

▶▶ 您现在的位置：『岁月如歌』 >> 文章中心 >> 电脑技术 >> 系统应用 >> 正文

▶ [推荐]linux完全教学手册

荐

【字体：小 大】

linux完全教学手册

作者：佚名 文章来源：本站原创 点击数：275 更新时间：2005-3-28

目 录

第一部分：系统

第一节课：Linux的发展、Linux的安装

第二节课：常用命令、管道与重定向

第三节课：vi的使用、用户配置

第四节课：文件系统

第五节课：系统管理（一）

第六节课：系统管理（二）

第二部分：网络服务

第七节课：DHCP服务器的配置

第八节课：DNS服务器的配置

第九节课：smb服务器的配置

第十节课：apache的配置

第十一节课：sendmail的配置

第十二节课：x-window的配置

第三部分：优化与安全

第十三节课：内核的设置与编译

第十四节课：包过滤防火墙与课程总结

第一节：Linux的发展、Linux的安装

一、Linux的定义

1、什么是Linux?

Linux 发音为 "Lin-noks".

Linux 的官方定义：“Linux 是一种 UNIX 操作系统的克隆，它（的内核）由 Linus Torvalds 以及网络上组织松散的黑客队伍一起从零开始编写而成。Linux 的目标是保持和 POSIX 的兼容。”

2、Linux的特点：

A、“Linux 具备现代一切功能完整的 UNIX 系统所具备的全部特征，其中包括真正的多任务、虚拟内存、共享库、需求装载、共享的写时复制程序执行、优秀的内存管理以及 TCP/IP 网络支持等。

B、“Linux 的发行遵守 GNU 的通用公共许可证(GPL)。

C、在原代码级上兼容绝大部分的Unix标准(如IEEE POSIX，System V，BSD)，它遵从POSIX规范，例如对于System V来说,把其上程序源代码拿到Linux下重新编译后就可以运行，而对于BSD Unix来说，它的可执行文件可以直接在Linux环境下运行。

更进一步地了解Linux, 请浏览下列站点:

<http://www.linux.org>

<http://www.linuxresources.com/what.html>

二、Linux的发展

1、UNIX的发展：

1969年由Ken Thompson在AT&T贝尔实验室的一台DEC PDP-7计算机上实现。后来Ken Thompson和Dennis Ritchie使用C语言对整个系统进行了再加工和编写，使得Unix能够很容易的移植到其他硬件的计算机上。由于此时AT&T还没有把Unix作为它的正式商品，因此研究人员只是在实验室内使用并完善它。正是由于Unix是被作为研究项目，其他科研机构 and 大学的计算机研究人员也希望能得到这个系统，以便进行自己的研究。AT&T以分发许可证的方法，对Unix仅仅收取很少的费用，大学和研究机构就能获得Unix的源代码以进行研究。Unix的源代码被散发到各个大学，一方面使得科研人员能够根据需要改进系统，或者将其移植到其他的硬件环境中去，另一方面培养了懂得Unix使用和编程的大量的学生，这使得Unix的普及更为广泛。

到了70年代末，在Unix发展到了版本6之后，由于AT&T认识到了Unix的价值，成立了Unix系统实验室（Unix System Lab,USL）来继续发展Unix。因此AT&T一方面继续发展内部使用的Unix版本7，一方面由USL开发对外正式发行的Unix版本，同时AT&T也宣布对Unix产品拥有所有权。几乎在同时，加州大学伯克利分校计算机系统研究小组（CSRG）使用Unix对操作系统进行研究，他们对Unix的改进相当多，增加了很多当时非常先进的特性，包括更好的内存管理，快速且健壮的文件系统等，大部分原有的源代码都被重新写过，很多其他Unix使用者，包括其他大学和商业机构，都希望能得到CSRG改进的Unix系统。因此CSRG中的研究人员把他们的Unix组成一个完整的Unix系统——BSD Unix（Berkeley Software Distribution）向外发行。

而AT&T的Unix系统实验室，同时也在不断改进他们的商用Unix版本，直到他们吸收了BSD Unix中已有的各种先进特性，并结合其本身的特点，推出了Unix System V版本从此以后，BSD Unix和Unix System V形成了当今Unix的两大主流，现代的Unix版本大部分都是这两个版本的衍生产品。如IBM的AIX4.0、HP/UX11、SCO的UnixWare等属于System V，而Minix、freeBSD、NetBSD、OpenBSD等属于BSD Unix。

2、Linux的发展历史

Linux的历史可以追溯到1990年，芬兰赫尔辛基大学的一名学生Linus Torvalds，最初用汇编语言写了一个在80386保护模式下处理多任务切换的程序，后来从Minix(用于操作系统教学、很小的Unix)中得到灵感，发誓要写一个比Minix更好的Minix，于是开始写了一些硬件的设备驱动程序、一个小的文件系统……，这样0.0.1版本的Linux就出来了，但是它必须在有Minix的机器上编译以后才能玩，这时候的Linus已经完全“走火入魔”了，决定踢开Minix“闹革命”，于是在1991年10月5号发布了Linux 0.0.2版本，这个版本已经可以运行bash(一种用户与操作系统内核通讯的软件)和gcc(GNU C编译器)了。

Linus从一开始，就决定自由扩散Linux、包括源代码，他把源代码发布在网上，随即就引起爱好者的注意，他们通过互连网也加入了Linux的内核开发工作，一大批高水平程序员的加入，使得Linux达到迅猛发展，到1993年底，Linux 1.0终于诞生。Linux 1.0已经是一个功能完备的操作系统了，其内核写得紧凑高效，可以充分发挥硬件的性能，在4M内存的80386机器上也表现得非常好。

很多人对Linux的认识有个误区，即总把Linux与低档硬件平台联系到一起，其实从2.1.xx系列内核开始，Linux就开始走高端的路子了，大约在1.3版本之后，开始向其他硬件平台上移植，包括号称最快的CPU---Digital Alpha(目前主频是最高的)，目前Linux能将硬件的性能充分发挥出来，可以囊括低端到高端的所有应用。

Linux加入GNU并遵循公共版权许可证(GPL)，由于不排斥商家对自由软件进一步开发，不排斥在Linux上开发商业软件，故而使Linux又开始了一次飞跃，出现了很多的Linux发行版，如Slackware、Redhat、TurboLinux、OpenLinux等十多种，而且还在增加，还有一些公司在Linux上开发商业软件或把其他Unix平台的软件移植到Linux上来，如今很多IT界的大腕如IBM、Intel、Oracle、Informix、Sysbase、、Netscape、Novell等都宣布支持Linux! 商家的加盟弥补了纯自由软件的不足和发展障碍，Linux得以迅速普及。

3、Linux 的主要发行版

目前比较流行的版本有：Slackware、OpenLinux、Mandrake、Red Hat Linux等等。建

简体中文 Linux 发行版：BluePoint Linux：最新发布的 Linux 中文版。利用 Linux 2.2 内核的 FrameBuffer，可在控制台获得中文输入输出。具备多内码支持，目前可以支持大陆国标码和港台大五码。与 RedHat Linux 兼容。

TurboLinux：国内最早的简体中文发行版之一。

XteamLinux：带有较好的图形配置工具googe。

4、几个常用术语

A、POSIX 及其重要地位

POSIX 表示可移植操作系统接口（Portable Operating System Interface，缩写为 POSIX 是为了读音更像 UNIX）。由电气和电子工程师协会（Institute of Electrical and Electronics Engineers，IEEE）开发，是为了提高 UNIX 环境下应用程序的可移植性。然而，POSIX 并不局限于 UNIX。许多其它的操作系统，例如 DEC OpenVMS 和 Microsoft Windows NT，都支持 POSIX 标准，尤其是 IEEE Std. 1003.1-1990（1995 年修订）或 POSIX.1，POSIX.1 提供了源代码级别的 C 语言应用编程接口（API）给操作系统的服务程序，例如读写文件。POSIX.1 已经被国际标准化组织（International Standards Organization，ISO）所接受，被命名为 ISO/IEC 9945-1:1990 标准。

POSIX 现在已经发展成为一个非常庞大的标准族，某些部分正处在开发过程中。

B、GNU 和 Linux 的关系

GNU 是 GNU Is Not UNIX 的递归缩写，是自由软件基金会的一个项目，该项目的目标是开发一个自由的 UNIX 版本，这一 UNIX 版本称为 HURD。尽管 HURD 尚未完成，但 GNU 项目已经开发了许多高质量的编程工具，包括 emacs 编辑器、著名的 GNU C 和 C++ 编译器（gcc 和 g++），这些编译器可以在任何计算机上运行。所有的 GNU 软件和派生工作均适用 GNU 通用公共许可证，即 GPL。GPL 允许软件作者拥有软件版权，但授予其他任何人以合法复制、发行和修改软件的权利。

Linux 的开发使用了许多 GNU 工具。Linux 系统上用于实现 POSIX.2 标准的工具几乎都是 GNU 项目开发的，Linux 内核、GNU 工具以及其他一些自由软件组成了人们常说的 Linux：C 语言编译器和其他开发工具及函数库。

X Window 窗口系统。

各种应用软件，包括字处理软件、图象处理软件等。

其他各种 Internet 软件，包括 FTP 服务器、WWW 服务器等。

关系数据库管理系统等。

C、GPL (General Public License) 公共许可协议

GPL 的文本保存在 Linux 系统的不同目录下的命名为 COPYING 的文件里。例如，键入 `cd /usr/doc/ghostscript*` 然后再键入 `more COPYING` 可查看 GPL 的内容。

GPL 和软件是否免费无关，它主要目标是保证软件对所有的用户来说是自由的。GPL 通过如下途径实现这一目标：

- 它要求软件以源代码的形式发布，并规定任何用户能够以源代码的形式将软件复制或发布给别的用户。
- 它提醒每个用户，对于该软件不提供任何形式的担保。
- 如果用户的软件使用了受 GPL 保护的软件的一部分，那么该软件就继承了 GPL 软件，并因此而成为 GPL 软件，也就是说必须随应用程序一起发布源代码。
- GPL 并不排斥对自由软件进行商业性质的包装和发行，也不限制在自由软件的基础上打包发行其他非自由软件。

遵照 GPL 的软件并不是可以任意传播的，这些软件通常都有正式的版权，GPL 在发布软件或者复制软件时声明限制条件。但是，从用户的角度考虑，这些根本不能算是限制条件，相反用户只会从中受益，因为用户可以确保获得源代码。

尽管 Linux 内核也属于 GPL 范畴，但 GPL 并不适用于通过系统调用而使用内核服务的应用程序，通常把这种应用程序看作是内核的正常使用。

假如准备以二进制的形式发布应用程序（像大多数商业软件那样），则必须确保自己的程序未使用 GPL 保护的软件。如果软件通过库函数调用而使用了别的软件，则不必受到这一限制。大多数函数库，受另一种 GNU 公共许可证，即 LGPL 的保护，将在下面介绍。

D、LGPL (Library General Public License)

GNU LGPL（GNU 程序库公共许可证）的内容全部包括在命名为 COPYING.LIB 的文件中。如果安装了内核的源程序，在任意一个源程序的目

录下都可以找到 COPYING.LIB 文件的一个拷贝。

LGPL 允许在自己的应用程序中使用程序库，即使不公开自己的源代码。但是，LGPL 还规定，用户必须能够获得在应用程序中使用的程序库的源代码，并且允许用户对这些程序库进行修改。

大多数 Linux 程序库，包括 C 程序库（libc.a）都属于 LGPL 范畴。因此，如果在 Linux 环境下，使使用 GCC 编译器建立自己的应用程序，程序所链接的多数程序库是受 LGPL 保护的。如果想以二进制的形式发布自己的应用程序，则必须注意遵循 LGPL 有关规定。

遵循 LGPL 的一种方法是，随应用程序一起发布目标代码，以及可以将这些目标程序和受 LGPL 保护的、更新的 Linux 程序库链接起来的 makefile 文件。

遵循 LGPL 的比较好的一种方法是使用动态链接。使用动态链接时，即使是程序在运行中调用函数库中的函数时，应用程序本身和函数库也是不同的实体。通过动态链接，用户可以直接使用更新后的函数库，而不用对应用程序进行重新链接。

但我们必须当心个别属于 GPL 范畴的库和实用程序：GNU dbm（即 gdbm）数据库类的程序库是非常著名的 GPL 库；GNU bison 分析器生成程序是另一个实用的 GPL 工具，如果使用 bison 生成代码，所得的代码也适用于 GPL。

在 GPL 的保护范围以外，也有 GNU dbm 和 GNU bison 的相应的替代程序。例如，对于数据库类的程序库，可以使用 Berkeley 数据库 db 来替代 gdbm；对于分析器生成器，可以使用 yacc 来替代 bison。

5、Linux 发展的重要里程碑

1990, Linus Torvalds 首次接触 MINIX

1991 中, Linus Torvalds 开始在 MINIX 上编写各种驱动程序等操作系统内核组件

1991 底, Linus Torvalds 公开了 Linux 内核

1993, Linux 1.0 版发行，Linux 转向 GPL 版权协议

1994, Linux 的第一个商业发行版 Slackware 问世????

1996, 美国国家标准技术局的计算机系统实验室确认 Linux 版本 1.2.13（由 Open Linux 公司打包）符合 POSIX 标准

1999, Linux 的简体中文发行版相继问世

2000 中, LinuxWorld China 2000 展览会召开，涌现大量基于 Linux 的嵌入式系统。

三、linux的安装（RedHat6.2的安装）

1、安装前的准备（了解你的硬件）

CPU C733

内存 256m

硬盘 希捷20G

显卡 TNT2 M64/16M

主板 技嘉694X

网卡 RTL8029

- RedHat6.2的安装光盘（带启动）
- 一张空白软盘

二、开始安装：

（1）、进入CMOS设置重光盘引导：

（2）、选择安装方式

缺省状态为图形安装方式（直接按ENTER键），我们采用文本安装方式：

输入text，然后按ENTER键。

（3）、选择语言：(US English)

这一步对大家都不陌生了，首先要选择自己需要的语言。或者选择和自己国家相近的语言也可以。建议大家使用US English，学习Red Hat Linux 还

是从英文版本开始学习比较正规，因为Red Hat Linux 里面很多程序都是英文的。缺省是没有中文支持的。

(4)、选择键盘布局：(选择us)

(5)、欢迎画面：

(6)、选择安装类型：

建议选择定制安装 (Install Custom System)，最后一项为升级安装。其它三项

Install GNOME Workstation

Install KDE Workstation

Install Server System

都不能对硬盘进行分区，安装程序自动创建分区。

(7)、选择分区方式：(FDISK)

(8)、硬盘进行分区：

A、先按m键查看帮助，按p查看分区信息：

B、按n键创建分区，并选择p创建主分区：

输入分区编号：1 (表示是第一个分区)，回车后输入起始柱面数，直接回车即可。输入分区容量2G (+2000M) 或结束柱面数。

同理创建第二个分区，容量为256M，作为交换分区。并用t命令改变分区类型为82 (swap)：

先输入l命令查看列表，找到linux swap的ID号为82，再输入82回车即可。

C、a命令设置引导标志位

再用p命令查看时，在boot列已有*号

D、用w命令保存并退出。

补充说明：

(1)、linux内物理硬盘的表示方法：

IDE硬盘 SCSI硬盘

Primary master hda SCSI : 0 sda

Primary slave hdb SCSI : 1 sdb

Secondary master hdc SCSI : 2 sdc

Secondary slave hdd SCSI : 3 sdd

SCSI : n ...

2)、linux内分区的表示方法：

hda1——hda4表示主分区或扩展分区，hda5——hda...表示扩展分区。

一个硬盘最多可以分4个主分区或3个主分区加一个扩展分区，扩展分区可以再分若干个逻辑分区。

(3)、linux对分区的要求：

安装linux时至少给硬盘分两个区，一个linux native (linux分区)、一个linux swap (交换分区)，交换分区一般为内存的1.5——2倍，

(7) 挂结分区：

(8) 格式化硬盘分区

(9) lilo的设置 (可以不用理会)

(10) lilo的安装位置 (推荐为MBR)

(11) linux安装到那个分区 (hda1)

(12) 输入主机名

(13) 设置IP地址、网关、DNS

(14) 选择鼠标驱动(如是两键鼠标，打开模拟三键 (Emulate 3 Buttons))

(15) 设置时区(选择China/Beijing)

(16) 设置root口令

- (17) 新建用户（可以不建）
- (18) 选择加密方式（缺省）
- (19) 选择安装组件
- (20) 如果安装了X-window，则选择显卡
- (21) 开始复制文件
- (22) 是否创建引导软盘（推荐创建）
- (23) 安装完毕（启出光盘，计算机自动从启）

第二节：常用命令、管道与重定向

一、常用命令：

二、标准输入/输出和重定向

1、标准输入与输出

我们知道，执行一个shell命令行时通常会自动打开三个标准文件，即标准输入文件（stdin），通常对应终端的键盘；标准输出文件（stdout）和标准错误输出文件（stderr），这两个文件都对应终端的屏幕。进程将从标准输入文件中得到输入数据，将正常输出数据输出到标准输出文件，而将错误信息送到标准错误文件中。我们以cat命令为例，cat命令的功能是从命令行给出的文件中读取数据，并将这些数据直接送到标准输出。若使用如下命令：

```
$ cat config
```

将会把文件config的内容依次显示到屏幕上。但是，如果cat的命令行中没有参数，它就会从标准输入中读取数据，并将其送到标准输出。例如：

```
$ cat
```

```
Hello world
```

```
Hello world
```

```
Bye
```

```
Bye
```

```
$
```

用户输入的每一行都立刻被cat命令输出到屏幕上。

另一个例子，命令sort按行读入文件正文（当命令行中没有给出文件名时，表示从标准输入读入），将其排序，并将结果送到标准输出。下面的例子是从标准输入读入一个采购单，并将其排序。

```
$ sort
```

```
bananas
```

```
carrots
```

```
apples
```

```
apples
```

```
bananas
```

```
carrots
```

```
$
```

这时我们在屏幕上得到了已排序的采购单。

直接使用标准输入/输出文件存在以下问题：

输入数据从终端输入时，用户费了半天劲输入的数据只能用一次。下次再想用这些数据时就得重新输入。而且在终端上输入时，若输入有误修改起来不是很方便。

输出到终端屏幕上的信息只能看不能动。我们无法对此输出作更多处理，如将输出作为另一命令的输入进行进一步的处理等。

为了解决上述问题，Linux系统为输入、输出的传送引入了另外两种机制，即输入/输出重定向和管道。

输入重定向：

输入重定向是指把命令（或可执行程序）的标准输入重定向到指定的文件中。也就是说，输入可以来自键盘，而来自一个指定的文件。所以说，输入重定向主要用于改变一个命令的输入源，特别是改变那些需要大量输入的输入源。

例如，命令wc统计指定文件包含的行数、单词数和字符数。如果仅在命令行上键入：

```
$ wc
```

wc将等待用户告诉它统计什么，这时shell就好象死了一样，从键盘键入的所有文本都出现在屏幕上，但并没有什么结果，直至按下 <ctrl+d>，wc才将命令结果写在屏幕上。

如果给出一个文件名作为wc命令的参数，如下例所示，wc将返回该文件所包含的行数、单词数和字符数。

```
$ wc /etc/passwd
```

```
20 23 726 /etc/passwd
```

```
$
```

另一种把/etc/passwd文件内容传给wc命令的方法是重定向wc的输入。输入重定向的一般形式为：命令<文件名。可以用下面的命令把wc命令的输入重定向为/etc/passwd文件：

```
$ wc < /etc/passwd
```

```
20 23 726
```

```
$
```

另一种输入重定向称为here文档（从这里开始，到这里结束），它告诉shell当前命令的标准输入来自命令行。here文档的重定向操作符使用<<。它将一对分隔符（本例中用delim表示）之间的正文重定向输入给命令。下例将一对分隔符delim之间的正文作为wc命令的输入，统计出正文的行数、单词数和字符数。

```
$ wc<<DELIM
```

```
>this text forms the content
```

```
>of the here document,which
```

```
>continues until the end of
```

```
>text delimiter
```

```
>delim
```

```
4 17 98
```

在<<操作符后面,任何字符都可以作为正文开始前的分隔符，本例中使用delim作为分隔符。here文档的正文一直延续到遇见另一个分隔符为止。第二个分隔符应出现在新行的开头。这时here文档的正文（不包括开始和结束的分隔符）将重新定向送给命令wc作为它的标准输入。

由于大多数命令都以参数的形式在命令行上指定输入文件的文件名，所以输入重定向并不经常使用。尽管如此，当要使用一个不接受文件名作为输入参数的命令，而需要的输入内容又存在一个文件里时，就能用输入重定向解决问题。

输出重定向：

输出重定向是指把命令（或可执行程序）的标准输出或标准错误输出重新定向到指定文件中。这样，该命令的输出就不显示在屏幕上，而是写入到指定文件中。

输出重定向比输入重定向更常用，很多情况下都可以使用这种功能。例如，如果某个命令的输出很多，在屏幕上不能完全显示，那么将输出重定向到一个文件中，然后再用文本编辑器打开这个文件，就可以查看输出信息；如果想保存一个命令的输出，也可以使用这种方法。还有，输出重定向可以用于把一个命令的输出当作另一个命令的输入（还有一种更简单的方法，就是使用管道，将在下面介绍）。

输出重定向的一般形式为：命令>文件名。例如：

```
$ ls > directory.out
```

```
$ cat directory.out
ch1.doc ch2.doc ch3.doc chimp config mail/ test/
$
```

将ls命令的输出保存为一个名为directory.out的文件。

注：如果>符号后边的文件已存在，那么这个文件将被重写。

为避免输出重定向中指定文件只能存放当前命令的输出重定向的内容，shell提供了输出重定向的一种追加手段。输出追加重定向与输出重定向的功能非常相似，区别仅在于输出追加重定向的功能是把命令（或可执行程序）的输出结果追加到指定文件的最后，而该文件原有内容不被破坏。如果要将一条命令的输出结果追加到指定文件的后面，可以使用追加重定向操作符>>。形式为：命令>>文件名。例如：

```
$ ls *.doc>>directory.out
$ cat directory.out
ch1.doc ch2.doc ch3.doc chimp config mail/ test/
ch1.doc ch2.doc ch3.doc
$
```

错误输出重定向：

和程序的标准输出重定向一样，程序的错误输出也可以重新定向。使用符号2>（或追加符号2>>）表示对错误输出设备重定向。例如下面的命令：

```
$ ls /usr/tmp 2> err.file
```

可在屏幕上看到程序的正常输出结果，但又将程序的任何错误信息送到文件err.file中，以备将来检查用。

还可以使用另一个输出重定向操作符（&>）将标准输出和错误输出同时送到同一文件中。例如：

```
$ ls /usr/tmp &> output.file
```

利用重定向将命令组合在一起，可实现系统单个命令不能提供的新功能。例如使用下面的命令序列：

```
$ ls /usr/bin > /tmp/dir
```

```
$ wc -w < /tmp/dir
```

```
459
```

统计了/usr/bin目录下的文件个数。

2、管道：

将一个程序或命令的输出作为另一个程序或命令的输入，有两种方法，一种是通过一个临时文件将两个命令或程序结合在一起，例如上个例子中的/tmp/dir文件将ls和wc命令联在一起；另一种是Linux所提供的管道功能。这种方法比前一种方法更好。

管道可以把一系列命令连接起来，这意味着第一个命令的输出会作为第二个命令的输入通过管道传给第二个命令，第二个命令的输出又会作为第三个命令的输入，以此类推。显示在屏幕上的是管道行中最后一个命令的输出（如果命令行中未使用输出重定向）。

通过使用管道符“|”来建立一个管道行。用管道重写上面的例子：

```
$ ls /usr/bin|wc -w
```

```
1789
```

再如：

```
$ cat sample.txt|grep "High"|wc -l
```

管道将cat命令（列出一个文件的内容）的输出送给grep命令。grep命令在输入里查找单词High，grep命令的输出则是所有包含单词High的行，这个输出又被送给wc命令，wc命令统计出输入中的行数。假设sample.txt文件的内容如下：

```
Things to do today:
```

```
Low:Go grocery shopping
```


High:Return movie

High:Clear level 3 in Alien vs. Predator

Medium:Pick up clothes from dry cleaner

那么该管道行的结果是2。

命令替换：

命令替换和重定向有些相似，但区别在于命令替换是将一个命令的输出作为另外一个命令的参数。常用命令格式为：

`command1 `command2``

其中，`command2`的输出将作为`command1`的参数。需要注意的是这里的`符号，被它括起来的内容将作为命令执行，执行后的结果作为`command1`的参数。例如：

`$ cd `pwd``

该命令将`pwd`命令列出的目录作为`cd`命令的参数，结果仍然是停留在当前目录下。

第三节课：vi的使用、用户配置

一、vi常用命令参考：

命令方式：

功能键 功能

:q 在未做修改的情况下退出

:q! 将任何修改作废，退出

:w 存盘

:wq 存盘并退出

:w file 将编辑的内容写到 file 文件中

:r file 将 file 文件读入，并放在当前光标行的后面

:e file 编辑新文件 file 代替旧内容

:f file 将当前正文的名称改为 file

:f 打印当前正文的名称和状态

:n 移动光标到第 n 行

:/str/ 将光标移动到下一个包含 str 字符串的行

?str? 将光标移动到最近的一个包含 str 字符串的行

:/str/w file 将第一个包含 str 字符串的行写到文件 file 中

:/str1/,/str2/w file 将从包含 str1 到 包含 str2 之间的行写入 file 文件

:s/str1/str2 用 str2 替换掉首次出现的 str1

:s/str1/str2/g 用 str2 替换掉每次出现的 str1

:d 删除当前光标行

插入方式：

功能键 功能

i 在光标左面插入正文

a 在光标右面插入正文

o 在光标所在行下面增加行

O 在光标所在行上面增加行

I 在光标行的开头插入
A 在光标行的末尾插入
s 用新的正文替换光标所指字符
cw 用新的正文替换光标右边的字
cb 用新的正文替换光标前边的字
cd 用新的正文替换光标所在的行
c\$ 用新正文替换从光标开始到本行末尾的所有字符
c0 用新正文替换从光标开始到本行开头的所有字符

编辑方式：

功能键 功能

ctrl-f 前翻一页

ctrl-b 后翻一页

H 将光标移动到屏幕最上面一行

M 将光标移动到屏幕中间

L 将光标移动到屏幕最下面一行

w 右移光标到下一个字的开头

e 右移光标到一个字的末尾

b 左移光标到前一个字的开头

0 左移光标到到本行的开始

^ 移动光标到行中第一个非空字符

\$ 右移光标到本行末尾

/string 向前搜索指定的字符串

? string 向后搜索指定的字符串

n 搜索下一个字符串

rc 用 c 替换当前光标处的字符

x 删除当前光标处的字符

dw 删除光标右面的字

db 删除光标左面的字

dd 删除光标所在行

d\$ 删除当前光标到行末尾之间的字符

d0 从当前光标起删除字符直到行的开始

J 删除本行的回车，并和下一行合并

p 将缓冲区内的内容粘贴到当前光标的后面

P 将缓冲区内的内容粘贴到当前光标的前面

yy 将当前行复制到剪切缓冲区

nyy 将n行复制到剪切缓冲区

u 撤消前一命令的结果

. 重复最后一条修改正文的命令

二、用户配置

第四节课：文件系统

文件系统形成了 Linux系统上所有数据的基础，Linux程序、库、系统文件和用户文件都驻留在文件系统上，因此对文件系统的适当管理是很重要的。

一、文件系统的目录结构

在 Linux下，用户所见到的文件空间是基于树状结构的，树的根在顶部。在这个空间中的各种目录和文件从树根向下分支，顶层目录(/)被称为根目录，

Linux操作系统由一些目录和许多文件组成。根据用户选择的安装不同，这些目录可能是不同的文件系统。通常，大多数操作系统都驻存在两个文件系统上：即称为/的根文件系统和安装在/usr下的文件系统。

如果你用 cd /命令将当前目录改变到根目录，并用ls命令列出目录清单，你就会看到一些目录。这些目录组成了根文件系统的内容，它们也为其他文件系统提供了安装点。

/bin目录包含称为二进制 (binary)文件的可执行程序 (事实上，名为/bin的目录是 binary的缩写)。这些程序是必需的系统文件，许多 Linux命令(如 ls)放在该目录中。

/sbin目录也用于存储系统二进制文件。这个目录中的大多数文件用于管理系统。

/etc目录非常重要，它包含许多Linux系统配置文件。从本质上说，这些文件使你的Linux系统具有自己的个性。口令文件(口令)就放在这里，在启动时安装的文件系统列表(fstab)也放在这里。另外，这个目录还包括 Linux的启动脚本、你想要永久记录的、带 IP地址的主机列表和许多其他类型的配置信息。

/lib目录中存储着程序运行时使用的共享库被存储在此。通过共享库，许多程序可以重复使用相同的代码，并且这些库可以存储在一个公共的位置上，因此能减小运行程序的大小。

/dev目录包含称为设备文件的特殊文件，这些文件用于访问系统上所有不同类型的硬件。例如，/dev/mouse文件是用于读取鼠标输入的。通过用这种方法组织对硬件设备的访问，Linux有效地使硬件设备的接口看起来就象一个文件。这意味着在许多情况下，你可以用对软件使用的相同语法来对计算机的硬设备进行操作。例如，为了在软盘驱动器上建立你的起始目录的磁盘档案，你可以使用下面的命令：

```
tar-cdf/dev/fd0~ tackett
```

/dev目录中的许多设备都放在逻辑组中，表列出了/dev 目录中一些最常用的设备。

设备文件 描述

/dev/console 系统控制台，它是与Linuxmso-hansi-font-family: "Times New Roman">系统实际连接的计算机显示器

/dev/hd IDE硬盘驱动器的设备驱动程序接口。/dev/hda设备指在硬盘驱动hda上的第一个分区，设备/dev/hda指整个硬盘 had。

/dev/sd SCSI磁盘的设备驱动程序接口。SCSI磁盘和分区使用与IDE/dev/hd设备相同的约定。

/dev/fd 提供支持软盘的设备驱动程序。/dev/fd0是第一个软盘驱动器，/dev/fd1是第二个软盘驱动器。

/dev/st SCSI磁带驱动器的设备驱动程序

/dev/tty 为用户输入提供不同控制台的设备驱动程序。它的名字来源于实际挂接到 UNIX系统的、被称为电传打字机(teletype)的终端。在Linux下，这些文件提供对虚拟控制台的支持，可以通过按 < Alt-F1 > 到 < Alt-F6 > 键来访问这些虚拟控制台。这些虚拟控制台提供独立的、同时进行的本地登录对话过程

/dev/pty 支持伪终端的设备驱动程序，伪终端用于远程登录进程，如使用Telnet的远程登录

/dev/ttyS 计算机的串行接口。/dev/ttySO对应MS-DOS下的 COM1。如果有一个串行鼠标，/dev/mouse是对连接了鼠标的ttyS设备的符号链接。

/dev/cua 与调制解调器一起使用的特殊呼出设备。

/dev/null 数据将永远丢失。如果你想要运行一个命令并丢弃标准输出或标准错误输出，那么这个设备是非常有用的。而且，如果用/dev/null作为一个输入文件，则会建立一个长度为零的文件。

/dev目录中一些最常用的设备

/proc 目录实际上是一个虚拟文件系统，它被用于从内存中读取处理信息。

/tmp目录用于存储程序运行时生成的临时文件。如果你有一个程序，它会生成许多大的临时文件，那么你可能想把/tmp目录作为一个独立文件系

统来安装，而不是只把它作为根文件系统上的一个目录。如果把/tmp作为根文件系统上的一个目录，并且有许多大文件要写入其中，那么根文件系统的空间就会用尽。

/home目录是用户起始目录的基础目录。通常把它作为一个独立的文件系统来安装，这样用户就能有大量用于自己文件的空间。事实上，如果系统上有许多用户，则可能需要把/home分成几个文件系统。为此，需要建立子目录，如用/home/staff和/home/admin目录分别作为公司职员和管理人员的子目录。把每一个这样的子目录做为不同的文件系统来安装，然后在其下建立用户的起始目录。

/var目录保存要随时改变大小的文件。通常，各种系统记录文件都放在这个目录下。

/usr目录及其子目录对Linux系统的操作是非常重要的。它包含这样一些目录，这些目录中保存系统上的一些最重要的程序。通常，/usr的子目录包含你安装的大型软件包。

子目录 描述

/usr/bin 这个目录保存许多在Linux系统上能找到的可执行程序。

/usr/etc 这目录包含许多各种各样的系统配置文件。

/usr/include 这个目录及其子目录是存放C编译程序的所有包含文件的地方。这些头文件定义常量和函数，对C编程是重要的。

/usr/g++-include 这个目录包含C++编译程序的包含文件。

/usr/lib 这个目录包含程序在链接时使用的各种库。

/usr/man 这个目录包含Linux系统上的程序的各种联机帮助。在/usr/man下是一些与联机帮助的不同段相对应的子目录。

/usr/src 这个目录包含保存系统的不同程序的源代码的目录。如果你得到了一个想要安装的软件包，那么/usr/src/packageName是安装该包前放置源代码的好地方。

/usr/local 这个目录用于存放系统的本地定制内容。通常，许多本地软件安装在这个目录的子目录中。这个目录的格式在几乎你所见到的每一个UNIX系统上都不相同。建立此目录的一种方法是：为二进制文件建立一个/usr/local/bin目录；为配置文件建立一个/usr/local/etc目录；为各种库建立一个/usr/local/lib目录；为源代码建立一个/usr/local/src目录。如果要使/usr/local具有许多空间的话，则可将整个/usr/local目录树作为一个独立的文件系统来安装。

/usr文件系统中重要的子目录

应当引起大家注意的是Linux文件系统这个词有两个不同且经常是相互矛盾的含义：一个含义是磁盘和磁盘机制的文件系统，另一个含义是用户看得见并能操作的逻辑文件系统。本章讨论用户看得见并能操作的逻辑文件系统。另外，如果用户熟悉如MS-DOS和OS/2这样的PC操作系统的话，那么就会发现下面许多话题都是熟悉的，这是因为2.0版本以上的MS-DOS的文件结构都是模仿UNIX的文件结构的，而UNIX的文件结构也是Linux使用的文件结构。

Linux中的每个实际的实体和逻辑实体在Linux文件系统中都用文件来表示。实际的实体包括磁盘、打印机和终端，逻辑实体包括目录、存储文档和程序的普通文件。

二、了解文件名和路径名

就象在MS-DOS等其他操作系统中一样，在Linux系统中必须区分文件名和路径名。一个文件名由字母、数字和某些标点符号组成的简单串组成，文件名中不能包含空格或作为字段分隔符的任何字符。例如，文件名johns . letter是正确的，而johns letter则是不正确的。文件名中不应包含任何对shell来说有特殊含义的字符。这些特殊字符如下所示：

```
!@#$%~&*()[]{}'"\|; < > `
```

另外，文件名中不能包含“/”符号，因为这个符号用于指示路径名。

早期的大多数UNIX版本把文件名限制在14个字符内，但Linux允许一个文件名有256个字符。有些最近的UNIX版本(如Berkeley版本(BSD))允许64个字符的文件名，但是只有前14个字符是有效的。由于Linux的目标之一是可移植，所以为了书写可移植程序和shell脚本，你应该把文件名限制在14个字符以内。

路径名可以有任意多个字符。在Linux中，文件不是存在于真空中的，而是存在于目录中。关于目录的知识，我们已在上一节作了比较详细的介绍。Linux中的最高层目录是根目录，用斜杠符号(/)标记，如果一个名为fred的文件存在于根目录下，它的绝对路径名就是/fred。当用adduser命令向系统添加新用户时，新用户分配到一个起始目录。按常规，该起始目录通常建在根目录下一个名为home的目录中。如果一位名为Fred的用户被分配了一个名为/home/fred的目录，那么Fred建立的所有文件都置于/home/fred目录中。Fred的某个文件的绝对路径名可能是/home/fred/freds .

file。绝对路径名准确地指出了在文件系统的什么地方能找到文件。

另一种路径名是相对路径名，它明确地指出了文件相对于当前目录的位置。如果Fred在他的起始目录中，那么文件名freds . file是一个相对于当前目录的相对路径名。为了找出哪个目录是当前目录，可以使用 pwd 命令(打印工作目录)。也可以用 echo\$PWD命令检查环境变量 \$PWD 的内容，以查看哪个目录是当前的工作目录。

通过使用两个建在所有目录下的别名，你可用相对路径名来定义在 Linux文件系统任何地方的文件。一个点(.)指当前目录；两个点(. .)指父目录。MS-DOS和OS/2使用与此相同的约定。

大多数 Linux命令按路径名操作。在大多数情况下，你使用的路径名是当前目录中的某个文件名，默认的路径名指向你的当前目录。如果Fred在他的起始目录(/home/fred)中，下面的三种方式都是等价的：

```
command freds . letter
command/home/fred/freds . letter
d . /freds . letter
```

三、Linux的文件类型

Linux只有四种基本的文件类型：普通文件、目录文件、连接文件和特殊文件。

可以用file命令来确定文件的类型，它可以识别文件类型属于可执行文件还是文本文件、数据文件等等。许多 UNIX命令只是 shell脚本，或者是与MS-DOS的批处理文件相似的解释执行的程序，可用 file命令来报告一个 UNIX命令是一个二进制的可执行程序还是一个简单的shell脚本；也可用该命令来确定文件是不是文本文件，即是否能浏览或编辑该文件。这个命令的语法是：

```
file [-vczL] [-f namefile] [-m magicfile] filelist
```

下表对file命令中的参数作了说明。

参数 描述

-c 打印输出magic文件(/usr/lib/magic)的语法分析形式，该参数是个二进制文件第一部分中的一个数值，这个数值标识了该文件的类型。它通常与-m一起使用，用于在安装一个新的magic文件之前对该文件进行调试

-z 查看压缩文件内部并试图找出该文件的类型

-L 引起后续的符号连接

-f namefile 告诉file，要识别的文件列表在文本文件namefile中。当需要识别许多文件时，该参数是很有用的

-m magicfile 指定用于确定文件类型的magic数字的一个替代文件。默认文件是/usr/lib/magic

filelist 列出想知道其类型的、以空格分隔的一组文件

1、普通文件

普通文件是用户大部分时间都在进行操作的文件。普通文件包括文本文件、C语言源代码、shell脚本(由 Linux shell解释的程序)、二进制的可执行程序和各种类型的数据。对比Linux来说，文件就是文件，Linux要知道的文件之间唯一区别是它是不是可执行文件。可执行文件可以直接执行，当然，这种文件应该含有要执行的内容，并且这种文件要在你的查找路径中。查找路径是你事先指定的路径名列表，Linux查找该路径名列表以找到可执行文件。可执行文件是二进制文件(也就是说，它包含可执行机器代码)和 shell脚本的文件。前面讨论过的 Linux的 file命令查看文件中的数据并对其进行合理的推测，如果你键入file*，你可能会看到类似于下面的内容：

```
INSTALL : symbolic link to/var/adm
ghostvw.txt : ascii text
linux : symbolic link to/usr/src/linux
mbox : mail text
mterm . txt : English text
seyon . txt : English text
xcalc . txt : English text
xclock . txt : English text
xeyes . txt : English text
xgrap.txt : English text
```



```
xlock . txt : English text
xsprea.txt : English text
xtriss . txt : empty
```

第一列中提到的所有文件都是普通文件，它们包含不同的数据类型，所有文件都置于file命令能被执行的那个目录中。

2、目录文件

目录也是文件，它们包含文件名和子目录名，并包含指向那些文件和子目录的指针。目录文件是Linux存储文件名的唯一地方，当你用ls命令列出一个目录的内容时，你所做的事只是列出了这个目录文件的内容，还没有触及文件本身。

当你用mv命令对存在当前目录中的一个文件重新命名时，你所做的事只是改变了该文件在目录文件中的条目。如果你把一个文件从一个目录移到另一个目录中，你所做的事只是把这个文件的说明从一个目录文件移到另一个目录文件中，当然，这要求新的目录在同一实际的磁盘上或同一分区上。如果不是这样的话，Linux将该程序的每一个字节都实际地拷贝到另一磁盘上。

3、目录与实际磁盘

Linux系统中的每个文件都被赋予一个唯一的数值，这个数值称做索引节点。索引节点存储在一个称作索引节点表 < inode table > 中，该表在磁盘格式化时被分配。每个实际的磁盘或分区都有其自己的索引节点表。一个索引节点包含文件的所有信息，包括磁盘上数据的地址和文件类型。文件类型包括如普通文件、目录和特殊文件这样的信息。

Linux文件系统把索引节点号1赋予根目录，这给予Linux的根目录文件在磁盘上的地址。根目录文件包括文件名、目录名及它们各自的索引节点号的列表，Linux可以通过查找从根目录开始的一个目录链来找到系统中的任何文件。根目录文件的内容可能像下面这样：

```
1
1 . .
45 etc
230 dev
420 home
123 . profile
```

注意文件.(点)和..(双点)在目录中的含义。因为这是根目录，所以“.”和它的父目录“..”是相同的，而在/home目录文件中它们的内容则不同：

```
420 .
1 . .
643 fred
```

注意，当前目录(.)的索引节点与建在根目录文件中的/home的索引节点相同，父目录(..)的索引节点与根目录的索引节点相同。

Linux通过上下连接目录文件系统来驾驭其文件系统。如果你想把文件移到另一实际磁盘的目录中，则Linux通过读取索引节点表来检测这种行动。在这种情况下，文件在被从原来的位置删除前，实际移动到新磁盘上并被赋予了该盘上的一个新的索引节点。

与使用mv命令一样，当你用rm命令删除文件时，你也不能触及文件本身。取而代之的是，Linux标记该索引节点是自由的并把它放回可供使用的空闲索引节点表中，目录中的该文件的条目被擦除。

4、连接

普通的连接实际上根本不是文件，它们只是指向同一索引节点的那些目录条目。索引节点表记录一个文件有多少个连接；只有当最后一个目录引用被删除时，该索引节点才最终被释放回空闲表中。显然，普通的连接不能跨越设备界线，因为所有的目录条目都指向同一个索引节点。

Linux及大多数UNIX的现代版本都有另一种连接，称为符号连接。对于这种连接，这个目录条目包含一个文件的索引节点(该索引节点本身又是对Linux逻辑文件系统上某处的另一个文件的引用)。一个符号连接可以指向同一磁盘或另一磁盘上的另一个文件或目录，也可以指向另一台计算机上的一个文件或目录。普通连接与符号连接之间一个主要的差别是：使用普通连接，每个连接都有同等的地位(也就是说，系统把每个连接都看作是原始文件)，并且在文件的最后一个连接被删除之前，实际的数据不会被删除；使用符号连接，当原始文件被删除时，所有对该文件的符号连接也都被删除，符号连接的文件没有与原始文件相同的地位。

除了连接与文件之间的这些细微的差别外，你可像直接访问文件那样对待和准确地访问连接。

你可用ls-l命令来看出某个文件是一个连接文件，因为该命令的回应显示了这个本地文件名，接着指示了被连接的文件：


```
lrwxrwxrwx 1 root root 4 Oct 17 15 : 27 Info- > info/
```

为了指示该文件是一个连接文件，文件权限标志以l开头。

5、特殊文件

每个与Linux系统相连的实际设备(包括磁盘、终端和打印机)，都在文件系统中表示出来。大多数设备都放在/dev目录中。例如，如果你正在系统控制台(console)上工作，那么你的相关设备就叫/dev/console。如果你正在标准终端上工作，那么你的设备名可能是/dev/tty01。终端或串行线都称为tty设备(它代表电传打字机，电传打字机是最初的UNIX(终端)。为了确定你的tty设备叫什么名字，键入tty命令，系统会显示你所连接的设备的名字。

打印机和终端称作特殊字符设备，它们可接受和产生字符流。另一方面，磁盘把数据存储在由柱面和扇区编址的块中，你不能只访问磁盘上的一个字符，你必须读写整个块。对于磁带通常也是这样的，这种类型的设备称作特殊块设备。对更复杂的情况，磁盘和其他特殊块设备必须能像面向字符的设备那样工作，因此每个特殊块设备都有一个与之匹配的特殊字符设备。Linux通过读取要送到字符设备的数据并为特殊块设备转换这些数据来实现这个转换过程，这一过程不需要你做任何事。

至少还有另一种类型的特殊设备你可能遇到，即称为管道的FIFO(先入先出缓冲区)。FIFO看起来像普通文件，如果你向里写入，它们就增长；如果你从FIFO读出，它的大小就会缩减。FIFO主要用于系统进程中，以允许众多程序向一个单一的控制进程发送信息。例如，当你用lp命令打印一个文件时，lp命令建立打印进程并通过向一个FIFO发送信息来通知lpsched守护程序。守护程序是一个不用用户要求就执行的系统进程。

一个非常有用的特殊设备文件是位存储桶(bit bucket)/dev/null，你送入/dev/null的任何东西都被忽略，当你不想看到命令的输出结果时，它是很有用的。例如，如果你不想看到打印在标准错误设备上的任何诊断报告，你就可以用下列命令把它们放入位存储桶中：

```
ls-la > /dev/null
```

6、文件权限

Linux中的文件极限的含义比你在文件或目录上所拥有的权限的含义多。虽然权限决定谁能读、写或执行一个文件，但它们也决定这个文件的类型及如何执行这个文件。

可以用长格式的列表命令ls-l来显示一个文件的权限。-l选项告诉ls命令使用长格式的列表，如果你键入ls-l，则你可能看到像下面这样的目录列表：

```
drwx-----2 sglines doc 512 Jan 1 13 : 44 Mail
drwx-----5 sglines doc IO24 Jan 17 08 : 22 News
-rw-----1 sglines doc 1268 Dec 7 15 : 01 biblio
drwx-----2 sglines doc 512 Dec 15 21 : 28 bin
-rw-----1 sglines doc 44787 Oct 20 06 : 59 books
-rw-----1 sglines doc 23801 Dec 14 22 : 50 bots.msg
-rw-r-----1 sglines doc 105990 Dec 27 21 : 24 duckie.gif
```

该列表几乎显示了目录条目和文件的索引节点所包含的与文件有关的所有信息。

第一列显示文件的权限，

第二列显示对一个文件(或一个目录中的附加块)的连接数量，

第三列显示谁拥有该文件，(所有权有三种可能：所有者、所有者的组和其他人。)

