

# 目 录

## 第一部分 Linux 安装与入门

|                                      |        |                        |                                                   |         |         |
|--------------------------------------|--------|------------------------|---------------------------------------------------|---------|---------|
| <b>1 Linux 历史及简介</b> .....           | ( 3 )  | 2.2.10 Linux 支持的声卡     | .....                                             | ( 50 )  |         |
| 1.1 什么是 Linux                        | .....  | ( 3 )                  | 2.2.11 Linux 支持的鼠标                                | .....   | ( 51 )  |
| 1.2 Linux 的发展历史                      | .....  | ( 4 )                  | 2.2.12 Linux 支持的游戏操作杆                             | .....   | ( 51 )  |
| 1.3 Linux 与 UNIX 的关系, Linux 的版本介绍    | .....  | ( 8 )                  | 2.2.13 Linux 支持的调制解调器                             | .....   | ( 52 )  |
| 1.3.1 UNIX 的起源介绍                     | .....  | ( 8 )                  | 2.2.14 Linux 支持的打印机/绘图机                           | .....   | ( 52 )  |
| 1.3.2 Linux 的特性以及为什么使用 Linux         | .....  | ( 10 )                 | 2.2.15 Linux 支持的扫描仪                               | .....   | ( 53 )  |
| 1.3.3 Linux 的版本介绍                    | .....  | ( 16 )                 | 2.2.16 Linux 支持(DPMS)显示器                          | .....   | ( 54 )  |
| 1.4 如何获得 Linux 以及 Linux 的网络资源        | .....  | ( 23 )                 | 2.2.17 Linux 支持的视频捕捉卡                             | .....   | ( 54 )  |
| 1.4.1 如何获得 Linux                     | .....  | ( 23 )                 | 2.2.18 Linux 支持的不间断电源 UPS                         | .....   | ( 54 )  |
| 1.4.2 Linux 的网络资源                    | .....  | ( 24 )                 | 2.2.19 Linux 支持的数据采集卡                             | .....   | ( 54 )  |
| 1.5 Linux 在中国的前景及 CLinux 计划          | .....  | ( 31 )                 | 2.3 Linux 安装前的准备工作                                | .....   | ( 55 )  |
| 1.6 常见问题解答                           | .....  | ( 32 )                 | 2.4 Slackware Linux 的安装指南                         | .....   | ( 59 )  |
| <b>2 Linux 的安装</b> .....             | ( 34 ) | 2.5 RedHat Linux 的安装指南 | .....                                             | ( 90 )  |         |
| 2.1 如何取得 Linux                       | .....  | ( 34 )                 | 2.5.1 制作 Redhat 的启动盘                              | .....   | ( 90 )  |
| 2.2 Linux 安装的硬件要求                    | .....  | ( 34 )                 | 2.5.2 使用启动盘来启动机器                                  | .....   | ( 90 )  |
| 2.2.1 Linux 支持的计算机/主板/CPU/内存         | .....  | ( 35 )                 | 2.5.3 开始系统安装                                      | .....   | ( 91 )  |
| 2.2.2 Linux 对 PCMCIA 的支持             | .....  | ( 39 )                 | 2.6 TurboLinux 中文版安装指南                            | .....   | ( 103 ) |
| 2.2.3 Linux 支持的显示卡和显示器               | .....  | ( 39 )                 | 2.7 常见问题解答                                        | .....   | ( 121 ) |
| 2.2.4 Linux 支持的硬盘控制器/硬盘              | .....  | ( 42 )                 | <b>3 Linux 的基本配置简介</b> .....                      | ( 122 ) |         |
| 2.2.5 Linux 支持的磁带机                   | .....  | ( 44 )                 | 3.1 网络的配置(TCP/IP, PPP, SLIP)                      | .....   | ( 122 ) |
| 2.2.6 Linux 支持的光盘驱动器(CDROM)(包括光盘刻录机) | .....  | ( 45 )                 | 3.1.1 记录必需的网络参数                                   | .....   | ( 122 ) |
| 2.2.7 Linux 支持的便携式驱动器                | .....  | ( 45 )                 | 3.1.2 网络配置文件:/etc/rc.d/rc.inet1, rc.inet2 和其他相关文件 | .....   | ( 124 ) |
| 2.2.8 Linux 支持的 I/O 接口               | .....  | ( 46 )                 | 3.1.3 设定 SLIP 设备                                  | .....   | ( 131 ) |
| 2.2.9 Linux 支持的网卡                    | .....  | ( 46 )                 | 3.1.4 设定 PLIP 设备(选项)                              | .....   | ( 137 ) |

|                                           |              |                      |       |
|-------------------------------------------|--------------|----------------------|-------|
| 3.1.5 设定 PPP 设备 .....                     | (139)        | 4.1.23 file .....    | (234) |
| 3.2 X Window 的配置 .....                    | (146)        | 4.1.34 find .....    | (234) |
| 3.2.1 x86config 和 XF86Setup 的使<br>用 ..... | (146)        | 4.1.35 fortune ..... | (235) |
| 3.2.2 设定窗口管理器 .....                       | (162)        | 4.1.36 grep .....    | (235) |
| 3.3 /etc 下的配置文件剖析 .....                   | (184)        | 4.1.37 gzexe .....   | (236) |
| 3.4 其他设备的配置(光驱,声卡,打印<br>机) .....          | (199)        | 4.1.38 gzip .....    | (237) |
| 3.5 常见问题解答 .....                          | (206)        | 4.1.39 head .....    | (237) |
| <b>4 Linux 使用入门 .....</b>                 | <b>(210)</b> | 4.1.40 hostid .....  | (238) |
| 4.1 Linux 基本命令的使用 .....                   | (210)        | 4.1.41 id .....      | (238) |
| 4.1.1 ash .....                           | (210)        | 4.1.42 join .....    | (238) |
| 4.1.2 at .....                            | (211)        | 4.1.43 kill .....    | (239) |
| 4.1.3 banner .....                        | (212)        | 4.1.44 last .....    | (240) |
| 4.1.4 bash .....                          | (213)        | 4.1.45 less .....    | (240) |
| 4.1.5 bc .....                            | (214)        | 4.1.46 ln .....      | (241) |
| 4.1.6 cal .....                           | (214)        | 4.1.47 logname ..... | (242) |
| 4.1.7 cat .....                           | (215)        | 4.1.48 ls .....      | (242) |
| 4.1.8 cd .....                            | (216)        | 4.1.49 man .....     | (244) |
| 4.1.9 chgrp .....                         | (216)        | 4.1.50 mtools .....  | (245) |
| 4.1.10 chmod .....                        | (217)        | 4.1.51 msg .....     | (246) |
| 4.1.11 chown .....                        | (218)        | 4.1.52 mkdir .....   | (247) |
| 4.1.12 chsh .....                         | (218)        | 4.1.53 mknod .....   | (247) |
| 4.1.13 clear .....                        | (219)        | 4.1.54 more .....    | (248) |
| 4.1.14 compress .....                     | (219)        | 4.1.55 mv .....      | (248) |
| 4.1.15 cp .....                           | (219)        | 4.1.56 newgrp .....  | (249) |
| 4.1.16 cpio .....                         | (221)        | 4.1.57 nice .....    | (250) |
| 4.1.17 crontab .....                      | (223)        | 4.1.58 passwd .....  | (250) |
| 4.1.18 csh .....                          | (225)        | 4.1.59 paste .....   | (251) |
| 4.1.19 cut .....                          | (225)        | 4.1.60 pr .....      | (252) |
| 4.1.20 date .....                         | (226)        | 4.1.61 ps .....      | (253) |
| 4.1.21 dd .....                           | (227)        | 4.1.62 pwd .....     | (254) |
| 4.1.22 df .....                           | (228)        | 4.1.63 quota .....   | (255) |
| 4.1.23 diff .....                         | (229)        | 4.1.64 rep .....     | (255) |
| 4.1.24 dialog .....                       | (229)        | 4.1.65 reset .....   | (256) |
| 4.1.25 du .....                           | (230)        | 4.1.66 rm .....      | (256) |
| 4.1.26 echo .....                         | (231)        | 4.1.67 rmdir .....   | (257) |
| 4.1.27 ed .....                           | (231)        | 4.1.68 set .....     | (257) |
| 4.1.28 egrep .....                        | (230)        | 4.1.69 sh .....      | (258) |
| 4.1.29 env .....                          | (232)        | 4.1.70 sleep .....   | (258) |
| 4.1.30 expand .....                       | (232)        | 4.1.71 sort .....    | (258) |
| 4.1.31 fdformat .....                     | (233)        | 4.1.72 split .....   | (260) |
| 4.1.32 fgrep .....                        | (234)        | 4.1.73 stty .....    | (260) |
|                                           |              | 4.1.74 sync .....    | (263) |
|                                           |              | 4.1.75 su .....      | (264) |

|                      |                    |       |        |                   |       |
|----------------------|--------------------|-------|--------|-------------------|-------|
| 4.1.76               | sudo               | (265) | 4.1.94 | wc                | (283) |
| 4.1.77               | tail               | (267) | 4.1.95 | whereis           | (284) |
| 4.1.78               | tar                | (267) | 4.1.96 | who               | (285) |
| 4.1.79               | tesh               | (270) | 4.1.97 | which             | (285) |
| 4.1.80               | tee                | (271) | 4.1.98 | write             | (286) |
| 4.1.81               | test               | (272) | 4.1.99 | zoo               | (286) |
| 4.1.82               | time               | (273) | 4.2    | Emacs 的使用         | (287) |
| 4.1.83               | top                | (273) | 4.2.1  | 功能强大的编辑器 Emacs 介绍 | (287) |
| 4.1.84               | touch              | (274) | 4.2.2  | Emacs 安装简介        | (287) |
| 4.1.85               | tr                 | (275) | 4.2.3  | 进入 Emacs 的世界      | (289) |
| 4.1.86               | tty                | (275) | 4.2.4  | 获得帮助              | (293) |
| 4.1.87               | umount, mount      | (276) | 4.2.5  | Emacs 使用技巧        | (297) |
| 4.1.88               | uname              | (277) | 4.3    | RPM 的使用           | (300) |
| 4.1.89               | unrar, unzip, lha  | (278) | 4.3.1  | 什么是 RPM           | (300) |
| 4.1.90               | uniq               | (278) | 4.3.2  | 安装和使用 RPM         | (301) |
| 4.1.91               | uudecode, uuencode | (279) | 4.4    | 常见问题解答            | (306) |
| 4.1.92               | vi                 | (280) |        |                   |       |
| 4.1.93               | wall               | (283) |        |                   |       |
| 第二部分 Linux 系统管理与中文环境 |                    |       |        |                   |       |
| 5                    | Linux 系统管理         | (311) | 5.8    | 交换空间              | (339) |
| 5.1                  | root 账号            | (311) | 5.8.1  | 交换空间              | (339) |
| 5.2                  | 引导系统               | (313) | 5.8.2  | 创建交换空间            | (340) |
| 5.2.1                | 从软盘引导              | (313) | 5.8.3  | 激活交换空间            | (341) |
| 5.2.2                | 用 LILO 引导          | (314) | 5.8.4  | 关闭交换空间            | (341) |
| 5.2.3                | 用 Loadlin 引导       | (318) | 5.9    | 设备文件              | (341) |
| 5.3                  | 系统启动和初始化           | (318) | 5.10   | 系统的备份与恢复          | (343) |
| 5.3.1                | 内核引导信息             | (318) | 5.10.1 | 简单备份              | (344) |
| 5.3.2                | init 和 initab      | (321) | 5.10.2 | 增量备份              | (346) |
| 5.4                  | 单用户模式              | (323) | 5.11   | 用 cron 制订作业日程     | (347) |
| 5.5                  | 关闭系统               | (323) | 5.12   | 管理打印服务            | (348) |
| 5.6                  | 管理用户帐号             | (324) | 5.12.1 | 检查打印机硬件           | (349) |
| 5.6.1                | passwd 文件          | (325) | 5.12.2 | 选择打印软件            | (351) |
| 5.6.2                | shadow 文件          | (326) | 5.12.3 | 配置 printcap 文件    | (352) |
| 5.6.3                | group 文件           | (327) | 5.12.4 | 打印过滤器             | (353) |
| 5.6.4                | 创建帐号               | (328) | 5.13   | 进程记帐              | (355) |
| 5.6.5                | 删除和查封帐号            | (330) | 5.14   | 软件升级              | (359) |
| 5.6.6                | 修改用户帐号             | (330) | 5.14.1 | 用 Linux 发行套件升级    | (360) |
| 5.7                  | 文件系统               | (331) | 5.14.2 | 升级函数库             | (362) |
| 5.7.1                | 文件系统类型             | (331) | 5.14.3 | 升级编译器             | (365) |
| 5.7.2                | 挂卸文件系统             | (333) | 5.14.4 | 升级其他软件            | (366) |
| 5.7.3                | 创建文件系统             | (336) | 5.15   | 编译新内核             | (368) |
| 5.7.4                | 检查和修补文件系统          | (337) |        |                   |       |

|                                     |              |                                       |              |
|-------------------------------------|--------------|---------------------------------------|--------------|
| 5.15.1 获取内核源程序 .....                | (369)        | 7.1.2 在以太网上设置 TCP/IP .....            | (416)        |
| 5.15.2 解开源程序包 .....                 | (370)        | 7.2 万维网(WWW)服务器的安装和配置 .....           | (421)        |
| 5.15.3 编译前的准备工作 .....               | (371)        | 7.2.1 安装 .....                        | (421)        |
| 5.15.4 配置内核 .....                   | (373)        | 7.2.2 配置指令 .....                      | (424)        |
| 5.15.5 编译内核和用新内核引导 .....            | (381)        | 7.3 代理服务器的设置 .....                    | (430)        |
| <b>6 Linux 编程简介 .....</b>           | <b>(383)</b> | 7.3.1 代理服务器 .....                     | (430)        |
| 6.1 GNU C 和 GNU Fortran 的安装和使 ..... | (383)        | 7.3.2 设置 Apache HTTP 服务器为 .....       | (430)        |
| 6.1.1 安装 gcc .....                  | (383)        | 7.3.3 设置浏览器使用代理服务器 .....              | (434)        |
| 6.1.2 使用 gcc 编译和连接 .....            | (385)        | 7.4 FTP 服务器的设置和维护 .....               | (435)        |
| 6.1.3 创建函数库 .....                   | (388)        | 7.4.1 wu-ftpd 和匿名 FTP 服务的配 .....      | (435)        |
| 6.1.4 利用 make 和 Makefile 实现 .....   | (388)        | 7.4.2 FTP 镜像节点的维护 .....               | (441)        |
| 6.2 C Shell 编程 .....                | (393)        | 7.5 BBS 服务器安装说明 .....                 | (449)        |
| 6.2.1 运行脚本文件 .....                  | (393)        | 7.5.1 安装 BBS 服务器 .....                | (449)        |
| 6.2.2 脚本的执行 .....                   | (394)        | 7.5.2 使用 bbs 的一些技巧 .....              | (453)        |
| 6.2.3 C Shell 脚本表达式 .....           | (395)        | 7.6 安装新闻服务器 .....                     | (457)        |
| 6.2.4 C Shell 脚本控制结构 .....          | (396)        | 7.7 Linux 和 Windows 95/NT 的文件和打 ..... | (461)        |
| 6.2.5 为命令提供输入 .....                 | (398)        | 7.7.1 Samba 的安装和配置 .....              | (461)        |
| 6.2.6 捕获中断 .....                    | (399)        | 7.7.2 Samba 的使用要点 .....               | (467)        |
| 6.2.7 一个 C Shell 脚本的例子 .....        | (399)        | 7.7.3 Linux 和 Windows 95/NT 的打 .....  | (471)        |
| 6.3 AWK 简介 .....                    | (400)        | <b>8 Linux 上的中文环境 .....</b>           | <b>(475)</b> |
| 6.3.1 名词定义 .....                    | (401)        | 8.1 互联网上的 Linux 中文资源 .....            | (475)        |
| 6.3.2 如何执行 AWK .....                | (401)        | 8.1.1 中文软件资源 .....                    | (475)        |
| 6.3.3 AWK 程序的主要结构 .....             | (401)        | 8.1.2 中文 Linux 计划 .....               | (476)        |
| 6.3.4 AWK 的域变量 .....                | (402)        | 8.1.3 Linux 的中文站点 .....               | (477)        |
| 6.3.5 AWK 的工作流程 .....               | (402)        | 8.1.4 开设 Linux 的中文讨论区的 .....          | (478)        |
| 6.3.6 输出合乎指定条件的记录 .....             | (404)        | 8.1.5 开设 Linux 的中文新闻组的 .....          | (478)        |
| 6.3.7 AWK 中数组的特色 .....              | (405)        | 8.2 中文字符集、编码和编码转换程序 .....             | (478)        |
| 6.3.8 AWK 程序中使用 Shell 命令 .....      | (407)        | 8.2.1 中文字符集及编码 .....                  | (478)        |
| 6.3.9 如何读取命令行上的参数 .....             | (407)        | 8.2.2 编码转换程序 .....                    | (482)        |
| 6.3.10 AWK 的内部变量( Built-in .....    | (409)        | 8.3 字符环境下的中文模拟终端—— .....              | (483)        |
| 6.3.11 AWK 的内部函数( Built-in .....    | (411)        | chdrv, ccc, kon .....                 | (483)        |
| <b>7 Linux 网络管理与服务 .....</b>        | <b>(414)</b> | 8.3.1 chdrv .....                     | (483)        |
| 7.1 TCP/IP 网络配置 .....               | (414)        | 8.3.2 ccc .....                       | (483)        |
| 7.1.1 TCP/IP 协议集 .....              | (414)        |                                       |              |

|                                                |       |                                                        |              |
|------------------------------------------------|-------|--------------------------------------------------------|--------------|
| 8.3.3 kon .....                                | (484) | 8.14.1 安装中文 PostScript 字库 .....                        | (532)        |
| 8.4 X Window 系统下的中文终端——<br>CXterm, crxvt ..... | (485) | 8.14.2 中文 PostScript 字库 .....                          | (536)        |
| 8.4.1 CXterm .....                             | (485) | 8.14.3 使用中文 PostScript 字库 .....                        | (540)        |
| 8.4.2 crxvt .....                              | (495) | 8.14.4 其他中文打印软件 .....                                  | (544)        |
| 8.5 X Window 系统的中文字库 .....                     | (495) | 8.15 X Window 系统的中文化“包装”<br>(WRAP)方案 .....             | (545)        |
| 8.5.1 在 X Window 中使用中文点<br>阵字库 .....           | (496) | 8.15.1 PRELOAD 机制 .....                                | (545)        |
| 8.5.2 在 X Window 中使用中文<br>True-Type 字库 .....   | (501) | 8.15.2 X 函数的中文“包装”原理 .....                             | (545)        |
| 8.6 中文输入服务器——Xcin, Chinput .....               | (509) | 8.15.3 中文化“包装”软件——XA<br>(Xcin Anywhere)和 Chinput ..... | (547)        |
| 8.6.1 Xcin + crxvt .....                       | (509) | 8.16 X Window 系统中文化的基本方法 .....                         | (549)        |
| 8.6.2 Chinput .....                            | (509) | 8.17 ZWinPro——Linux 上的外挂式中文<br>平台 .....                | (551)        |
| 8.7 中文化的 X Window 编程 Widget 集<br>——EZWCL ..... | (514) | <b>9 Linux 系统安全概述 .....</b>                            | <b>(554)</b> |
| 8.8 中文编辑器——vim, Emacs<br>(MULE), chpower ..... | (518) | 9.1 设置安全帐号 .....                                       | (554)        |
| 8.8.1 vim .....                                | (518) | 9.2 确保互联网的安全 .....                                     | (556)        |
| 8.8.2 Emacs(MULE) .....                        | (519) | 9.3 对安全性进行监控和管理 .....                                  | (560)        |
| 8.8.3 chpower .....                            | (522) | 9.4 备份系统 .....                                         | (561)        |
| 8.9 英汉/汉英辞典——edict(xedict),<br>xdict .....     | (523) | 9.5 安全建议 .....                                         | (561)        |
| 8.9.1 edict(xedict) .....                      | (523) | 9.5.1 系统可能已被入侵的信息 .....                                | (562)        |
| 8.9.2 xdict .....                              | (524) | 9.5.2 反入侵的对策 .....                                     | (563)        |
| 8.10 汉化的 X 窗口管理器 fvwm95 .....                  | (524) | 9.5.3 用其他软件加强安全性 .....                                 | (565)        |
| 8.11 Big5 码的中文 X 服务器——<br>CXWin .....          | (525) | 9.5.4 一些建议 .....                                       | (566)        |
| 8.12 中文文本处理软件——LaTeX 的<br>CJK 扩展 .....         | (526) |                                                        |              |
| 8.13 Netscape 的中文设置 .....                      | (530) |                                                        |              |
| 8.14 中文打印和中文 PostScript 字库 .....               |       |                                                        |              |

## 附 录

|                                                   |              |                           |              |
|---------------------------------------------------|--------------|---------------------------|--------------|
| <b>A GNU 公用许可证 (General Public License) .....</b> | <b>(571)</b> | <b>C Linux 软件资源 .....</b> | <b>(584)</b> |
| <b>B Linux 网络资源 .....</b>                         | <b>(576)</b> | <b>D 本书所附光盘介绍 .....</b>   | <b>(628)</b> |

# 第 一 部 分

## Linux 安装与入门

---

- 1 Linux 历史及简介
- 2 Linux 的安装
- 3 Linux 的基本配置简介
- 4 Linux 使用入门



# 1 Linux 历史及简介

## 1.1 什么是 Linux

在如今的操作系统应用中,除一些专业的领域外,可以说微软正在以其 Windows 95/NT 的强劲攻势横扫全球市场,能与其相抗衡的公司越来越少,连微软最大的对手——拥有一大批忠心耿耿用户的苹果公司也不得不败下阵来,并接受微软的合作意向。但在迅猛发展的国际互联网上,有一支由编程高手、业余计算机玩家和黑客们组成的奇怪队伍,完全独立地开发出在功能上毫不逊色于微软的商业操作系统的一个全新的免费 UNIX 操作系统——Linux(发音为 Li-nucks),并成为网络上的一支不可小觑的力量,在不到 4 年的时间里,以微薄的资格成为了微软的一个强劲对手。据不完全统计,全世界使用 Linux 操作系统的人已经有近 900 多万,而且绝大多数是在互联网上使用的,目前在互联网上的 WWW 服务器大约有 28% 是在运行 Linux,比排在第二位的 Windows 95/NT 大约多出 5 个百分点。而在中国,随着 Internet 大潮的卷入,一批主要以高等院校的学生和 ISP(Internet Service Provider)的技术人员组成的 Linux 爱好者队伍也已经蓬蓬勃勃地成长起来,随着网络的不断普及,既免费又性能优异的 Linux 操作系统必将发挥出越来越大的作用。

笔者编写这本书的目的,主要是为中国的 Linux 爱好者提供一本中文的参考书,让人家在学用 Linux 的过程中不断提高自己的计算机水平,尤其是 UNIX 方面的应用水平。

Linux 是什么?按照 Linux 开发者的说法,Linux 是一个遵循 POSIX<sup>①</sup> 标准的免费操作系统,具有 BSD 和 SYSV 的扩展特性(表明其在外表和性能上同常见的 UNIX 非常相像,但是所有系统核心代码已经全部被重新编写了)。它的版权所有者是芬兰籍的 Linus B. Torvalds 先生(Linus.Torvalds@Helsinki.FI)和另外一些开发人员,并且遵循 GPL 声明(GNU General Public License)。

Linux 可以在 Intel 386,486,Pentium,PentiumPro,Pentium MMX,PentiumII,Celeron 或 Pentium III 型处理器以及 Cyrix,AMD 的兼容芯片(如 6x86,K6,K6II 等芯片)的个人计算机上运行,它可以将一台普通的个人电脑立刻变成一台功能强劲的 UNIX 工作站,在 Linux 上可以运行大多数 UNIX 程序,如:TeX,X Window 系统和 GNU 的 C/C++ 编译器。它让用户端坐在家中就可以感受 UNIX 的全部威力。如今越来越多的商业公司采用 Linux 作为操作系统,例如科学工作者使用 Linux 来进行分布式计算,ISP 使用 Linux 配置 Intranet 服务器,电话拨号服务器等网络服务器,CERN(西欧核子中心)采用 Linux 做物理数据处理,美国邮政总局采用 Linux 作为邮件分拣系统,1998 年 1 月最卖座的影片《泰坦尼克号》

---

① POSIX —— 标准操作系统界面。

的片中计算机动画的最后处理工作就是在 Linux for Alpha 平台下进行的。更有趣的是 1997 年 InfoWorld 把年度最佳技术支持奖颁给了 Linux, 给批评自由软件没有良好服务的人好好地上了一课。目前越来越多的商业软件公司宣布支持 Linux 系统, 相继推出在 Linux 上的应用软件, 如 Sybase, Oracle, Lotus, 连微软也号称要推出 Office 2000 for Linux。当然对于大多数用户来说, 最重要的一点是, 现在我们可以自己家中的计算机上进行 UNIX 编程, 并享受阅读操作系统的全部源代码的乐趣!

## 1.2 Linux 的发展历史

如果以人类的年龄来计算, Linux 还是一个只有 8 岁的小娃娃。1991 年 8 月一位来自芬兰赫尔辛基大学的年轻人 Linus Benedict Torvalds, 对外发布了一套全新的操作系统。事情的缘起是这样的: 为了实习使用著名的计算机科学家 Andrew S. Tanenbaum 开发的 Minix (一套功能简单, 简单易懂的 UNIX 操作系统, 可以在 8086 上运行, 后来也支持 80386, 在一些 PC 机平台上非常流行), Linus 购买了一台 486 微机, 但是他发现 Minix 的功能还很不完善, 于是他决心自己编写一个保护模式下的操作系统, 这就是 Linux 系统的原型。最开始的 Linux 是用汇编语言编写的。主要工作是用来处理 80386 的保护模式, 按照 Linus 本人的描述, 刚开始的时候是这样的:

“最开始的确是一次痛苦的航行, 但是我终于可以拥有自己的一些设备驱动程序了, 并且排错也变得更简单, 我开始使用 C 语言来开发程序, 这大大加快了开发速度, 但开始担心我发的誓言: ‘作一个比 Minix 更好的 Minix’, 我梦想有一天我能在 linux 下重新编译 GCC……”

“我花了两个月来进行基本的设置工作, 直到我拥有了一个磁盘驱动程序 (有很多错误, 但碰巧能在我的机器上工作) 和一个小小的文件系统, 这就是我的第 0.01 版 (大约是 1991 年 8 月下旬的事情), 它并不完善, 连软盘驱动器的驱动程序都没有, 什么事情也做不了, 但是我已经被它吸引住了, 除非我能放弃使用 Minix, 不然我不会停止改进它。”

1991 年 10 月 5 日, Linus 发布了 Linux 的第一个“正式”版本: 0.02 版, 现在 Linus 可以运行 bash (GNU 的一个 UNIX shell 程序) 或 GCC (GNU 的 C 编译器), 它几乎还是什么事情也做不了, 但是它被设计成一个黑客的操作系统, 主要的注意力被集中在系统核心的开发工作上, 没有人去注意用户的支持、文档工作、版本发布等其他东西。

最开始的 Linux 版本被放置到一个 FTP 服务器上供大家自由下载, FTP 服务器的管理员认为这是 Linus 的 Minix, 因而就建了一个 Linux 目录来存放这些文件, 于是 Linux 这个名字就传开了, 如今已经成了约定俗成的名称了。

Linus 是在 USENET 讨论区 comp.os.minix 首先发布下面这条消息的:

“用户可曾渴望 Minix-1.1 会有这样美好的一天: 人们可以自己编写驱动程序, 用户是否发现这样一个美妙的计划——人们可以自己修改操作系统以适应自己的需要? 用户是否对所有东西都在 Minix 上运行这一点感到沮丧? 用户是否找到一个在业余时间可以干的好题目? 下面这篇文章也许正是用户所需要的:

‘如同我在一个月以前所提到的那样, 我正在开发一个类似于 Minix 的基于 AT-386 的操作系统, 它现在已经可以工作了 (当然得看用户怎么想), 现在我将公布它的源代码, 它

是第 0.02 版本,但是可以运行 `bash`, `GCC`, `gnu-make`, `gnu-sed`, `compress` 等等。”

然后这个娃娃操作系统就以两个星期出一次新的修正版本的速度迅速成长,在版本 0.03 之后 Linux 将版本号迅速提高到 0.10,这时候更多的人开始在这个系统上工作。在几次修正之后 Linux 将版本号提高到 0.95,这表明他希望这个系统迅速成为一个“正式”的操作系统,这时是 1992 年,但是直到一年半之后, Linux 的系统核心版本仍然是 0.99.pl14,已经非常接近 1.0 了。

Linux 终于在 1994 年的 3 月 14 日发布了它的第一个正式版本 1.0 版<sup>①</sup>,而 Linux 的讨论区也从原来的 `comp.os.minix` 中独立成为 `alt.os.linux`,后来又更名为 `comp.os.linux`。这是 USENET 上有名的投票表决之一,有好几万用户参加了投票。后来由于使用者越来越多,讨论区也越来越拥挤又不得不再细分成 `comp.os.linux.*`,如今已经有十几个讨论组,这还不把专门为 Redhat Linux 和 Debian Linux 设的讨论组计算在内。这个讨论组也是 USENET 上最热闹的讨论组之一,每天都有数以万计的文章发表。

目前 Linux 已经是一个完整的类 UNIX 操作系统了。其最新的稳定核心版本号为 2.0.34,测试核心版本为 2.1.105。

下面就是 Linux 的吉祥物,一只可爱的小企鹅(起因是因为 Linux 是芬兰人,因而挑选企鹅作为吉祥物)。

说到这里,就不得不说一下同 Linux 密切相关的 GNU 了。如果没有 GNU, Linux 也许不会发展得这么快,可是如果没有 Linux, GNU 也不会有如今这么巨大的影响力。

什么叫 GNU?

GNU 就是 GNU's Not UNIX 的缩写, GNU 的创始人 Stallman 认为 UNIX 虽然不是最好的操作系统,但是至少不会太差,而他自信有能力把 UNIX 不足的地方加以改进,使它成为一个优良的操作系统。定义就是名为 GNU 的一个同 UNIX 兼容的操作系统,开发这个系统的目的是为了让所有计算机用户都可以自由地获得这个系统。并且任何人都可以免费地获得这个系统的源代码,可以相互自由拷贝。因而在使用 GNU 软件时我们

可以理直气壮地说我们使用的是正版软件。当然 GNU 也是有自己的版权声明(即 GPL,见附录 A),就是它有名的 Copyleft(相对于版权的英文 Copyright),就是用户获得 GNU 软件后可以自由使用和修改,但是用户在散布 GNU 软件时,必须让下一个用户有获得源代码的权利并且必须告知他这一点。这一条看似古怪的规定是为了防止有些别有用心的人或公司将 GNU 软件稍加修改就去申请版权,说成是自己的产品,其目的就是要让 GNU 永远是免费和公开的。

GNU 是谁发起的? GNU 是由自由软件基金会(Free Software Foundation, FSF)的董事

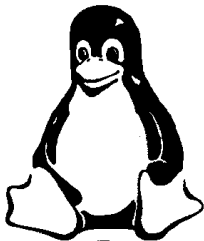


图 1.1 Linux 的吉祥物

<sup>①</sup> 按照编程人员的习惯,一个软件在第一次推出前的所有测试版本都不能称作 1.0 版,只能叫作 0.3、0.7 或 0.99 版,其数目越接近 1,就表明其越来越接近正式发布。

长 Richard M. Stallman (RMS) 于 1984 年发起的,如今已经有十几年的历史了。Stallman 本来是在美国麻省理工学院人工智能实验室从事研究工作的研究员,同时也是世界上可数的几个顶尖的程序员之一,他最著名的作品是 GNU 的第一个软件就是 GNU Emacs,UNIX 平台上的一个编辑器。这个软件推出后受到广大 UNIX 用户 的热烈欢迎,由于它同时提供源代码,大家都热心地替它排除错误、增加功能,使它的功能越来越强大,终于成为 UNIX 平台上最好的编辑器,上至 CRAY 超级计算机,下至最普遍的 PC 机,从 DOS 到 Windows,从 VMS 到 UNIX 都可以使用这个 Emacs。受到这个软件成功的鼓励,Stallman 成立了自由软件基金会,以推广 GNU 计划。基金会成立之后,主要靠一些厂家的捐献和出售 GNU 程序的使用手册,以及拷贝 GNU 软件的电脑磁带和光盘来维持,不过许多硬件厂家开始向基金会提供高性能的工作站,这其中包括 HP、SONY 和 AT&T 这样的国际性大公司。

GNU 目前已推出的软件主要有 Emacs——功能强大的编辑环境,GCC——性能优异的多平台的 C、C++、Fortran 编译器和其他 40 多种软件。其中 GCC 的成功为 GNU 带来了前所未有的影响。GCC 是一种可以在 11 种硬件平台上编译出可执行程序的超级编译器,而且其执行的效率更是惊人,与一般的编译器相比平均效率要高 20%~30%。这使得很多商业公司也采用 GCC 来开发软件,其中包括像 HP 公司这样的专业大公司。而现在回到苹果公司的前 NeXT 公司总裁史蒂夫·乔布斯也对 Stallman 尊敬有加,因为 NeXT 公司的软件都是用 GCC 开发的,在 PC 游戏中最好的 3D 射击游戏 quake(由 ID Software 公司开发)也是用 GCC 的 DOS 移植版本 djgpp 编写的。GCC 是 C、C++ 和 Objective C(由 NeXT 公司贡献)三者合一的编译器,它的编译原理同大多数的编译器不一样,它先有一个前端处理程序将 C、C++ 和 Objective C 的语句转换成为一个类似于 Lisp 的内部语言——RTL,再由一个后端处理程序将其优化后产生目的 CPU 可以执行的机器代码,因而对于每种新语言来说,只要写好一个新的前端处理程序就可以立刻将此语言移植到 GCC 已经支持的不同硬件平台上去,而且编译出的已经是经过优化的二进制代码。除了 C 系列以外,GCC 还有 Fortran 77,ada9x,Pascal 的前端处理程序。GCC 多平台的实现方法同 JAVA 的虚拟机技术不太一样,因而其运行速度远非 JAVA 能望其项背。

GNU 开始的策略就是先开发 UNIX 已有的程序(因为计划中 GNU 将会是一个与 UNIX 兼容的操作系统,而 UNIX 下的标准使用界面已经非常完善了,所以先依照这个标准开发应用程序,将来 GNU 自己的系统核心一出来,就可以使用所有先前开发的 GNU 应用程序),因此会先有 emacs、GCC 等工具软件开发出来。其中 GCC 更是 GNU 计划自立更生的一个重要步骤(不需要依靠商业软件来产生/编译整个 GNU 系统),如今这套程序开发工具已经很成熟。下一步要做的就是开发 GNU 的核心——Hurd 了,只要 Hurd 开发出来,GNU 就是一套能够自我开机,真正自给自足、完整的自由操作系统。Hurd 是基于 Mach 这个微内核(micro kernel)上的操作系统核心。美国犹他大学的 Mach4(一个更为方便和快速的微内核),由于 Hurd 基于 Mach,所以只要 Mach 移植到什么机器上,Hurd 也立刻就是这种硬件平台的操作系统,就这一点而言,它是一种很前卫的操作系统。

Linux 在短短的几年内崛起,也使用 GNU 的版权声明,大有取代 Hurd 之势。在网络上甚至有不少人认为 Hurd 应该停止开发,因为 Linux 系统已经十分成熟,GNU 实际上可以使用 Linux 当作核心,就能构成一套完整的操作系统。甚至连自由软件基金会的技术人员也承认,要是 Linux 早几年出现,也许就不用发展 Hurd 了。但是 Linux 还是晚了一

步,Hurd 已经计划很久了,技术上有很多非常领先的地方,能够支持的硬件平台也比 Linux 多,不完成就太可惜了。1994 年 11 月,Hurd 的第一个雏形已经发表;1995 年 4 月,发表了第二次 snapshot。Hurd 目前的版本是 0.2,在 1997 年 6 月发布,已经可以独立安装与稳定运行了,GNU 版本的 Mach 目前的版本是 1.1.3。与此同时,已经开发出一个基于微内核的 Linux 版本了,目前有一个可以在 HP-PA RISC 机器上运行的 Linux 系统。实际上,自由软件基金会日前有一套以 Linux 为核心的 GNU 系统计划。这个计划就是自由软件基金会支持的 Debian Linux。Debian Linux 是 GNU 软件与公共软件(Public Domain Software)最彻底的结合,它是一套完全由 GNU 和公共软件配置起来的,而且是具有商业操作系统水平的操作系统。

Linus 被视为 Linux 的教父,虽然他日前已经不如 Alan Cox 的投入大,但是作为 Linux 的代表,他对整个业界都有着举足轻重的影响。日前 Linus 的声望已经有超过 Stallman 的趋势,例如在 1998 年的 Linux Expo World 后有人就写了一篇文章:“All link for Linus”来替 Stallman 抱打不平,但是 Linus 比 Stallman 更为务实也是不可争议的事实,例如对于商业公司利用 Gnu 软件挣钱的想法,他们的意见就不太统一。

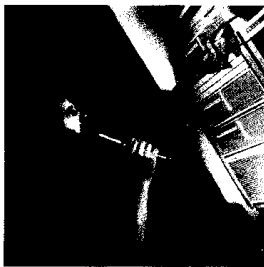


图 1.2 GNU 创始人 Stallman



图 1.3 Linux 创始人 Linus

## 1.3 Linux 与 UNIX 的关系, Linux 的版本介绍

### 1.3.1 UNIX 的起源介绍

首先让我们为 UNIX 下一个简短的定义。我们所提的 UNIX 指的是一个通常由 C 语言写成的操作系统,它有树状的文件系统,集合了文件和设备(device) I/O,其系统函数调用(system call)界面包含了如 fork(), pipe() 等函数,而且它的用户界面包含 cc, troff, grep, awk 之类的工具及一个 shell。UNIX 过去是 USL(AT&T) 的注册商标,现在则是 X/Open 的注册商标。但是我们一般所指的 UNIX 是一般通用的意义,而不是这个注册商标。

UNIX 的历史开始于 1969 年 Ken Thompson, Dennis Ritchie (即著名的 K&R, C 语言的发明人)与一群人在一部 PDP-7 上进行的一些工作,后来这个系统变成了 UNIX。它主要的几个版本为:

V1 (1971): 第一版的 UNIX,以 PDP-11/20 的汇编语言写成。包括文件系统, fork, roff 和 ed 等软件。

V4 (1973): 以 C 语言从头写起,这使得 UNIX 修改容易,可在几个月内移植到新的硬件平台上。最初 C 语言是为 UNIX 设计的,所以 C 与 UNIX 间有紧密的关系。

V6 (1975): 第一个在贝尔实验室外(尤其是大学中)广为流传的 UNIX 版本。这也是 UNIX 分支的起点与广受欢迎的开始。1.xBSD(PDP-11)就是由这个版本衍生出来的。

V7 (1979): 在许多 UNIX 玩家的心目中,这是“最后一个真正的 UNIX”,这个版本包括一个完整的 K&R C 编译器, Bourne shell, V7 移植到 VAX 机器后称为 32V。

目前开发 UNIX(System V) 的公司是 UNIX System Laboratories(USL)。USL 本为 AT&T 所有,1993 年初被 Novell 收购。Novell 于 1993 年末将 UNIX 这个注册商标转让给 X/Open 组织。

目前为止,UNIX 有两大流派:那就是 AT&T 的 System V 与 BSD(Berkeley Software Distribution)。SVR4 是两大流派融合后的产物。1991 年底,与 System V 针锋相对的开放软件基金会(Open Software Foundation)推出了 OSF/1。

下面是 SVR 系列和 BSD 系列特性的比较:

| 特性        | 典型的 SVR <sub>x</sub>                | 典型的 xBSD                            |
|-----------|-------------------------------------|-------------------------------------|
| 核心名称      | /UNIX                               | /vmUNIX(Linux)                      |
| 终端启动文件    | /etc/initab(Linux)                  | /etc/ttya                           |
| 开机启动文件    | /etc/rc.d 目录(Linux)                 | /etc/rc.* files                     |
| 挂上的文件系统   | /etc/mstab                          | /etc/mstab(Linux)                   |
| 常用的 shell | sh, ksh                             | csh                                 |
| 原生文件系统    | 55 (块大小: 512 ~ 2K)<br>文件名 <= 14 个字符 | UFS (块大小: 4K ~ 8K)<br>文件名 < 255 个字符 |
| 用户组       | 必须使用 newgrp(1)<br>SVR4: 多重用户组       | 自动加入成员                              |
| 打印系统      | lp, lpstat, cancel                  | lpr, lpq, lpnm lpd(Linux)           |
| 终端控制      | termio, termio.h                    | termios                             |

续表

| 特 性   | 典型的 SVR <sub>x</sub>  | 典型的 xBSD        |
|-------|-----------------------|-----------------|
| 进程控制  | SVR4: termios (POSIX) | termcap (Linux) |
| ps 指令 | > = SVR4 支持           | 支持              |
| 多重等待  | ps -ef                | ps -aux         |
| 字符串函数 | poll                  | select          |
| 程序对映  | memset, memcpy        | bzero, bcopy    |
|       | /proc (SVR4)(Linux)   |                 |

下面介绍目前几种常见的 UNIX 版本:

AIX: IBM 的 UNIX, 是根据 SVR2(最近已经出到 SVR3.2) 以及一部分 BSD 延伸而来, 加上各种硬件的支持, 具备特有的系统管理(SMIT)。

386BSD: Jolitz's 从 Net/2 software 移植过来的。支持 Posix, 32 位。

BSD/386 (80x86): 来自 BSDI, 附源程序。

Coherent (80x86): 另一种 UNIX, 与 V7 兼容, 有一些 SVR2 的特征(IPC)。

DELL UNIX: SVR4。

FreeBSD: 1.x 从 386bsd 0.1 而来, FreeBSD 2.x 版是用 4.4BSD lite 改写。

HP-UX (HP): 旧系统是从 S III (SVR<sub>x</sub>) 发展而来的, 现在是由 SVR2 (4.2BSD) 发展而来, 目前是 10.x 版。

Interactive SVR3.2 (80x86): 纯种 SVR3, Interactive 已经被 Sun 收购。

Idris: 第一个 UNIX 兼容产品, 是由 Whitesmith 完成的。

IRIX (SGI): Version 4 是基于 SVR3.2, 并且包括许多 BSD 的特性。Version 5.x 是基于 SVR4 的。

Linux (x86): 遵从 POSIX, SYSV 及 BSD 的扩展, 这一点从上表中即可看出。

Mach386: 来自 Mt Xinu。以 Mach 2.5 为基础, 附加 4.3BSD-Tahoe 增强功能。

Microport (80x86): 纯种的 SVR4, X11 和 OpenLook GUI。

Minix (80x86, Atari, Amiga, Mac): 与 V7 兼容的 UNIX, Linux 的鼻祖。

NetBSD: 与 FreeBSD 同一分支, 以 4.4BSD lite 为基础, 支持较多的硬件平台。

NEXTSTEP (Intel Pentium and 86486, Hewlett-Packard PA-RISC, NeXT 68040): 基于 Mach 核心的 4.3BSD, 具有良好的用户界面, NEXTSTEP 遵从 OpenStep 的标准。

OSF/1 (DEC): DEC 对 OSF/1 的移植。

SCO UNIX (80x86): SVR3.2, 目前影响较大的 PC UNIX。

SunOS (680x0, Sparc, i386): 根据 4.3BSD, 包含许多来自 System V 的东西。Sun 的主要成果在于: NFS, OpenLook GUI 标准, 现演变为 Solaris。

Ulrix (DEC): 根据 4.2BSD 再加上许多 4.3BSD 的东西。

Xenix (80x86): Intel 硬件平台上的 UNIX, 以 SVR2 为基础, 由微软推出。在中国使用较广泛。

由此可见, Linux 的确是 UNIX 的一种, 但是 Linux 是有史以来使用人数最多的一种

UNIX,也是对个人电脑玩家最友好的一种 UNIX。<sup>①</sup> 对于这一点,读者通过阅读后面的章节可亲身感受到这一点。很简单,Linux 是第一个完完全全由最终用户(电脑玩家)为了自己的需要而开发出来的操作系统。

### 1.3.2 Linux 的特性以及为什么使用 Linux

Linux 继承了 UNIX 不少优点,但是也具有自身的一些独特的特点,下面将分别予以介绍:

#### 多任务

多任务指的是计算机在同一时间内运行多个应用程序的能力。如用户可以一边编译系统核心一边编辑另外一个文件。这对于用户最大限度地利用计算机资源是很有好处的。UNIX 是典型的多任务系统,Linux 也是一个多任务的系统。

#### 多用户

多用户指的是多个用户在同一时间内使用一台机器。而且 Linux 不像某些商业操作系统那样有 licenses 的限制,在实际应用中,很多大学的 BBS(电子公告牌)服务器使用的就是 Linux,一个普通的 BBS 站使用操作系统为 Linux 的普通微机,同时上线人数能达到 200 人以上,这可不是一个小数目。

#### 多平台

虽然 linux 主要在 x86 平台上运行,但是目前已经移植到下列平台:Alpha、Sparc。RedHat 公司已经推出了这两个平台的发行套件。Linux 对其他硬件平台的移植也在进行之中。下面给出一些相应的网络地址:

|                 |                                                                                                                                         |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| ARM Linux       | <a href="http://whirligig.ees.soton.ac.uk/~rmk92/armlinux.html">http://whirligig.ees.soton.ac.uk/~rmk92/armlinux.html</a>               |
| Linux/68k       | <a href="http://www-users.informatik.rwth-aachen.de/~hn/linux68k.html">http://www-users.informatik.rwth-aachen.de/~hn/linux68k.html</a> |
| Linux/8086      | <a href="http://www.linux.org.uk/Linux8086.html">http://www.linux.org.uk/Linux8086.html</a>                                             |
| Linux/Alpha     | <a href="http://www.azstarnet.com/~axplinux/">http://www.azstarnet.com/~axplinux/</a>                                                   |
| Linux/MIPS      | <a href="http://www.waldorf-gulih.de/linux-mips-faq.html">http://www.waldorf-gulih.de/linux-mips-faq.html</a>                           |
| Linux/PowerPC   | <a href="http://liber.stanford.edu/linuxppc/">http://liber.stanford.edu/linuxppc/</a>                                                   |
| Linux for Acorn | <a href="http://www.ph.kcl.ac.uk/~amb/linux.html">http://www.ph.kcl.ac.uk/~amb/linux.html</a>                                           |
| MacLinux        | <a href="http://www.ibg.uu.se/maclinux/">http://www.ibg.uu.se/maclinux/</a>                                                             |

对应用程序使用的内存进行保护

在 Linux 下应用软件无法访问系统分配内存以外的内存区域。这样一个软件的错误操作不会造成整个系统的瘫痪,在 Windows 3.1 或 Windows 95 下经常出现一个软件把整个系统锁死的情况,但在 Linux 下是非常罕见的。作为一种非正式的解决方案,Linux 对 UNIX 系统中经常出现的由于溢出(overflow)而造成的系统安全漏洞也已经有了解决之道。“按需取盘”

在 Linux 下任何一个执行文件在执行时,只有那些确实被用到的代码段才会被系统读取到内存中,这样节约了大量的读取磁盘的时间,自然也就加快程序执行速度。并且这还是在操作系统级实现的,而不像 DOS 下是要靠应用程序 smartdrive 来管理,性能的差别是

---

<sup>①</sup> System V: AT&T 发行的 UNIX 操作系统。BSD (Berkeley Software Distribution): 加州大学伯克利分校发布的 UNIX 版本。其中 x 代表它们的发布版本号。

很大的。

共享内存页面

在 Linux 下,多个进程可以使用同一块的内存页面(每片大小为 4K),只有在某一个进程试图对这块页面执行写操作时 Linux 将把这块页面作为该进程复制到内存的另一块区域(copy-on-write),这样做的好处是不仅加快了程序运行的时间,而且节约了宝贵的物理内存。

使用分页技术的虚拟内存

在 Linux 下系统核心并不把整个进程交换到硬盘上,而是按照内存页面来交换的。虚拟内存的载体不仅可以是一个单独的分区,而且可以是一个文件。(如果用户同时使用 Windows 95 或 Windows 3.1 的虚拟内存时,Linux 还可以同他们共享同一个交换文件,这是对硬盘紧张的用户的一个非正式的解决办法。)Linux 还可以在系统运行时临时增加交换内存,而不用像某些 UNIX 系统那样需要重新启动才能使用新的交换空间。理论上 Linux 可以使用多达 16 个 128M 大小的交换文件,也就是说 Linux 的虚拟内存最多可以使用 2G ( $16 \times 128M = 2048M$ )的内存空间,这一点对某些进行科学计算的用户来说也许是非常有用的一个特性。

优秀的磁盘缓冲调度功能

Linux 最突出的一个优点就是它的磁盘 I/O 速度,因为它将系统没有用到的剩余物理内存全部用来作为硬盘的高速缓冲,当有对内存要求比较大的应用程序运行时,它将会自动将这部分内存释放出来给应用程序来使用。同 Dos/Windows 下的 Smartdrv 只能呆板的使用固定大小的缓冲区相比要先进得多。与 Linux 竞争的 FreeBSD 在这一点上同 Linux 一直有一些差距。因而对于那些需要运行大型软件(project)的用户来说,Linux 是 x86 上能找到的效率最高的操作系统。

动态链接共享库(dynamically linked shared libraries)

同 Windows 3.1 和 Windows 95 的 DLL 一样,Linux 也使用动态链接共享库(同时当然也提供静态链接库)这个特性可以大大减小 Linux 应用程序的大小,例如一个普通的 Motif 应用程序如果使用动态库,其程序大小只有 50K 左右,但是一旦在编译时改成静态链接时(static link),则该应用程序的大小将激增到 2M 左右。(动态链接共享库是在程序运行时才动态链接的,Linux 通过 LD\_PRELOAD 这个变量让开发人员可以使用自己的程序库中的模块来替换系统模块,这一点对于 X Window 下的汉化工作是非常重要的)并且被很多程序同时调用的一段代码只被加载一次,由众多程序共享。

Core Dump<sup>①</sup>

作为一个“自助”式的操作系统,Linux 可以让用户不仅在程序运行时对它进行排错(debug),还可以让用户在程序当掉之后产生出一个 coredump 文件(一般它的文件名为 core),然后用户就可以使用

`gdb -core = 文件名`

---

<sup>①</sup> Core Dump 的含义:在半导体出现之前,人们使用线圈来制作计算机内存,线圈就叫作 core,相应的内存就叫 core memory,如今虽然已经没有人使用线圈来制作内存,但是人们习惯上还是把内存称为 core,core dump 指的是系统在应用程序发生故障时,将当时的内存中的内容保存到硬盘上的文件中。

