
- ChordMiddle：如果是Logitech鼠标，那么得用这一配置项代替Emulate3Buttons

24.6 显示器配置小节

在该配置小节中可以设置显示器的水平、垂直扫描频率，同时定义各个显示模式与扫描频率之间的对应关系。X Server会根据显示器扫描频率的设置，配置用户指定的显示模式，自动在数十种的对应关系中，找到最恰当的显示刷新率。下面是一个配置实例：

```
Section "Monitor"
Identifier "Generic Monitor"
VendorName "Unknown"
ModelName "Unknown"
HorizSync 31.5
VertRefresh 60
ModeLine "640x480" 25.175 640 664 760 800
480 491 493 525
EndSection

Section "Monitor"
Identifier "cpq1355"
VendorName "Unknown"
ModelName "Unknown"
HorizSync 30 - 60

VertRefresh 50 - 125

# Mode "1024x768i"
# DotClock 45
# HTimings 1024 1048 1208 1264
# VTimings 768 776 784 817
# Flags "Interlace"
# EndMode
# --- 640x480 ---
# 640x480 @ 60 Hz, 31.5 kHz hsync
Modeline "640x480" 25.175 640 664 760 800
480 491 493 525
# 640x480 @ 72 Hz, 36.5 kHz hsync
Modeline "640x480" 31.5 640 680 720 864
480 488 491 521
# 640x480 @ 75 Hz, 37.50 kHz hsync
ModeLine "640x480" 31.5 640 656 720 840
480 481 484 500 -HSync -VSync
# 640x480 @ 85 Hz, 43.27 kHz hsync
Modeline "640x480" 36 640 696 752 832
480 481 484 509 -HSync -VSync
# 640x480 @ 100 Hz, 53.01 kHz hsync
Modeline "640x480" 45.8 640 672 768 864
480 488 494 530 -HSync -VSync

# --- 800x600 ---
# 800x600 @ 56 Hz, 35.15 kHz hsync
ModeLine "800x600" 36 800 824 896 1024
```

```
600 601 603 625
# 800x600 @ 60 Hz, 37.8 kHz hsync
Modeline "800x600" 40 800 840 968 1056
600 601 605 628 +hsync +vsync
# 800x600 @ 72 Hz, 48.0 kHz hsync
Modeline "800x600" 50 800 856 976 1040
600 637 643 666 +hsync +vsync
# 800x600 @ 85 Hz, 55.84 kHz hsync
Modeline "800x600" 60.75 800 864 928 1088
600 616 621 657 -HSync -VSync
# 800x600 @ 100 Hz, 64.02 kHz hsync
Modeline "800x600" 69.65 800 864 928 1088
600 604 610 640 -HSync -VSync

# --- 1024x768 ---
# 1024x768 @ 60 Hz, 48.4 kHz hsync
Modeline "1024x768" 65 1024 1032 1176 1344
768 771 777 806 -hsync -vsync
# 1024x768 @ 87 Hz interlaced, 35.5 kHz hsync
Modeline "1024x768" 44.9 1024 1048 1208 1264
768 776 784 817 Interlace
# 1024x768 @ 70 Hz, 56.5 kHz hsync
Modeline "1024x768" 75 1024 1048 1184 1328
768 771 777 806 -hsync -vsync
# 1024x768 @ 76 Hz, 62.5 kHz hsync
Modeline "1024x768" 85 1024 1032 1152 1360
768 784 787 823
# 1024x768 @ 85 Hz, 70.24 kHz hsync
Modeline "1024x768" 98.9 1024 1056 1216 1408
768 782 788 822 -HSync -VSync
# 1024x768 @ 100Hz, 80.21 kHz hsync
Modeline "1024x768" 115.5 1024 1056 1248 1440
768 771 781 802 -HSync -VSync
EndSection
```

正如上面所示，在一个配置文件中可以有多个Monitor配置小节，用来配置多个显示器。以供后面选择使用。每个Monitor配置小节都应该使用Section “Monitor”开始，以EndSection结束。下面我们就一起来看一下配置选项：

- Identifier、VendorName、ModelName：这三个配置项用来标识不同的显示器，第一个名字任意写，你自己知道就可以了，后面两个也可以随便新，不过写成“unkown”最好。

- HorizSync：设置水平扫描频率，它的缺省单位是kHz，它可以用逗号

分隔开多个独立的数值，也可以使用形如“30-64”的方式来表示一个范围。具体的值应该根据显示器的使用说明上的数据来写。

- VerRefresh：设置垂直扫描频率，它的缺省单位是Hz，表示方式与HorizSync一致，建议根据显示器的使用说明上的数据来写。

24.7 显卡配置小节

在该配置小节中，可以设置显卡的型号、芯片组、晶振芯片、显存。下面是一个配置实例：

```
Section "Device"
Identifier "ATI Mach64"
VendorName “Unknown”
BoardName “Unknown”
#Chipset “Generic”
VideoRam 1024
# Clocks 25.2 28.3
EndSection
```

其配置选项如下：

- Identifier：用来标识显卡
- Chipset：用来标识显卡芯片组
- VideoRam：用来标识显存
- Clocks：晶振芯片

24.8 屏幕配置小节

这一小节是最重要的一个配置小节，它将设置将采用的X Server、显卡及显示器之外，还有显示模式、色彩深度、分辨率和虚拟桌面的设置。下面是一个配置实例：

```
Section "Screen"
Driver "accel"
Device "ATI Mach64"
Monitor "cpq1355"
DefaultColorDepth 16

Subsection "Display"
Depth 16
Modes "800x600" "640x480"
ViewPort 0 0
EndSubsection
```

```
Subsection "Display"
Depth 8
Modes "1152x864" "1024x768" "800x600" "640x480"
ViewPort 0 0
EndSubsection

# BlankTime 3
# StandbyTime 10
# SuspendTime 30
#Offtime 50

EndSection
```

其配置选项如下：

- Driver：选择X Server驱动，在本例中是accel
- Device：选择所使用的显卡，这里的名字应与显卡配置小节的Identifier一致
- Monitor：选择所使用的显示器，这里的名字应与显示器配置小节的Identifier一致
- BlankTime：设置进入屏幕保护的时间，单位为分。
- StandbyTime：设置显示器进入待机状态的时间，单位为分
- SuspendTime：设置显示器进入挂起状态的时间，单位为分
- Offtime：设置显示器关闭的时间，单位为分

在这个小节中，还有Display子小节，用Subsection “Display”做为开始，用EndSubsection表示结束。每一个子小节设置一种显示模式，在此例子中设置了两种显示模式（这些模式在X Window中，可以使用“Ctrl”+“+”、“Ctrl”+“-”来切换。

每个Display子小节中可以有以下配置项：

- Depth：色深，如16代表16位色，8代表8位色
- Modes：可以使用的分辨率，如本配置中表示16位色可以使用”800x600” ”640x480”两种分辨率
- ViewPort：使用虚拟桌面时，设置起始画面的左上角坐标
- Virtual：设置虚拟桌面，后面跟上分辨率即可，如Virtual 1600 1400就表示模拟成为1600x1400大小的桌面。

25 quota配置指南

25.1 关于

这是我从Redhat linux 9的定制指南里摘出来的一段,写的不错,所以推荐给大家。

25.2 实现磁盘配额

除了监视系统上使用的磁盘空间，你还可以通过实现磁盘配额来限制磁盘空间，因此当用户使用了过多的磁盘空间或分区将要充满时，系统管理员就会接到警告。

磁盘配额可以为个体用户配置也可以为用户组配置。这种灵活性既能够给每个用户分配一个较小的配额来处理“个人”文件（如电子邮件和报告），又允许了他们正从事的项目能够拥有较大的配额（假定项目有自己的组群）。

除此以外，配额不仅能够被设置成对所用磁盘块数量的控制，还能够被设置成对内节点数量的控制。由于内节点包含文件相关的信息，对内节点的控制能够控制可被创建的文件数量。

要实现磁盘配额，quota RPM 必须在系统上被安装。

25.3 配置磁盘配额

要实现磁盘配额，请使用以下步骤：

1. 修改/etc/fstab来启用每个文件系统的配额
2. 重新挂载文件系统
3. 创建配额文件，重新生成磁盘用量表
4. 分配配额

以上步骤在下面各节中被详细讨论。

25.3.1 启用配额

以根用户身份使用你喜欢的编辑器来给需要配额的文件系统添加usrquota和（或）grpquota 选项：

```
LABEL=/ / ext3 defaults 1 1
LABEL=/boot /boot ext3 defaults 1 2
none /dev/pts devpts gid=5,mode=620 0 0
LABEL=/home /home ext3 defaults,usrquota,grpquota 1 2
none /proc proc defaults 0 0
none /dev/shm tmpfs defaults 0 0
/dev/hda2 swap swap defaults 0 0
/dev/cdrom /mnt/cdrom udf,iso9660 noauto,owner,kudzu,ro 0 0
/dev/fd0 /mnt/floppy auto noauto,owner,kudzu 0 0
```

在上面的例子中，/home 文件系统上启用了用户和组群配额。

25.3.2 重新挂载文件系统

添加了userquota 和grpquota 选项后，重新挂载每个相应fstab 条目被修改的文件系统。如果某文件系统没有被任何进程使用，使用umount 命令后再紧跟着mount 命令来重新挂载这个文件系统。如果某文件系统正在被使用，要重新挂载该文件系统的最简捷方法是重新引导系统。

25.3.3 创建配额文件

重新挂载了每个启用了配额的文件系统后，系统现在就能够使用磁盘配额了。不过，文件系统本身尚且不能支持配额。下一步是运行quotacheck 命令。

quotacheck 命令检查启用了配额的文件系统，并为每个文件系统建立一个当前磁盘用来的表。该表会被用来更新操作系统的磁盘用量文件。此外，文件系统的磁盘配额文件也被更新。

要在文件系统上创建配额文件（aquota.user 和aquota.group），使用quotacheck 命令的-c 选项。例如，如果用户和组群配额都为/home 分区启用了，在/home 目录下创建这些文件：

```
quotacheck -acug /home
```

> 牙色注:
> 这里有个错误,如果使用了a选项,就不能加入/home这个参数,因为a选项是检测所有的文件系统来决定是否起用了磁盘配额.如果你按这里说的输入,你会得到一个错误提示:
> Bad number of arguments.
> Utility for checking and repairing quota files.
> quotacheck [-gucfinvdmMR] [-F <quota-format>] filesystem|-a
> Bugs to mvw@planets.elm.net, jack@suse.cz

-a 选项意味着在/etc/mtab 中所有挂载了的非NFS 文件系统都会被检查来决定是否启用了配额。-c 选项指定每个启用了配额的文件系统都应该创建配额文件，-u 选项指定检查用户配额，-g 选项指定检查组群配额。

如果-u 或-g 选项被指定，只有用户配额文件被创建。如果只指定了-g 选项，只有组群配额文件会被创建。

文件被创建后，运行以下命令来生成每个启用了配额的文件系统的当前磁盘用量表：

```
quotacheck -avug
```

所用选项如下：

- a －检查所有启用了配额的在本地挂载的文件系统
- v －在检查配额过程中显示详细的状态信息

- u －检查用户磁盘配额信息
- g －检查组群磁盘配额信息

quotacheck 运行完毕后，和启用配额（用户和/或组群）相应的配额文件中就会写入用于每个启用了配额的文件系统（如/home）的数据。

25.3.4 为每用户分配配额

最后一步是使用edquota 命令分配磁盘配额。
要为用户配置配额，以根用户身份在shell 提示下执行以下命令：

edquota username

为每个你想实现配额的用户执行该步骤。例如，如果在/etc/fstab 中为/home 分区（/dev/hda3）启用了配额，执行了edquota testuser 命令后，系统默认的编辑器中就会有如下显示：

```
Disk quotas for user testuser (uid 501):
Filesystem blocks soft hard inodes soft hard
/dev/hda3 440436 0 0 37418 0 0
```

(edquota 使用EDITOR 环境变量所定义的文本编辑器。要改变这个编辑器，把EDITOR 环境变量设置为到你选中的编辑器的完整路径。)
第一列是启用了配额的文件系统的名称。第二列显示了用户当前使用的块数。随后的两列用来设置用户在该文件系统上的软硬块限度。inodes 列显示了用户当前使用的内节点数量。最后两列用来设置用户在该文件系统上的软硬内节点限度。
硬限是用户或组群可以使用的磁盘空间的绝对最大值。达到了该限度后，磁盘空间就不能再被用户或组群使用了。
软限定义可被使用的最大磁盘空间量。和硬限不同的是，软限可以在一段时期内被超过。这段时期被称为过渡期（grace period）。过渡期可以用秒钟、分钟、小时、天数、周数、或月数表示。
如果以上值中的任何一个被设置为0，那个限度就不会被设置。在文本编辑器中，改变想要的限度。如：,

```
Disk quotas for user testuser (uid 501):
Filesystem blocks soft hard inodes soft hard
/dev/hda3 440436 500000 550000 37418 0 0
```

要校验用户的配额是否被设置，使用以下命令：

```
quota testuser
```

25.3.5 为每组群分配配额

配额还可以根据组群来分配。例如，要为devel 组群设置组群配额，使用以下命令（在设置组群配额前，该组群必须存在）：

```
edquota -g devel
```

以上命令在文本编辑器中显示现存的组群配额：

```
Disk quotas for group devel (gid 505):
Filesystem blocks soft hard inodes soft hard
/dev/hda3 440400 0 0 37418 0 0
```

修改限度，保存文件，然后配置配额。

要校验组群配额是否被设置，使用以下命令：

```
quota -g devel
```

25.3.6 为每文件系统分配配额

要根据每个启用了组群的文件系统来分配配额，使用以下命令：

```
edquota -t
```

和另一个edquota 命令相似，这个命令也会在文本编辑器中打开当前的文件系统配额：

```
Grace period before enforcing soft limits for users:
Time units may be: days, hours, minutes, or seconds
Filesystem Block grace period Inode grace period
/dev/hda3 7days 7days
```

改变块过渡期或内节点过渡期，保存对文件的改变，然后退出文本编辑器。

25.4 管理磁盘配额

如果配额被实现，它们就需要被维护－主要维护方式是观察。查看配额是否被超出并确保配额的正确性。当然，如果用户屡次超出他们的配额或者持续地达到他们的软限，系统管理员就可以根据用户类型和磁盘空间对他们工作的影响来做出几种决策。管理员可以帮助用户来检索对磁盘空间的使用，也可以按需要增加用户的配额。

25.4.1 报告磁盘配额

创建磁盘用量报告需要运行repquota 工具。例如，repquota /home 命令会生成以下输出：

```
*** Report for user quotas on device /dev/hda3
Block grace time: 7days; Inode grace time: 7days
Block limits File limits
User used soft hard grace used soft hard grace
----------------------------------------------------------------
root -- 36 0 0 4 0 0
tfox -- 540 0 0 125 0 0
testuser -- 440400 500000 550000 37418 0 0
```

要查看所有启用了配额的文件系统的磁盘用量，使用以下命令：

```
repquota -a
```

这份报告虽然看起来很简单，有几点仍需要做一下说明。显示在每个用户后面的– 是一种判断用户是否超出其块限度或内节点限度的快速方法。如果任何一个软限被超出，相应的- 行就会被- 代替；第一个- 代表块限度，第二个代表内节点限度。

grace 列通常是空白。如果某个软限被超出，这一列就会包含过渡期中的剩余时间。如果过渡期已超过了，其中就会显示none。

25.4.2 保持配额的正确性

当某文件系统没有被完整地卸载（如，由于系统崩溃），这就有必要运行quotacheck。不过，即便系统没有崩溃，quotacheck 也可以被定期经常运行。定期运行以下命令来保持配额的正确性（所用选项在第6.1.1 节中被描述）：

```
quotacheck -avug
```

要定期运行它的最简单方法是使用cron。以根用户身份，你既可以使用crontab -e 命令来调度定期的quotacheck，也可以在以下目录之一内放置一个运行quotacheck 的脚本（使用最适合你需要的间隔期间）：

- /etc/cron.hourly
- /etc/cron.daily
- /etc/cron.weekly
- /etc/cron.monthly

最精确的配额统计数据可以在所分析的文件系统没有被活跃使用时获得。因此，cron 任务应该在文件系统被最少使用时调度。如果这一时间在使用配额的文件系统中并不统一，则使用多个cron 任务在不同的时间为每个文件系统运行quotacheck。

25.4.3 启用和禁用

你可以不必把配额设置为0 来禁用它们。要关闭用户和组群配额，使用以下命令：

```
quotaoff -vaug
```

如果-u 或-g 选项没有被指定，只有用户配额被禁用。如果只指定了-g 选项，只有组群配额会被禁用。

要重新启用配额，使用带有同样选项的quotaon 命令。

例如，要为所有文件系统启用用户和组群配额：

```
quotaon -vaug
```

要为指定文件系统（如/home）启用配额：

```
quotaon -vug /home
```

如果-u 或-g 选项没有指定，那么仅用户配额会被启用。如果只指定了-g 选项，仅组群配额会被启用。

26 vsftp配置实例

配置vsftp,使匿名用户只能下载,用户ftpup则可以上传和下载. 以下操作均以root身份完成

1. 建立ftpup用户和ftpup组,将系统用户ftp加入ftpup组
2. 建立ftp服务目录:

```
mkdir /var/incoming
chmod 755 /var/incoming
chown ftpup /var/incoming
chgrp ftpup /var/incoming
```

3. 修改/etc/vsftpd/vsftpd.conf为如下内容:

```
anonymous_enable=YES
local_enable=YES
write_enable=YES
local_umask=022
dirmessage_enable=YES
xferlog_enable=YES
connect_from_port_20=YES
xferlog_std_format=YES
chroot_list_enable=YES
chroot_local_user=NO
```

```
chroot_list_file=/etc/vsftpd.chroot_list
pam_service_name=vsftpd
userlist_enable=YES
userlist_deny=NO
listen=YES
tcp_wrappers=YES
anon_root=/var/incoming
local_root=/var/incoming
```

4. 修改/etc/vsftpd.ftpusers为如下内容

```
root
bin
daemon
adm
lp
sync
shutdown
halt
mail
news
uucp
operator
games
nobody
```

5. 修改/etc/vsftpd.chroot_list为如下内容

```
ftpup
```

6. 修改/etc/vsftpd.user_list为如下内容

```
ftpup
anonymous
```

27 RedHat Linux 8.0安装指南

27.1 安装前的准备

1. 硬件兼容:在安装时RedHat Linux 8.0会自动的检测到绝大部分硬件，但是硬件的标准更换很快，不可能保证能检测到所有硬件，如果没有检测到，请选择最符合标准的硬件配置，或者按照安装程序提供的方法填入硬件参数进行手工配置。

2. 磁盘空间：请保证提供给RedHat Linux 8.0使用的空间超过2.5G.

3. 安装方式：请先准备好RedHat Linux 8.0光盘（一共6张，3张系统安装盘，两张源代码盘，1张中文文档）。我们将从光盘安装RedHat Linux 8.0.

4. 安装类型：RedHat Linux允许选择以下几种安装类型:[个人桌面]、[工作站]、[服务器]、[定制]和[升级]。我们将选择[定制]安装RedHat Linux 8.0.

27.2 开始进行安装

安装说明：以下步骤中如果没有特别说明，请使用系统的默认配置。

1. 改变计算机的引导顺序

   启动计算机，按住[Del]键，进入BIOS设置程序，查找能够改变引导顺序的部分。请改变该顺序，使光盘位于该引导程序的最前面。存盘退出，机器重新启动，如果能够正确的从光驱读取RedHat Linux 安装光盘，这能继续下面的安装。

2. 光盘引导RedHat Linux 安装程序

   当读取RedHat Linux 8.0的第一张光盘成功后，不久一个包含boot:的屏幕就会出现。根据我们的安装方法，用户只需要按一下[Enter]键即可。如果用户在1分钟内没有采取任何行动，安装程序也会自动开始。接下来会进入图形化安装界面。

3. 进入使用RedHat Linux的欢迎界面，点击[下一步]继续；

4. 语言选择

   选择“中文（简体）（文（简体））”，点击[下一步]继续；

5. 键盘配置

   使用鼠标来选择你要在本次安装中和今后作系统默认的键盘布局类型（例如:U.S.Enlish），一般选系统检测到的键盘类型，点击[下一步]继续；

6. 鼠标配置

   为系统选择正确的鼠标类型。如果你找不到正确的匹配，选择你确定会与你的系统兼容的鼠标类型。点击[下一步]继续；

7. 安装类型

   RedHat Linux允许选择以下几种安装类型:[个人桌面]、[工作站]、[服务器]、[定制]和[升级]。请选择[定制]，点击[下一步]继续；

8. 磁盘分区设置

   请选择[用Disk Druid手工分区], 点击[下一步]继续；

9. 用Disk Druid手工分区

   到这一步，你必须告诉安装程序要在哪里安装RedHat Linux8.0。这是通过在将要安装RedHat Linux8.0的一个或多个磁盘分区上定义挂载点来做到的。

   我们推荐创建下列分区：

   - 一个交换分区（至少32M）。交换分区用来支持虚拟内存。当没有足够的内存来容纳你的系统正在处理的数据时，某些数据就被写如交换区。你的交换分区的最小值应该相当于你的计算机内存的2倍和32M中较大的一个值。如果你的内存小于等于1G，你的交换分区应至少与你的系统内存相等或是它的2倍。如果内存大于1G，建议使用2G交换区。创建一个有大量空间的交换分区将会在你未来升级内存的时候发挥作用。

   - 一个/boot分区（75M，ext3格式）。这个挂载在/boot上的分区包含操作系统的内核（允许你的系统引导RedHat Linux），以及其他几个在引导过程中使用的文件。鉴于多数PC BIOS的限制，创建一个小分区来容纳这些文件是较佳的选择。对大多数用户来说，75M引导分区应该是足够的。

   - 一个/（根）分区（1.5M-4.5M,ext3格式）。这个根目录被挂载的位置。在这个设置中，所有文件都都位于根分区上。一个大小约为1.5G的根分区可以容纳与个人桌面和工作站相当的安装（只剩极少空闲空间），而一个大于4.5G的根分区将会允许你安装每一个软件包。如果你的硬盘空间很大，并且都准备用来装RedHat Linux 8.0，您可以把除交换分区和/boot分区外的所有剩余空间分配给跟分区。也可以参照相关的讲解linux分区的资料增加其他的分区（/var,/opt,/etc/等）。

   分区完毕后点击[下一步]继续；

10. 引导装载程序的配置

    安装程序为你提供了2个引导装载程序：GRUB和LILO。用户可以任选其中一个，默认的是使用GRUB作引导装载程序。点击[下一步]继续；

11. 防火墙配置

    安全级别：中级选择[定制]来添加信任的设备或允许其他的进入接口。用鼠标在eth0、WWW(HTTP)、FTP、Telnet的前面打勾表示选中。点击[下一步]继续；

12. 语言支持的选择

    必须使用一种语言作为默认的语言。当安装结束后系统将使用默认的语言。如果选择了安装其他的语言，可以在安装完毕后改变默认

语言。除了默认的语言English(USA)之外，推荐选中Chinese(P.R.of China)。点击[下一步]继续；

13. 选择时区

    选择[亚洲/上海]。点击[下一步]继续；

14. 帐号配置

    输入管理员口令（请注意这里输入的口令是您安装完系统后用root登陆时必须输入的口令。请务必记住）。点击[下一步]继续；

15. 验证配置

    选择默认的选项即可，点击[下一步]继续；之后需要等待一段时间，安装程序提示：正在读取软件包.....