第四列显示文件所属的组，

第五列显示文件中的字节数，

第六列显示文件创建的日期和时间，

第七列显示文件本身的名字。

权限字段(第一列)分为四个子字段：

```
-rwx rwx rwx
```

第一个子字段定义文件的类型。普通文件以连字符(-)作为占位符，目录用一个d标记。

下表显示了文件类型子字段的权限值。

字符 含义

- 普通文件

b 特殊块文件

c 特殊字符文件

d 目录

l 符号连接

接下来的三个子字段显示文件的读、写和执行权限。例如，这三个子字段中的第一个子字段的rwx是说所有者对文件有读、写和执行权限。这三个子字段中的第二个子字段是说拥有该文件的组对文件有读、写和执行权限，第三个子字段显示其他人的权限。

这些权限字段可以显示更多的信息。事实上，有几种属性包含在这三个字段内。可惜，这些属性的含义是由你所使用Linux的版本及文件是否是可执行文件来决定的。

四、文件系统的管理

1、安装文件系统

首先介绍有关文件系统的一些概念。在一个文件系统被访问之前，它必须安装(mount)在某个目录下。例如：如果在软盘上有一个文件系统，则必须将它安装到某个目录下，比如mnt，以便能够访问其上的文件。安装了文件系统后，该文件系统的所有文件都会显示在那个目录下。卸载文件系统后，该目录(这里为/mnt)变为空。

硬盘上的文件系统也是如此。在启动时系统自动将文件系统安装到硬盘上，根文件系统被安装在目录“/”下，如果/usr有另一个单独的文件系统，它则安装在/usr下。如果只有一个根文件系统，所有的文件(包括那些在/usr下的)都存放在该文件系统中。

mount命令用来安装文件系统。

命令：

mount -av

从文件/etc/rc中执行(它是在启动时的系统初始化文件)，mount -av命令从文件/etc/fstab中获取文件系统和安装点的信息，fstab文件的一个例子如下：

```
$ device directory type options
```

```
/dev/hda2 / ext2 defaults
```

```
/dev/hda3 /usr ext2 defaults
```

```
/dev/hda4 none swap sw
```

```
/proc /proc proc none
```

第一个域是设备名，即安装的分区名；

第二个域是安装点；

第三个域是文件系统类型，如ext2(ext2fs文件系统)或minix(Minix文件系统)。

fstab文件的最后一个域包括了mount的选择项，通常设置为“ defaults ”。

可以看出，交换分区也包括在/etc/fstab中，其安装目录为none，类型为swap。命令swapon -a也是从/etc/rc中执行的，可用来交换列在/etc/fstab中的所有交换设备。

安装或卸载文件系统，有许多软件包可以使普通用户安装或卸载文件系统(尤其是软盘)而不破坏系统的安全性。

mount -av命令实际上安装除根文件系统(在fstab文件中为/dev/hda2)以外的全部文件系统，根文件系统在启动时由核心自动安装。

除了用mount -av以外，还可以手工安装文件系统。命令：

```
$ mount -t ext2 /dev/hda3 /usr
```

等价于在上例的fstab文件中安装文件系统/dev/hda3。

一般地，不会用手工安装或卸载文件系统，/etc/rc中的mount -av命令在启动时负责安装文件系统，文件系统在用shutdown或halt命令关闭系统之前自动卸载。

2、检查文件系统

通常情况下，不断地检查文件系统中被破坏的文件是一个好的办法，有些系统在启动时自动检查它们的文件系统(用/etc/rc中适当的命令)。

用于检查文件系统的命令依赖于出问题的文件系统类型，对ext2fs文件系统而言(最常用的类型)，命令为e2fsck，例如，

```
命令：$ e2fsck -av /dev/hda3
```

将检查/dev/hda3上的ext2fs文件系统，并自动纠正任何错误。

通常在检查文件系统之前最好卸载该文件。例如，

命令

```
$ umount /dev/hda3
```

将卸载/dev/hda3上的文件系统，然后再检查该文件系统，但是不可能卸载根文件系统。为了在没有卸载之前就检查根文件系统，可以使用维护磁盘boot/root；当文件系统中文件“忙”，即被其它运行进程使用着时，也不能卸载该文件系统。例如当某个用户的当前工作目录在某个文件系统上时就不能卸载该文件系统。如果试图卸载一个正在被使用的文件系统，将得到一个“Device busy”的错误信息。

其它类型的文件系统用e2fsck命令的其它形式，比如efscck和xfscck。在某些系统中，可以简单地使用fsck命令，它将首先决定文件系统类型，再执行相应的命令。

有一点很重要，那就是在检查完文件系统后，如果对该文件系统做了任何纠正，就应该立即重新启动系统(当然，一般情况下，不能在文件系统被安装时检查它)。例如，如果e2fsck报告对文件系统的错误作了些纠正的话，就应该立即用shutdown-r命令确保重新启动系统，这使得当e2fsck修改了文件系统后，系统能重新同步读文件系统的信息。

/proc文件系统从不需要以这种方式检查，/proc是一个内存文件系统，直接由核心管理。

除了为交换空间保留单独的分区以外，还可以使用交换文件。然而为了能做到这样，必须安装相应的Linux软件，并将一切安排就绪，才能创建交换文件。

如果安装了Linux系统，则可以用下列命令创建交换文件。下面将创建一个大小为8208块(约8MB)的交换文件：

```
$ dd if=/dev/zero of=/swap bs=1024 count=8208
```

这条命令创建交换文件，“count=8208”项为交换文件的块数。

```
$ mkswap /swap 8208
```

该命令将初始化交换文件，同样地也可以用相应的值替换交换文件的命令和大小。

```
$ /etc/sync
```

```
$ swapon /swap
```

同步以后便可对刚创建的/swap文件进行交换。同步是用来保证文件被写到磁盘上，这样做的一个主要缺点是，所有对交换文件的访问都是通过文件系统来实现的，这意味着组成交换文件的块可能不连续，因此性能也许不如使用交换分区，那样所有的块都是连续的，而且I/O请求是直接和设备打交道的。

使用交换文件的另一个缺点是在使用大的交换文件时，破坏文件系统中数据的机会变大，如果出了错误的话，那将破坏文件系统。将文件系统和交换分区分开可以避免该现象发生。如果暂时需要很大的交换空间，使用交换文件便十分有用。例如，如果编译一个很大的程序，并希望运行得快一点的话，可以暂时创建一个交换文件，并同常规的交换空间一起使用。

取消交换文件，首先用swapoff，例如：

```
$ swapoff /swap
```

然后再安全地删除该文件

```
$ rm /swap
```

请记住每个交换文件(分区)最大为16MB，但可以在系统中使用多达8个交换文件或分区。

3、其它任务

对系统管理员来说还有许多日常事务性的工作：

(1) 系统启动文件

系统启动时，在任何用户注册之前系统将自动执行一系列初始化程序。下面是对该过程的详细描述：

在启动时，内核产生进程/etc/init，init是一个程序，它读取其配置文件/etc/inittab，并根据该文件的内容产生其它进程。由inittab产生的最重要的进程之一就是可以在每个虚拟控制台(VC)上启动的/etc/getty进程。getty进程抢占虚拟控制台VC，并在VC上启动login进程，它允许你在每个VC上注

册，如果/etc/inittab不包含某个VC的getty进程，就不能在该VC上注册。

另一个从/etc/inittab中产生的进程是/etc/rc，即主系统的初始化文件。该文件是一个shell程序，它执行在启动时需要的任何初始化命令，诸如安装文件系统和初始化交换空间。

系统也可以执行其它的初始化程序，例如/etc/rc.local。/etc/rc.local通常包含针对用户自己系统的初始化命令，诸如设置主机名。rc.local可以从/etc/rc或从/etc/inittab中直接启动。

(2) 设置主机名

在一个网络环境中，主机名是用来识别某个机器的唯一标识，而在一个单机环境中，主机名只给出了系统的称呼。

设置主机名是一件很简单的事，只需用hostname命令即可。如果在一个网络中，主机名必须是机器的主机名全称，比如ccjsj.com；如果不是网络环境，则可以选择任意的主机名和域名，比如loomer.vpizza.com，shoop.nowhere.edn或floof.org。

设置主机名时，主机名必须在文件/etc/hosts中，它给每个主机赋一个IP地址，即使机器不在网络上，也应该在/etc/hosts中包含用户自己的主机名。

例如，如果主机不在TCP/IP网络上，主机名为floof.org，则只需在/etc/hosts中包括下列项：

```
127.0.0.1 floof.org localhost
```

它给主机floof.org赋了一个回送地址127.0.0.1(如果不在网络上就使用该地址)，别名localhost也赋给该地址。

如果在TCP/IP网络上，真正的IP地址和主机名也应该在/etc/hosts中，例如，如果主机名为www.ccjsj.com，IP地址为128.253.154.32，则在/etc/hosts中增加下列行：

```
128.253.154.32 www.ccjsj.com
```

如果主机名不在/etc/hosts中，那么将无法设置它。

设置主机名，只需使用hostname命令即可，例如命令：

```
$ hostname-S www.ccjsj.com
```

设置主机名为www.ccjsj.com。多数情况下，hostname命令是由某个系统启动文件执行的，例如/etc/rc或/etc/rc.local。编辑上述两个文件，改变其中的hostname命令来设置主机名，再重新启动系统时，主机名将被设置为新值。

(3)、紧急情况处理

在某些偶然的场合下，系统管理员将面临从一个彻底性的灾难中恢复系统的问题，例如忘记了根的口令或破坏了文件系统。

Linux并不是一个不稳定的UNIX版本，事实上Linux比商用UNIX在许多平台上出现的问题要少得多。Linux还提供一个强有力的向导，它能帮助你走出困境。

解决问题的第一步就是试图自己修复，到处看一看，看该如何办。系统管理员在仔细研究问题之前将会花大量的时间，不顾一切地发出请求帮助，多数情况下，你将发现自己解决问题实际上非常容易。

很少情况下需要重新安装被损坏的系统。许多新用户由于不小心删掉了一些基本的系统文件，便立即去找安装盘，这并不是一个好主意。在采取这项极端的方法之前，仔细研究一下并寻求其他人帮助解决问题。多数情况下，可以从维护磁盘上恢复系统。

* 用维护磁盘恢复

系统管理员的一个必不可少的工具就是所谓的"boot/root"盘，它是一个可以以完整的Linux系统启动的软盘，它与硬盘无关。boot/root盘实际上很简单，首先用软盘创建一个根文件系统，将所需要的应用例程放在上面，再在该软盘上安装LILO和可启动的内核。另一种方法是用一个软盘装内核，另一个装根文件系统，两种情况结果都一样，即可以完全从软盘上启动Linux系统。

sunsite.unc.edu中/pub/Linux/GCC/rootdisk中H.J.Lu的boot/root盘是另一个维护磁盘的例子，如果你想试一下的话也可以自己建一个。然而多数情况下，用事先做好的boot/root盘则比较容易而且可能更加完整。

使用boot/root盘是很简单的，只需要用该盘启动系统，并用root注册(通常没有口令)。为了存取硬盘上的文件，需要手工安装文件系统，例如命令

```
$ mount -t ext2 /dev/hda2 /mnt
```

把/dev/hda2上的ext2fs文件系统安装在/mnt上。记住"/"现在在boot/root盘上，用户需要将硬盘的文件系统安装在某个目录下，以便存取其上的文件。因此，如果将根文件系统安装在/mnt下的话，硬盘上的/etc/passwd则在/mnt/etc/passwd中。

* 修复根口令

如果忘记了根口令，可以用boot/root盘启动，将根文件系统安装在/mnt下，并在/mnt/etc/shadow中将root的口令域清除。

这时root没有口令。当用硬盘重新启动时，可以用root注册，并用passwd重新设置口令。如果知道如何使用vi，则可以在boot/root盘上使用它。因为在boot/root盘上，其它的编辑器，如Emacs，可能没有，但vi一定有。

* 修复损坏的文件系统

如果莫名其妙地破坏了文件系统，则可以用软盘运行 e2fsck来纠正文件系统中任何被破坏了的数据(当然指的是用ext2fs文件类型)，其它文件系统类型可以用 fsck命令的其它形式。

当从软盘检查文件系统时，最好此时文件系统没有被安装。

最常见的文件系统损坏就是超级块损坏。超级块是文件系统的头，它包含了文件系统的状态、大小、空闲块等，如果超级块被损坏(例如不小心将数据直接写到文件系统的分区表中)，系统将根本无法识别文件系统。任何试图安装文件系统的操作都将失效，e2fsck也无法修复系统。

幸运的是ext2fs类型的文件系统在驱动器“块组”的边界保存了超级块的备份，通常情况下是每8K块为一个块组。为了告诉e2fsck用超级块的拷贝，可以用诸如下面的命令：

```
$ e2fsck -b 8192 <partition>
```

其中<partition>是文件系统驻留的分区表，-b 8192选择项告诉e2fsck用文件系统中存储在块8193的超级块备份。

* 恢复丢失的文件

如果不小心删除了系统中的重要文件，则没有办法取消删除(undelete)。然而可以从软盘拷贝相关的文件到硬盘中，例如，如果删除了系统中的/bin/login(允许注册的文件)，只需用boot/root软盘启动，将根文件系统安装到/mnt上，并用命令：

```
$ cp-a/bin/login /mnt/bin/login
```

-a选择项告诉cp保留被拷贝文件的存取权限。

当然，如果删除的文件并不是基本的系统文件，而这些文件在boot/root软盘上没有，那么就没办法了。不过如果做了备份，当然总是可以恢复的。

* 恢复损坏的库

如果不小心破坏了库或在/lib中的符号连接，那么依赖于那些库的命令将无法运行。最简单的办法就是用boot/root盘启动，安装根文件系统，并在/mnt/lib下修复库。

第五节课：系统管理（一）

- 1、明确系统管理的目的是什么。
- 2、了解Linux口令加密的两种方式。
- 3、学会使用日志排除故障及系统的安全。
- 4、了解Linux的7个运行级。

一、用户管理

- 1、用命令行添加及删除用户帐号：

```
useradd passwd usermod userdel
```

- 2、手工添加、更改及删除用户：

- 用户帐户记录文件：/etc/passwd

passwd文件每行为7个域（字段），分别为：

用户名、口令、用户ID（UID）、组ID（GID）、用户说明、主目录、缺省shell。

- 用户口令文件：/etc/shadow

Linux的两种加密方式为DES（Data Encrypt Standard）和MD5（Standards for Message Digest）：

（1）DES是一种高级算法，对口令的长度有限制，必须在8个字符以内，加密后为13个字符的散列值。用常用字典可以很容易猜出口令。

(2) MD5口令长度为1到32个字符，加密后的散列值为32个字符。

/etc/shadow文件每条记录对应的字段分别为：

使用者名称：最多8个符号，可以用大小写，通常都是小写。直接对应/etc/passwd文件中的使用者名称。

密码：加密过的32个字节。一个空格(就是：)表示登入时不需密码(不是个好主意)，一个“*”项目(就是：*)指出帐号已经关闭。

密码最后一次变更起所经过的日数：(从1970年一月一日起)。

密码经过几天可以变更：(0表示可以随时变更)

密码经过几天必须变更：(99999表示使用者可以保留他们的密码很多很多年不变) 密码过期之前几天要警告使用者：(7为一周)

密码过期几天后帐号会被取消

帐号经过几天会被取消

保留栏位

- 组信息文件：/etc/group

Linux下的组只是单纯的分组，没有权限分配的功能。若要对组进行权限分配，必须借助工具软件，如sudo。

- 用户配置文件的模板：/etc/skel/. *

用户的提示符等缺省配置文件是从/etc/skel目录拷贝过来的。

练习：

1、手工建立用户user，要求：主目录为/user；说明行为mail user；shell为/bin/bash。

(1) vi /etc/passwd，在此文件内加入下行：

user:x:501:501:mail user:/user:/bin/bash

(2) vi /etc/shadow，在此文件内加入下行：

user::11533:0:99999:7:::

(3) 用passwd命令为用户设置口令：passwd user

(4) 在根目录下建立user目录：mkdir /user

(5) 拷贝/etc/skel/*到/user目录下：cd /etc/skel；cp -r ./* /user

(6) 更改/user目录的属主以及属组：chown -R user.user /user

2、把user用户更名为john，并且把主目录改为/home/john，其他缺省。

(1) vi /passwd，把user字段改为john，把主目录/user改为/home/john

(2) vi /shadow，把user字段改为john

(3) 拷贝/user /home/user：cp -r /user /home

(4) 更名/home/user为/home/john：mv /home/user /home/john

(5) 更改属主及属组：chown -R john.user /home/john

3、设置john的shell，使john不能交互（不能在本机登录，也不能利用telnet登录到本机。）但是能登录到本机的ftp服务。

(1) vi /bin/myshell（自己创建此文件）：

echo you can 't use telnet.

echo you can use ftp.

(2) chmod 777 myshell（更改为可执行）

(3) vi /etc/shells，在此文件内加入下行：

/bin/myshell（目的使myshell成为有效系统shell）

二、日志管理

- 日志文件位置：/etc/log/messages
- 日志的配置文件：/etc/syslog.conf

练习：

1、把一台主机名为bsd的主机的日志放入一台主机名叫Linux的主机日志内？

(1) 此时bsd主机为客户机，在客户端：

vi /etc/syslog.conf，加入下行：*. * @linux

(2) 此时Linux主机为服务器，在服务器端：

vi /etc/rc.d/init.d/syslog，加入下行：daemon syslogd -a -o -r

注：此种方法在集群服务器中常用。

注：如果发现自己的服务器有文件被更改或破坏的痕迹，而日志没有记录，则此系统不再被信任，可以考例重装并采取更高级的安全措施。

三、权限管理

1 给/tmp 单独分区（只需大约50M）可能发生缓冲区溢出或使用磁盘限额。

2 尽量避免使用suid如必须用最好用sgid代替

第六节课：系统管理（二）

一、理解运行级

其实init是Linux系统里的根进程，是系统所有进程的祖先。它的主要作用是根据记录在/etc/inittab里的一个脚本（script）程序产生进程。这个文件通常用于控制用户的登录模式。，不过沿用UNIX系统的至多6种的限制，一般只有1到6有效。

init一般在系统启动时自动运行，也可以由root用户调用。它的作用是切换系统的运行状态。它的命令格式是：init NUM 回车，其中NUM表示1到6的任一数字。

6种模式中有两种是系统内定的：

0：停止系统运行。init 0 回车 相当于 halt 回车。

6：重启系统。init 6 回车 相当于 reboot 回车。

注：如果你了解/etc/inittab文件，就千万不要把0设为缺省登录模式，否则你的Linux的运行就只能两点一线了——开机ç è 停机，或者是开机ç è 重启的无限循环。

init 1——进入单用户模式。

这和开机是选择Safe Mode是一样的，可以以root的权限对系统进行管理，但是不能进入图形模式，也不支持中文，声卡和网络服务等就更不用提了。它和Windows的安全模式是比较相似的。

init 2——进入无网络服务的多用户模式

这一模式下，在本机上的使用基本不变。你仍然可以访问网络，上Internet，ftp或telnet到网络中的任何一台服务器。只是你的计算机在网络中不给其他机器提供服务而已。计算机从第3运行级转入这一模式时你会看到一系列网络服务被停止。

注：但注意如果原来已经有用户ftp或telnet进入了你的系统，用init 2是不能中断他和你的连接的，小心让别人在这个时候黑你一把哦！——执行init 2前先用who命令检查一下有谁在线吧。

init 3——进入控制台登录的多用户模式

如果是从init 1 或init 2切换回init 3，我们可以看到一系列服务被重启。

init 5——进入图形化登录的多用户模式

系统启动后或执行init 5命令后，悦目的蓝绿过渡色背景上就出现了漂亮的登录窗口。

更深入的了解.....

如果你想知道更多，就不妨打命令man init看看完全版的手册，枯燥的手册会告诉你：命令init结合/etc目录下的一系列文件可以对你的系统的启动实现深入的定制化设置.....

练习：

1、进入单用户模式，并更改root用户的口令；不要重启系统，进入多用户网络维护模式；最后进入文本多用户网络模式。

二、启动脚本 /etc/inittab

当系统内核引导到最后一步时，init 进程被启动，它会自动查找 /etc/inittab 文件，看看是否有类型为 initdefault（参阅 inittab(5)）的记录。

initdefault 记录决定系统初始运行级别。如果没有这条记录（或者根本就没有 /etc/inittab），那么，你必须在系统控制台输入想要进入的运行级别。

练习：

1、编辑/etc/inittab文件，使系统的缺省运行级为文本多用户网络模式，并且禁止用户用Crlt+Alt+del重启计算机。

Trap ctrl+alt+del下面有一行 将此行注解掉

ca: : ctrlaltdel:/sbin/shutdown -t3 -r now (此行前面加#号即可)

三、系统启动的步骤

step 1 执行 /sbin/init 系统的第一个进程

step 2 执行 /etc/rc.d/rc.sysinit 第一个执行的script文件（负责设置环境变量，检查与挂载文件系统启动swat文件设置网络等）

step 3 执行运行级中的script文件

step 4 执行 /etc/rc.d/rc.local最后一个执行的script文件如果要每次开机时执行某个程序可在此文件中设置

信号

init 对以下信号产生响应

SIGHUP

当接收到该信号后，init会对 /etc/inittab 和 /var/log/inittab 文件进行检查。如果这两个文件之一存在而且文件中有 ASCII 字符的运行级别，init 会切换到相应的新的运行级别。

SIGUSR1

当接收到这个信号量，init 会关闭并重新打开它的控制 fifo，/dev/initctl。此操作对 /dev 被重新挂载后的启动脚本有用。

SIGINT

通常，当用户按了 CTRL-ALT-DEL 键后，内核会向 init 传递此信号。它所执行的操作与 ctrlaltdel 相同。

SIGWINCH

当键盘有 KeyboardSignal 按键按下时，内核向 init 传递此信号，它激活 kbrequest 动作。

- 其他启动脚本几乎都放在 /etc/rc.d 及 /etc/init.d下面。

四、redhat的系统配置命令setup的使用

用此命令可以配置redhat的启动服务等，在system services内可以配置系统启动时启动哪些服务。

五、软件包的管理——RPM的使用

1、rpm的特性

- 软件包的可升级性

使用RPM您可以单独升级系统中的某些部件而无须整个重新安装。当您获得了一套基于RPM 新版操作系统时(如红帽子 Linux), 您无须重新安装机器(而基于其它软件打包机制的常常需要这么做). RPM 允许智能的，全自动的就地升级系统。包中的配置文件在升级过程中会予以保留, 因此您不会丢失定制信息。

- 功能强大的软件包信息查询

RPM 拥有功能强大的查询选项. 您可以搜索数据库来查询软件包或文件。您可以简便的查出某个文件属于哪个软件包或出自哪儿。RPM软件包中的文件以压缩格式存放,拥有一个定制的二进制头文件，其中包含有关包和内容的有用信息，可以让您对单个软件包的查询简便又快速。

- 系统验证

.另一项强大的功能是进行软件包的验证。如果您担心误删了某个软件包中的某个文件，您就可以对它进行验证。任何非正常现象将会被通知。此时，如果需要的话，您可以重新安装该软件包。在重新安装过程中，所有被修改过的配置文件将被保留。

- 保持软件包的原始特征

一项严格的设计目标是要保持软件包的原始特征,就象该软件的原始发布者发布软件时那样。通过使用RPM,您可以拥有最初的软件 and 要使用的任何补丁程序还有详细的软件构建信息。从几方面来说这是非常之大的优点。例如，当某个软件的新版本发行时,您无须从头编译所有文件。您可以看一看补丁程序都需要做些什么。在这种方式下，所有编译条件以及为生成软件所做的修改都将是可见的。

2、rpm的使用

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 #####
```

- 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, 您可以在命令行上使用--replacepkgs 选项，这将忽略该错误信息。

- 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。.

- 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。

- 卸载

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 needed by bar-1.0-1
```

- 升级

升级软件包和安装软件包十分类似：

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

```
foo #####
```

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.rpmnew 这表示您对配置文件的修改不一定能向上兼容于该软件包中的配置文件。因此，RPM 会备份老文件，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。

因为升级其实就是软件包的卸载与安装的综合,您可能会遇见那些操作中所可能发生的错误。有一处不同，那就是当您使用旧版本的软件包来升级新版本的软件时，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
```

要使RPM 坚持这样“升级”，就使用 --oldpackage 命令行参数。

- 查询

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，您还可以连同-q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 <file> 将查询包含有文件<file>的软件包

-p <packagefile> 查询软件包文件名为<packagefile>的软件包

有几种方式来指定查询软件包时所显示的信息。以下选项用于选择您感兴趣的信息进行显示。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

验证软件包是通过比较compares从软件包中安装的文件和软件包中的原始文件信息来进行的。除了其它一些东西，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来验证某个软件包:

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

`rpm -q --whatprovides 文件名` (查询该文件是由哪个软件包提供的)

练习:

1、 查询sendmail软件包的版本号。

六、 源代码的管理和编译。

1、 tar命令的使用

tar的使用是命令行的，可以用来打包或者解包

`tar-xvzf<文档包名>`解压缩被tar打包并被gzip压缩过的文档包。此类文档包一般以.tgz或.tar.gz结尾。

`tar-xvf<文档包名>`

解压缩被tar打包但未被gzip压缩的文档包。此类文档包一般以.tar结尾。

`tar-tvf<文档包名>`

列出文档包内所包含的文件。

`tar-cvzf<文档包名><文件名>`

将符合命令行中文件名的文件打包并且压缩，制作成文档包。文件名中可以使用通配符*或?，文档包需要以.tgz或.tar.gz结尾。

2、 源代码的编译过程

Linux下应用软件安装包一般为filename.tar.gz压缩包格式，首先将DOWNLOAD的软件解包：

```
tar zxvf filename.tar.gz or tar zxvf filename.tgz
```

解包后如果得到的是可执行文件，则可直接按照INSTALL或README文件中的说明进行配置；如果解包后得到的是文件源代码，则还要完成一个源代码的编译过程。

进入文件展开的目录，如果看到：

A、 configure文件，则执行：

```
./configure
```

```
make
```

```
make install
```

B、 Imake文件，则执行：

```
xmkmf
```

```
make
```

```
make install
```

C、 Makefile文件，则执行：

```
make
```

```
make install
```

第二部分：网络服务

第七节课：DHCP服务器的配置

• DHCP的主要功能

DHCP的全称是动态主机配置协议（Dynamic Host Configuration Protocol），由IETF（Internet 网络工程师任务小组）设计，详尽的协议内容在RFC文档rfc2131和rfc1541里。目的就是为了减轻TCP/IP网络的规划、管理和维护的负担，解决IP地址空间缺乏问题。运行DHCP的服务器把TCP/IP网络设置集中起来，动态处理工作站IP地址的配置，用DHCP租约和预置的IP地址相联系，DHCP租约提供了自动在TCP/IP网络上安全地分配和租用IP地址的机制，实现IP地址的集中式管理，基本上不需要网络管理人员的人为干预。而且，DHCP本身被设计成BOOTP（自举协议）的扩展，支持需要网络配置信息的无盘工作站，对需要固定IP的系统也提供了相应支持。

• DHCP的工作原理

• 几个DHCP名词

在介绍DHCP工作原理以前，先解释这几个名词的含义：

DHCP客户：DHCP客户是一通过DHCP来获得网络配置参数的Internet主机，通常就是普通用户的工作站。

DHCP服务器：DHCP服务器是提供网络设置参数给DHCP客户的Internet主机。

DHCP/BOOTP 中继代理：在DHCP客户和服务器之间转发 DHCP 消息的主机或路由器。

DHCP是基于客户机/服务器模型设计的，DHCP客户和DHCP服务器之间通过收发DHCP

最后一项选项是零长度的End(选项 255)，表明这是选项的结束以便DHCP客户处理。采用选项编码的好处是不论选项有多长，DHCP客户都可以正确接收，即使是它不认识的选项（不见得所有的DHCP客户程序都完全遵循RFC标准）。

DHCP客户并没有自己的IP，所以包含DHCP消息的UDP数据报的IP头的源地址段是0.0.0.0，目的地址则是255.255.255.255。

• DHCP分配IP地址的过程:

DHCP客户机初始化TCP/IP，在本地物理子网上广播一个 DHCPDISCOVER 消息，以确定DHCP服务器位

置及其IP地址。如果DHCP服务器和客户不在同一个物理子网上，BOOTP中继代理将转发这个消息给DHCP服务器。由于网络上可能不止一个DHCP服务器，凡所有具有有效IP地址信息的DHCP服务器向客户机发出一个提议。客户机从接收到的第一个提议中选定IP地址信息，并广播一条租用地址的消息请求。由发出该提议的DHCP服务器响应该消息，指定IP地址信息给该客户机并发送一个确认，而所有其它DHCP服务器撤回各自的提议。客户机完成TCP/IP协议的初始化和绑定。配置完成后，客户机就可以使用普通网络通信和连接至其它IP主机时用到的所有IP服务和应用。

• DHCP 服务器软件的安装

在Linux几乎都采用的是Paul Vixie/ISC DHCPd，来实现DHCP服务器端功能。可以访问

<http://www.isc.org/isc>

获得最新消息。

• 修改配置文件

DHCPd默认的配置文件中/etc/dhcpd.conf，这是一个文本文件，DHCPd里有一个语法分析器，能对这个文件进行语法分析，获得配置参数。

dhcpd.conf 格式是递归下降的，关键字大小写敏感，可以有注释，注释以#开头，一直到该行结束。这里给出一个简单的dhcpd.conf的例子，所服务的网络为C类保留网络 192.168.1.0

```
#examples
```

```
# 缺省租约时间：default-lease-time 28800;
```

```
# 最大租约时间：max-lease-time 43200;
```

```
# 子网掩码选项：option subnet-mask 255.255.255.0;
```

```
# 广播地址：option broadcast-address 192.168.1.255;
```

```
# 路由器地址：option routers 192.168.1.1;
```

```
# DNS地址：option domain-name-servers 192.168.1.1;
```



```
# 域名：option domain-name "ccsj.com";
# 以上都是全局参数
```

```
# 子网声明和掩码：subnet 192.168.1.0 netmask 255.255.255.0 {
# 范围：
range 192.168.1.10 192.168.1.100;
range 192.168.1.150 192.168.1.200;
}
```

这段配置文件将允许DHCP服务器分配两段地址范围给DHCP客户，192.168.1.10-100 和192.168.1.150-200

如果DHCP客户在申请租约时不请求一个特定租约失效时间，则以default-lease-time(28800秒)为租约时间，如果有请求一个特定的租约失效时间，则采用max-lease-time (432000秒)

服务器发送下面的参数给DHCP客户机：

子网掩码是255.255.255.0，广播地址是192.168.1.255，默认网关是192.168.1.1，DNS是192.168.1.1。

如果要为一台叫做ftp的机器指定固定的IP地址，可以在dhcpd.conf文件加一条

```
host ftp {

# ftp上网卡的硬件地址
hardware ethernet 08:00:00:4c:58:23;
#固定IP
fixed-address 192.168.1.210;
}
```

- dhcpd.leases

dhcpd.leases 是DHCP客户租约的数据库文件，默认目录在/var/state/dhcp/，文件包含租约声明，每次一个租约被获取、更新或释放，它的新值就被记录到文件的末尾。在DHCPd第一次安装后，并不会生成这个文件。但DHCPd的运行需要这个文件，所以可以建立一个空的文件。

```
# touch /var/state/dhcp/dhcpd.leases
```

DHCPd记录这个文件的格式是

```
lease ip-address { statements... }
```

每个记录包含一个提供给客户的IP地址，在花括号里的语句包含一些租约信息。具体的租约信息因客户发出不同的DHCP请求而稍有差别。

- 运行DHCPd

要启动DHCPd，简单地键入 /usr/sbin/dhcpd 或用ntsysv 把DHCPd服务自动启动，也可以用

```
/etc/rc.d/init.d/dhcpd start，
```

这样启动后，DHCPd是启动在eth0 上，如果DHCPd上的服务器还有另外一块网卡eth1，想在eth1上启动

dhcpd，就键入

```
#/usr/sbin/dhcpd eth1
```

以上例子的dhcpd.conf来启动dhcpd，：

要注意的是dhcpd.leases的时间记录采用GMT时间，而不是本地时区的时间。要查看本机的GMT时间可以用

```
date -u
```

总结

以上就是DHCPd常用配置，实际应用DHCP还要考虑IP分配的一些策略问题，同时要保证网络的健壮性，必须至少要有两台DHCP服务器一起工作，如果一台出了故障，另一台可以继续为DHCP客户服务。然而目前DHCP协议里并没有能让两台DHCP服务器协同工作的机制，不能保证分配的地址的唯一性，所以这两台DHCP服务器里的可分配地址空间必须进行调整，不能有交叉重复的IP地址。

第八节课：DNS服务器的配置

一、什么是DNS

DNS是域名解析服务器的缩写，简单的说就是把域名转换成与其相对应的IP地址，还可以把IP地址反向解析转换成域名。有关DNS的说解，请参考其它的DNS文章。

二、前提条件

1、在安装Linux时你要选择安装DNS服务，并且要在Linux启动时同时也启动DNS。

2、可能用到的文件

文件名	所在目录	
named.conf	/etc/	"系统自带"
named.local	/var/named/	"系统自带"
name-ip (正向解析)	/var/named/	"创建或生成"
ip-name (反向解析)	/var/named/	"创建或生成"

说明：如果你在安装完成LINUX后在配置网络时同时也配置了DNS选项话，你可能只要做很少的工作就能完成DNS服务的安装。详细说明如下：假设你的IP为局域网的内部IP：192.168.1.1 域名为：ccjsj.com

则在控制台下键入：netconf后输入详细的DNS信息后，即在此选项：Domain Name Server (DNS)里配置DNS的相关信息后，再在控制台下输入命令：ndc restart 后就会在"/var/named"目录下自动生成“正向解析”和“反向解析”的2个文件，文件名应该是：“192.168.1”和“ccjsj.com”，你只需按照下面的格式添写就可以完成DNS的配置了。你也可以任意给这两个文件起名，只要你能看得懂就可以，但一定要在文件named.conf中对应上。

三、上述5个文件的内容如下：

1、named.conf

```
options {
    directory      "/var/named";
};
zone "." {
    type hint;
    file "named.ca";
};
zone "ccjsj.com" {
    type master;
    file "ccjsj.com";
};
zone "0.0.127.in-addr.arpa" {
    type master;
    file "named.local";
};
```

```

zone "1.168.192.in-addr.arpa" {
    type master;
    file "192.168.1";
};
3、named.local
@      IN      SOA      localhost.root. localhost (
1997022700 ; Serial (根据此数字判断是否要更新资料)
28800      ; Refresh (检查资料库文件是否更新的时间)
14400      ; Retry (更新失败后再次尝试时间)
3600000    ; Expire (更新失败后再次尝试时间)
86400 )    ; Minimum (缺省的生存时间)
IN         NS         localhost.
1          IN         PTR         localhost.
4、正向解析文件：ccjsj.com
@      IN      SOA      ns.ccjsj.com  root.ns.ccjsj.com (
1997022700 ; Serial
28800      ; Refresh
14400      ; Retry
3600000    ; Expire
86400 )    ; Minimum
NS         ns.ccjsj.com.
MX         10         ns.ccjsj.com.
;
localhost      IN      A      127.0.0.1
ns              IN      A      192.168.1.1
www             IN      A      192.168.1.1
mail            IN      A      192.168.1.1
ftp             IN      A      192.168.1.1
;
ns             CNAME     ns.ccjsj.com.
www            CNAME     ns.ccjsj.com.
mail           CNAME     ns.ccjsj.com.
ftp            CNAME     ns.ccjsj.com.
5、反向解析文件：192.168.1
@      IN      SOA      ns.ccjsj.com  root.ns.ccjsj.com (
1997022700 ; Serial
28800      ; Refresh
14400      ; Retry
3600000    ; Expire
86400 )    ; Minimum
NS         ns.ccjsj.com.
1          IN      PTR         ns.ccjsj.com.

```

至此，已经完成DNS的配置，还有几个相关的文件，你可以参考一下：

/etc/host.conf

order host,

bind

multi on

/etc/HOST

ns

/etc/host

127.0.0.1 ns.ccjsj.com ns localhost.localhostdomain localhost

192.168.1.1 ns.ccjsj.com ns

/etc/sysconfig/network

hostname=ns

至此，DNS的配置已经完成，现在要进行测试，在控制台下输入以下命令：

ndc restart

nslookup

会出现一个提示符“>”，你可以输入你的主机名myname,及mail,ftp,www,机器将会解析成具体的信息，相信你会成功的。

四、为新建的域添加一个子域（域名为sh.ccjsj.com）子域的dns服务器主机名为computer10

1 只需在/var/named/ccjsj.com文件中加入下面两句话

ns IN NS computer10.sh.ccjsj.com.

computer10. IN A 192.168.1.100

2 重新启动然后用nslookup解析一下

3 如果想在子域解析父域则需要在子域的主配置文件中“directory=/var/named下”加入 forward only;

forwarders {

192.168.1.1;

};

然后重新启动。

如果要其它机器要用本机作为dns服务器则只需修改/etc/resolv.conf文件nameserver指向本机（nameserver 192.168.1.1）

注意：在资料库文件中凡是完整的主机名结尾一定要加点

练习：

1 配置一个最为简单的DNS服务器域名为ccjsj.com用自己的姓作为主机名

2 为ccjsj.com添加一个子域域名为sh.ccjsj.com

要求：a 在子域可以解析父域。

b 在父域可以解析子域。

第九节课：smb服务器的配置

Smb服务的配置

SMB是Server Message Block (服务消息块) 协议的英文缩写，是Windows forWorkgroup,Windows NT或者Windows使用共享文件和打印服务的标准化协议。Samba是由澳大利亚的Andrew Tridgell开发，一种在UNIX环境下运行的SMB客户/服务器软件包，提供与实现了UNIX与Windows之间的文

件共享与打印，它可以在许多不同类型的UNIX上运行。由于Linux的分销商RedHat、Slackware、S.u.s.e等都提供Samba的软件包，并且由于Linux的文件系统性能优异，价格廉价以及爱好者甚多，所以本文以Redhat 6.0版的Linux为例来加以说明。

SMB以请求 - 回应方式运行，当客户机发送请求时，请求包含在SMB 中传到服务器。服务器收到这些SMB，经过解析，对客户机发回应答。当一台计算机通过SMB在网络上共享资源时，它就成为一台服务器。当一台计算机隶属于共享资源时，它就成为一台客户机。在这种计算环境中，一台计算机不管它的操作系统是Linux 或者Windows 98，同时作为服务器与客户机都是可能的。SMB 通信使用NetBIOS界面上发生，它可以运行在许多协议上，最常用到的NetBIOS是TCP/IP上的NetBIOS，所以Windows 98上必须安装与支持TCP/IP协议，安装方法已有许多文章介绍，这里就不说明

一、配置Samba服务器

Samba服务器有三种安全模式：share、user和server。

Samba的配置文件只有一个，就是/etc/smb.conf文件。

server模式需要NT域，如果对安全性能不是十分讲求的部门就不需要，实际应用上我想仅有user模式足够，甚至share模式足矣，因为这两种模式配置和使用起来方便多多。

1、先把Samba服务器配置为用户安全模式[user]

(1) . [global]的配置

netbios name=linux

workgroup=SambaServer #服务器的工作组名不必与Windows客户机相同

server string=Samba Server

hosts allow=192.168.0. 127.

security=user

encrypt passwords=yes

smb passwd file=/etc/smbpasswd

wins support=no

其它参数为默认设置。

[tmp]

说明：[tmp]为测试用，去掉前面的#即可。测试成功后，即可把该段的参数应用到共享资源的[public]段，并使用命令

#chmod a+w [public]里的共享资源目录

使所有用户都能读写该资源目录。

(2) . 建立Samba帐户，编辑帐户文件/etc/smbpasswd，删除不需要的帐户

#adduser user #生成的hgz帐户口令是空的。

#passwd user #需要口令时运行此命令增加口令。

#cat /etc/passwd | mksmbpasswd.sh > /etc/smbpasswd

注：生成smbpasswd口令文件。

#vi /etc/smbpasswd #删除不需要的帐户

(3) . 利用smbpasswd命令建立帐户的口令，最好与/etc/passwd的帐户口令相同。

#smbpasswd user #口令为空时亦要运行此命令，最多不加口令就是了。

(4) . 重新启动SMB服务器

#/etc/rc.d/init.d/smb restart

(5) . 配置/etc/hosts文件，增加以下行

192.168.1.1 linux #linux主机

二、Samba服务器配置为共享模式[share]：

share模式的安全性能最低。我想，只要不是为了连入Internet和共享重要的数据，应该够用了。

[global]配置：

在[user]配置的基础上，只需用；注释掉以下各行

```
; guest account=user
```

```
; encrypt passwords=yes
```

```
; smb passwd file=/etc/smbpasswd
```

再把security=user改为security=share即可。

三、配置Win 95/98客户机

1. 配置c:\windows\hosts文件，增加

192.168.1.1 linux

2. 配置网络

确定已安装了NetBEUI、TCP/IP网络协议，Microsoft网络客户。

如果在Linux Samba服务器上还配置了DNS服务的话，必须指定DNS服务器的ip地址。

3. 开机时以user作为用户名登录网络，鼠标右键点击“网上邻居”，选“映射网络驱动器”，选好驱动器符，然后输入\\linux\user，确定，再输入口令（如果口令不为空的话），确定即可。亦可输入\\linux\tmp(任何可用帐号都能共享)。如果映射成功，先前网上邻居看不到的SambaServer服务器这时应能看得到了。否则，从服务器上运行smbclient -L linux，再刷新网上邻居即可看到。

说明：此时，最好不要双击网上邻居里的Samba服务器，因为会出错。

我就是因为这样，以为Samba还没有配置好，不断地修改[global]，不断地调试而浪费了很多时间。为什么会这样呢？我不知道，大家最好去问微软。-.)

4. Samba设置成share模式时，Win95/98开机时用任何用户名

（不需要用Linux Samba服务器生成的有效帐号）登录都可以映射成\\linux\tmp。如果想映射成\\linux\user，则需要修改Win 95/98注册表。

在HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\VxD\VNetsup

中增加一DWORD值：值名：EnablePlainTextPassword 数值：0x01。

说明：Win95的早期版本不用修改注册表。

检测

1. 用ping命令来检查网络是否连通

Ping linux #从客户端

Ping ht2/ht3 #从服务器端

2. 用smbclient查看共享资源

#smbclient -L linux #查看Samba服务器上的共享资源

屏幕上显示如下：

```
Added interface ip=132.102.65.101 bcast=132.102.255.255 nmask=255.255.0.0
```

```
Password:
```

```
Domain=[SAMBASERVER] Os=[Unix] Server=[Samba 2.0.3]
```

```
Sharename Type Comment
```

```
tmp Disk Temporary file space
```

```
IPC$ IPC IPC Service (Samba Server)
```

```
Server Comment
```

```
LINUX Samba Server
```

```
Workgroup Master
```

```
SAMBASERVER LINUX
```

其实，如果Win 95/98客户机上设置了共享资源的话，可以用

```
#smbclient -L ht2/ht3 来查看Win95/98客户机上的共享资源。
```

用 #smbclient //ht2/共享名（如磁盘d）来共享Win95/98客户机上的共享资源。

练习题（1）配置samba能够和win98通信

要求：a 新建三个组manager、sales、tech；六个用户user1user2user3、

user4、user5、user6；让user1,user2属于manager组user3,user4属sales组user5,user6属于tech组。

b 新建三个目录manager、sales、tech设为共享并且让user1、user2只能访主目录和manager其他用同理。

（2）进行域认证

要求：a 设本域为samba

b 分别以不同的身份登陆（六个用户）

参考步骤

第一题

第一步 新建六个用户（adduser user1、user2、user6）；新建三个组groupadd manager（或修改/etc/group文件）然后把这六个用户加到这三个组中：

（修改/etc/passwd文件即把用户的gid改相应的编号）。。

第二步 在/home/dept下创建 manager、sales、tech三个目录（mkdir /home/dept/manager）然后更属主用命令chown root.manager /home/dept/manager 其两个同理。

第三步 将它们权限修改为770 chmod 770 /home/dept/manager

第四步 添加samba用户smbpasswd -a user1 其它同理

第五步 修改主配置文件/etc/smb.conf(7.0版在/etc/samba/smb.conf)

```
workgroup=samba
```

```
server string=This is NT server
```

```
security=user
```

```
hosts allow=192.168.1. 192.168.0. 192.168.2.
```

```
max log size=50 (单位k)
```

```
smb passwd files=/etc/smbpasswd
```

```
;domain logons=yes
```

第六步 将新建的三个目录设共享在[homes]段加入下面几行

```
[manager]
```

```
comment=the share for manager
```

```
browseable=yes
```

```
writeable=yes
guest ok=no
write list=@manager
(其它两个目录同上)
```

第七步 重新启动samba服务 /etc/rc.d/init.d/smb restart

第八步 在客户以不同身份登陆（如以user1用户则可以访问user1和manager1两目而没有权限访问其它目录）

第二题

第一步 domain logons=yes 前的分号去掉

第二步 将[homes]段的[netlogon]共享目录前的分号去掉之后保存退出

第三步 在/home 下创建netlogon目录 mkdir /home/netlogon

第四步 重新启动samba服务

第五步 在客户端打开登陆到NT域然后以某一用户登陆到samba域

第十节课：apache的配置

一、apache的启动

Linux捆绑了著名的Apache（阿帕奇）万维网服务器。

手工启动/停止/重起Apache Web服务器后台守护进程httpd是通过在命令“/etc/rc.d/init.d/httpd”后加上参数“start/stop/restart”来进行的：

```
[root@linux /root]# /etc/rc.d/init.d/httpd start
```

```
Starting httpd: httpd httpd: cannot determine local host name.
```

```
Use the ServerName directive to set it manually.
```

二、apache的配置

上面我们看到的是一般新用户在第一次启动Apache Web服务器时经常遇到的一个问题，就是没有向Apache Web服务器说明“ServerName（httpd服务器所服务的主机名）”。Apache Web服务器启动时必须知道自己所服务的主机名，否则是不行的，而系统自己是没办法猜测到用户的Web服务器所要服务的主机名的，所以用户必须主动的告诉Apache Web服务器它所服务的主机名。这个小问题通过修改Apache Web服务器配置文件“/etc/httpd/conf/httpd.conf”中的“ServerName”一行就可以解决：

把：

```
#ServerName new.host.name
```

改为：

```
ServerName www.ccjsj.com
```

注意，顶头的“#”号被去掉了。然后我们再来启动Apache Web服务器进程httpd：

```
[root@linux /root]# /etc/rc.d/init.d/httpd start
```

```
Starting httpd: httpd
```

接下来，我们再用运行于命令行的浏览器lynx来确认一下：

```
[root@linux /root]# lynx www.ccjsj.com
```

完全正常，我们如所期望的那样看到了在lynx浏览器中显示出来的Linux系统上缺省的主页。

在RedHat Linux中，Apache的缺省安装路径是在“/home/httpd”下面。由于RedHatLinux是基于RPM软件包管理系统的，用户可以使用命令：

```
[root@xteam /root]# rpm -ql apache | less
```

来查看Apache的详细的安装路径所在。Apache Web服务器主要的配置文件是在“/etc/httpd/conf”目录下的：httpd.conf

...