来对这个出错的地方来进行排错,换言之,Linux 在操作系统级给编程留出了调试接口。这个特性一般用户并不经常使用,但是对于开发人员来说却是一个非常有用的特性。如最新的 Visual C++ 就抄袭了这一点,只是并不完善而已。如果硬盘上到处都是这种垃圾文件,那么手工删除它就可以。(注意 /proc 下的 kcore 文件不是 coredump 文件,不可删除!!!)如果想不产生这种文件,那么在用户的根目录下的 .cshrc 文件中加入下面一行:

```
limit coredumpsiz: 0
```

系统就不会产生任何 coredump 文件。

支持伪终端设备(pseudoterminal,pty)

允许同时有很多用户从网络登录到系统上,每个登录进程使用一个伪终端设备。这些终端是动态收集的,一个废弃的终端很快就会被回收。Linux 缺省的伪终端数是 64 个,如果有超过这个数目的用户使用,只需要做一个简单的补丁就可以使用 256 个直到 1024 个虚拟终端。

支持多个虚拟控制台

用户可以在一个真实的控制台前登录多个虚拟控制台,并可以使用热键在这些虚拟控制台之间切换(缺省为 Alt + F1 - Alt + F6,或者是 Alt + 左右方向键)。

支持多种 CPU(支持多处理器 SMP)

Intel/AMI/Cyrix 出品的 x86 系列 CPU

386SX/DX/SL/DX1/SLC

486SX/DX/SL/SX2/DX2/DX4

Pentium

Pentium MMX

Pentium Pro

Pentium II

K6,M2,Cyrix 6x86 等

支持数学协处理器 387 的软件模拟

早期的一些 CPU 无 FPU 协处理器,由于 Linux Kernel 支持模拟 FPU,所以 Linux 可以在没有 FPU 的机器上运行,只是系统的速度将会有所下降,尤其是进行编译和图形处理等方面的工作时。但现在 Pentium 系列 CPU 已经成为主流,因此这个问题已不复存在。

支持的硬件多

Linux 支持的硬件列表相当广泛,从硬盘驱动器、软盘驱动器、主板、显示卡到 SCSI 卡、声卡、磁带机、光驱/光盘刻录机、网卡、ZIP/MO 驱动器和 video 设备等等,只要不是太冷僻的设备,在 Linux 中都可以找到相应的驱动程序。比起必须使用一些最通用的外设的 FreeBSD 来,Linux 的驱动程序的数目简直是一个天文数字,原因很简单:为 Linux 开发设备驱动程序的程序员实在是太多了。

在 Linux 开发的初期编写驱动程序是一件非常困难的事情,因为众多的硬件厂商不愿意向自由程序员提供相关的硬件细节,结果 Linux 的程序员们只有独自在黑暗中摸索。这种情况现在已经得到了显著的改善,许多硬件厂商不仅派出技术人员协助 Linux 开发者开发硬件驱动程序,甚至自己就提供 Linux 版的驱动程序。因为 Linux 在全世界拥有数亿万量级的计算机用户,并且其中大多数人是具有一定水平的技术人员,这个市场是任何一家硬件厂家无法忽视的。相信在不久的将来就会有“Linux compatible”的标志贴在硬件

产品的包装上,而“Linux Inside”也许像“Intel inside”一样会成为一时髦的标志。

支持多种键盘

包括最新的微软键盘,支持多国语言键盘布局,还可以自己定制键盘。

支持多种文件系统

Linux 支持的文件系统的种类包括: minix, ext, ext2, xiafs, hpfs, fat, msdos, umsdos, vfat, proc, ufs, iso9660, smbfs, ncfs, affs, ufs, ronfs, sysv, xenix 和 coherent(以上的定义在 /usr/src/linux/fs/filesystems.c 中可以找到), Linux 可以将这些文件系统直接装载(mount)为系统的一个目录。Linux 自己的文件系统 ext2fs 是非常先进的,最多可以支持到 2T 的硬盘,文件名长度的限制是 255 个字符。如果使用 umsdos 文件系统,用户可以将整个 Linux 安装到 DOS 下的一个目录中,不用将硬盘重新分区和格式化。同时在 DOS 和 Windows 95/NT 下也都有工具来直接读取 Linux 文件系统上的文件。同时 Linux 还支持以只读方式打开 HPFS-2 格式的 OS/2 2.1 的文件系统和 HFS 格式的 Macintosh 文件系统。

支持 POSIX 的任务控制

丰富的软件

Linux 大概是应用软件最多的 UNIX 操作系统了。下面简单列出 Linux 上的常用软件。

基本 UNIX 命令:

ls, tr, sed, awk 等等。

编程工具:

gcc, gdb, make, bison, flex, perl, rcs, cvs 和 prof 等。

编程语言和环境:

C, C++, Objective C, Java, Modula-3, Modula-2, Oberon, Ada95, Pascal, Fortran, ML, scheme, Tcl/tk, Perl, Python 和 Common Lisp 等。

图形用户界面:

X11R6.3 (XFree86 3.x), fvwm, olvwm, KDE, Afterstep, WindowMaker, Icewm, Gnome, Enlightenment, XFCE 和 CDE 等。

编辑器:

GNU Emacs, jove, ez, epoch, elvis(GNU vi), vim, joe, pico, jed 和 xwpe 等。

Shells:

bash (同 POSIX sh 兼容), zsh (同 ksh 兼容), pdksh, tesh, csh, rc, es 和 ash 等。

拨号软件:

Taylor UUCP, SLIP, CSLIP, PPP, kermi, szrz, minicom, pcomm, xcomm, term 和 Seyon 等以及收发传真的软件和语音信箱软件。

网络新闻阅读器和电子邮件:

C-news, inn, trn, nn, tin, smail, elm, mh 和 pine 等。

文字处理:

TeX, groff, doc, ez, linuxdoc-SGML, SiagOffice, Koffice, AbiWord 等。商业版的有 Star Office, 一套可以同 MSOffice 相媲美的重量级办公套件。

#### 游戏:

Nethack, X games, 现在最流行的 mud 中的 LPmud——东方故事系列和 DikuMud 等都可以在 Linux 上运行。也有商业公司 Id Software 推出的优秀射击游戏 doom, doom2, quake, quake2 以及 Heretic, CivIII 和利用 Wine 运行的 Starcraft 和 Unreal。

#### 图形图像处理:

XView, Gimp(X Window 上最好的图像处理软件)和 ImageMagick。

#### 中文支持:

X 下的中文终端: CXterm, Crxvt

字符方式下的中文终端: WZCE, Chdrv, Yaact, A4TTY, KONGB, CCE

中文字处理软件: Chpower, CJK(中文 latex)

中文 X 服务器: CXwin(目前只有 Big5 版本)ZWinPro, ZhXWin

中文输入软件: XA + Xcin, Chinput

中文内码即时转换软件: hzty

中文打印软件: cnprint

#### 模拟器:

DOSEMU, WINE, WABI(sun 公司的 windows 模拟器 Linux 版比较稳定)等, 还有任天堂红白机模拟器 snes, Apple II 模拟器街机模拟器 XMAME 等。

#### WWW 浏览器:

netscape, arena, Mosaic, lynx(功能强大的字符浏览器)

#### 多媒体:

l3dec, mpeg123, X11amp(mp3 播放器), mtv(Linux 下观看 VCD 的工具), midiplay, cdplayer 和 workbone

#### 科学工具:

magic, spice, gnuplot 等

以上所举只是 Linux 的软件的沧海一粟而已, 更多的软件可以参见附录 C, 另外每时每刻网络上都有 Linux 的新软件开发出来, 可以预见在 Linux 系统口的工作会越来越方便的。

#### 软件移植性好

由于早期的 Linux 不完全遵守 POSIX 标准, 使得在 Linux 平台上编译传统 UNIX 软件成为一件十分困难的事情。相对于 Linux 的竞争对手——FreeBSD 来说在 FreeBSD 平台上进行软件移植较为容易, 但是这个情况随着两个趋势的发展已经发生了逆转: 首先是 Linux 越来越靠近 POSIX 标准, 在最新的 RedHat 5.0 中更全面采用 glibc 2, 在 Linux 平台上编译软件的环境已经大为改善。其二是 Linux 的使用者随着互联网的发展而急剧扩大, 而 FreeBSD 的开发队伍(core team)还是只有 50 多个人。现在 Linux 上很多软件根本就不需要移植, 因为他们本来就是由 Linux 爱好者专门为 Linux 编写的, 如果要移植到其他 X86 UNIX 下反而要化大力气修改, 例如 DOSEMU, SMBFS 等, 现在反而是 FreeBSD 在移植 Linux 的设备驱动程序。

#### 与其他 UNIX 系统的兼容性

Linux 同大多数 POSIX, SYSTEM V, BSD 等 UNIX 系统在源代码级兼容, 通过 iBCS2 兼

容的模拟模块, Linux 可以直接运行 SCO、SVR3、SVR4 的可执行程序。

强大的网络功能

TCP/IP 网络包括 ftp, telnet, NFS 等。

Linux 在最新发展的核心中包含的基本协定有 TCP, IPv4, IPv6, AX.25, X.25, IPX, DDP (Appletalk), NetBEUI, Netrom 等。稳定的核心中目前包含的网络协议有 TCP, IPv4, IPX, DDP, and AX 等协议。

提供 Netware 的客户机和服务器,以及现在最热门的 Samba (让用户共享 Microsoft Network 资源,也就是用户在 Windows 95 中看到的网络邻居,或者让 Windows 95 共享 Linux 下的资源,也就是说用户可以在 Linux 文件系统中安装 Windows 95 的软件!!)当然不足之处就是 Samba 还只是字符界面,但图形界面的 Ksamba 很快就要问世, Linux 还包括 Appletalk 服务器。

提供全部源代码

Linux 最后也是最伟大的一个特性就是它的全部源代码是免费、公开的,这包括整个系统的核心,所有的驱动程序,开发工具包以及所有的应用程序。任何人只要有兴趣都可以将整个 Linux 系统重新编译一遍(当然如果用户有足够的兴趣和时间的話)。用户可以在 Linux 的源代码中观察系统核心的运转、查看 Telnet, Ftp 是如何实现的,整个 Linux 操作系统对于用户就像是一个透明的发动机。Linux 公布的全部源代码都遵循 GPL 宣言,在 Linux 早期的版本,因为开发者不愿意自己的程序被不法的商业软件厂商窃取,甚至喧宾夺主,因此使用了一个相当严格的版权声明,禁止一切的商业行为。后来因为广大的用户希望有人能够把 Linux 压成 CD-ROM,推广 Linux,以方便无法通过 Internet 获得 Linux 的用户(本书也附有一张 LinuxCD),所以将版权宣言换成 GPL。从此之后, Linux 便是公认的 GNU 操作系统。

提供源代码,其中的一大好处是:排错变得容易。固然公布源代码之后寻找系统漏洞更加容易,同样的弥补系统的漏洞也会容易,因而系统的健壮性要强于大多数的商业操作系统。下面有几个著名的例子:前一段在网络上传传甚广的“泪滴”(teardrop)攻击法,使得无数使用 Windows 95/NT 和 UNIX 作为操作系统的网络服务器瘫痪,还有硬件厂商 Intel 出品的奔腾,高能奔腾系列的 CPU 发现的 BUG 使在 Intel 平台上工作的任何一个用户都可以将整个系统在一瞬间卸栽掉。那些商业操作系统的用户只有停止一切工作,等待生产操作系统的公司公布补丁程序,而公布了源代码的 Linux 在几天内就得到了相应的“补丁”程序, Linux 的核心更是立刻升级到 2.0.33,系统很快就恢复了正常,相比而言 Linux 的解决方案是比较彻底的一种,而 Windows 95/NT 在给出新的 service pack 之后,等用户更新了系统,目前又有“新泪滴”来攻击 95/NT 的机器。在这种处理模式下用户永远是局外人,用户不清楚操作系统出了什么问题,所有的出错信息也许只是蓝底白字的一堆谁也看不懂的文字,或者是一声不响地结束。试想一下那些商业操作系统如果公布了源代码会出现什么情况?那将是一场灾难,无数的“臭虫”会出现,系统将在无处不在的网络上弱不禁风,现在固然不会出现这种情况,但用户只要一想到使用的操作系统里埋藏着这么多定时炸弹(逻辑锁?),不会不寒而栗吗?至少, Linux 之类的操作系统让用户死得像一个“明白鬼”。在这种不断搜寻“臭虫”的过程中,那些具有一定编程水平的用户,他的编程水平将会迅速提高,每天阅读世界上顶尖程序员编写的程序,对于用户的编程水平自然是一个

很大的促进。公布源代码后,为 Linux 开发新的驱动程序,替 Linux 下的应用程序开发新的功能,甚至自己编写新的应用程序都变得更加容易(在 Linux 下,需要什么功能,就自己去编写,“Just Programming it!”)。而目前一些商业操作系统的发展过程造成这样一种局面:用户使用越来越简单,为操作系统编写底层应用程序变得越来越困难,很多程序员变成只会用可视开发工具开发界面的机器人,对于底层系统熟悉的程序员越来越少,所有技术都越来越集中在几个大公司中。而 Linux 虽然对于初学者来说的确过于困难了一点,但 Linux 用户的计算机水平也不是其他用户能比拟的。与此同时,Linux 下的软件也会越来越多,用户也能更快地享受最新硬件的强大功能,而对于 UNIX 一向为人所批评的不友善的用户界面,现在正在紧锣密鼓地进行的 KDE 计划也正在迅速改变这一点。

说到这里,笔者不禁想提一段不相干的话。1998 年 1 月, Netscape 公司发布了一个爆炸性的宣言, Netscape 5.0 的 Client 端的源代码将公布且遵循 CPL, 这样在 GNU 的软件群中又增加了一个生力军,估计现在网络上会有不少的顶尖程序员等着为 Netscape 增加新的功能,替 Netscape 排错,将 Netscape 移植到更多的操作系统上,在 Netscape 与微软的 IE 的生死之战中, Netscape 作出的这个决定是福是祸,且让我们用户拭目以待,至少对于我们中国用户来说一定很快会出现专门为我们定制的中文 Netscape,相信对于国内的软件厂商来说是迅速抢占网络市场的一个绝好商机,希望不久的将来上网软件市场不再是国外软件厂商一统天下,当然在 Linux 下从事这项工作的自然是中国自己的自由程序员们。

提供源代码使 Linux 对于我们科研人员 and 高等院校的师生们是学习操作系统设计的一个很好的工具。在国外的大学中,很多计算机方面的教师们早就使用 Linux 来讲授操作系统的原理和设计,我们要想设计自己的操作系统就不能不掌握这些知识。

### 1.3.3 Linux 的版本介绍

任何一个软件都有版本号,例如微软的 Windows 95, Office 97 与 VC5.0 等等, Linux 也不例外。Linux 的版本号又分为两部分:内核(kernel)与发行套件(distribution)版本。Linux 初学者常会把内核版本与发行套件版本弄混,实际上内核版本指的是在 Linux 领导下的开发小组开发出的系统内核的版本号,目前最新的版本的序号大约是 2.2.5 与 2.1.132。(一般说来,以序号的第二位为偶数的版本表明这是一个可以使用的稳定版本,如 2.0.36,而序号的第二位为奇数的版本一般有一些新的东西加入,是不一定很稳定的测试版本,如 2.1.132。)而一些组织或厂家将 Linux 系统内核与应用软件和文档包装起来,并提供一些安装界面和系统设定与管理工具,这样就构成了一个发行套件,例如最常见的 Slackware, RedHat, Debian 与 TurboLinux 等。实际上发行套件就是 Linux 的一个大软件包而已。相对于内核版本,发行套件的版本号随发布者的不同而不同,与系统内核的版本号是相对独立的,例如 Slackware 4.0, RedHat 5.0, Debian 2.0 和 TurboLinux 3.0.2 等。

实际上对于 Linux 新手而言,发行套件版本也许更重要些,而对于自己经常更新系统的老手来说内核版本才是注意的焦点。下面简单介绍一下 Linux 的常见发行套件。因为中国用户能接触到的主要是 Slackware, RedHat 与 Debian 这三种发行套件,下文将针对这三种发行套件进行讲解,而对其他的发行套件仅仅作一个粗略的介绍。

## 1. Walnut Creek Slackware(见图 1.4)

Slackware Linux 的发源地是 [ftp.cdrom.com](http://ftp.cdrom.com), Walnut Creek CDROM 公司是正式的 Slackware 版本的发布者。其通信地址:

Walnut Creek CDROM  
4041 Pike Lane, Suite D  
Concord, CA 94520

E-mail 地址:

[info@cdrom.com](mailto:info@cdrom.com) (信息)  
[order@cdrom.com](mailto:order@cdrom.com) (订购)  
[support@cdrom.com](mailto:support@cdrom.com) (支持)。

网址:

WWW: <http://www.cdrom.com/titles/cn/slack96.htm>  
FTP: <ftp://ftp.cdrom.com/pub/linux/slackware>



图 1.4 Slackware 的标志

Slackware Linux 是最早出现的 Linux 发行套件之一,也是笔者最早接触的一个发行套件,它是由 Patrick Volkenking 先生制作的。开始,它的载体是软盘,因此其安装时的目录结构一直保留着 A1, A2, ... 以软盘为单位的安装系列,对于没有光驱的用户,它也许是唯一的选择。Slackware 的特点就是安装简单,目录结构清楚,版本更新快,在 1997 年的一年内就推出了好几个版本。其缺点是软件种类不如 RedHat 和 Debian 多(据粗略统计 Slackware 96 缺省的软件包约有 418 个 .tgz 文件, Red Hat 4.2 缺省的软件包约有 460 个 .rpm 文件, Debian 1.3.1 缺省的软件包则约有 1148 个 .deb 文件),并且其安装不如 RedHat 快速、简洁、直观。Slackware 只提供字符方式的安装界面,并且根据需要用户自己去寻找不同硬件的启动盘。其升级方式也不如 RedHat 和 Debian 简单,同 RPM 和 DEB 相比,Slackware 只有一个相对简陋的 pkgtool,经常会出现卸载软件后其他软件使用不了的故障。但是 Slackware Linux 最大的资本就是它是最为普及的 Linux 发行套件之一,有很多出版公司以 Slackware 为基础重新包装出版。其中 Walnut Creek CDROM 公司是其中最出名的一家(互联网上最大的软件 FTP 节点 [ftp.cdrom.com](http://ftp.cdrom.com) 就是由他们维护的)。此外 Slackware 简单的目录结构,清楚的配置文件也是它吸引用户的地方,其预装软件数目少的缺点却常常是喜欢自己安装软件的富有经验的 Linux 老手们选择 Slackware Linux 的原因。据 [www.li.org](http://www.li.org) 的统计,目前使用 Slackware Linux 的用户约占 55%。

## 2. RedHat Linux(见图 1.5)

RedHat Linux 是由 Red Hat software 公司发布的。其通信地址:

Red Hat Software  
3201 Yorktown Rd, Suite 123 DeKalb Center  
Durham, NC 27713

E-mail 地址:

[redhat@redhat.com](mailto:redhat@redhat.com)

网址:

WWW: <http://www.redhat.com>



图 1.5 RedHat 的标志

FTP: <ftp://ftp.redhat.com>

RedHat 的问世比 Slackware 和 Debian 要晚。其后来者居上,有凌驾于这两者之上的趋势。(RedHat Linux 曾被权威计算机杂志 InfoWorld 评为最佳 Linux 套件。)原因是 Red Hat Software 公司实力很强,将商业公司和自由软件开发者的优点融合起来,制作出一套非常优秀的 RedHat Linux(昵称为小红帽 Linux,其吉祥物也是一个头戴红帽的人)。RedHat Linux 的优点如下:

支持硬件平台多

Red Hat Linux 从 4.0 版起,便同时支持 Intel, Alpha, Sparc 三种硬件平台,这正是 Red Hat 公司引以为荣的地方。到目前为止其他发行套件仍没有同时发布支持这三个平台版本的。目前 RedHat 最新版本为 5.0 版。

优秀的安装界面

从 4.x 版开始,RedHat 公司对 Redhat Linux 的安装界面进行了大的手术,由原先 3.x 版本的在大批 img 文件中去寻找启动盘,变成只用制作一张启动盘就可以进行 CDROM 方式的安装工作,如要以 NFS,ftp 方式安装才需要制作第二张 supp 盘。整个安装过程非常简单明了,用户只需要选择很少的选项就可以开始安装,与 Slackware 相比要简单得多。并且在安装到老版本的 Redhat 上时,它可以保留系统中原有的许多设置(例如网络部分和 X Window 部分)让用户不必重复枯燥无味的配置工作。

独特的 RPM 升级方式

RedHat 所有的软件包都是以 RPM(RedHat Package Manager)方式包装的,这种包装方式可以让用户轻松地进行软件升级,彻底卸除(uninstall)应用软件和系统部件,RPM 使用简单,系统核心的升级也只用一行命令就可以轻松完成,而且还会检查程序运行时需要的库是否已经安装,同 Windows 95 上的 InstallShield 相比一点也不逊色。用户安装一遍 RedHat Linux 之后,就再也不用重新安装系统,只需要不断的升级就可以。这使得 RPM 成为 RedHat Linux 最大的卖点。并且 RedHat 公司将其置于 GPL 宣言保护下,因而对于广大的 Linux 用户都可以享用 RPM 方式带来的便利,免去了手工编译之苦。事实上很多 Slackware Linux 的用户都安装了 RPM 软件包来进行软件升级。

丰富的软件包

RedHat 收集的软件包是非常完整和精美的,不仅包括大量的 GNU 和自由软件,还包括了一些优秀的 ShareWare 软件。并都经过 RedHat 公司技术人员的认真调试和配置,让一个普通用户安装完 RedHat Linux 之后立刻就能享受配置完整的 WWW server、Samba 等需要用户花费大量时间和精力去编译安装的东西。

安全性能好

RedHat 缺省配置下的系统安全性能已经非同一般。并提供 Pluggable Authentication Modules (PAM)以加强系统的安全性能和系统管理的扩充性。如果用户计划增加系统的安全性,就要安装更多的安全软件,例如 TCP wrapper 也可以参阅用户指南(User Guide)的说明。并且 RedHat 提供快速的系统安全补丁建议,RedHat 公司有关安全方面的邮递列表(Mailing List)是最权威的 Linux 安全方面的消息来源之一。如果用户对安全方面十分注意的话,RedHat 公司的 Mailing list 是一定要订阅的,另外还可以经常光顾他们的主页,然后使用 RPM 来更新软件或做补丁工作。那么系统安全的维护工作会轻松得多。

方便的系统管理界面

RedHat 提供一套 X Window 下的系统管理软件,让用户可以在图形方式下进行“增加/删除”用户、改变系统设置、安装新软件、安装打印机等等系统管理方面的工作,与 UNIX 下通常采用的字符方式的界面相比要直观和方便得多,与商业 UNIX 提供的 SAM 和 Windows 95 下的控制面板相比也毫不逊色。

详细而完整的在线文档

在 /usr/doc 下面的目录中收录了完整的 HOWTO, LDP, FAQ 系列说明文件,还有 Redhat 独有的长达 250 多页的用户指南,详细说明各种软件安装及系统维护的方式,对于 Linux 初学者来说是非常有益的知识来源。

由此可见,RedHatLinux 与其他两套最常见的 Linux 版本相比,是初学者的最佳选择,对于初次接触 Linux 的用户来说,RedHat 可以让用户很快享受到 Linux 的强大功能,并免去繁琐的安装与设置工作。但是其复杂的目录结构与凌乱的系统配置文件是 RedHat 很大的一个毛病,用户经常为了更改设置而在大量目录之间漫无目的地寻找。据 www.li.org 的统计,目前使用 Redhat Linux 的用户约占 14%。

### 3. Debian Linux(见图 1.6)

通信地址:

P.O. Box 70152

Pt. Richmond CA 94807-0152

E-mail 地址:

info@debian.org

网址:

WWW: <http://www.debian.org/>

FTP: <ftp://ftp.debian.org/debian/>



图 1.6 Debian 的标志

这是由 GNU 发行的 Linux 套件,是完全由网络上的 Linux 爱好者负责维护的发行套件。这些志愿者的目的是制作一个可以同商业操作系统相媲美的免费操作系统。并且其所有的组成部分都是自由软件。Debian Linux 的特点是软件极其丰富、升级容易、软件之间的关联性强,是一个开放式的开发环境。

Debian 是独一无二的在相同的 FSF 精神指引下,由互不相识的世界各地的 Linux 爱好者通过互联网来进行协作开发的 Linux 发行套件,大约有 100 个维护人员工作于 500 个软件包之上,一个久经经验的错误跟踪系统允许用户迅速报告他们发现的臭虫或安全问题,然后由 Debian 的开发人员迅速进行维护。一般说来,很快就会有该软件包的一个新修正版本本放置到常见的 FTP 服务器上,由此可见 Debian 是一个动态的 Linux 发行套件。它每三个月发布一个“Snapshot”版本,其 ftp 服务器是每天更新的。

Debian 采用与 RedHat 的 RPM 方式类似的 deb 方式来进行软件的升级,因而在 Debian Linux 环境下安装软件是一件很轻松的工作,Debian 的所有软件都是自由软件,但是用户也可以获得非自由软件的 Debian 版本。Debian 安全性也是很好的,从其内置 shadow password 系统就可以看出这一点。据 www.li.org 的统计,目前使用 Debian Linux 的用户约占 12%。

对于喜欢安装大量软件的用户和想完全使用自由软件或者对系统要求干净简洁的用户来说,Debian 是一个不错的选择,只是中国用户要获得 Debian Linux 相对困难一些,国内只有少数 FTP 站点收集了 Debian Linux,其中有 [ftp.sepe.ac.cn](http://ftp.sepe.ac.cn)。

#### 4. TurboLinux(见图 1.7)

通信地址:

美国:

Pacific HiTech, Inc. (California)  
Communications Technology Cluster  
2201 Broadway, 2nd Floor  
Oakland, California 94612-1932 USA

中国:

北京拓林思软件有限公司  
北京市西城区金融街通泰大厦 C503 室  
100032

E-mail 地址:

[lonn@turbolinux.com](mailto:lonn@turbolinux.com)  
[info@turbolinux.com.cn](mailto:info@turbolinux.com.cn)

网址:

WWW: <http://www.turbolinux.com.cn/>  
FTP: <ftp://ftp.turbolinux.com.cn/>



图 1.7 TurboLinux 的标志

以 TurboLinux 为主要产品的 Pacific HiTech 公司(PHT)是亚洲市场最大的 Linux 产商。1998 年,TurboLinux 的全球发行量超过 120 万套,在日本占 Linux 市场的 50%,其成长速度是所有 Linux 产商中最快的。PHT 是唯一提供远东语言版本的 Linux 产商,目前同时发行中文版、日文版和英文版,国际化和本地化是其市场侧重点。

PHT 在中国市场上推出的 TurboLinux 中文版,是通过 ZWinPro 这套遵循 GPL 的外挂式中文平台,较好地解决了 Linux 下中文输入的显示和打印问题,并支持中文 TrueType 和中文 Postscript 字库。TurboLinux 软件包采用 RPM 格式,具有较好的兼容性,提供多种预先定制的安装方式,并提供中文字符界面的安装和系统管理程序。

相对于其他发行版本,TurboLinux 目前存在着体积较大和系统管理界面尚未完全图形化的缺点,不过对于远东地区用户来说,TurboLinux 是个不错的选择。

#### 5. S.u.S.E. Linux(见图 1.8)

在 Linux 的发展过程中,除美国的编程人员以外,起到最大贡献的就得算德国人了,连最新的压缩标准 bzip 都是他们写出来的,现在他们做出了自己的发行套件 S.u.S.E. Linux,这个 Linux 套件是在欧洲大陆上使用最广泛的 Linux 套件,其特点是易于安装使用,并且包含有一些其他发行套件不具有的一些软件,例如 S.u.S.E. X server,比 Xfree86 支持更多显示卡,例如最新的 SIS、MediaGX、NeoMagic 与 SaX,并且 S.u.S.E. 是采用新软件最多的一种发行套件,如他们的窗口管理器就是最新的 KDE 1.0,另外在标准的 S.u.S.E.5CD 的版本中还包括 850 个最新的软件,随 CD 的还有 400 页的使用说明书。

其通信地址:

S.u.S.E. GmbH  
Gebhardstrasse 2  
90762 Fuerth  
Germany

Email 地址:

suse@suse.de

网址:

<http://www.suse.de>



图 1.8 Suse 的标志

## 6. Craftworks Linux(见图 1.9)

这是一个基于 Intel 和 AXP 平台的 Linux 商业发行套件,其特点是内置大量的商业软件版本,例如最好的 Linux X 服务器——MetroX, SAM (Security and User Account Management)等,其系统还内置 NIS, NFS 与 shadow password 等,并且还提供商业公司应有的良好的技术支持。对于商业用户也许是一个不错的选择。

通信地址:

Craftwork Solutions, Inc.  
4320 Stevens Creek Blvd. # 170  
San Jose, CA 95129

E-mail 地址:

[info@craftwork.com](mailto:info@craftwork.com)

网址:

WWW: <http://www.craftwork.com>

FTP: <ftp://ftp.craftwork.com/pub/v2.2>(注,只能下载升级部分)



图 1.9 Craftworks 的标志

## 7. Linux Pro(见图 1.10)

这是由 WGS 公司推出的一个发行套件,其显著特点是系统极其稳定并且具有良好的安全性,他们遵循“最新的未必最好”的原则,对软件进行挑选,并且对每种软件都由公司的技术人员进行安全检查和补丁,他们的系统核心版本就是采用最稳定的 1.2.13,当然他们也提供最新版的软件供用户安装。对于要求稳定和安全的用户来说,可以考虑选择这个版本。

通信地址:

WorkGroup Solutions, Inc.  
P.O. Box 460190  
Aurora, CO 80046-0190

E-mail 地址:

[info@wgs.com](mailto:info@wgs.com)

网址:

WWW: <http://www.wgs.com/>

FTP: <ftp://ftp.wgs.com/pub2/wgs>



图 1.10 Linux

Pro 的标志

### 8. Trans-Ameritech Linuxware (见图 1.11)

特点是安装简单,提供 Windows 下的安装界面。对于 UNIX 新手是一个不错的选择。

通信地址:

Trans-Ameritech  
2342A Walsh Avenue  
Santa Clara, CA 95051

E-mail 地址:

info@trans-am.com, order@trans-am.com

网址:

WWW: <http://www.zoem.com/tae>



图 1.11 Linuxware 的标志

### 9. Yggdrasil Plug-and-Play Linux (见图 1.12)

特点是提供一本 Linux 圣经(《Linux Bible》),这是一本 Linux 初学者不可不读的入门读物。

通信地址:

Yggdrasil Computing, Incorporated  
4880 Stevens Creek Blvd., Suite 205  
San Jose, CA 95129-1034

E-mail 地址:

info@yggdrasil.com

网址:

WWW: <http://www.yggdrasil.com>

FTP: <ftp://ftp.yggdrasil.com>



图 1.12 Yggdrasil 的标志

### 10. Caldera (见图 1.13)

该公司就是最后买下微软的 MS-DOS 的唯一竞争对手——DR-DOS 的公司,其实点是 OpenLinux,有趣的是他们将 DR-DOS 改装成全面支持 Internet 的 DOS,这恐怕既是空前也是绝后的一个内嵌 Internet 支持的 DOS 版本了。Caldera Linux 是最早采用图形方式安装的,并且还提供 KDE 作为 Desktop。

通信地址:

Caldera, Inc.  
240 West Center Street  
Orem, Utah 84057  
USA

E-mail 地址:

info@caldera.com

网址:

WWW: <http://www.caldera.com>



图 1.13 Caldera OpenLinux 的标志

## 11. DLINUX

这实际上不是一个标准意义上的发行套件,之所以提出这个版本,是因为它是唯一的一个可以完全安装于 DOS 文件系统上的 Linux 发行套件(采用 UMSDOS 文件系统)。虽然它的体积压缩后只有 8.5M(展开为 22M),但是麻雀虽小五脏俱全,有 X Window, Mosaic, PPP 与 SLIP 等,是想学习 UNIX 又不想放弃 DOS 或 Windows 的用户的一个 Linux 品尝版。其安装方式也是前所未有的简单,只需要将该压缩文件在 DOS 下的任何一个目录下展开就可以了。

网址:

FTP: <ftp://ftp.sunsite.unc.edu/pub/Linux/distributions/dilinux>

其他还有很多对以上发行套件进行重新包装的厂商,如 InfoMagic、OxBow、Cheapbytes 等,这里就不再一一介绍了。

## 1.4 如何获得 Linux 以及 Linux 的网络资源

### 1.4.1 如何获得 Linux

中国用户获得 Linux 的方式主要有通过网络获取、购买 CDROM 或利用软盘。至于国外用户常用的使用直接拨号的 BBS 下载 Linux 对于中国用户并没有什么实际意义,这里不予以详细阐述。

#### 1. 通过网络获取 Linux

这利用 FTP 或者 Ftpmail 先下载整个 Linux 安装版本再安装,或者直接通过 NFS 或 FTP 安装 Linux(注:有些 Linux 的发行套件并不支持 NFS 或 FTP 安装)两种方式。下面给出互联网上提供这些服务的一些站点资源:

FTP:

| 节点名                   | IP 地址         | Linux 目录            |
|-----------------------|---------------|---------------------|
| tsx-11.mit.edu        | 18.172.1.2    | /pub/linux          |
| sunsite.unc.edu       | 152.2.22.81   | /pub/Linux          |
| ftp.turbolinux.com.cn | 210.77.38.126 | /pub (中国)           |
| ftp.buptnet.edu.cn    |               | /pub/slackware (中国) |
| ftp.hangwan.com.cn    |               | /pub                |
| ftp.pku.edu.cn        |               | /pub/slackware (中国) |
| ftp.synet.edu.cn      |               | /pub1/Linux (中国)    |
| ftp.scpc.ac.cn        |               | /pub/linux (中国)     |
| ftp.ihp.ac.cn         |               | /pub/linux (中国)     |
| ftp.funet.fi          | 128.214.248.6 | /pub/OS/Linux       |
| net.tamu.edu          | 128.194.177.1 | /pub/linux          |
| ftp.mcc.ac.uk         | 130.88.203.12 | /pub/linux          |

续表

| 节点名                           | IP地址           | Linux 目录                 |
|-------------------------------|----------------|--------------------------|
| src.doc.ic.ac.uk              | 146.169.2.1    | /packages/linux          |
| fgbl.fgl.mw.tu-muenchen.de    | 129.187.200.1  | /pub/linux               |
| ftp.informatik.tu-muenchen.de | 131.159.0.110  | /pub/comp/os/linux       |
| ftp.dlv.rwth-aachen.de        | 137.226.4.111  | /pub/linux               |
| ftp.informatik.rwth-aachen.de | 137.226.225.3  | /pub/linux               |
| ftp.Germany.EU.net            | 192.76.144.75  | /pub/os/linux            |
| ftp.ibp.fr                    | 132.227.60.2   | /pub/linux               |
| ftp.uu.net                    | 137.39.1.9     | /systems/UNIX/linux      |
| warehouse.wustl.edu           | 128.252.135.4  | /mirrors/linux           |
| ftp.win.tue.nl                | 131.155.70.100 | /pub/linux               |
| ftp.stack.tue.nl              | 131.155.2.71   | /pub/linux               |
| srcwgs.sra.co.jp              | 133.137.4.3    | /pub/os/linux            |
| ftp.denet.dk                  | 129.142.6.74   | /pub/OS/linux            |
| NGTUGGA.edu.tw                | 140.111.1.10   | /Operating-Systems/Linux |
| nie.switch.ch                 | 130.59.1.40    | /mirror/linux            |
| cnuxcc-arch.cnx.it            | 131.114.1.10   | /pub/Linux               |
| ftp.monash.edu.au             | 130.194.11.8   | /pub/linux               |
| ftp.dste.edu.au               | 130.102.181.31 | /pub/linux               |
| ftp.sydntech.usyd.edu.au      | 129.78.192.2   | /pub/linux               |

其中 TSX-11 和 SUNSITE 是互联网上 Linux 资源最丰富的两个站点。

#### FTPmail:

```

sunsite.unc.edu          ftpmail@sunsite.unc.edu
ftp.informatik.tu-muenchen.de  ftp@ftp.informatik.tu-muenchen.de

```

## 2. 购买 Linux 光盘

国内目前出版 Linux 光盘的公司很少,所以无法给用户列出国内出版商的通讯地址,国外出版商的通讯地址可以参见上一节。不过用户须注意的是,光盘提供商分两种,一种是只提供一个 FTP 站点上最新版本 Linux 的镜像,价格低廉,但是不提供任何技术支持,他们的光盘常常包括好几个 Linux 的发行套件。另外一种是公司出版自己的发行套件,用户可选择的余地较小,价格不菲,但是公司提供一定的技术支持,主要是安装方面的技术支持。与本书配套的光盘就包括 Slackware 和 RedHat 的最新发行套件。

至于通过软盘安装 Linux,现在变得不那么重要了,首先是现在光驱在中国已经很普及,其次软盘的故障率也实在是太高了一点。因此笔者并不提倡用户像安装某些商业版本的 UNIX 一样使用几十张软盘来安装 Linux,也许只是对于那些没有光驱或网卡的老式机器才适用。

### 1.4.2 Linux 的网络资源

对于 Linux 的初学者来说,能够获得帮助是一件非常重要的事情,尤其是对没有接触

过 UNIX 的人来说,要掌握 Linux 需要一定时间的学习。但是,用户常常遇到的问题不知道从何处获得帮助。下面笔者就简单介绍一下用户如何在 Linux 下解决遇到的问题。首先,遇到任何问题,不要有任何急躁情绪,把出现问题时的系统配置,运行环境都记下来,这一点对用户接下来寻求帮助是很有好处的。最好再将问题重复产生一遍,仔细研究故障现象,就自己的知识范围猜想一下可能的解决方案,自己尝试着去解决问题,要知道并不是所有问题都是用户以个人的力量无法解决的,试着去解决这些问题对提高用户自己 Linux 应用水平是有很大的帮助,自己解决一个问题之后的感觉是无比美妙的,这也正是 Linux 吸引人的地方。自己摸索出来的经验和方法虽然不是十全十美,但是只要能解决问题就是好的方法。如果自己实在没有办法解决,下一步就是去查看一下手边能找得到的相关资料,如有关 Linux 的文档,另外其实很多问题都能从 UNIX 书籍上找到答案,Linux 就是 UNIX 嘛。在 Linux 系统的 `/usr/doc` 目录下更存放着大量的 FAQ, HOWTO 等相关文档,在这里笔者建议 Linux 的初学者在使用 Linux 系统之前先阅读一遍 Linuxfaq, Infosheet 和那几十篇 Howto, 对用户使用 Linux 会有很大的帮助,防患于未然,即使读不懂也没有关系,可以使用户对于一些问题有个概念性的认识,明白将来遇到问题时可去何处查找资料。一般说来大多数问题都可以在此处找到解决的方案。如果这儿没有能够找到,而用户又有上网的条件,那么就可以到网络上查询相关的文档,使用 AltaVista, Infoseek, Yahoo 等搜索引擎去搜索,这是最容易查询到其相关方面已经存在的文档资料的途径。第二步就可以到与 Linux 相关的新闻组和 BBS(电子公告牌)上去发问,注意的是在发问之前希望大家先查询已经发表过的文章,因为很多问题重复出现过很多次,例如“我的 Linux 不能启动”、“如何输入中文”等问题,这些问题一般都已经有人作了详细的解答,用户如果盲目发表同样的问题,其结果很可能是无人理睬。在发表问题时还需要遵守新闻组和 BBS 上一些约定俗成的规定:不要发表的问题太笼统,使人不知如何回答,引来一大堆反问的信件。例如“我的 Linux 为什么不能开机?”、“我的程序为什么会死机?”等问题,徒然浪费宝贵的网络资源而于事无补,请用户务必要把信件写得清楚了,尽可能地提供有关用户所使用的机器和系统的信息,用户所遇到的问题的详细情况,否则别人很难给用户以恰当的帮助。用户至少应该告知用户的机器型号、硬件配置、用户的 Linux 的版本、使用的软件以及用户是如何运行、结果如何、出现的问题是怎么样的,如果用户已经试着自己解决问题,那么就能把问题说得更清楚。另外,在解决问题之后,如果下次有别的用户询问相同的问题,用户有义务给他们以解答,只有这样,网络上才能真正成为解决问题、交流技术的地方。

下面将给出网络上常见的 Linux 网络资源:

### 1. LDP(Linux Documentation Project)

<http://sunsite.unc.edu/mdw/linux.html>

<ftp://sunsite.unc.edu/pub/linux/doc/LDP/>

这是由 Matt Welsh 先生(《Linux 安装与使用指南》的作者)负责维护的 Linux Documentation Project(Linux 文档计划)的主页和获得 LDP 文档的 FTP 地址, LDP 目前有下列电子书籍供广大的 Linux 用户使用:

《Linux Installation and Getting Started》(《Linux 安装使用指南》)

作者: Matt Welsh.

《Linux Kernel Hacker's Guide》(《Linux 核心黑客手册》)

作者: Michael K. Johnson.

《Linux User's Guide》(《Linux 用户手册》)

作者: Larry Greenfield.

《Network Administrator's Guide》(《网络管理手册》)

作者: Olaf Kirch

《System Administrator's Guide》(《系统管理手册》)

作者: Lars Wirzenius.

这五本书都是重量级的文档,内容极其丰富详尽,非常值得大家一读,对于全面地掌握相关方面的知识是很有好处的。与此同时该主页(<http://sunsite.unc.edu/mdw/linux.html>)上有相当多的 Linux 网络资源和全部的 Linux Howto,FAQ。

## 2. Linux 新闻组

大家可以到 [news.clinux.ml.org](http://news.clinux.ml.org) (中国站点)上阅读下列新闻组,或者利用 [www](http://www.dejanews.com) 到 <http://www.dejanews.com> 去查询和浏览,该站点对于大家查询已经发表的旧文章是一个很不错的地方,比起直接去新闻服务器上去查询要方便和快捷得多,并且新闻服务器上通常不保留太多的旧信,而该站点保留了几几乎所有的信件。在网络新闻组中对中国用户最有意义的当然是中文新闻组,其中的 [cn.bbs.comp.linux](http://cn.bbs.comp.linux) 和 [tw.bbs.comp.linux](http://tw.bbs.comp.linux),这两个新闻组就很不错,不过在 Windows 下阅读 [tw.bbs.comp.linux](http://tw.bbs.comp.linux) 需要安装支持 big5 的中文平台,如 Njwin 和 Richwin 等,使用的 news 读取程序有微软的 Internet NEWS, Netscape News, Agent 等。在 Linux 下可以安装 [extern](http://www.clinux.org), [hztty](http://www.clinux.org) 等软件,读取程序有 [m](http://www.clinux.org), [trn](http://www.clinux.org) 等。具体实现将在后面的章节详细介绍。

如果用户英文水平很好的话,直接阅读英文新闻组也是一个不错的选择,其中不可不读的一个新闻组就是 [comp.os.linux.announce](http://news.clinux.org),该新闻组是由 Lars Wirzenius 先生维护,所有 Linux 的最新进展、新发表的软件及错误修正都首先在这儿发布。要订阅该新闻组,给 [linux-announce@news.ornl.gov](mailto:linux-announce@news.ornl.gov) 发信即可,信的正文写上 [subscribe](mailto:linux-announce@news.ornl.gov) 就可以了。有任何问题可直接与 [wirzeniu@kruuna.helsinki.fi](mailto:wirzeniu@kruuna.helsinki.fi) 联络。而 [comp.os.linux.answers](http://news.clinux.org) 是发表关于 Linux 的文档如 FAQ、Howto 等等的地方。[comp.os.linux.setup](http://news.clinux.org) 是讨论如何安装 Linux,以及安装过程中的问题的地方。[comp.os.linux.admin](http://news.clinux.org) 是讨论 Linux 系统管理的地方。[comp.os.linux.development.system](http://news.clinux.org) 讨论 Linux 核心开发方面的问题。而同 Linux 相关软件的开发可到 [comp.os.linux.development.apps](http://news.clinux.org) 讨论。[comp.os.linux.hardware](http://news.clinux.org) 主要讨论 Linux 硬件方面的问题。[comp.os.linux.networking](http://news.clinux.org) 讨论 Linux 网络方面的问题。[comp.os.linux.x](http://news.clinux.org) 讨论 X Window 使用的问题。如果用户不知道这个问题属于哪个新闻组,可以到 [comp.os.linux.misc](http://news.clinux.org) 这个讨论组来讨论。在发表文章时请注意不要在多个新闻组中交叉发表同一篇文章,因为 Linux 之所以要分成这么多新闻组,其目的就是要为了降低网络的负载,如果用户交叉发表文章,那么分组就没有意义了,并且会引起其他用户的满腔怒火,不仅用户的信箱会被抱怨的信件塞满,用户所在的节点也有可能被列为不受欢迎的节点,从而无法登录有关的新闻服务器。

有关 Linux 的新闻组列表如下:

|                         |              |                                  |
|-------------------------|--------------|----------------------------------|
| cn.bbs.comp.linux       | (大陆国标码)      | linux.debian.policy              |
| tw.bbs.comp.linux       | (台湾 big-5 码) | linux.debian.sparc               |
| linux.redhat.applixware |              | linux.debian.www                 |
| linux.redhat.asp        |              | comp.os.linux                    |
| linux.redhat.devel      |              | comp.os.linux.advocacy           |
| linux.redhat.digest     |              | comp.os.linux.alpha              |
| linux.redhat.install    |              | comp.os.linux.answers            |
| linux.redhat.list       |              | comp.os.linux.development        |
| linux.redhat.misc       |              | comp.os.linux.development.apps   |
| linux.redhat.pam        |              | comp.os.linux.development.system |
| linux.redhat.ppp        |              | comp.os.linux.hardware           |
| linux.redhat.rpm        |              | comp.os.linux.help               |
| linux.dev.kernel        |              | comp.os.linux.m68k               |
| linux.apps.cdwite       |              | comp.os.linux.misc               |
| linux.debian.68k        |              | comp.os.linux.networking         |
| linux.debian.alpha      |              | comp.os.linux.powerpc            |
| linux.debian.bugs       |              | comp.os.linux.questions          |
| linux.debian.changes    |              | comp.os.linux.setup              |
| linux.debian.devel      |              | comp.os.linux.x                  |

### 3. Linux 邮递列表 (Mailing List)

RedHat 公司的邮递列表:

<http://archive.redhat.com/>

applixware-list

asp-list

blinux-list

cde-list

gnome-announce

gtk-list

hurricane-list

linux-alert

linux-security

m68k-list

pam-list

redhat-announce-list

redhat-devel-list

redhat-install-list

redhat-list

redhat-ppp-list

rpm-list

sound-list

sparc-list

CLinux 计划的邮递列表:

服务器: [clinux.mf.org](mailto:clinux.mf.org)

clinux \_ announce

clinux \_ doc

clinux \_ misc

clinux \_ port

clinux \_ setup

linux \_ advocacy

linux \_ security

### 4. BBS 电子公告牌 (中文)

大陆:

清华大学 [bbs.net.tsinghua.edu.cn](mailto:bbs.net.tsinghua.edu.cn)

上海交大 [bbs.sjtu.edu.cn](http://bbs.sjtu.edu.cn)  
暨南大学 [melon.gznet.edu.cn](http://melon.gznet.edu.cn)  
曙光公司 [bbs.ncic.ac.cn](http://bbs.ncic.ac.cn)  
中国科大 [bbs.ustc.edu.cn](http://bbs.ustc.edu.cn)

台湾:

东吴计中 [bbs.scu.edu.tw](http://bbs.scu.edu.tw)  
中央电机 [bbs.ee.ccu.edu.tw](http://bbs.ee.ccu.edu.tw)  
交大资科 [bbs.cis.nctu.edu.tw](http://bbs.cis.nctu.edu.tw) 9001-9009  
清大资科 [bbs.cs.nthu.edu.tw](http://bbs.cs.nthu.edu.tw)  
清大电机 [bbs.ee.nthu.edu.tw](http://bbs.ee.nthu.edu.tw)  
云技计中 [bbs.yuntech.edu.tw](http://bbs.yuntech.edu.tw)  
成大计中 [bbs.ncku.edu.tw](http://bbs.ncku.edu.tw)  
中山计中 [bbs.nsysu.edu.tw](http://bbs.nsysu.edu.tw)  
SOB 电机分站 [freebsd.ee.ntu.edu.tw](http://freebsd.ee.ntu.edu.tw)  
物理学会 [physbls.iam.s.sinica.edu.tw](http://physbls.iam.s.sinica.edu.tw)

## 5. Linux Gopher 服务器(中文)

交大资工 [gopher://linux.csie.nctu.edu.tw/](http://gopher://linux.csie.nctu.edu.tw/)  
交大资科 [gopher://linux.cis.nctu.edu.tw/](http://gopher://linux.cis.nctu.edu.tw/)  
星舰总部 [gopher://140.138.246.113:7011/computer/](http://gopher://140.138.246.113:7011/computer/)

## 6. Linux WWW 资源(中文,英文资源见附录)

对于 Linux 的爱好者来说,下面几个站点是不可不去的:

<http://freshmeat.net> Linux 软件大本营  
<http://slashdot.org> Linux 论坛  
<http://linuxhq.com>

<http://freesoftware.cci.gov.cn> 中国自由软件协会  
<http://mud.263.net.cn/~linux/> 北京 Linux 俱乐部  
<http://www.slug.cx/> 上海 LINUX 用户组  
<http://www.lug.org.cn/> 南京 Linux 俱乐部  
<http://personal.hb.cninfo.net/~seeven/home.htm> 武汉 linux 联盟  
<http://www.stcore.com/linux/> Linux 中坚站  
<http://www.linux.org.tw/> 台湾 Linux 协会[BIG5]  
<http://cle.linux.org.tw/> 台湾 CLE  
<http://home.baoding.cn.net/~linuxman> Linux 服务社提供 CDROM 邮购服务。  
<http://studio.sinet.net.cn> 网络工作室,专门讨论 Linux/UNIX 的站点。  
<http://linux.okstation.com> 台湾一留学生的主页  
<http://www.hkbug.org> 香港 Linux 协会[KN]  
<http://clinux.ml.org/> 中文 Linux 主页  
<http://eldp.turkolinux.com.cn> 中文文档计划主页  
<http://esun01.ihp.ac.cn/> 本书的老主页

## 7. 杂志

### 《Linux Journal》

这是四年前开始发行的一本 Linux 杂志,它的内容包括针对各种水平的用户,订阅费用是 32 美元,地址是:

Linux Journal, PO Box 85867,

Seattle, WA 98145-1867 USA

Email: subs@ssc.com

fax: (U.S.) 1-206-782-7191

## 8. 书籍

### 中文:

- |                               |         |
|-------------------------------|---------|
| 1.《即学即用 Linux》                | 电子工业出版社 |
| 2.《Linux 的 Internet 站点建立与维护》  | 清华大学出版社 |
| 3.《微机上的 UNIX 操作系统 Linux 使用指南》 | 清华大学出版社 |
| 4.《用 GNU 软件编程》                | 电子工业出版社 |

