

目 录

第 1 章 Linux 的获取与安装初步	1
1.1 Linux 的获取	1
1.1.1 发行版本介绍	1
1.1.2 主要的发行版本	1
1.2 Linux 的安装初步	2
1.2.1 Linux 简介	3
1.2.2 安装准备	3
1.2.3 安装过程	5
1.3 Linux 和 FreeBSD	16
1.3.1 FreeBSD 简介	16
1.3.2 FreeBSD 表示硬盘分区的方式	16
1.3.3 Linux 与 FreeBSD 分享交换区空间	18
1.3.4 创建文件系统	20
1.3.5 程序的可移植性	21
第 2 章 Linux 的高级安装	22
2.1 Linux 的多重引导	22
2.1.1 与 DOS 和 Windows 95 共存于同一系统	22
2.1.2 与 Windows NT 共存于同一系统	23
2.1.3 用 Windows NT Loader 启动 FreeBSD 操作系统	26
2.2 Linux 启动盘的制作	27
2.3 Linux 和大内存	28
2.4 Linux 中模块的配置和使用	28
2.4.1 Linux 系统的模块	29
2.4.2 RPM	31
第 3 章 Linux 系统的配置	33
3.1 笔记本上的 Linux	33
3.1.1 简介	33
3.1.2 APM	33
3.2 声卡的配置	37
3.2.1 安装 isapnp 软件包	37

3.2.2	重新创建 Linux 内核.....	37
3.2.3	修改系统的引导程序脚本文件.....	37
3.2.4	一个 isapnp.conf 的例子.....	38
3.3	闪光警告.....	39
3.3.1	Speakerectomy.....	40
3.3.2	设置每一个主控制台发出“哔哔”声.....	40
3.3.3	termcap 和 terminfo.....	40
3.3.4	定义一个闪光警告.....	41
3.3.5	避免发出“哔哔”声的方法.....	41
3.4	设置与别的系统共享调制解调器.....	42
3.4.1	服务器端.....	42
3.4.2	客户端.....	43
3.4.3	安全问题.....	44
3.5	在 Linux 机器上设置串行通信设备.....	44
3.5.1	串行端口的名称.....	44
3.5.2	getty.....	46
3.5.3	setserial.....	46
3.5.4	拨号.....	46
3.5.5	用来接受拨号以及对外拨号的调制解调器的配置.....	48
3.5.6	设置终端连接到 PC 机上.....	52
3.5.7	使用两个以上的串行设备.....	53
3.5.8	将串行端口设到较高的速度.....	55
3.5.9	通信程序及工具.....	55
3.5.10	串行通信程序使用技巧.....	56
3.6	X Window.....	57
3.6.1	介绍.....	57
3.6.2	窗口、文件和窗口管理器.....	58
3.6.3	启动和退出 X Window 窗口管理器和桌面.....	59
3.6.4	X Window 系统中的设置工具.....	59
3.6.5	xterm.....	60
3.6.6	X Window 多任务.....	61
3.6.7	fvwm 窗口管理器和桌面.....	62
3.6.8	配置 X Window 系统.....	64
3.6.9	X Window 和窗口管理器.....	69
3.6.10	fvwm 配置文件.....	70
3.6.11	X Window 命令行参数.....	72
3.6.12	X Window 配置文件.....	73
3.6.13	字体.....	77
3.6.14	编译 X Window 应用程序.....	78

3.7	XFree86 配置	79
3.7.1	X11R5 和 XFree86 简介	79
3.7.2	支持的硬件	79
3.7.3	XFree86 的获取	80
3.7.4	配置 XFree86	81
3.7.5	tinyX	82
3.7.6	X 相关软件	83
3.7.7	X Window 常见问题解答	83
3.8	在 Linux 上启动 quota	84
3.8.1	quota 简介	84
3.8.2	quota 目前在 Linux 上的情况	85
3.8.3	配置 quota	85
3.8.4	高级设置	87
3.8.5	quotaon 和 quotaoff	89
3.9	ADSM 备份系统	89
3.9.1	安装 iBCS 模块	89
3.9.2	安装 ADSM 客户端	90
3.9.3	执行客户端	91
3.9.4	已知的问题	91
3.10	ZIP 磁盘	91
3.10.1	ZIP 的磁盘简介	92
3.10.2	设置 ZIP 磁盘	92
3.10.3	RedHat 4.2 的安装	95
3.10.4	Slackware 2.2 的安装	99
3.10.5	设立/etc/fstab	100
3.10.6	Debian 的安装	100
第 4 章	IP 别名和 IP Masquerade	103
4.1	Linux 下的 IP 别名	103
4.1.1	配置步骤	103
4.1.2	在重新启动机器后保持设置	105
4.1.3	IP 别名与电子邮件问题	105
4.2	Linux 主机的 IP Masquerade 功能	107
4.2.1	IP Masquerade 简介	107
4.2.2	IP Masquerade 现状	107
4.2.3	从 IP Masquerade 中获益	108
4.2.4	不需要 IP Masquerade 的用户	108
4.2.5	IP Masquerade 的运行	108
4.2.6	在 Linux 2.x 上使用 IP Masquerade	109

4.2.7	IP Masquerade 的设置	109
4.2.8	配置 IP 转发的方式	117
4.2.9	测试 IP Masquerade	118
4.2.10	IP Masquerade 的问题	118
4.2.11	IP Firewall Administration(ipfwadm)	120
4.2.12	IP Masquerade 以及需求方式(on-demand)拨号	123
4.2.13	IPautofw 数据包转发程序	123
第 5 章	Linux 充当防火墙和网络设备	124
5.1	Linux 充当防火墙	124
5.1.1	防火墙简介	124
5.1.2	安装与配置	125
5.1.3	代理服务器	127
5.2	Linux 充当网桥	130
5.2.1	设置	130
5.2.2	网卡模式	131
5.2.3	启动网桥	132
5.2.4	可能出现的问题	132
5.3	Linux 充当路由器	133
5.3.1	配置方法	133
5.3.2	可能出现的问题	135
第 6 章	电子讨论组与新闻组	136
6.1	以 majordomo 1.93 建立电子讨论组	136
6.1.1	系统要求	136
6.1.2	建立 majordom 帐号	136
6.1.3	获取 majordomo 软件	137
6.1.4	安装过程	137
6.2	新闻组软件 Leafsite	141
6.2.1	使用 Leafnode 软件	142
6.2.2	取得 Leafnode	142
6.2.3	安装 Leafnode	142
6.2.4	维护 Leafnode	143
6.2.5	工作原理	143
6.2.6	使用新闻阅读器	144
6.3	Usenet 和新闻阅读器	144
6.3.1	Usenet 新闻	145
6.3.2	安装 tm 和 tin	146
6.3.3	新闻传输代理	147

6.3.4	INN	147
6.3.5	trn 新闻阅读器	148
6.3.6	新闻组列表	149
6.3.7	trn 选择器	150
6.3.8	选择器的显示模式	152
6.3.9	trn 的使用	155
第 7 章	无盘工作站的配置	161
7.1	简介	161
7.1.1	网 络	161
7.1.2	工作原理	162
7.2	设 置	163
7.2.1	设置启动流程	164
7.2.2	设置 Linux	166
7.2.3	设置 DOS 及 Windows 3.1	171
7.2.4	设置 Windows 95	173
7.3	TCP/IP BootPROM 的相关工具程序	178
7.3.1	MENUEEDIT	178
7.3.2	BPHDBOOT	178
7.3.3	BPCLEAN	178
7.3.4	MrZip、MrUnzip 和 BpUnzip	179
7.3.5	Nobreak	180
第 8 章	DNS 配置	181
8.1	简介	181
8.2	DNS 的配置	181
8.2.1	启动 named 进程	183
8.2.2	设立简单的领域	185
8.3	维护	194
8.4	拨号连接的自动设置	195
第 9 章	Linux 的中文环境与 StarOffice	197
9.1	简介	197
9.1.1	chdrv	197
9.1.2	yact	197
9.1.3	extern	197
9.1.4	xcin 与 crvxt	198
9.2	安装与设置	198
9.2.1	chdrv	198
9.2.2	yact 的安装步骤	199

9.2.3	xcin 的安装与设置	200
9.3	中文环境的建立	201
9.4	StarOffice	202
9.4.1	简介	202
9.4.2	取得 StarOffice Office Suite 软件	203
9.4.3	安装 StarOffice	203
9.4.4	执行 StarOffice	205
9.4.5	StarOffice 使用技巧	205
第 10 章	Internet 服务器	206
10.1	介绍	206
10.2	服务器初始化脚本	207
10.3	inetd 服务器管理	209
10.4	FTP 服务器	210
10.4.1	FTP 服务器配置文件	211
10.4.2	FTP 用户帐号	213
10.4.3	FTP 服务器目录	213
10.4.4	FTP 文件	215
10.5	Web 服务器	215
10.5.1	配置 Web 服务器	216
10.5.2	启动 Web 服务器	220
10.6	Gopher 服务器	221
10.6.1	Gopher 的用户帐号和数据目录	223
10.6.2	配置明尼苏达大学的 Gopher 服务器	224
10.6.3	启动明尼苏达大学的 Gopher	228
10.6.4	GN Gopher 服务器	230
10.7	安装 PPP 服务器	235
10.7.1	让调制解调器接电话	235
10.7.2	通过串行线显示登录提示符	235
10.7.3	提供 PPP 服务	236
10.7.4	可能出现的问题	237
10.8	传真打印机服务器配置	238
10.8.1	设置传真打印机服务器	238
10.8.2	使用传真打印机服务器	240
10.8.3	指定目标的传真号码	240
10.8.4	从网络上的 UNIX 机器使用该功能	240
10.9	WAIS 服务器	240
10.9.1	配置和安装 freeWAIS 源代码	243
10.9.2	创建索引	244

10.9.3	WAIS 资源	245
10.9.4	测试 WAIS 服务器	246
10.9.5	启动 freeWAIS	247
10.10	Linux 上的 Java	248
10.10.1	Java 开发软件包 JDK	248
10.10.2	JDK1.0.2	248
10.10.3	JDK1.1.1	249
10.10.4	Java applet	250

第 1 章 Linux 的获取与安装初步

1.1 Linux 的获取

1.1.1 发行版本介绍

通常人们所指的 Linux 是它的内核(kernel)和许多应用程序的集合, Linux 是一个非常庞大的系统。原则上, 用户可以从因特网或其他地方获取 Linux 的内核, 并且自己编译它。然后用同样的方式获取许多应用程序的源代码, 编译以后装入系统。但对于复杂的程序, 这样做不但费时, 而且容易出错。

为了省时省力, 用户通常从 Linux 的发行商处获取操作系统和应用程序。Linux 一些发行版本之间的区别在于它们的软件、协议、包装、安装和跟踪应用软件包, 以及安装方法和维护工具。

1.1.2 主要的发行版本

目前, 几种有影响的 Linux 发行版本有: RedHat、Slackware、Debian、Caldera 和 SuSE。

这些 Linux 系统的发行版本有基本的共性(例如它们的内核), 但也有一些方面不同, 下面将分别讲述这几个发行版本。

1.1.2.1 Slackware Linux

Slackware 是 Walnut Creek CDROM 公司发行的。可以从下面的站点获取 Slackware Linux :

`ftp://ftp.cdrom.com`

Slackware 曾经是市场上最重要的一种发行版本, 其他一些发行版本都是以它为基础, 经过再包装并提供其他的服务而形成的。

它的优点是文档比较完整。Linux 的许多文档都是以 Slackware 为默认平台来编写的, 包括安装、配置和维护等。

国内的用户大多选用 Slackware, 互相之间容易交流和学习。其缺点是软件的更新速

度相对较慢，也没有给用户提供很好的升级和更新的策略和工具。

1.1.2.2 RedHat Linux

RedHat Linux 是由 RedHat Software 公司发行的，目前的最新版本是 5.2。它基于 2.0.18 内核。

它的 RPM 技术，即 RedHat 的软件包管理系统，使之可以方便地安装和卸载应用程序和操作系统本身，包括内核和操作系统的基础部分。

安装它非常容易，在安装时就提供了 X Window 界面。该软件在国外非常流行。

有关 RedHat 的信息可以参考以下的站点：

<http://www.redhat.com>

尤其值得一提的是，最近，包括 Intel 和 IBM 在内的许多大公司都宣布支持 RedHat，并为 RedHat 提供技术和财力上的支持。国内大部分高校大多也是使用 RedHat Linux。

1.1.2.3 Debian GNU/Linux

Debian GNU/Linux 是 Linux 操作系统的一个发行版本及在其上运行的许多软件包的集合。

Debian GNU/Linux 是一群志愿者努力的结果，他们的目的是为了开发一个免费的、高质量的且与 UNIX 兼容的操作系统和一系列应用程序。

开发一个免费的与 UNIX 兼容的 operating 系统的想法源于 GNU 项目，Debian GNU/Linux 的许多应用程序也是由 GNU 项目开发的。

1.1.2.4 Caldera

Caldera 又称为 OpenLinux，也是 Linux 一个比较受欢迎的发行版本，安装情况与 RedHat 差不多。它与其他发行版本的不同之处是提供了一系列公用软件，包括商业性软件包，如 Office Suite 等。Caldera 目前最新版本是 OpenLinux-1.3。

1.1.2.5 SuSE

SuSE 也是 Linux 的一个发行版本，它同样采用 RPM 技术，它在安装时提供 Save/Restore 配置选项。SuSE 有两个方面与众不同，其一它对 Xfree86 的支持比其他的 Linux 软件都要好，其二它包括很多应用程序和工具软件。

SuSE 目前的最新版本是 SuSE-5.3。

1.2 Linux 的安装初步

本节将讲述如何获得及安装 Linux 操作系统，是 Linux 新手必读的一节，也是本书后面内容的基础。

1.2.1 Linux 简介

Linux 是为普通个人计算机设计的, 免费的、与 UNIX 兼容的操作系统。它是在 386 计算机上开发出来的, 它现在支持 486、586、Pentium、PowerPC、Sun Sparc 和 DEC Alpha 等的硬件。它包含并支持各种各样的软件, 如: X Window、Emacs、TCP/IP 网络(包括 SLIP)和许多应用程序。

与其他的操作系统相比较, Linux 系统的最突出的特色是它的高效性, 它对硬件的要求非常低, 一个 4MB 内存的 486 个人计算机, 如果安装了 Linux 系统, 它的性能就足以使用户满意了。

下面就开始讲述如何安装这个新崛起的操作系统。

1.2.2 安装准备

本节介绍安装 Linux 前要做的准备工作, 包括对硬件和硬盘的要求等。

1.2.2.1 硬件要求

运行 Linux 需要什么样的硬件系统? 这是一个很关键的问题。

系统的实际硬件要求会定期性地变化。有关这个问题最详细的回答可以从下面的站点得到:

<http://sunsite.unc.edu/mdw/HOWTO/Hardware-HOWTO.html>

这是一个 Linux 硬件指南的主页, 里面详细介绍了支持 Linux 系统的硬件环境。

从下面的站点也可以找到该问题的答案:

<http://sunsite.unc.edu/mdw/HOETO/INFO-SHEET.html>

本书仅列出一些常见的硬件环境。

对于 Intel 类的计算机, 要求下面的硬件配置:

任一 ISA、EISA 和 VESA 内部总线或 PCI 80386、80486、奔腾和 P6 系统都支持 Linux。IBM PS/2 的 MAC 结构只支持最新开发的内核 2.1x, 在初期还不能使用。

计算机上至少要有 4MB 内存。从技术上看, Linux 只要有 2MB 的内存就能运行, 但许多安装程序和软件都需要 4MB 内存。内存越大系统会运行得越快。如果用户打算使用 X Window, 建议把内存增加到 8MB 或 16MB。

当然, 用户还需要一个硬盘和一个 AT 标准的硬盘控制器。所有 MFM、RLL 和 IDE 硬盘及硬盘控制器都能在 Linux 系统下工作。Linux 也支持多种 SCSI 驱动器和适配器。如果打算从头开始组建一个 Linux 系统, 那么加一个 SCSI 驱动器将给系统带来更高的可靠性和运行效率。

还需要一块 3.5 英寸的软盘, 因为有的 Linux 系统在安装时需要从软盘启动。

还需要一个 MDA、Hercules、CGA、EGA、VGA 或 SVGA 显示卡。一般来说, 如果

显示卡能在 DOS 下工作,也能在 Linux 系统下工作。然而,如果想运行 X Window,显示器硬件的支持就会受到一些限制,当前 X Window 对显示卡的支持是有限的。

用户还需要一个 CDROM 驱动器,ATAPI、SCSI 或 IDE CDROM 都能够在 Linux 系统下工作。如果使用专用接口卡,当从软盘引导安装内核时,很可能找不到 CDROM。另外,连接在并行接口上的 CDROM 根本不能工作,必须将 CDROM 接到适当的串行接口上。

如果用户使用的声卡和以太网网卡是即插即用的,那么在安装时会出现问题。在 Linux 2.0.25 和以前版本的内核中还没有实现对即插即用卡的支持。

1.2.2.2 硬盘的要求

在硬盘上还需为 Linux 准备自由空间。所需空间的数量根据用户要安装的软件的多少决定。大多数情况下只要求在硬盘上有一块 40~80MB 的自由空间,这个自由空间包括软件空间、交换区空间及用户所需的自由空间等。

随着 Linux 的发展,它所需要的硬盘空间越来越大,因为在其上运行的软件越来越多。用户可以有选择地安装软件,从而节省硬盘空间。

可以在 10MB 或更少的硬盘空间上运行最小配置的 Linux 系统,也可以占用 500MB 或更多硬盘空间来存放 Linux 应用程序。所需的空间很大程度上取决于用户安装软件的数量及用户需要自由空间的多少。

Linux 可以和其他操作系统共存于硬盘上,如 DOS、Microsoft Windows 或 OS/2(实际上可以从 Linux 中存取 DOS 文件,运行 DOS 程序)。换句话说,当在硬盘上为 Linux 创建分区时,DOS 或 OS/2 仍然存在于它自己的分区上,且 Linux 也拥有自己的分区。在后面的章节中将详细介绍多重引导系统。

完全没有必要为使用 Linux 而运行 DOS、OS/2 或其他任何操作系统。Linux 是一个完全独立的自成一体的操作系统,不需要依赖其他操作系统来安装和使用。

总之,Linux 的最小安装空间并不比目前销售的 DOS 或微软公司的 Windows 3.1 的要求多,并且大大少于 Windows 95 的最小安装空间。

1.2.2.3 Linux 发行版本的选择

在安装 Linux 前,必须选中一种适合自己的计算机的 Linux 发行版本。没有单一的、标准的 Linux 发行版本——实际上有许多类似的发行版本。每一版本都有自己的文档和安装命令。

Linux 的发行版本可以通过匿名 FTP 和写信订购硬盘、磁盘或 CDROM 来获取。Linux 发行指南中有大量关于通过 FTP 和写信订购 Linux 发行软件的描述。从下面的地址可以找到该指南:

<http://sunsite.unc.edu/mdw/HOWTO/Distribution-HOWTO.html>

在 Linux 出现初期,这篇指南第一次写出来的时候(1992—1993 年),许多人是从因特网或 BBS 上把 Linux 下载到 DOS 机器上,然后把下载的 Linux 转到多张软盘上,其中一张将引导机器及安装其他十几张软盘。如果顺利(没有磁介质损坏),几小时后将成功安装

Linux, 否则白费功夫。

虽然现在可以更容易地得到 Linux 系统, 但下载的方法还是可行的。最简便的方法是购买高质量的商业化的 Linux 发行光盘, 如 RedHat、Craftworks、Linux Pro 或 WGS。只要花不到 50 美元就可从书店或计算机商店买到 Linux 系统的安装盘, 可以免除数小时的下载之苦。

从下面的站点, 用户可以下载任意一个 Linux 系统的发行版本:

<http://sunsite.unc.edu/Linux/distributions/INDEX.html>

用户也可以买可选型(optional)Linux 的 CDROM, 如 InfoMagic Linux 开发者资源荟萃。这些光盘包括了几种 Linux 的发行版本和主要的 Linux 查询站点中的内容, 如 Sunsite 和 txs-ll。

1.2.3 安装过程

下面是 Linux 的基本安装过程:

1. 收集硬件配置资料。
2. 制作安装软盘。
3. 如果想运行一个双重引导系统(Linux 和 DOS 或 Windows), 必须重新对硬盘分区, 为 Linux 准备出空间。
4. 为了读取 CDROM, 要从软盘上启动最小配置的 Linux。
5. 准备 Linux 的文件系统。
6. 从 CDROM 安装 Linux 基本部分。
7. 从 CDROM 启动 Linux。
8. (可选)从 CDROM 安装更多软件。

1.2.3.1 安装工具的基本组成

下面是安装 Linux 发行版工具的基本组成:

- README 和 FAQ 文件

Linux 安装在硬盘上后, 这些文件就存放在 CDROM 的顶层目录下, 并可以阅读(根据光盘的制作方法, 它们也许在 DOS 或 Windows 环境下可读)。为了解系统重大的更新和修改, 建议用户一进入 CDROM, 就读这些文件。

- 引导硬盘映像(常常在子目录中)

在光盘中包含要写到软盘上用来制作引导盘的文件。根据系统的硬件类型, 将挑选下面引导盘的映像之一。这里的问题是有些硬件驱动程序互相冲突但只要使用一个仅带所需激活的驱动程序的引导盘映像就可以很容易解决这个问题。不要试图去找出系统硬件的问题, 因为问题在于驱动程序, 而不在于系统硬件。

- ◇ 一个根硬盘映像(也许有两个)

这是一个要写到一块软盘上创建安装盘的文件。现在, 根盘一般是独立于硬件类型的。

◆ 一个救援硬盘映像

万一硬盘上的内核或引导块遭到破坏, 通过这个映像制作的软盘可以恢复系统到正常状态。

- **RAWRITE.EXE**

这个 DOS 程序将把一个文件(如引导或根硬盘映像)的内容直接写到软盘上, 不需格式化。如果要从 DOS 系统创建引导和根软盘, 只需要 RAWRITE.EXE。如果用户有带软驱的 UNIX 工作站, 可用 UNIX 中的命令“dd”或供应商提供的创建命令创建这些盘。用户可以查阅 dd 命令的联机帮助以获取更详细的信息。

- **CDROM**

使用引导盘的目的是让系统准备装入根盘或安装盘, 根盘或安装盘则是下一步准备硬盘并复制 CDROM 上的一部分到硬盘的设备。

1.2.3.2 安装步骤

1. 制作引导盘和根盘

Linux 的 CDROM 通常带有安装助手, 它将通过交互提示指导用户建立引导盘、根盘和救援盘。安装助手也许是一个 DOS 安装程序(如 RedHat 的 redhat.exe), 也许是 UNIX 的脚本文件, 或两者皆有, 具体的取决于 Linux 的发行版本。

运行这种安装程序进行安装, 将避免很多容易出错的手工输入的步骤。

首先, 选择一个适合硬件的引导硬盘映像。如果是手工完成, 一般会发现 CDROM 上的引导硬盘映像或者起了一个容易识别的名字, 或者有一个描述每一映像的索引文件。

其次, 必须用所选中的引导硬盘映像、根盘和救援硬盘映像制作安装盘, 用户可以使用 DOS 程序 RAWRITE.EXE 来完成这个操作。

再次, 必须有 2~3 张 DOS 下格式化的高密度软盘。它们必须是同一种型号的, 也就是说, 如果引导盘驱动器是 3.5 英寸的, 那么两张软盘都必须是 3.5 英寸的高密度软盘。用 RAWRITE.EXE 把引导盘和根硬盘映像写到这些软盘上。

使用该程序无需参数, 如下所示:

```
C: \ > RAWRITE
```

然后, 根据提示输入要写的文件和目标盘(如 A:)。RAWRITE 命令将直接将这些文件复制到几张软盘。同样用 RAWRITE 写根盘映像(如: 目标文件名是 COLOR144)到软盘。做完这些后, 就得到两张盘: 一张是引导盘, 一张是根盘。

要注意的是, 这两张盘在 DOS 下不再可读(从某种意义上说, 它们是“Linux 格式”的软盘)。

就像前面提到的, 除了上面的方法之外, 还可以在 UNIX 系统中用 dd(1)命令做同样的工作。当然, 需要一个带软驱的 UNIX 工作站。在软驱设备为/dev/rfd0 的 Sun 工作站上, 可以使用下面的命令:

```
$ dd if=bare of=/dev/rfd0 obs=18k
```

在某些工作站上(如 Sun 系列), 必须提供正确的输出块大小参数(obs 参数), 否则该命令将失败。如果有问题, 可以求助相应系统的 dd(1)使用指南。

确保使用崭新的、高质量的软盘，软盘上必须没有坏道。



注意： 安装 Linux，不一定要使用 Linux 或 DOS。然而，运行 DOS 或 Linux 能使人更容易地从光盘上制作引导盘和根盘。如果自己的机器上没有 Linux 或 DOS 操作系统，可以用别人的机器上的 Linux 或 DOS 操作系统，并从别人的机器上安装 Linux 系统。

2. DOS/Windows 硬盘重分区

在目前所使用的大多数系统中，硬盘已经全部用于 DOS 或 Windows 等操作系统的分区上。为了使用 Linux 系统，需要重新划分这些分区的大小。如果要运行双重引导系统，则参看后面的章节，在后面的章节中详细讲述了如何实现多重引导系统。



注意： 有些版本的 Linux 不会安装到 DOS 分区的目录中(有别于从一个 DOS 分区的安装)。然而，如果使用“UMSDOS 文件系统”，它将把 Linux 文件系统作为 DOS 分区上的一个目录看待。在这种情况下，就不必对硬盘重新分区了。

使用重分区方法是有条件的。如果硬盘已有四个分区，就不值得再分区，再分区则会带来更多的麻烦。

但在大多数情况下必须重新分区，正如这里所描述的一样。如果计划用 UMSDOS 文件系统，就需要读者亲自查找有关这方面的资料了——本书中没有关于这方面的详细讲述。本书假定用户没有使用 UMSDOS 文件系统，并将重新分区硬盘。

一个分区就是某一操作系统所使用的硬盘上的一部分。如果只安装了 DOS，硬盘也许只有一个分区，全部用于 DOS。然而，要使用 Linux，就必须重新分区硬盘，以便一个分区安装 DOS，另一个(或更多)分区安装 Linux。

分区以三种类型出现：

- 主分区
- 扩展分区
- 逻辑扩展分区

如果想在硬盘上拥有多于四个的分区，需要创建一个扩展分区，扩展分区能分出许多逻辑扩展分区。

另外，大多数人仅仅使用主分区。然而，如果需要多于四个的分区，就需要建立扩展分区。

请注意，可以很容易地把 Linux 安装在系统第二块硬盘上(如 DOS 的 D:)。在创建 Linux 分区时，只需指出正确的设备名称即可。下面将详细介绍。

现在，回到硬盘重分区这个问题上来。重新划分分区大小所带来的问题是：如果不删除这些已经存在的分区上的数据，就无法(容易地)分区。因此，在重新分区之前，应该完全备份系统。为重新划分一个分区的容量，应先删除这个分区，然后重新创建一些小容量的分区。



注意： 对于 DOS 而言，可用一种叫 FIPS 的非破坏性硬盘重分区工具。在下面的站点中可以获取该工具：

<http://sunsite.unc.edu/pub/Linux/system/Install>

有了 FIPS 硬盘优化器(如 Norton 硬盘加速器),就应该在不破坏 DOS 分区上数据的情况下,对它重新划分大小。建议在操作之前,全面备份系统。

然而,如果没有 FIPS,典型的方法是用 fdisk 程序去修改分区。例如,假定用户有 80MB 的硬盘,全部用于 DOS,那最好把它分成两半——40MB 用于 DOS、40MB 用于 Linux。

为了分割硬盘,需要在 DOS 环境下运行 fdisk 命令,删除 80MB 的 DOS 分区,然后,就可以格式化这个分区并从备份中重新恢复 DOS 系统。

总之,应该用 fdisk 按下面的步骤重新划分 DOS 分区:

- ① 完全备份系统。
- ② 制作 DOS 的启动盘,使用命令如: `FORMAT A: /S`
- ③ 复制 `FDISK.EXE` 和 `FORMAT.COM` 到这张软盘上,同时复制其他需要的实用工具(例如:备份恢复工具)到这张软盘上。
- ④ 用这张 DOS 系统盘启动系统。
- ⑤ 运行 fdisk,注明要修改的硬盘(如 C: 或 D:)
- ⑥ 使用 fdisk 的菜单选项删除要改变大小的分区,这将破坏受影响分区上的数据。
- ⑦ 通过 fdisk 的菜单选项用更小的容量重建这些分区。
- ⑧ 退出 fdisk,用 `FORMAT` 命令重新格式化新的分区。
- ⑨ 从备份中恢复数据。

3. 建立 Linux 分区

重新分区硬盘后,需要为 Linux 建立分区。在介绍如何建立分区前,先讨论 Linux 下的分区和文件系统。

Linux 至少需要一个分区,由根文件系统使用。根文件系统将容纳 Linux 内核和其应用软件。

可以把一个文件系统当作 Linux 格式化的分区。文件系统用来存放文件。每一个操作系统都必须有一个根文件系统。然而,许多用户喜欢用多文件系统,其中一个文件系统用于容纳目录树的全部主要分支。举例来说,可以建立一个文件系统,用于安装 `/usr` 目录下的所有文件。这时就有一个根文件系统和一个 `/usr` 文件系统。



注意: UNIX 系统中用正斜杠 “/” 分隔目录名,而不像 DOS 用反斜杠。

每一个文件系统都需要拥有自己的分区。因此,如果要使用根文件系统和 `/usr` 文件系统,必须创建两个 Linux 分区。

另外,大多数用户建立了一个用作虚拟内存的交换分区。假定计算机有 4MB 内存和 10MB 交换分区,就 Linux 而言,就拥有了 14MB 的虚拟内存。也就是说,虚拟内存的容量是物理内存的容量与交换分区容量之和。

当使用交换空间时,Linux 把暂时不用的内存页面转移到硬盘上,这样一来,就可以在系统上运行更多的程序。但是,与物理内存相比,因为内存页面交换常常很慢,所以,交换分区并非实际物理内存的替代品。如果没有足够的物理内存,那么需要大量内存的程序(如 X Window)就要依靠交换分区空间获得内存。

几乎所有的 Linux 用户都使用了交换分区。如果只有 4MB 或更少内存,就要用交换

分区来安装 Linux。建议无论如何要用一个交换分区，除非有大量的物理内存。

交换分区的大小依据需要多大的虚拟内存而定。建议一般至少有 16MB 的虚拟内存。因此，如果有 8MB 的物理内存，只需 8MB 的交换分区。



注意： 每一交换分区在容量上不能超过 128MB。因此，如果内存需要大于 128MB，需要建几个交换分区。总共可建 16 个交换分区。

1.2.3.3 引导安装盘的使用

引导安装盘使用的第一步是用引导盘启动系统。

引导内核提示符将等待用户的输入，如果用户不做任何输入，10 秒钟后系统将自动启动内核。

在 Linux 启动内核前，通过在内核名后加参数，可以指明各种各样的硬件参数，如 SCSI 硬盘控制器的端口地址和 IRQ，或驱动器的几何参数。如果 Linux 不能检测到 SCSI 控制器和硬盘几何参数，就有必要这样做。特别是许多无 BIOS 的 SCSI 硬盘控制器需要在引导时指出端口地址和 IRQ。诸如 IBM PS/1、Thinkpad 和 ValuePoint 之类的计算机都不把硬盘参数存放在 CMOS 中，必须在引导时说明。

注意系统引导时的提示信息。它们将列出和描述安装 Linux 检测到的硬件。特别需要说明的是，如果装了 SCSI 硬盘控制器，应该看到 SCSI 控制器被检测到的列表。如果看到如下所示的信息：

```
SCSI: 0 hosts
```

就说明 SCSI 设备没有被检测到，用户将不得不考虑如何把 SCSI 的位置告诉内核。

另外，系统会显示硬盘分区和检测到的设备的信息。如果其中任何信息不正确或丢失，安装程序将不得不强制进行硬件检测。

如果一切正常并且硬件都检测到了，用户就可以跳到下一节“根盘的使用”。

要强制进行硬件检测，必须在引导提示符处输入正确的参数，格式如下：

```
Linux <parameters...>
```

下面列出了一些最常用的参数供用户使用。新式的 Linux 引导盘在启动前会给出选项让用户查看内核参数说明帮助。

- **hd=**柱面数，磁头数，扇区数

指定硬盘几何参数。只在 IBM PS/1、ValuePoint 和 ThinkPad 这样的系统上需要。如果硬盘有 683 个柱面，16 个磁头，每道 32 扇区。那么应输入：

```
Linux hd=683, 16, 32
```

- **tmc8xx=**端口地址，IRQ

如果用户使用的硬盘控制器是 Future Domain 公司的 TMC-8XX 系列 SCSI 硬盘控制器，需要指定这个选项，因为这种硬盘控制器没有 BIOS 接口。在这个选项中，用户需要指定端口地址和 IRQ。例如：

```
Linux tmc8xx=0xca000, 5
```



注意： 前缀 0x 必须用于以 16 进制表示的数。这一点对下面所有的选项均成立。

- st0x=内存地址, IRQ
如果用户使用的硬盘控制器是 Seagate 公司的 ST02 系列, 需要指定这个参数, 因为这种硬盘控制器不使用 BIOS 接口。
- t128=内存地址, IRQ
如果用户使用的硬盘控制器是 Trantor 公司的 T128b 系列, 需要指定这个参数, 因为这种硬盘控制器不使用 BIOS 接口。
- ncr5380=端口, IRQ, dma
如果是普通 NCR5380 系列硬盘控制器, 用户需要指定端口、IRQ 和 DMA 通道的参数值。
- aha152x=端口, IRQ, scsi_id
如果是没有 BIOS 接口的 ACI-6260 系列硬盘控制器, 用户需要指定端口 IRQ 和 SCSI ID 的参数值。

要详细了解这些选项的有关信息, 请阅读 SCSI 指南, 该指南可在任何 Linux FTP 查询站点都可得到。SCSI 指南还解释了 Linux SCSI 的兼容性。

1.2.3.4 根盘的使用

用引导盘启动之后, 系统会提示用户插入根盘。此时, 应该拿走引导盘插入根盘, 然后按回车键继续安装。

这里实际进行的操作是: 引导盘使用了最小内核的操作系统(因为硬盘还未准备好), 该系统使用部分内存作为虚拟硬盘(也称为“内存硬盘”)。根盘加载了一小组文件和安装工具到内存硬盘上, 根盘将通过这些文件和程序来从 CDROM 上安装一种 Linux 产品。

1. EGA 和 X Window 的安装选择

此时, 稍微老一点的 Linux 版本(包括 Slackware)提供一个 Shell 脚本程序, 要求用户手工地按指定顺序输入安装命令。更新的 Linux 产品是以运行一个鉴别适配器的程序开始的, 这些程序将以交互的方式带用户执行这几步, 并提供大量的帮助信息。

如果选择以 X Window 方式安装的方法, 在正确安装之前, 安装程序会向用户询问鼠标和监视器的类型。Linux 系统安装好后, 这些设置将进行保存。可以以后再调整显示器的显示性能, 因此在这一阶段, 设置为基本的 640X480 SVGA 模式是可以的。

虽然不一定要安装 X Window, 但是许多人发现图形界面更容易使用。无论怎样, 将来会用到 X Window, 不妨安装它。

程序提示将带用户开始必要的几步, 包括建立最初用户帐号, 从 CDROM 上安装软件包。

接下来, 介绍在安装过程中比较复杂的部分。这将帮助用户理解安装程序正在做什么, 以及为什么那么做。

2. 使用 fdisk 和 cfdisk 工具

根盘上的 Linux 启动后, 安装过程将进入到建立或编辑刚才通过 fdisk 设置了的分区。为建立或编辑 Linux 硬盘分区, 将使用 Linux 版本的 fdisk 程序, 或它的同类程序 cfdisk。

一般情况下,安装程序将搜索已经安装好的 fdisk 或 cfdisk,两者中 cfdisk 更容易使用;但它目前的版本还不允许创建新的分区。

因此,将 Linux 系统安装到新的硬盘上的时候,用户必须先用 fdisk 产生一个 cfdisk 能处理的分区。试着运行 cfdisk,如果不行,再运行 fdisk。

fdisk 和 cfdisk 都适用的一些注意事项:两者都带一个参数,即要在上面建立 Linux 分区的硬盘驱动器的名称。硬盘驱动器设备的名称为:

```
/dev/hda 第一 IDE 硬盘
/dev/hdb 第二 IDE 硬盘
/dev/sda 第一 SCSI 硬盘
/dev/sdb 第二 SCSI 硬盘
```

如果要在系统的第一 SCSI 硬盘上建立 Linux 分区,要使用如下的命令:

```
cfdisk /dev/sda
```

如果不带参数,fdisk 或 cfdisk 将默认为/dev/hda 设备。

要在第二硬盘上建立分区,只要在 fdisk 运行时指明硬盘驱动器设备是/dev/hdb(如果使用 IDE 硬盘),或/dev/sdb(如果使用 SCSI 硬盘)就行。

Linux 分区不必全都在同一硬盘上。例如,可以在/dev/hda 上建立根系统文件分区,而在/dev/hdb 设备上建立交换分区。如要这么做,只要对每个盘各运行一次 fdisk 或 cfdisk 就可以了。

在 Linux 中,分区被赋予一个基于它所属硬盘的名字。如:驱动器/dev/hda 上的第一个分区叫/dev/hda1,第二个分区叫/dev/hda2 等。



注意: 不要用 Linux 的 fdisk 或 cfdisk 建立或删除除 Linux 以外的操作系统所用的分区。也就是说,不要用 Linux 版本的 fdisk 或 cfdisk 建立或删除 DOS 分区,而要用 DOS 版本的 fdisk。如果用户试图用 Linux fdisk 建立 DOS 分区,则 DOS 的分区将不能正确引导。

下面是一个 fdisk 的应用实例。一个单一的 DOS 分区用 61 693 块,剩下的用于 Linux(在 Linux 中,1 块=1 024 字节,因此,61 693 块=61MB)。

在这个例子中只建立 2 个分区:交换区和根区。还可以按上面的建议扩展到 4 个 Linux 分区:交换区、根文件系统、系统软件区及 home 目录区。

首先,用 fdisk 的“p”命令显示当前分区表内容。由下面的输出可以看见,/dev/hda1(在/dev/hda 设备上的第一个分区)上含有一个大小为 61 693 块的 DOS 分区:

```
Command(m for help):  p
Disk /dev/hda: 16 heads, 38 sectors, 683 cylinders
Units = cylinders of 608 * 512 bytes

Device    Boot  Begin    Start    End  Blocks   Id  System
/dev/hda1  *      1         1      203   61693    6   DOS 16-bit >=32M
```

Command (m for help):

接下来, 用 “n” 命令建立新分区。Linux 根分区容量将是 80MB:

```
Command (m for help):  n
Command action
e   extended
p   primary partition (1-4)
p
```

这里会碰到这样的问题, 是建立一个扩展分区还是主分区? 大多数情况下, 只用到主分区, 除非要 在一个硬盘上用到 4 个以上的分区。以下是在创建一个新分区时的过程:

```
Partition number (1-4):  2
First cylinder (204-683):  204
Last cylinder or +size or +sizeM or +sizeK (204-683):  +80M
```

第一个柱面应该从上一个分区结束后的那一个柱面开始。这里, /dev/hda1 是在柱面 203 结束, 因此, 新的分区应该从柱面 204 开始。

可以看见, 如果标注 “+80M”, 就指定了容量为 80MB 的一个分区。同样, 标注 “+80K” 意味规定了容量为 80KB 的分区, “+80” 意味容量为 80B 的分区。

警告: Linux 目前的分区长度不能超过 33 090 个扇区。如果见到这一警告, 可以忽略它。这种限制是从只能用 64MB 硬盘的 Linux 文件系统中遗留下来的。如果用新的文件系统, 就不再是这种情况, 现在分区容量可高达 4TB。

接下来, 建立 10MB 的交换分区, 安装程序将把这部分分区存放在设备 /dev/hda3 上:

```
Command (m for help):  n
Command action
e   extended
p   primary partition (1-4)
p

Partition number (1-4):  3
First cylinder (474-683):  474
Last cylinder or +size or +sizeM or +sizeK (474-683):  +10M
```

下面, 需要显示硬盘上分区表中的信息, 命令如下:

```
Command (m for help):  p
Disk /dev/hda: 16 heads, 38 sectors, 683 cylinders
Units = cylinders of 608 * 512 bytes
```

Device	Boot	Begin	Start	End	Blocks	Id	System
/dev/hda1	*	1	1	203	61693	6	DOS 16-bit >=32M
/dev/hda2		204	204	473	82080	83	Linux native
/dev/hda3		474	474	507	10336	83	Linux native

应该注意到, Linux 的交换分区(这里是 /dev/hda3)类型为 “Linux native”。必须把交

换分区类型更改为“Linux swap”，以便安装程序能够识别。用 fdisk 中的“t”命令可以改变分区的类型，命令如下：

```
Command (m for help): t
Partition number (1-4): 3
Hex code (type L to list codes): 82
```

使用“L”命令将列出类型代码，可见 82 代表 Linux 交换分区类型。用“w”命令退出 fdisk，并将修改内容保存在分区表上。用“q”命令将不存盘而退出 fdisk。

退出 fdisk 后，系统会提示用户重新启动以保证修改生效。一般来说，使用 fdisk 后不必重新启动系统，因为新式的 fdisk 或 cfdisk 足够智能化，无需重新启动就能更新分区。

1.2.3.5 安装软件包

完成硬盘分区后，剩下的安装几乎就是自动的。

安装程序(无论是 EGA 还是基于 X Window 的显示器)将指导用户通过一系列菜单，这些菜单允许用户指定要安装的 CDROM 和要使用的分区等等。

在这里不讲述有关这一安装阶段的细节。这一阶段是大多数 Linux 发行版本之间差别最大的部分之一，但也是最简单的部分。此外，安装程序中每一步骤都会显示该步骤进行的操作内容，并带有良好的屏幕帮助。

1.2.3.6 安装的最后步骤

安装完成后，如果一切顺利，在首次从硬盘启动前，安装程序将提示用户完成一些系统的配置，包括 LILO 的配置和是否制作启动盘。

● LILO

LILO(代表 Linux 引导程序)是一个允许用户从硬盘启动 Linux(同样适用于其他操作系统，如 DOS)的程序，用户可能会见到安装 LILO 到硬盘上的选项。如果用户不运行 OS/2，就回答“是”。OS/2 有特殊的要求，参见下面“自选 LILO 配置”部分。

把 LILO 安装为主引导程序会使得单独的引导软盘不再必要，即用户不必再通过引导软盘来启动系统，而可以在每次启动系统时告诉 LILO 选择引导哪一个系统。

● 制作启动盘(可选)

在这一步中，用户将选择是否制作启动盘，以便用它来引导新的 Linux 系统。

为此，需要一张空白的高密度 DOS 格式的软盘，用它来启动系统。很简单，当系统提示时插入软盘，一个引导软盘就能制作出来。这张盘不同于引导安装盘，这两张盘不能相互替代。

其余各种系统配置安装的过程还会配置调制解调器、鼠标以及时区等。请根据菜单的指示进行操作。

系统也会提示建立用户帐号和设置根(系统管理)帐号口令。这并不复杂，用户只要按照屏幕说明操作即可。

1.2.3.7 启动新系统

如果一切按计划进行,现在用户能使用 LILO 的硬盘启动 Linux 系统。另外,也应该能从 Linux 引导软盘启动。(不是最初的引导安装盘,而是安装软件后制作的那一张引导软盘)。启动后,以根(root)身份登录。这时,用户就拥有了自己的 Linux 系统。

如果用 LILO 引导,在启动期间,试按下 Shift 键或 Ctrl 键,将显示一个引导提示符;按 Tab 键可以查看选项列表。用户能够根据显示列表选择从 LILO 直接启动 Linux、DOS 或其他的安装在同一块硬盘上的操作系统。

首次启动后,用户应该看见刚从硬盘启动的新的 Linux 的登录提示。

1.2.3.8 系统管理初步

根据安装进度情况,在这阶段,用户可以建立帐号、更改主机名或重新配置 X Window。用户能安装和配置许多内容,包括备份设备、设置网络服务、通过 SLIP/PPP 连接到因特网服务供应商等。

一本好的 UNIX 系统管理参考书对用户会大有帮助。用户可以阅读其他的 Linux 实用指南(HOWTO),如:网络指南(NET-2-HOWTO)和打印指南(Printing-HOWTO),以得到关于其他配置操作的信息,这些信息可以在目录/usr/doc 下找到。

1.2.3.9 自选LILO配置

LILO 是能在系统启动时用来选择 Linux、DOS 或其他操作系统之一的引导程序。这种是用户的 Linux 在安装过程中自动配置 LILO 的结果。

如果把 LILO 安装为主引导程序,它将会为硬盘上所有的操作系统处理引导步骤。如果 DOS 是用户唯一安装的其他操作系统,这一切将顺利进行。但是,如果硬盘上还存在自己引导管理器的 OS/2,用户应该把 OS/2 的引导管理器作为主引导程序,只用 LILO 引导 Linux(作为第二引导程序)。

对 EIDE 系统的一个重要的限制是:由于 BIOS 的限制,任何操作系统的引导扇区必须存在于最先两个物理硬盘之一上,即操作系统的引导扇区不能存放在第三块硬盘或第四块硬盘上。否则,无论用户从哪里启动,LILO 都将在显示“LILO:”提示符后挂起来。

如果用户不得不手工配置 LILO,将涉及到 LILO 的配置文件: /etc/lilo.conf。

下面提供了一个 LILO 配置实例,这里 Linux 分区在/dev/hda2(第一个硬盘)上,DOS 安装在/dev/hdb1(第二个硬盘)上。

```
# Tell LILO to install itself as the primary boot loader on /dev/hda.

boot = /dev/hda

# The boot image to install; you probably shouldn't change this

install = /boot/boot.b

# The stanza for booting Linux.
```

```
image = /vmlinuz      # The kernel is in /vmlinuz
label = Linux          # Give it the name "Linux"
root = /dev/hda2       # Use /dev/hda2 as the root filesystem
vga = ask              # Prompt for VGA mode
append = "aha152x=0x340, 11, 7, 1" # Add this to the boot options,
                                   # for detecting the SCSI controller

# The stanza for booting DOS

other = /dev/hdb1      # This is the DOS partition
label = DOS            # Give it the name "DOS"
table = /dev/hdb       # The partition table for the second drive
```

用户编辑了/etc/lilo.conf 文件后,就可以以根(root)身份运行/sbin/lilo 命令。这会把 LILO 安装在硬盘上。



注意: 为了正确使用引导程序,当用户重编译内核后,必须重运行/sbin/lilo 命令。

记住,在/etc/lilo.conf 文件中用追加选项“append”指定引导参数的操作与启动安装引导盘时是一样的。

现在可以从硬盘重新启动系统了。在默认方式下,LILO 将引导配置文件中列在第一项的操作系统(在本例中是 Linux)。为了弹出引导菜单选择另一个操作系统,在系统启动时按住 Shift 键或 Ctrl 键,用户应该看到如下提示符:

Boot:

这里,或者输入要引导的操作系统的名字(在配置文件中的 label 行给出名字:本例中是 Linux 或 DOS),或者按 Tab 键列出可选的列表。

现在假设要用 LILO 作第二引导加载程序。例如,要从 OS/2 的引导管理器引导 Linux 系统。但是,要从 OS/2 启动 Linux 分区,就得用 OS/2 的 fdisk(而非 Linux 的 fdisk)建立这个分区,并按 FAT 文件系统或 HPFS 文件系统格式化该分区,以便 OS/2 能识别它。

如果要从 OS/2 引导管理器允许让 LILO 启动 Linux,则把 LILO 安装在 Linux 根文件系统(上例为/dev/hda2)下即可。此时,对应的 LILO 配置文件应该如下所示:

```
boot = /dev/hda2
install = /boot/boot.b
compact

image = /vmlinuz
label = Linux
root = /dev/hda2
vga = ask
```

注意引导行的变化。

在运行/sbin/lilo 命令后，就会把 Linux 分区信息加到主引导记录 MBR 中。

1.3 Linux 和 FreeBSD

谈到 Linux，就不能不提到 FreeBSD，它们在许多方面具有相似的特性，本节就对这两个系统做一比较。

1.3.1 FreeBSD 简介

FreeBSD 和 Linux 一样是免费的类 UNIX 操作系统。最主要的差异是，Linux 是从头到尾重新发展的，而 FreeBSD 是基于 4.4BSD(美国伯克利大学软件发行部)自由发行的那一个部分而来。这个部分也被称为 4.4BSD-lite。

这也许会使人这么想：FreeBSD 比 Linux 更近于“真正”的 UNIX。但是，FreeBSD 只能在 Intel PC 平台 (i386 或更高)上执行，而移植到 DEC Alpha 平台的版本目前还没有实现。

FreeBSD 对硬件的要求与 Linux 非常相似。

FreeBSD 的发展与 Linux 比较起来是比较封闭的，主要因为它是由一个开发人员成立的中心团队负责。大的改变先在电子邮件列表上事先做讨论，讨论通过后才能做相应的改动，这与 Linux 的发展形成了鲜明的对比。

FreeBSD 有两个发展树：“-current”和“-stable”。在“-current”发展树中主要是新功能的发展，而“-stable”发展树中主要都是已经除去错误的系统。

FreeBSD 如同 Linux 一样，可自由免费地使用与再次分发。系统的大部分软件是在 BSD 版权下发行，而其他的则是在 GNU GPL 或 GNU LGPL 下发行。

可以在 FreeBSD Inc.1 下找到更多的有关信息或下载整个系统。最新“-stable”的版本和“-current”开发程序代码可以在 Walnut Creek CDROM2 购得。

1.3.2 FreeBSD 表示硬盘分区的方式

Linux 和 FreeBSD 使用两种不同的方式来表示硬盘分区。

1.3.2.1 FreeBSD的片和分区概念

FreeBSD 只需要用硬盘上的一个主硬盘分区。这个主分区在 FreeBSD 的术语中被称为“片”(slice)。然后它用 disklabel 这个程序在这个主硬盘分区中创建逻辑分区。这些逻辑分区在 FreeBSD 的术语中被称为“分区”(partitions)。

注意，Linux fdisk 程序的主分区表中不会将 FreeBSD 片中的分区显示出来，但是可

以使用命令“b”显示 FreeBSD 硬盘卷标的信息。它的输出将如下所示(/dev/hda4 是 FreeBSD 的片):

```
# fdisk /dev/hda

Command (m for help): p
Disk /dev/hda: 64 heads, 63 sectors, 621 cylinders
Units = cylinders of 4032 * 512 bytes
evice Boot Begin Start End Blocks Id System
/dev/hda1 * 1 1 27 54400+ 83 Linux native
/dev/hda2 28 28 55 56448 83 Linux native
/dev/hda3 56 56 403 701568 83 Linux native
/dev/hda4 404 404 621 439488 a5 BSD/386
Command (m for help): b
Reading disklabel of /dev/hda4 at sector 1624897.
BSD disklabel command (m for help): p
8 partitions:
# size offset fstype [fsize bsize cpgr]
a: 64512 1624896 4.2BSD 0 0 # (Cyl. 404 - 419)
b: 104832 1689408 swap # (Cyl. 420 - 445)
c: 878976 1624896 unused 0 0 # (Cyl. 404 - 621)
e: 64512 1794240 4.2BSD 0 0 # (Cyl. 446 - 461)
f: 645120 1858752 4.2BSD 0 0 # (Cyl. 462 - 621)
BSD disklabel command (m for help): q
#
```

字母“b”指的是交换分区(swap partition), 而“c”指的是整个片。

1.3.2.2 Linux和FreeBSD中的硬盘、片与分区表示方法

Linux 和 FreeBSD 中的硬盘以如下方法表示:

	Linux	FreeBSD
第一个 IDE 驱动器	/dev/had	/dev/wd0
第二个 IDE 驱动器	/dev/hdb	/dev/wd1
第一个 SCSI 驱动器	/dev/sda	/dev/sd0
第二个 SCSI 驱动器	/dev/sdb	/dev/sd1

FreeBSD 片以如下方法表示(以/dev/hda 为例):

	Linux	FreeBSD
第一个主分区	/dev/hda1	/dev/wd0s1
第二个主分区	/dev/hda2	/dev/wd0s2
第三个主分区	/dev/hda3	/dev/wd0s3
第四个主分区	/dev/hda4	/dev/wd0s4

FreeBSD 片中的分区以如下方法表示(/dev/hda4 是例子中的 FreeBSD 片):

Linux	FreeBSD	FreeBSD 的默认安装点
-------	---------	----------------

```

/dev/hda5      /dev/wd0s4a  /
/dev/hda6      /dev/wd0s4b  swap
/dev/hda7      /dev/wd0s4e  /var
/dev/hda8      /dev/wd0s4f  /usr

```

如果在 Linux 中执行 `dmesg` 命令，将会看到如下的输出：

```

Partition check:
hda: hda1 hda2 hda3 hda4 < hda5 hda6 hda7 hda8 >

```

如果在 `/dev/sdb3` 片中安装 FreeBSD，而且 `/dev/sdb2` 是一个含有两个逻辑分区的 Linux 扩展分区(`/dev/sdb5` 及 `/dev/sdb6`)，上面例子的结果看起来将如下所示：

Linux	FreeBSD	FreeBSD 的默认安装点
<code>/dev/sdb7</code>	<code>/dev/sd1s3a</code>	<code>/</code>
<code>/dev/sdb8</code>	<code>/dev/sd1s3b</code>	<code>swap</code>
<code>/dev/sdb9</code>	<code>/dev/sd1s3e</code>	<code>/var</code>
<code>/dev/sdb10</code>	<code>/dev/sd1s3f</code>	<code>/usr</code>

`dmesg` 命令的输出将如下所示：

```

Partition check:
sdb: sdb1 sdb2 < sdb5 sdb6 > sdb3 < sdb7 sdb8 sdb9 sdb10 >

```

如果有一个 Linux 扩展分区在 FreeBSD 片之后，就会出现这个问题。因为大多数的 Linux 内核没有内含 UFS 文件系统的支持，它们不能识别 FreeBSD 片中的分区。看到的(其中 `/dev/hda3` 是 FreeBSD 片，而 `/dev/hda4` 是 Linux 扩展分区)将是由：

```

Partition check:
hda: hda1 hda2 hda3 < hda5 hda6 hda7 hda8 > hda4 < hda9 hda10 >

```

变成：

```

Partition check:
hda: hda1 hda2 hda3 hda4 < hda5 hda6 >

```

这会导致错误的设备指定，并会使数据丢失。



注意： 一定要把 FreeBSD 片放在任何 Linux 扩展分区之后，并且不要在安装 FreeBSD 后改变 Linux 扩展分区中的任何逻辑分区。

1.3.3 Linux 与 FreeBSD 分享交换区空间

本节介绍如何让 Linux 和 FreeBSD 共用一个交换分区。除本书所述的方法之外，可能还有别的方法可以做到这一点。这里所用的 Linux 是 RedHat 4.1 和 4.2(Linux 内核 2.0.29 和 2.0.30)与 FreeBSD 2.2.2。可以在安装 Linux 之前先安装 FreeBSD，但要注意 FreeBSD 片中分区的顺序。

1.3.3.1 安装和准备Linux

Linux 的安装与前面所讲的没什么不同,只是必须在硬盘中留下给 FreeBSD 片的空间。

不用创建 Linux 的交换分区,但如果要这样一个分区,就把它放在要留给 FreeBSD 的空间中。

在完成 Linux 的安装后,必须重新编译内核。必须加入对 UFS 文件系统的支持和对 BSD disklabel(FreeBSD 分区表)的支持。在重新编译内核时,应按如下的方法配置:

```
UFS filesystem support (read only) (CONFIG_UFS_FS) [N/y/m/?] y
BSDdisklabel (FreeBSD partition tables) support (CONFIG_BSD_DISKLABEL) [N/y/?]
(NEW) y
```

安装新的内核并重新启动。如果已经创建了 Linux 交换分区,则把/etc/fstab 文件中所有含有 swap 的那几行删去。

1.3.3.2 安装FreeBSD

依照 FreeBSD 文件中所描述的方式来安装 FreeBSD。如果已经创建了 Linux 交换分区,把它删除(可以使用 FreeBSD 的 fdisk 程序完成删除工作),但注意 FreeBSD 片中分区的顺序。如果使用默认的卷标编码,第二个分区将是交换分区。完成 FreeBSD 的安装,并使用 Linux 的启动盘重新启动进入 Linux。

1.3.3.3 在Linux中设置FreeBSD的交换分区

在进入 Linux 后执行 dmesg 命令,它的输出类似这样:

```
Partition check:
hda: hda1 hda2 hda3 hda4 < hda5 hda6 hda7 hda8 >
```

这表示/dev/hda4 是 FreeBSD 片,而/dev/hda5、/dev/hda6、/dev/hda7 和/dev/hda8 是 FreeBSD 分区。如果交换分区是在片中的第二个分区,那么它对应的设备文件将会是 /dev/hda6。必须把下面的这一行放到/etc/fstab 文件中来打开交换分区的功能:

```
/dev/hda6 none swap sw 0 0
```

FreeBSD 可以用任何类型的分区来当作交换分区,但 Linux 的交换分区需要特别的签名。这个签名是用 mkswap 命令来创建的。FreeBSD 使用这个共用的交换分区时会对这个签名造成破坏,所以必须在每次进入 Linux 时执行 mkswap 命令。

要让这个程序自动执行,必须找出执行 swapon 的脚本并在启动时执行。对 RedHat 而言,这个脚本文件是/etc/rc.d/rc.sysinit。把下面这一行放到 swapon -a 前:

```
awk -- '/swap/ && ($1 !~ /#/)' { system("mkswap \"$1\"") }' /etc/fstab
```

这将会在每次启动时,在/etc/fstab 文件中的任何交换分区上执行 mkswap 命令,除非把它取消(使用"#"作为该行的第一个字符)。

在重新启动进入 Linux 后,使用 free 命令来检查交换区空间的大小。

此外,还必须重新启动进入 FreeBSD 来确认一切都如预期的那样运行。如果没有按

预期运行, 那么可能是使用了错误的分区来当作交换分区。唯一的解决办法便是重新安装 FreeBSD 后再试一次。

1.3.3.4 使用LILO来启动FreeBSD

可以非常容易地使用 LILO 来启动 FreeBSD。如果要使用 LILO, 不要安装 FreeBSD 的引导程序。在/etc/lilo.conf 文件中加入下列这几行, 并执行 lilo 命令(假设 FreeBSD 片在 /dev/hda4 设备):

```
other=/dev/hda4
table=/dev/hda
label=FreeBSD
```

如果把 FreeBSD 安装在第二块 SCSI 硬盘中, 使用如下的设置(假设 FreeBSD 片在 /dev/sdb2 设备):

```
other=/dev/sdb2
table=/dev/sdb
loader=/boot/chain.b
label=FreeBSD
```

1.3.4 创建文件系统

1.3.4.1 在Linux中创建UFS文件系统

很不幸, Linux 内核对 UFS 文件系统的支持有问题。当创建 UFS 文件系统时, 只会得到一些错误信息, 事实上 UFS 文件系统已经被创建了, 但却不能读到任何内容。

所幸 Linux 2.0.xx 内核(xx<30)有一个新的(Alpha)版本, 这个版本的 Linux 内核具有很好的 UFS 文件系统支持, 在 SunSite3 站点可以获取该内核文件。该文件被称为 U2FS, 这个文件目前的版本是 u2fs-0.4.3.tar.gz。U2FS 用如下的方式来安装(假设 u2fs-0.4.3.tar.gz 存放在/usr/src 目录下):

```
cd /usr/src
tar xvzf u2fs-0.4.3.tar.gz
patch -p0 -E < u2fs-0.4.3.patch
```

现在必须重新建立支持 U2FS 文件系统和 BSD disklabel 的内核。当使用 U2FS 时, 可以忽略内核的 UFS 文件系统支持。

安装完新的内核后, 可以使用如下的命令来装载 UFS 文件系统。

```
mount -t u2fs /dev/hda8 /mnt
```

这个 UFS 文件系统是只读的。那就是说, 可以从 UFS 文件系统中读取数据, 但不能对其写入数据。

1.3.4.2 在FreeBSD上创建ext2fs文件系统

要在 FreeBSD 中创建 ext2fs 文件系统，要先建立一个支持 ext2fs 文件系统的内核。在新内核的设置文件中写入如下的内容：

```
options "EXT2FS"
```

在使用新的内核启动后，可以用如下的命令来装载 ext2fs 文件系统：

```
mount -t ext2fs /dev/wd0s3 /mnt
```



注意： 不能把 ext2fs 文件系统创建在 FreeBSD 的扩展分区中。

由于 FreeBSD 的某些错误，必须在关闭 FreeBSD 前卸下所有的 ext2fs 文件系统。如果在 ext2fs 文件系统未卸下前就关闭 FreeBSD，FreeBSD 便不能同步化 UFS 文件系统，这会导致 FreeBSD 下次启动时文件系统检查程序 fsck 被执行。

1.3.5 程序的可移植性

1.3.5.1 在Linux下执行FreeBSD的可执行文件

iBCS 软件包提供了在 Linux 下执行 FreeBSD 可执行程序的支持。

用户可以安装该软件包，在安装该软件包之后，Linux 系统就可以直接执行 FreeBSD 的可执行代码。

1.3.5.2 在FreeBSD下执行Linux的可执行文件

FreeBSD 能够执行 Linux 的可执行文件，包括 a.out 和 ELF 格式。

要在 FreeBSD 下执行 Linux 的可执行文件，必须进行下列操作：

1. 打开与 Linux 相兼容的选项。需要修改/etc/rc.conf 文件，把

```
Linux_enable="NO"
```

改成

```
Linux_enable="YES"
```

2. 安装 Linux 共享程序库。这些程序在 FreeBSD 2.2.2 中被存放在软件包 linux_lib-2.4.tgz 中。执行如下命令：

```
pkg_add <path_to_package>/Linux_lib-2.4.tgz
```

来安装这个软件包。<path_to_package>是存放这个软件包的目录。

3. 安装要执行的 Linux 程序。这样就可以在 FreeBSD 下执行 Linux 系统中的可执行文件了。

第 2 章 Linux 的高级安装

2.1 Linux 的多重引导

2.1.1 与 DOS 和 Windows 95 共存于同一系统

本节介绍在同一个硬盘内共存 Linux、DOS 和 Windows 95 这三种操作系统时，使用 LILO 作为其启动的管理程序。

假设用户已经知道安装 Linux 的步骤。

2.1.1.1 问题详述

在同一个硬盘里，若将 DOS 与 Windows 95 分别安装在不同的硬盘主分区，且将两者同时设为可启动时，会导致系统工作不正常，所以需要有一个引导管理程序以选择启动操作系统，使得在执行其中一个操作系统时，该操作系统看不到另一个操作系统所在的硬盘分区，从而达到解决问题的目的。

这主要是因为 DOS 与 Windows 95 所在的硬盘区必须是 C:，即第一个 FAT 格式的硬盘主分区，所以即使用户将 DOS 安装在 D: 或其他非 C: 硬盘主分区，当 DOS 启动完成时，DOS 依然会把所在硬盘区显示为 C:。这是由于 DOS 本身设计的特点，详细情况请参见 Microsoft 官方文件。

要解决这个问题唯一的方法就是使用引导管理程序，以下将使用 Linux 的 LILO 作为引导管理程序。

2.1.1.2 设置步骤

1. 使用 Linux 的 fdisk 命令在用户的硬盘中分出三个硬盘主分区，然后先在其中一个硬盘主分区安装 Windows 95，在另一个硬盘主分区安装 DOS，在第三个硬盘主分区安装 Linux。

假如用户只有一个硬盘主分区，可以使用 FIPS 这个软件来重新分区用户的硬盘，并且不会损坏用户的数据，但使用 FIPS 这个软件时需要一点技巧，详情请见第 1 章。

FIPS 是一个免费的软件，用户可以在各个有关 Linux 的 FTP 站点找到。

2. 从各个 FTP 站点取得 lilo.17.tar.gz 或更新的版本, lilo.17.tar.gz 已经可以在启动时更新分区活动标记, 将其安装在用户的系统中。



注意: 如果用户使用的是 1996 年后的 Linux, 那么可以忽略步骤 2。

3. 在/etc 中用户可以找到 lilo.conf 这个文件, 将其内容更改如下:

```
boot = /dev/hda
compact
delay = 5           # 5 秒后, 进入到 Linux 操作系统。
vga = normal        # 指定一般 VGA 文字显示模式 (80X25)
ramdisk = 0         # 指定 ramdisk 大小, 一般不需要, 所以设 0
root = current

image = /vmlinuz.1.3.97      # 指定 Linux 的内核文件
append = "aha1542=0x230 ro"
label = Linux                # 启动 LILO 时, 操作系统的启动名称
                              # 此处是用 Linux 作为启动名称

other = /dev/hda1            # 指定其他操作系统所在的硬盘分区
table = /dev/hda             # 表示有包含分区表的硬
                              # 盘为 /dev/hda

rewrite-table
label = dos                  # 此处用 DOS 作为 DOS 操作系统的
                              # 启动名称

other = /dev/hda2
table = /dev/hda
rewrite-table
label = w95
```

4. 执行/sbin/lilo 命令, 更新在主引导记录(MBR)中的数据。

5. 使用方法:

在出现 LILO 提示符时, 按下 Tab 键将显示出系统上可以引导的各个操作系统的代号, 根据需要输入想要使用的操作系统, 在按下回车键后就会引导相应的操作系统。

2.1.2 与 Windows NT 共存于同一系统

本节描述使用 Windows NT 引导程序来启动 Linux 的方法。本过程在 Windows NT 4.0 Workstation 版本及 Linux 2.0 上测试通过。

2.1.2.1 Windows NT OS Loader的运行

Windows NT OS Loader 是通过来自其他操作系统的硬盘启动分区文件来实现的。它读入该文件并启动所选择的操作系统,也就是说,或者启动不同模式运行的 Windows NT 或者启动其他的操作系统。

2.1.2.2 安装Windows NT

先试着安装 Windows NT。如果用户想使用 NTFS 文件系统作为 Windows NT 分区的文件系统,记住,目前版本的 Linux 无法使用 NTFS 的分区。可以读取 NTFS 分区的 Alpha 版驱动程序可在下面的站点获取:

<http://www.informatik.hu-berlin.de/~loewis/ntfs>

用户可另外建立一个 FAT 分区作为数据交换分区,或是使用 DOS 格式化的软盘作数据交换用。

2.1.2.3 分区硬盘

建议不要使用 Windows NT 的硬盘管理器来建立 Linux 的交换分区和根目录分区。建议使用 Linux 的 fdisk 命令。

2.1.2.4 安装Linux

启动 Linux 系统,建立交换分区和根目录分区。fdisk 命令会将 NTFS 分区认为是 HPFS 分区,这是正常的。再次启动 Linux 系统,并按用户的喜好安装 Linux。

2.1.2.5 LILO

当用户进行到安装 LILO 的部分时,指定用户的 Linux 根目录分区作为用户的启动装置,因为用户硬盘的主引导记录现在是由 Windows NT 管理。

在文件/etc/lilo.conf的 root 记录及 boot 记录会有相同的值。如果用户有一个 IDE 硬盘,而且用户的 Linux 分区是在第二块分区上,则/etc/lilo.conf 文件的 boot 项目会如下所示:

```
boot=/dev/hda2
```

如果用户有两个硬盘,而且用户的 Linux 位于第二块硬盘的第一个分区上,则/etc/lilo.conf 文件的 boot 记录会如下所示:

```
boot=/dev/hdb1
```

2.1.2.6 启动硬盘分区文件

现在用户必须用 dd 命令从用户的 Linux 根目录分区中将硬盘启动分区存放到一个文件中,例如存放在 bootsect.lnx 文件中。如果/dev/hda2 是用户的 Linux 分区,这个 dd 命令是:

```
# dd if=/dev/hda2 of=/bootsect.lnx bs=512 count=1
```

现在将文件 `bootsect.lnx` 拷贝到一张 DOS 格式化过的软盘中。

用户可以使用如下命令来拷贝：

```
# mcopy /bootsect.lnx a:
```

或者使用下面的命令组合来拷贝：

```
# mount -t msdos /dev/fd0 /mnt
# copy /bootsect.lnx /mnt
# umount /mnt
```

2.1.2.7 Windows NT使用硬盘启动分区文件

从软盘上将文件拷贝到 `C:\bootsect.lnx` 目录下(命令为：`copy a: bootsect.lnx c:\`)。就像 Linux 下的 `lilo.conf` 文件，在 Windows NT 中对应的文件是 `c:\boot.ini`。

在编辑之前，用以下的命令取消它的系统及只读属性：

```
C: \attrib -s -r c: \boot.ini
```

现在使用任意一个编辑器更改 `boot.ini` 文件，例如 Notepad(记事本)，修改后的内容应该如下所示：

```
[boot loader]
timeout=30
default=multi(0)disk(0)rdisk(0)partition(1)\WINNT
[operating systems]
multi(0)disk(0)rdisk(0)partition(1)\WINNT="Windows NT Workstation ..."
multi(0)disk(0)rdisk(0)partition(1)\WINNT="Windows NT Workstation ..."
C: \BOOTSECT.LNX="Linux"
```

在本例中只加入了最后一行。保存 `boot.ini` 文件后用如下命令恢复它的属性：

```
C: \attrib +s +r c: \boot.ini
```

退出用户的 Windows NT 系统并重新启动后，用户应该会看到如下的显示：

```
OS Loader V4.00
```

请选择想启动的操作系统：

```
Windows NT Workstation Version 4.0
Windows NT Workstation Version 4.0 [VGA mode]
Linux
```

使用 `↑` 键和 `↓` 键选中想要的选项，选定后按 `Enter` 键确认。

选择 `Linux` 后按回车键会看到：

```
LILO loading zImage ....
```

2.1.2.8 排除可能的问题

如果事情并不按照预期的进行，用软盘来检查系统的可启动性。如果 `/dev/hdb1` 是 Linux

分区, 则/etc/lilo.conf 文件有如下记录:

```
root=/dev/hdb1
boot=/dev/fd0
```

插入软盘, 然后执行 lilo 命令。现在试着从软盘上启动。如果在/dev/hdb1 上的 Linux 无法被启动, 那么 Windows NT OS Loader 也将无法启动它。如果看到一大堆“01 01 01 01”的信息, 表示用户的硬盘根分区无法使用。检查是否所有的硬盘都被 BIOS 检测到。

如果这张软盘可以启动用户的 Linux 分区, 用户可以用如下命令得到用户的硬盘启动分区给 Windows NT OS Loader 使用:

```
# dd if=/dev/fd0 of=/bootsect.lnx bs=512 count=1
```

可以保留这张软盘, 在 Windows NT 安装出问题作为修复盘。

2.1.3 用 Windows NT Loader 启动 FreeBSD 操作系统

如果计算机中除了 Windows NT 4.0 操作系统及 Linux 操作系统之外, 还安装了 FreeBSD 2.2 操作系统, 就可以仿照上面的做法试试 Windows NT Loader 能不能启动 FreeBSD 操作系统。

在本例中, FreeBSD 装在/dev/sda2, 也就是 FreeBSD 的/dev/sd0s2。按照上面的做法可以先将/dev/sda2 的引导扇区存成一个文件:

```
# dd if=/dev/sda2 of=/bootsect.bsd bs=512 count=1
```

将它拷贝到 C:\下。再在文件 boot.ini 中加入如下一行:

```
C:\BOOTSECT.BSD="FreeBSD"
```

这个方法如果失败, 无法启动 FreeBSD。那么, 就可以用 LILO 来启动 FreeBSD 操作系统。于是, 可以使用下面的方法。进入 Linux 后, 将/etc/lilo.conf 文件的内容改为:

```
boot = /dev/fd0
install = /boot/boot.b
vga = normal

other = /dev/sda2
label = FreeBSD
table = /dev/sda
loader = /dev/os2_d.b
```

放入一张格式化过的软盘, 再运行 lilo-v 命令。然后将硬盘启动分区变成一个文件:

```
# dd if=/dev/fd0 of=/bootsect.bsd bs=512 count=1
```

并放到 C:\下。然后重新启动, 在 Windows NT Loader 的选项菜单下选择 FreeBSD 选项, 接着就看到:

```
LILO loading FreeBSD
>> FreeBSD BOOT @ 0x10000: 639/65536 k of memory
Usage: [[0: ][fd](0, a)]/kernel [-abcCdhrsv]
Use 1: sd(0, a)/kernel to boot sd0 if it is BIOS drive 1
Use ? for file list or press Enter for defaults
Boot:
```

这就代表成功了。

2.2 Linux 启动盘的制作

有时候为了调试新的内核，或者别的原因，可能要使用软盘来启动 Linux。

怎么制作一张启动盘呢？

首先，重新编译内核，记住要打开 ramdisk 功能。

重新编译内核的步骤是：

1. `cd /usr/src/Linux`
2. `make config`

然后，制作一张安装时用的启动盘，把这张盘插入 Linux 的软驱，执行下面的命令加载软盘：

```
mount /dev/fd0 /mnt
```

如果不行，试着加上参数 `-t ext2` (对 Redhat 版本) 或 `-t minix` (对 Slackware 版本)，一定要装载上去。例如：

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

接着，按照下面的方法将自己编译的内核拷贝到软盘上：

```
cp /usr/src/Linux/arch/i386/boot/zImage /mnt/vmlinuz
```

其中，`/usr/src/Linux/arch/i386/boot/zImage` 是内核存放的路径，`/mnt/vmlinuz` 是启动盘中启动内核存放的路径。

接着指定启动的设备：

```
rdev /mnt/vmlinuz /dev/fd0
```

再修改软盘上的 `lilo.conf` 文件以满足要求：

```
vi /mnt/etc/lilo.conf
```

```
TimesNewRoman
```

最后，运行 `lilo` 命令：

```
lilo -r /mnt
```

卸下软盘，以后就可以用这张软盘启动 Linux 系统了：

```
umount /dev/fd0
```

如果 Linux 版本比较旧, 那么在运行 `redv/mnt/vmlinuz/dev/fd0` 命令后还要运行下面的命令:

```
rdev /mnt/vmlinuz/dev/fd0H1440
rdev -r /mnt/vmlinuz 1440
```

至此, 就成功地完成了 Linux 启动盘的制作。

2.3 Linux 和大内存

有的机器在有 16MB 以上的内存的时候可能会出现 Linux 不支持大于 16MB 部分的内存的情况, 可以用如下的办法来解决这个问题:

1. 重新编译内核, 使内核支持 16MB 以上的内存(在默认情况下 2.0.x 内核一般都支持 16MB 以上的内存)。

重新编译内核要执行下面的步骤:

```
cd /usr/src/Linux
make config
```

2. 如果启动时还不支持 16MB 以上的内存, 可以在 LILO 显示提示符时敲入:

```
mem=32M ( 假设是 32MB 的内存 )
```

看看 Linux 支持不支持内存(用 `free` 查看内存, 或者用 `dmesg` 查看启动时候的信息), 如果支持, 在 `/etc/lilo.conf` 文件中加上如下 一行:

```
append="mem=32M"
```

即可, 如果已经有了 `append` 选项, 则直接在原来的 `append` 中加上 `mem=32M` 即可, 例如:

```
append="ether=0x3, 0x280, eth0 mem=32M root=/dev/hd1"
```

然后再次运行 `lilo` 命令即可, 下次启动的时候 Linux 就支持所有的内存了。

3. 如果经过以上的两步 Linux 还是不支持内存, 那么就是 Linux 不支持大内存了, 去掉多余的内存, 或者使用更新的内核版本重复以上的步骤。

2.4 Linux 中模块的配置和使用

Linux 一个重要的特性是核心可以根据需要来动态地加载模块, 不需要时自动卸载模块, 本节就详细讲述 Linux 中模块的配置和使用方法。

2.4.1 Linux 系统的模块

下面简介模块的概念及使用模块的方法。

2.4.1.1 模块

这是一个很重要的功能。模块的英文是“**module**”，在此指的是内核模块(**kernel modules**)；简单地说，一个内核模块提供一个功能，如 **isofs**、**minix**、**nfs** 或 **lp** 等。在 Linux 中可以将这些模块独立出来，在需要使用这些模块对应的功能时，内核会自动加载相应的模块，而在不使用时，内核会自动卸载相应的模块。

使用模块，有下面两点好处：

- 修改内核时，不必全部重新编译，可节省不少时间。
- 若需要用到新的模块，不必重新编译内核，只要插入对应的模块即可。

2.4.1.2 kerneld

kerneld 是一个智能的守护进程，它会自动把需要的模块插入内核，把没用到的模块脱离内核，省去用户的麻烦。若没有 **kerneld** 守护进程，就只好在 **/etc/rc.d/** 目录下加一个 **rc.modules** 文件了。

2.4.1.3 使用模块

1. 准备工作

获取文件 **modules-1.3.57.tar.gz**，这是模块一定要用到的系统文件。

把取回来的 **modules-x.x.xx.tar.gz** 解至适当的目录下，然后通过命令 **make** 把它们生成目标文件。执行 **make** 命令后，应该产生下面这些系统程序：

modprobe、**depmod** (这是一个到 **modprobe** 的符号链接)、**gensyms**、**makecrc32**、**insmod**、**rmmod** (这是一个到 **insmod** 的符号链接)、**lsmod** (这是一个脚本文件)、**ksyms** (这是一个到 **insmod** 的符号链接)以及 **kerneld**。

其中，以 **insmod**、**rmmod**(=**insmod**)、**lsmod**、**depmod**(=**modprobe**)和 **kerneld** 最重要：

- **lsmod**：列出当前内核中插入的模块。
- **insmod**：把某个模块插入内核中。
- **rmmod**：把某个不在使用的模块脱离内核。
- **depmod**：创建模块相关文件(**dependency file**)，以使 **insmod** 能找到要插入的模块。这个相关文件就在 **/lib/modules/[kernel 版本]/modules.dep** 目录下。
- **kerneld**：守护进程，可自动中断 **insmod** 插入模块，同时它也查看系统的运行状态，若某模块插入后很久没被使用，就会自动把它脱离出来，以节省内存。

在执行 **make** 命令之后，要使用命令 **make install** 来进行安装。

到此就完成第一个准备工作了。

把 **depmod** 和 **kerneld** 加入 **/etc/rc.d/rc.S** 文件(在 **SysVinit2.6** 系统中是加入到 **/etc/init.d/boot** 文件)中。

请看下面在 SysV init 2.6 中的 init.d/boot 范例:

```
# Create a generic link to the modules for the current kernel
#
# You can have generic symbolic links in /lib/modules/boot
# if you create links to: "../current/subsystem/module.o"
#
rm -f /lib/modules/current
ln -s /lib/modules/`uname -r` /lib/modules/current
if [ -r /lib/modules/current/modules.dep ]
then
    echo "Module dependencies file exists, refresh it."
else
    echo "No current module dependencies? creating it."
fi
/sbin/depmod -a
/sbin/modprobe -a -t boot \*
```

2. 计划

请列出一张列表, 看看内核有哪些选项是必需的, 也就是说, 哪些内容是不可以编译成模块的。

3. 新的内核及模块

下面开始重新编译内核, 使用的命令是:

```
# cd /usr/src/Linux
# make config
```

执行命令后, 屏幕上将会出现如下信息:

```
*
* Loadable module support
*
Enable loadable module support (CONFIG_MODULES) [Y/n/?]
Set version information on all symbols for modules (CONFIG_MODVERSIONS) [N/y/?]
Kernel daemon support (e.g. autoload of modules) (CONFIG_KERNELD)
[N/y/?]
```

对其中的提问分别回答 Y、N 或 M 即可。其中 CONFIG_KERNELD 的默认值是 N, 所以要注意别按错键。

其他的内核配置, 如果遇到现在不能编译成模块的功能(依自己的计划而定, 因人而异), 就只能回答 Y。其他的按如下方式回答:

M: 若提供这个选项。

Y: 没有提供 M 选项, 但又必须使用相应的功能。

N: 根本就用不到这个功能。

使用命令 `make config` 后，接着使用命令：

```
make dep; make clean
```

在使用 `make zliilo` 或 `make zlmage` 命令之后，将产生出没有模块的内核映像，所以还要再使用如下命令：

```
make modules; make modules_install
```

才行。

4. 使用 `depmod -a` 命令

最后一步使用下面的命令：

```
depmod -a 1.99.7  
reboot
```

如果没什么问题，就是成功了。

2.4.2 RPM

本节将描述怎样在 Linux 的 Slackware 版本中安装 RPM 及正常执行它的功能。本节的内容适合 Linux 的各版本。

2.4.2.1 简介

RPM 是“RedHat Package Manager”的缩写，在 Linux 系统的大部分发行版本中都提供它。它最基本的功能是安装和拆卸软件包。

2.4.2.2 取得软件

在 Linux 的一些发行版本中，没有直接提供 RPM，要使用它首先要获取该软件。

RPM 的最新版本可在如下站点得到：

```
ftp://ftp.rpm.org/pub/rpm/dist/latest
```

目前最新的版本是：

```
rpm-2.4.12-1.i386.tar.gz
```

2.4.2.3 安装软件

一定要用 `root` 的身份才能完成以下的步骤。

在 `root` 的目录中解开文件。所需要的命令如下(不妨设 RPM 版本安装在 `/home/wolf` 目录下)：

```
cd /  
tar zxvpf /home/wolf/rpm-2.4.5-1.i386.tar.gz
```

再在 `/var/lib` 之下建立一个 `rpm` 目录：

```
mkdir /var/lib/rpm
```

然后可以输入下面的命令：

```
rpm --initdb
```

来初始化 RPM 数据库。

如果一切顺利，便建立了一个具有 RPM 的系统。

可以取来一些 RPM 文件测试一下系统，用“rpm -Uvh filename.rpm”命令测试即可。

第 3 章 Linux 系统的配置

本章将详细讲述 Linux 系统中基本的配置方法，包括笔记本电脑上的配置、声卡配置、闪光警告、与其他系统共享调制解调器、X Window 配置和 ZIP 磁盘配置等。

3.1 笔记本上的 Linux

本节描述如何调整 Linux 系统的某些设置来减少系统的电源消耗。这对那些使用装有 Linux 系统的笔记本式计算机的用户来说是非常有益的。

3.1.1 简介

现在拥有笔记本式计算机的人越来越多，而且在这种计算机上安装 Linux 系统的人也越来越多。

在一台笔记本式计算机上安装与使用 Linux 已经没有问题了，放手去试试吧。不像其他有些操作系统，Linux 仍旧支持非常旧的硬盘并可在其上运行得很好，所以安装 Linux 也许可以使笔记本式计算机获得重生。

如果在笔记本式计算机上安装 Linux 时需要帮助或是遇到有关笔记本式计算机硬件上的问题，可查看如下的 Linux 笔记本式计算机网页：

<http://www.cs.utexas.edu/users/kharker/Linux-laptop/>

在该站点上可找到很多有用的信息和得到详细的帮助。这个 Linux 笔记本式计算机网页还介绍了特定型号的笔记本式计算机及芯片组等硬件。

本节主要讨论所有的可携式系统的共同问题——电源消耗。

3.1.2 APM

大部分的可携式系统及很多桌上型计算机都支持 APM(高级电源管理)。

本节将讲述如何在 Linux 系统下配置和使用 APM。

3.1.2.1 APM 的作用

通过 APM，当 CPU 空闲一段时间后，CPU 可以触发 BIOS，让 BIOS 使用它自己的

节省能源措施, 例如降低 CPU 主频、关闭硬盘和关掉屏幕等。

大部分的 Linux 系统将 `shutdown -r` 命令放在系统的 `/etc/inittab` 文件内, 所以当按下 `Ctrl+Alt+Del` 组合键时, 笔记本式计算机就能够自动关机。

3.1.2.2 在Linux中启用APM功能

如何在 Linux 中启用 APM 功能?