```
ServerType standalone
```

```
Port 80
```

```
...
```

“ ServerType ” 指令，说明的是Apache是以一个独立的（ standalone ）守护进程的身份运行于后台，还是由另外的后台守护进程（ Inetd ）在接到 HTTP 请求的时候来唤醒。对于正式的互联网站点来说， standalone 可以提供比 Inetd 方式更快的客户端响应速度。“ Port ” 指令说明了由 Apache Web 服务器所监听的端口号，端口 80 是由 RFC 所规定的标准的 HTTP 服务所驻留的端口，如果你有特殊的需要想让 Apache 监听其它的端口号，如 8080 等等，就可以在这里修改 “ Port ” 这一指令。

```
HostnameLookups off
```

```
...
```

接下来的这一指令 “ HostnameLookups ”，指示 Apache 是记录浏览本网站的客户的 IP 地址（形如：202.102.13.141），还是记录 DNS 地址（形如：www.ccjsj.com）。一般来说，记录 DNS 地址更省空间。

```
User nobody
```

```
Group nobody
```

```
...
```

上面这两行指令指示 Apache 服务器的运行身份，以用户 nobody.nobody 的身份来运行，对于系统安全是最合适的。切忌以用户 root 的身份来运行 Web 服务器。

```
ServerAdmin zhang@www.ccjsj.com
```

```
ServerRoot /etc/httpd
```

```
...
```

“ ServerAdmin ” 指令说明一个电子邮件地址。当 Apache Web 服务器在运行中遇到问题时，Apache 会把这一电子邮件地址提供给客户浏览器，以方便客户浏览器提交错误报告或寻求解决问题的办法。“ ServerRoot ” 指令指示了 Apache 的配置文件、错误记录文件以及 Log 文件等等在服务器上所在的位置。在 Red Hat Linux 中，这些文件缺省的驻留于路径 “ /etc/httpd ”。

```
ErrorLog logs/error_log
```

```
LogLevel warn
```

```
...
```

“ ErrorLog ” 指明相对于路径 “ ServerRoot ” 来说，Apache 的错误记录文件所在的位置。“ LogLevel ” 为 “ warn ” 一般来说是比较合适的，其它的 “ LogLevel ” 从简到繁还有 “ debug ， info ， notice ， warn ， error ， crit ， alert ， emerg ” 等等。

```
LogFormat "%h %l %u %t \"%r\" %>s %b" common
```

```
LogFormat "%{Referer}i -> %U" referer
```

```
LogFormat "%{User-agent}i" agent
```

```
CustomLog logs/access_log common
```

```
CustomLog logs/referer_log referer
```

```
CustomLog logs/agent_log agent
```

```
...
```

上面的指令 “ LogFormat ” 每一个都申明一种记录文件的格式，这些格式都很方便用一些程序脚本来处理，这就使得服务器管理员可以方便的分析网站各个方面的情况，获得远比一个计数器所提供的要多的多的信息。接下来呢，“ CustomLog ” 指令指示某一种格式的记录文件的所在路径（相对于 “ ServerRoot ”）。这里为了减小篇幅，我们就不再重复 Apache 手册里关于记录文件格式的说明了，由兴趣的朋友可以自己查阅一下。相关的网址在 “ <http://www.apache.org> ”

接下来的一个指令是：

```
#ServerName new.host.name
```

```
...
```

这一行在RedhatLinux的发行版中，缺省情况下是被注释掉了。用户可以在这里填写上自己的域名（主机名），以启动Apache Web服务器。这之后的“httpd.conf”配置文件中的指令一般是划归两个主题：性能调优以及虚拟主机的配置。由于这两个主题有其本身的逻辑结构，我们将在以后详述。

```
DocumentRoot /home/httpd/html
```

```
UserDir public_html
```

```
DirectoryIndex index.html index.shtml index.cgi
```

```
FancyIndexing on
```

```
...
```

“DocumentRoot”指令说明所有的那些HTML文件存放在服务器上的地点。在RedHatLinux中，存放这些HTML文件的缺省路径是在“/home/httpd/html”目录下。以后用户自己的Web主页文件等等也存放在“DocumentRoot”指令所指示的这个目录下。

“UserDir”指令说明个人主页存放路径。个人主页通过URL：<http://www.ccjsj.com/~bearzhang>，即“way”时，Apache Web服务器将调出“www.ccjsj.com”这台主机上的“/home/...[蹦柯枷碌奈募](#)”？/a>

紧接着的问题就是，当客户端浏览器试图访问诸如“<http://www.ccjsj.com>”这样的URL时，...睦 又校城pache Web服务器将首先试图给出index.html，如果不存在这个文件，就接着寻找index.shtml，如果还是找不到，就找index.cgi，那么，如果还是找不到怎么办呢？

接下来的指令“FancyIndexing”如果被设置成为“on”，Apache Web服务器将把相对应于被要求的URL的目录下的所有文件的列表以HTML超链接的格式给出；如果“FancyIndexing”被设置成为“off”，那么，相应的，将禁止这种行为。

```
...
```

```
Alias /icons/ /home/httpd/icons/
```

```
ScriptAlias /cgi-bin/ /home/httpd/cgi-bin/
```

```
...
```

这里的两个指令也是有关于URL到文件系统上的路径的映射的，上面的例子，“<http://www.ccjsj.com/icons/>”被映射...旁谗飧睢肪断隆？/a>

```
<Directory />
```

```
Options None
```

```
AllowOverride None
```

```
</Directory>
```

```
...
```

上面的例子其实就是设置一下缺省的访问控制，可以看出这是一个很严格的访问控制，这对于系统安全是有益的。

上面我们把Apache Web服务器三个主要部分大致的浏览了一下，用户在修改了Apache Web服务器的配置文件以适应自己特定的需要以后，需要重新启动Apach，以使新的配置文件生效。在Linux中以用户root的身份执行下面的命令：

```
[root@linux conf]# /etc/rc.d/init.d/httpd restart
```

三、设置实现基于名字的虚拟主机服务

基于名字的虚拟主机服务，不需要更多的IP地址，而且配置简单，无须什么特殊的软硬件支持。现代的浏览器大都支持这种虚拟主机的实现方法。当然，这也就是指一些早期的客户端浏览器也许不支持这种虚拟主机的实现方法。

正是以上原因，我们没有理由不使用基于名字的虚拟主机服务而使用基于IP地址的虚拟主机服务。配置基于名字的虚拟主机服务需要修改配置文件：“/etc/httpd/conf/httpd.conf”，在这个配置文件中增加以下内容。

```
NameVirtualHost 192.168.1.1
```

```
<VirtualHost 192.168.1.1>
ServerAdmin webmaster@www.ccjsj.com
DocumentRoot /home/httpd/www.ccjsj.com
ServerName www.ccjsj.com
ErrorLog /var/log/httpd/www.ccjsj.com/error.log
</VirtualHost>
```

```
<VirtualHost 192.168.1.1>
ServerAdmin webmaster@www.ccjsj.com
DocumentRoot /home/httpd/www.shccjsj.com
ServerName www.shccjsj.com
ErrorLog /var/log/httpd/www.shccjsj.com/error.log
</VirtualHost>
```

也就是在基于IP地址的配置基础上增加一句：NameVirtualHost 192.168.1.1而已。在本例中，为了体现只需要增加一次，所以特别地设置了两个虚拟主机服务。

最后也是建立相应的目录，将主页内容放到相应的目录中去就可以了。

设置实现基于ip的虚拟主机服务(与上述方法相同)

第十一节课：sendmail的配置

• 简单配置：

一 邮件协议SMTP（Simple Mail Transfer Protocol）用于主机与主机之传送邮件（一次处理一封邮件）

POP3 协议(Post office Protocol)用于下载整批邮件（连接110端口）

二 邮件的工具MUA（Mail user Agents）即用户处理邮件的界面（常见的有pine,elm）

MTA (Mail Transport Agents) 负责邮件的传输工作（常见的有sendmail,qmail,smail）

三sendmail服务器的安装 rpm -ivh sendmail-8.9.3-15.i386

四 控制启动/etc/rc.d/init.d/sendmail restart(setup-system service-sendmail开机是启动)

五 相关知识

/var/spool/mail存放用户收信信箱文件

/var/spool/mqueue/l存放每个用户尚未寄出的邮件

~/dead.letter每个用户无法寄出的邮件

/var/log/maillog 邮件系统的记录文件

~/mbox 读完之后邮件的存放地

六 主配置文件sendmail.cf可以不做修改只需在/etc/mail/access文件中加入一行即可

1 192 . 168 . 1 . RELAY（表示允许接收由192 . 168 . 1而来的邮并转寄）将/etc/inetd.conf文件中 #pop-3 stream tcp noswait root /usr/sbin/tcpd ipop3d—

行前面的#号去掉。

2 windows用户端打开outlook—工具—帐号--添加--邮件输入显示姓名—输入自己的电子邮件地址

下一步如图所示-----输入自己的帐号和密码

做完这些windows端的用户就可收发电子邮件了

(如果不能发信则将服务器端的/etc/mail/access文件中加入允许中继本机的IP)

七 如何收发附件

步骤1 使用uuencode将一个8bit的程序以七 bit的格式存储uuencode 原文件名 新文件名 > 123

2 在提示符下输入mail 在邮件的正文中加入 ~r 123 即可

3 收信端读完之后到主目下用 uudecode将程序转换回8bit uudecode mbox

4如果以linux作为客户端则只需配置好pop3服务和sendmail同时将dns指向同一台服务器即可。

八 安全控制

在/etc/sendmail.cw

在/etc/mail/access (或relay-domains) 加入允许中继的主机 (可以控制那些主机可以发邮件)

九 涉及到的命令 mail 收发邮件 mailq 检查代寄邮件encode uudecode 收发附件

第十二节课：x-window的配置

一 安装Xfree86

方式 1 安装rpm包 rpm -ivh Xfree86*

2 用安装盘启动选择安装的方式为upgrade(选择Xfree86套件)

二 XWINDOWS的设置

1 设置工具 (Xconfigurator,XF86Setup, xf86config,)

2 使用xf86config配置xwindows

xf86config (在提示符下执行xf86config)

step 1 设置鼠标 (命令执行之后第一个画面是一段说明文字可以直接按回车)

a.如果本机的鼠标是两键的可选1 microsoft compatible 三键则选2 mouse systems 如果是

ps/2的接口则选4 ps/2 mouse b. 选完回车之后会有一句 Do you want to enable Emulate 3 Buttons?问你是否要模拟三键。一般选择y 2 接下来鼠标装备文件名称即在 Mouse device : /dev/ttyS0 是com1 口 /dev/ttyS1 是com2 口 /dev/pauxx 是ps/2口

step 2 设置键盘

a.设置完鼠标之后的画面 Do you want to use XKB?输入Y可以设置键盘 (XKB是一种协议)

b.press enter to continue, or ctrl-c to abort. (按回车键继续按ctrl+c忽略) 当然继续啦

c.在之后的键盘列表中选1 Standrd 101-key,US encoding (标准的101键盘)

step 3 设置屏幕

a. 设置屏幕水平扫频率

有说明书则按说明书设置选11 Enter your own horizontal sync range自行输入如果不知到, 14, 15寸可以选6 31.5-48.5 ;Non-Interlaced SVGA, 1024*768 @70hz

17寸可以选7, 8。

b. 垂直扫描频率有说明书则按说明书设置选 5 Enter your own vertical sync range自行输入如果不知到, 14, 15寸可以选2 50-90 17寸选3 50-100 (接下来询问名称型号可直接回车)

step 4 设置显卡

- a. Do you want to look at the card database (是否查看支持的显卡资料)选Y回车
- b. 找到自己的型号之后输入其前面的编号
- c. Do not probe clocks or use any clocks line.Press enter to continue, or ctrl-c to abort.

询问是否侦测clock 可直接回车

- d. 接下来的画面选择一个X server 我们一般选择 5 choose the server from the card definition, Xf86_s3
- e. do you want me to set the symbolic link? (是否建立一个符号连接)回答Y

三 X的结构

客户机-服务器结构 linux的X是针对网络环境所做的所以有服务端的X-server与客户端的X-client 两者可以分开在网络中不同的主机上它们之间通过X protocol来沟通

- 1) X-server 提供的服务就是负责屏幕上的显示工作以及输入
- 2) X_client 对X-server 提出请求的X应用程序

也就是说可以利用远端的工作站执行X应用程序并使用其处理器资源而其执行情况可以在本地的

X-server上显示

- 3) 具体方法 (前提条件有两台主机其中一台IP为192. 168. 1. 10另一台为192. 168. 1. 11现在将192. 168. 1. 10作为服务端而将192. 168. 1. 11作为客户端)

step 1 xhost + 192.168.1.11 (允许192. 168. 1. 11使用本机作为X服务端)

step 2 telnet 192.168.1.11 (远端登陆这台客户机)

step 3 export DISPLAY=192.168.1.10:0.0 (指定X程序执行后显示的地方即服务端)

step 4 执行一个X程序如：xclock或xterm 执行的结果将在X服务端显示。

四 小技巧： 1 在图形界面下直接更改分辨率 ctrl + alt + +(-)

1 强行退出图形界面 ctrl + alt + backspace (F1-F6)

2 更改gnome桌面为 kde 桌面 例如更root的桌面就其主目录下编辑文件 .xinitrc 加入一句话exec startkde即可然后重新开启windows (startx)

五 练习题 使用xf86config 配置自己的X-window (步骤参考X-window设置一节)

第三部分：优化与安全

第十三节：内核的设置与编译

• 内核简介

内核：是一个操作系统的核心，它负责管理系统的进程、内存、设备驱动程序、文件和网络系统，决定着系统的性能和稳定性。就好比DOS下的IO . SYS和MSDOS . SYS一体，我们可以把这两个文件叫做DOS的核心。Linux也有它的核心，通常在根目录下，一个叫vmlinuz的文件。我们用这个文件来控制我们的整台PC，包括周边设备和软硬磁盘机、CD - ROM、声卡等。简单地说，核心就是操作系统本身。没有了它，就像一个无人住的家，没有人去维持这个家的动作。一部电脑需要很有效率已稳定的核心，也就是操作系统。因此，核心是整个系统维持下去的关键。

Linux的一个重要的特点就是其源代码的公开性，所有的内核源程序都可以在/usr/src/linux下找到，大部分应用软件也都是遵循GPL而设计的，你都可以获取相应的源程序代码。任何一个软件工程师都可以将自己认为优秀的代码加入到其中，由此引发的一个明显的好处就是Linux修补漏洞的快速以及对最新软件技术的利用。

先介绍一下编译核心的选项，希望能对大家消除对内核的神秘感有所帮助。

- 1.代码成熟等级
- 2.处理器类型和特色
- 3.对模块的支持
- 4.基本的选择
- 5.即插即用支持
- 6.块设备支持
- 7.网络选项
- 8.电话支持
- 9.SCSI设备的支持
- 10.I2O接口适配器
- 11.网络设备支持
- 12.配置业余无线广播
- 13.红外支持
- 14.ISDN的文件系统
- 15.旧型的光驱类型(非IDE界面的光驱)
- 16.字符设备
- 17.USB支持
- 18.文件系统
- 19.控制台驱动
- 20.声卡驱动
- 21.Kernel hacking

• 内核版本

由于Linux的源程序是完全公开的，任何人只要遵循GPL，就可以对内核加以修改并发布给他人使用。Linux的开发采用的是集市模型（bazaar，与cathedral--教堂模型--对应），为了确保这些无序的开发过程能够有序地进行，Linux采用了双树系统。一个树是稳定树（stable tree），另一个树是非稳定树（unstable tree）或者开发树（development tree）。一些新特性、实验性改进等都将首先在开发树中进行。如果在开发树中所做的改进也可以应用于稳定树，那么在开发树中经过测试以后，在稳定树中将进行相同的改进。一旦开发树经过了足够的发展，开发树就会成为新的稳定树。开发数就体现在源程序的版本号中；源程序版本号的形式为x.y.z：对于稳定树来说，y是偶数；对于开发树来说，y比相应的稳定树大一（因此，是奇数）。确定是以 root 的身份签入，然后cd到/usr/src。uname -r这个指令将会显示版本。内核版本的更新可以访问<http://www.kernel.org/>。

• 编译原因

Linux作为一个自由软件，在广大爱好者的支持下，内核版本不断更新。新的内核修订了旧内核的bug，并增加了许多新的特性。如果用户想要使用这些新特性，或想根据自己的系统度身定制一个更高效，更稳定的内核，就需要重新编译内核。

通常，更新的内核会支持更多的硬件，具备更好的进程管理能力，运行速度更快、更稳定，并且一般会修复老版本中发现的许多漏洞等，经常性地选择升级更新的系统内核是Linux使用者的必要操作内容。

为了正确的合理地设置内核编译配置选项，从而只编译系统需要的功能的代码，一般主要有下面四个考虑：

- 自己定制编译的内核运行更快（具有更少的代码）
- 系统将拥有更多的内存（内核部分将不会被交换到虚拟内存中）

- 不需要的功能编译进入内核可能会增加被系统攻击者利用的漏洞
- 将某种功能编译为模块方式会比编译到内核内的方式速度要慢一些

以上是针对成熟的Linux套件如Redhat Linux而言,我的目的是为建造嵌入式Linux操作系统做准备,也是必由之路。

一、编译内核的步骤：

1、准备工作

- 新版本内核的获取和更新

Linux内核版本发布的官方网站是<<http://www.kernel.org/>>, 国内各大ftp上一般都可以找到某些版本的内核。新版本的内核的发布有两种形式,一种是完整的内核版本,另外一种patch文件,即补丁。完整的内核版本比较大,比如linux-2.4.0-test8.tar.bz2就有18M之多。完整内核版本一般是.tar.gz (.tgz) 文件或者是.bz2文件,二者分别是使用gzip或者bzip2进行压缩的文件,使用时需要解压缩。patch文件则比较小,一般只有几十K到几百K,极少的会超过1M。但是patch文件是针对于特定的版本的,需要找到自己对应的版本才能使用。

编译内核需要root权限。把需要升级的内核拷贝到/usr/src/下(下面以2.2.16的内核的linux-2.2.16tar.gz为例), 命令为

```
#cp linux-2.2.16tar.gz /usr/src
```

先查看当前/usr/src的内容,注意到有一个linux的符号链接,它指向一个类似于linux-2.2.14(对应于现在使用的内核版本号)的目录。首先删除这个链接:

```
#cd /usr/src
```

```
#rm -f linux
```

现在解压下载的源程序文件。如果所下载的是.tar.gz (.tgz) 文件,使用命令:

```
#tar -xzf linux-2.2.16tar.gz (解压到 /usr/src)
```

如果下载的是.bz2文件,例如linux-2.2.16tar.bz2,使用命令

```
#bzip2 -d linux-2.2.16tar.bz2
```

```
#tar -xvf linux-2.2.16tar
```

现在再来看一下/usr/src下的内容,发现现在有了一个名为linux的目录,里面就是需要升级到的版本的内核的源程序。还记得那个名为linux的链接么?之所以使用那个链接就是防止在升级内核的时候会不慎把原来版本内核的源程序给覆盖掉了。现在也需要同样处理:

```
#mv linux linux-2.2.16 (重命名)
```

```
#ln -s linux-2.2.16 linux (建立链接)
```

如果还下载了patch文件,比如patch-2.2.16,就可以进行patch操作(下面假设patch-2.2.16已经位于/usr/src目录下了,否则需要先把该文件拷贝到/usr/src下):

```
#patch -p0 < patch-2.2.16
```

- 准备主板和相关硬件的说明手册

其实也不用太详细,只要知道您的硬件是属于哪一类型就行了。例如:有一张SCSI卡,那就要知道这张卡的名字,有一台cd-rom,就要知道这台光驱是哪一种牌子的,是否为标准的IDE/ATAPI界面,还是另有专属接口卡呢?或者,主机版是否有支持Triton芯片(通常586以上的电脑常有),这些信息能帮助我们,使得设定变得清楚且容易。

因此,不管您有什么使用手册,准备好吧。即使现在不用,将来还是会用到的(设X - window system时要显示卡的手册)。

- 检查声卡的IRQ设定和其种类

如果配有一张声卡,除了要知道卡的种类外(例如 Sound Blaster)还需要知道这张卡的IRQ地址。一般来说,声卡的IRQ地址是5或7而IO地址则为

220。DMA则I，不过，有时不同的声卡可能会有不同的设定。因为稍后的选项里，就会要填入这些数字。

- 编译核心的硬件需求

在编译核心时，确定您的RAM最好在8MB以上，否则可能会很慢而且问题会很多，记得查看swap有没有打开（用free指令）。此外，最好不要超频，不然很有可能会发生signal 11的错误，使得编到一半的核心停了下来，其实编译核心就好比编译程序一样，只是因为构成核心的程序太多了，因此我们能小心尽量小心。

2、内核编译的流程

- 编译开始

通常要运行的第一个命令是：

```
#cd /usr/src/linux
```

```
#make mrproper
```

该命令确保源代码目录下没有不正确的目标.o文件以及文件的互相依赖。如使用刚下载的完整的源程序包进行编译，本步可以省略。而如果多次使用了这些源程序编译内核，那么最好要先运行一下这个命令。

确保/usr/include/目录下的asm、linux和scsi等链接是指向要升级的内核源代码的。这是配置非常重要的一部分。删除掉/usr/include下的asm、linux和scsi链接后，再创建新的链接指向新内核源代码目录下的同名的目录。这些头文件目录包含着保证内核在系统上正确编译所需要的重要的头文件。也是上面又在/usr/src下"多余"地创建了个名为linux的链接的原因之一，按照下面的步骤进行：

```
# cd /usr/include
```

```
# rm -r asm linux scsi
```

```
# ln -s /usr/src/linux/include/asm-i386 asm
```

```
# ln -s /usr/src/linux/include/linux linux
```

```
# ln -s /usr/src/linux/include/scsi scsi
```

一旦万事俱备，转到/usr/src/linux。

第二节 配置内核 核心内容

配置内核可以根据需要与爱好使用下面命令中的一个：

```
#make config（基于文本的最为传统的配置界面，不推荐使用）
```

```
#make menuconfig（基于文本选单的配置界面，字符终端下推荐使用，必须安装ncurses-dev和tk4-dev库）
```

```
#make xconfig（基于图形窗口模式的配置界面，Xwindow下推荐使用）
```

```
#make oldconfig（如果只想在原来内核配置的基础上修改一些小地方，会省去不少麻烦）
```

如果不能使用Xwindow，那么就使用make menuconfig好了。界面虽然比上面一个差点，总比make config的要好多了。

选择相应的配置时，有三种选择，它们分别代表的含义如下：

Y - -将该功能编译进内核

N - -不将该功能编译进内核

M - -将该功能编译成可以在需要时动态插入到内核中的模块

在每一个选项前都有个括号，但有的是中括号有的是尖括号，还有一种圆括号。用空格键选择时可以发现，中括号里要么是空，要么是""，而尖括号里可以是空，""和"M"。这表示前者对应的项要么不要，要么编译到内核里；后者则多一样选择，可以编译成模块。而圆括号的内容是要在

所提供的几个选项中选择一项。

在编译内核的过程中，最烦杂的事情就是这步配置工作了，不清楚到底该如何选取这些选项。实际上在配置时，大部分选项可以使用其缺省值，只有小部分需要根据用户不同的需要选择。选择的原理是将与内核其它部分关系较远且不经常使用的部分功能代码编译成为可加载模块，有利于减小内核的长度，减小内核消耗的内存，简化该功能相应的环境改变时对内核的影响；不需要的功能就不要选；与内核关心紧密而且经常使用的部分功能代码直接编译到内核中。下面对选项分别加以介绍。

- 内核设置

1. Code maturity level options代码成熟等级

此处只有一项：prompt for development and/or incomplete code/drivers，如果要试验现在仍处于实验阶段的功能，比如khttpd、IPv6等，就必须把该项选择为Y了；否则可以把它选择为N。在Linux的世界里，每天都有许多人为它发展支持的driver和加强它的核心。但是有些driver还没进入稳定的阶段。但其作者很欢迎其他人去测试这些driver并提出一些bugs。这个问题是说，有一些drive还在做测试中，问您是否要选择这些drive或支持的程序码。

如果键入Y，往后将会出现一些还在测试中的东西给您做选择。（像Java的程序码和PCI bridge），否则就键入N。

2. Processor type and features处理器类型和特色

#Processor family (386, 486/Cx486, 586/K5/5x86/6x86, Pentium/K6/TSC,PPro/6x86MX) [PPro/6x86MX] -----选择处理器类型，缺省为Ppro/6x86MX。它会对每种CPU做最佳化，让它跑得快又好。一般来说，没有选择正确的CPU并不会有重大的影响（特别是选择386，这样编译出来的核心也许会比较小但它的速度可能就会变慢了）。所以，最好要知道您的CPU是哪一种。不过，如果您的gcc编译器是2.7.0版以前的。那么只能选择386或是486。

#High Memory Support-----内核支持的最大内存数，缺省为1G。可以支持到4G、6.4G，一般可以不选。

#Math emulation-----这项询问是否需Linux核心模拟数学浮点运算器。如果有486Dx、AMD以及Pentium机器的话，这个选项就不必选了，因为它们都有内建的浮点运算器。协处理器是在386时代的宠儿，现在早已不用了。不过，对于有内建浮点运算器的人来说，选了这个选项并不会因此让内建的浮点运算器失效。但它会增大核心约45KB。

#MTRR (memory type range register) support-----选择该选项，系统将生成/proc/mtrr文件对MTRR进行管理，供X server使用。同时用来启动pentium pro和pentium II 的特殊功能，如果你用的不是这类CPU就选N，否则也仅仅是使内核变大而已。

#Symmetric multi-processing support-----对称多处理支持。除非有多个CPU，否则就不用选了。

3. Loadable module support对模块的支持.

首先，了解一点关于模块的知识。模块就像你特意插入核心中的某些东西，如果办公室有一个小网络并且有时想用一下(但并不经常)，也许你想把网卡编译成一个模块。使用这个模块，机器必运行和存取/lib下模块，意思是驱动程序(IDE,SCSI等但必须是NFS支持的网卡)，文件系统(通常是ext2但也可以是nfs)和核心类型(最好是elf)必须编译在内核并且不能是模块，模块只有核心引导时才起作用，驱动程序(来网络)的存取，和文件系统安装。这些文件必须编译在核心内否则将不能安装启动分区。如果安装启动分区和网络，你需要网络系统文件，和已经编译的网卡。为什么要使用模块？模块化使核心变的更简捷，它减少核心释放大量的受保护的内存空间。模块的安装和卸载使用的空间是可重复分配利用的。如果你打开机器有90%以上的时间用到一个模块，编译它。运用这类模块是浪费内存的，原因是一旦你编译了模块它们同样将占用大量的内存，核心需要一些代码来挂上模块。记住，核心在保护空间运行，但模块并不是。这么说，并不经常使用我的设备，把它编译成只支持ext2,ide和elf。而一直使用的网

这或那用上那么几分钟。严格的说,这样做会使核心增大许多而降低它的执行速度。这时我们就可以把这些可能会用的驱动程序编译成一个一个的模块，在需要用的时候才用insmod这个指令加入核心，不用的时候也能rmmod把它从核心移除，或是用lsmod察看目前所载入的模块。这里面有三项：

#Enable loadable module support-----除非准备把所有需要的内容都编译到内核里面，否则该项应该是必选的。

#Set version information on all module symbols-----通常，我们更新核心版本之后，模块要重新的编译。这个选项使您不必更新编译模块而能使用以前的模块。可以不选它。但如果您选y，则按照它的说明，您必须有genksyms这个程序（可用whereis指令查看有无此程序）。

#Kernel module loader-----让内核在启动时有自己装入必需模块的能力，建议选上。

注意：在开机就会 mount 上来的 partition 的 FS、device driver 记得要 compiler 进 kernel，不能把它弄成 modules。请不要夸张到为了完全模块化而忘了把ext2fs和IDE dirver compiler 进 kernel 里。

4. General setup普通的属性设置

这部分内容非常多，一般使用缺省设置就可以了。下面介绍一下经常使用的一些选项：

#Networking support-----网络支持。因为在Linux里面，有虚拟的网络设备(lookback)，可以模拟整个网络。而且，一些程序需要它。必须，没有网卡也建议你选上。注意：选N，则7.(Networking options 网络选项)和11.(Network device support网络设备支持) 不会出现。

#Limit memory to low 16MB -----大部分的人这一选项 N。除了主机板没有办法处理16MB以上的内存，或者有超过16MB以上的内存但却常常发生一些很奇怪的问题。这时，您可以试试这个选项。有些主机板对超16MB内存的处理并不是很好，通常这些都是旧型的主机板。还有，在说明文件中有提到，如果内存超过64MB的话，用LILO加一些参数给Linux核心（例：mem=80M），并且把您主机板上的Cache加到512K。这样，整体效率才能提升。

#PCI support -----PCI支持。如果使用了PCI的卡，当然必选。

#PCI bios support -----主机板是否有PCI界面。如果有，则您必须回答y。

PCI是586电脑的主要界面（一些486主机板上也有），这个界面能让您插入所谓的PCI显示卡，或是PCI的网络卡等。这种界面是现在电脑的主要趋势，因此如果有PCI的插槽。您就可以选Y。除了一些很旧很旧但有支持PCI的主机板外（这些有bugs的旧型主机板可能会因为这个选项而让核心挂掉）。

#PCI access mode (BIOS, Direct, Any) [Any] -----设置Linux探测PCI设备的方式。选择“ BIOS ”，Linux将使用BIOS；选择“ Direct ”，Linux将不通过BIOS；选择“ Any ”，Linux将直接探测PCI设备，如果失败，再使用BIOS。

#Support for hot-pluggabel devices -----热插拔设备支持。支持的不是太好，可不选。

#PCMCIA/CardBus support-----PCMCIA/CardBus支持。有PCMCIA就必选了。

#PCI bridge optimization (experimental) -----在某些支持BIOS上，它能让存取速度加快，建议是选Y。

#Backward-compatible /proc/pci-----设备兼容，自己看help。

#System V IPC 如果将来想编译dosexmu（DOS模拟器），则这个选项一定要选，它是一个让各个程序（process）同步且能彼此交换数据的函数库和一些系统的调用，没它，很多的程序将会无法执行。

#BSD Process Accounting-----

#Sysctl support-----除非你的内存少的可怜，否则你应该启动这个功能，启用该选项后内核会大8K，但能让你直接改变内核的参数而不必重新开机。

#Kernel support for A.OUT binaries -----a . out的执行文件是比较古老的可执行码，用在比较早期的UNIX系统上。Linux最初也是使用这种码来执行程序，一直到ELF格式的可执行码出来后，有愈来愈多的程序码随着ELF格式的优点而变成了ELF的可执码。将来势必完全取代a . out格式的可执行码。但目前由于沿有许多的程序还没有取代过来，所以只好选择Y，等将来有一天，全部的程序都变成了ELF的天下时，那时再disable掉。

#Kernel support for Linux/Intel ELF binaries -----由上所述，这个当然y哩，因为目前gcc - 2.7.0以上的都有支持ELF了，如果没有选择这一项，可能会使用相当多的程序因此无法执行。

注意：编译模块成ELF和编译支持ELF二进制。不编译适当的支持 ' ' gotcha ' ' 是明智的，如果机器结构是Pentium或486你将得到高效的代码，但一个386的核心将运行在32-bit compatible clone；一个Pentium核心将不。为大多机器制作一张紧急启动盘，最好在386下编译，而386并不能运行在Pentium下编译的核心。

另外一点要注意的，你不能同时把 a.out 和 ELF 支援编译成 modules，一般把a.out编译为modules，把ELF编译为核心。

#Kernel support for JAVA binaries ----- 这一项是正在做测试中的产品，但是如果想写有关Java的程序，希望它能在Linux的机器上跑。那么，可以选择把它编成一个模块或是直接把它编进核心里。

#Power Management support -----电源管理支持。

##Advanced Power Management BIOS support-----高级电源管理BIOS支持。这通常是用在笔记本电脑上的东西，如果您有APM的BIOS，支持省电的设备的（有电池的那种），那么您可以选上这项，一般人这一项是选n，以避免一些可能会发生的问题。后有8个选项。

#Parallel port support -----串口支持。

5. Plug and Play configuration即插即用支持

Linux对即插即用目前支持的不如Windows，好有些情况下会和其他设备产生冲突（I/O，DMA，IRQ等）。这个选项对PCI设备没有影响，因为他们天生就是PNP设备。。

#Plug and Play support (CONFIG_PNP) ----- 选择“y”，内核将自动配置即插即用设备。原来PNP还有这个意思。

ISA Plug and Play support ----- 选择“y”，内核将自动配置基于ISA总线的即插即用设备。

6. Block devices块设备支持

这个就得针对自己的设备情况来选了：

#Normal PC floppy disk support (CONFIG_BLK_DEV_FD) [Y/m/n/?]-----普通PC软盘支持。

#Enhanced IDE/MFM/RLL disk/cdrom/tape/floppy support -----选择“y”，内核将提供对增强IDE硬盘、CDROM和磁带机的支持。在硬盘没有做得很大的时候，一般的IDE卡和BIOS只能支持小于540MB的硬盘。不但如此，那时也只能支持二颗硬盘。但现今的硬盘动不动就是1GB以上，今年主流是30~50G，而且常常都会超过一二颗硬盘。如此一来，新的主机板就开始支持加强型的IDE界面（Enhanced IDE），以支持到540MB以上的硬盘。所以，如果您的IDE界面是Enhanced的，请您选Y，底下就会出现八部分IDE界面的选项，这些选项能加快您的IDE界面的速度和对某些芯片做一些最佳化。但如果您的硬盘或光盘全都是SCSI界面的，那么选N以跳过下面选项。

#use old disk-only driver on primary interface-----通常是选择N。因为我们有其他新的drivers可用。这个选项的意思是说，如果您的IDE界面是很旧很旧的那种的。那么，就可以使用这个drives 驱动那个旧型的IDE界面（可装二台硬盘；或是一台硬盘，一台光盘）。而现今流行的 Enhanced IDE则

有两个界面，共可以接四台硬盘。我们稍后会有 driver支持它。

#Include IDE / ATAPI CDROM support-----如果希望核心支持IDE / ATAPI界面的光驱，选择Y。如果有光驱，但它附有一张接口卡，必须把排线接到那张专属接口卡上；或者是接到声卡上的，则这个选项也需要选N，稍后我们会有非IDE的光驱厂牌让我们挑选。现今的光驱通常是IDE / ATAPI界面的，所以这个选项通常是Y。

#Support removable IDE interfaces (PCMCIA) -----这个选项对大部分的人全选n，除非您有PCMCIA的东西，这通常是笔记本电脑上看得到的东西。PCMCIA是一个组织，在以前是设计内存条的。但现在他们对于PC CARDS定了一个标排，并很广泛的应用在laptop的电脑上。不但有所谓的PCMCIA的硬盘，甚至有网络卡、SCSI卡等，不过，大部分的人并不需要这个选项。

这个选项选完后，以下则是Linux核心对几种芯片的IDE界面做修正或是加强它。

#CMD640 chipset bugfix / support----- 很多 486和 586的主机板都是用 CMD640的芯片，它是Neptune芯片和 SIS芯片的结合。不过，这种芯片有它的缺点，在许多的情形下，它会造成数据的流失和错误。如果您选了这一项，则Linux核心会为您小心的寻找这些错误并修正它。而且，它会打开对二个IDE界面的支持。不过，在它的说明文件中提到，如果您的主机板没有PCI界面只有VESA总线界面却希望有这项功能的话，则您必须传一些参数给核心（ideo = cmd640__vlb）。如果不确定上面所说的，选择Y。

#CMD640 enhanced support-----一般来说，对于硬盘的存取速度来说，有所谓的PIO MODES值设定，现今的 IDE界面及 BIOS应该都能侦测到正确的硬盘PIO MODE值了。此值愈高表示硬盘的存取的速度愈快。可是，有些主机板的BIOS还是旧式的，不能抓到比较高的PIO MODE值，如此一来，便不能发挥整台硬盘的效率。这个设定告诉读者说：如果您的IDE界面是CMD640为基础的界面，但是您的BIOS并不能抓到正确的PIO MODE值，那么，这个选项可以自动的找到硬盘正确的PIO MODE值。

#RZ1000 chipset bugfix / support-----这个选项如同前面的CMD640一样。不过，它的芯片是RZ1000的芯片，这种芯片是以Neptune芯片为主的一种芯片，而目、有很多的486和 586的主机板都在使用它。可以查查主机板的说明书或是 IDE接口卡的说明书做确定。文件上提到，选择这个将会降低一些速度，但是数据能百分之百的正确。

#Intel 82371 PIIX (Triton I / II) DMA support-----对于586的Pentium电脑来说，有相当多的主机板都是用Intel的Triton芯片，使用这种芯片的最大好处是支持直接内存存取DMA，而节省您的CPU时间。在以往还没有DMA这个东西时，读取硬盘需要耗用许多的CPU时间。如此一来，CPU被占用，就不能充分的发挥它的功效。后来，DMA出来后，硬盘的读取便靠Triton或其他有支持DMA的IDE界面的芯片，直接与它们做沟通，而节省了大量的 CPU时间，但这必须您的硬盘和主机板有同时支持 DMA的 IDE界面的芯片，直接与它们做沟通而节省了大量的 CPU时间。但这必须您的硬盘和主机板有同时支持DMA MODE才行。

##other IDE chipset support-----

如果这上选项选y，则会出现下列六种其他的芯片或厂牌供您选择。

* NOte : most of these also require special kernel boot paraaeters

ALI M14xx support

DTC - 2278 support

Holtek HT65608 support

PROMISE DC4030 support

QDI QD6580 support

UMC 8672 support

上面这六种厂牌的芯片依硬件配备而使用，但它们有共同的特点就是必须传一些参数给核心。如果找不到您的芯片，那么也没关系。上这些选项只不过对这些芯片做最佳化罢了。

Additional Block Devices其他的块设备

#Loopback device support-----大部分的人这一个选项都选N，因为没有必要。这个选项的意思是说，可以将一个文件挂成一个文件系统。如果要烧光盘片的，那么您很有可能在把一个文件烧进去之前，看看这个文件是否符合ISO9660的文件系统的内容，是否符合您的需求。而且，可以对这个文件系统加以保护。不过，如果您想做到这点的话，您必须有最新的mount程序，版本是在2.5X版以上的。而且如果您希望对这个文件系统加上保护，则您必须有des.1.tar.gz这个程序。注意：此处与网络无关。

#Multinle devices driver support-----这个选项可以让把整个硬盘分区变成一个单独的区块设备，您必须有md035 . tgz这个程序。而且在做这件事之前请将您的硬盘备份，因为它尚在测试阶段。一般人对这个选项是选N。

#RAM disk support-----如果使用过DOS下的ramdrive程序，应该能了解这个选项的意义。它可以把内存当成硬盘来做存取就如同一般的硬盘一样，可以format它，或是放一些文件在里头。然后，当您关机这些数据也随着之而去了。如果的RAM够大，可以考虑玩玩这选项，但一般人都不需要。

#XT hard disk support-----支持XT的古董硬盘，这是IBM电脑时代的东西，如果您还有这种很旧很旧的硬盘。那么，您可以把它编进核心或是编成一个模块。大部分的人这个选项都是选择N的。

#Compaq SMART2 support-----

#Mulex DAC960/DAC1100 PCI RAID Controller support-----RAID镜像用的。

#Logical volume manager (LVM) support-----逻辑卷管理支持。

#Multiple devices driver support-----多设备驱动支持。

#RAM disk support-----RAM盘支持。

7. Networking options 网络选项

这里配置的是网络协议。

#Packet socket ----- 选择“ Y ”，一些应用程序将使用Packet协议直接同网络设备通讯，而不通过内核中的其它中介协议。

#Kernel / User Network link driver-----这个是在测试中的程序码，一般人不需要用。依它的说明，它允许在核心、模块或程序间的某些部分间，彼此做双向的沟通。如果想使用arpd，则这个程序码就要加进核心里。

#TCP/IP networking-----选择“ Y ”，内核将支持TCP/IP协议。这个选项无论如何请您选择Y，即使没有网络卡，或是没有连到网络上的设备，在linux上仍有所谓的lookback设备而且有些程序需要这个选项。在说明文件中提到，如果您没有打开这个设定，则X - window system可能会有问题（因为它也需要 TCP / IP）。

#Network firewalls-----选择“ Y ”，内核将支持防火墙。Firewalls依英文看是防火墙。在网络愈来愈发达的今天，网络安全的考虑也愈来愈重要了。在局域网上找一台电脑来保护自己的考虑也愈来愈多了。可以在一局域网上找一台电脑来保护自己区域内的电脑。这样的结果是，所有外部的电脑如果要连进内部的电脑就必须通过这台装有Firewalls电脑的同意。所以，如果您希望这台电脑有着过滤网络的功能的话，那么这个选项要选Y。而且，等一下有个IP firewalling的选项也要选y；但下面的forwarding / gatewaying要选n，如此才能让它正常动作。大部分的人这个选项选N。

#Network alasing-----允许有多个IP地址。

#IP : forwarding / gatewaying ----- 和 Firewall相反，这个选项是用来疏导网络的。一个gateway（也就是router），要帮忙疏导两个网络间的数据传送。这台机器必须要有两张网络卡，连接两个个同的网络，做疏导网络的工作。如果选择了这个选项，则表示想让这台负责做router。那么，就必

须有两张网络卡了。另外有一种情形是，如果您有MODEM（通过串列界面以SLIP和PPP协议）和网络卡，并用它们来连上Internet。这时您也一样可以执行IP - routing服务，也需把这个选项打开。

#IP：multicasting-----所谓的multicasting是群组广播，它是用在视频会议上的协议，如果想送一个网络封包（网络的数据），同样的一份数据将送往十部机器上。您可以连续送十次给十台机器（点对点的传送），也可以同时送一次，然后让十台机器同时接收到。当然后者比前者好，由于视频会议要求是最好每个人都能同时收到同一份信息，所以如果您有类似的需要，这个选项就要打开。同时您还必须去找相关的软件。

#IP：accounting-----如果您打开这个选项，您就可以在 / proc / net下看到系统对于整个网络状况的纪录。所以一般的人这个选项都是选y。而且，如果您设计把这台Linux机器当router用，那么读者可以因为这个选项而获得许多有关于网络IP控制的信息和它的输送情形。不过，您必须在底下的选择中选择proc系统（其实proc文件系统一定要选，不选很多程序会不能用！）

#IP：aliasing support-----也许您只有一张网络卡，但经由这个设定，您可以拥有数个IP地址。假设您已经有一个IP地址了，您还想再加入其他的IP地址，这时，您可以依下面的程序来做。

在 shell下键入：

```
sunlly： / #ifconfig eth0：0其他的IP地址（这个IP地址不能与其他机器重复）
```

```
sunlly： / # route - add - host其他的IP地址 dev eth0：0
```

如此您就可以同时拥有两个IP地址了。当然，如果想把这个IP地址去除，那么可以键入：