### 台湾出版的繁体版本:

- |                                       |          |
|---------------------------------------|----------|
| 1.《巡航 Linux 领域》                       | 和硕出版社    |
| 2.《漫步 Linux 世界》                       | 和硕出版社    |
| 3.《遨游 Linux 天地》                       | 和硕出版社    |
| 4.《攀登 Linux 高峰》                       | 和硕出版社    |
| 5.《Linux 的奥秘》                         | 松格出版社    |
| 6.《Linux 网路管理》                        | 松格出版社    |
| 7.《轻轻松松安装 Linux/UNIX》                 | 波全出版社    |
| 8.《Linux 安装与使用手册》                     | 第三波出版社   |
| 9.《Linux 的世界》                         | 第三波出版社   |
| 10.《Linux 中文应用手册》                     | 第三波出版社   |
| 11.《Linux 管理与操作手册》                    | 第三波出版社   |
| 12.《Linux 轻松入门》                       | 第三波出版社   |
| 13.《建构 Linux Web Server》              | 博硕出版社    |
| 14.《Internet Server DIY 架设 Web Serve》 | 资讯与电脑出版社 |
| 15.《Internet Server 建置深度指导手册》         | 资讯与电脑出版社 |
| 16.《LINUX 完全手册》                       |          |
| 17.《我真的用 Linux 架过商业网站》                | 旗标出版社    |
| 18.《UNIX 系统管理》                        | 松格出版社    |
| 19.《最新 X WINDOW 实用手册》                 | 第三波出版社   |
| 20.《自己动手装网路 LINUX》                    |          |
| 21.《Linux 初学与管理》                      | 松岗出版社    |

### 有关 Linux 的英文书籍有:

- |      |                                      |
|------|--------------------------------------|
| 书名:  | 《Learning the UNIX Operating System》 |
| 作者:  | Grace Todino & John Strang           |
| 出版商: | O'Reilly and Associates, 1987        |

ISBN: 0-937175-16-1, \$ 9.00

学习 Linux 的一本好书,如果用户是新手的话可以阅读本书。

书名: 《Learning the vi Editor》

作者: Linda Lamb

出版商: O'Reilly and Associates, 1990

ISBN: 0-937175-67-6, \$ 21.95

UNIX 编辑器 Vi 的使用说明。

书名: 《Essential System Administration》

作者: AEleen Frisch

出版商: O'Reilly and Associates, 1991

ISBN: 0-937175-80-3, \$ 29.95

讲解了如何管理一个 UNIX 系统的基本要点,非常全面。

书名: 《TCP/IP Network Administration》

作者: Craig Hunt

出版商: O'Reilly and Associates, 1990

ISBN: 0-937175-82-X, \$ 24.95

针对 Linux 深入剖析了如何设置并运行一个 TCP/IP 网络,同 Linux NET-2-HOWTO 和 Linux Network Administrator's Guide 相比,本书讲解了更多的技术细节。

书名: 《Managing UUCP and Usenet》

作者: Tim O'Reilly and Grace Todino

出版商: O'Reilly and Associates, 1991

ISBN: 0-937175-93-5, \$ 24.95

如何安装并设置 UUCP 软件? 如何读取 USENET 导航的新闻组? 如果用户对此有兴趣的话就可以阅读本书。

书名: 《The X Window System: A User's Guide》

作者: Niall Mansfield

出版商: Addison-Wesley

ISBN: 0-201-51341-2

一本详细的关于如何使用 X Window 的教程和参考手册。

书名: 《The C Programming Language》

作者: Brian Kernighan and Dennis Ritchie

出版商: Prentice-Hall, 1988

ISBN: 0-13-110362-8

如何在 UNIX 下使用 C 编程,这本书相信大家再也熟悉不过了。英文书常用作者的姓氏加以称呼,由 C 语言的设计者 Brian W Kernighan 和 Dennis Ritchie 写的这本书通常被称作 K&R。这本书被视为 C 语言的圣经,所以大家也称它“C Bible”。这本书不但是教 C 语言的课本,也是 C 语言的定义,但也因此这本书并不好读。当作课本很好,对于其他书籍讲解不详细的部份可以在这里找到明确的定义。

书名: 《The UNIX Programming Environment》

作者: Brian Kernighan and Bob Pike

出版商: Prentice-Hall, 1984

ISBN: 0-13-937681-X

讲述如何在 UNIX 下具体编程,包括所有常用的编程工具的使用,是要在 UNIX 下编程的使用者的最好的入门教材。

书名: 《Advanced Programming in the UNIX Environment》

作者: W. Richard Stevens

出版商: Addison-Wesley

ISBN: 0-201-56317-7

讲解在 UNIX 下编程的一些系统级的知识,如文件 I/O、进程控制或进程间通信等,本书还讲述了 UNIX 标准,包括 POSIX.1 标准。

书名: 《The Design of the UNIX Operating System》

作者: Maurice J. Bach

出版商: Prentice-Hall, 1986

ISBN: 0-13-201799-7

本书讲述了 UNIX 核心的工作原理,虽然不是针对 Linux 的,但是如果用户对 Linux 核心要有所了解的话,这本书就是用户开始的地方。

书名: 《The Magic Garden Explained》

作者: Benny Goodheart and James Cox

出版商: Prentice-Hall, 1994

ISBN: 0-13-098138-9

详细讲解 SVR4 核心的实现,比上本书的技术细节更多,而且更新。

## 1.5 Linux 在中国的前景及 CLinux 计划

近年来,我国计算机技术发展迅速,应用水平不断提高,随着计算机的进一步普及并进入大多数的普通家庭,广大的用户群对于实用软件的需求也进一步加大,而目前我国商业软件开发力量尚略显薄弱了一点。随着中国复关之日的来临,在人们知识产权保护意识越来越强的今天,相信不久的将来盗版软件在中国的市场会愈来愈小,缺乏应用软件这个迫在眉睫的矛盾会愈来愈明显。解决的办法:一是降低正版软件的价格,普及推广应用,但是中国计算机用户的经济水平决定了人们不可能使用太多的正版软件,试问如果没有盗版软件,有几个人能用得起 Visual C++、MSOffice 这些重量级的软件(但是使用 Linux 作为自己的 PC 上的操作系统,笔者就可以毫不羞涩地说使用的都是正版软件,连本书都是用自由软件 Emacs 输入的。)?并且用户对国内某些软件开发商的定价策略也颇有微词,但却无力改变;二是大力发展中国的自由软件事业,发展自由软件既能锻炼软件开发队伍;又为社会提供了有价值的应用软件,这对于广大的中国用户来说无疑是一件造福于民的事业。而 Linux 正是一个绝佳的自由软件的操作平台。在 Linux 平台上,使用网络、科学计算和工业控制都不会逊色于任何一套商业操作系统,用户可以在其他商业操作系统上能做的任何事情,而且最重要的是 Linux 上的软件都是正版软件。相信随着 Linux 在中国的普及,越来越多的用户不仅仅将 Linux 当作一个玩具。而会将 Linux 应用于更为广泛的用途上。但是自由软件和 Linux 的推广应用及发展需要良好的外部环境和社会大众的理解和支持,希望国内相关的计算机厂商和有识之士能够提供一些有益的赞助,为信

息产业在中国的健康发展打下一个坚实的基础。

中国目前已经出现了 Linux 的第一批用户,他们主要由各大科研院所的科研人员和大学里的学生们组成。在使用过程中他们也深刻体会到交换信息的重要性。一开始,他们在各地的电子公告牌上彼此交流使用心得,但是随着时间的流逝,电子公告牌的天地已经不够用了,为了更好地交流信息,为了在中国推广 Linux 的使用,他们成立了 CLinux 计划,起初大家在清华大学的电子公告牌上讨论该项议题,然后由网友 Homer 建立了 CLinux 的主服务器:clinux.ml.org,并且建立了中国 Linux 用户自己的邮递列表。该邮递列表的使用方法是:发一封信到 majordomo@clinux.ml.org,在信的正文写 help(注意不是在标题栏),用户就会收到一封关于该邮递列表使用方法的介绍。用户可以自由加入和退出个邮递列表的讨论。如果要订阅这些邮递列表,可以发封信给 majordomo@clinux.ml.org,在信的正文写:subscribe xxxxxx,这里 xxxxxx 指的是任何一个邮递列表,如 clinux\_doc。用户如果想要参加相关邮递列表的讨论,可以用 K-MAIL 把讨论的内容发给相应的邮递列表,如 clinux\_doc@clinux.ml.org。邮递列表的订阅和管理是自动进行的。现在,已经开设了以下邮递列表,并且可以应大家的要求随时调整:

|                 |              |
|-----------------|--------------|
| clinux_announce | CLinux 计划的公告 |
| clinux_doc      | CLinux 计划的文档 |
| clinux_misc     | CLinux 计划的杂项 |
| clinux_port     | 英文软件向中文软件的移植 |
| clinux_setup    | Linux 的安装    |
| linux_security  | Linux 安全问题   |

本书的作者也建立了一个 CLinux 的非正式 WWW 站点供大家使用,WWW 地址是 <http://cldp.turbolinux.com.cn> 上面收集了一些 Linux 的中文文档。另外本书的 WWW 站点也在这个服务器上,用户可以从这上面下载补丁程序。

## 1.6 常见问题解答

### 1. Linux 的标准发音?

<ftp://mud.263.net/~linux/linuxabc/linux.wav>,这是 Linus 本人的 Linux 发音的声音文件,应该是最权威的吧。另外有一个 UNIX 标准发音表:

<ftp://ftp.wg.omron.co.jp/pub/UNIX-faq/docs/Pronunciation-Guide>

### 2. 什么是 Jargon file?

由 Eric S. Raymond [eric@snark.thyrsus.com](mailto:eric@snark.thyrsus.com) 所维护的 Jargon file 是 UNIX 下的名词解释表(又称 hacker 字典),是 UNIX 新手必读的一本小册子。它的网址是:

<http://www.ccil.org/jargon>

这是 html 版,目前最新的是 4.10 版。

### 3. Linus 先生目前在什么地方? 他的个人情况?

Linus 本人已经结婚,刚刚生了一个女儿。他业余爱好阅读和斯诺克台球,研究生毕业后,目前在美國的 Santa Clara, CA, 他的一个朋友的公司中工作,工作可能是进行 SMP 方

面的研究。不过仍然领导 Linux 的开发工作。

Linus 的 E-mail 地址: [Linus.Torvalds@Helsinki.FI](mailto:Linus.Torvalds@Helsinki.FI)

主页: <http://www.cs.helsinki.fi/~torvalds/>

#### 4. 在何处登记为 Linux 用户?

<http://www.li.org>

目前中国用户登录的只有不到五百人, 希望有条件的网友多多努力。

# 2 Linux 的安装

## 2.1 如何取得 Linux

如何取得 Linux,这一点在上一章已经介绍过,这里就不再详细讲述,只是简单介绍一下如何通过网络获得 Linux 的手段:用户可以通过 FTP 取得 Linux 的最新版本,如果要安装 Slackware,请用户使用 Windows 下的 FTP 软件如 Wsftp、Cuteftp,将整个 Slackware 目录直接下载到 DOS 或 Windows 95 的分区上,就可以使用硬盘来安装 Linux。注意该分区不能是被压缩的分区(如微软的 DoubleSpace、DriveSpace、Stacker 等等)。而 Redhat 和 TurboLinux 最好不要直接下载到 DOS 或 Windows 95 的分区上,因为 Slackware 的目录下存放的是 8.3 格式的文件名,而 Redhat 和 TurboLinux 的安装文件是长文件名。下载后会变成短文件名,虽然不影响安装,但是用户得手工记录文件的对应关系,总而言之不太方便。

## 2.2 Linux 安装的硬件要求

首先让我们来讨论一下 Linux 对硬件的需求,即在什么样的机器上可以安装并运行 Linux? 实际上读者只需要回答这样一个简单的问题:用户使用的是一台 IBM PC 兼容机吗? 如果回答是的话,那么恭喜用户! 用户的机器可以安装并运行 Linux! 当然下面我们将详细讲解 Linux 支持的硬件,以便用户挑选和配置自己的 Linux 服务器,或者对照一下,看看自己的机器是否能安装 Linux。

由于 Linux 是一个完全由用户自己开发的操作系统,这就意味着 Linux 支持的硬件种类取决于 Linux 用户所能找到的相应硬件种类,实际上由于 Linux 用户群的庞大, Linux 支持的硬件数目是很广泛的。它支持绝大多数 80x86 平台设计的外部设备。但是对一些比较冷僻和“贵族化”的设备的支持可能不是很好,但随着时间的推移,使用 Linux 的用户数目会急剧增加, Linux 支持的硬件种类也会越来越多。

此外,由于 Linux 必须公开全部源代码,而很多生产硬件的厂家将自己产品的接口私有化或申请相应的专利保护,因此 Linux 的开发人员就不能为这种硬件开发驱动程序,因为如果他编写这样的程序,其程序所有权是属于该公司的,结果就违反了 GNU 的 GPL 版权宣言,此外因为众多的硬件厂商不愿意向自由程序员提供相关的硬件细节,结果 Linux 的程序员们只有自己独自摸索着去编写相应的驱动程序。这种情况现在已经得到了显著的改善,硬件厂商开始向 Linux 程序员提供技术资料,并派出技术人员协助 Linux 开发人员开发硬件驱动程序。主要是因为使用 Linux 的用户群的急剧扩大形成了一个新的硬件市场,失去这个市场对任何厂家都不是什么小的事情。

实际上选择一台能运行 Linux 的 PC 机,首要注意的事项就是用户要拿 Linux 做什么,即由 Linux 的用途来选择需要购买的硬件。如果对于 ISP 来说,使用 Linux 可能就是为了要建立 WWW, Mail, FTP, BBS 等网络服务器,他们一般对系统的内存、硬盘、网卡的要求要相对高一些,而不必太在意显示卡、声卡、CDROM 驱动器等设备,而一般个人电脑玩家就比较注意这些多媒体设备,以及光盘刻录机、调制解调器、活动硬盘、ZIP 驱动器等千奇百怪的硬件配备,以便自己可以在 Linux 下尽情地享受各种各样的声光效果和其他方便。由此可见在购买运行 Linux 的 PC 机前,针对自己使用 Linux 的主要目的,参考本文所提供的一些设备列表选择相应的硬件配备的确是一件很有必要的工作,不然等盲目购买回来的机器,出现一些设备不支持,或者运行速度无法忍受等等问题,那就为时已晚了。对此笔者有惨痛的教训,笔者曾经为配置一个基于 Linux 的局域网去购买网卡,由于事先未详细调查,结果购买了 Intel eepro 10 这个牌子的网卡,结果在 Linux 下化了很多时间也找不到相应的驱动程序,在网络上查询半天,联络到为 Linux 开发 Intel 系列网卡的那位仁兄,得到的答复是现在还不支持,结果只有重新更换网卡。(目前随着 Kernel 2.2 的出现,已经支持这种网卡)希望读者能仔细阅读下面的内容,减少安装 Linux 遇到困难的机率。

下文将分别就硬件的类型来介绍 Linux 的硬件需求,在阅读完下面的内容后相信用户已经有了一个大概的概念,如果仍有疑问,可以去阅读 Linux 自己附带的硬件 HOWTO——《Linux Hardware Compatibility HOWTO》,该文件是随时更新的,用户可以从其中得到 Linux 支持的最新硬件列表。笔者也是主要根据该文件向读者推荐比较适合的设备。

### 2.2.1 Linux 支持的计算机/主板/CPU/内存

Linux 支持的总线有 ISA, VLB, EISA 以及 PCI 等等。在标准的 Linux 核心中并不支持 PS/2 和微通道(MCA)这两种总线,但是可以找到支持这两种总线的 Alpha 版系统核心(<ftp://invaders.deri.nd.edu/pub/misc/>),不过并不推荐使用,对于中国用户来说,一般现在市面上见不到使用这两种总线的计算机,所以也不必考虑这个问题。一般市面上所售的 Pentium, Pentium Pro, Pentium MMX, Pentium II 系列微机都是 PCI 总线,而 486 和 386 系列微机的一部分是 ISA, EISA 和 VESA 等总线, Linux 对这些总线都支持得非常好,用户对此不用担心。

Linux 支持的 CPU 可谓十分广泛,从最老的 386 到最快最新的 Pentium II 都支持, Intel/AMD/Cyrix 出品的 x86 系列 CPU: 386SX/DX/SL/DXL/SLC, 486SX/DX/SL/SX2/DX2/DX4, Pentium, Pentium MMX, Pentium Pro, Pentium II, K6, M2, Cyrix 6x86 等,选择什么样的 CPU 当然视用户的经济状况和系统性能的要求而定,选择越快的 CPU,系统的整体效能自然就表现的优异。比如同样配置的一台 Linux 服务器在使用 Pentium 133 CPU 时,编译一遍 Glibc2 约需要 2.5 个小时左右,而更换为 Pentium Pro200 CPU 后只需要 1.5 个小时。如果用户对自己机器的速度感兴趣,那么在安装完 Linux 之后,可以查看一下 Linux 启动时的一个小信息 BogoMips,来确定用户的 CPU 的运行速度到底有多快。

Linux 启动的信息都是不断滚动的,用户可以在它停止滚动之后使用 Shift + PgUp 来翻阅前面的信息,或登录后用 `dmesg` 来查看,或用 `cat /proc/cpuinfo` 来查看。BogoMips 的输出就在前面几行,该参数被用来粗略估计系统 CPU 运行速度的。例如下面的这个表格就

是各种系统的 BogoMips:

80x86 系列:

|                       |        |                                         |        |
|-----------------------|--------|-----------------------------------------|--------|
| 8086                  | 0.5    | 6x86/133 Cyrix P166 +                   | 133.73 |
| 386SX/20              | 2.45   | K5/100 AMD                              | 200.32 |
| 386DX/40              | 8.06   | Pentium Pro/200                         | 197.42 |
| Nx586/90 NexGen       | 67.44  | Pentium MMX 200                         | 400.23 |
| 486DX/33              | 16.61  | (该值不太准确,因为 BogoMips 使用的测试程序在 MMX 中被优化了) |        |
| 486DX2/66             | 33.55  |                                         |        |
| Pentium/75 Intel      | 29.79  | SMP Pentium Pro/200                     | 398.14 |
| Pentium/133           | 53.25  | (使用双 CPU)                               |        |
| 5x86/133 Cyrix P166 + | 132.88 |                                         |        |

其他硬件平台:

|                   |        |                         |        |
|-------------------|--------|-------------------------|--------|
| ALPHA:21164/333   | 331.35 | HP-PA 9000-720          | 48.00  |
| PowerPC 601/60    | 59.38  | Sparc Sun USL/170       | 330    |
| Mips R4600/133    | 133.12 | Alpha server 1000-4/200 | 397.68 |
| Sun Sparcstation2 | 38.0   |                         |        |

看了这个表格,用户对自己的计算机的信心是不是又增强了?一台一流配置的 PC 同任何高级 UNIX 工作站相比一点也不逊色!但是价格的差距可实在令大多数用户心颤。

Linux 也支持 8086 和 80286,只是必须使用 ELKS(Embeddable Linux Kernel Subset)内核,但性能不敢恭维。另外注意的是对于 Cyrix 和 AMD 的质优价廉的 CPU 的支持虽然没有问题,但是值得用户注意的是有报告说,在少数配置下会出现系统死机的情况,因为 Cyrix 和 AMD 同 Intel 并不是完全兼容的,且其 CPU 的工作温度比 Intel CPU 要相对高一点。如果用户不是经验丰富的玩家,还是选择 Intel 的 CPU 比较保险,不管怎样,如果用户选择了 Cyrix 的 CPU,那么下面这个网址也许对用户有用,是用来打开 Cyrix 的缓存(cache)的:

<ftp://sunsite.unc.edu/pub/Linux/kernel/patches/linux.cpatch>

如果用户要选择多 CPU 的主板,那么恭喜用户,Linux 对多 CPU 的支持要比 NT 好得多。

Kernel 2.2 希望用户仔细阅读 <http://www.linux.org.uk/SMP/title.html> 这个网址。(另外用户还得重新编译 kernel,将 SMP=1 这一行打开。)

一般来说 Linux 能很好地支持市面上所售的大多数主板,Linux 支持最新的 TX、LX 和 BX 主板,支持使用双 CPU 的主板。主板是电脑中最重要的部分,它负责 CPU 和周边设备之间所有数据的流通,并为这些所有的 CPU、二级缓存、芯片组、BIOS、内存、I/O 芯片、键盘口、串口、并口、硬盘控制器、ISA 和 PCI 插槽提供安营扎寨的地方。下面笔者就自己的经验给出选择主机板的建议。

在选购 Pentium 主板之前,用户要做的第一个决定就是用户打算使用哪一种类型的芯片组(chipset);第二件事是选择什么样的制造商(manufacturer)。建议用户选择有品牌的主机板,最好该主机板的生产商还拥有自己的主页,因为这样用户就可以很方便地下载最新的

Flash BIOS 升级版本、驱动程序(drivers)和用户所需要的信息。用户应该先寻找拥有最高性能的主板,但是也应该注意主板的一些特性如 softmenu 等。目前,市面上最受欢迎的主板制造商分别是华硕、微星、技嘉、联想、大众、海洋和升技等,不要选择市面上销量非常小的主机板,这样如果出现问题,其他人也很难帮助用户解决问题。选择主板应考虑下面几点:

- (1)支持用户的 CPU 吗? 可别购买一块 T24 主板去运行 PII CPU。
- (2)支持兼容厂商的 CPU 吗? 如 K6、Cyrix M2 等等。
- (3)拥有一个好的变压器,包括很好的冷却功能并提供稳定的电流吗? 可以自行改变 CPU 电压(Standard/VRE/MMX)吗? 提供多少种 CPU 电压?
- (4)提供多少 PCI 和 ISA 的插槽? 这对系统的扩充很重要,一般显示卡和声卡网卡都要占用一个插槽。
- (5)有多少 SIMM 槽? 这对用户扩充内存很重要,用户如果要插 SDRAM 的话,还要有相应的 SDRAM 插槽。支持多少种 RAM(FPRAM,EDO RAM,SDRAM)? 假如是 430HX 的板子,是否有 11bit tag RAM 以支持容量高达 512 MB 的内存?
- (6)需要 ATX 的主板吗? 主板上的配置会不会挡到较长的插卡?
- (7)缺省是什么样的高速缓存? 笔者强烈推荐使用 512K 同步触发缓存(Pipelined Burst Cache)。

下列笔者给出一个仅供参考的推荐表:

• 最佳的 Pentium 430HX 主板

- (1)华硕(Asus) P/I-P55T2P4
- (2)升技(Abit) IT5H
- (3)微星(Micro Star Int) MS-5128
- (4)Supermicro P55T2S and P5STE
- (5)浩鑫(Shuttle/Spacewalker) HOT 553
- (6)技嘉(Gigabyte) GA-586HX and GA-586DX
- (7)松捷(Soyo) 5TF2
- (8)大众(FIC) PT2200

• 最佳的 Cyrix/IBM 6x86 主板

- (1)升技(Abit) IT5V or Abit IT5H
- (2)华硕(Asus) P/I-P55TVP4
- (3)华硕(Asus) P/I-P55T2P4
- (4)松捷(Soyo) SY-5VA2
- (5)技嘉(GigaByte) GA-586HX
- (6)捷邦(Jet Systems Jet Board) J-656 VXB

• 常见的 MMX 主板

- (1)微星 5148
- (2)华硕 TXP4
- (3)大众 2007

• 常见的 PII 主板

- (1)微星 MS6111/6112
- (2)华硕 P2L97

• 最佳的 Pentium 430VX 主板

- (1)升技(Abit) IT5V
- (2)华硕(Asus) P/I-P55TVP4
- (3)浩鑫(Shuttle/Spacewalker) HOT 557
- (4)PINE PJ7502
- (5)升技(ABIT) AB-PR5
- (6)技嘉(Gigabyte) GA-586ATV
- (7)松捷(Soyo) SY-5VA2
- (8)浩鑫(Shuttle/Spacewalker) HOT 555

• 最好的 Pentium Pro 主板

- (1)SuperMicro P6DOF
- (2)华硕(Asus) P/I-P6RP4
- (3)Tyan 1668 Titan Pro ATX
- (4)AMI Merlin
- (5)EPoX PP6-NB
- (6)Intel VS440FX (Venus)
- (7)华硕(Asus) P/I-P6NPS

• 常见的 K6 主板

- (1)大众 PA2007/2011(VP2)
- (2)EPOX P55-VP3
- (3)MYCOM PAISVG

• 常见的 Cyrix M2 主板

- (1)大众 PA2007/2011(VP2)

下面是这些主板厂商的主页：

|                                                                                                                     |                           |
|---------------------------------------------------------------------------------------------------------------------|---------------------------|
| <a href="http://www.megatrends.com/Motherboards/Mother.html">http://www.megatrends.com/Motherboards/Mother.html</a> | American Megatrends (AMI) |
| <a href="http://www.abit.com.tw/">http://www.abit.com.tw/</a>                                                       | 升技 ABIT                   |
| <a href="http://www.airwebs.com/">http://www.airwebs.com/</a>                                                       | AIR                       |
| <a href="http://www.dellnet.com/users/amptrom">http://www.dellnet.com/users/amptrom</a>                             | Amptrom                   |
| <a href="http://www.sopen.com.tw/">http://www.sopen.com.tw/</a>                                                     | A-Open                    |
| <a href="http://asustek.asus.com.tw">http://asustek.asus.com.tw</a>                                                 | 华硕 Asus                   |
| <a href="http://www.dataexpert.com">http://www.dataexpert.com</a>                                                   | DataExpert                |
| <a href="http://www.dfiusa.com/">http://www.dfiusa.com/</a>                                                         | DFI                       |
| <a href="http://www.ecsusa.com/ecsprods.html">http://www.ecsusa.com/ecsprods.html</a>                               | ECS                       |
| <a href="http://www.epox.com/">http://www.epox.com/</a>                                                             | EPoX                      |
| <a href="http://www.fic.com.tw/">http://www.fic.com.tw/</a>                                                         | 大众 FIC                    |
| <a href="http://www.fretech.com">http://www.fretech.com</a>                                                         | Free Tech                 |
| <a href="http://www.genosys.com">http://www.genosys.com</a>                                                         | Genoa                     |
| <a href="http://www.giga-byte.com/gbt/">http://www.giga-byte.com/gbt/</a>                                           | 技嘉 GigaByte               |
| <a href="http://www.intel.com/design/motherbd/">http://www.intel.com/design/motherbd/</a>                           | Intel                     |
| <a href="http://www.iwill.com.tw">http://www.iwill.com.tw</a>                                                       | 艾威 iwill                  |
| <a href="http://www.jbond.com">http://www.jbond.com</a>                                                             | 捷邦 J bond                 |
| <a href="http://www.j-mark.com">http://www.j-mark.com</a>                                                           | J Mark                    |
| <a href="http://www.micronics.com">http://www.micronics.com</a>                                                     | Micronics                 |
| <a href="http://www.msi.com.tw">http://www.msi.com.tw</a>                                                           | 微星 Microstar              |
| <a href="http://www.microway.com/">http://www.microway.com/</a>                                                     | Microway                  |
| <a href="http://www.mitac.com.tw/mboard.html">http://www.mitac.com.tw/mboard.html</a>                               | 神通 (Mitac)                |
| <a href="http://www.mtiusa.com/">http://www.mtiusa.com/</a>                                                         | M-Technology              |
| <a href="http://www.ocean-usa.com/ocean/">http://www.ocean-usa.com/ocean/</a>                                       | Ocean                     |
| <a href="http://www.pinegroup.com/">http://www.pinegroup.com/</a>                                                   | PINE                      |
| <a href="http://www.shuttlegroup.com/">http://www.shuttlegroup.com/</a>                                             | 浩鑫 Shuttle                |
| <a href="http://www.soyo.com">http://www.soyo.com</a>                                                               | 松捷 Soyo                   |
| <a href="http://www.supermicro.com">http://www.supermicro.com</a>                                                   | SuperMicro                |
| <a href="http://www.tyan.com">http://www.tyan.com</a>                                                               | Tyan                      |
| <a href="http://www.vextrec.com">http://www.vextrec.com</a>                                                         | Vextrec                   |

在 Linux 下,内存比 CPU 还要重要,例如 486DX4-100 + 64M 内存和 P133 + 8M 内存的两台 Linux 服务器,运行速度肯定是那台 486 快得多。Linux 可以运行的最小内存是 2M,如果用户不使用 X Window 系统,那么 Linux 只需要 4MB 的内存就可以运行,如果要运行 X Window 系统则至少需要 8 MB 内存,如果同时使用 Linux 的用户不止一个,或用户要同时执行好几个大型软件(如编译程序),那么用户至少需要 16M 内存。实际上 Linux 在内存不够时会使用交换区(swap disk)来作为虚拟内存,但是其运行速度和大量的 CPU 资源就浪费在磁盘的读取上了。笔者的推荐是,一般用户至少 16M 内存,用来提供网络服务的 Linux 服务器至少要 32M 内存,推荐 64M 内存。至于内存的种类,只要主板支持, Linux 就支持,不管是 FPM,EDO 还是 SDRAM 内存,至于内存的速度当然是越快越好了。

Linux 支持大多数笔记本式计算机,用句时髦的话来说就是支持移动计算, Linux 支持的笔记本型计算机有:

---

|                               |                                                                                                                                               |
|-------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| Compaq Concerto               | <a href="http://www.cs.nmsu.edu/~pfeiffer/">http://www.cs.nmsu.edu/~pfeiffer/</a>                                                             |
| Compaq Contura Aero           | <a href="http://domen.ninnett.no/~hta/linux/aero-faq.html">http://domen.ninnett.no/~hta/linux/aero-faq.html</a>                               |
| IBM ThinkPad                  | <a href="http://peipa.essex.ac.uk/tp-linux/tp-linux.html">http://peipa.essex.ac.uk/tp-linux/tp-linux.html</a>                                 |
| NEC Versa M and P             | <a href="http://www.santafe.edu:80/~nelson/versa-linux/">http://www.santafe.edu:80/~nelson/versa-linux/</a>                                   |
| Tadpole P1000                 | <a href="http://www.tadpole.com/Support/online/linux.html">http://www.tadpole.com/Support/online/linux.html</a>                               |
| TI TravelMate 4000M           | <a href="ftp://ftp.biomath.jussieu.fr/pub/linux/TM4000M-mini-HOWTO.txt.Z">ftp://ftp.biomath.jussieu.fr/pub/linux/TM4000M-mini-HOWTO.txt.Z</a> |
| TI TravelMate 5100            | <a href="http://www.wri.com/~cwikla/ti5100.html">http://www.wri.com/~cwikla/ti5100.html</a>                                                   |
| Toshiba Satellite Pro 4000CDT | <a href="http://terra.mpikg-teltow.mpg.de/~burger/T400CDT-Linux.html">http://terra.mpikg-teltow.mpg.de/~burger/T400CDT-Linux.html</a>         |

如果用户希望了解如何在笔记本上配置 X Window, 可以查阅 <http://www.castle.net/~darin/> 上的文件, 若使用高级电源管理 (APM) 就去 <ftp://ftp.cs.unc.edu/pub/users/faith/linux/> 目录下, 里面有相关的驱动程序可供用户下载。其他同笔记本有关的文档和资料用户可以在 <http://www.cs.utexas.edu/users/kharker/linux-laptop/> 上找到。

### 2.2.2 Linux 对 PCMCIA 的支持

Linux 上的 PCMCIA 驱动程序支持大多数常见的 PCMCIA 控制器: Databook TCIC/2、Intel i82365SL、Cirrus PD67xx 和 Vadem VG-468 chipsets。Linux 上的 PCMCIA 配置信息可以在 <http://hyper.stanford.edu/~dhinds/pcmcia/> 上找到。

### 2.2.3 Linux 支持的显示卡和显示器

如果用户要使用 X Window 的话, 就需要仔细挑选一块性能优异的显示卡。Linux 能够在所有显示卡的文本模式下工作, 也可在所有标准 VGA 卡上以 VGA 模式工作。但这不是用户要注意的重点, 因为 VGA 模式实在是太简陋和单调了, 完全不能发挥 X Window 的强大威力。Linux 支持视频加速卡如 ATI Mach 系列、ET4000 系列和 S3 系列, 它们工作起来比非加速卡要快得多。Linux 支持 8bit, 16bit, 24bit, 32bit 色彩模式, 因此用户在 Linux 下一样可以使用千变万化的色彩。Linux 下支持的显示卡分两种, 一部分是 SVGA 支持的显示卡, SVGA 是用来控制在控制台上运行图形模式的应用程序, 另外一部分是 XFree86 支持的显示卡, 而 XFree86 是用来控制 X Window 的, 两种类型的显示卡略有差别。SVGA 支持的显示卡有:

- 标准 VGA 卡
- 标准 EGA 卡
- ARK Logic ARK1000PV/2000PV
- ATI VGA 系列
- ATI Mach32, Mach64
- Cirrus 542x, 543x 系列
- OAK OTL-037/67/77/87
- S3 系列
- Trident TVGA8900/9000 系列
- Tseng ET3000/ET4000/W32 系列

Xfree86 3.3.3.1 支持的显示卡有：

Ark Logic

ARK1000PV, ARK1000VL, ARK2000PV, ARK2000MT

Alliance

AP6422, AT24

ATI

18800, 18800-1, 28800-2, 28800-4, 28800-5, 28800-6, 68800-3,  
68800-6, 68800AX, 68800LX, 88800GX-C, 88800GX-D, 88800GX-E,  
88800GX-F, 88800CX, 264CT, 264ET, 264VT, 264GT, 264VT-B, 264VT3,  
264GT-B, 264GT3 (this list includes the Mach8, Mach32, Mach64, 3D  
Rage, 3D Rage II and 3D Rage Pro)

Avance Logic

ALG2101, ALG2228, ALG2301, ALG2302, ALG2308, ALG2401

Chips & Technologies

65520, 65525, 65530, 65535, 65540, 65545, 65546, 65548, 65550,  
65554, 65555, 68554, 69000, 64200, 64300

Cirrus Logic

CLGD5420, CLGD5422, CLGD5424, CLGD5426, CLGD5428, CLGD5429,  
CLGD5430, CLGD5434, CLGD5436, CLGD5440, CLGD5446, CLGD5462,  
CLGD5464, CLGD5465, CLGD5480, CLGD6205, CLGD6215, CLGD6225,  
CLGD6235, CLGD6410, CLGD6412, CLGD6420, CLGD6440, CLGD7541(\*),  
CLGD7543(\*), CLGD7548(\*), CLGD7555(\*)

Cyrix

MediaGX, MediaGXm

Compaq

AVGA

Digital Equipment Corporation

TGA

Epson

SPC8110

Genoa

GVGA

IBM

8514/A (and true clones), XGA-2

IIT

AGX-014, AGX-015, AGX-016

Matrox

MCA2064W (Millennium), MGA1064SG (Mystique and Mystique 220),  
MGA2164W (Millennium II PCI and ACP), G100, G200

MX

MX68000(\*), MX680010(\*)

**NCR**

77C22( \* ), 77C22E( \* ), 77C22E + ( \* )

**NeoMagic**

2200, 2160, 2097, 2093, 2090, 2070

**Number Nine**

1128 (series I, II and IV), Revolution 3D (T2R)

**NVidia/SGS Thomson**

NV1, STG2000, RIVA128, Riva TNT

**OAK**

OTI067, OTI077, OTI087

**RealTek**

RTG3106( \* )

**Rendition**

V1000, V2x00

**S3**

86C911, 86C924, 86C801, 86C805, 86C805i, 86C928, 86C864, 86C964,  
86C732, 86C764, 86C765, 86C767, 86C775, 86C785, 86C868, 86C968,  
86C325, 86C357, 86C375, 86C375, 86C385, 86C988, 86CM65, 86C260

**SiS**

86C201, 86C202, 86C205, 86C215, 86C225, 5597, 5598, 6326

**3DLabs**

LINT 500TX, GLINT MX, Permedia, Permedia 2, Permedia 2v

**Tseng**

ET3000, ET4000AX, ET4000/W32, ET4000/W32i, ET4000/W32p, ET6000,  
ET6100

**Trident**

TVGA8800CS, TVGA8900B, TVGA8900C, TVGA8900CL, TVGA9000, TVGA9000i,  
TVGA9100B, TVGA9200CXR, Cyber9320( \* ), TVGA9400CXi, TVGA9420,  
TGUI9420DGi, TGUI9430DGi, TGUI9440AGi, TGUI9660XGi, TGUI9680, Pro-  
Vidia 9682, ProVidia 9685( \* ), Cyber 9382, Cyber 9385, Cyber 9388,  
3DImage975, 3DImage985, Cyber 9397, Cyber 9520

**Video 7/Headland Technologies**

HT216-32( \* )

**Weitek**

P9000, P9100

**Western Digital/Paradise**

PVGA1

**Western Digital**

WD90C00, WD90C10, WD90C11, WD90C24, WD90C24A, WD90C30, WD90C31

要想在 Linux 上很好地运行 X Window, 一定要选择一块视频加速卡, 否则 X Window 不仅不能支持真彩色, 而且窗口的重新绘制速度也会很成问题。说到这里, 笔者建议大家

尽量购买可以扩充到 4M 显存的显示卡,显示卡上配置的显存最少应为 2M,不仅是因为更多的显存支持更高的分辨率、更多的色彩,而且在同样分辨率情况下,显存多的显示卡的显示速度也会更快一些。如果读者的经济能力有限,可以考虑 S3 Trio64 V+,这块卡在 Linux 下的表现也很不错。至于目前最流行的 3D Voodoo 卡, Linux 也是支持的,针对 Voodoo 的驱动程序也在开发中,用户可以在

[http://www.3dfx.com/software/download\\_glide.html](http://www.3dfx.com/software/download_glide.html)

上下载 glide 的驱动程序,不过目前 Linux 上支持 voodoo 的软件只有 ID 公司的 QUAKE II,另外用户可以试验一下 Mesa,这个类似于 OpenGL 的软件也提供了 3DFX 的驱动程序,更详细的情况可以查阅最新的 3dFX howto (本书光盘中提供 VoodooBanshee 和 S3 Savage3D 的 X Server)。

如果 XFree86 不支持用户的显示卡,那么用户可以考虑选用商业 X 服务器包括:

Accelerated-X 1.2

X Inside 公司, Email 地址: info@xinside.com

Metro-X 2.3

Metro Link 公司 Email 地址: sales@metrolink.com

它们支持的显示卡种类都比 XFree86 多,并且都提供 Demo 版,只是正式版价格太高都为 199 美元,对于中国用户来说足够再买三四块显示卡了。

#### 2.2.4 Linux 支持的硬盘控制器/硬盘

##### 1. IDE

Linux 支持标准 IDE、EIDE 和 MFM/RTL 控制器,其中 EIDE 控制器上的两个 IDE 口最多可以支持 4 个硬盘或光驱。Linux 自动探测 EIDE 接口。目前市面上绝大多数 Pentium 级的主板都是 EIDE 接口的, EIDE 控制器包括:

- CMD-640
- DTC 2278D
- FGL/Holtek HT-6560B
- RZ1000
- Triton (82371FB) IDE ( busmaster DMA)

Linux 支持 BUSMASTER 模式和 Ultra DMA, Linux 也支持 ESDI 控制器和老式的 XT 控制器。

##### 2. SCSI

对于 ISP 来说,他们的 Linux 服务器同时可能要接受几百个的访问请求,允许几十个用户同时登录,这时如果选用 IDE 接口的硬盘,那么系统的负载将急剧上升(可以用 top 和 xload 查看),因为 IDE 硬盘读取数据也是要消耗 CPU 的时间,这时候选择 SCSI 接口的硬盘控制器就十分必要。另外像光盘刻录机、磁带机这些用来备份系统的常见外设大多数也是 SCSI 接口。因此建议 ISP 采用 SCSI 接口的硬盘驱动器。而对于个人用户,选择 SCSI 接口就不太划算,同等容量的 SCSI 硬盘要比 IDE 硬盘贵一倍以上,并且还得加上……

块一千多元的 SCSI 卡(挑选一块合适的 SCSI 卡十分重要,因为便宜的 SCSI 卡除了能驱动 CDROM 以外什么也干不了),因此用户还不如去买一块 IDE 硬盘,除了下面列出的 Linux 支持的 SCSI 接口卡列表以外,详细的信息请阅读 SCSI HOWTO。

Linux 支持的 SCSI 卡:

- AMI Fast Disk VLB/EISA (同 BusLogic 兼容)
- Adaptec AVA-1502E (ISA/VLB) (AIC-6360)
- Adaptec AVA-1505/1515 (ISA) (同 Adaptec 152x 兼容)
- Adaptec AHA-1510/152x (ISA) (AIC-6260/6360)
- Adaptec AHA-154x (ISA) (所有型号)
- Adaptec AHA-174x (EISA) (增强模式)
- Adaptec AHA-274x (EISA) / 284x (VLB) (AIC-7770)
- Adaptec AHA-284x (VLB) (AIC-7770)
- Adaptec AHA-2920 (PCI)
- Adaptec AHA-2940/3940 (PCI) (AIC-7870)
- Adaptec AHA-294x/U/W/UW/D/WD (AIC-7871, AIC-7844, AIC-7881, AIC-7884)
- Adaptec AHA-3940/U/W (PCI) (AIC-7872, AIC-7882)
- Adaptec AHA-398x/U/W (PCI) (AIC-7873, AIC-7883)
- Adaptec PCI 控制器 AIC-7850, AIC-7855, AIC-7860
- Adaptec 控制器 AIC-777x (EISA), AIC-785x, AIC-787x (PCI), AIC-788x (PCI)
- Advansys 5140 (ISA) <http://advansys.com/5140o.htm>
- Always IN2000
- BusLogic (ISA/EISA/VLB/PCI) (所有型号)
- DPT PM2001, PM2012A (EATA-PIO)
- DPT Smartcache (EATA-DMA) (ISA/EISA/PCI) (所有型号)
- DTC 329x (EISA) (Adaptec 154x 兼容)
- Future Domain TMC-16x0, TMC-3260 (PCI)
- Future Domain TMC-8xx, TMC-950
- Future Domain TMC-1800, TMC-18C50, TMC-18C30, TMC-36C70
- ICP-Vortex PCI-SCSI 磁盘阵列控制器
- ICP-Vortex EISA-SCSI 控制器
- Media Vision Pro Audio Spectrum 16 SCSI (ISA)
- NCR 5380 generic cards
- NCR 53c400 (Trantor T130B)
- NCR 53c406a (Acculogic ISApport / Media Vision Premium 3D SCSI)
- NCR 53c7x0, 53c8x0 (PCI)
- NCR 53C810, 53C815, 53C820, 53C825, 53C860, 53C875, 53C895
- Qlogic / Control Concepts SCSI/IDE (FAS408) (ISA/VLB)
- Quantum ISA-200S, ISA-250MC
- Seagate ST-01/ST-02 (ISA)

- SoundBlaster 16 SCSI-2 (Adaptec 152x 兼容) (ISA)
- Tekram DC-390, DC-390W/U/F
- Trantor T128/T128F/T228 (ISA)
- UltraStor 14F (ISA), 24F (EISA), 34F (VLB)
- Western Digital WD7000 SCSI

需要用户自己下载驱动程序的 SCSI 卡

- AMD AM53C974, AM79C974 (PCI) (Compaq, HP, Zeos 主板上的 SCSI)  
<ftp://sunsite.unc.edu/pub/Linux/kernel/patches/scsi/AM53C974-0.3.tgz>
- Adaptec ACB-40xx SCSI-MFM/RLL bridgeboard  
<ftp://sunsite.unc.edu/pub/Linux/kernel/patches/scsi/adaptec-40XX.tar.gz>
- Always Technologies AL-500  
<ftp://sunsite.unc.edu/pub/Linux/kernel/patches/scsi/al500-0.2.tar.gz>
- BusLogic (ISA/EISA/VLB/PCI)  
<ftp://ftp.dandelion.com/BUSLogic-1.0-beta.tar.gz>
- Iomega PC2/2B  
[ftp://sunsite.unc.edu/pub/Linux/kernel/patches/scsi/iomega\\_pc2-1.1.x.tar.gz](ftp://sunsite.unc.edu/pub/Linux/kernel/patches/scsi/iomega_pc2-1.1.x.tar.gz)
- Qlogic (ISP1020) (PCI)  
<ftp://sunsite.unc.edu/pub/Linux/kernel/patches/scsi/isp1020-0.5.gz>
- Ricoh GSI-8  
<ftp://tsx-11.mit.edu/pub/linux/ALPHA/scsi/gsi8.tar.gz>

注意:Linux 不支持并口 SCSI 卡。

只要 IDE 或 SCSI 接口能工作,那么挂上的所有硬盘都能被 Linux 支持,所有直接访问的 SCSI 硬盘只要块(block)的大小是 256,512 或 1024 字节就都可以在 Linux 下工作,其他大小的块(block)就不能工作,不过一般用户可以通过设置 SCSI 模式来修改这一数值。在 EIDE 接口下,Linux 支持以 LBA, Large, Normal 模式工作的硬盘,其实 Linux 并不通过 BIOS 来读取硬盘,只是在启动 Linux 时需要 BIOS 来读取启动程序。例如一些老式的硬盘控制器无法读取 500M 以上的大硬盘,结果在 DOS 或 Windows 95 下要安装 EZ-DRIVE, DM 之类的软件才能正常工作,而 Linux 不用安装任何软件就可以读取 500M 以上的硬盘,可是困扰 Linux 的一个常见问题就是对于柱面数大于 1024 的硬盘, Linux 的 LILO (Linux 多重启动控制程序)就不能工作,不过这个问题一般可以通过安装 System Commander 软件来解决。至于硬盘的种类, Linux 支持很多公司的产品,如昆腾的大脚系列硬盘(使用该硬盘作 FTP 服务器是个不错的选择)、IDE 硬盘中昆腾火球系列、Maxtor 的钻石系列、IBM 的水瓶系列质量都不错,在 SCSI 硬盘中 Seagate 也不错。被 Seagate 收购的 Conner 的 CFP1060S 型号的硬盘在 Linux 的 ext2fs 文件系统上有一些问题,

Linux 也支持磁盘阵列(RAID-0, RAID-1)。详情请阅网址:

<ftp://sweet-smoke.ufr-info-p7.ibp.fr/public/Linux/>

## 2.2.5 Linux 支持的磁带机

Linux 支持大多数 SCSI 接口的磁带机和使用软盘控制器的磁带机,以及 QIC-02 磁带

机 QIC-117, QIC-40/80, QIC-3010/3020 (QIC-WIDE) 磁带机, 也可使用 Colorado FC-10/FC-20, Mountain Mach-2 和 Iomega Tape Controller II 接口的磁带机。

驱动程序的网址为:

<ftp://sunsite.unc.edu/pub/Linux/kernel/tapes>

Linux 不支持下列磁带机:

- Emerald and Tecmar QIC-02 控制卡
- 连接在并口上的磁带机
- Colorado TC-15
- Irwin AX250L/Accutrak 250
- IBM Internal Tape Backup Unit
- COREtape Light

### 2.2.6 Linux 支持的光盘驱动器(CDROM)(包括光盘刻录机)

Linux 支持所有块(block)的大小为 512 或 2048 字节的 SCSI 接口的光盘驱动器。一般市面上所售的 SCSI 光盘驱动器都属于这种类型。

Linux 支持下列的 EIDE(ATAPI)接口的光盘驱动器:

- Aztech CDA268, Orchid CDS-3110, Okano/Wearnes CDD-110, Conrad TXC
- GoldStar R420
- LMS Philips CM 206
- Matsushita/Panasonic, Creative Labs, Longshine, Kotobuki (SBPCD)
- Mitsumi
- Optics Storage Dolphin 8000AT
- Sanyo H94A
- Sony CDU31A/CDU33A
- Sony CDU-535/CDU-531
- Teac CD-55A SuperQuad

Linux 支持下列公司生产的 CD-R 光盘刻录机和他们的兼容产品:

Philips, IMS, Kodak, Yamaha, Hewlett-Packard

一般来说只要是连接在硬盘口上的光盘驱动器,且主板的 BIOS 能支持该光盘驱动器,那么 Linux 都能支持,但连接在声卡上的光盘驱动器比较成问题,一般并不推荐。挑选光盘驱动器主要看速度快不快,挑不挑盘,稳定性如何,现在高速光驱的技术已经稳定下来,读盘能力大幅度提高,例如飞利浦的 24 倍速的光驱就不错。选择光盘刻录机最好选用 SCSI 接口的,IDE 接口的质量不太稳定,Linux 也支持 PhotoCD (XA)。

### 2.2.7 Linux 支持的便携式驱动器

Linux 支持所有 SCSI 接口的便携式驱动器,包括 Optical (MO), WORM, Floptical, Bernoulli, Zip, SyQuest, PD 等, Linux 也支持并口的 Zip 驱动器,相应的网址在 <ftp://gear.torque.net/pub/>, 在 Linux 下这些驱动器就像软驱一样工作,用户可以对它进行 fdisk/mkfs/mount 等操作。如果该驱动器提供锁定驱动器的活门的功能, Linux 也能够支持这种功能,并且 Linux 下的 mtools 也能作用于这些驱动器,用来存取 MS-DOS 格式的文件。

### 2.2.8 Linux 支持的 I/O 接口

Linux 支持所有的标准串口、并口及游戏操纵杆。Linux 支持的相应芯片有：8250, 16450, 16550 和 16550A UART, 所有使用中断大于 9 的接口卡都可以正常工作。

Linux 也支持多用户卡, Linux 支持的多用户卡的列表如下:

- AST Fourport and clones (4 口)
- Accent Async-4 (4 口)
- Arnet Multiport-8 (8 口)
- Bcll Technologies HUB6 (6 口)
- Boca BB-1004, 1008 (4, 8 口)
- Boca BB-2016 (16 口)
- Boca IO/AT66 (6 口)
- Boca IO 2by4 (4 串/2 并, 使用 5 个中断)
- Computone Valueport (4, 6, 8 口)
- DigiBoard PC/X (4, 8, 16 口)
- Control Hostess 550 (4, 8 口)
- PC-COMM 4-port (4 口)
- SIIG I/O Expander 4S (4 口, 使用 4 个中断)
- STB 4-COM (4 口)
- Twincom ACL/550
- Usenet Serial Board II (4 口)
- Cyclades Cyclom-8Y/16Y (8, 16 口) (ISA/PCI)
- Stallion EasyIO (ISA) / EasyConnection 8/32 (ISA/MCA)
- Stallion EasyConnection 8/64 / Onboard (ISA/EISA/MCA) / Brumby / Stallion (ISA)
- Control RocketPort (8/16/32 口)  
<ftp://tsx-11.mit.edu/pub/linux/packages/control/>
- Computone IntelliPort II (4/8/16 口)
- DigiBoard COM/Xi
- DigiBoard PC/Xe (ISA) and PC/Xi (EISA)  
<ftp://ftp.digibd.com/drivers/linux/>
- Hayes ESP8
- Moxa C218 (8 口) / C320 (8/16/24/32 可扩充)  
<ftp://ftp.moxa.com.tw/drivers/c-218-320/linux/>
- Specialix SIO/XIO (4 到 32 口)  
[ftp://sunsite.unc.edu/pub/Linux/kernel/patches/serial/sidrv0\\_5.taz](ftp://sunsite.unc.edu/pub/Linux/kernel/patches/serial/sidrv0_5.taz)