非常简单, 只需重新编译 Linux 内核程序。

当重新编译内核执行到“character devices”处时, 完全支持 APM BIOS 的设置如下(对于 2.0、3.0 及其以上版本):

```
Advanced Power Management BIOS support: Yes
Ignore USER SUSPEND: No
Enable PM at boot time: Yes
Make CPU Idle calls when idle: Yes
Enable console blanking using APM: Yes
Power off on shutdown: Yes
```



注意: 如果用户的系统不完全支持 APM BIOS 的标准, 选项中有些内容可能会让用户的系统死机。

3.1.2.3 APM 和 PCMCIA 驱动程序

在重新编译内核之后, 请勿忘记重新编译 Linux 的 PCMCIA 驱动程序。

随着供应商发行的 Linux 版本所附的编译前的 PCMCIA 驱动程序大多不支持 APM, 因此用户需要重新编译 PCMCIA 驱动程序, 使之支持 APM 功能。

3.1.2.4 apmd 软件包

用户的 Linux 系统安装 APM 后, 接着需要 `apmd` 软件。这是 APM 真正执行的软件, 如果用户在 Linux 系统上找不到这个软件, 可以从各大 Linux 站点及 SunSite 站点下载这个软件。

`apmd` 守护进程会记录用户的电池的特性, 并且还会在电压低时警告用户。`apm` 程序会使用一个 Shell 命令来使用户的系统工作在最小功能下, 运行在 X Window 系统下的 `xapm` 程序会以图形的方式显示出用户电池的状态。

3.1.2.5 笔记本式计算机不支持APM

如果用户的计算机的 BIOS 不支持任何节省能源的设置, 用户可以使用 `hdparm-S` 命令来定义用户硬盘的待命时间。这样会减少电源的消耗, 因为硬盘的运行是要消耗许多电源的。

用户的系统应该首先安装 `hdparm`, 然后使用命令 `man hdparm` 来查看该命令的语法及帮助信息。

3.1.2.6 改变某些一般的系统设置

有时得到 Linux 并在笔记本式计算机上安装、执行后,会发觉即使系统上没有用户登录,它也会每隔几秒就存取一次硬盘。这可能是由于硬盘没有进入节能模式。如果能够减少硬盘的访问,就可以很明显地增加电池的使用时间。以下是配置的方法:

1. crond 守护进程与 atrun 守护进程

通过 crond 守护进程,用户可以指定在特定的时间内执行特定的命令,具体的命令和时间存放在/etc/crontab 文件中。

使用 at 命令用户可以将以后要用到命令放在命令等待区内,时间到就会执行命令。有些发行版本的 Linux 系统使用 atd 守护进程来做这个工作,其他的发行版本(例如 RedHat)使用 crond 守护进程来完成这个工作。

2. update/bdfush 守护进程

Linux 随时都在处理着很多已打开文件的缓冲区,以保证文件有任何改变时能够尽快保存文件,否则,当系统死机时已做的改变就会丢失。

update/bdfush 守护进程就是用来做这种工作的。update 和 bdfush 是同一个程序的两个名字,可以使用其中一个来启动这个守护进程。默认的设置会让这个守护进程每 5 秒钟执行一次 flush 操作,每 30 秒钟执行一次 sync 操作。

• RedHat 4.1

在 /etc/inittab 文件内,将 update 命令改成:

```
ud::once:/sbin/update -s 3600 -f 3600
```

• Slackware

update 命令是在/etc/rc.d/rc.S 文件中被执行的,而不像 RedHat 中在/etc/inittab 文件中被执行。

3. syslogd 守护进程

syslogd 守护进程维护/var/log/目录下的很多 Linux 系统日志文件。在默认情况下,syslogd 守护进程在每次有系统信息被记录时会将信息保存到日志文件中。

syslogd 对应的配置文件是/etc/syslog.conf,用户可以通过修改这个文件来配置对哪些系统信息做日志记录,以及记录到哪个日志文件中。以下是在某个系统内的 syslog.conf 文件的内容:

```
# Log anything (except mail) of level info or higher.
# Don't log private authentication messages!
*.info;mail.none;authpriv.none -/var/log/messages
```

4. init 命令

在系统启动时,开始时的处理程序以及守护进程是通过 init 命令启动的。这个命令在每次它所建立的处理程序前会执行 sync 命令。

用户可以找到 init 命令的源代码,可去掉其中的 sync()调用,然后重新编译内核。

去掉 sync()调用后,可能会出现文件内容丢失的现象。要避免发生文件内容丢失的现象,用户应该在系统的/etc/rc.d/init.d/halt 脚本内在卸下文件系统前加上 sync()的调用。

5. 交换区

Linux 交换区以虚拟内存的方式来增加物理内存的空间。这也是硬盘经常会被存取的可能原因。如果用户的笔记本式计算机的物理内存很大，或是用户只用到很简单的应用程序(如 vi)，可以关闭交换区。

X Window 环境需要的动态内存空间很大，用户最好不要在没有交换区的情况下使用它，除非用户有 16MB 以上的物理内存。

如果已经安装了交换区，用户可以在/etc/rc.d/rc.sysinit 文件内的 swapon 命令行前加“#”符号关闭交换区。

如果用户不想永远地去掉交换区，可以让系统在启动时询问是否要使用交换区。实现方法是在文件/etc/rc.d/rc.sysinit(RedHat 4.1)或文件/sbin/init.d/boot(Suse 4.4.1)里加入如下的内容：

```
echo "Should the system use swap?"
echo " 0: No."
echo " 1: Yes."
/bin/echo "Your choice: \c"
read SWAPCHOICE

case "$SWAPCHOICE" in
0)
    # Do nothing.
    echo "(Swap partitions disabled)"
    ;;
*)
    # Start up swapping.
    echo "Activating swap partitions"
    swapon -a
esac
```

这样，用户就可以在使用交流电源时开启交换区，在使用电池时关闭交换区。

6. emacs 编辑器

如果使用过 emacs，用户也许已发现这个编辑器会自动保存文件。此功能很有用，故不能关闭它，但可以通过修改参数使之更适合笔记本式计算机的环境。

在/usr/share/emacs/site-lisp/site-start.el 文件内加入下列几行：

```
(setq auto-save-interval 2500)
(setq auto-save-timeout nil)
```

可以关闭基于时间的自动保存文件功能，而修改为每 2 500 次键盘的敲击时自动保存一次文件。当然，如果正好在打开一个文件，那么应该减少 2 500 这个数值。如果正在编写程序那么它已可满足需求了，因为每一个操作，如：向上、向下、向左、倒退和粘贴等都列入计算，所以很快就会满足 2 500 次的键盘操作的要求，从而自动保存文件。

7. 查看系统运行状态

如果 Linux 系统仍经常存取硬盘，用户可以使用下面的命令：

```
%ps ax
```

来查看系统上当前运行的进程情况。该命令会显示出所有正在运行的程序及其名称，有时还会列出每个运行程序的命令行参数。

可阅读每个运行程序的联机手册来了解该程序实际的操作。用这个方法，用户可以找到大部分运行程序所进行的操作。

3.2 声卡的配置

本节将讲述声卡的配置。首先要清楚的是，有些 Linux 系统不支持声卡。在这种情况下，用户需要先安装相应的软件包。

3.2.1 安装 isapnp 软件包

按照该软件包的帮助说明进行安装，在安装之后，需要进行配置。

编辑/etc/isapnp.conf 文件(该软件的配置文件)，使声卡的 IRQ 和输入/输出地址不会与用户的其他硬件冲突。

3.2.2 重新创建 Linux 内核

将声卡支持变成一个可装入模块，其中 IRQ、输入/输出及 DMA 地址就如 isapnp.conf 所配置的那样。同时也将用户的 3c509 网卡制作成模块。



注意： 在编译及安装完用户的内核之后不要忘了执行命令 `make modules` 和 `make modules-install`。

3.2.3 修改系统的引导程序脚本文件

本例中运行的是 Slackware 软件包，内核版本是 2.0.23。在设置网络之前，执行 isapnp 程序(在/etc/rc.d/rc.M 文件中则在设置 hostname 之后再执行 isapnp 程序)：

```
# EZA
# 设置即插即用装置
/sbin/isapnp /etc/isapnp.conf
```

然后，在/etc/rc.d/rc.inet1 中装入网卡驱动程序：

```
# EZA
# 装入网卡模块
```

```
/sbin/insmod 3c509
```

最后，在`/etc/rc.d/rc.local`文件中插入声卡驱动程序模块：

```
/sbin/insmod sound
```

关于 Linux 即插即用装置的更多信息可在如下站点中获取：

<http://www.redhat.com/Linux-info/prp/>

3.2.4 一个 isapnp.conf 的例子

下面是一个 `isapnp.conf` 的例子，大部分的注释都已删除。

```
# Trying port address 0203
# Board 1 has serial identifier 67 00 00 40 17 2b 00 8c 0e

# (DEBUG)
# (READPORT 0x0203)
# (ISOLATE)
# (IDENTIFY *)

# Card 1: (serial identifier 67 00 00 40 17 2b 00 8c 0e)
# CTL002b Serial No 16407 [checksum 67]
# Version 1.0, Vendor version 2.0
# ANSI string -->Creative SB16 PnP<--
#
# Logical device id CTL0031
#
# Edit the entries below to uncomment out the configuration required.
# Note that only the first value of any range is given, this may be changed
if required
# Don't forget to uncomment the activate (ACT Y) when happy

(CONFIGURE CTL002b/16407 (LD 0
#   ANSI string -->Audio<--

# Multiple choice time, choose one only !

#   Start dependent functions: priority preferred
#       IRQ 5.
#       High true, edge sensitive interrupt (by default)
(INT 0 (IRQ 5 (MODE +E)))
(DMA 0 (CHANNEL 1))
(DMA 1 (CHANNEL 5))
```

```
(IO 0 (BASE 0x0220))
(IO 1 (BASE 0x0330))
(IO 2 (BASE 0x0388))
(ACT Y)
))

(CONFIGURE CTL002b/16407 (LD 1
(INT 0 (IRQ 11 (MODE +E)))
(IO 0 (BASE 0x01e8))
(IO 1 (BASE 0x03ee))
(ACT Y)
))

(CONFIGURE CTL002b/16407 (LD 2
#   ANSI string -->StereoEnhance<--
#   Logical device decodes 16 bit IO address lines
#       Minimum IO base address 0x0100
#       Maximum IO base address 0x0138
#       IO base alignment 8 bytes
#       Number of IO addresses required: 1
#(IO 0 (BASE 0x0100))
#(ACT Y)
))

(CONFIGURE CTL002b/16407 (LD 3
(IO 0 (BASE 0x0200))
(ACT Y)))
```

3.3 闪光警告

本节将讲述在系统上如何使用 `termcap` 数据文件设置闪光警告并描述如何根据需求使喇叭声无效。

每当一个 BEL 字符(ASCII 代码为 7)输出时, Linux 主控制台就会发出喇叭声。

许多用户不喜欢他们的计算机发出“哔哔”声。本节将讲述如何使应用程序不产生 BEL 代码。本节大部分内容针对在图形环境下工作的用户。

3.3.1 Speakerectomy

Speakerectomy 针对喇叭声的问题最直接的解决方法就是取掉喇叭，这样就听不到喇叭的声音了。

PC 机上通常装一个切换较低 CPU 主频的开关(目前很多新型计算机已经取消这种开关)。

当用户在一般环境中工作时，这个开关从不使用。用户不能使用切换开关来增加 CPU 速度，但是可以使用它启动扩音器或关闭扩音器。

但是，使用笔记本式计算机的用户不能使用上面的方法。对于这样的用户，较好的解决方案是设置他们的软件避免发出“哔哔”声，见 3.3.2 节。

3.3.2 设置每一个主控制台发出“哔哔”声

在 Linux1.3.43 版本中，已经可以设置发出“哔哔”声的强弱和持续时间。通过修改 `console.c` 文件，每一个主控制台可以被设置成具有不同的持续时间或者不同的喇叭声音频率，可通过使用换码序列输出到主控制台设备来完成该操作。用户可以通过设置自己的 `~/profile` 或者 `~/login` 文件，设置不同的“哔哔”声来连接到各主控制台(或者不发出“哔哔”声)。

换码序列以如下方式运行：

`ESC-[10; xx]`用来选择喇叭声频率，单位是赫兹。这个值应在范围 21~32 766 之内，否则这个值就是没有被定义。如果 `xx` 参数值没有被指定，将会运行默认值 750Hz，被认为是 `ESC-[10]`。

`ESC-[11; xx]`用来选择喇叭声持续时间，单位是毫秒(ms)。如果 `xx` 参数值遗失了，将会使用默认值 `ESC-[11]`。

例如，想让一个频率 50Hz 的喇叭声持续一秒钟，用户可以使用命令：`echo -e"\033[10; 50]\033[11; 1000]"`。在 `bash` 程序下“-e”意思是“使用换码序列”。如果用户使用 `tcsh` 程序，命令为：`echo "\033[10; 50]\033[11; 1000]"`。

3.3.3 termcap 和 terminfo

`/etc/termcap` 是一个文本文件，它包含系统上所有终端的性能及终端参数。系统上有许多应用程序都使用 `termcap` 文件中的数据，完成在屏幕周围移动光标和其他的屏幕定位工作。`tcsh`、`bash`、`vi` 以及其他基于 `curses` 的应用程序都使用 `termcap` 数据库。

数据库描述了多个终端的类型。`TERM` 环境变量是在运行时选择的，应用程序通过指定一个 `termcap` 记录来使用终端。

在数据库内部，每一个终端的记录项显示一个或两个字符的代码和一个实际的换码序

列。不同部分之间的分隔符是冒号(:)。

举一实例,听得见的喇叭声,代码为 bl,在 termcap 文件中出现的形式通常是 bl=^G。系统根据 bl=^G 就可以确定喇叭声来自打印 Control+G 字符,即 ASCII 中的 BEL 控制符。

除 bl 代码外,vb 代码也经常使用。它被用来表现“闪光警告”。vb 代码在 termcap 文件中是不存在的。

目前大部分应用程序和程序库使用 terminfo 数据库来代替 termcap 文件,它们的作用是相同的。

3.3.4 定义一个闪光警告

如果还没有定义闪光警告,用户可以在自己的 termcap 文件中加入 vb 代码项,来定义一个闪光警告。建议用户在 termcap 文件中插入下列一行:

```
: vb=\E7\E[?5h\E[?5l\E[?5h\E[?5l\E[?5h\E[?5l\E[?5h\E[?5l\E8: \
```

其中符号反斜杠在该数据库中被用作换行符。

3.3.5 避免发出“哔哔”声的方法

下面是一份应用程序不完整的列表,它可以针对当前终端的状态(使用 termcap 数据或者 terminfo)来使用 vb 选项:

- **XServer:** 使用 `xset -b` 命令来选择喇叭声的行为。这个命令对应三个参数——大小、频率和持续时间。`xset -b` 使喇叭声完全无效。设置 X Window 服务器后将影响在显示器上运行的所有应用程序。
- **xterm:** `xterm` 可以把各喇叭声转换为一个看得见的或者听得见的信号。喇叭声在 `xterm` 默认情况下是听得见的,但是用户可以使用带有 `-vb` 选项的 `xset` 命令和在 `($HOME)/Xdefaults` 文件中加入“`xterm*visualbell: true`”命令的办法把它转换成一个看得见的闪光。如果用户正在执行 X 程序,可能不需要下列数据:
 - ✧ `tcsh`(6.04 版本和更新的版本) `set visiblebell`。重设喇叭声只要使用命令 `unset visiblebell`。使任何通知命令无效只需使用 `set nobeep` 取代 `unset visiblebell`。
 - ✧ `bash` 在用户的 `~/.bashrc` 文件中加入 `set bell-style visible` 语句。喇叭声状态可以设成“没有”或者“可听得见的”。
 - ✧ `bash` 在 `~/.inputrc` 文件中加入 `set prefer-visible-bell` 语句。
 - ✧ `nvi` 和 `elvis` 在 `~/.exrc` 文件中加入 `set flash` 语句。使闪光警告无效需要在把文件中的 `flash` 替换为 `noflash`。
 - ✧ `emacs` 在用户的 `~/.emacs` 文件中加入 `(setq visible-bell t)` 语句。它是用 `(setq visible-bell nil)` 语句使喇叭声失效。
 - ✧ `less` 使用闪光警告,则在命令行上使用 `-q` 语句,使所有报告无效则使用 `-Q`

语句。默认选项可以加入用户的环境变量 LESS。

3.4 设置与别的系统共享调制解调器

本节将描述在 TCP/IP 网络上的 Linux 系统中如何设置与别的系统共享安装在 Linux 系统上的调制解调器。

3.4.1 服务器端

假设在 Linux 服务器上安装以下两者之一的调制解调器：

1. 一台接在/dev/ttySx 上的调制解调器；
2. 在 /dev/ttylX 上接有综合业务数据网(ISDN)的调制解调器。

以下用 perl 脚本所写的几行是实现与其他系统共享调制解调器的基本程序：

```
$ cat /usr/sbin/modemd

#!/usr/bin/perl
select((select(STDOUT), $| = 1)[$]);
select((select(STDIN), $| = 1)[$]);
exec 'cu -s 115200 -l /dev/ttyS1';
die '$0: Cant exec cu: $!\n';
```

如果有一个客户端连接到适当的连接端口，则守护进程 inetd 会启动调制解调器守护进程 modemd。modemd 只是使用 cu 命令把标准输入和标准输出与套接字句柄连接起来，然后让 cu 命令来控制调制解调器。如果没有 cu 命令，可以在 UUCP 软件包中找到它。

用户也必须修改 inetd 的设置，把 modemd 加入到其中，这个配置文件通常都放在文件/etc/inetd.conf 中，加入的内容如下所示：

```
#
# modem daemon
#
modem stream tcp nowait root /usr/sbin/tcpd /usr/sbin/modemd /dev/ttyS1
```

为了能使 modemd 可以运行，必须在文件/etc/services 里加上如下的内容：

```
modem          2006/tcp          modemd
```

在这个例子中用 modemd 守护进程把连接端口 2006 当作服务程序，其中的端口号用户可以任意指定成还未被用过的端口号。在这些都完成后，用户还需要重新启动守护进程 inetd，使用下面的命令可以重新启动守护进程 inetd：

```
$ ps |grep inetd
```

```
194 ? S    0: 00 /usr/sbin/inetd
```

```
$kill -HUP 194
```

至此，用户的服务器端已经可以接受客户端的请求及为客户端服务了，用户可以用下面的命令检查实际的效果：

```
$ telnet localhost modem

Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
```

接着，用户可以连上调制解调器，也可以用调制解调器的一些 at 命令来确认一下：

```
atz
atz
OK

atd1
atd1
Linux ISDN
OK

^]
telnet>quit
$
```

除了可以用 perl 脚本来充当调制解调器服务器外，程序 mserver 也可以完成相同的功能，该程序可以从下面的站点获取：

```
ftp: //ftp.innet.be/pub/staff/car1/
```

也可以用 Archie 或其他搜索引擎来找到这个程序。

3.4.2 客户端

在本节中将只讲述在 Windows 系统上的设置。服务器端设置好后，客户端 PC 机的 COM 端口还需要被重新引导到 TCP/IP 通信协议上，完成这个功能的比较好的程序是位于 www.tactical-sw.com 的 DialOut/IP。

DialOut/IP 会使用虚拟 COM 端口来连上调制解调器，Windows 中的程序可以使用这个虚拟 COM 端口。除了一些传真应用程序或是其他需要直接存取 UART 控制线的程序之外，许多客户端程序(包括 Windows 95 的【拨号网络】)都可以像真的有一台调制解调器接在这些程序所在的机器上面一样地去使用调制解调器。

DialOut/IP 可以被设置成提供 Telnet 的通信协议，但是这个特性只能应用在调制解调器的某些产品上。



注意： 虽然命名为“拨出(DialOut)”，DialOut/IP 也可以被应用程序用来接收拨进来的电话。

在 www.tactical-sw.com 站点上有一个供用户下载但仅可使用一至二周的全功能版本，安装及设置过程由设置程序处理，详细的安装信息在 README.TXT 文件中。当用户运行 DialOut/IP 时，必须输入共享的调制解调器的 IP 地址和端口号。

DialOut/IP 是仅允许被一台调制解调器所使用的商业产品，也就是用户所付出的代价依用户所使用的共享调制解调器的数量而定，许可协议(license)中说用户可以把 DialOut/IP 安装在所有可以存取共享调制解调器的个人计算机上。

3.4.3 安全问题

如果在用户的局域网络中只有一台调制解调器给所有机器分享，那么就没有安全问题。然而，如果用户的局域网络上有一台或多台的机器被设置成调制解调器服务器且也接上了因特网，那就必须考虑安全问题了，否则任何人都可以用 `telnet your_host_name modem` 命令随意拨打国际长途电话。

为了保护用户的调制解调器服务器不被非法的存取，建议用户安装和设置 TCP_Wrapper。TCP_Wrapper 是一个软件包，可以从下面的站点获取：

`ftp://ftp.cert.org`

有关该软件包的安装和使用，可以参考该软件包的帮助信息。

3.5 在 Linux 机器上设置串行通信设备

本节将介绍如何在 Linux 系统上设置串行通信设备。

3.5.1 串行端口的名称

4 个串行设备对应 COM1~COM4 这 4 个端口，如下所示：

```
/dev/cua0, /dev/ttyS0 (COM1) address 0x3f8 IRQ 4
/dev/cua1, /dev/ttyS1 (COM2) address 0x2f8 IRQ 3
/dev/cua2, /dev/ttyS2 (COM3) address 0x3e8 IRQ 4
/dev/cua3, /dev/ttyS3 (COM4) address 0x2e8 IRQ 3
```

/dev/ttySN 设备给进来的连接所使用，而/dev/cuaN 设备则给出去的连接所使用。N 是串行端口号码。在本例中，是以 ttyS0 对应 COM1，以 ttyS1 对应 COM2，以 ttyS2 对应 COM3，

以 ttyS3 对应 COM4。

Linux 系统的某些安装方式会另外建立 2 个设备——调制解调器使用的/dev/modem 和鼠标使用的/dev/mouse。它们分别是调制解调器实际设备文件的链接和鼠标设备文件的链接。

强烈建议用户不要使用这些链接，如果用户打算用调制解调器接受拨号，却使用了/dev/modem 设备文件，那么用户将会因为加锁文件无法正确运行而遇到困难。如果要使用这些链接，一定要使它们指向正确的设备，即该设备文件符号链接的目标文件是正确的。

3.5.1.1 在/dev里的串行设备的主要和次要号码

在/dev 里的串行设备的主要和次要号码如下所示：

```
/dev/ttyS0 major 4, minor 64    /dev/cua0 major 5, minor 64
/dev/ttyS1 major 4, minor 65    /dev/cua1 major 5, minor 65
/dev/ttyS2 major 4, minor 66    /dev/cua2 major 5, minor 66
/dev/ttyS3 major 4, minor 67    /dev/cua3 major 5, minor 67
```

要注意所有的发行软件包都应该正确地建立了这些设备。

下面介绍在/dev 下建立设备的方法。如果用户没有其中的一个设备，必须自己用 mknod 命令来建立它。

假定用户需要建立 ttyS0 设备，可以使用如下命令：

```
Linux# mknod -m 666 /dev/cua0 c 5 64
Linux# mknod -m 666 /dev/ttyS0 c 4 64
```

可以取得 MAKEDEV 这个脚本文件，一般 FTP 站点上都会有。这个脚本文件简化了设备的建立过程。例如，如果用户需要建立 ttyS0 设备，则应该输入：

```
Linux# cd /dev
Linux# MAKEDEV ttyS0
```

3.5.1.2 多用户卡的注意事项

多用户卡所使用的设备视用户具体使用哪种多用户卡而定。这些详细信息列在随附于 setserial 程序的 rc.serial 文件中。

用户可能会需要建立这些设备。用 mknod 命令或是取得的 MAKEDEV 脚本文件都可以。多用户卡所用设备是以加上“64+端口号”来建立的。所以，如果用户想为 ttyS17 建立设备，就应该输入：

```
Linux# mknod -m 666 /dev/cua17 c 5 81
Linux# mknod -m 666 /dev/ttyS17 c 4 81
```

注意到“64+17=81”。如果使用 MAKEDEV 脚本文件，应该输入：

```
Linux# cd /dev
Linux# MAKEDEV ttyS17
```

3.5.2 getty

getty 是在用户登录 UNIX 机器的时候处理登录过程的一个程序。在 Linux 系统上经常使用的版本有 3 种: agetty、getty_ps 和 mgetty。

3.5.2.1 getty_ps

这个 getty 版本由 Paul Sutcliffe Jr.(paul@devon.lns.pa.us) 所编写, 目前由 Kris Gleason(gleasokr@boulder.colorado.edu) 来维护。最新的版本是 2.0.7e, 它兼容于任何旧的版本。大部分 Linux 安装软件包都包含 getty_ps, 并且会把 getty_ps 安装成程序文件 /sbin/getty。

getty_ps 软件包包含两个部分: getty 由主控台和终端设备所使用; uugetty 由调制解调器所使用。

3.5.2.2 mgetty

mgetty 是由 Gert Doering 所编写的 getty 版本。除了允许拨入之外, mgetty 也通过随附的 sendfax 程序提供传真支持。mgetty+sendfax 0.22 是这个软件包的最新版本。这套 mgetty 的文件写得相当好, 因而不需要再补充什么。更详尽的信息, 请参考它的安装说明。

3.5.2.3 agetty

agetty 是第三种 getty 程序, 由 W.Z.Venema(wietse@wzv.win.tue.nl) 所编写。它是 getty 的简单版本。

3.5.3 setserial

setserial 是可以让用户查看并修改串行设备各种属性(包括它们所使用的端口地址、中断以及其他串行端口选项)的程序。

setserial 刚开始是由 Rick Sladkey 编写, 然后由目前维护它的 Ted T'so(tytso@mit.edu) 修改过。最新的版本是 2.10 版, 可以在 Linux FTP 站点上找到。可以通过执行不加参数的 setserial 命令来获取该软件的版本信息。

在 Linux 系统启动时, 只有 ttyS{0-3} 会以默认的 IRQ4 和 IRQ3 来进行配置。所以, 如果有由其他网卡提供的串行端口或是 ttyS{0-3} 使用非标准的硬件中断(IRQ), 必须用这个程序来配置这些串行端口。

要了解该命令的所有选项, 请参考它的联机手册。

3.5.4 拨号

用户要利用调制解调器拨号, 进而与服务器建立连接, 需要通过一系列的步骤, 下面

就来看一看整个连接过程。

3.5.4.1 硬件需求

确定用户拥有正确的连接线。调制解调器需要一条直接的连接线，没有任何跳线。要确定接口的形式是对的，应把用户的调制解调器接到其中一个串行端口上。如果需要关于这个调制解调器的辅助说明，请参考调制解调器手册。

内置式调制解调器注意事项如下：

内置式调制解调器不需要连接线，也不需要串行端口，只要给它配置一个没有被使用的硬件中断，并且配置该端口的输入/输出地址即可。如果出现问题，请参考调制解调器手册。

如果想要将内置式调制解调器设在 `ttyS3` 可能会遇到一些问题。如果 Linux 没有检测到设在 `ttyS3` 上的内置式调制解调器，可以使用 `setserial` 程序来设置，然后调制解调器才会正常运行。设在 `ttyS{0-2}` 上的内置式调制解调器在检测时应该不会有任何问题。

3.5.4.2 与调制解调器联络

在使用复杂的通信程序之前，先以 `kermit` 或其他简单的通信程序来测试这些设置。因为版权的关系，`kermit` 并没有随着商业发行软件包一起发行。可以从 `sunsite.unc.edu:/pub/Linux/apps/comm` 及其镜像站点上找到最新的 `kermit` 版本。

例如，要是调制解调器设在 `ttyS3` 设备上；而且可以接受 38 400bps 传输速率，可以按如下方法做：

```
Linux# kermit
C-Kermit 5A(188), 23 Nov 92, POSIX
Type ? or HELP for help
C-Kermit>set line /dev/cua3
C-Kermit>set speed 38400
/dev/cua3, 38400 bps
C-Kermit>c
Connecting to /dev/cua3, speed 38400.
The escape character is Ctrl-\ (ASCII 28, FS)
Type the escape character followed by C to get back,
or followed by ? to see other options.
AT
OK
<ctrl>-\-C
(Back at Linux)
C-Kermit>quit
Linux#
```

如果调制解调器对 `at` 命令集有响应，那么可以假定调制解调器能正确运行。

当使用调制解调器对外拨号时，将调制解调器设置到它所能支持的最高速度。必须用 `setserial` 程序将用户的串行端口设置到更高的速度，然后在通信程序中将速度设

置到 38 400bps。

3.5.4.3 配置用来对外拨号的调制解调器

只要对外拨号，就可以以用户需要的任意方式配置调制解调器。如果用户计划用调制解调器来接受拨号，那么必须将调制解调器配置成与 `getty` 的速度选项在相同的速度设置下。所以，如果想要在 38 400bps 的速度下执行 `getty`，在配置调制解调器的时候应将速度设成 38 400bps。这样做是为了避免计算机与调制解调器之间的速度不匹配。

如果要查看结果响应，就可以设 `Q0`(这会显示结果响应)。如果在调制解调器上进行这个设置，需要以 `at` 命令指定寄存器名称。利用 `kermit` 或其他通信软件连接到调制解调器，然后输入如下命令：

```
ATQ0
```

如果调制解调器响应 `OK`，那么寄存器就已经被设置。

如果用户想查看自己正在输入什么，可以设置 `E1`(本地响应命令)。如果调制解调器具有数据压缩能力，用户要想启动它，就可以参考用户的调制解调器手册以获得更多的帮助说明以及完整的选项列表。

3.5.4.4 硬件流量控制

如果调制解调器支持硬件流量控制(RTS/CTS)，建议使用它。特别是对于支持数据压缩的调制解调器。

启动调制解调器的硬件流量控制最好是在启动系统的时候做，将命令放在 `/etc/rc.d/rc.local` 或是 `/etc/rc.d/rc.serial` 文件中。用户需要对每一个想要启动硬件流量控制的串行端口执行下列操作：

```
stty crtscts < /dev/cuaN
```

用户必须在调制解调器上启动 `RTS/CTS` 流量控制。操作方法参考调制解调器手册，因为不同的调制解调器有不同的操作方法。如果用户的调制解调器支持内置式脚本文件，则要确定将配置存入用户的调制解调器。

3.5.5 用来接受拨号以及对外拨号的调制解调器的配置

要用来接受拨号以及对外拨号，那么必须设置调制解调器(再一次对调制解调器使用 `at` 命令集)：

`E1` 本地响应命令。

`Q0` 回报结果响应。

`V1` 开启冗长模式。

`S0=0` 不要接听(`ugetty` 以 `WAITFOR` 选项来处理)。

如果还没有正确地设置，放在配置文件里的拨号代码可能会失败，因而使整个程序中途退出。配置文件还有其他选项：

&C1 DCD 只有在连接以后才开启。

&S0 DSR 永远为开启。

DTR 开启 / 关闭 / 重置调制解调器。

这些选项会影响调制解调器有关开始以及结束拨号的执行。

如果调制解调器不支持内置式脚本文件，可以通过在配置文件中的拨号代码来设置这些内容。具体的方法见下文。有些调制解调器以 DIP 开关来设定寄存器，也要确定这些设置是正确的。

3.5.5.1 安装getty_ps

依照默认值，getty_ps 将会按 Linux 的文件系统标准来配置，即二进制文件将会放在/sbin 目录下，且配置文件将会命名为/etc/conf.{uu}getty.ttySN。系统也会假定加锁文件将要放到/var/lock 目录下。要确定用户有/var/lock 这个目录。

如果用户不想依照 Linux 的文件系统标准配置，二进制文件将会放到/etc 目录下，配置文件将会命名为/etc/default/{uu}getty.ttySN，加锁文件则将会放到/usr/spool/uucp 目录下。如果用户使用 UUCP，那么建议这样做，因为如果把加锁文件放到 UUCP 无法找到的地方，将会出现问题。

getty_ps 使用 syslogd 守护进程来记录信息。如果目前没有执行它，参阅 syslogd(1)以及 syslog.conf(5)的联机手册来设置 syslogd。信息会优先以 LOG_AUTH 记录，错误信息使用 LOG_ERR 记录，而排除错误信息使用 LOG_DEBUG 记录。

如果不想使用 syslogd，可以编辑在 getty_ps 源文件里的 tune.h，将日志文件名的宏去掉，默认文件名为/var/adm/getty.log。

决定是否要依照 Linux 的文件系统标准以及是否使用 syslogd 后，相应地编辑在 getty_ps 原始目录下的 tune.h 文件以及 Makefile 文件。然后，根据说明安装。

请注意下文中所有的 getty 都是指 getty_ps，uugetty 则是指 getty_ps 软件包里的 uugetty。

3.5.5.2 设置uugetty

为了能从调制解调器进行拨号，需要用 uugetty。

uugetty 执行加锁文件检查。更新/etc/gettydefs，以便为调制解调器增加记录(注意到这里的记录仅供参考，它们并非固定速度)：

```
# Modem entries
38400# B38400 CS8 # B38400 SANE -ISTRIP HUPCL #QS @L @B login:
#19200
19200# B19200 CS8 # B19200 SANE -ISTRIP HUPCL #QS @L @B login:
#9600
9600# B9600 CS8 # B9600 SANE -ISTRIP HUPCL #QS @L @B login:
#2400
2400# B2400 CS8 # B2400 SANE -ISTRIP HUPCL #QS @L @B login:
#1200
1200# B1200 CS8 # B1200 SANE -ISTRIP HUPCL #QS @L @B login:
```

```
#300
300# B300 CS8 # B300 SANE -ISTRIP HUPCL #@S @L @B login: #38400
```

如果用户将调制解调器设成 RTS/CTS, 进行硬件流量控制, 那么用户可以将 CRTSCTS 加到这些记录中:

```
# 38400 fixed speed with hardware flow control
F38400# B38400 CS8 CRTSCTS # B38400 SANE -ISTRIP HUPCL CRTSCTS
#@S @L @B login: #F38400
# 19200 fixed speed with hardware flow control
F19200# B19200 CS8 CRTSCTS # B19200 SANE -ISTRIP HUPCL CRTSCTS
#@S @L @B login: #F19200
# 9600 fixed speed with hardware flow control
F9600# B9600 CS8 CRTSCTS # B9600 SANE -ISTRIP HUPCL CRTSCTS #@S
@L @B login: #F9600
```

如果需要, 用户可以让 `uugetty` 在登录时的欢迎信息中显示有趣的内容。在本例中, 显示系统名称、串行线路和目前的传输率。用户也可以加入其他的内容:

```
@B The current (evaluated at the time the @B is seen) bps rate.
@D The current date, in MM/DD/YY.
@L The serial line to which getty is attached.
@S The system name.
@T The current time, in HH: MM: SS (24-hour).
@U The number of currently signed-on users. This is a
count of the number of entries in the /etc/utmp file
that have a non-null ut_name field.
@V The value of VERSION, as given in the defaults file.
To display a single '@' character, use either '\@' or '@@'.
```

接下来, 确定调制解调器所接的串行端口有输出及输入的设备。

如果调制解调器设在 `ttyS3`, 用户将会需要 `/dev/cua3` 和 `/dev/ttyS3` 这两个设备文件。

如果没有正确的设备, 用户可以通过命令 `mknod` 来创建这些设备。`mknod` 的具体使用可参阅它的联机手册。

3.5.5.3 定制 uugetty

许多参数可以用来改变端口的属性, 以分别对每一个端口使用不同配置文件的方式来实现的。

`/etc/conf.uugetty` 文件会被所有的 `uugetty` 所使用, 而 `/etc/conf.uugetty.ttySN` 文件只由一个端口使用。默认的配置文件的范例可以在 `getty_ps` 的源文件里找到, 这在绝大部分的 Linux 发行软件包里都有。要注意的是, 如果用户用的是比较旧的 `getty` 版本(比 2.0.7e 版还旧), 或者是用户没有依照 Linux 的文件系统标准配置, 那么默认的文件将会是 `/etc/default/uugetty.ttySN`。

正确的 `/etc/conf.uugetty.ttyS3` 文件应如下所示:

```
# sample uugetty configuration file for a Hayes compatible
# modem to allow
# incoming modem connections
#
# alternate lock file to check... if this lock file exists,
# then uugetty is
# restarted so that the modem is re-initialized

ALTLOCK=cua3
ALTLINE=cua3

# line to initialize

INITLINE=cua3

# timeout to disconnect if idle...

TIMEOUT=60

# modem initialization string...
# format: <expect> <send> ... (chat sequence)

INIT="" AT\r OK\r\n
WAITFOR=RING
CONNECT="" ATA\r CONNECT\s\A

# this line sets the time to delay before sending the login
# banner

DELAY=1

#DEBUG=010
```

将下面一行内容加到/etc/inittab 文件中，uugetty 就会在用户的串行设备上等待用户的连接请求：

```
S3: 456: respawn: /sbin/uugetty ttyS3 F38400 vt100
```

重新启动程序：

```
Linux# init q
```

对于在文件 inittab 中的速度参数，最好使用调制解调器所能支持的最高传输率。因此，在文件 inittab 中使用 38 400bps。

以后 Linux 将会随时监视用户的串行端口是否有连接。

uugetty 还有更多的选项，参阅 getty (1M)联机手册可获得更详细的信息。

3.5.5.4 Supra注意事项

Supra 调制解调器对于 DCD 的处理与其他调制解调器不同。如果用户使用 Supra, 必须要设置 `&C0` 而不是 `&C1`。此外, 还必须设置 `&D2`, 以便正确地处理 DTR 信号。

3.5.6 设置终端连接到 PC 机上

下面讲述如何设置终端连接到 PC 机上。

3.5.6.1 硬件需求

确定有一条正确的连接线——哑终端连接线就可。确定用户使用的是串行端口, DB25 或是 DB9 接口, 而不是并行端口。

如果使用的并不是完全的哑终端连接线, 用户可能要使用下面的方法:

在用户计算机这边的接口上, 把 RTS 与 CTS 接在一起, 并把 DSR、DCD 和 DTR 接在一起。用这个方法, 当计算机要求一个正确的硬件连接联络信号时, 它就可以得到该信号(从它自己发出的)。

现在用户已经有正确的连接线, 把终端连接到了用户的计算机上。如果可以, 使用户的终端忽略调制解调器控制信号。试着把终端设置为: 9 600bps 的速度、8 位数据传输模式以及无奇偶校验位。

3.5.6.2 设置getty

参考前文将 `getty_ps` 安装好。在 `/etc/gettydefs` 文件中增加一个给 `getty` 用来连接用户终端的记录:

```
# 38400 bps Dumb Terminal entry
DT38400# B38400 CS8 CLOCAL # B38400 SANE -ISTRIIP CLOCAL #@S @L login:
#DT38400
# 19200 bps Dumb Terminal entry
DT19200# B19200 CS8 CLOCAL # B19200 SANE -ISTRIIP CLOCAL #@S @L login:
#DT19200
# 9600 bps Dumb Terminal entry
DT9600# B9600 CS8 CLOCAL # B9600 SANE -ISTRIIP CLOCAL #@S @L login:
#DT9600
```

如果需要, 用户可以让 `getty` 在登录的欢迎信息中上显示一些有趣的内容。在本例中, 显示系统名称以及串行线路。用户也可以加入其他的内容:

```
@B    The current (evaluated at the time the @B is seen) bps rate.
@D    The current date, in MM/DD/YY.
@L    The serial line to which getty is attached.
@S    The system name.
@T    The current time, in HH: MM: SS (24-hour).
```

```
@U   The number of currently signed-on users. This is a
      count of the number of entries in the /etc/utmp file
      that have a non-null ut_name field.
@V   The value of VERSION, as given in the defaults file.
To display a single '@' character, use either '\@' or '@@'.
```

编辑用户的/etc/inittab 文件, 使 `getty` 在串行端口上执行(依照用户自己的环境: 端口、速度和默认终端模式, 将正确的信息填入):

```
S1: 456: respawn: /sbin/getty ttyS1 DT9600 vt100
```

重新启动程序:

```
Linux# init q
```

现在用户应该会在终端上看到登录提示。有时用户需要按回车键几次才能使终端显示登录提示。

3.5.6.3 将 PC 机设置成终端的注意事项

许多人将其 PC 机设置成终端连接到 Linux 机器上。例如, 旧的 8088 或是 286 PC 非常适用于这种情况。用户需要的只是一个包含适合让终端-个人计算机(Terminal-PC)使用的 DOS 调制解调器, 以及让终端-个人计算机执行的通信程序。`kermi`t 是一个非常好的仿真终端程序, 用户可以在 atsun.cc.columbia.edu/pub/ftp/kermi 站点上找到适合于现有的任何操作系统使用的各种 `kermi`t 版本。使用其他的 DOS 通信程序, 例如 `telix` 或 `procomm` 也会有好的效果。

3.5.7 使用两个以上的串行设备

除非用户要使用 3 个或更多的串行设备(假定用户没有多用户卡), 否则可以跳过本节。

用户可以使用的串行端口数目受限于必须使用的硬件中断要求以及端口输入/输出地址的数目。这并不是 Linux 的限制, 而是 PC 机和硬件的限制。

必须给每一个串行设备指定中断和端口地址。串行设备是串行端口、内置式调制解调器或多用户卡。

3.5.7.1 选择串行设备的中断

用户的 PC 机一般会将 `ttyS0` 和 `ttyS2` 设在 `IRQ4`, 而将 `ttyS1` 和 `ttyS3` 设在 `IRQ3`。如果使用两个以上的串行设备, 用户必须放弃其中的一个中断。

可以从用户的并行端口重新指定一个中断。用户的 PC 机一般会设置 `IRQ5` 或 `IRQ7` 作为用户的并行端口, 也有少数人同时使用两个并行端口。用户可以给串行设备重新指定其中的一个中断, 并且仍然能顺利地使用并行端口。这时用户需要用到 `setserial` 程序。

此外, 用户必须调整主板上的跳线设置, 查看主板的说明文件, 设置用户想要给每个端口使用的 `IRQ`。

用户将需要做些设置，使每个串行设备都有且只有一个中断。以/etc/rc.local 文件中所用的设置为例，在启动系统时用户应该在某个文件中加入下面的内容：

```
/etc/setserial /dev/cua0 IRQ 3      # my serial mouse
/etc/setserial /dev/cua1 IRQ 4      # my Wyse dumb terminal
/etc/setserial /dev/cua2 IRQ 5      # my Zoom modem
/etc/setserial /dev/cua3 IRQ 9      # my USB modem
```

标准的 IRQ 设定如下：

IRQ 0	计时器通道 0
IRQ 1	键盘
IRQ 2	磁带控制器 2
IRQ 3	串行口 2
IRQ 4	串行口 1
IRQ 5	并行口 2
IRQ 6	软盘驱动器
IRQ 7	并行口 2
IRQ 8	实时时钟
IRQ 9	重定向到 IRQ2
IRQ 10	保留
IRQ 11	保留
IRQ 12	保留
IRQ 13	数学协处理器
IRQ 14	硬盘控制器
IRQ 15	保留

在选择中断时，要确定中断没有被主板或其他网卡所使用。IRQ 是 2、3、4、5 或 7。“保留”的意思是目前没有什么设备在标准情况下会固定使用这些硬件中断。注意到 IRQ2 与 IRQ9 是一样的，用户可以称它为 2 或 9，串行驱动程序会接受该设置的。

如果有一张 16 位的串行卡，也可以使用编号为 10、11、12 或 15 的硬件中断。

要确定用户使用的 IRQ 不是 0、1、6、8、13 或 14，它们是用户的主板所使用的。

3.5.7.2 设置串行设备的地址

接下来，用户必须设置端口地址。查看用户主板手册的跳线设置。就像中断，每个地址只能有一个串行设备。用户的端口通常会如下配置：

```
ttyS0 address 0x3f8
ttyS1 address 0x2f8
ttyS2 address 0x3e8
ttyS3 address 0x2e8
```

选择用户想要给每一个串行设备使用的地址并且以此来进行跳线设置。例如：将调制

解调器设在 ttyS3, 鼠标设在 ttyS0, 而终端设在 ttyS2。

当重新启动时, Linux 应该会在用户所设置的地址查看到用户的串行端口。Linux 所查看到的 IRQ 可能与用户跳线设置的 IRQ 不相符。不必担心, Linux 在它启动的时候不做任何 IRQ 检测。使用 `setserial` 程序指明串行端口使用哪个 IRQ。

3.5.8 将串行端口设到较高的速度

本节介绍通过通信程序或 `getty` 程序使用调制解调器时, 如何确定使用速度。

- 如果使用的调制解调器比 9 600bps(V.32) 调制解调器还要慢, 那么将速度设到调制解调器所能支持的最高速度。例如 300bps、1 200bps 或 2 400 bps。
- 如果使用 9 600bps(V.32)的调制解调器, 并且支持 V.42bis 数据压缩。就使用 38 400bps 作为用户串行端口使用的速度。V.42bis 的压缩比是 4: 1, 因此实际的传输速率是 “4 * 9 600 = 38 400”。
- 如果使用 14 400bps(V.32bis)的调制解调器, 并且支持 V.42bis 数据压缩的功能, 可以使用 `setserial` 程序, 加上 `spd_hi` 标志将用户的串行端口配置成使用 57 600bps(4 * 14 400 = 57 600)的速度。
- 如果使用 28 800bps(V.FC 或 V.34)的调制解调器, 则使用 `spd_vhi` 标志(4*28 800=115 200)。

然后, 在用户的通信程序或是 `/etc/inittab` 文件中使用 38 400 作为速度设置。这是目前能设的最高速度。要确定有 16550A 通用非同步接收转换器(UART)。

将用户所做的修改放进 `/etc/rc.d/rc.serial` 或 `/etc/rc.d/rc.local` 文件, 以便在系统启动时执行。

例如, 在 `/etc/rc.d/rc.local` 文件中设置 ttyS3 为 115 200 bps, 如下所示:

```
/sbin/setserial /dev/cua3 spd_vhi
```

3.5.9 通信程序及工具

一旦用户使每件事都能正确运行, 就可能会想要找些更高级的程序。如果所使用的发行软件包里没有, 那么需要到站点上下载相应的软件包。

这些软件包包括:

- `ecu` -- 一套通信程序。
- `minicom` - 类似 `telix` 的通信程序。
- `procomm` -- `procommish` 包含 `zmodem` 的通信程序。
- `seyon` - 以 X 为基础环境的通信程序。
- `xc` - `xcomm` 通信软件包。这些程序比单独的 `kermit` 提供更多的功能, 包括电话目录和自动拨号等。
- `term-term` 在一条串行线路上多重模拟连接。连接方式包括 `rlogin`、`telnet`、`ftp`、

finger、rdate、xmosaic 和 tredit。

tredit 是让用户能够重新引导远端的 TCP/IP 端口到本地主机上的特殊程序。它允许远端的 NNTP 和 SMTP 存取。term 的优点是它完全在使用者的空间中执行，即它不需要内核的支持或是系统管理上的支持(像 SLIP 所做的一样)。

- screen 多区段会话的程序。这个程序的功能就像虚拟控制台。
- callback- 这个程序会使调制解调器从用户刚刚被中断的地方立刻回拨给用户。
- mgetty+fax 处理传真事务，并且提供一个 getty 功能。
- ZyXEL ZyXEL U-1496 调制解调器的控制程序。它处理接受拨号、回拨的保密、传真和语音信箱功能。

其他程序可以在下面的站点找到：

sunsite.unc.edu /pub/Linux/system/Serial

和

sunsite.unc.edu /pub/Linux/apps/comm

或是其镜像站点。

3.5.10 串行通信程序使用技巧

以下介绍一些很有用的串行通信软件。

3.5.10.1 kermitt和zmodem

要在 kermitt 下使用 zmodem，则需将下列内容加入到用户的.kermrc 文件中：

```
define rz !rz < /dev/cua3 > /dev/cua3
define sz !sz \%0 > /dev/cua3 < /dev/cua3
```

要正确地定义端口的名称，使用时只需在 kermitt 的提示符下键入“rz”或“sz<文件名>”即可。

3.5.10.2 自动设置终端模式

如果要在登录的时候自动设置终端模式，需把终端的模式加入到/etc/inittab 文件的记录中。如果有一台 vt100 终端在 ttyS1 上，需要在 getty 命令中加入“vt100”：

```
S1: 456: respawn:/sbin/getty ttyS1 DT9600 vt100
```

也可以从 sunsite.unc.edu/pub/Linux/system/terminal-management 或其镜像站点取得 tset 程序。可以参阅随附于 tset 的文件学习它的使用方法。

tset 可以在用户登录的时候测试终端的特性。

3.5.10.3 串行连接下的彩色ls

如果 ls 因为彩色特性而扰乱了用户的终端模拟，可以把它关掉。

Ls--color 和 ls--colour 都使用彩色特性。有些安装方式使得 ls 默认为使用彩色。检查

/etc/profile 文件和/etc/cshrc 文件中的 ls 别名。如果不想修改系统默认值，用户也可以给 ls 设别名 ls--no-color。

3.5.10.4 从终端所连接的打印机进行打印

由 Garrett D'Amore(garrett@sdsu.edu)所编写的 vtprint 程序可以完成该功能。
这个程序可以从下面的站点获取：

`ftp.sdsu.edu/pub/vtprint`

也可以从下面的站点获取：

`http://www.sdsu.edu/~garrett`

vtprint 允许用户从远端 UNIX 主机打印至连接到他们本地终端或终端模拟器的打印机，它使得在家里打印文件等工作变得很方便。

3.5.10.5 Linux能不能自动地配置串行设备

要让 Linux 在启动的时候自动地检测并且设置串行设备，则把如下 一行：

`/sbin/setserial /dev/cuaN auto_IRQ skip_test autoconfig`

加到/etc/rc.d/rc.local 文件或/etc/rc.d/rc.serial 文件中。

对每一个用户希望自动配置的串行端口都要那样做。要确定所给的设备真正存在于计算机上。

3.5.10.6 多用户卡的注意事项

有关卡的地址和 IRQ 的信息可查阅 setserial 程序所附的 rc.serial 文件。它包含很多关于多用户卡的信息，如输入/输出地址和设备名称等。

3.6 X Window

3.6.1 介 绍

不用命令行的界面，用基于 X Window 的图形用户界面也可以与 Linux 系统进行交互。采用图形用户界面，用户可以使用图标、窗口和菜单来发布命令和运行程序。与普通的基于个人计算机的图形用户界面，如 Windows 或者是 Mac OS 不同的是，Linux 和 UNIX 系统把图形用户界面分成三个独立的部分：

1. X Window
2. 窗口管理器
3. 程序/文件管理器

X Window 是底层的标准图形工具，它提供所有基本的图形操作，如打开窗口或显示图像。窗口管理器负责处理窗口的操作，如改变大小或者移动窗口等。不同的窗口管理器在窗口被显示的方法上有所不同，使用不同的边框和窗口菜单。但是所有的窗口都采用了同样的底层 X Window 图形工具。文件管理器负责处理文件的操作，它使用图标、菜单和一个程序管理器来运行程序，通常允许用户从一个工具条上选取常用的文件操作命令。与窗口管理器不同，文件和程序管理器可能在外观和性能上差别很大。

Caldera 桌面是一个商业的文件和程序管理器。Caldera 桌面和 xfm 文件管理器都可以运行在 fvwm 窗口管理器上。

所有 Linux 和 UNIX 系统都使用同样标准的底层 X Window 图形工具。这意味着在大多数情况下，基于 X Window 的程序可以在任何一个窗口管理器和桌面的系统上运行。

基于 X Window 的软件常常可以在 Linux 和 UNIX 的 FTP 站点上看到，它们通常标注为 X11。可以下载这些文件，然后在 Caldera 桌面上的 fvwm 下直接运行它们。

有些文件是 Linux 的二进制形式，用户可以下载、安装，然后直接运行，另一些是源码的形式。一些应用程序，如 Motif 等，可能还需要一些特别的库。

3.6.2 窗口、文件和窗口管理器

使用窗口管理器，用户可以用鼠标来完成如打开、关闭、改变大小和移动窗口等的窗口操作。有些窗口管理器可以用在 Linux 系统上，还有一些更通用的窗口管理器，如 fvwm、fvwm95、LessTif(mwm)、Motif(mwm)、Xview(olwm)和 AfterStep(afterstep)。

fvwm 是 Robert Nation 写的用于绝大多数 Linux 系统上的主窗口管理器。它使用简单、功能强大，而且灵活。Fvwm 95 是一个有着类似于 Windows 95 的用户界面的 fvwm。它使用与 Windows 95 操作系统相类似的窗口、任务栏和菜单。LessTif 是 Motif 的免费的姊妹程序。所有 Motif 的应用程序都可以在 LessTif 下运行。Motif 是一个像 UNIX 一样的商业产品。Xview 是 Linux 版本的 Sun 系统的 OpenLook 界面(不是 Solaris)。AfterStep 是用于 NeXT 操作系统上的 NeXTSTEP 界面的姊妹程序。

窗口管理器在底层的 X Window 图形工具之上运行。X Window 实际上提供了允许用户打开、移动和关闭窗口以及显示菜单和选择图标这样的基本操作。fvwm 和 Xview 用各自的方式管理这些操作，提供不同的界面以供选择。所有的窗口管理器，无论它们外观上有多么大的差别，都是使用 X Window 工具。从这个意义上说，Linux 不是只限定于一种图形用户界面。在同一个 Linux 系统上，用户可以使用 fvwm 窗口管理器，也可以使用 Xview，甚至使用 Motif。

窗口管理器一般仅提供非常基本的窗口管理操作，如打开、移动和改变窗口大小。文件管理器和程序管理器使用户能用图标和菜单来管理和运行程序。使用文件管理器，用户可以在不同的目录窗口中拷贝、移动或者删除文件。使用程序管理器，用户可以用任务栏和程序图标来执行命令和运行程序。一个桌面程序利用文件管理器和程序管理器的功能来提供一个有菜单和图标的虚拟的桌面用以访问文件和管理程序。

有些窗口管理器还包括了桌面的一些特点。在 Linux 系统上使用的 fvwm 多数不但具

有窗口管理功能还具有程序管理功能。fvwm 具有一个任务栏和一个工作区菜单,使用菜单或者任务栏,用户可以在 fvwm 中直接运行任何 X Window 程序。其他的窗口管理器,如 fvwm95,具有和 Windows 95 一样的任务栏和初始菜单的桌面界面。fvwm95 基于 fvwm2、使用 Windows 95 的界面。除了支持程序管理功能, fvwm95 还将支持一个名叫 Explorer 的文件管理器。使用 fvwm95,用户可以像在 Windows 95 中那样显示文件和目录的图标。

3.6.3 启动和退出 X Window 窗口管理器和桌面

在命令行方式下输入 starx 启动 X Window。X Window 服务器将在窗口管理器运行之后加载。把 Linux 系统设置为在 Caldera 桌面中自动加载窗口管理器。桌面也是运行在窗口管理器里的。Caldera 加载窗口管理器后,可以看到: Caldera 是棕色或灰色的窗口。窗口上方是任务栏,窗口中显示的是程序的图标。可以用鼠标来访问菜单、打开窗口以及运行程序。

从文件菜单中选择退出桌面选项可以退出 Caldera 桌面。像许多窗口应用程序一样,Caldera 桌面在屏幕的上方有一个菜单栏。单击文件菜单就可以弹出其下拉菜单。退出桌面是其中最后一个选项。

当选择退出桌面选项的时候, Linux 系统仅关闭 Caldera 桌面,不关闭 fvwm 窗口管理器。

fvwm 窗口管理器在屏幕上的背景为深蓝色。用户必须退出窗口管理器才能关闭 X Window 退回到 Linux 系统的命令行方式界面。

要关闭 fvwm 窗口管理器,首先要打开 fvwm 工作区菜单。在深蓝色的背景的任意处单击鼠标就可以弹出这个菜单。用户必须单击菜单然后把选中条往下拉一点才能让这个菜单停留在屏幕上。选择这个菜单的最后一项,打开一个子菜单,该子菜单中的一个选项就是退出桌面选项。选择此选项就会退出窗口管理器和 X Window,回到有\$提示符的命令行方式界面。在命令行方式下,用户随时可以通过输入 starx 命令来启动 fvwm。

用户还可以在 fvwm 中启动其他的窗口管理器、桌面或 fvwm 的任务栏。在工作区菜单中选择桌面选项,将打开一个窗口管理器的子菜单。单击这个菜单中的开启任务选项将显示 fvwm 的任务栏,该任务栏上有一些 X Window 经常访问的程序的图标,如万维网浏览器。选中 Start Looking Glass 选项可以开启 Caldera 桌面(Looking Glass 是 Caldera 桌面的另一个名字)。选中窗口管理器选项将弹出一个子菜单,该子菜单上列出系统上已经安装的其他窗口管理器。单击子菜单中的其他的窗口管理器选项可以启动相应的窗口管理器。

3.6.4 X Window 系统中的设置工具

在 X Window 系统中,运行相应的设置工具,用户可以安装软件、设置打印机或添加

新用户等。

XF86Setup 是用来设置 X Window 系统的工具，就像 glint 用来安装软件一样。Lisa 是一个基于菜单的管理工具，它包括了系统管理的各个方面。在 Linux 的 Redhat 发行版本中还有一些基于窗口的工具，可以方便地设置打印机、用户和文件系统。

这些工具可以从 Caldera 桌面、fvwm 菜单或 fvwm 任务栏中启动。为访问这些工具，首先必须以 root 身份登录。然后用 startx 命令启动 X Window。在 Caldera 桌面上可以看到一个管理工具图标。双击管理工具图标，打开管理工具窗口，在窗口中可以看到每个工具的图标。

3.6.5 xterm

在 Linux 的窗口管理器中可以打开终端窗口，终端窗口提供一个标准的命令行界面。用户可以在此窗口的命令行上的提示符后面输入带有参数的命令。这个终端窗口是 xterm 程序创建的。在 fvwm 的任务栏、程序菜单或者像 Caldera 这样的桌面中可以开启一个终端窗口。

fvwm 任务栏上有一个 xterm 按钮，单击它即可打开一个 xterm 窗口。从 fvwm 工作区菜单中选中 Shells，再从其子菜单中选中 xterm 选项也可以打开一个 xterm 窗口。要从 Caldera 桌面上打开一个终端窗口，只要双击桌面左上角图标栏里的终端图标即可。

一旦打开，终端窗口就显示一个 Shell 提示符，一般是“\$”，用户可以像在命令行方式下一样输入命令。输入的 Linux 命令的执行结果将在终端的窗口上显示，其后是一个 Shell 提示符，表明命令行的开始。必须注意的是，终端窗口必须是活动窗口才能使用它的命令行。如果它不是活动窗口，需要移动鼠标指针使它变为活动窗口。

xterm 窗口支持几个文字处理特性。在窗口的左边是一个滚动条，使用它可以向前滚动来查看显示过的文本。当文本位于 xterm 窗口的一页时，可以通过回滚来查看它。如果想显示某个目录下的文件，而文件过多，无法在一个窗口内显示的时候，这个功能就很有用处。

xterm 窗口中能进行拷贝和粘贴文本操作，方法是：用鼠标的左键来拷贝文本，用第二个键来粘贴文本。用户可以拷贝先前显示的任何文本，如先前的命令或命令的输出结果。要拷贝文本，可以按住鼠标的左键拖动到目标处松开鼠标。双击将选择一个单词，三击将选择一行句子。如果想扩大选择范围，用鼠标的第三个按键。选择好文本后，单击鼠标的第二个按键，就可自动把文本粘贴到命令行的末尾。重复单击就会重复粘贴。在不同的终端窗口间也可以拷贝和粘贴。（术语“第二个”和“第三个鼠标按键”可能容易混淆。在一个双键鼠标上，第二个按键就是指右键，而第三个键就是指左右键同时按下。在一个三键鼠标上，第二个键就是指中键，而第三个按键就是指最右边的按键。这三个按键可用来访问 xterm 的菜单。）

一个 xterm 窗口有四个菜单：

主菜单、虚拟终端选项菜单、虚拟终端字体菜单和 TektroniX Window 选项菜单。

要弹出主菜单，按住 Ctrl 键并单击鼠标左键。要弹出虚拟终端选项菜单，按住 Ctrl 键

并单击鼠标的第二个(右边或者中间)按键。要弹出字体菜单,用 Ctrl 键和鼠标的第三个按键,从中可以设置字体和字符显示的大小。

xterm 窗口可以模拟 DEC VT 102 终端和 Tektronix 4014 终端。要显示 TektroniX Window,可以在虚拟终端选项菜单中(Ctrl 键和第二个按键)选择这个选项,会弹出一个可以显示图形的 TektroniX Window。

在 TektroniX Window 中,按住 Ctrl 键并单击鼠标的第二个按键,可以打开一个 TektroniX 选项菜单。

终端窗口有一个特别之处,即可以在它的命令行里运行任何 X Window 程序。这个终端窗口在 X Window 环境中运行。要运行 X Window 程序,只需打开一个 xterm 窗口,输入命令,再按一下回车键即可。

例如,要运行一个 Netscape,可以打开一个终端窗口,然后输入命令“netscape”。一个新的窗口就会打开并运行 Netscape。可以打开任意多个终端窗口,可以在它们中的任何一个中启动另一个 X Window 程序。要注意关闭一个终端窗口同时也会关闭从这个终端窗口中启动的程序。

用户会注意到输入 X Window 命令的终端窗口会把自己挂起。当按回车键运行程序以后它并没有立即显示命令提示符。那是因为终端窗口正忙于运行刚刚执行的 X Window 程序。可以通过带有一个“和”符号(&)的命令来激活该程序使终端窗口释放自己,在那个程序运行的同时执行其他的命令。从技术上来说,这是把该程序放在该终端窗口的后台运行。下例将运行 Netscape,并释放终端窗口来执行其他的命令。注意,在输入“netscape &”后,提示符立即出现,而不是处于挂起的状态。

```
$ netscape &  
$
```

当使用完终端窗口时,在命令行输入一个 exit 命令来结束它。每个终端窗口都是它自己的 Shell,Exit 就是结束这个 Shell 的命令。

3.6.6 X Window 多任务

Linux 的 X Window 界面最有用的特性之一就是它可以同时打开几个任务,每一个任务都有自己的窗口。发布一个命令,执行完该命令后,还可以执行另一个。在 X Window 中,用户可以同时运行几个不同的应用。把鼠标指针从一个窗口移动到另一个窗口,会有有效地使用户从一个应用切换到另一个应用。X Window 的这个特性表明了 Linux 最有用的特性之一——并发性,即可以同时运行多个进程的能力。

在 X Window 界面中,可以同时运行多个应用,每个应用都有自己的窗口。可以很容易地用终端窗口来演示这个特性。在同一时刻可以有多个终端窗口,每个窗口都有自己的命令行。要从 Caldera 桌面上打开一个终端窗口,双击图标栏的终端图标;在 fvwm 中,使用工作区菜单或 fvwm 图标栏上的 xterm 按钮。要打开第二个终端窗口,再次双击该图标即可。还要再打开一个终端窗口,只要再次双击这个图标,依此类推。每个终端窗口都有自己的命令行,把鼠标指针从一个窗口移动到另一个窗口可以从一个命令行切换到另一

个命令行。

用户可以在一个活动窗口中输入命令并执行它，然后切换到另一个终端窗口再输入另一个命令。每个终端窗口都相互独立地运行。如果发布的一个命令需要一段时间来运行，当用户移动到另一个窗口时，可以看到刚刚离开的那个窗口中的命令仍然在继续执行。

3.6.7 fvwm 窗口管理器和桌面

fvwm 窗口管理器不仅提供了基本的窗口、任务栏和菜单工具，而且还支持虚拟桌面等高级功能。用 fvwm 窗口管理器，用户可以有多个桌面，每个桌面都有不同的窗口和程序。