```
sunlly： / # ifconfig eth0：0 - IP地址
```

这样您就可以把加入的IP地址去除。不过在使用此选项前，前面的那一个aliasing Network选项也要选上去。

#IP：PC / TCP compatibility mode-----大部分人都选n。除非在使用DOS下的NCSA - TCP / IP软件连进Linux机器时遇到了困难，或者有不相容的情形出现。这时，您可以试着把这个选项打开，看看是否能解决这个问题。

#IP Reverse ARP-----如果您的Linux希望提供bootd的服务，就是让没有硬盘或软盘也能够开机并且上网络，只要它们有网络卡有连接到网络的话。此时，您必须执行一个指令叫rarp来设定哪些电脑的网络卡可以如此。不过一般人没有这个需要，所以答n。

#IP：Disable Path MTU Discovery（normall yenable）----- 大部分的人这个选项是选择N。除非发现用DOS下的ncsa的telnet程序连到Linux机器上出问题。这是很多人的问题，如果发生了DOS下的telnet程序不能连进Linux时，除了可以改用Nsysutel的telnet程序来解决外，还可以在编译核心的时候，把这一项选Y。MTU（Maximal Transfer Unit）叫做最大的传输单位，是说我们一次送往网络的信息大小。而Path MTU Discovery的意思是，当Linux发现一些机器的传输量比较小时，我们会分送网络信息给它。如此可以增加网络的速度，所以我们大部分都选N，也就是Enable。

#Ip：Dorp source routed frames-----通常我们一个网络的封包在丢出去后就不管它了，不过，在TCP / IP协议里，您可以设定让那些帮您绕路的机器回送一个是否这个封包已经送达了的消息。不过，这会导至网络安全上的问题，所以很少用，一般来说我们选择Y。

#IP：Allow large windows（not recommended if < 16Mb of memory = ）-----如果有超过16MB以上的内存，那么建议打开这个选项，可以增加传输的速度。在一般长距离的网络传输下要预备传输的数据可以先储存在缓冲区，等到对方的回应时再一次会过去。因此，您必须有内存来作为缓冲区。

#The IPX Protocol-----IPX是一种传输协议，它是Novell的一种网络协议，通常用在区域或是Windows的网络下。如果您希望Novell的机器资源共享（例如用他们的打印机或是硬盘），那么这个选项则选y。至于存取文件的格式是NCPFS的格式，稍后把这个文件系统选上来，以便支援这个文件系统。如此一来，您就可以通过Novell的IPX通讯协议去存取它们的数据了。或者，您希望从dosemu（DOS的模拟器）里用IPX协议，这时也要把它选进来。

#Full internal IPX network-----提供了一个完整的内部IPX网络，预设选项是N，因为它可能会让一些应用的服务程序（RIP / SAP）当掉。

#Appletalk DDP-----AppleTalk是存在于苹果机上的一种通讯协议，用来苹果电脑之间的网络通讯，通过AppleTalk，彼此的电脑间可以打印和分享文件。如果您需要连上这样的网络，可以把这个选项打开，如此就能加入他们与这些电脑做沟通了，或者把这项编成一个模块亦可。

#Amateur Radio AX.25 Level2-----

#Bridging (EXPERIMENTAL) -----

选这两个可以让Linux变成一个网络上的网桥，用来做不同网络间的沟通，通常一般人不需要。

#Qos and/or fair queueing (服务质量公平调度) 也支持了，还有kHTTPd，不过这些都还在实验阶段。

8. Telephony Support电话支持

原来是Linux下可以支持电话卡，这样你就可以在IP上使用普通的电话提供语音服务了。记住，电话卡可和modem没有任何关系。

9. SCSI support SCSI设备支持

如果有 SCSI 设备，就回答Y。现在一般PC机不会有SCSI。接着会有提示要求更进一步的资讯，像是你是否要支援光驱，硬盘，还有你使用的是那一种 SCSI 界面卡。这部份请参阅SCSI-HOWTO，有更详细的说明。如果你的启动分区是SCSI设备，不要选择SCSI模块支持。在一般的SCSI后是SCSI低级设备驱动程序。再次重申，模块仅仅是用在不在启动分区的设备。

#SCSI disk support-----指硬盘而言，如果有SCSI硬盘，那么就要选这个选项。

#SCSI tape support-----指磁带机而言，如果您有SCSI的磁带机，那么就要选这个选项。

#SCSI CDROM support-----指CDROM，如果您有SCSI光驱，这一项一定要选。

#SCSI generic support-----指其他有关SCSI的东西，也许您有一台SCSI的扫描器或是烧录机，或是其他有关SCSI的配备，您就要选这一项。而且，除此之外，您还必须准备关于这些配备的软件。

##Some SCSI devices (e.g. CD jukebox) support multiple LUNs

#Probe all LUNs on each SCSI device-----通常这个选项大部分的人都不会选。我们举个例子来说，如果您的SCSI光驱是那种多片装的，就是一台光驱，但可以一次放好几片光盘片的那种。这种我们叫做Lun。

#Verbose SCSI error reporting (kernel size+=12K) -----如果认为您的SCSI硬件配备有些问题，想了解一下它出现的错误信息。那么您可以把这个选项选y，Linux核心会告诉您有关于您的SCSI配备的问题（如果有的话）。不过，它会增加核心约12KB左右。

##SCSI low—level drivers

下面总共有接近30张的SCSI卡，您可以依需求做选择SCSI卡牌子。

c AIA1542 support

AdaPtec AIHA1740 support

AdaDtec AHA274X / 284X / 294X support

AdaPte 7000FASST SCSI support

AdaPtec AHA152X / 2825 support

Advansys SCSI support

Always IN2000 SCSI support

Advansys SCSI support

Always IN2000 scsi support
 AM53 / 79C974 PCI SCSI support
 Buslogic SCSI Support
 DTC3180 / 3280 SCSI support
 EATA ISA / EISA (DPT PM2011 / 021 / 012 / 022 / 122 / 322) support
 EATA - DMA (DPT , NEC , AT & T , SNI , AST , Olivetti , Alphatronix) support
 EATA - PIO (old DPT PM2001 , PM2012A) support
 Future Domain 16xx SCSI support
 Generic NCR5380 / 53c400 SCSI support
 NCR53c405a SCSI support
 NCR53c7 , 8xx SCSI support
 NCR53CSXX SCSI support
 IOMEGA Parallel Port ZIP drive SCSI support
 PAS16 SCSI Support
 Qlogic FAS SCSI support
 Qlogic ISP SCSI support
 Seagate ST-02 and future Domain TMC-8xx SCSI support
 Trantor T128 / T128F / T228 scsi support
 Ultrastor 14F / 34F support
 Ultrastor SCSI support

10. I2O device support

这个也不清楚，帮助里说是这个需要I2O接口适配器才能支持的，在智能Input/Output (I2O) 体系接口中使用，又是要硬件，不选了。

11. Network device support网络设备支持

上面选好协议了，现在该选设备了，内容多得很。还好里面大概分类了，有ARCnet设备、Ethernet (10 or 100 Mbit)、Ethernet (1000Mbit)、Wireless LAN (non-hamradio)、Token Ring device、Wan interfaces、PCMCIA network device support几大类。耐心点，一般说来都能找到自己用的网卡。如果没有，你只好自己到厂商那里去要驱动了。如果这个选项没有打开的话，那么以下的选项将不会出现。它是在选择网络卡或是网络的设备。例如，PLIP，PPP，SLIP，还有各式各样的网络卡，所以这个选项通常是选y。

#Dummy net driver support-----如果有SLIP或PPP的传输协议，那么要把这一项打开。因为一来它不会让您的Linux核心增大。二来，对某些应用程序来说，它可以让我们模拟出来的TCP / IP环境更像TCP / IP环境。如果您没有SLIP或PPP协议，就不用打开了。

#EQL (serial line load balancing) support-----如果有两个MODEM，两条电话线，而且用SLIP或PPP协议，可以用这个Driver以便让您的MODEM有两倍的速度。当然，在网络的另一端也要有同样的设备。

#PLIP (parallel port) support-----依字面上看，它是一种利用打印机的接口（平行接口），然后利用点对点来模拟TCP / IP的环境。它和SLIP / PPP全都属于点对点通讯，您可以把两台电脑利用打印机的连接接口串联起来，然后，加入此通讯协议。如此一来，这两部电脑就等于一个小小的网络了。不过，如果电脑有提供打印服务的话，这个选项最好不要打开，不然可能会有问题（因为都是用平行接口）。

#PPP (point - to - point) support-----点对点协议，近年来，PPP协议已经慢慢的取代SLIP的规定了，原因是PPP协议可以获取相同的IP地址，而SLIP则一直在改变IP地址，在许多的方面，PPP都胜过SLIP协议。

#SLIP (serial line) support-----这是MODEM族常用的一种通讯协议，必须通过一台Server（叫ISP）获取一个IP地址，然后利用这个IP地址，可以模拟以太网，使用有关TCP / IP的程序。

##Ethernet (10 or 100Mbit)

如果您在学校接了校园网络并且使用网络卡，那么这个选项一定要选y，否则以下对网络卡的选择将不会出现。或是您有网络卡，这时您同样的也要选y。之后，下面会列出许多网络卡让您选择。像我们平常用的都是NE2000相容卡。

#3COM cards

#AMD LANCE and PCnet (AT1500 and NE2100) support

#Western Digital / SMC cards

##other ISA Cards (CONFIG . ISA) -----选y，以下才会列出有关ISA的网络卡。包括NE2000的兼容卡。

Cabletron E21xx support

DEPCA , DE10x , DE200 , DE201 , DE202 , DE422 support

EtherWORKS 3 (DE203 , DE204 , DE205) support

EtherExpress 16 support

HP PCLAN + (27247B and 27252A) support

HP PCLAN (27245 and other 27xxx series) support

HP 10 / 100VG PCLAN (ISA , EISA , PCI) support

NE2000 / NE1000 support

SK . G16 support

#EISA , VLB , PCI and on board controllers -----选择网络卡，包括直接附在主机板上的那种。如果选择y，则底下会列出其他的网络卡让您做选择，这些卡对于一般人来说很少会去用到。所以大部分的人这项是选N的。

#Pocket and portable adaptors-----通常用在可携式的电脑上，这类型的网络卡（口袋型的），由于体积很小在安装和取下方面很方便，因此笔记本相关电脑上便常常采用这种网络卡。

#Token Ring driver support-----Token Ring是IBM电脑上的网络。它叫令牌环网络，和以太网是很类似的东西。如果您希望使用的Token Ring网络卡以便连接到这种网络，那么选Y，一般人都选N。

#ARCnet support-----这也是一种网络卡，通常一般人用不到，所以选n。如果您有这样的网络卡，请看Documentation / networking / arcnet . txt的说明。

12. Amateur Radio support 业余无线广播

可以用来启动无线网络的基本支持，目前的无线网络可以通过公众频率传输数据，如果你有此类设备就可以启用，具体请参考AX25和HAM HOWTO 文档。

13. IrDA (infrared) support 红外支持

14. ISDN subsystem

如果使用ISDN上网，这个就必不可少。ISDN (Integrated Services Digital Network)，它的中文名称是综合数字服务网络，是一个利用电话线，把声音，影片信息以数字的方式传送的数字网络，它需要电话交换机设备有支持ISDN，这通常需要电信局来做安装，对于在家工作的人来说，ISDN

可能是最舒适最便宜的一种方式，因此有愈来愈多的人使用它。不过，除非是公司，不然一般人很少会使用到ISDN的，所以这部分的选项大都选N。如果您选择Y，则下面会出现一些有关ISDN的问题。如果需要用到ISDN，可以去看看杂志的介绍。只要是有关网络的杂志应该都会有介绍。还需要启用Support synchronous PPP选项（参考PPP over ISDN）。

15. Old CD-ROM drivers (not SCSI、 IDE) 非SCSI/IDE口的光驱

如用IDE的CD-ROM，不选。

以下是选择非 IDE / ATAPI和 SCSI界面的光驱，这些光驱通常有自己专属的接口卡也是比较旧型的光驱类型。如果有这些光驱，则这个选项要选y，否则选n。如果您选择n，则会跳过以下光驱的选项。

Aztech / orchid / okano / Wearnes / IXC / CyDROM CDROM support

Goldstar R420 CDROM support

Matsllshita / panasonic / Creative , longshine , TEAC CDRW Support

Mitsllmi (no XA / MULTIsession) CDROM Support

Mitsumi (XA / Multisession) CDROM support

optics Storage DCLPHIN 8000AT CDROM support

Philips / LMS CM206 CDROM support

Sanyo CDR - h94A CDROM sunnort

Soft configurable CDROM interface card support

Sony CDU31A CDROM sunnort

Sony CDU535 CDROM support

16. Character devices字符设备

所谓字符设备通常是指以字符为单位做处理的设备，例如终端机就是其中一项。原则上，我们对于这些选项的选择也是以预设为主。这个内容又太多了，先使用缺省设置，需要的话自己就修改。把大类介绍一下吧：

#Virtual terminal ----- 选择“y”，内核将支持虚拟终端。

#Support for console on virtual terminal-----选择“y”，内核可将一个虚拟终端用作系统控制台。

#Standard/generic (dumb) serial support-----选择“y”，内核将支持串行口。标准序列接口的选定。如果您是用serial的鼠标（大部分的人都是用这个），或

是MODEM的话，则这一项一定要选。大部分的人这一项都选y。

#Support for console on serial port-----选择“y”，内核可将一个串行口用作系统控制台。

##I2C support-----I2C是Philips极力推动的微控制应用中使用的低速串行总线协议。如果你要选择下面的Video For Linux，该项必选。

#Mice鼠标。现在可以支持总线、串口、PS/2、C&T 82C710 mouse port、PC110 digitizer pad-----根据需要选择。非串行的鼠标，支持象PS/2。看一看你鼠标的键。许多双键鼠标是PS/2，即使它们的连接看起来象串行的。

#PS/2 mouse (aka "auxiliary device") support ----- 如果用户使用的是PS/2鼠标，则该选项应该选择“y”。

#Joysticks-----手柄。即使在Linux下把手柄驱动起来意义也不是太大，游戏太少了。

#Watchdog Cards：虽然称为Cards，这个可以用纯软件来实现，当然也有硬件的。如果你把这个选中，那么就会在你的/dev下创建一个名为watchdog的文件，它可以记录你的系统的运行情况，一直到系统重新启动的1分钟左右。有了这个文件，你就可以恢复系统到重启前的状态了。

#Video For Linux-----支持有关的音频/视频卡。

##Ftape, the floppy tape device driver---

#Digiboard PC / Xx Support-----这是一张叫 Digiboard PC / XX卡的 driver，这种卡上面有很多个 serial port的插槽（一般来说只有两个），可以用来连接很多个MODEM，在民间的BBS站很常用到，如果您有这样的东西，您必须选这项为y。有兴趣的读者可以读读Documentation / digiboard . txt的内容。

#Cyclades async mux support-----同上，这也是一种能接很多个serial port插槽的卡的驱动程序。

#Stallion multlport serial support-----同上，这也是其中一种卡。

#SDL RISCom / 8 card support-----这也是其中的一个支持muti - serial卡的driver。

#Parallel printer support-----有打印机的或是使用到并行接口的人这一项一定要选。除非是用serial的打印机。还有如果您有使用PLIP，那么这项也请选上。

#Mouse support-----大部分的人这一项并不用选y。因为大部分的人是用serial的鼠标，除非有些人是用一种附有接口卡的鼠标，这时这个选项才要选上，如果您选y，则底下会列出您的BUS鼠标所用的接口卡。

#support for user misc device modules-----除非您有所谓的触摸式显示器或是光笔等东西，否则这一项选n。

#QIC - 02 tape support-----非 SCSI界面的磁带机，除非您有，否则选 n。

#Ftape (QIC - 80 / Travan) support-----如果有磁带机，而这个磁带机是接在软盘控制卡上，这个选项才要选y。

#Watchdog Timer support-----一般人不需要这个选项，如果您选上这个选项，则您要用mknod在 / dev下建一个watchdog的文件。请看Documentation / watchlog . txt的解释。

#Enhanced Real Time Clock Support-----关于系统上Clock的东西，您必须自己用mknod在 / dev / 下建立一个文件叫rtc。如此一来，在 / proc / 下将可以看到rtc的信息。

有关于rtc的内容请看Documentation / rtc . txt。一般是选n。

17. USB support USB支持

很多USB设备，比如鼠标、调制解调器、打印机、扫描仪等，在Linux都可以得到支持，根据需要自行选择。

18. File systems文件系统

Linux上有支持约二十几种的文件系统，有支持某个文件系统的意思是，可以存取某个文件系统的数据或是做拷贝动作。在这些文件系统中，通常的选择方法是按照原来预设的方式，不过，在其中，EXT2FS那个选项无论如何一定要选，因为那是Linux系统所使用的文件系统。其他的则依需求做选择。

通常是ext2而让其余的使用模块。

#Kernel automounter support-----选择“ y ”，内核将提供对automounter的支持，使系统在启动时自动 mount远程文件系统。

#Standard (minix) -----新的套件不再建立 minix 文件系统，而且很多人不使用它，但是把它配置在核心里仍然是个好主意。某些 rescue-disk 程序会用到它，而且仍然有许多磁片可能用 minix 文件系统，因为 minix 文件系统对于处理磁片方面是最好的。当初Linus是因为对Minix这个小型的操作系统有很深的经验，所以才写出Linux这个操作系统。Minix文件系统通常用在磁盘上，有时会用到它。

#Extended fs ----- 这是扩充文件系统的第一版，现在已经不再使用。

#Second extented fs-----这是现在新发行的套件所广泛采用的文件系统，你可能会有一种。这个是linux文件系统，请务必选y，如果问我说选n会有什么后果，我也不知道。除非您能把 Linux装在 DOS的目录下。

#xiafs filesystem-----这个文件系统曾经一度很普遍，但是在写这份文件时，我已经不知道有任何人在使用它了。

#DOS FAT fs -----DOS FAT文件格式的支持，可以支持FAT16、FAT32。这个选项是DOS的文件系统，如果您没有选y，则下面的MSDOS，VFAT，umsdos将不会出现。

#msdos-----DOS文件系统的格式。如果你想要在 linux 下使用你硬盘中的 MS-DOS 分割区，或是想将用 MS-DOS 格式化的磁盘挂进来的话，回答y。

#VFAT (Windows95) fs -----windows95所支持的文件系统，是我们常说的vfat文件系统，如果您的系统中装有windows95，那么选择这个文件系统将对以让您看到windows95的长文件名。

#umsdos : UNIX like fs on top of std MSDOS FAT fs ----- 如果把Linux装进DOS的一个目录下那么您则要选择这一项。不建议这样做，因为如此一来，就不能看到长文件名了。而且效率上，并不是很好。

#ums-dos-----相当 slick 的文件系统，它能使 MS-DOS 文件系统拥有更多的特性，像是长档名等等。这对那些不使用 MS-DOS 的人（像我）并不是很有用。

#/proc-----这是最 slick 的文件系统之一。它不是你硬盘分割区里的任何东西，不占用硬盘的空间，而是核心与程序之间的文件系统介面，它表示的只是内存里头的状况和各个程序执行的情形，它也记录了您硬件上配备。。许多程序工具（像 ps ）都会用到它。如果已经将它安装好了，有空不妨试试看 cat /proc/meminfo 或者是 cat /proc/devices 。有些 shells ，像是 rc ，会用 proc/self/fd （在其它系统上为 /dev/fd ）来处理输出。几乎可以确定你在这里得要回答y，有许多重要的 Linux 标准工具是靠它来运作的，否则有些指令会出问题。

#Root file system on NFS-----一般不选，除非您的电脑上没有硬盘，希望通过网络由别人的硬盘开机过Linux如此一来才有需要选这项。同时对方也要执行rarp的服务。

#System V and Coherent-----这是为 System V 以及 Coherent 的分割区而设的。如果希望支持System V或Xenix的相关UNIX系统的FS并读取它们的数据，那么才有必要选这个，否则一般来说这个选项是选n的。

#Quota support -----Quota可以限制每个用户可以使用的硬盘空间的上限，在多用户共同使用一台主机的情况中十分有效。

#ISO 9660 CD-ROM file system support-----光盘使用的就是ISO 9660的文件格式。

#Mandatory lock support-----有些很特殊的database应用软件会用到它，一般人这个选项是选n。而且，如果选y的话，必须有最新版的NFS软件，最新版的samba软件等。

#NTFS file system support-----ntfs是NT使用的文件格式。

#UFS filesystem support-----这是BSD，SunoS，FreeBSD，NetBSD或Nextstep所使用的文件系统。如果您在电脑上有这些操作系统的话，那么可以选这一项。否则一般人都选n。

##Network File Systems-----网络文件系统

NFS-----如果你在网络环境下而且想要分享档案，回答 y。如果希望挂上别的电脑的文件系统，那么这个选项一定要选进去。它可以让您利用网络把别人的硬盘当成自己的来使用（把它变成一个目录）。对于一般人来说，这个选项是选y。

#SMB filesystem support-----这个文件系统让您可以挂上windows95或windowsNT的文件系统，也就是您也可以抓到在windows下，网上邻居上的电脑。

#SMB long filename support-----支持windows95的长文件名。

#NCP filesystem support-----NCP是一种网络的通讯协议，用在跑IPX协议上，它可以利用IPX协议让两台电脑之间的文件共享，并做沟通。如果您想挂上有关Novell的Netware文件系统，那么这个选项就选上去吧。

##Partition Types-----分区类型，该选项支持一些不太常用的分区类型，用户如果需要，在相应的选项上选择“y”即可。

##Native Language Support-----本地语言支持

附：不知道需要那些文件系统怎末办？

键入 mount 它看起来会像这样：

sunlyy% mount

/dev/hda1 on / type ext2 (defaults)

/dev/hda3 on /usr type ext2 (defaults)

none on /proc type proc (defaults)

/dev/fd0 on /mnt type msdos (defaults)

仔细看看每一行；在“type”后面的那个字就是文件系统的格式。在这个例子中，我的/和/usr分割区是second extended格式，我使用/proc，而且挂有一张以msdos (bleah)为文件系统格式的磁片。如果你有使用/proc，可以试试 cat /proc/filesystems。它会给你一份目前使用的核心所支援的文件系统列表。

19. Console drivers 控制台驱动

#VGA text console ----- 选择“y”，用户就可以在标准的VGA显示方式下使用Linux了。一般使用VGA text console就可以了，标准的80*25的文本控制台。

#Video mode selection support

20. Sound sound声卡驱动

如果你能在列表中找到声卡驱动那自然最好，否则就试试OSS了。阅读帮助文件从列表中小心的选取。确信为你声卡真确的选择了I/O和IRQ。声卡的MPU I/O是0选项。一般是330，如果不对不必担心。模块的好处就是在核心编译以后你还能重新编译、安装模块并挂上核心。

如果有声卡，请去了解一下声卡的IRQ和DMA等信息，并了解是属于哪一种的。现在大部分的人使用的都是Sound Blaster或是它的相容卡。有关这类的信息请看各个声卡的说明书。当选Y时，出现下面的画面。就依您的声卡来做选择吧！

Pro Audio Spectrum 16 support
 Sound Blaster (SB , SBPro , SB16 , clone) support
 Generic OPL2 / 0P13 FM synthesizer support
 Gravis Ultrasound support
 MPU-401 support (NOT for SB16)
 6850 CART Midi support
 PSS (ECho - AD12111) support (NOT for SB16)
 16 bit samplers option of GUS (NOT__GUS__MAX)
 GUS MAX support
 Microsoft Sound System support
 Ensoniq Soundscape support
 MediaTrix AudioTrix Pro support
 Support for MAD16 and / or Mozart based cards
 Support for Crystal CS4232 based (PnP) cards
 Support for Turtle Beach Wave Front (Maul , Tropez) synthesizers

/ dev / dsp and / dev / audio support----- 这个选项通常是必要的。因此大部分的人选y，如果没有这选项，则很多的游戏将没有声音效果。

#MIDI interface support-----支持MIDI界面。

#FM synthesizer (YM3812 / OPL - 3) support

#I / O base for SB Check from manual of the card-----声卡的I / O地址。括号是常用的选项。

#Sound Blaster IRQ Check from manual of the card-----声卡的IRQ，通常是1或5。

#Sound Blaster DMA 0 , 1 for 3-----声卡的DMA，通常是1。

#Sound Blaster 16 bit DMA 5 , 6 or 7 (use for 8 bitcards) (SB . DMA2) -----5

#MPU401 I / O base of SB16 , Jazz16 and ES1688 Check from manual of the card
 -----0

#SB MPU401 IRQ (Jazz16 , SM Wave and ES1688) Use with SB16-----1

#Audio DMA buffer size 4096 , 16384 , 32768 or 65536-----65536

##Additional low level drivers-----如有其他种类的声卡，则这项要选上去，以下会列出其他的声卡供选择。

21. Kernel hacking安全模式

通俗的说，这是windows安全模式,找不到明确解释，就引用这个说法。> 这是从 Linus 的 README 里摘录的：

kernel hacking 配置的细节通常会产生一个更大或是更慢的核心（或者是又大又慢），而且可能因为将一些常式配置成很活跃地去尝试中断一些不良的程序码以便找出核心里的问题，而使得核心变得比较不稳定。所以要产生一个核心 产品 的话，你在这里应该回答n。

配置了这个，即使在系统崩溃时，你也可以进行一定的工作了。

如果对Linux核心很熟，想了解这个操作系统上的很深的信息，则这个选项才打开，将会在 / proc下看到profile这个文件，记录着相当多的信息。

这些信息是供除错用的，对一般人来说，这个选项是选n，以免造成系统不稳固，或是引起其他的问题。

总算配置完了，现在存盘退出，当然你也可以把现在的配置文件保存起来，这样下次再配置的时候就省力气了。存盘退出的文件是一个可以编辑的文本文件。你可以用你的文本编辑器如：Emacs,vi,CrispLite,joe,等等。配置文件在/usr/src/linux目录名字是.config。前面的点使普通的.(ls)无法显示，加上-a选项就能看见。编辑文件中在配置时遇到麻烦的数字。然后，make dep 把你的配置信息从.config传送到特定的子目录完成安装。最后，make clean 准备最后的核心编译。

3、编译内核

编译有关的命令有如下几个：

```
#make dep
#make clean
#make zImage
#make bzImage /make zdisk
#make modules
#make modules_install
#depmod -a
```

第一个命令make dep实际上读取配置过程生成的配置文件，来创建对应于配置的依赖关系树，从而决定哪些需要编译而那些不需要；

第二命令make clean完成删除前面步骤留下的文件，以避免出现一些错误；

第三个命令make zImage和第四个命令make bzImage实现完全编译内核，二者生成的内核都是使用gzip压缩的，只要使用一个就够了，它们的区别在于使用make bzImage可以生成小一点的内核，比如在编译2.2.16版本的内核时如果使用make zImage命令，那么就会出现system too big的错误提示。建议使用make bzImage命令,压缩的核心它将留在/usr/src/linux/arch/i386/boot目录名为zImage。make zdisk写zImage到软盘在/dev/fd0(就是DOS中第一软驱A:)。在开始之前你必须要有张盘在软驱中。你可以同样用make zImage 然后把它拷贝到软盘cp /usr/src/linux/arch/i386/boot/zImage /dev/fd0 记住是高密度的，底密度的720k磁盘将不能启动核心。

后面三个命令只有在你进行配置的过程中，在回答Enable loadable module support时选了"Y"才是必要的，make modules和make modules_install分别生成相应的模块和把模块拷贝到需要的目录中。

严格说来，第七个命令和编译过程并没有关系，它是生成模块间的依赖关系，这样你启动新内核之后，使用modprobe命令加载模块时就能正确地定位模块。

4、启用内核

通常，核心安装叫做vmlinuz。过去Unix使用者共同起了这名字。"z"表示压缩，"v"和"m"意思是"virtual"(虚拟)和"sticky(粘性的)"，各自属于内存和磁盘管理。建议保留vmlinuz核心，直到知道它工作。

为了能够使用新版本的内核，还需要做一些改动：

```
#cp /usr/src/linux/System.map /boot/System.map-2.2.16
#cp /usr/src/linux/arch/i386/bzImage /boot/vmlinuz-2.2.16
```

以上这两个文件是刚才编译时新生成的。下面修改/boot下的两个链接System.map和vmlinuz，使其指向新内核的文件：

```
#cd /boot
#rm -f System.map vmlinuz
#ln -s vmlinuz-2.2.16 vmlinuz
```

```
#ln -s System.map-2.2..16 System.map
```

注意：要保留vmlinuz核心，以下列步骤进行

```
#cp /usr/src/linux/System.map /boot/System.map-2.2.16
```

```
#cp /usr/src/linux/arch/i386/bzImage /boot/vmlinuz-2.2.16
```

```
#cd /boot
```

```
#rm -f System.map
```

```
#ln -s System.map-2.2..16 System.map
```

现在#vi /etc/lilo.conf,增加如下一段：

image=/boot/vmlinuz-2.2.16 是设定为已经安装的核心

label=linux2.2.16 则是由 lilo 用来告诉你现在要启动的是那个核心或作业系统，

read-only

root=/dev/hda2 则是这个特别的作业系统的根目录 /

其中root=/dev/hda2一行要根据需要自行加以修改。

运行：#/sbin/lilo -v 保存执行命令：lilo你将看到核心标签，第一个是星号。如果你没有看到新核心的标签或LILO出现错误，你需要重新对/etc/lilo.conf工作（看下面的LILO分析）。

确认对/etc/lilo.conf的编辑无误，现在重新启动系统：

```
#shutdown -r now
```

不建议使用热启动或ctrl+Alt+del键。在一些情况下，文件系统不完全卸载会损坏打开的文件。在LILO提示时，如果你需要启动旧的核心或使用一些参数启动，如果你没看见启动提示，你可以试用shift或ctrl键，这样启动提示就出现了。一旦出现，按tab看核心标签。输入标签和可选参数启动。通常，在/etc/lilo.conf文件指定的时间后自动启动核心。启动时，你可能看见一些出错信息就象SIOCADDR。这常常显示模块（一般是网络模块）没有引导。处理这事很简单，如果有此一错，"VFS, cannot mount root",你就不要在核心中编译适当的磁盘或文件系统支持。

附录：LILO分析（技术性强，仅供参考）

第一部分 LILO介绍

LILO（Linux Loader）是Linux自带的一个优秀的引导管理器，使用它可以很方便地引导一台机器上的多个操作系统。与其他常用的引导加载程序相比，LILO引导方式显得更具有艺术性，对其深入的理解，将有助于我们方便地处理多操作系统、网络引导、大硬盘及大内存等诸多棘手的问题。

LILO的引导机制-----众所周知，计算机的最初启动是由BIOS控制的，在对一些硬件（如：内存、键盘等）初始化之后，它会试图加载硬盘的主引导记录（MBR）或软盘的引导扇区。MBR可通过两种方式运行，其一是定位到活动分区并加载相应的引导扇区，然后由引导扇区完成该分区内操作系统的基本组件的加载；其二是直接从一指定分区中加载信息，并通过它装入任一分区的操作系统，诸如LILO、OS/2 boot loader及Partition Magic等引导加载程序都可以配置成这种方式。软盘的引导扇区相当于硬盘活动分区的引导扇区，它通常用于装入软盘上的操作系统。由此可见，只要把LILO安装在MBR、活动分区或者引导软盘上，就能接管计算机的控制权，然后由LILO完成后继的引导过程。LILO中建有一个引导表地址编码，借此它的引导程序就能定位到Linux的内核文件，这种地址编码既可以按照柱面/磁头/扇区(CHS)模式，又可以采用LBA的线性块号模式，因此，即使对某些SCSI控制程序LILO也能运转良好。

当LILO定位到配置文件后，经过预引导过程，就显示提示符： LILO boot:

此时，系统允许选择引导不同的操作系统或者不同的内核配置，按Tab键显示可选项列表，然后输入可选项或者直接回车选择缺省配置，如果选择了引导Linux，还可以直接传递参数到系统内核。

和其他系统的引导加载程序相比，LILO具有更大的灵活性，其引导方式也更丰富多彩。

当LILO被安装在硬盘的MBR、活动分区或引导软盘上时，作为原引导程序的替身，它能引导任一硬盘任一分区上的Linux和其他操作系统；除了引导扇区，它没有任何隐含文件，也不需要使用特定的分区，它的配置文件可以在任何分区、甚至是存放在与Linux毫不相干的DOS分区的某个子目录下；它能引导几个不同的内核配置，甚至是几个不同的内核；它能引导同一机程序上的多个Linux版本；可达16个。

它能从网络上引导Linux。

LILO的灵活性使得其配置变得相当复杂，当有多个系统共存时，建议先安装其他操作系统，最后再装Linux，这样，设置LILO对其他系统的引导会相对简单一些。

第二部分 LILO参数

通常我们谈到LILO，会涉及到两个方面——LILO引导程序和LILO安装命令/sbin/lilo。

为了不至于混淆这两个概念，本文将用LILO表示LILO引导程序，而lilo表示/sbin/lilo。一般地，LILO使用一个文本文件/etc/lilo.conf作为其配置文件。lilo读取lilo.conf，按照其中的参数将特定的LILO写入系统引导区。任何时候，修改了/etc/lilo.conf，都必须重新运行lilo命令，以保证LILO正常运lilo.conf使用的配置参数很多，配置起来也相当复杂。下面以RedHat Linux为例作一些初步探讨，RedHat的lilo程序包版本为0.20，别的Linux发行版本可能会有所出入，但不会太大。

lilo.conf文件中的配置参数分为两部分，一部分是全局参数，另一部分是引导映像参数。引导映像参数作用于每一个引导映像区。如果某一引导映像参数（例如：password与全局参数的定义相抵触，则以该引导映像参数的定义为准，但仅限于该引导映像区。LILO的引导参数有很多，在此只对一些比较重要的参数作一介绍。与Linux系统其他的配置文件一样，“#”号后的一行文字表示注释。

1. “boot= ” 此参数指明包含引导扇区的设备名（如：/dev/had），若此项忽略，则从当前的根分区中读取引导扇区。
2. “root= ” 此参数告诉内核启动时以哪个设备作为根文件系统使用，其设定值为构造内核时根文件系统的设备名，可用的设备名有：
 - (1)/dev/hdaN~/dev/hddN：ST-506兼容硬盘，a到d上的N个分区
 - (2)/dev/sdaN~/dev/sdeN：SCSI兼容硬盘，a到e上的N个分区
 - (3)/dev/xdaN~/dev/xdnN：XT兼容硬盘，a到b上的N个分区
 - (4)/dev/fdN：软盘，A：（N=0）或B：（N=1）
 - (5)/dev/nfs：由网络取得根文件系统的标志
3. “nfsroot= ” 若需通过NFS提供根文件系统来引导无盘工作站，此参数为内核指定了网络根文件系统所在的机程序、目录及NFS，其格式为：


```
nfsroot= ( server_ip : ) root_dir ( ,nfs_options )
```
4. “nfsaddr= ” 设定网络通讯所需的各种网络界面地址，如无此参数，则内核会试图用反向地址解析协定(RARP)或启动协定(BOOTP)找出这些参数，其格式为：nfsaddr= 客户端IP ： 服务端IP ： 网关IP ： 子网屏蔽 ： 客户端名称 ： 网络设备名 ： auto
5. “image= ” 指定Linux的内核文件。
6. “delay= ” 设定引导第一个映像前的等待时间。
7. “disk= ” 此参数为某一特殊的硬盘定义非标准参数。

8. “ append= ” 为内核传递一个可选的参数行，其典型的应用是为不能完全由系统自动识别的硬盘指定参数，如：append = "hd=64,32,202"
9. “ label= ” 此参数为每个映像指定一个名字，以供引导时选择。
10. “ read-only ” 设定以只读方式挂入根文件系统，用于文件系统一致性检查（fsck）。
11. “ install= ” 安装一个指定文件作为新的引导扇区，缺省为/boot/boot.b。
12. “ loader= ” 说明所使用的链加载程序(chain loader)，缺省为/boot/chain.b，如果不是从首硬盘或软盘启动，那么，此选项必须说明。
13. “ table= ” 说明包含分区表的设备名，如果此参数忽略，引导加载程序将不能传递分区信息到已引导的操作系统。当此参数指向的分区表被修改时，必须重新运行/sbin/lilo。
14. “ init= ” 内核初始化时执行的程序，通常过程为init、getty、rc和sh，版本1.3.43以来的Linux内核能够执行/sbin/init说明的命令行，若在引导过程中出现问题，则可设置init=/bin/sh直接跳到Shell。
15. “ ramdisk_start= ” 由于内核不能放在压缩的内存文件系统映像内，为使内核映像能够和压缩的内存映像放在一张软盘内，加入“ ramdisk_start=offset ”，这样内核才能开始执行。
16. “ mem= ” 此参数的目的之一是为Linux指定使用的内存数量：如mem=96MB，目的之二是指定mem=nopentium告诉内核不要使用 4 MB分页表。
17. “ vga= ” 设置显示模式，如80 × 50、132 × 44等。
18. “ linear ” 产生用于替换硬盘sector/head/cylinder地址（硬盘几何参数）的linear扇区地址。linear地址在运行时产生并且不依赖于硬盘几何参数。某些SCSI硬盘和一些以LBA方式使用的IDE硬盘可能会需要使用这个参数。注意：在将LILO安装到软盘上时不能使用“ linear ” 参数。
19. “ prompt ” 给出“ boot: ” 提示，强制LILO等待用户的键盘输入，按下回车键则立即引导默认的操作系统，而按下Tab键则打印可供选择的操作系统。当“ prompt ” 被设置而“ timeout ” 没有被设置时，系统会一直处于等待状态而不引导任何操作系统。不设置该参数时，LILO不给出“ boot: ” 提示而直接引导默认操作系统，除非用户按下了Shift、Ctrl、Alt三键中的任何一个。大多数情况下，如果你的硬盘上有多个操作系统，建议使用参数，它留给用户一个选择的余地。
20. “ timeout= ” 设置等待键盘输入的时长，单位是0.1秒。超过这段时间没有输入则为超时，系统将自动引导缺省的操作系统。如果不设置本参数，缺省的超时时间长度为无穷大。
21. “ other= ” 设置包含非Linux操作系统，如DOS、SCO UNIX、Windows 95等系统引导映像的文件或设备。
22. alias=name 给当前操作系统起一别名。

第三部分 LILO典型配置方法

通常情况下，Linux的安装程序自身就可以完成LILO的安装配置，从而较好地解决多重系统的引导问题，如果系统不能自动完成这种配置，则可以通过手工修改配置文件/etc/lilo.conf来实现不同条件下的引导。

1.当系统能自动完成配置时

对于这种情况只有一个建议：将LILO安装到Linux分区的根上，而不是MBR这个多事地带。假设当前hda1中装有DOS/Windows，hda2中安装了Linux，则/etc/lilo.conf的内容大致如下：

```

boot=/dev/hda2 # 指定引导位置
compact
delay=50 # 延时 5 秒
root=current # 根在当前分区
image=/boot/vmlinuz # 指定linux的内核文件
label=linux # 用linux为代表名称
other=/dev/hda1 # 其他操作系统所在的分区
table=/dev/hda # 指定包含分区表的硬盘
label=dos # 用dos为代表名称

```

2. 当系统无法自动完成配置时

系统无法自动完成配置的情况不外乎两种：

- （1）BIOS不能直接看到Linux的根分区；
- （2）BIOS只能读写标准IDE硬盘的前504MB。

这时，必须遵循一个最基本的原则：建立一个BIOS能存取的较小的Linux分区，其中包含内核文件、映射文件及链加载程序等必要内容，而根则可以是另外一个独立的分区。至于配置上的其他细节，我们通过以下实例来进行说明。

第四部分 lilo.conf配置实例

有了这些基础知识，我们可以很容易地按照自己的意图配置LILO。

例一．lilo.conf文件

```

boot=/dev/hda #将LILO安装在MBR。LILO作为主引导管理器
message=/boot/message #注释为/boot/message
compact #产生一个更小的“map”文件
map=/boot/map #指定“map”文件为/boot/map
install=/boot/boot.b
password=zhoudi #设置口令
vga=normal #80x25文本模式
linear #使用“linear”地址
prompt #提示用户键盘输入
timeout=50 #超时时长为5秒
default=dos #缺省引导label为dos的操作系统
image=/boot/vmlinuz-2.0.34-1#设定Linux所用核心

```

#设置Linux核心引导映像

label=linux #标识为linux

root=/dev/hda1 #设置根文件系统

read-only #LILO以只读方式载入根文件系统

#设定MS-DOS或Windows 95

other=/dev/hda2 #DOS分区为第一个IDE硬盘的第二分区

```
label=dos #标识为dos
table=/dev/hda #主设备为第一个IDE硬盘
```

#设定SCO UNIX 注意：SCO分区必须设为活动（active）分区并将LILO安装在MBR上。

```
other=/dev/hda3
label=sco
table=/dev/had
```

这个例子中，LILO是作为主引导管理器来管理机器上所有操作系统的。LILO也可作为二级引导管理器，这只要将“boot”参数改为根分区就可做到。例如：`boot=/dev/hda1` 以这种方式使用LILO时，Linux根分区必须用DOS或Linux的fdisk程序将其设置为活动分区，并且这种方式只对硬盘主分区（不是扩展或逻辑分区）有效。

例二．一个标准的IDE大硬盘需安装Linux和DOS/Windows。

对于大硬盘问题，很多人只知道低于1024个柱面的限制，而不知为什么标准的IDE硬盘只能认前504MB。其实，BIOS的int13调用是采用三个位元组的CHS编码，10位为柱面号，8位为磁头号，6位为扇区号。可能的柱面号码是0～1023，可能的磁头号码是0～255，而磁道上可能的扇区号码是1～63，以这24位最多可以定址8455716864个位元组(7.875GB)。但不幸的是，标准的IDE介面容许256个扇区 / 磁道、65536个柱面及16个磁头。它本身可以存取 $2^3 \times 7 = 137438953472$ (128 GB)，但是加上BIOS方面63个扇区与1024个柱面的限制后只剩528482304(504MB)可以定址得到。

对策：在硬盘的前500MB中划分350MB(/dev/hda1)给DOS，150MB(/dev/hda2)给Linux，在相应的配置文件中应说明硬盘的参数。

```
boot=/dev/hda
... ..
disk=/dev/hda
bios=0x80
sectors=63
heads=16
cylinders=2100
image=/vmlunuz
append="hd=2100,16,23"
root=/dev/hda2
label=linux
```

例三．如果你有一块超过8 G的大硬盘,并且需要把Linux安装在比较靠后的位置,可以在安装的时候,选择linear模式,并且给它加上硬盘参数.

安装时候的硬盘参数可以这样写: `hd?=CYLs, HEADs, SECs` 其中的大写字母需要用实际的硬盘参数来替换, 这些参数可以从硬盘的标签上查到, 也可以看看BIOS设置里硬盘参数对应LBA模式的那一行. 问号是根据硬盘确定的, 实际使用时, 它可以是a, b, c, d四个字母中的一个. 比如:
`hda=1869,63,255` 这是IBM 15.2G硬盘的参数.

当然进入了Linux以后, 可以通过编辑/etc/lilo.conf加上这个文件, 然后运行一遍lilo达到同样的目的. 下面是本人未加参数前的lilo.conf的内容:

```
boot = /dev/hda
map = /boot/map
install = /boot/boot.b
prompt
timeout = 50
```



```

image = /boot/vmlinuz
label = linux
root = /dev/hda1
initrd = /boot/initrd-2.2.12-20.img
read-only

```

按照 linear 方式加入参数以后是如下格式:

```

boot = /dev/hda
map = /boot/map
install = /boot/boot.b
prompt
linear <-----加进了这一行
timeout = 50
image = /boot/vmlinuz
label = linux
root = /dev/hda1
initrd = /boot/initrd-2.2.12-20.img
read-only

```

append = "hda=1869,63,255" <-----加进了这一行 注意, append参数是针对每个系统引导记录的, 一定要放在 image 的下面或者是 other 的下面, 这样它才可以发挥作用.

当再次起动 Linux系统的时候, LILO 就按照线性模式对系统进行引导.除了在硬盘上寻址定位的方式不同以外, 对其他方面没有什么影响.

四 配置多重启动 (LILO有1024柱面的限制即如果把内核放到8.4个G之后则只能用软盘启动)

LILO 可以放在MBR (即主引导扇区) 也可以放在BOOT SECTOR 每一分区的第一个扇区

3 LILO的配置文件 /etc/lilo.conf

Prompt 设置成对话模式 (如不这样会自动起用第一组设置)

Timeout=50 等待用户输入的时间 (单位0.1秒)

Image=/boot/vmlinuz 核心

Label=linux 名称

Root=/dev/hda6 设置根目录在那一个分区

Read-only 方式为只读

(以上为第一个操作系统的设置)

Other=/dev/hda1 另一个操作系统所在分区

Label=dos 名称

Table=/dev/had 设置hda1的分区表在那里

设置完之后保存退然后在提示符下执行一下LILO即可

第十四节：包过滤防火墙

一 本节主要是关于怎样用 ipchains 做一个简单的包过滤防火墙。

1 Ipchains主要是通过通过在input,forward,output三个环节上添加规则来过滤。

2原则 假设一个防火墙已经做好，一个请求连接过来之后它会按照规则审查，如果有符合条件的则按照规则走，如果没有按照缺省策略走。

3常用的参数 -L 列出所有的规则 -A 追加一条规则 -D 删除一条规则

-s 源（即从那里发出的请求） -d 目标（向请求提供服务的主机）

-p 使用的协议类型（tcp,udp,icmp） -P 改变缺省策略

4 示例：1）本机的IP地址为192.168.1.10 如果我不想192.168.1.的的所有的主机ping本机则可以

```
ipchains -A input -s 192.168.1.0/24 -d 192.168.1.10 -p icmp -j DENY
```

/24(代表这个网段) 目标为1.10 协议为icmp 拒绝

2）凡是192.168.1.0网段登陆本机23端口的一律拒绝

```
ipchains -A input -s 192.168.1.0/24 -d 192.168.1.10 23 -p tcp -j DENY
```

源 目标 端口 协议

5练习题 1）凡来自192.168.1.112 的ping 则拒绝。

2）凡去往192.168.1.112的http则拒绝，

3）锁定本地的ftp服务

4）改变改变缺省策略为拒绝。

1 参考答案：

1）ipchains -A input -s 192.168.1.112 -d 192.168.1.10 -p icmp -j DENY

2）ipchains -A input -d 192.168.1.112 80 -p tcp -j DENY

3）ipchains -A input -d 192.168.1.10 21 -p tcp -j DENY

4）ipchains -P input DENY

二 IP伪装

1 起用IP转发 1）echo 1 > /proc/sys/net/ipv4/ip_forward

2) ipchains -A forward -s 192.168.0.0/24 -j MASQ

3)ipchains -P forward DENY

2 如果192.168.0.网段的主机将本机作为网关则可以通过本机上网

如果系统是linux只需要修改路由表（即更改缺省网关）