### 2.2.9 Linux 支持的网卡

Linux 的强大生命力一半来源于网络, 因此对于用户来说选择一块合适的网卡是十分重要的。而选择网卡要注意的地方主要在于它的性能和兼容性, 在网络带宽越来越大、应用越来越广泛的今天, 对于要使用 Linux 作为服务器的 ISP 来说, 挑选一块好网卡对系统性能的提升有很大的帮助, 对于这些 ISP 来说, 选择 100M 的以太网卡已经成为潮流。而对于没有这些高档应用的普通用户, 一块几十元的兼容 NE2000 网卡就能很好地工作了。

但是在兼容网卡中选择合适的品牌也是值得花一番心思的。目前市面上最多的就是 NE2000 和 D-link 的兼容网卡,一般说来选择 NE2000 的比较好一点。如果选择一块很冷僻的网卡,那么该网卡在 Linux 下也许不能正常工作,到时候可就要大费手脚了。

下面是 Linux 支持的网卡列表:

- 3Com 3C501
- 3Com 3C503, 3C505, 3C507, 3C509/3C509B (ISA) / 3C579 (EISA)
- 3Com 3C905 系列
- AMD LANCE (79C960) / PCnet-ISA/PCI (AT1500, HP J2405A, NE1500/NE2100)
- AT&T GIS WaveLAN
- Allied Telesis AT1700, Allied Telesis LA100PCI-T
- Ansel Communications AC3200 EISA
- Apricot Xen-II
- Cabletron E21xx
- Cogent EM110
- Crystal Lan CS8920, Cs8900
- DEC DE425 (EISA) / DE434/DE435 (PCI)
- DEC DE450/DE500-XA
- DEC DEPCA & EtherWORKS
- HP PCLAN (27245 and 27xxx series)
- HP PCLAN PLUS (27247B and 27252A)
- HP 10/100VG PCLAN (ISA/EISA/PCI)
- Intel EtherExpress
- Intel EtherExpress Pro
- KTI ET16/P-D2, ET16/P-DC ISA
- NE2000/NE1000(包括兼容网卡)
- Netgear FA-310TX
- New Media Ethernet
- Racal-Interlan NI5210 (使用 i82586 芯片)
- Racal-Interlan NI6510 (使用 am7990 芯片) -16M 内存以上就不能工作。
- PureData PDUC8028, PDI8023
- SEEQ 8005
- SMC Ultra
- SMC 9000
- SMC PCI EtherPower 10/100
- SMC EtherPower II
- Schneider & Koch G16
- Western Digital WD80x3
- Zenith Z-Note (IBM ThinkPad 300 笔记本内置的网卡)
- Znyx 312

便携式网卡:

- AT-Lan-Tec/RealTek 并口卡
- D-Link DE600/DE620 并口卡

无插槽卡：

- SLIP/CSLIP/PPP (串口)
- EQL
- PLIP (并口)

ARCnet 网卡：

- Linux 支持所有的 ARCnet 网卡

令牌环(Token Ring)网卡：

- 使用 IBM Tropic 芯片的网卡
- Amateur radio (AX.25)
- Ottawa PI/PI2
- Most generic 8530 based HDLC boards

下列网卡需要去网络上下载驱动程序：

- 3Com Demon Ethercards (3C592, 3C597 (100 mbps)) (EISA)  
<http://cesdis.gsfc.nasa.gov/linux/drivers/vortex.html>
- 3Com Vortex Ethercards (3C590, 3C595 (100 mbps)) (PCI)  
<http://cesdis.gsfc.nasa.gov/linux/drivers/vortex.html>
- DEC 21040/21140 "Tulip" / SMC PCI EtherPower 10/100  
<http://cesdis.gsfc.nasa.gov/linux/drivers/tulip.html>
- HP J2585 (PCI) / HP J2573 (ISA) (ATT2MDx1 / 100VG)  
<http://cesdis1.gsfc.nasa.gov:80/linux/drivers/100vg.html>

ISDN：

- Linux ISDN 主页  
<http://www.ix.de/ix/linux/linux-isdn.html>
- 3Com Sonix Arpeggio  
<ftp://sunsite.unc.edu/pub/Linux/kernel/patches/network/sonix.tgz>
- Combinet EVERYWARE 1000 ISDN  
<ftp://sunsite.unc.edu/pub/Linux/patches/network/combinet1000isdn-1.02.tar.gz>
- Diehl SCOM 卡  
<ftp://sunsite.unc.edu/pub/Linux/kernel/patches/network/isdndrv-0.1.1.tar.gz>
- ICN ISDN / Teles ISDN / Creatix AVM ISDN 卡  
<ftp://ftp.franken.de/pub/isdn4linux/>

ASUSCOM Network Inc. ISDNLink 128K PC adapter (HiSax)

AVM A1 (HiSax)

Combinet EVERYWARE 1000 ISDN

Compaq ISDN SO (ISA) (HiSax)

Creatix PnP SO (HiSax)

Dr. Neuhaus Niccy PnP/PCI (HiSax)

Dynalink IS64PH (HiSax)

Eicon.Diehl Diva 2.0 ISA & PCI (HiSax)

Eicon.Diehl Diva Piccola (HiSax)

Elsa Microlink PCC-16, PCF, PCF-Pro, PCC-8 (HiSax)  
ELSA QuickStep 1000/1000PCI/3000 (HiSax)  
HFC-2BS0 based cards (HiSax)  
IBM Active 2000 (ISA) (act2000)  
ICN ISDN cards (icn)  
Ith Kommunikationstechnik GmbH MIC 16 (ISA) (HiSax)  
ITK ix1-micro Rev.2 (HiSax)  
Octal PCBIT (pccbit)  
Sedlbauer Speed Card (HiSax)  
Teles 8.0/16.0/16.3 (HiSax)  
Teles 16.3c (HiSax)  
Teles S0 (HiSax)  
Traverse Technologie NETjet PCI S0 (HiSax)  
USR Sportster internal TA (HiSax)

如果 ISDN 卡模拟成 modem 或以网卡来工作,那么不用任何特别的驱动程序就能工作。

PCMCIA 卡:

目前支持所有通用的 PCMCIA 卡。

- 3Com 3c589, 3c589B
- Accton EN2212 EtherCard
- CNet CN30BC Ethernet
- D-Link DE-650
- EFA InfoExpress SPT EFA 205 10baseT
- EP-210 Ethernet
- Farallon Etherwave
- GVC NIC-2000P Ethernet Combo
- HYPERTEC HyperEnet
- IBM CreditCard Ethernet
- IC-Card Ethernet
- Katron PE-520 Ethernet
- Kingston KNE-PCM/M
- LANEED Ethernet
- Linksys EtherCard
- Maxtech PCN2000 Ethernet
- New Media Ethernet
- Novell/National NE4100 InfoMover
- Proteon Ethernet
- PreMax PE-200 Ethernet
- RPTI EP400 Ethernet

- Socket Communications Socket EA LAN 网卡
- Thomas-Conrad Ethernet
- Volktek Ethernet

ATM 卡：

- Efficient Networks ENI155P-MF 155 Mbps ATM 卡 (PCI)

<http://ircwww.epfl.ch/linux-atm/>

帧中继：

- Sangoma S502 56K Frame Relay card

<ftp://ftp.sovereign.org/pub/wan/fr/>

## 2.2.10 Linux 支持的声卡

在一般用户的概念中,要玩多媒体,就一定要用 Windows 95,似乎 UNIX 就从来没有多媒体功能。虽然 UNIX 历史上对多媒体的支持的确没有什么良好的口碑,但 Linux 的多媒体功能扩展比起 Win95 来一点也不逊色,用户可以用 xanim 看 avi, mov 等格式的影音文件,用 midiplay 来播放 midi,用 mpg123, l3dec, X11amp 来听 mp3 格式的歌曲,用 workbone / cdplayer 来听音乐 CD,用 RealAudio 来收听网络广播,在 QUAKE 的世界中听到敌人的惨叫声,用 mtv 来观看 VCD 等等。要享受这一切就都需要一块声卡。一般来说,只要是同 Sound Blaster 兼容的声卡, Linux 都能很好地支持,用户只需要记下声卡的中断、DMA 等参数,编译出一个可加载模块或重新编译一遍系统核心,用户的声卡就可以立刻开始工作。用户购买声卡时最好不要选购一些在 Windows 95 下还要运行自己的一些设置程序才能工作的声卡,如 Als007 之类的声卡,选择花王系列的声卡也很不错,质量好,价格也比较适中。如果对声音质量要求较高的话,选择 Sound Blaster AWE 32 吧!

Linux 支持的声卡列表如下:

- 6850 UART MIDI
- Adlib (OPL2)
- Audio Excell DSP16
- Aztech Sound Galaxy NX Pro
- Crystal CS4232 (PnP) based cards
- ECHO-PSS cards (Orchid SoundWave32, Cardinal DSP16)
- Ensoniq SoundScape
- Gravis Ultrasound
- Gravis Ultrasound 16-bit sampling daughterboard
- Gravis Ultrasound MAX
- Gravis Ultrasound ACE
- Gravis Ultrasound PnP
- Logitech SoundMan Games (SBPro)
- Logitech SoundMan Wave (Jazz16/OPL4)
- Logitech SoundMan 16 (PAS-16 兼容)
- MPU-401 MIDI
- MediaTriX AudioTriX Pro
- Media Vision Premium 3D (Jazz16)

- Media Vision Pro Sonic 16 (Jazz)
- Media Vision Pro Audio Spectrum 16
- Microsoft Sound System (AD1848)
- OAK OTI-601D cards (Mozart)
- OPTi 82C928/82C929 cards (MAD16/MAD16 Pro)
- Sound Blaster
- Sound Blaster Pro
- Sound Blaster 16
- Sound Blaster 32/64/AWE
- Turtle Beach Wavefront cards (Maui, Tropez)
- Wave Blaster (and other daughterboards)
- MPU-401 MIDI (intelligent mode)  
ftp://sunsite.unc.edu/pub/Linux/kernel/sound/mpu401-0.2.tar.gz
- 使用 PC 喇叭  
ftp://ftp.informatik.hu-berlin.de/pub/os/linux/hu-sound/
- Turtle Beach MultiSound/Tahiti/Monterey  
ftp://ftp.cs.colorado.edu/users/mccreary/archive/theach/multisound/

### 2.2.11 Linux 支持的鼠标

Linux 支持大多数市面上销售的鼠标。不过笔者不推荐大家购买常见的两键鼠标,因为在 UNIX 下用户会发现鼠标上的三个键个个都独挡一面,不像在 Windows 95/NT 下只使用两个键,夸张的苹果机鼠标只使用一个键。虽然 Linux 可以用两键来模拟三键,但是在使用中并不够方便舒适,所以建议大家购买三键的鼠标,注意有些鼠标有三键但是中间的那个键是不起作用的,只是装饰物。鼠标一般以机械鼠标的使用寿命较长,而光电鼠标的一般使用一段时间后就不太灵敏。质量最好的当然是罗技(logitech)的天貂鼠,只是价格不菲。Linux 支持的鼠标列表如下:

- Microsoft serial mouse
- Mouse Systems serial mouse
- Logitech Mouseman serial mouse
- Logitech serial mouse
- ATI XL Inport busmouse
- C&T 82C710 (QuickPort) (Toshiba, TI Travelmate)
- Microsoft busmouse
- Logitech busmouse
- PS/2 (auxiliary device) mouse
- Sejin J-mouse  
ftp://sunsite.unc.edu/pub/Linux/kernel/patches/console/jmouse.1.1.70-jmouse.tar.gz
- MultiMouse  
ftp://sunsite.unc.edu/pub/Linux/system/Misc/MultiMouse-1.0.tgz

另外所有支持标准鼠标协议的触摸屏在 Linux 下都能工作,如 Alps Glidepoint。

### 2.2.12 Linux 支持的游戏操作杆

- Joystick 驱动程序

<ftp://sunsite.unc.edu/pub/Linux/kernel/patches/console/joystick-0.7.3.tgz>

- Joystick 驱动程序 (可加载模块)

<ftp://sunsite.unc.edu/pub/Linux/kernel/patches/console/joyfixed.tgz>

### 2.2.13 Linux 支持的调制解调器

Linux 支持所有连接到串口的外置调制解调器和所有内置的调制解调器。有一些调制解调器需要在 DOS 下运行一个软件来加载控制程序才能正常工作,用户在 DOS 下加载完毕后进行一次热启动就可以在 Linux 下工作了。所有 PCMCIA 接口的调制解调器只要 Linux 支持相应的 PCMCIA 卡就能工作,支持传真的调制解调器需要安装相应的 FAX 软件才能工作。

- Digicom Connection 96 + /14.4 +

<ftp://sunsite.unc.edu/pub/Linux/system/Serial/smdl-linux.1.02.tar.gz>

- ZyXEL U-1496 series - ZyXEL 1.4, modem/fax/voice 控制程序

<ftp://sunsite.unc.edu/pub/Linux/system/Serial/ZyXEL-1.4.tar.gz>

### 2.2.14 Linux 支持的打印机/绘图机

所有连接到并口/串口的普通打印机都可以工作,一些特别的控制软件如下:

- HP LaserJet 4 系列的控制软件

<ftp://sunsite.unc.edu/pub/Linux/system/Printing/free-lj4-1.1pl.tar.gz>

- BiTronics 并口打印机

<ftp://sunsite.unc.edu/pub/Linux/kernel/misc/bt-ALPHA-0.0.1.tar.gz>

有很多的 Linux 程序输出 PS(PostScript)格式的文件,常见的非 PS 打印机可以通过安装 Ghostscript 来提供一个 PS 文件打印过滤器,从而能够让用户在非 PS 打印机上直接打印 PS 文件,Slackware/Redhat 都包括 Ghostscript,另外 RedHat 提供的 printtool 是一个很好的安装打印机的工具.Ghostscript 可以在下述网址下载:

<ftp://ftp.cs.wisc.edu/pub/ghost/aladdin/>

Linux 支持的打印机列表:

- Apple Imagewriter
- C. Itoh M8510
- Canon BubbleJet BJ10e, BJ200
- Canon LBP-8II, LIPS III
- DEC LA50/70/75/75plus
- DEC LN03, LJ250
- Epson 9 pin, 24 pin, LQ series, Stylus, AP3250
- HP 2563B
- HP DesignJet 650C
- HP DeskJet/Plus/500
- HP DeskJet 500C/520C/550C/1200C 彩色打印机
- HP LaserJet/Plus/II/III/4

- HP PainJet/XL/XL300 彩色打印机
- IBM 激光彩色打印机
- IBM Proprinter
- Imagen ImPress
- Mitsubishi CP50 彩色打印机
- NEC P6/P6+/P60
- Okidata MicroLine 182
- Ricoh 4081
- SPARCprinter
- StarJet 48 inkjet printer
- Tektronix 4693d color 2/4/8 bit
- Tektronix 4695/4696 喷墨绘图仪
- Xerox XES printers (2700, 3700, 4045, 等等)
- Canon BJC600 和 Epson Esc/P 彩色打印机

### 2.2.15 Linux 支持的扫描仪

- A4 Tech AC 4096  
ftp://ftp.informatik.hu-berlin.de/pub/local/linux/ac4096.tgz
- Epson GT6000  
ftp://sunsite.unc.edu/pub/Linux/apps/graphics/scanners/ppic0.5.tar.gz
- Fujitsu SCSI-2 扫描仪  
向 Dr. G. W. Wettstein 联系 (greg%wind.UUCP@plains.nodak.edu)
- Genius GS-B105G  
ftp://tsx-11.mit.edu/pub/linux/ALPHA/scanner/gs105-0.0.1.tar.gz
- Genius GeniScan GS4500 手持扫描仪  
ftp://tsx-11.mit.edu/pub/linux/ALPHA/scanner/gs4500-1.3.tar.gz
- HP ScanJet, ScanJet Plus  
ftp://ftp.Ctrl-c.liu.se/UNIX/linux/wingel/
- HP ScanJet II series SCSI  
ftp://sunsite.unc.edu/pub/Linux/apps/graphics/scanners/hpscanpbm-0.3a.tar.gz
- HP ScanJet 系列  
http://www.tummy.com/xvscan/
- Logitech Scanman 32 / 256  
ftp://tsx-11.mit.edu/pub/linux/ALPHA/scanner/logiscan-0.0.2.tar.gz
- Mustek M105 手持扫描仪  
ftp://tsx-11.mit.edu/pub/linux/ALPHA/scanner/scan-driver-0.1.8.tar.gz
- Mustek Paragon 6000CX  
ftp://sunsite.unc.edu/pub/Linux/apps/graphics/scanners/muscan-1.1.5.taz
- Nikon Coolscan SCSI 35mm 电影胶片扫描仪  
ftp://sunsite.unc.edu/pub/Linux/apps/graphics/scanners/

- UMAX SCSI 扫描仪  
同 Craig Johnston (mkshenk@u.washington.edu) 联系。

### 2.2.16 Linux 支持(DPMS) 显示器

节能性能已经被加入到 Linux 的核心中, 只要用 setterm 这个功能就可以让屏幕自动进入节电状态。

### 2.2.17 Linux 支持的视频捕捉卡

Linux 支持的视频捕捉卡如下:

- FAST Screen Machine II 视频捕捉卡  
`ftp://sunsite.unc.edu/pub/Linux/apps/video/ScreenMachineII.1.2.tgz`
- ImageNation Cortex I 视频捕捉卡  
`ftp://sunsite.unc.edu/pub/Linux/apps/video/cortex.drv.0.1.tgz`
- ImageNation CX100 视频捕捉卡  
`ftp://sunsite.unc.edu/pub/Linux/apps/video/cxdrv-0.1beta.tar.gz`
- Pro Movie Studio 视频捕捉卡  
`ftp://sunsite.unc.edu/pub/Linux/apps/video/PMS-grabber.2.0.tgz`
- Quanta WinVision 视频捕捉卡  
`ftp://sunsite.unc.edu/pub/Linux/apps/video/fggrabber-1.0.tgz`
- Video Blaster, Rombo Media Pro + 视频捕捉卡  
`ftp://sunsite.unc.edu/pub/Linux/apps/video/vid_src.gz`
- VT1500 TV 视频捕捉卡  
`ftp://sunsite.unc.edu/pub/Linux/apps/video/vt1500-1.0.5.tar.gz`

### 2.2.18 Linux 支持的不间断电源 UPS

Linux 支持的 UPS 列表如下:

- APC SmartUPS  
`ftp://sunsite.unc.edu/pub/Linux/system/UPS/apcd-0.1.tar.gz`
- 使用 RS-232 端口监视 UPS (unipower 包)  
`ftp://sunsite.unc.edu/pub/Linux/system/UPS/unipower-1.0.0.tgz`

要查询更多的信息请阅读 UPS HOWTO。

### 2.2.19 Linux 支持的数据采集卡

- Linux Lab 计划:  
`ftp://koala.chemie.fu-berlin.de/pub/linux/LINUX-LAB/`

上面罗罗嗦嗦地讲了一大篇, 读者是不是有一些眼花缭乱? 不要紧, 下面笔者就结合自己的经验向大家推荐一个可以很好地运行 Linux 的硬件配置(见表 2.1)。

当然, 笔者的建议只是参考, 而且现在 PC 机硬件的发展非常快, 很快现有的设备就会过时, 价格也会一落千丈。但只要大家多跑一跑电脑市场, 很快就会成为硬件专家的。另外千万别忘了多多阅读相关文档, 如果遇到什么拿不准的问题, 去 BBS 上问一问其他

用户会很有好处的。

表 2.1 硬件配置表

| 类 型   | 一般用户                 | 小型 ISP 用户          |
|-------|----------------------|--------------------|
| 主板    | 微星 6119              | 华硕 (Asus) P2B      |
| CPU   | Pentium Celeron 300A | Pentium II 400 x2  |
| 内存    | 32M EDO RAM          | 128M SDRAM         |
| 显示卡   | Winfast S680         | Winfast S600       |
| 显示器   | Samsung 15"          | Samsung 15"        |
| 硬盘    | 昆腾火球四代 6.4G          | Seagate SCSI 18.2G |
| 光驱    | Maxell 32X           | Maxell 32X         |
| 网卡    | NE2000 兼容            | Interl EE pro 100  |
| Modem | 33.6K                | /                  |
| 鼠标    | 真三键机械鼠标              | 真三键机械鼠标            |
| 打印机   | /                    | HP laserjet 6L     |

## 2.3 Linux 安装前的准备工作

选购完称心如意的 PC 机,下面就可以开始 Linux 的安装了,在进行 Linux 的安装之前,有几项重要的工作不可不做。下面就简单介绍一下安装 Linux 的准备工作,如果用户是 Linux 方面的新手,一定要仔细阅读。

### 1. 准备一张 DOS 启动盘,上面应该拷贝有下列程序

```

MSDOS.SYS          SYS.COM
IO.SYS              FORMAT.COM
COMMAND.COM         SCANDISK.EXE
FDISK.EXE           SCANDISK.INI

```

这张启动盘是为了防止万一在安装 Linux 的过程中出现致命错误后恢复硬盘使用的,如果用户有经验的话还应该自行加上 Norton 公司出品的 NU8.0 系列中的 NDD 和 DISKEDIT 这两套强劲的修复软件或是 RESURE 安全恢复软件。

准备三张空白的软盘,最好是 1.44M 的 3 吋盘,这三张软盘请在 DOS 下先格式化好。准备好纸和笔,用来记录安装过程中的一些参数。

### 2. 在硬盘上安装 MS-DOS 或 Windows 95 操作系统以及 CD-ROM 的驱动程序

如果用户的机器上已经安装了相应的操作系统或者用户不准备通过 CD-ROM 安装 Linux 就可以忽略这一步。(事实上, Linux 可以在一个光秃秃的系统上安装。但大多数用户在安装 Linux 前可能机器上已安装了 DOS 或 Windows 95 等操作系统,并希望以后也能使用它们。所以我们在这里说明了这一点。)因为大多数用户主要是通过 CD-ROM 方式安装,所以本书将侧重讲解如何通过 CD-ROM 安装 Linux,至于通过网络安装(包括 SMB, NFS, FTP 等方式),通过软盘安装,通过磁带安装,通过硬盘安装这些安装方式将只作一般

性的介绍,不再做详细的讲解。安装完操作系统和 CD-ROM 的驱动程序之后,就可以将 Linux 的安装光盘插入光驱检查,用户可以使用 `dir/s` 查看一下是否能够正确读出光盘上的文件,不然拷贝到一半读不出光盘可是一件很尴尬的事情。

### 3. 准备安装源文件

准备好 Slackware Linux CD-ROM 或相应的软盘,或者是拷贝到硬盘上的 Slackware 源文件。并且记录下正确的目录位置(如在 `cdrom` 上就是 `/slakware`)。如果通过 NFS 安装,则需要记录下远程主机上存放 Linux 的正确目录,如果用户有 UNIX 主机的话,可以用“`showmount -e 主机名`”来查看其 `export` 出来的安装盘。如果是 FTP 安装,就需要知道 FTP 服务器上存放 Linux 的目录路径以及登录用户名和密码,如果是 SMB 安装就需要知道服务器名和用户名及密码。

### 4. 备份系统

如果用户的机器上已经安装有 MS-DOS 或 Windows 95/NT 这些操作系统,建议用户将硬盘上现存的一些重要数据进行备份,如 Word 文档、程序源代码和数据文件等。虽然 Linux 的安装是很安全的,但是养成一个随时备份的好习惯对用户的工作是很有帮助的。此外如果用户有经验的话,将系统文件备份也是一个很好的主意,即用 Norton 公司的 NCU 最新版本中的 `diskedit` 或 `rescue` 程序将硬盘的 `boot`, `root` 分区表都备份到软盘上,这样就有备无患,即使整个硬盘都找不到那也没有关系,只要将软盘上的备份再恢复回去就可以完全恢复安装前的状态。

### 5. 收集并检查系统的硬件信息

Linux 对 PC 硬件的利用比 MS-DOS、Windows 95/NT 更有效,但是 Linux 需要一些硬件的相关信息才能正确配置相应的硬件,所以用户在安装前尽可能地收集用户的硬件信息:首先把用户所有的硬件使用手册(主板、显示卡、显示器、调制解调器、硬盘、声卡和光驱等使用手册)放在手边。然后利用 MS-DOS/Windows 95 下的软件进行硬件检测。如果用户的系统使用 MS-DOS5.0 或更高版本,就可以运行微软诊断工具 `MSD.EXE`,在 MSD 中选择 `print report`(打印报告)将系统的同硬件相关的信息打印出来,如果用户没有打印机,将它打印到文件,再将这个文件抄写到纸上。如果用户使用的是 Windows 95,只要选择开始(start) -> 设置(setup) -> 控制面板(control panel) -> 系统(system) -> 设备管理(device management),再查看或打印。其他类似的软件有 Norton 公司的 `SYSINFO` 等。普通用户需要记录下来的一般只有硬盘的参数、网卡的中断号和地址、鼠标和调制解调器的类型和端口号、显示卡的类型和显存的大小、显示器的类型和参数、内存的大小、光驱的类型以及连接到第几个 IDE 口、光驱是否是连接到声卡上,如果用户的机器连接在网络上,那么还要记下用户的机器的节点名、域名、网络掩码、路由地址、名字服务器地址等等信息。还有声卡的种类、中断号、DMA 和输出端口。用户别小看这些信息,下面在安装 Linux 时却是非常有用的,免去用户很多繁琐的工作,至少能让用户少启动机器几次。

## 6. 配置用户的机器

首先确定用户的机器是否能够从 3 吋盘启动,如果是从 5 吋盘启动那么查看一下在 CMOS 中是否能更改这一设置,将机器设置成从 3 吋盘启动,对于较老的机器,一般是没有这个选项的,只有靠手工去更换两个软驱的连线顺序,那样就实在太麻烦了,这种情况下就只有选择 1.2M 的 bootdisk 了。此外在 CMOS 的设置中要注意以下几点:

① 千万要把 Anti virus(防止病毒写主引导区)这个选项关闭!!! 否则在安装 LILO 时会出现死机。

② 将 PnP bios(即插即用)关闭,否则 Linux 有可能不能正确检测用户的网卡、声卡和光驱的参数。

③ 关闭 Swap A,B(交换 AB 盘)这个选项,否则启动时可能会出现问題。

④ 将启动顺序调整为(A,C),很多用户为了防止病毒和加快启动速度,将 C 作为第一启动盘,现在安装 linux 需要从软盘启动,所以用户可以将这个选项打开。如果用户的 BIOS 支持 CDROM 启动(当然用户首先得查看一下手中的 Linux 光盘是否支持光盘启动),则用户可以选择为 CDROM,C,并且用户可以省略下面如何制作 bootdisk/rootdisk 这个步骤。

⑤ 在 CMOS 中查看一下 Memory Hole 这个选项,如果打开的话,不管用户的内存有多大,Linux 将只能检测到 16M 内存。

⑥ 如果用户的硬盘是 IDE 的,并且柱面数大于 1024,那么用户的 LILO 安装也许会出现问题。详细情况请参阅 LILO 的配置和 Large Disk mini-HOWTO。

## 7. 在硬盘上给 Linux 分配可以安装的分区

一般说来,用户计算机上的硬盘已经全部用于 MS-DOS、Windows 95/NT 或者 OS/2 这些操作系统的分区。为了给 Linux 分配安装空间,用户需要重新划分这些分区的大小,如果用户不想将硬盘重新分区,或者用户只是想尝试一下 Linux,那么用户在选择安装时,可以选择 UMSDOS 方式安装,这种安装方式将 Linux 安装到用户的硬盘上 MS-DOS 分区上的 \linux 目录下,使用 MS-DOS 的文件格式来存放 Linux 文件,通过 UMSDOS 的驱动程序将 MS-DOS 的 8.3 文件名转化为 Linux 下的长文件名。用户可以在 DOS 下通过运行 Loadlin 这个工具程序来启动该 Linux,这种做法的好处是不用将硬盘重新分区,对没有经验的用户来说相对要安全一点,而且卸载 Linux 也要简单得多,只需要在 DOS 下删除这个目录就可以了。缺点是以这种方式安装的 Linux 磁盘读取速度较慢,并且如果用户安装软件较多的话,在硬盘上产生过多的文件,那么 MS-DOS 文件系统就有可能崩溃,并且这种安装方式的安全性是很差的,任何人都可以在 MS-DOS 下阅读和删除 Linux 文件,笔者并不推荐大家使用这种方式安装 Linux。

硬盘分区就是为某一操作系统分配的硬盘上的一部分空间。如果用户只安装了 MS-DOS,用户的硬盘也许只有一个分区,全部用于 MS-DOS。分区有三种类型:主分区(primary partition)、扩展分区(extended partition)和逻辑分区(logical partition)。一个硬盘最多只能有四个主分区。如果用户想在每一硬盘上拥有多于四个的分区,用户需要创建扩展分区,扩展分区上能划分出许多逻辑分区。用户不要把数据直接存在扩展分区上——它仅仅用来存放逻辑分区,数据只存放在主分区或逻辑分区上。如果用户目前的磁盘空间已经全部

分配给 DOS/Windows 95 使用,那么有以下方式给 Linux 重新分配硬盘空间:

(1) 将 MS-DOS 下的某个逻辑盘(例如 D:)上的数据全部移到其他分区上,将该盘上的文件全部删除,这样就可以让 Linux 使用这个分区来安装。如果用户有两个硬盘,直接将第二个硬盘分配给 Linux 就可以了。

(2) 重新将硬盘分区,这是最直接,也是最麻烦的一步,用户需要备份用户硬盘上的全部数据,重新安装 MS-DOS, Windows 95/NT,以及数以百计的软件。

这个方法就是使用 FDISK 程序去修改分区。例如,用户现在有一个 1G 的硬盘,全部用于 MS-DOS。那么就重新分区分配 600M 给 MS-DOS 使用,400M 给 Linux 使用。为了重新分配硬盘空间,用户得在 DOS 下运行 FDISK,删除全部的 DOS 分区,在原来的地方建立一个 600M 的 DOS 分区,然后格式化这个分区为 MS-DOS 格式,再重新安装 MS-DOS 操作系统,其他的 400M 空间用户不必进行任何操作。在安装 Linux 时再让 Linux 对这个分区进行操作,具体步骤如下:

- ① 备份用户的系统;
- ② 制作 MS-DOS 的启动盘,命令如下:FORMAT /S A:;
- ③ 复制 FDISK.EXE,FORMAT.COM,SYS.COM 到这片软盘上;
- ④ 使用这片 MS-DOS 系统盘启动系统;
- ⑤ 运行 FDISK,选择要修改的硬盘(如 C:或 D:);
- ⑥ 使用 FDISK 的菜单选项去删除要该变大小的分区,该分区的数据将全部被删除;
- ⑦ 用 FDISK 菜单选项选择要分配给 MS-DOS 的容量,重建这些分区;
- ⑧ 退出 FDISK,用 FORNAT 命令重新格式化新的分区;
- ⑨ 在新分区上重新安装 MS-DOS 操作系统和相应软件。

如果用户的操作系统是 OS/2 或 Windows 95/NT,那么重新分区的原理是相似的,请阅读相应的系统文档。

(3) 最安全和方便的办法,使用 Linux CDROM 上自带的 FIPS 程序,该程序其实是一个功能非常强大的 MS-DOS 工具程序,它可以在不删除硬盘上原有数据的情况下将硬盘重新分区。这个程序是由 Arno Schaefer(schaefer@rbg.informatik.th-darmstadt.de)先生编写的。该程序只能分割 MS-DOS 的主分区,不能分割 MS-DOS 的扩展分区。下面简单介绍一下该程序的使用过程:

① 建立一张 DOS 启动盘,将 CDROM 上的/install/FIPS 目录下的 RESTORRB.EXE, FIPS.EXE 和 ERRORS.TXT 拷贝到该软盘上,笔者不建议直接在硬盘上使用 FIPS;

② 接下来请用户使用 MS-DOS 或 Windows 95 提供的 SCANDISK(或 Norton 公司最新版本的 NDD)来检查要划分的硬盘,修复错误,然后用 DEFRAG(或 Norton 公司最新版本的 Speedisk)来将硬盘重新整理,保证硬盘的后半部分没有任何 MS-DOS 文件,注意像 mirror 或 image 这样的程序会重新把一些文件放在硬盘的最后面,请用“attrib -r -s -h image.idx”或“attrib -r -s -h mirrorsav.fil”将这些文件的属性改为可读并删除他们,另外在 autoexec.bat 或 config.sys 中除去 image 或者 mirror 及其他防病毒程序如 NAV,防止他们在 FIPS 重新启动机器时恢复原来的硬盘分区表。此外要用 FIPS 分割被 double space, stacker, drivespace 压缩的磁盘时只能分割压缩盘的宿主盘。接下来用做好的 FIPS 启动盘启动机器;

③ 当用户启动 FIPS 的时候,程序可以让用户把硬盘上 boot 区及 root 区的备份写到

软盘上一个 ROOTBOOT.00x (其中的 x 代表一个从 0 到 9 的数字)的文件里。如果在使用 FIPS 时发生任何错误,用户可以从软盘启动并执行 RESTORRB 来恢复硬盘上原先的配置。

**注意:**如果用户不止一次地使用 FIPS 的话(这一般是不必要的,但是可能发生),程序会把一个以上的 ROOTBOOT 档案写到软碟上。RESTORRB 会让用户选择要回存哪个配置文件。RESTORRB.000 这个文件包含用户原先的配置。别把版本弄混淆了。

④ 运行 FIPS 将出现以下的画面:

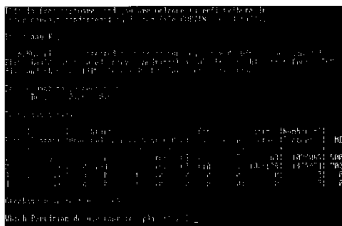


图 2.1 FIPS 运行画面

用户现在必须输入新的分区应该从哪个磁柱开始。使用左/右方向键来递增或递减磁柱的计数。剩下的分区以及新分区的大小会在程序中显示,所以选择正确的磁柱对用户来说并不困难。用户可以使用上/下方向键一次递增十个磁柱。准备好后,按下 enter 继续。在此之后,可以选择重新编辑分区表(这会让用户重新选择划分分区)或是继续。如果用户键入“c”,则 FIPS 将会计算改变过的启动分区,再检查一次并提示用户是否要继续。如果用户键入“y”,那么 FIPS 将会把这些改变写入硬盘并结束。重新启动机器,用 FDISK 查看新配置是否已经起作用了。

现在用户的准备工作已经全部完成,下面就可以开始进行系统的安装,笔者将分两节分别讲述如何安装 Slackware Linux 和 Redhat Linux。

## 2.4 Slackware Linux 的安装指南

不论是 Slackware 还是 Redhat,在安装时都需要先启动一个最简单的 Linux 系统,然后再进行系统的安装,这就像安装 Windows 95/NT 时,安装程序先向硬盘上拷贝一个最简单系统,然后运行一个 Windows 95/NT 的简单版本,再进行系统的安装。但是 Linux 不用向硬盘上拷贝任何东西,它只需要一到两张软盘就可以运行一个最简单的小系统,包括 tar、gzip、install、fdisk、ls、mount 和 setup 等安装必须的一些程序,然后就可以进行系统的安装过程。

对于 Slackware,安装盘有两张:rootdisk 和 bootdisk, bootdisk 就相当于一张启动盘,用来启动机器,展开系统内核,然后等待第二张 bootdisk 建立一个简单的系统,展开一些必

要的系统程序。

安装盘是以映像方式存放在光盘上的 `bootdsk.144`, `bootdsk.12`, `rootdsk` 这三个目录下,如同用 `hd-copy`, `dup` 等磁盘复制软件复制的映像文件一样,只不过他们是用 UNIX 下的 `dd` 来复制的。

`bootdisk` 既然是用来启动机器,它就必须包含有尽可能多的硬件驱动程序,否则读者使用比较不通用的硬件设备时,无法用这张制作好的 `bootdisk` 成功启动机器,就更不用说执行进一步的系统安装。这就是为何用户必须仔细选择适当的 `bootdisk` 映像文件来制作 `bootdisk` 盘的原因。

用户可以在光盘上的 `bootdsk.144` 下看到下列文件:

```
Volume in drive E is slakware32
Volume Serial Number is 9791-E14D
Directory of E:\BOOTDSKS.144

.                < DIR >          03-06-97  20:38  .
..               < DIR >          03-16-97   1:40  ..
00 _ INDEX .TXT      1,369  03-06-97  20:44  00 _ INDEX .TXT
00 _ TRANS .TBL      2,871  03-17-97  13:18  00 _ TRANS .TBL
7000FAST .S          575,488  03-04-97  17:15  7000FAST .S
ADVANSYS .S          600,064  03-04-97  17:15  ADVANSYS .S
AHA152X .S           583,680  03-04-97  17:15  AHA152X .S
AHA1542 .S           577,536  03-04-97  17:15  AHA1542 .S
AHA1740 .S           574,464  03-04-97  17:15  AHA1740 .S
AHA2X4X .S           589,824  03-04-97  17:15  AHA2X4X .S
AM53C974 .S          579,584  03-04-97  17:15  AM53C974 .S
AZTECH .I           535,552  03-04-97  17:15  AZTECH .I
AZTECH .S           778,240  03-04-97  17:15  AZTECH .S
BARE .I             568,320  03-04-97  17:15  BARE .I
BAREAPM .I          570,368  03-04-97  17:15  BAREAPM .I
BAREPNP .I          579,584  03-04-97  17:15  BAREPNP .I
BUSLOGIC .S         598,016  03-04-97  17:15  BUSLOGIC .S
CDU31A .I           535,552  03-04-97  17:15  CDU31A .I
CDU31A .S           777,216  03-04-97  17:15  CDU31A .S
CDU535 .I           533,504  03-04-97  17:15  CDU535 .I
CDU535 .S           775,168  03-04-97  17:15  CDU535 .S
CM206 .I            534,528  03-04-97  17:15  CM206 .I
CM206 .S            778,240  03-04-97  17:15  CM206 .S
DTC3280 .S          577,536  03-04-97  17:15  DTC3280 .S
EATA _ DMAS         582,656  03-04-97  17:15  EATA _ DMA .S
EATA _ ISA .S       579,584  03-04-97  17:15  EATA _ ISA .S
EATA _ PIO .S       577,536  03-04-97  17:15  EATA _ PIO .S
FAT32 .I            570,368  03-04-97  17:15  FAT32 .I
FAT32 .S            774,144  03-04-97  17:15  FAT32 .S
FDOMAIN .S          577,536  03-04-97  17:15  FDOMAIN .S
```

|            |         |                  |          |       |             |
|------------|---------|------------------|----------|-------|-------------|
| GOLDSTAR   | I       | 530,432          | 03-04-97 | 17:15 | GOLDSTAR.I  |
| GOLDSTAR   | S       | 779,264          | 03-04-97 | 17:15 | GOLDSTAR.S  |
| HPFS       | I       | 532,480          | 03-04-97 | 17:15 | HPFS.I      |
| IDE-BAT    | < DIR > |                  | 03-06-97 | 20:38 | IDE-BAT     |
| IN2000     | S       | 579,584          | 03-04-97 | 17:15 | IN2000.S    |
| IOMEGA     | S       | 574,464          | 03-04-97 | 17:15 | IOMEGA.S    |
| MCD        | I       | 533,504          | 03-04-97 | 17:15 | MCD.I       |
| MCD        | S       | 776,192          | 03-04-97 | 17:15 | MCD.S       |
| MCDX       | I       | 533,504          | 03-04-97 | 17:15 | MCDX.I      |
| MCDX       | S       | 775,168          | 03-04-97 | 17:15 | MCDX.S      |
| N53C406A   | S       | 575,488          | 03-04-97 | 17:15 | N53C406A.S  |
| NET        | I       | 704,512          | 03-04-97 | 17:15 | NET.I       |
| N_5380     | S       | 580,608          | 03-04-97 | 17:15 | N_5380.S    |
| N_53C7XX   | S       | 594,944          | 03-04-97 | 17:15 | N_53C7XX.S  |
| OPTICS     | I       | 533,504          | 03-04-97 | 17:15 | OPTICS.I    |
| OPTICS     | S       | 777,216          | 03-04-97 | 17:15 | OPTICS.S    |
| PAS16      | S       | 578,560          | 03-04-97 | 17:15 | PAS16.S     |
| QLOG_FAS   | S       | 574,464          | 03-04-97 | 17:15 | QLOG_FAS.S  |
| QLOG_ISP   | S       | 588,800          | 03-04-97 | 17:15 | QLOG_ISP.S  |
| RAWRITE    | EXE     | 36,064           | 03-04-97 | 17:15 | RAWRITE.EXE |
| README     | TXT     | 9,816            | 03-04-97 | 17:15 | README.TXT  |
| SANYO      | I       | 532,480          | 03-04-97 | 17:15 | SANYO.I     |
| SANYO      | S       | 776,192          | 03-04-97 | 17:15 | SANYO.S     |
| SBPCD      | I       | 548,864          | 03-04-97 | 17:15 | SBPCD.I     |
| SBPCD      | S       | 789,504          | 03-04-97 | 17:15 | SBPCD.S     |
| SCSI-BAT   | < DIR > |                  | 03-06-97 | 20:38 | SCSI-BAT    |
| SCSI       | S       | 772,096          | 03-04-97 | 17:15 | SCSI.S      |
| SCSINET    | S       | 939,008          | 03-04-97 | 17:15 | SCSINET.S   |
| SCSIPNP    | S       | 783,360          | 03-04-97 | 17:15 | SCSIPNP.S   |
| SEAGATE    | S       | 575,488          | 03-04-97 | 17:15 | SEAGATE.S   |
| TRANTOR    | S       | 577,536          | 03-04-97 | 17:15 | TRANTOR.S   |
| ULTRASTR   | S       | 575,488          | 03-04-97 | 17:15 | ULTRASTR.S  |
| USTOR14F   | S       | 578,560          | 03-04-97 | 17:15 | USTOR14F.S  |
| WHICH      | ONE     | 5,749            | 03-04-97 | 17:15 | WHICH.ONE   |
| XT         | I       | 532,480          | 03-04-97 | 17:15 | XT.I        |
| 61 file(s) |         | 35,043,901 bytes |          |       |             |
| 4 dir(s)   |         | 0 bytes free     |          |       |             |

其中 \*.i 文件是 IDE 接口的 bootdisk, \*.s 文件是选择 SCSI 接口的 bootdisk, RAWRITE.EXE 是用来复制 bootdisk 到软盘上的程序, IDE-BAT/SCSI-BAT 目录下存放的是如何用 RAWRITE 复制磁盘的批处理文件。WHICH.ONE 文件描述了如何选择各种安装盘,列表如下:

表 2.2 安装盘的种类

| Slackware 的安装盘                                                                                                    | IDE                                                         | SCSI                  | MF   |
|-------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------|-----------------------|------|
| 硬盘                                                                                                                | bare.i                                                      | 使用 SCSI 控制器的 bootdisk | xt.i |
| SCSI 光驱                                                                                                           | 使用 SCSI 控制器的 bootdisk                                       | 使用 SCSI 控制器的 bootdisk |      |
| IDE/ATAPI 光驱                                                                                                      | bare.i                                                      | 使用 SCSI 控制器的 bootdisk |      |
| Aztech, Orchid,<br>Okano, Wearnes,<br>Conrad, CyCDROM<br>non-IDE 光驱                                               | aztech.i                                                    | aztech.s              |      |
| Sony CDU31a,<br>Sony CDU33a 光驱                                                                                    | edu31a.i                                                    | edu31a.s              |      |
| Sony CDU531,<br>Sony CDU535 光驱                                                                                    | edu535.i                                                    | edu535.s              |      |
| Philips/LMS cm206 光驱                                                                                              | cm206.i                                                     | cm206.s               |      |
| Goldstar R420 光驱                                                                                                  | goldstar.i                                                  | goldstar.s            |      |
| Mitsumi non-IDE<br>光驱                                                                                             | medx.i<br>med.i                                             | medx.s<br>med.i       |      |
| Optics Storage 8000 AT 光驱<br>("Dolphin")                                                                          | optics.i                                                    | optics.s              |      |
| Sanyo CDR-H94A 光驱                                                                                                 | sanyo.i                                                     | sanyo.s               |      |
| Matshushita,<br>Kotobuki,<br>Panasonic,<br>CreativeLabs<br>(Sound Blaster)<br>Longshine and<br>Teac non-IDE<br>光驱 | sbpcd.i                                                     | sbpcd.s               |      |
| NFS                                                                                                               | net.i                                                       | scsinet.s             |      |
| 磁带机                                                                                                               | bare (软驱接口磁带机)<br>SCSI 接口的磁带<br>机使用 SCSI 控制<br>器的 bootdisks | 使用 SCSI 控制器的 bootdisk | xt.i |

用户可以根据自己的硬件种类来选择相应的 bootdisk,对于一般个人用户可以选择 bare.i 或者 net.i(要使用网络安装的用户)。

选择完 bootdisk,让我们来看看 rootdisk,rootdisk 文件包含有一个最简单的 Linux 文件系统,以及在该文件系统下附带的一些基本工具程序,用来把 Linux 系统安装到硬盘。rootdisk 的映像文件存放在光盘下的 rootdsk 目录中,在该目录下有下列文件:

Volume in drive E is slackware32

Volume Serial Number is 9791-E14D

Directory of E:\ROOTDSKS

< DIR >

03-07-97 21:04

```

..          < DIR >          03-16-97  1:40  ..
00 _ INDEX  TXT              1,622 03-06-97 20:45 00 _ INDEX.TXT
00 _ TRANS  TBL              801 03-17-97 13:18 00 _ TRANS.TBL
COLOR       BAT              887 03-04-97 17:21 COLOR.BAT
COLOR       GZ               588,713 03-07-97 21:03 COLOR.GZ
PCMCIA      BAT              805 03-04-97 17:21 PCMCIA.BAT
PCMCIA      GZ               735,579 03-07-97 21:03 PCMCIA.GZ
RAWRITE     EXE              36,064 03-04-97 17:21 RAWRITE.EXE
README      2,155 03-04-97 17:21 README
README     UMS              1,051 03-04-97 17:21 README.UMS
README _ T APE              1,408 03-04-97 17:21 README _ T.APE
RESCUE      BAT              805 03-04-97 17:21 RESCUE.BAT
RESCUE      GZ               536,909 03-07-97 21:03 RESCUE.GZ
TAPE        BAT              803 03-04-97 17:21 TAPE.BAT
TAPE        GZ               533,511 03-07-97 21:03 TAPE.GZ
TEXT        BAT              886 03-04-97 17:21 TEXT.BAT
TEXT        GZ               546,036 03-07-97 21:03 TEXT.GZ
UMSDOS      BAT              888 03-04-97 17:21 UMSDOS.BAT
UMSDOS      GZ               589,661 03-07-97 21:04 UMSDOS.GZ

18 file(s)                3,578,584 bytes
2 dir(s)                  0 bytes free

```

其中 COLOR.GZ 是彩色文本菜单界面安装方式,这种安装方式是最常用的,但是它对击键非常敏感,不小心乱敲键盘就有可能出问题。TEXT.GZ 是字符界面的安装盘,可以用 shift + PageUp 来回溯前面的信息。PCMCIA.GZ 是使用 PCMCIA 网卡通过 NFS 方式向笔记本式计算机上安装 Linux 的安装盘。UMSDOS.GZ 在上文说过是用来安装到 MS-DOS 分区的安装盘。TAPE.GZ 是用来从磁带机安装 Linux 的安装盘。RESCUE.GZ 是在用户的 Linux 系统崩溃后用来修复系统的急救盘。

值得注意的是,不要自作聪明地用 gzip 将这些 .gz 文件解压缩,RAWRITE 必须使用 .gz 文件来进行 rootdisk 的制作,在早期的 Linux 安装中需要先 gzip -d 这些文件,现在已经不用了。系统内核将自动解压这些文件,除非用户需要挂上软盘作为根文件系统才需要解压这些文件。

安装磁盘的制作方法:选择适当的文件,准备一张格式化(Format)好的磁盘,之后利用 RAWRITE.EXE(在 DOS 下用)或 dd(在 UNIX 下用)把该文件写到磁盘上。在 MS-DOS 下具体操作步骤如下:

```
E: \ INSTALL> rawrite
```

```
RaWrite 1.3 - Write disk file to raw floppy diskette
```

```
Enter source file name: e: \ bootdisks.144 \ bare.i
```

```
Enter destination drive: a:
```

```
Please insert a formatted diskette into drive A: and press -ENTER- :
```

```
Number of sectors per track for this disk is 18
```

Writing image to drive A: . Press ^C to abort.

Track: 38 Head: 0 Sector: 7

Done.

将刚写好的那张盘写上 bootdisk 的字样, 以避免同别的软盘混淆起来。换一张软盘:

E: \INSTALL> rawrite

RaWrite 1.3 - Write disk file to raw floppy diskette

Enter source file name: e: \rootdsk\color.gz

Enter destination drive: a:

Please insert a formatted diskette into drive A: and press -ENTER- :

Number of sectors per track for this disk is 18

Writing image to drive A: . Press ^C to abort.

Track: 31 Head: 1 Sector: 16

Done.

E: \INSTALL>

将这张软盘做上 rootdisk 的标记, 等一会儿要用。如果用户要觉得麻烦的话, 直接执行相应的批处理程序也是可以的。

如果用户要在 Linux 下复制这两张软盘, 请用如下命令进行复制:

```
$ dd if=base.i of=/dev/fd0 obs=18k
```

```
$ dd if=color.gz of=/dev/fd0 obs=18k
```

这里假设用户的软驱为 /dev/fd0。

在复制完这两张软盘之后就可以开始系统的安装了。

将 bootdisk 插入软驱, 重新启动机器。稍等一会将出现下面的启动画面:

```
LILO
```

```
Welcome to the Slackware Linux (v. 3.4.0) bootkernel disk!
```

If you have any extra parameters to pass to the kernel, enter them at the prompt below after one of the valid configuration names (ramdisk, mount, drive2)

Here are some examples (and more can be found in the BOOTING file):

```
ramdisk hd=cyl,hds,secs
```

(Where "cyl", "hds", and "secs" are the number of cylinders, sectors, and heads on the drive. Most machines won't need this.)

In a pinch, you can boot your system with a command like:

```
mount root=/dev/hda1
```

On machines with low memory, you can use mount root=/dev/fd1 or

mount root=/dev/fd0 to install without a ramdisk. See LOWMEM.TXT for details

If you would rather load the root/install disk from your second floppy drive:

```
drive2 (or even this: ramdisk root=/dev/fd1)
```

DON'T SWITCH ANY DISKS YET! This prompt is just for entering extra parameters

If you don't need to enter any parameters, hit ENTER to continue.