3.6.7.1 fvwm 工作区菜单

用鼠标单击工作区然后稍微向下拖动，就会弹出工作区菜单。这个主菜单包含子菜单，其中一些子菜单本身也包含子菜单。例如，选择应用选项，将弹出其子菜单，其中列出当前所有的 X Window 程序类别。如果用户选择 xpaint，xpaint 程序将会开始运行。

当安装或者删除一个 X Window 程序的时候，fvwm 将会自动更新工作区菜单。经常使用的 X Window 程序项目已经被设置。如果把 fvwm 设置成检查在 /usr/X11r6/bin 和 /usr/openwin/bin 目录下的任何新程序，这些程序将被加到工作区菜单中。

要退出 fvwm，从工作区菜单中选择退出桌面选项，接着将弹出一个包括停止桌面选项的子菜单。当选择这个选项，fvwm 将会与 X Window 一同退出。

3.6.7.2 GoodStuff 任务栏

fvwm 的最新版本包括 GoodStuff 任务栏，任务栏上的按钮用来显示经常使用的 X Window 命令。GoodStuff 任务栏将在启动 fvwm 窗口管理器的时候被显示；如果没有显示任务栏，可以通过在工作区菜单的桌面菜单中选择启动 GoodStuff 选项启动它。每个按钮都将显示程序的名字。只要单击按钮就可以启动相应程序。例如，要打开一个 xterm 窗口，只需要单击 xterm 按钮。

使用 WWW 按钮将启动因特网浏览器。使用 xfm 按钮启动 xfm 文件管理器，通过菜单和图标可以访问系统上的目录和文件。如果用户选择标准安装方式，那么可能没有安装所有程序。如果没有，用户将被问到是否愿意安装它们。

要使 GoodStuff 任务栏永久地显示在 fvwm 桌面上，在 fvwm 工作区菜单所在的桌面上的设置菜单的 GoodStuff 子菜单中选择位置选项。用户可以选择把它放在屏幕的顶部、中间、左边或者右边。用户可以随意地移动或者重置任务栏的大小。要移动任务栏，用鼠标右边的按钮单击它的边缘，然后拖动到想移动到的地方。要重置大小，用鼠标左键单击它的边缘或者一角并拖动。

如果不想再显示任务栏，在 fvwm 工作区菜单所在的桌面上的设置菜单中的 GoodStuff 子菜单中选择无效选项。

任务栏的配置文件为 system.fvwmrc.goodstuff，它位于 /usr/lib/X11/fvwm 目录下。文

件中含有任务栏的按钮的记录。用户可以通过在一行的前面加上一个“#”来删除一个按钮。还可以给新按钮添加其他行。一个按钮记录用关键字*GoodStuff 开始,其后是用户想显示的按钮和想在按钮上显示的图片。行末是要运行的命令,通常用 Exec 命令开始,接着是一个空字符串和 xlaunch 命令然后是程序名。下例给 Crisplite 编辑器添加一个按钮。激活 Crisplite 编辑器的命令是 mcr。

```
*GoodStuff Crisplite dtp.xpm Exec "" xlaunch mcr
```

.xpm 文件就是用来显示的图片文件。用户可以在 /usr/share/data/pixmaps 和 /usr/openwin/lib/pixmaps 目录下找到很多这样的文件。

3.6.7.3 虚拟桌面

开始用户可能会为虚拟桌面而不安,因为虚拟桌面它含有一种内置的放大特性。用户将发现在屏幕上显示的区域可能只是桌面的一部分。移动鼠标指针到屏幕边缘将移动到桌面的被隐藏的部分。用户还会注意到在桌面的右上角有一个小方块。这叫做 pager, 用它查看虚拟桌面的不同区域。这个 pager 被分成四个小正方形。用户可以把每个正方形想象成一个独立的桌面的扩充。虚拟桌面就像用户有一个很大的桌子,只有部分被显示在屏幕上。这个桌子活动的部分就是加亮的正方形,通常是白色的,这个正方形就是显示在屏幕上的那部分桌面。

通过移动这个加亮的正方形,可以移动到桌面的不同地方。记住用鼠标的右键来单击和拖动这个加亮的正方形。也可以单击 pager 四个部分中的一个来移动到桌面的对应部分。可以把不同的窗口放在桌面的不同部分,然后需要的时候再移动到那个部分中。用这种方法,用户放在桌面的所有内容就不一定要一起显示在屏幕上而使屏幕显得凌乱不堪。

如果在桌面上工作,突然所有的内容都不见了,这可能是因为碰巧单击了其中的一个正方形。只要再次单击右上方的正方形就会回到原来的状态。

不论虚拟桌面显示在哪个部分,某些项目始终显示在屏幕上。这些项目被称为“粘性(sticky)”项目。fvwm pager 与 fvwm 任务栏一样都是这种项目,例如, fvwm pager 总是显示在屏幕的右上角上,而不管用户在查看虚拟桌面的哪一部分。而窗口不是粘性的。

Caldera 桌面既可以是粘性地停留在屏幕上,也可以是作为一个窗口,保留在虚拟桌面上。

Caldera 任务栏最左边的图标用于在粘性图像和窗口之间切换。如果它是一个窗口,用户将看到桌面有一个标准的 fvwm 窗口边界,可以随意移动或者重置大小。作为一个粘性图像,则桌面没有边界。不论屏幕移动到虚拟桌面的哪一部分,Caldera 桌面都在屏幕上显示任务栏、菜单和图标。

开始虚拟桌面的屏幕区域可能不到 pager 的一个部分。可以在/etc/XF86Config 文件中把虚拟选项设置成显示器的分辨率。在大多数 15 英寸的显示器中,分辨率是 800*600。

大多数窗口管理器,包括 fvwm,也支持多种桌面。在这个意义上,桌面是一个包含了图标、菜单和窗口的屏幕。使用像 fvwm 这样的窗口管理器,用户可以使用多种不同的桌面。fvwm 可以支持 40 亿种桌面。

3.6.8 配置 X Window 系统

X Window 设计得十分灵活,用户可以通过各种方式配置它。X Window 可以在目前的大多数显示卡上运行,甚至也可以在图形加速卡上运行。

X Window 不拘泥于任何特定的桌面界面。X Window 提供一套底层的图形操作函数集,供用户界面应用程序(如窗口管理器和文件管理器)使用。窗口管理器利用这些操作函数来构建控制窗口的软件包,如滚动条、尺寸调整按钮以及关闭按钮。不同的窗口管理器能够构建出不同的软件包,从而提供不同的界面。所有的这些窗口管理器都能在 X Window 上运行,存在多种的窗口管理器供用户选择。

在系统中,各用户可以运行不同的窗口管理器,但各用户使用同一个 X Window 底层图形操作函数集。用户甚至能在没有任何窗口管理器和文件管理器的情况下运行 X Window 程序。

为了运行 X Window,必须安装与系统显示卡匹配的 XFree86 服务器软件包,并提供有关当前系统的显示器、鼠标以及键盘的配置信息。这些信息存放于 `/etc/XF86Config` 配置文件中。这个文件使用的配置信息最好由 X Window 配置程序生成,如 `XF86Setup`。在用户安装系统的过程中进行 X Window 配置时,会自动生成 `/etc/XF86Config` 文件。

用户可以通过 `.xinitrc` 和 `/usr/X11R6/lib/X11/xinit/xinitrc` 这两个配置文件来配置自己的 X Window 界面,通过这两个文件用户可以选择并且启动窗口管理器、文件管理器以及初始 X Window 应用程序(在 Linux 系统里,目录 `/usr/X11R6/lib/X11/xinit/` 是目录 `/etc/X11/xinit/` 的一个链接,文件实际存放在目录 `/etc/X11/xinit/` 中)。

还有一组特殊的 X Window 命令,用户可以用它们来配置根窗口、装入字体或者配置 X Window 资源,例如设定窗口边界颜色。用户还可以从联机资源下载 X Window 实用程序,那些作为 Linux 镜像站点的联机资源通常存放在 `/pub/Linux/X11` 目录下。如果不得不编译一个 X Window 应用程序,用户可能必须采取特殊的步骤来安装,有关这部分信息将在 3.6.14 节中介绍。

除了可以配置 X Window 之外,用户还可以配置窗口管理器。窗口管理器 `fvwm` 有它自己的一套配置文件,通过它们可以增加任务栏按钮或增加菜单项,甚至建立键盘映射。

3.6.8.1 XFree86 服务器

XFree86 服务器支持的显示卡和显示器范围很广,有支持单色、VGA 和 SVGA 的服务器,有为图形加速卡设计的一系列服务器,每种芯片组对应一种服务器。如需要更详细的信息,可查看位于目录 `/usr/X11R6/lib/X11/doc` 中的 `XFree86` 文件。在该文件中能找到对应于各种服务器的文件以及各服务器所支持的显示卡类型的信息。文件 `AccelCards` 列出了目前被支持的所有硬件,包括芯片组;文件 `Monitors` 列出了显示器配置信息。也可以查看不同驱动程序的 `man` 帮助页。各驱动程序及相应的 `man` 帮助页如下所示:

显示卡	<code>man</code> 帮助页
加速卡	<code>XF86_Accel</code>
单色显示卡	<code>XF86_Mono</code>

VGA 显示卡 XF86_VGA16

SVGA 显示卡 XF86_SVGA

安装系统时所指定的 XFree86 服务器会自动安装, 用户将不得不使用 `lisa`、`glint` 或者 `rpm` 来安装其他服务器, 它们存放在 CDROM 的目录 `Linux/install/xbasis1` 下。每个包含 XFree 服务器的软件包以“XFree86”开头。用户必须安装与当前的显示卡匹配的软件包。大多数标准显示卡能在 SVGA 服务器上工作。如果用户正在使用简单的单色或 VGA 显示卡, 只要安装一般的 XF86_Mono 或 XF86_VGA 服务器。但是, 如果装有图形加速卡, 必须首先确定它使用的是什么芯片组。查看随卡买回来的手册或文档。如果不能确定芯片的类型, 可以使用 `Superprobe` 实用程序, 它会分析当前的加速卡, 确定所需信息。

首先, 看一下 `Superprobe` 的 `man` 帮助页以便在启动它之前确定要使用的参数。一个特定服务器的名字中通常包含它所支持的芯片组的名字。例如, 如果正在使用的显示卡上有 S3 芯片组, 将使用 XFree86_S3 软件包。如果使用 `ViRGE` 芯片(`Diamond Stealth 3D 2000`), 将使用 S3V 服务器。在某些情况下, 不同品牌的显示卡使用同一种芯片。如果用户在 CDROM 中没有找到所需的服务器软件包, 可以查看任一 Linux 镜像站点寻找更新的版本(通常在目录 `/pub/Linux/xii/Servers` 下)。

通过 PRM 可以安装相应的 X Window 服务器, 下面是安装 S3 芯片组显示卡的例子:

```
rpm -i XFree86_S3-3.2-1.i386.rpm
```

3.6.8.2 /etc/XF86Config 文件

安装了 XFree86 服务器后, 就该配置 X Window 界面了。

这会涉及到配置文件 `XF86Config` 的生成, 它位于 `/etc` 目录下, 包含了有关当前图形卡、显示器、键盘以及鼠标的详细说明。用户可以直接创建这个文件, 但最好使用诸如 `XF86Setup` 或 `XF86Config` 之类的实用配置程序。用这些实用程序配置时, 只需回答有关硬件的一些问题, 然后程序会自动生成相应的 `XF86Config` 文件。

用户应该了解关于当前系统硬件的详细信息。

对于显示器, 用户必须知道它的水平和垂直同步扫描频率范围以及带宽。对于图形卡, 应该知道它的芯片型号, 有时甚至还应知道时钟频率。

对于鼠标, 应该知道它是 Microsoft 兼容系列还是其他品牌, 诸如 `Logitech`。同时还应知道鼠标所连接的端口号。

`XF86Config` 命令提供给用户一个全屏幕界面, 用户可以非常轻松地鼠标、键盘、图形卡和显示器选择各项特性。如果在用 `XF86Setup` 进行配置时碰到问题, 用户可以使用 `XF86Config` 命令。它的界面是命令行提示符形式, 可以敲入字符作为响应, 每一步都有提示, 使用户方便地完成 X Window 的配置。

文件 `/etc/XF86Config` 分为如下文所示的几个节。用户可以在 `XF86Config` 的 `man` 帮助页中找到关于所有这些节和对它们各项的详细信息。所有这些记录项都是 `XF86Setup` 程序建立的。例如, 显示器配置屏生成 `/etc/XF86Config` 文件中的显示器节、鼠标配置屏生成鼠标节等。文件里每一个节以关键词 `Section` 开头, 紧接着是用引号括起来的节的名字。每一个节以 `EndSection` 结束。注释行都以“#”开头。每一个节的各项都以数据说明开头, 然后是一系列取值。例如, 列出 `rgb` 色彩数据的 `Files` 节, 以数据说明 `RgbPath` 开头, 然

后是 rgb 颜色数据文件的路径名。

下面列出了/etc/XF86Config 文件中的各节及其说明:

Files	字体和颜色文件目录
Module	动态装载模块
ServerFlags	选项
Keyboard	键盘规格
Pointer	鼠标配置信息
Monitor	显示器配置信息
Device	显示卡配置信息
Screen	屏幕配置信息

尽管用户可以用标准的文本编辑器直接编辑文件/etc/XF86Config, 但最好是使用诸如 XF86Setup 之类的配置程序进行修改。用户可能从来不会去修改文件中的大多数节, 但是用户可能不得不对 Screen 节进行修改。为了这样做, 可以编辑文件并增加或修改 Screen 节的各项。由于 Screen 节是用户最有可能修改的节, 所以尽管它出现在文件最后, 还是最先讨论它。

1. Screen 节

Screen 节首先是 Driver 项, 它指定了驱动程序的名字。

有 5 种驱动程序, 每一种对应于一种 XFree86 服务器: Accel、Mono、SVGA、VGA2 和 VGA16。

驱动程序 Accel 用于诸如 S3_XFree86 之类的所有加速型 X Window 服务器。

Mono 用于 XF86_Mono 服务器支持的非 VGA 单色驱动器。

VGA2 和 VGA16 用于 VGA 服务器, SVGA 用于 XF86_SVGA 服务器。

如果使用 XFree86_SVGA 服务器, Driver 项将是 "svga"。如果使用任意一种加速型服务器, 这一项将是 "Accel"。诸如 XF86Setup 之类的配置程序将为以上这些显示方式生成 Screen 节。如果有一块加速卡并且能为它安装服务器, 那么 X Window 将使用 Accel Screen 节。如果正在使用 SVGA 服务器, 将使用 SVGA Screen 节。

在 Driver 项之后是 Device 和 Monitor 项, 它们分别指定用户正在使用的显示器和显示卡。

下面是/etc/XF86Config 文件中的一部分:

```
Section "Screen"
    Driver      "Accel"
    Device      "Primary card"
    Monitor     "Primary Monitor"
    DefaultColorDepth 16
    SubSection  "Display"
        Depth   8
        Modes    "1152x864" "1024x768" "800x600" "640x480"
               "640x400" "480x300" "400x300" "320x240" "320x200"
    Virtual 800 600
    EndSubSection
```

SubSection "Display"

Screen 节下有 Display 子节, 指定所支持的颜色深度。前面各部分是配置硬件的, 而 Display 子节则配置一些显示特征, 例如显示颜色数和虚拟屏幕尺寸等。子节中主要有三项: Depth、Modes 和 Virtual。

Depth 项指明屏幕分辨率: 8、16 或 24。不论当前的 X Window 服务器支持什么显示卡, 用户都可以加入 DefaultColorDepth 项为服务器设定默认颜色深度: 8 表示 256 色、16 表示 32K 色、24 表示 16M 色。

Modes 项指定在所给分辨率下允许的显示模式。Virtual 项指定虚拟屏幕的尺寸。用户能够拥有一个比正常显示面积大的虚拟屏幕。当移动鼠标至显示屏边缘时, 它将向隐藏的那部分屏幕移动, 通常将虚拟屏幕设定为 1024×768, 相当于 17 英寸显示器的尺寸。也可以设为 1152×864, 相当于 21 英寸显示器的尺寸。如果想取消虚拟屏幕, 可以把 Virtual 项设为 800×600, 使虚拟显示尺寸和屏幕物理尺寸大小相等。

下面是几种常见虚拟屏幕对应的空间大小:

Virtual 1024 768	17 英寸虚拟屏幕
Virtual 1152 864	21 英寸虚拟屏幕
Virtual 800 600	15 英寸屏幕(取消虚拟屏幕)

可以改变这一节中的任何特性。为了改变虚拟屏幕尺寸, 必须修改这一节。但是, 不要轻易修改 XF86Config 文件中的其他节, 除非用户胸有成竹。

2. Files、Modules、ServerFlags 和 Keyboard 节

Files 节列出了 XFree86 所需资源所在的目录。一个字体项以数据说明 FontPath 开头, 然后是那种字体的字体文件所在的路径名。下面是这些项的一部分:

```
RgbPath          "/usr/X11R6/lib/X11/rgb"
FontPath          "/usr/X11R6/lib/X11/fonts/misc:unscaled"
```

Module 节指定了一些动态加载的模块。Linux 的 ELF 系统支持动态模块。Load 项将加载一个模块。详细情况请查看 XF86Config 的 man 帮助页。

可以为 XFree86 服务器设置一些标志。用户可以在 XF86Config 的 man 帮助页中找到所有标志。例如, 标志 NoTrapSignals 允许清除内核, 以便调试程序。标志 DontZap 禁止用 Ctrl+Alt+Backspace 组合键关闭服务器。DontZoom 标志禁止图形模式之间的切换。

Keyboard 节决定键盘类型并且设定布局、模型和所用协议。例如, 下面这一项就设置了布局。这一节有很多选项。如想得到全部列表, 可参考 XF86Config 的 man 帮助页。

```
XkbLayout          "us"
```

3. Pointer 节

Pointer 节配置鼠标和其他所有定位设备。这一节只有几个项, 其中一些用于定制某些特定类型的鼠标。Protocol 项指定鼠标所使用的协议, 如 Microsoft 或 Logitech。Device 项指定了鼠标设备文件的路径名。

下面的例子显示了一个标准的 Pointer 节, 它配置的是一个波特率为 1200 的 Microsoft 鼠标。设备文件为 /dev/mouse。

```
Section "Pointer"
```

```

        Protocol      "Microsoft"
        Device         "/dev/mouse"
        BaudRate       1200
    EndSection

```

以下列出了 Pointer 节各项:

Protocol	鼠标协议(运行 <code>man XF86Config</code> 命令可得到完整的表)
Device	设备文件路径, 如 <code>/dev/mouse</code> 或 <code>/dev/cua0</code>
BaudRate	串行鼠标的波特率
Emulate3Buttons	允许二键鼠标通过同时按左右键来模拟三键鼠标
ChordMiddle	配置某些 Logitech 三键鼠标
ClearDTR 和 ClearRTS	清除 DTR 和 RTS 行, 只对 Mouse Systems 鼠标有效
SampleRate	设置抽样率(Logitech)

4. Monitor 节

在用户的系统上使用的每一台显示器都应该有一个 Monitor 节。正如前面所指出的, 垂直和水平扫描频率必须精确, 否则可能会损坏显示器。

Monitor 节开始的几项用来标识显示器, 如销售商和型号。HorizSync 和 VerRefresh 项指定了垂直和水平扫描频率。大多数显示器支持多种分辨率。Monitor 节中的各 ModeLine 项指定了分辨率。每一项 ModeLine 对应一种分辨率。每项有 5 个值:

分辨率名称、时钟频率, 然后是两组值(每组 4 个值), 一组是水平定时参数, 一组是垂直定时参数, 以标志位结束。

标志位指定显示模式的各种特征, 比如 Interlace 表明显示模式是隔行扫描的, 而 +hsync 和 +vsync 则是用来选择信号的极性。

```

Modeline "name" dotclock  horizontal-frep  vertical-frep  flags

```

如下是 ModeLine 的一部分。最好不要改动 Monitor 节, 而完全由 XF86Setup 配置程序生成各项。

```

Modeline "800 x 600" 50.0 800 856 976 1040 600 637 643 666 +hsync +vsync

```

下面列出 Monitor 节中通常使用的一些项:

Identifier	显示器标识名
VendorName	生产厂家
ModelName	样式和型号
HorizSync	水平扫描频率, 一个范围或一系列值
VerRefresh	垂直刷新频率, 一个范围或一系列值
Gamma	Gamma 校正
ModeLine	通过指定时钟频率、水平定时参数和垂直定时参数来确定相应的分辨率

5. Device 节

Device 节指定了系统显示卡。Device 节的开头几项用来标识显示卡, 比如 VendorName(销售商)、BoardName(显示卡名称)和 Chipset(芯片组)。

VideoRam 项指明了显存大小。Clocks 项列出了一组系统的时钟值。这一节中可以设定很多不同的项,比如,如果显示卡上有 Ramdac 芯片,可以用 Ramdac 项指定 Ramdac 芯片的类型;如果允许访问主缓冲区,MemBase 项指出了主缓冲区的基地址。详细列表和描述可参考 XF86Config 的 man 帮助页。

用户可以修改 VideoRam 项。例如,增加显示卡的内存,但最好别修改 Clocks 项。如果设错了时钟值,很容易损坏显示器,所以最好由 XF86Setup 或其他 XFree86 配置程序来生成时钟值。如果没有时钟值,服务器会自动确定。

3.6.9 X Window 和窗口管理器

用户可以放弃命令行界面而使用 X Window 窗口管理器和文件管理器,这样通过使用窗口、按钮和菜单,用户就能够与 Linux 系统进行交互。

窗口管理器提供了基本的窗口管理操作,例如打开、关闭窗口或改变窗口尺寸。文件管理器则允许用户使用图标和菜单来管理和运行程序。此外,使用文件管理器还能够拷贝、移动或删除文件,也能够分别为不同的目录打开不同的窗口。

为 Linux 设计的窗口管理器有多种。其中较为流行的 Linux 管理器有:Free Virtual Window Manager(fvwm)、Xview(olwm)、twm、fvwm95(fvwm95)、LessTif(mwm)、AfterStep(afterstep)以及 Motif(mwm)。

除了 Motif 之外,其他几种都是免费的。

Xview 是 Sun 公司的 OpenLook 界面的 Linux 版本。fvwm95 是 fvwm 的一个扩展,它提供包含任务栏的 Windows 95 界面。LessTif 是一个免费的 Motif 姊妹程序,它能够提供 Motif 界面,并能运行 Motif 应用程序。twm 是一个稍微旧一点的窗口管理器,它提供基本的打开窗口功能。在 Linux 上已经安装了 fvwm、twm 和 Xview。其他的窗口管理器可以从相应的因特网站或 Redhat contrib 目录下载。只要在相应的网址上有较新版本的窗口管理器,用户就可以下载并安装。

窗口管理器对底层的 X Window 系统进行操作,后者实际上提供各项基本的窗口操作,允许用户打开、移动和关闭窗口,以及显示菜单和选择图标。fvwm 和 Xview 分别以自己的方式管理这些操作,并分别具有不同风格的界面。这样设计的好处是用户可以用不同的窗口管理器操作同一个 Linux 系统。

从这层意义上来说,Linux 并不局限于一种图形用户界面(GUI)。在同一个 Linux 系统上,一个用户可能正在使用 fvwm 窗口管理器,另一个用户可能正在使用 Xview 窗口管理器,而第三个用户则在使用 fvwm95,所有用户都在同时使用系统。

fvwm 是 GNU 软件,用户免费拥有它。如果只使用 fvwm,用户仍然能够运行任何 X Window 程序。fvwm 有它自己的菜单和任务栏。用户也可以从 xterm 终端窗口运行任何 X Window 程序。从 fvwm 启动一个 xterm 窗口,输入 X Window 应用程序的名字,然后按回车键,X Window 应用程序将启动自己的窗口。最好通过在命令后面加符号“&”的方法以后台进程方式启动 X Window 应用程序。系统将为用户使用的 X Window 应用程序打开一个窗口。用户可以以后台方式运行 Caldera CDROM 上的任何 X Window 软件。X

Window 应用程序的名字经常是以字母 x 开头的, 但是也有例外, 比如 Netscape 和 Mosaic。用户可以从 Linux 的 FTP 站点下载各种各样的 X Window 软件。它们通常在目录 X11 下, 比如/pub/Linux/X11。

只能在特定的窗口管理器下运行 X Window 应用程序的情况是很少见的。用户可以在 fvwm、Xview、fvwm95 或其他任何窗口管理器下运行同一个 X Window 应用程序, 例如 Netscape。甚至在没有任何窗口管理器的情况下, 也可以运行 X Window 应用程序。用户也可以打开一个非常简单的 xterm 窗口, 然后从 xterm 窗口启动 X Window 应用程序。X Window 应用程序运行所需要的只是 X Window, 而不是窗口管理器。

3.6.10 fvwm 配置文件

fvwm 窗口管理器启动时将执行自己的配置文件。这些文件将完成一些任务, 诸如在 fvwm 任务栏上显示按钮, 在工作区菜单里建立菜单项, 决定启动什么初始化程序(如果有)。这些配置文件在目录/usr/X11R6/lib/X11/fvwm 下。每个文件的名字以单词 system 开头。

基本配置文件是 system.fvwmrc。这个文件是其他文件的开头, 如 system.fvwmrc.desk 和 system.fvwmrc.goodstuff 等。

用户可以修改这些配置文件, 按照自己的喜好配置 fvwm 窗口管理器。例如, 如果想往任务栏添加按钮, 可以往文件 system.fvwmrc.goodstuff 里添加记录。

对文件 system.fvwmrc 的修改将改变系统上的所有用户的配置。当然, 每个用户可以在自己的主目录下建立个人的配置文件.fvwmrc, 它只对创建它的个人用户起作用。每次用户执行 startx 命令时, 这个文件就被执行, 并被用来配置 fvwm。

3.6.10.1 任务栏: system.fvwmrc.goodstuff

要往任务栏添加按钮, 可以在文件 system.fvwmrc.goodstuff 中建立新的记录。记录以关键词*GoodStuff 开头, 接着是用户想在按钮的图像下显示的文本, 然后是用户想显示的图像文件名。图像是 pixmap 型, 文件扩展名是.xpm。用户能够在目录/usr/openwin/lib/pixmaps 和/usr/share/data/pixmaps 下找到图像文件。图像文件名后面是一个 fvwm 操作。尽管这个操作通常是 Exec 命令, 但它实际上可以是任何 fvwm 命令。Exec 将执行一个 Linux 命令或程序。Exec 命令后面是一个可以作为菜单项的名字的字符串。对按钮条来说, 它只是一个空串“ ”。字符串后面是 Xlaunch 命令和程序名字。下例往 Crisplite 编辑器中添加一个按钮。启动 Crisplite 编辑器的命令是 mcr。

```
*GoodStuff    Crisplite    dtp.xpm    Exec    ""    xlaunch    mcr
```

如果想从任务栏移去一个记录, 可以删除文件中对应于此记录的那一行, 也可以在那一行开头插入“#”符号作为注释。支持任务栏的其他窗口管理器也有类似的配置文件。

3.6.10.2 工作区菜单: system.fvwmrc.menu

文件 system.fvwmrc.menu 保存着工作区菜单的配置信息。工作区主菜单在该文件末尾定义, 子菜单则在主菜单前定义。工作区菜单和其他主要子菜单都用命令 PopupSMenu 定

义。每一项占一行，每一项通常对应一个菜单定义，**Popup** 项指定了这种定义。**Popup** 项后面是将显示在菜单上的记录名称，然后是相关菜单的名字。菜单名称中前面有“&”的字母在菜单显示时带有下划线，并作为快捷键的关键字母。按下那个字母将选择该字母对应的菜单项。部分菜单定义如下所示：

```
PopSMenu('Workplace',
  Title      "Workplace"
  Nop        " "
  Popup      "&Help on Linux"   Help
  Nop        " "
  Popup      "&Shells"           Shells
)
```

很多子菜单用 **PopupMenu** 命令定义。紧跟着 **PopupMenu** 的是一串记录名称和命令，它们之间以逗号隔开，所有这些字符串被一对括号括起来。第一项通常是子菜单名称；然后是子菜单名称，它后面是该子菜单执行的命令；紧接着又是一对子菜单名称和命令。可以用符号“&”来指定快捷键关键字母。子菜单 **Text and Publishing** 就是这样的弹出式(**popup**)菜单，如下是部分菜单定义：

```
PopupMenu('Text and Publishing','MenuItem',
  'L&yX(WYSIWIN TeX)', 'lyx',
  'X&wpick(Screen shot)', 'xwpick HOME/xwpick.gif',
  'X&fig(Vector graphic)', 'xfig'
)
```

如要往菜单里添加一个菜单项，可以先找到该菜单的定义部分，然后插入一空行，输入想添加的菜单名称和命令。例如，要在文本和菜单中插入对应于 **Crisplite** 编辑器的菜单项，用户将插入以下这行。字母 **c** 前面的符号“&”表示建立了关键字母为 **c** 的快捷键，别忘了加上逗号。

```
"Crisplite",      m&c,r,
```

千万不要修改文件 **system.fvwmrc.menu.prep**。这个文件是由 **fvwm** 生成的，它实际上创建了工作区菜单。这个文件是由文件 **menu** 提供的信息自动生成的。

安装在目录 **/usr/X11R6/bin** 下的 **X Window** 程序会自动加载到 **Application** 菜单下的 **Other Applications** 子菜单上。因此，用户能够通过工作区菜单执行任何 **X Window** 程序。往菜单文件中添加一条记录将把该记录对应的菜单项放在一个特定子菜单中，而不是放在 **Other Applications** 菜单中。

3.6.10.3 图标风格与设备映射：.styles和.bindings

文件 **system.fvwmrc.styles** 存放着用于指定特定程序的窗口风格的记录。例如，下面显示的指定页面调度程序风格的 **StaysOnTop** 使页面调度程序始终位于所有窗口前面。**Style** 项也被用来指定程序窗口图标化时所使用的图像文件。用户能够改变这些项目，以便显示不同的 **pixmap** 图像，或者为新的程序添加新记录。

```
Style    "fvwmPager"      StaysOnTop
Style    "xgraph"         Icon    graphs.xpm
```

文件 `system.fvwmrc.bindings` 存放着所有鼠标按键和键盘的映射。鼠标按键和键盘对工作区菜单的操作都在这儿定义。按下组合键 `Alt+F1` 将显示工作区菜单，单击鼠标左键(键 1)可以得到相同的结果。下面这一项就是定义了组合键 `Alt+F1` 与工作区菜单之间的映射。Context 域是指组合键的适用窗口，字母 A 代表任何窗口；字母 R 代表根窗口，Modifi 域指任何需要同时按下的控制键。字母 M 代表元键(Alt 键)；字母 S 代表 Shift 键。

```
#Key      Context      Modifi      Function
Key F1     A                M           Popup      "Workplace"
```

3.6.10.4 初始启动应用程序: `system.fvwm.desk`

在 Linux 系统里命令 `startx` 首先启动 X Window，然后是 `fvwm` 窗口管理器，接着是 `Caldera Desktop Looking Glass`。`Caldera Desktop` 的自动启动是由 `fvwm` 的配置脚本文件完成的。这些配置文件可以用来指定在 `fvwm` 启动时被自动启动的程序。

通常，`startx` 命令通过配置文件 `/etc/X11/init/xinitrc` 调用 `xinit` 命令。这个 `xinitrc` 文件执行基本的 X Window 配置操作，最后启动 `fvwm` 窗口管理器。`fvwm` 窗口管理器有它自己的启动文件，该文件位于目录 `/usr/X11R6/lib/X11/fvwm` 下。`system.wmrc` 文件是基本配置文件，它执行 `StartupFunction` 命令，文件 `system.fvwmdesk` 定义了这个命令。在这个文件中，`StartupFunction` 被定义为执行命令 `lg`，该命令启动 `Caldera Desktop`(`lg` 代表 `Caldera Desktop` 的真正名字——`Looking Glass`)。如果用户不想让 `Caldera Desktop` 随着 `startx` 命令自动启动，只需在有 `lg` 命令的那一行开头插入符号“#”，使这一行成为注释行。如果用户想让 `fvwm` 自动启动其他的文件管理器或者桌面，可以把命令插入到文件 `system.fvwmdesk` 中。用户能够很轻松地把文件 `system.fvwmrc` 里的整个 `StartupFunction` 注释掉，如下所示：

```
#      Function      "I"      StartupFunction
```

如果从 `fvwm` 取消 `Caldera Desktop` 的自动启动功能，用户可以使用工作区菜单来启动 `Caldera Desktop`。在 `Desktop` 子菜单中，用户将能找到对应于 `Looking Glass`(`Caldera Desktop`) 的菜单项。选择这一菜单项将启动 `Caldera Desktop`。用户当然也可以用这种方法启动别的文件管理器或桌面。

3.6.11 X Window 命令行参数

用户能够以 `xinitrc` 脚本文件的方式，或者以 `xterm` 窗口里的命令行方式启动任何 X Window 应用程序。大多数 X Window 应用程序支持一组标准的 X Window 参数，这些参数用来配置应用程序所使用的窗口和显示。用户可以设定标题栏的颜色、确定窗口的标题、指定文本颜色和字体，也可以把窗口定位于屏幕的特定位置。它们在 X 的 `man` 帮助页中有详细说明(运行 `man X`)。下面就介绍 X Windows 命令常用的参数：

`-geometry` 是一个经常使用的参数。它还需要另外的参数，通过它用户能够指定应用

程序的窗口在屏幕上显示的位置。在下面的例子里, X Window 应用程序 `xclock` 在带有 `geometry` 参数的情况下被启动。那些数字(最多可以有 4 个)用来确定窗口位置。`+0+0` 表示左上角; `-0-0` 表示右上角。用户启动 X Window 窗口时, 将能看到显示的时钟。

```
& xclock -geometry +0+0 &
```

用户可以通过 `-title` 参数设定应用程序窗口的标题。注意当标题不只一个单词时, 要用引号把它们括起来。`-fn` 参数用来设定字体; `-fg` 设定文本和图表颜色; `-bg` 设定背景颜色。下例打开了一个标题为“`My New Window`”的窗口。它的文本和图表颜色是绿色, 背景颜色是灰色, 字体是 `Helvetica`。

```
$ xterm -title "My New Window" -fg green -bg gray -fn  
/usr/fonts/helvetica &
```

3.6.12 X Window 配置文件

X Window 使用配置文件和 X Window 命令来配置 X Window 系统。其中一些配置文件属于系统, 不能修改。但是, 每个用户可以有一组自己的配置文件, 诸如 `.xinitrc` 和 `Xresources` 等。

使用这些文件可以建立个人风格的 X Window 界面。当用 `startx` 命令启动 X Window 时, 这些配置文件被自动读取并执行。在这些配置文件里, 用户能够执行 X Window 命令来配置系统。通过诸如 `xset` 和 `setroot` 之类的命令, 用户能够增加字体或者控制根窗口的显示。

用户可以通过使用 `xdpyinfo` 命令得到当前使用的 X Window 系统配置的完整描述。`man` 帮助页有关于 X Window 命令和配置文件的详细介绍。

```
$ man X
```

X Window 是由命令 `xinit` 启动的。但是用户无需直接执行这个命令, 可以通过经常用来启动 X Window 的 `startx` 命令来执行它。这两个命令都在目录 `/usr/X11R6/bin` 下, 和许多基于 X Window 的其他程序存放在一起。

`startx` 命令是一个执行 `xinit` 命令的 Shell 脚本文件。`xinit` 命令按顺序首先查找位于用户主目录的 X Window 初始化脚本文件 `.xinitrc`。如果在主目录里没有脚本文件 `.xinitrc`, `xinit` 将把 `/usr/X11R6/lib/X11/xinit/xinitrc` 作为其初始化脚本文件。`.xinitrc` 和 `/usr/X11R6/lib/X11/xinit/xinitrc` 中都有配置 X Window 服务器的命令, 它们执行诸如启动窗口管理器之类的 X Window 初始化命令。可以把 `/usr/X11R6/lib/X11/xinit/xinitrc` 脚本文件看作默认脚本文件。在 Linux 系统里, `/usr/X11R6/lib/X11/xinit` 目录实际上是目录 `/etc/X11/xinit` 的一个符号链接, 默认初始化文件都存放在 `/etc/X11/xinit` 目录中。

Linux 系统并不自动地在任何用户主目录下创建 `.xinitrc` 脚本文件, 需要 `.xinitrc` 脚本文件的用户将不得不亲自创建 `.xinitrc` 脚本文件。每一个用户(包括根用户)可以在自己的主目录下创建个人风格的 `.xinitrc` 脚本文件, 以便按照自己的意愿来配置和启动 X Window。在某个用户创建自己的 `.xinitrc` 脚本文件前, 默认使用 `/usr/X11R6/lib/X11/xinit/xinitrc` 脚本

文件。用户才可以通过查看这个脚本文件来了解 X Window 是如何启动的。X Window 的某些配置操作必须在文件 `.xinitrc` 中完成。对于想要创建自己的 `.xinitrc` 脚本文件的用户，最好首先把脚本文件 `/usr/X11R6/lib/X11/xinit/xinitrc` 拷贝到自己的用户主目录下，命名为 `.xinitrc`。然后，用户才可以随心所欲地修改这个脚本文件(注意：系统的 `.xinitrc` 脚本文件在名字前面不带“.”号，而用户自己建立的主目录下的 `.xinitrc` 文件名字前面带“.”号)。用户如需要更多的信息，可参考 `xinit` 和 `startx` 的 man 帮助页。

除了 `xinit` 命令之外，其他命令，如 `xrdb` 和 `xmodmap` 等，也能够用来配置 X Window 的界面。在资源文件 `.Xresources` 中列出了 X Window 的一些图形配置。每个用户可以在自己的用户主目录下拥有定制的 `.Xresources` 文件，用它来配置特定的 X Window。

`.Xresources` 文件包括一些配置特定程序的记录，比如某个工具箱的颜色。`.Xresources` 文件也有一个适用于整个系统的版本：`/usr/X11R6/lib/X11/xinit/.Xresources`(注意：与 `/usr/X11R6/lib/X11/xinit/xinitrc` 不同，在 `/usr/X11R6/lib/X11/xinit/.Xresources` 文件名中，`Xresources` 前面带有“.”号)。与文件 `.Xresources` 一样，文件 `.Xdefaults` 也包括一些用来配置 X Window 资源的记录。

Linux 系统安装后只有一个系统 `.Xresources` 文件，但用户可以在自己的主目录下创建自己的 `.Xresources` 文件，并且往此文件中添加一些资源记录。用户也可以拷贝文件 `/usr/X11R6/lib/X11/xinit/.Xresources` 到自己的主目录下，然后编辑或添加自己的记录。配置操作是由命令 `xrdb` 执行的，它会读取系统的 `.Xresources` 文件和用户主目录下的任何 `.Xresources` 文件和 `.Xdefaults` 文件。`xrdb` 通常在脚本文件 `/usr/X11R6/lib/X11/xinit/xinitrc` 中被执行。如果用户在主目录下创建了自己的 `.xinitrc` 脚本文件，务必在文件最后运行 `xrdb` 命令，使它读取 `.Xresources` 文件或 `/usr/X11R6/lib/X11/xinit/xinitrc` 文件(最好两者都读)。要做到这一点，用户只需用系统的 `.xinitrc` 脚本文件的一个拷贝作为自己的 `.xinitrc` 脚本文件，然后按照自己的喜好修改它。关于 X Window 资源的细节，请看 `xrdb` 的 man 帮助页。当然也可以在 X 的 man 帮助页中找到关于 `.Xresources` 以及其他 X Window 配置文件的详细说明。

`.Xresources` 文件的每一条记录由一个分配给资源、类或一个应用程序资源的值组成。通常它们是工具箱的资源或一个应用程序中工具箱的类的资源。这些资源一般由三个元素的组合(元素之间用“.”号分隔)来指定：应用程序、对象和资源。

例如，假设想在 `oclock` 应用程序中把时针的颜色改为蓝色。应用程序是 `oclock`，对象是 `clock`，资源是 `hour: blue`。这条记录如下所示：

```
oclock.clock.hour:blue
```

对象元素实际上是由一系列等级的对象组成，由此确定一个特定对象。在 `oclock` 这个例子中只有一个对象，但在很多应用程序中，对象分级是非常复杂的。用户将需要列出一串冗长的对象才能确定用户想指定的那个对象。为了避免这种复杂性，用户可以使用“*”号来直接引用想要指定的对象，具体做法是用“*”号来替代“.”号。下面的例子是把 `oclock` 的分针和时针设置为绿色。

```
oclock*hour :green
oclock*minute: green
```

用户也可以使用“*”号把一个值赋于对象的所有类。用户可以使用类的名字来引用这个类中的所有资源。类名以大写字母开头。例如，在 `xterm` 应用程序中，背景和鼠标指针颜色都是 `Background` 类的一部分。引用 `xterm*Background` 将改变 `xterm` 窗口中所有这些资源。

用户也可以使用“*”号来改变所有应用程序的对象中的资源值。在这种情况下，可以在资源名前加一个“*”号。例如，要把每一个应用程序中的所有对象的前景颜色改为红色，可以输入：

```
*foreground: red
```

如果只想改变所有应用程序中滚动条的前景颜色，可以输入：

```
*scrollbar*foreground:blue
```

命令 `showrgb` 将列出系统提供的各种颜色。用户可以改变某个对象中显示的字体、图形应用程序中的背景或边框图像。各个应用程序的资源各不相同。应用程序可能支持各种各样的对象和它们的资源。如果想了解一个应用程序所能支持的资源以及这些资源所能接受的值，可以查看此应用程序的帮助页和文档数据。某些资源的取值是逻辑量，用它们可以关闭或启动一些特性。其他一些资源则能够指定某些选项。

文件 `/usr/X11R6/X11/xinit/.Xmodmap` 存放着诸如鼠标和键盘之类的系统输入设备的配置信息(例如，用户可以建立键的映射，如 `BACKSPACE` 也可以改变鼠标左右键的定义)。每一个用户可以在自己的主目录下创建自己的定制 `Xmodmap` 文件，用它来配置系统的输入设备。如果用户是通过自己的终端和 Linux 系统相连接，这种定制的文件会对用户很有帮助。`.Xmodmap` 文件由 `xmodmap` 命令读取，通过它完成配置。`xmodmap` 命令将首先在用户主目录下寻找 `Xmodmap` 文件，然后读这个文件。如果在主目录下没有 `.Xmodmap` 文件，`xmodmap` 命令将读文件 `/usr/X11R6/lib/X11/xinit/.Xmodmap`。`xmodmap` 命令通常在文件 `/usr/X11R6/lib/X11/xinit/xinitrc` 内被执行。详细情况请参考 `xmodmap` 的帮助页。

通常，在 `xinitrc` 脚本文件里有一些 X Window 命令，比如 `xset` 和 `xsetroot`，用它们来配置 X Window 的各种不同特性。`xset` 命令设置一些不同的选项，例如启动屏幕保护和设定铃与扬声器的音量等。也可以用 `xset` 来加载字体。详细情况请参考 `xset` 的帮助页。`xset` 在选项 `b` 和参数 `on` 或 `off` 存在的情况下，将打开或关闭扬声器。下面的例子中，打开了扬声器：

```
xset      b      on
```

如果要设置屏幕保护，可以执行带选项 `-s` 的 `xset`。用户可以通过参数 `on` 或 `off` 来启动或关闭屏幕保护。作为参数输入的两个数值指定长度和间隔(单位：秒)。长度表示屏幕保护程序在激活前等待的秒数。间隔是指屏幕保护程序在产生新的图案前等待的时间。

`xsetroot` 命令使用户能够设置根窗口的特性(设置颜色或显示位图图案，用户甚至能够自己设置光标)。要了解 `xsetroot` 命令的各选项和详细情况请参考 `xsetroot` 的帮助页。下面的 `xsetroot` 命令通过使用 `-solid` 选项把根窗口的背景设置为蓝色。

```
xsetroot -solid blue
```

文件/usr/X11R6/lib/X11/xinit/xinitrc 中还包含一些命令和变量，它们用来配置 Looking Glass 桌面、Xview 窗口管理器和 fvwm 窗口管理器。

在.xinitrc 脚本文件里，窗口管理器的启动命令往往是这个脚本文件中的最后一条命令。无论.xinitrc 脚本文件的最后一条命令是什么，在执行完这条命令后 X Window 将会退出。通过把启动窗口管理器的命令放在最后，在退出窗口管理器时将关闭 X Window。用户想初始启动的其他任何程序都应放在窗口管理器命令前面。如果想在执行 startx 时都自动启动 Netscape 或 xfm 文件管理器，用户必须把启动这些应用程序的命令放在窗口管理器命令前面。在应用程序后面应加一个“&”号，使它们能在后台运行。下例在 fvwm 启动时，自动启动 xfm 文件管理器和 Netscape。

```
xfm &
netscape &
exec fvwm
```

下面是一个简单的.xinitrc 文件的例子。它被设计成在启动 fvwm 窗口管理器的同时启动 xfm 文件管理器。变量 OPENWINHOME 和 HELPPATH 为需要 Xview 库的应用程序指出了这些库的存放目录。如果.xinitrc 文件存放在用户主目录中，它会指示 xinit 在启动 fvwm 的同时启动 xfm(注意文件.xinitrc 名字前的“.”)。当 fvwm 启动时，任务栏与页面调度程序将和 xfm 窗口一起显示。在屏幕的任何位置单击鼠标左键都会弹出工作区菜单。

```
.xinitrc

#!/bin/sh
#$Xconsortium: xinitrc.cpp,v 1.4 91/08/22 11:41:34 rws Exp $

userresources=$HOME/.Xresources
usermodmap=$HOME/.Xmodmap
sysresources=/usr/X11R6/lib/X11/xinit/.Xresources
sysmodmap=/usr/X11R6/lib/X11/xinit/.Xmodmap

export OPENWINHOME=/usr/openwin
export HELPPATH=$ OPENWINHOME/help

#merge in defaults and keymaps
if [ -f $sysresources ] ; then
    xrdp -merge $sysresources
fi

if [-f $sysmodmap]; then
    xmodmap $sysmodmap
fi

if [ -f $userresources ]; then
    xrdp -merge $userresources
```

```
fi

if [-f $usermodmap]; then
    xmodmap $usermodmap
fi

#start some nice programs
xfm &

exec fvwm
```

在目录/etc/X11/xinit 下有一个名为 `xinitrc.before.fvwm` 的仅有简单框架的 `xinitrc` 文件样本。它没有对应于 `fvwm`、`Xview` 和 `Looking Glass(Caldera)`桌面的配置命令。但是，这是一个仅包含运行 `X Window` 所必须的基本配置操作的好例子。该脚本文件将打开三个 `xterm` 窗口并且运行 `twm` 窗口管理器。如果需要，用户可以把这个脚本文件拷贝到自己的主目录下，作为自己的 `.xinitrc` 文件(如果用户现在有自己的 `.xinitrc` 文件,务必对它进行备份)，然后就可以按照自己的意愿添加配置操作和命令。例如，可以去掉一些 `xterm` 命令，启动另一个不同的窗口管理器(添加需要的配置变量)，或者启动一个应用程序，如 `Netscape`。如果不想启动 `xclock`，用户可以在它的命令前加一个“#”号，把这个命令注释掉。

3.6.13 字 体

`X Window` 的字体文件存放在目录 `/usr/X11R6/lib/X11/fonts` 下。

`X Window` 的字体文件由命令 `xfst` 程序来装载。`xfst` 读取配置文件 `/etc/X11/fs/config`，然后根据这个文件的内容来处理字体文件。在 `xman` 帮助页里有关于字体的详细信息。为了安装字体集，把相应的字体文件放在一个用户能够把它的路径添加到 `catalogue` 项的目录下面，这样字体就可以自动安装。用户也可以用带 `+fp` 选项的 `xset` 命令单独地安装某种特定字体。用户系统里的字体由字体路径指定。字体路径是一些文件名，每一个文件存放一种字体。文件名包括它们的完整路径。下面是文件 `/etc/X11/config` 中 `catalogue` 项的一个例子。这是一串以逗号隔开的目录列表。`X Window` 首先在这些目录中寻找字体文件。

```
Catalogue+
/usr/X11R6/lib/X11/fonts/misc,/usr/X11R6/lib/X11/fonts/Speedo/,
/usr/X11R6/lib/X11/fonts/Type1/,/usr/X11R6/lib/X11/fonts/75dpi/,
/usr/X11R6/lib/X11/fonts/100dpi/
```

在使用新安装的字体之前，用户必须首先使用 `mkfontdir` 命令建立字体索引。从包含新的字体的目录中，输入 `mkfontdir` 命令。用户也可以把目录路径作为 `mkfontdir` 命令的一个参数。在对字体索引完以后，用户可以使用带有选项 `fp rehash` 的 `xset` 命令装入这些字体。为了能够自动装载这些字体，把目录以完整路径名的形式添加到 `xfst` 配置文件的

catalogue 项中去。下面的例子显示了如何安装一种新的字体并加载它：

```
$ cp newfont.pcf ~/myfonts
$ mkfontdir ~/myfonts
$ xset fp rehash
```

在一个字体目录下，有几个保存着关于字体的信息的文件。fonts.dir 文件列出了该目录下的所有字体。此外，用户可以通过建立一个 fonts.alias 文件给一个字体文件取其他的名字。字体文件名一般很长、很复杂。详细情况请参考 xfs 和 mkfontdir 的帮助页。

用带有 +fp 和 -fp 选项的 xset，用户能够分别地添加或者去掉特定的字体。带有参数 rehash 的 fp 选项将装载字体。如果使用命令“xset fp default”，将恢复默认字体集。+fp 向字体路径添加一种字体。对于自己定义的字体，用户可以把它们放在任何目录下并且指定它们的名字(包括它们的完整路径名)。以下的例子把在目录/usr/local/fonts 下的 myfont 字体添加到字体路径中。然后用带有参数 rehash 的 fp 选项装入字体。

```
xset +fp /usr/local/fonts/myfont
xset fp rehash
```

要去掉这种字体，需运行 xset -fp /usr/home/myfont 命令，然后运行命令 xset fp rehash。如果想把系统恢复到默认字体集，输入如下命令：

```
xset fp default
xset fp rehash
```

使用 xlsfonts 命令，可以列出目前安装在系统上的所有字体。如果想显示已安装的字体以查看它的外观，使用 xselfonts 命令。

3.6.14 编译 X Window 应用程序

为了编译 X Window 应用程序，用户必须确信安装了 XFree86 开发包和其他可能要用到的开发包。这些开发包包含了头文件和 X Window 程序使用的库。这类包的文件名一般含有 devel，例如，XFree86-devel-3.1.2-2.i386.rpm。

许多 X Window 应用程序可能需要特殊的共享库。例如，有些应用程序可能需要 xforms 库或 qt 库。这些库中的某些部分可能不得不从联机站点得到，但大多数都可以从 RPM 形式的 Redhat contrib 目录得到。

许多 X Window 应用程序需要生成一个为配置系统而创建的 Makefile 文件。如果有应用程序源代码，这个任务可以由 Imakefile 文件完成。Linux 系统的有些发行版本含有 xmkmf 命令，这个命令能够根据 Imakefile 文件的内容自动生成相应的 Makefile 文件。

有了 Makefile 文件后，就可以用 make 命令编译应用程序。xmkmf 命令调用程序 imake 从 Imakefile 生成 Makefile 文件。但是用户不能直接使用 imake。关于 xmkmf 和 imake 的详细情况可查看相应的 man 帮助页。许多程序使用配置脚本文件生成 Makefile。在那种情况下，用户只需运行 ./configure 命令，而不是 xmkmf。

3.7 XFree86 配置

3.7.1 X11R5 和 XFree86 简介

X11R5 是针对 UNIX 系统的窗口系统。X Window 由 MIT Consortium 提出, 有自由的版权, 它允许任何源程序被使用, 使用时只要将原来的版权注意事项包含进去即可。

因为 X Window 是 UNIX 操作系统的标准窗口系统, 所以也就有相当多的应用程序使用 X Window, X Window 有免费版本与商业版本。

XFree86 也是 X11R5 的一个分支, 主要支持 Intel 平台的 UNIX 操作系统和类似 UNIX 的操作系统。XFree86 服务器部分源自 x386 1.2 版(它是随 x11R5 一起公诸于世的 X Window 服务器), 但是 XFree86 大部分是新开发的。从 XFree86-1.3 之后, 最重要的改进是图形加速卡, 这是一个全新的特性。

3.7.2 支持的硬件

这是一个常见的问题, 在此先做简单的说明。目前 2.0 版本的 XFree86 支持如下的显示卡硬件。

非加速卡:

```
Tseng ET3000, ET4000AX, ET4000/W32
Western Digital/Paradise PVGA1
Western Digital WD90C00, WD90C10, WD90C11, WD90C30
Genoa GVGA
Trident TVGA8800CS, TVGA8900B, TVGA8900C, TVGA8900CL, TVGA9000
ATI 28800-4, 28800-5, 28800-a
NCR 77C22, 77C22E
Cirrus Logic CLGD5420, CLGD5422, CLGD5424, CLGD6205, CLGD6215,
        CLGD6225, CLGD6235
Compaq AVGA (cf the question in the appendix)
OAK OTI067, OTI077
```

加速卡:

```
Cirrus CLGD5426, CLGD5428
Western Digital WD90C31
NOTE WD90C33 在 XFree86-2.0 无法运行
```

这些卡都支持 256 色(XF86_SVGA), 而 ATI 和 Cirrus 的卡不支持单色模式(这两种卡只支持 256 色)。

有关各种卡的说明可在/usr/X386/lib/X11/etc 目录下的 README 文档中找到。

3.7.3 XFree86 的获取

本节主要针对 Linux 的可执行的软件而言。可执行的 XFree86-2.0 for Linux 软件可通过 FTP 协议到下面的站点取得：

tsx-11.mit.edu: /pub/Linux/packages/X11/XFree86-2.0

或者

sunsite.unc.edu: /pub/Linux/X11/XFree86-2.0

这些软件都是以 gzip 压缩的归档文件：

XF86_8514.tar.gz	IBM8514 卡的服务器
XF86_Mach32.tar.gz	Mach32 卡的服务器
XF86_Mach8.tar.gz	Mach8 卡的服务器
XF86_Mono.tar.gz	单色卡的服务器
XF86_S3.tar.gz	S3 卡的服务器
XF86_SVGA.tar.gz	SVGA 卡的服务器
XF86_VGA16.tar.gz	16VGA 卡的服务器
xf86-svr-2.0.tar.gz	以上所有卡的服务器
xf86-bin-2.0.tar.gz	一些常用的用户程序
xf86-cfg-2.0.tar.gz	XDM 配置文件和选择器
xf86-fnt-2.0.tar.gz	字体
xf86-kit-2.0.tar.gz	Linkkit 用以建置 X Window 服务器
xf86-lib-2.0.tar.gz	动态库、位图和字体
xf86-man-2.0.tar.gz	联机手册
xf86-pex-2.0.tar.gz	PEX 库
xf86-prg-2.0.tar.gz	静态库和头文件
xf86-doc-2.0.tar.gz	XFree86-2.0 的文档
xf86-doc2-2.0.tar.gz	Xconfig 的联机手册

另外需要 libc4.4.1 或更新的版本才能运行 XFree86-2.0。而且必须安装 David Engel 的动态链接库 ld.so 1.3 或更新版本，这可以从 tsx-11 的镜像站点获取，目录名是 /pub/Linux/packages/GCC。

内存的需求是至少 8MB 物理内存和 16MB 虚拟内存(交换区方式)。若能小心管理和使用内存，4MB 也是可以运行的。建议是至少有 8MB 的内存，因为用交换区方式会降低系统的性能。

若要运行很消耗内存的程序，像 gcc，那么用户至少应该有 16MB 物理内存和 16MB 虚拟内存。

不必安装 LinkKit 软件包，除非有 17MB 的硬盘空间，如果完全安装该软件包，则需 21MB 空间。

如果去掉其他不需要的服务器，也可以节省几 MB 的空间。

安装之前，请备份已变更过的文件，以备不时之需。

安装的步骤如下(请以 root 身份在/目录下执行)：

```
gzip -dc tarfilename | tar xvzof -
```



警告： 这个命令的执行结果将会覆盖用户的旧的 XFree86 版本。

安装之后，用户必须正确设置用户的系统。

如何编译在该软件包中的 INSTALL、README 和 README.Linux 的几个文件中都有说明。

如果要添加 SVGA 服务器的驱动程序，那么用户需要 LinkKit 软件包。LinkKit 用户可在目录/usr/X386/lib/Server/VGADriverDoc 下找到。

3.7.4 配置 XFree86

从 2.0 版开始，所有的 XFree86 文件都被存放在一起，并用 tar 将其结合成一个文件——xf86-doc-2.0.tar.gz。

用户必须取得这个文件，并详读其中的文件之后，才会正确地建立和运行 XFree86。下面就是有关这些文件的一些说明：

README.Config 文件是有关搭建 XFree86 的简介，先看此文件，并依照其指示一步一步做。

接下来用 man 命令查看 Xconfig、XFree86 和有关的服务器(XF86_SVGA、XF86_Mono、XF86_VGA16、XF86_S3、XF86_Mach8、XF86_Mach32 或 XF86_8514 等)的使用说明。

如果无法确定使用的显示卡是哪一种，可以用 SuperProbe 程序来检查显示卡种类。不过若有显示卡的手册，那么依据手册的说明。

要注意的是 SuperProbe 不一定能正确地检查出显示卡类型，在这种情况下 XFree86 的服务器很难正确运行。

在 AccelCards 文件中列出了一些可以在 XFree86-2.0 版下正常运行的视频加速卡。看看自己的卡是否列在其中。可能有一些卡可能没有列在其中，但也能正常运行。若使用 XS3，那么该看看 README.XS3，因为要加入 XS3 服务器时，要对系统稍做一些改变。

配置 XFree86 的一些问题可以用改变 BIOS 设置和缓存区的方法来解决。

在安装时用户将需要如下的参数：

1. 屏幕规格(如水平和垂直的频率、频宽等等)。
2. 显示卡的名称。有些公司会用“S3”代表“accelerated”。
3. 用户的显示卡所提供的显示内存。



注意： WD90C3x 显示卡在检测时钟时会有问题。请先用 XFree86-1.3 服务器测得时钟，再将其记录在 Xconfig 文件中，然后就可以运行 2.0 版的服务器了。

4. 鼠标使用的协议和连接的设备。在 Xconfig 的联机手册中列出了可用的鼠标协议。以下是一些鼠标连接的设备名称：

/dev/atibm ATI XL 总线型鼠标(注意：ATI GU 总线型鼠标事实上就是 Logitech busmouse)

/dev/logibm Logitech 总线型鼠标(注意：它使用总线型鼠标协议，而不是 Logitech 协议)

/dev/inportbm microsoft 总线型鼠标

/dev/psaux ps/2 鼠标



注意： 上列的名称是新的名称，也许在旧的版本中都还是使用旧的名称。下面给出新旧名称对照列表：

新	旧	主设备号	次设备号
atibm:	bmouseatixl	10	3
logibm:	bmouselogitec	10	0
inportbm:	bmousems	10	2
psaux:	bmouseps2 或 ps2aux	10	1

其他的鼠标就属于串行鼠标，所以也都连上一些串行端口，如/dev/ttyS?或/dev/ttyS??。

总线型鼠标(bus mouse)有时不易看出连接上哪一个设备，要判别就得将所有的驱动程序放到系统内核中，并在启动时看看启动时的信息，这些信息会说明系统所检测到的总线型鼠标是哪一种。此时用户就知道了它连上哪个设备。

接下来查看 README.Config 这个文件。若用户的屏幕型号在 modeDB.txt 文件中没有被列出，应该选一个一般的模式。

若想调整或换一个模式，则请先详读 VideoModes.doc 文件，并依其指示去设置。



警告： 别人的 Xconfig 文件不一定适合自己用，因为各人的硬件设备有所不同。特别是屏幕模式的设置，要确定是自己的屏幕的规格所有的，以免损坏硬件。

下面介绍如何配置非US-layout模式的键盘。

若用户不改变原先标准的设置，服务器会自动启动 US-american 键盘模式。用户若要不同的设置，请看 xmodmap 的帮助说明。

在 sunsite.unc.edu: /pub/Linux/X11/misc 站点下中有 xmodmaps 关于语言键盘的几个设置实例。

另外有几个特别键，用户可以定义新的对应。这方面请参考 XFree86kbd(1) 联机手册。

3.7.5 tinyX

如果用户的内存不足以运行 XFree86，可以试试 tinyX 软件。

tinyX 以 XFree86 为基准，所以安装或设置的方式与 XFree86 相同。

要运行 tinyX 至少要有 4MB 物理内存和 8MB 交换区空间。目前的 tinyX 有几个版本, 服务器程序可依前面的 XFree86-2.0 的说明去选择。软件的名称是: tinyX-YYY-2.0.tar.gz(YYY 代表服务器名称)。

此软件得可在 Sunsite 的 FTP 站点取得。取得之后先解压缩, 然后设置 Xconfig 文件, 在 readme 目录下也可以找到 tinyX 的一些相关文件, 先看完这文件, 再设置 Xconfig 文件。

由于 tinyX 不包含完全版的 XFree86-2.0, 所以可能会在执行时出现程序不存在的信息。

3.7.6 X 相关软件

目前有许多 X 相关软件。在此介绍两个在 X 方面有比较大的影响的软件。

3.7.6.1 xview3L5.1.tar.gz

该软件源自 Sun(TM)的 xview3.2 软件, 是由 Kenneth Osterberg 修订的 Linux 版本。包含 xview toolkit 及一些 X Window 的扩充等等。

用户会觉得它是 Open Look(TM)在 X Window 方面的扩充。

要安装它, 首先要解开归档文件, 读取一些随附的说明文件(如 README 或 install 等等)。然后, 安装这个脚本。要完全安装此软件要有 25MB 的硬盘空间。

要执行 Xview 至少要有 8MB 的内存, 最好有 16MB。另外, 要在 XFree86-2.0 上运行 xview, 则共享库 ld.so 必须是 1.3 或以上的版本。

3.7.6.2 Motif for Linux

这是一个商业版的软件, 所以只能用钱去买。

从 tsx-11 站点下的目录/pub/Linux/advertisements 下可找到 Metrolink 公司的产品广告。

这个软件提供了不同的窗口管理器。其他的一些的窗口管理器程序可从 Sunsite 站点的目录/pub/Linux/X11/Window-managers 下找到。

这些软件都有不同的设置方法。所以要设置这些窗口管理器就必须详读它们各自的安装文件, 以便能正确安装。

可以从下面的站点获取一些现成的 X Window 应用软件:

`sunsite.unc.edu/pub/Linux/X11`

或

`tsx-11.mit.edu/pub/Linux/binaries` 与 `/pub/Linux/sources`

3.7.7 X Window 常见问题解答

Q: 用“X”命令来启动 X, 只看到网状的背景, 却没有看到窗口, 是不是在哪出错了?

A: 用户只是启动服务器, 没有启动窗口管理器和其他的应用程序。一般的启动方式

是用 `startx` 命令。

Q: 有支持 16 位或 24 位的彩色模式吗?

A: 目前只支持 1 位, 4 位和 8 位的彩色模式。而且 4 位的服务器还处在实验阶段, 可能会有错误。

Q: 服务器无法找到某些字体。

A: 检查用户的目录 `/usr/X386/lib/X11/Xconfig` 中的字体路径, 是否包含这些路径, 若有则执行 `mkfontdir` 命令可能会给用户一些帮助。

Q: 当结束 X 时, 字体有时无法恢复正常, 屏幕会出现奇怪的内容。

A: 此问题尚未完全解决, 不过有些软件有时会有帮助, 如 `runx` 和 `restoretext` 等。这些软件可以从 Sunsite 站点的 `/pub/Linux/libs` 目录下获取。

Q: 有一张正常的 SVGA 卡, 有 1MB 的显示内存。用此卡可以使用 1152×900 或 1024×1024 的虚拟分辨率。但是用 S3 卡却不能, 为什么?

A: 目前的 S3 卡无法用 1 152 个像素点的线长度, 所以用 1 280 代替。而且 S3 服务器要 1KB 显存给鼠标, 所以 1MB 的虚拟解析度是无法运行的。

用户需要一些卡上的内存做字体与图素的快速存取之用, 才能有最大的效率。所以这些内存并不是浪费的。在 1MB 显存的卡上, 建议用 1024×768 和 1152×800 分辨率。若使用后者, 所剩的供字体快速存取的内存就有限了, 因此当使用比较大的字体时, 性能也有明显的差别。一般当字体的点数小于 18 时, 还是很正常。

Q: 要用 LinkKit 来配置 XF86_S3 或 XF86_VGA16 服务器, 在修改过 `sited.def` 和运行过 `mkmf` 后, 执行 `make` 命令时却失败了, 为什么?