```
route del default gw
```

```
route add default gw 192.168.1.10
```



作者：长江后浪



头衔：初涉菜地

发布时间：2004-12-09 15:21:45

编辑

引用

Aps大侠出马,一个顶.....

一个顶一个就不错了。

第1楼：

不错，我加精了...



作者：梦飞



头衔：版主

发布时间：2004-12-09 16:42:48

编辑

引用

众多精华,皆在《中国IT考试网》- <http://bbs.cnitexam.com>



第2楼：

经典！好东西！！



赛迪网友

作者：zdfbj



头衔：初涉菜地

发布时间：2005-01-05 09:36:35

编辑

引用



第3楼：jo

有利用价值~



赛迪网友

作者：安氏一族



头衔：初涉菜地

发布时间：2005-03-19 08:34:20

编辑

引用

第4楼：

好。收藏



作者：andy



头衔：初涉菜地

发布时间：2005-03-22 18:17:42

编辑

引用

时间没有等我。是你，忘了带我走。

第5楼：

谢谢楼主！



作者：时磊



头衔：初涉菜地

发布时间：2005-03-23 12:33:07

编辑

引用

第6楼：

目录

第一部分：系统

第一节课：Linux的发展、Linux的安装

第二节课：常用命令、管道与重定向

第三节课：vi的使用、用户配置

第四节课：文件系统

第五节课：系统管理（一）

第六节课：系统管理（二）

第二部分：网络服务

第七节课：DHCP服务器的配置

第八节课：DNS服务器的配置

第九节课：smb服务器的配置

第十节课：apache的配置

第十一节课：sendmail的配置

第十二节课：x-window的配置

第三部分：优化与安全

第十三节课：内核的设置与编译

第十四节课：包过滤防火墙与课程总结

第一节：Linux的发展、Linux的安装

一、Linux的定义

1、什么是Linux?

Linux 发音为 "Lin-noks".

Linux 的官方定义：“Linux 是一种 UNIX 操作系统的克隆，它（的内核）由 Linus Torvalds 以及网络上组织松散的黑客队伍一起从零开始编写而成。Linux 的目标是保持和 POSIX 的兼容。”

2、Linux的特点：

A、“Linux 具备现代一切功能完整的 UNIX 系统所具备的全部特征，其中包括真正的多任务、虚拟内存、共享库、需求装载、共享的写时复制程序执行、优秀的内存管理以及 TCP/IP 网络支持等。

B、“Linux 的发行遵守 GNU 的通用公共许可证(GPL)。

C、在原代码级上兼容绝大部分的Unix标准(如IEEE POSIX，System V，BSD)，它遵从 POSIX规范，例如对于System V来说,把其上程序源代码拿到Linux下重新编译后就可以运行，而对于BSD Unix来说，它的可执行文件可以直接在Linux环境下运行。

更进一步地了解Linux, 请浏览下列站点:

<http://www.linux.org>

<http://www.linuxresources.com/what.html>

二、Linux的发展

1、UNIX的发展：

1969年由Ken Thompson在AT&T贝尔实验室的一台DEC PDP-7计算机上实现。后来Ken Thompson和Dennis Ritchie使用C语言对整个系统进行了再加工和编写，使得Unix能够很容易的移植到其他硬件的计算机上。由于此时AT&T还没有把Unix作为它的正式商品，因此研究人员只是在实验室内部使用并完善它。正是由于Unix是被作为研究项目，其他科研机构 and 大学的计算机研究人员也希望能得到这个系统，以便进行自己的研究。AT&T以分发许可证的方法，对Unix仅仅收取很少的费用，大学和研究机构就能获得Unix的源代码以进行研究。Unix的源代码被散发到各个大学，一方面使得科研人员能够根据需要改进系统，或者将其移植到其他的硬件环境中去，另一方面培养了懂得Unix使用和编程的大量的学生，这使得Unix的普及更为广泛。

到了70年代末，在Unix发展到了版本6之后，由于AT&T认识到了Unix的价值，成立了Unix系统实验室（Unix System Lab,USL）来继续发展Unix。因此AT&T一方面继续发展内部使用的Unix版本7，一方面由USL开发对外正式发行的Unix版本，同时AT&T也宣布对Unix产品拥有所有权。几乎在同时，加州大学伯克利分校计算机系统研究小组（CSRG）使用Unix对操作系统进行研究，他们对Unix的改进相当多，增加了很多当时非常先进的特性，包括更好的内存管理，快速且健壮的文件系统等，大部分原有的源代码都被重新写过，很多其他Unix使用者，包括其他大学和商业机构，都希望能得到CSRG改进的Unix系统。因此CSRG中的研究人员把他们的Unix组成一个完整的Unix系统——BSD Unix（Berkeley Software Distribution）向外发行。

而AT&T的Unix系统实验室，同时也在不断改进他们的商用Unix版本，直到他们吸收了BSD Unix中已有的各种先进特性，并结合其本身的特点，推出了Unix System V版本从此以后，BSD Unix和Unix System V形成了当今Unix的两大主流，现代的Unix版本大部分都是这两个版本的衍生产品。如IBM的AIX4.0、HP/UX11、SCO的UnixWare等属于System V，而Minix、freeBSD、NetBSD、OpenBSD等属于BSD Unix。

2、Linux的发展历史

Linux的历史可以追溯到1990年，芬兰赫尔辛基大学的一名学生Linus Torvalds，最初用汇编语言写了一个在80386保护模式下处理多任务切换的程序，后来从Minix(用于操作系统教学、很小的Unix)中得到灵感，发誓要写一个比Minix更好的Minix，于是开始写了一些硬件的设备驱动程序、一个小的文件系统……，这样0.0.1版本的Linux就出来了，但是它必须在有Minix的机器上编译以后才能玩，这时候的Linus已经完全“走火入魔”了，决定踢开Minix“闹革命”，于是在1991年10月5号发布了Linux 0.0.2版本，这个版本已经可以运行bash(一种用户与操作系统内核通讯的软件)和gcc(GNU C编译器)了。

Linus从一开始，就决定自由扩散Linux、包括源代码，他把源代码发布在网上，随即就引起爱好者的注意，他们通过互连网也加入了Linux的

内核开发工作，一大批高水平程序员的加入，使得Linux达到迅猛发展，到1993年底，Linux 1.0终于诞生。Linux 1.0已经是一个功能完备的操作系统了，其内核写得紧凑高效，可以充分发挥硬件的性能，在4M内存的80386机器上也表现得非常好。

很多人对Linux的认识有个误区，即总把Linux与低档硬件平台联系到一起，其实从2.1.xx系列内核开始，Linux就开始走高端的路子了，大约在1.3版本之后，开始向其他硬件平台上移植，包括号称最快的CPU---Digital Alpha(目前主频是最高的)，目前Linux能将硬件的性能充分发挥出来，可以囊括低端到高端的所有应用。

Linux加入GNU并遵循公共版权许可证(GPL)，由于不排斥商家对自由软件进一步开发，不排斥在Linux上开发商业软件，故而使Linux又开始了一次飞跃，出现了很多的Linux发行版，如Slackware、Redhat、TurboLinux、OpenLinux等十多种，而且还在增加，还有一些公司在Linux上开发商业软件或把其他Unix平台的软件移植到Linux上来，如今很多IT界的大腕如IBM、Intel、Oracle、Informix、Sysbase、Netscape、Novell等都宣布支持Linux! 商家的加盟弥补了纯自由软件的不足和发展障碍，Linux得以迅速普及。

3、Linux 的主要发行版

目前比较流行的版本有：Slackware、OpenLinux、Mandrake、Red Hat Linux等等。建

简体中文 Linux 发行版：BluePoint Linux：最新发布的 Linux 中文版。利用 Linux 2.2 内核的 FrameBuffer，可在控制台获得中文输入输出。具备多内码支持，目前可以支持大陆国标码和港台大五码。与 RedHat Linux 兼容。

TurboLinux：国内最早的简体中文发行版之一。

XteamLinux：带有较好的图形配置工具googe。

4、几个常用术语

A、POSIX 及其重要地位

POSIX 表示可移植操作系统接口（Portable Operating System Interface，缩写为 POSIX 是为了读音更像 UNIX）。由电气和电子工程师协会（Institute of Electrical and Electronics Engineers，IEEE）开发，是为了提高 UNIX 环境下应用程序的可移植性。然而，POSIX 并不局限于 UNIX。许多其它的操作系统，例如 DEC OpenVMS 和 Microsoft Windows NT，都支持 POSIX 标准，尤其是 IEEE Std. 1003.1-1990（1995 年修订）或 POSIX.1，POSIX.1 提供了源代码级别的 C 语言应用编程接口（API）给操作系统的服务程序，例如读写文件。POSIX.1 已经被国际标准化组织（International Standards Organization，ISO）所接受，被命名为 ISO/IEC 9945-1:1990 标准。

POSIX 现在已经发展成为一个非常庞大的标准族，某些部分正处在开发过程中。

B、GNU 和 Linux 的关系

GNU 是 GNU Is Not UNIX 的递归缩写，是自由软件基金会的一个项目，该项目的目标是开发一个自由的 UNIX 版本，这一 UNIX 版本称为 HURD。尽管 HURD 尚未完成，但 GNU 项目已经开发了许多高质量的编程工具，包括 emacs 编辑器、著名的 GNU C 和 C++ 编译器（gcc 和 g++），这些编译器可以在任何计算机系统上运行。所有的 GNU 软件和派生工作均适用 GNU 通用公共许可证，即 GPL。GPL 允许软件作者拥有软件版权，但授予其他任何人以合法复制、发行和修改软件的权利。

Linux 的开发使用了许多 GNU 工具。Linux 系统上用于实现 POSIX.2 标准的工具几乎都是 GNU 项目开发的，Linux 内核、GNU 工具以及其他一些自由软件组成了人们常说的 Linux：C 语言编译器和其他开发工具及函数库。

X Window 窗口系统。

各种应用软件，包括字处理软件、图象处理软件等。

其他各种 Internet 软件，包括 FTP 服务器、WWW 服务器等。

关系数据库管理系统等。

C、GPL (General Public License)公共许可协议

GPL 的文本保存在 Linux 系统的不同目录下的命名为 COPYING 的文件里。例如，键入 `cd /usr/doc/ghostscript*` 然后再键入 `more COPYING` 可查看 GPL 的内容。

GPL 和软件是否免费无关，它主要目标是保证软件对所有的用户来说是自由的。GPL 通过如下途径实现这一目标：

- 它要求软件以源代码的形式发布，并规定任何用户能够以源代码的形式将软件复制或发布给别的用户。

- 它提醒每个用户，对于该软件不提供任何形式的担保。
- 如果用户的软件使用了受 GPL 保护的软件的一部分，那么该软件就继承了 GPL 软件，并因此而成为 GPL 软件，也就是说必须随应用程序一起发布源代码。
- GPL 并不排斥对自由软件进行商业性质的包装和发行，也不限制在自由软件的基础上打包发行其他非自由软件。

遵照 GPL 的软件并不是可以任意传播的，这些软件通常都有正式的版权，GPL 在发布软件或者复制软件时声明限制条件。但是，从用户的角度考虑，这些根本不能算是限制条件，相反用户只会从中受益，因为用户可以确保获得源代码。

尽管 Linux 内核也属于 GPL 范畴，但 GPL 并不适用于通过系统调用而使用内核服务的应用程序，通常把这种应用程序看作是内核的正常使用。假如准备以二进制的形式发布应用程序（像大多数商业软件那样），则必须确保自己的程序未使用 GPL 保护的软件。如果软件通过库函数调用而使用了别的软件，则不必受到这一限制。大多数函数库，受另一种 GNU 公共许可证，即 LGPL 的保护，将在下面介绍。

D、LGPL (Library General Public License)

GNU LGPL（GNU 程序库公共许可证）的内容全部包括在命名为 COPYING.LIB 的文件中。如果安装了内核的源程序，在任意一个源程序的目录下都可以找到 COPYING.LIB 文件的一个拷贝。

LGPL 允许在自己的应用程序中使用程序库，即使不公开自己的源代码。但是，LGPL 还规定，用户必须能够获得在应用程序中使用的程序库的源代码，并且允许用户对这些程序库进行修改。

大多数 Linux 程序库，包括 C 程序库（libc.a）都属于 LGPL 范畴。因此，如果在 Linux 环境下，使使用 GCC 编译器建立自己的应用程序，程序所链接的多数程序库是受 LGPL 保护的。如果想以二进制的形式发布自己的应用程序，则必须注意遵循 LGPL 有关规定。

遵循 LGPL 的一种方法是，随应用程序一起发布目标代码，以及可以将这些目标程序和受 LGPL 保护的、更新的 Linux 程序库链接起来的 makefile 文件。

遵循 LGPL 的比较好的一种方法是使用动态链接。使用动态链接时，即使是程序在运行中调用函数库中的函数时，应用程序本身和函数库也是不同的实体。通过动态链接，用户可以直接使用更新后的函数库，而不用对应用程序进行重新链接。

但我们必须当心个别属于 GPL 范畴的库和实用程序：GNU dbm（即 gdbm）数据库类的程序库是非常著名的 GPL 库；GNU bison 分析器生成程序是另一个实用的 GPL 工具，如果使用 bison 生成代码，所得的代码也适用于 GPL。

在 GPL 的保护范围以外，也有 GNU dbm 和 GNU bison 的相应的替代程序。例如，对于数据库类的程序库，可以使用 Berkeley 数据库 db 来替代 gdbm；对于分析器生成器，可以使用 yacc 来替代 bison。

5、Linux 发展的重要里程碑

1990, Linus Torvalds 首次接触 MINIX

1991 中, Linus Torvalds 开始在 MINIX 上编写各种驱动程序等操作系统内核组件

1991 底, Linus Torvalds 公开了 Linux 内核

1993, Linux 1.0 版发行，Linux 转向 GPL 版权协议

1994, Linux 的第一个商业发行版 Slackware 问世???

1996, 美国国家标准技术局的计算机系统实验室确认 Linux 版本 1.2.13（由 Open Linux 公司打包）符合 POSIX 标准

1999, Linux 的简体中文发行版相继问世

2000 中, LinuxWorld China 2000 展览会召开，涌现大量基于 Linux 的嵌入式系统。

三、linux的安装（RedHat6.2的安装）

1、安装前的准备（了解你的硬件）

CPU C733

内存 256m

硬盘 希捷20G

显卡 TNT2 M64/16M

主板 技嘉694X

网卡 RTL8029

- RedHat6.2的安装光盘（带启动）
- 一张空白软盘

二、开始安装：

（1）、进入CMOS设置重光盘引导：

（2）、选择安装方式

缺省状态为图形安装方式（直接按ENTER键），我们采用文本安装方式：

输入text，然后按ENTER键。

（3）、选择语言：(US English)

这一步对大家都不陌生了，首先要选择自己需要的语言。或者选择和自己国家相近的语言也可以。建议大家使用US English，学习Red Hat Linux 还是从英文版本开始学习比较正规，因为Red Hat Linux 里面很多程序都是英文的。缺省是没有中文支持的。

（4）、选择键盘布局：（选择us）

（5）、欢迎画面：

（6）、选择安装类型：

建议选择定制安装（Install Custom System），最后一项为升级安装。其它三项

Install GNOME Workstation

Install KDE Workstation

Install Server System

都不能对硬盘进行分区，安装程序自动创建分区。

（7）、选择分区方式：（FDISK）

（8）、硬盘进行分区：

A、先按m键查看帮助，按p查看分区信息：

B、按n键创建分区，并选择p创建主分区：

输入分区编号：1（表示是第一个分区），回车后输入起始柱面数，直接回车即可。输入分区容量2G（+2000M）或结束柱面数。

同理创建第二个分区，容量为256M，作为交换分区。并用t命令改变分区类型为82（swap）：

先输入l命令查看列表，找到linux swap的ID号为82，再输入82回车即可。

C、a命令设置引导标志位

再用p命令查看时，在boot列已有*号

D、用w命令保存并退出。

补充说明：

(1)、linux内物理硬盘的表示方法：

IDE硬盘 SCSI硬盘

Primary master hda SCSI：0 sda

Primary slave hdb SCSI：1 sdb

Secondary master hdc SCSI：2 sdc

Secondary slave hdd SCSI：3 sdd

SCSI：n...

2）、linux内分区的表示方法：

hda1——hda4表示主分区或扩展分区，hda5——hda...表示扩展分区。

一个硬盘最多可以分4个主分区或3个主分区加一个扩展分区，扩展分区可以再分若干个逻辑分区。

(3)、linux对分区的要求：

安装linux时至少给硬盘分两个区，一个linux native（linux分区）、一个linux swap（交换分区），交换分区一般为内存的1.5——2倍，

(7) 挂结分区：

(8) 格式化硬盘分区

(9) lilo的设置（可以不用理会）

(10) lilo的安装位置（推荐为MBR）

(11) linux安装到那个分区（hda1）

(12) 输入主机名

(13) 设置IP地址、网关、DNS

(14) 选择鼠标驱动(如是两键鼠标，打开模拟三键 (Emulate 3 Buttons))

(15) 设置时区(选择China/Beijing)

(16) 设置root口令

(17) 新建用户（可以不建）

(18) 选择加密方式（缺省）

(19) 选择安装组件

(20) 如果安装了X-window，则选择显卡

(21) 开始复制文件

(22) 是否创建引导软盘（推荐创建）

(23) 安装完毕（启出光盘，计算机自动从启）

第二节：常用命令、管道与重定向

一、常用命令：

二、标准输入/输出和重定向

1、标准输入与输出

我们知道，执行一个shell命令行时通常会打开三个标准文件，即标准输入文件（stdin），通常对应终端的键盘；标准输出文件（stdout）和标准错误输出文件（stderr），这两个文件都对应终端的屏幕。进程将从标准输入文件中得到输入数据，将正常输出数据输出到标准输出文件，而将错误信息送到标准错误文件中。我们以cat命令为例，cat命令的功能是从命令行给出的文件中读取数据，并将这些数据直接送到标准输出。若使用如下命令：

```
$ cat config
```

将会把文件config的内容依次显示到屏幕上。但是，如果cat的命令行中没有参数，它就会从标准输入中读取数据，并将其送到标准输出。例如：

```
$ cat
Hello world
Hello world
Bye
Bye

$
```

用户输入的每一行都立刻被cat命令输出到屏幕上。

另一个例子，命令sort按行读入文件正文（当命令行中没有给出文件名时，表示从标准输入读入），将其排序，并将结果送到标准输出。下面的例子是从标准输入读入一个采购单，并将其排序。

```
$ sort
```

```
bananas
carrots
apples
```

```
apples
bananas
carrots
```

```
$
```

这时我们在屏幕上得到了已排序的采购单。

直接使用标准输入/输出文件存在以下问题：

输入数据从终端输入时，用户费了半天劲输入的数据只能用一次。下次再想用这些数据时就得重新输入。而且在终端上输入时，若输入有误修改起来不是很方便。

输出到终端屏幕上的信息只能看不能动。我们无法对此输出作更多处理，如将输出作为另一命令的输入进行进一步的处理等。

为了解决上述问题，Linux系统为输入、输出的传送引入了另外两种机制，即输入/输出重定向和管道。

输入重定向：

输入重定向是指把命令（或可执行程序）的标准输入重定向到指定的文件中。也就是说，输入可以不来自键盘，而来自一个指定的文件。所以说，输入重定向主要用于改变一个命令的输入源，特别是改变那些需要大量输入的输入源。

例如，命令wc统计指定文件包含的行数、单词数和字符数。如果仅在命令行上键入：

```
$ wc
```

wc将等待用户告诉它统计什么，这时shell就好象死了一样，从键盘键入的所有文本都出现在屏幕上，但并没有什么结果，直至按下 <ctrl+d>，wc才将命令结果写在屏幕上。

如果给出一个文件名作为wc命令的参数，如下例所示，wc将返回该文件所包含的行数、单词数和字符数。

```
$ wc /etc/passwd
```

```
20 23 726 /etc/passwd
```

```
$
```

另一种把/etc/passwd文件内容传给wc命令的方法是重定向wc的输入。输入重定向的一般形式为：命令<文件名。可以用下面的命令把wc命令的输入重定向为/etc/passwd文件：

```
$ wc < /etc/passwd
```

```
20 23 726
```

```
$
```

另一种输入重定向称为here文档（从这里开始，到这里结束），它告诉shell当前命令的标准输入来自命令行。here文档的重定向操作符使用<<。它将一对分隔符（本例中用delim表示）之间的正文重定向输入给命令。下例将一对分隔符delim之间的正文作为wc命令的输入，统计出正文的行数、单词数和字符数。

```
$ wc<<DELIM
```

```
>this text forms the content
```

```
>of the here document,which
```

```
>continues until the end of
```

```
>text delimiter
```

```
>delim
```

```
4 17 98
```

在<<操作符后面,任何字符都可以作为正文开始前的分隔符，本例中使用delim作为分隔符。here文档的正文一直延续到遇见另一个分隔符为止。第

二个分隔符应出现在新行的开头。这时here文档的正文（不包括开始和结束的分隔符）将重新定向送给命令wc作为它的标准输入。

由于大多数命令都以参数的形式在命令行上指定输入文件的文件名，所以输入重定向并不经常使用。尽管如此，当要使用一个不接受文件名作为输入参数的命令，而需要的输入内容又存在一个文件里时，就能用输入重定向解决问题。

输出重定向：

输出重定向是指把命令（或可执行程序）的标准输出或标准错误输出重新定向到指定文件中。这样，该命令的输出就不显示在屏幕上，而是写入到指定文件中。

输出重定向比输入重定向更常用，很多情况下都可以使用这种功能。例如，如果某个命令的输出很多，在屏幕上不能完全显示，那么将输出重定向到一个文件中，然后再用文本编辑器打开这个文件，就可以查看输出信息；如果想保存一个命令的输出，也可以使用这种方法。还有，输出重定向可以用于把一个命令的输出当作另一个命令的输入（还有一种更简单的方法，就是使用管道，将在下面介绍）。

输出重定向的一般形式为：命令>文件名。例如：

```
$ ls > directory.out
$ cat directory.out
ch1.doc ch2.doc ch3.doc chimp config mail/ test/
$
```

将ls命令的输出保存为一个名为directory.out的文件。

注：如果>符号后边的文件已存在，那么这个文件将被重写。

为避免输出重定向中指定文件只能存放当前命令的输出重定向的内容，shell提供了输出重定向的一种追加手段。输出追加重定向与输出重定向的功能非常相似，区别仅在于输出追加重定向的功能是把命令（或可执行程序）的输出结果追加到指定文件的最后，而该文件原有内容不被破坏。如果要将一条命令的输出结果追加到指定文件的后面，可以使用追加重定向操作符>>。形式为：命令>>文件名。例如：

```
$ ls *.doc>>directory.out
$ cat directory.out
ch1.doc ch2.doc ch3.doc chimp config mail/ test/
ch1.doc ch2.doc ch3.doc
$
```

错误输出重定向：

和程序的标准输出重定向一样，程序的错误输出也可以重新定向。使用符号2>（或追加符号2>>）表示对错误输出设备重定向。例如下面的命令：

```
$ ls /usr/tmp 2> err.file
```

可在屏幕上看到程序的正常输出结果，但又将程序的任何错误信息送到文件err.file中，以备将来检查用。

还可以使用另一个输出重定向操作符（&>）将标准输出和错误输出同时送到同一文件中。例如：

```
$ ls /usr/tmp &> output.file
```

利用重定向将命令组合在一起，可实现系统单个命令不能提供的新功能。例如使用下面的命令序列：

```
$ ls /usr/bin > /tmp/dir
$ wc -w < /tmp/dir
459
```

统计了/usr/bin目录下的文件个数。

2、管道：

将一个程序或命令的输出作为另一个程序或命令的输入，有两种方法，一种是通过一个临时文件将两个命令或程序结合在一起，例如上个例子中的/tmp/dir文件将ls和wc命令联在一起；另一种是Linux所提供的管道功能。这种方法比前一种方法更好。

管道可以把一系列命令连接起来，这意味着第一个命令的输出会作为第二个命令的输入通过管道传给第二个命令，第二个命令的输出又会作为第三个命令的输入，以此类推。显示在屏幕上的是管道行中最后一个命令的输出（如果命令行中未使用输出重定向）。

通过使用管道符“|”来建立一个管道行。用管道重写上面的例子：

```
$ ls /usr/bin|wc -w
```

```
1789
```

再如：

```
$ cat sample.txt|grep "High"|wc -l
```

管道将cat命令（列出一个文件的内容）的输出送给grep命令。grep命令在输入里查找单词High，grep命令的输出则是所有包含单词High的行，这个输出又被送给wc命令，wc命令统计出输入中的行数。假设sample.txt文件的内容如下：

```
Things to do today:
```

```
Low:Go grocery shopping
```

```
High:Return movie
```

```
High:Clear level 3 in Alien vs. Predator
```

```
Medium:Pick up clothes from dry cleaner
```

那么该管道行的结果是2。

命令替换：

命令替换和重定向有些相似，但区别在于命令替换是将一个命令的输出作为另外一个命令的参数。常用命令格式为：

```
command1 `command2`
```

其中，command2的输出将作为command1的参数。需要注意的是这里的`符号，被它括起来的内容将作为命令执行，执行后的结果作为command1的参数。例如：

```
$ cd `pwd`
```

该命令将pwd命令列出的目录作为cd命令的参数，结果仍然是停留在当前目录下。

第三节课：vi的使用、用户配置

一、vi常用命令参考：

命令方式：

功能键 功能

:q 在未做修改的情况下退出

:q! 将任何修改作废，退出

:w 存盘

:wq 存盘并退出

:w file 将编辑的内容写到 file 文件中

:r file 将 file 文件读入，并放在当前光标行的后面

:e file 编辑新文件 file 代替旧内容

:f file 将当前正文的名称改为 file

:f 打印当前正文的名称和状态

:n 移动光标到第 n 行

:/str/ 将光标移动到下一个包含 str 字符串的行

?str? 将光标移动到最近的一个包含 str 字符串的行

:/str/w file 将第一个包含 str 字符串的行写到文件 file 中

:/str1/,/str2/w file 将从包含 str1 到 包含 str2 之间的行写入 file 文件
:s/str1/str2 用 str2 替换掉首次出现的 str1
:s/str1/str2/g 用 str2 替换掉每次出现的 str1
:d 删除当前光标行

插入方式：

功能键 功能

i 在光标左面插入正文

a 在光标右面插入正文

o 在光标所在行下面增加行

O 在光标所在行上面增加行

I 在光标的开头插入

A 在光标的末尾插入

s 用新的正文替换光标所指字符

cw 用新的正文替换光标右边的字

cb 用新的正文替换光标前边的字

cd 用新的正文替换光标所在的行

c\$ 用新正文替换从光标开始到本行末尾的所有字符

c0 用新正文替换从光标开始到本行开头的字符

编辑方式：

功能键 功能

ctrl-f 前翻一页

ctrl-b 后翻一页

H 将光标移动到屏幕最上面一行

M 将光标移动到屏幕中间

L 将光标移动到屏幕最下面一行

w 右移光标到下一个字的开头

e 右移光标到一个字的末尾

b 左移光标到前一个字的开头

0 左移光标到到本行的开始

^ 移动光标到行中第一个非空字符

\$ 右移光标到本行末尾

/string 向前搜索指定的字符串

?string 向后搜索指定的字符串

n 搜索下一个字符串

rc 用 c 替换当前光标处的字符

x 删除当前光标处的字符

dw 删除光标右面的字

db 删除光标左面的字

dd 删除光标所在行

d\$ 删除当前光标到行末尾之间的字符

d0 从当前光标起删除字符直到行的开始
 J 删除本行的回车，并和下一行合并
 p 将缓冲区内的内容粘贴到当前光标的后面
 P 将缓冲区内的内容粘贴到当前光标的后面
 yy 将当前行复制到剪切缓冲区
 nyy 将n行复制到剪切缓冲区
 u 撤消前一命令的结果
 . 重复最后一条修改正文的命令

二、用户配置

第四节课：文件系统

文件系统形成了 Linux系统上所有数据的基础，Linux程序、库、系统文件和用户文件都驻留在文件系统上，因此对文件系统的适当管理是很重要的。

一、文件系统的目录结构

在 Linux下，用户所见到的文件空间是基于树状结构的，树的根在顶部。在这个空间中的各种目录和文件从树根向下分支，顶层目录(/)被称为根目录，

Linux操作系统由一些目录和许多文件组成。根据用户选择的安装不同，这些目录可能是不同的文件系统。通常，大多数操作系统都驻存在两个文件系统上：即称为/的根文件系统和安装在/usr下的文件系统。

如果你用 cd /命令将当前目录改变到根目录，并用ls命令列出目录清单，你就会看到一些目录。这些目录组成了根文件系统的内容，它们也为其他文件系统提供了安装点。

/bin目录包含称为二进制(binary)文件的可执行程序(事实上，名为/bin的目录是 binary的缩写)。这些程序是必需的系统文件，许多 Linux命令(如 ls)放在该目录中。

/sbin目录也用于存储系统二进制文件。这个目录中的大多数文件用于管理系统。

/etc目录非常重要，它包含许多Linux系统配置文件。从本质上说，这些文件使你的Linux系统具有自己的个性。口令文件(口令)就放在这里，在启动时安装的文件系统列表(fstab)也放在这里。另外，这个目录还包括 Linux的启动脚本、你想要永久记录的、带 IP地址的主机列表和许多其他类型的配置信息。

/lib目录中存储着程序运行时使用的共享库被存储在此。通过共享库，许多程序可以重复使用相同的代码，并且这些库可以存储在一个公共的位置上，因此能减小运行程序的大小。

/dev目录包含称为设备文件的特殊文件，这些文件用于访问系统上所有不同类型的硬件。例如，/dev/mouse文件是用于读取鼠标输入的。通过用这种方法组织对硬件设备的访问，Linux有效地使硬件设备的接口看起来就象一个文件。这意味着在许多情况下，你可以用对软件使用的相同语法来对计算机的硬设备进行操作。例如，为了在软盘驱动器上建立你的起始目录的磁盘档案，你可以使用下面的命令：

```
tar-cdf/dev/fd0~ tackett
```

/dev目录中的许多设备都放在逻辑组中，表列出了/dev 目录中一些最常用的设备。

设备文件 描述

/dev/console 系统控制台，它是与Linuxmso-hansi-font-family: "Times New Roman">系统实际连接的计算机显示器

/dev/hd IDE硬盘驱动器的设备驱动程序接口。/dev/hdali设备指在硬盘驱动hda上的第一个分区，设备/dev/hda指整个硬盘 had。

/dev/sd SCSI磁盘的设备驱动程序接口。SCSI磁盘和分区使用与IDE/dev/hd设备相同的约定。

/dev/fd 提供支持软盘的设备驱动程序。/dev/fd0是第一个软盘驱动器，/dev/fd1是第二个软盘驱动器。

/dev/st SCSI磁带驱动器的设备驱动程序

/dev/tty 为用户输入提供不同控制台的设备驱动程序。它的名字来源于实际挂接到 UNIX系统的、被称为电传打字机(teletype)的终端。在Linux下，这些文件提供对虚拟控制台的支持，可以通过按 < Alt-F1 > 到 < Alt-F6 > 键来访问这些虚拟控制台。这些虚拟控制台提供独立的、同时进行的本地登录对话过程

/dev/pty 支持伪终端的设备驱动程序，伪终端用于远程登录进程，如使用Telnet的远程登录

/dev/ttyS 计算机的串行接口。/dev/ttySO对应MS-DOS下的 COM1。如果有一个串行鼠标，/dev/mouse是对连接了鼠标的ttyS设备的符号链接。

/dev/cua 与调制解调器一起使用的特殊呼出设备。

/dev/null 数据将永远丢失。如果你想要运行一个命令并丢弃标准输出或标准错误输出，那么这个设备是非常有用的。而且，如果用/dev/null作为一个输入文件，则会建立一个长度为零的文件。

/dev目录中一些最常用的设备

/proc 目录实际上是一个虚拟文件系统，它被用于从内存中读取处理信息。

/tmp目录用于存储程序运行时生成的临时文件。如果你有一个程序，它会生成许多大的临时文件，那么你可能想把/tmp目录作为一个独立文件系统来安装，而不是只把它作为根文件系统上的一个目录。如果把/tmp作为根文件系统上的一个目录，并且有许多大文件要写入其中，那么根文件系统的空间就会用尽。

/home目录是用户起始目录的基础目录。通常把它作为一个独立的文件系统来安装，这样用户就能有大量用于自己文件的空间。事实上，如果系统上有许多用户，则可能需要把/home分成几个文件系统。为此，需要建立子目录，如用/home/staff和/home/admin目录分别作为公司职员和管理人员的子目录。把每一个这样的子目录做为不同的文件系统来安装，然后在其下建立用户的起始目录。

/var目录保存要随时改变大小的文件。通常，各种系统记录文件都放在这个目录下。

/usr目录及其子目录对 Linux系统的操作是非常重要的。它包含这样一些目录，这些目录中保存系统上的一些最重要的程序。通常，/usr的子目录包含你安装的大型软件包。

子目录 描述

/usr/bin 这个目录保存许多在Linux系统上能找到的可执行程序。

/usr/etc 这目录包含许多各种各样的系统配置文件。

/usr/include 这个目录及其子目录是存放 C编译程序的所有包含文件的地方。这些头文件定义常量和函数，对 C编程是重要的。

/usr/g++-include 这个目录包含 C++编译程序的包含文件。

/usr/lib 这个目录包含程序在链接时使用的各种库。

/usr/man 这个目录包含Linux系统上的程序的各种联机帮助。在/usr/man下是一些与联机帮助的不同段相对应的子目录。

/usr/src 这个目录包含保存系统的不同程序的源代码的目录。如果你得到了一个想要安装的软件包，那么/usr/src/packageName是安装该包前放置源代码的好地方。

/usr/local 这个目录用于存放系统的本地定制内容。通常，许多本地软件安装在这个目录的子目录中。这个目录的格式在几乎你所见到的每一个UNIX系统上都不相同。建立此目录的一种方法是：为二进制文件建立一个/usr/local/bin目录；为配置文件建立一个/usr/local/etc目录；为各种库建立一个/usr/local/lib目录；为源代码建立一个/usr/local/src目录。如果要使/usr/local具有许多空间的话，则可把整个/usr/local目录树作为一个独立的文件系统来安装

/usr文件系统中重要的子目录

应当引起大家注意的是Linux文件系统这个词有两个不同且经常是相互矛盾的含义：一个含义是磁盘和磁盘机制的文件系统，另一个含义是用户看得见并能操作的逻辑文件系统。本章讨论用户看得见并能操作的逻辑文件系统。另外，如果用户熟悉如 MS-DOS和 OS/2这样的 PC操作系统的话，那么就会发现下面许多话题都是熟悉的，这是因为2.0版本以上的 MS-DOS的文件结构都是模仿UNIX的文件结构的，而 UNIX的文件结构也是 Linux使用的文件结构。

Linux中的每个实际的实体和逻辑实体在Linux文件系统中都用文件来表示。实际的实体包括磁盘、打印机和终端，逻辑实体包括目录、存储文档和程序的普通文件。

二、了解文件名和路径名

就象在 MS-DOS等其他操作系统中一样，在Linux系统中必须区分文件名和路径名。一个文件名由字母、数字和某些标点符号组成的简单串组成，文件名中不能包含空格或作为字段分隔符的任何字符。例如，文件名 johns . letter是正确的，而 johns letter则是不正确的。文件名中不应包含任何

对shell来说有特殊含义的字符。这些特殊字符如下所示：

```
!@#$%~&*()[]{}'"\|; <>`
```

另外，文件名中不能包含“/”符号，因为这个符号用于指示路径名。

早期的大多数 UNIX版本把文件名限制在14个字符内，但 Linux允许一个文件名有256个字符。有些最近的 UNIX版本(如Berkeley版本(BSD))允许64个字符的文件名，但是只有前14个字符是有效的。由于 Linux的目标之一是可移植，所以为了书写可移植程序和 shell脚本，你应该把文件名限制在14个字符以内。

路径名可以有任意多个字符。在Linux中，文件不是存在于真空中的，而是存在于目录中。关于目录的知识，我们已在上一节作了比较详细的介绍。Linux中的最高层目录是根目录，用斜杠符号(/)标记，如果一个名为fred的文件存在于根目录下，它的绝对路径名就是/fred。当用 adduser命令向系统添加新用户时，新用户分配到一个起始目录。按常规，该起始目录通常建在根目录下一个名为home的目录中。如果一位名为 Fred的用户被分配了一个名为/home/fred的目录，那么Fred建立的所有文件都置于/home/fred目录中。Fred的某个文件的绝对路径名可能是/home/fred/freds . file。绝对路径名准确地指出了在文件系统的什么地方能找到文件。

另一种路径名是相对路径名，它明确地指出了文件相对于当前目录的位置。如果Fred在他的起始目录中，那么文件名freds . file是一个相对于当前目录的相对路径名。为了找出哪个目录是当前目录，可以使用 pwd 命令(打印工作目录)。也可以用 echo\$PWD命令检查环境变量 \$PWD 的内容，以查看哪个目录是当前的工作目录。

通过使用两个建在所有目录下的别名，你可用相对路径名来定义在 Linux文件系统任何地方的文件。一个点(.)指当前目录；两个点(..)指父目录。MS-DOS和OS/2使用与此相同的约定。

大多数 Linux命令按路径名操作。在大多数情况下，你使用的路径名是当前目录中的某个文件名，默认的路径名指向你的当前目录。如果Fred在他的起始目录(/home/fred)中，下面的三种方式都是等价的：

```
command freds . letter
```

```
command/home/fred/freds . letter
```

```
d . /freds . letter
```

三、Linux的文件类型

Linux只有四种基本的文件类型：普通文件、目录文件、连接文件和特殊文件。

可以用file命令来确定文件的类型，它可以识别文件类型属于可执行文件还是文本文件、数据文件等等。许多 UNIX命令只是 shell脚本，或者是与MS-DOS的批处理文件相似的解释执行的程序，可用 file命令来报告一个 UNIX命令是一个二进制的可执行程序还是一个简单的shell脚本；也可用该命令来确定文件是不是文本文件，即是否能浏览或编辑该文件。这个命令的语法是：

```
file [-vczL] [-f namefile] [-m magicfile] filelist
```

下表对file命令中的参数作了说明。

参数 描述

-c 打印输出magic文件(/usr/lib/magic)的语法分析形式，该参数是个二进制文件第一部分中的一个数值，这个数值标识了该文件的类型。它通常与-m一起使用，用于在安装一个新的magic文件之前对该文件进行调试

-z 查看压缩文件内部并试图找出该文件的类型

-L 引起后续的符号连接

-f namefile 告诉file，要识别的文件列表在文本文件namefile中。当需要识别许多文件时，该参数是很有用的

-m magicfile 指定用于确定文件类型的magic数字的一个替代文件。默认文件是/usr/lib/magic

filelist 列出想知道其类型的、以空格分隔的一组文件

1、普通文件

普通文件是用户大部分时间都在进行操作的文件。普通文件包括文本文件、C语言源代码、shell脚本(由 Linux shell解释的程序)、二进制的可执行程序和各种类型的数据。对比Linux来说，文件就是文件，Linux要知道的文件之间唯一区别是它是不是可执行文件。可执行文件可以直接执行，当然，这种文件应该含有要执行的内容，并且这种文件要在你的查找路径中。查找路径是你事先指定的路径名列表，Linux查找该路径名列表以找到可执行文件。可执行文件是二进制文件(也就是说，它包含可执行机器代码)和 shell脚本的文件。前面讨论过的 Linux的 file命令查看文件中的数

据并对其内容进行合理的推测，如果你键入file*，你可能会看到类似于下面的内容：

```
INSTALL : symbolic link to/var/adm
ghostvw.txt : ascii text
linux : symbolic link to/usr/src/linux
mbox : mail text
mterm . txt : English text
seyon . txt : English text
xcalc . txt : English text
xclock . txt : English text
xeyes . txt : English text
xgrap.txt : English text
xlock . txt : English text
xsprea.txt : English text
xtriss . txt : empty
```

第一列中提到的所有文件都是普通文件，它们包含不同的数据类型，所有文件都置于file命令能被执行的那个目录中。

2、目录文件

目录也是文件，它们包含文件名和子目录名，并包含指向那些文件和子目录的指针。目录文件是Linux存储文件名的唯一地方，当你用ls命令列出一个目录的内容时，你所做的事只是列出了这个目录文件的内容，还没有触及文件本身。

当你用mv命令对存在当前目录中的一个文件重新命名时，你所做的事只是改变了该文件在目录文件中的条目。如果你把一个文件从一个目录移到另一个目录中，你所做的事只是把这个文件的说明从一个目录文件移到另一个目录文件中，当然，这要求新的目录在同一实际的磁盘上或在一分区上。如果不是这样的话，Linux把该程序的每一个字节都实际地拷贝到另一磁盘上。

3、目录与实际磁盘

Linux系统中的每个文件都被赋予一个唯一的数值，这个数值称做索引节点。索引节点存储在一个称作索引节点表 < inode table > 中，该表在磁盘格式化时被分配。每个实际的磁盘或分区都有其自己的索引节点表。一个索引节点包含文件的所有信息，包括磁盘上数据的地址和文件类型。文件类型包括如普通文件、目录和特殊文件这样的信息。

Linux文件系统把索引节点号 1 赋予根目录，这给予 Linux 的根目录文件在磁盘上的地址。根目录文件包括文件名、目录名及它们各自的索引节点号的列表，Linux可以通过查找从根目录开始的一个目录链来找到系统中的任何文件。根目录文件的内容可能像下面这样：

```
1
1 . .
45 etc
230 dev
420 home
123 . profile
注意文件 . (点)和 . . (双点)在目录中的含义。因为这是根目录，所以 “ . ” 和它的父目录 “ .. ” 是相同的，而在/hom目录文件中它们的内容则不同：
420 .
1 . .
643 fred
```

注意，当前目录(.)的索引节点与建在根目录文件中的/home的索引节点相同，父目录(..)的索引节点与根目录的索引节点相同。

Linux通过上下连接目录文件系统来驾驭其文件系统。如果你想把文件移到另一实际磁盘的目录中，则Linux通过读取索引节点表来检测这种行动。在这种情况下，文件在被从原来的位置删除前，实际移动到新磁盘上并被赋予了该盘上的一个新的索引节点。

与使用mv命令一样，当你用rm命令删除文件时，你也不能触及文件本身。取而代之的是，Linux标记该索引节点是自由的并把它放回可供使用的

空闲索引节点表中，目录中的该文件的条目被擦除。

4、连接

普通的连接实际上根本不是文件，它们只是指向同一索引节点的那些目录条目。索引节点表记录一个文件有多少个连接；只有当最后一个目录引用被删除时，该索引节点才最终被释放回空闲表中。显然，普通的连接不能跨越设备界线，因为所有的目录条目都指向同一个索引节点。Linux及大多数 UNIX的现代版本都有另一种连接，称为符号连接。对于这种连接，这个目录条目包含一个文件的索引节点(该索引节点本身又是对Linux逻辑文件系统上某处的另一个文件的引用)。一个符号连接可以指向同一磁盘或另一磁盘上的另一个文件或目录，也可以指向另一台计算机上的一个文件或目录。普通连接与符号连接之间一个主要的差别是：使用普通连接，每个连接都有同等的地位(也就是说，系统把每个连接都看作是原始文件)，并且在文件的最后一个连接被删除之前，实际的数据不会被删除；使用符号连接，当原始文件被删除时，所有对该文件的符号连接也都被删除，符号连接的文件没有与原始文件相同的地位。

除了连接与文件之间的这些细微的差别外，你可像直接访问文件那样对待和准确地访问 连接。

你可用 `ls-l` 命令来看出某个文件是一个连接文件，因为该命令的回应显示了这个本地文件名，接着指示了被连接的文件：

```
lrwxrwxrwx 1 root root 4 Oct 17 15 : 27 Info- > info/
```

为了指示该文件是一个连接文件，文件权限标志以 `l` 开头。

5、特殊文件

每个与Linux系统相连的实际设备(包括磁盘、终端和打印机)，都在文件系统中表示出来。大多数设备都放在 `/dev` 目录中。例如，如果你正在系统控制台(`console`)上工作，那么你的相关设备就叫 `/dev/console`。如果你正在标准终端上工作，那么你的设备名可能是 `/dev/tty01`。终端或串行线都称为 `tty` 设备(它代表电传打字机，电传打字机是最初的UNIX(终端)。为了确定你的 `tty` 设备叫什么名字，键入 `tty` 命令，系统会显示你所连接的设备的名字。

打印机和终端称作特殊字符设备，它们可接受和产生字符流。另一方面，磁盘把数据存储在由柱面和扇区编址的块中，你不能只访问磁盘上的一个字符，你必须读写整个块。对于磁带通常也是这样的，这种类型的设备称作特殊块设备。对更复杂的情况，磁盘和其他特殊块设备必须能像面向字符的设备那样工作，因此每个特殊块设备都有一个与之匹配的特殊字符设备。Linux通过读取要送到字符设备的数据并为特殊块设备转换这些数据来实现这个转换过程，这一过程不需要你做任何事。

至少还有另一种类型的特殊设备你可能遇到，即称为管道的FIFO(先入先出缓冲区)。FIFO看起来像普通文件，如果你向里写入，它们就增长；如果你从FIFO读出，它的大小就会缩减。FIFO主要用于系统进程中，以允许众多程序向一个单一的控制进程发送信息。例如，当你用 `lp` 命令打印一个文件时，`lp` 命令建立打印进程并通过向一个FIFO发送信息来通知 `lpsched` 守护程序。守护程序是一个不用用户要求就执行的系统进程。

一个非常有用的特殊设备文件是位存储桶(`bit bucket`)/`dev/null`，你送入 `/dev/null` 的任何东西都被忽略，当你不想看到命令的输出结果时，它是很有用的。例如，如果你不想看到打印在标准错误设备上的任何诊断报告，你就可以用下列命令把它们放入位存储桶中：

```
ls-la > /dev/null
```

6、文件权限

Linux中的文件极限的含义比你在文件或目录上所拥有的权限的含义多。虽然权限决定谁能读、写或执行一个文件，但它们也决定这个文件的类型及如何执行这个文件。

可以用长格式的列表命令 `ls-l` 来显示一个文件的权限。`-l` 选项告诉 `ls` 命令使用长格式的列表，如果你键入 `ls-l`，则你可能看到像下面这样的目录列表：

```
drwx-----2 sglines doc 512 Jan 1 13 : 44 Mail
drwx-----5 sglines doc IO24 Jan 17 08 : 22 News
-rw-----1 sglines doc 1268 Dec 7 15 : 01 biblio
drwx-----2 sglines doc 512 Dec 15 21 : 28 bin
-rw-----1 sglines doc 44787 Oct 20 06 : 59 books
-rw-----1 sglines doc 23801 Dec 14 22 : 50 bots.msg
-rw-r-----1 sglines doc 105990 Dec 27 21 : 24 duckie.gif
```

该列表几乎显示了目录条目和文件的索引节点所包含的与文件有关的所有信息。

第一列显示文件的权限，
 第二列显示对一个文件(或一个目录中的附加块)的连接数量，
 第三列显示谁拥有该文件，(所有权有三种可能：所有者、所有者的组和其他人。)
 第四列显示文件所属的组，
 第五列显示文件中的字节数，
 第六列显示文件创建的日期和时间，
 第七列显示文件本身的名字。
 权限字段(第一列)分为四个子字段：

-rwx rwx rwx

第一个子字段定义文件的类型。普通文件以连字符(-)作为占位符，目录用一个d标记。

下表显示了文件类型子字段的权限值。

字符 含义

- 普通文件

b 特殊块文件

c 特殊字符文件

d 目录

l 符号连接

接下来的三个子字段显示文件的读、写和执行权限。例如，这三个子字段中的第一个子字段的rwx是说所有者对文件有读、写和执行权限。这三个子字段中的第二个子字段是说拥有该文件的组对文件有读、写和执行权限，第三个子字段显示其他人的权限。

这些权限字段可以显示更多的信息。事实上，有几种属性包含在这三个字段内。可惜，这些属性的含义是由你所使用Linux的版本及文件是否是可执行文件来决定的。

四、 文件系统的管理

1、 安装文件系统

首先介绍有关文件系统的一些概念。在一个文件系统被访问之前，它必须安装(mount)在某个目录下。例如：如果在软盘上有一个文件系统，则必须将它安装到某个目录下，比如mnt，以便能够访问其上的文件。安装了文件系统后，该文件系统的所有文件都会显示在那个目录下。卸载文件系统后，该目录(这里为/mnt)变为空。

硬盘上的文件系统也是如此。在启动时系统自动将文件系统安装到硬盘上，根文件系统被安装在目录“/”下，如果/usr有另一个单独的文件系统，它则安装在/usr下。如果只有一个根文件系统，所有的文件(包括那些在/usr下的)都存放在该文件系统中。

mount命令用来安装文件系统。

命令：

mount-av

从文件/etc/rc中执行(它是在启动时的系统初始化文件)，mount-av命令从文件/etc/fstab中获取文件系统和安装点的信息，fstab文件的一个例子如下：

\$ device directory type options

/dev/hda2 / ext2 defaults

/dev/hda3 /usr ext2 defaults

/dev/hda4 none swap sw

/proc /proc proc none

第一个域是设备名，即安装的分区名；

第二个域是安装点；

第三个域是文件系统类型，如ext2(ext2fs文件系统)或minix(Minix文件系统)。

fstab文件的最后一个域包括了mount的选择项，通常设置为“ defaults ”。

可以看出，交换分区也包括在/etc/fstab中，其安装目录为none，类型为swap。命令swapon-a也是从/etc/rc中执行的，可用来交换列在/etc/fstab中的所有交换设备。

安装或卸载文件系统，有许多软件包可以使普通用户安装或卸载文件系统(尤其是软盘)而不破坏系统的安全性。

mount-av命令实际上安装除根文件系统(在fstab文件中为/dev/hda2)以外的全部文件系统，根文件系统在启动时由核心自动安装。

除了用mount-av以外，还可以手工安装文件系统。命令：