16. 软件包组的选择

    分为几个组：桌面、应用程序、软件开发、服务器等，在安装程序默认的基础上，作为邮件服务器建议不选桌面、应用开发、软件开发，另外服务器组选种如下软件包组：

    - 服务器配置工具
    - 万维网服务器
    - 邮件服务器
    - windows文件服务器
    - DNS服务器
    - FTP服务器
    - 网络服务器

    点击[下一步]继续；

17. 准备安装

    你应该看到一个为你安装作准备的屏幕。点击[下一步]继续；

18. 安装软件包

    到了这一步，在所有软件包被安装完的过程中你除了要换2次光盘外将不必进行其他任何操作。安装软件包完毕后将跳到下一步。

19. 创建引导盘

    强烈建议创建一张引导盘，按照安装程序的提示，插入一张干净的未被写保护的软盘，选择[是，我想创建引导盘]，点击[下一步]继续，等待数分钟，这期间引导盘将被创建。

20. 图形化界面(X)配置

    如果要为你的系统配置一个X服务器，安装程序会给你一个视频卡列表，一般选用系统检测的结果即可。如果没有检测到，请选择跳过X配置，转到下一步骤。点击[下一步]继续；

    (a) 显示器配置为了完成X配置，必须配置你的显示器并定制X设置。系统会自动检测到你的显示器并给出相应的参数（水平频率范围和垂直频率范围）。如果你选择了跳过X配置，请转到下一步骤。点击[下一步]继续；

    (b) 定制图形化配置为你的X配置选择正确的色彩深度和分辨率。点击[测试设置]来试用这个配置。如果你不喜欢在测试中见到的，点击[否]来选择另一种分辨率。

    另外可以选择你要登陆的类型：图形化或文本。选定后点击[下一步]继续；

21. 安装完成

    恭贺你！你的RedHat Linux 8.0安装现已完成！安装程序会提示你做好重新引导系统的准备。点击退出完成安装。

27.3 安装后的注意事项

1. 保管好创建的RedHat Linux 8.0系统引导盘，这在系统不能正常启动的情况下是很重要的。
2. 不要外泄root口令。并且管理员要经常更改root口令。
3. 管理员在平时的工作中不要用root身份登陆，以免因为不正确或失误的操作破坏系统。当用root登陆系统后，如果离开或暂时不用，必须用exit命令退出root身份，以免别非法使用。

28 Samba的三种典型配置

pub －不需要密码，且可读写及删除文件。
read-only －不需要密码，但只可以读取文件。
user1 －需要密码，可读写及删除文件。

步骤如下：

1. 首先以root身分登录进入系统。
2. 编辑/etc/smb.conf文件，将”unix password sync = no”这个一句改为”unix password sync = yes”。这样子的话，以后系统增加使用者时，会自动将该使用者的密码也更新到/etc/smbpasswd内(Samba的帐号密码文件)。

3. 到/home目录下增加下列目录，并指定这些目录的权限：

```
/home/pub nobody:nobody 777
/home/read-only root:root 755
/home/user1 user1:user1 700
```

4. 编辑/etc/smb.conf这个文件，修改：

```
security = share
```

5. 编辑/etc/smb.conf这个文件，到文件最后面增加下面几句：

```
[public]
comment = Public Areas
path = /home/pub
browseable = yes
guest ok = yes
writable =yes

[read-only]
comment = Read-Only Areas
path = /home/read-only
browseable = yes
guest ok = yes

[user1]
comment = Password Required
path = /home/user1
browseable = yes
writable = yes
```

完成后存盘离开。

6. 重行运行Samba：

```
/etc/rc.d/init.do/smb restart
```

就可以在其它的电脑上共享到这些目录了。

29 ls命令选项详解

ls 命令可以说是Linux下最常用的命令之一。它有众多的选项，其中有很多是很有用的，你是否熟悉呢？下面列出了ls 命令的绝大多数选项。

- -a 列出目录下的所有文件，包括以. 开头的隐含文件。

- -b 把文件名中不可输出的字符用反斜杠加字符编号(就象在C语言里一样)的形式列出。
- -c 输出文件的i 节点的修改时间，并以此排序。
- -d 将目录象文件一样显示，而不是显示其下的文件。
- -e 输出时间的全部信息，而不是输出简略信息。
- -f -U 对输出的文件不排序。
- -g 无用。
- -i 输出文件的i 节点的索引信息。
- -k 以k 字节的形式表示文件的大小。
- -l 列出文件的详细信息。
- -m 横向输出文件名，并以“，”作分格符。
- -n 用数字的UID,GID 代替名称。
- -o 显示文件的除组信息外的详细信息。
- -p -F 在每个文件名后附上一个字符以说明该文件的类型，“*”表示可执行的普通文件；“/”表示目录；“@”表示符号链接；“|”表示FIFOs；“=”表示套接字(sockets)。
- -q 用?代替不可输出的字符。
- -r 对目录反向排序。
- -s 在每个文件名后输出该文件的大小。
- -t 以时间排序。
- -u 以文件上次被访问的时间排序。
- -x 按列输出，横向排序。
- -A 显示除“.”和“..”外的所有文件。
- -B 不输出以“~”结尾的备份文件。
- -C 按列输出，纵向排序。
- -G 输出文件的组的信息。
- -L 列出链接文件名而不是链接到的文件。
- -N 不限制文件长度。

- -Q 把输出的文件名用双引号括起来。
- -R 列出所有子目录下的文件。
- -S 以文件大小排序。
- -X 以文件的扩展名(最后一个. 后的字符)排序。
- -1 一行只输出一个文件。
- --color=no 不显示彩色文件名
- --help 在标准输出上显示帮助信息。
- --version 在标准输出上输出版本信息并退出。

30 用Kickstart批量安装Linux

日常工作中，我们经常需要在多台硬件完全相同的电脑上安装同样的Linux，尤其是在做集群应用或互为备份的数据库服务器的时候。笔者是兼职Linux教师，出于讲课的需要，经常要在硬件配置完全相同的电脑上为学生安装多达十台或更多的Linux系统。如果使用最常见的CDROM安装方式，恐怕一次就要用掉一天的时间。而采用Kickstart 安装方式，大约60分钟就全部安装完毕了。

Kickstart是Red Hat发展的快速定制安装方式，可以让电脑按照事先设计好的方式自动安装。最常见的方式是网络安装，也可以采用CDROM或硬盘安装。本文主要介绍Kickstart的网络安装。

30.1 制作启动盘

制作安装盘的工作既可以在Windows系统下完成，也可以在Linux系统下完成。

在Windows系统中，使用Red Hat安装光盘中第一张的rawritewin程序。此程序在CDROM下的dosutils/rawritewin目录下。制作第一张软盘时，Image File请选择images bootnet.img ，这张叫启动盘。制作第二张软盘的时候，Image File请选择images drvnet.net，这张叫驱动盘(Driver Disk)。根据你网卡的型号，驱动盘也许并不需要。

如果是在Linux系统中制作安装盘，命令如下：

```
dd if=/mnt/cdrom/images/bootnet.img of=/dev/fd0 (启动盘)
dd if=/mnt/cdrom/images/drvnet.img of=/dev/fd0 (驱动盘)
```

启动盘里包含了最常见的网卡驱动，如3Com 509/Intel eepro NE 2000等。如果你的网卡不是最常见的，比如联想D-Link530TX，则必须多做一张网卡的驱动盘。

30.2 编辑文件syslinux.cfg

编辑启动盘上的syslinux.cfg 文件，将第一行改写为：

```
default linux ks=floppy
```

如果需要用到网卡的驱动盘，则第一行改写为：

```
default linux ks=floppy dd
```

此时系统会自动提示插入驱动盘。然后删除prompt 和timeout 行。

30.3 编辑ks.cfg文件

这一步也是Kickstart安装中最重要的步骤。ks.cfg 是Kickstart安装的核心文件，它指明了以什么方式、将Linux安装到何处、安装什么package等内容。ks.cfg 放置在软盘的根目中。

在新安装的Red Hat系统下，/root/下有个anaconda.cfg文件，可以它为基础进行编辑，也可以用X-Window下的ksconfig程序进行这项工作。如果读者用ksconfig程序进行设定，可能还需要手动进行修改。下面是ks.cfg 文件的内容，为了方便解说，我将原文加了行号。请注意正式的文件是不能加行号的。

```
1 #Generated by Kickstart Configurator
2 #System language
3 lang en_US
4 #Language modules to install
5 langsupport --default en_US en_US zh_CN.GB2312
6 #System keyboard
7 keyboard us
8 #System mouse
9 mouse genericps/2
10 #System timezone
11 timezone --utc Asia/Shanghai
12 #Root password
13 rootpw 12345
14 #System bootloader configuration
15 bootloader --location=mbr
16 #Install Red Hat Linux instead of upgrade
17 install
18 #Use FTP installation media
19 url --url ftp://192.168.203.2/download
20 #Disk partitioning information
21 clearpart --all
22 part / --size 4200
23 part swap --size 300
```

```
24 #Use DHCP networking
25 network --bootproto dhcp
26 #System authorization information
27 auth --useshadow --enablemd5
28 #Firewall configuration
29 firewall --disabled
30 #XWindows configuration information
31 #Probe for video card
32 #Probe for monitor
33 xconfig --depth 16 --resolution 1024x768 --defaultdeskt-
   op=GNOME
34 %packages
35 @KDE
36 @Emacs
37 lynx
38 %pre
39 echo " Welcome to my kickstart"
40 %post
41 echo "192.168.10.55 Server" >> /etc/hosts
```

所有以“”号开头的都是注释，可以忽略。

第2行指明了安装时采用的语言。其实在配置正确的时候，kickstart安装不需要人工干预，本文将其定为英语。

第4行指明了系统支持的语言环境，如果不安装X-Window，只需要英语即可。本文增加了对简体中文的支持。

第6-9行指明了系统采用的键盘和鼠标的类型，通常键盘都是us兼容行的。带滚轮的PS/2鼠标应写为:

```
mouse msintellips/2
```

第6-9行建议用ksconfig 程序配置。

第10-11行是时区，中国的用户一般可以选择上海。

第12-13行指明了root用户的密码，还有一种形式就是将密码进行加密。

第16-17行指明了是安装还是升级，如果是升级，第17行应该用upgrade代替Install参数。

第18-19行指明了安装介质所在的位置，这是Kickstart安装的关键之一。安装介质可以放置在NFS/FTP/HTTP服务器上，也可以放置在本机硬盘上。本文不打算探讨硬盘安装，只研究网络安装。具体的办法是把3张Red Hat安装光盘中的Red Hat目录拷贝到到服务器的某个位置，比如说/tmp/install 下面，如果是用NFS安装，则需要把/tmp/install 共享出去，同时要保证将要安装Red Hat Linux的客户机可以访问。只读的权限可以按如下配置:

```
/etc/exports 文件如下
/tmp/install 192.168.10.0/255.255.255.0(ro)
```

如果是FTP或者HTTP安装，则可以把Red Hat目录放在ftproot或者httproot下。值得注意的是，在Windows系统中，可能对“/”和“\”产生歧意，如果FTP Server 是Windows系统，最好将Red Hat目录放置到ftproot的再下一级目录中。

第21-23行是描述Linux将安装到哪个分区，clearpart -all 的意思是清除所有分区。第22行新建一个4.2GB的分区作为“/”分区。第23行新建立一个300MB的swap分区。

如果使用已存在的分区，则写为：

```
partition / --onpart sda1
partition swap -onpart sda2
```

注意：第21-23行最好不要用ksconfig程序自动设定参数，手工配置比较稳妥。

第25行的意思是用DHCP Server自动分配IP地址。如果装机的数量较多，DHCP的方式毫无疑问是首选。如果想自己指定IP地址，则应该写为：

```
network --bootproto static --ip 192.168.10.55 --ne-
tmask 255.255.255.0 --gateway 192.168.11.1 --na-
meserver 202.96.134.133
```

多网卡的机器稍微复杂一点，本文就不介绍了。

第27行指明了系统验证用户的方式。本文用的是缺省Shadow密码，MD5方式加密，这是最常见的方式。

第28行把自定义的防火墙关闭。因为安装Red Hat时用的是ipchains 防火墙，因此笔者建议安装以后采用iptables防火墙。

第30-33行是关于X-Window的配置。意思是使用系统自动侦测的配置。Linux一般都能正确地检测到显示卡和显示器的型号。

第33行的意思是以GNOME做为缺省的窗口管理器，颜色为16位色，屏幕分辨为1024×768。如果在末尾加上“–startxonboot”，就是开机后X-Window登陆。

第34-37行指明了安装时候选择安装的软件。前面有个“@”符号代表一组RPM包，每组包将安装的软件列表,这样软件可以在Red Hat第一张安装光盘中的Redhat/base/comps 文件中找到。如果一个组都不指定，则会安装Redhat/base/comps文件中的Base组合，也就是最基本的一些RPM包，如Sendmail等。写一个@Everything 则是完全安装。第37行中有单独安装了lynx的RPM包。

第38-39行的%pre表示安装之前执行的命令。

第40-41行%post以后代表安装之后将要执行的命令。第40行执行了一个简单的命令。

最后的工作是把写好的ks.cfg 拷贝到软盘上：

```
cp ks.cfg /mnt/floppy
```

笔者在Windows 2000下安装了VMware，在虚拟机上用光盘安装Linux时，不知道什么原因光驱的速度非常慢。因此又在Windows 2000下安装了Server-U FTP服务器，然后在VMware下进行Kickstart安装，结果速度非常令人满意。另外，也可以用Omni-NFS在Windows 2000下实现NFS Server的功能，使用NFS安装。

善用Kickstart安装中的%post 命令，就可以直接配置好服务器。

Kickstart安装是学习Linux的重点之一，在Red Hat RH133培训课程中，就有Kickstart安装的课程，但这方面的中文资料一直都比较少，希望本文能起到抛砖引玉的作用。本文介绍的内容也适用于最新的Red Hat 8.0。

31 samba服务器

31.1 Samba协议基础

在NetBIOS出现之后，Microsoft就使用NetBIOS实现了一个网络文件/打印服务系统，这个系统基于NetBIOS设定了一套文件共享协议，Microsoft称之为SMB（Server Message Block）协议。这个协议被Microsoft用于它们Lan Manager和Windows NT服务器系统中，实现不同计算机之间共享打印机、串行口和通讯抽象（如命名管道、邮件插槽等）。

随着Internet的流行，Microsoft希望将这个协议扩展到Internet上去，成为Inter net上计算机之间相互共享数据的一种标准。因此它将原有的几乎没有多少技术文档的SMB协议进行整理，重新命名为CIFS（Common Internet File System），并打算将它与NetBIOS相脱离，试图使它成为Internet上的一个标准协议。

因此，为了让Windows和Unix计算机相集成，最好的办法即是在Unix计算机中安装支持SMB/CIFS协议的软件，这样Windows客户就不需要更改设置，就能如同使用Windows NT服务器一样，使用Unix计算机上的资源了。Samba是用来实现SMB的一种软件，它的工作原理是，让NETBIOS（Windows95网络邻居的通讯协议）和SMB（Server Message Block）这两个协议运行于TCP/IP通信协议之上，并且使用Windows的NETBEUI协议让Unix计算机可以在网络邻居上被Windows计算机看到。它的功能有：

- 共享Linux磁盘给Win95/NT
- 共享Win95/NT磁盘给Linux机器
- 共享Linux打印机给win95/NT
- 共享win95/NT打印机给Linux机器。

同时它的文件服务功能比NT系统还高，而且在Windows2000之前就提供了用户磁盘空间限制的功能。

31.2 Samba的配置

Samba的配置文件是smb.conf，现在根据配置文件讲讲它的配置选项：

31.2.1 全局设置:

```
workgroup = MYGROUP
```

定义该Samba服务器所在的工作组或者域（如果下面的security=domain的话）。

```
server string = MY Samba Server
```

设定机器的描述，当我们通过网络邻居访问的时候可以在备注里面看见这个内容，而且还可以使用samba设定的变量。这里说一下samba定义的变量:

```
%S = 当前服务名（如果有的话）
%P = 当前服务的根目录（如果有的话）
%u = 当前服务的用户名（如果有的话）
%g = 当前用户说在的主工作组
%U = 当前对话的用户名
%G = 当前对话的用户的主工作组
%H = 当前服务的用户的Home目录
%v = Samba服务的版本号。
%h = 运行Samba服务机器的主机名
%m = 客户机的NETBIOS名称
%L = 服务器的NETBIOS名称
%M = 客户机的主机名
%N = NIS服务器名
%p = NIS服务的Home目录
%R = 说采用的协议等级(值可以是CORE, COREPLUS, LANMAN1, LAN-
     MAN2, NT1)
%d = 当前服务进程的ID
%a = 客户机的结构（只能识别几项：Samba，WfWg，WinNT，Win95）
%I = 客户机的IP
%T = 当前日期和时间

hosts allow = 网络或者主机
```

这里可以设置允许访问的网络和主机IP，比如允许192.168.1.0/24和192.168.2.1/32访问，就用host allow = 192.168.1. 192.168.2.1 127.0.0.1(网络注意后面加”.”号，各个项目间用空格隔开，记得把本机也加进去)

```
printcap name = printcapFile
```

到printcapFile（一般是/etc/printcap）这个文件中取得打印机的描述信息

```
load printers = yes|no
```

设定是否自动共享打印机而不用设置下面的[printer]一节的相关东西

```
printing = PrintSystemType
```

定义打印系统的类型，缺省是lprng,可选项有：bsd, sysv, plp, lprng, aix, hpux, qnx。

```
guest account = pcguest
```

定义游客帐号，而且需要把这个帐号加入/etc/passwd，不然它就用缺省的nobody

```
log file = LogFileName
```

定义记录文件的位置LogFileName（一般是用/var/log/samba/%m.log）

```
max log size = size
```

定义记录文件的大小size（单位是KB，如果是0的话就不限大小）

```
security = security_level
```

定义Samba的安全级别，按从低到高分为四级：share，user，server，domain。它们对应的验证方式如下：

- share:没有安全性的级别，任何用户都可以不要用户名和口令访问服务器上的资源。
- user:samba的默认配置，要求用户在访问共享资源之前资源必须先提供用户名和密码进行验证。
- server:和user安全级别类似，但用户名和密码是递交到另外一个服务器去验证，比如递交给一台NT服务器。如果递交失败，就退到user安全级。
- domain:这个安全级别要求网络上存在一台Windows的主域控制器，samba把用户名和密码递交给它去验证。

后面三种安全级都要求用户在本Linux机器上也要系统帐户。否则是不能访问的。

```
password server = <NT-Server-Name>
```

当前面的security设定为server或者domain的时候才有必要设定它。

```
password level = n
```

这是设定针对一些SMB客户像OS/2之类而设的，这样的系统在发送用户密码的时候，会把密码转换成大写再发送，这样就和samba的密码不一致，这个参数可以设定密码里允许的大写字母个数，这样samba就根据这个数目对接收到的密码进行大小写重组，以重组过的密码尝试验证密码的正确性。n越大，组合的次数就越多，验证时间就越长，安全性也会因此变得越低。例如n=2，用户的密码是abcd，但发送出去其实

是ABCD，samba就会把这个ABCD进行大小写重组，组合后的结果可以是: Abcd,aBcd,abCd,abcD,abcd,ABcd,AbCd,AbcD,aBCd,aBcD,abCD。所以如果没有必要，就把n定为是零。这样的话samba只尝试两次，一个是接收到的密码，另一个尝试的是这个密码都是小写的情况。

```
username level = n
```

这个是对于用户名的情况，说明和上面一项类似。

```
encrypt passwords = yes|no
```

设置是否对密码进行加密，samba本身有一个密码文件/etc/samba/smbpasswd，如果不对密码进行加密则在验证会话期间客户机和服务器之间传递的是明文密码，samba直接把这个密码和Linux里的/etc/samba/smbpasswd密码文件进行验证。但是在Windows 95 OS/R2以后的版本和Windows NT SP3以后的版本缺省都不传送明文密码，要让这些系统能传送明文密码必须在其注册表里更改，比较麻烦，好的方法就是把这里的这个开关设置为yes。

```
smb passwd file = smbPasswordFile
```