A: LinkKit 的 `Imakefile` 有一些错误, 用户可以修改 `Imakefile` 的第 326 行, 把

```
#if XF86SVGAserver || XF86MonoServer
```

改成

```
#if XF86SVGAserver || XF86MonoServer || XF86VGA16Server ||  
XF86S3Server
```

或者也可以配置 XF86_Mono 或 XF86_SVGA 服务器, 然后就可以了。

3.8 在 Linux 上启动 quota

在 Linux 系统上, 管理员可以控制某个用户对硬盘的使用, 例如限制 `demo` 用户最多占用硬盘上的 10MB 空间。这种工作可以通过 Linux 系统的 quota 机制来实现。下面就看一看如何设置 quota。

3.8.1 quota 简介

quota 可以让用户从两方面指定硬盘的储存限制:

1. 使用者所能够支配的索引节点数。

2. 使用者可以取用的硬盘分区块数。

quota 的功能是强制使用者在大部分的时间中保持在他们的硬盘使用限制之下，取消他们在系统上无限制地使用硬盘空间的能力。

quota 是以使用者和文件系统为基础的。如果使用者在一个以上的文件系统上建立文件，那么必须在每一文件系统上分别设置 quota。

3.8.2 quota 目前在 Linux 上的情况

quota 的支持自从 1.3.8x 版开始已经包含到内核之中，如果用户的系统不支持 quota，那么建议用户更新内核。

到目前为止，quota 只能在 ext2 文件系统上运行。

在Linux上使用quota的功能需求

1. Linux 内核

支持 quota 的 Linux 内核可以从以下站点取得：

<http://sunsite.unc.edu/pub/Linux/kernel/v2.0>

2. quota 软件

视用户的 Linux 发行软件包而定，在用户的系统上可能已经装好，也可能没有 quota 软件。quota 软件的程序源代码可以从如下站点取得：

<ftp://ftp.funet.fi/pub/Linux/PEOPLE/Linus/subsystems/quota/all.tar.gz>

3.8.3 配置 quota

下面介绍如何配置 quota。

3.8.3.1 配置用户的内核

重新配置用户的内核并且对这个选项回答 y 以加入 quota 的支持：

```
quota support (CONFIG_QUOTA) [n] y
```

重新配置用户的内核需要执行下面的命令：

```
cd /usr/src/Linux
make config
```

3.8.3.2 编译并安装quota软件

如果用户的系统上还没有 quota 软件，则需要安装该软件。

quota 软件的程序源代码可以从如下站点取得：

`ftp://ftp.funet.fi/pub/Linux/PEOPLE/Linus/subsystems/quota/all.tar.gz`

3.8.3.3 设置

修改用户系统的启动脚本文件以检查 quota 并且在启动时打开 quota，下例为某个脚本程序：

```
#Check quota and then turn quota on.
if [ -x /usr/sbin/quotacheck ]
then
    echo "Checking quotas. This may take some time."
    /usr/sbin/quotacheck -avug
    echo " Done."
fi

if [ -x /usr/sbin/quotaon ]
then
    echo "Turning on quota."
    /usr/sbin/quotaon -avug
fi
```



注意： 一定要在加载用户 `/etc/fstab` 里的文件系统之后打开 quota，否则 quota 将不会运行。

3.8.3.4 修改 `/etc/fstab` 文件

为了在系统每次启动的时候都使 quota 有效，需要修改 `/etc/fstab` 文件。

没有启用 quota 的分区一般如下所示：

```
/dev/hda1    /          ext2      defaults          1 1
/dev/hda2    /usr       ext2      defaults          1 1
```

要在文件系统上加入 quota，则在包含 “defaults” 的第四个项后面加上 “usrquota”：

```
/dev/hda1    /          ext2      defaults          1 1
/dev/hda2    /usr       ext2      defaults,usrquota 1 1
```

如果用户需要在一个文件系统中的用户组 quota 支持，将 “usrquota” 改为 “grpquota”：

```
/dev/hda1    /          ext2      defaults          1 1
/dev/hda2    /usr       ext2      defaults,grpquota 1 1
```

如果要同时支持 quota 与用户组 quota，则修改如下：

```
/dev/hda1    /          ext2      defaults          1 1
/dev/hda2    /usr       ext2      defaults,usrquota,grpquota 1 1
```

3.8.3.5 建立 quota.user 文件

quota 记录文件 `quota.user` 应该由 root 拥有，而且只有 root 有读写的权限而其他任何

人都没有。

以 root 的身份登录。进入用户想要启用 quota 的分区的最上层，并且按如下方式建立 quota.user 文件：

```
touch /partition/quota.user
touch /partition/quota.group
chmod 600 /partition/quota.user
chmod 600 /partition/quota.group
```

3.8.3.6 重新启动

现在重新启动系统以便让用户所做的改变生效。

同时注意以后对于用户还想要启用 quota 的其他分区只需要最后三步，即：修改 /etc/fstot.b 文件、建立 quota.use 文件及重新启动。

3.8.4 高级设置

为使用者及用户组指定 quota 这项操作需要执行 edquota 命令。(详情参考 man edquota)。

通常在编辑 quota 之前先使用 quotacheck 并加上参数 -avug，以取得最新的文件系统的使用状况。这只是一个人的习惯问题，并非必要步骤。

3.8.4.1 为特定使用者指定 quota

举个例子。在本系统上有一个名叫 bob 的使用者。使用 edquota bob 命令将会把用户带进 vi(或是在用户的 \$EDITOR 环境变量中所指定的编辑器)以便为使用者 bob 编辑在每个启用 quota 的分区上的 quota:

```
quotas for user bob:
/dev/hda2: blocks in use: 2594, limits (soft = 5120, hard = 6400)
          inodes in use: 356, limits (soft = 1000, hard = 1500)
```

“blocks in use”是使用者在某个分区上已经使用的区块总数(以数据块为单位)。
“inodes in use”是使用者在某个分区上所拥有的总数文件。

3.8.4.2 为特定用户组指定 quota

举个例子，系统上有一个用户组 games。

使用 “edquota -g games”，把用户带入 vi 以便为 games 用户组编辑 quota:

```
quotas for group games:
/dev/hda4: blocks in use: 5799, limits (soft = 8000, hard = 10000)
          inodes in use: 1454, limits (soft = 3000, hard = 4000)
```

3.8.4.3 为一群使用者指定相同的quota值

为了快速地为系统上的一群使用者，例如 100 名，设置和 bob 相同的 quota 值，首先以手动编辑 bob 的 quota 有关信息，然后执行如下命令：

```
edquota -p bob `awk -F: '$3 > 499 {print $1}' /etc/passwd`
```

这里假设用户使用 csh，并且用户的使用者 UID 从 500 开始。

除了 edquota 之外，用户还应该熟悉如下三个专有名词：软性限制、硬性限制和缓冲期间。

1. 软性限制

一般来说，软性限制指出 quota 使用者在分区上拥有的硬盘用量总数。

2. 硬性限制

硬性限制只在设置了缓冲期间时才会运行。它指出硬盘用量的绝对限制，quota 使用者不能超越他的硬性限制。

3. 缓冲期间

缓冲期间是对 quota 使用者实行软性限制之前的时间限制。可以使用的时间单位是秒、分、小时、日、星期和月。用户使用“edquota -t”命令后，将会得到如下信息：

```
Time units may be: days, hours, minutes, or seconds
Grace period before enforcing soft limits for users:
/dev/hda2: block grace period: 0 days, file grace period: 0 days
```

把 0 days 这个部分修改为用户觉得合适的任意期限。

3.8.4.4 其他的quota命令

1. quotacheck

quotacheck 用来扫描文件系统的硬盘用量，并更新 quota.user 这个 quota 记录文件到最新的状态。建议定期在系统启动时执行 quotacheck 或通过 cron 任务每隔一段时间(例如一星期)执行一次。

2. repquota

repquota 产生文件系统的有关 quota 的信息摘要。如下是 repquota 的输出例子：

```
# repquota -a
```

		Block limits				File limits			
User		used	soft	hard	grace	used	soft	hard	grace
root	--	175419	0	0		14679	0	0	
bin	--	18000	0	0		735	0	0	
uucp	--	729	0	0		23	0	0	
man	--	57	0	0		10	0	0	
user1	--	13046	15360	19200		806	1500	2250	
user2	--	2838	5120	6400		377	1000	1500	

3.8.5 quotaon 和 quotaoff

quotaon 是用来打开 quota 的命令, quotaoff 则是将其关闭。实际上两者很相似。它们分别是在系统启动与关机时执行的。

quotaon 的使用方法是:

```
# quotaon /filesystem
```

quotaoff 的使用方法是:

```
# quotaoff /filesystem
```

更详细的信息可以查阅这两个命令的联机帮助手册。

3.9 ADSM 备份系统

这部分将描述如何在 Linux/i386 之中针对商业性的 ADSM 备份系统去安装和使用客户端。

ADSM 是一个由 IBM 开发的以网络为基础的备份系统, 它应用广泛, 其中包含各式各样的系统(各种 UNIX、Windows 系列、Novell、Macintosh 和 Windows NT 系统)。但是, 目前没有 Linux 版本。

在 Linux 下, 必须使用 SCO 二进制文档, 用户需要安装 iBCS2 模拟器来执行 ADSM。本节内容针对的是 ADSM V2 R1。

3.9.1 安装 iBCS 模块

要使用 ADSM 备份系统, 必须先安装 iBCS 模块。

iBCS2 模块可以从以下站点取得:

```
ftp://tsx-11.mit.edu/pub/Linux/BETA/ibcs2
```

如果 Linux 内核版本是 1.2.13, 相应的文件是 ibcs-1.2-950721.tar.gz, 解开它后安装补丁程序: ibcs-1.2-950808.patch1 和 ibcs-1.2-950828.patch2。

再键入 make 命令, 然后使用 insmod 命令来安装 iBCS 模块。

针对一个 2.0 内核版本, 取得 ibcs-2.0-960610.tar.gz, 解开它放在一个合适的放置, 进入到此目录中, 使用下面的补丁程序:

```
--- iBCSemul/ipc.c.orig Wed Jan 15 21: 32: 15 1997
+++ iBCSemul/ipc.c Wed Jan 15 21: 32: 31 1997
@@ -212, 7 +212, 7 @@
 switch (command) {
     case U_SEMCTL:
```

```

        cmd = ibcs_sem_trans(arg3);
-        arg4 = (union semun *)get_syscall_parameter (regs, 4);
+        arg4 = (union semun *)(((unsigned long *) regs->esp) + (5));
        is_p = (struct ibcs_semid_ds *)get_fs_long(arg4->buf);
#ifdef IBCS_TRACE
        if ((ibcs_trace & TRACE_API) || ibcs_func_p->trace)

```

然后, 复制 CONFIG.i386 文件到 CONFIG, 键入 make 命令。

如果用户还没有下面的设备, 按照下列命令建立必要的设备文件:

```

# cd /dev
# ln -s null XOR
# mknod socksys c 30 0
# mknod spx c 30 1

```

3.9.2 安装 ADSM 客户端

SCO 二进制文档由三个归档文件(tar)提供。

改变路径到根目录, 根据用户的策略设置 umask 值, 然后在一个目录下解开它们(如根目录)。在目录/tmp 下会找到相应的程序文件, 执行该文件。

这时必须手动编辑文件/usr/adsm/dsm.sys/usr/adsm/dsm.opt。

在文件 dsm.sys 中, 重要的参数详细说明如下:

Servername 服务器名称

TCPServeraddress 服务器 DNS 全名

NODename 用户自己的主机名

在文件 dsm.opt 中, 必须要详细说明如下参数:

Server 如同前面的描述

Followsymbolic 要或不要跟随符号链接点

SUBdir 支持子目录(通常需要它)

domain 文件系统备份

这时必须从/etc/fstab 文件创建一个与 SCO 相兼容的/etc/mnttab 文件。可以使用下面的 perl 程序文件 fstab2mnttab 来完成这个步骤:

```

#!/usr/bin/perl

$mnttab_struct = "a32 a32 I L";
open(MTAB, "/etc/mtab") || die "Cannot open /etc/mtab: $!\n";
open(MNTTAB, ">/etc/mnttab") || die "Cannot open /etc/mnttab: $!\n";

while(<MTAB>) {
    next if /pid/;
    chop;
    /^(\S*)\s(\S*)\s(\S*)\s.*$/;

```

```
$device = $1;
$mountpt = $2;
$fstype = $3;
if($fstype ne "nfs" && $fstype ne "proc") {
    $mnttab_rec =
        pack($mnttab_struct, $device, $mountpt, 0x9d2f, time());
    syswrite(MNTTAB, $mnttab_rec, 72);
    print "Made entry for: $device $mountpt $fstype\n";
}

close(MNTTAB);
exit 0;
```

3.9.3 执行客户端

在 ADSM 备份系统中，有两种客户端：

1. dsm

该客户端是 X11 界面。

2. dsmd

该客户端是命令行界面。

具体的使用，在该程序中有详细的说明，例如：

```
dsmd schedule -quiet 2>&1 >/dev/null &
```

3.9.4 已知的问题

但是，SCO 只能处理少于 8 个字符的主机名。如果用户的主机名更长，或者用 Internet 全名，还需要在 /usr/adsm/dsm.sys 中的 NODename 行详细说明系统的机器名。

如果使用 DISPLAY 变量，必须提供服务器的 DNS 全名（如：DISPLAY=host.full.domain: 0 而不是 DISPLAY=host: 0）。

3.10 ZIP 磁盘

ZIP 磁盘是类似于软盘的可移动式磁盘，不同于软盘的是，它提供了比软盘更大的存储容量，一般是 100MB。ZIP 磁盘通常用于备份数据，除此之外，它还可以完成其他的许多功能，本节将介绍 ZIP 磁盘的使用情况。

3.10.1 ZIP 磁盘简介

本节分为四部分,第一部分描述如何设置 ZIP 磁盘,适用于 Linux 的 RedHat 和 Slackware 版本。

第二、第三和第四部分则分别描述如何在 ZIP 磁盘上安装 RedHat 4.2、Slackware 2.2 和 Debian 1.2。

3.10.2 设置 ZIP 磁盘

(这部分内容适用于 RedHat 和 Slackware 版本。)在开始之前先确定用户可以使用 ZIP 磁盘;在内核中已经支持 ppa 或是安装了 ppa 的模块。

可以用一个简单的方法来检查:

```
==> dmesg
```

假如 dmesg 的内容相当长,用户可以用管道输送到 more:

```
==> dmesg | more
```

下列是输出的一部分:

```
scsi0 : PPA driver version 0.26 using 4-bit mode on port 0x3bc.
scsi : 1 host.
  Vendor: IOMEGA   Model: ZIP 100           Rev: D.08
  Type:   Direct-Access                ANSI SCSI revision: 02
Detected scsi removable disk sda at scsi0, channel 0, id 6, lun 0
SCSI device sda: hdwr sector= 512 bytes. Sectors= 196608 [96 MB] [0.1 GB]
  sda: Write Protect is off
  sda: sdal sda2
```

假如用户看到的内容如下所示:

```
scsi : 0 hosts.
scsi : detected total.
```

表示用户有 SCSI 的支持,但没有发现 ZIP 磁盘。

3.10.2.1 分区ZIP磁盘

要分区 ZIP 磁盘,执行 fdisk 命令:

```
==> fdisk /dev/sda
```

下面举例说明分区表的设置:

```
Disk /dev/sda: 64 heads, 32 sectors, 96 cylinders
Units = cylinders of 2048 * 512 bytes
```

Device	Boot	Begin	Start	End	Blocks	Id	System
/dev/sda1		1	1	81	82928	83	Linux native
/dev/sda2		82	82	96	15360	82	Linux swap

3.10.2.2 格式化和安装ZIP磁盘

在执行 `fdisk` 之后，格式化新的分区：

```
==> mke2fs -c /dev/sda1
```

最后用户将需要安装 ZIP 磁盘：

```
==> mount /dev/sda1 /iomega -t ext2
```

用户还需要格式化交换分区：

```
==> mkswap -c /dev/sda2
```

3.10.2.3 建立启动磁盘

因为 **ppa**(打印机端口)版本的 ZIP 驱动程序不是真正的 SCSI 设备，即它不是可启动的装置，所以需要有一个内核含有 **ppa** 支持，而不是模块化的引导磁盘。

1. 设置和制作内核

首先用户需要设置并制作一个内核，其中要有启动 **ppa** 支持的设置，而不只是设置成装入式的模块。

为了要有 **ppa** 的选择，选择 SCSI 支持：

```
SCSI support (CONFIG SCSI) Y/m/n/?
```

另外，还要加入 SCSI 磁盘支持：

```
SCSI disk support (CONFIG_BLK_DEV_SD) Y/m/n/?
```

最后，在 SCSI 低级驱动器之下是 **ppa** 支持：

```
IOMEGA Parallel Port ZIP drive SCSI support (CONFIG SCSI_PPA) Y/m/n/?
```



注意： 再一次提醒用户，确定不要将 **ppa** 设置成模块，而应该放在内核中。

另外，因未使用 **parport** 内核补丁程序，**ppa** 驱动程序不允许 ZIP 磁盘上的被动式端口给打印机使用，用户可能必须在 **Parallel printer support** 项出现时选择 N：

```
Parallel printer support (CONFIG_PRINTER) N/y/m/?
```

设置好内核，编译生成内核文件：

```
==> make dep; make clean; make zImage
```

新的内核应该出现在 `arch/i386/boot/zImage` 文件中。

2. 安装 LILO

以下步骤就是建立 LILO 的过程，使系统在启动的时候可以使用相应的设备。

1) 建立 ext2 文件系统

在磁盘上建立 ext2 文件系统, 使用如下命令:

```
==> mke2fs -t /dev/fd0
```

2) 拷贝基本的文件

接下来是确定有一个目录可做安装点并设置安装软盘文件系统(这儿使用 /mnt/floppy):

```
==> mount /dev/fd0 /mnt/floppy -t ext2
==> cp /boot/boot.b /mnt/floppy
==> cp /boot/map /mnt/floppy
==> cp /usr/src/Linux/arch/i386/boot/zImage /mnt/floppy/vmlinuzDESK
```

为了顺利启动, 还需要用户设置 LILO 部分:

建立 LILO 的配置文件。首先给在 ZIP 磁盘上的那些内核建立 LILO 的配置文件 (/mnt/floppy/lilo.conf)。下面是配置文件的一个例子:

```
boot=/dev/fd0
map=/mnt/floppy/map
install=/mnt/floppy/boot.b
prompt
compact
timeout=50
image=/mnt/floppy/vmlinuzLAP
    label=Laptop
    root=/dev/sda1
    read-only
image=/mnt/floppy/vmlinuzDESK
    label=Desktop
    root=/dev/sda1
    read-only
image=/mnt/floppy/vmlinuzDESK
    label=rescue
    root=/dev/hdc1
    read-only
```

上面的配置文件中有两个内核, 一个是给需要在内核中有数学模拟程序的笔记本型 486, 另一个是给桌上型计算机用。

然后, 执行 LILO 命令把系统的引导设置安装到软盘上:

```
==> lilo -C /mnt/floppy/lilo.conf
```

LILO 被安装到磁盘上后, 就可以跳过下面两个步骤。

3. 内核的安装

新创建的内核拷贝到软盘上:

```
==> cp arch/i386/boot/zImage /dev/fd0
```

或

```
==> cat arch/i386/boot/zImage > /dev/fd0
```

4. 设置软盘的根目录和交换区

内核安装在软盘上后, 用户需要把根目录设备设置到 ZIP 磁盘上:

```
==> rdev /dev/fd0 /dev/sda1
```

然后, 执行下面的命令:

```
==> rdev -s /dev/fd0 /dev/sda2
```

3.10.3 RedHat 4.2 的安装

在安装时, 假设使用的是 Kernel 2.0.30. Imoega ppa (打印机端口版的 ZIP 机) RedHat 4.2 系统。

3.10.3.1 软件包的安装

1. 要安装的软件包

在本次安装中, 要包含网络的支持。但因为 RedHat 的 X Window 网络设置方式, 将必须手工调整, 或是试着去设置在 /etc/sysconfig/ 目录内的网络设置工作脚本文件。

在目前版本的 Linux 系统中, 在 ZIP 磁盘中编译程序是相当慢的, 所以用户最好不安装任何开发软件包。

基于空间的考虑, 也选择不安装 X Window 系统。稍后, 用户可以试着去挂上硬盘并建立一个与 /usr 的符号链接, 看看是否可让 X Window 运行。

下面以安装的顺序列出本例子中安装的软件包。有星号的表示在 RedHat 的勘误中有更新。括弧中的则是更新的软件包号码。

例如, NetKit-B-0.09-6 被更新到 NetKit-B-0.09-8, 所以内容将是(使用命令 rpm--root/iomega-qa):

- | | |
|--------------------------|------------------------|
| 1) setup-1.7-2 | 2) pamconfig-0.51-2 |
| 3) filesystem-1.3-1 | 4) MAKEDEV-2.2-9 |
| 5) adduser-1.7-1 | 6) libc-5.3.12-18 |
| 7) SysVinit-2.64-8 | 8) ash-0.2-8 |
| 9) at-2.9b-2 | 10) libtermcap-2.0.8-4 |
| 11) bash-1.14.7-1 | 12) bc-1.03-6 |
| 13) bdflush-1.5-5 | 14) cpio-2.4.2-4 |
| 15) cracklib-dicts-2.5-1 | 16) tmpwatch-1.2-1 |
| 17) crontabs-1.5-1 | *18) db-1.85-10 (-11) |
| 19) dev-2.5.1-1 | 20) diffutils-2.7-5 |
| 21) etcskel-1.3-1 | 22) file-3.22-5 |
| 23) fileutils-3.16-1 | 24) findutils-4.1-11 |
| 25) grep-2.0-5 | 26) groff-1.10-8 |

- | | |
|---------------------------------|-----------------------------------|
| *27) ld.so-1.7.14-4 (-5) | 28) getty_ps-2.0.7h-4 |
| 29) gzip-1.2.4-7 | 30) mingetty-0.9.4-3 |
| *31) initscripts-2.92-1 (93-1) | 32) ed-0.2-5 |
| 33) info-3.9-1 | 34) ncurses-1.9.9e-4 |
| 35) libg++-2.7.1.4-5 | *36) pwdb-0.54-3 (-4) |
| 37) rootfiles-1.5-1 | *38) pam-0.57-2 (-4) |
| 39) redhat-release-4.2-1 | 40) less-321-3 |
| 41) mount-2.51-2 | 42) zlib-1.0.4-1 |
| 43) rpm-2.3.11-1 | 44) e2fsprogs-1.10-0 |
| 45) sysklogd-1.3-15 | 46) tar-1.11.8-11 |
| 47) passwd-0.50-7 | 48) gawk-3.0.2-1 |
| 49) gdbm-1.7.3-8 | 50) gpm-1.10-8 |
| 51) hdparm-3.1-2 | 52) kbd-0.91-9 |
| 53) slang-0.99.37-2 | 54) newt-0.8-1 |
| 55) kbdconfig-1.4-1 | 56) ncompress-4.2.4-7 |
| *57) sh-utils-1.16-4 (-5) | 58) procinfo-0.9-1 |
| *59) logrotate-2.3-3 (4-1) | 60) LILO-0.19-1 |
| 61) losetup-2.51-2 | 62) Linuxthreads-0.5-1 |
| *63) mkinitrd-1.6-1 (7-1) | 64) mailcap-1.0-3 |
| *65) man-1.4h-5 (j-1) | 66) mt-st-0.4-2 |
| 67) modules-2.0.0-5 | 68) mailx-5.5.kw-6 |
| 69) net-tools-1.32.alpha-2 | 70) procmail-3.10-10 |
| 71) procps-1.01-11 | 72) psmisc-11-4 |
| 73) quota-1.55-4 | 74) readline-2.0-10 |
| 75) sed-2.05-6 | 76) setconsole-1.0-1 |
| 77) sendmail-8.8.5-4 | 78) shadow-utils-960530-6 |
| 79) stat-1.5-5 | 80) tcsh-6.06-10 |
| 81) termcap-9.12.6-5 | 82) textutils-1.22-1 |
| 83) time-1.7-1 | 84) timeconfig-1.8-1 |
| 85) util-Linux-2.5-38 | 86) vim-4.5-2 |
| 87) vixie-cron-3.0.1-14 | 88) which-1.0-5 |
| 89) zoneinfo-96i-4 | 90) tcp_wrappers-7.5-1 |
| *91) NetKit-B-0.09-6 (-8) | *92) lpr-0.18-1 (19-1) |
| *93) bind-4.9.5p1-2 (9.6-1) | *94) bind-utils-4.9.5p1-2 (9.6-1) |
| *95) wu-ftpd-2.4.2b12-6 (b15-1) | 96) anonftp-2.3-3 |
| 97) zip-2.1-1 | 98) unzip-5.12-5 |
| 99) statserial-1.1-7 | 100) minicom-1.75-2 |
| 101) lrzsz-0.12.14-1 | 102) dip-3.3.7o-9 |
| 103) ppp-2.2.0f-3 | 104) portmap-4.0-3 |

105) perl-5.003-8	*106) traceroute-1.0.4.4bsd-2 (1.4a5-1)
*107) elm-2.4.25-7 (-8)	108) lynx-2.6-2
109) ncftp-2.3.0-5	110) pine-3.95-2
111) rdate-0.960923-1	112) apache-1.1.3-3
*113) nfs-server-2.2beta16-7 (2.2beta16-8)	*114) nfs-server-clients-2.2beta16-7 (2.2beta16-8)

2. 更新和勘误

正如许多 RedHat 使用者所知, 部分的软件包可能被发现有一些安全漏洞, 或有其他会引起争议的问题。因此 RedHat 的这些软件包被更新和勘误。上例中已经列出了那些用户安装且有更新的软件包, 并且在更新的软件包前标上了星号。

参考关于更新软件包的 RedHat 网页可获得更详细的信息:

<http://www.redhat.com/support/docs/rhl/rh42-errata-general.html>



注意: 在更新那些软件包前, 必须执行命令 `ldconfig`。

3. 10. 3. 2 使用rpm而不用glint安装

在 `rpm` 中, 使用 `--root` 选项去指定加载目录为安装装置的根目录。因为安装前和安装后的执行脚本文件基于不同的根目录而无法正确地被执行, 现已发现有许多软件包无法安装。所以使用 `--noscripts` 选项:

```
==> rpm --root /iomega -i --noscripts PACKAGE.i386.rpm
```

肯定许多人会注意到, 将得到一个错误信息, 如下所示:

```
failed to open /iomega/var/lib/rpm/packages.rpm
```

```
error: cannot open /iomega/var/lib/rpm/packages.rpm
```

所以需要建立 `var/lib/rpm` 目录:

```
==> mkdir /iomega/var mkdir /iomega/var/lib mkdir /iomega/var/lib/rpm
```

3. 10. 3. 3 pamconfig与pam的执行顺序

假如有人试图安装 `pamconfig`, 它将会显示关于 `pam` 的相关属性错误; 当用户再试着安装 `pam`, 它会显示关于 `pamconfig` 的相关属性错误。

用户可以强迫 `pamconfig` 被安装。

强迫 `pamconfig` 被安装需要使用 `--nodeps` 选项, 如:

```
==> rpm --root /iomega -i --nodeps --noscripts pamconfig-0.51-2
```

3. 10. 3. 4 软件包安装后的问题

所有的内容都安装后, 该磁盘的功能还不完整。

假如用户试着启动, 会得到两个错误:

第一个是/etc/ld.so.cache 文件不存在;

第二个是找不到 libc.so.5。

- /etc/ld.so.cache

为了建立这个文件, 当 ZIP 磁盘正在安装时, 用户将必须执行 ldconfig 命令:

```
==> chroot /iomega; /sbin/ldconfig
```

- libc.so.5

为了解决找不到程序库的问题, 用户必须建立一个安装执行脚本文件时应该建立的符号链接:

```
==> cd /iomega/lib; ln -s libc.so.5.3.12 libc.so.5
```

- 设置 root 的密码

正如执行 ldconfig 一样, 用户也要为新系统改变或建立 root 的密码:

```
==> chroot /iomega; passwd root
```

- 安装程序所建立的内容

当查看 RedHat 4.2 光盘或硬盘时, 会看到一些内容: 安装程序的程序源代码。用户还会发现在/misc/src/install 目录下有一个有用的 net.c 文件。

在 net.c 文件里面, 还会发现安装程序执行时建立的那些文件。大部分只是产生网络应用(这就是被称为 net.c 的原因), 即使没有网卡, 用户仍然可用 localhost 当作网络。这些文件包括:

/etc/hosts

/etc/HOSTNAME

/etc/resolv.conf

/etc/sysconfig/network

/etc/sysconfig/network-scripts/ifcfg-eth0

或用户可能有的其他网络装置。

3.10.3.5 网络的设置

本节中, 将要配置 3COM3C595 快速以太网网卡。

首先, 需要一个主机名字, 例如: dash-dot.wig.org。有名字而没有 IP 地址也是无意义的, 所以用了一个非因特网保留的可用的网址 192.168.10.0。即使主机名称通常存在 /etc/HOSTNAME 中, RedHat 也会在 /etc/sysconfig/network 中检查这个名称, 所以从 etc/sysconfig/network 文件开始执行。

/etc/sysconfig/network 的样本:

```
NETWORKING=yes
HOSTNAME=dash-dot
DOMAINNAME=wig.org
GATEWAY=
GATEWAYDEV=eth0
NS1=192.168.10.7
```

拷贝基本上相同的数据到/etc/HOSTNAME、/etc/resolv.conf 和/etc/hosts 中。

/etc/HOSTNAME:

```
dash-dot.wig.org
```

/etc/resolv.conf:

```
search wig.org
```

```
nameserver ns.wig.org
```

/etc/hosts:

```
127.0.0.1 localhost
```

```
192.168.10.99 dash-dot.wig.org dash-dot
```

```
192.168.10.7 ns.wig.org ns
```

RedHat 在/etc/sysconfig/network-scripts 的执行脚本文件中设置所有的网络装置。

网络的设置通常先由安装程序建立，所以必须手动建立这些配置文件。它们全是由 ifcfg-xxx 开头，其中 xxx 是 ifconfig 启动的网络接口，如 ppp0 和 eth0 等等。在这本例中，必须建立一个 ifcfg-eth0 文件，内容包括：

```
DEVICE=eth0
ONBOOT=yes
BOOTPROTO=none
BROADCAST=192.168.10.255
NETWORK=192.168.10.0
NETMASK=255.255.255.0
IPADDR=192.168.10.99
```

最后，为了让这个设置会成功，还要在/etc/sysconfig/network-scripts 目录下，执行如下命令：

```
==> ./ifup ifcfg-eth0 boot
```

假如接口被设成在系统启动时自动启动，这将会启动设置网络接口的脚本文件。

3.10.4 Slackware 2.2 的安装

下面简单介绍 Slackware 2.2 的安装。

3.10.4.1 需求

- ZIP 磁盘和计算机
- 有 ZIP 支持的内核
- ZIP HOWTO 文档(推荐)
- 格式化的 1.44 英寸软盘
- 一到两个小时的时间

3.10.4.2 安装

最简单的开始方法是直接安装 Slackware 到 ZIP 磁盘中。因为 Slackware 是一个比 RedHat 小的发行版本，而且它可能比较容易除去不想要的软件包。

安装 Slackware 2.2 的步骤与前面讲的安装 RedHat 4.2 的步骤是一样的，用户可以参阅 3.10.3 节的内容。

3.10.4.3 安装内容

最难的部分是决定安装内容。显然用户需要 A 系列(大约 8MB)，其他就可随便了。

最后，用户可以把 Slackware 版缩小到 70MB 安装，其中包含 gcc/g++、perl、X11R6(不是全部)、sendmail 和联机手册文件，最后大约剩下 10MB 给使用者文件用。

3.10.5 设立/etc/fstab

(适合于 RedHat 和 Slackware 发行版本)

在重新启动前，最后一件需要做的事是在将会成为根分区的文件系统上设立 fstab 文件。下面是/iomega/etc/fstab 文件的样本，用户在这个文件中至少要包含如下的内容：

```
/dev/sda1    /          ext2        defaults    1    1
/dev/sda2    none       swap        sw
none         /proc      proc        defaults    1    1
```

保存该文件并存档，就可以用新制作的引导磁盘重新启动系统。

3.10.6 Debian 的安装

下面介绍 Debian 的安装。

3.10.6.1 需求

- 连接打印机端口的 ZIP 磁盘
- 两张 1.44MB 的空白软盘
- 一套完整的 Debian 安装磁盘(假如用户不记得如何制作这些磁盘，复习 Debian 的安装文件说明)
- 几个小时的时间

3.10.6.2 制作修复盘

Debian 修复盘是一种 SYSLinux 式的引导磁盘，它使用和一个特殊的引导程序加载器 (boot loader) 去避免装入 DOS。下面讲述制作修复盘的具体步骤：

1. 使用 dd(或是 DOS 中的 RAWRITE)创建新的修复盘

本部分的内容在前面都已经讲过了，假如用户不记得如何做，复习 Debian 的安装文件。

2. 创建一个有 ZIP ppa 支持的内核

创建一个有 ZIP ppa 支持的内核，用户还需要规划 msdos、fat、minix、ext2fs 和 procfs 文件系统。

同时设立用户想最终在 ZIP 磁盘上安装的模块。

内核规划好后，用下面的命令：

```
==> make dep; make clean
==> make bzImage
```

重新创建内核。

创建模块用下面的命令：

```
==> make modules
```



注意： 确定用 make bzImage 命令，而不是 make zImage 命令。

3. 安装新的修复盘

使用的命令是：

```
==> fdmount fd0
```

或：

```
==> mount /dev/fd0 /mnt
```

4. 把内核拷贝到软盘

这一步很简单，只要用 cp 命令将 Linux 内核(在 i386 的平台上，它位于 arch/i386/boot/bzImage 中)拷贝到软盘上，软盘上的目标文件是/mnt/Linux：

```
#cp arch/i386/boot/bzImage /mnt/Linux
```

5. 编辑脚本文件 rdev.sh

用用户最喜欢的编辑器编辑位于修复盘上的脚本文件 rdev.sh。

修改最后一行：

即把：

```
'rdev /mnt/Linux /dev/ram0'
```

改成：

```
'rdev /mnt/Linux /dev/sda1'
```

用户也需要把目录名/mnt/Linux 改成正确的路径名。例如，如果用户安装的软盘在 /fd0，必须把/mnt/Linux 修改为成/fd0/Linux：

如果用户使用 C Shell，则使用如下命令：

```
#setenv PATH $PATH:/fd0/Linux
```

6. 执行修改后的脚本文件 rdev.sh

在用户编辑位于修复盘上的脚本文件 rdev.sh 后，就可以执行该文件了：

```
==> ./rdev.sh
```

3.10.6.3 把基本系统安装到ZIP磁盘

用修改后的修复盘启动。假如一切正常，用户将会得到熟悉的 Debian 的菜单式的安装过程，但它现在已经可以识别用户的 ppa ZIP 磁盘了。

这一步骤与将系统安装到硬盘上是相同的，但需要将 `/dev/sda1` 安装成根目录并将 `/dev/sda2` 设置为交换区。

3.10.6.4 创建启动盘

用户可以创建启动盘，可以使用 Debian 安装过程中的 Create Boot Disk(创建启动盘)选项来创建启动盘。

3.10.6.5 重新启动系统

插入启动盘并在安装菜单里选择 Reboot(重启动)选项。

3.10.6.6 安装在前面步骤中创建的模块

用户还可以加载包含用户创建的内核的硬盘分区。改变目录到正确的目录并执行 `make modules_install` 命令。

下面是详细的步骤：

```
==> mount /dev/hda2 /mnt
==> cd /mnt/usr/src/Linux
⇒ make modules_install
```

第 4 章 IP 别名和 IP Marsquerade

IP 别名和 IP Marsquerade 是 Linux 系统很特别的功能，能够帮助用户实现在其他系统中不能实现的功能，本章将对这两个功能做详细的介绍。

4.1 Linux 下的 IP 别名

本节将说明如何在 Linux 机器上设置并执行 IP 别名。除此之外，还介绍如何设置在 IP 别名上接收电子邮件等。

IP 别名是指在一块网卡上绑定多个 IP 地址。

在普通状态下，一块网卡只能绑定一个 IP 地址，但通过 IP 别名，就可以实现在一块网卡上绑定多个 IP 地址。

下面讲述如何在 Linux 系统下配置和使用 IP 别名。

4.1.1 配置步骤

1. 装入 IP Alias 模块(如果用户将它编译进内核可跳过此步骤)，需要使用如下命令：

```
/sbin/insmod /lib/modules/`uname -r`/ipv4/ip_alias.o
```

2. 建立回送装置，以及 eth0 接口上以主 IP 地址开头的的所有 IP 地址：

```
/sbin/ifconfig lo 127.0.0.1
/sbin/ifconfig eth0 up
/sbin/ifconfig eth0 172.16.3.1
/sbin/ifconfig eth0: 0 172.16.3.10
/sbin/ifconfig eth0: 1 172.16.3.100
```

172.16.3.1 是主 IP 地址，而.10 及.100 是别名。

值得注意的是 eth0: x，其中 x=0、1、2、...、n 在每一不同的 IP 地址上。

主 IP 地址不需要这个。

3. 建立回送路由。

首先，加路由到回送装置，然后是局域网络，最后是不同的 IP 地址及默认回送路由：

```
/sbin/route add -net 127.0.0.0
```

```

/sbin/route add -net 172.16.3.0 dev eth0
/sbin/route add -host 172.16.3.1 dev eth0
/sbin/route add -host 172.16.3.10 dev eth0: 0
/sbin/route add -host 172.16.3.100 dev eth0: 1
/sbin/route add default gw 172.16.3.200

```

上面的例子只显示了三个 IP 地址。在 `/usr/include/Linux/net_alias.h` 文件中定义的系统可以具有的 IP 别名个数的最大值是 256，也就是说，在一块网卡上最多只能绑定 256 个 IP 地址。

执行 `/sbin/ifconfig` 命令的结果如下所示：

```

lo          Link encap: Local Loopback
            inet addr: 127.0.0.1 Bcast: 127.255.255.255 Mask: 255.0.0.0
            UP BROADCAST LOOPBACK RUNNING MTU: 3584 Metric: 1
            RX packets: 5088 errors: 0 dropped: 0 overruns: 0
            TX packets: 5088 errors: 0 dropped: 0 overruns: 0

eth0        Link encap: 10Mbps Ethernet HWaddr 00: 8E: B8: 83: 19: 20
            inet addr: 172.16.3.1 Bcast: 172.16.3.255 Mask: 255.255.255.0
            UP BROADCAST RUNNING PROMISC MULTICAST MTU: 1500 Metric: 1
            RX packets: 334036 errors: 0 dropped: 0 overruns: 0
            TX packets: 11605 errors: 0 dropped: 0 overruns: 0
            Interrupt: 7 Base address: 0x378

eth0: 0     Link encap: 10Mbps Ethernet HWaddr 00: 8E: B8: 83: 19: 20
            inet addr: 172.16.3.10 Bcast: 172.16.3.255 Mask: 255.255.255.0
            UP BROADCAST RUNNING MTU: 1500 Metric: 1
            RX packets: 0 errors: 0 dropped: 0 overruns: 0
            TX packets: 0 errors: 0 dropped: 0 overruns: 0

eth0: 1     Link encap: 10Mbps Ethernet HWaddr 00: 8E: B8: 83: 19: 20
            inet addr: 172.16.3.100 Bcast: 172.16.3.255 Mask: 255.255.255.0
            UP BROADCAST RUNNING MTU: 1500 Metric: 1
            RX packets: 1 errors: 0 dropped: 0 overruns: 0
            TX packets: 0 errors: 0 dropped: 0 overruns: 0

```

`/proc/net/aliases` 文件的内容是：

device	family	address
eth0: 0	2	172.16.3.10
eth0: 1	2	172.16.3.100

`/proc/net/alias_types` 文件的内容是：

type	name	n_attach
2	ip	2

当然，`/proc/net` 文件的内容是由 `ifconfig` 命令产生，而非用手写的。

4.1.2 在重新启动机器后保持设置

不论用 BSD 形式或 Sys V 形式(例如 RedHat)的 init 启动, 用户都可以把设置 IP 别名的命令放在 `/etc/rc.d/rc.local` 文件中。

下面是 Sys Vinit 系统(RedHat 3.0.3 及 4.0 版本)中的设置。

`/etc/rc.d/rc.local` 的内容(修改过只显示相关的部分)如下所示:

```
#setting up IP alias interfaces
echo "Setting 172.16.3.1, 172.16.3.10, 172.16.3.100 IP Aliases ..."
/sbin/ifconfig lo 127.0.0.1
/sbin/ifconfig eth0 up
/sbin/ifconfig eth0 172.16.3.1
/sbin/ifconfig eth0: 0 172.16.3.10
/sbin/ifconfig eth0: 1 172.16.3.100
#
#setting up the routes
echo "Setting IP routes ..."
/sbin/route add -net 127.0.0.0
/sbin/route add -net 172.16.3.0 dev eth0
/sbin/route add -host 172.16.3.1 eth0
/sbin/route add -host 172.16.3.10 eth0: 0
/sbin/route add -host 172.16.3.100 eth0: 1
/sbin/route add default gw 172.16.3.200
#
```

4.1.3 IP 别名与电子邮件问题

下面介绍在 IP 别名的机器上在不同的 IP 别名地址上接收电子邮件的方法。

建立一个文件(如果还没有), 如 `/etc/mynames.cw`。该文件名及存放目录由用户自定。

在该文件中, 放上别名 IP 地址的完整域名称。如果别名的 IP 地址没有域名称, 可以用这个 IP 地址本身。

```
/etc/mynames.cw:
-----
# /etc/mynames.cw - include all aliases for your machine here; # is a comment.
domain.one.net
domain.two.com
domain.three.org
4.5.6.7
```

在 `sendmail.cf` 文件中, 在定义文件类别 Fw 处, 加上下面的内容:

```
.
.
```

```

.
#####
#  local info  #
#####
.
.
# file containing names of hosts for which we receive email
Fw/etc/mynames.cw
.
.
.

```

这样应该就可以了。

以测试模式执行 sendmail 以测试新的设置：

```

ganymede$ /usr/lib/sendmail -bt
ADDRESS TEST MODE (ruleset 3 NOT automatically invoked)
Enter < ruleset> < address>
> 0 me@4.5.6.7
rewrite: ruleset 0  input: me @ 4 . 5 . 6 . 7
rewrite: ruleset 98  input: me @ 4 . 5 . 6 . 7
rewrite: ruleset 98 returns: me @ 4 . 5 . 6 . 7
rewrite: ruleset 97  input: me @ 4 . 5 . 6 . 7
rewrite: ruleset 3   input: me @ 4 . 5 . 6 . 7
rewrite: ruleset 96  input: me < @ 4 . 5 . 6 . 7 >
rewrite: ruleset 96 returns: me < @ 4 . 5 . 6 . 7 . >
rewrite: ruleset 3 returns: me < @ 4 . 5 . 6 . 7 . >
rewrite: ruleset 0   input: me < @ 4 . 5 . 6 . 7 . >
rewrite: ruleset 98  input: me < @ 4 . 5 . 6 . 7 . >
rewrite: ruleset 98 returns: me < @ 4 . 5 . 6 . 7 . >
rewrite: ruleset 0 returns: $# local $: me
rewrite: ruleset 97 returns: $# local $: me
rewrite: ruleset 0 returns: $# local $: me
> 0 me@4.5.6.8
rewrite: ruleset 0  input: me @ 4 . 5 . 6 . 8
rewrite: ruleset 98  input: me @ 4 . 5 . 6 . 8
rewrite: ruleset 98 returns: me @ 4 . 5 . 6 . 8
rewrite: ruleset 97  input: me @ 4 . 5 . 6 . 8
rewrite: ruleset 3   input: me @ 4 . 5 . 6 . 8
rewrite: ruleset 96  input: me < @ 4 . 5 . 6 . 8 >
rewrite: ruleset 96 returns: me < @ 4 . 5 . 6 . 8 >
rewrite: ruleset 3 returns: me < @ 4 . 5 . 6 . 8 >
rewrite: ruleset 0   input: me < @ 4 . 5 . 6 . 8 >
rewrite: ruleset 98  input: me < @ 4 . 5 . 6 . 8 >
rewrite: ruleset 98 returns: me < @ 4 . 5 . 6 . 8 >

```

```
rewrite: ruleset 95 input: < > me < @ 4 . 5 . 6 . 8 >
rewrite: ruleset 95 returns: me < @ 4 . 5 . 6 . 8 >
rewrite: ruleset 0 returns: $# smtp $@ 4 . 5 . 6 . 8 $: me < @ 4 . 5 . 6 . 8 >
rewrite: ruleset 97 returns: $# smtp $@ 4 . 5 . 6 . 8 $: me < @ 4 . 5 . 6 . 8 >
rewrite: ruleset 0 returns: $# smtp $@ 4 . 5 . 6 . 8 $: me < @ 4 . 5 . 6 . 8 >
>
```

这样,就把 Linux 系统下的 IP 别名设置好了。值得一提的是,当前许多虚拟主机的实现都是基于 IP 别名的。

4.2 Linux 主机的 IP Masquerade 功能

本节将描述如何在一台 Linux 主机上启动 IP Masquerade(伪装)功能,允许没有注册因特网 IP 地址的连接计算机通过用户的 Linux 机器连接因特网。用户的机器可能以以太网网络连接 Linux,也可能是其他种类,如拨号的点对点(ppp)连接。本节将着重介绍以太网网络连接的情况,因为这是最常见的情况。

4.2.1 IP Masquerade 简介

IP Masquerade 是 Linux 系统中的一种网络功能。如果一台 Linux 主机使用 IP Masquerade 功能连接到因特网上,那么接上它的计算机(不论是在同一个局域网络上还是通过调制解调器连接)即使没有获得正式指定的 IP 地址,也可以访问因特网。

这使得一些计算机可以隐藏在网关系统后面存取因特网而不被发现,看起来就像只有这个系统在使用因特网。突破设置良好的伪装系统的安全防护比突破良好的数据包过滤式防火墙更加困难。

4.2.2 IP Masquerade 现状

IP Masquerade 仍然在实验阶段。内核从 1.3.x 开始已经内建这项支持。许多个人甚至公司正在使用它,且有满意的结果。

文件传输(FTP),网络交谈(IRC)等现在可以装入某些模块配合使用。其他的网络数据流音频如 True Speech 和 Internet Wave 等,也能运行。一些邮件列表中的成员甚至还尝试过视频会议软件。目前 ping 命令配合新近可以取得的因特网控制信息协议(ICMP)补丁程序也能运行。

IP Masquerade 在多种不同的操作系统及平台上与客户端机器配合良好。成功的例子有使用 UNIX、Windows 95、Windows NT、Windows for Workgroup(带有 TCP/IP 包)、OS/2、Mac TCP 的 Macintosh OS、VAX 和 Linux 的 Alpha,甚至带有 AmiTCP 或 AS225-stack 的

Amiga 系统。

4.2.3 从 IP Masquerade 中获益

如果用户有一台连接因特网的 Linux 主机，而且有一些通过 TCP/IP 连接到 Linux 机器的计算机在局域网络上，那么该用户将从 IP Masquerade 中获益。

4.2.4 不需要 IP Masquerade 的用户

如果用户的机器是单独一台连接因特网的 Linux 主机，那么使用 IP Masquerade 没什么意义；如果用户的其他机器拥有正式指定的 IP 地址，那么用户就不需要 IP Masquerade；如果用户不喜欢免费使用因特网，当然也不需要 IP Masquerade。

4.2.5 IP Masquerade 的运行

简单的设置草图如下所示：

```

SLIP/PPP      +-----+          +-----+
to provider    | Linux |    SLIP/PPP    | Anybox |
<----- modem1|      |modem2 ----- modem |      |
111.222.333.444 |      | 192.168.1.100 |      |
               +-----+          +-----+

```

上面的草图中一台安装并执行 `ip_masquerading` 的 Linux 机器使用 `modem1` 通过 SLIP/PPP 连接因特网。它有一个指定的 IP 地址 111.222.333.444。它设置 `modem2` 允许拨号者登录并启动 SLIP 或 PPP 连接。

第二个系统(不必是运行 Linux 的系统)拨号进入 Linux 机器并启动 SLIP 或 PPP 连接。它在因特网上没有指定的 IP 地址，所以它使用 192.168.1.100。

典型的例子如下所示：

```

+-----+
|      | Ethernet
| abox  | : : : : :
|      | 2      : 192.168.1.x
+-----+
:      :
:      : +-----+ PPP
+-----+      : 1| Linux | link
|      |      : : : : | masq-gate| : : : : : // Internet
| bbox  | : : : : :
|      | 3      : +-----+

```

```

+-----+      :
      :
+-----+      :
|       |      :
| cbox   | : : : :
|       | 4
+-----+

```

<-Internal Network->

在这个例子中有 4 台计算机系统。

这个 Linux 系统 masq-gate 是 abox、bbox 和 cbox 内部网络机器连接因特网的伪装网关。内部网络使用指定的私用网络地址,在这个例子中是 C 类地址网络 192.168.1.0, Linux 机器拥有地址 192.168.1.1, 其他系统也拥有此地址。

这三台机器 abox、bbox 和 cbox(它们可以执行任何操作系统,如 Windows 95 或 Macintosh MacTCP,甚至是另一台 Linux 机器)可以连接到因特网上的其他机器,而伪装系统网关 masq-gate 转换它们所有的连接,所以这些连接看起来像原本即从伪装网关 masq-gate 本身发出的,而且还安排伪装连接传回的数据转回原先的系统——所以在内部网络上的系统看到的是直接通往因特网的递送路径而且不知道它们的数据被伪装过。

4.2.6 在 Linux 2.x 上使用 IP Masquerade

内核 2.0.x 的程序源代码可以从以下站点取得:

<ftp://ftp.funet.fi/pub/Linux/kernel/src/v2.0/>

用户需加入一些支持来编译用户的内核,建议使用最新的正式版本。

可装入内核模块,最好是 2.0.0 或更新的版本(至少需要 modules-1.3.57),可以从以下站点取得:

<http://www.pi.se/blox/modules/modules-2.0.0.tar.gz>

Ipfwadm 2.3 或更新的版本可以从以下站点取得:

<ftp://ftp.xos.nl/pub/Linux/ipfwadm/ipfwadm-2.3.tar.gz>

在 Linux Ipfwadm 网页上有更多的关于版本的信息。用户可以有选择地加上一些 IP Masquerade 补丁程序以增加其他功能。

4.2.7 IP Masquerade 的设置

如果用户的私用网络里有重要的信息,在使用 IP Masquerade 之前请三思。这能成为

用户通往因特网的网关，也能成为外界进入用户私用网络的途径。

4.2.7.1 编译内核加入 IP Masquerade 的支持

首先，用户需要内核的程序源代码(最好是稳定的 2.0.0 版或以上的内核)。

使用如下命令：

```
tar xvzf Linux-2.0.x.tar.gz -C /usr/src
```

把内核的程序源代码解至 /usr/src/ 目录。

以下是用户要编译的选项。

下列选项要回答 YES(Y)：

```
Prompt for development and/or incomplete code/drivers
CONFIG_EXPERIMENTAL
- 这将让用户能选择把实验性的 IP Masquerade 程序代码编译到内核中
Enable loadable module support
CONFIG_MODULES
- 让用户能够装入模块
Networking support
CONFIG_NET
Network firewalls
CONFIG_FIREWALL
TCP/IP networking
CONFIG_INET
IP: forwarding/gatewaying
CONFIG_IP_FORWARD
IP: firewalling
CONFIG_IP_FIREWALL
IP: masquerading (EXPERIMENTAL)
CONFIG_IP_MASQUERADE
- 这虽然是实验性的，但却是“必须”的
IP: ipautofw masquerade support (EXPERIMENTAL)
CONFIG_IP_MASQUERADE_IPAUTOFW
- 建议使用
IP: ICMP masquerading
CONFIG_IP_MASQUERADE_ICMP
- 支持 ICMP 数据包伪装，可选用
IP: always defragment
CONFIG_IP_ALWAYS_DEFRAG
- 强烈建议使用
Dummy net driver support
CONFIG_DUMMY
- 建议使用
```



注意： 以上这些只是 IP Masquerade 所需要的，用户还需选择其他的设置需要的选项。

编译内核之后，用户应该编译并安装模块：

```
make modules modules_install
```

然后用户应该在/etc/rc.d/rc.local(或其他用户认为合适的文件)文件中加上几行以便每次启动时自动装入/lib/modules/2.0.x/ipv4/所需的模块：

```
...
/sbin/depmod -a
/sbin/modprobe ip_masq_ftp
/sbin/modprobe ip_masq_raudio
/sbin/modprobe ip_masq_irc
...
```

以及其他如 ip_masq_cuseeme 和 ip_masq_vdolive 等模块。

4.2.7.2 指定私用网络的IP地址

因为其他所有机器都没有正式指定的地址，所以必须以正确的方式给这些机器分配地址。

因特网地址指定局(IANA, Internet Assigned Numbers Authority)已经保留下列三个区域的 IP 地址空间给私用网络：

10.0.0.0	-	10.255.255.255
172.16.0.0	-	172.31.255.255
192.168.0.0	-	192.168.255.255

第一个区域称为“24 位区域”，第二个为“20 位区域”，而第三个则称为“16 位区域”。

注意到第一个区域是 A 类地址网络号码，第二个区域则是 16 个连续的 B 类地址网络号码，而第三个区是一组 255 个连续的 C 类地址网络号码。

所以，如果用户要使用一个 C 类网络地址，那么用户的机器应该以 192.168.1.1、192.168.1.2、192.168.1.3、…、192.168.1.x 来命名。

192.168.1.1 通常是网关，在这里就是用户连上因特网的 Linux 主机。

注意，192.168.1.0 和 192.168.1.255 分别为网络和广播地址，是保留的。用户应该避免在机器上使用这些地址。

4.2.7.3 配置其他机器

除了为每台机器设置适当的 IP 地址之外，用户还应该设置适当的网关。一般说来，这非常简单，只需输入 Linux 主机的地址(通常是 192.168.1.1)作为网关地址即可。

关于域名服务，用户可以加入任何 DNS(域名系统)系统。最好是加入 Linux 使用的那个。用户也可以有选择地加上任意网域字尾。

在重新配置这些 IP 地址后，要重新启动适当的服务或是重新启动系统。

下例假设用户使用一个 C 类网络地址并且以 192.168.1.1 作为 Linux 主机的地址。请注意 192.168.1.0 和 192.168.1.255 是保留的。

1. 配置 Windows 95

- 1) 如果用户还没有安装网卡以及网卡驱动程序, 现在就安装。
- 2) 选择【控制面板】|【网络】选项。
- 3) 如果用户的网络配置里没有 TCP/IP 协议, 则添加。
- 4) 在【TCP/IP 属性】对话框中, 选择【IP 地址】选项卡, 把 IP 地址设置为“192.168.1.x”(1<x<255), 把子网掩码设为“255.255.255.0”。
- 5) 在网关中加入“192.168.1.x”作为用户的网关。
- 6) 在【DNS 配置/DNS 服务器】文字输入框中加入用户的 Linux 主机使用的 DNS(通常可以在/etc/resolv.conf 文件中可以找到)。用户可以有选择地加入适当的网域字尾搜寻顺序。
- 7) 不要改变原来的其他设置, 除非用户胸有成竹。
- 8) 在所有的对话框中按下【确定】按钮, 然后重新启动系统。
- 9) 测试网络连接, 用 ping 命令连接用户的 Linux 主机: 单击【开始】|【运行】选项, 在【运行】对话框中输入“ping 192.168.1.1”。
- 10) 用户可以在 Windows 目录下有选择地建立一个 HOSTS 文件, 用户这样就可以使用局域网络里的机器名称。在 Windows 目录里有一个 HOSTS.SAM 文件, 可以作为例子参考。

2. 配置 Windows for Workgroup 3.11

- 1) 如果用户还没有安装网卡以及界面驱动程序, 现在就安装。
- 2) 如果用户还未安装 TCP/IP 32b 软件包, 现在就安装。
- 3) 选中 Network Drivers 选项组里的 Microsoft TCP/IP-32 3.11b 选项, 然后单击 Setup 按钮。
- 4) 设置 IP 地址为“192.168.1.x”(1<x<255), 然后设置 Subnet Mask 为“255.255.255.0”, Default Gateway 为“192.168.1.1”。
- 5) 不要打开 Automatic DHCP Configuration 选项, 并在 WINS Server 文字输入框中放入任何内容, 除非用户在 Windows NT 域中, 而且很有把握。
- 6) 单击 DNS 按钮, 填入在“配置 Windows 95”中步骤 6 提到的信息, 完成后单击 OK 按钮。
- 7) 单击 Advanced 按钮, 如果使用类似于“配置 Windows 95”步骤 10 中提到的主机文件, 选中 Enable DNS for Windows Name Resolution 和 Enable LMHOSTS lookup 选项。
- 8) 在所有对话框中, 单击 OK 按钮, 然后重新启动系统。
- 9) 通过 ping 命令连接用户的 Linux 主机以测试网络连接, 在 File/Run 文字输入框中输入: “ping 192.168.1.1”。

3. 配置 Windows NT

- 1) 如果用户还没有安装网卡以及界面驱动程序, 现在就安装。
- 2) 选择 Main[Control Panel]Network 选项。
- 3) 如果用户还没要装 TCP/IP 服务, 从 Add Software 菜单中加入 TCP/IP 协议及相关的部分。
- 4) 选择 Network Software and Adapter Cards 选项组里的 Installed Network Software

选项框中的【TCP/IP 协议】选项。

5) 在 TCP/IP Configuration 选项组中, 选择适当的接口驱动程序; 例如, 选择[1]Novell NE2000Adapter。然后设置 IP 地址为“192.168.1.x”(1<x<255), 再设置 Subnet Mask 为“255.255.255.0”, Default Gateway 为“192.168.1.1”。

6) 不要打开 Automatic DHCP Configuration 选项并在 WINS Server 文字输入框中放入任何内容, 除非用户在 Windows NT 域中而且头脑清醒。

7) 单击 DNS 按钮, 填入在“配置 Windows 95”中步骤 6 提到的信息, 完成后单击 OK 按钮。

8) 单击 Advanced 按钮, 如果用户使用类似“配置 Windows 95”步骤 10 中提到的主机文件, 选中 Enable DNS for Windows Name Resolution 和 Enable LMHOSTS lookup 选项。

9) 在所有对话框中, 单击 OK 按钮, 然后重新启动系统。

10) 用 ping 命令直接用户的 Linux 主机以测试网络连接: 在 File/Run 文字输入框输入: “ping 192.168.1.1”。

4. 配置 UNIX 系列的系统

1) 如果用户还未安装网卡并以适当的界面驱动程序重新编译内核, 现在就做吧。

2) 如果还未安装, 则安装 TCP/IP 网络, 如 nettools 软件包。

3) 将 IPADDR 设为“192.168.1.x”(1<x<255), 然后将 NETMASK 设为“255.255.255.0”, GATEWAY 设为“192.168.1.1”, BROADCAST 设为“192.168.1.255”。

例如, 在 Linux 系统的 RedHat 版本上用户可以编辑/etc/sysconfig/network-scripts/ifcfg-eth0, 或从【控制面板】的【软件包配置】里设置。

4) 将用户的名称服务器及域搜寻字尾加到/etc/resolv.conf 中。

5) 依据用户的设置, 可能要更新/etc/networks 文件。

6) 重新启动适当的服务。

7) 发出 ping 命令: “ping 192.168.1.1”以测试网关的连接性。

5. 配置使用 NCSA telnet 软件包的 DOS 机器

1) 如果用户还没有安装网卡, 现在就安装。

2) 装入适当的数据包驱动程序。对于 NE2000 卡来说, 如果用户的卡设置为 IRQ 10 及硬件地址为 0x300, 则用 nwpd 0x60 10 0x300。

3) 建立一新目录, 然后解开 NCSA telnet 软件包: pkunzip tel2308b.zip

4) 使用文字编辑器打开 config.tel 文件。

5) 设置 myip=192.168.1.x(1<x<255), netmask=255.255.255.0。

6) 在本例中, 应该设置 hardware=packet, interrupt=10, ioaddr=60。

7) 至少要有一台独立的机器设置为网关, 也就是 Linux 主机:

```
name=default
host=yourLinuxhostname
hostip=192.168.1.1
gateway=1
```

8) 此外还要指定一个域名服务器:

```
name=dns.domain.com : hostip=123.123.123.123: nameserver=1
```



注意： 用 Linux 主机使用的 DNS 的适当信息来取代上述一行。

9) 保存用户的 config.tel 文件。

10) 通过 Telnet 命令连接 Linux 主机以测试网络连接：“telnet 192.168.1.1”。

6. 配置执行 MacTCP 的 MacOS 机器

1) 如果还没有为以太网适配器安装适当的驱动程序，最好现在就安装。

2) 打开 MacTCP Control Panel，选择适当的网络驱动程序(Ethernet，而不是 EtherTalk)然后单击 More 按钮。

3) 在 Obtain Address 选项组中，单击 Manually 单选按钮。

4) 在 IP Address 选项下，从弹出菜单中选择 Class C，忽略对话框中的其他部分。

5) 在 Domain Name Server Information 文字输入框中输入适当信息。

6) 在 Gateway Address 文字输入框中，输入“192.168.1.1”。

7) 单击 OK 按钮，保存设置。在 MacTCP Control Panel 的主窗口中，在 IP Address 文字输入框中输入 Mac 的 IP 地址(192.168.1.x, 1<x<255)。

8) 关闭 MacTCP Control Panel 窗口。如果弹出窗口提醒用户重新启动，那就重新启动。

9) 用户可以通过 ping 命令连接 Linux 主机来测试网络连接。如果有 MacTCP Watcher 免费程序，单击 Ping 按钮，然后在弹出的对话框中键入用户的 Linux 主机的地址(192.168.1.1)。

10) 用户有可选择地在 System Folder 中建立 HOSTS 文件以便可以使用局域网络里机器的主机名称。

7. 配置执行 Open Transport 的 MacOS 系统

1) 如果还没有为以太网适配器安装适当的驱动程序，最好现在就安装。

2) 打开 TCP/IP Control Panel 窗口，然后从 Edit 菜单中选择 User Mode 选项，确定使用者模式至少是 Advanced，然后单击 OK 按钮。

3) 从 File 菜单中选择 Configurations 选项。选择 Default 选项并单击 Duplicate 按钮。在 Duplicate Configuration 对话框中输入“IP Masq(或是其他能让用户知道这是个特殊配置的字)”。再单击 OK 按钮，然后单击 Make Active 按钮。

4) 从 Connect via 弹出菜单中选择 Ethernet 选项。

5) 从 Configure 弹出菜单选择适当的选项。如果不知道应该选什么，可以重新选择 Default 配置，然后退出。

6) 在 IP Address 文字输入框中输入 Mac 的 IP 地址(192.168.1.x, 1<x<255)。

7) 在 Subnet mask 文字输入框中输入“255.255.255.0”。

8) 在 Router address 文字输入框中输入“192.168.1.1”。

9) 在 Name server addr 文字输入框中输入域名服务器的 IP 地址。

10) 在 Implicit Search Path 文字输入框里的 Starting domain name 文字输入框输入用户的因特网域名(例如：输入“microsoft.com”)。

11) 接下来的步骤是选择性的。不正确的值可能导致严重的错误。如果用户不确定，

最好留下空白,不要选。目前,不能在 TCP/IP 对话框中设置系统不使用以前选过的别的 HOSTS 文件。如果用户的网络需要 802.3 框架则选中 802.3。

12) 按下 Options 按钮以确定 TCP/IP 有作用。使用 Load only when needed 选项。如果多次执行并结束 TCP/IP 应用程序而未重新启动机器,用户将发现不选 Load only when needed 选项会抑制或降低机器的内存管理效率。不选此选项将使 TCP/IP 协议总是被装入以便于使用。如果选中了, TCP/IP 协议会自动在需要时装入并在不需要时释放。装入与释放的过程可能使机器的内存变得不连续。

13) 用户可以通过 ping 命令连接 Linux 主机来测试网络连接。如果用户有 MacTCP Watcher 免费程序,按下 Ping 按钮,然后在弹出的对话框中键入 Linux 主机的地址 (192.168.1.1)。

14) 可以在 System Folder 中建立一个 HOSTS 文件以使用户可以使用局域网络里机器的主机名称。如果还没有这个文件,用户可以从一个正在运行 MacTCP 的系统中取回,或自己建一个(它遵循 UNIX 的/etc/hosts 文件格式,在 RFC 1035 中描述)。用户建立了这个文件后,就可以打开 TCP/IP Control Panel 窗口,单击 Select Hosts File 按钮,然后打开 HOSTS 文件。

15) 关闭对话框或从 File 菜单中选择 Close 或 Quit 选项然后单击 Save 按钮以保存用户所做的改变。这些改变会立刻生效。

8. 配置使用 DNS 的 Novell 网络

- 1) 如果用户还没有为以太网适配器安装适当的驱动程序,最好现在就安装。
- 2) 从 URL 网站 ftp.novell.com/pub/updates/Unixconn/lwp5 中取回 tcpip16.exe。
- 3) 编辑 c:\nwclient\startnet.bat

下面是该文件的一个例子:

```
SET NWLANGUAGE=ENGLISH
LH LSL.COM
LH KTC2000.COM
LH IPXODI.COM
LH tcpip
LH VLM.EXE
F:
```

- 4) 编辑 c:\nwclient\net.cfg(将驱动程序改为用户的,如 NE2000):

```
Link Driver KTC2000
  Protocol IPX 0 ETHERNET_802.3
  Frame ETHERNET_802.3
  Frame Ethernet_II
  F 物理内存 E Ethernet_802.2

NetWareDOS Requester
  FIRST NETWORK DRIVE = F
  USE DEFAULTS = OFF
  VLM = CONN.VLM
  VLM = IPXNCP.VLM
```

```

VLM = TRAN.VLM
VLM = SECURITY.VLM
VLM = NDS.VLM
VLM = BIND.VLM
VLM = NWP.VLM
VLM = FIO.VLM
VLM = GENERAL.VLM
VLM = REDIR.VLM
VLM = PRINT.VLM
VLM = NETX.VLM

```

Link Support

```

Buffers 8 1500
MemPool 4096

```

Protocol TCPIP

```

PATH SCRIPT      C: \NET\SCRIPT
PATH PROFILE     C: \NET\PROFILE
PATH LWP_CFG     C: \NET\HSTACC
PATH TCP_CFG     C: \NET\TCP
ip_address       xxx.xxx.xxx.xxx
ip_router        xxx.xxx.xxx.xxx

```

5) 最后建立 c: \bin\resolv.cfg.

```

SEARCH DNS HOSTS SEQUENTIAL
NAMESERVER 207.103.0.2
NAMESERVER 207.103.11.9

```

9. 配置 OS/2 Warp

- 1) 如果用户还没有为以太网适配器安装适当的驱动程序，最好现在就安装。
- 2) 如果用户还没有安装 TCP/IP 通信协议，现在就装。
- 3) 选择 Programs/TCP/IP(LAN)/TCP/IP 设置。
- 4) 在 Network 栏加上用户的 TCP/IP 地址并设置用户的子网掩码(255.255.255.0)。
- 5) 在 Routing 栏按下 Add 按钮。将 Type 栏设置为 default 并在 Router Address 栏中键入 Linux 主机的 IP 地址(192.168.1.1)。
- 6) 将 Hosts 栏设置为与 Linux 主机使用相同的 DNS 地址。
- 7) 关闭 TCP/IP 控制台。在接下来的问题中回答 yes。
- 8) 重新启动系统。
- 9) 用户可以通过 ping 命令连接用户的 Linux 主机以测试网络配置。在 OS/2 文字输入框中输入“ping 192.168.1.1”。如果收到 ping 数据包则一切就都没问题了。

4.2.8 配置 IP 转发的方式

至此，用户已经安装好内核以及其他需要的软件包，也装入了模块。同时，其他机器的 IP 地址、网关和 DNS 也全都设置完成。

剩下的事是使用 `ipfwadm` 程序转发适当的数据包给相应的机器。这可以用多种不同的方法来做。

下面举例说明其中的一种方法：

```
ipfwadm -F -p deny
ipfwadm -F -am -S yyy.yyy.yyy.yyy/x -D 0.0.0.0/0
```

其中 `x` 由用户的子网而定，为下列数字之一，而 `yyy.yyy.yyy.yyy` 则是用户的网络地址。

netmask	x	Subnet
255.0.0.0	8	Class A
255.255.0.0	16	Class B
255.255.255.0	24	Class C
255.255.255.255	32	Point-to-point

例如，如果是在一个 C 类网络地址上，需要输入：

```
ipfwadm -F -p deny
ipfwadm -F -a m -S 192.168.1.0/24 -D 0.0.0.0/0
```

因为 `bootp` 请求数据包没有合法的 IP 地址，客户端并不知道它的地址，对于在伪装/防火墙上执行 `bootp` 服务器的用户必须在拒绝服务之前执行下列命令：

```
ipfwadm -I -a accept -S 0/0 68-D 0/0 67 -W bootp_clients_net_if_name -P udp
```

用户也可以分别对每台机器进行设置。例如，如果想让 192.168.1.2 和 192.168.1.8 能够存取因特网，但不允许其他机器使用，需要输入：

```
ipfwadm -F -p deny
ipfwadm -F -a m -S 192.168.1.2/32 -D 0.0.0.0/0
ipfwadm -F -a m -S 192.168.1.8/32 -D 0.0.0.0/0
```

另外，用户可以输入子网掩码以取代该值，例如 192.168.1.0/255.255.255.0

常见的错误命令如下：

```
ipfwadm -F -p masquerade
```

不要把用户的默认方式定为 `masquerade`，否则可以操纵传送路由的人将能够直接穿过用户的网关，以此伪装他们的身份！

4.2.9 测试 IP Masquerade

这些配置工作完成后，下面将进行测试。

首先，确定用户的 Linux 主机到因特网的连接是通的。

用户可以试着在其他机器上浏览一些因特网上的网页，看是否能见到。建议第一次尝试时使用 IP 地址而不要用主机名称，因为用户的 DNS 设置有可能不正确。

例如，用户可以使用地址 `http://152.2.254.81/mdw/Linux.html` 来访问 Linux 文件计划网页 `http://sunsite.unc.edu/mdw/Linux.html`。

如果用户看见漂亮的帆船，那么表示成功了。

接着用户可以使用主机名称试试看，然后用 Telnet、FTP、RealAudio、TrueSpeech 以及其他 IP Masquerade 支持的内容。

4.2.10 IP Masquerade 的问题

某些协议现在无法配合 masquerading 使用，对这些协议，需要在 masquerade 程序中加入特定的代理服务程序才能使之有效。

4.2.10.1 进入系统的服务

masquerading 完全不能处理外界的服务请求。

如果并不要求高度的安全性，那么可以重新指定，从而解决这个问题。如使用一个修改过的 redir 程序(该程序能从 Sunsite 及其镜像站点取得)。

如果用户希望能够对外界进入系统的服务请求有某种程度的身份认证，那么用户可以在 redir 的顶层(0.7 或更高)使用 TCP_Wrapper 或 Xinetd 来允许特定 IP 地址通过，或使用其他的工具。

4.2.10.2 已支持的客户端软件以及设置方面的其他注意事项

一般说来，使用传输控制协议(TCP)或是用户数据报协议(UDP)的应用程序应该都能运行。

1. 可以使用的客户端软件

● 一般客户端软件

HTTP 应用于所有支持它的平台，浏览网页。

POP & SMTP 应用于所有支持它的平台，电子邮件软件。

Telnet 应用于所有支持它的平台，远端登录程序。

FTP 应用于所有支持它的平台，配合 `ip_masq_ftp.o` 模块(不是所有都能配合各种客户端软件；例如某些不能使用 `ws_ftp32` 连接的却可以使用 Netscape 进入)。

Archie 应用于所有支持它的平台，文件搜索软件(并非所有 Archie 客户端软件都支持)。

NNTP(USENET) 应用于所有支持它的平台，网络新闻软件。

VRML 应用于 Windows (可能所有支持它的平台都可以), 虚拟现实浏览。

traceroute 主要应用于 UNIX 系列的平台, 某些变种可能无法运行。

ping 应用于所有平台, 配合 ICMP 补丁程序来使用。

anything based on IRC 应用于所有支持它的平台, 配合 ip_masq_irc.o 模块来使用。

gopher client 应用于所有支持它的平台。

WAIS client 应用于所有支持它的平台。

- 多媒体客户端软件

Real Audio Player 应用于 Windows, 声音播放器, 用户还需装入 ip_masq_audio 模块。

True Speech Player 1.1b 应用于 Windows, 网络数据流声音播放器。

Internet Wave Player 应用于 Windows, 网络数据流声音播放器。

Worlds Chat 0.9a 应用于 Windows, 客户/服务器端立体交谈(3D chat)程序。

Alpha Worlds 应用于 Windows, 客户/服务器端立体交谈程序。

Powwow 应用于 Windows, 点对点文字声音白板通信。

CU-SeeMe 应用于所有支持它的平台, 需要装入 cuseeme 模块, 详细信息请参阅 IP Masquerade Resource16。

VDOLive 应用于 Windows, 用户还需装入 vdolive 补丁程序。

- 其他客户端软件

NCSA telnet 2.3.08 应用于 DOS, 包含 Telnet、FTP 和 ping 等软件软件包。

PC-anywhere for Windows 2.0 应用于 MS-Windows, 通过 TCP/IP 远端遥控 PC, 只有在作为客户端而非主机端的情形下才能运行。

Socket Watch 使用网络时间协议(NTP)。

Linux net-acct package 应用于 Linux 平台, 网络帐号管理软件包。

2. 无法使用的客户端软件

Intel Internet Phone Beta 2 可以连上但声音只能单向(往外)传送。

Intel Streaming Media Viewer Beta 1 无法连上服务器。

Netscape CoolTalk 无法连接对方。

Talk 和 ntalk 不能运行, 需要编写一个内核代理程序。

WebPhone 目前无法运行(它做了不合法的地址假设)。

4. 2. 10. 3 已测试过可以作为其他机器的平台/操作系统

Linux

Solaris

Windows 95

Windows NT(workstation 和 server)

Windows For Workgroup 3.11(带有 TCP/IP 包)

Windows 3.1(带有 Chameleon 包)

Novel 4.01 Server

OS/2(包括 Warp v3)

Macintosh OS 带有(MacTCP 或 Open Transport)
 DOS(带有 NCSA telnet 包, 部分带有 DOS Trumpet)
 Amiga (带有 AmiTCP 或 AS225-stack)
 带 UCX 的 VAX Stations 3520 和 3100 (用于 VMS 的 TCP/IP 栈)
 Alpha/AXP with Linux/Redhat
 SCO Openserver(版本 3.2.4.2 和 5)
 运行 AIX 的 IBM RS/6000

4.2.11 IP Firewall Administration(ipfwadm) ,

本节提供关于 ipfwadm 的更深入的使用指导。

下面是一个 Shell 脚本, 名称是/etc/rc.d/rc.firewall, 使用这个名称的原因是它可以在系统启动时自动执行。这个脚本将在 IP 地址为 192.168.255.1 的 Linux 主机上配置防火墙, 它使用的命令主要是 ipfwadm:

```
#!/bin/sh
#
# /etc/rc.d/rc.firewall, 定义防火墙配置, 从 rc.local 执行.
#

PATH=/sbin: /bin: /usr/sbin: /usr/bin

# 测试用, 等待一段时间然后清除所有的防火墙规则.
# 如果用户希望防火墙 10 分钟之后自动关闭就取消下列几行的注释.
# (sleep 600; \
# ipfwadm -I -f; \
# ipfwadm -I -p accept; \
# ipfwadm -O -f; \
# ipfwadm -O -p accept; \
# ipfwadm -F -f; \
# ipfwadm -F -p accept; \
# ) &

# 进入伪装网关的设置, 更新以及设置拒绝的策略.

ipfwadm -I -f
ipfwadm -I -p deny

# 伪装网关的本地界面, 局域网络里的机器, 允许连往任何.
# 地方.

ipfwadm -I -a accept -V 192.168.255.1 -S 192.168.0.0/16 -D 0.0.0.0/0

# 伪装网关的远程界面, 声称是局域网络里的机器。
```

```
ipfwadm -I -a deny -V your.static.PPP.address -S 192.168.0.0/16 -D  
0.0.0.0/0 -o
```

伪装网关的远程界面，无论来自哪个 IP 地址的请求，都允许送往固定 PPP。
地址。

```
ipfwadm -I -a accept -V your.static.PPP.address -S 0.0.0.0/0 -D  
your.static.PPP.address/32
```

回送界面是允许的。

```
ipfwadm -I -a accept -V 127.0.0.1 -S 0.0.0.0/0 -D 0.0.0.0/0
```

捕捉所有规则，其他任何的进入方式都会被拒绝并记录。

```
ipfwadm -I -a deny -S 0.0.0.0/0 -D 0.0.0.0/0 -o
```

送出伪装网关的设置，更新以及设置拒绝的策略。

```
ipfwadm -O -f  
ipfwadm -O -p deny
```

本地界面，允许来自任何 IP 地址的请求送出至局域网络。

```
ipfwadm -O -a accept -V 192.168.255.1 -S 0.0.0.0/0 -D 192.168.0.0/16
```

远程界面送出至局域网络。

```
ipfwadm -O -a deny -V your.static.PPP.address -S 0.0.0.0/0 -D 192.168.0.0/16 -o
```

局域网络的机器从远程界面送出。

```
ipfwadm -O -a deny -V your.static.PPP.address -S 192.168.0.0/16 -D 0.0.0.0/0 -o
```

局域网络的机器从远程界面送出。

```
ipfwadm -O -a deny -V your.static.PPP.address -S 0.0.0.0/0 -D 192.168.0.0/16 -o
```

其他任何远程界面送出的内容都是允许的。

```
ipfwadm -O -a accept -V your.static.PPP.address -S your.static.PPP.address/32 -D  
0.0.0.0/0
```

回送界面是允许的。

```
ipfwadm -O -a accept -V 127.0.0.1 -S 0.0.0.0/0 -D 0.0.0.0/0
```

捕捉所有规则，其他任何的送出方式都会被拒绝并记录。

```
ipfwadm -O -a deny -S 0.0.0.0/0 -D 0.0.0.0/0 -o
```

伪装网关的转送设置，更新以及设置拒绝的策略。

```
ipfwadm -F -f
ipfwadm -F -p deny
```

伪装局域网的机器从本地界面送出至任何地方的数据。

```
ipfwadm -F -a masquerade -W ppp0 -S 192.168.0.0/16 -D 0.0.0.0/0
```

捕捉所有规则，其他任何的转送方式都会被拒绝并记录。

```
ipfwadm -F -a deny -S 0.0.0.0/0 -D 0.0.0.0/0 -o
```

用户可以使用-I、-O 或-F 规则来控制到某特定节点的流量。记得这些规则集是由上往下扫描的，而-a 代表附加(append)到目前现有的规则集中，所以任何限制必须在全域规则之前出现。

使用-I 规则的例子：

拒绝并记录本地界面，局域网络里的机器通往 204.50.10.13。

```
ipfwadm -I -a reject -V 192.168.255.1 -S 192.168.0.0/16 -D 204.50.10.13/32 -o
```

本地界面，局域网络里的机器，允许通往任何地方。

```
ipfwadm -I -a accept -V 192.168.255.1 -S 192.168.0.0/16 -D 0.0.0.0/0
```

使用-O 规则最慢，因为数据包首先经过伪装，但这个规则阻止防火墙存取禁止的节点。使用-O 规则的例子：

拒绝并记录送出至 204.50.10.13 的数据

```
ipfwadm -O -a reject -V your.static.PPP.address -S your.static.PPP.address/32 -D 204.50.10.13/32 -o
```

允许其他任何远程界面送出的内容

```
ipfwadm -O -a accept -V your.static.PPP.address -S your.static.PPP.address/32 -D 0.0.0.0/0
```

使用-F 规则的例子：

拒绝并记录 PPP 界面送出的从局域网络到 204.50.10.13 的数据。

```
ipfwadm -F -a reject -W ppp0 -S 192.168.0.0/16 -D 204.50.10.13/32 -o
```

伪装本地界面从局域网络送出至任何地方的数据。

```
ipfwadm -F -a masquerade -W ppp0 -S 192.168.0.0/16 -D 0.0.0.0/0
```

不需要有特定的规则来允许 192.168.0.0/16 通往 204.50.11.0，这包含于全域规则中。

设置上述规则的方法有多种。例如可以使用-W eth0 来取代-V 192.168.255.1，可以使

用-W ppp0 来取代-V your.static.PPP.address。

4.2.12 IP Masquerade 以及需求方式 (on-demand) 拨号

1. 如果用户想把网络设置成自动拨号上因特网, 那么 diald demand 拨号软件包将会是很好的工具。
2. 设置 diald。
3. diald 和 IP masq 设置完后, 用户可以在任何客户端机器上启动 Web、Telnet 或 FTP 连接。
4. Diald 将会检测到进入系统的请求, 然后拨号到用户的 ISP 并建立连接。

4.2.13 IPautofw 数据包转发程序

IPautofw18 是一个给 Linux 的 IP Masquerade 使用的 TCP 和 UDP 转发程序。当使用需要 UDP 的软件包的时候, 需要装入特定的 ip_masq 模块: ip_masq_raidio 和 ip_masq_cuseeme 等。

第 5 章 Linux 充当防火墙和网络设备

Linux 是一个功能非常强大的操作系统，利用它可以充当防火墙、网桥和路由器等设备，本章将讲述这些内容。

5.1 Linux 充当防火墙

目前，防火墙成了因特网安全上的热门话题，

本文将简单介绍如何在 Linux 上安装防火墙，同时也讲述代理服务器的安装和使用。利用代理服务器，可以使在防火墙后方的用户也能对因特网拥有比较大的访问权限。

5.1.1 防火墙简介

防火墙原来是用在汽车上的一个术语，它用来隔离引擎和乘客，在引擎爆炸时可以保护乘客的安全。计算机上的防火墙是一个逻辑装置，用来保护私人的区域不受外界的侵害，方法是：

1. 找一台有路由能力的计算机(例如 Linux)。
2. 加入两个接口(例如：串行端口、Ethernet 和 Token Ring 等)。
3. 关掉 IP Forwarding 的功能。
4. 把其中一个接口接上因特网。
5. 把受保护的网络接在另一个接口上。

然后把计算机接到两个不同的网络上。现在这台计算机就称为防火墙了。它可以接到因特网上，也可以接到受保护网络上。但受保护网络无法连上因特网，因特网也连不上受保护网络。