```
$ mount -t ext2 /dev/hda3 /usr
```

等价于在上例的fstab文件中安装文件系统/dev/hda3。

一般地，不会用手工安装或卸载文件系统，/etc/rc中的mount-av命令在启动时负责安装文件系统，文件系统在用shutdown或halt命令关闭系统之前自动卸载。

2、检查文件系统

通常情况下，不断地检查文件系统中被破坏的文件是一个好的办法，有些系统在启动时自动检查它们的文件系统(用/etc/rc中适当的命令)。

用于检查文件系统的命令依赖于出问题的文件系统类型，对ext2fs文件系统而言(最常用的类型)，命令为e2fsck，例如，

```
命令：$ e2fsck -av /dev/hda3
```

将检查/dev/hda3上的ext2fs文件系统，并自动纠正任何错误。

通常在检查文件系统之前最好卸载该文件。例如，

```
命令
```

```
$ umount /dev/hda3
```

将卸载/dev/hda3上的文件系统，然后再检查该文件系统，但是不可能卸载根文件系统。为了在没有卸载之前就检查根文件系统，可以使用维护磁盘boot/root；当文件系统中存在文件“忙”，即被其它运行进程使用着时，也不能卸载该文件系统。例如当某个用户的当前工作目录在某个文件系统上时就不能卸载该文件系统。如果试图卸载一个正在被使用的文件系统，将得到一个“ Device busy ”的错误信息。

其它类型的文件系统用e2fsck命令的其它形式，比如efscck和xfscck。在某些系统中，可以简单地使用fsck命令，它将首先决定文件系统类型，再执行相应的命令。

有一点很重要，那就是在检查完文件系统后，如果对该文件系统做了任何纠正，就应该立即重新启动系统(当然，一般情况下，不能在文件系统被安装时检查它)。例如，如果e2fsck报告对文件系统的错误作了些纠正的话，就应该立即用shutdown-r命令确保重新启动系统，这使得当e2fsck修改了文件系统后，系统能重新同步读文件系统的信息。

/proc文件系统从不需要以这种方式检查，/proc是一个内存文件系统，直接由核心管理。

除了为交换空间保留单独的分区以外，还可以使用交换文件。然而为了能做到这样，必须安装相应的Linux软件，并将一切安排就绪，才能创建交换文件。

如果安装了Linux系统，则可以用下列命令创建交换文件。下面将创建一个大小为8208块(约8MB)的交换文件：

```
$ dd if=/dev/zero of=/swap bs=1024 count=8208
```

这条命令创建交换文件，“ count=8208 ”项为交换文件的块数。

```
$ mkswap /swap 8208
```

该命令将初始化交换文件，同样地也可以用相应的值替换交换文件的命令和大小。

```
$ /etc/sync
```

```
$ swapon /swap
```

同步以后便可对刚创建的/swap文件进行交换。同步是用来保证文件被写到磁盘上，这样做的一个主要缺点是，所有对交换文件的访问都是通过文件系统来实现的，这意味着组成交换文件的块可能不连续，因此性能也许不如使用交换分区，那样所有的块都是连续的，而且I/O请求是直接和设备打交道的。

使用交换文件的另一个缺点是在使用大的交换文件时，破坏文件系统中数据的机会变大，如果出了错误的话，那将破坏文件系统。将文件系统和交换分区分开可以避免该现象发生。如果暂时需要很大的交换空间，使用交换文件便十分有用。例如，如果编译一个很大的程序，并希望运行得快一点的话，可以暂时创建一个交换文件，并同常规的交换空间一起使用。

取消交换文件，首先用swapoff，例如：

```
$ swapoff /swap
```

然后再安全地删除该文件

```
$ rm /swap
```

请记住每个交换文件(分区)最大为16MB，但可以在系统中使用多达8个交换文件或分区。

3、 其它任务

对系统管理员来说还有许多日常事务性的工作：

(1) 系统启动文件

系统启动时，在任何用户注册之前系统将自动执行一系列初始化程序。下面是对该过程的详细描述：

在启动时，内核产生进程/etc/init，init是一个程序，它读取其配置文件/etc/inittab，并根据该文件的内容产生其它进程。由inittab产生的最重要的进程之一就是可以在每个虚拟控制台(VC)上启动的/etc/getty进程。getty进程抢占虚拟控制台VC，并在VC上启动login进程，它允许你在每个VC上注册，如果/etc/inittab不包含某个VC的getty进程，就不能在该VC上注册。

另一个从/etc/inittab中产生的进程是/etc/rc，即主系统的初始化文件。该文件是一个shell程序，它执行在启动时需要的任何初始化命令，诸如安装文件系统和初始化交换空间。

系统也可以执行其它的初始化程序，例如/etc/rc.local。/etc/rc.local通常包含针对用户自己系统的初始化命令，诸如设置主机名。rc.local可以从/etc/rc或从/etc/inittab中直接启动。

(2) 设置主机名

在一个网络环境中，主机名是用来识别某个机器的唯一标识，而在一个单机环境中，主机名只给出了系统的称呼。

设置主机名是一件很简单的事，只需用hostname命令即可。如果在一个网络中，主机名必须是机器的主机名全称，比如ccjsj.com；如果不是网络环境，则可以选择任意的主机名和域名，比如loomer.vpizza.com，shoop.nowhere.edn或floof.org。

设置主机名时，主机名必须在文件/etc/hosts中，它给每个主机赋一个IP地址，即使机器不在网络上，也应该在/etc/hosts中包含用户自己的主机名。

例如，如果主机不在TCP/IP网络上，主机名为floof.org，则只需在/etc/hosts中包括下列项：

```
127.0.0.1 floof.org localhost
```

它给主机floof.org赋了一个回送地址127.0.0.1(如果不在网络上就使用该地址)，别名localhost也赋给该地址。

如果在TCP/IP网络上，真正的IP地址和主机名也应该在/etc/hosts中，例如，如果主机名为www.ccjsj.com，IP地址为128.253.154.32，则在/etc/hosts中增加下列行：

```
128.253.154.32 www.ccjsj.com
```

如果主机名不在/etc/hosts中，那么将无法设置它。

设置主机名，只需使用hostname命令即可，例如命令：

```
$ hostname-S www.ccjsj.com
```

设置主机名为www.ccjsj.com。多数情况下，hostname命令是由某个系统启动文件执行的，例如/etc/rc或/etc/rc.local。编辑上述两个文件，改变其中的hostname命令来设置主机名，再重新启动系统时，主机名将被设置为新值。

(3)、紧急情况处理

在某些偶然的场合下，系统管理员将面临从一个彻底性的灾难中恢复系统的问题，例如忘记了根的口令或破坏了文件系统。

Linux并不是一个不稳定的UNIX版本，事实上Linux比商用UNIX在许多平台上出现的问题要少得多。Linux还提供一个强有力的向导，它能帮助你走出困境。

解决问题的第一步就是试图自己修复，到处看一看，看该如何办。系统管理员在仔细研究问题之前将会花大量的时间，不顾一切地发出请求帮助，多数情况下，你将发现自己解决问题实际上非常容易。

很少情况下需要重新安装被损坏的系统。许多新用户由于不小心删掉了一些基本的系统文件，便立即去找安装盘，这并不是一个好主意。在采取这项极端的方法之前，仔细研究一下并寻求其他人帮助解决问题。多数情况下，可以从维护磁盘上恢复系统。

* 用维护磁盘恢复

系统管理员的一个必不可少的工具就是所谓的"boot/root"盘，它是一个可以以完整的Linux系统启动的软盘，它与硬盘无关。boot/root盘实际上很简单，首先用软盘创建一个根文件系统，将所需要的应用例程放在上面，再在该软盘上安装LILO和可启动的内核。另一种方法是用一个软盘装内核，另一个装根文件系统，两种情况结果都一样，即可以完全从软盘上启动Linux系统。

sunsite . unc . edu中/pub/Linux/GCC/rootdisk中H . J Lu的boot/root盘是另一个维护磁盘的例子，如果你想试一下的话也可以自己建一个。然而多数情况下，用事先做好的boot/root盘则比较容易而且可能更加完整。

使用boot/root盘是很简单的，只需要用该盘启动系统，并用root注册(通常没有口令)。为了存取硬盘上的文件，需要手工安装文件系统，例如命令
\$ mount -t ext2 /dev/hda2 /mnt

把/dev/hda2上的ext2fs文件系统安装在/mnt上。记住"/"现在在boot/root盘上，用户需要将硬盘的文件系统安装在某个目录下，以便存取其上的文件。因此，如果将根文件系统安装在/mnt下的话，硬盘上的/etc/passwd则在/mnt/etc/passwd中。

* 修复根口令

如果忘记了根口令，可以用boot/root盘启动，将根文件系统安装在/mnt下，并在/mnt/etc/shadow中将root的口令域清除。

这时root没有口令。当用硬盘重新启动时，可以用root注册，并用passwd重新设置口令。如果知道如何使用vi，则可以在boot/root盘上使用它。因为在boot/root盘上，其它的编辑器，如Emacs，可能没有，但vi一定有。

* 修复损坏的文件系统

如果莫名其妙地破坏了文件系统，则可以用软盘运行 e2fsck来纠正文件系统中任何被破坏了的数据(当然指的是用ext2fs文件类型)，其它文件系统类型可以用 fsck命令的其它形式。

当从软盘检查文件系统时，最好此时文件系统没有被安装。

最常见的文件系统损坏就是超级块损坏。超级块是文件系统的头，它包含了文件系统的状态、大小、空闲块等，如果超级块被损坏(例如不小心将数据直接写到文件系统的分区表中)，系统将根本无法识别文件系统。任何试图安装文件系统的操作都将失效，e2fsck也无法修复系统。

幸运的是ext2fs类型的文件系统在驱动器“块组”的边界保存了超级块的备份，通常情况下是每8K块为一个块组。为了告诉e2fsck用超级块的拷贝，可以用诸如下面的命令：

```
$ e2fsck -b 8192 <partition>
```

其中<partition>是文件系统驻留的分区表，-b 8192选择项告诉e2fsck用文件系统中存储在块8193的超级块备份。

* 恢复丢失的文件

如果不小心删除了系统中的重要文件，则没有办法取消删除(undelete)。然而可以从软盘拷贝相关的文件到硬盘中，例如，如果删除了系统中的/bin/login(允许注册的文件)，只需用boot/root软盘启动，将根文件系统安装到/mnt上，并用命令：

```
$ cp -a /bin/login /mnt/bin/login
```

-a选择项告诉cp保留被拷贝文件的存取权限。

当然，如果删除的文件并不是基本的系统文件，而这些文件在boot/root软盘上没有，那么就没办法了。不过如果做了备份，当然总是可以恢复的。

* 恢复损坏的库

如果不小心破坏了库或在/lib中的符号连接，那么依赖于那些库的命令将无法运行。最简单的办法就是用boot/root盘启动，安装根文件系统，并在/mnt/lib下修复库。

第五节课：系统管理（一）

- 1、明确系统管理的目的是什么。
- 2、了解Linux口令加密的两种方式。
- 3、学会使用日志排除故障及系统的安全。

4、了解Linux的7个运行级。

一、用户管理

1、用命令行添加及删除用户帐号：

useradd passwd usermod userdel

2、手工添加、更改及删除用户：

- 用户帐户记录文件：/etc/passwd

passwd文件每行为7个域（字段），分别为：

用户名、口令、用户ID（UID）、组ID（GID）、用户说明、主目录、缺省shell。

- 用户口令文件：/etc/shadow

Linux的两种加密方式为DES（Data Encrypt Standard）和MD5（Standards for Message Digest）：

（1）DES是一种高级算法，对口令的长度有限制，必须在8个字符以内，加密后为13个字符的散列值。用常用字典可以很容易猜出口令。

（2）MD5口令长度为1到32个字符，加密后的散列值为32个字符。

/etc/shadow文件每条记录对应的字段分别为：

使用者名称：最多8个符号，可以用大小写，通常都是小写。直接对应/etc/passwd文件中的使用者名称。

密码：加密过的32个字节。一个空格（就是：）表示登入时不需密码（不是个好主意），一个“*”项目（就是：*）指出帐号已经关闭。

密码最后一次变更起所经过的日数：（从1970年一月一日起）。

密码经过几天可以变更：（0表示可以随时变更）

密码经过几天必须变更：（99999表示使用者可以保留他们的密码很多很多年不变）密码过期之前几天要警告使用者：（7为一周）

密码过期几天后帐号会被取消

帐号经过几天会被取消

保留栏位

- 组信息文件：/etc/group

Linux下的组只是单纯的分组，没有权限分配的功能。若要对组进行权限分配，必须借助工具软件，如sudo。

- 用户配置文件的模板：/etc/skel/.*

用户的提示符等缺省配置文件是从/etc/skel目录拷贝过来的。

练习：

1、手工建立用户user，要求：主目录为/user；说明行为mail user；shell为/bin/bash。

（1）vi /etc/passwd，在此文件内加入下行：

user:x:501:501:mail user:/user:/bin/bash

（2）vi /etc/shadow，在此文件内加入下行：

user::11533:0:99999:7:::

（3）用passwd命令为用户设置口令：passwd user

（4）在根目录下建立user目录：mkdir /user

（5）拷贝/etc/skel/. *到/user目录下：cd /etc/skel；cp -r . * /user

（6）更改/user目录的属主以及属组：chown -R user.user /user

2、把user用户更名为john，并且把主目录改为/home/john，其他缺省。

（1）vi /passwd，把user字段改为john，把主目录/user改为/home/john

（2）vi /shadow，把user字段改为john

（3）拷贝/user /home/user：cp -r /user /home

（4）更名/home/user为/home/john：mv /home/user /home/john

(5)更改属主及属组：chown -R john.user /home/john

3、设置john的shell,使john不能交互（不能在本机登录，也不能利用telnet登录到本机。）但是能登录到本机的ftp服务。

（1）vi /bin/myshell (自己创建此文件)：

echo you can 't use telnet.

echo you can use ftp.

(2) chmod 777 myshell (更改为可执行)

（3）vi /etc/shells，在此文件中加入下行：

/bin/myshell (目的使myshell成为有效系统shell)

二、日志管理

- 日志文件位置：/etc/log/messages
- 日志的配置文件：/etc/syslog.conf

练习：

1、把一台主机名为bsd的主机的日志放入一台主机名叫Linux的主机日志内？

（1）此时bsd主机为客户机，在客户端：

vi /etc/syslog.conf，加入下行：*.*@linux

(2) 此时Linux主机为服务器，在服务器端：

vi /etc/rc.d/init.d/syslog，加入下行：daemon syslogd -a -o -r

注：此种方法在集群服务器中常用。

注：如果发现自己的服务器有文件被更改或破坏的痕迹，而日志没有记录，则此系统不再被信任，可以考例重装并采取更高级的安全措施。

三、权限管理

1 给/tmp 单独分区（只需大约50M）可能发生缓冲区溢出或使用磁盘限额。

2 尽量避免使用suid如必须用最好用sgid代替

第六节课：系统管理（二）

一、理解运行级

其实init是Linux系统里的根进程，是系统所有进程的祖先。它的主要作用是根据记录在/etc/inittab里的一个脚本（script）程序产生进程。这个文件通常用于控制用户的登录模式。，不过沿用UNIX系统的至多6种的限制，一般只有1到6有效。

init一般在系统启动时自动运行，也可以由root用户调用。它的作用是切换系统的运行状态。它的命令格式是：init NUM 回车，其中NUM表示1到6的任一数字。

6种模式中有两种是系统内定的：

0：停止系统运行。init 0 回车 相当于 halt 回车。

6：重启系统。init 6 回车 相当于 reboot 回车。

注：如果你了解/etc/inittab文件，就千万不要把0设为缺省登录模式，否则你的Linux的运行就只能两点一线了——开机çè停机，或者是开机çè重启的无限循环。

init 1——进入单用户模式。

这和开机是选择Safe Mode是一样的，可以以root的权限对系统进行管理，但是不能进入图形模式，也不支持中文，声卡和网络服务等就更不用提了。它和Windows的安全模式是比较相似的。

init 2——进入无网络服务的多用户模式

这一模式下，在本机上的使用基本不变。你仍然可以访问网络，上Internet，ftp或telnet到网络中的任何一台服务器。只是你的计算机在网络中不给其他机器提供服务而已。计算机从第3运行级转入这一模式时你会看到一系列网络服务被停止。

注：但注意如果原来已经有用户ftp或telnet进入了你的系统，用init 2是不能中断他和你的连接的，小心让别人在这个时候黑你一把哦！——执行init 2前先用who命令检查一下有谁在线吧。

init 3——进入控制台登录的多用户模式

如果是从init 1 或init 2切换回init 3，我们可以看到一系列服务被重启。

init 5——进入图形化登录的多用户模式

系统启动后或执行init 5命令后，悦目的蓝绿过渡色背景上就出现了漂亮的登录窗口。

更深入的了解.....

如果你想知道更多，就不妨打命令man init看看完全版的手册，枯燥的手册会告诉你：命令init结合/etc目录下的一系列文件可以对你的系统的启动实现深入的定制化设置.....

练习：

1、进入单用户模式，并更改root用户的口令；不要重启系统，进入多用户网络维护模式；最后进入文本多用户网络模式。

二、启动脚本 /etc/inittab

当系统内核引导到最后一步时，init 进程被启动，它会自动查找 /etc/inittab 文件，看看是否有类型为 initdefault（参阅 inittab(5)）的记录。

initdefault 记录决定系统初始运行级别。如果没有这条记录（或者根本就没有 /etc/inittab），那么，你必须在系统控制台输入想要进入的运行级别。

练习：

1、编辑/etc/inittab文件，使系统的缺省运行级为文本多用户网络模式，并且禁止用户用Ctrl+Alt+del重启计算机。

Trap ctrl+alt+del下面有一行 将此行注解掉

ca: : ctrlaltdel:/sbin/shutdown -t3 -r now (此行前面加#号即可)

三、系统启动的步骤

step 1 执行 /sbin/init 系统的第一个进程

step 2 执行 /etc/rc.d/rc.sysinit 第一个执行的script文件（负责设置环境变量，检查与挂载文件系统启动swat文件设置网络等）

step 3 执行运行级中的script文件

step 4 执行 /etc/rc.d/rc.local最后一个执行的script文件如果要每次开机时执行某个程序可在此文件中设置

信号

init 对以下信号产生响应

SIGHUP

当接收到该信号后，init会对 /etc/inittab 和 /var/log/inittab 文件进行检查。如果这两个文件之一存在而且文件中有 ASCII 字符的运行级别，init 会转换到相应的新的运行级别。

SIGUSR1

当接收到这个信号量，init 会关闭并重新打开它的控制 fifo，/dev/initctl。此操作对 /dev 被重新挂载后的启动脚本有用。

SIGINT

通常，当用户按了 CTRL-ALT-DEL 键后，内核会向 init 传递此信号。它所执行的操作与 ctrlaltdel 相同。

SIGWINCH

当键盘有 KeyboardSignal 按键按下时，内核向 init 传递此信号，它激活 kbrequest 动作。

- 其他启动脚本几乎都放在 /etc/rc.d 及 /etc/init.d下面。

四、redhat的系统配置命令setup的使用

用此命令可以配置redhat的启动服务等，在system services内可以配置系统启动时启动哪些服务。

五、软件包的管理——RPM的使用

1、rpm的特性

- 软件包的可升级性

使用RPM您可以单独升级系统中的某些部件而无须整个重新安装。当您获得了一套基于RPM 新版操作系统时(如红帽子 Linux), 您无须重新安装机器(而基于其它软件打包机制的常常需要这么做)。RPM 允许智能的，全自动的就地升级系统。包中的配置文件在升级过程中会予以保留, 因此您不会丢失定制信息。

- 功能强大的软件包信息查询

RPM 拥有功能强大的查询选项. 您可以搜索数据库来查询软件包或文件。您可以简便的查出某个文件属于哪个软件包或出自哪儿。RPM软件包中的文件以压缩格式存放,拥有一个定制的二进制头文件，其中包含有关包和内容的有用信息，可以让您对单个软件包的查询简便又快速。

- 系统验证

另一项强大的功能是进行软件包的验证。如果您担心误删了某个软件包中的某个文件，您就可以对它进行验证。任何非正常现象将会被通知。此时，如果需要的话，您可以重新安装该软件包。在重新安装过程中，所有被修改过的配置文件将被保留。

- 保持软件包的原始特征

一项严格的设计目标是要保持软件包的原始特征,就象该软件的原始发布者发布软件时那样。通过使用RPM,您可以拥有最初的软件 and 要使用的任何补丁程序还有详细的软件构建信息。从几方面来说这是非常之大的优点。例如，当某个软件的新版本发行时,您无须从头编译所有文件。您可以看一看补丁程序都需要做些什么。在这种方式下，所有编译条件以及为生成软件所做的修改都将是可见的。

2、rpm的使用

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 #####
```

- 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, 您可以在命令行上使用--replacepkgs 选项，这将忽略该错误信息。

- 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。

- 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。

- 卸载

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 needed by bar-1.0-1

- 升级

升级软件包和安装软件包十分类似：.

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

```
foo #####
```

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于该软件包中的配置文件。因此，RPM 会备份老文件，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。

因为升级其实就是软件包的卸载与安装的综合,您可能会遇见那些操作中所可能发生的错误。有一处不同，那就是当您使用旧版本的软件包来升级新版本的软件时，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

要使RPM 坚持这样“ 升级 ”，就使用 --oldpackage 命令行参数。

- 查询

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，您还可以连同-q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 <file> 将查询包含有文件<file>. 的软件包

-p <packagefile> 查询软件包文件名为<packagefile>. 的软件包

有几种方式来指定查询软件包时所显示的信息。以下选项用于选择您感兴趣的信息进行显示。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

验证软件包是通过比较compares从软件包中安装的文件和软件包中的原始文件信息来进行的。除了其它一些东西, 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来验证某个软件包:

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

rpm -q - whatprovides 文件名 (查询该文件是由哪个软件包提供的)

练习:

1、查询sendmail软件包的版本号。

六、源代码的管理和编译。

1、tar命令的使用

tar的使用是命令行的, 可以用来打包或者解包

tar-xvzf<文档包名>解压缩被tar打包并被gzip压缩过的文档包。此类文档包一般以.tgz或.tar.gz结尾。

tar-xvf<文档包名>

解压缩被tar打包但未被gzip压缩的文档包。此类文档包一般以.tar结尾。

tar-tvf<文档包名>

列出文档包内所包含的文件。

tar-cvzf<文档包名><文件名>

将符合命令行中文件名的文件打包并且压缩, 制作成文档包。文件名中可以使用通配符*或?, 文档包需要以.tgz或.tar.gz结尾。

2、源代码的编译过程

Linux下应用软件安装包一般为filename.tar.gz压缩包格式, 首先将DOWNLOAD的软件解包:

```
tar zxvf filename.tar.gz or tar zxvf filename.tgz
```

解包后如果得到的是可执行文件, 则可直接按照INSTALL或README文件中的说明进行配置; 如果解包后得到的是文件源代码, 则还要完成一个源代码的编译过程。

进入文件展开的目录, 如果看到:

A、configure文件, 则执行:

```
./configure
```

```
make
```

```
make install
```

B、Imake文件，则执行：

```
xmkmf
make
make install
```

C、Makefile文件，则执行：

```
make
make install
```

第二部分：网络服务

第七节课：DHCP服务器的配置

- DHCP的主要功能

DHCP的全称是动态主机配置协议（Dynamic Host Configuration Protocol），由IETF（Internet 网络工程师任务小组）设计，详尽的协议内容在RFC文档rfc2131和rfc1541里。目的就是为了减轻TCP/IP网络的规划、管理和维护的负担，解决IP地址空间缺乏问题。运行DHCP的服务器把TCP/IP网络设置集中起来，动态处理工作站IP地址的配置，用DHCP租约和预置的IP地址相联系，DHCP租约提供了自动在TCP/IP网络上安全地分配和租用IP地址的机制，实现IP地址的集中式管理，基本上不需要网络管理人员的人为干预。而且，DHCP本身被设计成BOOTP（自举协议）的扩展，支持需要网络配置信息的无盘工作站，对需要固定IP的系统也提供了相应支持。

- DHCP的工作原理
- 几个DHCP名词

在介绍DHCP工作原理以前，先解释这几个名词的含义：

DHCP客户：DHCP客户是一通过DHCP来获得网络配置参数的Internet主机，通常就是普通用户的工作站。

DHCP服务器：DHCP服务器是提供网络设置参数给DHCP客户的Internet主机。

DHCP/BOOTP 中继代理：在DHCP客户和服务器之间转发 DHCP 消息的主机或路由器。

DHCP是基于客户机/服务器模型设计的，DHCP客户和DHCP服务器之间通过收发DHCP

最后一项选项是零长度的End(选项 255)，表明这是选项的结束以便DHCP客户处理。采用选项编码的好处是不论选项有多长，DHCP客户都可以正确接收，即使是它不认识的选项（不见得所有的DHCP客户程序都完全遵循RFC标准）。

DHCP客户并没有自己的IP，所以包含DHCP消息的UDP数据报的IP头的源地址段是0.0.0.0，目的地址则是255.255.255.255。

- DHCP分配IP地址的过程:

DHCP客户机初始化TCP/IP，在本地物理子网上广播一个 DHCPDISCOVER 消息，以确定DHCP服务器位

置及其IP地址。如果DHCP服务器和客户不在同一个物理子网上，BOOTP中继代理将转发这个消息给DHCP服务器。由于网络上可能不止一个DHCP服务器，凡所有具有有效IP地址信息的DHCP服务器向客户机发出一个提议。客户机从接收到的第一个提议中选定IP地址信息，并广播一条租用地址的消息请求。由发出该提议的DHCP服务器响应该消息，指定IP地址信息给该客户机并发送一个确认，而所有其它DHCP服务器撤回各自的提议。客户机完成TCP/IP协议的初始化和绑定。配置完成后，客户机就可以使用普通网络通信和连接至其它IP主机时用到的所有IP服务和应用。

- DHCP 服务器软件的安装

在Linux几乎都采用的是Paul Vixie/ISC DHCPd，来实现DHCP服务器端功能。可以访问

<http://www.isc.org/isc>

获得最新消息。

- 修改配置文件

DHCPd默认的配置文件是/etc/dhcpd.conf，这是一个文本文件，DHCPd里有一个语法分析器，能对这个文件进行语法分析，获得配置参数。dhcpd.conf 格式是递归下降的，关键字大小写敏感，可以有注释，注释以#开头，一直到该行结束。这里给出一个简单的dhcpd.conf的例子，所服务的网络为C类保留网络 192.168.1.0

```
#examples
# 缺省租约时间：default-lease-time 28800;
# 最大租约时间：max-lease-time 43200;
# 子网掩码选项：option subnet-mask 255.255.255.0;
# 广播地址：option broadcast-address 192.168.1.255;
# 路由器地址：option routers 192.168.1.1;
# DNS地址：option domain-name-servers 192.168.1.1;
# 域名：option domain-name "ccjsj.com";
# 以上都是全局参数

# 子网声明和掩码：subnet 192.168.1.0 netmask 255.255.255.0 {
# 范围：
range 192.168.1.10 192.168.1.100;
range 192.168.1.150 192.168.1.200;
}
```

这段配置文件将允许DHCP服务器分配两段地址范围给DHCP客户，192.168.1.10-100 和192.168.1.150-200

如果DHCP客户在申请租约时不请求一个特定租约失效时间，则以default-lease-time(28800秒)为租约时间，如果有请求一个特定的租约失效时间，则采用max-lease-time（432000秒）

服务器发送下面的参数给DHCP客户机：

子网掩码是255.255.255.0，广播地址是192.168.1.255，默认网关是192.168.1.1，DNS是192.168.1.1。

如果要为一台叫做ftp的机器指定固定的IP地址，可以在dhcpd.conf文件加一条

```
host ftp {

# ftp上网卡的硬件地址
hardware ethernet 08:00:00:4c:58:23;
#固定IP
fixed-address 192.168.1.210;
}
```

- dhcpd.leases

dhcpd.leases 是DHCP客户租约的数据库文件，默认目录在/var/state/dhcp/，文件包含租约声明，每次一个租约被获取、更新或释放，它的新值就被记录到文件的末尾。在DHCPd第一次安装后，并不会生成这个文件。但DHCPd的运行需要这个文件，所以可以建立一个空的文件。

```
# touch /var/state/dhcp/dhcpd.leases
```

DHCPd记录这个文件的格式是

```
lease ip-address { statements... }
```

每个记录包含一个提供给客户的IP地址，在花括号里的语句包含一些租约信息。具体的租约信息因客户发出不同的DHCP请求而稍有差别。

- 运行DHCPd

要启动DHCPd，简单地键入 `/usr/sbin/dhcpd` 或用 `ntsysv` 把DHCPd服务自动启动，也可以用 `/etc/rc.d/init.d/dhcpd start`，

这样启动后，DHCPd是启动在eth0上，如果DHCPd上的服务器还有另外一块网卡eth1，想在eth1上启动dhcpd，就键入

```
#/usr/sbin/dhcpd eth1
```

以上例子的dhcpd.conf来启动dhcpd，：

要注意的是dhcpd.leases的时间记录采用GMT时间，而不是本地时区的时间。要查看本机的GMT时间可以用 `date -u`

总结

以上就是DHCPd常用配置，实际应用DHCP还要考虑IP分配的一些策略问题，同时要保证网络的健壮性，必须至少要有两台DHCP服务器一起工作，如果一台出了故障，另一台可以继续为DHCP客户服务。然而目前DHCP协议里并没有能让两台DHCP服务器协同工作的机制，不能保证分配的地址的唯一性，所以这两台DHCP服务器里的可分配地址空间必须进行调整，不能有交叉重复的IP地址。

第八节课：DNS服务器的配置

一、什么是DNS

DNS是域名解析服务器的缩写，简单的说就是把域名转换成与其相对应的IP地址，还可以把IP地址反向解析转换成域名。有关DNS的说解，请参考其它的DNS文章。

二、前提条件

1、在安装Linux时你要选择安装DNS服务，并且要在Linux启动时同时也启动DNS。

2、可能用到的文件

文件名	所在目录	
named.conf	/etc/	"系统自带"
named.local	/var/named/	"系统自带"
name-ip (正向解析)	/var/named/	"创建或生成"
ip-name (反向解析)	/var/named/	"创建或生成"

说明：如果你在安装完成LINUX后在配置网络时同时也配置了DNS选项话，你可能只要做很少的工作就能完成DNS服务的安装。详细说明如下：

假设你的IP为局域网的内部IP：192.168.1.1 域名为：ccjsj.com

则在控制台下键入：`netconf`后输入详细的DNS信息后，即在此选项：Domain Name Server (DNS)里配置DNS的相关信息后，再在控制台下输入命令：`ndc restart`后就会在"/var/named"目录下自动生成“正向解析”和“反向解析”的2个文件，文件名应该是：“192.168.1”和“ccjsj.com”，你只需按照下面的格式添写就可以完成DNS的配置了。你也可以任意给这两个文件起名，只要你能看得懂就可以，但一定要在文件named.conf中对应上。

三、上述5个文件的内容如下：

1、named.conf

```
options {
    directory      "/var/named";
};
```

```

zone "." {
    type hint;
    file "named.ca";
};
zone "ccjsj.com" {
    type master;
    file "ccjsj.com";
};
zone "0.0.127.in-addr.arpa" {
    type master;
    file "named.local";
};
zone "1.168.192.in-addr.arpa" {
    type master;
    file "192.168.1";
};

```

3、named.local

```

@      IN      SOA      localhost.root. localhost (
1997022700 ; Serial (根据此数字判断是否要更新资料)
28800      ; Refresh (检查资料库文件是否更新的时间)
14400      ; Retry (更新失败后再次尝试时间)
3600000    ; Expire (更新失败后再次尝试时间)
86400 )    ; Minimum (缺省的生存时间)
IN        NS        localhost.
1         IN        PTR      localhost.

```

4、正向解析文件：ccjsj.com

```

@      IN      SOA      ns.ccjsj.com  root.ns.ccjsj.com (
1997022700 ; Serial
28800      ; Refresh
14400      ; Retry
3600000    ; Expire
86400 )    ; Minimum
NS        ns.ccjsj.com.
MX        10      ns.ccjsj.com.
;
localhost IN      A      127.0.0.1
ns         IN      A      192.168.1.1
www        IN      A      192.168.1.1
mail       IN      A      192.168.1.1
ftp        IN      A      192.168.1.1
;
ns         CNAME    ns.ccjsj.com.

```

```

www      CNAME      ns.ccjsj.com.
mail     CNAME      ns.ccjsj.com.
ftp      CNAME      ns.ccjsj.com.

```

5、反向解析文件：192.168.1

```

@      IN      SOA      ns.ccjsj.com  root.ns.ccjsj.com (
1997022700 ; Serial
28800      ; Refresh
14400      ; Retry
3600000    ; Expire
86400 )     ; Minimum
NS      ns.ccjsj.com.
1      IN      PTR      ns.ccjsj.com.

```

至此，已经完成DNS的配置，还有几个相关的文件，你可以参考一下：

/etc/host.conf

order host,

bind

multi on

/etc/HOST

ns

/etc/host

```

127.0.0.1      ns.ccjsj.com      ns      localhost.localhostdomain  localhost
192.168.1.1    ns.ccjsj.com      ns

```

```

192.168.1.1    ns.ccjsj.com      ns

```

/etc/sysconfig/network

hostname=ns

至此，DNS的配置已经完成，现在要进行测试，在控制台下输入以下命令：

ndc restart

nslookup

会出现一个提示符“>”，你可以输入你的主机名myname,及mail,ftp,www,机器将会解析成具体的信息，相信你会成功的。

四、为新建的域添加一个子域（域名为sh.ccjsj.com）子域的dns服务器主机名为computer10

1 只需在/var/named/ccjsj.com文件中加入下面两句话

```
ns IN NS computer10.sh.ccjsj.com.
```

```
computer10. IN A 192.168.1.100
```

2 重新启动然后用nslookup解析一下

3 如果想在子域解析父域则需要在子域的主配置文件中“directory=/var/named下”加入 forward only;

```
forwarders {
```

```
192.168.1.1;
```

```
};
```

然后重新启动。

如果要其它机器要用本机作为dns服务器则只需修改/etc/resolv.conf文件nameserver指向本机（nameserver 192.168.1.1）

注意：在资料库文件中凡是完整的主机名结尾一定要加点

练习：

1 配置一个最为简单的DNS服务器域名为ccjsj.com用自己的姓作为主机名

2 为ccjsj.com添加一个子域域名为sh.ccjsj.com

要求：a 在子域可以解析父域。

b 在父域可以解析子域。

第九节课：smb服务器的配置

Smb服务的配置

SMB是Server Message Block (服务消息块) 协议的英文缩写，是Windows forWorkgroup,Windows NT或者Windows使用共享文件和打印服务的标准化协议。Samba是由澳大利亚的Andrew Tridgell开发，一种在UNIX环境下运行的SMB客户/服务器软件包，提供与实现了UNIX与Windows之间的文件共享与打印，它可以在许多不同类型的UNIX上运行。由于Linux的分销商RedHat、Slackware、S.u.s.e等都提供Samba的软件包，并且由于Linux的文件系统性能优异，价格廉价以及爱好者甚多，所以本文以Redhat 6.0版的Linux为例来加以说明。

SMB以请求 - 回应方式运行，当客户机发送请求时，请求包含在SMB中传到服务器。服务器收到这些SMB，经过解析，对客户机发回应答。当一台计算机通过SMB在网络上共享资源时，它就成为一台服务器。当一台计算机隶属于共享资源时，它就成为一台客户机。在这种计算环境中，一台计算机不管它的操作系统是Linux 或者Windows 98，同时作为服务器与客户机都是可能的。SMB 通信使用NetBIOS界面上发生，它可以运行在许多协议上，最常用到的NetBIOS是TCP/IP上的NetBIOS，所以Windows 98上必须安装与支持TCP/IP协议，安装方法已有许多文章介绍，这里就不说明

一、配置Samba服务器

Samba服务器有三种安全模式：share、user和server。

Samba的配置文件只有一个，就是/etc/smb.conf文件。

server模式需要NT域，如果对安全性能不是十分讲求的部门就不需要，实际应用上我想仅有user模式足够，甚至share模式足矣，因为这两种模式配置和使用起来方便多多。

1、先把Samba服务器配置为用户安全模式[user]

(1) . [global]的配置

netbios name=linux

workgroup=SambaServer #服务器的工作组名不必与Windows客户机相同

server string=Samba Server

hosts allow=192.168.0. 127.

security=user

encrypt passwords=yes

smb passwd file=/etc/smbpasswd

wins support=no

其它参数为默认设置。

[tmp]

说明：[tmp]为测试用，去掉前面的#即可。测试成功后，即可把该段的参数应用到共享资源的[public]段，并使用命令

#chmod a+w [public]里的共享资源目录

使所有用户都能读写该资源目录。

(2). 建立Samba帐户, 编辑帐户文件/etc/smbpasswd, 删除不需要的帐户

#adduser user #生成的hgz帐户口令是空的。

#passwd user #需要口令时运行此命令增加口令。

#cat /etc/passwd | mksmbpasswd.sh > /etc/smbpasswd

注: 生成smbpasswd口令文件。

#vi /etc/smbpasswd #删除不需要的帐户

(3). 利用smbpasswd命令建立帐户的口令, 最好与/etc/passwd的帐户口令相同。

#smbpasswd user #口令为空时亦要运行此命令, 最多不加口令就是了。

(4). 重新启动SMB服务器

#/etc/rc.d/init.d/smb restart

(5). 配置/etc/hosts文件, 增加以下行

192.168.1.1 linux #linux主机

二、Samba服务器配置为共享模式[share]:

share模式的安全性能最低。我想, 只要不是为了连入Internet和共享重要的数据, 应该够用了。

[global]配置:

在[user]配置的基础上, 只需用; 注释掉以下各行

```
; guest account=user
```

```
; encrypt passwords=yes
```

```
; smb passwd file=/etc/smbpasswd
```

再把security=user改为security=share即可。

三、配置Win 95/98客户机

1. 配置c:\windows\hosts文件, 增加

192.168.1.1 linux

2. 配置网络

确定已安装了NetBEUI、TCP/IP网络协议, Microsoft网络客户。

如果在Linux Samba服务器上还配置了DNS服务的话, 必须指定DNS服务器的ip地址。

3. 开机时以user作为用户名登录网络, 鼠标右键点击“网上邻居”, 选“映射网络驱动器”, 选好驱动器符, 然后输入\\linux\user, 确定, 再输入口令(如果口令不为空的话), 确定即可。亦可输入\\linux\tmp(任何可用帐号都能共享)。如果映射成功, 先前网上邻居看不到的SambaServer服务器这时应能看得到了。否则, 从服务器上运行smbclient -L linux, 再刷新网上邻居即可看到。

说明: 此时, 最好不要双击网上邻居里的Samba服务器, 因为会出错。

我就是因为这样, 以为Samba还没有配置好, 不断地修改[global], 不断地调试而浪费了很多时间。为什么会这样呢? 我不知道, 大家最好去问微软。-:)

4. Samba设置成share模式时，Win95/98开机时用任何用户名（不需要用Linux Samba服务器生成的有效帐号）登录都可以映射成\\linux\\tmp。如果想映射成\\linux\\user，则需要修改Win 95/98注册表。在HKEY_LOCAL_MACHINE\\System\\CurrentControlSet\\Services\\VxD\\VNETSUP中增加一DWORD值：值名：EnablePlainTextPassword 数值：0x01。

说明：Win95的早期版本不用修改注册表。

检测

1. 用ping命令来检查网络是否连通

Ping linux #从客户端

Ping ht2/ht3 #从服务器端

2. 用smbclient查看共享资源

#smbclient -L linux #查看Samba服务器上的共享资源

屏幕上显示如下：

Added interface ip=132.102.65.101 bcast=132.102.255.255 nmask=255.255.0.0

Password:

Domain=[SMBASERVER] Os=[Unix] Server=[Samba 2.0.3]

Sharename Type Comment

tmp Disk Temporary file space

IPC\$ IPC IPC Service (Samba Server)

Server Comment

LINUX Samba Server

Workgroup Master

SMBASERVER LINUX

其实，如果Win 95/98客户机上设置了共享资源的话，可以用

#smbclient -L ht2/ht3 来查看Win95/98客户机上的共享资源。

用 #smbclient //ht2/共享名（如磁盘d）来共享Win95/98客户机上的共享资源。

练习题（1）配置samba能够和win98通信

要求：a 新建三个组manager、sales、tech；六个用户user1user2user3、user4、user5、user6；让user1,user2属于manager组user3,user4属sales组user5,user6 属于tech组。

b 新建三个目录manager、sales、tech设为共享并且让user1、user2只能访主目录和manager其他用同理。

（2）进行域认证

要求：a 设本域为samba

b 分别以不同的身份登陆（六个用户）

参考步骤

第一题

第一步 新建六个用户（adduser user1、user2、user6）；新建三个组groupadd manager（或修改/etc/group文件）然后把这六个用户加到这三个组中：（修改/etc/passwd文件即把用户的gid改相应的编号）。。

第二步 在/home/dept下创建 manager、sales、tech三个目录（mkdir /home/dept/manager）然后更属主用命chown root.manager /home/dept/manager 其两个同理。

第三步 将它们权限修改为770 chmod 770 /home/dept/manager

第四步 添加samba用户smbpasswd -a user1 其它同理

第五步 修改主配置文件/etc/smb.conf(7.0版在/etc/samba/smb.conf)

```
workgroup=samba
```

```
server string=This is NT server
```

```
security=user
```

```
hosts allow=192.168.1. 192.168.0. 192.168.2.
```

```
max log size=50 (单位k)
```

```
smb passwd files=/etc/smbpasswd
```

```
;domain logons=yes
```

第六步 将新建的三个目录设共享在[homes]段加入下面几行

```
[manager]
```

```
comment=the share for manager
```

```
browseable=yes
```

```
writable=yes
```

```
guest ok=no
```

```
write list=@manager
```

(其它两个目录同上)

第七步 重新启动samba服务 /etc/rc.d/init.d/smb restart

第八步 在客户以不同身份登陆（如以user1用户则可以访问user1和manager两目而没有权限访问其它目录）

第二题

第一步 domain logons=yes 前的分号去掉

第二步 将[homes]段的[netlogon]共享目录前的分号去掉之后保存退出

第三步 在/home 下创建netlogon目录 mkdir /home/netlogon

第四步 重新启动samba服务

第五步 在客户端打开登陆到NT域然后以某一用户登陆到samba域

第十节课：apache的配置

一、apache的启动

Linux捆绑了著名的Apache（阿帕奇）万维网服务器。

手工启动/停止/重起Apache Web服务器后台守护进程httpd是通过在命令“/etc/rc.d/init.d/httpd”后加上参数“start/stop/restart”来进行的：

```
[root@linux /root]# /etc/rc.d/init.d/httpd start
```

```
Starting httpd: httpd: cannot determine local host name.
```

```
Use the ServerName directive to set it manually.
```

二、apache的配置

上面我们看到的是一般新用户在第一次启动Apache Web服务器时经常遇到的一个问题，就是没有向Apache Web服务器说明“ServerName（httpd服务器所服务的主机名）”。Apache Web服务器启动时必须知道自己所服务的主机名，否则是不行的，而系统自己是没办法猜测到用户的Web服务器所要服务的主机名的，所以用户必须主动的告诉Apache Web服务器它所要服务的主机名。这个小问题通过修改Apache Web服务器配置文件“/etc/httpd/conf/httpd.conf”中的“ServerName”一行就可以解决：

把：

```
#ServerName new.host.name
```

改为：

```
ServerName www.ccjsj.com
```

注意，顶头的“#”号被去掉了。然后我们再来启动Apache Web服务器进程httpd：

```
[root@linux /root]# /etc/rc.d/init.d/httpd start
```

Starting httpd: httpd

接下来，我们再用运行于命令行的浏览器lynx来确认一下：

```
[root@linux /root]# lynx www.ccjsj.com
```

完全正常，我们如所期望的那样看到了在lynx浏览器中显示出来的Linux系统上缺省的主页。

在RedHat Linux中，Apache的缺省安装路径是在“/home/httpd”下面。由于RedHatLinux是基于RPM软件包管理系统的，用户可以使用命令：

```
[root@xteam /root]# rpm -ql apache | less
```

来查看Apache的详细的安装路径所在。Apache Web服务器主要的配置文件是在“/etc/httpd/conf”目录下的：httpd.conf

```
...
```

```
ServerType standalone
```

```
Port 80
```

```
...
```

“ServerType”指令，说明的是Apache是以一个独立的（standalone）守护进程的身份运行于后台，还是由另外的后台守护进程（inetd）在接到HTTP请求的时候来唤醒。对于正式的互联网站点来说，standalone可以提供比inetd方式更快的客户端响应速度。“Port”指令说明了由Apache Web服务器所监听的端口号，端口80是由RFC所规定的标准的HTTP服务所驻留的端口，如果你有特殊的需要想让Apache监听其它的端口号，如8080等等，就可以在这里修改“Port”这一指令。

```
HostnameLookups off
```

```
...
```

接下来的这一指令“HostnameLookups”，指示Apache是记录浏览本网站的客户的IP地址（形如：202.102.13.141），还是记录DNS地址（形如：www.ccjsj.com）。一般来说，记录DNS地址更省空间。

```
User nobody
```

```
Group nobody
```

```
...
```

上面这两行指令指示Apache服务器的运行身份，以用户nobody.nobody的身份来运行，对于系统安全是最合适的。切忌以用户root的身份来运行Web服务器。

```
ServerAdmin zhang@www.ccjsj.com
```

```
ServerRoot /etc/httpd
```

```
...
```

“ServerAdmin”指令说明一个电子邮件地址。当Apache Web服务器在运行中遇到问题时，Apache会把这一电子邮件地址提供给客户浏览器，以方便客户浏览器提交错误报告或寻求解决问题的办法。“ServerRoot”指令指示了Apache的配置文件、错误记录文件以及Log文件等等在服务器上所在的位置。在RedHatLinux中，这些文件缺省的驻留于路径“/etc/httpd”。

```
ErrorLog logs/error_log
```

```
LogLevel warn
```

```
...
```

“ErrorLog”指明相对于路径“ServerRoot”来说，Apache的错误记录文件所在的位置。“LogLevel”为“warn”一般来说是比较合适的，其它的“LogLevel”从简到繁还有“debug，info，notice，warn，error，crit，alert，emerg”等等。

```
LogFormat "%h %l %u %t \"%r\" %>s %b" common
```

```
LogFormat "%{Referer}i -> %U" referer
```

```
LogFormat "%{User-agent}i" agent
CustomLog logs/access_log common
CustomLog logs/referer_log referer
CustomLog logs/agent_log agent
```

...