设置存放samba用户密码的文件smbPasswordFile(一般是/etc/samba/smbpasswd)。

```
ssl CA certFile = sslFile
```

当samba编译的时候支持SSL的时候，需要指定SSL的证书的位置（一般在/usr/share/ssl/certs/ca-bundle.crt）。

```
unix password sync = yes|no
passwd program = /usr/bin/passwd %u
passwd chat = *New*UNIX*password* %n
*ReType*new*UNIX*password* %n
*passwd:*all*authentication*tokens*updated*successfully*
```

这三项设置能否从windows的应用程序修改unix系统的用户密码

```
username map = UsermapFile
```

指定用户映射文件（一般是/etc/samba/smbusers），当我们在这个文件里面指定一行root = administrator admin的时候，客户机的用户是admin或者administrator连接时会被当作用户root看待。

```
include = MachineConfFile
```

指定对不同机器的连接采用不同的配置文件MachineConfFile（一般为了灵活管理使用/etc/samba/smb.conf.%m，由于采用了samba的变量，把配置文件和客户机的NETBIOS名称关联起来，能很容易地控制这些客户机的权限和设置）。

```
socket options = TCP_NODELAY SO_RCVBUF=8192 SO_SNDBUF=8192
```

这个是网络socket方面的一些参数，能实现最好的文件传输性能。相关的选项还有SO_KEEPALIVE、SO_REUSEADDR、SO_BROADCAST、IPTOS-_LOWDELAY、IPTOS_THROUGHPUT、SO_SNDLOWAT（*）、SO_RCVLOWAT（*），带*号的要指定数值。一般如果在本地网络，就只用IPTOS_LOWDELAY,如果是有一个本地网络的，就用IPTOS_LOWDELAY TCP_NODELAY，如果是广域网络，就试试IPTOS_THROUGHPUT。

```
interfaces = interface1 interface2
```

如果有多个网络接口，就必须在这里指定。如interface = 192.168.12.2/24 192.168.13.2/24 remote browse sync = host(subnet) 这里指定浏览列表同步信息从哪里取得，如果用host（比如192.168.3.25）或者整个子网（192.168.5.255）。

这里说明一下什么是浏览（Browse）：

在SMB协议中，计算机为了访问网络资源，就需要了解网络上存在的资源列表（例如在Windows下使用网络邻居查看可以访问的计算机），这个机制就被称为浏览（Browse）。虽然SMB协议中经常使用广播的方式，但如果每次都使用广播的方式了解当前的网络资源（包括提供服务的计算机和各个计算机上的服务资源），就需要消耗大量的网络资源和浪费较长的查找时间，因此最好在网络中维护一个网络资源的列表，以方便查找网络资源。只有必要的时候，才重新查找资源，例如使用Windows下的查找计算机功能。

但没有必要每个计算机都维护整个资源列表，维护网络中当前资源列表的任务由网络上的几个特殊计算机完成的，这些计算机被称为Browser，这些Browser通过记录广播数据或查询名字服务器来记录网络上的各种资源。Browser并不是事先指定的计算机，而是在普通计算机之间通过自动进行的推举产生的。不同的计算机可以按照其提供服务的能力，设置在推举时具备的不同权重。为了保证一个Browser停机时网络浏览仍然正常，网络中常常存在多个Browser，一个为主Browser（Master Browser），其他的为备份Browser。

```
remote announce = host(subnet)
```

指定这些机器向网络宣告自己，而不是有Browser得到。

```
local master = yes|no
```

这个参数指定nmbd是否试图成为本地主浏览器，默认值是yes，如果设为no则samba服务器就永远都不会成为本地主浏览器。但即使设置了yes，也不等于samba服务器就会成为本地主浏览器。只是参与本地主浏览器选择。

```
os level = n
```

n的值是个整数，决定了nmbd是否有机会成为本地广播区域的工作组里的本地主浏览器，默认值是零，零则意味着nmbd失去浏览选择。如果要nmbd更有机会成为本地主浏览器的话，可以设为65。

```
domain master = yes|no
```

这个参数让nmbd成为一个域浏览器，取得各本地主浏览器的浏览列表，并将整个域的浏览列表递交给各本地主浏览器。

```
preferred master = yes|no
```

这个参数指定nmbd是否是工作组里的首要的主浏览器，如果指定为yes，nmbd在启动的时候就强制一个浏览选择。

Domain master和local master

工作组和域这两个概念在进行浏览时具备同样的用处，都是用于区分并维护同一组浏览数据的多个计算机。事实上他们的不同在于认证方式上，工作组中每台计算机都基本上是独立的，独立对客户访问进行认证，而域中将存在一个（或几个）域控制器，保存对整个域中都有效的认证信息，包括用户的认证信息以及域内成员计算机的认证信息。浏览数据的时候，并不需要认证信息，Microsoft将工作组扩展为域，只是为了形成一种分级的目录结构，将原有的浏览和目录服务相结合，以扩大Mircrosoft网络服务范围的一种策略。

工作组和域都可以跨越多个子网，因此网络中就存在两种Browser，一种为Domain Master Browser ，用于维护整个工作组或域内的浏览数据，另一种为Local Master Browser，用于维护本子网内的浏览数据，它和Domain Master Browser通信以获得所有的可浏览数据。划分这两种Browser 主要是由于浏览数据依赖于本地网广播来获得资源列表，不同子网之间只能通过浏览器之间的交流能力，才能互相交换资源列表。

但是，为了浏览多个子网的资源，必须使用NBNS名字服务器的解析方式，没有NBNS的帮助，计算机将不能获得子网外计算机的NetBIOS名字。Local Master Browser也需要查询NetBIOS名字服务器以获得Domain Master Browser的名字，以相互交换网络资源信息。

由于域控制器在域内的特殊性，因此域控制器倾向于被用做Browser，主域控制器应该被用作Domain Master Browser，他们在推举时设置的权重较大。

```
preserve case = yes|no
short preserve case = yes|no
```

指定拷贝DOS文件的时候保持大小写，缺省是no

```
default case = lower|upper
```

所有的DOS文件的缺省是大写还是小写

```
case sensitive = yes|no
```

大小写敏感，一般是no,不然会出现一些问题。

31.3 共享设置

共享资源:

每个SMB服务器能对外提供文件或打印服务，每个共享资源需要被给予一个共享名，这个名字将显示在这个服务器的资源列表中。如果一个资源的名字的最后一个字母为$，则这个共享名就为隐藏共享，不能直接表现在浏览列表中，而只能通过直接访问这个名字来进行访问。在SMB协议中，为了获得服务器提供的资源列表，必须使用一个隐藏的资源名字IPC$来访问服务器，否则客户无法获得系统资源的列表。

共享设置中有个比较奇怪的段:

homes

，在smb.conf文件中一般没有对这个目录的设定特定内容比如路径等。当客户机发出服务请求时，就在smb.conf文件的其它部分查找友特定内容的服务。如果没有发现这些服务，并且提供了homes段时，那么就搜索密码文件得到用户的Home目录。通过Homes段，Samba可以得到用户的Home目录并使之共享。下面是这个段的最基本的几个设置。

```
[homes]
comment=Home Directory
browseable=no
writable=yes
```

比较正常的共享的配置如下例：

```
[MyShare]
comment = grind' s file
path = /home/grind
allow hosts = host(subnet)
deny hosts = host(subnet)
writable = yes|no
user = user(@group)
valid users = user(@group)
invalid users = user(@group)
read list = user(@group)
write list = user(@group)
admin list = user(@group)
public = yes|no
hide dot files = yes|no
create mode = 0755
directory mode = 0755
sync always = yes|no
short preserve case = yes|no
preserve case = yes|no
case sensitive = yes|no
mangle case = yes|no
```

```
default case = upper|lower
force user = grind
wide links = yes|no
max connections = 100
delete readonly = yes|no
```

其中[]里面的MyShare指定共享名，一般就是网络邻居里面可以看见的文件夹的名字。comment指的是对改共享的备注。

path指定共享的路径，其中可以配合samba变量使用。比如你可以指定path=/data/%m，这样如果一台机器的NETBIOS名字是grind,它访问MyShare这个共享的时候就是进入/data/grind目录,而对于NETBIOS名是glass的机器，则进入/data/glass目录。

allow hosts和deny hosts和前面的全局设置的方法一样这里不再提及。

writeable指定了这个目录缺省是否可写，也可以用readonly = no来设置可写。

user设置所有可能使用该共享资源的用户，也可以用@group代表group这个组的所有成员，不同的项目之间用空格或者逗号隔开。

valid users指定能够使用该共享资源的用户和组。

invalid users指定不能够使用该共享资源的用户和组。

read list 指定只能读取该共享资源的用户和组。

write list指定能读取和写该共享资源的用户和组。

admin list指定能管理该共享资源（包括读写和权限赋予等）的用户和组。

public指明该共享资源是否能给游客帐号访问，这个开关有时候也叫guest ok，所以有的配置文件中出现guest ok = yes其实和public = yes是一样的。

hide dot files指明是不是像unix那样隐藏以“.”号开头的文件。

create mode指明新建立的文件的属性，一般是0755。

directory mode指明新建立的目录的属性，一般是0755。

sync always指明对该共享资源进行写操作后是否进行同步操作。

short preserve case指明不管文件名大小写。

preserve case指明保持大小写。

case sensitive指明是否对大小写敏感，一般选no,不然可能引起错误。

mangle case指明混合大小写。

default case指明缺省的文件名是全部大写还是小写。

force user强制把建立文件的属主是谁。如果我有一个目录，让guest可以写，那么guest就可以删除，如果我用force user= grind强制建立文件的属主是grind，同时限制create mask=0755，这样guest就不能删除了。

wide links指明是否允许共享外符号连接，比如共享资源里面有个连接指向非共享资源里面的文件或者目录，如果设置wide links = no将使该连接不可用。max connections = n设定同时连接数是n。

delete readonly指明能否删除共享资源里面已经被定义为只读的文件。

有两类特殊的共享，分别是光驱和打印机光驱的共享设置:

```
[cdrom]
comment = grind’s cdrom
path = /mnt/cdrom
public = yes
browseable = yes
root preexec = /bin/mount -t iso9660 /dev/cd0 /mnt/cdrom
root postexec = /bin/umount /mnt/cdrom
```

这里root preexec指明了连接时用root的身份运行mount命令，而root postexec则指明了断开时用root身份运行umount,有效实现了对光驱的共享。

打印机共享的设置：

```
[printers]
path = /var/spool/samba
writeable = no
guest ok = yes
printable = yes
printer driver = HP LaserJet 5L
```

这里printable指明该打印机可以打印，guest ok说明游客也能打印，path指明打印的文件队列暂时放到/var/spool/samba目录下。printer driver的作用是指明该打印机的类型，这样我们在安装网络打印机的时候可以直接自动安装驱动而不必选择。

添加用户：

samba添加用户比较方便，一般是用smbadduser，用法是smbadduser unixid:netid，举个例子：如果你的本机有个叫grind的用户，你用smbadduser grind:grind，这样从网上邻居访问的时候用户名就用grind，而如果你用了smbadduser grind:glass的话，网上邻居访问的时候提供的用户名就是glass而不是grind了。

关于samba服务端的设置基本就是这些，我想一般应用中所要使用的上面几乎都覆盖到了，所以有些不重要的就省略了，如果要更加详细的信息可以用man smb.conf参考。

32 Linux系统各文件、目录介绍

32.1 简介

Linux操作系统中,以文件来表示所有的逻辑实体与非逻辑实体。逻辑实体系指文件与目录; 非逻辑实体则泛指硬盘、终端机、打印机等。

一般而言,Linux文件名称的组成除由连续字母、标点符号、数字等构成外,中间不能有空格符、路径名称符号 / 或 # * % & {} [] ……等与Shell有关的特殊字符。

Linux文件系统中,结构上以root file system 位最上位也最为重要,所谓root file system乃于开机时将root partition挂载在/ 的目录,若无法mount / 则开机时无法进入Linux系统中此时仅能remount / 的目录。该目录下有/etc、/dev、/boot、/home、/lib、/lost+found、/mnt、/opt、/proc、/root、/bin、/sbin、/tmp、/var、/usr等重要目录,以下兹分别介绍之。

32.2 目录与文件简介

1. /etc:本目录下存放着许多系统所需的重要设定与管理文件,有一些为纯档名,有些是以.conf的型态出现另亦有一些自成单一目录:当然亦有些设定文件并非放在/etc目录下,例如使用者家目录之.bashrc、.bash_profile等文件;通常祇要更动过/etc目录下之设定档内容必须重新激活设定档使设定生效,且一般亦无须重开机。以下列出主要文件如后:

   - HOSTNAME

     本文件内容仅记载主机+网域名称,在系统提示符号下输入hostname可以显示HOSTNAME之文件内容。如：www.webrj.com、www.webrj.net等。

   - XF86Config

     本文件内容为X Window System的主要设定档,Caldera的版本放在/etc目录下,RedHat版本则放在/etc/X11目录下,有关显示卡、鼠标、键盘均可在此加以设定。

   - aliases

     本文件内容用来设定邮件别名及邮件清单可以让特定的地址转寄给不同的使用者或群组;编辑完/etc/aliases后须下newaliases指令使之生效，对于由远程寄送至local端的邮件具有备份功能，为mail server重要的设定档之一。

   - amd.conf

     本文件为一常驻服务程序(daemon)，全名为automatically mount file system，Caldera的版本称为am.d；是一个能自动mount和umount cdrom和floppy的文件系统。

   - at.deny

     本文件为对能否使用at指令的使用者加以限制, at 是一个非常有用的工具，可以让使用者指定在特定时刻执行某个程序或指令，通常只执行一次。如果你想定时定期的执行某项工作，应该使用cron工具而非at。

   - crontab

     本文件可让系统定时执行排程工作为系统管理上极为重要之文件，Cron是一个常驻程序(daemon)，在开机时激活cron的daemon时，它会自动去检查/var/spool/cron 目录下面看看是否有任

何cron文件。每一个user的可以去设定自己所要排定执行的工作。在这一个目录底下，每一个user会有一个属于他loginid名称的cron文件，crond会自动将这些user的cron文件加载至内存中，并定期去执行每个user的cron文件。另外，crond也会去读取/etc/crontab 的内容。

这是属于系统的cron工作设定文件，主要系统会定期去执行/etc/cron.d/ 目录下面四个cron.daily、cron.hourly、cron.weekly、cron.monthly目录下(RedHat版本则集中放在/etc目录下)的命令。

- dhcpd.conf

  本文件为DHCP Server的设定档,可烤贝/usr/doc/dhcp-serial number目录下的dhcpd.conf.sample至/etc目录下，该文件内容一般有记载着subnet、netmask、routers、domain-name、default-lease-time、max-lease-time及IP range；亦可一并指派固定IP给特定主机。

- dosemu.conf

  本文件为Dos仿真器的设定文件可以在Linux上仿真Dos环境，用Dos 指令。

- dumpdate

  dump指令可对Linux ext2文件系统进行检查备份(例如dumpe2fs /etc　对/etc目录作备份)，dumpdate则是存放dump指令的执行日期。

- exports

  本文件为NFS(Network File System)设定档，NFS主要是运用在UNIX 系统上，用来使UNIX系统能够在几部计算机间做文件的分享。其功能类似windows的网络磁盘驱动器，可以mount的方式，分享其它linux或UNIX主机的目录或文件。exports文件内则分别记载着分享目录、分享对象及权限等项目。

- fdprm

  本文件为软盘机参数表，含有各大小格式扇区磁道等资料。

- fstab

  本文件包含了开机时需加载的文件系统,每一行都表示一个文件系统,各字段分别有特定的装置或远程的文件系统、挂载点、加载的文件型态、挂载选项、dump设定及fsck设定。此文件可供Linux于开机时加载到系统上，并于关机时卸载。

- ftp*

  以ftp开头的文件如ftpaccess(主要设定档)、ftpconversions(文件的相关压缩规定)、ftpgroups、ftphosts、ftpusers(分别为ftp群

组、主机、使用者拒绝存取设定)等均为ftp server相关的设定档。

- group

  本文件为群组的资料文件，可以使每个使用者均拥有自己的群组；一般而言新的使用者建立后的的同时会产生相同于使用者名称的群组名称(RedHat Distribution)(Caldera Distribution 则会随adduser 或useradd 指令的不同而会纳入users群组或相同于使用者名称的群组名称)；在GNOME环境中可用LinuxConf设定，在KDE环境中可用User Manger加以设定之。

- host*

  在/etc目录下以host开头的文件有host.conf、hosts、hosts.allow、host.deny等目录，兹分述如下:

  - host.conf

    本文件设定网络搜寻顺序系依hosts或DNS之先后顺序定之;同时亦可一并设定是否将多重IP指定给一台主机与否。

  - hosts

    本文件设定主机的IP及网域名称，利用此文件可加速特定主机的搜寻速度无须借助DNS之功能，早期未有DNS前即是依赖hosts档作名称查寻。

  - hosts.allow

    本文件记载着允许那些主机联机到你的主机，在系统安全上为Tcpwrapper机制的存取控制文件。通常是维护主机安全或作测试用。

  - hosts.deny

    本文件记载着拒绝那些主机联机到你的主机，在系统安全上为Tcpwrapper机制的存取控制文件。通常是维护主机安全或作测试用。一般而言，常与hosts.allow檔原则上先deny ALL 再设定allow，被allow同意的存取便不会再被deny否决。

- httpd

  本目录下置放有conf目录、log 及module等文件其中尤以conf目录下的httpd.conf(主要设定档) 、access.conf(网页目录及资料来源路径文件)、srm.conf(设定apache的存取控制檔)，目前Apache 1.3.9以后版本已整合成httpd.conf加以设定即可；/etc/httpd为Apache Server重要目录所在。

- inetd.conf

  通常当系统激活时，有部份的service是并没有在开机时被激活的，以节省系统资源。他们是利用inetd(internet daemon)–来监

控网络服务的要求，再激活适当的daemon。inetd 是用来监控各种service的daemon，依不同的port提供不同的监控。/etc/inetd.conf即为其设定档，并由tcpd来提供监控。例如将telnet加上，系统便不提供telnet的服务。

本文件最常与Tcpwrapper机制的存取控制文件hosts.allow及hosts.deny搭配负责网络安全的监控；举ftp为例，当inetd接收到使用ftp的请求时，便会激活tcpd，tcpd先纪录这项ftp的请求，然后检查hosts.allow、host.deny这二个存取设定控制档，如果同意存取，就会激活。另外,Tcpwapper只能影响inetd激活，并且在inetd.conf里面经过编辑，由tcpd呼叫的服务，所以并不能为其它服务提供安全保障(如sendmail、NFS等)。管理者可以看log文件知道系统是否有被别人侵入。

- inittab

  一般Linux系统激活时，LILO 执行后加载kernel，kernel激活后呼init program(/sbin/init)激活系统必备程序，init为parent process(呼叫后fork 许多child process)，接着便检视/etc/inittab，视/etc/inittab以runlevel? 启动再执行/etc/rc.d/rc.?d/下所有s开头之shell script,直到完成(ex.s10network ; s40cron)是以inittab文件乃在规范使用者要以那一种runlevel登入Linux系统，我们亦可直接在系统提示符号下输入init 或telinit 后加上欲登入之第几层runlevel,例如init 3 或telinit 5，即可进入该环境模式。

- isapnp.gone

  本文件包含ISA适配卡所使用的资源,可用来设定硬件所需的内存、I/O Base、IRQ及DMA。

- issue(net)

  本文件是记载使用者在登入本机时所出现的一段文字讯息，例如[Red Hat Linux release 6.2 (Zoot) + CLE V0.9P1 (Yami) Kernel 2.2.17-4CLE on an i586]而issue.net则用于远程登入时之文字讯息显示之用。issue文件可加以修改成你需要出现的内容, 但需同时将/etc/rc.d/rc.local文件中`[echo “” > /etc/issue`至`echo >> /etc/issue]`等几行加上批注起来，始能生效。

- ldso.conf

  本文件存放了系统中共享函式库(shared libraries)的路径；Linux提供了两种形式函式库：shared及static，而Unix只提供shared libraries。当程序被编译时，程序便会去连结该程序所需用到的函式库。有时候程序为了便于侦错，或是为了某些考量，我们不希望程序去使用共享函式库，而是把会用到的链接库全部连结进程序的执行文件，让程序本身拥有一份函式库中函式的副本，这种方式称为静态连结程序(static linked)，而依赖共享函式库所建立的程序称动态连结程序(Dynamically Linked)。系统

安装了library后，必须告诉程序library放置于何处，使用共享函式库的好处是免除不同的程序里重复使用静态函数库(static libraries)，与其在每个呼叫这些函式的程序里都储存一份copy，不如把函式库集中在系统的文件里，让执行的程序都可以读到这个程序。

当编辑完ldso.conf文件增加新的函式库或新安装某个套件之后(可能会安装某个新的共享函式库)，后须下ldconfig -v，使之生效。

- lilo.conf

  lilo即linux loader，本文件lilo.conf内容可以分为两部分: Global及per-image，负责指定开机时所使用的kernel及开机时所要读取的lilo是放在哪一个装置，另亦有设定多重开机功能，使多个操作系统并存；如果对本文件进行修改，须下lilo －v －v －v 的指令重改设定由于lilo并非为一Daemon，仅为一内部程序，通常须重开机后设定始能生效。