Boot:

这时候,用户可以在这里输入一些特别的参数,例如硬盘参数、网卡参数等等,这是为万一 Linux 自己无法辨别相应的设备而准备的:

hd = 柱面数、磁头数、扇区数

指定硬盘几何参数。只在 IBM PS/1, ValuePoint, ThinkPad 这样的系统上需要。如果用户的硬盘有 683 个柱面、16 头、每道 32 扇区,那么输入:hd = 683,16,32

ether = 中断号, 断口地址 m[, 参数 1 到参数 8], 网卡名(eth0, eth1)

这是为辨别网卡而设置的,如:ether = 9,0x300,0xd0000,0xd4000,eth0  
各种光驱

aztcd = 端口地址[, magic\_number]

edu31a = 端口地址[, 中断号[, is\_pas\_card]]

cm206 = [端口地址][, 中断号]

gsed = 端口地址

mcd = 端口地址[, 中断号[, wait\_value]]

optcd = 端口地址

sbpcd = 端口地址, 类型

sjcd = 端口地址[, 中断号[, dma\_channel]]

sonycd535 = 端口地址[, 中断号]

tmc8xx = 端口地址, 中断

为无 BIOS 的 Future Domain 公司的 TMC-8XX 系列 SCSI 硬盘控制器指定端口地址和中断。如:

linux tmc8xx = 0xca000,5

注意:0x 前缀必须用于以 16 进制表示的数字,这一点对所有下面的选项均成立。

st0x = 内存地址, 中断

为无 BIOS 的 Seagate 公司的 ST02 系列硬盘控制器指定内存地址和中断。

t128 = 内存地址, 中断

为无 BIOS 的 Trantor 公司的 T128b 系列硬盘控制器指定内存地址和中断。

ncr5380 = 端口, 中断, dma

为普通 NCR5380 系列硬盘控制器指定端口、中断、和 DMA 通道。

aha152x = 端口, 中断, scsi\_id, l

为无 BIOS 的 ACL-6260 系列硬盘控制器指定端口, 中断, SCSI ID. 包括 Adaptec 1510, 152x 系列及 SCSI 声霸控制器。

mem = 内存大小

可以为 16 进制, 如 16M 内存可以写成 0x6000000 或 16384k 或 16M, 这是为了使用超过 64M 以上的内存而设的。

更具体的请查阅 The Linux BootPrompt HOWTO。

一般来说用户不用输入任何参数,直接按回车即可。下面将展开系统内核并且启动系统内核,进行系统检测,用户可以看到屏幕上有大量信息输出,可以用 Shift + Pageup 来浏览一下 Linux 已经检测到用户系统上的哪些硬件。接着启动画面会停住:

VFS: Insert ramdisk floppy and press Enter

这里提醒用户插入 rootdisk。取出 bootdisk, 插入 rootdisk 后按回车, 系统就开始展开 rootdisk, 将出现下面的画面:

Welcome to the Slackware Linux installation disk! (version 4.0.0)

##### IMPORTANT! READ THE INFORMATION BELOW CAREFULLY. #####

- You will need one or more partitions of type "Linux native" prepared. It is also recommended that you create a swap partition (type "Linux swap") prior to installation. Most users can use the Linux "fdisk" utility to create and tag the types of all these partitions. OS/2 Boot Manager users, however, should create their Linux partitions with OS/2 "fdisk", add the bootable (root) partition to the Boot i Manager menu, and then use the Linux "fdisk" to tag the partitions as type "Linux native".
- If you have 4 megabytes or less of RAM, you MUST activate a swap partition before running setup. After making the partition with fdisk, use: `mkswap /dev/<partition> <number of blocks>; swapon /dev/<partition>`
- Once you have prepared the disk partitions for Linux, and activated a swap partition if you need one, type "setup" to begin the installation process.
- If you want the install program to use monochrome displays, type: `TERM=vt100` before you start "setup".

You may now login as "root".

Slackware login:

请用户用 root 这个帐号登录, 这个帐号目前是没有密码的。登录后先做给 Linux 分区的工作, 运行的程序同 MS-DOS 下的程序一样都叫 fdisk, 但是使用方法却不一样。首先介绍一下下面要用到的一些名词:

表 2.3 设备及名称

| 设备           | 设备名       |
|--------------|-----------|
| 第一个软驱(A:)    | /dev/fd0  |
| 第二个软驱(B:)    | /dev/fd1  |
| 第一个硬盘(整个硬盘)  | /dev/hda  |
| 第一个硬盘 主分区 1  | /dev/hda1 |
| 第一个硬盘 主分区 2  | /dev/hda2 |
| 第一个硬盘 主分区 3  | /dev/hda3 |
| 第一个硬盘 主分区 4  | /dev/hda4 |
| 第一个硬盘 逻辑分区 1 | /dev/hda5 |
| 第一个硬盘 逻辑分区 2 | /dev/hda6 |
| ...          |           |
| 第二个硬盘(整个硬盘)  | /dev/hdb  |
| 第二个硬盘 主分区 1  | /dev/hdb1 |
| ...          |           |

续表

| 设备                 | 设备名       |
|--------------------|-----------|
| 第一个 SCSI 硬盘 (整个硬盘) | /dev/sda  |
| 第一个 SCSI 硬盘, 主分区 1 | /dev/sda1 |
| ...                |           |
| 第二个 SCSI 硬盘 (整个硬盘) | /dev/sdb  |
| 第二个 SCSI 硬盘, 主分区 1 | /dev/sdb1 |
| ...                |           |

下面就对用户要安装 Linux 的硬盘进行分区操作, 键入 fdisk 回车, 则 fdisk 将缺省地以 /dev/hda 作为工作硬盘。要对第二个硬盘进行操作, 键入 fdisk /dev/hdb 即可, 依此类推。工作画面如下:

```
# fdisk
```

```
Using /dev/hda as default device!
```

```
Command (m for help):
```

键入 m 就可以获得 fdisk 的命令帮助表:

表 2.4 参数及功能表

| 参数 | 功能                                                | 参数 | 功能                                               |
|----|---------------------------------------------------|----|--------------------------------------------------|
| a  | toggle a bootable flag<br>切换分区为可启动与不可启动。          | q  | quit without saving changes<br>离开本程序而不更改分区表内容。   |
| c  | toggle the dos compatibility flag<br>切换 DOS 兼容标志。 | t  | change a partition's system id<br>修改某一个分区的类型。    |
| d  | delete a partition<br>删除一个分区。                     | u  | change display/entry units<br>改变硬盘资料的显示方式。       |
| l  | list known partition types<br>列出本程序所支持的分区类型。      | v  | verify the partition table<br>校验分区表。             |
| m  | print this menu<br>输出本帮助菜单。                       | w  | write table to disk and exit<br>将更改后的分区表写入硬盘。    |
| n  | add a new partition<br>新增一个分区。                    | x  | extra functionality (experts only)<br>一些低级的编辑功能。 |
| p  | print the partition table<br>列出目前硬盘上的分区。          |    |                                                  |

下面首先讨论这样一个问题: 我们到底需要多少个分区? 硬盘交换区 (swap) 是一定要一个的, 其他的是不是都可以放在一个分区上呢, 答案是当然可以, 这对初学者来说也是最简单的一个选择, 但是对于那些要用 Linux 进行实际工作的用户来说这是远远不够

的,一定要分成好几个分区,原因主要有以下几条:

(1)安全性。所有文件都放在一个分区时,一旦分区损坏就会丢失所有数据,如果已经分割成好几个分区,只要重新安装那个分区就可以了。而且用户可以把一些重要的分区挂载成只读的,这对防止黑客破坏也是一个很有效的防范措施。

(2)在 UNIX 下,一旦根分区被塞满,那么系统的许多重要文件都将无法写入,很可能使整个系统崩溃,例如利用 mail bomb 来攻击的话,对于只有一个分区的用户来说,在网络速度快的情况下,十几分钟硬盘就被塞满了。如果将根分区同/home和/var、/tmp这些经常写入数据的目录分开,那么就可以避免这种情况的发生。

(3)对于使用 Linux 来作新闻组/BBS 服务器的用户来说,因为新闻组和 BBS 上的文件经常是数以万计的小文件,将他们放在一个单独的盘上对提升系统的性能是很有帮助的,并且可以将这个硬盘的簇设小一点,以节省硬盘空间。

一般说来,我们可以建立下面几个分区:

/ 根分区,这个分区不用太大,对于 Slackware 只要 40M 就可以启动 Linux 了。

/var 存放系统日志文件和用户邮件的地方,视情况而定,一般个人用户可以不用再划分这个分区。

/usr 这个分区存放系统应用软件,要设得大一点,一般以 300M 以上为宜。

/usr/local 如果经常安装新软件的话也可以多分一个分区。

/home 这个分区存放用户自己的文件,对于个人用户也可以不划分。

/tmp 这个分区用来存放临时文件,因为有些大软件在运行时有可能产生一些交换文件,一般设 50M 以上比较适合。

如果用户使用 Linux 当新闻组服务器,可以考虑将/var/spool/news 另分一个分区,如果是 BBS 服务器那就是/home/bbs(Firebird bbs),如果是 ftp 服务器就是/home/ftp。如果在安装时候没有分出来怎么办?很简单,再安装分区时,在相应的位置作一个连接就可以了。

接着让我们讨论一个经常被问到的问题,用户的交换分区应该分多大?一个经典的回答是设置等于用户内存大小的交换分区,有用户要问,我的内存有 64M,还需要交换文件吗?答案是需要,因为用户无法预测在同一时刻运行的所有应用程序会不会消耗 64M 以上的内存,如果为了节省磁盘空间而让程序出现 Memory exhausted 或 Bus error 之类的信息,那就是因小失大了,而且对于内存小于 16M 的用户,笔者建议是设立加上物理内存总共 32M 的交换分区比较适合。并且建议用户将交换分区分配在硬盘上尽可能靠前的位置,这样对提高交换的速度是很有帮助的(硬盘上越靠前,寻道时间越短)。

不过为了便于初学者的学习,下面的安装过程,还是以安装到一个分区为例讲解。

在进入 fdisk 后,先用 p 来显示目前硬盘的分区情况:

Command (m for help): p

Disk /dev/hda: 64 heads, 63 sectors, 612 cylinders

Units = cylinders of 4032 \* 512 bytes

| Device | Boot | Begin | Start | End | Blocks | Id | System |
|--------|------|-------|-------|-----|--------|----|--------|
|--------|------|-------|-------|-----|--------|----|--------|

```
/dev/hda1 * 1 1 254 512052 + 6 DOS 16-bit > = 32M
```

这表示硬盘上只有一个 DOS 分区,下面首先增加一个 swap 区:

Command (m for help): n

Command action

e extended

p primary partition (1-4)

p

Partition number (1-4): 2

First cylinder (255-612): 255

Last cylinder or + size or + sizeM or + sizek ([255]-612): + 32M

笔者使用的是 32M 的交换分区,所以输入 + 32M。现在再看一下分区表:

Command (m for help): p

Disk /dev/hda: 64 heads, 63 sectors, 612 cylinders

Units = cylinders of 4032 \* 512 bytes

| Device    | Boot | Begin | Start | End | Blocks   | Id | System             |
|-----------|------|-------|-------|-----|----------|----|--------------------|
| /dev/hda1 | *    | 1     | 1     | 254 | 512032 + | 6  | DOS 16-bit > = 32M |
| /dev/hda2 |      | 255   | 255   | 271 | 34272    | 83 | Linux native       |

但是 Linux 的 fdisk 对所有新增加的分区都缺省设成 Linux native 分区(类型为 0x83)。下面用 l 命令查看一下 Linux 支持的分区类型:

Command (m for help): l

|   |                    |    |                 |    |              |    |            |
|---|--------------------|----|-----------------|----|--------------|----|------------|
| 0 | Empty              | 9  | AIX bootable    | 80 | Old MINIX    | c7 | Syrinx     |
| 1 | DOS 12-bit FAT     | a  | OS/2 Boot Manag | 81 | Linux/MINIX  | db | CP/M       |
| 2 | XENIX root         | 40 | Venix 80286     | 82 | Linux swap   | e1 | DOS access |
| 3 | XENIX usr          | 51 | Novell?         | 83 | Linux native | e3 | DOS R/O    |
| 4 | DOS 16-bit < 32M   | 52 | Microport       | 93 | Amoeba       | f2 | DOS second |
| 5 | Extended           | 63 | GNU HURD        | 94 | Amoeba BBT   | ff | BBT        |
| 6 | DOS 16-bit > = 32M | 64 | Novell Netware  | a5 | BSD/386      |    |            |
| 7 | OS/2 HPFS          | 65 | Novell Netware  | b7 | BSDI fs      |    |            |
| 8 | AIX                | 75 | PC/IX           | b8 | BSDI swap    |    |            |

可见我们要把这个分区设定成 82 即 Linux 交换分区,使用 t 这个命令来更改 hda2 的类型为 82:

Command (m for help): t

Partition number (1-4): 2

Hex code (type l to list codes): 82

Changed system type of partition 2 to 82 (Linux swap)

下面让我们来增加 Linux 分区:

Command (m for help): n

Command action

e extended

p primary partition (1-4)

```

p
Partition number (1-4): 3
First cylinder (272-612): 272
Last cylinder or + size or + sizeM or + sizeK ([272]-612): 612

```

我们将剩下的所有空间都用来作为 Linux 分区,所以直接输入磁柱数目。i 现在的分区表是:

```
Command (m for help): p
```

```

Disk /dev/hda: 64 heads, 63 sectors, 612 cylinders
Units = cylinders of 4032 * 512 bytes

```

| Device    | Boot | Begin | Start | End | Blocks   | Id | System            |
|-----------|------|-------|-------|-----|----------|----|-------------------|
| /dev/hda1 | *    | 1     | 1     | 254 | 512032 + | 6  | DOS 16-bit >= 32M |
| /dev/hda2 |      | 255   | 255   | 271 | 34272    | 82 | Linux swap        |
| /dev/hda3 |      | 272   | 272   | 612 | 687456   | 83 | Linux native      |

确定一切无误,下面就要将分区表写入硬盘了,如果用户发现有错的话,直接输入 q 退出就可以了。写入命令是:

```
Command (m for help): w
```

写入硬盘后,fdisk 会建议用户重新启动,实际上除非出现错误代码,不然一般情况下是不用重新启动的。接下来就请用户在命令行输入“setup”命令,正式开始我们的安装过程了。

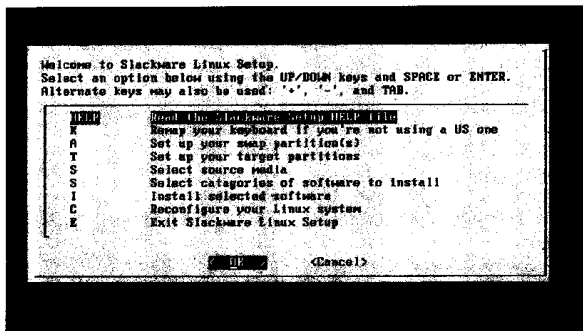


图 2.2 安装的主菜单

下面是安装主界面菜单的简单介绍:

- |              |                     |              |             |
|--------------|---------------------|--------------|-------------|
| (1)HELP      | 阅读在线帮助              | (6)SOURCE    | 选择源文件存放位置   |
| (2)KEYMAP    | 更改键盘映射              | (7)DISK SETS | 选择要安装的软件包   |
| (3)MAKE TAGS | 改变 Slackware 缺省安装选项 | (8)INSTALL   | 开始安装已选软件包   |
| (4)ADDSWAP   | 设定交换分区              | (9)CONFIGURE | 设定 Linux 系统 |
| (5)TARGET    | 设定要安装的目标分区          | (10)EXIT     | 结束设定        |

当用户选择 HELP 选项就可以看见下面的帮助画面,对于第一次安装的用户,可以阅读这个帮助文件,熟悉常用的热键。如空格键选定、上下键选择、Tab 和 Shift + Tab 键切换,回车确认等等。

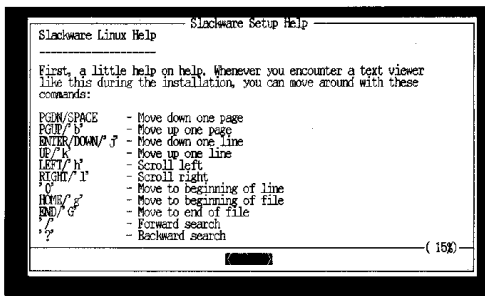


图 2.3 安装程序帮助画面

安装过程分以下几步:

#### (1)设定交换分区(ADDSWAP)

安装主界面菜单的第二项和第三项没有什么用处,用户可以不用理会。首先用户需要选择第四项“ADDSWAP”,设定系统的交换分区。然后系统会按顺序依次执行第四到第十项。由于上面用户已经使用 fdisk 增加了交换分区,现在选择这个选项,就会出现如图 2.4 所示菜单。

这表明安装程序已经探测到这个交换分区,下面将对它进行格式化。回车后出现:

Do you wish to install this as your swap partition?

如果用户确定分区没有错就选择 Yes 继续,安装程序会出现 MKSWAP WARNING,不要理睬,按 Enter 继续,然后系统就开始格式化交换分区,用户等待一段时间后格式化完毕,出现 USE MKSWAP 的问题,这是询问用户是否使用该交换分区,选择 Yes,然后系统将询问 ACTIVATE SWAP SPACE? 选择 Yes 后系统将挂上交换分区,然后是 SWAP SPACE CONFIGURED,用户应该设置为一启动就自动挂上交换分区。到这里为止,系统的交换分区的安装就算全部完成了。

#### (2)目标分区选择(TARGET)

下面将自动切入 TARGET 的设置,即让用户选择安装的目标分区。如图 2.5:

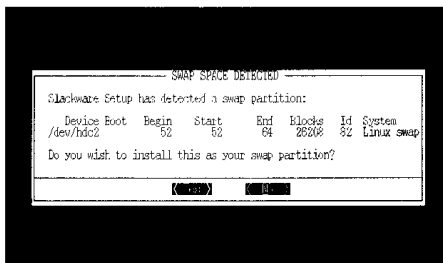


图 2.4 选择交换分区

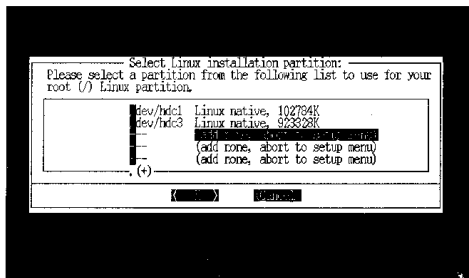


图 2.5 选择安装目的分区

用上下键选择用户要把系统安装到的那个分区,按回车确认,接下来的 **FORMAT PARTITION** (格式化分区)建议用户选择 **Check** 一项,这一选项会检查用户硬盘上是否有坏磁道,选择自动检测相对而言会更加保险一点,如果用户嫌慢的话就可以不选。下一项为 **SELECT INODE DENSITY** 这个选项是选择每个磁盘簇的大小。一般用户选择 **DEFAULT** 即可,如果用户有很多小文件就选 1024,要是硬盘大就选 4096(笔者推荐大家使用 4096 比较好。因为 1024 的 inode 表占的空间就很大了。约占系统空间的 1/8,也就是说 1G 的硬盘有 120M 刚格式化完就没有了,而 4096 的 inode 表约占 4%,两者差异,一目了然)。回车后系统就开始格式化这个目标分区,等待一段时间之后,格式化工作就完成了。如果用户划分了多个分区的话,这里就会提示用户输入要分配的目录,然后会出现下面的画面让用户选择是否要在 Linux 下自动挂上 MS-DOS 下的分区,见图 2.6。

用户选择 **Yes** 后就进入下面的菜单,让用户输入相应的分区名称,见图 2.7。

用户输入 **/dev/hda1** 后系统将出现下面的菜单,让用户输入在 Linux 下分配给这个

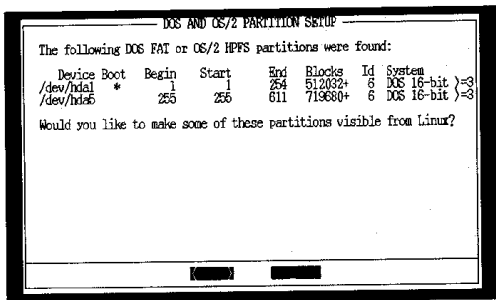


图 2.6 显示查找到的 DOS 或 OS/2, NT 分区

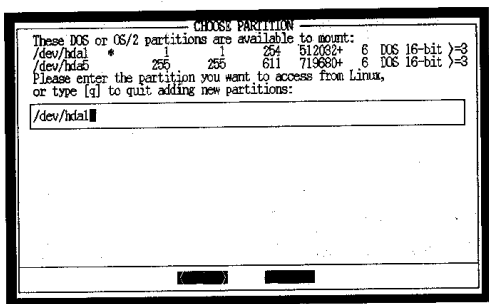


图 2.7 输入要自动挂载的分区

DOS 分区的一个目录,用来将该 DOS 分区挂载到这个目录下,见图 2.8。

这里笔者输入 /dose, 而后输入 /dev/hda1 (即 dos 下的 c 盘, 将被挂载到 /dose 这个目录下), 依次类推, 用户可以输入其他 DOS 分区, 输入完毕后请输入 q 就结束了目标分区选择这一步。

### (3) 选择源文件存放位置(SOURCE)

下面是选择从什么地方安装 Linux, 因为用户现在是从 CD-ROM 安装, 因此就选择第五项 Install from cdrom, 如果系统在这里无法识别 CD-ROM 的话, 那么用户只有将光盘上的 /slakware 这个目录全部拷贝到硬盘上再选择第一个选项 Install from a hard drive partition 从硬盘安装了(图 2.9)。

图 2.10 安装程序将询问用户是否继续进行下一步, 选择要安装的软件包。选择 Yes,

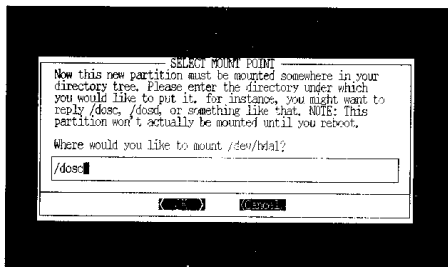


图 2.8 输入挂载的目录

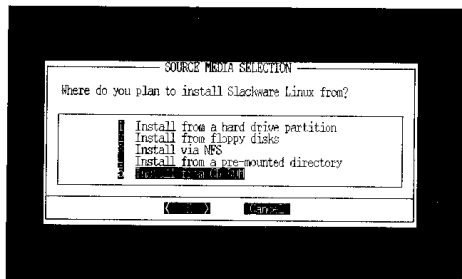


图 2.9 选择安装方式

就进入下一步。

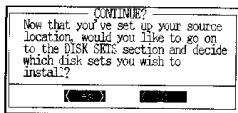


图 2.10 询问是否继续安装过程

#### (4) 选择要安装软件包 (DISK SETS)

如图 2.11, 现在要用用户选择要安装的软件包, 用户可以选择下面各系列的软件包:

- A 系统最基本的软件。
- AP 基本应用软件与文档, Man 文档, groff, ispell, joe, jed, jove, ghostscript, ac, bc, quota。

- D 程序开发环境, GCC/G++/ObjectiveC/Fortran-77 2.7.2.1, make (GNU and BSD), yacc and GNU bison, flex, 5.4.23 C libraries, glib, SVGAlib, ncurses, gcc (LISP), p2c, m4, perl, res。
- E GNU Emacs 19.34 功能强大的编辑器。
- F 一系列的 FAQ (常见问与答) 文件。
- K 2.0.30 Linux kernel 源码。
- N 网络软件, TCP/IP, UUCP, mailx, dip, PPP, deliver, cba, pine, BSD sendmail, Apache httpd, arena, lynx, cnews, nn, tin, trn, inn。
- T teTeX 即著名的排版软件。
- TXL Tel, Tk, TeX, 一种可视化编程工具。
- Y 游戏, The BSD games collection, Tetris for terminals, Koules, Lizards, and Sasteroids。
- CONTRIB Slackware 独有目录, 包括 ada 编译器和 一个多媒体创作工具。
- X 基本 X Window 系统。
- XAP X 应用软件: X11 ghostscript, libgr, seyon, workman, xfilemanager.xv 3.10a, GNU chess 和 xboard, xfm 1.3.2, i ghostview, gnuplot, xpaint, xfractint, fawn95-2, X 游戏。
- XD X11 开发工具库。
- XV xvview3.2pl-X11R6, XView 库, openwin。

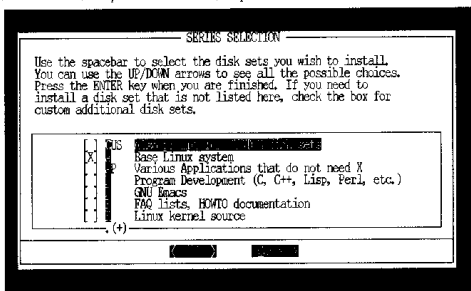


图 2.11 选择要安装的软件包

用户可以选择安装, 不过 A, AP, D, N, X 和 XAP 系列最好全部安装。如果用户要作正式用途的话, 一定要全部安装, 否则将来重新安装就比较麻烦。如果用户的 Linux 分区大于 400M 的话, 也可以选择全部安装。使用上下键和空格键选定后, 选择 OK 就会出现图 2.12 的画面,

选择 Yes 就进入下一步安装, 选择 No 就回到主菜单。

#### (5) 开始安装已选软件包 (INSTALL)

如图 2.13, 现在是让用户选择安装方式: 有 normal, menu, custom, path, expert, none 这几种, normal 方式将出现如图 2.14 的一个安装界面。

系统已经替用户选择了缺省的软件, 如果用户不满意可以手工更改, 选择 menu 方式

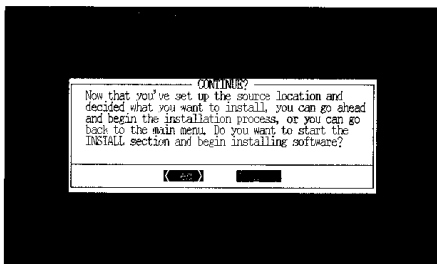


图 2.12 询问是否开始安装

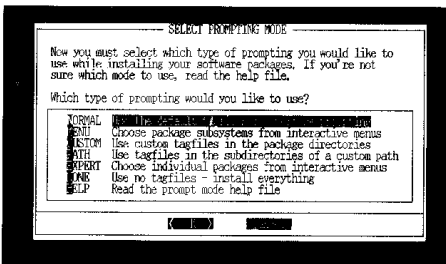


图 2.13 选择安装方式

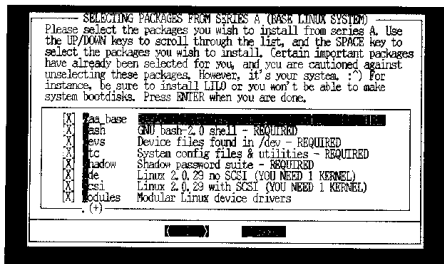


图 2.14 normal 方式

同 normal 方式差不多,只是系统必须的软件不包括在内,系统将自动安装那些软件,如图 2.15。

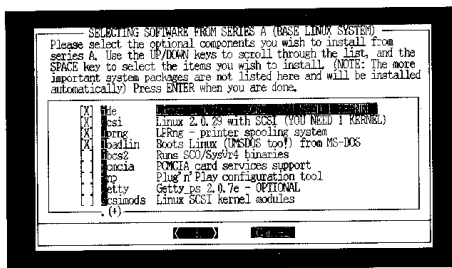


图 2.15 menu 方式

如果选择 custom 和 path 两种方式,系统都要求用户提供自己的 tag 文件,这对于大多数用户都是毫无必要的,如图 2.16。

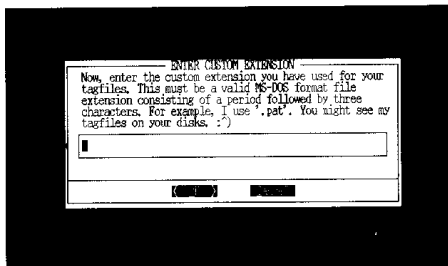


图 2.16 custom 安装方式

下面是笔者推荐大家使用的 none 方式,就是全部安装,不仅免去了选择的烦恼,还避免了安装不全的问题,当然这需要用户的硬盘有足够大的空间(如图 2.17)。

最后是专家模式,用户可以选择每一个软件包(如图 2.18)

用户选择 none 后,系统就开始安装 Linux 系统,这个过程取决于用户的光驱速度和硬盘速度,一般从 10~30 分钟不等,用户可以暂时离开一会,等待安装结束。在安装过程中,会有窗口显示现在正在安装的软件,安装顺序是按照字母顺序排列,所以根据观察已经安装到哪个系列就可以知道现在已经安装了多少。

(6) 设定 Linux 系统(CONFIGURE)

全部安装完毕后就进入系统设定菜单如图 2.19。

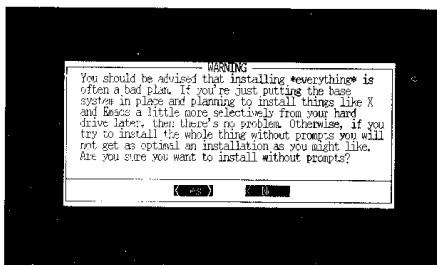


图 2.17 none 方式

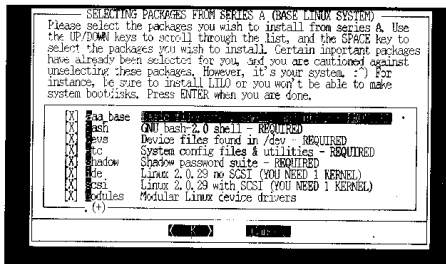


图 2.18 expert 安装方式

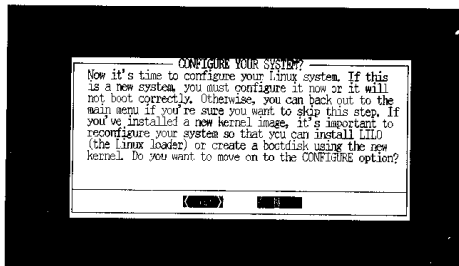


图 2.19 设置系统

选择 Yes, 则首先要安装系统内核, 如下图所示, 可以选择从 bootdisk.cdrom 还是从一张 MS-DOS 格式的软盘上拷贝系统内核, 一般选择 bootdisk, 这时候退出软驱中的 rootdisk, 插入 bootdisk, 选择 OK, 则系统开始将 bootdisk 上的 vmlinuz 拷贝到系统的根目录下。这就是新系统的内核(图 2.20)。

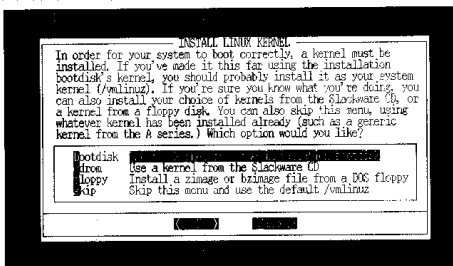


图 2.20 安装系统内核

然后系统会提示用户是否要做一张急救盘, 这样在系统出现问题时可以使用这张盘来启动系统, 如果用户需要做的就选择 Yes(图 2.21)。

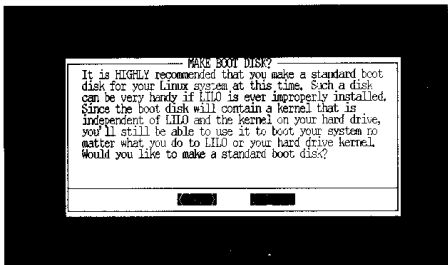


图 2.21 制作急救盘

接下来系统会提示用户插入一张空白软盘, 插入软盘后选择 Yes, 系统就格式化这张软盘, 并且拷贝一个最简单的系统到这张软盘上。制作这张启动盘并不妨碍用户安装 LILO(多重引导系统)。这张软盘在系统出现问题的时候可以用来启动机器做修复工作(图 2.22)。

然后系统会让用户设定调制解调器如图 2.23。

如果用户准备使用调制解调器, 选择 “Yes” 并按回车键(图 2.24)。

选择正确的端口。

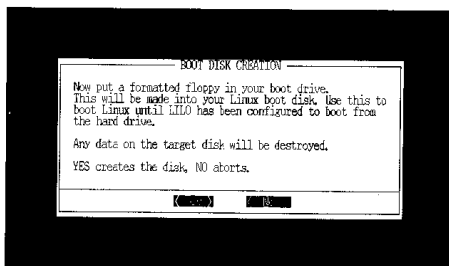


图 2.22 开始制作启动盘

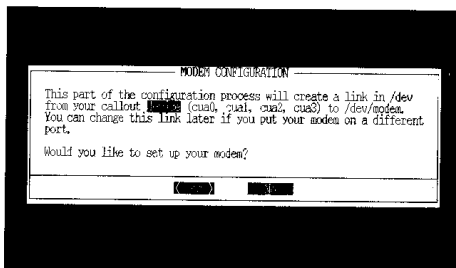


图 2.23 设置调制解调器

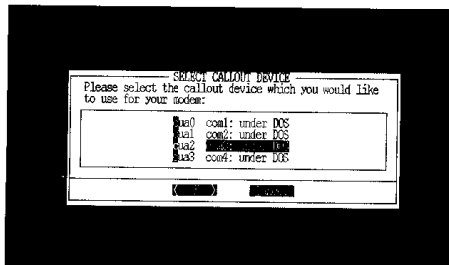


图 2.24 选择 MODEM 使用端口

下面是配置鼠标(图 2.25)。

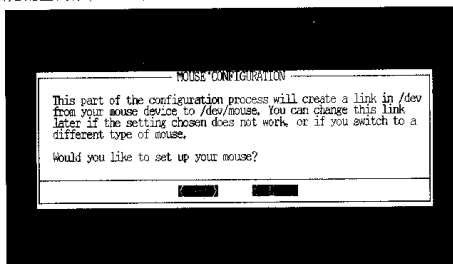


图 2.25 开始设置鼠标

选择“Yes”并按回车键(图 2.26)。

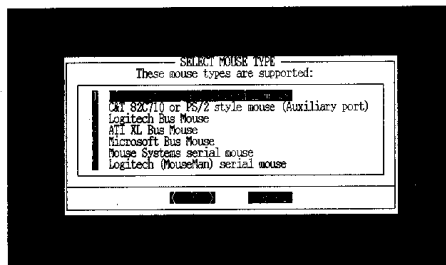


图 2.26 选择鼠标类型

如果是一般的两键鼠标就请用户选择选择“1 Microsoft compatible serial mouse”。如果是真三键鼠标就请用户选择选择“6 Mouse systems serial mouse”。如果是 PS/2 鼠标就请用户选择“2 C&T82c710 or PS/2 style mouse( Auxiliary port)”(图 2.27)。

根据用户鼠标连接到哪个端口来选择端口设备，一般是选择“ttyS0 COM1: under DOS”。选择“OK”并按回车键。

选择字体，用来选择文本模式下的屏幕字体，一般不用选择，否则看起来屏幕会乱乱的。选择“No”并按回车键(图 2.28)。

如果有调制解调器下面就会让用户设定调制解调器的速度：16540 串口选择 9600。16550 串口选择一个相应的速度。选择“OK”并按回车键。

下面就是非常重要的多重引导系统 LILO 的安装过程了：

LILO 是一个能在系统启动时用来选择 Linux, MS-DOS, 或其他操作系统之一的引导

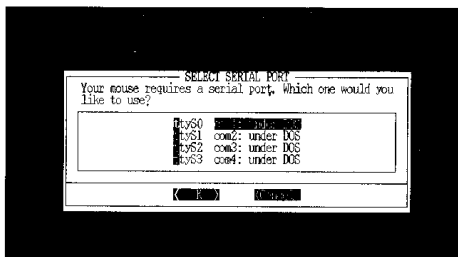


图 2.27 选择鼠标端口

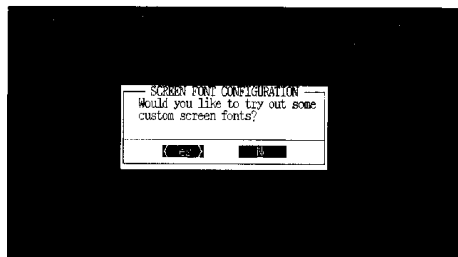


图 2.28 选择屏幕字体

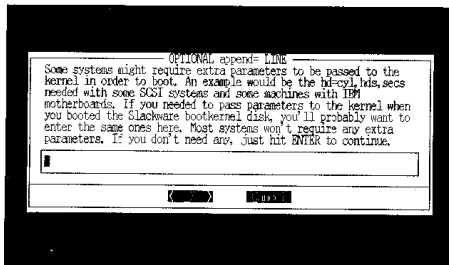


图 2.29 设置 LILO 参数

加载程序(除非用户使用 OS/2, 应该已经安装了 boot manager。如果这样, 用户可以跳过以下部分)(图 2.29)。

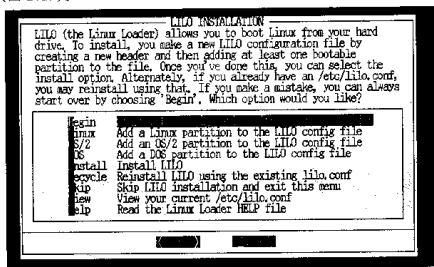


图 2.30 安装 LILO

首先选择“Begin Start LILO configuration with new LILO header”(图 2.30)接下来会出现下面的画面让用户输入一些特别的参数,具体解释可以参照前面的章节。用户一般不用输入任何参数,直接选择“OK”并按回车键(图 2.31)。

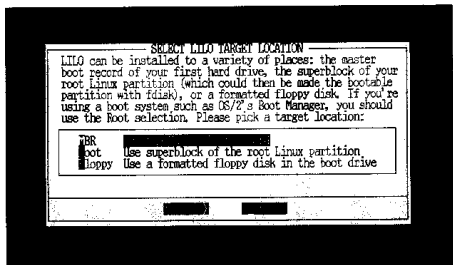


图 2.31 选择安装 LILO 的目的地

这个菜单让用户选择把 LILO 安装到什么地方, MBR 是第一个硬盘的主引导区(master boot record)root 是安装到可以启动的 Linux 分区的 Superblock 上,而 floppy 是安装到一张软盘上,即前面做的启动盘,用户一般应该选择 MBR,然后选择“OK”并按回车键(图 2.32)。

让用户选择在启动 LILO 时的等待时间,这是留给用户选择引导哪种操作系统的时间。一般选择“5 seconds”就可以了,选择“OK”并按回车键。

下面选择是先启动 Linux 还是启动 DOS,--般选择先启动 dos:

- (1)选择“DOS Add a DOS partition to the LILO config file”;
- (2)选择“OK”并按回车键;

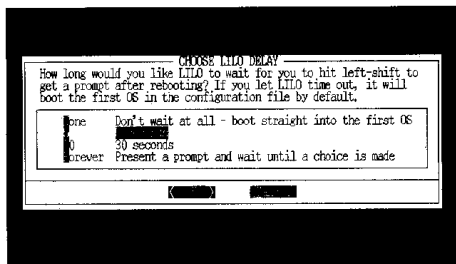


图 2.32 选择 LILO 延迟时间

(3) 输入 DOS 启动分区的设备名 (如 /dev/hda1, 见图 2.33), 选择 “OK” 并按回车键;

(4) 输入一个代号用来启动 dos, 选择 “OK” 并按回车键;

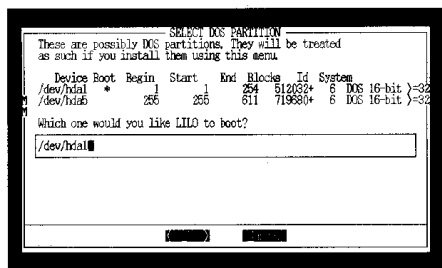


图 2.33 加入 DOS 启动分区

(5) 选择 “Linux Add a Linux partition to the LILO boot config file”;

(6) 选择 “OK” 并按回车键;

(7) 输入 LINUX 启动分区的设备名 (如 /dev/hde1, 见图 2.34), 选择 “OK” 并按回车键;

(8) 输入一个代号用来启动 Linux (如 Linux) 选择 “OK” 并按回车键 (如图 2.35);

(9) 然后用户可以选择 view 来查看目前的 LILO.conf 的设置 (图 2.36);

如果都正确就选择 “Install LILO” 选择 “OK” 并按回车键, 将 LILO 安装到硬盘上。

下面是让用户设定网络, 用户应该已经记录下下面的信息:

Host name      节点名  
Domain name    域名

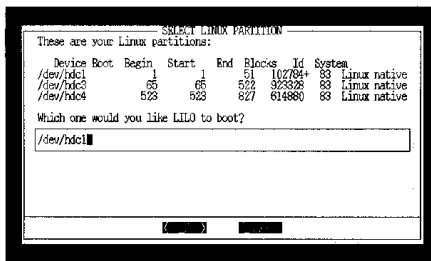


图 2.34 加入 LINUX 启动分区

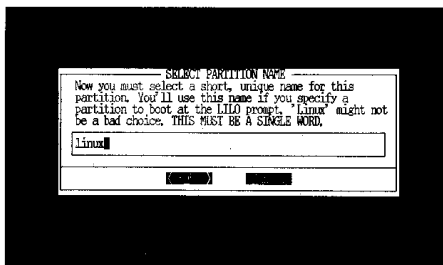


图 2.35 加入 Linux 启动代号

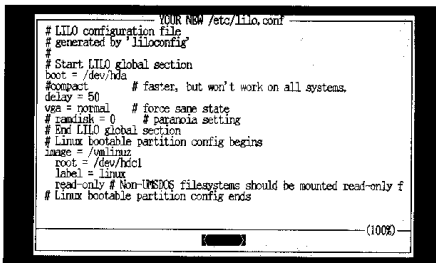


图 2.36 观察目前 LIL0 配置

TCP/IP address IP      IP 地址  
Address of default gateway      路由地址  
Subnet mask      网络掩码  
Address of name server      名字服务器地址

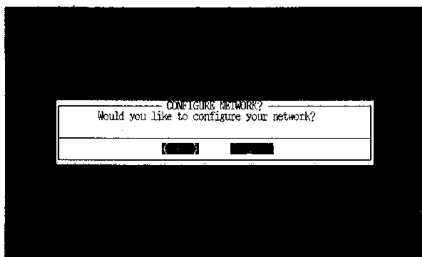


图 2.37 开始配置网络

(10) 选择 “Yes” 并按回车键(图 2.37);

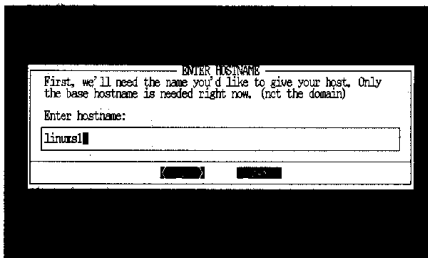


图 2.38 输入节点名

(11) 输入节点名, 如 linuxsl 选择 “OK” 并按回车键(图 2.38);

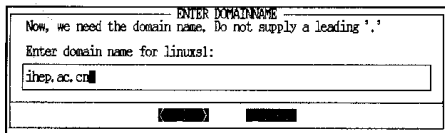


图 2.39 输入域名

(12) 输入域名, 如 ihp.ac.cn 选择 “OK” 并按回车键(图 2.39);

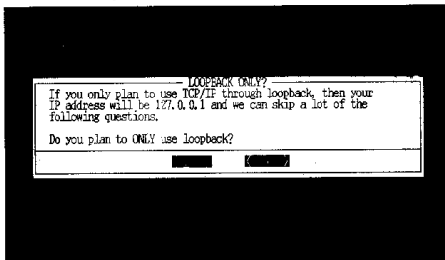


图 2.40 选择是否只使用 loopback

(13) 选择是否只使用 loop back, 这种模式是只在本机使用网络, 不同外界相联, 因而即使没有网卡也能工作。如果用户需要安装网络, 就选择 “No” 并按回车(图 2.40)。

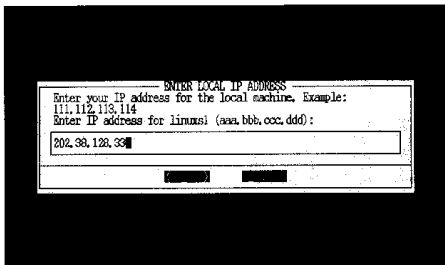


图 2.41 输入 IP 地址

(14) 让用户输入 IP 地址, 如 202.38.128.33, 选择 “OK” 并按回车键(图 2.41);

(15) 输入路由地址, 如 202.38.128.62, 选择 “OK” 并按回车键(图 2.42);

(16) 输入网络掩码, 如 255.255.255.0, 选择 “OK” 并按回车键(图 2.43);

(17) 输入名字服务器的地址, 如 202.38.128.58, 选择 “OK” 并按回车键(图 2.44);

(18) 网络配置完毕, 选择 “OK” 并按回车键(图 2.45);

(19) 设定使用什么样的 sendmail 配置, 一般选择 SMTP + BIND, 选择 “OK” 并按回车键(见图 2.46);

(20) 选择时区, 可以选择 Asia/PRC 或 Asia/Shanghai 选择 “OK” 并按回车键(图 2.47)。

现在整个系统已经安装完毕, 选择 “Exit Exit Slackware Linux Setup”, 选择 “OK” 并按回车键, 从驱动器中取出启动盘, 就可以按 “Ctrl-Alt-Del” 启动机器了。在启动机器后,

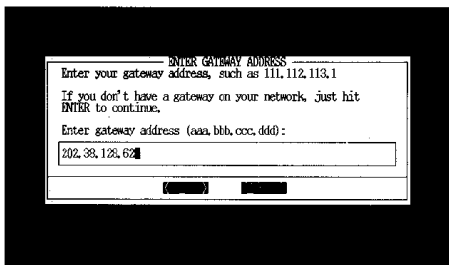


图 2.42 输入路由器地址

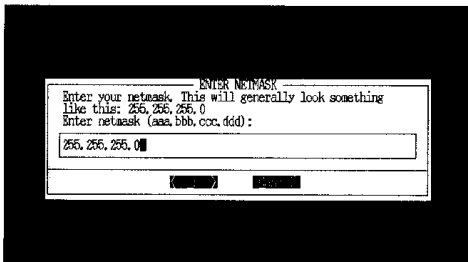


图 2.43 输入网络掩码

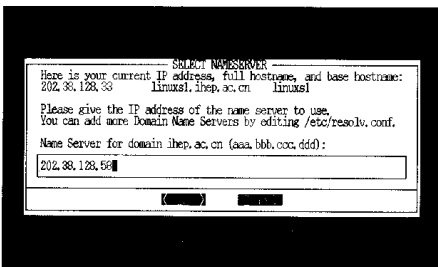


图 2.44 输入名字服务器的地址

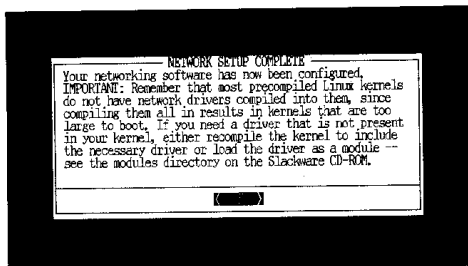


图 2.45 网络配置完毕

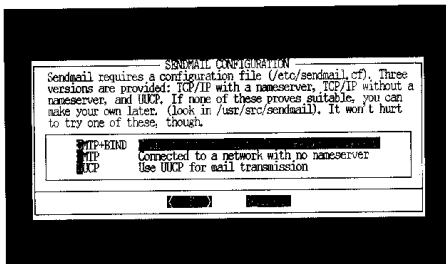


图 2.46 配置 sendmail

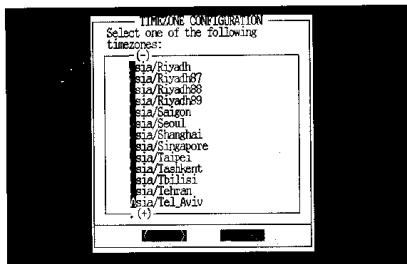


图 2.47 选择时区

如果不出意外的话,用户就可以看到在启动时会出现 LILO 的字样,按下 Shift 或 Ctrl 键就会出现 LILO BOOT: 的提示符,按 Tab 键就会列出用户可以选择启动哪个操作系统,输入 Linux 的代号,就会进入用户盼望已久的 Linux 系统了。如上所述:整个 Slackware 的安装过程就已经结束了。在下一章中我们将继续讨论如何配置整个系统,如配置 Xfree86,配置 /etc 下的各种配置文件等。

## 2.5 RedHat Linux 的安装指南

### 2.5.1 制作 Redhat 的启动盘

在用户 Redhat 的 CD 上会包括一个启动 RedHat 的启动盘映象文件(bootdisk)和扩充盘(supplemental disk)的映象文件,如果用户通过 CD-ROM 或 NFS 方式安装,就不需要扩充盘。而通过硬盘,FTP,PCMCIA 或 SMB FS 方式安装就需要扩充盘。Redhat 的启动盘只能用 3 寸盘来制作。

#### 1. 在 DOS 下制作启动盘

同 Slackware 一样,都是使用 RAWRITE 这个程序来做盘。

```
X: > cd \images (X 为 Redhat 源目录所在盘符)
```

```
X: > \dosutils\rawrite.exe
```

RAWRITE 询问文件名时回答启动盘文件名为 boot.img, 询问盘符时回答 A:。

如果要制作扩充盘,扩充盘的名称为 supp.img。做完盘别忘了将盘作上记号,标明是 boot 还是 supp 盘。

#### 2. 在 Linux 下制作启动盘

制作启动盘: `dd if=boot.img of=/dev/fd0 bs=1440k`

制作扩充盘: `dd if=supp.img of=/dev/fd0 bs=1440k`

#### 3. 从 CDROM 启动

如果用户的 RedHat 光盘上有 autoboot.bat 这个文件,那么就可以直接从光盘启动安装而不需要制作启动盘:

```
X: > cd \dosutils (X 为 Redhat 源目录所在盘符)
```

```
X: > autoboot.bat
```

### 2.5.2 使用启动盘来启动机器

将启动盘插入机器的驱动器中并启动机器。将出现如图 2.48 的画面:

直接按回车,如需要可以加上一些特别的参数如: boot: linux hdc = cdrom 等。

**注意:** RedHat 不像 Slackware, 它没有为某种特别的设备制作启动盘,而是通过可加载模块来实现的。

如果的机器上的设备在启动时没有被辨认出来,那么在后面的安装过程中可以通过动态加载可加载模块来使得 Linux 支持这些外设,另外在安装过程中的键盘使用如下:

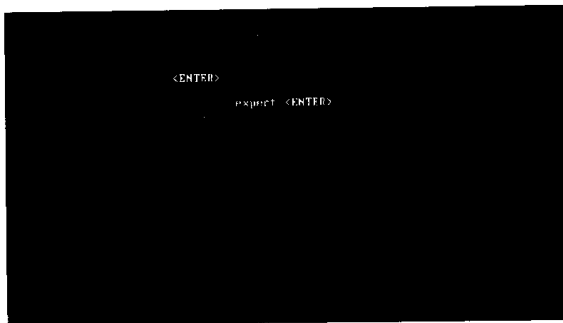


图 2.48 RedHat 启动画面

使用上下左右键和 Tab 以及 Alt + Tab 键来移动光标,例如要按下 OK 键就移动光标到 OK 键上按空格键,在选项中选择更详细的选项就移动光标到那儿之后按回车,选定选项使用空格,不选定只要再按一次空格键。按下 F12 就接受当前设置移动到下一个菜单。用户还可以用下列组合键在各虚拟终端之间切换:

- [Alt] + [F1] 安装界面
- [Alt] + [F2] shell 界面
- [Alt] + [F3] 安装日志 (安装信息)
- [Alt] + [F4] 系统日志 (核心信息)
- [Alt] + [F5] 其他信息