上面的指令“LogFormat”每一个都申明一种记录文件的格式，这些格式都很方便用一些程序脚本来处理，这就使得服务器管理员可以方便的分析网站各个方面的情况，获得远比一个计数器所提供的要多的多的信息。接下来呢，“CustomLog”指令指示某一种格式的记录文件的所在路径（相对于“ServerRoot”）。这里为了减小篇幅，我们就不再重复Apache手册里关于记录文件格式的说明了，由兴趣的朋友可以自己去查阅一下。相关的网址在“<http://www.apache.org>”

接下来的一个指令是：

```
#ServerName new.host.name
```

...

这一行在RedhatLinux的发行版中，缺省情况下是被注释掉了。用户可以在这里填写上自己的域名（主机名），以启动Apache Web服务器。这之后的“httpd.conf”配置文件中的指令一般是划归两个主题：性能调优以及虚拟主机的配置。由于这两个主题有其本身的逻辑结构，我们将在以后详述。

```
DocumentRoot /home/httpd/html
UserDir public_html
DirectoryIndex index.html index.shtml index.cgi
FancyIndexing on
```

...

“DocumentRoot”指令说明所有的那些HTML文件存放在服务器上的地点。在RedHatLinux中，存放这些HTML文件的缺省路径是在“/home/httpd/html”目录下。以后用户自己的Web主页文件等等也存放在“DocumentRoot”指令所指示的这个目录下面。

“UserDir”指令说明个人主页存放路径。个人主页通过URL：<http://www.ccjsj.com/~bearzhang>，即“...way”时，Apache Web服务器将调出“www.ccjsj.com”这台主机上的“/home/...[蹦柯枷碌奈募 ?/a>](#)

紧接着的问题就是，当客户端浏览器试图访问诸如“<http://www.ccjsj.com>”这样的URL时，...睦 又校城pache Web服务器将首先试图给出index.html，如果不存在这个文件，就接着寻找index.shtml，如果还是找不到，就找index.cgi，那么，如果还是找不到怎么办呢？

接下来的指令“FancyIndexing”如果被设置成为“on”，Apache Web服务器将把相对应于被要求的URL的目录下的所有文件的列表以HTML超链接的格式给出；如果“FancyIndexing”被设置成为“off”，那么，相应的，将禁止这种行为。

...

```
Alias /icons/ /home/httpd/icons/
ScriptAlias /cgi-bin/ /home/httpd/cgi-bin/
```

...

这里的两个指令也是有关于URL到文件系统上的路径的映射的，上面的例子，“<http://www.ccjsj.com/icons/>”被映射...旁谗飧睢肪断隆?[/a>](#)

```
<Directory />
Options None
AllowOverride None
</Directory>
```

...

上面的例子其实就是设置一下缺省的访问控制，可以看出这是一个很严格的访问控制，这对于系统安全是有益的。

上面我们把Apache Web服务器三个主要部分大致的浏览了一下，用户在修改了Apache Web服务器的配置文件以适应自己特定的需要以后，需要重新启动Apache，以使新的配置文件生效。在Linux中以用户root的身份执行下面的命令：

```
[root@linux conf]# /etc/rc.d/init.d/httpd restart
```

三、设置实现基于名字的虚拟主机服务

基于名字的虚拟主机服务，不需要更多的IP地址，而且配置简单，无须什么特殊的软硬件支持。现代的浏览器大都支持这种虚拟主机的实现方法。当然，这也就是指一些早期的客户端浏览器也许不支持这种虚拟主机的实现方法。

正是以上原因，我们没有理由不使用基于名字的虚拟主机服务而使用基于IP地址的虚拟主机服务。配置基于名字的虚拟主机服务需要修改配置文件：“/etc/httpd/conf/httpd.conf”，在这个配置文件中增加以下内容。

```
NameVirtualHost 192.168.1.1
```

```
<VirtualHost 192.168.1.1>
```

```
ServerAdmin webmaster@www.ccjsj.com
```

```
DocumentRoot /home/httpd/www.ccjsj.com
```

```
ServerName www.ccjsj.com
```

```
ErrorLog /var/log/httpd/www.ccjsj.com/error.log
```

```
</VirtualHost>
```

```
<VirtualHost 192.168.1.1>
```

```
ServerAdmin webmaster@www.ccjsj.com
```

```
DocumentRoot /home/httpd/www.shccjsj.com
```

```
ServerName www.shccjsj.com
```

```
ErrorLog /var/log/httpd/www.shccjsj.com/error.log
```

```
</VirtualHost>
```

也就是在基于IP地址的配置基础上增加一句：NameVirtualHost 192.168.1.1而已。在本例中，为了体现只需要增加一次，所以特别地设置了两个虚拟主机服务。

最后也是建立相应的目录，将主页内容放到相应的目录中去就可以了。

设置实现基于ip的虚拟主机服务(与上述方法相同)

第十一节课：sendmail的配置

• 简单配置：

一 邮件协议SMTP（Simple Mail Transfer Protocol）用于主机与主机之传送邮件（一次处理一封邮件）

POP3 协议(Post office Protocol)用于下载整批邮件（连接110端口）

二 邮件的工具MUA（Mail user Agents）即用户处理邮件的界面（常见的有pine,elm）

MTA (Mail Transport Agents) 负责邮件的传输工作（常见的有sendmail,qmail,smail）

三sendmail服务器的安装 rpm – ivh sendmail-8.9.3-15.i386

四 控制启动/etc/rc.d/init.d/sendmail restart(setup-system service-sendmail开机是启动)

五 相关知识

/var/spool/mail存放用户收信信箱文件

/var/spool/mqueue/l存放每个用户尚未寄出的邮件

~/dead.letter每个用户无法寄出的邮件

/var/log/maillog 邮件系统的记录文件

~/mbox 读完之后邮件的存放地

六 主配置文件sendmail.cf可以不做修改只需在/etc/mail/access文件中加入一行即可

1 192 . 168 . 1 . RELAY (表示允许接收由192 . 168 . 1而来的邮并转寄) 将/etc/inetd.conf文件中 #pop-3 stream tcp noswait root /usr/sbin/tcpd ipop3d一行前面的#号去掉。

2 windows用户端打开outlook—工具—帐号--添加--邮件输入显示姓名—输入自己的电子邮件地址

下一步如图所示-----输入自己的帐号和密码

做完这些windows端的用户就可收发电子邮件了

(如果不能发信则将服务器端的/etc/mail/access文件中加入允许中继本机的IP)

七 如何收发附件

步骤1 使用uuencode将一个8bit的程序以七 bit的格式存储uuencode 原文件名 新文件名 > 123

2 在提示符下输入mail 在邮件的正文中加入 ~r 123 即可

3 收信端读完之后到主目下用 uudecode将程序转换回8bit uudecode mbox

4如果以linux作为客户端则只需配置好pop3服务和sendmail同时将dns指向同一台服务器即可。

八 安全控制

在/etc/sendmail.cw

在/etc/mail/access (或relay-domains) 加入允许中继的主机 (可以控制那些主机可以发邮件)

九 涉及到的命令 mail 收发邮件 mailq 检查代寄邮件encode uudecode 收发附件

第十二节课：x-window的配置

一 安装Xfree86

方式1 安装rpm包 rpm -ivh Xfree86*

2 用安装盘启动选择安装的方式为upgrade(选择Xfree86套件)

二 XWINDOWS的设置

1 设置工具 (Xconfigurator,XF86Setup, xf86config,)

2 使用xf86config配置xwindows

xf86config (在提示符下执行xf86config)

step 1 设置鼠标 (命令执行之后第一个画面是一段说明文字可以直接按回车)

a.如果本机的鼠标是两键的可选1 microsoft compatible 三键则选2 mouse systems 如果是

ps/2的接口则选4 ps/2 mouse b. 选完回车之后会有一句 Do you want to enable Emulate 3 Buttons?问你是否要模拟三键。一般选择y 2 接下来鼠标装备文件名称即在 Mouse device : /dev/ttyS0 是com1 □ /dev/ttyS1 是com2 □ /dev/paux 是ps/2 □

step 2 设置键盘

a.设置完鼠标之后的画面 Do you want to use XKB?输入Y可以设置键盘 (XKB是一种协议)

b.press enter to continue, or ctrl-c to abort. (按回车键继续按ctrl+c忽略) 当然继续啦

c.在之后的键盘列表中选1 Standrd 101-key,US encoding (标准的101键盘)

step 3 设置屏幕

a. 设置屏幕水平扫频率

有说明书则按说名书设置选11 Enter your own horizontal sync range自行输入如果不知到, 14, 15寸可以选6 31.5-48.5 ;Non-Interlaced SVGA, 1024*768 @70hz

17寸可以选7, 8。

b. 垂直扫描频率有说明书则按说名书设置选 5 Enter your own vertical sync range自行输入如果不知到, 14, 15寸可以选2 50-90 17寸选3 50-100 (接下来询问名称型号可直接回车)

step 4 设置显卡

a. Do you want to look at the card database (是否查看支持的显卡资料)选Y回车

b. 找到自己的型号之后输入其前面的编号

c. Do not probe clocks or use any clocks line.Press enter to continue, or ctrl-c to abort.

询问是否侦测clock 可直接回车

d. 接下来的画面选择一个X server 我们一般选择 5 choose the server from the card definition, Xf86_s3

e. do you want me to set the symbolic link? (是否建立一个符号连接)回答Y

三 X的结构

客户机-服务器结构 linux的X是针对网络环境所做的所以有服务端的X-server与客户端的X-client 两者可以分开在网络中不同的主机上它们之间通过X protcol来沟通

1) X-server 提供的服务就是负责屏幕上的显示工作以及输入

2) X_client 对X-server 提出请求的X应用程序

也就是说可以利用远端的工作站执行X应用程序并使用其处理器资源而其执行情况可以在本地的

X-server上显示

3) 具体方法 (前提条件有两台主机其中一台IP为192. 168. 1. 10另一台为192. 168. 1. 11现在将192. 168. 1. 10作为服务端而将192. 168. 1. 11作为客户端)

step 1 xhost + 192.168.1.11 (允许192. 168. 1. 11使用本机作为X服务端)

step 2 telnet 192.168.1.11 (远端登陆这台客户机)

step 3 export DISPLAY=192.168.1.10:0.0 (指定X程序执行后显示的地方即服务端)

step 4 执行一个X程序如: xclock或xterm 执行的结果将在X服务端显示。

四 小技巧: 1 在图形界面下直接更改分辨率 ctrl + alt + +(-)

1 强行退出图形界面 ctrl + alt + backspace (F1-F6)

2 更改gnome桌面为 kde 桌面 例如更root的桌面就在其主目录下编辑文件 .xinitrc 加入一句话exec startkde即可然后重新开启windows (startx)

五 练习题 使用xf86config 配置自己的X-window (步骤参考X-window设置一节)

第三部分：优化与安全

第十三节：内核的设置与编译

• 内核简介

内核：是一个操作系统的核心，它负责管理系统的进程、内存、设备驱动程序、文件和网络系统，决定着系统的性能和稳定性。就好比DOS下的IO.SYS和MSDOS.SYS一体，我们可以把这两个文件叫做DOS的核心。Linux也有它的核心，通常在根目录下，一个叫vmlinuz的文件。我们用这个文件来控制我们的整台PC，包括周边设备和软硬磁盘机、CD-ROM、声卡等。简单地说，核心就是操作系统本身。没有了它，就像一个无人住的家，没有人去维持这个家的动作。一部电脑需要很有效率已稳定的核心，也就是操作系统。因此，核心是整个系统维持下去的关键。

Linux的一个重要的特点就是其源代码的公开性，所有的内核源程序都可以在/usr/src/linux下找到，大部分应用软件也都是遵循GPL而设计的，你都可以获取相应的源程序代码。任何一个软件工程师都可以将自己认为优秀的代码加入到其中，由此引发的一个明显的好处就是Linux修补漏洞的快速以及对最新软件技术的利用。

先介绍一下编译核心的选项，希望能对大家消除对内核的神秘感有所帮助。

- 1.代码成熟等级
- 2.处理器类型和特色
- 3.对模块的支持
- 4.基本的选择
- 5.即插即用支持
- 6.块设备支持
- 7.网络选项
- 8.电话支持
- 9.SCSI设备的支持
- 10.I2O接口适配器
- 11.网络设备支持
- 12.配置业余无线广播
- 13.红外支持
- 14.ISDN的文件系统
- 15.旧型的光驱类型(非IDE界面的光驱)
- 16.字符设备
- 17.USB支持
- 18.文件系统
- 19.控制台驱动
- 20.声卡驱动
- 21.Kernel hacking

- 内核版本

由于Linux的源程序是完全公开的，任何人只要遵循GPL，就可以对内核加以修改并发布给他人使用。Linux的开发采用的是集市模型（bazaar，与cathedral--教堂模型--对应），为了确保这些无序的开发过程能够有序地进行，Linux采用了双树系统。一个树是稳定树（stable tree），另一个树是非稳定树（unstable tree）或者开发树（development tree）。一些新特性、实验性改进等都将首先在开发树中进行。如果在开发树中所做的改进也可以应用于稳定树，那么在开发树中经过测试以后，在稳定树中将进行相同的改进。一旦开发树经过了足够的发展，开发树就会成为新的稳定树。开发数就体现在源程序的版本号中；源程序版本号的形式为x.y.z：对于稳定树来说，y是偶数；对于开发树来说，y比相应的稳定树大一（因此，是奇数）。确定是以 root 的身份签入，然后cd到/usr/src。uname -r这个指令将会显示版本。内核版本的更新可以访问<<http://www.kernel.org/>>。

- 编译原因

Linux作为一个自由软件，在广大爱好者的支持下，内核版本不断更新。新的内核修订了旧内核的bug，并增加了许多新的特性。如果用户想要使用这些新特性，或想根据自己的系统度身定制一个更高效，更稳定的内核，就需要重新编译内核。

通常，更新的内核会支持更多的硬件，具备更好的进程管理能力，运行速度更快、更稳定，并且一般会修复老版本中发现的许多漏洞等，经常性地选择升级更新的系统内核是Linux使用者的必要操作内容。

为了正确的合理地设置内核编译配置选项，从而只编译系统需要的功能的代码，一般主要有下面四个考虑：

- 自己定制编译的内核运行更快（具有更少的代码）
- 系统将拥有更多的内存（内核部分将不会被交换到虚拟内存中）
- 不需要的功能编译进入内核可能会增加被系统攻击者利用的漏洞
- 将某种功能编译为模块方式会比编译到内核内的方式速度要慢一些

以上是针对成熟的Linux套件如Redhat Linux而言,我的目的是为建造嵌入式Linux操作系统做准备，也是必由之路。

一、编译内核的步骤：

1、准备工作

- 新版本内核的获取和更新

Linux内核版本发布的官方网站是<<http://www.kernel.org/>>，国内各大ftp上一般都可以找到某些版本的内核。新版本的内核的发布有两种形式，一种是完整的内核版本，另外一种patch文件，即补丁。完整的内核版本比较大，比如linux-2.4.0-test8.tar.bz2就有18M之多。完整内核版本一般是.tar.gz（.tgz）文件或者是.bz2文件，二者分别是使用gzip或者bzip2进行压缩的文件，使用时需要解压缩。patch文件则比较小，一般只有几十K到几百K，极少的会超过1M。但是patch文件是针对于特定的版本的，需要找到自己对应的版本才能使用。

编译内核需要root权限。把需要升级的内核拷贝到/usr/src/下（下面以2.2.16的内核的linux-2.2.16tar.gz为例），命令为

```
#cp linux-2.2.16tar.gz /usr/src
```

先查看当前/usr/src的内容，注意到有一个linux的符号链接，它指向一个类似于linux-2.2.14（对应于现在使用的内核版本号）的目录。首先删除这个链接：

```
#cd /usr/src
```

```
#rm -f linux
```

现在解压下载的源程序文件。如果所下载的是.tar.gz（.tgz）文件，使用命令：

```
#tar -xvzf linux-2.2.16tar.gz （解压到 /usr/src）
```

如果下载的是.bz2文件，例如linux-2.2.16tar.bz2，使用命令

```
#bzip2 -d linux-2.2.16tar.bz2
```

```
#tar -xvf linux-2.2.16tar
```

现在再来看一下/usr/src下的内容，发现现在有了一个名为linux的目录，里面就是需要升级到的版本的内核的源程序。还记得那个名为linux的链接么？之所以使用那个链接就是防止在升级内核的时候会不慎把原来版本内核的源程序给覆盖掉了。现在也需要同样处理：

```
#mv linux linux-2.2.16 （重命名）
```

```
#ln -s linux-2.2.16 linux （建立链接）
```

如果还下载了patch文件，比如patch-2.2.16，就可以进行patch操作（下面假设patch-2.2.16已经位于/usr/src目录下了，否则需要先把该文件拷贝到/usr/src下）：

```
#patch -p0 < patch-2.2.16
```

- 准备主机板和相关硬件的说明手册

其实也不用太详细，只要知道您的硬件是属于哪一类型就行了。例如：有一张SCSI卡，那就要知道这张卡的名字，有一台cd-rom,就要知道这台光驱是哪一种牌子的，是否为标准的IDE/ATAPI界面，还是另有专属接口卡呢？或者，主机版是否有支持Triton芯片（通常586以上的电脑常有），这些信息能帮助我们，使得设定变得清楚且容易。

因此，不管您有什么使用手册，准备好吧。即使现在不用，将来还是会用到的（设X - window system时要显示卡的手册）。

- 检查声卡的IRQ设定和其种类

如果配有一张声卡，除了要知道卡的种类外（例如 Sound Blaster）还需要知道这张卡的IRQ地址。一般来说，声卡的IRQ地址是5或7而IO地址则为220。DMA则I，不过，有时不同的声卡可能会有不同的设定。因为稍后的选项里，就会要填入这些数字。

- 编译核心的硬件需求

在编译核心时，确定您的RAM最好在8MB以上，否则可能会很慢而且问题会很多，记得查看swap有没有打开（用free指令）。此外，最好不要超频，不然很有可能会发生signal 11的错误，使得编到一半的核心停了下来，其实编译核心就好比编译程序一样，只是因为构成核心的程序太多了，因此我们能小心尽量小心。

2、内核编译的流程

- 编译开始

通常要运行的第一个命令是：

```
#cd /usr/src/linux
```

```
#make mrproper
```

该命令确保源代码目录下没有不正确的目标.o文件以及文件的互相依赖。如使用刚下载的完整的源程序包进行编译，本步可以省略。而如果多次使用了这些源程序编译内核，那么最好要先运行一下这个命令。

确保/usr/include/目录下的asm、linux和scsi等链接是指向要升级的内核源代码的。这是配置非常重要的一部分。删除掉/usr/include下的asm、linux和scsi链接后，再创建新的链接指向新内核源代码目录下的同名的目录。这些头文件目录包含着保证内核在系统上正确编译所需要的重要的头文件。也是上面又在/usr/src下"多余"地创建了个名为linux的链接的原因之一，按照下面的步骤进行：

```
# cd /usr/include
```

```
# rm -r asm linux scsi
```

```
# ln -s /usr/src/linux/include/asm-i386 asm
```

```
# ln -s /usr/src/linux/include/linux linux
```

```
# ln -s /usr/src/linux/include/scsi scsi
```

一旦万事俱备，转到/usr/src/linux。

第二节 配置内核 核心内容

配置内核可以根据需要与爱好使用下面命令中的一个：

```
#make config（基于文本的最为传统的配置界面，不推荐使用）
```

```
#make menuconfig（基于文本选单的配置界面，字符终端下推荐使用，必须安装ncurses-dev和tk4-dev库）
```

```
#make xconfig（基于图形窗口模式的配置界面，Xwindow下推荐使用）
```

```
#make oldconfig（如果只想在原来内核配置的基础上修改一些小地方，会省去不少麻烦）
```

如果不能使用Xwindow，那么就使用make menuconfig好了。界面虽然比上面一个差

点，总比make config的要好多了。

选择相应的配置时，有三种选择，它们分别代表的含义如下：

Y - -将该功能编译进内核

N - -不将该功能编译进内核

M - -将该功能编译成可以在需要时动态插入到内核中的模块

在每一个选项前都有个括号,但有的是中括号有的是尖括号，还有一种圆括号。用空格键选择时可以发现，中括号里要么是空，要么是"*"，而尖括号里可以是空，"*"和"M"。这表示前者对应的项要么不要，要么编译到内核里；后者则多一样选择，可以编译成模块。而圆括号的内容是要在所提供的几个选项中选择一项。

在编译内核的过程中，最烦杂的事情就是这步配置工作了，不清楚到底该如何选取这些选项。实际上在配置时，大部分选项可以使用其缺省值，只有小部分需要根据用户不同的需要选择。选择的原理是将与内核其它部分关系较远且不经常使用的部分功能代码编译成为可加载模块，有利于减小内核的长度，减小内核消耗的内存，简化该功能相应的环境改变时对内核的影响；不需要的功能就不要选；与内核关心紧密而且经常使用的部分功能代码直接编译到内核中。下面对选项分别加以介绍。

- 内核设置

1. Code maturity level options代码成熟等级

此处只有一项：prompt for development and/or incomplete code/drivers，如果要试验现在仍处于实验阶段的功能，比如khttpd、IPv6等，就必须把该项选择为Y了；否则可以把它选择为N。在Linux的世界里，每天都有许多人为它发展支持的driver和加强它的核心。但是有些driver还没进入稳定的阶段。但其作者很欢迎其他人去测试这些driver并提出一些bugs。这个问题是说，有一些drive还在做测试中，问您是否要选择这些drive或支持的程序码。

如果键入Y，往后将会出现一些还在测试中的东西给您做选择。（像Java的程序码和PCI bridge），否则就键入N。

2. Processor type and features处理器类型和特色

#Processor family (386, 486/Cx486, 586/K5/5x86/6x86, Pentium/K6/TSC,PPro/6x86MX) [PPro/6x86MX] -----选择处理器类型，缺省为Ppro/6x86MX。它会对每种CPU做最佳化，让它跑得快又好。一般来说，没有选择正确的CPU并不会有重大的影响（特别是选择386，这样编译出来的核心也许会比较小但它的速度可能会变慢了）。所以，最好要知道您的CPU是哪一种。不过，如果您的gCC编译器是2.7.0版以前的。那么只能选择386或是486。

#High Memory Support-----内核支持的最大内存数，缺省为1G。可以支持到4G、6.4G，一般可以不选。

#Math emulation-----这项询问是否需Linux核心模拟数学浮点运算器。如果有486Dx、AMD以及Pentium机器的话，这个选项就不必选了，因为它们都有内建的浮点运算器。协处理器是在386时代的宠儿，现在早已不用了。不过，对于有内建浮点运算器的人来说，选了这个选项并不会因此让内建的浮点运算器失效。但它会增大核心约45KB。

#MTTR (memory type range register) support-----选择该选项，系统将生成/proc/mtrr文件对MTRR进行管理，供X server使用。同时用来启动pentium pro和pentium II 的特殊功能，如果你用的不是这类CPU就选N，否则也仅仅是使内核变大而已。

#Symmetric multi-processing support-----对称多处理支持。除非有多个CPU，否则就不用选了。

3. Loadable module support对模块的支持.

首先，了解一点关于模块的知识。模块就像你特意插入核心中的某些东西，如果办公室有一个小网络并且有时想用一下(但并不经常)，也许你想把网卡编译成一个模块。使用这个模块，机器必运行和存取/lib下的模块，意思是驱动程序(IDE,SCSI等但必须是NFS支持的网卡)，文件系统(通常

是ext2但也可以是nfs)和核心类型(最好是elf)必须编译在内核并且不能是模块,模块只有核心引导时才起作用,驱动程序(来网络)的存取,和文件系统安装。这些文件必须编译在核心内否则将不能安装启动分区。如果安装启动分区和网络,你需要网络系统文件,和已经编译的网卡。为什么要使用模块?模块化使核心变的更简捷,它减少核心释放大量的受保护的空間。模块的安装和卸载使用的空間是可重复分配利用的。如果你打开机器有90%以上的時間用到一个模块,编译它。运用这类模块是浪费内存的,原因是一旦你编译了模块它们同样将占用大量的内存,核心需要一些代码来挂上模块。记住,核心在保护空間运行,但模块并不是。这么说,并不经常使用我的设备,把它编译成只支持ext2,ide和elf。而一直使用

是在这或那用上那么几分钟。严格的说,这样做会使核心增大许多而降低它的执行速度。这时我们就可以把这些可能会用的驱动程序编译成一个一个的模块,在需要用的时候才用insmod这个指令加入核心,不用的时候也能rmmod把它从核心移除,或是用lsmod察看目前所载入的模块。这里面有三項:

#Enable loadable module support-----除非准备把所有需要的內容都编译到内核里面,否则该项应该是必选的。

#Set version information on all module symbols-----通常,我们更新核心版本之后,模块要重新编译。这个选项使您不必更新编译模块而能使用以前的模块。可以不选它。但如果您选y,则按照它的说明,您必须有gensyms这个程序(可用whereis指令查看有无此程序)。

#Kernel module loader-----让内核在启动时有自己装入必需模块的能力,建议选上。

注意:在开机就会 mount 上来的 partition 的 FS、device driver 记得要 compiler 进 kernel,不能把它弄成 modules。请不要夸张到为了完全模块化而忘了把ext2fs和IDE driver compiler 进 kernel 里。

4. General setup普通的属性设置

这部分內容非常多,一般使用缺省设置就可以了。下面介绍一下经常使用的一些选项:

#Networking support-----网络支持。因为在Linux里面,有虚拟的网络设备(loopback),可以模拟整个网络。而且,一些程序需要它。必须,没有网卡也建议你选上。注意:选N,则7.(Networking options 网络选项)和11.(Network device support网络设备支持) 不会出现。

#Limit memory to low 16MB -----大部分的人这一选项 N。除了主机板没有办法处理16MB以上的内存,或者有超过16MB以上的内存但却常常发生一些很奇怪的问题。这时,您可以试试这个选项。有些主机板对超16MB内存的处理并不是很好,通常这些都是旧型的主机板。还有,在说明文件中有提到,如果内存超过64MB的话,用LILO加一些参数给Linux核心(例:mem=80M),并且把您主机板上的Cache加到512K。这样,整体效率才能提升。

#PCI support -----PCI支持。如果使用了PCI的卡,当然必选。

#PCI bios support -----主机板是否有PCI界面。如果有,则您必须回答y。

PCI是586电脑的主要界面(一些486主机板上也有),这个界面能让您插入所谓的PCI显示卡,或是PCI的网络卡等。这种界面是现在电脑的主要趋势,因此如果有PCI的插槽。您就可以选Y。除了一些很旧很旧但有支持PCI的主机板外(这些有bugs的旧型主机板可能会因为这个选项而让核心挂掉)。

#PCI access mode (BIOS, Direct, Any) [Any] -----设置Linux探测PCI设备的方式。选择“BIOS”,Linux将使用BIOS;选择“Direct”,Linux将不通过BIOS;选择“Any”,Linux将直接探测PCI设备,如果失败,再使用BIOS。

#Support for hot-pluggabel devices -----热插拔设备支持。支持的不是太好,可不选。

#PCMCIA/CardBus support-----PCMCIA/CardBus支持。有PCMCIA就必选了。

#PCI bridge optimization (experimental) -----在某些支持BIOS上，它能让存取速度加快，建议是选Y。

#Backward-compatible /proc/pci-----设备兼容，自己看help。

#System V IPC 如果将来想编译dosexmu（DOS模拟器），则这个选项一定要选，它是一个让各个程序（process）同步且能彼此交换数据的函数库和一些系统的调用，没它，很多的程序将会无法执行。

#BSD Process Accounting-----

#Sysctl support-----除非你的内存少的可怜，否则你应该启动这个功能，启用该选项后内核会大8K，但能让你直接改变内核的参数而不必重新开机。

#Kernel support for A.OUT binaries -----a.out的执行文件是比较古老的可执行码，用在比较早期的UNIX系统上。Linux最初也是使用这种码来执行程序，一直到ELF格式的可执行码出来后，有愈来愈多的程序码随着ELF格式的优点而变成了ELF的可执码。将来势必完全取代a.out格式的可执行码。但目前由于沿有许多程序还没有取代过来，所以只好选择Y，等将来有一天，全部的程序都变成了ELF的天下时，那时再disable掉。

#Kernel support for Linux/Intel ELF binaries -----由上所述，这个当然y哩，因为目前gcc - 2.7.0以上的都有支持ELF了，如果没有选择这一项，可能会使用相当多的程序因此无法执行。

注意：编译模块成ELF和编译支持ELF二进制。不编译适当的支持 ' ' gotcha ' ' 是明智的，如果机器结构是Pentium或486你将得到高效的代码，但一个386的核心将运行在32-bit compatible clone；一个Pentium核心将不。为大多机器制作一张紧急启动盘，最好在386下编译，而386并不能运行在Pentium下编译的核心。

另外一点要注意的，你不能同时把 a.out 和 ELF 支援编译成 modules，一般把a.out编译为modules，把ELF编译为核心。

#Kernel support for JAVA binaries ----- 这一项是正在做测试中的产品，但是如果写有关Java的程序，希望它能在Linux的机器上跑。那么，可以选择把它编成一个模块或是直接把它编进核心里。

#Power Management support -----电源管理支持。

##Advanced Power Management BIOS support-----高级电源管理BIOS支持。这通常是用在笔记本电脑上的东西，如果您有APM的BIOS，支持省电的设备的（有电池的那种），那么您可以选上这项，一般人这一项是选n，以避免一些可能会发生的问题。后有8个选项。

#Parallel port support -----串口支持。

5. Plug and Play configuration即插即用支持

Linux对即插即用目前支持的不如Windows，好有些情况下会和其他设备产生冲突（I/O，DMA，IRQ等）。这个选项对PCI设备没有影响，因为他们天生就是PNP设备。。

#Plug and Play support (CONFIG_PNP) ----- 选择“y”，内核将自动配置即插即用设备。原来PNP还有这个意思。

ISA Plug and Play support ----- 选择“y”，内核将自动配置基于ISA总线的即插即用设备。

6. Block devices块设备支持

这个就得针对自己的设备情况来选了：

#Normal PC floppy disk support (CONFIG_BLK_DEV_FD) [Y/m/n/?]-----普通PC软盘支持。

#Enhanced IDE/MFM/RLL disk/cdrom/tape/floppy support -----选择“y”，内核将提供对增强IDE硬盘、CDROM和磁带机的支持。在硬盘没有做得很大的时候，一般的IDE卡和BIOS只能支持小于540MB的硬盘。不但如此，那时也只能支持二颗硬盘。但现今的硬盘动不动就是1GB以上，今年主流是30~50G，而且常常都会超过一二颗硬盘。如此一来，新的主机板就开始支持加强型的IDE界面（Enhanced IDE），以支持到540MB以上的硬盘。所以，如果您的IDE界面是Enhanced的，请您选Y，底下就会出现八部分IDE界面的选项，这些选项能加快您的IDE界面的速度和对某些芯片做一些最佳化。但如果您的硬盘或光盘全都是SCSI界面的，那么选N以跳过下面选项。

#use old disk-only driver on primary interface-----通常是选择N。因为我们有其他新的drivers可用。这个选项的意思是说，如果您的IDE界面是很旧很旧的那种的。那么，就可以使用这个drives 驱动那个旧型的IDE界面（可装二台硬盘；或是一台硬盘，一台光盘）。而现今流行的 Enhanced IDE则有两个界面，共可以接四台硬盘。我们稍后会有 driver支持它。

#Include IDE / ATAPI CDROM support-----如果希望核心支持IDE / ATAPI界面的光驱，选择Y。如果有光驱，但它附有一张接口卡，必须把排线接到那张专属接口卡上；或者是接到声卡上的，则这个选项也需要选N，稍后我们会有非IDE的光驱厂牌让我们挑选。现今的光驱通常是IDE / ATAPI界面的，所以这个选项通常是Y。

#Support removable IDE interfaces (PCMCIA) -----这个选项对大部分的人全选n，除非您有PCMCIA的东西，这通常是笔记本电脑上看得到的东西。PCMCIA是一个组织，在以前是设计内存条的。但现在他们对于PC CARDS定了一个标排，并很广泛的应用在1aptap的电脑上。不但有所谓的PCMCIA的硬盘，甚至有网络卡、SCSI卡等，不过，大部分的人并不需要这个选项。

这个选项选完后，以下则是Linux核心对几种芯片的IDE界面做修正或是加强它。

#CMD640 chipset bugfix / support----- 很多 486和 586的主机板都是用 CMD640的芯片，它是Neptune芯片和 SIS芯片的结合。不过，这种芯片有它的缺点，在许多的情形下，它会造成数据的流失和错误。如果您选了这一项，则Linux核心会为您小心的寻找这些错误并修正它。而且，它会打开对二个IDE界面的支持。不过，在它的说明文件中提到，如果您的主机板没有PCI界面只有VESA总线界面却希望有这项功能的话，则您必须传一些参数给核心（ideo = cmd640__v1b）。如果不确定上面所说的，选择Y。

#CMD640 enhanced support-----一般来说，对于硬盘的存取速度来说，有所谓的PIO MODES值设定，现今的IDE界面及BIOS应该都能侦测到正确的硬盘PIO MODE值了。此值愈高表示硬盘的存取的速度愈快。可是，有些主机板的BIOS还是旧式的，不能抓到比较高的PIO MODE值，如此一来，便不能发挥整台硬盘的效率。这个设定告诉读者说：如果您的IDE界面是CMD640为基础的界面，但是您的BIOS并不能抓到正确的PIO MODE值，那么，这个选项可以自动的找到硬盘正确的PIO MODE值。

#RZ1000 chipset bugfix / support-----这个选项如同前面的CMD640一样。不过，它的芯片是RZ1000的芯片，这种芯片是以Neptune芯片为主的一种芯片，而目、有很多的486和 586的主机板都在使用它。可以查查主机板的说明书或是 IDE接口卡的说明书做确定。文件上提到，选择这个将会降低一些速度，但是数据能百分之百的正确。

#Intel 82371 PIIX (Triton I / II) DMA support-----对于586的Pentium电脑来说，有相当多的主机板都是用Intel的Triton芯片，使用这种芯片的最大好处是支持直接内存存取DMA，而节省您的CPU时间。在以往还没有DMA这个东西时，读取硬盘需要耗用许多的CPU时间。如此一来，CPU被占用，就不能充分的发挥它的功效。后来，DMA出来后，硬盘的读取便靠Triton或其他有支持DMA的IDE界面的芯片，直接与它们做沟通，而节省了大量的CPU时间，但这必须您的硬盘和主机板有同时支持DMA的IDE界面的芯片，直接与它们做沟通而节省了大量的CPU时间。但这必须您的硬盘和主机板有同时支持DMA MODE才行。

##other IDE chipset support-----

如果这上选项选y，则会出现下列六种其他的芯片或厂牌供您选择。

* NOte : most of these also require special kernel boot paraaeters

ALI M14xx support

DTC - 2278 support

Holtek HT65608 support

PROMISE DC4030 support

QDI QD6580 support

UMC 8672 support

上面这六种厂牌的芯片依硬件配备而使用，但它们有共同的特点就是必须传一些参数给核心。如果找不到您的芯片，那么也没关系。上这些选项只不过对这些芯片做最佳化罢了。

Additional Block Devices其他的块设备

#Loopback device support-----大部分的人这一个选项都选N，因为没有必要。这个选项的意思是说，可以将一个文件挂成一个文件系统。如果要烧光盘片的，那么您很有可能在把一个文件烧进去之前，看看这个文件是否符合ISO9660的文件系统的内容，是否符合您的需求。而且，可以对这个文件系统加以保护。不过，如果您 想做到这点的话，您必须有最新的mount程序，版本是在2.5X版以上的。而且如果您希望对这个文件系统加上保护，则您必须有des.1.tar.gz这个程序。注意：此处与网络无关。

#MultInle devices driver support-----这个选项可以让把整个硬盘分区变成一个单独的区块设备，您必须有md035 . tgz这个程序。而且在做这件事之前请将您的硬盘备份，因为它尚在测试阶段。一般人对这个选项是选N。

#RAM disk support-----如果使用过DOS下的ramdrive程序，应该能了解这个选项的意义。它可以把内存当成硬盘来做存取就如同一般的硬盘一样，可以format它，或是放一些文件在里头。然后，当您关机这些数据也随着之而去了。如果的RAM够大，可以考虑玩玩这选项，但一般人都不需要。

#XT hard disk support-----支持XT的古董硬盘，这是IBM电脑时代的东西，如果您还有这种很旧很旧的硬盘。那么，您可以把它编进核心或是编成一个模块。大部分的人这个选项都是选择N的。

#Compaq SMART2 support-----

#Mulex DAC960/DAC1100 PCI RAID Controller support-----RAID镜像用的。

#Logical volume manager (LVM) support-----逻辑卷管理支持。

#Multiple devices driver support-----多设备驱动支持。

#RAM disk support-----RAM盘支持。

7. Networking options 网络选项

这里配置的是网络协议。

#Packet socket ----- 选择“ Y ”，一些应用程序将使用Packet协议直接同网络设备通讯，而不通过内核中的其它中介协议。

#Kernel / User Network link driver-----这个是在测试中的程序码，一般人不需要用。依它的说明，它允许在核心、模块或程序间的某些部分间，彼此做双向的沟通。如果想使用arpd，则这个程序码就要加进心里。

#TCP/IP networking-----选择“ Y ”，内核将支持TCP/IP协议。这个选项无论如何请您选择Y，即使没有网络卡，或是没有连到网络上的设备，在linux上仍有所谓的lookback设备而且有些程序需要这个选项。在说明文件中提到，如果您没有打开这个设定，则X - window system可能会有问题

(回为它也需要 TCP / IP)。

#Network firewalls-----选择“ Y ”，内核将支持防火墙。Firewalls依英文看是防火墙。在网络愈来愈发达的今天，网络安全的考虑也愈来愈重要了。在局域网上找一台电脑来保护自己的考虑也愈来愈多了。可以在一局域网上找一台电脑来保护自己区域内的电脑。这样的结果是，所有外部的电脑如果要连进内部的电脑就必须通过这台装有Firewalls电脑的同意。所以，如果您希望这台电脑有着过滤网络的功能的话，那么这个选项要选 Y。而且，等一下有个IP firewalling的选项也要选y；但下面的forwarding / gatewaying要选n，如此才能让它正常动作。大部分的人这个选项选N。

#Network alasing-----允许有多个IP地址。

#IP : forwarding / gatewaying ----- 和 Firewall相反，这个选项是用来疏导网络的。一个gateway（也就是router），要帮忙疏导两个网络间的数据传送。这台机器必须要有两张网络卡，连接两个个同的网络，做疏导网络的工作。如果选择了这个选项，则表示想让这台负责做router。那么，就必须有两张网络卡了。另外有一种情形是，如果您有MODEM（通过串行界面以 SLIP和 PPP协议）和网络卡，并用它们来连上Internet。这时您也一样可以执行IP - routing服务，也需把这个选项打开。

#IP : multicasting-----所谓的multicasting是群组广播，它是用在视频会议上的协议，如果想送一个网络封包（网络的数据），同样的一份数据将送往十部机器上。您可以连续送十次给十台机器（点对点的传送），也可以同时送一次，然后让十台机器同时接收到。当然后者比前者好，由于视频会议要求是最好每个人都能同时收到同一份信息，所以如果您有类似的需要，这个选项就要打开。同时您还必须去找相关的软件。

#IP : accounting-----如果您打开这个选项，您就可以在 / proc / net下看到系统对于整个网络状况的纪录。所以一般的人这个选项都是选y。而且，如果您设计把这台Linux机器当router用，那么读者可以因为这个选项而获得许多有关于网络IP控制的信息和它的输送情形。不过，您必须在底下的选择中选择proc系统（其实proc文件系统一定要选，不选很多程序会不能用！）

#IP : aliasing support-----也许您只有一张网络卡，但经由这个设定，您可以拥有数个IP地址。假设您已经有一个IP地址了，您还想再加入其他的IP地址，这时，您可以依下面的程序来做。

在 shell下键入：

```
sunlly : / #ifconfig eth0 : 0其他的IP地址（这个IP地址不能与其他机器重复）
```

```
sunlly : / # route - add - host其他的IP地址 dev eth0 : 0
```

如此您就可以同时拥有两个IP地址了。当然，如果想把这个IP地址去除，那么可以键入：

```
sunslyy : / # ifconfig eth0 : 0 - IP地址
```

这样您就可以把加入的IP地址去除。不过在使用此选项前，前面的那一个aliasing Network选项也要选上去。

#IP : PC / TCP compatibility mode-----大部分人都选n。除非在使用DOS下的NCSA - TCP / IP软件连进Linux机器时遇到了困难，或者有不相容的情形出现。这时，您可以试着把这个选项打开，看看是否能解决这个问题。

#IP Reverse ARP-----如果您的Linux希望提供bootd的服务，就是让没有硬盘或软盘也能够开机并且上网络，只要它们有网络卡有连接到网络的话。此时，您必须执行一个指令叫rarp来设定哪些电脑的网络卡可以如此。不过一般人没有这个需要，所以答n。

#IP : Disable Path MTU Discovery（ normall yenable ）----- 大部分的人这个选项是选择N。除非发现用DOS下的ncsa的telnet程序连到Linux机器上出问题。这是很多人的问题，如果发生了DOS下的telnet程序不能连进Linux时，除了可以改用Nsysutel的telnet程序来解决外，还可以在编译核心的时候，把这一项选 Y。MTU（Maximal Transfer Unit）叫做最大的传输单位，是说我们一次送往网络的信息大小。而Path MTUD iscovery的意思是，当Linux发现一些机器的传输量比较小时，我们会分送网络信息给它。如此可以增加网络的速度，所以我们大部分都选N，也就是Enable。

#Ip : Dorp source routed frames-----通常我们一个网络的封包在丢出去后就不管它了，不过，在TCP / IP协议里，您可以设定让那些帮您绕路的机器回送一个是否这个封包已经送达了的消息。不过，这会导至网络安全上的问题，所以很少用，一般来说我们选择Y。

#IP : Allow large windows (not recommended if < 16Mb of memory =) -----如果有超过16MB以上的内存，那么建议打开这个选项，可以增加传输的速度。在一般长距离的网络传输下要预备传输的数据可以先储存在缓冲区，等到对方的回应时再一次会过去。因此，您必须有内存来作为缓冲区。

#The IPX Protocol-----IPX是一种传输协议，它是Novell的一种网络协议，通常用在区域或是Windows的网络下。如果您希望Novell的机器资源共享（例如用他们的打印机或是硬盘），那么这个选项则要选y。至于存取文件的格式是NCPFS的格式，稍后把这个文件系统选上来，以便支援这个文件系统。如此一来，您就可以通过Novell的IPX通讯协议去存取它们的数据了。或者，您希望从dosexmu（DOS的模拟器）里用IPX协议，这时也要把它选进来。

#Full internal IPX network-----提供了一个完整的内部IPX网络，预设选项是N，因为它可能会让一些应用的服务程序（RIP / SAP）当掉。

#Appletalk DDP-----AppleTalk是存在于苹果机上的一种通讯协议，用来苹果电脑之间的网络通讯，通过AppleTalk，彼此的电脑间可以打印和分享文件。如果您需要连上这样的网络，可以把这个选项打开，如此就能加入他们与这些电脑做沟通了，或者把这项编成一个模块亦可。

#Amateur Radio AX.25 Level2-----

#Bridging (EXPERIMENTAL) -----

选这两个可以让Linux变成一个网络上的网桥，用来做不同网络间的沟通，通常一般人不需要。

#Qos and/or fair queueing（服务质量公平调度）也支持了，还有kHTTpd，不过这些都还在实验阶段。

8. Telephony Support电话支持

原来是Linux下可以支持电话卡，这样你就可以在IP上使用普通的电话提供语音服务了。记住，电话卡可和modem没有任何关系。

9. SCSI support SCSI设备支持

如果有SCSI设备，就回答Y。现在一般PC机不会有SCSI。接着会有提示要求更进一步的资讯，像是你是否要支援光驱，硬盘，还有你使用的是那一种SCSI界面卡。这部份请参阅SCSI-HOWTO，有更详细的说明。如果你的启动分区是SCSI设备，不要选择SCSI模块支持。在一般的SCSI后是SCSI低级设备驱动程序。再次重申，模块仅仅是用在不在启动分区的设备。