- mediaprm

  本文件用来定义磁盘驱动器位于Linux下的代号ex. /dev/fd0、/dev/cdrom

- modules.conf

  本文件为模块的主要设定文件，在Red Hat 版本之名称命名为conf.modules ，Modules一般来说大部分为一些装置、网络、文件系统等的驱动程序，传统上，驱动程序是核心的一部份，因为几乎所有的核心都需要藉由成为核心码的一部份来取得使用硬件的能力，至于模块化程序则是一种可加载之式驱动程序(Loadable device driver)，在系统执行时是直接由加载内存或从内存卸载；而使用module的方式，可以不用重新建立kernel，同时也可以节省时间及内存空间。

- motd

  本文件为系统显示与user的提示讯息提示的时机则在user login并输入帐号及密码后出现之讯息，例如:[Last login: Mon Dec 18 10:19:15 on tty1]。

- mtab

  本文件记载着已挂载(mount)的文件系统，当你下mount指令时所显现之内容与mtab内容常相一致。

- ntp.conf

  NTP全名为Network Time Protocol(网络对时协议)，系client端针对NTP Server作同步化对时所使用之协议，为达到NTP同步

化获致正确而可信赖时间。ntp.conf则为主要设定文件提供模式的选择和使用的servers设定。

- pam.d

  PAM（Pluggable Authentication Modules）即可插拔的认证模块；当login时，必需提供username和password，然后系统根据所给予的username和password来验证可否login，确认使用者身份，PAM允许设置多种认证方式，不须再重新编译核心要进行认证的程序。使用PAM，可编辑配置文件动态的去读取配置模块，然后再去执行验证；pam.d目录下则放置与PAM相关的文件。

- passwd

  每一位使用者皆有使用者名称供以辨识身份，亦必须一并设定密码在/etc/passwd文件中可以root身份来加入新的使用者，通常user输入username及密码，系统会先将输入之密码加密成13位码，再与/etc/passwd作一比对，无误与否。至于实际的密码则放在/etc/shadow文件中。

- protocols

  通讯协议号码是IP数据段表头中的一个字节，藉以显示出数据应该传给IP层以上的何种协议；至于协议号码则规范在本文件中，protocols内容是一个对照表，包含协议名称及协议号码和批注等栏项，利用协议号码与协议名称的对照便可知使用何种协议。

- rc.d

  本目录下以etc/rc.d/rc0.d ~ `/etc/rc.d/rc6.d`里的文件代表各种不同runlevel所激活的daemon尤为重要，K代表kill，S代表start。在渠等目录下的文件通常为一连结档link至/etc/rc.d/init.d里的daemon以供激活。/etc/inittab文件乃在规范使用者要以那一种runlevel登入Linux系统，至于本文件案则视以runlevel? 登入，代表激活rc?.d目录下依序激活各连结档的shell script(/etc/rc.d/init.d 里的daemon)。

- resolv.conf

  BIND的伺服端透过daemon(/etc/rc.d/init.d/named)来执行；BIND的客户端则透过resolver，它是一个程序函式库组成，会发出名称查询，而由/etc/resolv.conf设定之(即将nameserver加入该文件中)，例如`echo “nameserver 172.17.0.10” >> /etc`/resolv.conf 或者手动vi编辑之。

- rpc

rpc全名为remote procedure call即远程程序呼叫，通常系指local端程序对位于远程系统中的程序进行呼叫，俟其完成任务并将该任务传回给local端；/etc/rpc文件内存rpc程序号码数据库，含有使用者可识别之替代rpc程序号码的名称基本格式则含有rpc程序的server名称、rpc程序号码、别名等记载栏项。

- samba.d/smb.conf:

  Samba(Server Message Block protocol)使用来将linux system与windows system透过网络上的芳邻作沟通整合之用，而/etc/samba.d/smb.conf则为samba的重要设定档，该档中主要分global与share defini tions二大项；/etc/samba.d/ smbpasswd则是samba password的设定档。

- securetty

  本文件定义了root可login的terminal，一般而言预设为tty1至tty8，root只能从定义中的terminal登入。

- sendmail.cf

  本文件为mail server的重要设定档，sendmial是一种MTA，目前在unix或linux上使用相当广泛，由柏克莱大学大学生所写的程序，功能强大，但sendmail.cf内容由宏指令写成较为艰涩难懂，因此一般除伪装名称、主机的别名、收信的主机位置名称等少数项目由sendmail.cf文件直接于其中设定外，一般而言，均另用m4产生sendmail.cf的宏定义档，RedHat版本名称为/etc/sendmail.mc，OpenLinux版本则放在/usr/share/sendmail/cf/cf/gerneric-col2.2.mc；此外/etc/aliases、/etc/access、/etc/mail/relay-domains亦为sendmail的相关设定檔。

- services

  一般而言Linux或Unix的操作系统port号码常定义在/etc/services文件中由port号码可辨示出application process(应用程序行程)即网络服务例如ftp、telnet；在services文件中256以下port号码保留给常用的网络服务，256到1024的port号码则由特定的Linux服务使用，另不同的Transport Layer(传输层)可以共享相同的port号码；port号码必须与协议号码共同搭配始能将资料导引到正确的网络服务。

- shadow

  Shadow passwording将密码移到另一个文件，本文件即属密码文件而与/etc/passwd搭配；权限上则较为严格，一般而言，仅有root有读写(Caldera distribution)或读(RedHat distribution)的权利。

- skel

本目录下之文件内容例如:.bash_logout、.bash_profile、.bashrc、.kde、.kderc、.netscape、.screenrc、Destop等即是用来在建立一新的使用者时在/home/目录下的使用者名称目录的内容即烤贝至skel此目录下之文件内容。

- sysconfig/network-sccripts/ifcfg-eth?

  本文件为网络卡的设定档内容有设备名称、IP地址、屏蔽、广播地址、网段、开机激活设定等项目。

- wgetrc

  wget乃使用来将World wide web上之文件取回之工具指令，wgetrc文件则是wget初始化的文件其中有quota、mail header、重传文件的预设次数、firewall及proxy的相关设定。

- z*

  以下z开头的文件均为zshell相关的设定档zlogin系指z shell的登入设定；zlogout系指z shell的注销设定；zprofile系指z shell的使用者设定；zshenv系指z shell的使用环境设定；zshrc系指z shell的资源档设定。

2. dev:

   本目录中存放了device file(装置文件)，使用者可以经由核心用来存取系统中之硬设备，当使用装置文件时核心会辨识出I/O Request传递到相对应装置的驱动程序以便完成特定的动作；每个装置在/dev目录下均有一个相对应的项目；另/dev目录下尚有一些项目是没有的装置这通常是安装系统时所建立的，它不一定对应到实体的硬件装置；此外亦有一些虚拟的装置，不对应到任何实体装置，例如俗称黑洞装置的/dev/null，任何写入该档的请求均会被执行但被写入的资料均会如进入黑洞般的消失无踪。

   在/dev目录下第一栏的文件型态会发现常存有既非directory亦非file而是“b”或”c”，”b”指block device file(区块装置文件)，”c”指character device(字符装置文件)；区块装置(例如硬盘)通常是外接装置，资料的读写都是以整个区块的形式进行读写，字符装置(例如串行端口)，资料的读写都是以一个byte来进行读写。

   (a) /dev/fd*

   fd0、fd1———等指第几个软盘支持的装置驱动接口，fd0表示第一个fd1表示第二个。

   (b) /dev/hd

   系指IDE硬盘的装置驱动接口，在/dev/lilo.conf设定中boot=/dev/hda即指整颗硬盘，/dev/hda1则指硬盘中的第一个partition。

(c) /dev/sd

系指SCSI磁盘的装置驱动接口。

(d) /dev/console

系指系统的操作控制台乃实际连接到Linux操作系统的屏幕。

(e) /dev/tty

系指提供使用者不同的terminal操作控制台的装置驱动接口，并有virtual console的功能切换上可使用ctrl+alt+F1~F6。

(f) /dev/ttyS?

ttyS系指串行端口接口ttyS0即为COM1，ttyS1即为COM2

3. /boot:

本目录下放置有系统激活的相关文件例如initrd.img、vmlinuz、System.map，均为重要的文件，是以本目录不可任意删除。

initrd.img为系统激活时最先加载的文件。

Vmlinuz即为kernel的image文件。

System.map包括了kernel的功能及位置。top、ps指令会去读此文件来显示系统目前的信息状态。因此System.map必须对应到相同的kernel，不然会显示错误的信息。

4. /home:

一般而言，使用者的家目录就($HOME)是放在/home这个目录下，而以使用者名称作为/home目录下各个目录的名称例如使用者col的家目录路径即为/home/col目录下当使用者col login时，其所在的目录即为/home/col，此外ftp站台目录(ftp)及web站台目录(httpd)亦置于/home目录下。

5. /lib:

许多系统激活时所需要用到重要的共享函式库shared libraries均放于此，包含最重要的GNU C library在内，凡档名为library.so.version的共享函式库，通常放在/lib目录下。

/usr/lib/:

本目录与/lib目录不同的是/lib乃系统激活时所需要用到重要的共享函式库而/usr/lib乃关于应用软件常置放函式库之处；例如放置一些其它应用程序(如Netscape、X server)等的share libraries。其中，最重要的函式库为libc或glibc (glibc 2.x便是libc 6.x版本，标准C语言函式库)。几乎所有程序都会用到libc或glibc，因为这两个程序提供了对于Linux kernel的标准接口。又档名为library.a的静态函式库，通常亦放在/usr/lib目录下。

6. /lost+found:

   一般重启或关闭system，可下sync;sync.避免有些message会留在硬盘之cache上，此时dirty bit为1，当再开机时，system会去检查每一个dirtybit是否为0，如果为1则会执行fsck。作fsck时，常会问要否删除dirtybit，如选yes时，会把inode集中放在lost+found用file指令去查寻，不重要再行删除即可（inode number）。简单言之，本目录乃记录硬盘上的partition于资料流失时作fsck寻找回来的遗失文件片段。

7. /mnt:

   本目录为系统内定的mount point(挂载点)，预设则有/mnt/cdrom和/mnt/floppy，使用自动的挂载程序例如KDE桌面上的cdrom与floppy或者GNOME的Drive Mount Applet，均会自动地把光驱和软盘机挂载至这二个目录。如果要挂加载额外的文件系统，一般而言我们都会将/mnt目录当作挂载点，然后在该目录下建立任一目录名称作为挂载目录；家目录把同一颗硬盘上的另一个Window 2000操作系统mount到本机Linux操作系统上则可用此法。

8. /proc:

   本目录为一个虚拟的文件系统，其功能乃在统一文件与行程，它不占用任何硬盘空间，因为该目录下的文件均放置于内存内；每当你存取/proc文件系统时kernel会拦截你的存取动作撷取相关信息再动态的产生目录与文件内容。

   本目录下主要有二大形态的文件；其一是以PID数字为文件名，/proc会记录系统每个行程，有助于系统的管理于与除错；其二乃kernel所搜集到的系统使用的硬件信息例如/proc/ioports、/proc/dma、/proc/meminfo、/proc/interrupts均属之。

9. /root:

   本目录为系统管理者root的家目录。

10. /bin:

    bin为binary的简写主要放置一些系统的必备执行文件例如:cat、cp、chmod df、dmesg、gzip、kill、ls、mkdir、more、mount、rm、su、tar等。

    /usr/bin:

    主要放置一些应用软件工具的必备执行文件例如c++、g++、gcc、chdrv、diff、dig、du、eject、elm、free、gnome*、gzip、htpasswd、kfm、ktop、last、less、locale、m4、make、man、mcopy、ncftp、newaliases、nslookup passwd、quota、smb*、wget等。

    /sbin:

主要放置一些系统管理的必备程序例如:cfdisk、dhcpcd、dump、e2fsck、fdisk、halt、ifconfig、ifup、ifdown、init、insmod、lilo、lsmod、mke2fs、modprobe、quotacheck、reboot、rmmod、runlevel、shutdown等。

/usr/sbin:

放置一些网络管理的必备程序例如:dhcpd、httpd、imap、in.*d、inetd、lpd、named、netconfig、nmbd、samba、sendmail、squid、swap、tcpd、tcpdump等。

11. /tmp:

    本目录乃供全体使用者暂时放文件的目录，有时某些应用程序执行中产生的临时文件亦会暂放至此目录；然而系统预设本目录权限为可读、写、执行但无法删除之1777(drwxrwxrwt)亦即多加上save program text on swap device即t 的权限；以避免有人任意删除他人存放于/tmp目录的文件。

12. /var:

    Linux操作系统经常需要变动的或暂存的资料常放在固定的目录而后系统新产生的资料都会在这个文件中作更新；在这个目录下有几个重要的目录例如:/var/log；/var/spool；/var/run等以下兹分别介绍之:

    (a) /var/log

        log文件乃记载着Linux操作系统运作过程的记录分别有激活纪录(boot)、网站存取、错误、安全记录(httpd/access_log、httpd/error_log、httpd/ssl*)、邮件记录(maillog)、登入讯息(secure)、FTP讯息(xferlog)。

    (b) /var/spool

        spool乃Linux操作系统将工作暂放置于内存特定区域或硬盘上特定设备，/var/spool是一个队列目录提供spool服务，其底下最为重要的目录即为lpd、mail、squid等三个目录；打印文件的print queue会在/var/spool/lpd。(完)

33 linux下刻盘四部曲

1. 编译内核(依系统情况可以略过).
2. 加载相应驱动模块.
3. 制作iso9660文件系统的映象文件
4. 刻盘

缺省情况下,你的linux内核或许不支持SCSI设备,而现在的刻录机却许多接在一个scsi卡上的.因此,在这种情况下,需要从新编译内核,让它支持SCSI设备,并且要根据SCSI卡所采用的芯片,将其编译到内核或编译成模块.此外,为方便测试将要刻录的映象文件,还应该编译时在块设备里面选择支持"LOOPBACK DEVICE SUPPORT".最后,还要选择"SCSI generic support",将其编译成模块,这专门是为支持这种连接在一块SCSI卡上的刻录机刻录时用的.

然后,重新启动系统,启动过程中会检测SCSI设备,因为检测的特别快,可能看不清. 没有关系,登录后用"dmesg"命令来看系统查到了什么东西,比如屏幕上打印了这么一块信息:

```
sym53c8xx: at PCI bus 0, device 9, function 0
sym53c8xx: setting PCI_COMMAND_PARITY...(fix-up)

sym53c8xx: 53c810a detected
sym53c810a-0: rev=0x23, base=0xe6000000, io_port=0xe400, i-
              rq=11
sym53c810a-0: ID 7, Fast-10, Parity Checking
sym53c810a-0: restart (scsi reset).
scsi0 : sym53c8xx - version 1.3c
scsi : 1 host.
Vendor: MATSHITA Model: CD-R CW-7502 Rev: 4.17
Type: CD-ROM ANSI SCSI revision: 02
```

当然了,这是在我的系统启动时打印的信息,如果你没有看到类似的信息,说明你没有把SCSI卡所用芯片编译到内核或在系统启动时没有把自动加载相应的驱动模块(通过配制/etc/conf.modules在启动时加载相应模块).这时候,你可以手工加载模块.当模块加载到系统中后,就应该出现以上类似的信息了.在我用的SCSI卡上采用的芯片BIOS是"SYMBIOS 53C810AE",因此在编译内核时选择了"SYM53C8XX SCSI SUPPORT",并把它编译成模块.下面简单介绍一下以上信息:

首先它显示在PCI插槽上发现了SCSI卡,紧接着,发现了SCSI卡上连接的设备,它的ID号是7,在scsi0上.

当找到设备后,剩下的事情就简单了,现在需要找的就是一些制作ISO映象程序刻录时用的软件.我用的软件是cdrecord,软件很不错,本身可以在SCSI卡上检测所连接设备,也支持multi-session刻录(允许你一次刻不完下次再刻).制作ISO映象文件用的是mkisofs这个程序,一般的linux都带这个包,在redhat中包的名字和文件的名字一样.

假如我有一个整理好待刻的目录,里面有一堆很好的收藏.我首先把用下面的命令作成一个ISO格式的映象文件(这并不是刻录,只是在硬盘上按1:1的比例将要刻的东西作并不是刻录,只是在硬盘上按1:1的比例将要刻的东西作成一个映象):

```
mkisofs -r -o cd_image my_private/
```

"-r"指明将所有的文件属性变成对所有人是可读属性的,"-o"是输出, my_private是要刻东西所在的目录.

作完映象文件后,可以检测它是否正确,此时要加载曾经编译的loop模块,将此映象文件作为一个文件系统"mount"上来.

```
mount -t iso9660 -o ro,loop=/dev/loop0 cd_image /mnt/cdrom
```

进入/mnt/cdrom就可以查看文件是否正确.主意的是,用mkisofs可以制作带光盘启动的映象,这要通过"-b"参数,"-b"后面紧接着的是可以启动的内核文件,它的大小是有限制的,必须是1.2或1.44或2.88MB,它的路径也是相对于my_private目录而言的.

最后一步,"天下无盘",你可别刻费了吆.:-) 现在就开始真正动手了.用下载来的刻录软件进行刻录.

这里只对cdrecord进行简略介绍,还有一些图形界面的刻录工具,各位就自己找吧.关于cdrecord可以在http://hkt.linuxberg.com这个网站找到.用

```
cdrecord -scanbus
```

可以显示出下列类似信息:

```
Cdrecord release 1.6.1 Copyright (C) 1995-1998 J鯢g Schill-
ing  scsibus0:
0) *
1) *
2) *
3) *
3) *
4) *
5) *
6) 'MATSHITA' 'CD-R CW-7502 ' '4.17' Removable CD-ROM
```

ID号为7的一个刻录机找到,下面开始了.

```
cdrecord -v speed=2 dev=6,0 cd_image
```

"speed=2"指明刻盘速度为2倍速, "dev=6,0"指明关于刻录设备的一些信息,完整的形式是"dev=devicename:scsibus,target,lun",我们这里只有一个接在SCSI卡上的设备, 因此只些了简略形式"dev=target,lun",其中devicename是CD-R所对应的设备名,SCSIBUS是SCSI总线号,TARGET就是刚才检测出来的第六项,LUN指的是SCSI设备的逻辑单元号,一般的SCSI设备只支持一个LUN.

34 如何更改X-Windows的刷新频率

如果你用的是Redhat的话你的X window配置文件是/etc/X11/XF86Config。Slackware的X Window配置文件是/etc/XF86Config。

在那个文件中你会看到象下面这样的东西：

```
# 640x400 @ 70 Hz, 31.5 kHz hsync
Modeline "640x400" 25.175 640 664 760 800 400 409 411 450
# 640x480 @ 60 Hz, 31.5 kHz hsync
Modeline "640x480" 25.175 640 664 760 800 480 491 493 525
# 800x600 @ 56 Hz, 35.15 kHz hsync
ModeLine "800x600" 36 800 824 896 1024 600 601 603 625
# 1024x768 @ 87 Hz interlaced, 35.5 kHz hsync
Modeline "1024x768" 44.9 1024 1048 1208 1264 768 776 784 817
Interlace

# 640x480 @ 72 Hz, 36.5 kHz hsync
Modeline "640x480" 31.5 640 680 720 864 480 488 491 521
# 800x600 @ 60 Hz, 37.8 kHz hsync
Modeline "800x600" 40 800 840 968 1056 600 601 605 628
+hsync +vsync

# 800x600 @ 72 Hz, 48.0 kHz hsync
Modeline "800x600" 50 800 856 976 1040 600 637 643 666
+hsync +vsync
# 1024x768 @ 60 Hz, 48.4 kHz hsync
Modeline "1024x768" 65 1024 1032 1176 1344 768 771 777 806
-hsync -vsync

# 1024x768 @ 70 Hz, 56.5 kHz hsync
Modeline "1024x768" 75 1024 1048 1184 1328 768 771 777 806
-hsync -vsync
# 1280x1024 @ 87 Hz interlaced, 51 kHz hsync
Modeline "1280x1024" 80 1280 1296 1512 1568 1024 1025 1037
 1165
Interlace

# 1024x768 @ 76 Hz, 62.5 kHz hsync
Modeline "1024x768" 85 1024 1032 1152 1360 768 784 787 823
# 1280x1024 @ 61 Hz, 64.2 kHz hsync
Modeline "1280x1024" 110 1280 1328 1512 1712 1024 1025 102-
8 1054

# 1280x1024 @ 74 Hz, 78.85 kHz hsync
Modeline "1280x1024" 135 1280 1312 1456 1712 1024 1027 103-
0 1064

# 1280x1024 @ 76 Hz, 81.13 kHz hsync
Modeline "1280x1024" 135 1280 1312 1416 1664 1024 1027 103-
0 1064
```

这些东西控制这你的显示卡的设置，例如下面这一行注释说分辨率为1280x1024，刷新速率为76赫兹，行扫频率为81.13千赫兹

```
# 1280x1024 @ 76 Hz, 81.13 kHz hsync
```

下面这一行具体设置显示卡：

```
Modeline "1280x1024" 135 1280 1312 1416 1664 1024 1027 103-
0 1064
```

这行中每个项目的意义如下：（从左到右）
模式行，分辨率，像元频率（兆赫兹），每行像元数，行同步（消隐）脉冲开始的时钟周期，行同步（消隐）脉冲结束的时钟周期，每行的时钟周期数，每帧的图象行数，帧同步脉冲开始的扫描行数，帧同步脉冲结束的扫描行数，每帧的扫描行数。

调整这些数目你可以最大限度地使用你的显示卡和显示器，例如您的显示卡有一兆存储器，您可以设置成1152x900的分辨率。:-)需要当心的是仔细核对你的显示器所允许的扫描频率，有些显示器当收到过高的扫描同步脉冲后会烧毁行扫描晶体管。我就烧过一个显示器，那晶体管还不太好买，彩电的行扫描晶体管通常不能用，频率太低，功率也不够大，上去就烧。:-(