### 2.5.3 开始系统安装

#### 1. 选择显示模式

首先系统会询问用户是否使用一个彩色的显示器(图 2.49),缺省是 Yes,一般情况下,按回车就可以。

然后安装程序将会显示下面的欢迎画面(图 2.50)。

#### 2. 选择键盘

系统将询问用户使用什么样的键盘,其他语言的用户可以选择其他的键盘映射方式,选择之后的设置将自动加入系统设置,如果用户想更改设置的话,请在安装完 RedHat Linux 之后使用 `/usr/sbin/kbdconfig` 命令来设置。而中国用户一般都直接键入回车使用缺省设置 US 键盘(图 2.51)。

#### 3. 选择 PCMCIA 支持

如果用户使用 PCMCIA 方式来安装,这儿就要选择 Yes,系统将提示用户插入扩充

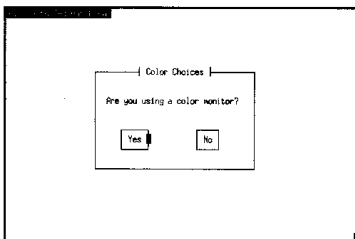


图 2.49 选择是否使用彩色终端

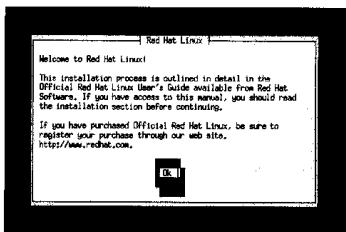


图 2.50 RedHat 欢迎画面

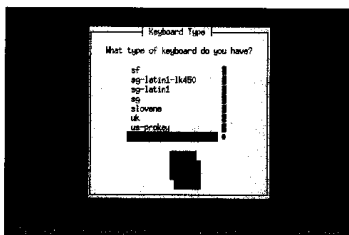


图 2.51 选择键盘类型

盘。一般用户选择 No 就可以继续安装(图 2.52)。

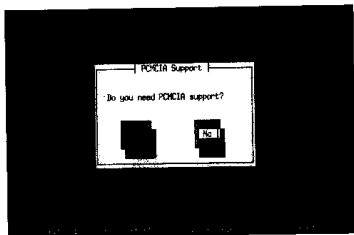


图 2.52 是否使用 PCMCIA

#### 4. 选择安装方式

现在系统将提示用户选择安装方式,如图 2.53。

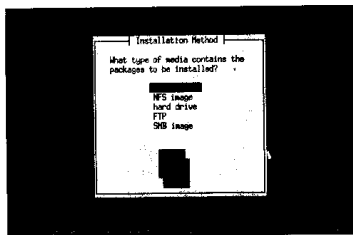


图 2.53 选择安装方式

RedHat 支持下列安装方式:

- CD-ROM 方式:如果用户有 RedHat 光盘,就不需要扩充盘。
- NFS 方式:从远端 NFS 服务器上安装不需要扩充盘。
- 硬盘方式:用户从本地硬盘安装,需要扩充盘。
- FTP 方式:用户从远端 FTP 服务器上安装,需要扩充盘。
- SAMBA 方式:从远端的 Windows NT 或 Windows 95 共享盘上安装,需要扩充盘。

如果用户选择除 CD-ROM 外的其他方式,需要扩充盘,这里系统将提示插入扩充盘,本章将主要以 CD-ROM 安装为主要安装方式进行讲解。

#### 5. 选择光驱类型

如图 2.54,系统将让用户选择光驱类型。

大多数 PC 机器上的光驱都是 IDE(ATAPI)CD-ROM,所以用户一般选择 IDE 就可以

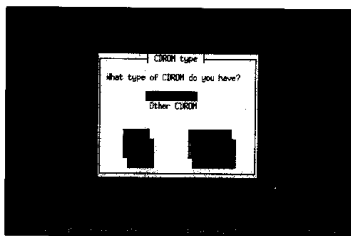


图 2.54 选择光驱类型

了,如果选择 SCSI 光驱,这两种方式系统都将自动测试光驱类型,不过有时候 SCSI 光驱需要用户输入一些参数。如果用户的光驱是其他类型的光驱,系统将提示用户选择光驱类型并加载相应的模块,如下图 2.55。

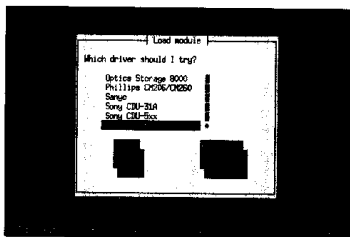


图 2.55 配置光驱

## 6. 选择 NFS 配置 (或 FTP, SAMBA)

如果用户不使用 NFS 安装模式、FTP 或 SAMBA 方式,就可以跳过这一节,NFS、FTP、SAMBA 方式安装只是在配置安装目录时有所不同,前面配置网络部分都是相同的。

选择网络驱动程序

用户选择 NFS、FTP 方式安装,都必须首先让系统安装网卡的驱动程序。

大多数情况下,系统将自动配置这些网卡(图 2.56),这种操作有时将把整个系统弄死,如果这种情况发生,请启动机器,在下次安装时手工输入相应的设置(图 2.57)。

配置 TCP/IP 网络

在安装程序配置好用户的网络设置之后,用户需要输入一些信息来配置 TCP/IP 网络,输入的信息只在安装时候使用,在安装之后,用户仍然需要配置系统网络设置。

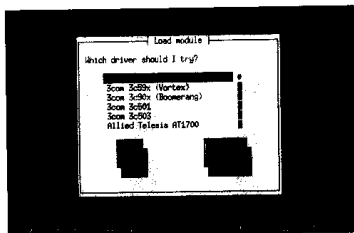


图 2.56 选择网卡

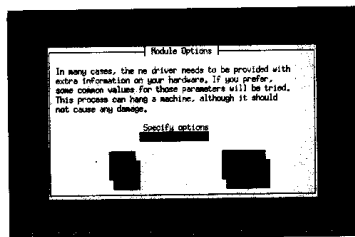


图 2.57 选择是自动探测还是手工设置网卡

首先用户需要输入下列信息：

|        |                  |
|--------|------------------|
| IP 地址： | 202.38.128.33    |
| 网络掩码：  | 255.255.255.0    |
| 缺省路由：  | 202.38.128.62    |
| 名字服务器： | 202.38.128.58    |
| 域名：    | ihcp.ac.cn       |
| 节点名：   | linux.ihcp.ac.cn |

如图 2.58：首先输入 IP 地址、网络掩码、缺省路由及名字服务器，接下来要输入域名和节点名，以及可以不输入的候补名字服务器地址(图 2.59)。

#### 输入 NFS 服务器配置

下面系统将提示用户输入 NFS 服务器的节点名和存放 Redhat 源文件的目录，例如：  
/mnt/cdrom或/pub/redhat/i386 等(用户可以通过在 UNIX 下使用 showmount -a 节点名来看这个目录的具体位置)(图 2.60)。

如果是 FTP 方式安装，输入 NFS 服务器的节点名和存放 Redhat 源文件的目录，如图 2.61。

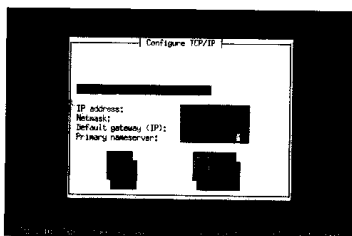


图 2.58 设定 TCP/IP

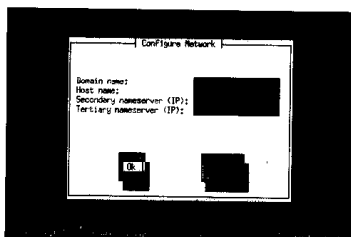


图 2.59 设定节点名和域名

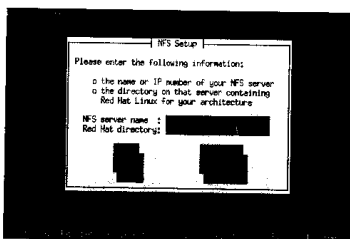


图 2.60 设定 NFS 安装

如果选择有帐号的 FTP 方式还需要输入用户名和密码,通过代理服务器时还要输入代理服务器的节点和端口。

如果是 SAMBA 方式安装,用户需要输入 SAMBA 服务器的节点名和存放 Redhat 源文

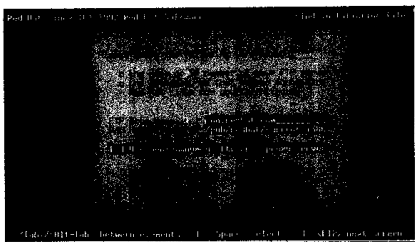


图 2.61 设定 FTP 安装

件的共享目录名,还有用户名和密码(缺省是无密码的 GUEST 方式)(图 2.62)。

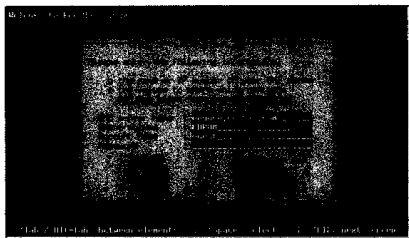


图 2.62 设定 SAMBA 安装

## 7. 选择安装或升级方式

如图 2.63 所示,系统将提示用户选择是安装还是升级,如果用户已经安装了 RedHat 系统,选择升级方式将不覆盖原有系统的各种配置文件,而是直接使用,这样用户可以在保留原有配置的情况下升级整个系统(所有改动将记录在/tmp/upgradelog 这个文件中)。对于第一次安装的用户来说当然选择 Install 方式来安装。

## 8. SCSI 支持

系统将询问是否要支持 SCSI 设备,如果用户选择 Yes,就需要选择相应的 SCSI 卡的类型,大多数 SCSI 设备系统都是可以自动检测的。一般用户不用选择这个选项。请选择 No,进入下一步安装。

## 9. 使用 FDISK 来给 RedHat 划分硬盘空间

这一部分请参照上一节 FDISK 的使用。不过 RedHat 体贴用户的一点是自动列出了

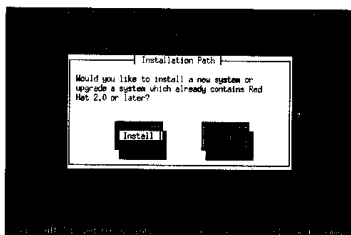


图 2.63 选择是安装还是升级

用户所有的硬盘让用户选择对哪个硬盘进行操作(图 2.64)。

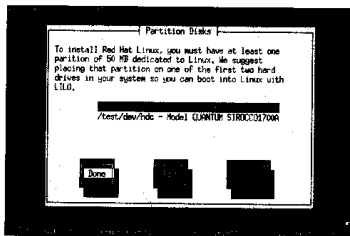


图 2.64 选择要划分的硬盘

划完硬盘,系统将搜寻交换分区,如果找到就提示用户是否初始化此分区,用户安装提示选择 OK 就可以了。

然后系统将让用户选择安装目的分区,即使用那个分区来启动 Redhat Linux,接下来就是选择系统启动时自动挂载的分区名,在这儿用户可以输入要挂载的分区(图 2.65)。

用户甚至可以选择系统启动时自动挂载 NFS 文件系统,选择 add NFS 就可以,如图 2.66。

接下来系统将提示用户选择要格式化的硬盘,对于初次安装的用户建议全部格式化,并且选择检测(check bad block)模式来格式化。然后系统将格式化这些分区,用户需要等待一定时间。系统将在屏幕上显示格式化的进度。

#### 10. 选择安装软件包

现在用户可以选择安装哪些软件包了,用户可以选择下列软件包,如图 2.67。



图 2.65 设置要挂载的分区

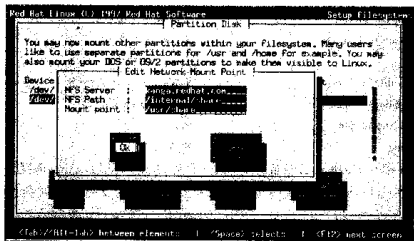


图 2.66 设定要挂载的 NFS 盘

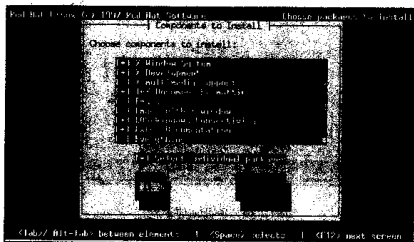


图 2.67 选择安装软件包

下面是相应的软件包列表：

[ ] C Development

C 编程包

[ ] Development Libraries

编程时使用的连接库

|                                    |                       |
|------------------------------------|-----------------------|
| [ ] C++ Development                | C++ 编程包               |
| [ ] Print Server                   | 打印服务器软件               |
| [ ] News Server                    | 新闻组服务器软件              |
| [ ] NFS Server                     | NFS 服务器软件             |
| [ ] Networked Workstation          | 网络支持软件                |
| [ ] Anonymous FTP/Copier Server    | 匿名 FTP 和 GOPHER 服务器软件 |
| [ ] Web Server                     | WWW 服务器软件             |
| [ ] Network Management Workstation | 网络管理软件包               |
| [ ] Dialup Workstation             | 拨号网络软件                |
| [ ] Game Machine                   | 游戏软件                  |
| [ ] Multimedia Machine             | 多媒体软件                 |
| [ ] X Window System                | X Window 系统           |
| [ ] X Development                  | X Window 编程软件         |
| [ ] X multimedia support           | X 多媒体支持软件             |
| [ ] TeX Document Formatting        | TeX 文字处理软件            |
| [ ] Emacs                          | Emacs 编辑器             |
| [ ] Emacs with X Window            | X Window 下的 EMACS     |
| [ ] DOS/Windows Connectivity       | DOS/windows 模拟器       |
| [ ] Extra Documentation            | 文档                    |
| [ ] Everything                     | 所有软件                  |

同 Slackware 相比, Redhat 已经替用户分好了类, 用户不用自己去选择一个又一个的软件包, 并且会在用户选择一个软件包后如没有安装相应的支持软件时会给出警告, 如图 2.68。

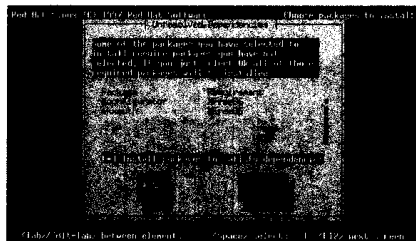


图 2.68 RedHat 的警告

当然, Redhat 也可以让用户自己选择详细的软件包目录。只要用户在上面选择:

Select individual packages.

这个选项可以自己选择安装每个软件, 如图 2.69。

用户在选择时可以使用 F1 键来观看软件包的说明。

最后系统将开始安装所有软件, 记录文件在 /tmp/install.log 中, 这个过程需要一定的时间, 不过系统将有一个非常直观的进度表以供用户观察安装过程。



#### 14. 配置打印机

如用户没有打印机,可跳过这一部分。

用户可以选择配置本地打印机(直接同系统相连的打印机)、远程打印机(连接在其他 UNIX 机器上的打印机)或网络打印机(通过 SMABA 方式共享出来的打印机)

用户还要输入以下的配置:

Queue: 用户产生的打印队列名称。

Spool Dir: 用来存放打印暂存文件的目录。

Type of Printer: 打印机的类型。

Paper Size: 打印机支持的纸张大小。

Resolution: 打印机的分辨率(DPI)。

Color: 如果是彩色打印机,须输入色彩深度。

本地打印机需要输入下列配置:

Printer Device: 打印机端口。

远程打印机需要输入下列配置:

Remote Host: 远程打印服务地址。

Remote Queue: 远程打印机的队列名称。

网络打印机需要输入下列配置:

LAN Manager Host: SAMBA 打印机的节点名。

LAN Manager IP(可选): SAMBA 打印机的 IP 地址。

Share Name: 共享的 SAMBA 打印机的名称。

Username: 登录 SAMBA 打印机使用的用户名:(Windows 服务器缺省是 GUEST, Samba 服务器缺省是 nobody)。

Password: 登录 SAMBA 打印机使用的密码,缺省是无。

设置完毕选择确认就将保留这些设置。

#### 15. 配置时区

选择 asia/shanghai 或 asia/PRC,安装后可以使用/usr/sbin/timeconfig 来修改。

#### 16. 设置 root 密码

用户需要输入两次密码并确认即可,下次系统启动时,可以用这个密码登录进入系统。

#### 17. 安装 LILO

如图 2.70, 选择安装 LILO 的位置,一般选择 MBR (Master Boot Record),然后同 Slackware 一样设置 LILO 即可。

要给 Linux 启动时加上固定参数,可以在图 2.71 中输入参数。

用户如果使用 LBA 硬盘,请选择 Liner mode 这个选项。

最后选择 OK 安装程序,将自动安装 LILO 到硬盘上。

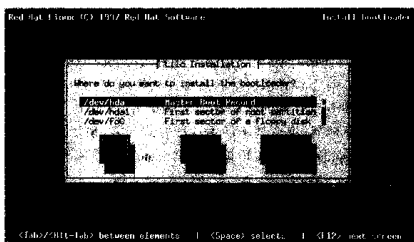


图 2.70 选择 LILO 安装目的地

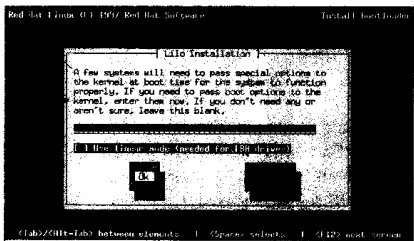


图 2.71 设定 LILO 参数

## 18. 重新启动

现在就可按照安装程序的提示重新启动机器,记住移去软驱中的软盘。  
以上是 RedHat 的安装过程。至于 Redhat 的配置将在下一章中介绍。

## 2.6 TurboLinux 中文版安装指南

启动 TurboLinux 安装程序有三种途径。

### 1. 直接用 CDROM 启动

如果读者的 BIOS 支持 CD-ROM 启动,请进入读者的机器的 CMOS 设置,并且在 Boot Sequence 中将 CDROM 设在最前面,查看读者的主板手册来改变启动顺序,如果读者使用的是支持光盘启动的 SCSI 接口的 CDROM,读者也可以用 SCSI 接口启动,读者需要先设置 SCSI 控制器。

## 2. 在 DOS 下用 Loadlin 启动

如果读者不能直接从 CDROM 启动,请读者使用光盘上的 LOADLIN 启动 TurboLinux 安装程序。在 Windows95/98 下读者需要重新启动并切换到 MS-DOS 方式下,然后请读者在 DOS 下切换当前的驱动器到读者的光盘驱动器上如 D:,或者 E:。(如果读者不能在 DOS 下访问读者的光盘驱动器,读者就不能用这种方式安装,请使用下面的软盘安装方式)然后进入子目录 dosutils/。现在请输入命令 autoboot 并回车,就可以开始安装 TurboLinux。

## 3. 利用软盘启动

如果上面的两种安装方式都不能工作,读者就必须利用软盘来安装 TurboLinux,读者需要制作的软盘数目在 1~3 张之间,这取决于读者的系统的硬件配置:

- 一张软盘 如果读者的系统上没有网卡、SCSI 卡或者 PCMCIA 卡,那么读者就只需要制作一张启动盘 (boot.img) 并用它启动安装。

这种安装途径支持下列硬件:

网卡: NE2000, 3c509, 3c59x, 3c90x, Intel EtherExpress Pro 10/100, SMC Ultra, SMC Ultra32, DEC Tulip。

SCSI 卡: Adaptec 2940 兼容 (aic7xxx), 所有 BusLogic 卡。

- 两张软盘 如果读者不使用 PCMCIA, 但是读者拥有不在上面的硬件列表之中的硬件设备,读者必须制作两张软盘来安装 TurboLinux:

启动盘 (boot.img) 特殊设备盘: (extrahw.img)。

- 三张软盘 如果读者使用 PCMCIA 卡,读者需要制作三张软盘 (boot.img, extrahw.img, supp.img)。如果读者要使用 FTP 或者 Samba 安装,其方式,读者也必须制作三张软盘。

制作软盘的过程非常简单: 将一张格式化好的空白软盘放入读者的小盘驱动器, 切换目录到第一张 TurboLinux CD 上的 dosutils/ 下运行 rawrite.exe。当询问写入的 Image 文件名的时候,请输入 .. \images \ boot.img, 当询问写入的驱动器的时候,输入 a 并回车。

**使用并口 IDE 驱动器的用户请注意:** 读者需要另外的一个磁盘映像文件: extra2.img, 写入一张磁盘, 以便安装程序能够使用读者的并口 IDE 光盘驱动器。

TurboLinux 安装程序

TurboLinux 的安装程序是由 consloe 上的一系列简洁、方便的中文菜单界面组成, 并且带有在线帮助, 这有助于用户进行一步一步的安装。

下面是安装程序中的热键列表:

Alt + F1 返回主菜单。

Alt + F2 进入 shell 环境, 此命令只有在进入第二阶段的安装过程之后才起作用。

Alt + F3 显示安装程序的测试信息, 供调试系统时使用。

Alt + F4 显示系统核心的信息。

Alt + F5 显示 mke2fs 的输出信息, 此程序格式化读者的 Linux 分区。

本窗口会询问用户是否使用彩色的显示模式 (图 2.72), 缺省时为 Yes, 一般情况下,

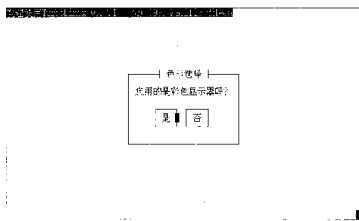


图 2.72 选择显示模式

按回车就可以。除了某些单色显示器以外,然后安装程序将会显示一个欢迎画面,读者只需要按回车跳过即可。

用户在此处设置键盘类型(图 2.73),其他非英语语言的用户也可选择其他的键盘映射方式,选择之后的设置将会自动加入系统设置,如果用户想更改设置的话,请在安装完 TurboLinux 之后使用 `/usr/sbin/kbdconfig` 命令来设置。而中国用户一般都直接键入回车使用缺省设置 US 键盘。



图 2.73 选择键盘布局

如果读者的系统上有 PCMCIA 卡(有时被称作 PC Card),选择‘是’。如果读者选择‘是’,安装程序会提醒读者插入特别设备驱动盘,不然读者将无法驱动读者的 PCMCIA 设备。

如果安装程序提醒读者需要特别的设备驱动程序(图 2.74),移去启动盘,并且插入特别设备驱动程序盘。读者将会看到一个状态条指示加载程序进度。在系统将特别设备驱动模块加载到内存中后,安装程序会自动去寻找这些设备驱动,这时候读者就可以取出读者的设备驱动盘。如果读者的系统不需要这张软盘,请读者选择‘忽略’来跳过这一步。

TurboProbe 是 TurboLinux 安装程序所特有的一部分, TurboProbe 可以自动检测读者系

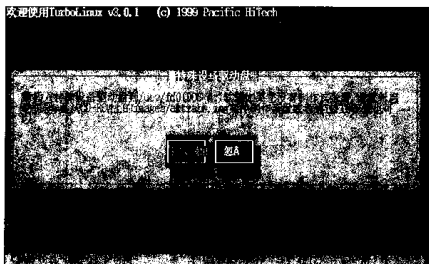


图 2.74 特别设备驱动盘



图 2.75 TurboProbe

统上安装的 ISA 设备、PCI 设备、SCSI 设备、网卡及并口 IDE 设备(图 2.75)。如果读者拥有一个并口 IDE 设备,还需要读者制作'extra2'软盘(用同上文相同的方式从 TurboLinux 光盘上的 images/extra2.img 制作)。如果读者没有加载特别设备驱动盘,那么 TurboProbe 的检测能力会受到影响,它只能检测到启动盘支持的那些硬件设备。请查看前面的'制作启动盘'一节的启动盘支持的硬件设备列表。检测过程将持续好几分钟,特别是系统拥有 SCSI 设备时(因为这时 TurboProbe 需要等待 SCSI 总线复位,SCSI 控制器检测所有的 SCSI 设备,比如读者系统上的 SCSI 卡是 Adaptec3940,需要 60 秒的时间来初始化 SCSI 卡),这时请读者耐心等待。读者可以按 Alt+F4 来查看检测的进程,按 Alt+F1 返回安装程序主画面。检测结束之后会显示所有检测到的设备列表。

**注意:**如果读者使用 PCMCIA 设备,就可以忽略这一节。系统在加载 PCMCIA 设备驱动程序后将自动启动 PCMCIA 管理器,将检测所有的 PCMCIA 设备并显示到屏幕上。

下面是五种安装 TurboLinux 的方式(图 2.76),其中四种我们将提供技术支持,除了'硬盘安装'方式由读者自己决定是否采用,其他四种安装方式是:

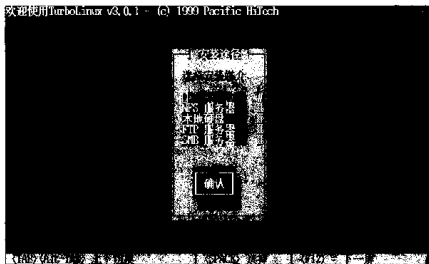


图 2.76 安装途径

- CD-ROM 安装

最常用的安装方式,如果读者选择了这种安装方式,安装程序会让读者插入第一张 TurboLinux 光盘,插入光盘后按回车。如果读者拥有的是一个 IDE/ATAPI 或 SCSI CD-ROM 驱动器,并且所有的硬件设备都能正常工作,那么安装程序将会自动找到 TurboLinux 光盘,并且开始第二阶段的安装。如果出现什么问题,或者读者拥有一个特别的 CDROM 驱动器,那么安装程序会提醒读者给出读者的 CDROM 驱动器的类型,在选择正确的驱动器类型之后,就会开始第二阶段的安装过程。

- NFS 方式安装

如果读者的机器在一个局域网中,并且该局域网上有另外一台服务器以 NFS 方式共享出一份 TurboLinux 的拷贝,那么读者就可以利用这台机器作为 NFS 服务器来安装 TurboLinux。在安装之前读者必须从系统管理员处获得读者系统的 TCP/IP 设置和 NFS 服务器的配置信息。选择这种安装方式将会弹出对话框网络设置。读者需要输入读者得到的配置信息来进行 NFS 安装。

- FTP 方式安装

如果读者拥有 Internet 的快速接入,那么读者可以通过 FTP 方式安装 TurboLinux。选择这个选项后,读者会被提示插入安装支持盘,移去软盘驱动器中的软盘并插入安装支持盘,按下回车键,然后系统会将其加载进内存,并且启动第二阶段的安装程序,进行 FTP 安装。

- Samba 方式安装

TurboLinux 安装光盘支持 Joliet 扩展和标准 UNIX Rock-Ridge 扩展。这标志着在 Windows 95/98/NT 下能够看见光盘上的长文件名。如果读者将读者的 TurboLinux 安装光盘放入读者的局域网的某台机器上,并且将其通过 windows 共享出来,读者就可以通过 Samba 方式来在该局域网上的任何一台机器上安装 TurboLinux 了。读者需要输入该 Windows 机器的名称和共享目录名,以及登录该 Windows 机器的用户名和密码,如果该共享目录没有密码保护,那么读者可以输入任意的用户名和密码。

选择安装程序方式

此处大多数用户应该选择缺省,如果读者是富有经验的用户,希望使用更加精密的安装方式,就请读者选择‘扩展方式’。



图 2.77 网络类型选项 (只限普通方式)

选择读者系统上使用的网络类型,缺省是只有 PPP/拨号(图 2.77)。如果读者的系统连接在一个局域网(以太网、令牌网或 Arcnet),那么就请读者选择局域网。

系统选项 (只限扩展方式)

本窗口同网络类型窗口很类似,但是允许读者选择系统启动时预先加载的设备驱动程序,PPP/拨号选项的作用同上节一样,其他的选项的作用如下:

**IP Aliasing** 如果读者的系统需要给一个或多个网络设备分配多个 IP 地址,读者就需要选择这个选项。

**Reverse ARP** 如果读者的系统需要提供反向地址查询,就需要选择这个选项。请读者确认后再选。

**VFAT 文件系统** 本模组允许读者在加载 Dos/Windows 分区时能够看到 Win95 的长文件名。

**Samba 文件系统** 本模组允许读者使用 smbmount 来加载 windows 服务器的共享卷

**NCP/Netware 文件系统** 本模组允许读者通过 IPX 来加载 Netware 的文件系统

**SysV 文件系统** 本模组允许读者加载 SysV 的文件系统。

**UFS 文件系统** 本模组允许读者加载 BSD UFS 文件系统。

**HPFS 文件系统** 本模组允许读者加载 OS/2 的 HPFS 文件系统。

**HFS 文件系统** 本模组允许读者加载 Macintosh 的 HFS 卷(硬盘、光盘、软盘)

**lp (打印机)** 本模组允许读者使用并口打印机,缺省为选中。

**Cyclades** 如果读者的系统上装有 cyclades 多串口卡,选中该选项。on。

**Riscom/8** 如果读者的系统上装有 Riscom/8 多串口卡,选中该选项。Riscom/8

**FTape** 如果读者的系统上装有基于软盘扩展器的磁带机,选中该选项。

**iBCS** 如果读者需要在 TurboLinux 上允许 SCO UNIX,FreeBSD,或者 Solaris 系统的可执行文件,那么就选中该选项。

**普通 SCSI** 如果读者要在 TurboLinux 上使用 CDR/CDRW 的设备,选中该选项。

## SCSI 检测

如果读者的系统上安装有 SCSI 设备,但是没有连接任何 SCSI 设备,读者将会看到一

条警告信息。



图 2.78 硬盘分区

现在屏幕上将显示读者计算机系统上安装的硬盘(图 2.78),使用上下键可加亮读者希望分区的硬盘,并且按 TAB 键直到光标停留在 FDISK 或者 CFDISK 上,按回车来选择。完成分区后返回‘硬盘分区的窗口’选择‘完成’来继续。

使用交换分区

下面的窗口让读者选择哪一个交换分区将被使用,以便安装 TurboLinux,并且要求格式化交换分区,这时系统缺省选择了检查坏块这个功能,用户一般应该选择继续,这时屏幕上会出现一个显示完成的进度条,如果不检查坏块,这一步将一闪而过。

选择根分区

现在读者将看到系统上所有可能的 Linux ext2 分区,选择其中的一个作为读者的根分区(将被作为/加载)。如果读者只有一个 Linux 分区,请选择 OK 继续。

设置文件系统

读者现在能够设置文件系统,所有包含 TurboLinux 能识别的文件系统的分区(包括 MS-DOS 或 Windows 分区)将出现在下一个对话框中。这使读者能够将这些分区设置为 TurboLinux 文件系统的不同部分,读者指派的分区的 TurboLinux 系统启动时自动加载。选择读者想要设置的分区并按回车,输入这些分区的载入点。例如如果/dev/hda1 是一个 DOS/Windows FAT 分区,那么输入/dosc, 这样在安装完成后,读者的系统将自动将/dev/hda1 加载为/dosc 目录。

现在读者也可以增加 NFS 的文件卷,按 Add NFS 键,如果读者还没有配置网络,安装程序会询问读者是否要现在配置,建议读者取消,在后面再配置网络,现在读者将要输入 NFS 服务器名、NFS 文件卷的路径、这个 NFS 卷的载入点,还有读者是否希望在启动时自动加载为可写还是只读(缺省为只读)。

在完成设置之后选择 OK, 进行下一步的安装过程  
格式化分区

读者现在可以选择读者要格式化的 Linux ext2 分区, 读者一定要格式化根(/)分区以及 /user 和 /var 分区(如果读者分有多个分区), 特别是在对老的 Linux 系统升级的时候, 如果不格式化, 那么有可能会造成版本间的冲突。

格式化时检查坏块将更安全, 但是会花费大量的时间。如果读者选择 'Ok', 读者的系统将会被格式化, 格式化硬盘需要一定的时间, 主要取决于读者分区的大小和读者的系统速度。

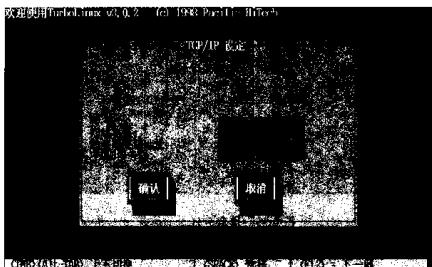


图 2.79 配置 TCP/IP 网络

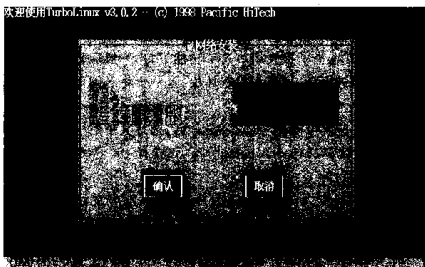


图 2.80 配置域名

如果读者没有选择 NFS 安装方式, 读者现在将需要配置网络。不然读者只需要保持目前的网络配置, 或者重新配置网络。如果读者没有在前面网络类型中选择局域网的选项, 读者将只需要输入节点名和域名, 以及名字服务器的地址。在这种情况下, 读者可以跳过这一步, 不然读者就需要配置 TCP/IP 网络(图 2.79)。

如果读者的局域网上有 BOOTP 服务器, 并且知道读者机器的网络硬件地址, 那么可

以选中“Use BOOTP”，此时不用输入其他设置就可以完成网络的配置。如果读者的 Bootp 服务器被正确配置，那么正确的 IP 信息会被传递以重新安装，读者将会看到一个新窗口，包含所有的节点名、域名、第二名字服务器（图 2.80），读者只需要按“OK”就可以完成网络配置。一般情况下读者需要手工输入 IP 信息、IP 地址、网络掩码、缺省路由和 DNS 名字服务器，然后选择“OK”即可。系统现在将通过 DNS 查询读者机器的节点名和域名。如果读者没有 DNS 服务器或者没有注册域名，那么读者只需要等待 DNS 查询失败返回，然后手工输入域名和节点名。

### 安装记录信息

现在安装程序将提醒读者将把所有的安装记录输出到文件 `/tmp/install.log` 中, 读者可以在安装完成后查看这个文件。



图 2.81 安装软件包

现在读者将首次使用 TurboPkg, TurboPkg 是 TurboLinux 的 RPM 软件包管理系统, 此程序在安装时间安装后有些细微的差别。安装时, 它将提供给读者一个预定义的安装类型(图 2.81), 包括所需硬盘大小, 读者也可以选择‘定制’安装, 手工选择需要安装的软件包。

如果读者使用 FTP 安装,读者将看到一个对话框让读者输入 FTP 服务器的一些信息:

**使用代理服务器** 如果读者需要一个代理服务器来访问 FTP 服务器, 选择“使用代理”, 读者将被提示输入代理服务器的设置, 读者可以向读者的系统管理员查询这些信息。

**FTP 服务器** 请输入 FTP 服务器的节点名或 IP 地址, 存放 TurboLinux 的目录, 读者也可以输入用户名和密码 (如果不是匿名登录, 缺省是匿名登录, 因此不用改变这些设置)。

**建立 FTP 连接** TurboPkg 现在将连接 FTP 服务器并下载一些文件来执行安装。

现在读者将看到 TurboPkg 欢迎画面,按回车继续。

安装 TurboLinux 最简单的方式就是选择一个预定义的安装类型,在读者选择安装类型时可按 F1 来查看该安装类型更多的附加信息。这种软件包的依赖关系将自动被处理,并自动计算出需要的硬盘工具,现在读者可以选择继续安装或者定制读者需要安装的软件包。

有经验的用户也许希望定制要安装的软件包,或者修改一个现有的安装类型,读者可以进入定制画面,这是一个树状结构,由软件包群组或单独软件包组成,如果读者是修改

现有的安装类型,那么只有该安装类型需要的软件包才被选中,如果读者选择定制安装,所有的软件包将被选中。用上下键来选择群组或软件包,按回车来选择/取消软件包群组或软件包。下面是一些重要的热键:

- F1 帮助。
- F2 展开/缩进目录树。
- F3 显示选中软件包的信息。
- F4 进行大小写不敏感的搜索。

为了正常工作,许多软件包要求读者的系统中安装了其他的软件包或库。例如,许多图形化的 TurboLinux 系统管理工具需要 `tel/tk` 和 `perl` 软件包。为了让读者的系统拥有所有需要的软件包, TurboLinux 在每次读者安装或删除软件包的时候须检查它们的依赖关系。

在读者选择完要安装的包之后,安装检查这些软件包的依赖关系。如果有的软件包需要读者没有安装的软件包,程序会列出这些未解决的依赖关系并让读者有机会解决。如果读者简单地按 `OK`,程序自动将需要的包增加到选择的包当中。在读者完成选择后,按 `OK` 继续, TurboPkg 将自动检查需要安装的软件包大小,并且如果读者硬盘剩余空间不够,将给出警告(注意:此特性在 FTP 或 Samba 安装时无效)。

现在安装程序开始安装 TurboLinux 系统,这将需要一定的时间,取决于读者系统的硬盘速度、光驱速率、CPU 频率,完全安装一般需要 20~30 分钟。所以读者现在可以开始阅读本手册,以便安装完成后立刻开始使用读者的 TurboLinux 系统。

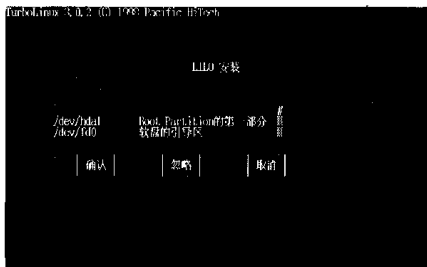


图 2.82 安装 LILO(Linux 启动管理器)

LILO 是一个能在系统启动时用来选择 LINUX、MS-DOS 或其他操作系统之一的引导加载程序。(除非用户使用 OS/2,应该已经安装了 boot manager。如果这样,用户可以跳过以下部分。)如果读者需要更特别的启动类型,建议读者使用商业软件 System Commander。

我们将此步骤移动到系统配置的最前面,这样如果读者在后面的系统配置中造成系统死锁(一般不会出现这种情况,除非读者的显示卡或者 ISA PNP 的设备是一些很特殊的设备),读者的系统仍然可以启动,而不必重新安装。

首先让读者选择把 LILO 安装到什么地方(图 2.82),MBR 是第一个硬盘的主引导区

(master boot record), ROOT 是安装到可以启动的 Linux 分区的 Superblock 上, 而 floppy 是安装到一张软盘上, 即前面做的启动盘, 用户一般应该选择 MBR, 然后选择“OK”并按回车键。

下一个窗口让读者输入需要在启动时传递给 LILO 的信息, 如果读者使用的是某些 SCSI 的驱动器, 读者还需要选中‘Linear’模式, 除非读者确证需要这个参数, 不然一般情况下建议读者不选中这个选项。最值得用户注意的是如果读者的系统拥有 64M 以上的内存, Intel x86 系列的 BIOS 有时会向操作系统报告最大只有 64M 内存, 这种情况下读者就需要通知系统核心读者的真实内存大小, 不然 Linux 将只能使用 64M 内存, 例如, 读者系统拥有 128M 内存, 读者就需要输入下面的参数:

```
mem = 127M
```

为什么我们使用 127M 而不是 128M? 因为某些系统保留一部分内存用来做 BIOS 或显示内存的 cache。所以我们保留 1M 内存给系统。如果读者输入比读者真实内存大的参数时将会造成一些很严重的结果, 如系统死机。现在读者选择‘OK’继续。

现在读者需要输入希望启动的分区, 读者的 Linux 分区将被自动配置为标志‘Linux’。读者的第一个 DOS 分区将被自动配置为标志‘DOS’, 目前还不能自动辨认 Win98/NT 的 FAT32 分区, 它将被显示为‘未知’, 读者需要手工输入。用方向键来上下移动, 使用 TAB 键来选中编辑按钮并按回车, 缺省的启动分区将是排在最前面。目前还不能选择缺省安装分区, 但是安装完成后读者可以手工修改。如果读者是安装前面章节 硬盘分区中所说的那样分区, 读者的 DOS/Windows 分区将被作为缺省启动分区, 后面将详细讲解这一点, 现在选中‘Ok’继续, 配置将被写入硬盘, 完成 LILO 的安装。

配置显示卡

现在安装程序将执行 TurboXCfg 来配置读者的显示卡(图 2.83)。几乎所有的显示卡都可以用 TurboXCfg 来自动检测, 目前已经能够检测大多数 AGP 显示卡和 PCI 模式的显示卡。



图 2.83 TurboXCfg 欢迎画面

如果读者选中自动检测, TurboXCfg 自动检测读者的显示卡, 并显示检测结果(图 2.84), 按回车 继续, 如果读者需要手工配置读者的显示卡, 选中“否”。

如果读者选中手工配置显卡, 安装程序将给出一个庞大的显卡清单, 以便读者能够手工选择读者的显卡。用上下键加亮读者的显卡, 并且按回车选中。读者必须输入显存大小 (单位为 K, 例如 2M 为 2048K), 有些显卡还需要读者输入时钟芯片和 RAMDAC 的型号。如果读者的确知道这些参数, 就可以输入, 不然使用缺省设置即可。

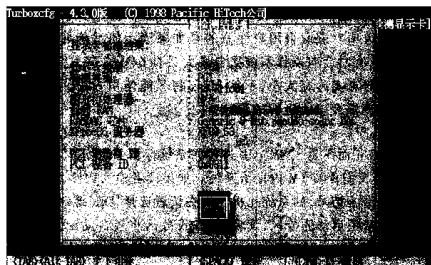


图 2.84 TurboXCFG 检测结果

现在读者正在以系统安装模式运行 TurboXCFG, 这种模式将一步一步地帮助读者安装 X Window (图 2.85)。如果是安装完成后运行则是分离的菜单项, 如果读者某一步骤失败, 读者也会回到主菜单画面来重新开始。



图 2.85 TurboXCFG 配置键盘

此处读者可以选择不同的 Console 键盘布局, 读者选中的键盘布局是在 Linux 控制台上运行的, 而不是在 X Window 下, 基于安装程序的语言版本, TurboXCFG 将自动加亮一个缺省的键盘布局, 对于 TurboLinux 中文版用户, 那么 'us' 键盘布局是缺省的键盘布局, 请确认读者选中了正确的键盘布局, 并按回车继续。

现在须读者输入键盘型号, TurboXCFG 将自动加亮一个缺省设置, 请确认读者选中了正确的键盘型号, 并按回车继续。大多数用户应该是 'Generic 101-key' 键盘, 某些欧洲的

计算机需要 102 或 104 键盘,而日本用户一般是 106 键盘。

下面请读者输入键盘布局,一般是依照键盘语种选择。中文版用户按回车选择美国英语即可。

#### TurboXCfg - 配置鼠标

TurboXCfg 将自动探测读者系统上的鼠标,如果它检测到一个 PS/2 鼠标或者 Bus 鼠标,将自动加亮相应的选项,如果没有鼠标或读者使用的是串行鼠标,将缺省为‘No Mouse’。这时需要读者手工选择鼠标类型,一般的鼠标类型都是串行鼠标,接下来使用 TAB 键来移动光标到参数选项,如果读者拥有的是两键鼠标,读者就需要选中第一个按钮“模拟三键”,如果读者的三键鼠标的中键不起作用,读者就需要选中第二个选项“中键送出(左+右)”,一般可以不用选中。下面的两个选项没有什么作用,最后一个选项是如果读者使用的是飞轮鼠标,就可以选中。请确认读者选中了正确的鼠标类型,并按回车继续。

如果读者选择的是串行鼠标,读者现在需要输入读者鼠标连接的串口大多数系统使用的是 串口 1,即 /dev/ttyS0,即使读者输入了错误的端口,也可以再次返回修改。

现在将询问读者鼠标上的键数,一般用户选择两键即可,如果读者使用的是真三键鼠标,就请选中三键。如果读者使用的是 Logitech MouseManPlus,读者可能就需要选择 4~6 键。

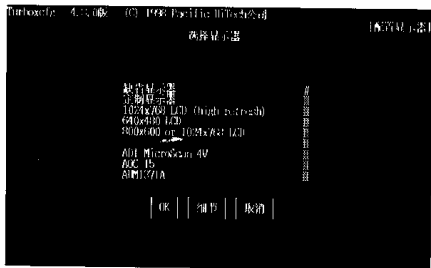


图 2.86 TurboXCfg 配置显示器

TurboXCfg 现在列出一个长长的显示器清单(图 2.86)。如果读者的显示器出现在清单中,选择它并按[Enter],或者选择缺省的设置,此时的屏幕刷新频率比较安全但是效果比较差。否则,选择 Custom。如果读者选了 Custom,首先读者需要输入读者的显示器的制造商和型号,还有读者的显示器的频宽、垂直和水平扫描频率。读者可以选择一个预定义的设置,或者手工输入,读者可以在读者的显示器的手册中找到这些参数,如果读者不知道这些参数,建议读者选择‘未知’。

**警告:**不推荐选择一个和读者的显示器‘相似’的显示器,除非读者能肯定读者选择的显示器不会超过读者实际的显示器能力。如果读者这样做,读者可能会超频读者的显示器并毁灭它。

### TurboXCfg - 选择缺省色彩深度

现在读者需要选择读者希望的色彩深度,色彩越多,显示效果越好。当然,色彩深度受限于读者的显存大小。如果读者选择 16bpp 色彩(每点 16 位色彩),分辨率为 1024 × 768,读者就需要

$1024 \times 768 = 786432 \text{ 点} \times 2 \text{ bytes/点} = 1572864 \text{ byte}$  即 1.5M 显存。

TurboLinux 的 TurboDesk 的桌面环境需要最小为 16bpp 的色彩深度,不然色彩就会溢出,如果读者没有 2M 显存,读者就必须选择其他的窗口管理器。如果读者的显卡拥有 4M 或以上的显示内存,读者就可以选择 24bpp 或 32bpp 来获得最佳的效果。

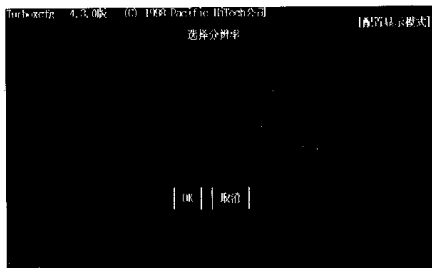


图 2.87 TurboXCfg 选择显示模式

在这一步中 TurboXCfg 将运行 X -probeonly 来自动检测读者系统上能够达到的显示分辨率,屏幕将会闪动几次,给出一个可能的分辨率列表(图 2.87),可移去读者不需要的显示分辨率,读者的 X Window 将自动使用读者选中的最大分辨率。如果读者使用虚拟桌面,读者就需要移动鼠标来查看整个窗口。

### TurboXCfg - 设置其他选项

如果读者选择‘自动配置’,TurboXCfg 将自动选择最高显示模式和刷新频率,如果读者选择‘手工配置’,读者就可以手工输入分辨率和扫描频率,有经验的用户可以这样做。

### TurboXCfg - 选择显示字体

此处读者需要选择是 75dpi 字体,还是 100dpi 字体,如果读者的分辨率高于 800 × 600,一般请使用 100dpi 字体,否则请使用 75dpi 字体。在安装完成后,读者可以随时再运行 TurboXCfg 来重新配置。

### TurboXCfg - 选择登录方式

TurboLinux 有两种登录方式:

- (1) 标准控制台文本方式登录
- (2) 通过 XDM 进行图形界面登录

建议新用户和没有经验的 Linux 用户使用图形登录方式,登录后可立刻开始使用 X 桌面。有经验的用户一般愿意使用文本方式登录,这对于那些主要作为服务器和工作站来使用的 TurboLinux 用户来说是非常方便的。

### TurboXCfg - 测试 X 配置

现在读者可以检查读者的 X 服务器的设置是否正确,然后再写入读者的系统配置。现在 X Window 将使用当前配置启动,然后读者将看到一个标准 X 光标和左上角的一个窗口,此窗口将显示当前的分辨率和刷新频率,按 Next 按钮将在所有的显示模式中切换,按 Quit 按钮或按 Ctrl + Alt + Backspace 键盘将返回安装程序,返回后将询问读者配置是否正确,如果读者选择 'No',读者将可以重新配置读者的 X,选择 '是'继续。

### TurboXCfg - 完成

现在读者的 X Window 系统已经配置完毕,TurboXCfg 将把所有的配置写入读者的系统。读者可以在安装完毕后以 root 身份在文本方式下运行 turboboxfg,来重新配置读者的 X Window 系统。

### 配置时区

选择读者所在的时区,如果系统 CMOS 的时钟设置为 GMT(格林威治)时间,那么就需选中该选项。选择 'OK' 继续。

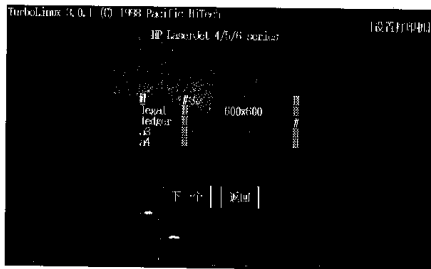


图 2.88 配置打印机

读者现在可以配置打印机,如果读者选择 '是',读者将可以有如下选择。

打印机连接 读者的打印机是如何连接到读者的计算机上的?