#SCSI disk support-----指硬盘而言，如果有SCSI硬盘，那么就要选这个选项。

#SCSI tape support-----指磁带机而言，如果您有SCSI的磁带机，那么就要选这个选项。

#SCSI CDROM support-----指CDROM，如果您有SCSI光驱，这一项一定要选。

#SCSI generic support-----指其他有关SCSI的东西，也许您有一台SCSI的扫描器或是烧录机，或是其他有关SCSI的配备，您就要选这一项。而且，除此之外，您还必须准备关于这些配备的软件。

##Some SCSI devices (e.g. CD jukebox) support multiple LUNs

#Probe all LUNs on each SCSI device-----通常这个选项大部分的人都不会选。我们举个例子来说，如果您的SCSI光驱是那种多片装的，就是一台光驱，但可以一次放好几片光盘片的那种。这种我们叫做Lun。

#Verbose SCSI error reporting (kernel size+=12K) -----如果认为您的SCSI硬件配备有些问题，想了解一下它出现的错误信息。那么您可以把这个选项选y，Linux核心会告诉您有关于您的SCSI配备的问题（如果有的话）。不过，它会增加核心约12KB左右。

##SCSI low—level drivers

下面总共有接近30张的SCSI卡，您可以依需求做选择SCSI卡牌子。

c AIA1542 support

AdaPtec AIHA1740 support

AdaDtec AHA274X / 284X / 294X support

AdaPte 7000FASST SCSI support

AdaPtec AHA152X / 2825 support

Advansys SCSI support

Always IN2000 SCSI support

Advansys SCSI support

Always IN2000 scsi support

AM53 / 79C974 PCI SCSI sppport

Buslogic SCSI Support

DTC3180 / 3280 SCSI support

EATA ISA / EISA (DPT PM2011 / 021 / 012 / 022 / 122 / 322) support

EATA - DMA (DPT , NEC , AT & T , SNI , AST , 01ivetti , Alpatronix) support

EATA - PIO (old DPT PM2001 , PM2012A) support

Future Domain 16xx SCSI support

Generic NCR5380 / 53c400 SCSI support

NCR53c405a SCSI support

NCR53c7 , 8xx SCSI support

NCR53CSXX SCSI support

IOMEGA Parallel Port ZIP drive SCSI support

PAS16 SCSI Support

Qlogic FAS SCSI support

Qlogic ISP SCSI support

Seagate ST-02 and future Domain TMC-8xx SCSI support

Trantor T128 / T128F / T228 scsi support

Ultrastor 14F / 34F support

Ultrastor SCSI support

10. I2O device support

这个也不清楚，帮助里说是这个需要I2O接口适配器才能支持的，在智能Input/Output (I2O) 体系接口中使用，又是要硬件，不选了。

11. Network device support网络设备支持

上面选好协议了，现在该选设备了，内容多得很。还好里面大概分类了，有ARCnet设备、Ethernet (10 or 100 Mbit)、Ethernet (1000Mbit)、Wireless LAN (non-hamradio)、Token Ring device、Wan interfaces、PCMCIA network device support几大类。耐心点，一般说来都能找到自己用的网卡。如果没有，你只好自己到厂商那里去要驱动了。如果这个选项没有打开的话，那么以下的选项将不会出现。它是在选择网络卡或是网络的设备。例如，PLIP，PPP，SLIP，还有各式各样的网络卡，所以这个选项通常是选y。

#Dummy net driver support-----如果有SLIP或PPP的传输协议，那么要把这一项打开。因为一来它不会让您的Linux核心增大。二来，对某些应用程

序来说，它可以让我们模拟出来的TCP / IP环境更像TCP / IP环境。如果您没有SLIP或PPP协议，就不用打开了。

#EQL (serial line load balancing) support-----如果有两个MODEM，两条电话线，而且用SLIP或PPP协议，可以用这个Driver以便让您的MODEM有两倍的速度。当然，在网络的另一端也要有同样的设备。

#PLIP (parallel port) support-----依字面上看，它是一种利用打印机的接口（平行接口），然后利用点对点来模拟TCP / IP的环境。它和 SLIP / PPP全都属于点对点通讯，您可以把两台电脑利用打印机的连接接口串联起来，然后，加入此通讯协议。如此一来，这两部电脑就等于一个小小的网络了。不过，如果电脑有提供打印服务的话，这个选项最好不要打开，不然可能会有问题（因为都是用平行接口）。

#PPP (point - to - point) support-----点对点协议，近年来，PPP协议已经慢慢的取代SLIP的规定了，原因是PPP协议可以获取相同的IP地址，而SLIP则一直在改变IP地址，在许多的方面，PPP都胜过SLIP协议。

#SLIP (serial line) support-----这是MODEM族常用的一种通讯协议，必须通过一台Server（叫ISP）获取一个IP地址，然后利用这个IP地址，可以模拟以太网，使用有关TCP / IP的程序。

##Ethernet (10 or 100Mbit)

如果您在学校接了校园网络并且使用网络卡，那么这个选项一定要选y，否则以下对网络卡的选择将不会出现。或是您有网络卡，这时您同样的也要选y。之后，下面会列出许多网络卡让您选择。像我们平常用的都是NE2000相容卡。

#3COM cards

#AMD LANCE and PCnet (AT1500 and NE2100) support

#Western Digital / SMC cards

##other ISA Cards (CONFIG . ISA) -----选y，以下才会列出有关ISA的网络卡。包括NE2000的兼容卡。

Cabletron E21xx support

DEPCA , DE10x , DE200 , DE201 , DE202 , DE422 support

EtherWORKS 3 (DE203 , DE204 , DE205) support

EtherExpress 16 support

HP PCLAN + (27247B and 27252A) support

HP PCLAN (27245 and other 27xxx series) support

HP 10 / 100VG PCLAN (ISA , EISA , PCI) support

NE2000 / NE1000 support

SK . G16 support

EISA , VLB , PCI and on board controllers -----选择网络卡，包括直接附在主机板上的那种。如果选择y，则底下会列出其他的网络卡让您做选择，这些卡对于一般人来说很少会去用到。所以大部分的人这项是选N的。

#Pocket and portable adaptors-----通常用在可携式的电脑上，这类型的网络卡（口袋型的），由于体积很小在安装和取下方面很方便，因此笔记本相关电脑上便常常采用这种网络卡。

#Token Ring driver support-----Token Ring是IBM电脑上的网络。它叫令牌环网络，和以太网是很类似的东西。如果您希望使用的Token Ring网络卡以便连接到这种网络，那么选Y，一般人都选N。

#ARCnet support-----这也是一种网络卡，通常一般人用不到，所以选n。如果您有这样的网络卡，请看Documentation / networking / arcnet . txt的说明。

12. Amateur Radio support 业余无线广播

可以用来启动无线网络的基本支持，目前的无线网络可以通过公众频率传输数据，如果你有此类设备就可以启用，具体请参考AX25和HAM HOWTO 文档。

13. IrDA (infrared) support红外支持

14. ISDN subsystem

如果使用ISDN上网，这个就必不可少。ISDN (Integrated Services Digital Network)，它的中文名称是综合数字服务网络，是一个利用电话线，把声音，影片信息以数字的方式传送的数字网络，它需要电话交换机设备有支持ISDN，这通常需要电信局来做安装，对于在家工作的人来说，ISDN可能是最舒适最便宜的一种方式，因此有愈来愈多的人使用它。不过，除非是公司，不然一般人很少会使用到ISDN的，所以这部分的选项大都选N。如果您选择Y，则下面会出现一些有关ISDN的问题。如果需要用到ISDN，可以去看看杂志的介绍。只要是有关网络的杂志应该都会有介绍。还需要启用Support synchronous PPP选项 (参考PPP over ISDN)。

15. Old CD-ROM drivers (not SCSI、 IDE) 非SCSI/IDE口的光驱

如用IDE的CD-ROM，不选。

以下是选择非 IDE / ATAPI和 SCSI界面的光驱，这些光驱通常有自己专属的接口卡也是比较旧型的光驱类型。如果有这些光驱，则这个选项要选y，否则选n。如果您选择n，则会跳过以下光驱的选项。

Aztech / orchid / okano / Wearnes / IXC / CyDROM CDROM support

Goldstar R420 CDROM support

Matsllshita / panasonic / Creative , longshine , TEAC CDRW Support

Mitslmi (no XA / Multisession) CDROM Support

Mitsumi (XA / Multisession) CDROM support

optics Storage DCLPHIN 8000AT CDROM support

Philips / LMS CM206 CDROM support

Sanyo CDR - h94A CDROM sunnort

Soft configurable CDROM interface card support

Sony CDU31A CDROM sunnort

Sony CDU535 CDROM support

16. Character devices字符设备

所谓字符设备通常是指以字符为单位做处理的设备，例如终端机就是其中一项。原则上，我们对于这些选项的选择也是以预设为主。这个内容又太多了，先使用缺省设置，需要的话自己就修改。把大类介绍一下吧：

#Virtual terminal ----- 选择 “ y ”，内核将支持虚拟终端。

#Support for console on virtual terminal-----选择 “ y ”，内核可将一个虚拟终端用作系统控制台。

#Standard/generic (dumb) serial support-----选择 “ y ”，内核将支持串行口。标准序列接口的选定。如果您是用serial的鼠标（大部分的人都是用这个），或

是MODEM的话，则这一项一定要选。大部分的人这一项都选y。

#Support for console on serial port-----选择“y”，内核可将一个串行口用作系统控制台。

##I2C support-----I2C是Philips极力推动的微控制应用中使用的低速串行总线协议。如果你要选择下面的Video For Linux，该项必选。

#Mice鼠标。现在可以支持总线、串口、PS/2、C&T 82C710 mouse port、PC110 digitizer pad-----根据需要选择。非串行的鼠标，支持象PS/2。看一看你鼠标的键。许多双键鼠标是PS/2，即使它们的连接看起来象串行的。

#PS/2 mouse (aka "auxiliary device") support ----- 如果用户使用的是PS/2鼠标，则该选项应该选择“y”。

#Joysticks-----手柄。即使在Linux下把手柄驱动起来意义也不是太大，游戏太少了。

#Watchdog Cards：虽然称为Cards，这个可以用纯软件来实现，当然也有硬件的。如果你把这个选中，那么就会在你的/dev下创建一个名为watchdog的文件，它可以记录你的系统的运行情况，一直到系统重新启动的1分钟左右。有了这个文件，你就可以恢复系统到重启前的状态了。

#Video For Linux-----支持有关的音频/视频卡。

##Ftape, the floppy tape device driver---

#Digiboard PC / Xx Support-----这是一张叫 Digiboard PC / XX卡的 driver，这种卡上面有很多个 serial port的插槽（一般来说只有两个），可以用来连接很多个MODEM，在民间的BBS站很常用到，如果您有这样的东西，您必须选这项为y。有兴趣的读者可以读读Documentation / digiboard . txt 的内容。

#Cyclades async mux support-----同上，这也是一种能接很多个serial port插槽的卡的驱动程序。

#Stallion multlport serial support-----同上，这也是其中一种卡。

#SDL RISCom / 8 card support-----这也是其中的一个支持muti - serial卡的driver。

#Parallel printer support-----有打印机的或是使用到并行接口的人这一项一定要选。除非是用serial的打印机。还有如果您有使用PLIP，那么这项也请选上。

#Mouse support-----大部分的人这一项并不用选y。因为大部分的人是用serial的鼠标，除非有些人是用一种附有接口卡的鼠标，这时这个选项才要选上，如果您选y，则底下会列出您的BUS鼠标所用的接口卡。

#support for user misc device modules-----除非您有所谓的触摸式显示器或是光笔等东西，否则这一项选n。

#QIC - 02 tape support-----非 SCSI界面的磁带机，除非您有，否则选 n。

#Ftape (QIC - 80 / Travan) support-----如果有磁带机，而这个磁带机是接在软盘控制卡上，这个选项才要选y。

#Watchdog Timer support-----一般人不需要这个选项，如果您选上这个选项，则您要用mknod在 / dev下建一个watchdog的文件。请看 Documentation / watchlog . txt的解释。

#Enhanced Real Time Clock Support-----关于系统上Clock的东西，您必须自己用mknod在 / dev / 下建立一个文件叫rtc。如此一来，在 / proc / 下将可以看到rtc的信息。

有关于rtc的内容请看Documentation / rtc . txt。一般是选n。

17. USB support USB支持

很多USB设备，比如鼠标、调制解调器、打印机、扫描仪等，在Linux都可以得到支持，根据需要自行选择。

18. File systems文件系统

Linux上有支持约二十几种的文件系统，有支持某个文件系统的意思是，可以存取某个文件系统的数据或是做拷贝动作。在这些文件系统中，通常的选择方法是按照原来预设的方式，不过，在其中，EXT2FS那个选项无论如何一定要选，因为那是Linux系统所使用的文件系统。其他的则依需求做选择。

通常是ext2而让其余的使用模块。

#Kernel automounter support-----选择“y”，内核将提供对automounter的支持，使系统在启动时自动 mount 远程文件系统。

#Standard (minix) -----新的套件不再建立 minix 文件系统，而且很多人不使用它，但是把它配置在核心里仍然是个好主意。某些 rescue-disk 程序会用到它，而且仍然有许多磁片可能用 minix 文件系统，因为 minix 文件系统对于处理磁片方面是最好的。当初Linus是因为对Minix这个小型的操作系统有很深的经验，所以才写出Linux这个操作系统。Minix文件系统通常用在磁盘上，有时会用到它。

#Extended fs ----- 这是扩充文件系统的第一版，现在已经不再使用。

#Second extented fs-----这是现在新发行的套件所广泛采用的文件系统，你可能会有一种。这个是linux文件系统，请务必选y，如果问我说选n会有什么后果，我也不知道。除非您能把 Linux装在 DOS的目录下。

#xiafs filesystem-----这个文件系统曾经一度很普遍，但是在写这份文件时，我已经不知道有任何人在使用它了。

#DOS FAT fs -----DOS FAT文件格式的支持，可以支持FAT16、FAT32。这个选项是DOS的文件系统，如果您没有选y，则下面的MSDOS，VFAT，umsdos将不会出现。

#msdos-----DOS文件系统的格式。如果你想要在 linux 下使用你硬盘中的 MS-DOS 分割区，或是想将用 MS-DOS 格式化的磁盘挂进来的话，回答y。

#VFAT (Windows95) fs -----windows95所支持的文件系统，是我们常说的vfat文件系统，如果您的系统中装有windows95，那么选择这个文件系统将对以让您看到windows95的长文件名。

#umsdos : UNIX like fs on top of std MSDOS FAT fs ----- 如果把Linux装进DOS的一个目录下那么您则要选择这一项。不建议这样做，因为如此一来，就不能看到长文件名了。而且效率上，并不是很好。

#ums-dos-----相当 slick 的文件系统，它能使 MS-DOS 文件系统拥有更多的特性，像是长档名等等。这对那些不使用 MS-DOS 的人（像我）并不是很有用。

#/proc-----这是最 slick 的文件系统之一。它不是你硬盘分割区里的任何东西，不占用硬盘的空间，而是核心与程序之间的文件系统介面，它表示的只是内存里头的状况和各个程序执行的情形，它也记录了您硬件上配备。。许多程序工具（像 ps ）都会用到它。如果已经将它安装好了，有空不妨试试看 cat /proc/meminfo 或者是 cat /proc/devices 。有些 shells，像是 rc，会用 proc/self/fd（在其它系统上为 /dev/fd）来处理输出入。几乎可以确定你在这里得要回答y，有许多重要的 Linux 标准工具是靠它来运作的，否则有些指令会出问题。

#Root file system on NFS-----一般不选，除非您的电脑上没有硬盘，希望通过网络由别人的硬盘开机过Linux如此一来才有需要选这项。同时对方也要执行rarp的服务。

#System V and Coherent-----这是为 System V 以及 Coherent 的分割区而设的。如果希望支持System V或Xenix的相关UNIX系统的FS并读取它们的数据，那么才有必要选这个，否则一般来说这个选项是选n的。

#Quota support -----Quota可以限制每个用户可以使用的硬盘空间的上限，在多用户共同使用一台主机的情况中十分有效。

#ISO 9660 CD-ROM file system support-----光盘使用的就是ISO 9660的文件格式。

#Mandatory lock support-----有些很特殊的database应用软件会用到它，一般人这个选项是选n。而且，如果选y的话，必须有最新版的NFS软件，最新版的samba软件等。

#NTFS file system support-----ntfs是NT使用的文件格式。

#UFS filesystem support-----这是BSD，SunOS，FreeBSD，NetBSD或Nextstep所使用的文件系统。如果您在电脑上有这些操作系统的话，那么可以选这一项。否则一般人都选n。

##Network File Systems-----网络文件系统

NFS-----如果你在网络环境下而且想要分享档案，回答 y。如果希望挂上别的电脑的文件系统，那么这个选项一定要选进去。它可以让您利用网络把别人的硬盘当成自己的来使用（把它变成一个目录）。对于一般人来说，这个选项是选y。

#SMB filesystem support-----这个文件系统让您可以挂上windows95或windowsNT的文件系统，也就是您也可以抓到在windows下，网上邻居上的电脑。

#SMB long filename support-----支持windows95的长文件名。

#NCP filesystem support-----NCP是一种网络的通讯协议，用在跑IPX协议上，它可以利用IPX协议让两台电脑之间的文件共享，并做沟通。如果您想挂上有关Novell的Netware文件系统，那么这个选项就选上去吧。

##Partition Types-----分区类型，该选项支持一些不太常用的分区类型，用户如果需要，在相应的选项上选择“y”即可。

##Native Language Support-----本地语言支持

附：不知道需要那些文件系统怎未办？

键入 mount 它看起来会像这样：

```
sunly% mount
```

```
/dev/hda1 on / type ext2 (defaults)
```

```
/dev/hda3 on /usr type ext2 (defaults)
```

```
none on /proc type proc (defaults)
```

```
/dev/fd0 on /mnt type msdos (defaults)
```

仔细看看每一行；在“type”后面的那个字就是文件系统的格式。在这个例子中，我的/和/usr分割区是second extended格式，我使用/proc，而且挂有一张以msdos(bleah)为文件系统格式的磁片。如果你有使用/proc，可以试试 cat /proc/filesystems。它会给你一份目前使用的核心所支援的文件系统列表。

19. Console drivers 控制台驱动

#VGA text console ----- 选择“y”，用户就可以在标准的VGA显示方式下使用Linux了。一般使用VGA text console就可以了，标准的80*25的文本控制台。

#Video mode selection support

20. Sound sound声卡驱动

如果你能在列表中找到声卡驱动那自然最好，否则就试试OSS了。阅读帮助文件从列表中小心的选取。确信为你声卡真确的选择了I/O和IRQ。声卡的MPU I/O是0选项。一般是330，如果不对不必担心。模块的好处就是在核心编译以后你还能重新编译、安装模块并挂上核心。

如果有声卡，请去了解一下声卡的 IRQ和 DMA等信息，并了解是属于哪一种的。现在大部分的人使用的都是Sound Blaster或是它的相容卡。有关这类的信息请看各个声卡的说明书。当选Y时，出现下面的画面。就依您的声卡来做选择吧！

Pro Audio Spectrum 16 support

Sound Blaster (SB , SBPro , SB16 , clone) support

Generic OPL2 / 0P13 FM synthesizer support

Gravis Ultrasound support

MPU-401 support (NOT for SB16)

6850 CART Midi support

PSS (ECho - AD12111) support (NOT for SB16)

16 bit samplings option of GUS (NOT__GUS__MAX)

GUS MAX support

Microsoft Sound System support

EnsolliQ Soundscape support

MediaTrix AudioTrix Pro support

Support forMAD16 and / or Mozart based cards

Support for Crystal CS4232 based (PnP) cards

Support for Turtle Bench Wave Front (Maul , Tropez) synthesizers

/ dev / dsp and / dev / audio support----- 这个选项通常是必要的。因此大部分的人选y，如果没有这选项，则很多的游戏将没有声音效果。

#MIDI interface support-----支持MIDI界面。

#FM synthesizer (YM3812 / 0PL - 3) support

#I / O base for SB Check from manual of the card-----声卡的I / O地址。括号是常用的选项。

#Sound Blaster IRQ Check from manual of the card-----声卡的IRQ，通常是1或5。

#Sound Blaster DMA 0 , 1 for 3-----声卡的DMA，通常是1。

#Sound Blaster 16 bit DMA 5 , 6 or 7 (usel for 8 bitcards) (SB . DMA2) -----5

#MPU401 I / O base of SB16 , Jazz16 and ES1688 Check from manual of the card

-----0

#SB MPU401 IRQ (Jazz16 , SM Wave and ES1688) Usel with SB16-----1

#Audio DMA buffer size 4096 , 16384 , 32768 or 65536-----65536

##Additional low level drivers-----如有其他种类的声卡，则这项要选上去，以下会列出其他的声卡供选择。

21. Kernel hacking安全模式

通俗的说，这是windows安全模式,找不到明确解释，就引用这个说法。> 这是从 Linus 的 README 里摘录的：

kernel hacking 配置的细节通常会产生一个更大或是更慢的核心（或者是又大又慢），而且可能因为将一些常式配置成很活跃地去尝试中断一些不良的程序码以便找出核心里的问题，而使得核心变得比较不稳定。所以要产生一个核心 产品 的话，你在这里应该回答n。

配置了这个，即使在系统崩溃时，你也可以进行一定的工作了。

如果对Linux核心很熟，想了解这个操作系统上的很深的信息，则这个选项才打开，将会在 / proc下看到profile这个文件，记录着相当多的信息。这些信息是供除错的，对一般人来说，这个选项是选n，以免造成系统不稳固，或是引起其他的问题。

总算配置完了，现在存盘退出，当然你也可以把现在的配置文件保存起来，这样下次再配置的时候就省力气了。存盘退出的文件是一个可以编辑的文本文件。你可以用你的文本编辑器如：Emacs,vi,CrispLite,joe,等等。配置文件在/usr/src/linux目录名字是.config。前面的点使普通的.(ls)无法显示，加上-a选项就能看见。编辑文件中在配置时遇到麻烦的数字。然后，make dep 把你的配置信息从.config传送到特定的子目录完成安装。最后，make clean 准备最后的核心编译。

3、编译内核

编译有关的命令有如下几个：

```
#make dep
#make clean
#make zImage
#make bzImage /make zdisk
#make modules
#make modules_install
#depmod -a
```

第一个命令make dep实际上读取配置过程生成的配置文件，来创建对应于配置的依赖关系树，从而决定哪些需要编译而那些不需要；

第二命令make clean完成删除前面步骤留下的文件，以避免出现一些错误；

第三个命令make zImage和第四个命令make bzImage实现完全编译内核，二者生成的内核都是使用gzip压缩的，只要使用一个就够了，它们的区别在于使用make bzImage可以生成小一点的内核，比如在编译2.2.16版本的内核时如果使用make zImage命令，那么就会出现system too big的错误提示。建议使用make bzImage命令,压缩的核心它将留在/usr/src/linux/arch/i386/boot目录名为zImage。make zdisk写zImage到软盘在/dev/fd0(就是DOS中第一软驱A:)。在开始之前你必须要有张盘在软驱中。你可以同样用make zImage 然后把它拷贝到软盘cp /usr/src/linux/arch/i386/boot/zImage /dev/fd0 记住是高密度的，底密度的720k磁盘将不能启动核心。

后面三个命令只有在你进行配置的过程中，在回答Enable loadable module support时选了"Y"才是必要的，make modules和make modules_install分别生成相应的模块和把模块拷贝到需要的目录中。

严格说来，第七个命令和编译过程并没有关系，它是生成模块间的依赖关系，这样你启动新内核之后，使用modprobe命令加载模块时就能正确地定位模块。

4、启用内核

通常，核心安装叫做vmlinuz。过去Unix使用者共同起了这个名字。"z"表示压缩，"v"和"m"意思是"virtual"(虚拟)和"sticky(粘性的)"，各自属于内存和磁盘管理。建议保留vmlinuz核心，直到知道它工作。

为了能够使用新版本的内核，还需要做一些改动：

```
#cp /usr/src/linux/System.map /boot/System.map-2.2.16
```

```
#cp /usr/src/linux/arch/i386/bzImage /boot/vmlinuz-2.2.16
```

以上这两个文件是刚才编译时新生成的。下面修改/boot下的两个链接System.map和vmlinuz，使其指向新内核的文件：

```
#cd /boot
```

```
#rm -f System.map vmlinuz
```

```
#ln -s vmlinuz-2.2.16 vmlinuz
```

```
#ln -s System.map-2.2..16 System.map
```

注意：要保留vmlinuz核心，以下列步骤进行

```
#cp /usr/src/linux/System.map /boot/System.map-2.2.16
```

```
#cp /usr/src/linux/arch/i386/bzImage /boot/vmlinuz-2.2.16
```

```
#cd /boot
```

```
#rm -f System.map
```

```
#ln -s System.map-2.2..16 System.map
```

现在#vi /etc/lilo.conf,增加如下一段：

image=/boot/vmlinuz-2.2.16 是设定为已经安装的核心

label=linux2.2.16 则是由 lilo 用来告诉你现在要启动的是那个核心或作业系统，

read-only

root=/dev/hda2 则是这个特别的作业系统的根目录 /

其中root=/dev/hda2一行要根据需要自行加以修改。

运行：#sbin/lilo -v 保存执行命令：lilo你将看到核心标签，第一个是星号。如果你没有看到新核心的标签或LILO出现错误，你需要重新对/etc/lilo.conf工作（看下面的LILO分析）。

确认对/etc/lilo.conf的编辑无误，现在重新启动系统：

```
#shutdown -r now
```

不建议使用热启动或ctrl+Alt+del键。在一些情况下，文件系统不完全卸载会损坏打开的文件。在LILO提示时，如果你需要启动旧的核心或使用一些参数启动，如果你没看见启动提示，你可以试用shift或ctrl键，这样启动提示就出现了。一旦出现，按tab看核心标签。输入标签和可选参数启动。通常，在/etc/lilo.conf文件指定的时间后自动启动核心。启动时，你可能看见一些出错信息就象SIOCADDR。这常常显示模块（一般是网络模块）没有引导。处理这事很简单，如果有此一错，"VFS, cannotmount root",你就不要在核心中编译适当的磁盘或文件系统支持。

附录：LILO分析（技术性强，仅供参考）

第一部分 LILO介绍

LILO（Linux Loader）是Linux自带的一个优秀的引导管理器，使用它可以很方便地引导一台机器上的多个操作系统。与其他常用的引导加载程序相比，LILO引导方式显得更具有艺术性，对其深入的理解，将有助于我们方便地处理多操作系统、网络引导、大硬盘及大内存等诸多棘手的问题。

LILO的引导机制-----众所周知，计算机的最初启动是由BIOS控制的，在对一些硬件（如：内存、键盘等）初始化之后，它会试图加载硬盘的主引导记录（MBR）或软盘的引导扇区。MBR可通过两种方式运行，其一是定位到活动分区并加载相应的引导扇区，然后由引导扇区完成该分区内操作系统的基本组件的加载；其二是直接从一指定分区中加载信息，并通过它装入任一分区的操作系统，诸如LILO、OS/2 boot loader及Partition Magic等引导加载程序都可以配置成这种方式。软盘的引导扇区相当于硬盘活动分区的引导扇区，它通常用于装入软盘上的操作系统。由此可见，只要把LILO安装在MBR、活动分区或者引导软盘上，就能接管计算机的控制权，然后由LILO完成后继的引导过程。LILO中建立一个引导表地址编码，借此它的引导程序就能定位到Linux的内核文件，这种地址编码既可以按照柱面/磁头/扇区(CHS)模式，又可以采用LBA的线性块号模式，因此，即使对某些SCSI控制程序LILO也能运转良好。

当LILO定位到配置文件后，经过预引导过程，就显示提示符： LILO boot:

此时，系统允许选择引导不同的操作系统或者不同的内核配置，按Tab键显示可选项列表，然后输入可选项或者直接回车选择缺省配置，如果选择了引导Linux，还可以直接传递参数到系统内核。

和其他系统的引导加载程序相比，LILO具有更大的灵活性，其引导方式也更丰富多彩。

当LILO被安装在硬盘的MBR、活动分区或引导软盘上时，作为原引导程序的替身，它能引导任一硬盘任一分区上的Linux和其他操作系统；除了引导扇区，它没有任何隐含文件，也不需要使用特定的分区，它的配置文件可以在任何分区、甚至是存放在与Linux毫不相干的DOS分区的某个子目录下；它能引导几个不同的内核配置，甚至是几个不同的内核；它能引导同一机程序上的多个Linux版本；可达16个。

它能从网络上引导Linux。

LILO的灵活性使得其配置变得相当复杂，当有多个系统共存时，建议先安装其他操作系统，最后再装Linux，这样，设置LILO对其他系统的引导会相对简单一些。

第二部分 LILO参数

通常我们谈到LILO，会涉及到两个方面——LILO引导程序和LILO安装命令/sbin/lilo。

为了不至于混淆这两个概念，本文将用LILO表示LILO引导程序，而lilo表示/sbin/lilo。一般地，LILO使用一个文本文件/etc/lilo.conf作为其配置文件。lilo读取lilo.conf，按照其中的参数将特定的LILO写入系统引导区。任何时候，修改了/etc/lilo.conf，都必须重新运行lilo命令，以保证LILO正常运lilo.conf使用的配置参数很多，配置起来也相当复杂。下面以RedHat Linux为例作一些初步探讨，RedHat的lilo程序包版本为0.20，别的Linux发行版本可能会有所出入，但不会太大。

lilo.conf文件中的配置参数分为两部分，一部分是全局参数，另一部分是引导映像参数。引导映像参数作用于每一个引导映像区。如果某一引导映像参数（例如：password与全局参数的定义相抵触，则以该引导映像参数的定义为准，但仅限于该引导映像区。LILO的引导参数有很多，在此只对一些比较重要的参数作一介绍。与Linux系统其他的配置文件一样，“#”号后的一行文字表示注释。

1. “boot=” 此参数指明包含引导扇区的设备名（如：/dev/had），若此项忽略，则从当前的根分区中读取引导扇区。
2. “root=” 此参数告诉内核启动时以哪个设备作为根文件系统使用，其设定值为构造内核时根文件系统的设备名，可用的设备名有：
 - (1)/dev/hdaN~/dev/hddN：ST-506兼容硬盘，a到d上的N个分区
 - (2)/dev/sdaN~/dev/sdeN：SCSI兼容硬盘，a到e上的N个分区
 - (3)/dev/xdaN~/dev/xdnN：XT兼容硬盘，a到b上的N个分区
 - (4)/dev/fdN：软盘，A：（N=0）或B：（N=1）
 - (5)/dev/nfs：由网络取得根文件系统的标志

3. “ nfsroot= ” 若需通过NFS提供根文件系统来引导无盘工作站，此参数为内核指定了网络根文件系统所在的机程序、目录及NFS，其格式为：
nfsroot= (server_ip :) root_dir (,nfs_options)
4. “ nfsaddr= ” 设定网络通讯所需的各种网络界面地址，如无此参数，则内核会试图用反向地址解析协定(RARP)或启动协定(BOOTP)找出这些参数，其格式为：nfsaddr= 客户端IP : 服务端IP : 网关IP : 子网屏蔽 : 客户端名称 : 网络设备名 : auto
5. “ image= ” 指定Linux的内核文件。
6. “ delay= ” 设定引导第一个映像前的等待时间。
7. “ disk= ” 此参数为某一特殊的硬盘定义非标准参数。
8. “ append= ” 为内核传递一个可选的参数行，其典型的应用是为不能完全由系统自动识别的硬盘指定参数，如：append = "hd=64,32,202"
9. “ label= ” 此参数为每个映像指定一个名字，以供引导时选择。
10. “ read-only ” 设定以只读方式挂入根文件系统，用于文件系统一致性检查（ fsck ）。
11. “ install= ” 安装一个指定文件作为新的引导扇区，缺省为/boot/boot.b。
12. “ loader= ” 说明所使用的链加载程序(chain loader)，缺省为/boot/chain.b，如果不是从首硬盘或软盘启动，那么，此选项必须说明。
13. “ table= ” 说明包含分区表的设备名，如果此参数忽略，引导加载程序将不能传递分区信息到已引导的操作系统。当此参数指向的分区表被修改时，必须重新运行/sbin/lilo。
14. “ init= ” 内核初始化时执行的程序，通常过程为init、getty、rc和sh，版本1.3.43以来的Linux内核能够执行/sbin/init说明的命令行，若在引导过程中出现问
题，则可设置init=/bin/sh直接跳到Shell。
15. “ ramdisk_start= ” 由于内核不能放在压缩的内存文件系统映像内，为使内核映像能够和压缩的内存映像放在一张软盘内，加入 “ ramdisk_start= offset ”，这样内核才能开始执行。
16. “ mem= ” 此参数的目的之一是为Linux指定使用的内存数量：如mem=96MB，目的之二是指定mem=nopentium告诉内核不要使用 4 MB分页表。
17. “ vga= ” 设置显示模式，如80 × 50、132 × 44等。
18. “ linear ” 产生用于替换硬盘sector/head/cylinder地址（硬盘几何参数）的linear扇区地址。linear地址在运行时产生并且不依赖于硬盘几何参数。某些SCSI硬盘和一些以LBA方式使用的IDE硬盘可能会需要使用这个参数。注意：在将LILO安装到软盘上时不能使用 “ linear ” 参数。
19. “ prompt ” 给出 “ boot: ” 提示，强制LILO等待用户的键盘输入，按下回车键则立即引导默认的操作系统，而按下Tab键则打印可供选择的操作系统。当 “ prompt ” 被设置而 “ timeout ” 没有被设置时，系统会一直处于等待状态而不引导任何操作系统。不设置该参数时，LILO不给出 “ boot: ” 提示而直接引导默认操作系统，除非用户按下了Shift、Ctrl、Alt三键中的任何一个。大多数情况下，如果你的硬盘上有多个操作系统，建议使用参数，它留给用户一个选择的余地。
20. “ timeout= ” 设置等待键盘输入的时长，单位是0.1秒。超过这段时间没有输入则为超时，系统将自动引导缺省的操作系统。如果不设置本参数，缺省的超时时间长度为无穷大。

21 . “ other= ” 设置包含非Linux操作系统，如DOS、SCO UNIX、Windows 95等系统引导映像的文件或设备。

22 . alias=name 给当前操作系统起一别名。

第三部分 LILO典型配置方法

通常情况下，Linux的安装程序自身就可以完成LILO的安装配置，从而较好地解决多重系统的引导问题，如果系统不能自动完成这种配置，则可以通过手工修改配置文件/etc/lilo.conf来实现不同条件下的引导。

1.当系统能自动完成配置时

对于这种情况只有一个建议：将LILO安装到Linux分区的根上，而不是MBR这个多事地带。假设当前hda1中装有DOS/Windows，hda2中安装了Linux，则/etc/lilo.conf的内容大致如下：

```
boot=/dev/hda2 # 指定引导位置
compact
delay=50 # 延时 5 秒
root=current # 根在当前分区
image=/boot/vmlinuz # 指定linux的内核文件
label=linux # 用linux为代表名称
other=/dev/hda1 # 其他操作系统所在的分区
table=/dev/hda # 指定包含分区表的硬盘
label=dos # 用dos为代表名称
```

2. 当系统无法自动完成配置时

系统无法自动完成配置的情况不外乎两种：

- （1）BIOS不能直接看到Linux的根分区；
- （2）BIOS只能读写标准IDE硬盘的前504MB。

这时，必须遵循一个最基本的原则：建立一个BIOS能存取的较小的Linux分区，其中包含内核文件、映射文件及链加载程序等必要内容，而根则可以另外是一个独立的分区。至于配置上的其他细节，我们通过以下实例来进行说明。

第四部分 lilo.conf配置实例

有了这些基础知识，我们可以很容易地按照自己的意图配置LILO。

例一 . lilo.conf文件

```
boot=/dev/hda #将LILO安装在MBR。LILO作为主引导管理器
message=/boot/message #注释为/boot/message
compact #产生一个更小的“ map ” 文件
map=/boot/map #指定“ map ” 文件为/boot/map
install=/boot/boot.b
password=zhoudi #设置口令
vga=normal #80x25文本模式
linear #使用“ linear ” 地址
```

```
prompt #提示用户键盘输入
timeout=50 #超时时长为5秒
default=dos #缺省引导label为dos的操作系统
image=/boot/vmlinuz-2.0.34-1#设定Linux所用核心
```

```
#设置Linux核心引导映像
label=linux #标识为linux
root=/dev/hda1 #设置根文件系统
read-only #LILO以只读方式载入根文件系统
```

```
#设定MS-DOS或Windows 95
other=/dev/hda2 #DOS分区为第一个IDE硬盘的第二分区
label=dos #标识为dos
table=/dev/hda #主设备为第一个IDE硬盘
```

```
#设定SCO UNIX 注意：SCO分区必须设为活动（active）分区并将LILO安装在MBR上。
other=/dev/hda3
label=sco
table=/dev/had
```

这个例子中，LILO是作为主引导管理器来管理机器上所有操作系统的。LILO也可作为二级引导管理器，这只要将“boot”参数改为根分区就可做到。例如：`boot=/dev/hda1` 以这种方式使用LILO时，Linux根分区必须用DOS或Linux的fdisk程序将其设置为活动分区，并且这种方式只对硬盘主分区（不是扩展或逻辑分区）有效。

例二．一个标准的IDE大硬盘需安装Linux和DOS/Windows。

对于大硬盘问题，很多人只知道低于1024个柱面的限制，而不知为什么标准的IDE硬盘只能认前504MB。其实，BIOS的int13调用是采用三个位元组的CHS编码，10位为柱面号，8位为磁头号，6位为扇区号。可能的柱面号码是0～1023，可能的磁头号码是0～255，而磁道上可能的扇区号码是1～63，以这24位最多可以定址8455716864个位元组(7.875GB)。但不幸的是，标准的IDE介面容许256个扇区 / 磁道、65536个柱面及16个磁头。它自己本身可以存取 $2^3 \times 7 = 137438953472$ (128 GB)，但是加上BIOS方面63个扇区与1024个柱面的限制后只剩528482304(504MB)可以定址得到。

对策：在硬盘的前500MB中划分350MB(/dev/hda1)给DOS，150MB(/dev/hda2)给Linux，在相应的配置文件中应说明硬盘的参数。

```
boot=/dev/hda
... ..
disk=/dev/hda
bios=0x80
sectors=63
heads=16
cylinders=2100
image=/vmlinuz
append="hd=2100,16,23"
root=/dev/hda2
label=linux
```

例三 . 如果你有一块超过 8 G 的大硬盘, 并且需要把 Linux 安装在比较靠后的位置, 可以在安装的时候, 选择 linear 模式, 并且给它加上硬盘参数.

安装时候的硬盘参数可以这样写: hd?=CYLs, HEADs, SECs 其中的大写字母需要用实际的硬盘参数来替换, 这些参数可以从硬盘的标签上查到, 也可以看看 BIOS 设置里硬盘参数对应 LBA 模式的那一行. 问号是根据硬盘确定的, 实际使用时, 它可以是 a, b, c, d 四个字母中的一个. 比如:
hda=1869,63,255 这是 IBM 15.2G 硬盘的参数.

当然进入了 Linux 以后, 可以通过编辑 /etc/lilo.conf 加上这个文件, 然后运行一遍 lilo 达到同样的目的. 下面是本人未加参数前的 lilo.conf 的内容:

```
boot = /dev/hda
map = /boot/map
install = /boot/boot.b
prompt
timeout = 50
image = /boot/vmlinuz
label = linux
root = /dev/hda1
initrd = /boot/initrd-2.2.12-20.img
read-only
```

按照 linear 方式加入参数以后是如下格式:

```
boot = /dev/hda
map = /boot/map
install = /boot/boot.b
prompt
linear <-----加进了这一行
timeout = 50
image = /boot/vmlinuz
label = linux
root = /dev/hda1
initrd = /boot/initrd-2.2.12-20.img
read-only
```

append = "hda=1869,63,255" <-----加进了这一行 注意, append 参数是针对每个系统引导记录的, 一定要放在 image 的下面或者是 other 的下面, 这样它才可以发挥作用.

当再次起动 Linux 系统的时候, LILO 就按照线性模式对系统进行引导. 除了在硬盘上寻址定位的方式不同以外, 对其他方面没有什么影响.

四 配置多重启动 (LILO 有 1024 柱面的限制即如果把内核放到 8.4 个 G 之后则只能用软盘启动)

LILO 可以放在 MBR (即主引导扇区) 也可以放在 BOOT SECTOR 每一分区的第一个扇区

3 LILO 的配置文件 /etc/lilo.conf

Prompt 设置成对话模式 (如不这样会自动起用第一组设置)

Timeout=50 等待用户输入的时间 (单位 0.1 秒)

Image=/boot/vmlinuz 核心

Label=linux 名称

Root=/dev/hda6 设置根目录在那一个分区

Read-only 方式为只读

(以上为第一个操作系统的设置)

Other=/dev/hda1 另一个操作系统所在分区

Label=dos 名称

Table=/dev/had 设置hda1的分区表在那里

设置完之后保存退然后在提示符下执行一下LILO即可

第十四节：包过滤防火墙

一 本节主要是关于怎样用 ipchains 做一个简单的包过滤防火墙。

1 Ipchains主要是通过通过在input,forward,output三个环节上添加规则来过滤。

2原则 假设一个防火墙已经做好，一个请求连接过来之后它会按照规则审查，如果有符合条件的则按照规则走，如果没有按照缺省策略走。

3 常用的参数 -L 列出所有的规则 -A 追加一条规则 -D 删除一条规则

-s 源 (即从那里发出的请求) -d 目标 (向请求提供服务的主机)

-p 使用的协议类型 (tcp,udp,icmp) -P 改变缺省策略

4 示例：1) 本机的IP地址为192.168.1.10 如果我不想192.168.1.的的所有的主机ping本机则可以

ipchains -A input -s 192.168.1.0/24 -d 192.168.1.10 -p icmp -j DENY

/24(代表这个网段) 目标为1.10 协议为icmp 拒绝

2) 凡是192.168.1.0网段登陆本机23端口的一律拒绝

ipchains -A input -s 192.168.1.0/24 -d 192.168.1.10 23 -p tcp -j DENY

源 目标 端口 协议

5练习题 1) 凡来自192.168.1.112 的ping 则拒绝。

2) 凡去往192.168.1.112的http则拒绝，

3) 锁定本地的ftp服务

4) 改变改变缺省策略为拒绝。

1 参考答案：

1) ipchains -A input -s 192.168.1.112 -d 192.168.1.10 -p icmp -j DENY

2) ipchains -A input -d 192.168.1.112 80 -p tcp -j DENY

3) ipchains -A input -d 192.168.1.10 21 -p tcp -j DENY

4) ipchains -P input DENY

二 IP伪装

1 起用IP转发 1) echo 1 > /proc/sys/net/ipv4/ip_forward

2) ipchains -A forward -s 192.168.0.0/24 -j MASQ

3)ipchains -P forward DENY

2 如果192.168.0.网段的主机将本机作为网关则可以通过本机上网

如果系统是linux只需要修改路由表 (即更改缺省网关)

```
route del default gw
route add default gw 192.168.1.10
```

文章录入：admin 责任编辑：admin

上一篇文章：注册表趣味应用小集

下一篇文章：Windows2003的全攻略(一)

【发表评论】 【加入收藏】 【告诉好友】 【打印此文】 【关闭窗口】

最新热点

[x é ¼]PhotoshopÖÆ x ÷ ± ÆÖæµÄj-
 [x é ¼]ÐÞ»µÖÖÆ¬ ÖÖí ê ÄÄ È ¥Ö öj-
 [íÆ¼ö]Flash MX¿ ì ÈÜ È ëÄÄ Ö ëj-
 [íÆ¼ö]Flash MX¿ ì ÈÜ È ëÄÄ Ö ëj-
 [x é ¼]Ò»¼â¿Ü £ ¨ È Ê °ïÐÄ È Öj-
 [x é ¼]Ôö¼ Ó³¼ ° ÖÆ x ÷ Loadinj-
 [x é ¼]FireworksÄ È ° æÖÆ x ÷ í¼j-
 [x é ¼]PhotoshopÖÖÄ à Ò »¿Äí È j-
 [x é ¼]º ì Ä ± x ÓRed Hat Linuj-
 [x é ¼]Windows XP¹âÄìÆö ¶ j-

最新推荐

[x é ¼]PhotoshopÖÆ x ÷ ± ÆÖæµÄj-
 [x é ¼]ÐÞ»µÖÖÆ¬ ÖÖí ê ÄÄ È ¥Ö öj-
 [íÆ¼ö]Flash MX¿ ì ÈÜ È ëÄÄ Ö ëj-
 [íÆ¼ö]Flash MX¿ ì ÈÜ È ëÄÄ Ö ëj-
 [x é ¼]Ò»¼â¿Ü £ ¨ È Ê °ïÐÄ È Öj-
 [x é ¼]Ôö¼ Ó³¼ ° ÖÆ x ÷ Loadinj-
 [x é ¼]FireworksÄ È ° æÖÆ x ÷ í¼j-
 [x é ¼]PhotoshopÖÖÄ à Ò »¿Äí È j-
 [x é ¼]º ì Ä ± x ÓRed Hat Linuj-
 [x é ¼]Windows XP¹âÄìÆö ¶ j-

相关文章


Linux下硬盘分区的最佳方案
 Windows 2003全攻略（四）
 Windows 2003全攻略（三）
 Windows 2003全攻略（二）
 Windows2003的全攻略(一)

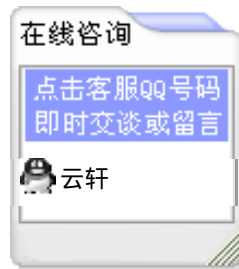


Ä » ÓÐ È î îÆ Ä ÄÜ

| 设为首页 | 加入收藏 | 联系站长 | 友情链接 | 版权申明 | 管理登录 |



Copyright © 2005-2010 『岁月如歌』 Powered By : 云轩 
 程序内核：PowerEasy2005+DVBS7.1 E-mail:cnlxtx@669.com QQ：81186540


[首页](#) | [业界新闻](#) | [文章中心](#) | [影音中心](#) | [下载中心](#) | [图片中心](#) | [音乐厅](#) | [语聊室](#) | [在线翻译](#) | [岁月论坛](#) | [岁月博客](#)


[设为首页](#)
[加入收藏](#)
[联系站长](#)