如果要从受保护网络内连上因特网，必须先远程登录 Telnet 到防火墙，再利用防火墙使用因特网。同样地，因特网上的人要进入受保护网络，也必须先通过防火墙。

这种做法对于因特网上的攻击有很好的免疫作用。如果有人企图对受保护的网络进行攻击，则必须先穿过防火墙，攻击必须分成两步，增加了难度。如果有人想借助一般的方法，如邮件炸弹或因特网蠕虫，来攻击受保护的网路，他们可能无法如愿。

5.1.1.1 防火墙的缺点

防火墙最大的问题在于由内部访问因特网将变得很困难。

基本上, 防火墙利用 Dial-Up Shell 的帐号来减少对于因特网的使用, 内部网络的用户必须先登录到防火墙才能对因特网进行其他的访问操作。因此一些需要直接连接因特网的程序(如 Netscape), 就无法在防火墙后方顺利操作, 对这种问题解决办法是使用代理服务器, 这部分内容将在下面简述。

5.1.1.2 代理服务器

代理服务器可用来协助用户从防火墙后方直接访问因特网。

它的工作是在服务器上打开一个套接字(socket)程序, 作为与因特网沟通的管道。举例来说, 用户的计算机 drig 在受保护的网内, 当用户要使用 Netscape 去浏览 Web 时, 得在防火墙上建一个代理服务器, 并把这个代理服务器设置为可以接受用户的计算机的请求, 即把要连上端口 80 的请求转为接到代理服务器的端口 1080 上, 从而访问因特网。

用过 TIA 或 TERM 的用户应该对端口转向不会陌生, 在这两个程序里, 可以对一个端口做转向的操作。例如: 有人用 TIA 设置允许大家用代理服务器地址 192.251.139.21、防火墙的端口 4024 去连上他的 Web 服务器。这里提到的代理服务器, 其原理也差不多, 只是恰好相反, 使用端口 1080(或某特定值)去连接其他人的端口 80。代理服务器最了不起的地方在于它的安全性, 如果设置正确, 它不会允许任何人逾越它。

5.1.2 安装与配置

下面就来看一看如何在 Linux 系统上安装防火墙和配置防火墙。

5.1.2.1 硬件需求

本例中, 计算机采用 486-DX66, 8MB 物理内存, 500MB 硬盘, 以一台 14 400bps 的调制解调器连到 ISP(因特网服务供应商)上。这就是 Linux 的基本配备, 要架成防火墙, 还得加上一块 NE2000 Ethernet 网卡。接着连上三台 PC 机(Windows 3.1 和 Trumpet Winsock)和两台 Sun 工作站(SunOS 4.1)。选择这样的框架是因为这种框架的应用比较普遍。

5.1.2.2 设置软件

首先, 重新编译系统内核, 执行 make config 命令时得做适当的调整:

1. 使用 Networking Support。
2. 使用 TCP/IP Networking。
3. 取消对 IP Forwarding(CONFIG_IP_FORWARD)的选择。
4. 使用 IP 防火墙。
5. 可以使用 IP accounting, 这样比较审慎一点。
6. 使用 Networking Device Support。
7. 使用 PPP 及 Ethernet support, 视接口而定。

接着, 重新编译, 重新安装内核后重新启动。应该可以在启动过程中看到所加入的接口的信息。

5.1.2.3 设置网络地址

因为不希望因特网直接访问用户的网络，所以不必使用真实地址。C 类地址的 192.168.2.xxx，它是被设置来做测试的，所以尽可放心地用。在设置中，只需要一个真实的地址，其他的自己设定就可以了。将防火墙上的以太网卡的地址设为 192.168.2.1，将其他计算机的地址设为 192.168.2.xxx。

5.1.2.4 测试

试着从防火墙上通过 ping 命令测试一个因特网上的节点。使用 nic.ddn.mil 作为测试点。如果不行，查 PPP 有没有设好，然后再试一次。

现在，试着通过 ping 命令测试保护网络内的计算机，所有网络内的计算机应该可以通过 ping 连接其他任何一台计算机，如果不行，再试一次。接着，所有在受保护网络内的机器应该都可以通过 ping 命令连通防火墙。若不行，再重复以上步骤，记住，应该通过 ping 命令测试 192.168.2.1，而不是 PPP 的真实 IP 地址。

然后试着从防火墙里的计算机去通过 ping 命令测试 PPP 的 IP 地址，如果可以，表示 IP Forwarding 的功能未取消，必须重新编译内核。虽然已经把受保护的网路设为 192.168.2.1 地址，不会收到来自因特网的数据包，可是把 IP Forwarding 关掉还是比较保险，可以自己控制全局而不必依赖 ISP。

最后，从防火墙上通过 ping 命令测试受保护的网路内的机器，到此，应该没有什么问题。这样就已经完成了防火墙的最基础的安装。

5.1.2.5 防火墙的安全性

目前的防火墙还不够好，因为它还敞开着大门等待别人来攻击。首先，可以找到 /etc/inetd.conf 文件，它又被称为“超级服务器”，因为其上有很多服务器守护进程被执行。例如：

- Telnet
- Talk
- FTP
- Daytime
- Finger
- rlogin
- rexec
- ruser

取消所有不需要的功能，例如 netstat、sysstat、tftp、bootp 和 finger。甚至可以关掉 Telnet 而只允许 rlogin。

只要在该功能前加上“#”符号就可以轻易地把它取消掉。利用 kill-HUP <pid> 发送一个 SIG-HUP 信号给 inetd 守护进程，使 inetd 守护进程重新去读取刚才更改的设置 (inetd.conf)，同时重新启动 inetd 守护进程。

试着远程登录(Telnet)防火墙的端口 15(netstat 端口)，如果还读得到 netstat 的数据，

表示没有正确重新启动 `inetd` 守护进程。

5.1.3 代理服务器

下面介绍安装和设置代理服务器的方法及代理服务器的使用等内容。

5.1.3.1 安装代理服务器

与防火墙服务不同，代理服务器需要额外的软件，可以从下面的站点获取：

```
ftp: //sunsite.unc.edu/pub/Linux/system/Network/misc/socks-Linux-src.tgz
```

在 `sock-conf` 目录下还有一个配置文件范例。解压缩之后，按照指示对它使用 `make` 命令，在使用 `make` 命令时可能遇到很多问题，确定 `Makefile` 正确无误。记住要把代理服务器加到 `/etc/inetd.conf` 中，可以加入如下的行：

```
socks stream tcp nowait nobody /usr/local/etc/sockd sockd
```

5.1.3.2 设置代理服务器

`socks` 的程序需要两个配置文件。一个说明哪些访问是被允许的，另一个则是把请求发送到适当的代理服务器。访问文件必须设在代理服务器上，而路由文件则要装在每台 UNIX 机器上。

1. 访问文件

在 `socks4.2 Beta` 中，访问控制文件称为“`sockd.conf`”，应该包括两行，一行是允许的(`permit`)，一行是禁止的(`deny`)，每行有三个项目：

- 关键字(`permit/deny`)
- IP 地址
- 地址修正值

关键字不是 `permit` 就是 `deny`，应该要有一行 `permit`、一行 `deny`。IP 地址采用因特网上的标准记法，例如 `192.168.2.0`。地址修正值也是采用 IP 地址的格式，而有子网掩码的效果。把它想象成一个 32 位的二进位数，若该位为 1，表示在地址比较时，此一位和之前 IP 地址的该位相符。例如：

```
permit 192.168.2.23 255.255.255.255
```

意思是只有 `192.168.2.23` 才是相符，而

```
permit 192.168.2.0 255.255.255.0
```

会允许 `192.168.2.0` 到 `192.168.2.255` 间的 IP 地址，即整个 C 类地址域。如果加入如下一行：

```
permit 192.168.2.0 0.0.0.0
```

则代表是所有的地址都允许了。所以先允许要开放权限的地址，再拒绝其他的地址。例如：

```
permit 192.168.2.0 255.255.255.0
deny 0.0.0.0 0.0.0.0
```

会允许所有的 192.168.2.xxx 地址访问，注意 deny 行里的“0.0.0.0”使用了 0.0.0.0 做修正，“0.0.0.0”并没什么特殊用意，修正值就拒绝了所有的地址，用 0 只是因为打字方便罢了，要用 255.255.255.255 也可以。除这两行外，还可以加上额外的限制。

也可以允许或拒绝某一特定的用户，但这需要 ident 认证方法，并非所有的系统都有此功能，如 Trumpet Winsock 就没有，所以就不再深入讨论。

2. Routing 文件

Routing 文件称作 socks.conf，它很容易与访问控制文件相混淆。

Routing 文件用来控制 socks 客户程序何时用 socks 服务，何时不用。例如，在用户的网络上，192.168.2.3 和 192.168.2.1(防火墙)使用 talk 命令时就用不到 socks 服务，它们直接用以太网相连。

因为系统自动定义 127.0.0.1 作为回路之用，和自己的主机使用 talk 命令时当然也不需 socks 服务。

文件中有三个项目：

- deny
- direct
- sockd

deny 表明何时要拒绝请求，它的语法和 sockd.conf 相同。一般来说，把修正值设为 0.0.0.0 就可以轻易挡掉所有的外来者。direct 项指出哪些地址不必用到 socks，即不必通过代理服务器就可连到的地方。同样有三个项：关键字、IP 位址和地址修正值，例如：

```
direct 192.168.2.0 255.255.255.0
```

这使用户可以在受保护网络内通行无阻。sockd 项说明哪些计算机上有 socks 守护进程在执行，语法是：

```
sockd @=<serverlist> <IP address><modifier>
```

注意“@=”那一栏，它将列出一些代理服务器的 IP 地址。在这个例子中，只用一台代理服务器，但可以多设几个代理服务器，以容纳高负载，同时减少故障的风险，IP 地址和修正值用法同前。

3. 防火墙内的 DNS

相比之下，在防火墙内安装 DNS 是很简单的，只要在充当防火墙的机器上装一个 DNS，并且将防火墙内的机器 DNS 设置改成防火墙的 IP 地址就行了。

5.1.3.3 使用代理服务器

1. UNIX

若应用程序想要使用代理服务器，先决条件是它们必须是被套接字化型的，即 msocket 为接口的，所以必须有两个 Telnet，一个是直接连接的，另一个则是通过代理服务器连接。Socks 指导用户怎么去套接字化一个程序，同时也有一些已经被套接字化的程序，如果使用一套接字化的版本去直接做连接，Socks 会自动转换成直接版。所以，得把所有受保护

网内的原有程序改为套接字化的版本，如将“finger”改为“finger.orig”，“telnet”改成“telnet.orig”等等，必须在 include/socks.h 文件中告诉 Socks 这些数据。

有些程序会自动路由并套接字化自己，Netscape 就是一个例子。可以在 Netscape 中使用代理服务器，只要在 Proxies 选项中输入服务器的地址和对应的端口就可以了。

2. Windows/Trumpet Winsock

Trumpet Winsock 具有使用代理服务器的能力，在 Setup 菜单里输入服务器 IP 地址和其他可以直接连通的计算机地址，Trumpet 会处理所有外送的数据包。

5.1.3.4 使代理服务器处理UDP数据包

有点美中不足的是 Socks 软件只能处理 TCP 数据包，而不处理 UDP 数据包。很多程序，如 talk 和 Archie 等都使用 UDP。有一个软件可以用来当作 UDP 的代理服务器，该软件是 UDPrelay，由 Tom Fitzgerald 所写(fitz@wang.com)。不过，到目前为止，它还不兼容于 Linux。

5.1.3.5 代理服务器的缺点

代理服务器是个安全装置，用它在有限的 IP 地址上增加因特网的访问会有一些缺点。代理服务器允许在受保护网络内的计算机大量访问因特网，而让外界不能触及受保护网络内部(亦即外部的服务器)。talk、Archie 和 mail 都无法直接连接受保护网络内部的计算机，看起来没什么，可是仔细想想：

- 可能在受保护网络内用计算机打了一篇报告，回家之后，想把它拿出来看看。抱歉，这是不可能的，根本无法访问计算机，因为它在防火墙内部。试着登录防火墙，可是因为每个人都用代理服务器访问，所以没有人会在上面另设帐号。
- 与女友用电子邮件通信，有些“不可告人”的私事要讲，请她把电子邮件直接寄到计算机上会比较好，但是不行。
- 无法处理 UDP 数据包是代理服务器的致命弱点，UDP 的用途会愈来愈多。例如：CoolTalk 等。

FTP会造成代理服务器上的另外一个问题，当下载文件或是进行一个ls操作时，FTP 服务器会开一个 socket 在客户端机器上，并由它来传送信息。而一个代理服务器不会允许这么做，所以 FTP 就无法顺利完成。另外，代理服务器运行起来挺慢的，因为开销太大了，而其他的方法相比之下就快多了。

如果只有一个 IP 地址，也不担心安全的问题，就用不到防火墙或是代理服务器；如果不是，且也不担心安全问题，可以找找 IP 仿真器之类的工具，如 Term、Slirp 或 TIA。Term 可以在以下的站点获取：

```
ftp: //sunsite.unc.edu
```

Slirp 可以在下面的站点获取：

```
ftp: //blitzen.canberra.edu.au/pub/slirp
```

而 TIA 可以在下面的站点获取：

```
ftp: //arketplace.com
```

这些工具运行起来快多了，连接的效率也较高，同时由因特网连入内部网络的权限也大多了。代理服务器适合那些有很多主机要连上因特网却不想安装和设置更多内容的用户。

5.2 Linux 充当网桥

这部分内容描述如何去设置以太网网桥。

什么是以太网网桥？

以太网网桥是一个设备，它可以控制在一个子网内的网络数据包，以减少网络的大量流量。

网桥通常安放在两组分开的计算机用户组之间，使计算机在自己组内所属的相互沟通，而不是频繁地和其他用户组的计算机沟通。

网桥的作用是分隔子网，它从硬件的层次上查看通过它的数据包，就像实际中的桥的作用一样，决定是否将数据包发送给“对岸”，结果使网络变得更快、更安静且碰撞更少。

5.2.1 设置

1. 取得网桥配置软件

有些 Linux 系统，并不直接提供网桥软件，对这些系统而言，需要安装网桥软件，而最新版本的 Linux 系统，都是本身就带有网桥软件。

从下面的站点可以获取该软件：

```
ftp://shadow.cabi.net/pub/Linux/BRCFG.tgz
```

2. 取得和阅读 Multiple ethernet HOWTO

在进行网桥配置之前，如果对网络不是很熟悉，最好阅读一下 Multiple ethernet HOWTO 帮助，里面详细讲述了如何将以太网分段。

从下面的站点可以获取该文件：

```
ftp://sunsite.unc.edu/pub/Linux/docs/HOWTO/mini/Multiple-Ethernet
```

3. LILO 配置网卡

为了使 Linux 系统成为网桥，首先该系统必须至少安装两块网卡。

使多个以太网设备在机器上运行。如果系统上安装了两个网卡，需要加入如下一行到 `/etc/lilo.conf` 文件中，然后重新执行 `lilo` 命令：

```
append = "ether=0, 0, eth1"
```

如果在网桥上有三张网卡，则使用如下一行：

```
append = "ether=0, 0, eth1 ether=0, 0, eth2"
```

靠加入更多的以太网语句可以找到更多的网卡。默认的情况下，一个传统的 Linux 内

核只检测一块以太网网卡，而且当找到一个网卡后，检测就会停止。上面所附加的陈述要求内核在找到第一个网卡后，继续检测，以找到更多的以太网设备。

另一个方法是使用启动变量。对于两个网卡的系统，使用如下语句：

```
Linux ether=0, 0, eth1
```

对于三个网卡的系统，则使用：

```
Linux ether=0, 0, eth1 ether=0, 0, eth2
```

4. 重新编译 Linux 内核以启动 BRIDGING

通过下面的命令来重新编译 Linux 内核：

```
cd /usr/src/Linux
make config
```

在提示是否充当网桥时，输入“Y”。

这样，就完成了基本的网桥的配置。

5. 一个网桥不必有一个 IP 地址

一个网桥可以有一个 IP 地址，但是一个普通的网桥不需要一个 IP 地址。

如果要从网桥上删除 IP 地址，可以到/etc/sysconfig/network-scripts/(对于一个 RedHat 系统而言)目录下，并拷贝 ifcfg-lo0 文件到 ifcfg-eth0 和 ifcfg-eth1。在这两个以太网配置文件中，将包含“DEVICE=lo”的语句修改为“DEVICE=eth0”和“DEVICE=eth1”。

其他的 Linux 版本也许略有不同，可以自己动手试试。

6. 重新启动

重新启动，执行包含网桥功能的内核，另外也查明并确认没有 IP 地址指定到这个网卡。

```
#reboot
```

5.2.2 网卡模式

以太网网卡在默认的方式下，仅仅接受发送给自己的数据包，而丢弃其他的数据包。

超级用户可以将以太网网卡设置成纷乱(promiscuous)模式，在纷乱模式的工作方式下，以太网网卡接受通过它的所有数据包。

要使 Linux 系统成为网桥，必须把以太网网卡设置为纷乱模式，它们将查看每一个通过它的数据包。

使网卡处于纷乱模式可以使用下面的命令：

```
ifconfig promisc eth0 : ifconfig promisc eth1
```

这样，连接到同一网段的所有的网卡被网桥分开。

5.2.3 启动网桥

将内核重新编译成支持网桥还不够，必须使用网桥软件进行配置。

下一步，要启动网桥软件。

启动网桥的方法是使用 `brcfg` 程序：

```
brcfg -ena
```

1. 检验每一个网卡有不同的交通流量

要检查某个网卡上的数据传输情况，用程序：`tcpdump`。

下面是使用该程序的一个例子：

```
tcpdump -i eth0 (在一个窗口)
tcpdump -i eth1 (在另一个窗口)
```

2. sniffer

除了上面的 `tcpdump` 程序之外，还有另外一个功能类似的程序：`sniffer`。

它的作用也是检查通过某个网卡的所有数据包，并显示在屏幕上。

因此，可以执行 `sniffer`(监看器)或 `tcpdump` 在其他机器上去检验网桥是否能正确地分开网段。



注意： `sniffer` 和 `tcpdump` 都必须在超级用户状态下才能够运行。

5.2.4 可能出现的问题

1. ioctl(SIOCGIFBR) failed: Package not installed 这是什么意思？

用户的内核没有网桥的能力。取得一个 2.0 或者更新的内核，并重新编译启动网桥功能的选项 `BRIDGING`。

有关重新编译启动网桥功能的操作，在前面已经讲述过，在此就不再重复了。

2. 机器的一边不能通过 ping 命令连到另外一边

已经使用“`brcfg -ena`”命令使网桥运行了吗？如果是，则 `brcfg` 命令将显示“`bridging is ENABLED`”。

已经把网卡设在纷乱模式了吗？使用“`ifconfig`”命令，具体的操作在前面已经讲过了。如果正在用多媒体网卡，确定启动的是否正确的那一端。

也许需要使用网卡附带的配置/设置程序来进行设置。

3. 不能从网桥上使用 Telnet/FTP

这是正常的现象。

这是因为没有 IP 地址指定到任何的网桥的网卡。

一个网桥是一个网络的通透部分，是为其他主机提供服务的。

因此，不能从网桥上使用 Telnet/FTP 是正常的现象。

4. 在设置路由方面需要做什么

不需要做任何的事情。整个的路由数据由内核的网桥的代码所掌握。

5. traceroute 问题

网桥看起来已经在运行，但是为什么 traceroute 命令的执行结果没有显示那网桥是路径的一个部分？

由于网桥的通透子网的功能，traceroute 将不显示网桥作为路径的一部分。一个网桥是网络的一个通透组件。

6. IP_FORWARD 是否必要编译到内核中

不必要。IP_FORWARD 对于一个网关，它要有 IP 地址指定到它的网络接口，而充当网桥的 Linux 系统上的网络接口是没有 IP 地址的。

7. 网桥的启动

在内核配置的时候，针对以下的问题回答“Y”：

```
"Prompt for development and/or incomplete code/drivers (CONFIG_EXPERIMENTAL)
[Y/n/?]"。
```

8. 同时连接 10MB 和 100MB 两个以太网网络区段

一个网桥界面能够同时连接 10MB 和 100MB 两个以太网网络区段吗？这样的配置会造成高速流量的另一边慢下来吗？

能，一个网桥能够把 10MB 的区段和 100MB 的区段绑在一起，只要在网桥的快速网络的那一边的是有 100MB 能力的。

这时它发生的情形是：从一个在 100MB 网络的主机发送数据包给另一个在 10MB 网络的主机时只有用 10Mbps 的速度传输，其余的那网络传输速度在快速以太网网络不会被拖慢。

5.3 Linux 充当路由器

Linux 系统本身可以被配置成路由器，为其他主机提供路由信息。本节就来看一看如何具体配置 Linux，使其充当路由器。

5.3.1 配置方法

利用 Linux 充当路由器其实很简单。

首先在重新编译内核时要选中 IP Forwarding 功能。用这个内核启动，看看能不能找到两块网卡。如果不行，则在/etc/lilo.conf 文件上加上下面的语句：

```
append="ether=IRQ, io-port, eth0 ether=IRQ, io-port, eth1"
```

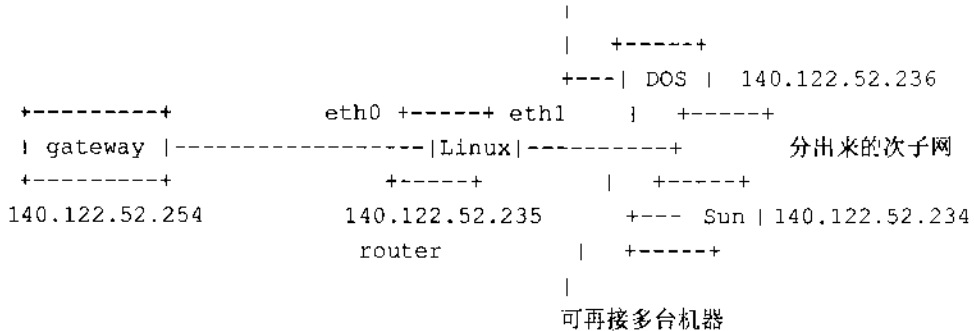
然后，再重运行 lilo 命令。

或者，如果用 loadlin，则用下面的命令：

```
loadlin vmlinuz root=/dev/??? ro ether=IRQ, io, eth0 ether=IRQ, io, eth1
```

即可。注意看启动时有没有找到网卡。或看 `/proc/net/dev` 有没有 `eth0` 和 `eth1`，有就对了(如果没有，就要重新编译内核了)。

接下来就是配置这两个接口了。假设网络环境如下：



`eth0` 接口是用来跟外面的网络相连的。可以用一般的 `netconfig` 程序来设置它。

利用 Linux 的 Proxy Arp 功能，当外界有数据包要送到次子网络时，Linux 会响应此数据包。所以对外界来说次子网就是存在的了。因此很重要的一点是，次子网必须是从原来的子网络上分割出来的。

以上面的情形为例，如果估计次子网有 5 台左右的机器，可以决定要分割出来的次子网的 IP 范围是从 140.122.52.232~140.122.52.239 共 8 个地址，计算一下，`netmask` 应是 255.255.255.248，`broadcast` 是 140.122.52.239。于是 `eth1` 应该这样配置：

```
# ifconfig eth1 140.122.52.235 netmask 255.255.255.248 broadcast 140.122.52.239
# route add -net 140.122.52.232 netmask 255.255.255.248 gw 140.122.52.235 eth1
```

这样，所有从 140.122.52.232 到 140.122.52.239 的数据包都会往 `eth1` 送。试试看能不能从次子网通过 Telnet 连接到 Linux 上？若可以，就成功了--大半。

不知用户是否注意到，在这里是将 `eth0` 和 `eth1` 的 IP 地址设为一样。常有人认为，有两块网卡就要有两个 IP 地址。但这是没有必要的，可以设为相同。

但是若想试着直接通过 Telnet 连接到外面网络的机器，就会发现不通。这是因为尚未设置 arp 表。以上面为例，如果想要 DOS 那台机器可以跨出次子网，就要在 Linux 上设置 arp：

```
# arp -s 140.122.52.236 08: 00: 77: 14: 00: 00 pub
      ^^^^^^^^^^^^^^^^^^^^^
      ↑
      这是 eth0 接口的硬件地址，可用 ifconfig 查到
```

这时应该就可以直接从 DOS 机器远程登录到外面网络上的机器了。也可以从外面网络上的机器通过远程登录连接到内部的 DOS 机器上。

如果在外网用 `traceroute` 命令查看 route 的路径，会发现充当路由器的那台 Linux 机器就在上面。

可以把这些命令都放在 `/etc/rc.d/rc.inet1` 中，这样启动时就可以设置好两个接口了。

5.3.2 可能出现的问题

下面是用 Linux 充当路由器时一些应该注意的小问题。

1. 网卡的选择

如果不打算做路由器，什么样的网卡都行。但若准备用 Linux 作路由器，最好别用两块同型号的网卡。不能肯定用了就一定不行，但有些情况下两块 NE2000 兼容网卡是不能同时正常工作的。

2. 启动盘和根盘的制作

值得注意的是在做根盘时，RAWRITE 命令输入的参数应是带.gz 的压缩文件，千万别自作聪明地先把它解压缩了。

为什么呢？

这是因为 Linux 的内核是压缩的，每次启动需要自解的缘故。

3. NFS 安装

如有条件最好采用 NFS(网络文件系统)安装。可以先从网上把 slackware 软件取过来，放入自己的或附近的一台主机里，通过 NFS 把它共享，然后在本地安装时，在“安装方式”步骤中选择“通过 NFS 安装”就可以了，这时用户需要输 NFS 服务器的 IP 地址和存放 slackware 系统的目录，之后安装程序会自动从 NFS 服务器上安装 Linux 系统。

4. 编译内核

如果需要充当路由器或用户的网卡不支持，那必须做这一步。编译时注意打开 IP Forwarding，并选择所用的网卡。

5. 调试路由器时最好用 PC 机

比如调试过程中路由器换了网卡，子网外的工作站就需要更改它的 ARP 信息，如果对 UNIX 系统管理不熟就比较麻烦。

也可以使用笨办法，重新启动：

```
#reboot
```

6. 路由器的性能

用 Linux 作路由器，会使子网的出口速度较没有路由器时稍慢，大概是原来的 70%。当然，有了路由器，可以使子网免受外网故障的影响。有所失也必有所得。

第 6 章 电子讨论组与新闻组

电子讨论组和新闻组是在因特网上使用较多的两种服务，Linux 系统不仅支持这两种服务的访问，同时还能够提供这两种服务。本章就来详细讲述如何在 Linux 系统上建立电子讨论组服务和新闻组服务。

6.1 以 majordomo 1.93 建立电子讨论组

本章将介绍如何在 Linux 系统上建立一个电子讨论组，并结合实例来使用户掌握配置的整个过程。

6.1.1 系统要求

要建立电子讨论组服务器，需要以下的软件：

1. Sendmail 8.xx
2. perl 4.036 到 5.001a 及其库文件
3. C 编译器
4. majordomo 1.93 软件

6.1.2 建立 majordom 帐号

在建立电子讨论组之前，首先要建立一个 majordom 帐号。

使用下面的命令来创建 majordom 帐号：

```
adduser majordom
```

其工作组名也是 majordom，其主目录是 /usr/local/majordomo。

```
#chmod 775 /usr/local/majordomo
```

6.1.3 获取 majordomo 软件

通过 su 命令转换成 majordomo 用户，然后到下面的站点获取文件：

```
ftp://nctuccca.edu.tw/packages/mail-server/majordomo/majordomo-1.93.tar.gz
```

并存放到目录/usr/local/majordomo 下，然后解压缩。

6.1.4 安装过程

下面是软件的安装过程：

1. 修改 Makefile(/usr/local/majordomo/majordomo-1.93/Makefile)。

1) 设置环境变量 W_BIN, W_BIN=/usr/local/majordomo。

2) 去掉 POSIX 部分的注释。

2. 执行 make wrapper 命令。



注意： wrapper 是用来设置 set uid 的，所以在 Makefile 中的 W_UID 与 W_GID 一定要设对。

3. make install

4. make install-archive2

5. make install-wrapper

此步骤需以 root 身份执行。

6. ln -s /usr/local/majordomo/majordomo.cf/etc/majordomo.cf

7. 在 \$HOME(/usr/local/majordomo) 下执行下面的命令：

```
mkdir archives, digests, lists;
```

```
chmod 775 archives, digests, lists;
```

```
touch Log
```

```
chmod 644 Log
```

8. ln -s /usr/bin/perl /usr/local/bin/perl

该步骤是为了避免一些 perl 脚本指向 /usr/local/perl 而不是 /usr/bin/perl。

9. 对某些文件要做必要的修改：

1) request-answer

- 在 request-answer 的开始处插入一行 require "shlock.pl"

- 将所有的 do_exec_sendmail 改成 do_exec_sendmail_in_request_answer

- 第 41 行：

```
exec("/usr/lib/sendmail", "-f$list-request", "-t")||
```

将 -f\$list-request 改成 -f\$list-approval

2) resend 第 348 行：

```
&bounce("Approval required")改成&bounce()
```

10. edit majordomo.cf

范例如下:

```
# $whereami -- What machine am I running on?
# Need modified

$whereami = "girl.ee.ntu.edu.tw";

# $whoami -- Who do users send requests to me as?

$whoami = "majordomo@$whereami";

# $whoami_owner -- Who is the owner of the above, in case of problems?

$whoami_owner = "owner-majordomo@$whereami";

# $homedir -- Where can I find my extra .pl files, like majordomo.pl?
# the environment variable HOME is set by the wrapper

$homedir = "/usr/local/majordomo";

# $listdir -- Where are the mailing lists?

$listdir = "/usr/local/majordomo/lists";

# $digest_work_dir -- the parent directory for digest's queue area
# Each list must have a subdirectory under this directory in order for
# digest to work. E.G. The bblisa list would use:
# /usr/local/mail/digest/bblisa
# as its directory.

$digest_work_dir = '/usr/local/majordomo/digests';

# $log -- Where do I write my log?

$log = "$homedir/Log";

# $mailer -- What program and args do I use to send mail?
# The variable $to can be interpolated into this command line,
# however the $to variable is provided by the person sending mail,
# and much mischief can be had by playing with this variable.
# Use $to with care.

$mailer = "/usr/lib/sendmail -f\u0024sender -t";

# Majordomo will look for "get" and "index" files related to $list in
# directory "$filedir/$list$filedir_suffix", so set $filedir and
```

```

# $filedir_suffix appropriately. For instance, to look in
# /usr/local/mail/files/$list, use:
# $filedir = "/usr/local/mail/files";
# $filedir_suffix = ""; # empty string
# or to look in $listdir/$list.archive, use:
# $filedir = "$listdir";
# $filedir_suffix = ".archive";

$filedir = "/usr/local/majordomo/archives";
$filedir_suffix = "";

# What command should I use to process an "index" request?

$index_command = "/bin/ls -lG | /usr/bin/sed -e 's/^...../' -e 'ls/.*/ SIZE DATE
NAME/'";

#$index_command = "/bin/ls -lL | /usr/bin/awk '{print $5, $6, $7, $8, $9}'";
# If you want to use FTPMAIL, rather than local access, for file transfer
# and access, define the following:
# $ftpmail_address = "ftpmail@decwr1.dec.com";
# $ftpmail_location = "FTP.$whereami";
# if you want the subject of the request to be included as part of the
# subject of the reply (useful when automatically testing, or submitting
# multiple command sets), set $return_subject to 1.

$return_subject = 1;

# If you are using majordomo at the -request address, set the
# following variable to 1. This affects the welcome message that is
# sent to a new subscriber as well as the help text that is generated.

$majordomo_request = 0;

# Set the umask for the process. Used to set default file status for
# config file.

umask(007);

# Very Important!!!!!!!!!!!!!!!!!!!!
# the safe locations for archive directories. This should be defined as
# a series of root anchored directory paths as will be used as prefixes
# to the file names specified to the archive2.pl script.

@archive_dirs = ( "/usr/local/majordomo/archives/sample",
"/usr/spool/archive/firewalls"
);

```

```

# Set this to 1 if you want to use the experimental mechanism for allowing
# / in user names. People with lots of X.400 addresses on their lists or
# HP mail whatever may want to set this. However use it at your own risk.

$analyze_slash_in_address = 0;

#
# these tune the experimental matching that is done for addresses with / in
# them. If you haven't turned on the experimental analyze_slash_in_address
# they are ignored. See the source for full explanation of these variables.
#
# if set to 1 ignore the requirement that addresses have an @ sign in the
# address component after the last /.

$no_x400at=0;

# if set to 1 do not look for "/c=" and "/ad=" or "/am=" in the address.
# X.400 seems to require these components.

$no_true_x400=0;

#Edit /etc/aliases(after you edit the /etc/aliases
#you should run newaliases to make it work)
#owner-owner: postmaster

#
#Majordomo
#

owner-majordomo: long, nobody
majordomo: "|/usr/local/majordomo/wrapper majordomo"
majordom: owner-majordomo

#Needed "just in case" for Majordomo version 1.9x:

Majordomo-Owner: owner-majordomo
nobody: /dev/null

#
# Sample mailing list
#
# Adding "nobody" prevents expansion to "long" under sendmail v8:

owner-sample: long, nobody
sample-owner: long, nobody

```

```
#sample: "|/usr/local/majordomo/wrapper resend -p bulk -M 10000 -R -l sample -f

sample-owner -h girl.ee.ntu.edu.tw -s sample-outgoing"
sample: "|/usr/local/majordomo/wrapper resend -l sample -h girl.ee.ntu.edu.tw
sample-outgoing"
sample-approval: owner-sample
sample-outgoing: ; include: /usr/local/majordomo/lists/sample.
sample-archive.
"/usr/local/majordomo/wrapper digest -r -C -l sample-digest sample-
digest-outgoing"

#
# For digest version list
#

sample-digest: sample
sample-digest-outgoing: "|/usr/local/majordomo/wrapper request-answer
sample-digest"
owner-sample-digest: owner-sample
owner-sample-digest-outgoing: owner-sample
owner-sample-digest-request: owner-sample
sample-digest-approval: sample-approval

#
# End of the digest version list
#

owner-sample-outgoing: owner-sample
sample-archive: "|/usr/local/majordomo/wrapper archive -f
/usr/local/majordomo/archives/sample/sample -m -a"
owner-sample-archive: owner-sample

#
# For someone who may use listname-request to ask questions
#sample-request: "|/usr/local/majordomo/wrapper request-answer sample"

sample-request: "|/usr/local/majordomo/wrapper majordomo -l sample"
owner-sample-request: owner-sample

#
# End of listname-request version
```

6.2 新闻组软件 Leafsite

本节讲述如何在 Linux 系统下建立新闻组服务器。

6.2.1 使用 Leafnode 软件

如果用户要在本地计算机上脱机看新闻，必须安装一个新闻服务器软件，如 INN 或者 Cnews。也要使用 NNTP 或者 UUCP 才可连接到新闻服务器上浏览。

Leafnode 软件使用方便而且非常小，但是有一些缺点：在发生某些错误的情况下 Leafnode 运行缓慢并且丢失新闻。这就是不用它作为一个大新闻服务器的原因。然而，对于个人的用户这是适当的，因为用户不必花许多时间去设置 INN。

6.2.2 取得 Leafnode

要使用它，必须先取得该软件。

Leafnode 可以在下面的站点取得：

```
ftp://ftp.troll.no/pub/freebies/1
```

Leafnode 最新的版本是 1.4，在 Linux 的一些发行版本(例如 Debian)中，包含 Leafnode 软件。无论如何，确保使用最新的版本 1.4，它修正了几个重要的问题。

6.2.3 安装 Leafnode

Leafnode 软件包附带一个安装帮助。如果用户的发行版本是高于 1.4 的版本，最好的是使用事先翻译好的版本并且忽略步骤 1 到 4。

1. 确保没有其他新闻服务器在计算机上运行。输入：

```
$ telnet localhost nntp
```

如果得到一个错误信息，则表示系统上没有其他的新闻服务器在运行。如果得到一个连接的提示响应到 Shell 上，则表示有新闻服务器在运行，应先卸载 INN 软件。

2. 在/etc/passwd 中查看有没有用户名 news。如果没有，则建立一个，或者输入：

```
# adduser news
```

或者使用发行版本中所附的工具。

3. 解开源代码：

```
$tar xzf leafnode-1.4.tar.gz
```

改变目录到源代码目录：

```
$ cd leafnode-1.4
```

4. 编译程序和安装它：

```
$ make; make install
```

5. 编辑/usr/lib/leafnode/config 文件：

该文件也许在其他位置, 如/etc/leafnode。在该配置文件中, “server=” 一行要指出 ISP 的新闻服务器地址。

6. 编辑/etc/nntpserver 文件:

这将包括本地主机名(localhost 或者本地计算机的名称, 可以用命令 hostname 取得)。如果在一些启动文件, 如/etc/profile 或 ~/.bash_profile 中环境变量是被预先定义的, 也要调整计算机名称。

7. 编辑/etc/inetd.conf 文件:

查看有没有使用 “nntp” 开头的行。如果有这样的行, 在这行前加一个 “#”。然后加下面一行:

```
nntpsream tcp nowait news /usr/sbin/tcpd /usr/local/sbin/leafnode
```

当有人连接到计算机的 NNTP 端口时, Leafnode 将开始运行。

8. 以 root 或者 news 的身份进行连接和运行程序 fetch。第一次运行 fetch, 将下载 ISP 的新闻新闻组的列表。这需花一定时间, 依用户的连接速度和用户的 ISP 而定。

9. 执行用户最喜欢的新闻阅读器:

执行新闻阅读器。Sln、(r)tin 和 knews 是比较好的新闻阅读器。

10. 重新启动 fetch 命令, 下载所有想要取得的新闻组的新闻。

6.2.4 维护 Leafnode

运行新闻服务器后, 仍有一些事要做。也许要编辑文件/usr/lib/leafnode/config 来设置新闻组的期满日期。这个日期的意思是, 什么时候之前的信息应被删除。针对新闻组改变期满时间的方法如下:

```
groupexpire foo.bar n
```

这将针对新闻组 foo.bar 设置期满时间为 n 天。

这个设置不会使 Leafnode 删除以前的旧信息, 另有一个程序会负责完成工作——texpire。如果用户的计算机是全天候启动的, 也许要把下列的行加入到 news 用户的 crontab 文件中(用 news 的身份登录然后输入 “crontab-e”; 或以 root 身份登录, 然后输入 “crontab -u news -e”):

```
0 19 * * * /usr/local/sbin/texpire
```

这行内容会使 cron 守护进程在每一天的 19:00 开始执行命令 texpire。查看 crontab 帮助页可以得到更多的配置信息。如果用户的计算机是没有规则地启动, 也许每一次启动时都需要执行命令 texpire。

6.2.5 工作原理

Leafnode 是 NNTP 服务器, 它的意思是可以从一个不同的计算机上登录(通过万维网

或局域网等等)。每次用户用自己的新闻阅读器(newsreader)进入一个新闻组,用户的阅读器(reader)发送数据到 Leafnode 并请求它。如果新闻组不存在,Leafnode 将会建立一个空文件/var/spool/news/interesting.groups,名字很像新闻组。当下次运行 fetch 程序时,将会取得新闻组的信息。

如果一个新闻组在一个固定的时间内没有被访问,Leafnode 将会停止去取它的文章并从/var/spool/news/interesting.groups 中删除它。默认的情况下 Leafnode 会自动删除连续一个星期无人读取的数据文件。

一个星期对用户而言可能是不够的。如果用户要去度假三个星期,但仍然要取得新闻呢?但是,在 Leafnode 中没有用来改变它的选项。可以编辑文件 leafnode.h 并重新编译它。

#defined 后面的变量通常是 TIMEOUT_LONG 和 TIMEOUT_SHORT, 时间单位是秒。另外一个解决方法是定义一个 cron 任务,每晚执行“touch/var/spool/news/interesting.groups/*”这个命令。

如果要取得一个整个的新闻组的列表,查看文件/usr/lib/leafnode/groupinfo,在该文件中将会找到简短的描述新闻组的信息。

如果要重新读取新闻组的列表(例如,当要看一个新的新闻组),只需删除文件/var/spool/news/active.read。

6.2.6 使用新闻阅读器

没有针对 Linux 的新闻阅读器,就像没有针对 Linux 的编辑器。有人最喜欢的新闻阅读器是 gnus 模式下的 emacs,它可以设置的阅读器最多。许多人在终端上使用 slrn 和 tin,在 X Window 之下使用 knews。也有人使用 trn、nn 和其他阅读器。用户可以自由选择阅读器,最好别选 Netscape,因为 Netscape 是人的、没有特色和不稳定的。

6.3 Usenet 和新闻阅读器

Usenet 是一种开放的系统,在此系统上,用户可以发布新闻和阐述自己的某些观点。它操作起来像一个邮件系统,使得 Linux 系统上的任何用户能够从此邮箱阅读和发送消息。用户的消息被表现为 Usenet 文件,这些文件分布在能接受和识别它们的系统中。每个接受 Usenet 文件的系统被认为是一个站点。某些站点完成 Usenet 文件的组织和发布,它们从一些站点接收到消息,然后把它们组织成 Usenet 文件,并以广播的形式发送给别的用户。这些站点叫做主干站点,它们担当发行人的角色,收到文章并把它们组织分配到不同的组中。

要想访问 Usenet 新闻,首先要能访问一个新闻服务器。新闻服务器搜集到每天的新闻素材,并使得它们能被别的系统访问。在用户的网络环境中可以有一个系统充当新闻服务器。如果连接到某个因特网服务供应商(ISP),则 ISP 会提供一个新闻服务器供用户使用。

用户需要用新闻阅读器来阅读 Usenet 上的文章。新闻阅读器是连接到新闻服务

器并能存取文章的客户端程序。在因特网和基于 TCP/IP 的网络中, 新闻服务器和新闻阅读器之间的信息传输使用的是网络新闻传输协议(NNTP), 新闻服务器也被称为是 NNTP 服务器。

另外, 还可以在 Linux 上创建自己的新闻服务器, Linux 上新闻传输代理程序(news transport agents), 便能用来创建这样的新闻服务器。

6.3.1 Usenet 新闻

Usenet 文件具有刊物的功能, 包含在文件中的消息被认为是刊物中的文章。用户可以在 Usenet 上发表文章, 而且使它立刻发布到世界各地的 Usenet 站点。别人可以通过 Usenet 马上阅读文章, 而不必等待书面刊物的出版。因为每种刊物都属于不同的类别, 所以对 Usenet 文件也进行分类, 每类叫做一个新闻组。当用户发表文章时, 此文章便被指定到相应的新闻组中。其他用户可以在此新闻组中找到此文章来阅读。可以把每个新闻组看作是一个不断更新的杂志。例如, 如果想读关于计算机的文章, 可以到 Usenet 的计算机新闻组中找到此类文章。

近来, Usenet 文件也被用作电子公告牌, 人们可以在上面进行讨论。这些文件夹被划分为不同的新闻组。文件夹中的文章不像是杂志刊物中的文章, 而像是在谈话。

每个新闻组都有一个名字。为了分类, 通常每个名字包括三部分: 概括的主题、副标题和特定的主题。

每个部分之间用圆点分隔。例如, 有些关于娱乐的新闻组, 叫做 rec。其中有一些是关于食物的, 叫 food。这些组中有专门讨论特定主题的组, 如烹饪, 叫 recipes。那么这个新闻组的名字为 rec.food.recipes。

大多数的电子公告牌只是用来讨论。这些组用 alt 或 talk 作为概括的主题。例如, talk.food.chocolate 组中可能包括讨论巧克力多么好或多么坏的谈话, 而 alt.food.cholate 中可能包括一些不成熟的思考, 如巧克力在人类文明中的重要性。

下面是一些 Usenet 新闻组:

comp.ai.neural-nets
comp.lang.pascal
sci.physics.fusion
rec.ares.movies
rec.food.recipes
talk.politics.theory

Linux 有关于各种各样主题的新闻组。一些是讨论组, 还有一些包括 Linux 的最新开发信息。如果有问题, 可以在这些组中寻求帮助。下面列出的是最近很流行的一些 Linux 新闻组:

comp.os.Linux.announce	关于 Linux 开发的公告
comp.os.Linux.admin	关于系统管理的问题
comp.os.Linux.misc	关于特殊的问题和观点

comp.os.Linux.setup	关于安装中的问题
comp.os.Linux.help	问题和特殊难题的答案

6.3.2 安装 trn 和 tin

阅读 Usenet 文章需要一个新闻阅读器, 例如: trn 或 tin。这些新闻阅读器需要首先指定一个新闻组, 然后阅读其中的文章。

新闻阅读器有一个友好的用户界面, 允许浏览、选择、阅读、保存或打印文章。trn 可能是当今使用最广泛的新闻阅读器, 它是早期的新闻阅读器 rn 的最新的版本, 功能强大。它使用了一种高级的检索机制——线索, 用它来找出同一个主题的文章。

用户可以写一篇文章, 并把它加到一个新闻组中以供别人阅读。往一个新闻组中加一篇文章被称为发表文章。可以用一个特殊的功能——Pnews 来发表文章。

在使用 trn 和 tin 之前, 先要安装它们。Linux 系统的快速安装程序, 不能自动安装 tin。

首先要以超级用户注册进入系统。在开始安装之前, 要确信已用下面的命令把 CDROM 安装到 Linux 系统中:

```
#mount /dev/cdrom /mnt
```

要安装 trn 和 tin, 可以用安装程序 glint, 或在命令行上直接用程序 rpm。

用 glint 前, 首先需要为软件包打开一个窗口, 接着用鼠标单击 Application 图标, 然后单击 News 图标, 将看到 trn 和 tin 软件。选中相应的图标, 然后单击 Install 按钮开始安装。

另外, 还可以在命令行上进行操作。先转到 /mnt/cdrom/packages/RPMS 目录下, 接着输入如下命令:

```
#rpm -i trn-3.6.1.i386.rpm
```

```
#rpm -I tin-1.22-2.i386.rpm
```

用 tin 和 trn 都能阅读远程新闻服务器上提供的 Usenet 新闻。这些新闻服务器使用网络新闻传输协议(NNTP), 可以通过因特网访问它们。为了连接到远程的新闻服务器, 首先把新闻服务器的因特网地址赋给一个 Shell 变量 NNTPSERVER, 接着把这个 Shell 变量传给当前环境。Shell 变量 NNTPSERVER 的赋值和传送工作可以定义在初始化文件 .bash_profile 中, 当注册时自动对此变量进行赋值和传送, 例如:

```
$NNTPSERVER=news.servername.com
```

```
$export NNTPSERVER
```

使用 tin 连接到远程新闻服务器时, 要在 tin 命令行中使用 -r 参数:

```
$tin -r
```

或使用 rtin 命令。

6.3.3 新闻传输代理

因特网上的 Usenet 新闻被作为每日新闻素材提供给 10 000 多个新闻组。这些素材提供给一些站点, 这些站点允许其他的系统通过新闻阅读器来阅读新闻。这些站点作为新闻服务器, 访问它们的新闻阅读器作为它们的客户端。

新闻服务器软件又叫做新闻传输代理程序, 它能给新闻阅读器提供新闻, 使用户能阅读新闻组和在新新闻组上发表文章。Linux 上三个最流行的新闻传输代理程序为: INN、NNTP 和 Cnews。

Cnews 和 NNTP 都是小型的、简单的软件, 它们是专为小型的网络设计的。INN 是一种功能强大的软件, 专为大型的网络设计。

Usenet 上的每日新闻素材通常数量巨大, 而且消耗服务器的大量时间和内存, 所以用户可能不想用自己的 Linux 系统来接收这些素材。如果用户的系统运行在一个网络环境中, 可以用网络中的一个系统作为新闻服务器。在此系统中装上新闻传输代理软件, 此服务器便可以用来接收和管理新闻素材。网络中使用其他系统的用户可以通过新闻阅读器来访问新闻服务器。

如果用户的网络环境中已有一个新闻服务器, 就不需要再安装新闻传输代理程序。只需用新闻阅读器来访问此服务器(参见前面章节中关于 NNTPSERVER 的部分)即可。如果使用因特网服务供应商(ISP), 这些 ISP 都有自己的新闻服务器, 能用新闻阅读器 tm 或 tin 连接到这些远程服务器。但是要记住 tm 和 tin 将花费一些时间去下载选定的新闻组上的所有文章, 以及下载新闻组上所有的更新信息。

可以用新闻传输代理软件来建立一个本地网络新闻组, 以供系统用户和本地网络上的其他用户来访问。能安装 NNTP 或 Cnews 来管理本地新闻组。这样, 系统的用户就可以发表文章和阅读本地新闻。虽然 NNTP 和 Cnews 更适合局域网, 但是也可以使用 INN。

6.3.4 INN

在 Caldera CDROM 中的 InterNetNews(INN)新闻传输代理软件, 提供了新闻服务器的所有功能。能使用 glint 或 rpm 从 Caldera CDROM 中安装它, 或从 Linux 的 FTP 站点上下载, 这样的 FTP 站点很多, 如 sunsite.unc.edu 等。

当用 rpm 安装时, 记住要选用 INN 软件的 RedHat 版本。下例用 rpm 安装 INN:

```
$rpm -i inn-3.6.1.i386.rpm
```

通过此命令可以安装 INN 程序, 同时在相应的目录下安装配置文件。

在 /usr/doc 目录下, 可以看到一个关于 HOWTO 文件的目录。INN 被配置为启动系统时自动启动。INN 的主要程序是 innd, d 代表这是一个守护进程, 当系统启动时, 此进程便被启动, 并在用户工作时始终在后台运行。

大部分 INN 配置文件放在 /etc/news 目录下。在此目录下, 文件 inn.conf 用来为 INN

设置选项, `host.nttp` 用来控制对系统新闻的访问。可以按需要编辑任意的配置文件。`man` 帮助页中有关于所有的 INN 程序和大多数配置文件, 如 `innd`、`inn.conf`、`rnews` 和 `host.nttp`。正确配置 INN 是一个复杂且费时的过程, 一定要仔细地阅读参考手册及光盘上的 `/docs/HTML/news.html` 页面和 HOWTO 文件。

6.3.5 trn 新闻阅读器

用流行的 `trn` 新闻阅读器, 可以浏览一系列新闻组, 从中可以选择一个新闻组以阅读其中的文章。`trn` 具有许多强大的功能, 如可以在一组文章中进行模式搜索。

`trn` 新闻阅读器是 `rn` 的增强版本, 它允许用户通过主题、文章或线索来显示和搜索文章。

`trn` 中的 `t` 代表线索。线索是文章之间的连接, 表示文章之间的关系, 如文章和前面发表的文章之间是同一主题, 或此文章是前文的后续文章等。此外, 用户也可以使用标准的 `rn` 命令从一篇文章切换到下一篇。

`trn` 具有特殊的界面, 被称为选择器, 在其中可以通过线索来切换文章。例如, 如果读一篇关于特定主题的文章, 当输入 `n` 命令时, 便切换到此主题的下一篇文章, 而不像通常的切换到下一篇顺序发表的文章。除了按发表文章的顺序在文章中切换外, 还可以用不同的线索或不同主题来切换不同的文章。对于文章和它的后续文章也是如此。一篇文章和它的后续文章之间用线索连接, 当阅读一篇文章时, 用 `n` 命令将从此文章中切换到它的第一篇后续文章, 而不是它的下一篇顺序发表的文章。这样, 用 `n` 命令使用线索就可以阅读一篇文章和它的所有后续文章, 而不用再一一搜索。

`trn` 在两个列表中操作: 新闻组列表和文章列表。

当第一次执行 `trn` 时, 可以从新闻组列表中选择新闻组。也可以使用命令在新闻组列表中上下切换, 一旦找到想看的, 选中它, 就可以选择和阅读其中的文章。当读完后, 还可以离开此新闻组, 从新闻组列表中再选择其他感兴趣的新闻组来阅读。可以用 `trn` 的选择器来显示、组织和切换文章, 也可以使用标准的 `rn` 命令来管理文章列表。

`trn` 新闻阅读器通过是否包含未读的新闻来区分新闻组。能用 `trn` 的特殊命令来寻找那些只包含未读新闻的新闻组, 未读新闻是指新闻组中未读过的文章。`trn` 通过放在用户的帐号下的 `.newsrsc` 文件来跟踪文章的已读或未读状态。每个用户都有自己的 `.newsrsc` 文件, 它包含了 Usenet 服务器提供的所有的新闻组, 每一项用来跟踪其中是否有未读或已读的新闻。

`trn` 新闻阅读器允许用户订阅一个新闻组或取消订阅。当第一次运行 `trn` 时, 将获得 Usenet 服务器上所有新闻组的访问权, 并自动地订阅所有的新闻组。然而, 用户可能对许多新闻组并不感兴趣, 为了不让那些不感兴趣的新闻组扰乱界面, 可以撤消对它们的订阅。如果以后又对某些新闻组感兴趣, 还可以再订阅。在 `.newsrsc` 文件中记录着订阅的新闻组。

6.3.6 新闻组列表

在 Linux 系统提示符后面输入“m”，则启动 m 新闻阅读器。m 将首先显示新闻组标题的一个简短列表。而启动 tm 时，在显示列表前，tm 将首先把新增的新闻组列表和.newsrc 文件中的列表相对照。如果有新增的新闻组未列在文件中，tm 将逐个询问用户是否要订阅此新闻组。在提示下输入“y”表示订阅，输入“n”表示不订阅。

也许需要添加大量新的新闻组。如果不想在启动后进行此项工作，使用 tm 命令时加上-q 参数即可跳过此过程。tm 将直接显示新闻组标题的一个简短列表，跳过增加新闻组的询问过程。

```
$trn -q
```

此后，tm 将检查.newsrc 文件中列出的新闻组中是否有未读的新闻。如果有，将显示这些新闻组标题。新闻组标题中会提示此新闻组中有多少未读新闻。接着 tm 会提示是否要阅读这些文章。如果不想阅读，输入 n 命令则切换到下一个新闻组。输入 p 命令可以退回前一个新闻组。

在提示符后输入“+”命令，可以进入 tm 选择器，其中列出新闻组中的文章。用户能从中选择想显示的文章。如果输入“y”，将显示新闻组中的第一篇文章。接着会提示用户是阅读下一篇文章还是退回到新闻组列表。在提示符后输入“q”，即可离开此新闻组返回到新闻组列表。

下例中，用户首先进入 tm 界面，显示一系列新闻组标题。提示第一个新闻组，用 n 命令跳过此新闻组。下一提示中，用户输入一个“+”命令，则显示新闻组 comp.os.Linux.misc 中的文章。

```
$trn
comp.a.language          3 articles
comp.os.Linux.misc       1 article
rec.arts.movies          7 articles
rec.fool.recipes         245 articles
sci.physics.fusion       32 articles
talk.politics.theory     126 articles
etc.

===== 3 unread articles in comp.ai.language - read now ? [+ynq] n
===== 1 unread article in comp.os.Linux.misc - read now ?[ynq]+
```

tm 有大量在新闻组列表中切换的命令。使用这些命令能切换到第一个或最后一个新闻组，切换到上一个或下一个新闻组，或是直接移动到特定的新闻组。例如，“\$”将使用户直接切换到新闻组列表的末尾。用来区分已读和未读的新闻组的命令也不少。“^”将使用户切换到第一个包含未读新闻的新闻组，而数字 1 将使用户切换到所有新闻组栏目的第一个栏目。小写字母 n 和 p 将使用户切换到下一个和上一个包含未读新闻的新闻组。大写的 N 和 P 将使用户切换到新闻组列表中的下一个新闻组和上一个新闻组。

当第一次运行 **tm** 时，新闻组中将包含大量未读的文章。可以使用一个简单的命令把它们标为已读，而不用一一阅读。在新闻组的提示符下输入命令 **c**，则把该新闻组中所有的文章标为已读。此时，用于选择未读新闻组的命令 **n** 和 **p** 不可用，直到有新的文章加入到新闻组中才可使用。

通常用户都知道要访问的新闻组的名字，可以不必使用 **n** 和 **p** 命令逐个切换，而使用模式查找直接切换到想访问的新闻组。模式查找命令使得 **tm** 在定位新闻组方面的功能更强大。在命令提示符之后，键入 “/” 和要查找的模式。“/” 表示在新闻组列表中向前搜索，“?” 表示向后搜索。下例中，用户搜索新闻组 **food.recipes**。

```
$trn
comp.a.language          3 articles
comp.os.Linux.misc       1 article
rec.arts.movies          7 articles
rec.fool.recipes         245 articles
sci.physics.fusion       32 articles
talk.politics.theory     126 articles
etc.

===== 3 unread articles in comp.ai.language - read now ? [+ynq] n
/food.recipes
Searching...
===== 245 unread articles in rec.food.recipes - read now? [+ynq] +
```

也可以通过使用新闻组的全名的方式来定位。命令 **g** 加上新闻组的名字将定位到此新闻组。

```
===== 3 unread articles in comp.ai.language - read now ? [+ynq]
g rec.food.recipes
Searching...
===== 245 unread articles in rec.food.recipes - read now? [+ynq]
```

tm 列出和查找的命令只是针对于已订阅的新闻组。用命令 **l** 能列出和查找未订阅的新闻组。单独使用命令 **l** 可以列出所有未订阅的新闻组。在 **l** 后加上一个模式，则查找包含此模式的所有未订阅的新闻组。例如，**ltrek** 将查找未订阅的新闻组中包含 “**trek**” 的新闻组。

可以使用命令 **a** 来订阅一个新闻组，用 **a** 加上要订阅的新闻组名字。还可以使用命令 **u** 来取消订阅。例如，**u rec.foods.recipes** 将取消对此新闻组的订阅。取消后，就不能再用查找命令 “/” 和 **g** 来选定它。当然，可以用命令 **l** 来定位它：**l rec.foods.recipes**。用命令：**a rec.foods.recipes** 可以再次订阅此新闻组。

6.3.7 trn 选择器

前面已经提到，在 **tm** 的提示符后输入 “+”，就可以进入选择器。在选择器中可以

使用线索。选择器的界面中包括每个文章的作者、关联的线索数和主题。任何后续的文章前面加有“>”符号。文章是按照它所属的线索来分类的。每一个线索的第一篇文章前加有一个 ID，ID 包括一个小写的字母或单个数字，从字母 a 开始。下例是一个 tm 选择器：

```
rec.food.recipes                258 articles (moderated)

a   Dylan Chris                 1   Fruit Salad
b   Cecelia Petersen            1   Fudge Cake
d   Richard Leland              2   Chocolate News
    Larisa@atlash
    Aleina Petersen             1   >White chocolate
    Maryann Price               1   >Chocolate Fudge
    mark@pacific                1   >
    Justin G.                   1   >Chocolate
e   George Petersen             1   Apple Muffins
f   Marylou Carrion             1   REQUEST: romantic dinners
g   Valerie Fuller              1   REQUEST: Dehydrated Goodies
i   Carolyn Blacklock           1   REQUEST: Devonshire Cream
    Bill Bode                   1   >
j   Bonnie Matoza               1   Sauces
l   Gabriel Matoza              1   Passion Fruit
o   Ken Blacklock               1   REQUEST: blackened (red)fish
    augie@napa                  3   >blackened (red)fish
    John Carrion
    Anntoinnete

-- Select threads (date order) - 24%
```

上例是进入选择器后显示的第一屏，第一篇文章前加有字母 a。按下空格或“>”键即可显示文章的下一屏。按一下“<”键即可显示前一屏。从上例开始显示下一屏，线索仍然从 a 开始显示。每一个线索前的 ID 在一屏中是唯一的，它们只是用来标识屏幕中的文章线索。

为了阅读一篇文章，首先应选择文章的线索，然后在选择器中显示此文章。可以按下和 ID 相应的键来选择一个线索。例如，按下 d 即可选择第一行前加有 d 的线索。在上例中，如果选定一个线索，则在其 ID 之前会显示一个符号“+”。选定一个线索后，按下 Enter 键或大写字母 Z 即可显示该线索的文章。

首先显示的是此线索的第一篇文章。按 n 键可以使切换到该线索的下一篇文章。找到需要的文章后，用标准的 tm 命令就能显示该文章以阅读它。如果文章超过一屏，按空格键可显示下一屏。

任何时候都可以通过按“+”键返回到选择器。当不想阅读一个选定线索中的文章时，需要删除对此线索的选定。通过按两次和 ID 所对应的字母键，可以删除选定。当第一次按下相应的字母键，在 ID 前将显示一个符号“+”。再按一次，则符号“+”消失，代表删除选定。

也可以用光标的命令来选定一个线索。当初次进入选择器屏幕时，光标在第一个线索的 **a** 的前面。用命令 **n** 和 **p** 或箭头键，可以切换到下一个或退回到前一个线索。用 **n** 命令或下箭头切换到下一个线索，**p** 命令或上箭头退回到前一个线索。要显示一个特定线索中的文章，只需把光标移到此线索的 ID，然后按 **Enter** 或 **Z** 键即可。

如果想一次阅读多个线索，通过按下和每个线索的 ID 相应的键，就可以给每个想阅读的线索做一个标记。例如想选定 ID 为 **b** 的线索，按 **b** 键即可。切换到下一个需要的线索，重复同一操作。在每个选择的线索前都显示一个“+”。也可以从一屏移到下一屏，选择需要的线索。全部选定后就可以显示这些线索的文章。

6.3.8 选择器的显示模式

选择器可以按三种显示模式来显示文章，这三种模式为：文章、主题和线索。

输入命令 **S** 后，输入 **a** 则选定文章模式，输入 **s** 则选定主题模式，输入 **t** 则选定线索模式。在不同模式下，按“=”键可以向后和向前转换。

模式也可以看作是一种线索。当使用主题模式时，文章按照主题项进行分组。主题是在用户在文章标题的主题区域中输入的内容。有相同主题项的文章被放在同一线索下。主题分组只限于那些有相同主题项的文章。文章的主题稍有不同便不会被放在一起。

当使用线索模式时，文章和它的后续文章以及有相同主题的文章被放在一起。后续文章前带有符号“>”。线索模式和主题模式的不同之处在于线索模式把文章的所有后续文章放在一起，尽管一些后续文章可能有不同的主题。文章模式不显示线索，在此模式下，文章只是按发表的顺序排列，每篇文章带有自己的 ID。

根据选用的显示模式不同，选择器的屏幕显示也将不同。在主题模式下，选择器显示每篇文章的作者，并把它们按照主题分类。首先是主题下的第一位作者，然后是此主题下的文章数，接着是主题。主题只在第一篇文章后显示一次。在主题模式下，ID 代表一个主题，不是某一篇文章。在屏幕上，ID 只被放在不同主题分组的开始处。按下和 ID 相应的字母键，即可选定一个主题。主题模式提供一种简单的方法来访问不同主题的文章。同时主题标题的列表也是对新闻组中讨论的主题的总结。

下例中，选择器是在主题模式下。注意 **d** 项指定的主题为 **Chocolate News**。在此主题下有两篇文章，一篇的作者是 **Richard Leland**，另一篇的作者是 **Larisa@atlash**。项 **i** 也有两篇文章，只是第二篇是一篇后续文章，用符号“>”说明。项 **u** 说明主题下有三篇文章，都是后续文章。注意它实际是项 **o** 的后续文章，虽然项 **o** 有不同的主题。

```
rec.food.recipes                258 articles(moderated)

a  Dylan Chris                  1  Fruit Salad
b  Cecelia Petersen             1  Fudge Cake
d  Richard Leland               2  Chocolate News
   Larisa@atlash
e  George Petersen              1  Apple Muffins
f  Marylou Carrion              1  REQUEST: romantic dinners
```

```

g  Valerie Fuller          1  REQUEST: Dehydrated Goodies
i  Carolyn Blacklock      1  REQUEST: Devonshire Cream
   Bill Bode              1  >
j  Bonnie Matoza         1  Sauces
l  Gabriel Matoza        1  Passion Fruit
o  Ken Blacklock         1  REQUEST: blackened (red)fish
r  dylan@sf              1  REQUEST: Cheese Toast
s  Penny Bode            1  REQUEST: Sausage Recipes
t  gloria@stlake        1  Biscuit Recipe
u  augie@napa            3  >blackened (red)fish entree
   John Carrion
   Anntoinnete
v  John Gunther

```

```

-- Select threads (date order) - 24%
Selector mode: threads, Subjects, Articles? [tsa] s

```

在线索模式中，选择器显示的文章是通过后续连接或是通过主题连接的。也就是说，在线索模式下，无论通过何种方式互相连接的文章均被分在一组。后续文章被列在原始文章的后面，且后续文章带有符号“>”。这种分类方式就是线索。用线索模式可以方便地访问一篇文章和它所有的后续文章，以查看所有的讨论和意见。通过主题连接的文章也处在同一线索中，并和它们的后续文章放在一起。每个线索有自己的 ID，按下和 ID 相应的字母键即可选择一个线索。

下例中，选择器是在线索模式下。注意屏幕的显示和主题模式下的显示有所不同。项 d 包括同主题的文章，也包括后续文章，共有六篇。前两篇有相同的主题，后四篇是它们的后续文章，带有符号“>”。大多数的后续文章的主题是不同的，其中的两个，作者是 Maryann Price 和 mark@pacific 的文章有同一主题。项 o 代表一个线索，它的第一篇是 Ken Blacklock 的文章，还包括它的三篇后续文章：augie@napa、John Carrion 和 Anntoinette。虽然这三篇文章是同一线索中不同的文章，但它们有相同的主题。

```

rec.food.recipes          258 articles(moderated)

a  Dylan Chris           1  Fruit Salad
b  Cecelia Petersen      1  Fudge Cake
d  Richard Leland        2  Chocolate News
   Larisa@atlash
   Aleina Petersen       1  >White chocolate
   Maryann Price         2  >Chocolate Fudge
   mark@pacific
   Justin G.             1  >Chocolate
e  George Petersen       1  Apple Muffins
f  Marylou Carrion       1  REQUEST: romantic dinners
g  Valerie Fuller        1  REQUEST: Dehydrated Goodies
i  Carolyn Blacklock     1  REQUEST: Devonshire Cream

```

```

      Bill Bode                1  >
j    Bonnie Matoza           1  Sauces
l    Gabriel Matoza           1  Passion Fruit
o    Ken Blacklock            1  REQUEST: blackened (red)fish
      augie@napa              3  >blackened (red)fish
      John Carrion
      Anntoinnete

```

```
-- Select threads (date order) - 24%
```

```
Selector mode: threads, Subjects, Articles? [tsa] t
```

下例中，文章是按照发表的顺序排列的，每篇文章有自己的 ID。注意项 d 和项 e 有同一主题。在文章模式中没有线索，文章和它们的后续文章是分开排列的。例如，虽然 **augie@napa** 的文章是 **Ken Blacklock** 的文章的后续，但它们分布在不同的部分，且有各自的 ID——r 和 v。

```

rec.food.recipes                258 articles(moderated)

a    Dylan Chris              Fruit Salad
b    Cecelia Petersen         Fudge Cake
d    Richard Leland           Chocolate News
e    Larisa@atlash            Chocolate News
f    George Petersen          Apple Muffins
g    Marylou Carrion          REQUEST: romantic dinners
i    Valerie Fuller           REQUEST: Dehydrated Goodies
j    Carolyn Blacklock        REQUEST: Devonshire Cream
l    Bonnie Matoza            Sauces
o    Gabriel Matoza           Passion Fruit
r    Ken Blacklock            REQUEST: blackened (red)fish
s    dylan@sf                 REQUEST: Cheese Toast
t    Penny Bode               REQUEST: Sausage Recipes
u    gloria@stlake           Biscuit Recipe
v    augie@napa               >blackened (red)fish
w    John Gunther             Oatmeal Cookies
x    Margaret                 REQUEST: Potato Salad
y    Frank Moitoza            REQUEST: Sesame Chicken
z    maryann@sebast           >Summer desserts

```

```
-- Select threads (date order) - 24%
```

```
Selector mode: threads, Subjects, Articles? [tsa] a
```

6.3.9 trn 的使用

下面介绍 tm 的使用。

6.3.9.1 显示文章: trn 线索树

当选定一篇文章时,便显示文章标题、文章的第一屏和一个 more 提示行。文章被一屏一屏地显示,就像用 more 命令显示文件一样。按空格键即可显示下一屏,按 b 键可以退回到前一屏,用命令 q 可以中止显示并退出。还可以在文章中进行模式查找。用命令 g 加上要查找的模式即可以查到此模式在所有文章中第一次出现的地方。用命令 G 可以重复此查找。

```
rec.food.rcipes(moderated) #7155 (229 more)
From : richpete@garnet.berkeley.edu (Richard Petersen)
[1] Spelling Cookies
Followup -To: poster
Date: Mon Dec 20 04:48:01 PST 1995
Organization: University of California, Berkeley
Lines: 43
```

Spelling Cookies

INGREDIENTS

```
1 cup flour
2 teaspoons single acting baking powder, or 1 teaspoon double acting
baking powder,
    or 1 teaspoon baking soda
1/2 teaspoon nutmeg
1/4 teaspoon cinnamon
3/4 cup butter ( or 1 1/2 sticks)
1 cup brown or dark brown sugar
1/2 cup regular sugar
1 egg
1 teaspoon vanilla extract
1/4 cup milk
--More--(44%)
```

在上面显示的状态下按空格键可以显示下面的信息:

```
rec.food.rcipes(moderated) #7155 (229 more)
[1] Spelling Cookies

12 ounces semisweet chocolate chips
pecan or walnut bits
```

3/4 cup wheat germ
2 3/4 cups rolled oats (Old Fashioned Quaker Oats)

STEPS

1. Mix brown and regular sugars together
2. cream butter, then mix in sugars. Then mix in egg, vanilla extract, and milk until creamy.
3. Separately mix flower, cinnamon, nutmeg, baking powder (or baking soda), salt (if wanted)
4. Mix in flower concoction into butter/sugar concoction until creamy.
5. With spatula add in wheat germ, then nuts, then chocolate chips, then oats.
6. Preheat oven to about 300 degrees. Spoon out onto cookie sheets. Cook for about 20 minutes or so. Watch carefully. When top of cookies begin to brown they are done.
7. Eat immediately or refrigerate. Aging only improves taste.
8. These are called spelling cookies because when I gave the recipe to a friend in my class he daughter found so many spelling mistakes that she called them spelling cookies.
9. Eat at you own risk. Gook luck.