本地打印机 读者的打印机直接同计算机相连。选择这个选项后读者还需要输入读者的打印机连接到的端口。

远程打印机 连接在其他 UNIX 机器上的打印机,或者是直接连接到网络上的打印机服务器,选择这个选项后读者还需要输入远程的节点名和远程打印机的队列名称。

LAN Manager/SMB 网络打印机 由 Microsoft Windows for Workgroups, Windows 95/98, 或者 Windows NT 服务器共享出的打印机,读者需要输入 SAMBA 打印机的节点名、SAMBA 打印机的 IP 地址、共享的 SAMBA 打印机的名称、登录 SAMBA 打印机使用的用户名(Windows 服务器缺省是 GUEST, Samba 服务器缺省是 nobody)、登录 SAMBA 打印机使用的密码,缺省时无。

通用打印机选项(图 2.88) 用户产生的打印队列名称,用来存放打印暂存文件的目录。建议使用缺省设置。

Paper Size 中打印机支持的纸张大小。

Resolution 中打印机的分辨率(DPI)。

Color 中如果是彩色打印机,请输入色彩深度。

打印机类型 读者使用的打印机类型。

**注意:**如果读者的打印机是 PS 打印机,读者可以选择 'PostScript Printer'。

**特别选项** 取决于读者选择的打印机类型,对于不同的打印机,这一步会出现不同的选项,共同的地方是:打印机支持的纸张大小、打印机的分辨率(DPI);如果是彩色打印机,请输入色彩深度。如果是 HP 打印机,读者还要输入 'Fix Stair-Stepping of Text'。

检查读者的打印机配置信息- 检查读者的设置,选中 '编辑' 来改变他们,或者选择 '完成' 来继续。



图 2.89 TurboPNPCfg - ISA 即插即用配置

**注意:**读者可以在安装后以 root 身份运行 `turbopnpcfg`。

如果读者选择 '继续',读者的系统将自动检测即插即用设备(图 2.89),如果读者的系统上没有即插即用设备或者读者想跳过这一步,请选择 '忽略',如果读者没有即插即用设备而选择继续,那么安装程序会自动跳到下一步。

如果读者拥有即插即用设备,检测完后会产生一个列表。大多数设备可以简单地选择 '自动探测',并且对 TurboPNPCfg 的所有问题回答是,可以配置完毕。有经验的用户可以使用 '设置' 来手工配置即插即用设备,配置完成后选择 '完成',此时读者的即插即用设备即被配置完成并且初始化完毕。

**注意:**某些即插即用设备,如 USRobotics Sportster 内置调制解调器和 SE2000 兼容网卡,既可以设置成 PnP 模式又可以设成非 PnP 模式,建议读者设置成非 PnP 模式使用,这样会避免一些不必要问题。

**TurboPPPCfg** - 配置调制解调器和拨号网络

**注意:**读者可以在安装后以 root 身份运行 `turbopppcfg`。

如果在网络类型选项窗口中选中了拨号网络,读者现在就可以配置读者的拨号网络。TurboLinux 3.0.2 现在可以在 `turbonetcfg` 中利用拨号网络配置 plug-in 来配置 PPP。读者看到的第一个画面是配置读者的 PPP 设备,使用 '增加' 按钮或者按 `a` 键来增加一个 PPP 设

备,并且按回车来编辑。

首先必须给这个设备分配一个串口,按下‘串口’按钮。现在读者需要选择读者的串口数量,在下面的对话框中输入速率,普通的值一般为 38400、57600 或者 115200。如果读者使用的是 56K 调制解调器或者 ISDN 读者就可以输入更高的值,如果读者使用比 128KBps 更快的 ISDN 接入,那么读者需要安装一个特别的高速串口卡才可以使用更高的速率。

**注意:**38400 即为 33.6K(或者更低)调制解调器的标准速率。

现在读者可以不用填写其他栏,‘描述’一栏是辨别不同拨号连接的标准,其他的都可以自行填写。

**注意:**‘用户名’和‘密码’只有在使用 PAP 验证的时候才有用,一般情况下是不用输入的。

还有更多的一些选项,不过都不是必须填写的。值得注意的选项有:

使用 PAP 身份验证——大多数 ISP 能够使用 PAP (Password Authentication Protocol)。Windows 下的用户经常使用这种接入方式。

被动模式 LCP——有些 ISP 不希望客户端自行处理验证信息,这时需要打开这个选项。如果读者不确定,请使用缺省值。如果读者的 PPP 设备不工作,读者可以以 root 身份再次运行 `turbopppcfg` 重新配置读者的 PPP 设备。

自动重新连接——如果有时 PPP 连接出错中断,选择这个选项可以自动重新启动连接。

启动时连接——如果读者想一开机就启动拨号网络,就请打开这个选项。

使用静态 IP——如果读者的 ISP 给读者分配了一个静态的 IP 地址,就请打开这个选项。

访问时接通——如果读者希望在别人访问读者的节点时才打开 PPP 连接,就请打开这个选项。

如果读者没有使用 PAP 身份验证,读者需要输入‘登录脚本’设置,如某些 ISP 需要读者在登录时手工输入用户名和密码,这时需要自行编写登录脚本。例如:

|           |             |
|-----------|-------------|
| EXPECT    | SEND        |
| login:    | 用户名         |
| password: | 密码          |
| 263.net > | ppp default |

大多数 ISP 提供 PAP 身份验证,设置完毕后选择‘完成’继续。

如果读者选择了‘静态 IP 地址’的选项,读者将被提示输入读者的静态 IP 地址,如果读者选择‘访问时接通’,读者也要输入读者要连接的远端节点的 IP 地址。

此程序允许读者系统启动时自动运行的系统服务/守护进程,首先它将给出一张已经安装的系统服务的列表(图 2.90),其中某些服务是已经被缺省配置为自动运行的,某些服务仍然需要设定。已经被配置为自动运行的服务由 \* 表示,使用上下键和空格键来选中/移去自动运行的服务。

`TurboWMCfg` - 配置读者的缺省 X Window 桌面

如果读者安装了 X Window 系统,读者现在将可以选择读者的缺省 X Window 桌面。读者所选择的桌面将成为系统的缺省桌面。如果读者希望安装完后再修改系统的缺省

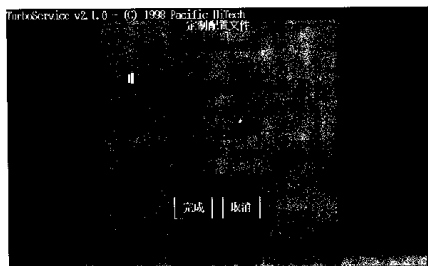


图 2.90 TurboService - 配置系统服务

桌面, 请以 root 身份运行 turbowmcfg。

读者将只能选择你已经安装的窗口管理器, 请注意在每种窗口管理器的边上有一列数字, 表明该窗口管理器需要的最小色彩深度, 请不要选择比读者的 X Window 的色彩深度大的窗口管理器。



图 2.91 设置超级用户密码

安装程序然后要求读者为系统设置一个 root 口令(图 2.91), 口令至少六个字符。读者输入的口令不会显示在屏幕上。读者必需输入口令两次, 如果两次不一致, 安装程序会要求读者重新输入。

增加普通用户

现在读者可以增加普通用户。读者需要输入用户名(最大长度为 8 个字母), 并且输入密码(两次), 选择 OK 继续。

安装完成

现在安装已经完成, 读者可以移去软驱中的软盘, 或者读者的光盘驱动器中的光盘, 按回车重新启动 TurboLinux, 读者也可以返回主画面重新执行某些安装步骤。

## 2.7 常见问题解答

### 1. LILO 出错的意义

没有信息

LILO 没有安装或者安装 LILO 的分区没有被激活。

“LI”

LILO 第一部分被加载,第二部分出错。一般由于 LILO 定位错误或者使用了错误的 boot.b 文件。

“LIL”

LILO 第二部分开始加载,但是没有从 map 文件中成功读取 descriptor tables。

“LIL?”

LILO 第二部分在错误的地方加载,原因同“LI”。

“LIL-”

descriptor table 错误,或者是改变了 boot.b 文件却没有重新安装 LILO。

“LILO”

LILO 的所有部分都安装完毕。

“1010101010”

如果用户的分区情况改变却没有重新安装 LILO,LILO 开机时就会出现 1010 现象。这时用软盘开机重新运行 LILO 就可以修复。另外用户超频的时候也有可能出现这种情况。

### 2. 什么是 /proc/kcore?

/proc/kcore 并不会占据硬盘空间,它只是核心对内存的一个映射文件。

```
$ ls -la /proc/kcore
-r----- 1 root  root  16781312 May 13 17:44 /proc/kcore
$ du /proc/kcore
0 /proc/kcore
```

它的大小同系统内存的大小是一样的。/proc 下的文件属于 proc 文件系统并不会消耗硬盘空间,如果用户误删除之后会有一些问题,但是系统重新启动后将会自动产生这些文件。

### 3. 用户自己制作 bootdisk

```
rdev /vmlinuz /dev/hda1      (其中 hda1 换成用户存放 vmlinuz 的地方)
dd if = vmlinuz of = /dev/fd0
```

### 4. 如果用户误删除了 /etc 下的全部文件怎么办?

如果用户还以 root 在线上还可以挽救,不要 logout,先用 ldconfig 作出 /etc/ld.so.cache,之后就可以用大部份的指令了,试着造出其他的必要文件,若有备份 etc,就用 tar 展开就好了。要不然就用软盘启动,挂上这个分区,再将 etc 的备份文件拷贝回去展开就可以了。

# 3

## Linux 的基本配置简介

在这一章中,我们将讲解如何配置 Linux 环境,使之更适合用户的需要。

在上一章的安装过程结束之后,用户启动机器,在 LILO 中选择启动 Linux,在一长串信息之后,将会出现下面的画面:



图 3.1 Linux 登录画面

用户输入在安装过程中设置的 root 密码之后就可以用超级用户登录系统,进行系统配置工作了。

虽然在安装过程中用户已经进行了一些简单的配置,但是要让用户的 Linux 系统正常运行,用户还需要进行很多配置工作,本章将分以下几部分来讲述 Linux 下的系统配置:

- 网络的配置(TCP/IP,PPP,SLIP)
- X Window 的配置(XFree86Config 的使用)
- /etc/下的配置文件剖析
- 其他设备的配置(光驱、声卡、打印机)

### 3.1 网络的配置(TCP/IP,PPP,SLIP)

要发挥 Linux 的强大威力,网络是必不可少的,没有网络的 Linux 就像没有轮子的跑车。下面我们首先来看看如何在 Linux 下配置网络。

#### 3.1.1 记录必需的网络参数

在用户配置 Linux 的网络之前,用户需要记录一些网络的参数,可以向用户的 ISP 或系统管理员查询相关的信息。

##### 1. IP 地址

这是每一台机器的唯一地址,是由四个小于 255 的整数组合而成。例如: 202.38.128.54,如果用户要使用 SLIP 或 PLIP 拨号上网就可能不需要这些参数,如果用户只使用 loopback 设备(用户不使用 Ethernet/SLIP/PPP/PLIP 这些网络连接,但是用户可以使用 127.0.0.1 来当作本机的网络地址,可以安装开发一切网络软件而不用进行实际的网络连接),用户将不需要实际的地址。

## 2. 网络掩码(netmask)

为了执行效率起见,需要限制网络上特定区段的节点数,通常系统管理员把他们的网络分成几个子网(Subnet),然后给每个子网分配部份网络地址,当网络掩码以比特形式覆盖在用户自己的网络上的地址,用来告知用户它所在的子网。这对路由来说很重要。系统管理员在规划网络时,就已经选定网络掩码了,他会告知用户正确的网络掩码。

一般说来,大部分 class-C 的子网,都使用 255.255.255.0 当作网络掩码,Class-B 的网络使用 255.255.0.0。

| 网络级别      | 子网掩码          |           |
|-----------|---------------|-----------|
| 1 - 127   | 255.0.0.0     | (Class A) |
| 128 - 191 | 255.255.0.0   | (Class B) |
| 192 +     | 255.255.255.0 | (Class C) |

## 3. 网络地址(network address)

用机器的 IP 地址同网络掩码做 AND 运算而得:

```
网络掩码:      255.255.255.0
IP 地址:       202.38.128.32    & &
-----
网络地址:      202.38.128.0    =
```

## 4. 广播地址

用用户的网络地址与网络掩码的反码来作 OR 运算而得:

```
用户的网络掩码: 255.255.255.0    !
网络掩码的反码:  0.  0.  0.255    =
网络地址是:      202.38.128.0    ||
-----
广播地址:        202.38.128.255    =
```

很多网络使用网络地址当作广播地址。广播地址有什么用,用户用下面这个命令知道:

```
ping 202.38.128.255
```

这个命令将对 202.38.128 子网上的所有机器作 ping 操作,这就是著名的广播风暴,使用这个命令会造成整个子网上有大量的 IP 包在广播,网络速度会急剧下降,请用户注意。

## 5. 路由地址——Router('Gateway') Address

这是一个将用户的局域网连至外部网络的机器的地址,一般路由器的地址使用下面两个惯例:

```
路由器使用最低的地址:202.38.128.1,
路由器使用最高的数字:202.38.128.254。
```

实际上路由器可以使用网络上任何一个网络地址。

如果用户使用 loopback 的方式,就不需要路由器地址。如果用户使用 PPP,那用户也不需要知道自己使用的路由器地址,因为 PPP 会自动替用户决定正确的地址。如果用户使用 SLIP,用户的路由器地址将会是用户的 SLIP 服务器的地址。

#### 6. 名字服务器的地址 (Domain Nameserver Address)

网络上的机器都要使用名字服务器来将机器名翻译成相应的 IP 地址,或将 IP 地址翻译成机器名,用户可以使用 nslookup 来查询机器地址:

```
nslookup ftp.ihep.ac.cn
```

如果用户在自己的机器上运行了 named,用户也可以将自己的机器作为名字服务器来使用,使用 loopback 的用户不用设置名字服务器的地址。

如果用户使用调制解调器拨号上网,还需要知道拨号服务的电话号码。确认调制解调器设定正确而且用户知道它连接到哪个串行口。

#### 3.1.2 网络配置文件: /etc/rc.d/rc.inet1, rc.inet2 和其他相关文件

用户每次手动设定网络设备后,系统就会将这些设置记录在/etc下的一些文件中,这样系统每次重新启动时,都会自动设定网络。

/etc/rc.d 目录下“rc”开头的文件是用来做启动系统的初始化文件。对一些非标准的 UNIX 系统 (non-UNIX-wizard)来说,rc 系列的文件是在系统启动时用 init 程序来执行的,其中执行所有基本的系统程序:syslogd,updated 及 crond。rc 系列文件与 MS-DOS 系统下的 autoexec.bat 很类似。rc 的意思是“runtime commands”。这些文件一般是放在 /etc 目录下,Linux 目前并没硬性规定要放在哪里。依 BSD (/etc/rc.\*) 或 System-V (/etc/rc.d/rc\*) 的惯例都可以,一般使用 SYSV 的惯例:也就是说这些文件可在 /etc/rc.d 目录下找到。rc.inet1 及 rc.inet2 是两个网络配置文件。

用户应该使用 /sbin/ifconfig 及 /sbin/route 来设定用户机器基本的 tcp/ip 接口。

一个标准的 rc.inet1 文件如下:

```
# ! /bin/sh
#
# rc.inet1      This shell script boots up the base INET system.
#
# Version:      @( # )/etc/rc.d/rc.inet1      1.01      05/27/93
#

HOSTNAME='cat /etc/HOSTNAME'      # 设定节点名

# Attach the loopback device.      设定 loopback 设备
/sbin/ifconfig lo 127.0.0.1
/sbin/route add -net 127.0.0.0 netmask 255.0.0.0 lo

# IF YOU HAVE AN ETHERNET CONNECTION, use these lines below to configure the
```

```
# eth0 interface. If you're only using loopback or SLIP, don't include the
# rest of the lines in this file.

# Edit for your setup.
IPADDR="202.38.128.33"    # REPLACE with YOUR IP address!    设定 IP 地址
NETMASK="255.255.255.0"  # REPLACE with YOUR netmask!    设定网络掩码
NETWORK="202.38.128.0"   # REPLACE with YOUR network address! 设定网络地址
BROADCAST="202.38.128.255" # REPLACE with YOUR broadcast address, if you
                           # have one. If not, leave blank and edit below.
                           # 设定广播地址
GATEWAY="202.38.128.63"  # REPLACE with YOUR gateway address!
# 设定路由地址

# Uncomment the line below to initialize the ethernet device.
# /sbin/ifconfig eth0 $ {IPADDR} broadcast $ {BROADCAST} netmask $ {NETMASK}

# Uncomment these to set up your IP routing table.
# /sbin/route add -net $ {NETWORK} netmask $ {NETMASK} eth0
# /sbin/route add default gw $ {GATEWAY} netmask 0.0.0.0 metric 1

# End of rc.inet1
```

其中/sbin/ifconfig 设定用户的网络界面,如果不加参数就可以显示目前的网络配置,ifconfig 的语法是:

```
/sbin/ifconfig eth0 IP 地址 broadcast 广播地址 netmask 网络掩码
```

Linux 的第一个 ethernet 设备叫做 'eth0', 第二个叫做 'eth1', 依次类推。

/sbin/route 是用来开启、修改、删除路由表中的项目。如果网络上有一个数据包要传送的话,网络服务程序会查询这个路由表列出目的地址,及这个地址可能使用到的 interface。用户可以使用 route 这个命令不加参数来显示 routing table 的内容。route 的语法如下:

```
/sbin/route add -net 网络地址 netmask 网络掩码 eth0
```

```
/sbin/route add default gw 路由器 netmask 0.0.0.0 metric 1
```

rc.inet2 会启动所有网络守护进程,像 inetd, portmapper 等。在设定完网络设备之后,用户必须要执行必要的网络应用程序和守护进程。

一个标准的 rc.inet2 文件如下:

```
# ! /bin/sh
#
# rc.inet2    This shell script boots up the entire INET system.
#
#    Note, that when this script is used to also fire
#
#    up any important remote NFS disks (like the /usr
#
#    distribution), care must be taken to actually
#
#    have all the needed binaries online _ now _ . . .
#
```

```

# Author:   Fred N. van Kempen, <waltje@u.walt.nl.mugnet.org>
#

# Constants.
NET="/usr/sbin"
IN_SERV="lpd"
LPSPPOOL="/var/spool/lpd"

# At this point, we are ready to talk to The World...
echo "Mounting remote file systems..."
/sbin/mount -a -t nfs          # This may be our /usr runtime!!!   加载 nfs 文件系统

echo -n "Starting daemons:"

# Start the SYSLOGD/Klogd daemons.  These must come first. 启动 syslogd
if [ -f ${NET}/syslogd ]; then
    echo -n " syslogd"
    ${NET}/syslogd & # Backgrounded to avoid an ugly notice from bash-2.0
    echo -n " klogd"
    ${NET}/klogd
fi

# Start the SUN RPC Portmapper. 启动 portmap
if [ -f ${NET}/rpc.portmap ]; then
    echo -n " portmap"
    ${NET}/rpc.portmap
fi

# Start the INET SuperServer 启动 inetd
if [ -f ${NET}/inetd ]; then
    echo -n " inetd"
    ${NET}/inetd
else
    echo "no INETD found.  INET cancelled!"
    exit 1
fi

# # Start the NAMED/BIND name server   启动名字服务器.
# if [ -f ${NET}/named ]; then
#     echo -n " named"
#     ${NET}/named
# fi

```

```

# # Start the ROUTEd server.      启动路由服务器
# if [ -f ${NET}/routed ]; then
#     echo -n " routed"
#     ${NET}/routed -g -s
# fi

# # Start the RWHO server.      启动 rwho 服务器
# if [ -f ${NET}/rwhod ]; then
#     echo -n " rwhod"
#     ${NET}/rwhod -t -s
# fi

# Start the various INET servers.      启动 Inet 服务器
for server in ${IN_SERV} ; do
    if [ -f ${NET}/${server} ]; then
        echo -n " ${server}"
        ${NET}/${server}
    fi
done

# # Start the various SUN RPC servers.      启动 rpc 守护进程
if [ -f ${NET}/rpc.portmap ]; then
    # Start the NFS server daemons.
    if [ -f ${NET}/rpc.mountd ]; then
        echo -n " mountd"
        ${NET}/rpc.mountd
    fi
    # if [ -f ${NET}/rpc.nfsd ]; then
    #     echo -n " nfsd"
    #     ${NET}/rpc.nfsd
    # fi
    # # Fire up the PC-NFS daemon(s).
    # if [ -f ${NET}/rpc.pcnfsd ]; then
    #     echo -n " pcnfsd"
    #     ${NET}/rpc.pcnfsd ${LPSPool}
    # fi
    # if [ -f ${NET}/rpc.bwnfsd ]; then
    #     echo -n " bwnfsd"
    #     ${NET}/rpc.bwnfsd ${LPSPool}
    # fi
fi # Done starting various SUN RPC servers.

# The 'echo' below will put a carriage return at the end

```

```
# of the list of started servers.
echo

# # Setting up NIS: 设定 NIS
# # (NOTE: For detailed information about setting up NIS, see the
# # documentation in /usr/doc/yp-clients and /usr/doc/ypserv)
# #
# # First, we must set the NIS domainname. NOTE: this is not
# # necessarily the same as your DNS domainname, set in
# # /etc/resolv.conf! The NIS domainname is the name of a domain
# # served by your NIS server.
# #
# if [ -r /etc/defaultdomain ]; then
#     domainname `cat /etc/defaultdomain`
# fi
#
# # Then, we start up ybind. It will use broadcast to find a server.
# #
# if [ -d /var/yp ]; then
#     echo "Running ybind..."
#     /usr/sbin/ybind
# fi

# Done!
```

其他一些与网络配置相关的文件有：

/etc/hosts

/etc/hosts 包含了一个 IP 地址和节点名的列表。Linux 在由节点名查地址的时候先到这个文件中去搜索,这个文件中至少要包含 127.0.0.1 localhost 这一行,如果用户不只使用 loopback,还要增加用户机器的地址。

例如,如果 siskin.ihep.ac.cn 的 ip 地址是 202.38.128.32,/etc/hosts 文件中要设成:

```
# /etc/hosts
# List of hostnames and their ip addresses
127.0.0.1          localhost
202.38.128.32     siskin.ihep.ac.cn
# end of hosts
```

用户还可以在其中加入自己常用的一些节点和 IP 的对应表,这样会加快连线速度。因为省去了到名字服务器上查询的时间。

/etc/networks

/etc/networks 是给 route 用的,格式跟 /etc/hosts 很像,以下是个例子:

```
#
# /etc/networks: list all networks that you wish to add route commands
```

```
#               for in here
#
default         0.0.0.0           # default route - recommended
loopnet         127.0.0.0       # loopback network - recommended
mynet           202.38.128.0    # Example network CHANGE to YOURS
#
# end of networks
```

/etc/host.conf

这个文件说明用户的系统会如何查询节点名，它应该包含至少以下两行：

```
order hosts,bind
multi on
```

这两行通知先检查 /etc/hosts 文件，然后去查名字服务器，并且可以使用多个名字服务器。

/etc/resolv.conf

这个文件设定系统使用的名字服务器。

例如：如果用户的机器是 goober.norelco.com，而且有一个 127.253.154.5 的名字服务器，那用户的 /etc/resolv.conf 文件如下：

```
domain norelco.com
nameserver 127.253.154.5
```

用户可以指定多个名字服务器，每个都在 resolv.conf 文件中有一行。

/etc/HOSTNAME

设定用户自己机器的名字。设定机器名之后，用户自己的机器才能被其他机器辨识出来。用户可以使用 hostname 这个命令来设置。

**注意：**bin/domainname 这个命令是用来设定 NIS 的域名，而不是用来设定节点名的，除非用户运行 NIS。

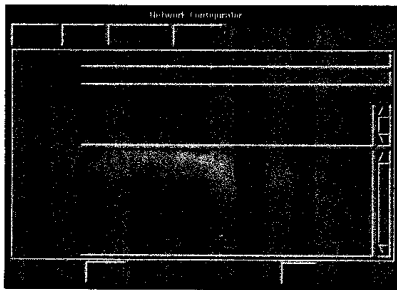


图 3.2 在 Redhat 下配置网络

实际上在 Linux 中设置这些网络选项并不需要用户自己去手工修改这些配置文件，

系统都已经提供了相应的应用程序供用户使用,在 Slackware 下有文本界面的 netconfig,它的使用同安装时候的网络配置使用是一样的,这里就不再作重复讲述了,而 Redhat 提供了一个 X Window 下的网络配置程序,可以通过选择 control panel 中的 Network 来启动也可以通过手工输入 netcfg 来启动,如图 3.3。

选择 interface 就可以设定网络设备了如图 3.4。

输入 IP 地址和网络掩码。要增加一个新设备,按下 add 就可以了。如图 3.5。

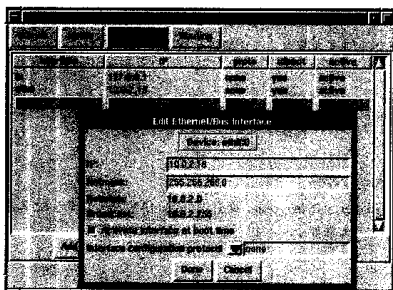


图 3.3 设定 IP 地址和网络掩码

在这里可以设定 PPP,SLIP,Ethernet 和 PLIP 等多种网络设备。

设定路由如下图:



图 3.4 选择网络设备

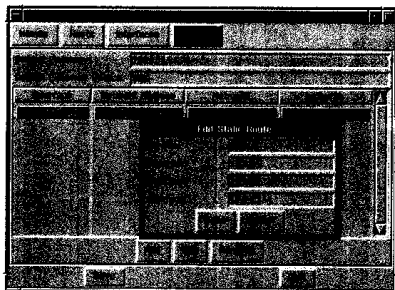


图 3.5 设定路由

这种图形界面的设置程序给用户带来了很大的方便。用户使用这些程序就可以很轻松地配置网络了。

### 3.1.3 设定 SLIP 设备

SLIP (Serial Line Internet Protocol) 允许用户在串行线上使用 TCP/IP, 串行线可以是电话线 + 调制解调器, 或是一条专线 (leased line) 之类的东西。中国目前已经有 ISP 提供 SLIP 服务了。

SLIP 使用用户机器上的串行口来传输 IP 包 (datagram), 为了达到更好的效果它会直接控制串行设备。SLIP 设备叫做 `sl0` 或 `sl1` 等, 网络服务程序使用 `ioctl` 来将串口设备变成 SLIP 设备, 例如 `dip` 或 `slattach`。

SLIP 在实际应用中分为标准 SLIP 与 IP 包压缩 CSLIP 两种。事实上, 所谓的 IP 包压缩只是压缩 IP 包的表头部分, 它能提高数据传输的速度。然而, 在与服务器连线时必须确认对方所使用的协议种类, 否则大部分的功能将无法使用。

完成 SLIP 连线共包含三个步骤: 调制解调器拨号接通主机、设定硬件界面和设定路由表。这三个步骤可分别执行 `dial`, `ifconfig` 与 `route` 三个指令来完成。这里我们不分别说明这三个程序工作原理, 我们把焦点集中在 `dip` 这个程序, 它可以自动调用其他程序, 完成整个 SLIP 连线工作。

#### 1. dip

`dip` (Dialup IP) 能设定串行设备的速度, 也可执行命令自动连线到远端服务器, 自动帮用户登录到服务器端, 找寻服务器端送出的字符串, 分析出它给的 IP 地址, 并执行必要的 `ioctl` 来切换用户的串行口到 SLIP 模式下, `dip` 有一个很强大的使用脚本文件的能力, 能让用户的登录程序自动化。用户可以从下列地址取得最新的 DIP 程序:

`ftp://sunsite.unc.edu/pub/Linux/system/Network/serial/dip *`

用 `dip -t` 指令可以进入测试模式, 得到 `dip` 所有指令列表, 我们不对测试模式多作介绍, 所有动作都可以写入脚本文件中, 拨号、登录和从主机送出的字符串获得用户的 IP 一次完成。

下面是 `dip` 所有指令列表:

DIP: Dialup IP Protocol Driver version 3.3.71-uri (27 Dec 94)

Written by Fred N. van Kempen. MicroWalt Corporation.

DIP > help

DIP knows about the following commands:

|                       |                       |                       |                      |                       |
|-----------------------|-----------------------|-----------------------|----------------------|-----------------------|
| <code>bootp</code>    | <code>break</code>    | <code>chatkey</code>  | <code>config</code>  | <code>databits</code> |
| <code>dec</code>      | <code>default</code>  | <code>dial</code>     | <code>echo</code>    | <code>flush</code>    |
| <code>get</code>      | <code>goto</code>     | <code>help</code>     | <code>if</code>      | <code>inc</code>      |
| <code>init</code>     | <code>mode</code>     | <code>modem</code>    | <code>netmask</code> | <code>parity</code>   |
| <code>password</code> | <code>proxyarp</code> | <code>print</code>    | <code>port</code>    | <code>quit</code>     |
| <code>reset</code>    | <code>secridf</code>  | <code>secrid</code>   | <code>send</code>    | <code>skey</code>     |
| <code>sleep</code>    | <code>speed</code>    | <code>stopbits</code> | <code>term</code>    | <code>timeout</code>  |
| <code>wait</code>     |                       |                       |                      |                       |

下面我们通过一个脚本文件范例来看整个连线过程。有几点说明：

(1) dip 只支持四个内部参数, \$local 存放主机指定给我们的 IP 地址, \$remote 存放 SLIP 服务器的 IP 地址, \$mtu 存放最大传输 IP 包大小, \$errlvl 存放指令执行返回值。

(2) dip 的脚本文件没有循环的功能, 只有简单的 if-goto 逻辑判断及 goto 的分支执行功能。

(3) 在完成 SLIP 连线后, 我们所拥有的 IP 及节点名皆为服务器所指定, 在执行 ftp、telnet 等程序时不会有问题。特别要注意发送 E-mail 时, 发信人地址是用我们所设的节点名, 别人在回信时会无法投递, 因为自设的节点名没有注册, 对方的信件程序无法从节点名反查出用户的 IP。如果要测试信件收发, 可以用 <user> @ <ip addr> 的方式发送。

dip 的脚本文件 test.dip 如下:

```
#
# test.dip      Dialup IP connection support script. (dynamic)
#
# Version:      test.dip 1.0      01/22/97
#
# Author:       Roman
#
# Features:     Support multiple phone numbers
#               Redial automatically when busy
#
main:

#
# Set serial port and speed.
#
port cua1                # 指定 modem 连接端口 (COM2)
speed 57600              # RS-232 传输速度
netmask 255.255.0.0

#
# Prepare for dialing.
#
# send <string> : 传递指令给 modem
# \r 代表回车符, modem 在收到 CR 后开始解释、执行指令
# \n 代表 LF + CR
#
send AT&FL1S0:0S7=25\r
if $errlvl != 0 goto modemerr      # modem 指令执行失败的处理
#
# wait <string> <n>
# 在 <n> 秒内从 modem 取得 <string>, 成功传回 0
```

```
#
wait OK 3                                # 等待 modem 传回执行结果
if $errlvl != 0 goto modemerr
#
# print <string> : 在屏幕上显示 <string>
#
print DIP script for XXX SLIP server by Vincent Chen 08/12/1994
print -----
print ..... AutoDial procedure started .....
print ..... Press Ctrl-C to abort .....
dialsvr:

#
# Now trying the 1st number
# 先拨第一条电话
#
# sleep <n> : 暂停执行 n 秒
#
sleep 3
print ..... Trying 63689279 .....
send ATDT63689279 \r                      # 拨 server 电话号码
if $errlvl != 0 goto modemerr
wait BUSY 15                             # 判断是否占线
if $errlvl == 0 goto next
wait CONNECT 10                          # 等待连线字符串
if $errlvl != 0 goto next
goto login                               # 连线成功,开始登录主机

next:

#
# Now trying the 2nd number
# 第一条电话无法连上,改拨第二条
#
sleep 3
print ..... Trying 63689280 .....
send ATDT63689280 \r
if $errlvl != 0 goto modemerr
wait BUSY 15
if $errlvl == 0 goto dialsvr
wait CONNECT 15
if $errlvl != 0 goto dialsvr
goto login
```

```

#
# 调制解调器连线成功,开始登录服务器
#

login:

    wait login: 10                                # 等主机送出登录的提示字符串
    if $errlvl != 0 goto error
    send slip \r                                    # 送出用户的帐号
    if $errlvl != 0 goto error
    # wait password: 10                            # 等待主机要求输入密码
    # if $errlvl != 0 goto error
    # send xxxxx \r                                # 送出用户的密码
    # if $errlvl != 0 goto error
    #
    # Catch the key word to confirm that IP will show up shortly.
    #
    wait Enter 5                                    # 等待主机送出用户 IP 前的信息
    if $errlvl != 0 goto error
    #
    # Get dynamic IP from strings that sent by server
    #
    # get $local < remote/ip address > < n >
    # 在 < n > 秒内取得我们的 ip, 可指定 dip 自行取得或由我们直接指定
    # get $local remote < n > : 让 dip 从服务器送出的字符串取得我们的 ip 地址
    # get $local xxx.xxx.xxx.xxx < n > : 直接告诉 dip 我们的 ip 地址
    #
    # 注意:在下 get 指令之前必须确定不会有类似 ip 的字符串重复出现,
    # 否则会造成 dip 误判断,前面之所以要等 Enter 字符串就是
    # 考虑到这一点。
    #
    get $local remote 3                             # 由 dip 分析我们的 IP
    if $errlvl != 0 goto error
    #
    # Get gateway IP from strings that sent by server
    #
    # get $remote < remote/ip address > < n >
    # 在 < n > 秒内取得 server 的 ip, 可指定 dip 自行取得或由我们直接指定
    # get $remote remote < n > : 让 dip 从 server 送出的字符串取得服务器的 ip 地址
    # get $remote xxx.xxx.xxx.xxx < n > : 直接告诉 dip 服务器的 ip 地址
    #
    get $remote remote 3                             # 由 dip 分析服务器的 IP 地址

```

```

if $ errlvl != 0 goto error

#
# Set up the SLIP operating parameters.
#
get $ mtu 1024                # 设 IP 包长度最大值

#
# Set Destination net/address as type 'default' (vice an address).
# This is used by the 'route' command to set the kernel routing table.
# Some machines seem to require this be done for SLIP to work properly.
#
default                       # 由 dip 自行建立路由表
#
# Give the connection report !
#

done;

print * * * * * Connected * * * * *
print IP address is $ local
print Remote gateway is $ remote
#
# mode < SLIP/CSLIP/PPP >
#
mode SLIP                     # 进入 SLIP 传输模式
goto exit
error:

print SLIP connection failed.
goto exit

modemerr:

print Modem initializing failed.
goto exit

exit:

```

在脚本文件编辑完成后,只要输入 `dip <script filename>`,dip 将根据预设的程序自动登录服务器。以下是整个脚本文件的执行过程:

```

bash # dip /etc/test.dip
DIP: Dialup IP Protocol Driver version 3.3.71-uri (27 Dec 94)
Written by Fred N. van Kempen, MicroWalt Corporation.

```

DIP 脚本文件 for xxx SLIP server by Roman 01/22/1997

```
..... AutoDial procedure started .....
..... Press Ctrl-C to abort .....
..... Trying xxx-xxxx .....
* * * * * Connected * * * * *
IP address is xxx.xxx.xxx.xxx
Remote gateway is yyy.yyy.yyy.yyy
```

从上面看到第一条拨号后,可成功地连上服务器。服务器分配的 IP 是 xxx.xxx.xxx.xxx,服务器的 IP 地址是 yyy.yyy.yyy.yyy。用户可以自己修改这个脚本文件支持多条电话线自动拨号。

连线完成后我们可以用 ifconfig 指令,看到 dip 已经为我们设定好 sl0 这个 SLIP 通道,在 1.1.78 版之前的核心只支持 4 个通道,目前新的核心可以支持 16 个通道,所有客户程序都共用一个通道,所以要拿这台计算机做服务器的话,还可以支持 15 个用户拨号。

接下来先用 ifconfig 指令检查 dip 是否已经设置好 slip 界面。

```
bash # ifconfig

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:2000   Metric:1
            RX packets:0 errors:0 dropped:0 overruns:0
            TX packets:20 errors:0 dropped:0 overruns:0

sl0         Link encap:VJ Serial Line IP
            inet addr:xxx.xxx.xxx.xxx  P-t-P:yyy.yyy.yyy.yyy  Mask:255.255.255.0
            UP POINTOPOINT RUNNING  MTU:1024   Metric:1
            RX packets:0 errors:0 dropped:0 compressed:0
            TX packets:0 errors:0 dropped:0 compressed:0
```

在下表我们可以看到除原有的 loop back 之外,又多了 xxx.xxx.xxx.xxx 及 default 这两个 dip 为我们设好的 IP 包传输路径。default route 是一个特殊的 IP 包传输路径,所有目的地不在传输路径表上的 IP 包都通过这个管道对外传送,我们可以看到缺省路由是由 xxx.xxx.xxx.xxx,因此必须另外指定送往 xxx.xxx.xxx.xxx 的 IP 包是由 sl0 设备传送的。

```
bash # route
```

Kernel routing table

| Destination     | Gateway         | Genmask         | Flags | MSS  | Window | Use | Iface |
|-----------------|-----------------|-----------------|-------|------|--------|-----|-------|
| xxx.xxx.xxx.xxx | *               | 255.255.255.255 | UH    | 960  | 0      | 0   | sl0   |
| loopback        | *               | 255.0.0.0       | U     | 1936 | 0      | 20  | lo    |
| default         | xxx.xxx.xxx.xxx | *               | UG    | 960  | 0      | 0   | sl0   |

要结束连线只要执行 dip -k 指令,dip 会帮我们挂断电话、清除路由表、关闭 sl0 设备

等所有工作,恢复没有拨号前的系统状况。当然用 `kill -9 < dip process id >` 也有同样的效果。

```
bash # dip -k
```

```
DIP: Dialup IP Protocol Driver version 3.3.71-uri (27 Dec 94)
```

```
Written by Fred N. van Kempen, MicroWalt Corporation.
```

```
DIP: process 115 killed.
```

## 2. slattach

DIP 是给调制解调器用户使用的,对于用永久性的串行设备相连的用户来说使用 `slattach` 是更为方便。

既然用户的连接是固定不变的,用户可能会想要在 `rc.inet1` 中加入一些设定。用户所需要作的就是为串行设备设定好正确的速度与切换串行设备成 SLIP 模式。`slattach` 这个指令可以帮助用户做到上述的设定。把下面这段范例加到用户的文件 `rc.inet1` 中:

```
#
# Attach a leased line static slip connection
#
# configure /dev/cua0 for 19.2kbps and csip
/sbin/slattach -p csip -s 19200 /dev/cua0 &
/sbin/ifconfig sl0 IPA.IPA.IPA.IPA pointopoint IPR.IPR.IPR.IPR up
#
# End static slip.
```

这里:

```
IPA.IPA.IPA.IPA  用户的 IP 地址
IPR.IPR.IPR.IPR  远端的机器的 IP 地址
```

`slattach` 允许用户用 `-p` 的参数设定使用不同的协议。可依据用户是否需要压缩来决定使用 CSLIP 还是 SLIP,不过连线的两端必须使用同样的协议就可以了。

### 3.1.4 设定 PLIP 设备(选项)

PLIP (Parallel Line IP)用来提供在两台机器间做点对点的连线,除了使用电脑上的并口(parallel port)而不使用串行口(serial port外),它与 SLIP 很像。因为使用并口可以一次传输超过 1 比特的数据,而且并行设备可以达到比串行设备更高的传输速率。

下面是一个范例程序,用户需要增加下列的数行到 `rc.inet1`:

```
#
# Attach a PLIP interface
#
# configure first parallel port as a plip 设备
/sbin/ifconfig plip0 IPA.IPA.IPA.IPA pointopoint IPR.IPR.IPR.IPR up
#
# End plip
```

这里：

IPA .IPA .IPA .IPA 表示用户自己机器的 IP address.

IPR .IPR .IPR .IPR 表示远端机器的 IP address.

这个点对点的参数跟上文 SLIP 有一样的含义，它指定另外一端的地址。除了不需要 dip 与 slattach 之外，其他 PLIP 同 SLIP 是一样的。

#### PLIP 引脚图

PLIP 的引脚图同 DOS 下 interlnk 程序的引脚图是一样的。

这个引脚图在 /usr/src/linux/drivers/net/plip.c 内：

| 引脚名       | 连接对应关系  |
|-----------|---------|
| GROUND    | 25 - 25 |
| D0-ERROR  | 2 - 15  |
| ERROR-D0  | 15 - 2  |
| D1-SLCT   | 3 - 13  |
| SLCT-D1   | 13 - 3  |
| D2-PAPOUT | 4 - 12  |
| PAPOUT-D2 | 12 - 4  |
| D3-ACK    | 5 - 10  |
| ACK-D3    | 10 - 5  |
| D4-BUSY   | 6 - 11  |
| BUSY-D4   | 11 - 6  |
| D5        | 7 *     |
| D6        | 8 *     |
| D7        | 9 *     |
| STROBE    | 1 *     |
| FEED      | 14 *    |
| INIT      | 16 *    |
| SLCTIN    | 17 *    |

注意:不要去连接到标有\* \* '的接脚。

### 3.1.5 设定 PPP 设备

PPP (点对点协议)是在 null-modem/modem/ISDN 上运行 IP 协议以及其他网络协议的一种协议。使用 PPP, 用户可以把 Linux PC 连接到一台 PPP 服务器上并存取该服务器所连接的网络资源,如同用户直接连接在该网络上一样。它提供更强的功能、错误侦测及额外的数据安全,同时修正一些 SLIP 的缺点,且适合做同步的传输或异步的传输。

使用 PPP 代替 SLIP 通常的原因有:

- (1)ISP 只提供 PPP。
- (2)线路质量不好,PPP 修复错误速度比 SLIP 快。
- (3)PPP 上可以运行其他协议如 DECNET/AppleTalk。

Slackware 和 RedHat 缺省都安装了 PPP,如果用户要手动更新,网络地址为

`ftp://sunsite.unc.edu/pub/Linux/system/network/serial/ppp/`

首先用户得确定系统核心已经支持 PPP,如果没有,就要重新编译核心使核心支持 PPP 协议。

如果是 RedHat 的用户,用户就可以使用 `ezppp` 来配置 PPP,这个软件为一个类似于 Windows 95 下的拨号设置的图形界面的程序,如下图:

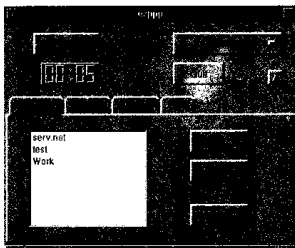


图 3.6 EZPPP 的画面

如果用 TurboLinux 可以使用 `turbopppefg` 来配置 PPP。或者也可以使用 KDE 中的 `kppp`,这是一个非常好用的图形拨号程序。

用户设定 PPP 的第一步是设定 PPP 连接文件,用户必须以 `root` 身份登录来建立这些目录并且编辑这些设立 PPP 连线所需的文件:

```

/usr/sbin/pppd           # PPP 执行文件
/etc/ppp/scripts/ppp-on  # 拨号/连线指令文件
/etc/ppp/scripts/ppp-on-dialer # 拨号的 chat 指令文件部份
/etc/ppp/scripts/ppp-off # 断线指令文件
/etc/ppp/options         # 所有连线所使用的选项

```

```
/etc/ppp/options.ttyXX      # 给某一特定通讯口使用的选项
```

Red Hat Linux 的使用者应注意标准的 Red Hat 4.X 版本,这些指令脚本文件存放在目录 `/usr/doc/ppp-2.0f-2/scripts` 下面。

在用户的 `/etc` 目录里应该要有个目录:

```
drwxrwxr-x  2 root   root      1024 Oct  9 11:01 ppp
```

这个目录存放了所有的 PPP 配置文件。其中有一个 `options.tpl` 文件是一个例子文件,用户可以编辑它来产生自己的 `options` 文件。

如果用户有多个串行线路/调制解调器(典型的例子是 PPP 服务器),那么建立一个通用的 `/etc/ppp/options` 文件,其中包含每个用户提供支持拨入的串行口所使用的共同选项,并且为每一个需要单独设定以建立 PPP 连线的串行线路设立单独的选项文件。这些文件名分别为 `options.ttyx1`, `options.ttyx2`, 依此类推(其中 `x` 是用户串行口的顺序)。

对于单一 PPP 连线,用户可以直接使用 `/etc/ppp/options` 这个文件。另外一种办法,用户可以把所有的选项放进 `pppd` 指令作为参数。下面是一个简单的例子:

```
# /etc/ppp/options (No PAP/CHAP)
#
# 避免 pppd 进入后台执行
-detach
#
# 使用调制解调器控制线
modem
# 使用 uucp 形态的锁定文件以避免它人取用串行设备
lock
# 使用硬件流量控制
crtscts
# 在路由表中将此连结建立为缺省发送设备
defaultroute
# 不使用任何“溢出”控制序列
asynmap 0
# 最大传送 IP 包大小为 552 bytes
mtu 552
# 最大接收 IP 包大小为 552 bytes
mru 552
#
# -----END OF SAMPLE /etc/ppp/options (no PAP/CHAP)
```

接下来用户可以手动建立 PPP 连结,手动成功之后就可以建立自动拨号文件。输入:

```
% pppd -d -detach /dev/ttySx 38400 &
```

用户调制解调器的灯现在开始闪烁,需要等待 `pppd` 完成拨号过程,此时用户可以观察 PPP 设备的状况:

```
% ifconfig
ppp0      Link encap:Point-Point Protocol
```

```
inet addr:10.144.153.104 P-t-P:10.144.153.51 Mask:255.255.255.0
UP POINTOPOINT RUNNING MTU:552 Metric:1
RX packets:0 errors:0 dropped:0 overruns:0
TX packets:0 errors:0 dropped:0 overruns:0
```

其中 inet addr: 10.144.153.10 是该连结中用户一端的 IP 地址。  
P-t-P:10.144.153.5 是服务器端的 IP 地址。

如果是下面的信息,用户就需要检查一下各项设置了:

```
ppp0      Link encap:Point-Point Protocol
          inet addr:0.0.0.0 P-t-P:0.0.0.0 Mask:0.0.0.0
          POINTOPOINT MTU:1500 Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0
          TX packets:0 errors:0 dropped:0 overruns:0
```

用户现在还可以使用 route 命令来查看路由表

```
% route -n
Kernel routing table
```

| Destination  | Gateway      | Genmask         | Flags | MSS  | Window | Use | iface |
|--------------|--------------|-----------------|-------|------|--------|-----|-------|
| 10.144.153.3 | *            | 255.255.255.255 | UH    | 1500 | 0      | 1   | ppp0  |
| 127.0.0.0    | *            | 255.0.0.0       | U     | 3584 | 0      | 11  | lo    |
| 10.0.0.0     | *            | 255.0.0.0       | U     | 1500 | 0      | 35  | eth0  |
| default      | 10.144.153.3 | *               | UG    | 1500 | 0      | 5   | ppp0  |

现在试着 ping 其他机器(不是 PPP 服务器自己),例如

```
% ping sunsite.unc.edu
PING sunsite.unc.edu (152.2.254.81): 56 data bytes
64 bytes from 152.2.254.81: icmp_seq=0 ttl=254 time=190.1 ms
64 bytes from 152.2.254.81: icmp_seq=1 ttl=254 time=180.6 ms
64 bytes from 152.2.254.81: icmp_seq=2 ttl=254 time=169.8 ms
64 bytes from 152.2.254.81: icmp_seq=3 ttl=254 time=170.6 ms
64 bytes from 152.2.254.81: icmp_seq=4 ttl=254 time=170.6 ms
(按下 Ctrl+C 终止)
```

```
--- sunsite.unc.edu ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 169.8/176.3/190.1 ms
```

如果没有回答,就 ping 一下 ISP 的主机的 IP 地址,如果 ping 通就是/etc/resolv.conf 文件有问题,如果还是不对,那么确定在其他操作系统上能否正确运行,如果可以,那就是 Linux 配置的问题。

如果一切正常,输入 ppp-off 结束连线,一段短暂的停顿之后,调制解调器应该会自己挂断电话。

如果这个指令不起作用的话,那么直接关掉用户的调制解调器,另外记得输入下面的指令:

```
rm -f /var/lock/LCK.ttySx
```

下面用户就可以编辑自动连结指令文件。

## 1. ppp-on 指令文件

这是实际启动连线的一对指令文件中的第一个。

```
# ! /bin/sh
#
# Script to initiate a PPP connection. This is the first part of the
# pair of scripts. This is not a secure pair of scripts as the codes
# are visible with the 'ps' command. However, it is simple.
#
# These are the parameters. Change as needed.
TELEPHONE = 5551212      # The telephone number for the connection
ACCOUNT = george         # The account name for logon (as in 'George Burns')
PASSWORD = gracie        # The password for this account (and 'Gracie Allen')
LOCAL_IP = 0.0.0.0       # Local IP address if known. Dynamic = 0.0.0.0
REMOTE_IP = 0.0.0.0      # Remote IP address if desired. Normally 0.0.0.0
NETMASK = 255.255.255.0  # The proper netmask if needed
#
# Export them so that they will be available to 'ppp-on-dialer'
export TELEPHONE ACCOUNT PASSWORD
#
# This is the location of the script which dials the phone and logs
# in. Please use the absolute file name as the $PATH variable is not
# used on the connect option. (To do so on a 'root' account would be
# a security hole so don't ask.)
#
DIALER_SCRIPT = /etc/ppp/ppp-on-dialer
#
# Initiate the connection
#
#
exec /usr/sbin/pppd debug /dev/ttySx 38400 \
    $ LOCAL_IP: $ REMOTE_IP \
    connect $ DIALER_SCRIPT
```

用户编辑这个指令文件,须输入用户在 ISP 的用户名和密码,用户 ISP 的电话号码。而且,如果用户在 /etc/ppp/options 文件里设定了 IP 地址,删除“LOCAL\_IP: \$ REMOTE\_IP \”这一行。另外确定 DIALER\_SCRIPT 指向用户将要使用的拨号指令文件的全路径名。

## 2. ppp-on-dialer 指令文件

```
# ! /bin/sh
#
# This is part 2 of the ppp-on script. It will perform the connection
```

```
# protocol for the desired connection.
#
/usr/sbin/chat -v                                \
TIMEOUT 3                                         \
ABORT '\nBUSY\r'                                  \
ABORT '\nNo ANSWER\r'                            \
ABORT '\nRINGING\r\n\r\nRINGING\r'              \
'' \rAT                                           \
'OK-+++\c-OK' ATH0                               \
TIMEOUT 30                                        \
OK ATDT $ TELEPHONE                              \
CONNECT ''                                        \
login:--gin: $ ACCOUNT                            \
password: $ PASSWORD
```

第二个指令文件实际是建立用户的 PPP 连结。一份 chat 指令文件是一系列“期待字符串”与“送出字符串”的配对。我们总是在送出某些字符串之前先期待某些字符串出现。如果我们在没有先接收到任何字符串的情况下要送出某些字符串的话,我们必须使用空的期待字符串,而在没有送出任何字符串的情况下要期待某些字符串作法也类似。而且,如果字符串包含好几个字(例如 No CARRIER),用户必须把字符串用引号括住,这样 chat 会把字符串当作是一个字符串。下面逐行解释:

```
exec /usr/sbin/chat -v      运行 chat 程序。
TIMEOUT 3                  设定 MODEM 响应时间为 3 秒。
ABORT '\nBUSY\r'           如果接收到 BUSY 字符串,就中止执行。
ABORT '\nNo ANSWER\r'     如果接收到 No ANSWER 字符串,中止执行。
ABORT '\nRINGING\r\n\r\nRINGING\r' 如果有人打电话进来,所以要终止执行。
'' \rAT                    送出 AT 字符串。
OK-+++\c-OK ATH0          这用户的指令可以解决用户的调制解调器连在线上无法断线的情况。
TIMEOUT 30                设定指令文件其他部分超时时间为 30 秒
OK ATDT $ TELEPHONE       期待 OK (调制解调器对 ATH0 指令的回应)并且拨号到我们想要拨入的号码。
CONNECT ''                期待 CONNECT 字符串(连接成功)并且不送出任何恢复字符串。
login:--gin: $ ACCOUNT     再一次,这里面我们有一些错误的恢复设定。期待登录提示符号(...gin:)当接收到提示时,送出用户名(存放在 $ ACCOUNT 环境变量里)。
password: $ PASSWORD      期待密码提示并且送出密码。
```

更详细的使用请参阅 chat 的 man 文件。

### 3.ppp-off 指令文件

```
#!/bin/sh
#####
```

```

#
# Determine the device to be terminated.
#
if [ "$1" = "" ]; then
    DEVICE = ppp0
else
    DEVICE = $1
fi

#####
#
# If the ppp0 pid file is present then the program is running. Stop it.
if [ -r /var/run/ $ DEVICE.pid ]; then
    kill -INT `cat /var/run/ $ DEVICE.pid`

#
# If the kill did not work then there is no process running for this
# pid. It may also mean that the lock file will be left. You may wish
# to delete the lock file at the same time.
    if [ ! "$?" = "0" ]; then
        rm -f /var/run/ $ DEVICE.pid
        echo "ERROR: Removed stale pid file"
        exit 1
    fi

#
# Success. Let pppd clean up its own junk.
    echo "PPP link to $ DEVICE terminated."
    exit 0
fi

#
# The ppp process is not running for ppp0
echo "ERROR: PPP link is not active on $ DEVICE"
exit 1