下一个问题是你的显示卡有什么像元频率可以用，例如上面的行要求135MHz，但你的卡只有125兆赫兹，怎么办呢？也好办，用emacs或vi把135改为125就完了。

如果你的配置文件中对应于您所要的分辨率有多个模式行的话，您可以将其它频率较低的模式行用号给封上，只留下你想要的频率就能改变扫描频率了。

祝你成功！烧显示器我可不能负责！ :-)

35 Grub多重启动管理器

35.1 什么是grub

grub 是一个多重启动管理器。grub是GRand Unified Bootloader的缩写，它可以在多个操作系统共存时选择引导哪个系统。它可以引导的操作系统包括Linux,FreeBSD,Solaris,NetBSD,BeOSi,OS/2,Windows95/98,Windows NT,Windows2000。它可以载入操作系统的内核和初始化操作系统（如Linux,FreeBSD），或者把引导权交给操作系统（如Windows 98）来完成引导。

35.2 grub的特点

grub可以代替lilo来完成对Linux的引导，特别适用于linux与其它操作系统共存情况，与lilo相比，它有以下特点：

- 支持大硬盘

  现在大多数Linux发行版本的lilo都有同样的一个问题：根分区(/boot分区)不能分在超过1024柱面的地方，一般是在8.4G左右的地方，否则lilo不能安装，或者安装后不能正确引导系统。而grub就不会出现这种情况，只要安装时你的大硬盘是在LBA模式下，grub就可以引导根分区在8G以外的操作系统。

- 支持开机画面

  grub支持在引导开机的同时显示一个开机画面。对于玩家来说，这样可以制作自己的个性化开机画面；对于PC厂商，这样可以在开机时显示电脑的一些信息和厂商的标志等。grub支持640x480,800x600,1024x768各种模式的开机画面，而且可以自动侦测选择最佳模式，与Windows那320x400的开机画面不可同日而语。

- 两种执行模式

  grub不但可以通过配置文件进行例行的引导，还可以在选择引导前动态改变引导时的参数，还可以动态加载各种设备。例如你在Linux下编译了一个新的核心，但不能确定它能不能工作，你就可以在引导时动态改变grub的参数，尝试装载这个新的核心进行使用。Grub的命令行有非常强大的功能，而且支持如bash或doskey一样的历史功能，你可以用上下键来寻找以前的命令。

- 菜单式选择

  在lilo下，你需要手工输入操作系统的名字来引导不同的操作系统。而grub使用一个菜单来选择不同的系统进行引导。你还可以自己配置各种参数，如延迟时间，默认操作系统等。

- 分区位置改变后不必重新配置

  lilo是通过读取硬盘上的绝对扇区来装入操作系统，因此每次分区改变都必须重新配置lilo，例如你用PQ magic调整了分区的大小，那lilo在你重新配置好之前就不能引导这个分区的操作系统了。而grub是通过文件系统直接把核心读取到内存，因此只要操作系统核心的路径没有改变，grub就可以引导系统。除此之外，Grub还有许多非常强大的功能。例如支持多种外部设备，动态装载操作系统内核，甚至可以通过网络装载操作系统核心。Grub支持多种文件系统，支持多种可执行文件格式，支持自动解压，可以引导不支持多重引导的操作系统等。

35.3 grub的使用

安装grub

如果已经安装了蓝点Linux2.0则grub是默认安装的。要把grub重新安装到主引导扇区上，只需要简单打入makebootable命令就可以了。

制作grub启动盘

首先确定grub已经安装，然后进入grub的目录，键入：

```
#cd /boot/grub
```
放入一张软盘，然后敲入命令：
```
#dd if=stage1 of=/dev/fd0 bs=512 count=1
#dd if=stage2 of=/dev/fd0 bs=512 seek=1
```
这样就可以做好一张启动盘了。也可以用mkbootdisk命令
```
#mkbootdisk 2.2.16
```
2.2.16是指内核版本号
开机

安装了grub开机后会出现一个菜单，列出所有的启动选项。如果设置了启动画面则会显示启动画面，按Esc键则可以取消启动画面显示菜单选项。蓝点Linux所带的grub的命令提示是全中文的，在菜单下面详细列出如按e是编辑启动命令，按c是使用命令行等。用上下键可以选择菜单项，按回车启动所选项。按e键可以编辑所选项的启动命令，你可以用这个功能临时改变你的系统的启动参数，参见配置grub一节。按c键则进入命令行模式。　　在命令行模式下可以打入命令直接执行，例如你可以敲入poweroff关闭计算机。按Tab键可以列出所有支持的命令。蓝点Linux已经把grub汉化了，其中一部分命令敲入名字后会给出中文提示，显示命令的用法和参数。

35.4 配置grub

grub启动时会在/boot/grub/中寻找一个名字为menu.lst的配置文件，如果找不到此文件则不进入菜单模式而直接进入命令行模式。

menu.lst 是一个文本文件，你可以用任何一个文本编辑器来打开它。每一行代表一个配置命令，如果一行的第一个字符为井号"#"则这一行为注释，你可以简单地用增加或减少注释行来改变配置。

编辑menu.lst，一般会有以下各行
```
timeout second
```
设定在second秒之后引导默认的操作系统。

蓝点Linux默认是timeout 5，就是5秒没有其他指令就引导系统，如果设成–1，则grub会一直等待直到用户选择一个选项为止。
```
default num
```
默认启动第num+1行选项，也就说default=0则默认启动菜单第一行的操作系统，default=1则启动第2行的系统，如此类推。
```
splash pathname/filename
```
指出开机画面的文件所存放的路径和文件名，如splash /boot/logo/800x600x8.img是指用在/boot/logo路径下的800x600.img文件作为开机画面
```
title OSname title
```

后面的字符就是你在菜单项上所看见的选项，你可以写上操作系统的名字和描述，如用title BluePoint Linux, Single Mode 代表这一选项是引导蓝点Linux的单用户模式。

下面结合两个系统引导描述来解释几个引导选项的意义

```
title BluePoint Linux, Default Mode
root (hd0,1)
kernel /boot/vmlinuz vga=auto root=/dev/hda2
```

hd0是指第一个硬盘(主硬盘) (hd0,1)是指第一个硬盘的第二个分区。kernel /boot/vmlinuz 是指出Linux核心的路径在/boot/vmlinuz中。vga=auto 是设定显示模式，root=/dev/hda2是指把第一个硬盘的第二个分区作为根挂载点（"/"）。

```
title Microsoft Windows
root (hd1,0)
chainloader (hd1,0)+1
```

root (hd1,0)这是指第二个硬盘(从硬盘)上第一个分区。chainloader (hd1,0)+1装入一个扇区的数据然后把引导权交给它。

35.5 从软盘启动grub

制作启动盘后可以用软盘启动引导硬盘上的操作系统插入制作好的启动软盘，进入BIOS设定软盘启动。软盘启动成功后就会进入grub的命令行模式

```
grub>
```

要启动一个操作系统，首先指定引导哪个分区上的系统，例如要引导指第一个硬盘上的第一个分区的操作系统，先键入

```
grub>root (hd0,0)
```

接着如果要启动的是Windows系统，键入

```
grub>chainloader (hd0,0)+1
```

注意(hd0,0)要随着硬盘和分区的不同而改变数字。如果要引导Linux或其他系统，应键入

```
grub>kernel （hd0,0）/boot/vmlinuz root=/dev/hda1
```

注意hda1参数也要随着硬盘和分区的不同而改变，如从第二个硬盘的第一个分区引导则用hdb1。

最后敲入boot就可以启动系统了。

在任何时候不能确定命令或者命令的参数都可以按Tab获得相关的帮助。用上下键可以获得命令的历史记录。其实这些命令就是menu.lst的启动描述，您也可以根据那些描述来自己键入启动命令，最后敲入boot就可以引导系统了。

36 GRUB is great

一直想找个FreeBSD bootloader那样的启动程序, 但是因为懒, 所以也没把lilo试着换成别的. 前两天才知道grub很象FreeBSD的bootloader, 有自己的shell, 并且可以认识文件系统(lilo就不行), 而且还支持netboot.所以试了一把, 感觉很不错. 以后升级内核的时候再也不用胆战心惊了.(前两天给一个带scsi硬盘的服务器升级内核, 结果lilo起不来, 搞得我要拆机箱最后在lilo.conf里面加上disk=/dev/sda以及bios=0x80才搞定.)

唯一美中不足的是, grub的password机制不够灵活: 要么整个menuitem都需要password来unlock, 要么就可以直接修改grub的启动命令, 不受任何限制. 而lilo可以只在输入kernel参数时要求密码.

36.1 安装简要

1. ./configure 可以加上一些参数, 如网卡驱动模块
2. make install
3. mkdir /boot/grub cp /usr/share/grub/i386-pc/* /boot/grub
4. grub-install your-boot-device (eg. /dev/hda2)

如果需要menu的话, 要编辑/boot/grub/menu.lst重起就行了

36.2 basic steps to boot linux on grub commandline

```
grub> root (hd0,1)
```

(hd0,1)即/dev/hda2, 注意所有编号都从0开始。指定kernel 所在分区

```
grub> kernel /boot/vmlinuz
```

/boot/vmlinuz为你的kernel, 可以使用TAB键来列出文件名

```
grub> boot
```

36.3 basic steps to boot DOS/Windows

```
grub> rootnoverify (hd0,0)
```

(hd0,0)就是通常的C:

```
grub> chainloader +1
grub> boot
```

在文档里面还用了makeactive来激活DOS分区, 但如果grub不在mbr上, 下次就进不了grub了

37 xinetd使用指南

redhat7.0以后inetd都已换成xinetd了—好用了很多。先看一下/etc/xinetd.conf。instances指定最大实例数—如果你用的是wu-ftpd,想控制最大人数为60,那就改这个了。注意最后一句includedir其实是把telnet,wu-ftpd等脚本包含到此文件中的(直接写在这也行)。

```
defaults
{
instances = 60
log_type = SYSLOG authpriv
log_on_success = HOST PID
log_on_failure = HOST
cps = 25 30
}
includedir /etc/xinetd.d
```

这篇xinetd完全指南已非常全了—看了它你对xinetd就会很熟了：）

许多人在装了redhat 7.x后开始找不到北(我就是其中一个)!!!因为redhat 7.x开始注重系统安全,最大的特征就是用xinetd.conf代替原来的inetd.conf。并且7.1中默认安装没有开ftp,telnet等熟悉的服务,而是更安全的ssh!7.1还加入firewall等服务(感谢paradise提供下载地点给我安装redhat7.1)。

大家对被称作超级服务器的Inetd一定很熟悉，其实现控制对主机网络连接。当一个请求到达由Inetd管理的服务端口，Inetd将该请求转发给名为tcpd的程序。Tcpd根据配置文件hosts.allow和hosts.deny来判断是否允许服务该请求。如果请求被允许则相应的服务器程序(如：ftpd、telnetd)将被启动。这个机制也被称作`tcp_wrapper`。

xinetd(eXtended InterNET services daemon)提供类似于inetd+`tcp_wrapper`的功能，但是更加强大和安全。它能提供以下特色：

- 支持tcp、ucp、RPC服务(但是当前对RPC的支持不够稳定)
- 基于时间段的访问控制
- 功能完备的log功能，即可以记录连接成功也可以记录连接失败的行为
- 能有效的防止DoS攻击(Denial of Services)
- 能限制同时运行的同一类型的服务器数目
- 能限制启动的所有服务器数目
- 能限制log文件大小
- 将某个服务绑定在特定的系统接口上，从而能实现只允许私有网络访问某项服务

- 能实现作为其他系统的代理。如果和ip伪装结合可以实现对内部私有网络的访问

它最大的缺点是对RPC支持的不稳定性，但是可以启动protmap，与xinetd共存来解决这个问题。

1.编译安装

可以从www.xinetd.org下载xinetd，当前最新的版本是xinetd 2.1.8.8p3。默认编译和安装xinetd是非常简单的，按照如下的步骤进行即可完成：

```
#./configure; make; make install
```

在进行configure时，可以支持如下几个有用处的选项：

`--with-libwrap` 如果使用该选项xinetd将会察看tcpd配置文件(/etc/hosts.allow和/etc/host.deny)来进行访问控制，但是如果要利用该功能，系统上必须安装有`tcp_wrapper`和相关库。

`--with-loadavg` 使用该选项，xinetd将而已处理max-load配置选项。从而在系统负载过重时关闭某些服务进程，来实现某些DoS攻击。

`--with-inet6` 使用该选项xinetd将支持IPv6。

如果是是用redhat7.0，则其默认将安装xinetd，而不需要自行安装。

2.配置

xinetd的默认配置文件是/etc/xinetd.conf。其语法和/etc/inetd.conf完全不同且不兼容。它本质上是/etc/inetd.conf和/etc/hosts.allow，/etc/hosts.deny功能的组合。/etc/xinetd.conf中的每一项具有下列形式:

```
service service-name
{
        ......
}
```

其中service是必需的关键字，且属性表必须用大括号括起来。每一项都定义了由service-name定义的服务。

Service-name是任意的，但通常是标准网络服务名，也可增加其他非标准的服务，只要它们能通过网络请求激活，包括localhost自身发出的网络请求。有很多可以使用的attribute，在下表中进行了详细的说明。稍后将描述必需的属性和属性的使用规则。

操作符可以是=，+=，或－=。所有属性可以使用=，其作用是分配一个或多个值，某些属性可以使用+=或－=的形式，其作用分别是将其值增加到某个现存的值表中，或将其值从现存值表中删除。表10.10中说明了可以用后一种形式的属性。

Value是为给定属性设置的参数。

扩展的lnernet服务进程属性

属　性
描述和允许值

`Socket_type`
使用的TCP/IP socket类型，值可能为stream(TCP)，dgram(UDP)，raw和seqpacket(可靠的有序数据报)

protocol
指定该服务使用的协议，其值必须是在/etc/protocols中定义的。如果不指定，使用该项服务的缺省协议。

Server
要激活的进程，必须指定完整路径

`Server_args`
指定传送给该进程的参数，但是不包括服务程序名

Port
定义该项服务相关的端口号。如果该服务在/etc/services中列出，它们必须匹配

Wait
这个属性有两个可能的值。如果是yes，那么xinetd会启动请的进程并停止处理该项服务的请求直到该进程终止。这是个单线程服务。如果是no，那xinetd会为每个请求启动的一个进程，而不管先前启动的进程的状态。这是个多线程服务

User
设置服务进程的UID，但是若xinetd的有效UID不是0，该属性无效

Group
设置进程的GID。若xinetd的有效UID不是0，这个属性无效

Nice
指定进程的nice值

Id
该属性被用来唯一地指定一项服务。因为有些服务的区别仅仅在于使用不同的协议，因此需要使用该属性加以区别。默认情况下服务id和服务名相同。如echo同时支持dgram和streama服务。设置`id=echo_dgram`和id=echo-`_streams`来分别唯一标识两个服务

Type
可以是下列一个或多个值：RPC（对RPC服务），INTERNAL(由xinetd自身提供的服务，如echo)，UNLISTED(没有列在标准系统文件如/etc/rpc或/etc/service中的服务)

`Access_time`
设置服务可用时的时间间隔。格式是`hh:mm_hh:mm`; 如08：00-18：00意味着从8A.M到6P.M.可使用这项服务

Banner
无论该连接是否被允许，当建立连接时就将该文件显示给客户机

Flags
可以是以下一个或多个选项的任意组合：

> REUSE:设置TCP/IP socket可重用。也就是在该服务socket中设置`SO_REUSEADDR`标志。当中断并重新启动xinetd
> INTERCEPT:截获数据报进行访问检查，以确定它是来自于允许进行连接的位置。不能和INTERNAL服务和多线程服务不可使用该属性值
> NORETRY:如果fork失败，不重试
> IDONLY: 只有在远程端识别远程用户时才接受该连接(也就是远程系统必须运行ident服务器)，该标记只适用于面向连接的服务。若没有使用USERID记录选项则该标记无效`log_on_success`和/或`log_on_failure`属性设置USERID值以使该值生效。仅用于多线程的流服务
> NAMEINARGS:允许`server_args`属性中的第一个参数是进程的完全合格路径，以允许使用`TCP_Wrappers`
> NODELAY:若服务为tcp服务，并且NODELAY标记被设置，则`TCP_NODELAY`标记将被设置。若服务不是tcp服务则该标记无效

`Rpc_version`
指定RPC版本号或服务号。版本号可以是一个单值或者一个范围中如2–3

`rpc_number`
如果RPC程序号不在/etc/rpc中，就指定它

Env
用空格分开的VAR=VALUE表，其中VAR是一个shell环境变量且VALUE是其设置值。这些值以及xinetd的环境都在激活时传送给服务程序。这个属性支持=和+=操作符

Passenv
用空格分开的xinetd环境中的环境变量表，该表在激活时传递给服务程序。

设置no就不传送任何变量。该属性支持所有操作符

`Only_from`
用空格分开的允许访问服务的客户机表。表2种给出客户机语法。如果不为该属性指定一个值，就拒绝访问这项服务。该属性支持所有操作符。

`No_access`
用空格分开的拒绝访问服务的客户机表。表2给出客户机语法。该属性支持所有操作符

Instances
接受一个大于或等于1的整数或UNLIMITED。设置可同时运行的最大进程数。UNLIMITED意味着xinetd对该数没有限制。

`Log_type`
指定服务log记录方式，可以为：

> SYSLOG facility[level]：设置该工具为daemon，auth，user或local0–7。设置level是可选的，可以的level值为emerg，alert，crit，err，warning，notice，info，debug，默认值为info
> file[soft[hard]]：指定file用于记录log，而不是syslog。限度soft和hard用KB指定（可选）。一旦达到soft限，xinetd就登记一条消息。一旦达到hard限，xinetd停止登记使用该文件的所有服务。如果不指定hard限，它成为soft加1%，但缺省时不超过20MB.缺省soft限是5MB

Redirect
该属性语法为redirect=Ipaddress port。它把TCP服务重定向到另一个系统。如果使用该属性，就忽略server属性

Bind
把一项服务绑定到一个特定端口。语法是bind=Ipaddress。这样有多个接口（物理的或逻辑的）的主机允许某个接口但不是其他接口上的特定服务（或端口）

`Log_on_success`
指定成功时登记的信息。可能值是:

> PID:进程的PID。如果一个新进程没被分叉，PID设置为0。
> HOST：客户机主机IP地址
> USERID:通过RFC1413高用捕获客户机用户的UID。只可用于多线程流服务。
> EXIT：登记进程终止和状态
> DURATION：登记会话持续期

缺省时不登记任何信息。该属性支持所有操作符

Log_on_failure
指定失败时登记的信息。总是登记表明错误性质的消息。可能值是:

ATTEMPT：记录一次失败的尝试。所有其他值隐含为这个值。
HOST：客户机主机IP地址。
USERID：通过RFC1413调用捕获客户机用户的UID。只可用于多线程流服务。
RECORD：记录附加的客户机信息如本地用户，远程用户和终端的类型。缺省时不登记任何信息。该属性支持所有操作符。

Disabled
只可用于defaults项（参看本小节后面的defaults项），指定被关闭的服务列表，是用空格分开的不可用服务列表来表示的。它和在/etc/xinetd.conf文件中注释掉该服务项有相同的效果。

我们首先看一个简单的例子。例1是配置文件/etc/xinetd.conf的一个范例。这两种服务的定义看上去像/etc/inetd.conf的原因是因为它们是用itox工具从/etc/inetd.conf转换得来的，只把/etc/inetd.conf项对应转换成适当的xinetd语法。这样，这些属性（在大括号中的=号的左边）意义是非常直接的，其相关值（在大括号中的=号的右边）也是如此。

例1　文件/etc/xinetd.conf中的一部分

```
Serice ftp
{
Socket_type=stream
    protocol=tcp
    wait=no
    user=root
    server=root
    Server_args= - 1  - a
}

Service telnet
{
    Socket_type=stream
    protocol=tcp
    wait=tcp
    user=root
    server=/usr/sbin/in.telnetd
}
```

创建/etc/xinetd.conf文件最容易的方法是用itox工具（该例假定当前工作目录是xinetd的编译目录）：

```
#xinetd/itox  -daemon_dir /usr/sbin  /etc/xinetd.conf
```

itox的参数-daemon_dir /usr/sbin指定服务程序的目录位置，如果实现了TCP_Wrappers，从/etc/inetd.conf中是不能确定它的，转换完成以后，就开始增加属性和值，以限制访问并增加登记，最后要手工修改/etc/xinetd.conf以充分利用xinetd的特性；否则，如果只把/etc/inetd.conf转换为/etc/xinetd.conf，xinetd的行为就和inetd一样了。

表1详述了在/etc/xinetd.conf中最常使用的一些属性和值。当然还有许多其他属性，详细配置选项可以在安装xinetd以后通过man xinetd.conf来得到。在本小节后面的“配置实例”中，将用一些例子阐明其中的许多属性。

表2中给出only_from和no_access表的语法，定义了指定主机名，IP地址和网络的语法。注意表2中最后一项netmask的语法和之前看到的有所不同。它没有采用传统的十进制或十六进制netmask的表示方法，而是采用一个整数表示从netmask（用二进制表示）的最高位（最左端）开始起每位都为1的位数。因此，给定例子的netmask值设置为20，意味着其最左端的20位都设置为1，而余下12位设置为0，或

```
11111111        11111111        11110000        00000000
```

它是十进制netmask255.255.240.0的二进制表示。

表2 /etc/xinetd.conf的访问控制表的语法

语　　法 描　　述
hostname 可解析的主机名。使用和这个主机名相关的所有IP地址
IPaddress 点和十进制形式的标准IP地址，如192.168.0.1
Net_name /etc/networks中的网络名
x.x.x.0 x.x.0.0 x.0.0.0 0.0.0.0 0作为通配符看待。如项88.3.92.0匹配从88.3.92.0到88.3.92.255的所有IP地址。项0.0.0.0匹配所有地址 x.x.x.{a，b，…} x.x{a，b，…} x.{a，b，…}