End of article 7155(of 7158) - what next ? [npq]+

文章显示结束时, 会出现一个提示, 问用户下一步要做什么。可选择的命令 n、p 和 q 在方括号中。按 n 将显示新闻组中的下一篇文章, 按 p 显示前一篇文章。按 q 将退回到新闻组列表中。如下例, 按 “+” 可以回到选择器中:

End of article 7155(of 7158) - what next ? [npq]+

当显示一个线索的第一篇文章时, 屏幕的右上角会显示一个线索树。线索树代表一个线索中文章之间的联系。每一篇未读的文章被一个数字代表, 该数字从 1 开始, 位于一个方括号中。每个数字之间用线连接。当前正在显示的文章的代表数字会增亮显示。当读完一篇文章, 切换到下一篇文章时, 读完的文章的代表数字会用圆括号括起来, 而下一篇文章的数字会增亮显示。

线索能标明文章之间的关系。线索中的第一篇文章位于线索树的左上角。树的分支位于它的右下方, 下一列是后续文章。所有的后续文章用线连接起来。例如, 第一篇文章的后续文章的代表数用括号括起来, 并放在第一篇文章的代表数的右边作为新的一列。文章数和后续文章数之间用线连接。其他的后续文章被放在第一篇后续文章所在列的下面, 每篇文章之间也用线连接。最后一篇后续文章用斜线连接。

任何一个后续文章可能有不止一篇的后续文章。则所有的后续文章被放在同一列中, 因此, 一个线索树可以向水平和垂直方向扩展。水平方向是后续文章和后续文章的后续文章。垂直方向是一篇文章的所有后续文章。

同第一篇文章有相同主题的文章被列在第一篇文章的下面。它们之间没有用线连接。

线索树中的代表文章的数字用来表明是否有文章共享主题。第一篇文章用数字 1 来代

表,表明是此线索的初始文章。有相同主题的文章有相同的代表数字。其他代表数字也为1的文章和第一篇文章有相同的主题。在同一线索中的有不同主题的文章有不同的代表数字。随后的和前面文章有相同主题的文章使用前面文章的代表数字。第一篇和前面文章有不同主题的文章用数字2来代表,其他代表数字为2的文章则和该文章有相同的主题。在线索中遇到一个不同的主题,则代表数字加1。这种记数方式使用户能识别线索中不同主题的部分。另外,线索中可能包含具有用户不感兴趣的主题的文章,能很容易的识别出来,并不阅读它们。

可以在线索树中从一篇文章切换到另一篇,直接显示想看的文章,而不用显示中间的文章。还可以在线索树中向后切换。能用箭头在线索树中上下切换,切换到的数字便会增亮。例如,用下箭头从当前的文章切换到下一篇,下一篇的数字增亮。按右箭头切换到右边的数字。按左箭头切换到左边的数字,按上箭头能向上切换。当前增亮的数字所代表的文章被显示出来。当在线索树中切换到不同的数字时,显示不同的文章。

6.3.9.2 保存文章

用户能保存阅读的文章。显示文章时,输入命令a和文件名。如果文件还未保存,系统将提示用户是否想以邮件格式保存文件。输入“y”,文件将保存为邮件格式,文章将作为邮箱中的一个消息。能用mailx命令和-f参数来读取存在文件中的文章。按Enter键,则文件被保存为文本格式。

```
rec.food.recipes(moderated) #7155 (229 more)
From : richpete@garnet.berkeley.edu (Richard Petersen)
[2] Spelling Cookies
Followup -To: poster
Date: Mon Dec 20 04:48:01 PST 1995
Organization: University of California, Berkeley
Lines: 43

Spelling Cookies

                                INGREDIENTS

1 cup flour
2 teaspoons single acting baking powder, or 1 teaspoon double acting
baking powder,
    or 1 teaspoon baking soda
1/2 teaspoon nutmeg
1/4 teaspoon cinnamon
3/4 cup butter ( or 1 1/2 sticks)
1 cup brown or dark brown sugar
1/2 cup regular sugar
1 egg
1 teaspoon vanilla extract
1/4 cup milk

End of article 7155( of 7158) - what next ? [npq] s spellcookies
```

```
File /h/garnet_d/richpete/News/spellcookies doesn't exist --
  use mailbox format ? [ynq] y
Saved to mailbox /h/garnet_d/richpete/News/spellcookies

End of article 7155(of 7158) - what next ? [npq]
```

如果把文章保存到一个已存在的文件中，则此文章将被加到文件的后面。如果存到一个邮件格式的文件中，新加的文章将成为一个新的消息。当多篇文章保存在同一个文件中时，把此文件存为邮件格式有许多好处。可以通过邮件方便地存取特定的文章，也可以用 R 命令答复文章的作者，还可以很容易地把文章合并到给其他用户的消息中。

6.3.9.3 在trn中回复文章

可以通过两种方式回复一篇特定的文章：发表一篇后续文章或通过邮件发送给作者一个消息。后续文章是发表在 Usenet 上的，用来回复读过的文章，Usenet 上的其他用户都可以看到此文章。而邮件则不同，邮件是通过 Linux 的 mail 发送的一个私人的消息。用命令 f 和 F 可以发表一篇文章，而用命令 r 和 R 可以发送一个回复消息。

6.3.9.4 发表后续文章

用 tm 新闻阅读器可以发表后续文章。当阅读一篇文章时，使用命令 f 和 F 可以发表后续文章。这两个命令都是调用 Pnews，用它输入和发表文章。F 命令将在用户的后续文章中包括回复文章的正文。显示的正文的每行前有一个“>”。

通过定位要回复的文章并按下 f 命令，可以发表一篇后续文章。进入带有后续文章标题的编辑器中，可以输入文章的正文或编辑文章标题。在退出编辑器之前，将提示用户是发送、忽略、编辑还是显示后续文章。用 send 命令能发表文章。

在所在的新闻组中，后续文章可以把用户带入关于一篇文章的讨论中，用户不仅能阅读文章，还读到别人在后续文章中对它的一些观点。如前所述，可以对一篇后续文章发表后续，来讨论其他用户对文章的观点。发表后续文章时，只需定位到要回复的后续文章，按 f 并输入回复即可。

当回复一篇文章时，在回复中包含文章的原文对理解会有帮助。用 F 命令可以做到这点。被包含的原文的每行前加有“>”。不用在回复文章中包含所有的原文。假设只对文章的一部分进行回复，可以在编辑器中删除原文的其余部分，只保留原文中用户要回复的部分和回复一起发表。甚至可以用编辑器把回复插入到原文中，使回复看起来更像是注释。用命令 F 将使回复文章更清晰易懂且避免重复输入别人的注释。用 F 命令对后续文章发表回复时十分方便，它能自动加入回复的用户的评论，而不用用户亲自输入。

6.3.9.5 寄回复给作者

可以直接给文章的作者寄一封邮件，以代替发表一篇后续文章。寄邮件时用命令 r 和 R。首先定位到要回信的文章，按 r。显示文章标题，接着启动邮件程序，进入一个编辑器中，然后输入消息。文章标题也在编辑器中，可以修改主题或总结行。当离开编辑器时，

系统会提示用户是发送、中止、编辑还是显示消息。使用 `send` 命令可以把此邮件寄给作者。

用 `R` 命令可以把原文包含在消息中。像 `F` 命令一样, `R` 命令也将把原文包含在消息中, 其每行原文前加有 “>”。在编辑器中, 能按需要对原文进行删除、复制或移动操作。

6.3.9.6 标记文章

阅读一篇文章后, `trn` 将不在文章列表中显示此文章标题, 以后也不能用 `trn` 来访问它。如果想以后能阅读此文章, 可以把它标记为未读, 使得 `trn` 在文章列表中继续显示此文章标题。当阅读文章时, 用命令 `m` 可以标记文章为未读。如果下次想阅读此文章, 只需用命令 `M` 即可。

6.3.9.7 文章的选择

能选择一组文章来进行模式查找或引用序号。一个模式查找命令加一个冒号 “:” 和一个 `trn` 命令将使得此命令作用于所有的文章, 而不仅仅是下一篇文章。一个模式查找加一个冒号 “:” 将有效地选定一组文章, 并在其中进行模式查找。例如, 如果想保存所有的主题涉及 “cookies” 的文章, 可以输入命令 “/cookies/:s cookfile”, 如下所示:

```
End of article 6914(of 7158) - what next ? [npq] /cookies/:s
cookfile
Searching ... 7094
File /h/garnet_d/richpete/News/cookfile doesn't exist -
    use mailbox format? [ynq] y
Saved to mailbox /h/garnet_d/richpete/News/cookfile
7128 Appended to mailbox /h/garnet_d/richpete/News/cookfile
7155 Appended to mailbox /h/garnet_d/richpete/News/cookfile
done
End of article 6914 ( of 7158) - what next ? [npq]
```

“^” 是一个特殊的模式符号, 它代表所有的文章。能用下面的命令把所有的文章保存到一个文件中:

```
/^/:s myfile
```

用一个模式和 “=” 命令, 可以显示一系列关于某一特定主题的文章。使用 “=” 命令可以列出每篇文章的序号和主题。当 “=” 命令和模式查找连用时, 能列出所匹配的文章的序号和主题。下例将列出关于 `cookies` 的所有未读文章的序号和主题:

```
/cookies/:s
```

如果想查找已读和未读的所有文章, 需要用命令 `r` 来限定模式。下面的命令将列出所有关于 `cookies` 的文章, 其中包括已读过的文章:

```
/cookies/r:s
```

默认状态下, 模式查找只搜索文章的主题。当使用 `a` 限定模式后, 将搜索文章的全文。下例中将搜索出所有用到 `chocolate` 的食谱, 这种查找的结果通常很长而不便打印:

```
/chocolate/a:=
```

可以通过列出文章相应的序号来引用一组文章，各序号之间用逗号分隔，最后用分号加上一个应用到这些文章的命令。当引用的是连续的文章序号时，可以用一个范围来表示。例如，3-5 代表文章 3、4 和 5。要保存文章 34、17 和 9~12，可以用如下命令：

```
34,17,9-12:s myfile
```

如下的例子中，用户把文章 7155 和 7128 存入文件 `goodcook` 中：

```
7155,7128:s goodcook.
```

在模式和序号引用中，可以扩展使用另外的命令。每个命令之间用冒号分隔。下例中，文章 34、17 和 9~12 分别被保存并打印：

```
34,17,9-12:s myfile: | lp
```

6.3.9.8 trn选项

使用 `trn` 时有一些参数。用 `-n` 参数可以指定用户希望阅读的新闻组或新闻组的类型。用 `-l` 参数可以列出新闻组的标题。下例中，用户列出 `rec.food.recipes` 新闻组的标题：

```
$ trn -l -n rec.food.recipes
```

用 `-q` 参数，可以跳过订阅阶段而直接阅读新闻组：

```
$ trn -q rec.food.recipes
```

第 7 章 无盘工作站的配置

本章将说明如何使用无盘工作站，这种方法可使每台 PC 机在启动时选择使用何种操作系统。这种方法的实现原理就是使用 TCP/IP BootPROM，它可以让使用者在启动时选择不同的引导分区。

7.1 简介

本节所描述的方法是从 1996 夏季开始在美国 CUI, University of Geneva 所使用的。这里有许多服务器(大多是 UNIX 与 Novell)和一些个人计算机，可依用途分为两类：

- 学生使用
- 研究与辅助教学

以下是最终实现的目标：

- 任何一台计算机都可以运行 Linux、DOS、Windows 3.1 或 Windows 95。每次启动时可以选择所需要的操作系统。
- 所有的软件，包含操作系统，可以放在服务器上，便于安装与更新。
- 客户端计算机可以在不对服务器做写入操作的状态下正常工作(为了安全性)，home 目录除外。
- 尽可能减少客户端的设置，客户端可以自动由服务器取得 IP 设置，这些有关信息放在一个小文件中，可供各种操作系统使用。
- 大部分的计算机都各自拥有硬盘，客户端可以靠这种优点来降低网络负载与存放用户的缓存文件。
- 所有的用户都必须注册才能使用计算机。
- 学生所使用的计算机在每次启动时必须清理一次。
- 每台计算机都可以防止病毒感染。

这些限制使人们选用来自 EDV GmbH 的 TCP/IP BootPROM，因为这种 BootPROM 与操作系统无关，因此令人们相当感兴趣。另外，启动硬盘的映像文件，可由其他的程序来取代，这个功能使人们能在操作系统启动前进行许多初始设置。

7.1.1 网 络

Geneva 大学拥有 B 级网域，分成许多子网络。CVI 共使用了四个子网络，其中之一

供学生使用。

开始,个人计算机使用两种网络通信协议:IPX 与 IP。IPX 这一类,由一台 Novell Netware 3 的服务器共用 DOS 与 Windows 的软件;IP 这一类,由一台 Sun 服务器通过 NFS 共用 Linux 软件及个人的分区。

在最新的规划中,已不再使用 IPX 通信协议。服务器改由一台 UNIX 服务器(Linux 或 Sun)担任, Linux 客户端可通过 NFS 访问服务器上的文件,而 DOS 与 Windows 则由 SMB (NetBIOS) over TCP/IP, 来共享软件 and 文件数据。

7.1.2 工作原理

1. 客户端个人计算机启动后,在 TCP/IP BootPROM 获得控制权之前先做自我测试。
2. BootPROM 送出 BOOTP/DHCP 请求而取得 IP 参数。
3. 如果服务器收到个人计算机所送出的要求,就会送回 BOOTP/DHCP 响应,内容包括客户端的 IP 地址、默认通信网关及引导映像文件。
4. BootPROM 由 TFTP 通信协议从服务器下载引导映像文件。
5. 个人计算机通过这个引导映像文件启动,这个引导文件只是单纯的引导程序(并不包括操作系统)。
6. 如果这台计算机是学生的个人计算机,启动程序开始就由 TFTP 通信协议下载一个小的文本文件,其内容是叙述这台计算机的硬盘状态。通过这个小文件,硬盘的分区将重新规划,且 DOS 所使用的分区也被快速格式化。
7. 接着引导程序让用户选择所需的操作系统。
8. 依照用户的选择,新的引导映像文件将从服务器通过 TFTP 通信协议下载。
9. 如果选择使用 Linux,引导映像文件将包含内核加载程序及压缩过的内核,此内核将支持 NFS 及缓冲区文件系统。

在以上步骤中,要注意如下几点:

- IP 地址是通过 BOOTP/DHCP 通信协议由 Novell 服务器所接收。
 - 这个内核可以通过 NFS 安装一个只读的根文件系统。
 - 挂上一块有写入功能的小型内存磁盘(ramdisk)。
 - 如果在硬盘上发现交换区,则为其做好准备并启动它。
 - 如果在硬盘上发现 Linux 分区,挂上它并为其做好缓冲区 NFS 分区。
 - IP 配置一完成,启动所有的服务,也启动 xdm。
 - 要求用户登录到这台工作站。
10. 如果选择使用 DOS 或 Windows,引导映像文件是 FAT16 分区的压缩映像程序。通过 TFTP 下载引导映像文件,并且储存在硬盘中以供未来使用。更严谨的说法是,这个程序可以在以下的情况中执行:
- 这个程序下载一个检查文件(512 字节)以确认所选操作系统的引导映像文件。
 - 如果所需的映像文件并不在硬盘中,或者与检查文件不吻合(可能是由于引导映像文件已被改变或者一个新的版本被安装到服务器上),整个影像文件将通过

TFTP 通信协议来传输。

- 操作系统的映像文件解压缩到第一个 FAT16 的分区，传输速度大约是每秒一兆字节。
- 接着程序跳到所选的操作系统的引导硬盘分区，该分区已存在于本机的硬盘中。

对 DOS 与 Windows 3.1 而言，使用免费的 Microsoft LanManager for DOS 软件。它由三个文件所组成，充当 SMB 客户端。Microsoft LanManager 可通过 DHCP 而支持动态 IP 地址设置。

在登录后，用户会看到 DOS。也可以输入“win”，以启动 Windows 3.1。

请注意 DOS 与 Windows 3.1 就如同安装在客户端一般。Windows 95，也使用 Microsoft SMB 客户端(Client for the Microsoft Network)，也可通过 DHCP 而支持动态 IP 地址设置。

学生所使用的计算机可以直接关掉电源而不会有任何问题，因为硬盘的数据在每次启动时都会重新建立。

对于比较“安全的”计算机(如辅助教学的计算机)，有时会使用上述的方法来启动，有时则直接装入硬盘内的引导映像文件而不再清理硬盘。这样可以允许用户在硬盘中保留数据。

7.2 设 置

首先，应该存在两台计算机：服务器，用一台 UNIX 主机；客户端，一台已经启动 TCP/IP BootPROM 的个人计算机，硬盘内不要存放重要数据。

如果想要测试设置而还没有取得 TCP/IP BootPROM，可以到如下站点：

<http://www.incom.de>

下载该软件的测试版。这个软件可以使计算机就好像是插上 TCP/IP BootPROM 一般。

学生所使用的计算机，设置 BootPROM 由网络启动，不能使用硬盘启动及软盘启动。

教学用计算机，可以设置 BootPROM 由网络启动，也允许使用硬盘与软盘启动；在客户端设置 BootPROM。

服务器端，设置 DHCP 守护进程(使用 Internet Software Consortium 所提供的正式版)。也必须启动 TFTP 守护进程。这份文件将假设用户使用 TCP/IP BootPROM 软件包中的加强型 TFTP 守护进程。

别忘了 BOOTP/DHCP 通信协议被限制在子网域中使用。如果客户端与服务器端不在同一网段，必须在这两网段间建立网关。现在，先假设两台机器在相同的网段中。

先做在所有操作系统上都要做的事，如：

- 设置初始化硬盘的分区及格式化
- 设置操作系统菜单
- 测试启动流程

然后，对每种操作系统，完成下列步骤：

- 建立单独的客户端

- 把它的配置信息放到服务器上。
- 测试远程客户端启动

完成这些事后,以后所添加的客户端只要插上 BootPROM,再加几行配置信息到 DHCP 设置文件中即可。

7.2.1 设置启动流程

在服务器端的/tftpboot目录下,存放下列的引导映像文件(二进制格式):

bpclean 硬盘清理工具

bpmenu TCP/IP BootPROM 菜单程序

bpunzip 硬盘还原工具

bphdboot 传送引导程序到硬盘的映像文件

7.2.1.1 硬盘的初始化与清理

在相同的目录下,建立符号链接到 bpclean(或者直接用拷贝命令),命名为 XXXclean(或者其他可以帮助自己记住这是用来清理客户端机器的名称),并且建立一个 XXXclean.tab 文件,说明客户端所需的硬盘分区及要连接的引导映像文件。例如:对 2GB 硬盘使用下面的方法:

```
# 允许不超过 512 bytes 的说明叙述
# 十六进制请在字首加一个 $ 美元符号
# Part |      | Part
# type | Boot? | Size
   6    Y      +500 MB
  $82    N      +31 MB
  $83    N      -50 MB
    0
# 链接的映像文件
    /tftpboot/XXXmenu
```

分区类型 6 是指 BIGDOS 等,如: DOS FAT-16 由 32MB 到 500MB

分区类型 hex 82 是 LinuxSwap。

分区类型 hex 83 是 LinuxExt2fs。大小如为负值,则表示第三个分区至少要占用 50MB 空间。

分区类型 0 表示是空的(未使用的)分区。

现在, bpclean 在不清除分区数据的情况下,将重新写入主引导记录(master boot record),当然也包括分区表。

7.2.1.2 操作系统菜单

同样地,建立符号链接到 bpmenu(或者直接拷贝),命名为 XXXmenu(或者可以帮助自

已记住这是客户端机器引导菜单的名称), 并且建立 XXXmenu.m 文件, 说明客户端所需的引导菜单。

也可以手动产生这个文件或通过全屏幕菜单编辑器 menuedit.exe 来产生这个文件。例如, 假设用下面这个文件:

```
.CLS 23
.ATT 23
.POS 23 4
.WRT Simple Boot Menu          \
.POS 23 5
.WRT -----                  \
.POS 23 8
.WRT 1. Boot from local hard disk \
.POS 23 10
.WRT 2. Boot DOS and Windows 3   \
.POS 23 12
.WRT 3. Boot Windows 95         \
.POS 23 14
.WRT 4. Boot RedHat Linux       \
.POS 23 17
.WRT Your choice ;              \
.POS 37 17
.KEY 1 ; bphdboot
.KEY 2 ; Linux.PX
.KEY 3 ; win31.P
.KEY 4 ; win95.P
```

7.2.1.3 测试引导程序

在 DHCP 设置文件中加入客户端, 并设置引导映像文件为 /tftpboot/XXXclean。可能需要重新启动 DHCP 服务器使设置生效。

启动客户端, 可以看到由 bpclean 所送出的信息, 这些信息告诉用户它所建立的分区的大小, 然后会看到引导菜单出现在屏幕上。可以用键盘上的 Pause 键来暂停以便读取刚才的信息, 但是这可能会导致 TFTP 的连接超时。如果按下按键 1, 将收到如下信息:

“the boot partition contains not valid boot sector.”

对于未被格式化过的引导分区而言这是正常的。因为现在还未建立任何引导映像文件, 所以其他的按键也会失效。

现在要开始安装各种操作系统。可依需要来决定先安装哪一种。对任何一种操作系统, 都需要先从软盘来做初始化的操作。所以在看到 TCP/IP BootPROM 的标志后应该立刻按空格键换成从软盘启动。

有些操作系统会更新主引导记录, 特别是 Linux 内核引导程序(LILO)。而这种变动会因 bpclean 这个程序而无法完成, 因此最好改变客户端的 DHCP 设置, 使引导映像文件直接跳到 /tftpboot/XXXmenu(不做清理的操作)。别忘了要使改变生效, 必须重新启动 DHCP 服务器。

7.2.2 设置 Linux

下面在客户端建立 RedHat Linux 4.1, 使它支持网络功能。需要准备内核程序源代码及所需要的软件包, 准备未来要加载的点(放在/mnt/tmp 比较好)以及设置 X 服务器等。在 /usr/src/Linux-2.0.27 目录中, 必须有内核 2.0.27 的程序源代码。

现在要加入一些修正文件, 使它升级到 2.0.30, 并且支持 TCP/IP BootPROM 及文件缓存的功能。文件缓存是一个把“未确定的”NFS 文件储存在硬盘中的功能, 它可以降低网络负载。TCP/IP BootPROM 的支持已经由 Marc Vuilleumier Stuckel berg 完成, 并且由 David Clerc 转移到内核 2.0。TCP/IP BootPROM 及文件缓存的支持都已经由它们的作者免费地发布。

值得注意的是 Linux 对 NFS-Root 的支持只能工作在 BOOTP 通信协议下, 不能工作在 DHCP 通信协议。尽管 DHCP 只是 BOOTP 的扩展, Linux 也可以在 DHCP 服务器下工作(如果没有设定 DHCP 服务器拒绝 BOOTP 请求)。

7.2.2.1 产生内核

首先, 进入 /usr/src 目录并加入下列修正文件, 使用命令“patch -p0<修正文件的名称”:

patch-2.0.28 正式的内核更新, 一定要加入这个修正文件。

patch-config-sound 美化声音设置的修正文件, 来自 Unifix Linux 2.0。

patch-PCSP 较大的修正文件, 用来使 PC 机的扬声器模拟声卡, 来自 Unifix Linux 2.0。

patch-bootprom 小的修正文件, 用来产生特别的内核 image, 使它可由 TCP/IP BootPROM 启动。

patch-filecache 小的修正文件, 使内核加入一些特别功能, 如 Unifix 文件缓存功能, 来自 Unifix Linux 2.0。

patch-penguinlogo 小的修正文件, 帮助用户等到 Linux 系统完全装入。

patch-2.0.29 小的正式的内核更新文件, 如果不需要最新版的内核则不用加入它。

patch-2.0.30 正式的内核更新文件。一样, 不一定要加入这项修正(但它会改善 TCP/IP 通信协议)。如果在机器上没有 alpha 的程序源代码(这很有可能发生), 这个修正文件将出现两次显示某些文件不存在的信息。不要紧张, 只要回答跳过这些缺少的文件, 不会发生任何问题。

接着执行 make mrproper 及 make xconfig, 创建内核。记住这将是客户端计算机以 Linux 启动时, 唯一接收到的软件, 所以它必须包含启动整个操作系统所需的内容。

网络功能一定要放在内核中。简而言之, 内核至少需要以下这些功能:

- 支持网络
- 在有 BOOTP 通信协议的情况下, 支持 NFS-Root
- 支持文件缓存

当选择完毕后, 照常地输入“make clean; make dep”接着输入“make zImage”、“make modules”与“make modules_install”。这会花一点时间。现在准备测试新内核, 先用 LILO

安装内核(参考 LILO 文件), 并重新启动计算机(由硬盘启动)。如有任何错误, 修正它们并再试一次。执行 `demod-a` 命令。当完全无误时, 执行 `make bplmage` 命令产生一个供 TCP/IP BootPROM 使用的引导映像文件。

7.2.2.2 把root文件系统移到NFS服务器

服务器要有足够的空间来保存整个 Linux 文件系统(几百兆字节)。建立一个新的目录 `rootfs`, 并在其中建立另一个新目录 `runtime`。使用 `/export/Linux/rootfs/runtime` 目录。使 NFS 的输出信息只有 Linux 客户端可读可写, 并只有 root 用户才有存取权(`anon=0`)。例如, NFS 在 Solaris 下执行, 用以下这个命令:

```
share -F nfs -o rw=pc7971, anon=0 /export/Linux/rootfs/runtime.
```

在 Linux 客户端挂上这个分区, 并用 GNU tar 把整个 Linux 文件系统复制过去(默认是 RedHat Linux)。然后编辑文件 `/export/Linux/rootfs/runtime/etc/fstab`, 并改变根目录的设置, 使根目录符合以 NFS 加载方式取代客户端自己的硬盘。

还要删除(或者至少要改变名称) `/export/Linux/rootfs/runtime/etc/sysconfig/network-scripts/ifcfg-eth0` 文件, 因为网络装置将由 NFS-root 初始化但不能被初始化两次。

在 `/etc/lilo.conf` 文件中复制一行 `append` 命令, 例如: 以 `Linux-nfs` 为名, 并加入以下的参数:

```
append="root=/dev/nfs nfsroot=/export/Linux/rootfs/runtime nfsaddr=用户的-ip: 服务器的-ip: 网关的-ip: 子网掩码: 机器名称"
```

其中, `-ip` 是 Linux 客户端的 IP 地址, 服务器的 `-ip` 是 NFS 服务器的 IP 地址, 网关的 `-ip` 是 Linux 客户端的默认网关, 子网掩码是 Linux 客户端的子网掩码, 而机器名称是 Linux 客户端的主机名称)。再执行一次 `lilo` 命令, 重新启动计算机(仍然是由硬盘启动), 并选择 `Linux-nfs` 启动选项。

计算机将如先前一般启动, 但可能有点慢。如果这里有某些部分不能正常工作, 则重新启用原来本地的 Linux 启动设置并试着修正它。大部分的情况是, NFS root 设置错误。如果有不了解的内容, 看一下 `/usr/src/Linux/Documentation` 文件。

可以再试一次, 这次只用 `append="root=/dev/nfs"` 以确定 Linux 内核可以通过 DHCP/BOOTP 请求而获得所设置的 IP 参数。如果要这样做, 必须在 DHCP 设置文件中加入以下这些选项, 加上机器的硬件及 IP 地址:

```
option subnet-mask 255.255.252.0;
option routers 129.194.68.1;
option root-path "/export/Linux/rootfs";
```

下一步要使系统在只读的 NFS 文件系统下可以工作。

7.2.2.3 产生只读的NFS root文件系统

既然要使 root 文件系统在大部分的 Linux 客户端以只读的方式加载上, 那就要使文件系统有些不同, 因此将需要写入权限的地方以 `ramdisk` 或 `file cache` 取代。把这个文件系统

建立在`/export/Linux/rootfs`目录下，而标准的版本则直接放到`/runtime`目录下。登录到 NFS 服务器并在`/export/Linux/rootfs`目录下，建立以下的目录链接：

```
bin -> cache/bin
dev -> ramdisk/dev
etc -> ramdisk/etc
lib -> cache/lib
root -> ramdisk/root
sbin -> cache/sbin
tmp -> ramdisk/tmp
usr -> cache/usr
var -> ramdisk/var
cache/
bin -> /runtime/bin
lib -> /runtime/lib
sbin -> /runtime/sbin
usr -> /runtime/usr
mnt/
cdrom/
floppy/
tmp/
proc/
ramdisk/
dev -> /runtime/dev
etc -> /runtime/etc
root -> /runtime/root
tmp -> /runtime/tmp
var -> /runtime/var
```

用只读的 NFS 文件系统启动，将加载上一个已初始化过的 `ramdisk` 放在`/ramdisk`目录下。同样地，自己硬盘的分区将被挂在`/cache`目录下给 NFS 缓存使用。如果目标本身是一个子目录，则每一个子目录会转换成一个符号链接并链接到外部文件系统的原始位置。如果不喜欢这样，那么可以在 NFS 服务器上，建立一个符号链接，由`/runtime`目录链接到`/export/Linux/rootfs/runtime`目录。

在只读的客户端加入一些设置的内容是必须的，如此才可以加载上 `ramdisk`，设置文件缓存和检测硬件以改变设置文件。

`runtime/etc/rc.d/rc.ramdisk` 文件可以快速设置并挂上 `ramdisk`。以下是该文件的部分内容：

```
#!/bin/sh
#
# 设置一个 ramdisk 因为 root 通过 NFS 以只读方式挂上
#
modprobe rd
gzip -c -d /runtime/lib/ramdisk.gz | dd of=/dev/ram bs=1k > /dev/null 2>&1
mount -n -t ext2 /dev/ram /ramdisk
```

`runtime/etc/rc.d/rc.sysdetect` 是所有的机器各自独立的设置文件, 包括检测及分配区域硬盘的分区给文件缓存使用。

`runtime/etc/rc.d/init.d/filecache.init` 文件用来启动文件系统。以下是该文件的内容:

```
#!/bin/sh
#
# filecache:      文件缓存:      启动文件缓存(给 NFS root 用)
#
# 启动功能函数库
. /etc/rc.d/init.d/functions

# 看看如何中断函数调用.
case "$1" in
    start)
        if [ -e /cache -a -r /etc/filecache.conf ]; then
            echo -n "Starting NFS filecache: "
            # 把 var 与 tmp 移到区域性的硬盘
            rm -rf /cache/var /cache/tmp
            (cd /ramdisk; tar cf - var tmp) | (cd /cache; tar xf -)
            (cd /ramdisk; rm -rf var tmp; ln -s /cache/var; ln -s /cache/tmp
        )

            chmod 777 /cache/tmp
            # 启动文件缓存
            daemon filecache -d on
            echo ""
            touch /var/lock/subsys/filecache

        fi
        ;;
    stop)
        filecache off
        rm -f /var/lock/subsys/filecache
        ;;
    *)
        echo "*** Usage: filecache.init {start|stop}"
        exit 1
esac

exit 0
```

`runtime/etc/filecache.conf` 是设置缓存的文件。以下是该文件的部分内容:

```
Max 100 MB 50 % #
Cache /runtime /cache
```

前两个文件应该放在 `runtime/etc/rc.d/rc.sysinit` 文件内最前面的位置, 如下所示:

```
# 设置所需的 ramdisk (给 NFS root 只读的机器)
if [ -e /ramdisk -a -r /etc/rc.d/rc.ramdisk ]; then
```

```

        /etc/rc.d/rc.ramdisk
    fi
    # 设置硬件各自独立的参数 (给每个 NFS root 使用)
    if [ -r /etc/rc.d/rc.sysdetect ]; then
        /etc/rc.d/rc.sysdetect
    fi

```

检查 `rc.sysdetect` 文件, 并使它与硬件设备匹配。特别要注意的是, 如果没有使用相同的显示卡及屏幕, 看看它们在 `/proc/pci` 文件中相应的内容, 并依此修改配置文件。在每台机器中, 其文件 `rc.sysdetect` 之中有一段内容为可自己设定的设置文件(例如 `printcap`), 为了使每台机器能独立工作, 需要在服务器的 `dhcpd.conf` 文件中, 用特别的标志 `option-132` 设置每台机器所使用的地址。

在继续安装之前, 至少要先建立几个基本的文件, 如 `runtime/etc/fstab.ref` 和 `runtime/etc/hosts.ref`, 这些将由 `rc.sysdetect` 配置文件在启动时通过检测设置来完成, 为了动态地设置 X Window 服务器, 使用 RedHat 软件包, 必须进行下面的修改: 在 `/usr/X11R6/bin` 及 `/usr/X11R6/lib/X11` 目录下, 把一些相关的链接设置文件及目录改成硬链接。别忘了以后 X Window 服务器的更新安装也要如法炮制。

把 `filecache` 安装在 `runtime/bin` 目录下, 并把它的使用说明 `man page` 安装在 `runtime/usr/man/man8` 目录下, 把 `bootptag` 或 `dhcptag` 安装在 `runtime/usr/local/bin` 目录下, `bootptag.c`(或 `dhcptag.c`)必须放在 `runtime/usr/local/src` 目录下。

`filecache` 是一个发送 BOOTP/DHCP 要求, 并以兼容于 Shell 的标准输出格式响应 BOOTP/DHCP 内容的小程序, 如下所示:

```

bootp_your_ip='129.194.71.32'
bootp_server_ip='129.194.77.31'
bootp_filename='XXXclean'
bootp_subnet_mask='255.255.252.0'
bootp_routers='129.194.68.1'
bootp_domain_name_servers='129.194.69.200 129.194.8.7 129.194.4.32'
bootp_host_name='pc7132'
bootp_domain_name='unige.ch'
bootp_root_path='/export/Linux/rootfs'
bootp_broadcast_address='129.194.71.255'
bootp_nis_domain='cuisunnet.unige.ch'
bootp_nis_servers='129.194.69.200'
bootp_option_132='dufour'

```

标志的名称与 RFC 2132 很相似。如果使用 `dhcptag`, 将得到一些以 `dhcp_` 开头的标志, 这些是 DHCP 协议选项内容; 为了顾及相容性, 其他的信息仍然以 `bootp_` 开头。使用这个程序自动设定 `rc.sysdetect`。最后, 在 `runtime/lib` 目录下安装 `makeramdisk` 脚本文件。将该脚本文件自动产生 `ramdisk` 映像文件。

试着由客户端以可读写 NFS 文件系统启动(仍然由硬盘启动)。

它将检测客户端的个别配置信息, 并产生适当的文件检查 `/etc/fstab`、`/etc/hosts` 和 `/etc/sysconfig/network` 是否设置正确。如果看到的不是正确的结果, 请在单用户工作模式

下重试一次,并找出在 rc.sysdetect 文件中的错误。

等到它可以正常工作,到/lib 目录下执行 makeramdisk 命令。这将会花几秒钟,产生一个给只读的 NFS 客户端用的 ramdisk 映像文件。然后把产生出来的 ramdisk 映像文件放在/lib/ramdisk.gz 中,就可以了。

7.2.2.4 由BootPROM启动

请安装 TCP/IP BootPROM 兼容的内核映像文件(可在/usr/src/Linux/arch/i386/boot/bplImage 目录下找到),即服务器上的/tftpboot/Linux.PX.rc.sysdetect,这个文件用来初始化 Linux 交换区及 Linux 分区。

编辑服务器上的 XXXclean.tab 文件,并改变分区的类型,把 hex 82 改为 hex 28,把 hex 83 改为 hex 38。这是一种未知的分区类型,但是安装程序文件能把它当成预备分区来辨识。在 DHCP 的设置文件中,把引导文件设置为 XXXclean,使它能重新建立分区表。别忘了在修改完设置文件后要重新启动 DHCP 守护进程。

最后,收回公布的可读写的 runtime 目录,并公布只读的/export/Linux/rootfs 目录。重新启动客户端,这一次启动使用 Linux 菜单的选项。这时,用户的机器已是远程引导 Linux 系统了。

7.2.2.5 系统维护与更新

如果以后想更新软件,安装一些错误修正文件或安全性修正文件,请按照下列步骤进行:

1. 不公布 rootfs 目录。
2. 给客户端公布 runtime 目录为可读写。
3. 设置客户端的 nfsroot 目录为 runtime(在/etc/bootptab 中设置)。
4. 启动 Linux 客户端,并安装想安装的内容。不要怕使用 rpm,它可以做得很好(只要在安装任何软件包时,注意一些做过的修正可能要重新做修改)。
5. 当完成后重新换成正常的公布。这是指必须在本地安装的状态下,才能更新基于服务器设置的软件。

7.2.3 设置 DOS 及 Windows 3.1

在客户端,使用所喜欢的 DOS 软盘启动(启动时按空格键取消 BootPROM 启动)。格式化硬盘中的 DOS 分区,并加上/S 参数,使它放入操作系统。

建立一个子目录 DOS,把 DOS 系统命令拷贝进去。安装所喜欢的网络客户端程序 Microsoft LanManager 或 Windows 3.1 等等。使用 DHCP 通信协议设置 IP 地址。

必须恢复原本被 BootPROM 所使用的内存空间(当 DOS 启动后这些内存不再需要)加入下面这一行到 config.sys 文件内的第一行:

```
device=\util\bputil.sys -r
```

如果不想让客户端机器使用无效的登录名称，把 `nobreak.sys` 伪设备驱动程序放在 `config.sys` 文件的开头处，并把以下内容放到 `autoexec.bat` 文件中：

```
rem -- 使用 c: \logged 为一个标记
del c: \logged >nul
: loginneeded
cls
echo Please type in your login name and password
echo.
net logon *
rem -- the login script 已经建立好 c: \logged
if not exist c: \logged goto loginneeded
del c: \logged
rem -- 使 break 功能生效
echo Yes >NOBRK
```

重新启动并在引导菜单中选择 **Boot from local hard-disk** 选项。

7.2.3.1 把设置存放到服务器

回到服务器上，建立一个共享目录 `admin`，用来存放一些系统管理者使用的文件。

如果服务器运行 UNIX，建议用户建立一个 `/tftpboot` 目录到 `admin` 的软链接，这样可以直接由客户端传回引导映像文件。在 `admin` 目录内，建一个子目录 `utils` 并放入如下的工具：

mrzip.exe 用来压缩客户端硬盘映像文件的程序

mrunzip.exe 用来将硬盘映像文件存放到服务器的程序

还需要把一些用来清理硬盘及产生压缩映像的批处理文件放到相同的目录下，这些批处理文件如下所示：

```
@echo off
if "%1"==" " goto error
echo >c: \lanman.dos\lmuser.ini
l: \utils\mrzip l: \tftpboot\%1
goto end
: error
echo Usage: MAKEIMG {image-name-without-extension}
: end
```

现在回到客户端，以下将以 `admin` 用户身份装载硬盘。

接着执行建立的批处理文件，可以按下列的命令执行(不一定要用绝对路径)：

```
L: \util\mrzip L: \tftpboot\win31
```

不久，服务器上的子目录 `/tftpboot` 下将有两个新文件：`win31.imz` 是硬盘内容的压缩映像文件；`win31.chk` 是相关的检查文件。在这目录内，要建立一个 `bpunzip` 的符号链接且命名为 `win31.P`。

7.2.3.2 测试远程启动客户端

现在重新启动客户端并在引导菜单中选择 DOS and Windows 3.1 选项。bpunzip 程序将传送一些它产生映像的表格信息，并从网络下载整个引导映像文件，这将需要一分钟左右的时间。

接着 bpunzip 程序会解压缩这个映像文件到 DOS 分区，并由该分区启动。

下一次重新启动，bpunzip 程序只解压缩这个映像文件，大概在 30 秒内完成。

7.2.3.3 转换设置文件给其他机器

如果要为每台机器修正一些设置(例如默认打印机)，或者如果需要改变一些不能通过 DHCP 通信协议修改的网络设置，可以使用 unzipreg.exe 程序来处理，它必须放到客户端的 autoexec.bat 文件中。这个程序将读取一个由 bpunzip 产生的特殊隐藏文件 BOOTP.ANS，它的内容是由服务器通过 BOOTP/DHCP 通信协议传回的。接着，它将读取第一个参数所指示的文件，通过 BOOTP/DHCP 的响应内容，以“UNZIPREG: 卷标名:”这种格式取代所有的字符串，并把结果写入第二个参数所指示的文件。例如，有个名为 input.bat 文件，内容为：

```
set hostname=UNZIPREG: HOSTNAME:
set domainname=UNZIPREG: DOMAINNAME:
set gateway=UNZIPREG: ROUTER:
set subnetmask=UNZIPREG: SUBNET:
set printer=UNZIPREG: T180:
```

而执行了这个命令：

```
unzipreg input.bat output.bat
```

将得到一个名为 output.bat 的文件，该文件包含如下内容：

```
set hostname=pc7179
set domainname=unige.ch
set gateway=129.194.76.1
set subnetmask=255.255.252.0
set printer=laserwriter1
```

假设 DHCP 设置文件定义这台机器名为 pc7179，域名为 unige.ch，而 option-180 标志是 laserwriter1。

在设置一台客户端机器后，别忘了使用 mrzip 重新建立硬盘映像文件以保存所有改变。

7.2.4 设置 Windows 95

用户需要知道的是，因为微软公司不提供 Windows 95 在实模式下的与 TCP/IP 相兼容的驱动程序，所以无法使用 Microsoft Network Client over TCP/IP，这也就意味着不能用 Samba。

因为当客户端一启动就会锁住一些动态链接文件造成许多程序几乎无法更新, 如果试着去更新它, 则会出现错误。

其他的方面, 安装 Windows 95 的方法就与安装 DOS 一样容易。

7.2.4.1 建立一台独立的客户端

在客户端用 DOS 启动, 如果已经设置好 DOS/Windows 3.1, 那就可以在引导选项中任选一个, 或者用一个软盘启动(按空格键可以跳过 BootPROM)。

使用第一种方法的优点是用户已经有一台可使用网络的客户端, 并且只需要把 Windows 95 的安装盘放到服务器上的某个地方就可以了。

如果由软盘启动, 第一步是安装操作系统, 方法是格式化硬盘的 DOS 分区并使用/S 选项。如果使用 DOS/Windows 3.1 的结合, 那么首先需要清除一些安装 Windows 95 时不需要的文件以及一些在最后的引导映像中不需要的文件(例如, Windows 目录)。

开始安装 Windows 95, 按部就班地把它装在本地的硬盘上。最后, 程序会重新启动客户端计算机, 进行一些设置并再一次重新启动。这几次的重新启动, 必须在引导菜单中选择 Boot from local hard-disk 选项。

等到设置好所需要的驱动程序后, 就可以执行 defrag 命令进行完整的硬盘重组(包含重组未使用的硬盘空间)。

如果用户要使用 BootPROM 所占的内存, 只要在 config.sys 文件的第一行加上如下语句:

```
device=\util\bputil.sys -r
```

bputil 存放在 TCP/IP BootPROM 工具集中。不同于 DOS, 在 Windows 95 中必须避免使用 EMM386。

如果要使用 Windows 95 OSR2, 需要在 MSDOS.SYS 文件中的[Options]段加一行(MSDOS.SYS 是文本文件):

```
AUTOSCAN=0
```

这样可以使 Windows 启动时不自动启动 ScanDisk 程序。

当所有的软件都放在服务器时, 如果要减轻网络及服务器的负担(这将增进系统效率), 可以考虑安装 Shared LAN Cache 软件, 该软件性能优良, 由 Measurement Techniques, Inc 开发, 可以从下面的站点获取:

```
http://www.lancache.com)
```

该软件在客户端计算机上运行, 它会把所有从网络上取得的数据放在区域性的硬盘中作为缓存。

7.2.4.2 把设置文件移到服务器

如果目前服务器端还没有 admin 目录, 那就建立一个。

admin 目录将存放一些系统管理要用的信息。如果服务器是 UNIX 的机器, 把 admin 目录用软链接链接到/tftpboot 子目录下, 这样可以从客户端直接把映像文件放到这里。在

admin 目录内，建立一个子目录/utls 并把下列工具放进去：

mrzip.exe 用来产生客户端硬盘压缩映像的程序

mrunzip.exe 用来把压缩映像文件从服务器回存到客户端硬盘的程序

在客户端打开一个 DOS 窗口，可以把 admin 目录用网络硬盘映射在 L:，并执行下列的命令：

```
L: \util\mrzip L: \tftpboot\win95
```

执行该命令后在服务器的/tftpboot 子目录下产生两个新文件：win95.imz 是客户端硬盘的压缩映像；win95.chk 是检查用的文件。

至此，就完成了 Windows 95 远程启动。

7.2.4.3 测试远程启动客户端

现在重新启动客户端计算机并从引导菜单中选择 Windows 95。

bpunzip 程序将给出一些关于更新映像表的信息，并且从网络下载整个引导映像。这大约需要两分钟的时间。接着 bpunzip 程序将解压缩这个映像文件到 DOS 分区，并启动该文件。就这样，远程启动客户端就绪了。下一次重新启动，bpunzip 程序只解压缩这个映像文件，大概需要 40 秒的时间。

7.2.4.4 其他的设置

与 Windows 3.1 最大的不同是：Windows 95 加入了即插即用的程序代码，即自动检测硬件设备。它本身没有坏处，但问题出在它过于灵敏，而且有时会出错。

如果试着用同样的引导映像去启动另一台客户端机器，将显示一些 Windows 检测到新的硬件设备的信息：一个新的声卡、一块新的硬盘、一块新的网卡或者一只新的鼠标。

造成这样的原因有两个：

1. 这些设备可能使用不同的硬件资源。
2. 这些设备可能向 Windows 95 报告它们的序号(如，运行 Windows 95 的每台机器的网卡，每台机器的以太网地址都各不相同)。

如果即插即用功能失效时，就会出现问题。例如，Windows 95 不能即插即用罗技 PS2 鼠标，更糟的是鼠标消失了。要解决这类问题的方法就是尽可能使所有的计算机有相同的设备。

很不幸，SMC EtherEX 使用的即插即用的程序代码会使计算机死机。唯一的解决办法是使 Windows 95 确认它已经知道有这片网卡，不触动即插即用机制。这就是通过文件 autoexec.bat，自动地把关于这块网卡的设置值加入到有关 Windows 95 的系统注册信息中。要注意的是使用 PCI 的网卡并不需要这么做。

在客户端计算机编辑文件 autoexec.bat 并加入下列几行：

```
rem --- 修正 Windows 系统注册有关信息以避免即插即用功能检测
cls
unzipreg c: \lib\smc.reg c: \temp\smc.reg
regedit /L: c: \win95\system.dat /R: c: \win95\user.dat c: \temp\smc.reg
echo.
```

```
del c: \temp\smc.reg
```

regedit 是标准的 Windows 95 程序。如果在 Windows 95 下执行它，用户可以浏览系统登录的有关信息，如果在 DOS 下执行它，用户可以对系统登录的有关信息做一些简单的修正。

unzipreg.exe 是一个小程序，可以放在用户的路径下的任何地方。它将读取一个由 bpunzip 所产生的隐藏文件 BOOTP.ANS，该文件包含来自服务器的 BOOTP/DHCP 响应的有关信息。然后，它将读取第一个参数的文件(c: \lib\smc.reg)，然后以 BOOTP/DHCP 响应的数据取代文件内所有“UNZIPREG: tagname: ”格式的内容，接着写入第二个参数的文件(c: \temp\smc.reg)。

在 lib 子目录下有 smc.reg 文件，其内容如下：

```
REGEDIT4

[HKEY_LOCAL_MACHINE\Enum\ISAPNP\SMC8416\UNZIPREG: MACID: C0]
"HardwareID"="*SMC8416, ISAPNP\SMC8416"
"HWRevision"="1.0.10"
"DeviceDesc"="SMC EtherEZ (8416)"
"Class"="Net"
"Driver"="Net\\0001"
"CompatibleIDs"="*SMC8416"
"Mfg"="SMC"
"ConfigFlags"=hex: 10, 00, 00, 00

[HKEY_LOCAL_MACHINE\Enum\ISAPNP\SMC8416\UNZIPREG : MACID :
C0\Bindings]
"MSTCP\\0001"=""

[HKEY_LOCAL_MACHINE\Enum\ISAPNP\SMC8416\UNZIPREG : MACID :
C0\LogConfig]
"0000"=hex: 00, 04, 00, 00, 00, 20, 00, 00, 10, 00, 00, 00, 04, 00, 30, 00,
00, 00, 00, 00, a8, 0e, \
00, 00, 20, 30, 00, 00, 02, 00, 00, 00, 01, 00, 0c, 00, 00, 00, 30, 00,
00, 00, 00, 00, e0, ff, 20, \
00, 40, 02, ff, 03, 00, 00, 04, 03, 2c, 00, 00, 00, 01, 00, 00, 00, 31, 00,
14, 00, 00, 00, 00, 00, \
00, 00, 00, 30, 00, 00, 00, 00, 00, e0, ff, ff, 03, 20, 00, 00, 30, 00,
0c, 00, ff, ff, 0f, 00, 00, \
00, 00, 00, 2c, 00, 00, 00, 01, 80, 00, 00, 01, 00, 14, 00, 00, 00, 00,
00, 00, 00, 00, 00, 00, \
00, 00, 00, e0, ff, ff, 00, 80, 00, 00, 00, 00, 0c, 00, ff, 5f, 10, 00,
00, 00, 00, 00, 00, 00, 00, \
00

[HKEY_LOCAL_MACHINE\Enum\ISAPNP\SMC8416\UNZIPREG: MACID: C1]
"HardwareID"="*SMC8416, ISAPNP\SMC8416"
```

```

"HWRevision"="1.0.10"
"DeviceDesc"="SMC EtherEZ (8416)"
"Class"="Net"
"Driver"="Net\\0001"
"CompatibleIDs"="*SMC8416"
"Mfg"="SMC"
"ConfigFlags"=hex: 10, 00, 00, 00

[HKEY_LOCAL_MACHINE\Enum\ISAPNP\SMC8416\UNZIPREG : MACID :
C1\Bindings]
"MSTCP\\0001"=""

[HKEY_LOCAL_MACHINE\Enum\ISAPNP\SMC8416\UNZIPREG:MACID:C1\LogConfig]
"0000"=hex: 00, 04, 00, 00, 00, 20, 00, 00, 10, 00, 00, 00, 04, 00, 00, 00,
00, 00, 00, 00, a8, 0e, \
00, 00, 20, 00, 00, 00, 02, 00, 00, 00, 01, 00, 0c, 00, 00, 00, 00, c0, 00,
00, 00, 00, e0, ff, 20, \
00, 40, 02, ff, 03, 00, 00, 04, 03, 2c, 00, 00, 00, 01, 00, 00, 00, c1, 00,
14, 00, 00, 00, 00, 00, \
00, 00, 00, 00, 00, 00, 30, 00, 00, e0, ff, ff, 00, 20, 00, 00, 00, c0, 0c,
00, ff, ff, 0f, 00, 00, \
00, 00, 00, 2c, 00, 00, 30, 01, 80, 00, 00, 01, 00, 14, 00, 00, 00, c0, 00,
00, 00, 00, 00, 00, 00, \
00, 00, 00, e0, ff, ff, 00, 80, 00, 00, 00, 00, 0c, 00, ff, 5f, 10, 30,
00, 00, 00, 00, 00, 00, 00, \
00

```

这个文件由 Windows 95 的 regedit 产生。

把关于网卡的设置信息输出成一个文件 HKEY_LOCAL_MACHINE/Enum/ISAPNP/SMC8416, 然后用“UNZIPREG:MACID:”取代网卡的硬件地址。当在该文件中执行 unzipreg 命令时, 它会自动地以网卡真实的硬件地址值取代用户所设置的内容。注意这里在 MACID 之后有个数字有时是 C0 有时是 C1。虽然在这个登录文件中放一个不存在的网卡不会出错, 但还是要两项都放进去。

值得一提的是, 可以用同样的方法自动地配置主计算机名称, 当 Windows 95 不能通过 DHCP 取得设置时, 要加入下列几行到 smc.reg 文件中:

```

[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\VxD\VNETSUP]
"ComputerName"="UNZIPREG: HOSTNAME: "

[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\VxD\MSTCP]
"HostName"="UNZIPREG: HOSTNAME: "

[HKEY_LOCAL_MACHINE\System\CurrentControlSet\control\ComputerName\Co
mputerName]
"ComputerName"="UNZIPREG: HOSTNAME: "

```

在改变客户端机器的设置后，千万别忘了用 `mrzip` 命令重新建立硬盘映像，否则所有的设置都会丢失。

如果无法避免 Windows 去检测一些机器上的新硬件，试着在这台机器上重新建立硬盘映像文件，包含这台机器所设置的有关登录的信息也要放入硬盘映像中。如果顺利应该可以解决问题。

如果想得到更多有关设置 Samba 的信息与工具，可以查看下面的站点：

`http://cuiwww.unige.ch/info/pc`

7.3 TCP/IP BootPROM 的相关工具程序

本节将把人们为 TCP/IP BootPROM 所写的程序做一些说明。

7.3.1 MENUEDIT

这是在 DOS 下执行的程序，用来编辑给 TCP/IP BootPROM 的引导菜单的说明文件。该程序很简单。用户按 F1 键可以查看该程序的一些联机帮助信息。

7.3.2 BPHDBOOT

该引导映像(`boot image`)程序将装入硬盘的主引导记录并执行。

该程序对于那些需要在重新启动后才能完成所有设置的操作系统是相当方便的，也可以用在要强迫清理硬盘的时候。

7.3.3 BPCLEAN

该根映像(`boot image`)程序将重新写入硬盘的主引导记录，包含硬盘的分区。此外，该程序可以快速格式化 DOS(FAT16)分区(但无法使它可启动)。因为版权的关系，必须重新规划主引导记录与 FAT16 引导硬盘分区，除了一些远程启动的信息外，它们与标准的分区大致上是一样的。

这个程序下载与这个程序文件名相同的分区表叙述文件，其文件扩展名为 `.tab`。这个文件可以包含空行，注释由记号“#”开始，但不得超过 512 个字符。

前四个不含空白及注释的几行应该包含四个硬盘分区的叙述。

第五个不含空白及注释的那一行应包含下一个要装入的引导映像文件名称。

分区的说明行由空格或 `tab` 键来分开，必须使用如下三种格式：

```
typeboot 1st-cyl    1st-head 1st-sect last-cyl last-head last-sect
```

```
type    boot 1st-cyl  1st-head 1st-sect relative-size
type    boot relative-size
```

第一种格式，精确的分区叙述。

第二种格式，第一个扇区位置有定义，但分区的结束位置是依要求的分区大小自动计算的。

第三种格式，第一个扇区依照前一个分区的结束位置自动推算，且分区的结束位置按所要求的分区大小自动计算。

对于小于 32MB 的 DOS 分区的 **type** 为 4，在 32MB 到 500MB 之间的 DOS 分区的 **type** 为 6。其他的值可以使用 Linux **fdisk help** 找到。

1st-cyl、**1st-head** 及 **1st-sect** 是关系着这个分区的第一个硬盘分区。千万别忘记，柱面及磁头是从 0 开始，而硬盘分区是从 1 开始。

last-cyl、**last-head** 及 **last-sect** 是与这个分区有关的最后一个硬盘分区。分区通常在柱面的边界结束。

分区所占的大小比例可以用下列的方法表示：

+10MB 指分区至少要有 10MB(即 2048 硬盘分区)。

-100MB 指分区必须至少预留 100MB(即 20480 硬盘分区)可用的空间给下一个分区。

+ 30 % 指分区必须在目前的位置，占用至少 30%的空间。

- 70 % 指分区应在当前的位置留下至少 70%的空间给下一个分区。

以大小比例定义分区时通常会在柱面的边界结束，并且除非第一个分硬盘分区的位置是正确的，才会从磁头的边界处开始。

7.3.4 MrZip、MrUnzip 和 BpUnzip

MrZip 是一个在 DOS 下执行的程序，用来建立压缩的 DOS FAT 16 分区。

MrZip 先分析硬盘的使用状况，然后只对使用到的数据做压缩，接着用相当快的(并不是最有效率的)统计的压缩算法来压缩数据。

MrZip 支持 Windows 95 的长文件名，但是不储存文件扩展名为 .SWP 的文件。在各种不同的文件类型中，已经包含许多 **magic** 号。当文件容量的高位序号数字增大时，这个检查码储存在文件的容量的低位序号数字中。如果在建立压缩映像文件前把硬盘的序号归零，则可以使用这个数字来追踪映像文件的更新次数。

因为 MrZip 会直接使用硬盘存取，建议用户在执行前先把硬盘缓冲区的数据存放回硬盘。

MrUnzip 是一个在 DOS 下执行的程序，用来解压缩硬盘映像文件到硬盘中，直接使用硬盘存取。不要与其他任何具有缓冲功能的程序一起使用，因为只这一个程序就已经给 DOS 很大的负担了。

总之，如果要恢复一个无法启动的机器那么它将是很有用的。

BpUnzip 是引导映像程序用来管理压缩硬盘的映像文件。

BpUnzip 先读取分区表并检查第一个 DOS 分区。

接着 BpUnzip 读取第一个未用到的柱面的第一个硬盘分区并检查是否已经有一个映像表。

如果不是这样,或者映像表内容有矛盾,或者两个 Shift 键同时被按下(一种特别的清理信号),这个映像表将会被清除。

如果映像表还没有包含所需的映像,它会通过 TFTP 通信协议装入并加在映像表内。

这个映像文件接着会被解压缩,所有的 magic 号都再确认,并检查解压缩后的数据。

如果解压缩失败,或者检查结果与最新的引导硬盘分区不吻合,则映像文件将会重新装入并修正一次。如果解压缩成功,程序会把控制权交给引导硬盘分区,操作系统就会启动。

如果 BpUnzip 通过文件扩展名.P 装入(如 win95.P),则它会假设 TFTP 服务器在端口 59 有加强型的界面(在正规的端口 69 之外所附加的)。BpUnzip 将通过它以较大的数据包格式装入映像文件,通常是以每个数据包 1 408 字节来取代每个数据包 512 字节(这种使用大数据包转换的技巧有点像 TCP/IP BootPROM 所使用的)。

同理,如果 BpUnzip 是用文件扩展名.G 装入(如 win95.GP),则它先下载一个文件名相同的 GIF 图形文件(如 win95.gif)并在启动时把它显示在屏幕上。这个程序只能在 800×600, 256 色的模式下工作(即使 GIF 图形文件可以更小或者使用更少的颜色)。如果想要看看启动的流程记录,则 GIF 文件显示可以按 Esc 键删除。如果显示卡不是 VESA 兼容型,则这个功能不能使用。

目标分区不必与原分区完全相同,而只要有足够的空间来存放分区中所有使用到的柱面即可。

如果目标分区小于原分区,则硬盘分区表(FAT)将会因此而减小(但柱面大小则不变)。如果目标分区小于原分区,则文件对应表(FAT)将会尽可能地扩大。然而,如果目标分区比原始分区大更多,可能 65 518 个柱面就不够覆盖所有的空间(因为柱面大小是无法改变的)。这时, BpUnzip 将警告用户有一些空间损失掉了。

默认值, BpUnzip 编译成支持 LBA(不超过 1 024 个柱面,但可达 256 个磁头)。某些 BIOS 及某些操作系统较喜欢另一种 NORMAL 模式(可达 4 096 个柱面,但不能超过 64 个磁头);如果需要这种模式,那么把 LBA 的定义从原始程序里改成注释然后重新编译一遍。

7.3.5 Nobreak

nobreak.sys 是一个相当小(大约 350 字节)的驱动程序,它将加在文件 config.sys 中的开始处。

它的目的是保护引导程序,直到使用者登录。驱动程序工作原理是,当产生中断要求时修改键盘的扫描码,直接在 BIOS 层工作。

这个驱动程序必须放在 config.sys 文件中(或者使用 Undocumented DOS 的 devlod 程序)。

第 8 章 DNS 配置

8.1 简介

对新手而言, DNS 是网络管理中比较难于理解的。本节将介绍一些有关内容, 如设立一个简单的 DNS 域名服务器的方法等。至于更高级的设置用户将得阅读“真正的”文件。本节最后将说明所谓“真正的”文件是哪些文件。

开始进行配置 DNS 前, 应该先配置好机器以便可以从远程登录到它, 然后由它对其他机器做远程登录, 并且使所有网络的连接都能成功地进行。尤其是, 应该能够在 127.0.0.1 地址上(本地回送)进行远程登录并且进入用户自己的这台机器。

同时用户也需要正确的/etc/host.conf、/etc/resolv.conf 和/etc/hosts 文件作为一个起始点, 在本节中不解释它们的功能。

如果使用 SLIP 或 PPP, 那么要让它运行。若是它还不能运行, 请阅读前面的有关章节。

“用户的机器”是指用户要在上面设立 DNS 的机器, 而不是在用户网络环境里的其他机器。

在 UNIX 上的名称服务是通过 named 程序来执行的。named 是 BIND(Berkley Internet Name Domain)软件包的一部分, 这个软件包是由 Paul Vixie 为因特网软件集团(Internet Software Consortium)所开发的。

Linux 的大部分发行软件包都包含 named, 而且通常安装在/usr/sbin/named 目录下。如果用户的系统有 named, 那么用户可以直接使用它; 如果用户的系统没有, 那么可以从某个 Linux 的 FTP 站点取得一份二进制文件, 或是从站点ftp://ftp.vix.com/pub/bind中的 release 或者 testing 目录下取得最新的程序源代码, 并判断哪种版本最符合用户的平台。

DNS 是以整个因特网网络为范围的数据库。如果用户放些无用的信息, 用户本人以及其他人都将会从中取出一些无用的信息。

8.2 DNS 的配置

一台缓存专用名称服务器将会为名称查询找出答案并且在下一次用户需要该名称的时候提供答案。

首先用户需要一个/etc/named.boot 文件。当 DNS 守护进程 named 启动时会自动读取这个文件。目前它应该仅包含下面的内容:

```
; Boot file for nicolais caching name server
;
directory /var/named
;
; type          domain                  source file or host
cache           .                      root.cache
primary         0.0.127.in-addr.arpa    pz/127.0.0
```

“directory”指定 named 寻找文件的路径。后面所有的文件都将是相对于此目录的。这个/var/named/root.cache 文件应该包含下面的内容:

```
\&.      518400 NS      D.ROOT-SERVERS.NET.
\&.      518400 NS      E.ROOT-SERVERS.NET.
\&.      518400 NS      I.ROOT-SERVERS.NET.
\&.      518400 NS      F.ROOT-SERVERS.NET.
\&.      518400 NS      G.ROOT-SERVERS.NET.
\&.      518400 NS      A.ROOT-SERVERS.NET.
\&.      518400 NS      H.ROOT-SERVERS.NET.
\&.      518400 NS      B.ROOT-SERVERS.NET.
\&.      518400 NS      C.ROOT-SERVERS.NET.
;
D.ROOT-SERVERS.NET.  3600000 A      128.8.10.90
E.ROOT-SERVERS.NET.  3600000 A      192.203.230.10
I.ROOT-SERVERS.NET.  3600000 A      192.36.148.17
F.ROOT-SERVERS.NET.  3600000 A      192.5.5.241
G.ROOT-SERVERS.NET.  3600000 A      192.112.36.4
A.ROOT-SERVERS.NET.  3600000 A      198.41.0.4
H.ROOT-SERVERS.NET.  3600000 A      128.63.2.53
B.ROOT-SERVERS.NET.  3600000 A      128.9.0.107
C.ROOT-SERVERS.NET.  3600000 A      192.33.4.12
```

这个文件描述了在因特网上的根名称服务器。这会随时间而改变,并且必须加以维护。

在 named 的联机使用手册里有关于这个文件的描述。

接下来,用户需要一份如下所示的/etc/resolv.conf 文件:

```
search subdomain.your-domain.edu your-domain.edu
nameserver 127.0.0.1
```

“search”行指出各个域搜寻的先后顺序。“nameserver”行指出用户的机器可以在哪个地址上找到一台名称服务器,在本例中是用户自己的这台机器,因为用户在它上面执行 named(注意: named 从不读取这个文件,而是使用 named 的名称解析器读取它)。

下面简单介绍这个文件有什么作用:如果某个客户端要寻找 foo,那么首先尝试的是 foo.subdomain.your-domain.edu,接下来是 foo.your-domain.edu,最后则是 foo。如果有某个客户端要寻找 sunsite.unc.edu,那首先尝试的是 sunsite.unc.edu.subdomain.your-domain.edu,然后接下来是 sunsite.unc.edu.your-domain.edu,最后则会是 sunsite.unc.edu。

用户最好别在 `search` 行放太多领域，因为搜寻它们会多花时间。

本例假设用户属于 `subdomain.your-domain.edu` 这个领域，那么用户的机器，可能会称为 `your-machine.subdomain.your-domain.edu`。在 `search` 行里不应该包含用户的 TLD(top level domain, 顶层领域，在这个例子中是 `edu`)。如果用户经常需要连接到在另外一个领域里的主机，那么可以把该领域加进 `search` 行里，如下所示：

```
search subdomain.your-domain.edu. your-domain.edu. other-domain.com.
```

依此类推。很明显，用户得放入真实的域名来取代这些名称。接下来，根据用户 `libc` 版本的不同需要修正 `/etc/nsswitch.conf` 或 `/etc/host.conf` 文件。

- `/etc/nsswitch.conf`

这是个很长的文件，它指出到何处去取得各种不同的数据类型，从什么文件或数据库取得。它的文件头部经常会包含一些有帮助的注释。找出以 “`hosts:`” 作为开头的那一行，应该如下所示：

```
hosts:      files dns
```

如果文件里没有以 “`hosts:`” 作为开头的行，那么把上面这一行放进去。它表示程序应该先在 `/etc/hosts` 文件里搜寻，然后根据 `resolv.conf` 检查 DNS。

- `/etc/hosts.conf`

它可能包含数行，其中应该有一行以 `order` 作为开头，如下所示：

```
order hosts, bind
```

如果文件里没有 “`order`” 行，那么用户应该加入一行。它设置系统，使主机名称解析先在 `/etc/hosts` 里寻找，然后查询名称服务器(在 `resolv.conf` 里用户说明该服务器在 `127.0.0.1`)。有关这个文件的更详尽的信息在 `resolv(8)` 的联机手册中可以找到(执行 `man 8 resolv` 即可)。每个人，特别是 DNS 管理者，都应该阅读这份手册。

8.2.1 启动 `named` 进程

接下来启动 `named`。如果用户使用拨号连接，那么请先连上网络。输入 “`ndc start`” 并且按下回车键。如果它显示 “找不到 `ndc` 命令”，那么试着使用 “`/usr/sbin/ndc start`” 来取代 “`start`”。

然后可以测试用户的设置。在启动 `named` 的时候，如果用户观察一下(使用 `tail-f /var/adm/messages` 命令)系统记录信息文件(通常称为 `/var/adm/messages`)，那么应该会看见如下所示的内容：

```
Jun 30 21: 50: 55 roke named[2258]: starting.  named 4.9.4-REL
Sun Jun 30 21: 29: 0
    3 MET DST 1996   janl@roke.slip.ifi.uio.no: /var/tmp/bind/named
Jun 30 21: 50: 55 roke named[2258]: cache zone "" loaded(serial 0)
Jun 30 21: 50: 55 roke named[2258]: primary zone "0.0.127.in-addr.arpa"
loaded (serial 1)
```

如果有任何关于错误的信息, `named` 将会显示有错误的文件的名称(`named.boot` 和 `root.cache` 两者之一), 杀掉 `named` 进程并回头检查 `named` 的文件。

```
$ nslookup
Default Server:  localhost
Address:  127.0.0.1
```

>

如果以上是用户所得到的响应, 那么它已经能够运行。得到其他任何响应, 都请回头检查每一步。每一次用户改变 `named.boot` 文件之后, 都得使用 `ndc restart` 命令重新启动 `named` 进程。

然后可以输入查询。试寻找某些靠近用户的机器。如寻找 `pat.uio.no`:

```
> pat.uio.no
Server:  localhost
Address:  127.0.0.1

Name:  pat.uio.no
Address:  129.240.2.50
```

`nslookup` 命令会要求用户的 `named` 找寻 `pat.uio.no` 这台机器。然后 `named` 与在用户 `root.cache` 文件里所指定的名称服务器中的一台机器联系, 并且从那里查询它该如何继续查找的路径。在取得结果之前可能得花费一段时间, 因为它查找用户在 `/etc/resolv.conf` 文件里指定的所有领域。

如果用户再查找一次, 那么用户将会得到如下信息:

```
> pat.uio.no
Server:  localhost
Address:  127.0.0.1

Non-authoritative answer:
Name:  pat.uio.no
Address:  129.240.2.50
```



注意: 这次所得到的“Non-authoritative answer:”行代表 `named` 此次并未到网络外去查询, 而是在它的缓存区里查找并且在那里发现答案。但是缓存的有关信息可能会过时。所以它通过“Non-authoritative answer:”来通知用户有这个(很轻微的)危险性存在。当 `nslookup` 显示这是用户第二次查询某台主机时, 这是 `named` 能缓存该项有关信息并且正常运行一个信号。用户可以通过使用 `exit` 命令退出 `nslookup` 程序。

如上所述, 用户应该知道如何设立一个能够缓存的 `named` 系统。

8.2.2 设立简单的领域

本节讲述如何设立用户自己的领域。

首先讲述一些关于 DNS 如何运行的理论知识。了解这些理论知识对用户理解下面的内容有帮助。

DNS 是一个层次式的系统。其顶端写作“.”，读作“root”。在“.”之下有几个顶层领域(TLD)，最广为人知的是 org、com、edu 和 net 这四个。当用户想要找出 prep.ai.mit.edu 的地址时，用户的名称服务器必须找到服务 edu 的一台名称服务器。DNS 会去查询 root.cache 文件，而“.”服务器会给 DNS 一份 edu 服务器列表。以下介绍查询过程：

```
$ nslookup
Default Server:  localhost
Address:  127.0.0.1
```

开始查询某台根服务器：

```
> server c.root-servers.net.
Default Server:  c.root-servers.net
Address:  192.33.4.12
```

设置查询类型为 NS(name server records, 名称服务器记录)：

```
> set q=ns
```

查询关于 edu.的数据：

```
> edu.
```

结尾的“.”在这里非常重要，它表明该服务器所查询的 edu 是在“.”之下的某一个，这能稍微缩小查询的范围。

```
edu      nameserver = A.ROOT-SERVERS.NET
edu      nameserver = H.ROOT-SERVERS.NET
edu      nameserver = B.ROOT-SERVERS.NET
edu      nameserver = C.ROOT-SERVERS.NET
edu      nameserver = D.ROOT-SERVERS.NET
edu      nameserver = E.ROOT-SERVERS.NET
edu      nameserver = I.ROOT-SERVERS.NET
edu      nameserver = F.ROOT-SERVERS.NET
edu      nameserver = G.ROOT-SERVERS.NET
A.ROOT-SERVERS.NET      internet address = 198.41.0.4
H.ROOT-SERVERS.NET      internet address = 128.63.2.53
B.ROOT-SERVERS.NET      internet address = 128.9.0.107
C.ROOT-SERVERS.NET      internet address = 192.33.4.12
D.ROOT-SERVERS.NET      internet address = 128.8.10.90
E.ROOT-SERVERS.NET      internet address = 192.203.230.10
I.ROOT-SERVERS.NET      internet address = 192.36.148.17
F.ROOT-SERVERS.NET      internet address = 192.5.5.241
```

```
G.ROOT-SERVERS.NET      internet address = 192.112.36.4
```

以上信息表明*.root-servers.net 服务于 edu.领域, 所以可以由此继续查询 c.服务器。下面查询哪个名称服务器服务于下一层的域名:

```
mit.edu.:

> mit.edu.
Server:  c.root-servers.net
Address:  192.33.4.12

Non-authoritative answer:
mit.edu nameserver = STRAWB.mit.edu
mit.edu nameserver = W20NS.mit.edu
mit.edu nameserver = BITSY.mit.edu

Authoritative answers can be found from:
STRAWB.mit.edu  internet address = 18.71.0.151
W20NS.mit.edu   internet address = 18.70.0.160
BITSY.mit.edu   internet address = 18.72.0.3
```

Steawb、w20ns 和 bitsy 服务于 mit 领域, 选择其中一个并且查询 ai.mit.edu:

```
> server W20NS.mit.edu.
```

主机名称不分大小写, 以下的数据是屏幕的拷贝:

```
Server:  W20NS.mit.edu
Address:  18.70.0.160

> ai.mit.edu.
Server:  W20NS.mit.edu
Address:  18.70.0.160

Non-authoritative answer:
ai.mit.edu      nameserver = WHEATIES.AI.MIT.EDU
ai.mit.edu      nameserver = ALPHA-BITS.AI.MIT.EDU
ai.mit.edu      nameserver = GRAPE-NUTS.AI.MIT.EDU
ai.mit.edu      nameserver = TRIX.AI.MIT.EDU
ai.mit.edu      nameserver = MUESLI.AI.MIT.EDU

Authoritative answers can be found from:
AI.MIT.EDU      nameserver = WHEATIES.AI.MIT.EDU
AI.MIT.EDU      nameserver = ALPHA-BITS.AI.MIT.EDU
AI.MIT.EDU      nameserver = GRAPE-NUTS.AI.MIT.EDU
AI.MIT.EDU      nameserver = TRIX.AI.MIT.EDU
AI.MIT.EDU      nameserver = MUESLI.AI.MIT.EDU
WHEATIES.AI.MIT.EDU  internet address = 128.52.32.13
WHEATIES.AI.MIT.EDU  internet address = 128.52.35.13
```

```

ALPHA-BITS.AI.MIT.EDU  internet address = 128.52.32.5
ALPHA-BITS.AI.MIT.EDU  internet address = 128.52.37.5
GRAPE-NUTS.AI.MIT.EDU  internet address = 128.52.32.4
GRAPE-NUTS.AI.MIT.EDU  internet address = 128.52.36.4
TRIX.AI.MIT.EDU internet address = 128.52.32.6
TRIX.AI.MIT.EDU internet address = 128.52.38.6
MUESLI.AI.MIT.EDU      internet address = 128.52.32.7
MUESLI.AI.MIT.EDU      internet address = 128.52.39.7

```

可见 `wheaties.ai.mit.edu` 是 `ai.mit.edu` 的一台名称服务器:

```

> server WHEATIES.AI.MIT.EDU.
Default Server:  WHEATIES.AI.MIT.EDU
Addresses:  128.52.32.13, 128.52.35.13

```

下面改变查询的类型。

```

> set q=any
> prep.ai.mit.edu.
Server:  WHEATIES.AI.MIT.EDU
Addresses:  128.52.32.13, 128.52.35.13

```

```

prep.ai.mit.edu CPU = dec/decstation-5000.25  OS = Unix
prep.ai.mit.edu
    inet address = 18.159.0.42, protocol = tcp
    #21 #23 #25 #79
prep.ai.mit.edu preference = 1, mail exchanger = life.ai.mit.edu
prep.ai.mit.edu internet address = 18.159.0.42
ai.mit.edu      nameserver = alpha-bits.ai.mit.edu
ai.mit.edu      nameserver = wheaties.ai.mit.edu
ai.mit.edu      nameserver = grape-nuts.ai.mit.edu
ai.mit.edu      nameserver = mini-wheats.ai.mit.edu
ai.mit.edu      nameserver = trix.ai.mit.edu
ai.mit.edu      nameserver = muesli.ai.mit.edu
ai.mit.edu      nameserver = count-chocula.ai.mit.edu
ai.mit.edu      nameserver = life.ai.mit.edu
ai.mit.edu      nameserver = mintaka.lcs.mit.edu
life.ai.mit.edu internet address = 128.52.32.80
alpha-bits.ai.mit.edu  internet address = 128.52.32.5
wheaties.ai.mit.edu    internet address = 128.52.35.13
wheaties.ai.mit.edu    internet address = 128.52.32.13
grape-nuts.ai.mit.edu  internet address = 128.52.36.4
grape-nuts.ai.mit.edu  internet address = 128.52.32.4
mini-wheats.ai.mit.edu internet address = 128.52.32.11
mini-wheats.ai.mit.edu internet address = 128.52.54.11
mintaka.lcs.mit.edu    internet address = 18.26.0.36

```

从“.”开始连续找出在域名里的下一层名称服务器。如果用户使用自己的 DNS 服务器取代其他所有的服务器，则 `named` 缓存所有这些在为用户寻找这个答案时所找到的有关信息，而且在一段时间内它不必再次查询。

一个相对较少被论及，但是同样重要的领域是 `in-addr.arpa`。它也是分层的树状结构。`in-addr.arpa` 让用户可以在拥有主机地址的时候得知该主机的名称。在这里要注意在 `in-addr.arpa` 领域中 `ip#s` 是以反向顺序书写的。如果用户有某台机器的地址——`192.128.52.43`，那么 `named` 会以 `prep.ai.mit.edu` 例子的方式来处理：找出 `in-addr.arpa` 的服务器，再找出 `192.in-addr.arpa` 的服务器，然后找出 `128.192.in-addr.arpa` 的服务器，接着找出 `52.128.192.in-addr.arpa` 的服务器，最后再找出所需的 `43.52.128.192.in-addr.arpa` 的记录。

现在来定义自己的领域。将创建 `Linux.bogus` 领域并定义其中的机器。使用一个完全是虚拟出来的域名以免扰乱网络。

下面以文件 `named.boot` 里的如下一行开始这个部分的设置：

```
primary          0.0.127.in-addr.arpa          pz/127.0.0
```