指定主机表。如172.19.32.{1，56，59}意味着含IP地址172，19.32.1，172.19.32.56和172.19.32.59的表

Ipaddress/netmask
定义要匹配的网络或子网。如172.19.16.0/20匹配从172.19.16.0到172.19.31.2-55的所有地址

在看了这些基本属性之后，下面我们仔细讨论那些必需的属性，特定服务和一些配置实例。

必需的属性

对每种服务都必须指定某些属性。一些服务比其他服务需要更多属性，因为它们不被缺省定义（即不在/etc/services或/etc/rpc中）。表3列出了必需的属性。

表3　必需的属性

语　法
描　述

`Socket_type`
所有服务

Wait
所有服务

User
在/etc/services或/etc/rpc中列出的服务

Server
非内部服务

Port
不在/etc/services中的非RPC服务

Protocol
不在/etc/services中的所有RPC服务和所有其他服务

`Rpc_version`
所有RPC服务

`Rpc_number`
不列在/etc/rpc中的任何RPC服务

特定的xinetd服务

/etc/xinetd.conf文件中有4个特殊项。它们分别是defaults，servers，services和xadmin。Defaults项不是一项服务，且不需要前置service关键字（否则它会被当成称为defaults的服务对待）。这些特殊项在以下4小段中描述。

Defaults项
/etc/xinetd.conf文件中的defaults项是实现为该文件中的所有服务指定某些属性的默认值。这些默认值可被每个服务项取消或修改。表4中列出可在defaults项中指定的属性。这个表也指明了具体服务项中可以修改哪些属性。

表4 defaults可用的属性

属性
服务修改

Log_on_success
Log_on_failure
Only_from
No_access
Passenv
可以用=操作符改写或用+ =或- =操作符修改

Instances
Log_type
可以用=操作符改写

disabled
可注释掉的服务，但disabled属性可用于某个服务项内

例2是defaults项的一个实例。从中看到对所有服务而言，登记消息将通过loca14.info有选择地送到syslogd进程。对成功的服务连接，将登记PID，客户机IP地址，中止状态和连接时间。对不成功的连接企图，将登记客户机IP地址。每项服务的最大实例数设置为8。禁止两项服务：in.tftpd和in.rexecd。

defaults项从本质上提供了在整个文件中建立某些属性的默认值，它应用于没有设置这些属性的所有服务。

例2 /etc/xinetd.conf中defaults项的示例

```
Defaults
{
Log_type = SYSLOG loca14 info
Log_on_success = PID HOST EXIT DURATION
Log_on_failure = HOST
Instances =8
Disabled = in.tftpd in.rexecd
}
```

注:如果在/etcxinetd.conf文件中没有defaults项，且之后决定增加这一项，你必须中止和重新启动xinetd以使defaults生效。这对任何要增加到/etc/xinetd.conf中的新服务也是正确的。它可以如下完成。

```
#kill-TERM xinetd
#/usr/sbin/xinetd
```

或者如果使用了启动脚本，则只需要简单执行。

```
#/etc/rc.d/init.d/xinetd restart
```

Servers项servers特殊服务是实现提供当前运行在服务器上的进程表，以及有关这些进程的确切信息。换句话说，它提供了活动连接的列表。这对排除故障和检查xinetd状态是个有用机制。例3显示了/etc/xinetd.conf文件中的一个实例servers项。注意这项服务的类型是INTERNAL，UNLISTED，这意味着它是xinetd的内部功能，且不列在/etc/services中。使用的端口是完全任意的。

例3　servers项的示例

```
Service servers
{
        type = INTERNAL UNLISTED
        Socket_type = stream
        Protocol = tcp
        Port = 9997
        Wait = no
        Only_from = 172.17.33.111
        Wait = no
}
```

注意这项服务仅用于特定IP地址172.17.33.111，它是服务器自身的IP地址。这表示不允许任何其他主机从这个服务器获得当前运行在服务器上的进程列表。这样做的原因是显而易见的：如果这条信息可被其他系统上的主机获取，基于对当前正在运行的进程的了解就加以利用。除用于调试之外，一般不要运行该服务，因为172.17.33.111上的任何用户通过执行例4中的telnet 172.17.33.111　9997都能获取这条信息。注意xinetd仅提供这条信息就退出，不提供交互连接。例4中的输出告诉我们有两个

正在运行的telnet进程（第5行和第31行），一个进程PID为5931，另一个为5961（分别为第6行和第32行），有一个ftp进程（第18行），其运行PID为5960（第19行）。

例4　servers服务的输出示例

```
1  $ telnet topcat 9997
2  Trying 172.17.33.111……
3  Connected to topcat
4  Escape character is  '^]'
5  telnet server
6  Pid=5931
7  Start_time=Sat Apr 17 10:32:15 1999
8  Connection info:
9  State=CLOSED
10 Service=telnet
11 Descriptor=20
12 Flags=9
13 Remote_address=10.48.3.2, 39958
14 Alternative services=
15 Log_remote_user=YES
16 Writes_to_log=YES
17
18 ftp server
19 Pid=5960
20 Start_time=Sat Apr 17 10:49:06  1999
21 Connection info:
22 State=CLOSED
23 Service=ftp
24 Descriptor=20
25 Flags=9
26 Remote_address=172.17.55.124, 2320
27 Alternative services=
28 Log_remote_user=YES
29 Writes_to_log=YES
30
31 telnet server
32 Pid=5961
33 Start_time=Sat Apr 17 10:49:20  1999
34 Connection info:
35 State=CLOSED
36 Service=telnet
37 Descriptor=20
38 Flags=9
```

```
39 Remote_address=172.17.1.3，35461
40 Alternative services=
41 Log_remote_user=YES
42 Writes_to_log=YES
43
44 Connection closed by foreign host
45 $
```

Services项
services特定项的目的是提供可用服务的列表。对services特定项来说，这是个有用的排除故障工具，但为了上述同样的安全因素，可能不会去用它。尽管如此，还是要看一看它如何工作。

例5是services项的一个示例。端口号的选择也是任意的。也应注意访问限制于topcat，这是服务器自身的主机名。

例5 /etc/xinetd.conf中services项示例

```
Service services
{
        type = INTERNAL UNLISTED
        Socket_type = stream
        protocol = tcp
        port = 8099
        wait = no
        Only_ from = topcat
}
```

对于servers服务来说，任何用户可执行telnet topcat 8099，并获得来自services服务的输出。例6给出了连接8099端口的实例信息信息。注意xinetd仅提供这条信息就退出，而不提供任何交互连接。

例6　查询services内部服务的输出

```
$ telnet topcat 8099
Trying 172.17.33.111……
Connected to topcat.
Escape character is  '^]'
Servers tcp 9997
Services tcp 8099
ftp tcp 21
telnet tcp 23
Shell tcp 514
Login tcp 513
Talk udp 517
Ntalk udp 518
```

```
Pop-2 tcp 109
Pop-3 tcp 110
Imap tcp 143
Linuxconf tcp 98
Connection closed by foreign host.
$
```

Xadmin项
这个特定服务项提供以交互方式获得services特定服务所提供信息的方法。例7是/etc/xinetd.conf项的一个示例（端口号的选择也是任意的），类似于services和servers服务，这项服务也没有口令或其他保护，所以要谨慎设置服务的only_from项，以增强安全性。

例7 xadmin项

```
Service xadmin
{
        type = INTERNAL UNLISTED
        socket_type = stream
        protocol = tcp
        port = 9967
        wait = no
        Only_from = topcat
}
```

对前两个特定服务类型来说，你只需telnet到所列端口上，即telnet topcat 9967。和前两项服务不同，xadmin提供了一个交互环境。一旦连接到xadmin服务器上，就可执行5个命令。它们是help，show，run，bye和exit。Help命令显示其他命令以及一个简短的用法消息。Bye和exit命令都关闭这个连接。Show run和show avail命令分别提供servers和services提供的信息。

警告:像前3段中指出的一样，这些特定服务servers，services和xadmin产生的信息可用于攻击系统。可能不需要一直运行这些服务，或许只当你调试时才需要。在任何时候，如果的确运行了这些服务，要确信用access_from和/或no_access配置了访问控制。也可以为这些服务使用bind属性以进一步限制访问。

3.配置实例
这节中将看到一些不同的例子，与之相关的行为以及它们产生的登记消息。

访问控制

从一个简单的访问控制例子开始。在例10中，服务器topcat的login服务项允许IP地址以172开始但不以172.19开始的任何系统访问。这个例子包括

了defaults部分。假定这一项用于login服务，且从客户机上用rlogin命令激活，让我们检查成功和不成功企图，以及它们产生的登记。

假定主机underdog和IP地址是172.18.5.9。那么当Mary执行rlogin登记topcat时，会给予她访问权。这相成功的登录如例11所示。尽管例11说明了Mary的动作产生的log内容。如例12所示，该例中的最后一项反映了当Mary拆除登录时的退出情况。可用PID跟踪某次会话的退出，只要你指明这个PID将在/etc/xinetd.conf中登记。登记项如下：每个xinetd登记项记录日期和时间戳，之后服务器主机名，然后是xinetd，之后在括号中是xinetd的PID。例12中的第一条记录以start关键字开始，表明这个会话的开始，之后识别的激活进程（login），然后激活进程的PID，最后是客户机地址。

例10　在/etc/xinetd.conf中rlogin service项的示例

```
Defaults
{
       Log_type=SYSLOG loca14 info
       Log_on_success=PID HOST EXIT DURATION
       Log_on_failure=HOST
       instances=8
}

Service login
{
       Socket_type=stream
       protocol=tcp
       wait=no
       user=root
       flags=REUSE
       Only_from=172.0.0.0
       No_access=172.19.0.0
       Olg_on_success+=USERID
       Olg_on_failure+=USERID
       server=/usr/sbin/in.ftpd
       Server_args=-1  -a
}
```

例11 成功的rlogin企图

```
[mary@underdog]$ rlogin topcat
password:
last login:Wed Apr 14 17:45:02 from roadrunner
[mary@topcat]$
```

例12　和例11相关的登记项

Apr 15 11:01:46 topcat xinetd[1402]:START:login pid=1439 From=1-72.18.5.9
Apr 15 11:01:46 topcat xinetd[1439]:USERID:login OTHER:mary
...

...
apr 15 11:39:31 topcat xinetd[1402]:EXIT:login status:1 pid=1439 dura-tion=2265(sec)

第2项以USERID关键字开始，表明成功地发出了RFC1413调用。之后是服务名（login），远程系统对RFC1413调用（此时为OTHER）的响应，最后远程用户名（mary）。

这些log项的含义是很清楚的，但表4提供了这些关键字（如START，USERID和EXIT）和其含义的解释。

现在假定Joe想在主机sly.no.good.org(IP地址为19.152.1.5)上使用rlogin。例13显示了这一结果。看上去Joe连接被拒绝，或者可能他想强入。让我们看一看例14中的这三次企图所产生的登记项。注意登记项不包括远程用户名，尽管我们在例10中用`log_on_failure`属性特别请求那个信息。这是因为远程主机sly.no.good.org没有运行identd或类似进程。因为主机sly.no.good.org不在例10中的`only_from`表中，尽管在login服务项中增加了flags=IDONLY一项，它不会记录sly.no.good.org没有运行identd的事实。仅当主机得到许可时，这样一项登记记录才会出现。

表4 xinetd登记项的描述

登记关键词
格式和描述

START
`START: service_id[pid=PID][from=Ipaddress]`
当启动一项服务时记录该项。`Service_id`是服务名，像id属性指定的一样（如果不明确设置这个属性，它采用该服务参数的值：参看表10.10）；PID是被激活进程的标识符，或者如果没有进程被激活，就为0（仅当`log_on_S`-uccess指定了PID时才记录）：Ipaddress是客户机的IP地址（仅当HOST是l-`og_on_success`时才记录）

EXIT
`EXIT: service_id[type=s][pid=PID][duration=#(sec)]`
仅当为`log_on_success`指定了EXIT时，若进程终止就记录这项。Service-`_id`和PID项和以前一样。Type项记录退出状态或产生终止的信号。Durati-on捕获会话时间（秒数）且需要在`log_on_success`中说明DURATION选项

FAIL

FAIL: service_id reason[from=Ipaddress]
当失败的请求发生且至少为log_on_success属性指定了一个值时生成该项。Service_id如前。Reason是一个解释失败原因的简单词或短语。Ipaddress是客户机地址且需要为log_on_success属性设置HOST值才出现。

DATA
DATA: service_id data
仅当为log_on_failure指定了RECORD时才记录。Service_id如前。记录的data取决于服务，但通常包括远程用户名（如果可得到）和状态信息

USERID
USERID:service_id text
仅当为log_on_success或log_on_failure或者指定了USERID时，才记录这个住处。Service_id如前。Text包括客户机对RFC1413调用的响应且特别是远程用户名

NOID
NOID:service_id Ipaddress reason
仅当为flags属性设置了IDONLY值且至少为log_on_success或log_on_failure设置了USERID值时该项出现。Service_id如前。给出的Ipaddress是主机地址。Reason是失败状态

例13 来自未授权主机的失败rlogin企图

```
Sly.no.good.org $ rlogin topcat
Topcat:Connection reset by peer
Sly.no.good.org $ rlogin -1 paul topcat
Topcat:Connection reset by peer
Sly.no.good.org $ rlogin -1 mary topcat
Topcat:Connection reset by peer
Sly.no.good.org $
```

例14 和例10-50相关的登记项

```
Apr 15 12:08:40 topcat xinetd[1402]:FAIL:login address fr-
om-19.152.1.5
Apr 15 12:08:52 topcat xinetd[1402]:FAIL:login address fr-
om-19.152.1.5
Apr 15 12:08:49 topcat xinetd[1402]:FAIL:login address fr-
om-19.152.1.5
```

有一个最后登记项要检查。注意例10中的instances属性设置为8。对第9个登录会话会发生什么？当第9个用户试图rlogin到topcat时，那个用户会看到下列错误消息

```
rcmd:topcat:address already in use
```

并且topcat中为这一事件记录的登记项是

```
Apr 15 13:37:33 topcat xinetd(1402):FAIL:login service_li-
mit from-172.17.55.124
```

把这个记录和表4中的描述相对照，FAIL关键字之后是服务名，然后是对失败的解释（此时为service_limit），最后是客户机地址。

使用bind属性

bind属性允许把一个特定接口的IP地址和一个特定的服务关联。假定有一个内部ftp服务器，它通过匿名ftp为公司职员提供只读资源。再假定这个ftp服务器有两个接口，一个连接在公司环境中，另一个连接到专用内部网，通常只有在那个特定组中工作的职员可访问它。尽这个例子中只考虑基于接口提供两个不同ftp服务的需求。例15在/etc/xinetd.conf中说明了两个ftp服务项。它可以实现所期望的功能。出于完整性再次提供了defaults部分。

注意每个ftp服务项有一个唯一的id属性。对有相同名字的服务数目没有任何限制，只要每个有唯一的标识符。在这个例子中，为内部ftp服务器设置id属性为ftp，为外部匿名服务器设置id属性为ftp_chroot。注意在后一种情况下，激活的进程是/usr/sbin/anon/in.aftpd(对TCP_Wrappers来说是twist)，这和以前的服务是不同的。

例15　把服务绑定到特定地址上

```
Defaults
{
       Log_type=SYSLOG loca14 info
       Log_on_success=PID HOST EXIT DURATION
       Instances=8
}

Service ftp
{
       id=ftp
       Socket_type=stream
       protocol=tcp
       wait=no
       user=root
       Only_from=172.17.0.0 172.19.0.0/20
       bind=172.17.1.1
       Log_on_success+=USERID
       Log_on_failure+=USERID
       server=/usr/sbin/in.ftpd
       Server_args=-1 -a
}
```

```
Service ftp
{
       id=ftp_chroot
       Socket_type=stream
}


Service telnet
{
       Socket_type=stream
       Wait=no
       flags=REUSE
       user=root
       bind=172.17.33.111
       server=usr/sbin/in.telnetd
       Log_on_success=PID HOST EXIT DURATION USERID
       Log_on_failure=RECORD HOST
}

Service telnet
{
       Socket_type=stream
       protocol=tcp
       wait=no
       flags=REUSE
       user=root
       bind=201.171.99.99
       redirect=172.17.1.1 23
       Log_on_success=PID HOST EXIT DURATION USERID
       LOG_ON_FAILURE=record host
}
```

例16 redirect的结果

```
telnet 201.171.99.99
Trying 201.171.99.99
Connected to 201.171.99.99
Escape character is '^]'
UNIX(r) System V Release 4.0 (foghorn)
Login:
```

因为Linux对每个物理端口最多支持256个逻辑接口，因此理论上可以为系统上的每个物理地址代理256个不同的地址。

尽管redirect机制可能是非常有用的，但实现它时要小心。要确保在代理服务器和终端系统上进行登记。可是在高度受控的内部网络中，这个实

现可能是方便的。

包含TCP_Wrappers

/etc/xinetd.conf中包含TCP_Wrappers功能是如此简单,TCP_Wrappers的所有功能可通过xinetd包括进去，就像通过inetd一样。例17是/etc/xinetd.conf文件的一个实例，它为许多服务使用了TCP_Wrappers。当属性服务器设置为/usr/sbin/tcpd后，那个服务将被包裹。注意这样的项总是把server_args属性设置为要激活的进程（全路径）。载讨论xinetd的编译时，用到了libwrap
配置选项，但是无论是否用libwrap编译，例17中的配置文件都能发挥作用。用libwrap编译的作用是包含/etc/hosts.allow和/etc/hosts.deny中的访问限制。例7所示提供了TCP_Wrappers的一组完整特征，banners，spawn，twist等。

例17 在/etc/xinetd.conf中使用TCP_Wrappers

```
Defaults
{
       Log_type=SYSLOG loca14 info
       Log_on_success=PID HOST EXIT DURATION
       Log_on_failure=HOST
       instances=8
}
Service ftp
{
       id=ftp
       Socket_type=stream
       protocol=tcp
       wait=no
       user=root
       Only_from=172.17.0.0
       Log_on_success+=USERID
       Log_on_failure+=USERID
       Access_times=8:00-16:30
       server= /usr/sbin/tcpd
       Server_args= /usr/sbin/in.ftpd -l -a
}

Service telnet
{
       Socket_type=stream
       wait=no
       flags=NAMEINARGS REUSE
       User=root
```

```
        Bind=172.17.33.111
        server= /usr/sbin/tcpd
        Server_args= /usr/sbin/in.telnetd
        Log_on_success=PID HOST EXIT DURATION USERID
        Log_on_failure=RECORD HOST
}

Service telnet
{
        Socket_type=stream
        protocol=tcp
        wait=no
        flags=REUSE
        user=root
        bind=201.171.99.99
        redirect=172.17.1.1 23
        Log_on_success=PID HOST EXIT DURATION USERID
        Log_on_failure=RECORD HOST
}
...

...
```

xinetd进程

xinetd进程接受若干参数。这些参数可被特定服务default中的属性改写，或在一个或多个服务的单个属性项改写。然而，这里给出的所有参数或它们的缺省值控制xinetd自身的行为。例如，如果filelog标记指定为xinetd，那么将在那里登记所有状态转换消息，尽管/etc/xinetd.conf文件中为和服务相关消息指定了其他登记位置。可用参数列在表5中。

应注意xinetd报告的所有状态信息，总是出现在-syslog或-filelog标记指定的登记文件中，不管设置如何，即通过defaults还是在/etc/xinetd.conf中。如果要在一个文件中捕获xinetd的PID，可以用

```
xinetd -pid 2> /var/run.xinetd.pid
```

和xinetd一起使用的可用信号 xinetd进程也基于收到的信号采取特定的行动。表16描述了它接受的每个信号的功能。注意每当增加了新服务或defaults项，或每当改变了任何服务的如下属性：protocol，socket_type，type或wait时，必须用SIGTERM(或更简单的TERM)信号中止xientd。每当给xinetd发布一个软性或硬性重配置信号时，将写入例19中所示类型的登记项。这个特定例子是硬性重配置的结果。注意这次硬性重配置的结果是中止了一项服务（用dropped=1标识）。

表5 xinetd的标记

标　记
描　述

-d
调试模式。输出可和调试器如gdb一起使用

-sysllog facility
指定syslogd工具。是daemon，auth，user和loca10-7其中之一

-filelog file
指定登记写到file而不是syslog中。必须是完整路径名

-fconfig_file
指定配置文件。必须是完整路径名。缺省是/etc/xinetd.conf

-pid
把PID写入标准错误中

-loop rate
指定每秒钟分叉的进程数。缺省是10.对较快机器来说可能希望改变它

-reuse
设置可重用的TCP socket，这意味着以前的实例运行时也可启动其他进程。当和flags属性一起使用时，有更特殊的服务控制（参看表10.10）

-limit numproc
限制由xinetd启动的同时运行的进程总数为numproc

-
限制同时发生的RFC1413请求数为limit

-shutdownprocs limit
当log_on_failure属性中使用了RECORD值时，xinetd分叉称为shutdown的服务以收集服务终止时的信息。该选项限制同时运行的shutdown进程总数为limit

-cc interval
使xinetd每隔interval秒运行对其内部状态的一致性检查。用killall －IOT xinetd可手工实现

表18 xinetd信号

信　号

作　用

SIGUSR1
软性重配置。重读/etc/xinetd.conf并作相应调整

SIGUSR2
硬性重配置。重读/etc/xinetd.conf并杀死和配置文件中的建立准则不再匹配的所有进程。例如，如果一个客户机连接到这个服务器且又增加到no_access表中，那么这个信号会终止该客户机的会话