请注意在这个文件里的域名结尾处并没有加上“.”符号。第一行把定义 `0.0.127.in-addr.arpa` 的文件命名为 `pz/127.0.0`。这个文件的内容如下：

```
@          IN      SOA      Linux.bogus. hostmaster.Linux.bogus.
(
                                1          : Serial
                                28800     : Refresh
                                7200      : Retry
                                604800    : Expire
                                86400)    : Minimum TTL
                                NS         ns.Linux.bogus.
1          PTR      localhost.
```

请注意在这个文件里所有的完整域名末尾的“.”符号，这与上面提到的 `named.boot` 文件形成对比。有人喜欢以 `$ORIGIN` 命令启始每个区域文件，但这是不必要的。一个区域文件的基点是在 `named.boot` 文件的“`primary`”行里指定的，在这个例子里是 `0.0.127.in-addr.arpa`。

这个区域文件中包含三种资源记录(resource record)：一个 SOA 资源记录，一个 NS 资源记录和一个 PTR 记录。SOA 是授权开始(start of authority)的缩写。“@”是用于标识一个基点，而因为这个文件的“`primary`”行指定 `0.0.127.in-addr.arpa`。所以第一行实际上是：

```
0.0.127.IN-ADDR.ARPA. IN      SOA ...
```

NS 是名称服务器资源记录，它设置 DNS 什么机器是这个领域的名称服务器。而最后的 PTR 记录表明 1(等于是 `1.0.0.127.IN-ADDR.ARPA`，例如 `127.0.0.1`)的名称是 `localhost`。

SOA 这个记录是所有区域文件的序文，在每一个区域文件里都应该有唯一的一个最开头的记录。它描述该区域本台计算机的主机名(一台称为 `Linux.bogus` 的机器)，哪个用户负责其内容(`hostmaster@Linux.bogus`)，这个区域文件是什么版本(`serial: 1`)，以及其他必须做的有关缓存与备份 DNS 服务器的事。其他的项如 `refresh`、`retry`、`expire` 和 `minimum` 可以使用这份说明文件里所用的数字，应该不会有问题。

ns	A	127.0.0.2
mail	A	127.0.0.4

在这个文件里有一种新的资源记录类型,即MX类型,也称为邮件交换器资源记录(Mail eXchanger RR)类型。这种资源记录类型表明邮件系统地址 `someone@Linux.bogus` 的邮件要寄送到哪里,换句话说也就是应该寄送到 `mail.Linux.bogus` 或 `mail.friend.bogus`。在每个机器名称前面的数字是MX资源记录的优先权,数字比较小(10)的资源记录是邮件主要应该寄往的机器。如果失败可以把它寄往数字比较大的机器,如本例中是具有优先权20的 `mail.friend.bogus`。

通过执行 `ndc restart` 命令,重新启动 `named`。以 `nslookup` 检查结果:

```
$ nslookup
> set q=any
> Linux.bogus
Server: localhost
Address: 127.0.0.1

Linux.bogus
    origin = Linux.bogus
    mail addr = hostmaster.Linux.bogus
    serial = 199511301
    refresh = 28800 (8 hours)
    retry   = 7200 (2 hours)
    expire  = 604800 (7 days)
    minimum ttl = 86400 (1 day)
Linux.bogus    nameserver = ns.Linux.bogus
Linux.bogus    nameserver = ns.friend.bogus
Linux.bogus    preference = 10, mail exchanger = mail.Linux.bogus.Linux.bogus
Linux.bogus    preference = 20, mail exchanger = mail.friend.bogus
Linux.bogus    nameserver = ns.Linux.bogus
Linux.bogus    nameserver = ns.friend.bogus
ns.Linux.bogus internet address = 127.0.0.2
mail.Linux.bogus internet address = 127.0.0.4
```

如果小心地检查,用户将会发现一个错误。如下一行:

```
Linux.bogus    preference=10, mail exchanger=mail.Linux.bogus.Linux.bogus
```

全都错了。它应该是:

```
Linux.bogus    preference = 10, mail exchanger = mail.Linux.bogus
```

故意犯了个错误使用户可以借此学习。仔细看看该区域文件会发现如下行:

```
@             MX      10 mail.Linux.bogus      : Primary Mail Exchanger
```

遗漏了一个句点。或是说多了个“Linux.bogus”。在区域文件里如果一个机器名称不是以

句点结尾那么会在其结尾处加入基点。所以不论是：

```
@           MX      10 mail.Linux.bogus.      ; Primary Mail
Exchanger
```

还是：

```
@           MX      10 mail                      ; Primary Mail
Exchanger
```

都是正确的。在一个区域文件里，域名应该不是完全写出来并以“.”结尾，就是不包含“.”符号的形式，而在这种情况下其领域默认为基点。必须强调的是，在 `named.boot` 文件中域名后面不应该有“.”的存在。有很多情况下，因为多了或少了一个“.”而出现问题，使许多人困惑不解。

下面是一份新的区域文件，其中还包含其他信息：

```
;
; Zone file for Linux.bogus
;
; Mandatory minimum for a working domain
;
@      IN      SOA      Linux.bogus. hostmaster.Linux.bogus.
                        199511301      ; serial, todays date +
todays serial #
                        28800           ; refresh, seconds
                        7200            ; retry, seconds
                        604800          ; expire, seconds
                        86400 )         ; minimum, seconds

server      NS      ns              ; Inet Address of name
server      NS      ns.friend.bogus.
server      MX      10 mail          ; Primary Mail Exchanger
server      MX      20 mail.friend.bogus. ; Secondary Mail
Exchanger

localhost   A      127.0.0.1
ns          A      127.0.0.2
mail        A      127.0.0.4
;
; Extras
;
@           TXT      "Linux.Bogus, your DNS consultants"

ns          MX      10 mail
ns          MX      20 mail.friend.bogus.
```

```

                                HINFO  "Pentium" "Linux 1.2"
                                TXT     "RMS"
richard                        CNAME   ns
www                           CNAME   ns

                                A        127.0.0.3
donald                         MX       10 mail
                                MX       20 mail.friend.bogus.
                                HINFO    "i486" "Linux 1.2"
                                TXT      "DEK"

                                MX       10 mail
mail                           MX       20 mail.friend.bogus.
                                HINFO    "386sx" "Linux 1.0.9"

                                A        127.0.0.5
ftp                            MX       10 mail
                                MX       20 mail.friend.bogus.
                                HINFO    "P6" "Linux 1.3.59"

```

用户也许想移动前面三个 A 记录以便让它们邻近于它们其他的相关记录，而不是像上面一样放在最前端。

这里有几个新的资源记录：主机有关信息(HINFO: Host INfOrMation)包括两个部分，第一个部分是机器上的硬件或是中央处理单元，而第二个部分是机器上的软件或是操作系统。

TXT 记录是个自由的文字记录，用户可以用它来做任何想做的事。每台计算机有一个正式名称(CNAME: Canonical NAME)，但可以具有几个名称，这些其他的名称叫作别名。richard 以及 www 都是 ns 的一个别名。很重要的一点是 A、MX、CNAME 和 SOA 记录永远不该引用 CNAME 记录设置的别名，它们只应该引用 A 记录所设置的名称，所以如下的记录是错的：

```
fooobar CNAME richard           ; NO!
```

但如下的记录是对的：

```
fooobar CNAME ns               ; Yes!
```

还有也很重要的一点是，注意正式名称所设置的名称对电子邮递地址而言不是合法主机名称：webmaster@www.Linux.bogus。即使它在用户的系统上可以运行，也很少有电子邮件管理者会去实行这项规则。避免这个问题的方法是使用 A 记录(或者其他的，如 MX 记录)来取代之：

```
www      A      127.0.0.2
```

通过执行 `ndc reload` 命令装入新的数据库，这会使 `named` 再一次读取其文件。

```

$ nslookup
Default Server:  localhost
Address:  127.0.0.1

```

```
> ls -d Linux.bogus
```

这个命令表示列出所有的记录:

```
[localhost]
Linux.bogus.          SOA      Linux.bogus
hostmaster.Linux.bogus. (1995
11301 28800 7200 604800 86400)
Linux.bogus.          NS       ns.Linux.bogus
Linux.bogus.          NS       ns.friend.bogus
Linux.bogus.          MX       10 mail.Linux.bogus
Linux.bogus.          MX       20 mail.friend.bogus
Linux.bogus.          TXT      "Linux.Bogus, your DNS
consultants"
localhost             A        127.0.0.1
mail                  A        127.0.0.4
mail                  MX       10 mail.Linux.bogus
mail                  MX       20 mail.friend.bogus
mail                  HINFO    386sx Linux 1.0.9
donald                A        127.0.0.3
donald                MX       10 mail.Linux.bogus
donald                MX       20 mail.friend.bogus
donald                HINFO    i486      Linux 1.2
donald                TXT      "DEK"
www                   CNAME    ns.Linux.bogus
richard CNAME ns.Linux.bogus
ftp                   A        127.0.0.5
ftp                   MX       10 mail.Linux.bogus
ftp                   MX       20 mail.friend.bogus
ftp                   HINFO    P6        Linux 1.3.59
ns                    A        127.0.0.2
ns                    MX       10 mail.Linux.bogus
ns                    MX       20 mail.friend.bogus
ns                    HINFO    Pentium   Linux 1.2
ns                    TXT      "RMS"
Linux.bogus           SOA      Linux.bogus
hostmaster.Linux.bogus. (1995 11301 28800 7200 604800 86400)
```

检查 www.Linux.bogus:

```
> set q=any
> www.Linux.bogus.
Server: localhost
Address: 127.0.0.1
```

```
www.Linux.bogus canonical name = ns.Linux.bogus
```

```
ns.Linux.bogus
```

```
Linux.bogus      nameserver = ns.Linux.bogus
```

```
Linux.bogus      nameserver = ns.friend.bogus
```

```
ns.Linux.bogus   internet address = 127.0.0.2
```

ns.Linux.bogus 拥有 127.0.0.2 这个地址。

当然，这个领域是虚拟的，里面所有的地址也都是。真正的领域要在其中插入真正的域名和地址以及其他所有的有关信息。这些工作完成时用户需要一个反向寻找用的区域文件，并且为其中所使用的每个地址包含唯一的一个 PTR 资源记录，如：

```
127.0.0.2      PTR      ns.Linux.bogus.
127.0.0.3      PTR      donald.Linux.bogus.
127.0.0.4      PTR      mail.Linux.bogus.
127.0.0.5      PTR      ftp.Linux.bogus.
```

对于作为范例的领域(当然是附加于 SOA 资源记录的)，该文件的域名(基点)应该是反向的，这个情况正如同 127.0.0 与 0.0.127.in-addr.arpa 的关系一样。

当熟悉了 Linux.bogus 领域的操作后，从 named.boot 文件里把它删除。

8.3 维 护

对于 named，用户除了使之正常运行之外，还有个维护的任务得要做，那就是维持 root.cache 文件的更新。

最简单的方法是使用 dig 程序，首先执行不任何参数的 dig 程序，用户将会得基于自己的服务器的 root.cache 文件。然后 dig rootserver 命令查询所列出的根服务器之一。用户将会注意到如下的输出看起来非常像一个 root.cache 文件，除了某些数字以外，这些数字不会有什么妨碍。把输出信息存放到文件(例如，dig .@e.root-server.net>root.cache.new)中并且用该文件来取代原来旧的 root.cache 文件。取代了原来的 cache 文件之后要记得重新启动 named 程序。

下例是一个脚本文件，它可以自动执行来更新 named.cache 文件，为它安装一个 crontab 项目。这个脚本文件假设用户的电子邮件可以运行，并且定义了“hostmaster”这个邮件别名：

```
#!/bin/sh
#
# Update the nameserver cache information file once per month.
# This is run automatically by a cron entry.
#
```

```
(
echo "To: hostmaster <hostmaster>"
echo "From: system <root>"
echo "Subject: Automatic update of the named.boot file"
echo

export PATH=/sbin: /usr/sbin: /bin: /usr/bin:
cd /var/named

dig . @rs.internic.net >named.cache.new

echo "The named.boot file has been updated to contain the following
information: "
echo
cat named.boot.new

chown root.root named.cache.new
chmod 444 named.cache.new
rm -f named.cache.old
mv named.cache named.cache.old
mv named.cache.new named.cache
ndc restart
echo
echo "The nameserver has been restarted to ensure that the update is complete."

echo "The previous named.cache file is now called /var/named/named.cache.old."
) 2>&1 | /usr/lib/sendmail -t
exit 0
```

8.4 拨号连接的自动设置

本节解释如何设置拨号连接以便自动完成每个过程。这里使用的是 PPP 拨号，然而有许多人使用 `slip` 或是 `cslip` 连接方式，所以用户的实际设置里几乎每个地方都可能与此有所不同。但是 `slip` 使用的 `dip` 程序应该能够完成这里所做的许多事情。

一般来说，当没有连上网络时，`resolv.conf` 文件只包含如下一行：

```
domain uio.no
```

当连上网络的时候，需要启动 `named` 进程，并且拥有一个如前所述的 `resolv.conf` 文件。

通过保持两个名为 `resolv.conf.local` 和 `resolv.conf.connected` 的文件以便作为该 `resolv.conf` 的模板文件来解决这个问题。后面的文件看起来如前所述的 `resolv.conf` 文件。

要自动连接到网络，需执行一个“`ppp-on`”脚本文件：

```
#!/bin/sh
echo calling...
pppd
```

pppd 有个 options 文件，它用于配置关于如何取得连接的一些特殊选项。PPP 连接完成后，pppd 启动一个 ip-up 命令(这在 pppd 的联机手册里有说明)。以下是该脚本文件的一部分：

```
#!/bin/sh
interface="$1"
device="$2"
speed="$3"
myip="$4"
upip="$5"
cp -v /etc/resolv.conf.connected /etc/resolv.conf
/usr/sbin/named
```

当 PPP 脱机时，pppd 执行 ip-down 脚本文件：

```
#!/bin/sh
cp /etc/resolv.conf.local /etc/resolv.conf
read namedpid </var/run/named.pid
kill $namedpid
```

当用户没有连接到网络上时执行 named 可能并不好，因为 named 将会尝试送出查询到网络上而且其终止时限很长，而每次有某些程序尝试解析一个名称的时候用户就得等待到这个终止时限。如果使用拨号上网，应该在连上网时启动 named 并且在脱机时杀掉它。

某些人喜欢在慢速的连接上使用 forwarders 命令。如果用户的因特网服务供应商 ISP 在 1.2.3.4 以及 1.2.3.5 设有 DNS 服务器，那么用户可以插入如下一行：

```
forwarders 1.2.3.4 1.2.3.5
```

到 named.boot 文件中。同时使 named.cache 文件保持为空白。这将会减低源自用户主机的 IP 流量，并可能提升速度。如果用户按线路的数据量付费，这特别有用。这还有个好处，让用户不必负作为一个缓存的 named 维护者所应负起的责任：用户不需要去更新一个空的 named.cache 文件。

第 9 章 Linux 的中文环境与 StarOffice

为了使 Linux 系统可以处理中文数据，必须在系统上安装中文系统。

目前在 Linux 上可以用来处理中文的系统有几种，以下将简单介绍这些系统。

9.1 简介

目前在 Linux 上支持几种中文处理系统，用户可以择一安装，或者全部都安装。

本章简单地介绍四种较常用的中文处理系统。关于那些系统的详细信息请参阅其说明文件。

9.1.1 chdrv

一种在终端模式下提供显示、输入的中文显示程序。

chdrv 只能在主控制台(也就是用户的 Linux 计算机)使用，而无法提供给其他通过网络进入系统的用户使用。该软件最大的缺点是执行的速度比较慢。

9.1.2 yact

yact 也是在终端模式下执行的中文显示、输入系统。

yact 与 **chdrv** 最大不同的地方是：**yact** 是通过 **svglib** 函数库使用显示卡，如果用户的 **svglib** 没有包含所使用的显示卡数据，可能无法使用这个系统。

9.1.3 cxtterm

cxtterm 是在 X Window 模式下执行的中文虚拟终端。**cxtterm** 提供最古老的中文显示、输入环境，并提供各种中文内码模式，包含 BIG 5、HZ 和 GB 等等。因为每一个 **cxtterm** 虚拟终端都必须装入中文数据，耗用的系统资源相当大。

9.1.4 xcin 与 crvxt

xcin 与 crvxt 是在 X Window 模式下执行的中文输入系统, 因为 xcin 是利用 X Window 的客户/服务器方式执行的, 所以只要启动一个 xcin 输入窗口, 便可以对应许多 crvxt 显示虚拟终端, 占用系统资源比较小。

xcin 与 crvxt 也可以提供类似 DOS 环境下的五笔字型或自然输入法的智能型输入选字环境。目前 xcin 与 crvxt 需配合倚天中文系统来使用, 制作字体与输入法。

9.2 安装与设置

各系统的安装方法各不相同, 请用户参阅其 README 或相关文件。大致的安装步骤是:

1. 取得文件。
2. 解开。
3. 设置。
4. 编译。
5. 安装。
6. 使用环境的设置。
7. 完成。

以下仅简略介绍上面所提到的系统的安装法, 详细的安装法还是请用户参阅其随附的文件。

9.2.1 chdrv

程序源代码的安装步骤:

1. 在一个是够大的目录下(建议/tmp 或/usr/src)解开 chdrv 原程序压缩文件。
2. 把 chdrvfont.tar.gz(chdrv 的字体文件)也放到解开的目录下。
3. 进入 chdrv 目录, 键入 make unpack 命令把原程序整理后安装在/root/chinese 目录下。
4. 进入/root/chinese 目录。
5. 键入 make 命令。
6. 键入 make install 进行安装。
7. 进行设置的工作。
8. 完成。

可执行文件(二进制文件)安装步骤:

1. 在/usr/src 或/tmp 目录下解开 chdrv 的可执行程序压缩文件。

2. 把 chdrv 的字体压缩文件在/usr/local/lib/chinese 目录下解开。
3. 进入 chdrv 目录。
4. 键入 installbin 命令。
5. 修改/etc/chinese.conf 的设置。
6. 完成。

安装输入法:

/etc/chinese.conf 文件是中文输入法的设置文件, 请用户依照自己的需要, 修改它的内容, 现在以增加一个输入法来做示例。

```
# define input method table
# PHONETIC : the phonetic input method table
# MULTI : multiple input methods, it is order-dependent

BEGIN INPUT
PHONETIC /usr/local/lib/chinese/phone.def
#MULTI /usr/local/lib/chinese/dayi.tbl
MULTI /usr/local/lib/chinese/Boshiamy.tbl
END INPUT
```

修改/etc/chinese.conf 文件后, 把 Boshiamy.tbl 放到/usr/local/lib/chinese/目录中。最后利用 chconfig 程序使/etc/chinese.conf 的内容生效。



注意:

- ① 如果是使用程序源代码的方式来安装, 请在编译前确定 Linux 内核是否已经编译过, 因为 chdrv 需要读取一些系统的程序源代码, 如果少了这些文件, 那么 chdrv 无法顺利编译成功。
- ② 使用 chdrv 前请先阅读其随附文件。

9.2.2 yact 的安装步骤

1. 解开 yact 原程序压缩文件。
2. 进入 yact 目录。
3. 键入 make 编译其程序源代码。
4. 拷贝字体并更改字体设置文件。
5. 修改输入法。
6. 完成。

9.2.2.1 系统需求

yact 通过 svgalib 来使用 VGA 显示卡, 所以用户使用的显示卡必须在 svgalib 所支持的范围之内。至于用户的显示卡是否可以使用 yact, 请参阅 svgalib 相关的说明文件。

9.2.2.2 安装字体

yact 本身并没有附送任何的字体, 如果用户想要使用中文字体, 有几个方法可以达到目的:

- 使用倚天中文系统的字体
倚天中文系统已经是国内非常普及的中文系统, 相信用户要取得倚天的字体应该不困难。yact 需要倚天中文系统的 ASCFONT.24、STDFONT.24、SPCFSUPP.24 和 SPCFONT.24 这四个字体文件。
- 转换 bdf 格式的中文字体
用户可以去 ftp.nctu.edu.tw 站点获取一些免费的中文 bdf 字体文件, 通过 bdftohbf 程序把 bdf 字体转换成 hbf 字体。

9.2.2.3 安装输入法

yact 使用与 xterm 相同的 cit 版本 2 的输入格式, 用户可以自己在网络上面获取所需要的输入法, 并把输入法安装到 /usr/local/lib/yact/ 目录下, 当启动 yact 时, 会自动地读取这个目录的输入法文件。至于输入法启动的按键方式, 是使用文件名来对应相应的按键组合, 说明如下:

```
/usr/local/lib/yact/1 - Ctrl + Alt + 1
/usr/local/lib/yact/2 - Ctrl + Alt + 2
/usr/local/lib/yact/3 - Ctrl + Alt + 3
...
Ctrl + Space - 切换中 / 英文输入模式。
```

9.2.3 xcin 的安装与设置

下面介绍 xcin 的安装与设置方法。

9.2.3.1 安装步骤

1. tar xzvf xcin-2.1b1.tgz
在 /usr/src 目录中解开 xcin 文件, 以便进行编译的工作。
2. cd /usr/src/big5-pack
进入 xcin 软件包程序源代码目录。
3. cd xcin
进入 xcin 主程序目录。
4. vi Imakefile
修改 Imakefile 文件中的 ETDIR 变量, 把它指向倚天中文系统的目录。
5. xmkmf; make; make install
编译并安装 xcin 主程序。
6. cd ../crxvt; xmkmf; make; make install

编译并安装 `crxvt` 虚拟终端程序。

7. `cd ../big5font`
进入字体文件转换程序的目录。
8. `vi Makefile`
编辑 `Makefile` 文件, 把下面这两个参数修改一下:
`ETDIR`: 倚天中文系统存放的位置。
`FONDIR`: X Window 系统存放中文字体的位置。
9. `make;make install`
编译字体。
10. 完成。

9.2.3.2 设置

因为 `xcin` 是一个遵守 X Window 标准的程序, 所以可以使用一般的设置 X Window 程序的方法来设置(例如在 `$HOME/.Xdefaults` 中设置 `xcin` 的资源), 或是在启动的时候以附加参数的方法设置。详细的设置说明, 还是请用户参阅 `xcin` 所附的文件。

9.2.3.3 启动

在 X Window 模式下直接输入:

```
[cdchen@Linux cdchen]$ xcin &
```

就可以执行 `xcin` 了。

9.3 中文环境的建立

完成了中文处理系统的建立工作后, 用户已经可以在 Linux 机器上显示中文了。但如果使用中文的文本编辑软件, 用户会发现, Linux 系统只能显示中文, 却无法接受中文的输入。

如果要改进这个问题, 必须自己亲自修改两个地方, 使 Linux 系统可以接受中文的输出与输入。首先必须在用户使用的 Shell 启动文件(例如 `.login` 文件)增加 `locale` 环境变量的设置, 其次也得在自己的主目录中的 `.inputrc` 文件(如果主目录中没有这样的文件, 请自己建立)增加关于输入的设置。

以下把 Shell 启动文件与 `.inputrc` 相关设置公开如下, 用户可以参考使用:

```
Shell:
~ -----
Bash Shell:
```

请在 `/etc/profile` 增加下面的内容:

```
stty cs8 -istrip
stty pass8
```

```
export LANG=C
export LC_CTYPE=iso-8859-1
```

Tcsh Shell:

请在/etc/csh.login 和/etc/csh.cshrc 中增加如下设置:

```
stty cs8 -istrip
stty pass8
setenv LANG C
setenv LC_CTYPE=iso-8859-1
```

在\$HOME/.inputrc 文件中增加如下设置:

```
set convert-meta off
set output-meta on
```

最后, 请自己准备一个已经包含中文的文件, 利用 `grep` 程序来搜寻其中的文字。如果可以找得到, 表示 Linux 系统已经可以处理中文了。

9.4 StarOffice

本节将讲述 Linux 系统上常用的办公软件包——StarOffice, 通过这个软件包, 用户可以完成在 Windows 95 下 Office 软件包能完成的大部分工作。

9.4.1 简介

StarOffice Office Suite 是在 Linux 下执行的一套办公室工具软件, 是通过 C 语言和 Motif 编写的。

StarOffice 包含:

- StarWriter 文本处理器
- StarCalc 运算器
- StarImage 图形编辑器
- StarDraw 点像 CorelDRAW 的绘图软件
- StarChart 制作图表的程序
- StarMath 数学编辑器

StarOffice Office Suite 的 Linux 版在非商业性使用下是个免费软件。若要用在商业性使用, 需要授权才行。目前授权费用是与 Caldera's Linux 系统分开的, 多少尚未确定。

Motif信息

StarOffice 由商业版 Motif 2.0 GUI 软件包所写成。动态链接版不能在 Motif 1.2/LessTif 下工作, 而静态版本(使用 Motif 2.0 链接的)则不管用户是否有 Motif 都可以工作。然而,

执行效率会降低。

在 Pentium 100、48MB 物理内存且装有 Redhat 4.1 及 Motif 2.0 的机器上执行 StarOffice，使用动态链接执行的 StarOffice 只需花 1 分钟来装入。然而，值得注意的是，等它被装入后，使用动态链接的 StarOffice 比使用静态链接的 StarOffice 占用更少的系统资源。

9.4.2 取得 StarOffice Office Suite 软件

目前最新的版本是 StarOffice 3.1 版。可从下面的站点获取：

```
ftp://ftp.gwdg.de/pub/Linux/staroffice
```

README.StarOffice 文件包含关于下载 StarOffice 3.1 版的信息。用户需要获取以下三个文件：

StarOffice31-english.tar.gz(英文版)基本文件或 StarOffice31-german.tar.gz(德文版)基本文件，StarOffice31-dynbin.tar.tgz(动态链接用)或 StarOffice31-statbin.tar.gz(静态链接用)。二进制文件和 StarOffice31-common.tar.gz。

Caldera 站点也有 StarOffice 版本，请查找它的网站：

```
http://www.caldera.com
```

9.4.3 安装 StarOffice

安装 StarOffice 的步骤如下：

1. 以 root 身份解开源文件到/usr/local 目录下。
2. 执行 setup 程序。
3. 最后使用命令：

```
source.sd.sh
```

或者

```
source.sd.csh
```

9.4.3.1 libc版本及修补处

StarOffice 与 libc 5.4.4 进行链接。

StarOffice 3.1 可在 libc 5.3.x 上工作。但是，它的设置程序必须在 5.4.4 或更高的版本才行。如果有 libc 5.3.x，应该可以取得 libc 5.4.4+并在执行 setup 脚本前把它加到 LD_LIBRARY_PATH 环境变量内，这样就可解决问题了。

例如，在 C Shell 下，执行如下命令：

```
setenv LD_LIBRARY_PATH 对应的目录名称
```

如果用户试着在比 5.4.4 版还早的 libc 版本上执行 setup 脚本，将会得到如下的错误信

息:

```
line 1: Syntax error at token 'I' expected declarator;  
i.e. File ...
```

要升级系统的 `libc`, 可以通过 FTP 到站点 `sunsite.unc.edu` 看看 `/pub/Linux/GCC` 目录内的 `libc-5.4.33.bin.tar.gz` 文件。把这个文件的内容解开到一个缓存目录内, 会有个新的 `lib/` 目录被建立。通过命令 `su` 转换成 `root` 后, 从这个缓存目录拷贝 `libc.so.5.4.33` 文件到 `/lib` 目录内。再以下的命令建立从 `libc.so.5` 到 `libc.5.4.33` 的符号链接:

```
ln -sf /lib/libc.so.5.4.33 /lib/libc.so.5
```

然后执行 `ldconfig` 命令。

9.4.3.2 安装tar文件

下载完 `StarOffice` 后, 通过命令 `su` 转换成或登录成 `root`, 并将取得的文件放在 `/usr/local/` 目录下。

改变目录到 `/usr/local/` 后解开这些文件。以下是解开被 `gzip` 压缩后再被 `tar` 存档过的文件的命令:

```
tar -xzf filename.tar
```

在旧的系统上可能需要先使用 `gzip -d` 命令来解压缩被 `gzip` 压缩过的文件, 然后使用 `tar-xvf` 命令来解档该文件。

这些文件会解开到新建立的 `usr/local/StarOffice-3.1` 目录内。

9.4.3.3 设置与搭建

以 `root` 身份解开 `StarOffice` 的文件之后, 改变目录到 `/usr/local/StarOffice-3.1` 内并且执行 `setup` 程序。这个程序会安装非共享的文件和符号链接到个别用户需要的文件。如果用户不接受内定的安装路径有可能会有问题。

`StarOffice` 使用几个环境变量。

文件 `.sd.sh` (给 Bourne Shell 使用) 以及文件 `.sd.csh` (给 C Shell 使用) 含有提供给 `StarOffice` 使用的环境变量。这些文件都放在用户的主目录下。

如果用户使用的 Shell 是 `bash`, 编辑 `.bashrc` 文件并且加上以下一行:

```
source ~/.sd.sh
```

做完后, 重新执行 `bash`, 环境变量的改变就会生效。

如果用户使用不同的 Shell, 请先查看 Shell 的 `man` 帮助页的相关信息。

9.4.3.4 修改LANG变量的问题

在 `.sd.sh` 和 `.sd.csh` 这两个文件中设置 `LANG` 变量后, 会在使用 `perl` 和 `man` 命令时产生问题。

使用 `man` 命令时, 会得到下面的错误信息:

```
' Failed to open the message catalog man on the path NLSPATH=''
```

使用 `perl` 命令会得到如下的错误的信息

```
' warning: setlocale(LC_CTYPE, "")...'
```

`.sd.sh` 文件中设置了 `LANG=us`, 去掉 `LANG=us` 这行, 并将 `LANG` 从变量列表中去掉, 就可以修正该问题了。

在 `.sd.csh` 文件中(以 C Shell 的格式所写), 需要去掉有 `setenv LANG us` 的行。

9.4.4 执行 StarOffice

StarOffice 的执行文件被放在 `/usr/local/StarOffice-3.1/Linux-x86/bin/` 目录下。

应用程序包括:

- `sdraw3`
- `swriter3`
- `scal3`
- `smath3`
- `schart3`
- `simage3`

要使用联机手册, 必须执行 `svdaemon` 命令。要使用 `StarWriter`、`StarCalc` 及 `StarDraw` 应用程序间的通信功能, 必须执行 `svportmap` 命令。

9.4.5 StarOffice 使用技巧

在 StarOffice 使用过程中, 还需要搞清楚下面的一些细节:

1. 可以不采用原本在 `.bash_profile` 脚本内的 `.sd.sh`(或 `.sd.csh`) 文件, 而将 `.sd.sh` 命令脚本拷贝起来, 把它改名成 `swriter`, 再把它拷贝到 `/usr/local/bin` 中。然后, 把它修改成:

```
#!/bin/bash
```

在该文件的最后一行加入下面的内容:

```
exec swriter3 $*
```

现在只要执行 `swriter` 命令, 让它自动地设置好适当的环境变量, 然后就可以执行 `StarWriter` 程序。

2. 有很多人在装入 StarOffice 时要花很多时间。下面介绍解决此问题的方法。在很局限的环境下执行 StarOffice, 只放 StarOffice 所使用的二进制程序代码以及程序库(除 `/usr/X11R6/lib`、`libc/libm` 和 `libg++/libstdc++` 之外的所有程序库)。

第 10 章 Internet 服务器

为了反映出 UNIX 和 Internet 之间的紧密关系, Linux 提供了很好 Internet 服务, 如 Web、WAIS、FTP 和 Gopher。除了访问其他的站点之外, 还能设置自己的 Linux 系统作为一个 Web 站点或 FTP 站点。

别人可以用用户创建的 Web 页来访问系统或下载用户提供的文件。这样的系统便被称为服务器, 它能提供服务。

一个单一的 Linux 系统可以提供几种不同的服务。Linux 系统可以是一个 Web 服务器和一个 FTP 服务器, 也可以同时是一个 Gopher 服务器和 WAIS 服务器。一个用户可以用 FTP 服务下载文件, 同时另一个用户可以阅读 Web 页。为了使 Linux 系统提供服务, 要为每个服务安装并运行合适的服务器软件。

一般的 Linux 系统已经是一个 Web、FTP 和 Gopher 服务器。在设计此系统时已经把它设计为 Internet 服务器。为了使 Linux 系统成为一个 Web 服务器, 只需创建 Web 页面。另外, 只需把要使之可用的文件放在 FTP 目录下, 即成为 FTP 服务器。

为了使 Linux 系统作为一个 Internet 服务器, 必须取得一个到 Internet 的连接并提供远程用户到系统的访问。访问通常是指允许匿名注册到服务器保留的目录中。Linux 系统已配置允许 Web 和 FTP 用户进行访问。

建立一个能容纳服务器操作的 Internet 连接并不容易。需要一个专门的连接, 或由 Internet 服务供应商提供一种连接。不仅是自己连接到 Internet, 还要允许许多用户通过 Internet 和系统建立一个很大流量的连接。

如果只想给本地网络提供服务, 就不需要专门的连接。另外, 还可以允许用户通过调制解调器连接和注册到系统, 以提供信息服务。用户可以拨号到系统中使用 Web 页或用 FTP 下载文件。无论想提供什么样的服务, 都需要安装合适的服务器软件并运行它。

本章将讲述 Internet 的四部分服务: Web、FTP、WAIS 和 Gopher。

10.1 介 绍

服务器就是一个和其他程序同时运行的守护进程, 它不断地查找是否有对服务的请求, 此请求可以来自系统的其他用户和通过网络连接到系统中的远程的用户。当它收到一个来自用户的服务请求, 它便启动一个会话以提供服务。例如, 如果一个用户想从系统中下载一个文件, 它能使用它自己的 FTP 客户程序来请求 FTP 服务器为它启动一个会话。在此会话期间, 它能访问系统并从中下载文件。

为了使用户能访问服务器, 服务器必须运行。如果在系统中设置一个带有 HTML 文

件的 Web 站点，在用户能访问 Web 站点和显示这些文件之前，必须运行 httpd Web 服务器程序。

启动一个服务器有几种方法。一种方法是从命令行手工输入服务器程序的名字和它的参数选项。然后按 Enter 键，服务器将被启动并且继续显示命令行提示。

为了查看服务器是否运行，可以输入下面的命令以查看当前运行的所有进程(将能看到为服务器程序启动的进程)：

```
#ps -ax
```

系统可以自动地启动这些服务器，而不用每次开机后都要自己手工执行这些服务器程序。依照用户想如何使用一个服务器，配置自动启动有两种方法：

- 可以让服务器自开机后便启动，并一直运行直到关闭它。
- 另外还可以让服务器收到用户对它的第一个服务请求时再启动。

如果一个服务器经常被使用，可能想让它始终运行。如果它不常被使用，可能希望当有服务请求时再启动它。例如，如果运行一个 Web 站点，Web 服务器将随时可能收到来自 Internet 上的请求，而对于 FTP 站点来说，可能不常收到请求，在这种情况下，可能想让 FTP 服务器收到请求时再启动。当然有些 FTP 站点会收到频繁的请求，它们将被设置为不断运行的 FTP 服务器。

为了启动系统时自动启动一个服务器，需要用到目录/etc/rc.d/init.d 中的启动脚本。可以通过配置 inetd 守护进程来指定当收到一个服务请求时再启动一个服务器。inetd 将查找服务请求并在收到请求时启动服务器。

Linux 系统已经配置自动启动并不间断运行 Web 服务器。它的配置脚本叫做 httpd.init，存放在目录/etc/rc.d/init.d 中。

FTP 服务器在 inetd 下配置运行。当有人初始化一个 FTP 会话时才启动它。在配置文件/etc/inetd.conf 中能找到一个关于 FTP 服务器的项，而不像 Web 服务器那样在目录/etc/rc.d/init.d 中没有关于 FTP 的脚本。

10.2 服务器初始化脚本

Linux 用了一种补充的初始化文件，此文件在其他的系统中不存在。对于其他系统，可能不得不把服务器命令放在系统初始化文件 rc.local 中。对于 Linux，能使用目录/etc/rc.d/init.d 中的初始化脚本。

目录/etc/rc.d/init.d 中有一些当启动系统时自动执行的 Shell 脚本。这些脚本中有些命令可以执行特殊的程序，如服务器程序。在这些程序中，能找到一个关于 Web 服务器的脚本叫做 httpd。小心不要更改其他的脚本程序。这些都是关于网络接口或打印机守护程序的必须的启动程序。在目录中有一个简单的启动脚本叫做 skeleton。为了给服务器创建一个新的脚本，可以首先复制一个 skeleton 脚本，接着编辑新脚本，把服务器路径名和命令放在合适的位置。下面以 httpd 脚本为例，介绍脚本程序。

```
/etc/rc.d/init.d/httpd
```

httpd 脚本首先执行一个定义脚本中使用的函数的脚本，然后执行一个网络脚本以决定网络配置。接着它检查这个守护进程的配置文件是否存在，如存在，将执行此文件。对于 httpd 守护进程，配置文件名为 `/etc/sysconfig/daemons/httpd`。这个文件中设置了一个在以后命令中将使用的变量 `$VARIANT` 的值，在此处该值为 `Apache`。

```
./etc/rc.d/init.d/functions
./etc/sysconfig/network
[! -r /etc/sysconfig/daemons/$NAME ] && exit 0
./etc/sysconfig/daemons/$NAME
```

Web 服务器程序将执行脚本中下面的命令行。变量 `$NAME` 被设置为 `httpd`，`$DAEMON` 被设置为 Web 服务器程序 `/usr/sbin/httpd`。-f 选项指定此 Web 服务器的配置文件，在配置文件中 `$VARIANT` 指定一个使用的 Web 服务器，在此处该值为 `Apache`。start-stop-daemon 函数加上 -S 选项将启动此守护进程。

```
start-stop-daemon -S -n $NAME -x $DAEMON -- -f
/etc/httpd/$VARIANT/conf/httpd.conf
```

start-stop-daemon 函数加上 -K 选项将关闭此守护进程，用 -p 选项指定进程号，用 -n 选项指定名字。

```
start-stop-daemon -K -p/var/run/$NAME.pid -n $NAME
```

系统中为每个运行级别都设置一个子目录，这些初始化的文件能被这些子目录中的链接访问。在目录 `/etc/rc.d` 中，有一组命名规则为 `rcN.d` 的子目录，在此 `N` 是用来指定运行级别的数字。rc 脚本将在系统启动时决定运行级别，并只执行为此运行级别设置的子目录中的脚本文件。默认的运行级别为 3，它是多用户级。当启动系统时，rc 脚本将执行指定在 `rc3.d` 目录中的启动脚本。

`rc3.d` 目录中有符号链接到 `/etc/rc.d/init.d` 目录中的启动脚本。所以在目录 `/etc/rc.d/init.d` 中的 httpd 脚本实际是通过 `rc3.d` 目录中的符号链接被调用。在 `rc3.d` 目录中链接到 `/etc/rc.d/httpd` 脚本的是 `S85httpd`。为了使得服务器能自动启动，首先要在目录 `/etc/rc.d/init.d` 中创建一个启动脚本，然后在目录 `/etc/rc.d/rc3.d` 中创建一个符号链接到此脚本。

假设已在系统中安装一个 Gopher 服务器，现在想自动启动它。最简单的方法是创建一个启动脚本 `gopherd`，如果已经存在 `httpd` 脚本，则只需把 `httpd` 复制一份，并命名为 `gopherd`。`httpd` 和 `gopherd` 都是网络服务器的自动启动脚本，它们的内容基本上相同的，只是对应的启动命令不同：

```
#cp httpd gopherd
```

接着要编辑 `gopherd` 脚本并替换 `httpd` 中和 `gopherd` 不同的地方。对于变量 `$DAEMON`，应该分配 `gopherd` 服务器程序的路径名，如 `/usr/sbin/gopherd`：

```
$NAME=gopherd
$DAEMON=/usr/sbin/gopherd
```

下面的行将执行 `gopherd` 服务器。注意命令中包括一些参数，如 Gopher 文件的目录。

服务器不同参数也有所不同。HTTP Web 服务器没有参数，能把引用移到变量\$VARIANT 中。

```
start-stop-daemon -S -n $NAME -x $DAEMON /usr/lib/gopher-data
```

现在切换到目录/etc/rc.d/rc3.d 中，创建一个符号链接到/etc/rc.d/init.d/gopherd 脚本。命令 ln 加-s 选项将创建此链接：

```
#ln -s /etc/rc.d/init.d/gopherd S94gopherd
```

以后再启动系统时，gopherd 服务器将自动启动，不断地运行等待着请求。

除了复制脚本 httpd 外，还可以复制脚本 skeleton。还可以加一些命令来执行 /etc/sysconfig/network 脚本并检查网络是否正常连接。

用户也许会注意到这里没有 FTP 服务器的脚本。在 Linux 中，FTP 服务器是由 inetd 守护进程管理的，在 10.4 节中会详细地介绍。如果希望 FTP 服务器时刻运行，可以像在操作 gopherd 服务器那样创建一个 init 脚本和一个符号链接。FTP 服务器程序的名称为 in.ftpd。还需要给文件 inetd.conf 中的对应的 ftpd 项加注释，使它不再管理 FTP 服务器。

10.3 inetd 服务器管理

如果系统中的某一服务平均只有很少的请求。就不需要此服务的服务器时刻运行。可以在远程用户访问它时再运行它。

inetd 守护进程管理 Internet 服务器，并只在系统收到对服务的请求时才启动它们。inetd 将不停地检查远程用户对某一特定服务的请求，当收到请求时便启动相应的服务器守护进程。例如，当 inetd 收到一个用户访问 FTP 的请求，它启动 ftpd(FTP 守护进程)。ftpd 将处理此请求，允许用户下载文件。

要使得 inetd 调用合适的服务器守护进程，必须配置此服务。要在文件/etc/services 和 /etc/inetd.conf 中配置服务器的选项。

/etc/services 文件中列出系统中可用的服务。文件/etc/services 中的项包含服务的名字和它的端口名以及用斜杠分隔开的协议。出现在 RedHat 的文件/etc/services 中的 FTP 项如下所示(别的系统可能只要求一项)：

```
ftp-data    20/tcp
ftp        21/tcp
```

/etc/inetd.conf 文件是 inetd 的配置文件。需要为文件中的项指定服务、协议和调用的服务器程序。下面显示的是一个 FTP 的项(服务器路径和参数根据不同的 Linux 也有所不同)：

```
#<service> <sock_type> <proto> <flags> <user> <server_path> <args>
ftp        stream      tcp        nowait root    /usr/sbin/ftpd ftpd
```

在别的 Linux 中，配置行可能存在，但前面加有一个“#”号把它注释掉。只须除去“#”号。如没有此配置项，就需要加上它。

tcpd

可以用 tcpd 守护进程给 inetd 管理服务加上一个安全级。能设置 tcpd 来监视服务器通过 inetd 的连接。tcpd 将检查远程的用户认证以确认它们进行一个有效的请求。用 tcpd 还能限制远程用户访问系统。

如果使用 tcpd 守护进程，则一系列主机名存放在文件 `hosts.allow` 和 `hosts.deny` 中。文件中的项目有如下格式：

```
service:hostname:domain.
```

`domain` 是可选项。对于 `service` 可以指定一个特定的服务，如 FTP 或输入 `ALL` 指定所有的服务。对于 `hostname` 可以指定一个特定主机名，或用 `ALL` 指定所有的主机。下例中，第一个记录允许所有的主机访问 Web 服务 HTTP。第二个记录允许主机 `pangol.train.com` 访问所有的服务。第三和第四个记录允许主机 `rose.berkeley.edu` 和 `caldera.com` 进行 FTP 访问。

```
http:ALL
ALL:pangol.train.com
ftp:rose.berkeley.edu
ftp:caldera.com
```

文件 `hosts.allow` 中包含允许访问的主机。如果只是不允许一些特定的主机访问，可以在文件 `hosts.allow` 中对服务指定 `ALL`，而在文件 `hosts.deny` 中列出要禁止的主机名。tcpd 的帮助页(`man tcpd`)提供关于 tcpd 的更多信息。

为了使 tcpd 监视一个服务器，需要在文件 `inetd.conf` 中的服务器项的路径名区域中放置 tcpd 的路径名。用 tcpd 守护进程的路径名 `/usr/sbin/tcpd`，可以代替 `ftpd` 程序的路径名 `/usr/sbin/in.ftpd`。参数域中列出 `in.ftpd` 服务器程序：

```
#<service> <sock_type> <proto> <flags> <user> <server_path> <args>
ftp          stream      tcp        nowait root    /usr/sbin/tcpd  in.ftpd
```

当 inetd 守护进程收到一个 FTP 服务的请求时，它调用 tcpd 守护进程，用它来接管和监视连接。接着它启动 `in.ftpd` 服务器程序。默认情况下，tcpd 将允许所有的请求。为了允许所有的对 FTP 服务的请求，要在文件 `/etc/hosts.allow` 中输入如下所示的记录(项 `ALL:ALL` 将使系统对所有的用户开放所有的服务)：

```
ftp:ALL
```

10.4 FTP 服务器

Linux 已经安装一个 FTP 服务器并创建一个用户放置供 FTP 访问的文件的目录。目录

已经被配置控制远程用户的访问,限制用户只能访问 FTP 目录和它的子目录。留给 FTP 的目录为/home/ftp。要把允许 FTP 访问的文件放在目录/home/ftp/pub 中。还可以在其中创建子目录来存放文件。一旦连接到网络中,一个远程的用户就可以 FTP 到系统中,并从目录/home/ftp/pub 或它的子目录中下载文件。

虽然安装 Linux 系统后 FTP 服务就已被配置好了,但是理解它是如何配置的也是十分有用的。当安装另一个系统时,就需要知道如何安装 FTP 服务器和创建它的数据目录。然而在 Linux 中已经完成这些设置,不需要再一一设置。

在不同的 Linux 的 FTP 站点上都有 FTP 服务器软件。在 sunsite.unc.edu 和它的镜像站点上,FTP 服务器软件位于/pub/Linux/systems/Network/file-transfer 目录中。美国华盛顿大学的 FTP 服务器是一个 wu-ftp-2.4-fixed.tar.gz 文件。用户可能需要从这些站点中下载 FTP 服务器软件的更新版本,并用它们来更新服务器。

如果从一个 FTP 站点下载 FTP 服务器软件,将首先要解压和展开它们。将会自动创建一些目录存放文件和源代码。服务器软件包中将有安装说明,可以用它们指导来创建服务器目录和编译软件。

为了允许远程用户访问,必须运行服务器软件 ftpd。和其他的服务器一样,可以在启动机器时开始 FTP 服务,也可以当收到一个请求时通过 inetd 启动 FTP 服务器,或直接从命令行中启动它。为了在初启系统时便启动 FTP 服务器,需要像在 Web 服务器中描述的那样,在目录/etc/rc.d/inin.d 中创建一个 init 脚本。要想从命令行中启动 ftpd 服务器,要输入 ftpd 命令和参数或选项。

ftpd 服务器能通过选项被调用。通常要带允许注册的-l 选项。选项-t 和-T 用来给用户设定中断时间,一段时间内没有动作,就将切断那些连接。选项-d 显示调试信息,-u 设置上传的文件 r umask 值。

下面是 ftpd 的选项及其含义:

- d 为系统日志写调试信息。
- l 在系统日志中注册每个会话。
- t seconds 设置不活动的超时秒数(默认的是 15 分钟)。
- T seconds 允许用户设定的最大的超时时间(默认的是两小时)。
- a 使用 ftpaccess(5)配置文件。
- A 禁用 ftpaccess(5)配置文件。
- L 把设置到 ftpd 服务器的命令注册到系统日志中。
- I 把 ftpd 收到的文件注册到 xferlog。
- o 把 ftpd 传输的文件注册到系统日志。

10.4.1 FTP 服务器配置文件

这有一组配置文件,能被用来管理 FTP 站点。在 Linux 系统中,这些文件位于目录/etc 下,并以 FTP 作为名字的开头。ftpaccess 文件将决定访问 FTP 站点的用户的权限。

访问、信息、权限、记录和其余的各项功能将被指定。例如,可以给某一目录创建一

个别名, 当一个 FTP 用户注册时显示一个信息或不允许任何匿名用户删除文件。文件中的 `loginfails` 项定义一个用户在断出之前试着登录的次数, `email` 项指定 FTP 管理员的 E-mail 地址。 `ftppaccess` 的帮助页中列出可能的选项。当安装 Linux 系统, FTP 服务器将被自动安装和配置。下文将列出 `ftppaccess` 文件和它的配置。

权限是为三种不同类型的用户设置的, 三种用户为:

- 匿名
- 客户
- 真正的用户

匿名用户是指用匿名注册的用户。客户用户是指用客户帐号注册或访问的用户。一个真正的用户是指一个在系统中有帐号并用 FTP 连接访问系统的用户。用类选项能定义用户自己的类。

当显示一个信息时, `message` 项指定一个带有此信息的文件。当用户注册时显示一个信息, 当进入某一个目录时显示另一个信息。例如, 下面的项将使得在用户注册时显示在文件 `welcome.msg` 中的信息:

```
message /welcome.msg login
```

为了设定权限, 用命令加上一个 `yes` 或 `no` 接着列出用户类型或类。如下显示的文件 `ftppaccess`, 所有的用户能执行 `tar` 和 `compress` 操作, 但是匿名和客户用户将被禁止使用 `chmod`、`delete`、`overwrite` 和 `rename` 操作。它们不能删除文件、改变它们、更改它们的名字或权限。

能用 `ftphosts` 文件来允许或禁止其他的主机访问 FTP 站点。

`ftpusers` 文件列出不能通过 FTP 访问的用户。例如, 即使用户知道口令, 也不能通过 FTP 连接注册成为一个超级用户。

`ftpgroups` 是一个组存取文件, 它允许 FTP 用户成为系统中一个特定组的成员。此文件中列出特定的组口令。

在 `ftppaccess` 文件中 `private` 项必须被设置成 `yes`。下面列出了 FTP 对应的所有配置文件。

ftppaccess FTP 用户的访问、信息、权限、记录和其余的各项功能。

ftpconversions 压缩和归档文件操作的 FTP 转换表。

ftpusers 不允许通过 FTP 访问的用户列表, 如 `root`。

ftpgroups FTP 用户用特定的口令能访问的组。格式如下:

```
access-group:encrypted:real-group
```

`access-group` 是一个任意的字符串。 `real-group` 是列在 `/etc/group` 中的一个有效的组名。

ftphosts 允许或禁止不同的主机访问 FTP 站点, 如:

```
allow username addrglob[addrglob...]
```

只允许匹配 `addrglob` 的主机名以 `username` 注册, 如:

```
deny username addrglob[addrglob...]
```

将拒绝匹配 `addrglob` 的主机名以 `username` 注册。

10.4.2 FTP 用户帐号

为了允许别的用户用匿名 FTP 访问系统，必须创建一个名为 FTP 的帐号。能给帐号 FTP 设置一些限制，使得任何远程的 FTP 用户不能访问系统的其他部分。必须改变此帐号在文件 `/etc/passwd` 中的项，使一般的用户不能访问它。在很多系统上已经设置允许匿名用户进行 FTP 注册，下面的项在 `/etc/passwd` 文件中：

```
ftp:*:14:50:FTP User:/home/ftp:
```

在口令区域中的星号用来保护此帐号，它将阻止其他的用户通过以此帐号注册，并因此控制它的文件或访问系统的其他部分。用户 ID14，注释域是 FTP User，注册目录是 `/home/ftp`。当 FTP 用户注册到系统时，它将处于此系统中。如果没有设置一个根目录，创建一个并用命令 `chown` 改变 FTP 用户的权限。

组 ID 是 FTP 组的 ID，是专门为匿名 FTP 用户设置的。能通过为 FTP 组设置限制来限制匿名的 FTP 用户。下面是一个能在 `/etc/group` 文件中找到的关于 FTP 组的项。对于有的 Linux 系统，如果没有此项，应该加上它：

```
ftp::50
```

目录 `/home/ftp` 的权限中应该否定写权限。如果不希望 FTP 用户创建和删除目录，可以用 `chmod` 命令设置权限 555 来禁止写访问：

```
chmod 555 /home/ftp
```

目录 `/home/ftp` 和它的权限、口令和组项已经设置在有的 Linux 系统上。如果在另外的 Linux 上设置一个 FTP 服务器，要自己设置 `/home/ftp`。

10.4.3 FTP 服务器目录

为了防止系统遭到 FTP 用户的一些意外的访问，应在 FTP 目录中，如 `/home/ftp` 中，创建一组有限制的目录。保护的一个重要部分是阻止远程用户使用不在限制目录中的命令或程序。例如，因为 `ls` 命令位于 `/bin` 目录中，可能不希望用户使用 `ls` 列出文件名。同时，又希望 FTP 用户使用 `ls` 命令。为了做到这一点，需要在目录 `/home/ftp` 中创建一个新的目录 `bin`，接着复制一份命令 `ls` 放到 `/home/ftp/bin` 中。此目录将限制 FTP 用户的使用，它们使用的命令 `ls` 是目录 `/home/ftp/bin` 中的命令，而不是用户自己用的 `/bin` 中的 `ls` 命令。通过同样的方法，可以让 FTP 用户使用其他命令。用户需要的其他命令是：`cd`，它允许用户切换目录；命令 `more`，它让用户显示文本文件。用户还可以加入一些希望使用的其他命令。

`/home/ftp` FTP 服务器目录，所有者是 `root`（在一些系统中，此目录是 `/usr/local/ftp`）：

所有的目录和子目录有权限 555 或 755 以限制其他用户或组只有读或执行的权限。

`/home/ftp/bin` 此目录中含有 FTP 远程用户能执行的命令，如 `ls`。

`cd/home/ftp/etc` 此目录中含有配置文件如它自己的 `passwd` 文件/`home/ftp/pub` 用户提供的可下载的文件放置在此目录中；可以在其中设置子目录。

`/home/ftp/lib` 如果 `ls` 命令需要 `ld.so.1` 文件，就需要此目录来存放此文件。

还需要一个 `/home/ftp/etc` 目录，存放 `passwd` 和 `group` 文件的副本。另外它也将阻止 FTP 用户访问 `/etc` 目录下的原文件。编辑 `/home/ftp/etc/passwd` 文件，删除系统的一般用户的项。剩余的项的口令应被设置为 “*”，以保护访问。对于 `group` 文件，除去所有的用户组并设置所有的口令为 “*”。

```
#cat /home/ftp/etc/passwd
root:*:0:0:::
bin:*:1:1:::
operator:*:11:0:::
ftp:*:14:50:::
nobody:*:99:99:::
```

```
#cat /home/ftp/etc/group
root::0:
bin::1:
daemon::2:
sys::3:
adm::4:
ftp::50:
```

目录 `/home/ftp/pub` 中存放用户想让远程 FTP 用户下载的文件。当 FTP 用户注册到系统时，它将处于目录 `/home/ftp` 中，并能切换到目录 `/home/ftp/pub` 中开始访问其中的文件。在 `/home/ftp/pub` 中能加入任何目录和文件。甚至可以指定一些目录为上载目录，允许 FTP 用户上传文件到系统中。

一些 Linux 系统要求 `ls` 命令工作时要访问 `libc.so.1` 和 `rld` 文件。它们通常存放在 `/lib` 目录中。因为不希望 FTP 用户直接访问系统，要创建一个 `/home/ftp/lib` 目录，并复制这些文件，把复制的文件放在此目录中。另外，因为 `rld` 使用 `/dev/zero` 文件，还要创建一个 `/home/ftp/dev` 目录并用 `mknod` 命令复制设备文件 `/dev/zero`，然后把它放到此目录中。

权限

为了限制 FTP 用户只能访问目录 `/home/ftp` 和它的子目录，需要对 FTP 用户隐藏文件结构的其他部分。要让目录 `/home/ftp` 呈现为 FTP 用户的根目录。实际的根目录 “/” 和其他的目录结构则对 FTP 用户隐藏。可以用命令 `chroot` 加上参数 FTP，使得目录 `/home/ftp` 呈现为根目录：

```
#chroot ftp
```

此后当 FTP 用户用 `cd/` 命令来切换目录时，它总是切换到目录 `/home/ftp` 中。

作为进一步的限制，在 `/home/ftp` 中所有的命令的目录和命令本身都是为 `root` 所有，而不是被 FTP 用户所有。也就是说，FTP 用户不能控制这些目录。能用命令 `chown` 来改变目录的所有权。下例便是把目录 `/home/ftp/bin` 和命令 `/home/ftp/bin/ls` 的所有权设置成 `root`。`root` 应该拥有 `/home/ftp/bin/`、`/home/ftp/etc` 以及它们包含的所有文件：

```
chown root /home/ftp/bin
chown root /home/ftp/bin/ls
```

FTP 目录的权限应该设置为允许 FTP 用户访问。对于所有者、组和另外的用户，有三组权限：读、写和执行。为了允许 FTP 用户访问，组和目录的其他权限应设置为可读和执行。执行权限允许 FTP 用户访问目录，读权限则允许列出目录中的内容。目录不允许 FTP 用户具有写权限。没人想让 FTP 用户删除或添加一个目录。例如，目录 `/home/ftp/bin` 需要读和执行权限，因为 FTP 用户要访问和执行它的命令。对于拥有可以下载的文件目录 `/home/ftp/pub` 来说，FTP 用户必须拥有读和执行的权限。

作为目录的所有者，需要写权限以便能添加新文件或子目录。当然，只有做改变时才需要写权限。为了进一步的安全，当不需要做改动时，能设置这些目录对所有的用户包括所有者都只有读和执行的权限。用命令 `chmod` 加上数字 `555` 和目录名将设置对所有的用户为读和执行权限。在有的 Linux 系统中，FTP 目录的权限设置为 `755`，它给了所有者写权限。

```
#chmod 555 /home/ftp/bin
```

对于目录 `/home/ftp/bin` 中文件的权限和其他指定的 FTP 目录的权限能有更多的限制。一些文件需要执行，而另一些文件只要被读。目录 `/home/ftp/bin` 或 `/home/ftp/lib` 中的文件 `ls` 和 `rld` 需要执行，可以设置权限为 `555`。在目录 `/home/ftp/etc` 中的文件如 `passwd` 和 `group` 可以设置权限为 `111`，即只读的权限。如下面的例子所示，能用命令 `chmod` 来设置文件的权限。在有的 Linux 系统系统中已经为用户设置了这些权限。对于别的系统，可能要自己设置。

```
#chmod 111 /home/ftp/etc/passwd
```

10.4.4 FTP 文件

在目录 `/home/ftp/pub` 下的目录中有 FTP 文件，作为礼貌，应为 FTP 用户创建一个 `README` 文件和一个 `index` 文件。`README` 文件应包含对此目录中的文件的一个简单说明，`index` 文件应包含一系列文件和对文件中的内容的简单说明。

10.5 Web 服务器

在安装 Linux 期间，将自动地在系统中安装 Apache Web 服务器和所有的必需的目录

及配置文件。Linux 系统已经是一个全功能的 Web 站点。每当启动系统时，Web 服务器也同时启动，并不断地运行。Web 站点的数据文件的目录为/home/httpd/html。把 Web 页放在此目录或它的子目录下。

在 Apache 的站点 <http://www.apache.org> 中提供了它的 Web 服务器的联机支持和手册。在 Caldera 的站点 <http://www.caldera.com> 中也有联机支持。在这个两个站点中能得到关于开发 Web 站点和遇到的问题的更多的信息。

不需要做更多的事情。连接到一个网络中后，远程的用户就能访问 Web 站点。但是，了解一个 Web 站点是如何建立的将是十分有好处的。如果使用其他的系统，可能需要知道安装 Web 服务器和创建目录的过程。下面将介绍此过程。

安装在 Linux 系统中的 Web 服务器把 Web 站点设置在目录/home/httpd 中。为了管理此站点，还要设置一些目录。cgi-bin 有网关接口，文件 icons 拥有主页所用的图标。Web 页被放在目录/home/http/html 中。把主页的 index.html 文件也放在此目录下。配置文件在不同的目录/etc/httpd/conf 中。

下面列出 Web 服务器目录和 httpd 选项。

Web 服务器目录及说明：

/home/httpd Web 服务器的目录。

/home/httpd/cgi-bin 公共网关接口和脚本。

/home/httpd/icons 主页的图标。

/home/httpd/html Web 站点的 Web 页。

/etc/httpd/variant/conf 存放 Web 服务器的配置文件，variant 是一个特定的 Web 服务器，例如，/etc/httpd/apache/conf 中的配置文件是安装在 Linux 上的 Apache Web 服务器的。

httpd 选项及说明：

-d 如果不用默认的目录，用此选项指定一个目录。

-f 允许指定一个不同于 httpd.conf 的配置文件。

-v 显示版本。

还有一些免费的 Web 服务器软件可供使用。NCSA httpd Web 服务器是最早开发的服务器之一。Apache 和它非常相似，并改正了 NCSA httpd 服务器软件中的一些问题。可以从很多 Linux FTP 站点上下载服务器软件。在 Linux FTP 站点和它的镜像站点上，Web 服务器软件位于目录/pub/Linux/systems/Networks/info-systems/www/server 中。可以用 FTP 或 Netscape 来访问此目录。以后能从此目录中下载 Web 服务器的更新版本并升级服务器。

如果从一个 FTP 站点上下载一个 Web 服务器软件，需要解压文件并提取文件。此过程中将用到许多以前的目录，并在目录中新增一些文件和源代码。服务器软件包中将包括关于服务器目录的创建说明和编译软件的说明。

10.5.1 配置 Web 服务器

用目录/etc/httpd/conf 中的一些配置文件能配置 Web 服务器软件。根据需要不同配置

也有所不同,可以让 Web 服务器作为一个守护进程不间断地运行,或在 `inetd` 需要时再调用它。如果服务器使用频繁,应该让它作为一个守护进程直接运行。

`httpd.conf` 文件配置 Web 服务器。它列出了一系列变量和它们的值。每一项包含一个单独的变量名和一个值,中间用一个空格隔开。这些变量为 Web 服务器设置不同的特征。一些变量需要路径名,另外的一些只用设置关键字 `on` 或 `off`。在有的 Linux 系统中已经设置了这些变量。可以增加一个或改变一个变量。下面是一个例子,用变量 `ServerAdmin` 设定用户发送关于管理的问题的邮件地址。用想接收关于系统管理的邮件的地址来代替 `you@your.address` 项:

```
#ServerAdmin: Your address, where problems should be e-mailed.  
ServerAdmin you@your.address
```

在此文件中只设置一些变量,不需要每个变量都进行设置。一些设置需要系统的特殊信息。例如, `ServerName` 为 Web 服务器指定另外一个主机名。它必须是有效的主机名。假设系统的主机名是 `richlp.ix.com`, 还有另外一个主机名 `www.ix.com`。注意此项前有一个“#”号。去掉“#”号,并输入 Web 服务器的主机名以取代 `new.host.name`:

```
#ServerName allows you to set a hostname which is sent back to clients for  
#your server if it's different than the one the program would get (i.e.  
use  
#"www" instead of the host's real name).  
  
#ServerName new.host.name
```

文件 `srn.conf` 配置 Web 服务器的资源。它列出一组变量和它们的值。项的前面通常带有一个注释来解释此项。一些项已经被赋值,另外的一些要用户自己输入值。可以用标准的文本编辑器来改变。下例显示项 `DocumentRoot`。此项前有一个注释。项本身包含一个变量名和它的值,在此值为一个目录的路径名:

```
DocumentRoot /usr/local/etc/httpd/htdocs
```

以下列出 `httpd.conf` 中的变量及说明:

AccessConfig 文件 `access.conf` 的位置(默认的是 `/conf/access.conf`)。

AgentLog 动作的记录文件(默认的是 `logs/agent_log`)。

ErrorLog 记录错误的文件的位置;如果不是以 `/` 开头,则指相对于服务器根目录的地址(默认的是 `log/error_log`)。

Group 服务器作为一个守护进程运行的组 ID。

IdentityCheck 对远程用户进行识别检查。

MaxClients 限定服务器运行的最大数目,例如,限定能同时连接的客户数;超过此数,客户将不能注册(默认值为 150)。

MaxRequestsPerChild 在子进程退出之前,每个子进程允许执行的请求数(默认数是 30)。

PidFile 服务器将记录它的 `pid` 到此文件(默认的是 `/logs/httpd.pid`)。

Port 等待请求的端口。

ResourceConfig 文件 `srm.conf` 的位置(默认是 `conf/srm.conf`)。

ServerAdmin 管理员的 E-mail 地址。

ServerName 另外的服务器的主机名。

ServerRoot Web 服务器的用户的根目录;也是服务器的配置、错误和记录文件所放置的目录(默认的是 `/usr/local/etc/httpd`)。

ServerType 或是单机或是 `inetd`。

StartServers 启动的服务器数(默认的是 5)。

TimeOut 等待用户请求的秒数;如果在此时间内没有收到请求,用户将退出注册(默认的是 400)。

TransferLog 记录的路径(默认的是 `log/access_log`)。

TypesConfig MIME 配置文件的位置(默认的是 `conf/mime.conf`)。

User 服务器的用户 ID。

以下是 `srm.conf` 中的变量及说明:

AccessFileName 在每个目录中拥有访问控制信息的文件。

AddDescription 放置在服务器产生的索引中的文件的简短描述。

AddEncodingextensions-list 当浏览器查到压缩文件时,对它们进行解压;extensions 列表中包括不同的扩展名,用来指定压缩的类型,如 `gz` 指定 `gzip`,方法是在 `srm.conf` 文件中加入一行:
`AddEncoding x-gzip gz`。

AddIcon image file file-extensions 用于特定文件类型的图标;指定图像文件和一系列文件扩展名,如 `.mpg`、`.bin` 或 `.ps`。举例说明: `AddIcon /icons/movie.gif .mpg .qt`。

AddIconbyEncoding 指定图标并加一些解码的信息。

AddIconByType 用 MIME 类型来确定图标的使用。

AddLanguage language 允许指定一个文件的语言;如果一个文件使用浏览器能识别的语言,可以使用浏览器的内容可变的功能,指定语言和扩展名,使浏览器能显示此文件,如 `AddLanguage en .en`。

AddType type/subtype 允许用户不用编辑即可覆盖 MIME 类型,并指定特定的文件为某一种类型;在 Apache 的文件 `srm.conf-dist` 中有一些项能使用,如 `map` 文件,这些项被注释掉。

Alias alias-name path-name 为不同的路径名创建别名,如: `Alias /icons/ /usr/local/etc/httpd/icons/`。

DefaultType 文件使用的默认的 MIME 类型,服务器不能从文件的扩展名中找到的类型(默认是 `text/html`)。

DefaultIcon 不包含图标的文件所显示的图标。

DirectoryIndex Web 站点索引的文件名;这些文件被作为 HTML 目录的索引;多个项之间用空格分隔(默认是 `index.html`)。

DocumentRoot 用户提供的文件的目录;默认情况下,所有的请求都从此目录中获取

文件，但是符号链接和别名可能指向其他位置(默认是 `/usr/local/etc/httpd/htdocs`)。

FancyIndexing 为了做索引，加文件名和图标到文件列中；可以设置为 `on` 或 `off`
HeaderName 作为目录索引扩展的文件(默认是 `HEADER`)。

IndexIgnore file-list 被目录索引忽视的文件集；包括一些文件，如 `README` 或 `HEADER`
IndexOptions 索引选项。

LanguagePriority language-list 允许给定一些语言的一个优先顺序；忽略内容具有多义性，例如 `LanguagePriority en fr de`。

OldScriptAlias 同 `Alias`。

Readmename 服务器默认情况下查找的 `README` 文件的名字(默认是 `README`)。

Redirect 说明何处能找到不在服务器上的文件。

ScriptAlias alias-name path-name 控制包含服务器脚本的目录，如 `ScriptAlias /cgi-bin//usr/local/etc/httpd/cgi-bin/`。

UserDir 如果收到一个用户请求，把用户请求附加到用户主目录中的目录名(默认是 `/public_html`)。

文件 `access.conf` 决定一些服务是否允许用户访问，这些服务位于服务器的某一目录中。此文件中包括一系列的命令，每个命令都被包括在一组目录标识符中。开始的标识符是单词“`Directory`”加上一个目录路径名，并用尖括号括起来：`<Directory pathname>`。结束的标识符用相同的`<>`符号，只是在“`Directory`”前加有一个斜杠：`</Directory>`。在标识符中能加入一些命令。

在命令 `Options` 中，可以授权一些特征，如使用符号链接。用命令 `AllowOverride`，可以决定文件 `htaccess` 能覆盖哪些被命令设定的特征。用 `Limit` 命令，可以改变 `access.conf` 文件以控制 Web 用户访问系统。`Limit` 命令用了一组标识符，用法如 `Directory` 标识符。用`<Limit>`开始一个 `Limit` 命令并用`</Limit>`结束它。`Limit` 命令指明谁能访问 Web 服务器。还有另外一些可用选项。例如，`allow` 选项加上一列主机名，将限制只允许这些主机访问。`deny` 选项加上一列主机名将拒绝这些主机的访问。

在文件 `access.conf` 中，将找到两个目录项。第一个目录项意味着把此目录作为 Web 站点的根目录。Web 页也位于此目录中。确认在第一个目录项中使用的路径名是否就是 Web 站点使用的根路径名。如果不是，改正它。下面是一个 `Directory` 项的例子：

```
#This should be changed to whatever you set DocumentRoot to.
<Directory /home/httpd/htdocs>
Options Indexes FollowSymLinks
AllowOverride All
# Controls who can get stuff from this server.
<Limit GET>
order allow, deny
allow from all
</Limit>
</Directory>

#Place any directories you want access information for after this one.
```

下面说明 `access.conf` 中的内容。

目录标识符：

`<Directory path-name>...</Directory>` 指定设置控制的目录；以 `</Directory>` 结束。

命令：

`Options feature-list` 指定目录的服务器选项；放置在目录和限制命令中。

其特征值包括：

`All` 使用所有的特征。

`ExecCGI` 使 CGI 脚本可执行。

`FollowSymLinks` 使用符号链接。

`Includes` 允许使用 `include` 文件。

`IncludeNoExec` 允许 `include` 文件但不允许 `exec` 操作。

`Indexes` 允许用户查询索引。

`None` 禁用所有特征。

`SymLinksIfOwnerMatch` 使用符号链接前检查用户的 ID。

`AllowOverride feature-list` 目录中 `.htaccess` 文件能覆盖的控制选项。

其特征值包括：

`All` 不限制访问。

`AuthConfig` 通过基本认证限制访问。

`AuthName` 目录的授权名。

`AuthType` 目录的授权类型。

`AuthUserFile` 包含用户名和口令的文件。

`AuthGroupFile` 包含容许的组名的文件。

`FileInfo` 使用 `AddType` 和 `AddEncoding` 命令。

`Limit` 使用限制命令。

`None` 不允许访问。

`Options` 使用 `Options` 命令。

`<Limit>...</Limit>` 用下列指令控制对 Web 服务器的访问；项 `all` 指定所有的主机。

`allow host-list host-list` 允许在 `host-list` 中指定的主机访问。

`denies host-list host-list` 拒绝在 `host-list` 中指定的主机访问。

`orders options` 对 `deny` 和 `allow` 列表进行排序：`order deny,allow`。

`requires host-list host-list` 请求使用文件 `AuthUserFile` 进行认证。

10.5.2 启动 Web 服务器

可以用命令 `httpd` 手工启动 Web 服务器。此命令有一些选项。选项 `-d` 允许用户为 `httpd` 程序指定一个不同于默认的目录。用选项 `-f`，可以指定一个不是 `httpd.conf` 的配置文件。选项 `-v` 显示版本号。

在前面描述 `init` 脚本时提到，Linux 系统启动时将自动地启动 Web 服务器守护进程。

Web 服务器的启动脚本叫做 `httpd`，它位于目录 `/etc/rc.d/init.d` 中。一个叫做 `S85httpd` 的符号链接连接并运行目录 `/etc/rc.d/rc3.d` 中的 `rc` 程序。对于其他的 Linux 系统，可以把 Web 服务器的命令放在系统启动脚本如 `rc.local` 或 `rc.sysinit` 中。

如果希望 `inetd` 守护进程调用 `httpd`，在文件 `/etc/services` 和 `/etc/inetd.conf` 中放置一个关于 `httpd` 的项。`/etc/services` 中列出可用于系统中的不同的服务。对于 Web 服务器，可以输入 `http` 和指定一个端口，例如：

```
http 80/tcp
```

在文件 `/etc/inetd.conf` 中的 Web 服务器项和 FTP 项很相似。安装在有的发行版本的 Linux 系统中的 Web 服务器的路径名是 `/usr/sbin/httpd.httpd`，它没有参数：

```
http stream tcp nowait nobody /usr/sbin/httpd httpd
```

为了使 `tcpd` 能监视和控制 Web 服务器的请求，需要用路径名 `/usr/sbin/tcpd` 代替路径名 `/usr/sbin/httpd`：

```
http stream tcp nowait nobody /usr/sbin/tcpd httpd
```

还要为 `httpd.config-dist` 文件中的 `ServerType` 变量指定值 `inetd`：

```
#ServerType is either inetd, or standalone.  
ServerType inetd
```

Linux 没有配置成从 `inetd` 运行 Web 服务器。如果希望这样，要从自动启动的列表的启动守护进程列中除去 Web 服务器，并在文件 `/etc/services` 和 `/etc/inetd.conf` 中加入合适的项。

如果安装另一个 Web 服务器，要确保配置文件被正确设置和安装。如果用户在使用其他版本的 Linux，并且自己安装 Apache，将会注意到会出现带有扩展名 `.conf-dist` 的配置文件。需要把这些文件复制到有相同前缀，但后缀为 `.conf` 的文件中。Web 服务器将只从带有后缀 `.conf` 的文件中读取配置信息。然后可以进一步创建配置项。

为了检查 Web 服务器，启动 Web 浏览器并输入系统的 Internet 域名。对于系统 `turtle.trek.com`，如果用户输入 “`http://turtle.trek.com`”，则将显示放在 Web 根目录中的主页。最简单的方法是用 `lynx` 浏览器。启动 `lynx` 并按 `g` 以打开一行，输入自己系统的 URL。`lynx` 将显示 Web 站点的主页。不过首先要确保把文件 `index.html` 放在 `/home/httpd/html` 目录中。

也能用 `Telnet` 命令来检查 Web 服务器的操作。用 `Telnet`、系统主机名和 Web 服务器操作使用的端口 80：

```
telnet turtle.trek.com 80
```

10.6 Gopher 服务器

Gopher 服务器用一种索引方法来访问 Internet 资源，如数据文件和图形。

不像 FTP，Gopher 能提供给用户一个菜单以供选择。每个菜单项能引出另一个菜单

或另一个 Gopher 站点。从这个方面讲, Gopher 像 Web 那样, 允许用户从一个站点转到另一个站点以查找资源, 但是它又像 FTP 那样只列出资源, 没有文本或图形来进行解释。

Gopher 使用一个叫做 Gopher 的 TCP/IP 协议。它使得 Gopher 菜单文件进行高速地传输。Gopher 信息包含在 Gopher 文件中, 这些文件包含一系列记录, 这些记录在某些站点上是可访问的。每一记录被组织成五个区域来指定区域中的信息和这些记录的地址。每个区域之间用一个制表符分开: 类型、显示名字、选择器字符串、主机名和端口。

每种类型都对应了一个编号, 正如下面所列出的。显示名是对显示在 Gopher 菜单中的记录的一个描述。选择器字符串是记录的独特标识符。主机名是记录所处的系统的主机名, 端口是访问此主机系统时所用的端口(通常是 70)。

Gopher 的文件类型及说明:

- 0 文本文件
- 1 Gopher 目录。
- 2 CSO 电话号码服务器。
- 3 错误。
- 4 BinHex Machintosh 文件, HQX。
- 5 二进制的 DOS 文件。
- 6 UNIX UUencoded 文件。
- 7 全文本索引(Gopher 菜单文件)。
- 8 Telnet 连接, 包括远程主机的地址。
- 9 二进制文件。
- g GIF 图像文件。
- h HTML 文件。
- l Graphic 图像文件(不是 GIF)。
- M MIME 多部分混合信息。
- P Adobe PDF 文件。
- s 声音文件。
- T TN3270 Telnet 连接。

Gopher 是明尼苏达大学开发的, 而且还在不断地开发新的版本。可以从明尼苏达大学的 Gopher FTP 站点 boombox.micro.unm.edu 中目录 `/pub/gopher/Unix` 下取得一个 Gopher。也能从大多数的 Linux FTP 站点取得 Gopher 服务器软件。例如, 在站点 sunsite.unc.edu 和它的镜像站点的 `/pub/Linux/systems/Network/info-systems/gopher` 目录或 `/pub/packages/info-systems/gopher/boombox-mirror/Unix` 目录中能取得此软件。还有一个 GNU 免费的 Gopher 服务器软件, 叫做 GN Gopher。

用 rpm 或 glint 安装和配置 GN Gopher 服务器。明尼苏达大学为任何教育机构或非商业应用提供免费的 Gopher 软件。但是对于商业应用用户需要一个授权的费用。GN Gopher 服务器软件对任何人, 商业或非商业的均免费。明尼苏达大学也有一个更高的版本叫做 Gopher+, 目前它是一个商业的产品。

安装 GN 和明尼苏达大学的 Gopher 服务器有所不同。本节中的例子用的是明尼苏达大学的 Gopher 的 2.3 版和 2.20 版的 GN Gopher, 它们能从许多 Linux FTP 站点上取得。

用户还可能得到更新的版本。明尼苏达大学的 Gopher 软件包中包括 Gopher 客户程序和服务器软件。GN Gopher 软件包只包括服务器软件。

Gopher 客户程序是一种速度很快的命令行界面的客户程序。另外还有其他一些客户程序可以使用，如 xgopher。

10.6.1 Gopher 的用户帐号和数据目录

需要一个数据目录来存放 Gopher 数据文件，还要有能进行 Gopher 访问的用户帐号和组。虽然有的 Linux 系统已经创建了 Gopher 用户，但它没有创建数据目录。如果用户使用 Linux 的其他发行版本，可能需要自己创建 Gopher 用户。

10.6.1.1 Gopher 的用户帐号

为了更好地控制其他 Gopher 用户访问系统，系统上应该有一个名为 gopher 的用户帐号。有的 Linux 系统已经创建了 Gopher 用户帐号并对它进行了配置。在有的 Linux 系统的文件/etc/passwd 中将能发现下面的关于 Gopher 的项。

```
gopher:*:13:30:gopher:/home/gopher
```

口令域中的星号用来保护此帐号，以防止其他的用户通过此帐号注册，并因此控制它的文件或访问系统的其他部分。用户 ID 为 13，注释域为 gopher。注册目录为/home/gopher。当 Gopher 用户注册到系统，它将进入此目录。

对于某些 Linux 系统，需要自己创建 Gopher 帐号。可以给此帐号加上一些限制以阻止远程的 Gopher 用户访问系统中的其他部分。可以改变/etc/passwd 文件中关于此帐号的项，通过设置口令域为一个星号来阻止其他一般的用户访问它。

组 ID 是指 Gopher 组的 ID，它专为 Gopher 用户设置。可以在 Gopher 组上设置一些限制来限制所有的 Gopher 用户。下面是一个在文件/etc/group 中的关于 Gopher 组的项。其他的 Linux 系统中如果没有此项，可以加上。

```
gopher::30:
```

10.6.1.2 Gopher 数据目录

有些系统为它的 GN Gopher 服务器指定/home/gopher 目录作为 Gopher 数据目录。Gopher 文件的数据目录应该和 Gopher 用户的根目录相同。如果想使用不同的目录，需要用命令 chown 把目录的所有权改为 Gopher 用户。当配置 Gopher 服务器软件时，确保指定的目录是 Gopher 数据目录。否则，服务器将找不到 Gopher 文件。

明尼苏达大学的 Gopher:

在下面的例子中，用户下载了 Linux 版本的明尼苏达大学的 Gopher 软件包 gopher2_3.tar.gz。首先用 gunzip 解压此软件包，然后用 tar 展开文件和目录。将产生一个 gopher2_3 目录。此目录中包含文件和应用的不同的子目录。gopherd 目录中含有 Gopher 服务器的源代码，Gopher 目录中含有 Gopher 客户程序的源代码。doc 目录中包含文件，

包括帮助文件。

```
#gunzip gopher2_3.tar.gz
#ls
gopher2_3.tar
#tar xvf gopher2_3.tar
#ls
gopher2_3 gopher2_3.tar
#cd gopher2_3
#ls -F
Copyright MANIFEST Makefile.config.in Makefile.in README conf.h
config.guess config.h.in config.sub configure configure.in
copyright doc/ gopher/ gophfilt/ install-sh make.com
object/ patchlevel.h
```

为了安装 Gopher 服务器，首先要在 gopherd 目录中的一个配置文件中指定选项。需要提供一些信息，如在什么目录中放置 Gopher 菜单文件。接着编译 Gopher 软件。编译软件只不过是 Gopher 在源代码目录下输入命令 `make`。`make` 将使用 `Makefile` 文件正确地编译 Gopher 程序。明尼苏达大学的 Gopher 2.3 有配置功能，它能自动决定如何配置系统并为系统创建合适的 `Makefile` 文件。任何系统的特定的信息都被精确地设定在配置文件中。

10.6.2 配置明尼苏达大学的 Gopher 服务器

在创建 Gopher 服务器之前，要用文件 `gopherd.conf` 和 `gopherdlocal.conf` 来配置它。在明尼苏达大学的 Gopher 服务器中，这些配置文件在子目录 `gopherd` 中。`gopherd.conf` 用来配置系统的一些特定特征，如限制的链接数。`gopherdlocal.conf` 为 gopher 服务器提供信息，如管理员的名字和控制指定的远程系统的访问。

`gopherd.conf` 和 `gopherdlocal.conf` 文件中包含对 gopher 服务器的配置说明。文件中已列出一组命令的默认规范。用户只需删除每行开头的“#”，以取消它的注释功能，并按需要改动任何值即可。

可以设置选项如允许的最大用户数或传输文件的压缩方法等。输入选项时，首先输入选项的说明符和一个冒号，然后加上一个空格和选项的值。下面的例子是设定最大的用户数：

```
MaxConnections: 15
```

必须为系统中的 Gopher 服务指定一个别名。用 `hostalias:` 记录来指定。通常此记录指的是系统的主机名的全称，有的系统是主机名的一部分，比如 Gopher。下面的例子中 `hostalias` 表示主机名的全称的别名 `garnet.train.com`：

```
hostalias: garnet.train.com
```

还需要为文件 `gopherdlocal.conf` 指定全路径名。用命令 `include` 加上文件 `gopherdlocal.conf` 的路径名即可。在有的 Linux 系统中，`gopherd` 服务器、文件 `gopherd.conf`

和 `gopherdlocal.conf` 应该安装在目录 `/usr/sbin` 中。当 `gopherd` 服务器运行时，它首先读取文件 `gopherd.conf` 的配置信息，接着读取文件 `gopherdlocal.conf` 的特定的配置信息。在文件 `gopherd.conf` 中要为 `include` 指令指定 `gopherdlocal.conf` 的全路径名，在此为 `/usr/sbin/gopherdlocal.conf`：

```
include: /usr/sbin/gopherdlocal.conf
```

用 `ignore` 和 `ignore_patt` 选项，可以限制 Gopher 用户可访问的文件。`ignore` 将拒绝访问带某一扩展名的文件，例如，`ignore: conf` 命令将拒绝任何对扩展名为 `.conf` 文件的访问。`ignore_patt` 限制有特定模式的文件，例如，`ignore_patt` 命令将限制对名字中包含模式“`bin`”的所有文件的访问。

`gopherdlocal.conf` 文件中拥有本地的用户定制信息。此文件中的项将覆盖文件 `gopherd.conf` 中的可与之相比较的项。在文件 `gopherdlocal.conf` 中用户可以指定一些管理信息，如系统管理员的名字和 Gopher 服务的描述。在此文件中有两项是关于系统管理员的：`Admin:`和 `AdminEmail:`。用 `Admin:` 项来添加一个系统管理员的名字。此外，还可以加一些其他的信息，如一个电话号码。用 `AdminEmail:`项来指定系统管理员的 E-mail 地址。

```
Admin: Richard Petersen
AdminEmail: rp@richlp.com
```

有些项提供关于 Gopher 服务器的信息。项 `Abstract:`将显示给系统中的用户一个布告。它简明地描述 Gopher 服务器提供的服务。项 `Language` 将告诉说明系统的大多数的文件使用什么语言。用户应该设定这两项。别的信息项都是可选的。在项 `Org:`中可以输入组织名，在 `Loc:` 项中可以输入地址。项 `BummerMsg:`指定一个信息，当站点中有太多的用户访问而不能再被其他用户访问时，产生错误并显示此信息。

用 `access:`项可以限制其他用户访问。`Access:` 项的格式如下：

```
access: hostname permission-list num
```

`hostname` 是主机名或是网络或远程系统的 IP 地址。`num` 指定的远程网络或系统中同时允许多少用户访问。`permission-list` 设定远程网络或系统的用户的权限。有 4 种权限：浏览、FTP、读和查找。为了禁止某权限，可以在它前面加上一个感叹号“`!`”。浏览权限允许用户列出目录中的文件，FTP 允许系统作为一个 FTP 网关，读权限允许访问文件，查找权限允许访问索引。

对于 `hostname`，可以指定一个网络地址，一个特定系统的地址或用默认项以使得任何用户都可以访问。默认项是指用关键字“`default`”。下面的例子中，Gopher 服务器对任何用户开放读和查找权限，同时关闭浏览和 FTP 权限。没有网络或系统中能同时有 15 个以上的用户访问 Gopher 服务器。

```
access: default !ftp read search !browse 15
```

能用 `access:` 项加上网络的域名来控制一个网络的访问。下面的例子中，网络 `train.com` 中最多能有 5 个用户访问 Gopher 服务器，但是没有浏览功能。

```
access: train.com ftp read search !browse 5
```

能用网络或系统的 IP 地址代替域名。对于网络，能只用 IP 地址中的网络部分。不要

忘记在结束时用一个句号。在下面的例子中，网络地址为 199.189.的网络被拒绝使用系统作为一个 FTP 网关。同时只能有 7 个用户访问。

```
access: 199.189. !ftp read search browse 7
```

为了给特定的系统提供访问，可以输入主机的系统名称。在下面的例子中，IP 地址为 204.166.189.21 的系统能完全访问 Gopher 系统。如果系统是个人使用的单机，可以指定此系统只能有一个用户访问。

```
access:204.166.189.21 ftp read search browse 1
```

在编译 Gopher 服务器之前，应该检查 Makefile.config 和 conf.h 文件中的某些配置。在文件 Makefile.config 文件中首先检查要操作的 Gopher 程序的路径名。它被指定给变量 PREFIX。如果想在不同的目录中操作 Gopher，改变赋给变量 PREFIX 的默认路径名。Linux 将从目录/usr/sbin 中查找服务器程序。应给变量 PREFIX 赋值为此目录。例如：

```
PREFIX=/usr/sbin
```

还应该检查 DOMAIN、SERVERPORT、SERVERDATA 和 SERVEROPTS 变量。DOMAIN 变量指定域名的网络部分。例如 richlp.ix.com 的网络部分应是.ix.com。一定要带有前一个圆点。如果域名命令能显示全域名，可以让此项为空。

```
DOMAIN=.ix.com
```

SERVERPORT 变量被设置为 70，为做通常的访问应该保留此值。变量 SERVERDATA 有 Gopher 数据文件所在的目录，默认值为/gopher-data。如果想用另一个目录，能把此目录赋给 SERVERDATA。在有的 Linux 系统中应该设置为/home/gopher。变量 SERVEROPTS 被赋以一系列选项以控制 Gopher 服务。可以在标题中加入日期和时间或设置的最大用户数。这些选项将在下面列出。

```
SERVERDATA=/home/gopher
```

在文件 conf.h 中，可以设置一些特征的值，如超时的时间，和某一特定操作，如查看一个图形文件时使用的程序。在文件 conf.h 中含有一列定义项，它们和 C 编程中的定义项很相似。

每项以#define 开头，加上一个大写的变量和一个值。为了改变一个特定变量的值，用文本编辑器编辑此文件，删除此值并输入一个新值。默认的值已经输入。下面的例子设定打印选项为值：lpr 命令。

```
#define PRINTER_COMMAND "lpr"
```

对于 Gopher 客户程序可以用 CLIENT1_HOST 项来指定它将连接的默认的 Gopher 服务器。下面的例子连接到一个 Gopher 的 Internet 站点：gopher.tc.umn.edu：

```
#define CLIENT1_HOST gopher.tc.umn.edu
```

文件 conf.h 容易被混淆。它是一个 C 程序，不是一个 Shell 脚本文件。它和其他配置文件不同。在许多配置文件中，一个“#”是一个注释。而在文件 conf.h 中，它是一个定义命令的开始。在 conf.h 中，一个注释是用“/*”开头并以“*/”结尾的。加注释的项便失去作用，要使它起作用只需删除“/*”和“*/”符号。

文件 `conf.h` 的默认项工作得很好,也可以按需要对它们做改变。然而,改变用于 Linux 系统的项时要小心。Linux 将使用那些不是专为其他系统设计的项。如果看到一个项前有一个 `#if defined(system)`, `system` 是一个操作系统的名字,那么下面的项将用于此系统,直到下一个 `#endif`。如 `#if defined(sun)` 将只用于 Sun 系统。`conf.h` 文件中有一大部分是为 VMS 操作系统定义的,在文件的开头和结尾有些项是用于 Linux 的。

现在可以开始编译 Gopher 软件。能用下面的命令编译 Gopher 客户程序和 Gopher 服务器:

```
#make install
```

可以分别编译客户程序或服务器,用命令 `make` 加上项目 `client` 或 `server`:

```
#make client
```

```
#make server
```

它将创建一个名为 `gopherd` 的服务器程序。可以从 `inetd` 直接调用 `gopherd`。
明尼苏达大学的 Gopher 的 `Makefile.config` 和 `conf.h` 文件中的项及说明如下。

Makefile.config 文件:

PREFIX 安装软件的基本的路径名(默认的是 `/usr/local`)。

CLIENTDIR Gopher 客户程序被安装的目录(默认的是 `/usr/local/bin`)。

CLIENTLIB Gopher 客户程序的帮助文件。

MAN1DIR Gopher 客户程序的帮助文件被安装的目录。

MAN8DIR Gopher 服务器的帮助文件被安装的目录。

SERVERDIR Gopher 服务器和配置文件被安装的目录(`gopherd` 和 `gopherd.conf`——默认的是 `/usr/local/etc`)。

CLIENTOPTS Gopher 客户程序的选项:

-DONOMAIL 远程用户不能邮寄文件。

-DAUTOEXITONU 同时处理 `q` 和 `u`, 并自动地从主菜单中退出(如果 Gopher 被其他应用程序调用时,此选项很有用)。

DOMAIN 如果系统不能返回有效的全域名,需要指定系统的网络地址;否则,保持为 `null`。

SERVERPORT=num 设置 Gopher 服务器用来等待请求的端口,通常为 70。

SERVERDATA=pathname 对于 Gopher 服务器可用的 Gopher 数据文件的所在目录;对于有的 Linux 系统此目录为 `/usr/lib/gopher-data`。

SERVEROPTS=options-list Gopher 服务器的选项:

-DADD_DATE_AND_TIME 往 Gopher 标题中增加日期和时间。

-DDL 设置支持 `dl` 数据库功能。

-DLOADRESTRICT 限制用户访问。

-DSETPROCTITLE 设置 `ps` 显示的进程标题。

DLPATH 到 `dl` 数据库的路径。

conf.h 文件:

CLIENT1_HOST 首先连接的主机(列在下面的是默认的主机):

```

#define CLIENT1_HOST gopher.tc.umn.edu.
CLIENT2_HOST 连接的主机:
#define CLIENT2_HOST "gopher2.tc.umn.edu"
CLIENT1_PORT 首先连接的端口(默认的是 70):
#define CLIENT1_PORT 70
CLIENT2_PORT 连接的端口(默认的是 70):
PAGER_COMMAND 文本的分页命令。
MAIL_COMMAND Gopher 用户发送邮件的命令(默认的是/bin/mail)。
TELNET_COMMAND Telnet 会话使用的命令(默认的是 telnet)。
PAINTER_COMMAND 打印的命令(默认的是 lpr)。
PLAY_COMMAND 播放声音的命令(默认的是/bin/false)。
IMAGE_COMMAND 显示图像的命令(默认的是 xloadimage)。
HTML_COMMAND 显示 Web 页的命令(没有默认值, 一般用 lynx)。

```

10.6.3 启动明尼苏达大学的 Gopher

可以从命令行中直接启动 Gopher 服务器, 或用 `init` 脚本引导系统时启动, 或是当收到一个 Gopher 服务请求是通过 `inetd` 来启动。明尼苏达大学的 Gopher 服务器选项如下所示:

- C 取消目录请求的高速缓存。
- c 不受 `chroot` 的限制运行; 允许通过符号链接访问 Gopher 数据根目录外的文件, 如系统的帮助文件; 有潜在的安全问题, 应该和 `-u username` 选项一起使用。
- D 启动调试功能。
- I 用 `inetd` 调用 Gopher。
- L *num* 指定最大的平均负载。
- l *logfile* 记录到日志文件的连接。
- o *conf-file* 指定备用 `gopherd.conf` 配置文件。
- u *username* 在指定的用户名下运行 `gopherd`; 为安全目的提供附加的限制。
- U *userid* 在 *userid* 下运行 `gopherd`; 为安全目的提供附加的限制(同-u)。

有的 Linux 系统已经配置好了 Gopher 系统, 使之通过 `inetd` 来运行 Gopher。此配置将把 Gopher 数据目录定为 `/home/gopher`。在 `/etc/services` 文件中将找到一个关于 Gopher 的项。它通过端口号和协议指定服务名。在其他的 Linux 系统中, 需要自己输入。

```
gopher 70/tcp
```

在文件 `/etc/inetd.conf` 中也有一个关于 Gopher 的项。目前此项引用 GN Gopher 服务器 `gn`。可以把它改为引用 `gopherd` 服务器。把项中的 `gn` 替换为 `gopherd`, 并加一个 `-I` 选项和 Gopher 数据目录的路径名。选项 `-I` 指定 `gopherd` 被 `inetd` 调用。可以加一个端口号, 在此为 70。此项还被设置为用 `tcpd` 来监视和控制远程用户的访问。下面的例子列出此项。Gopher

文件在目录/home/gopher 中，端口号为 70:

```
gopher stream tcp nowait root /usr/sbin/tcpd gopherd -I
/home/gopher 70
```

如果不想使用 tcpd，能用 gopherd 服务器程序的路径名取代它:

```
gopher stream tcp nowait root /usr/sbin/gopherd gopherd -I
/home/gopherd 70
```

如果在命令行模式下启动 gopherd 服务器，应输入 gopherd 和它的参数、gopher 数据文件目录的路径名和端口号。gopherd 有一些选项，列在下面。作为防范措施，可以用 -u 选项指定一个非 root 的所有者来运行 gopherd。

需要首先创建一个用户并在它的 passwd 项中输入一个“*”。用户的根目录将是 Gopher 数据目录。有些 Linux 系统已经创建一个名为 Gopher 的用户。下面的例子中，gopherd 服务器程序的全路径名为 /usr/sbin/gopherd，Gopher 数据文件在 /home/gopher 中，端口号为 70:

```
/usr/sbin/gopherd -u gopher/home/gopher 70
```

为使 Gopher 服务器在启动系统时自动启动，需要在目录/etc/rc.d/init.d 中创建一个 init 脚本。本章的开始部分对 init 脚本和如何创建 gopherd 服务器的 init 脚本有详细的说明。

下面列出 gopherd.conf 和 gopherdlocal.conf 文件中的项及说明。

gopherd.conf 文件:

hostalias: *DNS-alias-name* 用此主机名来代替系统的主机名，必须有一个 DNS，用来引用一个系统，比如 gopher.ix.com 上 gopher 服务器，命令如下:

```
hostalias: gopher.turnip.com
```

cachetime: *seconds* 一个缓存文件中保存的 Gopher 目录的时间。

viewext: *extension gophertype Prefix gopher+type[Language]*

改变文件名的扩展名为一个特定的 Gopher 类型；大多数的名字已经在文件 gopherd.conf 中设置了。第一个参数是一个扩展名，如.gif；第二个参数是单个字符的 Gopher 类型(1、0 或 I 等)；第三个参数是要加到一般文件名路径前的前缀；第四个参数是 gopher+视图特征或 Internet Media 类型(正规叫做 MIME 内容类型)，如 image/gif；可选的第五个参数是指定文件使用的语言以替换默认的语言。如:

```
viewext:.jpg I 9 image/JPEG
```

```
viewext:.html h 0 text/html
```

ignore: *extension* 忽略带有指定扩展名的文件；这些文件不提示给 Gopher 的用户，命令如下:

```
ignore:binignore_patt: regular-expression
```

忽略匹配 *regular expression* 的文件；这些文件不提示给 Gopher 的用户。

blockext: extension 映射带有指定扩展名的文件到属性块中。

decoder: extension program 当检索到文件时，用指定的程序对带有指定扩展名的文件进行操作；和压缩文件一起使用，如：

```
decoder: .gz /usr/gnu/bin/zcat
```

pids_directory: path-name 一个存放 pid 文件的临时目录。

maxconnections: num Gopher 服务器能同时处理的并发连接的数量。

gopherdlocal.conf 文件：

admin: administrator-name-and-info 管理员的名字和附加信息，如电话号码。

adminemail: email-address Gopher 服务器的管理员的 E-mail 地址。

site: site-description 站点的描述名。

loc: address 站点的地址：街道、城市等。

geog: 站点的经度和纬度。

language: default-language 站点使用的默认语言。

secureusers: filename 列出经过认证的主机和网络的文件。

bummermsg 当拒绝一个客户访问时显示的信息。

access: domain name access-list 允许用户决定谁能浏览目录、阅读文件和搜索系统，第一个参数是域名、IP 地址或默认；第二个参数是一列逗号分隔的单词，用来控制访问，这些单词是：**browse**、**read**、**search** 和 **ftp**。每个单词前加一个“！”就可拒绝此种访问；如：

```
default !browse,read,search,!ftp
```

该设置默认为拒绝浏览和 FTP，但允许读和查找。如果首先设定默认值，以后的项中将继承这些属性，只需要指定不同的项。另外，还可以通过增加一个可选的参数来指定并发事务的访问数，如：

```
access: default browse,read,search,ftp 5
```

10.6.4 GN Gopher 服务器

有的发行版本的 Linux 系统已经配置了 GN Gopher 服务器。如果用户选择完全安装的选项，GN Gopher 将自动安装，如果选用推荐的安装选项，需要用 rpm 或 glint 来安装 *gn-2.22-1.i386.rpm*。无论选择哪种安装方式，都可以使用 Gopher 服务器。可以开始把 Gopher 文件和菜单加到 */home/gopher* 目录中。要运行 *mkcach* 来使得 Gopher 客户程序能访问 Gopher 文件。然而，如果用的是 Linux 的其他版本，需要能从一个 Linux FTP 站点中下载标准的 GN Gopher 服务器软件包 *gn-2.20.tar.gz*。然后配置并安装它。解压和展开此文件将创建一个名为 *gn-2.20* 的目录，其中放有 GN Gopher 服务器的源程序和文件。

GN Gopher 软件包中包含一些在编译之前需要为 Linux 配置的源代码。还有一个配置

文件, `config.h` 和一个 `Makefile` 文件中包含编译命令。在 `doc` 子目录中的文件 `INSTALL` 有关于如何创建 GN 服务器和设置 Linux 站点的详细说明。在目录 `/doc/examples` 中有 GN Gopher 菜单的大量的例子。

10.6.4.1 配置GN Gopher服务器

为了配置 GN Gopher 服务器, 需要改变文件 `config.h` 和 `Makefile` 中的项。有些项是必须配置的, 需要指定它们。在文件的开头的标题 “Compulsory items to fill in” 中指定了这些项。必须输入主机名和 Gopher 数据文件目录的路径名, 以及指定使用的 Linux 系统。还可以按照需要进行其他的定制。在文件 `conf.h` 中每项的格式为:

一个 `#define` 加上一个变量和它的值。可以改变此值。

下面列出的项对 Caldera Network Desktop 版本的 Linux 是有效的。实际应用中, 应该用系统主机名取代 `gannet.train.com`:

```
#define GN_HOSTNAME "garnet.train.com"
#define ROOT_DIR "/home/gopher"
#define LINUX
#define MAINTAINER "mailto:justin@garnet.train.com"
#define ROOT_MENU_NAME "GN -- A gopher/HTTP Server"
#define GN_LOGFILE "/var/log/gn.log" /* "/path/to/gn.log" */
#define MIME_TYPE_FILE "/usr/local/gn_mime.types"
#define WAISGN "/usr/sbin/waisgn"
```

在必须配置的项后有一些其他的项, 它们在默认状态下能很好地工作, 但也可以按需要对它进行改变。比如能通过指定 `DEFAULTPORT` 项为其他的端口号, 从而使用该端口。可以设置 `TIMEOUT` 项为超时的时间, 设置 `MAXDEPTH` 项为菜单查找的最大深度。在 `Makefile` 中, 可以指定用户使用的 C 编译器的类型和服务器程序存放的目录。用户将能找到两个关于 C 编译程序 `CC` 的类型的项:

1. 一项指定为 `gcc`, 前面加有一个 “#” 把它注释掉。
2. 而另一项指定为 `cc`, 前面没有 “#” 号。

可以在 `cc` 前加一个 “#”, 而去掉 `gcc` 前的 “#”。这样将设置 `CC` 项为 `gcc` 编译器。另外在 `Makefile` 中可以设置 `SERVERBINDIR`, 它是用户将存放的服务器程序的目录路径。`BINDER` 是 `mkcache` 和 `uncache` 程序的目录。设置这些目录为希望存放程序的目录, 如 `/usr/sbin`。在 Caldera Network Desktop 中用 RedHat 版本, 目录 `SERVERFINDER` 应被设置为 `/usr/sbin`, 此目录中有守护程序。虽然 `mkcache` 和 `uncache` 程序可以放在任何目录中, 最好还是把它们和 GN 服务器放在一起, 只需把 `BINDER` 也设定为 `/usr/sbin` 即可以做到这点。

在这些变量的下面, 是关于 `include` 目录的项。如下所示, 除去关于 Linux 的项前面的 “#”, 以使它们能起作用:

```
#INCLUDES= -I.. -I../gn
#For Linux use
INCLUDES= -I.. -I../gn -I/usr/include/bsd
```

在文件 `Makefile` 中，再往下便是一个关于库函数的空项，如下所示：

```
# Libraries to be included.
LIBS =
```

下面的项是为不同的系统指定不同的库函数，这些项被注释掉。关于 Linux 的项如下所示：

```
#For Linux use
#LIBS = -lbsd
```

可以除去 `LIBS` 项前的“#”号，或是在上面的空 `LIBS` 后加上 `-lbsd` 的值：

```
LIBS= -lbsd
```

配置完成后，就可以输入 `make` 来创建一个服务器。它将产生两个可执行的服务器程序 `sgn` 和 `gn`。`gn` 是为 `inetd` 使用的，而 `sgn` 是可以直接使用的一个独立的守护进程。还将产生两个应用程序：`mksync` 和 `unsync`。然后输入 `make install`，在系统中安装这些程序。

10.6.4.2 启动GN Gopher服务器

现在可以启动服务器了。在启动 Gopher 服务器之前，应该首先创建一些 Gopher 菜单文件。用户可以启动服务器做为一个独立的守护或通过 `inetd` 来启动服务器。

有的 Linux 系统已经把系统配置为通过 `inetd` 运行 GN Gopher 服务器。创建和安装 GN Gopher 服务器后，为了有一个全功能的 Gopher 服务器，必须创建 Gopher 数据文件并把它们放在目录 `/home/gopher` 中。

为了使用 `inetd`，在文件 `/etc/services` 和 `/etc/inetd.conf` 中必须有关于 Gopher 服务器的项。有的 Linux 系统已经包括此项。可以在文件 `/etc/services` 中找到下面的项：

```
gopher 70/tcp
```

在文件 `/etc/inetd.conf` 中，用户将能找到关于 GN Gopher 服务器的项。注意 GN Gopher 服务器是被 `gn` 调用的。此版本的 GN 服务器通过 `inetd` 工作。另外，`tcpsn` 用来监视和控制 Gopher 访问。

```
gopher stream tcp nowait root /usr/sbin/tcpsn gn
```

也可以在启动系统时用 `init` 脚本或直接从命令行启动 GN Gopher。在这两种情况下，服务器使用 GN Gopher 的 `sgn` 版本来启动，并以一个独立的守护进程运行。如果想在引导时启动 Gopher，需要创建一个目录 `/etc/rc.d/init.d` 中的 `init` 脚本。输入 `sgn` 和参数，用来指定 Gopher 数据文件的目录的路径名和端口号(通常为 70)。端口号用 `-p` 选项来指定。作为一种预防措施，`sgn` 将按照用户在文件 `config.h` 中的 `USERID` 项的指定来自动运行。

```
sgn -p 70 /home/gopher
```

下面是 GN 服务器配置文件：

`GN_HOSTNAME` 显示服务器的主机名；在此是系统的主机名。

`ROOT_DIR` 设置 Gopher 数据目录；对于有的 Linux 系统，它是 `/home/gopher`。

`LINUX` 定义 Linux 作为操作系统；默认的是 `Sun_OS`；需要用 `LINUX` 代替它。

`MAINTAINER` 管理员的邮件地址。

ROOT_MENU_NAME 用户想显示的作为 Gopher 菜单标题的名字。
GN_LOGFILE GN 记录文件的位置。
MIME_TYPE_FILE MIME 配置文件的位置；默认的是 `/usr/local/gn-mime.types`；可以把文件放在任何地方，但是确认设置正确的目录。
WAISGN 控制 WAIS 索引 `/usr/sibn/waisgn` 的 `waisgn` 程序的路径名。
DEFAULTPORT 设置 Gopher 访问的默认端口，当前是 70。
TIMEOUT 设置等待请求的超时时间。
MAXDEPTH 设置查询菜单的最大长度。
USERID 为了安全，运行 GN 服务器所使用的用户 ID。
GROUPID 为了安全，运行 GN 服务器所使用的组 ID。
DECOMPRESS 解压文件所用的程序；默认的是 `/usr/local/bin/zcat`。
MENUFNAME 设置菜单文件所使用的名字；默认的是 `menu`。
TEMPDIR 设置临时目录；默认的是 `/tmp`。
 GN 服务器的 Makefile 文件如下：
CC 设置 C 编译器；默认的设置是 `cc`；可以把它改成 `gcc`，命令是：`CC=gcc`。
INCLUDE 设置 Include 目录；应该被设置成使用 `/usr/include/bsd`。
SERVBINDIR GN 和 SGN 服务器的程序；默认的是 `./bin`；应该设置为 `/usr/sbin`。
BINDIR 存放可执行二进制文件的目录；默认的是 `./b`；应该设置为 `/usr/bin`。
LIBS= 设置使用的库；除去关于 Linux 的那一项前的“#”号；应该设置为 `-lbsd`。

10.6.4.3 测试Gopher服务器

当 Gopher 服务器运行时，可以用 Telnet 或 Gopher 客户程序来测试它。用 Telnet 时，首先 Telnet 到自己的系统中并指定 Gopher 使用的端口。下面的命令将测试在系统 `garnet.train.com` 中的一个 Gopher 服务器，将显示启动信息：

```
telnet garnet.train.com 70
```

如果按 Enter 键，将显示服务器上的 Gopher 主目录中的菜单项，并以下面的格式输出：

```
type display-name selector hostname port
```

下面是一个测试 Gopher 服务器的例子：

```
#telnet garnet.train.com 70
Trying 127.0.0.1 ...
Connected to garnet.train.com
Escape character is '^]'.

0About My Weather Site 0/intro  garnet.train.com  70  +
1California Weather Information  1/calif  garnet.train.com  70  +
1New york Weather this week  1/newyork  garnet.train.com  70  +
1The Weather in Hawaii  1/weather/hawaii/
garnet.train.com  70
```

还可以用 Gopher 客户程序来访问 Gopher 服务器。下面的例子中，用 `garnet.train.com`

中的 Gopher 客户程序访问本系统中的 Gopher 服务器。将显示 Gopher 服务器的主菜单，用户能从中选择或访问。

```
gopher garnet.train.com
```

10.6.4.4 GN Gopher 目录

GN Gopher 目录下的 menu 文件中包含菜单项。每个 Gopher 目录有它自己的 menu 文件，文件中有关于每个菜单项的记录。GN Gopher 使用和明尼苏达大学的 Gopher 相同的记录集，仅有一些例外。在文件 menu 中，为每个菜单项列出路径、名字、号码和摘要。然而，在路径项的路径名之前，必须指定 Gopher 类型。它通常是一个数字，如 0 表示文本文件，1 表示目录。

```
Name=About My Weather Site
Path=0/intro
Type=0
Numb=1
Abstract=Important Weather Information
```

GN Gopher 要求为每个目录创建一个.cache 文件。在目录中执行 mkcache 程序即可生成一个.cache 文件。可以在每个 Gopher 目录中用此方法创建.cache 文件，或从 gopher-data 主目录中执行 mkcache 加上-r 选项。命令 mkcache -r 将为当前目录和它的所有子目录产生.cache 文件。在有的 Linux 系统中，可以从/home/gopher 目录中运行 mkcache -r 来为所有 gopher 目录创建.cache 文件。

GN Gopher 也完全支持 Web 页。它能显示 Web 页，并可以通过 Web 页面的 HREF 引用它的菜单项。参见 GN 文件中关于这些功能的详细描述。

10.6.4.5 Gopher 索引: gopherindex

用 gopherindex 命令能为 Gopher 数据文件创建全文索引。

gopherindex 在明尼苏达大学的 gopher 软件包中，它用 waisindex 来执行它的索引，所以用户必须安装 WAIS。它使用参数的几个选项和一个 Gopher 数据目录。在数据目录中的所有文件将被做一个索引。-N 选项指定用在一个 Gopher 菜单中的索引文件的一个描述。下面的例子将对目录/home/gopher/baseball 中的文件进行索引：

```
/usr/sbin/gopherindex -v -N "Search CIS Services Short Courses"
/home/gopher/baseball
```

除了使用 gopherindex，还能直接用 waisindex 并对索引文件创建一个连接：

```
waisindex -r /home/gopher/baseball
```

在目录连接或 menu 文件中可以为索引文件创建连接项。类型 7 表示一个文件 WAIS 索引类型：

```
Type=7
Name=Baseball Index
Host=+
```

```
Port=+
Paht=7/.index/index
```

10.7 安装 PPP 服务器

要在 Linux 上安装 PPP 服务器，接受拨入的电话，连上整个公司的以太网网络，再通过路由器接上因特网，过程并不简单，整台 Linux 有许多地方需要设置。本章内容能够帮助用户顺利地安装 PPP 服务器。

要建立 PPP 服务器有三个条件：

1. 调制解调器必须要能接电话。
2. Linux 必须能通过串行线显示登录提示符。
3. 建立 pppd 的帐号。

10.7.1 让调制解调器接电话

用 Linux 来设置太麻烦，所以可以用 Windows 3.1 的终端来设置调制解调器，执行 Windows 的终端。把调制解调器设置成 ATSO=1，就是调制解调器在电话响一声后就接，ATSO=2 就是响两声后接。

10.7.2 通过串行线显示登录提示符

在/etc 目录下，有一个 inittab 文件，其中有一段内容要设置如下：

```
# 串行线 s 这是本文的设置，可能与用户的不同。
#s1: 45: respawn: /sbin/getty ttyS0 19200 vt100 s2: 45: respawn: /sbin/getty
ttyS1 19200 vt100
```

每一行之前若加上“#”，代表注释。

这段内容的各个域的含义如下：

```
id: runlevel: action: /etc/getty [options] line [speed [term [lined]]]
```

其 getty 必须为 getty_ps。有了这段设置，重新启动，并连上终端后，执行 ps aux 命令，应可看到下所示的一些数据：

```
.... root 88 232 v02 S 10: 58 0: 00 /sbin/getty tty2 3840
root 88 233 v03 S 10: 58 0: 00 /sbin/getty tty3 3840
root 92 264 psf S 11: 33 0: 00 /sbin/getty ttyS1 19200
```

/dev/ttyS1 是给拨入的连接使用的。

/dev/cua1 是拨出用的。

以上两者都是 com2 端口。

到目前为止，如果用调制解调器拨进来，将可出现登录提示符的画面如下：

```
login:
```

此时整条串行线已通，可以登录，并执行一些简单的命令，但本地端和远程端并未开始使用 PPP 协议。

10.7.3 提供 PPP 服务

为了提供 PPP 服务，要建立一个帐号，如：ppp01，它的登录 Shell 可以调用 pppd，并和对方沟通。

做法如下：运行 adduser 命令来增加帐号：

```
Linux#adduser
-----这以下为 adduser 的用户界面：

Adding a new user. The username should be not exceed 8 characters in
length, or you may run into problems later.
Enter login name for new account (^C to quit): ppp01
Editing information for new user [ppp01]
Full Name: ppp01
GID [100]: 100
Checking for an available UID after 500
501...502...503...504...505...506...507...508...
First unused uid is 509
UID [509]: 509
Home Directory [/home/ppp01]: /home/usr/ppp01
Shell [/bin/bash]: /etc/ppp/ppplogin
Password [ppp01]: xxxxx
-----以上为 adduser
```

一般登录 Shell 为/etc/bash 或/etc/csh，但为了在用户登录时就执行 pppd，建立一个 PPP 连接，所以写了一个可执行的脚本文件，内容如下：

```
-----以下为/etc/ppp/ppplogin
#!/bin/csh
exec /usr/etc/pppd passive netmask 255.255.255.0 proxyarp : 140.xxx.xxx.xxx
-----以上为/etc/ppp/ppplogin
```

passive 的意思是 pppd 一开始会尝试协商，若没有回应，则等待。

netmask 255.255.255.0 设置子网掩码。

proxyarp 向系统的 ARP 表中加入一项记录。

140.xxx.xxx.xxx 设置拨入端的 IP 地址。

关于 pppd 这部分的选项，可用 man pppd 命令来查看。

执行 `adduser` 命令后, `/etc/passwd` 文件的内容如下:

```
ppp01: (encrypted password): 505: 100: ppp01: /home/usr/ppp01:
/etc/ppp/ppplogin
```

上述是服务器端建立的方法, 至于客户端连接 PPP 服务器就不多做介绍了。

至此大功告成, 用 `ppp01` 帐号登录后, `pppd` 就开始和对端协商 PPP 协议了。

10.7.4 可能出现的问题

但事情真的这么顺利吗? 可能没有, 可能碰到如下情况:

1. 调制解调器可以接电话了, 但接下来没有任何动作, 登录提示符并不会出现。

解决办法:

`/etc/default` 目录下, 有 `getty` 程序初始化的文件, 其内容可能如下所示:

```
CLEAR = NO HANGUP=YES INIT="" ATZ\r OK WAITFOR=RING
CONNECT="" ATA\r CONNECT \s\A
```

若是用命令 `man getty` 命令查看其选项就知道为什么不出提示符, 所以改成:

```
CLEAR = NO
HANGUP=YES
```

登录提示符就可以出现了。

2. 一切设对, PPP 连接也建立起来了, 可以从本地远程登录 Telnet 到 Linux 上了, 但却无法远程登录到其他机器上。

原因:

`pppd` 的 ARP 表并没有成功建立, 数据包无法路由出去。理论上 `pppd` 应可自动建立 ARP 表, 但不知哪里设错, 无法建立。

解决办法:

手动建立 ARP 表:

```
#
#Proxy ARP for those dialin users who will be using this
#machine as a server
#
/sbin/arp -s 140.xxx.xxx.xxx 00: 00: C0: AD: 37: 1C pub
#End proxy arps
```

以 `arp -a` 命令将可发现如下的输出:

```
(140.xxx.xxx.xxx) at 00: 00: C0: AD: 37: 1C
```

这样才可以正确路由出去。



注意: 00: 00: C0: AD: 37: 1C 是以太网地址, 也就是网卡的硬件地址, 在 Linux 启动时注意看可发现它。

3. 可以正常显示, 但用 `ppp01` 帐号登录后就断了。

原因:

pppd 根本没有执行, 可在主控制台以 ppp01 帐号登录, 看看该帐号是否的确可以执行 pppd。

10.8 传真打印机服务器配置

在 Linux 系统下, 可以设置传真打印机服务器。

一个传真打印机服务器是由很少的程序所设置的, 其中, 传真服务器是通过 efax 程序来实现的。在计算机上实现传真如同实现打印一样简单。

10.8.1 设置传真打印机服务器

安装 efax 当作传真打印服务器包含很少的问题。

在这里描述这些问题和解决方法, 以及简单的步骤。

10.8.1.1 efax软件

查明确实已经有了 efax 软件。

在 RPM 基础的系统下使用下面的命令:

```
rpm -qv efax
```

就可以安装该软件。

如果系统上没有安装该软件, 可以取得 efax, 从下面的站点用户可以获取该软件:

```
ftp: //sunsite.unc.edu/pub/Linux/apps/serialcomm/fax/efax08a.tar.gz
```

用户可以从下面的站点获取 RPM 格式的 efax 软件:

```
ftp://ftp.redhat.com/pub/redhat/redhat-4.2/i386/RedHat/RPMS/efax-  
0.8a-3.i386.rpm
```

10.8.1.2 printcap项

在 printcap 文件中没有 efax 项。

解决方法:

在/etc/printcap 文件中加入下面的内容:

```
fax: \  
: lp=/dev/rull: \  
: sd=/var/spool/fax: \  
: if=/usr/bin/faxlpr:
```

10.8.1.3 传真命令

使用硬链接而不是符号链接到传真命令。

执行下面的命令:

```
ln /usr/bin/fax /usr/bin/faxlpr
```

10.8.1.4 /usr/bin/fax 文件

编辑/usr/bin/fax 成需要的文件。

对于调制解调器选择正确的传真种类:

```
CLASS=2.0
```

设置 FROM="your international phone number"项:

```
NAME="Your Name"]
```

加上下列一行对国际的电话号码做本地的转换。

例如, 在以色列需要使用:

```
TELCVT='sed -e s/+972/0/ -e s/+/00/' # israel (以色列)
```

文件中有个问题在 586 行, 改成如下两行:

```
cfile="/usr/bin/tail -1 lock"
cfile="cat $cfile"
```

在 586 和 587 行中有个“-”符号:

```
0) echo "$1" | mail -s "fax to $num succeeded" $user@$host ;
*) echo "$1" | mail -s "fax to $num failed " $user@$host ;
```

10.8.1.5 /var/spool/fax目录

建立目录/var/spool/fax:

```
mkdir /var/spool/fax
chmod 777 /var/spool/fax
```

10.8.1.6 加锁文件

加锁文件/var/spool/fax/lock 建立的权限是不正确的, 使用下列命令把它设置正确:

```
touch /var/spool/fax/lock ; chmod 644 /var/spool/fax/lock
```

10.8.1.7 /dev/modem特殊文件

efax 程序在运行时假定有一个真的设备文件在/dev/modem, 否则它将无法与符号链接一起运行, 所以要以与/dev/cua 设备相同的主要号码和附属号码建立一个设备文件, 通过它连接到调制解调器。文件必须有 rw-rw-rw-模式使任何一个用户都能够使用传真软件。

```
ls -lL /dev/modem
rm /dev/modem
mknod /dev/modem c Mj Mi
chmod 666 /dev/modem
```

此处 Mi 应为 5, Mi 随调制解调器连接的串行端口而定, 若连到 cua0 则 Mi 为 64, 连

到 `cual` 则 `Mi` 为 65。

例如(对于 `cual`):

```
mknod /dev/modem c 5 65
```

10.8.1.8 /var/lock 目录

改变 `/var/lock` 目录的属性:

```
chmod 1777 /var/lock
```

10.8.2 使用传真打印机服务器

使用下列 `lpr` 命令之一:

```
lpr -Pfax -J <Fax-Number> [file-names] any command |
```

或者

```
lpr -Pfax -J <Fax-Number>
```

10.8.3 指定目标的传真号码

在 `lpr` 的 `-J` 选项之后指定目标的传真号码。即:

```
lpr -J fax-number
```

10.8.4 从网络上的 UNIX 机器使用该功能

加入下列各行到文件 `/etc/printcap` 和客户端的 Linux 系统, 建立 `spool` 目录等等。

```
fax: \
    : sd=/var/spool/fax: \
    : mx#0: \
    : sh: \
    : rm=host.domain: \
    : rp=fax:
```

在传真服务器机器中加上客户端机器的名称 `/etc/hosts.lpd`。

10.9 WAIS 服务器

WAIS(广域信息服务)用关键字查找一个文件数据库, 并按照它们的重要程度排列和显示结果。这是使得信息能在网络上使用的一种非常有效的方法。WAIS 是由 Thinking

Machines 开发的, 现在由 WAIS Inc 管理。WAIS 的免费版本称为 freeWAIS, 它能从 Clearinghouse for Networking Information Discovery and Retrieval(CNIDR)获得。还能从 CNIDR 和 LinuxFTP 站点上取得 freeWAIS 的 Linux 版本。

freeWAIS 软件包中包括客户程序、一个服务器程序和一个索引程序。

客户程序被称为 swais、xwais 和 waissearch。它们用来输入请求和显示结果。

索引软件被称为 waisindex。可以用它为 WAIS 文件创建一个关键字索引, 以提供快速有效的访问。

服务器软件被称为 waisserver。用它能够创建自己的 WAIS 站点, 并允许其他的用户在 WAIS 文件上执行查找。

WAIS 服务器:

z3950 的选项及说明如下:

-p *portnum* 监听端口, 如果提供 *portnum*, 则用端口号为 *portnum* 的端口。

-s 监听标准的输入/输出。

-d *directory* 用此目录做为索引的默认位置。

-e *path-name* 把错误输出重定向到 *path-name*, 如果为指定此目录, 则重定向到 /dev/null。

-l *log_level* 设置记录的级别 0、1、5 和 10:

0 不记录

1 只记录错误和警告信息

5 记录 MEDIUM 优先级的信息

10 记录所有的信息

-u *user* 已指定的用户运行服务器。

-v 显示当前的版本和数据。

waisindex 选项和文件类型及说明如下。

waisindex 文件:

index-name.doc 关于文件的信息, 包括大小和名字。

index-name.dct 字典文件, 其中包含交叉索引和倒置文件的所有的关键词列表。

index-name.fn 列出为索引创建的文件。

index-name.hl 所有标题的表; 标题是题目并当它们在查找结果中时显示它们。

index-name.inv 倒排文件, 其中包括一个单词表、它们重要程度的排序以及它们到索引文件的连接。

index-name.src 包含索引信息的源描述文件, 包括它位于什么系统, 它涉及的主题, 谁来管理等信息。

index-name.status 包含用户定义的信息。

waisindex 选项:

-a 给存在的一行附加索引。

-contents 给文件的内容做索引(默认)。

-d *path-name* 为索引文件指定路径名; 路径名将被加到索引文件名的前面。

-e *logfile* 重定向错误信息到 *logfile*。

- export 为了允许 Internet 访问，加主机名和 TCP 端口到源文件中；否则将不包括连接信息，文件只能被本地访问。
 - l num 设置记录的级别 0、1、5 和 10：
 - 0 不记录
 - 1 只记录错误和警告信息
 - 5 记录 MEDIUM 优先级的信息
 - 10 记录所有的信息
 - mem 索引时使用的内存数。
 - M 连接不同类型的文件。
 - contents 决定是否对文件内容做索引；-contents 对整个文件做索引，
 - nocontents -nocontents 只对文件名和标题做索引，不对内容做索引。
 - pairs 如何对待大写字母；-pairs(默认)把大写的单词作为一项。
 - nopairs -nopairs 把大写单词中字母分别作为一项。
 - pos 和-nopos 是否在索引中包括单词的位置信息；-pos 将包括此信息，并允许近似匹配搜索，但索引的大小将增加；-nopos 不包括此信息。
 - r 递归地对子目录做索引。
 - register 登记一个索引到 WAIS 服务器的目录中。
 - t 指定文件的类型。
 - T 设置文件的类型。
- 文件的类型如下：
- filename 用文件名作为标题行的文本类型。
 - first_line 用文件中的第一行作为标题行的文本类型。
 - one_line 对每个句子做索引的文本类型。
 - text 对文件做索引的文本类型，标题行是路径名。
 - ftp 包含访问系统的 FTP 代码。
 - GIF GIF 图像文件。
 - PICT PICT 图像文件。
 - TIFF TIFF 图像文件。
 - MPEG MPEG 文件。
 - MIDI MIDI 文件。
 - HTML 用于 Web 页的 HTML 文件。
 - mail_or_rmail 索引邮箱文件。
 - mail_digest 用标题行中的主题对 E-mail 做索引。
 - netnets 对 USENET 新闻做索引。
 - ps PostScript 文件。

可以从站点 ftp.cnidr.org 或大多数的 Linux FTP 站点上取得 freeWAIS。在 FTP 站点 ftp.cnidr.org 上有一个 freeWAIS 的 Linux 版本，并且是已经编译过的，马上可以使用。当前的用于 Linux 的软件包是 `freeWAIS-0.5-Linux.tar.gz`，位于目录 `/pub/CNIDR.tools/freewais` 中。也可以下载源代码自己编译。源代码软件包叫做 `freeWAIS-0.5.tar.Z`，它能按照不同

的系统进行不同的配置。freeWAIS 也可以从 sunsite.unc.edu 或它的镜像站点上的 `/pub/packages/info-systems/wais` 目录中下载。此目录中有一些不同的版本。

创建一个要存放 freeWAIS 的目录，通常是 `/home/wais`。建议用户下载预编译过的用于 Linux 的二进制的软件包 `freeWAIS-0.5-Linux.tar.gz`。接着用 `gunzip` 解压，用 `tar xvf` 展开文件。将创建一个名为 `freeWAIS-0.5-Linux` 的目录。此目录中包含二进制文件。需要把它们安装到合适的目录中，如 `/usr/bin`。如果下载源代码想自己编译 freeWAIS，首先要配置软件，然后再编译创建它的二进制文件。

10.9.1 配置和安装 freeWAIS 源代码

如果下载源代码，它的目录中将有一些子目录：`doc` 中含有说明文件，`src` 中含有源代码。还有一个特殊的子目录 `wais-test` 中含有服务器的测试文件。

首先要在 Makefile 文件中设定变量 TOP。TOP 需要赋值为 freeWAIS 源代码所在的目录的路径名。输入 WAIS 源代码所在目录的路径名。

除了 Makefile 之外，还有一些特定的 Makefile，每个文件都有一个扩展名指定它是特定操作系统的 Makefile。Makefile.Linux 文件便是用来创建 Linux 版本的 freeWAIS。此文件已经为 Linux 做了一些配置。然而，可以指定或改动其他一些选项。下面将对这些选项进行详细的说明。这些选项被赋值给变量 CFLAGS。开始已经设置了默认的选项。下面是一个 CFLAGS 变量的例子。记住用户只能修改此行：

```
CFLAG = -Wall -m486 -fwritable-strings -Who-unuesd -I$(INCLUDE)
-DTELL_USER -DUSG -DSECURE_SERVER -DRELEVANCE_FEEDBACK -DBOOLEANS
-DPARTIALWORD -DLITERAL -DSOUND -DBIBDS -DLINUX
```

下面是其他一些选项及其说明：

- DBOOLEANS 允许用 AND, OR 和 NOT 进行布尔查找。
- DBINGINDER 为文件的最大集合做索引。
- DLITERAL 文字串查找。
- DPARTIALWORD 允许在模式匹配中使用 “*” 以匹配模式的任何变化。例如，用 `hum*` 匹配 `human`、`hummingbird`。
- DRELEVANCE_FEEDBACK 允许用户从查找中选择相关的文件并在这些相关文件内进行新的查找。
- DSECURE_SERVER 提供最好的服务器安全。
- DTELL_USER 告诉服务器正在连接的用户名称。
- DUSE_SYSLOG 使用系统日志进入系统。

如果需要，可以限制 WAIS 服务只允许特定的网络或系统访问。为了做到这点，创建一个 `SERV_SEC` 文件并输入允许访问的网络和系统的域名和 IP 地址。文件 `SERV_SEC` 被定义在文件 `ir.h` 中，如下所示：

```
#define SERVSECURITYFILE "SERV_SEC"
```

`SERV_SEC` 文件中的项包含域名和 IP 地址，其中 IP 地址是可选项，例如：

```
pangol.train.com 204.166.189.21
```

可以给进一步细化限制到一个特定的数据库，只允许某些指定的网络和系统访问某些数据库。需要建立一个 DATA_SEC 文件。此文件也被定义在文件 ir.h 中，如下所示：

```
#define DATASECURITYFILE "DATA_SEC"
```

文件 DATA_SEC 中的每一项首先列出数据库，接着是域名和可选的 IP 地址。在域名的区域用一个 “*” 则表示所有的用户均可访问此数据库。下面的例子显示文件 DATA_SEC 中的项，第二项为所有的用户打开一个 oldata 数据库：

```
mydata pangol.train.com 204.166.189.21
oldata      *           *
```

一旦准备编译 freeWAIS，要使用下面的 make 命令和项 Linux。它将为 Linux 系统创建 WAIS 客户程序、索引器和服务器程序。接着首先要运行目录 src/x 中的 xmkmf 来为 xwais 创建 Makefile。

```
#make Linux
```

10.9.2 创建索引

为了使用 WAIS，必须为需要使之可用的文件创建索引。索引的过程是由 waisindex 命令完成的，它将创建一个特定的 WAIS 数据库。可以为单个文件、一组文件或文件的整个目录和子目录创建索引。数据文件和它的索引一起组成一个 WAIS 数据库。可以分别索引不同的文件或文件组，在服务器上建立不同的 WAIS 数据库。WAIS 数据库应该位于当 WAIS 服务器被调用时指定的 WAIS 数据目录中。

waisindex 能创建一个倒排文件索引，它将涉及指定的文件中的每个单词。这样就可以在文件中进行全文的关键字搜索。waisindex 命令有一些选项，在选项的最后是被索引的文件、文件组或目录。

用 -d 选项能指定索引的名字。waisindex 将为文件创建几个索引文件。每个索引文件有自己的扩展名来说明自己的功能，但它们名字的前缀是用 -d 选项指定的名字。如果没在此选项中指定名字，“index” 将作为前缀名。

另外，如果希望数据库能被 Internet 上的用户访问，还要加一个 -export 选项。没有此选项，数据库只能被系统中的其他用户访问。下面将讨论 -export 选项。

```
waisindex -d index-file -export file-list
```

如果列出多个文件被索引，则对所有的文件将做成一个索引。如果想对一个子目录中的所有文件做索引，只需在命令中用 -r 选项加上目录名。

```
waisindex -d index-file -export -r directory-name
```

若要加一个文件或目录到一个已存在的数据库中，需要在对它做索引时用选项 -a。还要用选项 -d 和数据库名以把此索引文件加到指定的数据库中。可以在命令行中列出几个文件以把它们加入数据库中。如果想加一个目录，要使用 -r 选项加上目录名。

```
waisindex -d index-file -a -export file-list
```

下例中, 用户首先对文件 `cookies` 和 `cakes` 做索引, 为它们创建一个索引名为 `recipes`。从 `recipes` 中查询将搜索 `cookies` 和 `cakes` 两个文件。接着对文件 `pies` 做索引并把它加到索引 `recipes` 中。WAIS `recipes` 数据库中现在包含文件 `cookies`、`cakes` 和 `pies`。然后对目录 `snack`, 包括它的所有文件和它的子目录中的所有文件做索引, 索引名为 `junkfood`。最后一个例子中, 又一次对 `sancks` 做索引, 但这一次索引将被加到 `recipes` 数据库中。`junkfood` 数据库仍然存在并引用 `snacks` 目录。

```
#waisindex -d recipes -export cookies cakes
#waisindex -d recipes -export -a pies
#waisindex -d junkfood -export -r snacks
#waisindex -d recipes -export -a -r snacks
```

用 `-t` 选项, 将能对不同类型的文件做索引。能索引图片、邮箱文件、HTML 页和标准的文本文件。对于文本文件, 可以进一步指定对每行做索引。如果按行索引, 当用关键字查找时, WAIS 将指出关键字所在的行。下面的例子中, 使用者对文件 `bread`s 的每行做索引, 产生索引文件叫做 `cereals`。

```
#waisindex -d cereals -t one_line breads
```

`waisindex` 命令还能把不同类型的文件关联到一个文件中。例如, 如果有一些图像、视频或声音文件, 想把它们和一个指定的文本文件相关联, 可以用 `waisindex` 把这些文件连接在一起。当一个用户检索一个文本文件时, 同时将检索关联的图像、视频或声音文件。当用户阅读文本时, 它也能显示一个图片或播放声音。关联文件必须和它们所关联的文件有相同的前缀。例如, 如果有一个文件名为 `train.txt`, 有一个图形文件 `trani.gif` 和声音文件 `train.midi`。用 `-M` 选项和一系列文件类型, 加上 `-export` 选项把一组文件连接到一个索引文件。

```
#waisindex -d train -Mtext, tiff, mpeg, midi -export /user/waisdata/train/*
```

为了让 WAIS 和 Web 资源很好的结合, 需要为 Web 页创建 WAIS 索引。用 `waisindex` 和 `-T HTML` 选项, 用它来指定被索引的文件类型是 HTML 文件。索引名可以是诸如 `myweb` 之类的名字。此索引将允许 WAIS 搜索 Web HTML 文件。在下面的例子中, 使用者将对目录 `/home/httpd/html` 中的 Web 页做索引。索引名为 `myweb`, 类型是 HTML。通过指定 `-contents` 选项对每个 Web 页的全部内容做索引。选项 `-export` 将包括 Internet 访问的主机信息。

```
#waisindex -d myweb -T HTML -contents -export -r
/home/httpd/html/*.html
```

10.9.3 WAIS 资源

当为文件做索引以使得数据库可用时, `waisindex` 将为数据库创建一个源文件, 通过它其他的用户能访问数据库; 它将提供一些信息, 如数据库的名称等。一些 WAIS 数据库访问时要收费, 用源文件能显示收费信息。还能从中找到管理者的地址以发送一些建议。在源文件的结束是对 WAIS 数据库的一个简短说明。

当用命令 `waisindex` 创建数据库时, 如果指定 `-export` 选项, 为了使其他系统中的用户访问数据库, 将加入一些字段。其中有两个字段是关于 Internet 信息的, 一个是 IP 名, 另一个是主机的 IP 地址。被加入的另一个字段是用来指定访问计算机上的 WAIS 数据库的端口。如果没有用 `-export` 选项, 这些字段将是空的。这种情况下, 只有系统中的用户才能访问数据库。

如果需要, 可以改变源文件中的任何字段。完整的源文件是用括号括起来的。每行中的字段是以一个开始的冒号加上字段名。可以编辑源文件并加入更多的说明。注意描述是用双引号括起来的。第一个引号是在 “description” 的后面, 然后是说明的文本, 最后是结束的引号, 它单独占一行。下面的例子是一个 zipcodes 的源文件, 名为 `zipcodes.src`。

```
(:source
:versiton 3
:ip-address "192.31.181.1"
:ip-name "quake.think.com"
:tcp-port 210
:database-name "/proj/wais/db/sources/zipcodes"
:cost 0.00
:cost-unit :free
:maintainer "wais@quake.think.com"
:description "
WAIS index of USA Zip Code database.
The full Zipcodes file may be obtained via FTP using the URL:
<ftp://obi.std.com/obi/ZIPCODES/zipcode.txt>
"
)
```

其他用户用源文件来访问 WAIS 数据库。一个源文件将告诉用户它位于的是哪个主机以及它调用的是什么。可以把它看作是 WAIS 数据库的 URL。

远程的用户为了访问数据库必须首先拥有源文件。可以把源文件发送给用户, 它可以把源文件插入它的主机的 `wais-sources` 目录中, 或者把源文件登记注册到一个如 `quake.think.com` 和 `cnidr.org` 之类的 WAIS 服务器中, 这些服务器中有一个服务器的目录。源文件和其他的源文件都放在这些服务器中。用一个 WAIS 客户程序如 `swais`, 使用者能访问服务器的此目录并找到列在其中的 WAIS 数据库。接着它们可以选择和查询数据库。

当创建一个数据库时, 在对文件做索引时用 `-register` 选项就可以注册数据库。也可以等到以后再注册, 可能需要先测试它。注册时只须用命令 `waisindex` 和选项 `-d` 加索引名, 然后再加上 `-register` 选项。

```
waisindex -d recipes -register
```

10.9.4 测试 WAIS 服务器

在 `freeWAIS` 源代码包中, 有一个目录名为 `test-wais`, 此目录中含有一组测试文件,

用它们可以测试索引或服务器的访问。在此目录中有一个 Shell 脚本为 `test.waisindex`。如果查看此 Shell 脚本, 将看到几个 `waisindex` 命令, 用测试文件创建几个不同的数据库。它将创建四个测试索引。索引 `test-Bool` 测试布尔查找能力。`test-Comp` 索引测试压缩文件的操作。`test-Docs` 索引对使用目录 `docs` 中的文件进行递归查找进行测试。`test-Multi` 检查对不同类型的文件如 GIF 图形文件的操作。(这些命令名前都有一个路径 `../bin/`, 如果已经下载 WAIS 二进制文件并已进行安装, 则应该删除前面的 `../bin/`。)

还能注意到为每个测试数据库都创建两个源文件: `boolean.src` 和 `doc.src`。为了在本地测试服务器, 把这些源文件复制到 WAIS 客户程序用来访问 WAIS 资源的目录。例如, 如果 `swais` 用 `/usr/lib/wais-sources` 作为它的源目录, 然后复制 `test-boolean.src`(源)文件到此目录中。现在或者通过 `inetd` 或直接启动 WAIS 服务器(如果直接启动, 在命令的最后加上一个“&”号)。用 `test-wais` 目录的路径名作为 WAIS 数据目录。当启动 `swais`, 它将列出在它的源目录所有资源, 包括 `test-boolean` 资源和其他测试资源。可以选择和查询测试资源, 它将访问 WAIS 服务器并返回结果。可以试着查找关键字“`boolean`”。也能把目录 `wais-test` 移到 WAIS 数据目录中。为了创建一个总体的数据库, 可以对 `wais-test` 用 `-r` 选项做索引。

10.9.5 启动 freeWAIS

用户可以让 WAIS 不间断地运行, 或当需要是通过 `inetd` 调用它。用 `waissserver` 命令和一些选项能启动 WAIS 服务器。可以用 `-d` 选项指定 WAIS 索引的默认位置。还能用 `-p` 设置端口, 用 `-u` 设置用户名。在命令的最后加上一个“&”号。当直接调用时, `waissserver` 将在后台运行。为了使 WAIS 以一个持续运行的守护进程运行, 用户应在一个 `rc.d/init` 启动文件如 `wais.init` 中输入 `waissserver` 命令。下面的例子是一个 `waissserver` 命令:

```
waissserver -d /usr/wais/wais_index &
```

为了更安全地运行 WAIS 服务, 以一个普通用户而不是以 `root` 来运行 `waissserver`。新建一个用户并在它的 `passwd` 项中输入一个“*”。然后启动 `waissserver` 时能用 `-u` 选项加上用户名。在下面的例子中, `waissserver` 以用户 `sports` 运行:

```
waissserver -u sports -d /usr/wais/wais_index &
```

为了使 `inetd` 启动 WAIS 服务器, 必须在文件 `/etc/services` 和 `/etc/inetd.conf` 中设置合适的项。对于文件 `/etc/services`, 可以设置如下的项:

```
z3590 210/tcp #z39_50 protocol for WAIS
```

接着在文件 `inetd.conf` 中, 放置调用 WAIS 服务器的项。当 WAIS 服务器通过 `waissserver.d` 文件被调用, 它将知道是通过 `inetd` 守护进程来运行的。由于这个原因, 参数列中的第一个参数是 `waissserver.d`:

```
z3590 stream tcp nowait root /usr/sbin/waissserver  
waissserver.d /home/wais -e server.log
```

10.10 Linux 上的 Java

现在已有大量的基于 Java 的产品可用于 Linux 中。有 Linux 版本的 Java 开发软件包 JDK, HotJava 浏览器和 Java 的 Web 服务器。可以从 Blackdown 的 Web 站点上下载这些产品, Blackdown 的 Web 页为:

<http://www.blackdown.org/java-Linux.html>

用户可以从一列可用的 Java 产品中选择需要的项。下面列出一些产品:

JDK1.0.2 Java 开发软件包 1.0.2。

JDK1.1.3 Java 开发软件包 1.1.3。

JDBC JDBC 数据库访问 API。

BDK JavaBeans 开发软件包(要求 Java 开发软件包 1.1.1)。

Java Web Server 用 Java 编写的 Web 服务器。

JSDK Java Servlet 开发软件包。

JWS20 工作组 2.0 Build 125。

HotJava HotJava 1.0 Web 浏览器(要求 Java 开发软件包 1.1.1)。

10.10.1 Java 开发软件包 JDK

Java 开发软件包是一种工具,用它来可以创建和调试 Java 程序,并对 Java 应用提供支持,如 HotJava 浏览器。软件包中包含演示的 Java 程序和源代码。从 Sun 的 Web 站点可以取得关于 JDK 的详细信息的文件:

<http://java.sun.com/products/jdk/>

现在有两个版本的 JDK 可用, 1.0 和 1.1。

1.0 是一个老的版本,它被大多数的浏览器支持。

1.1 是一个较新的版本,有许多老版本的浏览器不支持它。

JDK1.1 包括一些先进的功能,比如 JavaBean 和数据库链接。两个 JDK 版本都可用于 Linux。可以从一个 Java-Linux 的镜像站点下载它们,从前面提到的 Blackdown Web 页能链接到此镜像站点。在 Linux CDROM 上也有 JDK1.0。许多的 Java 的应用如 HotJava 浏览器需要 JDK1.1。为了使用它们,需要下载和安装 JDK1.1。JDK1.1 也有两个版本 JDK1.1.1 和最新的 JDK1.1.3。

10.10.2 JDK1.0.2

Java 开发软件包的 1.0.2 版本在 Linux CDROM 上的 contrib 目录中,作为一个 RPM

软件包。它也是 JDK1.0 的最后一个版本。安装它之前，首先要安装上 CDROM 并切换到它的 contrib 目录下，进入 RPMS 子目录，用户将发现 JDK 的静态的和共享的两个版本。共享的版本是为那些有 Motif 共享库的用户使用的。对于 Linux Lite，安装静态的版本。用命令 rpm-i 安装 JDK 软件包：

```
#mount /mnt/cdrom
#cd /mnt/cdrom/Linux/contrib/RPMS
#ls JDK*
JDK_shared-1.0.2.p12-3.i386.rpm  JDK_static-1.0.2.p12-3.i386.rpm
#rpm -i JDK_static--1.0.2.p12-3.i386.rpm
```

JDK 软件包在目录/usr/local/java 中安装 Java 应用软件、库函数和演示程序。

Java 应用软件包括一个 Java 编译器 javac、一个 Java 调试程序 jdb 和一个 applet 查看程序 appletviewer。为了使用这些应用程序，需要把目录/usr/local/java/bin 加到变量 PATH 中，变量 PATH 在文件/etc/profile 或.profile 中。appletviewer 位于目录/usr/local/java 中，需要把此目录也加到 PATH 中。

10.10.3 JDK1.1.1

JDK1.1 提供了比 JDK1.0 更多的功能。

JDK1.1 支持国际化、带符号的 applet、JAR 文件格式、增强的 AWT(window 工具集)、JavaBeans 组件模式、增强的网络功能、针对大数的数学软件包、远程方法调用 RMI、反射、数据库的链接 JDBC、新的 Java 界面、串行的对象、内部的类和增强的执行功能。关于这些特征的具体说明可以在 JDK 的文件中找到。

当前有两个版本的 JDK1.1：JDK1.1.1 和最新的 JDK1.1.3。下面将介绍如何安装 JDK1.1.1。JDK1.1.3 的安装过程也很相似。

当前 JDK1.1.1 的 Linux 版本是一个压缩文件。需要下载、解压和展开.tar.gz 文件。一旦下载此文件，把它放在希望安装 JDK1.1.1 的目录中，通常是/usr/local。然后用 gunzip 解压它，并用 tar xvf 展开，用户也可以把两个操作合在一起用如下所示的 tar xvzf 解压并展开。

```
#mv jdk.1.1.1-Linux-v3.tar.gz /usr/local
#cd /usr/local
#tar xvzf jdk.1.1.1-Linux-v3.tar.gz
```

展开后将创建一个名为 jdk1.1.1 的目录，其中有包含 Java 应用软件和函数库的两个子目录 bin 和 lib。需要把目录/usr/local/jdk1.1.1/bin 加到变量 PATH 中，变量 PATH 在文件/etc/profile 或.profile 脚本中(TCSh Shell 是.login)。另外在/etc/profile 或.profile 中，需要加一项，把目录/usr/local/jdk1.1.1/bin 指定给变量 JAVA_HOME，如下所示。此变量被许多 Java 应用程序如 HotJava 用来定位 Java 解释器。

以下是 JDK 工具和目录及简要说明：

JDK1.0 和 1.1 工具：

Java Java 解释器。
Javac Java 编译器。
Jdb Java 调试器。
Javadoc 以 HTML 格式产生 API 文件。
Javah 为一个 Java 类创建 C 头文件和 C 存根文件。
Javap 对编译过的 Java 文件进行反编译。
Appletviewer 允许用户在没有 Web 浏览器时运行 applet。
JDK1.1 附加的工具：
Jar Java 存放文件工具。
JDK1.0 和 1.1 工具：
Javakey 数字符号工具。
native2ascii Native-To-ASCII 转换器。
rmic Java RMI 转换器。
rmiregistry Java 远程对象寄存器。
Serialver 串行版本的命令。
UpdateAWT AWT1.1 转换工具。
Java 目录：
Bin Java 应用软件。
Lib Java 类库。
Include Java 包含文件。
Demo Java applet 的例子。

10.10.4 Java applet

可以像使用标准编程语言编写程序一样来创建 Java applet。

首先用文本编辑器来编写源代码。源代码被保存在扩展名为.java 的文件中。接着可以用编译器 javac 来编译源文件，产生一个 Java applet。这个 applet 文件将有一个扩展名.class。例如在目录/usr/local/java/demo/blink 中有一个名为 Blink.java 的 java 源文件。可以切换到此目录中并编译文件 Blink.java，产生文件 Blink.class。

```
#javac Blink.java
```

在目录中的文件 example1.html 将调用 Blink.class applet。启动浏览器并调用此文件即可运行 Blink applet。

在 Web 页中使用 HTML 标识符<applet>调用 applet。此标识符中包含一些属性，其中一个为：code。可以给 code 分配编译过的 applet 的名字。这里还有一些可选的属性，可以通过它们设置一些特性，像 applet 的显示区域和它的排列。甚至可以访问远程 Web 站点上的 applet。一个关于不同 applet 特性的完整列表在文件/usr/local/java/README.jdk 中。下面的例子中，一个名为 Blink.class 的 applet 将显示在 Web 浏览器的中间的方框中，此方框高 140 像素，宽 100 像素。

```
< applet code="Blink.class" width=100 height=140  
align=center></applet>
```

为了调用调试器，可以使用命令 `appletviewer` 加 `-debug` 选项和运行 `applet` 的 HTML 文件名，如下所示：

```
appletviewer -debug mypage.html
```

[General Information]
书名=Linux 安装与配置指南
作者=高鹏
页数=251
SS号=10028737
出版日期=1999年4月第1版