SIGQUIT
终止xinetd但不终止它分叉的任何进程

SIGTERM
终止xinetd分叉的所有进程；然后终止xinetd

SIGHUP
把xinetd状态信息写到/tmp/xinetd.dump中

SIGIOT
检查内部数据库毁坏情况并报告结果

例19　xinetd硬性重配置的登记记录

```
Apr 15 14:42:31 topcat xinetd[1402]:Starting hard reconfig-
uration
Apr 15 14:42:31 topcat xinetd[1402]:readjusting service ser-
vers
Apr 15 14:42:31 topcat xinetd[1402]:readjusting service ser-
vces
Apr 15 14:42:31 topcat xinetd[1402]:readjusting service tel-
net
Apr 15 14:42:31 topcat xinetd[1402]:readjusting service she-
ll
Apr 15 14:42:31 topcat xinetd[1402]:readjusting service log-
in
Apr 15 14:42:31 topcat xinetd[1402]:readjusting service talk
Apr 15 14:42:31 topcat xinetd[1402]:readjusting service nta-
lk
Apr 15 14:42:31 topcat xinetd[1402]:readjusting service pop-2
Apr 15 14:42:31 topcat xinetd[1402]:readjusting service pop-3
Apr 15 14:42:31 topcat xinetd[1402]:readjusting service imap
Apr 15 14:42:31 topcat xinetd[1402]:readjusting service linu-
xconf
```

```
Apr 15 14:42:31 topcat xinetd[1402]:readjusting service ftp
Apr 15 14:42:31 topcat xinetd[1402]:Reconfigured:new=1 ol-
d=12 dropped=1 (services)
```

注:确定某个修改的/etc/xinetd.conf文件被读的最可靠方式是停止并重启动xinetd进程。最好用SIGTERM信号中止xinetd。如这节中描述的，发给xzinetd一个SIGTERM使它中止（用SIGKILL或信号号9）其控制之下的每个进程。有时在xinetd的子进程中止之前有一个延时，这意味着如果杀死并立即重启动xinetd，它不可能绑定所有端口（对此xinetd的登记文件—-而不是这项服务指定的登记文件—-中含一个错误消息）。这就是为什么sleep3命令出现在例中的stop和start命令间的脚本中。对TCP服务如telnet和ftp用flags=REUSE属性及其值或指定xinetd自身的-reuse选项可完全消除这个问题。

38 大教堂和市集

大教堂和市集 原著：Eric Raymond 翻译：HansB

38.1 大教堂和市集

Linux的影响是非常巨大的。甚至在5年以前，有谁能够想象一个世界级的操作系统能够仅仅用细细的Internet连接起来的散布在全球的几千个开发人员有以业余时间来创造呢？

我当然不会这么想。在1993年早期我开始注意Linux时，我已经参与Unix和自由软件开发达十年之久了。我是八十年代中期GNU最早的几个参与者之一。我已经在网上发布了大量的自由软件，开发和协助开发了几个至今仍在广泛使用的程序（Nethack,Emacs VC和GND模式，xlife等等）。我想我知道该怎样做。

Linux推翻了许多我认为自己明白的事情。我已经宣扬小工具、快速原型和演进式开发的Unix福音多年了。但是我也相信某些重要的复杂的事情需要更集中化的，严密的方法。我相信多数重要的软件（操作系统和象Emacs一样的真正大型的工具）需要向建造大教堂一样来开发，需要一群于世隔绝的奇才的细心工作，在成功之前没有beta版的发布。

Linus Torvalds的开发风格（尽早尽多的发布，委托所有可以委托的事，对所有的改动和融合开放）令人惊奇的降临了。这里没有安静的、虔诚的大教堂的建造工作——相反，Linux团体看起来像一个巨大的有各种不同议程和方法的乱哄哄的集市（Linux归档站点接受任何人的建议和作品，并聪明的加以管理），一个一致而稳定的系统就象奇迹一般从这个集市中产生了。

这种设计风格确实能工作，并且工作得很好，这个事实确实是一个冲击。在我的研究过程中，我不仅在单个工程中努力工作，而且试图理解为什么Linux世界不仅没有在一片混乱中分崩离析，反而以大教堂建造者们不可想象的速度变得越来越强大。

到了1996年中，我想我开始理解了。我有一个极好的测试我的理论的机会，以一个自由软件计划的形式，我有意识的是用了市集风格。我这样做了，并取得了很大的成功。

在本文的余下部分，我将讲述这个计划的故事，我用它来明确一些自由软件高效开发的格言。并不是所有这些都是从Linux世界中学到的，但我们将看到Linux世界给予了它们一个什么样的位置。如果我是正确的，它们将使你理解是什么使Linux团体成为好软件的源泉，帮助你变得更加高效。

38.2 邮件必须得通过

1993年以前我在一个小的免费访问的名为Chester County InterLink的ISP的做技术工作，它位于Pennsylvania的West Chester。（我协助建立了CCIL，并写了我们独特的多用户BBS系统——你可以telnet到locke.ccil.org来检测一下。今天它在十九条线上支持三千的用户）。这个工作使我可以一天二十四小时通过CCIL的56K专线连在网上，实际上，它要求我怎么做！

所以，我对Internet email很熟悉。因为复杂的原因，很难在我家里的机器（snark.thyrsus.com）和CCIL之间用SLIP工作。最后我终于成功了，但我发现不得不时常telnet到locke来检查我的邮件，这真是太烦了。我所需要的是我的邮件发送到snark,这样biff(1)会在它到达时通知我。

简单地sendmail的转送功能是不够的，因为snark并不是总在网上而且没有一个静态地址。我需要一个程序通过我的SLIP连接把我的本地发送的邮件拉过来。我知道这种东西是存在的，它们大多使用一个简单的协议POP（Post Office Protocol）。而且，locke的BSD/OS操作系统已经自带了一个POP3服务器。

我需要一个POP3客户。所以我到网上去找到了一个。实际上，我发现了三、四个。我用了一会pop-perl，但它却少一个明显的特征：抽取收到的邮件的地址以便正确回复。

问题是这样的：假设locke上一个叫“joe”的人向我发了一封邮件。如果我把它取到snark上准备回复时,我的邮件程序会很高兴地把它发送给一个不存在的snark上的“joe”。手工的在地址上加上“@ccil.org”变成了一个严酷的痛苦。

这显然应是计算机替我做的事。（实际上，依据RFC1123的5.2.18节，sendmail应该做这件事）。但是没有一个现存的POP客户知道怎样做！于是这就给我们上了第一课：

1.每个好的软件工作都开始于搔到了开发者本人的痒处。

也许这应该是显而易见的（“需要是发明之母”长久以来就被证明是正确的），但是软件开发人员常常把他们的精力放在它们既不需要也不喜欢的程序，但在Linux世界中却不是这样——这解释了为什么从Linux团体中产生的软件质量都如此之高。

那么，我是否立即投入疯狂的工作中，要编出一个新的POP3客户与现存的那些竞争呢？才不是哪！我仔细考察了手头上的POP工具，问自己“那一个最接近我的需要？”因为：

2.好程序员知道该写什么，伟大的程序员知道该重写（和重用）什么。

我并没有声称自己是一个伟大的程序员，可是我试着效仿他们。伟大程序员的一个重要特点是建设性的懒惰。他们知道你是因为成绩而不是努力得到奖赏，而且从一个好的实际的解决方案开始总是要比从头干起容易。

例如，Linux并不是从头开始写Linux的。相反的它从重用Minix（一个386机型上的类似Unix的微型操作系统）的代码和思想入手。最后所有的Minix代码都消失或被彻底的重写了，但是当它们在的时候它为最终成为Linux的雏形做了铺垫。

秉承同样的精神，我去寻找良好编码的现成的POP工具，用来作为基础。

Unix世界中的代码共享传统一直对代码重用很友好（这正是为什么GNU计划不管Unix本身有多么保守而选取它作为基础操作系统的原因）。Linux世界把这个传统推向技术极限：它有几个T字节的源代码可以用。所以在Linux世界中花时间寻找其他几乎足够好的东西，会比在别处带来更好的结果。

这也适合我。加上我先前发现的，第二次寻找找到了9个候选者——fetchPOP，PopTart，get-mail，gwpop，pimp，pop-perl，popc，popm-

ail 和upop）。我首先选定的是“fetchpop”。我加入了头标重写功能，并且做了一些被作者加入他的1.9版中的改进。

但是几个星期之后，我偶然发现了Carl Harris写的“popclient”的代码，然后发现有个问题，虽然fetchpop有一些好的原始思想(比如它的守护进程模式)，它只能处理pop3，而且编码的水平相当业余(Seung-Hong是个很聪明但是经验不足的程序员)，Carl的代码更好一些，相当专业和稳固，但他的程序缺少几个重要的相当容易实现的fetchpop的特征(包括我自己写的那些)。

继续呢还是换一个? 如果换一个的话，作为得到一个更好开发基础的代价，我就要扔掉我已经有的那些代码。

换一个的一个实际的动机是支持多协议，pop3是用的最广的邮局协议，但并非唯一一个，Fetchpop和其余几个没有实现POP2.RPOP，或者APOP，而且我还有一个为了兴趣加入IMAP(Internet Message Access Protocol，最近设计的最强大的邮局协议)的模糊想法。

但是我有一个更加理论化的原因认为换一下会是一个好主意，这是我在Linux很久以前学到的：

3.“计划好抛弃，无论如何，你会的”(Fred Brooks,《神秘的人月》第11章)

或者换句话说，你常常在第一次实现一个解决方案之后才能理解问题所在，第二次你也许才足够清楚怎样做好它，因此如果你想做好，准备好推翻重来至少一次。

好吧(我告诉自己)，对fetchpop的尝试是我第一次的尝试，因此我换了一下。

当我在1996年6月25日把我第一套popclient的补丁程序寄给Carl Harris之后，我发现一段时间以前他已经对popclient基本上失去了兴趣，这些代码有些陈旧，有一些次要的错误，我有许多修改要做，我们很快达成一致，我来接手这个程序。不知不觉的，这个计划扩大了，再也不是我原先打算的在已有的pop客户上加几个次要的补丁而已了，我得维护整个的工程，而且我脑袋里涌动着一些念头要引起一个大的变化。

在一个鼓励代码共享的软件文化里，这是一个工程进化的自然道路，我要指出：

4.如果你有正确的态度，有趣的问题会找上你的，但是Carl Harris的态度甚至更加重要，他理解：

5.当你对一个程序失去兴趣时，你最后的责任就是把它传给一个能干的后继者。

甚至没有商量，Carl和我知道我们有一个共同目标就是找到最好的解决方案，对我们来说唯一的问题是我能否证明我有一双坚强的手，他优雅而快速的写出了程序，我希望轮到我时我也能做到。

38.3 拥有用户的重要性

于是我继承了popclient，同样重要的是，我继承了popclient的用户基础，用户是你所拥有的极好的东西，不仅仅是因为他们显示了你正在满足

需要，你做了正确的事情，如果加以适当的培养，他们可以成为合作开发者。

Unix传统另一有力之处是许多用户都是黑客，因为源优码是公开的，他们可以成为高效的黑客，这一点在Linux世界中也被推向了令人高兴的极致，这对缩短调试时间是极端重要的，在一点鼓励之下，你的用户会诊断问题，提出修订建议，帮你以远比你期望快得多的速度的改进代码。

6. 把用户当做协作开发者是快速改进代码和高效调试的无可争辩的方式。

这种效果的力量很容易被低估，实际上，几乎所有我们自由软件世界中的人都强烈低估了用户可以多么有效地对付系统复杂性，直到Linus让我们看到了这一点。

实际上，我认为Linus最聪明最了不起的工作不是创建了Linux内核本身，而是发明了Linux开发模式，当我有一次当着他的面表达这种观点时，他微笑了一下，重复了一句他经常说的话：“我基本上是一个懒惰的人，依靠他人的工作来获取成绩。”象狐狸一样懒惰，或者如Robert Heinlein所说，太懒了而不会失败。

回顾起来，在GNU Emacs Lisp库和Lisp代码集中可以看到Linux方法的成功，与Emacs的C内核和许多其他FSF的工具相比，Lisp代码库的演化是流动性的和用户驱动的，思想和原型在达到最终的稳定形式之前往往要重写三或四次，而且经常利用Internet的松散合作。

实际上，我自己在fetchmail之前最成功的作品要算Emacs VC模式，它是三个其他的人通过电子邮件进行的类似Linux的合作，至今我只见过其中一个人(Richard Stallman)，它是SCCS、RCS和后来的CVS的前端，为Emacs提供“one-touch”版本控制操作，它是从一个微型的、粗糙的别人写好的sccs.el模式开始演化的，VC开发的成功不像Emacs本身，而是因为Emacs Lisp代码可以很快的通过发布 / 测试 / 改进的过程。

(FSF的试图把代码放入GPL之下的策略有一个未曾预料到的副作用，它让FSF难以采取市集模式，因为他们认为每个想贡献二十行以上代码的人都必须得到一个授权，以使受到GPL的代码免受版权法的侵扰，具有BSD和MITX协会的授权的用户不会有这个问题，因为他们并不试图保留那些会使人可能受到质询的权力)。

38.4 早发布、常发布

尽量早尽量频繁的发布是Linux开发模式的一个重要部分，多数开发人员(包括我)过去都相信这对大型工程来说是个不好的策略，因为早期版本都是些充满错误的版本，而你不想耗光用户的耐心。

这种信仰强化了建造大教堂开发方式的必要性，如果目标是让用户尽可能少的见到错误，那你怎能不会仅仅每六个月发布一次(或更不经常)，而且在发布之间象一只狗一样辛勤“捉虫”呢? Emacs C内核就是以这种方式开发的，Lisp库，实际上却相反，因为有一些有FSF控制之外的Lisp库，在那里你可以独立于Emacs发布周期地找寻新的和开发代码版本。

这其中最重要的是Ohio州的elisp库，预示了今天的巨大的Linux库的许多特征的精神，但是我们很少真正仔细考虑我们在做什么，或者这个库的

存在指出了FSF建造教堂式开发模式的什么问题，1992年我曾经做了一次严肃的尝试，想把Ohio的大量代码正式合并到Emacs的官方Lisp库中，结果我陷入了政治斗争中，彻底失败了。

但是一年之后，在Linux广泛应用之后，很清楚，一些不同的更加健康的东西诞生了，Linus的开发模式正好与建造教堂方式相反，Sunsite和tsx-11的库开始成长，推动了许多发布。所有这些都是闻所未闻的频繁的内核系统的发布所推动的。

Linus以所有实际可能的方式把它的用户作为协作开发人员。

7. 早发布、常发布、听取客户的建议

Linus的创新并不是这个(这在Unix世界中是一个长期传统)，而是把它扩展到和他所开发的东西的复杂程度相匹配的地步，在早期一天一次发布对他来说都不是罕见的!而且因为他培育了他的协作开发者基础，比其他任何人更努力地充分利用了Internet进行合作，所以这确实能行。

但是它是怎样进行的呢?它是我能模仿的吗?还是这依赖于Linus的独特天才?

我不这样想，我承认Linus是一个极好的黑客(我们有多少人能够做出一个完整的高质量的操作系统内核?)，但是Linux并不是一个令人敬畏的概念上的飞跃，Linus不是(至少还不曾是)象Richard stallman或James Gosling一样的创新天才，在我看来，Linus更象一个工程天才，具有避免错误和开发失败的第六感觉，掌握了发现从A点到B点代价最小的路径的决窍，确实，Linux的整个设计受益于这个特质，并反映出Linus的本质上保守和简化设计的方法。

如果快速的发布和充分利用Internet不是偶然而是Linus的对代价最小的路径的洞察力的工程天才的内在部分，那么他极大增强了什么?他创建了什么样的方法?

问题回答了它自己，Linus保持他的黑客用户经常受到激励和奖赏：被行动的自我满足的希望所激励，而奖赏则是经常(甚至每天)都看到工作在进步。

Linus直接瞄准了争取最多的投入调试和开发的人时，甚至冒代码不稳定和一旦有非常棘手的错误而失去用户基础的险，Linus似乎相信下面这个：

8. 如果有一个足够大的beta测试人员和协作开发人员的基础，几乎所有的问题都可以被快速的找出并被一些人纠正。

或者更不正式的讲：“如果有足够多的眼睛，所有的错误都是浅显的”(群众的眼睛是雪亮的)，我把这称为“Linus定律”。

我最初的表述是每个问题“对某些人是透明的”，Linus反对说，理解和修订问题的那个人不一定非是甚至往往不是首先发现它的人，“某个人发现了问题”，他说，“另一个理解它，我认为发现它是个更大的挑战”，但是要点是所有事都趋向于迅速发生。

我认为这是建造教堂和集市模式的核心区别，在建造教堂模式的编程模式看来，错误和编程问题是狡猾的、阴险的、隐藏很深的现象，花费几个月的仔细检查，也不能给你多大确保把它们都挑出来的信心，因此很长的发布周期，和在长期等待之后并没有得到完美的版本发布所引起的失望都是不可避免的。

以市集模式观点来看，在另一方面，我们认为错误是浅显的现象，或者至少当暴露给上千个热切的协作开发人员，让他们来对每个新发布进行测试的时候，它们很快变得浅显了，所以我们经常发布来获得更多的更正，作为一个有益的副作用，如果你偶尔做了一个笨拙的修改，也不会损失太多。也许我们本不应该这样的惊奇，社会学家在几年前已经发现一群相同专业的(或相同无知的)观察者的平均观点比在其中随机挑选一个来得更加可靠，他们称此为“Delhpi效应”，Linus所显示的证明在调试一个操作系统时它也适用——Delphi效应甚至可以战胜操作系统内核一级的复杂度。

我受Jeff Dutky (dutky@wam.umd.edu)的启发指出Linus定律可以重新表述为“调试可以并行”，Jeff观察到虽然调试工作需要调试人员和对应的开发人员相交流，但它不需要在调试人员之间进行大量的协调，于是它就没有陷入开发时遇到的平方复杂度和管理开销。

在实际中，由于重复劳动而导致的理论上的丧失效率的现象在Linux世界中并不是一个大问题，“早发布、常发布策略”的一个效果就是利用快速的传播反馈修订来使重复劳动达到最小。

Brooks甚至做了一个与Jeff相关的更精确的观察：“维护一个广泛使用的程序的成本一般是其开发成本的40%，奇怪的是这个成本受到用户个数的强烈影响，更多的用户发现更多的错误”(我的强调)。

更多的用户发现更多的错误是因为更多的用户提供了更多测试程序的方法，当用户是协作开发人员时这个效果被放大了，每个找寻错误的人都有自己稍微不同的感觉和分析工具，从不同角度来看待问题。“Delphi效应”似乎因为这个变体工作变得更加精确，在调试的情况下，这个变体同时减小了重复劳动。

所以加入更多的beta测试人员虽不能从开发人员的P.O.V中减小“最深”的错误的复杂度，但是它增加了这样一种可能性，即某个人的工具和问题正好匹配，而这个错误对这个人来说是浅显的。

Linus也做了一些改进，如果有一些严重的错误，Linux内核的版本在编号上做了些处理，让用户可以自己选择是运行上一个“稳定”的版本，还是冒遇到错误的险而得到新特征，这个战略还没被大多数Linux黑客所仿效，但它应该被仿效，存在两个选择的事实让二者都很吸引　人。

38.5　什么时候玫瑰不是玫瑰?

在研究了Linus的行为和形成了为什么它成功的理论之后，我决定在我的工程(显然没有那么复杂和雄心勃勃)里有意识的测试这个理论。但我首先做的事是熟悉和简化Popclient。Carl Harris的实现非常好，但是有一种对许多C程序来说没有必要的复杂性。他把代码当作核心而把数据结构当作对代码的支持，结果是代码非常漂亮但是数据结构设计得很特别，相当丑陋(至少对以这个老LISP黑客的标准来看)，然而除了提高代码和数据结构设计之外，重写它还有一个目的，就是要把它演化为我彻底理解的东西，对修改你不理解的程序中的错误负责可不是一件有趣的事。

第一个月我只是在领会Carl's的基本设计的含义，我所做的第一个重大修改是加入了IMAP支持，我把协议机重新组织为一个通用驱动程序和三个方法表(对应POP2、POP3和IMAP)，这个前面的修改指出一个需要程序

员(特别是象C这种没有自然的动态类型支持的语言)记在脑中的一般原理：

9. 聪明的数据结构和笨拙的代码要比相反的搭配工作的更好

Fred Brooks也在他第11章中讲道："让我看你的［代码］，把你的[数据结构]隐藏起来，我还是会迷惑；让我看看你的[数据结构]，那我就不需要你的[代码]了，它是显而易见的"。

实际上，他说的是"流程图"和"表"，但是在三十年的术语／文化演进之后，事情还是一样的。

此时(1996年9月初，在从零开始六个月后)，我开始想接下来修改名字——毕竟，它已不仅仅是一个POP客户，但我犹豫了，因为还没有什么新的漂亮设计呢，我的popclient版本需要有自己的特色。

当fetehmail学会怎样把取到的邮件转送到SMTP端口时，事情就完全改变了，但是首先：上面我说过我决定使用这个工程来测试我关于Linus Torualds所做的行为的理论，(你可能会问)我怎样做到这点呢? 以下面的方式：

- 我尽早尽量频繁的发布(几乎从未少于每十天发布一次；在密集开发的时候是每天一次)。
- 我把每一个和我讨论fetchmail的人加入一个beta表中。
- 每当我发布我都向beta表中的人发出通告，鼓励人们参与。
- 我听取beta测试员的意见，向他们询问设计决策，对他们寄来的补丁和反馈表示感谢。

这些简单的手段立即收到的回报，在工程的开始，我收到了一些错误报告，其质量足以使开发者因此被杀掉，而且经常还附有补丁、我得到了理智的批评，有趣的邮件，和聪明的特征建议，这导致了：

10. 如果你象对待最宝贵的资源一样对待你的beta测试员，他们就会成为你最宝贵的资源。

38.6 popclient变成了Fetchmail

这个工程的真正转折点是Harry Hochleiser寄给我他写的代码草稿，他把邮件转发到客户端机器的SMTP端口，我立即意识到这个特征的可靠实现将淘汰所有其他的递送模式。

几个星期以来我一直在修改而不是改进fetchmail，因为我觉得界面设计虽然有用但是太笨拙琐碎了，到处充满了太多的粗陋的细小选项。

当我思考SMTP转发时我发现popclient试图做的事太多了，它被设计成既是一个邮件传输代理(MTA)也是一个本地递送代理(MDA)。使用SMTP转发，它就可以从MDA的事务中解脱出来而成为一个纯MTA，而象sendmail一样把邮件交给本地递送程序来处理。

既然端口25在所有支撑TCP／IP的平台上早已被预留，为什么还要为一个邮件传输代理的配置或为一个邮箱设置加锁的附加功能而操心呢?尤其是当这意味着抽取的邮件就象一个正常的发送者发出的SMTP邮件一样，而这就是我们需要的。

这里有几个教益：第一，SMTP转发的想法是我有意识地模拟Linus的方法以来的最大的单个回报，一个用户告诉我这个非同寻常的想法－－我所需做的只是理解它的含义。

11. 想出好主意是好事，从你的用户那里发现好主意也是好事，有时候后者更好。

很有趣的是，你很快将发现，如果你完全承认你从其他人那里得到多少教益的话，整个世界将会认为所有的发明都是你做出的，而你会对你的天才变得谦虚。我们可以看到这在Linus身上体现得多明显!(当我在1997年8月的Perl会议上发表这个论文时，Larry Wall坐在前排，当我讲到上面的观点时，他激动的叫了出来："对了!说对了!哥们!"所有的听众都哄堂大笑起来，因为他们知道同样的事情也发生在Perl的发明者身上)。

于是在同样精神指导下工程进行了几个星期，我开始不光从我的用户那儿也从听说我的系统的人那儿得到类似的赞扬，我把一些这种邮件收藏起来，我将在我开始怀疑自己的生命是否有价值时重新读读这些信。:)

但是有两个更基本的，非政治性的对所有设计都有普遍意义的教益。

12. 最重要和最有创新的解决方案常常来自于你认识到你对问题的概念是错误的。

一个衡量fetchmail成功的有趣方式是工程的beta测试人员表(fegtchmail的朋友们)的长度，在创立它的时候已经有249个成员了，而且每个星期增加两到三个。

实际上，当我在1997年5月校订它时，这张表开始因为一个有趣的原因而缩短了，有几个人请求我把他们从表中去掉，因为fetchmail已经工作的如此之好，他们不需要看到这些邮件了!也许这是一个成熟的市集风格工程的生命周期的一部分。

我以前一直在解决错误的问题，把popclient当作MTA和具有许多本地递送模式的MDA的结合物，Fetchmail的设计需要从头考虑为一个纯的MTA，做为一个普通Internet邮件路径的一部分。

当你在开发中碰了壁时(当你发现自己很难想通下一步时)，那通常不是要问自己是否找到正确答案，而是要问是否问了正确问题，也许需要重新构造问题。

于是，我重新构造了我的问题，很清楚，要做的正确的事是(1)把SMTP转发支持放在通用驱动程序中，(2)把它做为缺省模式，(3)最终分离所有其他的递送模式，尤其是递送到文件和标准输出的选项。

我在第三步上犹豫了一下，担心会让popdiant的长期用户对新的递送方法感到烦心，在理论上，他们可以立即转而转发文件或者他们的非sendmail等价物来得到同样的效果，在实际中这种转换可能会很麻烦。但是当我这么做之后，证明好处是巨大的，驱动程序代码的冗余的部分消失了，配置完全变得简单了－－不用屈从于系统MDA和用户的邮箱，也不用为下层OS是否支持文件锁定而担心了。

而且，丢失邮件的唯一漏洞也被堵死了，如果你选择了递送到一个文件而磁盘已满，你的邮件就会丢失，这在SMTP转发中不会发生，因为SMTP侦听器不会返回OK的，除非邮件可以递送成功或至少被缓冲留待以后递送。

还有，性能也改善了(虽然在单次执行中你不会注意到)，这个修改的另一个不可忽视的好处是手册变得大大简单了。

后来，为了允许处理一些罕见的情况，包括动态SLIP，我必须回到让用户定义本地MDA递送上来，但是我发现了一个更加简单的方法。

所有这些给了我们什么启发呢?如果可以不损失效率，就要毫不犹豫抛弃陈旧的特性，Antonine de Saint-Exupery(在他成为经典儿童书籍作家之前是一个飞行员和飞机设计师)曾说过：

13. “最好的设计不是再也没有什么东西可以添加了，而是再也没有什么东西可以去掉。”

当你的代码变得更好和更简单时，这就是你知道它是正确的时候了，而且在这个过程中，fetehmail的设计具有了自己的特点，而区别于其前身popclient。

现在是改名的时候了，这个新的设计看起来比老popclient更象一个sendmail的复制品，它们都是MTA，但是Senmail是推然后递送，而新的popclient是拉然后递送。于是，在两个月之后，我把它重新命名为fetehmail。

38.7 Fetchmail成长起来

现在我有了一个简洁和富有创意的设计，工作得很好的代码，因为我每天都用它，和一直在增长的beta表，它让我渐渐明白我已经不是在从事只能对少数其他人有用的工作中，我写了一个所有有一个Unix邮箱和SLIP / PPP邮件连接的人都真正需要的程序。

通过SMTP转发功能，它成为一个潜在的“目录杀手”，远远领先于它的竞争者，这个程序如此能干以至于其他的程序不但被放弃简直被忘记了。

我知道你不可以真得瞄准或计划出这样的结果，你只能努力去设计这些强大的思想，以后这些结果就好象是不可避免的、自然的、注定了的，得到这种思想的唯一办法是获取许多思想，或者用工程化的思考其他人的好主意而超过原来想到它的人的设想。

Andrew Tanenbanm原来设想建造一个适合386的简单的Unix用做教学，Linus Torvalels把Andrew的可能想到的Minix可以做什么的概念推进了一步，成长为一个极好的东西，同样的(虽然规模较小)，我接受了Card Harris和Harry Hochheiser的想法，把它们变得更强大，我们都不是人们所浪漫幻想的天才的创始人，但是大多数科学和工程和软件开发不是被天才的创始人完成的，这和流传的神话恰恰相反。

结果总是执着的原因－－实际上，它是每个黑客为之生存的成功!而且它们意味着我必须把自己的标准定高一点，为了把fetchmail变得和我所能设想的那样好，我必须不仅为我自己的需要写代码，而且也要包括对在我生活围主页外的人们的需求的支持，而且同时也要保证程序的简单和健壮。

在实现它之后我首先写的最重要的特征是支持多投－－从集中一组用户的邮件的邮箱中取出邮件，然后把它路由到每个人手中。

我之所以加上多投功能部分是因为有些用户一直在闹着要它，更是因为我想它可以从单投的代码中揭露出错误来，让我完全一般地处理寻址，而

且这被证明了。正确解释RFC822花了我相当长的时间，不仅因为它的每个单独部分都很难，而且因为它有一大堆相互依赖的苛刻的细节。

但是多投寻址也成为一个极好的设计决策，由此我知道：

14. 任何工具都应该能以预想的方式使用，但是一个伟大的工具提供你没料到的功能。

Fetchmant多投功能的一个没有料到的用途是在SLIP / PPP的客户端提供邮件列表、别名扩展。这意味着一个使用个人机器的人不必持续访问ISP的别名文件就能通过一个ISP帐户管理一个邮件列表。我的beta测试员提出的另一个重要的改变是支持8位MIME操作，这很容易做，因为我已经仔细的保证了8位代码的清晰，不仅因为我预见到了这个特性的需求，而且因为我忠实于另一准则：

15. 当写任何种类的网关型程序时，多费点力，尽量少干扰数据流，永远不要抛弃信息，除非接收方强迫这么作!

如果我不遵从这个准则，那么8位MIME支持将会变得困难和笨拙，现在我所需要做的，是只读一下RFC 1652，在产生信头的逻辑加上一点而已。

一些欧洲用户要求我加上一个选项来限制每次会话取得消息数(这样他们就可以从昂贵的电话网中控制花费了)，我很长一段时间拒绝这样做，而且我仍然对它不很高兴，但是如果你是为了世界而写代码，你必须听取顾客的意见——这并不随他们不付给你钱而改变。

38.8 从Fetchmail得来的另一些教益

在他们回到一般的软件工程问题以前，还有几个从fetchmail得到的教益需要思考。

rc文件语法包括可选的“noise”关键字，它被扫描器完全忽略了，当你把它们全抽取出的时候，关键字 / 值对更具可读性。

当我注意到rc文件的声明在多大程度上开始象一个微型命令语言时，这是一个Late-night的体验(这也是我为什么把popclient原来的“server”关键字改成了“poll”)。

对我来说似乎把这个微型命令语言变得更象英语可能会使它更容易使用。现在，虽然我对经过Emacs和HTML及许多数据库引擎所证实的“把它做成一个语言”的设计方式确信不疑，但是我并不是一个通常的“类英语”语法的狂热拥护者。

传统程序员容易控制语法使它尽量精确和紧凑，完全没有冗余，这是计算机资源还很昂贵时遗留下的一种文化传统，所以扫描策略需要尽可能的廉价和简单，而具有50%冗余度的英语，看来好象是一个非常不合适的模型。

这并不是我不用类英语语法的原因，我提到这一点是为了推翻它，在更廉价的时钟周期与核心的时代，简洁并没有走到尽头，今天对一个语言来说，对人更方便比对机器更廉价来的更加重要。

然而，有几个原因提醒我们小心一点，一个是扫描策略的复杂度开销——你并不想把它变成一个巨大的错误来源和让用户困惑，另一个是试图使

语言表面上的类似可以和传统语言一样令人困惑(你可以在许多4GL和商业数据库查询语言上看到这一点)。

Fetchmail的控制语法避免了这些问题，因为语言的领域是极其有限的。它一点也不象一个一般性的语言，它很简单地描述的东西并不复杂，所以很少可能在英语的一个小子集与实际的控制语言之间发生混淆，我想这有一个更广泛的教益：

16. 如果你的语言一点也不象是图灵完备的，严格的语法会有好处。

另一个教益是关于安全的，一些fetchmail用户要求我修改软件把口令加密存贮在rc文件里，这样觑探者就不能看到它们了。

我没有这样做，因为这实际上起不到任何保护作用，任何有权读取你的rc文件的人都可以以你的名义运行fetchmail——如果他们要破你的口令，它们可以从fetchmail的代码中找到制作解码器的方法。

所以fetchmail口令的加密都会给那些不慎重思考的人一种安全的错觉，这里一般性的准则是：

17. 一个安全系统只能和它的秘密一样安全，当心伪安全。

38.9 集市风格的必要的先决条件

本文的早期评审人员和测试人员坚持提出成功的市集模式开发的先决条件，包括工程领导人的资格问题和在把项目公开和开始建造一个协作开发人员的社团的时候代码的状态。

相当清楚，不能以一个市集模式从头开发一个软件，我们可以以市集模式、测试、调试和改进，但是以市集模式从头开始一个项目将是非常困难的，Linus没有这样做，我也没有，初期的开发人员的社团应该有一此可以运行和测试的东西来玩。

当你开始创建社团时，你需要演示的是一个诺言，你的程序不需要工作的很好，它可以很粗糙、很笨拙、不完整和缺少文档、它不能忽略的东西是要吸引哪些人卷入一个整洁的项目。

Linux和fetchmail都是以一个吸引人的基本设计进入公共领域的，许多和我一样在思考市集模式的人已经正确的认为这是非常关键的，然后得出了一个结论，工程领导者的高度的设计直觉和聪颖是必不可少的。

但是Linus是从Unix得到他的设计的，我最初是从先前的popmail得到启发的(虽然相对Linux而言，它最后改变巨大)，所以市集风格的领导人 / 协调人需要有出众的设计才能，或者他可以利用别人的设计才能?

我认为能够提出卓越的原始设计思想对协调人来说不是最关键的，但是对他 / 她来说绝对关键的是要能把从他人那里得到的好的设计重新组织起来。

Linux和fetchmail项目都显示了这些证据，Linus(如同前面所说)并不是惊人的原始设计者，但他显示了发现好的设计并把它集成到Linux内核中的强大决窍。还有我也描述了怎样从别人那里得到了fetchmail中最强大的设计思想(SMTP转发)。

本文的早期读者称赞我，说因为我做了许多关于原始设计的事，所以倾向于低估原始设计在市集项目中的价值，也许有些是对的吧，但是设计(而不是编码或调试)本来就是我最强的能力。

变得聪明和软件设计的原始创作的问题是它会变成一个习惯，当需要保持事物健壮和简洁的时候，你却开始把事情变得漂亮但却复杂。我曾经犯过错误，使得一些项目因我而崩溃了，但我努力不让它发生在fetchmail身上。

所以我相信fetchmail项目的成功部分是因为我抑制自己不要变得太聪明，这说明(至少)对市集模式而言原始设计并不是本质的，请考察一下Linux假设Linus Torvalds在开发时试图彻底革新操作系统设计，它还会象今天我们所拥有的内核那样稳定和成功吗?

当然基本的设计和编码技巧还是必需的，但我希望每个严肃考虑发起一个市集计划的人都已至少具备这些能力，自由软件社团的内部市场对人们有某些微妙的压力，让他们不要发起自由不能搞定的开发，目前为止，这工作得仍然相当好。

对市集项目来说，我认为还有另一种通常与软件开发无关的技能和设计能力同样重要——或者更加重要，市集项目的协调人或领导人必须有良好的人际和交流能力。

这是很显然的，为了建造一个开发社团，你需要吸引人，你所做的东西要让他们感到有趣，而且要保持他们对他们正在做的工作感到有趣，而且要保持他们对他们正在做的工作感到高兴，技术方面对达成这些目标有一定帮助，但这远远不是全部，你的个人素质也有关系。

并不是说Linus是一个好小伙子，让人们喜爱并乐于帮助他，也并不是说我是个积极外向的，喜欢扎堆儿工作，有出众的幽默感的人，对市集模式的工作而言，至少有一点吸引人的技巧是非常有帮助的。

38.10 自由软件的社会学语境

下述如实：最好的开发是从作者解决每天工作中的个人问题开始的，因为它对一大类用户来说是一个典型问题，所以它就推广开来了，这把我们带回到准则1，也许是用一个更有用的方式来描述：

18. 要解决一个有趣的问题，请从发现让你感兴趣的问题开始。

这是Carl Harris和原先的popclient的情形，也是我和fetchmail的情形，但这已在很长一段时间被大家知晓了，Linux和fetchmail的历史要求我们注意的有趣之处是下一个阶段——软件在一个庞大的活跃的用户和协作开发人员的社团中的进化。

在《神秘的人月》一书中，Fred Brooks观察到程序员的工作时间是不可替代的：在一个误了工期的软件项目中增加开发人员只会让它拖得更久，他声称项目的复杂度和通讯开销以开发人员的平方增长，而工作成绩只是以线性增长，这个说法被称为“Brooks定律”，被普遍当作真理，但如果Brooks定律就是全部，那Linux就不可能成功。

几年之后，Gerald Weinbeng的经典之作“The Psychology Of Computer Progromming”为我们更正了Brooks的看法，在他的“忘我(egoless)的编程”中，Weinberg观察到在开发人员不顽固保守自己的代码，鼓励其他人寻找错误和发展潜力的地方，软件的改进的速度会比其他地方有戏剧性的提高。

Weinberg的用词可阻止了他的分析得到应有的接受，人们对把Internet黑客称为“忘我”的想法微笑，但是我想今天他的想法比以往任何时候都要引人注目。

Unix的历史已经为我们准备好了我们正在从Linux学到的(和我在更小规模上模仿Linus的方法所验证的)东西，这就是，虽然编码仍是一个人干的活，真正伟大的工作来自于利用整个社团的注意和脑力，在一个封闭的项目中只利用他自己的脑力的人会落在知道怎样创建一个开放的、进化的，成百上千的人在其中查找错误和进行修改的环境的开发人员之后。

但是Unix的传统中有几个因素阻止把这种方法推到极致。一个是各种授权的法律约束、商业机密和商业利益，另一个(事后来看)是Internet还不够好。

在Internet变得便宜之前，有一些在地理上紧密的社团，它们的文化鼓励Weingberg的“忘我”编程，一个开发人员很容易吸引许多熟练的人和协作开发人员，贝尔实验室，MIT A1实验室，UC Berkeley，都成为传统的、今天仍然是革新的源泉。

Linux是第一个有意识的成功的利用整个世界做为它的头脑库的项目，我不认为Linux的孕育和万维网的诞生相一致是一个巧合，而且Linux在1993-1994的一段ISP工业大发展和对Internet的兴趣爆炸式增长的时期中成长起来，Linus是第一个学会怎样利用Internet的新规的人。

廉价的Internet对Linux模式的演化来说是一个必要条件，但它并不充分，另一个关键因素是领导风格的开发和一套协作的氛围使开发人员可以吸引协作开发人员和最大限度地利用媒体。

但是这种领导风格与氛围到底是什么呢?它不能建立在权力关系之上——甚至如果它们可以，高压的领导权力不能产生我们所看到的结果，Weinberg引用了19世纪俄国的无政府主义者Kropotkin的“Memoris of a Revolutionist”来证明这个观点：

“我从小生活在一个农奴主的家庭中，我有一个活跃的生活，象我们时代的所有年轻人一样，我深信命令、强制、责骂、惩罚等等的必要性。但是当我(在早期)必须管理一个企业，和(自由)人打交道时，当每一个错误都会产生严重后果时，我开始接受以命令和纪律为准则来行动和以普通理解为准则来行动的区别。前者在军事阅兵中工作的很好，但是它在现实生活中一文不值，目标达成只是靠许多愿望的聚合的简单后果。”“许多聚合在一起的愿望的直接后果”精确地指出了象Linux的项目所需要的东西。“命令的准则”在Internet这种无政府主义的天堂中一群自愿者之中是没有市场的，为了更有效的操作和竞争，想领导协作项目的黑客们必须学会怎样以Kropotkins含糊指出的“理解的准则”模式来恢复和激活社团的力量，他们必须学会使用Linus定律。

前面我引用“Delhpi效应”来作为Linus定律的一个可能的解释，但是来自生物学和经常学的自适应系统的更强大的分析也提出了自己的解释，Linus世界的行为更象一个自由市场或生态系统，由一大群自私的个体组成，它们试图取得(自己)最大的实效，在这个过程中产生了比任何一种中央计划都细致和高效的自发的改进的结果，所以，这里就是寻找“理解的准则”的地方。

Linux黑客取得的最大化的“实际利益”不是经典的经济利益，而是无

形的他们的自我满足和在其他黑客中的声望，(有人会说他们的动机是“利他的”，但这忽略了这样的事实：利他主义本身是利他主义者的一种自我满足的形式)，自愿的文化以这种方式工作的实际上并非不寻常，我已参与一个科幻迷团体很长时间了，它不象黑客团体一样，显式地识别出“egoboo”(一个人在其他爱好者之中的声望的增长)作为自愿者活动背后的基础驱动力)。

Linus成功地把自己置于项目的守门人的位置，在项目中开发大部分是别人做的，他只是在项目中培养兴趣直到它可以自己发展下去，这为我们展示了对Kropokin的“共同理解原则”的敏锐把握，对Linux这种类似经济学的观点让我们看到这种理解是怎样应用的。

我们可以把Linus的方法视为创建一个高效的关于“egoboo”(而不是钱)的市场，来把自私的黑客个体尽可能紧密的联系起来，达成只能通过高度协作才能得到的困难的结果，在fetchmail项目中我展示了(在较小规模上)这种模式可以复制，得到良好的结果，也许我比他更有意识一点、更加系统一点。

许多人(尤其是哪些由于政治原因不信任自由市场的人)会盼望自我导向的自我主义者的文化破碎、报废、秘密和敌对，但这种盼望很明显地被Linux的文档的多样性、质量和深度打破了，程序员讨厌写文档似乎已是圣训，但Linux的黑客们怎么产生了这么多?显然Linux的egoboo自由市场比有大量资金的商业软件产品的文档部在产生有品德的、他人导向的行为方面工作的更好。

Fetchmail和Linux内核项目都表明，通过恰当的表彰许多其他黑客，一个强大的开发者 / 协调者可以用Internet得到许多协同开发人员而不是让项目分崩离析为一片混乱，所以关于Brooks定律我得到了下面的想法：

19. 如果开发协调人员有至少和Internet一样好的媒介，而且知道怎样不通过强迫来领导，许多头脑将不可避免地比一个好。

我认为自由软件的将来将属于那些知道怎样玩Linus的游戏的人，把大教堂抛之脑后拥抱市集的人，这并不是说个人的观点与才气不再重要，而是，我认为自由软件的前沿将属于从个人观点和才气出发的人，然后通过共同兴趣自愿社团的高效建造来扩展。

可能不只是自由软件的将来，在解决问题方面，没有任何商业性开发者可以与Linux社团的头脑库相匹敌，很少有人能负担起雇佣200多个为fetchmail出过力的人!

也许最终自由软件文化将胜利，不是因为协作在道德上是正确的或软件“囤积居奇”在道德上是错的(假设你相信后者，Linus和我都不)，而仅仅是因为商业世界在进化的军备竞赛中不能战胜自由软件社团，因为后者可以把更大更好的开发资源放在解决问题上。