```

下面就可以测试用户创建的脚本文件了。

以 root 登录,在另一个窗口输入命令:

```
tail -f /var/log/messages
```

以查看调试信息。

在第一个窗口中输入这个命令

```
ppp-on &
```

切换到新窗口就会看到下面的信息:

```

Oct 21 16:09:58 hwin chat[19868]: abort on (No CARRIER)
Oct 21 16:09:59 hwin chat[19868]: abort on (BUSY)
Oct 21 16:09:59 hwin chat[19868]: send (ATZM)

```

```
Oct 21 16:09:59 hwin chat[19868]: expect (OK)
Oct 21 16:10:00 hwin chat[19868]: ATZ`M`M
Oct 21 16:10:00 hwin chat[19868]: OK -- got it
Oct 21 16:10:00 hwin chat[19868]: send (ATDT722298`M)
Oct 21 16:10:00 hwin chat[19868]: expect (CONNECT)
Oct 21 16:10:00 hwin chat[19868]: `M
Oct 21 16:10:22 hwin chat[19868]: ATDT722298`M`M
Oct 21 16:10:22 hwin chat[19868]: CONNECT -- got it
Oct 21 16:10:22 hwin chat[19868]: send (`M)
Oct 21 16:10:22 hwin chat[19868]: expect (ogin:~)
Oct 21 16:10:23 hwin chat[19868]: kepler login: -- got it
Oct 21 16:10:23 hwin chat[19868]: send (hartr`M)
Oct 21 16:10:23 hwin chat[19868]: expect (ssword:~)
Oct 21 16:10:23 hwin chat[19868]: hartr`M
Oct 21 16:10:23 hwin chat[19868]: Password: -- got it
Oct 21 16:10:23 hwin chat[19868]: send (??????`M)
Oct 21 16:10:23 hwin chat[19868]: expect (hartr)
Oct 21 16:10:24 hwin chat[19868]: [hartr -- got it
Oct 21 16:10:24 hwin chat[19868]: send (ppp`M)
Oct 21 16:10:27 hwin pppd[19872]: pppd 2.1.2 started by root, uid 0
Oct 21 16:10:27 hwin pppd[19873]: Using interface ppp0
Oct 21 16:10:27 hwin pppd[19873]: Connect: ppp0 <--> /dev/cua1
Oct 21 16:10:27 hwin pppd[19873]: fsm_sddata(LCP): Sent code 1, id 1.
Oct 21 16:10:27 hwin pppd[19873]: LCP: sending Configure-Request, id 1
Oct 21 16:10:27 hwin pppd[19873]: fsm_rconffreq(LCP): Rcvd id 1.
Oct 21 16:10:27 hwin pppd[19873]: lcp_reqci: rcvd MRU
Oct 21 16:10:27 hwin pppd[19873]: (1500)
Oct 21 16:10:27 hwin pppd[19873]: (ACK)
Oct 21 16:10:27 hwin pppd[19873]: lcp_reqci: rcvd ASYNCMAP
Oct 21 16:10:27 hwin pppd[19873]: (0)
Oct 21 16:10:27 hwin pppd[19873]: (ACK)
Oct 21 16:10:27 hwin pppd[19873]: lcp_reqci: rcvd MAGICNUMBER
Oct 21 16:10:27 hwin pppd[19873]: (a098b898)
Oct 21 16:10:27 hwin pppd[19873]: (ACK)
Oct 21 16:10:27 hwin pppd[19873]: lcp_reqci: rcvd PCOMPRESSION
Oct 21 16:10:27 hwin pppd[19873]: (ACK)
Oct 21 16:10:27 hwin pppd[19873]: lcp_reqci: rcvd ACCOMPRESSION
Oct 21 16:10:27 hwin pppd[19873]: (ACK)
Oct 21 16:10:27 hwin pppd[19873]: lcp_reqci: returning CONFACK.
Oct 21 16:10:27 hwin pppd[19873]: fsm_sddata(LCP): Sent code 2, id 1.
Oct 21 16:10:27 hwin pppd[19873]: fsm_rconffack(LCP): Rcvd id 1.
Oct 21 16:10:27 hwin pppd[19873]: fsm_sddata(IPCP): Sent code 1, id 1.
Oct 21 16:10:27 hwin pppd[19873]: IPCP: sending Configure-Request, id 1
```

```
Oct 21 16:10:27 hwin pppd[19873]: fsm_rconfreq(IPCP): Rcvd id 1.
Oct 21 16:10:27 hwin pppd[19873]: ipcp: received ADDR
Oct 21 16:10:27 hwin pppd[19873]: (10.144.153.51)
Oct 21 16:10:27 hwin pppd[19873]: (ACK)
Oct 21 16:10:27 hwin pppd[19873]: ipcp: received COMPRESSTYPE
Oct 21 16:10:27 hwin pppd[19873]: (45)
Oct 21 16:10:27 hwin pppd[19873]: (ACK)
Oct 21 16:10:27 hwin pppd[19873]: ipcp: returning Configure-ACK
Oct 21 16:10:28 hwin pppd[19873]: fsm_sdata(IPCP): Sent code 2, id 1.
Oct 21 16:10:30 hwin pppd[19873]: fsm_sdata(IPCP): Sent code 1, id 1.
Oct 21 16:10:30 hwin pppd[19873]: IPCP: sending Configure-Request, id 1
Oct 21 16:10:30 hwin pppd[19873]: fsm_rconfreq(IPCP): Rcvd id 255.
Oct 21 16:10:31 hwin pppd[19873]: ipcp: received ADDR
Oct 21 16:10:31 hwin pppd[19873]: (10.144.153.51)
Oct 21 16:10:31 hwin pppd[19873]: (ACK)
Oct 21 16:10:31 hwin pppd[19873]: ipcp: received COMPRESSTYPE
Oct 21 16:10:31 hwin pppd[19873]: (45)
Oct 21 16:10:31 hwin pppd[19873]: (ACK)
Oct 21 16:10:31 hwin pppd[19873]: ipcp: returning Configure-ACK
Oct 21 16:10:31 hwin pppd[19873]: fsm_sdata(IPCP): Sent code 2, id 255.
Oct 21 16:10:31 hwin pppd[19873]: fsm_rconfack(IPCP): Rcvd id 1.
Oct 21 16:10:31 hwin pppd[19873]: ipcp: up
Oct 21 16:10:31 hwin pppd[19873]: local IP address 10.144.153.104
Oct 21 16:10:31 hwin pppd[19873]: remote IP address 10.144.153.51
```

由此完成对 PPP 的配置,下面用户就可以使用 PPP 网络了。

当用户不想使用 PPP 的时候,可以使用标准的 `ppp-off` 指令终止 PPP 连结。

## 3.2 X Window 的配置

X Window 是 UNIX 上的标准图形界面,由 MIT 开发,最后的版本是 X11R6.3, XFree86 是在 Intel 平台上运行的免费 X Window 系统。目前的最新版本是 XFree86 3.3.2,现在先简单介绍如何配置用户的 XFree86 系统。

### 3.2.1 xf86config 和 XF86Setup 的使用

配置 X Window 系统首先要配置 X 服务器(X Server),使 X 服务器能够正常运行,在 Linux 下配置 X 服务器有两个程序可以用:xf86config 和 XF86Setup,前者是字符界面的,后者是图形界面的,下面让我们先来看看如何使用 Xfree86config。

**准备需要的资料:**请先准备好用户的显示器及显示卡的说明书,在设定 X Window 时将会用到显示器的垂直及水平扫描频率。还有请查看一下显示卡所使用的芯片,显示卡上的显存大小。如果是使用 S3 系列的显示卡或者是比较特殊的显示卡的用户可能还需要 DAC 芯片和 Clock 芯片的型号。如果用户不知道,可以运行 Superprobe 这

个程序来查询,不过可能会造成死机。希望用户注意。

执行设定程序,显示如下:

```
# xf86config
This program will create a basic XF86Config file, based on menu selections you make.
```

The XF86Config file usually resides in `/usr/X11R6/lib/X11` or `/etc`. A sample XF86Config file is supplied with XFree86; it is configured for a standard VGA card and monitor with 640x480 resolution. This program will ask for a pathname when it is ready to write the file.

You can either take the sample XF86Config as a base and edit it for your configuration, or let this program produce a base XF86Config file for your configuration and fine-tune it. Refer to `/usr/X11R6/lib/X11/doc/README.Config` for a detailed overview of the configuration process.

For accelerated servers (including accelerated drivers in the SVGA server), there are many chipset and card-specific options and settings. This program does not know about these. On some configurations some of these settings must be specified. Refer to the server man pages and chipset-specific READMEs.

Before continuing with this program, make sure you know the chipset and amount of video memory on your video card. SuperProbe can help with this. It is also helpful if you know what server you want to run.

Press enter to continue, or ctrl-c to abort. < RET >

The directory `'/usr/X386/bin'` exists. You probably have an old version of XFree86 installed (XFree86 3.1 installs in `'/usr/X11R6'` instead of `'/usr/X386'`).

It is important that the directory `'/usr/X11R6'` is present in your search path, \* before \* any occurrence of `'/usr/X386/bin'`. If you have installed X program binaries that are not in the base XFree86 distribution in `'/usr/X386/bin'`, you can keep the directory in your path as long as it is after `'/usr/X11R6'`.

Your PATH is currently set as follows:

```
/usr/local/qt/bin:/usr/local/kde/bin:./:/usr/openwin/bin:/usr/local/bin:/usr/sbin:/usr/local/qt/bin:/usr/local/kde/bin:/usr/local/qt/bin:/usr/local/kde/bin:./:/usr/openwin/bin:/usr/local/bin:/usr/sbin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/usr/X11/bin:/usr/local/netscape:/usr/openwin/bin:/usr/games:./usr/lib/teTeX/bin
```

Note that the X binary directory in your path may be a symbolic link.

In that case you could modify the symbolic link to point to the new binaries.

Example: 'rm -f /usr/bin/X11; ln -s /usr/X11R6/bin /usr/bin/X11', if the link is '/usr/bin/X11'.

Make sure the path is OK before continuing.

Press enter to continue, or ctrl-c to abort. <RET>

在执行设定程序后,下面是一些无关紧要的信息,用户可以一直按回车忽略,然后用户将会看到下面几个选项:

First specify a mouse protocol type. Choose one from the following list:

1. Microsoft compatible (2-button protocol)
2. Mouse Systems (3-button protocol)
3. Bus Mouse
4. PS/2 Mouse
5. Logitech Mouse (serial, old type, Logitech protocol)
6. Logitech MouseMan (Microsoft compatible)
7. MM Series
8. MM HiTablet

If you have a two-button mouse, it is most likely of type 1, and if you have a three-button mouse, it can probably support both protocol 1 and 2. There are two main varieties of the latter type: mice with a switch to select the protocol, and mice that default to 1 and require a button to be held at boot-time to select protocol 2. Some mice can be convinced to do 2 by sending a special sequence to the serial port (see the ClearDTR/ClearRTS options).

Enter a protocol number: 2

市面上所售的鼠标一般都是 Microsoft compatible 的鼠标,但是笔者强烈建议大家购买三键鼠标即 Mouse Systems,并且可以 2 键与 3 键切换的,建议用户将开关拨到 3 键。因为 X Window 下用第三个键来粘贴及激活菜单。另外 logitech 的天貂鼠一般都应该要选第六项。

You have selected a Mouse Systems protocol mouse. If your mouse is normally in Microsoft-compatible mode, enabling the ClearDTR and ClearRTS options may cause it to switch to Mouse Systems mode when the server starts.

Please answer the following question with either 'y' or 'n'.

Do you want to enable ClearDTR and ClearRTS?

一般不用选,直接按回车。

You have selected a three-button mouse protocol. It is recommended that you do not enable Emulate3Buttons, unless the third button doesn't work.

Please answer the following question with either 'y' or 'n'.

Do you want to enable Emulate3Buttons?

如果是两键鼠标选择 yes, 就可以用两键同时按下来模拟三键鼠标。

Now give the full device name that the mouse is connected to, for example /dev/tty00. Just pressing enter will use the default, /dev/mouse.

Mouse device:

询问用户的 mouse 的设备文件的路径, 按回车选择缺省的 /dev/mouse 就可以了, 当用户的鼠标不工作的时候, 用户可以先查看一下 /dev/mouse 是否连接到正确的设备文件,

下面是让用户设定在 X Window 下的键盘。

Beginning with XFree86 3.1.2D, you can use the new X11R6.1 XKEYBOARD extension to manage the keyboard layout. If you answer 'n' to the following question, the server will use the old method, and you have to adjust your keyboard layout with xmodmap.

Please answer the following question with either 'y' or 'n'.

Do you want to use XKB? n

选择 Yes 就可以使用 XKB, 选择 No 就必须使用 xmodmap 来管理键盘。

If you want your keyboard to generate non-ASCII characters in X, because you want to be able to enter language-specific characters, you can set the left Alt key to Meta, and the right Alt key to ModeShift.

Please answer the following question with either 'y' or 'n'.

Do you want to enable these bindings for the Alt keys?

按回车用缺省的, 按 yes 就可以用 alt 加小键盘输入 ASCII 字符。

下面是设定显示器的部分。

Now we want to set the specifications of the monitor. The two critical parameters are the vertical refresh rate, which is the rate at which the whole screen is refreshed, and most importantly the horizontal sync rate, which is the rate at which scanlines are displayed.

The valid range for horizontal sync and vertical sync should be documented in the manual of your monitor. If in doubt, check the monitor database /usr/X11R6/lib/X11/doc/Monitors to see if your monitor is there.

Press enter to continue, or ctrl-c to abort.

You must indicate the horizontal sync range of your monitor. You can either select one of the predefined ranges below that correspond to industry-standard monitor types, or give a specific range.

It is VERY IMPORTANT that you do not specify a monitor type with a horizontal sync range that is beyond the capabilities of your monitor. If in doubt, choose a conservative setting.

hsync in kHz; monitor type with characteristic modes

- 1 31.5; Standard VGA, 640x480 @ 60 Hz
- 2 31.5 - 35.1; Super VGA, 800x600 @ 56 Hz
- 3 31.5, 35.5; 8514 Compatible, 1024x768 @ 87 Hz interlaced (no 800x600)
- 4 31.5, 35.15, 35.5; Super VGA, 1024x768 @ 87 Hz interlaced, 800x600 @ 56 Hz
- 5 31.5 - 37.9; Extended Super VGA, 800x600 @ 60 Hz, 640x480 @ 72 Hz
- 6 31.5 - 48.5; Non-Interlaced SVGA, 1024x768 @ 60 Hz, 800x600 @ 72 Hz
- 7 31.5 - 57.0; High Frequency SVGA, 1024x768 @ 70 Hz
- 8 31.5 - 64.3; Monitor that can do 1280x1024 @ 60 Hz
- 9 31.5 - 79.0; Monitor that can do 1280x1024 @ 74 Hz
- 10 31.5 - 82.0; Monitor that can do 1280x1024 @ 76 Hz
- 11 Enter your own horizontal sync range

Enter your choice (1-11): 3

现在是在让用户设定水平扫描频率,此时请拿起用户先前准备的显示器的资料对比上面的选项,如果有一样的数据,就选相应的选项,如果找不到的话,就请按输入“11”,手工输入自己显示器的水平扫描频率。不过一般选择5是没有什么问题的。

You must indicate the vertical sync range of your monitor. You can either select one of the predefined ranges below that correspond to industry-standard monitor types, or give a specific range. For interlaced modes, the number that counts is the high one (e.g. 87 Hz rather than 43 Hz).

- 1 50-70
- 2 50-90
- 3 50-100
- 4 40-150
- 5 Enter your own vertical sync range

Enter your choice: 3

设定垂直扫描频率,查看显示器的手册,如果有相同的就输入该选项的代码。若没有的话,就选5然后输入自己的参数,注意一般来说市面上所售的显示器在50~90范围内是安全的,再高的扫描频率就有可能把显示器烧坏。特别要提醒的是,如果设定完之后,执行X Window时显示器显示不正常。就要重新设定用户的垂直扫描频率,从1、2、3的选项中挑一个范围比用户小却最接近用户的选项。

然后程序会让用户输入显示器的型号厂家等信息,没有任何作用,可以随便输入。

You must now enter a few identification/description strings, namely an identifier, a vendor name, and a model name. Just pressing enter will fill in default names.

The strings are free-form, spaces are allowed.

Enter an identifier for your monitor definition: CASPER 14

Enter the vendor name of your monitor: CASPER

Enter the model name of your monitor: CASPER

下面可设定显示卡。首先让用户在显示卡数据库中选择对应自己显示卡的代码,使用回车翻页,使用 **q** 退出,如果知道自己的显示卡的显示芯片的也可以不选。

Now we must configure video card specific settings. At this point you can choose to make a selection out of a database of video card definitions.

Because there can be variation in Ramdacs and clock generators even between cards of the same model, it is not sensible to blindly copy the settings (e.g. a Device section). For this reason, after you make a selection, you will still be asked about the components of the card, with the settings from the chosen database entry presented as a strong hint.

The database entries include information about the chipset, what server to run, the Ramdac and ClockChip, and comments that will be included in the Device section. However, a lot of definitions only hint about what server to run (based on the chipset the card uses) and are untested.

If you can't find your card in the database, there's nothing to worry about.

You should only choose a database entry that is exactly the same model as your card; choosing one that looks similar is just a bad idea (e.g. a GemStone Snail 64 may be as different from a GemStone Snail 64+ in terms of hardware as can be).

Do you want to look at the card database? y

|                                                   |               |
|---------------------------------------------------|---------------|
| 0 2 the Max MAXColor S3 Trio64V +                 | S3 Trio64V +  |
| 1 928Movie                                        | S3 928        |
| 2 AGX (generic)                                   | AGX-014/15/16 |
| 3 ALG-5434(E)                                     | CL-GD5434     |
| 4 ASUS PCI-AV264CT                                | ATI-Mach64    |
| 5 ASUS PCI-V264CT                                 | ATI-Mach64    |
| 6 ASUS Video Magic PCI V864                       | S3 864        |
| 7 ASUS Video Magic PCI VT64                       | S3 Trio64     |
| 8 ATI 3D Xpression                                | ATI-Mach64    |
| 9 ATI 8514 Ultra (no VGA)                         | ATI-Mach8     |
| 10 ATI Graphics Pro Turbo                         | ATI-Mach64    |
| 11 ATI Graphics Pro Turbo 1600                    | ATI-Mach64    |
| 12 ATI Graphics Ultra                             | ATI-Mach8     |
| 13 ATI Graphics Ultra Pro                         | ATI-Mach32    |
| 14 ATI Graphics Xpression with 68875 RAMDAC       | ATI-Mach64    |
| 15 ATI Graphics Xpression with AT&T 20C408 RAMDAC | ATI-Mach64    |

---

|                                                  |            |
|--------------------------------------------------|------------|
| 16 ATI Graphics Xpression with CH8398 RAMDAC     | ATI-Mach64 |
| 17 ATI Graphics Xpression with Mach64 CT (264CT) | ATI-Mach64 |

Enter a number to choose the corresponding card definition.

Press enter for the next page, q to continue configuration.

下面就让用户选择自己的 X 服务器,这个选项必须要选,如果用户在上面选择了显示卡,下面就会多出来一个选项,用户选择 5 就是自己显示卡对应的 X 服务器,如果用户的显示卡是 S3/Mach32/Mach8/8514/P9000/AGX/W32/Mach64/I128/S3Virge 系列就可以在第四个选项中挑选自己的 X 服务器,如果用户的显示卡是 ET 系列的就只有选择 3 了,如果用户设定好 X Window 之后不能正常运行,那么恐怕只有选择第 2 个选项,使用简单的 16 色模式的 X Window 了,要么就只有更换显示卡这一条路可以选择了。

Now you must determine which server to run. Refer to the manpages and other documentation. The following servers are available (they may not all be installed on your system):

- 1 The XF86\_Mono server. This is a monochrome server that should work on any VGA-compatible card, in 640x480 (more on some SVGA chipsets).
- 2 The XF86\_VGA16 server. This is a 16-color VGA server that should work on any VGA-compatible card.
- 3 The XF86\_SVGA server. This is a 256 color SVGA server that supports a number of SVGA chipsets. On some chipsets it is accelerated or supports higher color depths.
- 4 The accelerated servers. These include XF86\_S3, XF86\_Mach32, XF86\_Mach8, XF86\_8514, XF86\_P9000, XF86\_AGX, XF86\_W32, XF86\_Mach64, XF86\_I128 and XF86\_S3V.

These four server types correspond to the four different "Screen" sections in XF86Config (vga2, vga16, svga, accel).

Which one of these screen types do you intend to run by default (1-4)?

下面将询问用户是否要作一个连接,是将 /usr/X11R6/bin/X 这个文件连接到实际上使用的 X 服务器程序上。

The server to run is selected by changing the symbolic link 'X'. For example, `'rm /usr/X11R6/bin/X; ln -s /usr/X11R6/bin/XF86_SVGA /usr/X11R6/bin/X'` selects the SVGA server.

The directory /var/X11R6/bin exists. On many Linux systems this is the preferred location of the symbolic link 'X'. You can select this location when setting the symbolic link.

Please answer the following question with either 'y' or 'n'.

Do you want me to set the symbolic link? y

Do you want to set it in /var/X11R6/bin? y

Select an accel server:

- 1 XF86\_S3
- 2 XF86\_Mach32
- 3 XF86\_Mach8
- 4 XF86\_8514
- 5 XF86\_P9000
- 6 XF86\_AGX
- 7 XF86\_W32
- 8 XF86\_Mach64
- 9 XF86\_I128
- 10 XF86\_S3V

Which accel server: 10

上面可是让用户挑选加速 X 服务器,请用户自行选择,如果用户上面研究选择了自己的显示卡,这个选项就不会出现。

下面将询问用户的显存大小。

Now you must give information about your video card. This will be used for the "Device" section of your video card in XF86Config.

You must indicate how much video memory you have. It is probably a good idea to use the same approximate amount as that detected by the server you intend to use. If you encounter problems that are due to the used server not supporting the amount memory you have (e.g. ATI Mach64 is limited to 1024K with the SVGA server), specify the maximum amount supported by the server.

How much video memory do you have on your video card:

- 1 256K
- 2 512K
- 3 1024K
- 4 2048K
- 5 4096K
- 6 Other

Enter your choice: 4

如果用户不知道自己显示卡的显存大小,可以运行 SuperProbe 这个程序来查询,或者在 dos 下运行 univbe 这个程序来查看,或在 WIN95 下看显示卡能够达到的最高色彩深度就可以知道自己有多大显存,例如最多有 16bit 色彩的是 1M 显存,32bit 的是 2M 显存。选择相应的显存大小之后就是让用户输入一些没有用的显示看的生产厂家、型号等信息,可输也可不输。

You must now enter a few identification/description strings, namely an identifier, a vendor name, and a model name. Just pressing enter will fill in default names (possibly from a card definition).

The strings are free-form, spaces are allowed.

Enter an identifier for your video card definition:

You can simply press enter here if you have a generic card, or want to describe your card with one string.

Enter the vendor name of your video card:

Enter the model (board) name of your video card:

下面是让用户选择 S3, AGX 和 W32 这些显示卡上的 RAMDAC.

Especially for accelerated servers, Ramdac, Dacspeed and ClockChip settings or special options may be required in the Device section.

The RAMDAC setting only applies to the S3, AGX, W32 servers, and some drivers in the SVGA servers. Some RAMDAC's are auto-detected by the server.

The detection of a RAMDAC is forced by using a Ramdac "identifier" line in the Device section. The identifiers are shown at the right of the following table of RAMDAC types:

|                                                |           |
|------------------------------------------------|-----------|
| 1 AT&T 20C490 (S3 and AGX servers, ARK driver) | att20c490 |
| 2 AT&T 20C498/21C498/22C498 (S3, autodetected) | att20c498 |
| 3 AT&T 20C409/20C499 (S3, autodetected)        | att20c409 |
| 4 AT&T 20C505 (S3)                             | att20c505 |
| 5 BrookTree BT481 (AGX)                        | bt481     |
| 6 BrookTree BT482 (AGX)                        | bt482     |
| 7 BrookTree BT485/9485 (S3)                    | bt485     |
| 8 Sierra SC15025 (S3, AGX)                     | sc15025   |
| 9 S3 GenDAC (86C708) (autodetected)            | s3gendac  |
| 10 S3 SDAC (86C716) (autodetected)             | s3_sdac   |
| 11 STG-1700 (S3, autodetected)                 | stg1700   |
| 12 STG-1703 (S3, autodetected)                 | stg1703   |

Enter a number to choose the corresponding RAMDAC.

Press enter for the next page, q to quit without selection of a RAMDAC.

用户如果不太清楚的话,一般 S3 的显示卡选择 9 没有什么问题,或者干脆选择 q 退出,如果不选择 DAC,系统会自己检测。

下面是设定显示卡上的时钟芯片。

A Clockchip line in the Device section forces the detection of a programmable clock device. With a clockchip enabled, any required clock can be programmed without requiring probing of clocks or a Clocks line. Most cards don't have a programmable clock chip.

Choose from the following list:

|    |                                                          |            |
|----|----------------------------------------------------------|------------|
| 1  | Chrontel 8391                                            | ch8391     |
| 2  | ICD2061A and compatibles (ICS9161A, DCS2824)             | icd2061a   |
| 3  | ICS2595                                                  | ics2595    |
| 4  | ICS5342 (similar to SDAC, but not completely compatible) | ics5342    |
| 5  | ICS5341                                                  | ics5341    |
| 6  | S3 GenDAC (86C708) and ICS5300 (autodetected)            | s3gendac   |
| 7  | S3 SDAC (86C716)                                         | s3_sdac    |
| 8  | STG 1703 (autodetected)                                  | stg1703    |
| 9  | Sierra SC11412                                           | sc11412    |
| 10 | TI 3025 (autodetected)                                   | ti3025     |
| 11 | TI 3026 (autodetected)                                   | ti3026     |
| 12 | IBM RGB 51x/52x (autodetected)                           | ibm_rgb5xx |

Just press enter if you don't want a Clockchip setting.

What Clockchip setting do you want (1-12)?

用户如果不知道的话也可以不选择,千万不要乱选,有可能造成 X 服务器不能正常工作。下面是让用户进行自动检测时钟,如果用户第一次运行时死机或者出问题退出后,下次就不要选择这一项,如果用户嫌麻烦也可以不选择。

For most configurations, a Clocks line is useful since it prevents the slow and nasty sounding clock probing at server start-up. Probed clocks are displayed at server startup, along with other server and hardware configuration info. You can save this information in a file by running 'X -probeonly 2 > output\_file'. Be warned that clock probing is inherently imprecise; some clocks may be slightly too high (varies per run).

At this point I can run X -probeonly, and try to extract the clock information from the output. It is recommended that you do this yourself and add a clocks line (note that the list of clocks may be split over multiple Clocks lines) to your Device section afterwards. Be aware that a clocks line is not appropriate for drivers that have a fixed set of clocks and don't probe by default (e.g. Cirrus). Also, for the P9000 server you must simply specify clocks line that matches the modes you want to use. For the S3 server with a programmable clock chip you need a 'ClockChip' line and no Clocks line.

You must be root to be able to run X -probeonly now.

Do you want me to run 'X -probeonly' now? n

下面是让用户选择每种显示模式的分辨率:

For each depth, a list of modes (resolutions) is defined. The default resolution that the server will start-up with will be the first listed

mode that can be supported by the monitor and card.

Currently it is set to:

"640x480" "800x600" "1024x768" "1280 x 1024" for 8bpp

"640x480" "800x600" "1024x768" for 16bpp

"640x480" "800x600" for 24bpp

"640x480" "800x600" for 32bpp

Note that 16, 24 and 32bpp are only supported on a few configurations.

Modes that cannot be supported due to monitor or clock constraints will be automatically skipped by the server.

- 1 Change the modes for 8pp (256 colors)
- 2 Change the modes for 16bpp (32K/64K colors)
- 3 Change the modes for 24bpp (24-bit color, packed pixel)
- 4 Change the modes for 32bpp (24-bit color)
- 5 The modes are OK, continue.

Enter your choice:

用户选择第一项就是选择 256 色的显示模式的分辨率顺序,例如输入 4321 就是缺省为  $1280 \times 1024$ , 以下依次为  $1024 \times 768$ ,  $800 \times 600$  及  $640 \times 480$ , 然后还会询问用户是否使用屏幕漫游功能。XFree86 缺省使用 256 色, 并且在 `xf86config` 中不好更改, 如果用户的显示卡的显存足够, 想使用 16bit/24bit 色彩, 用户可以用以下这些方法(选一种用就可以了):

1. 编辑用户根目录下的 `~/.xserverrc`  
`exec /usr/X11R6/bin/X -bpp 16`  
`startx`
2. 启动 X Window 使用下列参数  
`startx -- -bpp 16`
3. 使用 `xdm` 时候  
 改 `/usr/X11R6/lib/X11/xdm/Xserver`  
 在 `xdm` 加个参数 `-bpp 16`
4. 编辑 `/usr/X11R6/bin/startx`  
 增加一行:  
`serverargs = -bpp 16`
5. 手工修改 `/etc/XF86Config`  
 在 `Section Screen` 中加入一行  
`DefaultColorDepth 16`

如果用户要使用 24 位色彩 只要把以上的 16 改成 24 就可以。

I am going to write the XF86Config file now. Make sure you don't accidentally overwrite a previously configured one.

Shall I write it to /etc/XF86Config? Y

好了,大功告成,选择 Yes 将设定成果写入 XF86Config 中。

输入 startx,如果没什么问题的话,用户就可以看到 X Window 了。在 X Window 中可以通过按下 Ctrl + Alt + ' + ' 和 Ctrl + Alt + '-' 来切换分辨率, Ctrl + Alt + Backspace 强制退出 X Window, Ctrl + Alt + F1 + F7 在 X Window 和各个虚拟控制台之间切换。如果有问题,例如 X Window 不能运行,查看出错信息,重新配置一下,如果仍有问题,就先选用 VGA 或 SVGA 的 X 服务器,然后再询问有经验的用户。

如果用户觉得字符界面的设定程序不方便,就可以使用图形界面的 XF86Setup,如下图所示:

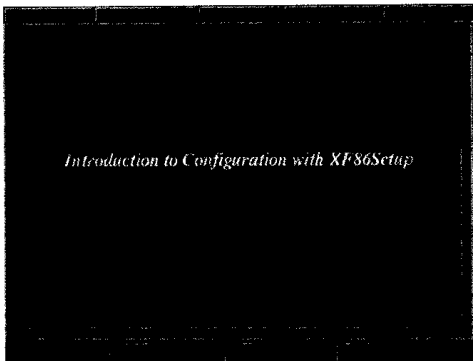


图 3.7 XF86Setup

实际上这个 XF86Setup 的工作模式同 xfree86config 是一样的,首先选择鼠标图 3.8:再选择键盘:

再选择显示卡,用户可以直接用鼠标来选择,要设置显存等选项就选择 detail setup 就可以设置图 3.9:

在这里选择显示器的扫描频率:

至于其他选项都可以不理睬,设置完毕之后就选择 done 就可以了。

在配置完 X 服务器之后,如果用户的 X Window 已经能够正常运行,只是屏幕是歪的,怎么办? 如果用户去调整显示器上的旋钮那自然是可以调整好的,可是回到 DOS/Windows 95 下那屏幕又歪了,实际上 X Window 可以在启动时自己调整,方法是首先将 /etc/XF86Config 这个文件拷贝到一个备份文件中例如 XF86Config.old,然后在 X Window 下先开两个 xterm,在其中一个 xterm 上编辑 /etc/XF86Config 这个文件,在另外一个 xterm 上输入命令:

```
xvidtune
```

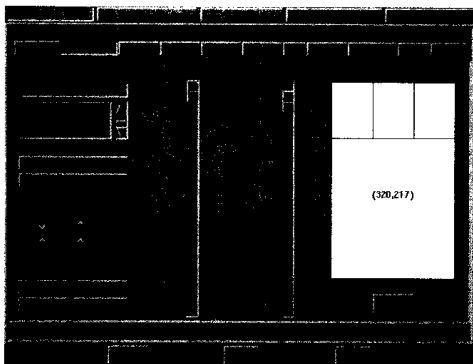


图 3.8 设置鼠标

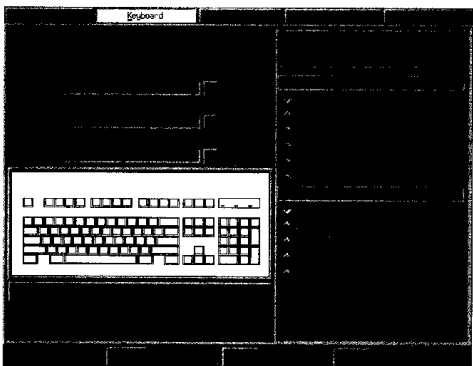


图 3.9 设置键盘

就会出现下面的画面：

这个程序可用来调整 X Window 的屏幕位置和大小,用户先按下 show 这个按钮就会在那个 xterm 上出现目前 X Window 所使用的显示模式,对照这一行输出在另外一个 Xterm 中找到 XF86Config 中相应的一行,然后在这个 xvidtune 中调整 left, right, wider, narrower, up, down, shorter 和 taller 等选项,然后不断用 test 选项来测试,如果用户满意

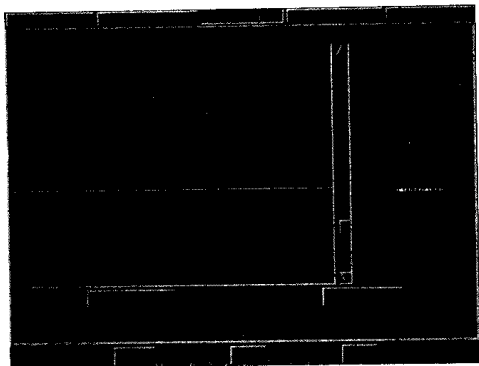


图 3.10 设置显示卡

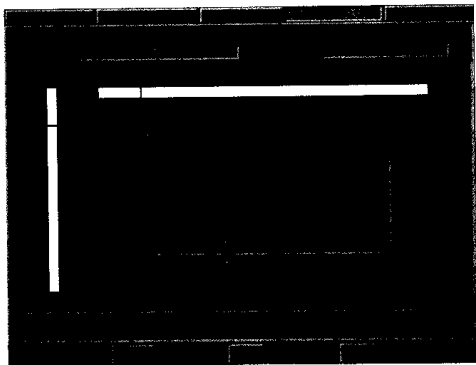


图 3.11 设置显示器

就选择 apply 接受这个设置,再选择 show 将目前设置输出到那个 Xterm 中,在另外一个 Xterm 中改变找到的那一行为新的设定值,这样每次启动 X Window 时就可以使用正确的屏幕设定。那么用户可能要问如何手工编辑 XF86Config 这个文件呢?下面笔者就介绍一下 XF86Config 的格式,实际上有很多功能只能通过手工编辑来解决,不过用户在编辑前切记要做备份,否则出了问题没有办法恢复。

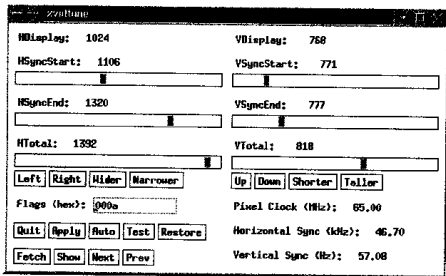


图 3.12 调整屏幕

XF86Config 由许多段落组成。每个段落的格式如下：

```
Section SectionName      段落名称
SectionEntry             设置
...
EndSection              段落结束符
```

一般有以下这些段落：

Files {File pathnames}

这是用来设定 X Window 使用字体的路径和调色板的路径,如果用户要增加字体路径就在这个段落中增加。格式为

FontPath 路径名

如果使用字体服务器 xfs 那么格式是

Fontpath tcp/lark:7000

这一行的意思是使用网络上 lark 这台机器上的 7000 端口上运行的字体服务器。

要改变 X Window 下使用的调色板的配置就使用下面的格式：

RGBPath 路径名

ServerFlags section

设定 X 服务器的一些参数：

NoTrapSignals 防止 X 服务器因为出错而退出,一般用户不用设置。

DontZap 这个选项可以屏蔽 Ctrl + Alt + Backspace 强制退出这个功能。

DontZoom 屏蔽 Ctrl + Alt + '+' 和 Ctrl + Alt + '-' 来调整 X 服务器的分辨率这个功能。

Keyboard section

设定键盘：

Protocol kbd-protocol 键盘协议可以是 Standard 或 Xqueue。

AutoRepeat delay rate 更改键盘重覆的速度。

ServerNumLock 让用户可以使用小键盘。

LeftAlt mapping

RightAlt mapping

AltGr mapping  
 ScrollLock mapping  
 RightCtl mapping

允许用户将上面的键映射到这些功能:

Meta  
 Compose  
 ModeShift  
 ModeLock  
 ScrollLock  
 Control  
 XLeds led 允许应用重新使用三个指示灯。  
 VTSysReq 允许切换到其他控制台。

#### Pointer section

设定鼠标参数:

| Protocol | protocol-type | 鼠标类型 |
|----------|---------------|------|
|          | BusMouse      |      |
|          | Logitech      |      |
|          | Microsoft     |      |
|          | MMSeries      |      |
|          | Mouseman      |      |
|          | MouseSystems  |      |
|          | PS/2          |      |
|          | MMHitTab      |      |
|          | Xqueue        |      |
|          | OSMouse       |      |

Device pointer-dev 鼠标端口,一般是/dev/mouse

BaudRate rate 串行鼠标的波特率,一般为 1200

Emulate3Buttons 用两键模拟三键

Emulate3Timeout timeout 模拟三键时,两键同时按下的时间间隔。缺省为 50 毫秒。

ChordMiddle 使用三键时,按下中间的键送出 left + right 事件。

SampleRate rate

ClearDTR

ClearRTS

适用于某些特殊鼠标,一般用户不用设置。

#### Monitor sections

设置显示器的参数,在 XF86Config 中可以有多不同的段落:

Identifier ID string 指定这个显示器的名称。每个段落只能有一个唯一的显示器名称。

VendorName vendor 设定显示器制造厂家名称。

ModelName model 显示器的型号

HorizSync horisync-range 显示器的水平扫描频率范围。用逗号分开,用减号表示范围,单位为

KHZ

VertRefresh vertrefresh-range 显示器的垂直扫描频率范围。用逗号分开,用减号表示范围,单

位为 KHZ。

Gamma gamma-value(s) 调整显示器的 gamma 值。用户不用更改

Mode name 设定显示器的工作模式以 Video Mode 开始, EndMode 结束。

#### Device sections

设定显示卡的参数:

Identifier ID string 显示卡的名称。

VendorName vendor 显示卡制造厂家名称。

BoardName model 显示卡型号。

Chipset chipset-type 显示卡的芯片型号。

Ramdac ramdac-type 显示卡的 RAMDAC 型号。

DacSpeed speed RAMDAC 的速率。

Clocks clock ... 显示卡的时钟。

VideoRam mem 显存大小。

#### Screen section

设定 X 服务器的参数:

Driver X 服务器的种类

Accel

Mono

SVGA

VGA2

VGA16

Device device-id 显示卡的段落名。

Monitor monitor-id 显示器的段落名。

ScreenNo scrnum 使用的虚拟控制台。

BlankTime time 屏幕保护时间缺省为 10 分钟。

SuspendTime time 屏幕进入休眠模式时间。缺省为 15 分钟。

OffTime time 关掉显示器的时间。缺省为 30 分钟。

SubSection Display 设定分辨率, 色彩深度等参数。

以上就是 XF86Config 的简单介绍, 有兴趣的用户可以自行设定。

### 3.2.2 设定窗口管理器

实际上 X 服务器只管理如何显示图形的界面, 如何相应键盘和鼠标的输入, 如何建立和摧毁一个窗口, 但是它并不提供用户界面, 用户可以输入 X 这条命令来观察一下没有窗口管理器(window manager 简称 wm)的 X Window 是什么样, 这时候只会出现一个什么也没有的图形窗口, 用户如果不用 Ctrl + Alt + Backspace 甚至不能退出这个窗口。但是实际上用户输入 startx 时会出现下面的画面:

这就是窗口管理器——fvwm95, 它提供管理桌面、提供虚拟工作台、建立窗口、移动窗口、改变窗口大小、将窗口变成图标和将图标变成窗口的功能。这一节我们就将介绍如何设置 X Window 下的窗口管理器 fvwm95 和最新的 AfterStep 以及 Enlightenment。

fvwm95 的全名是 F(?) Virtual Window Manager, 是专门为 X Window 设计的, 界面和用法类似于 Windows 95 界面的一个窗口管理器, fvwm95 是由 fvwm 演变而来, fvwm

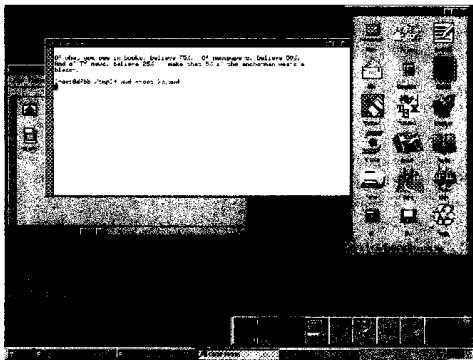


图 3.13 fvwm95 的画面

是从 twm 演变而来。fvwm95 提供了虚拟工作区,可以用键盘来切换和移动屏幕移动窗口和对窗口进行基本操作等,提供了外部和内部命令两种方式进行操作,并且提供窗口聚焦随鼠标的移动而变化的功能。甚至还提供事件音乐的功能,即像 Windows 95 一样在操作时有音响效果。

fvwm95 的窗口就像 Windows 95 一样,提供了 3D-look 的外观,要调整窗口大小,可将鼠标移到窗口的角落,此时会出现一个小 L 的光标,移动它来控制窗口的大小。当将鼠标移至窗口的标题栏时,按下鼠标左键便可以移动此窗口,按下中键可以显示出窗口的控制菜单(包含窗口的移动、放大、缩小、关闭和摧毁等),当用户将鼠标移到标题栏的左方的小图标,并按下鼠标任一键也会显示出窗口控制菜单。标题栏里用户最多可以定义 10 个按钮,缺省的有最小化最大化关闭这些按钮,并且用户可以用 Alt + Tab 键在不同的窗口之间切换。fvwm95 还提供了虚拟工作台给用户使用。用户既可以使用屏幕漫游来使用虚拟屏幕,也可以通过使用 pager 这个程序来切换不同的虚拟工作区。

fvwm95 启动时会先寻找配置文件。首先 fvwm95 会在用户的根目录中寻找一个叫 fvwm2rc95 的文件。如果找不到这个文件,就会使用系统的缺省文件 /usr/lib/X11/fvwm95-2/system.fvwm2rc95,如果再找不到,那 fvwm95 就无法正确执行了。(在 RedHat 下是用 m4 这个程序来处理 fvwm2rc95,结果这个配置文件被移到/etc/X11/TheNextLevel 下,上面的两个文件再怎么设置也没有作用。)

fvwm 95 有两种初始化模式: InitFunction 和 RestartFunction,在配置文件中 InitFunction 在第一次运行 fvwm95 时起作用,而 RestartFunction 是在用户重新启动 fvwm95 时起作用,用户还可以使用配置文件中的 AddToFunc 来增加自己需要的菜单和图标,用户也可以自己编写模块(modules)来让 fvwm95 增加更多的新功能。

下面我们通过分析一个 fvwm2rc95 文件来说明如何配置自己的 fvwm95:

```

#
# .fvwm2rc95 example configuration file for fvwm95-2.0.41
# .fvwm2rc95 的设定文件范例 fvwm95-2.0.41 版

# =====

#
# Fonts - one for window titles, another for icons, and another for the menus
# 第一个用来设定窗口标题栏的字体, 第二个是图标字体, 第三个是菜单的字体
WindowFont      -adobe-times-bold-i-*-*-14-*-*-*-*-*
IconFont        fixed
MenuFont        -adobe-helvetica-medium-r-*-*-140-*-*-*-*-*

#
# Set the default window colors. The syntax is:
# DefaultColors winforecolor winbackcolor titleforecolor titlebackcolor
# where:
# winbackcolor   is the background (border) color of all windows,
#                 including the menu windows
# winforecolor   is used for the menu text
# titlebackcolor is the background color used for title bars of
#                 non selected windows
# titleforecolor is the color of the title text of non selected windows
#
# 设定缺省的窗口颜色, 语法如下:
# DefaultColors 窗口前景颜色 窗口背景颜色 标题栏前景颜色 标题栏背景颜色
# 窗口背景颜色      : 给所有窗口用的背景(边界)颜色, 包括 menu 窗口
# 窗口前景颜色      : 菜单的文字的颜色
# 窗口标题背景颜色   : 在后台的窗口标题栏背景颜色
# 窗口标题前景颜色   : 在后台的窗口标题栏文字颜色
# 用户可以参考/usr/X11R6/lib/X11/rgb.txt 这个文件来确定使用什么样的颜色
#
DefaultColors Black #c0c0c0 #c0c0c0 grey51

#
# Text and background color for the title bar of the active (selected) window
# 前台的窗口文字和背景颜色
#
HighlightColors yellow #f0f0f0

#
# Text and background color for the title bar of sticky windows
# 锁定的窗口文字和背景颜色

```

```

#
StickyColors      blue #60c080

# You can also use the Style command to set the colors of individual
# title bars of some non-selected windows
# 可以使用 Style 命令来设定其他后台窗口的标题颜色
#
# Style * ForeColor #c0c0c0
# Style * BackColor grey51

# = = = = =

#
# Set up the major operating modes
# 设定主要的操作模式
#

Style * MouseFocus
# Style * SloppyFocus
# Style * ClickToFocus
# Comment the above and uncomment one of the following if you prefer
# focus follow mouse.
# 选择是否使用鼠标跟随功能,即:
# MouseFocus,只要鼠标移动到那个窗口,该窗口就变为前台
# 其他两个就是要鼠标单击才能激活窗口

# If we use the FvwmTaskBar we don't need the icons :-))
# Style ... Icon are also commented below.
# 因为有任务栏所以在这里取消了窗口图标化的功能
# 其格式为:
# Style 窗口名称 功能
Style * NoIcon
Style * MWMFunctions
Style * HintOverride
Style * MWMDecor
Style * DecorateTransient

OpaqueMoveSize 100
# 移动窗口时显示窗口的多少部分,
# 0:只显示边界 ---- 100:完全显示
EdgeScroll 100 100
# 虚拟桌面滚动方式,当鼠标光标移过一定百分比时,桌面便会切换到其他桌面
# 格式

```

```

# EdgeScroll 水平比例 垂直比例
EdgeResistance 10000 0
# 虚拟桌面移动条件
# 第一个数字是当鼠标移到边界多少毫秒之后便切换到下个虚拟桌面
# 第二个数字表示光标移动多少个像素之后才会移到下个窗口

#
# RandomPlacement prevents user interaction while placing windows;
# RandomPlacement 可以让用户开启新窗口时, 出现的位置是随机的
# SmartPlacement 由窗口管理器决定新窗口的位置, 一般是选一个空白的地方
Style * RandomPlacement
Style * SmartPlacement

#
# NoPPosition instructs fvwm to ignore the PPosition field in window
# geometry hints. Emacs annoyingly sets PPosition to (0,0)!
# 忽略 Pposition 域
Style * NoPPosition

# = = = = =
#
# Set the desk top size in units of physical screen size
# 设定虚拟桌面的个数
# DeskTopSize 水平 x 垂直
DeskTopSize 3x2

# = = = = =
#
# Module path and paths to the icons
# 模块的路径, 图标的路径
ModulePath /usr/X11R6/lib/X11/fvwm95-2/
PixmapPath /usr/include/X11/pixmaps/:/usr/lib/X11/mini-icons/
IconPath /usr/include/X11/bitmaps/:/usr/lib/TheNextLevel/pixmaps/

# = = = = =
#
# Set the decoration styles and window options
# If compatible styles are set for a single window in multiple Style
# commands, then the styles are ORed together. If conflicting styles
# are set, the last one specified is used.
# 设定特定窗口的风格, 如果同时设定一个窗口, 有冲突的地方以后面的为准。

```

```

# These commands should command before any menus or functions are defined.
# and before the internal pager is started.
# 这些设定必须在定义菜单和函数前面定义,否则不起作用
# 一般 NoTitle 是没有标题栏, StaysOnTop 是总在最前面, NoHandles 和 WindowListSkip
# 是不出现在任务栏
# Default border and handle widths
# 缺省的窗口边界大小
Style * BorderWidth 4, HandleWidth 4

Style Fvwm *      NoTitle, Sticky, WindowListSkip
Style Fvwm Pager  StaysOnTop, NoHandles
Style FvwmBanner  StaysOnTop
Style FvwmButtons NoTitle, NoHandles, Sticky, WindowListSkip, BorderWidth 3
Style * lock      NoTitle, NoHandles, Sticky, WindowListSkip
Style xbuff       NoTitle, Sticky, WindowListSkip

#
# Some mini-icons for the title bar
# 设定一些在窗口上的小图标
# 用户可以在下面设定自己的应用程序的图标, 注意图标文件必须放在前面设定的图标
# 路径中, 应用程序名支持通配符。
Style *          TitleIcon mini-x2.xpm
Style * xterm *   TitleIcon mini-term.xpm
Style rxvt       TitleIcon mini-term.xpm
Style xcalc      TitleIcon mini-calc.xpm
Style xsession   TitleIcon mini-bx2.xpm
Style xv         TitleIcon mini-xv.xpm
Style Netscape   TitleIcon mini-nscape.xpm
Style xmag       TitleIcon mini-zoom.xpm
Style textedit   TitleIcon mini-edit.xpm
Style tkps       TitleIcon mini-run.xpm
Style Ghostview  TitleIcon mini-gv.xpm
Style Xcolorsel  TitleIcon mini-colors.xpm
Style IDL *      TitleIcon mini-zoom.xpm
Style xeyes      TitleIcon mini-eyes.xpm
Style * fm *     TitleIcon mini-filemgr.xpm
# xman resource names
# 设定 xman 使用的小图标
Style topBox     TitleIcon mini-book1.xpm
Style help       TitleIcon mini-book2.xpm
Style manualBrowser TitleIcon mini-book2.xpm
# xpaint resource names
# 设定 xpaint 使用的小图标

```

```

Style xpaint          TitleIcon mini-paint.xpm
Style Canvas          TitleIcon mini-paint.xpm
Style fatbits         TitleIcon mini-paint.xpm
Style filebrowser     TitleIcon mini-olfolder.xpm

Style FvwmFile        TitleIcon mini-olfolder.xpm
Style XDbx            TitleIcon mini-bug2.xpm

# = = = = =

#
# Stuff to do at start-up
# 启动 fvwm95 时的工作

# AddToFunc 功能名称、动作方式、具体模块或动作
# 动作方式有 I          当此功能被使用时，立即动作
#           M          则是鼠标有动作时才启动
#           D          双击鼠标键才会动作
AddToFunc InitFunction I Exec xsetroot -solid cyan4 &
+
+ I Module FvwmBanner
+ I Module FvwmButtons
+ I Module FvwmTaskBar
+ I Module FvwmPager 0 3

AddToFunc RestartFunction I Module FvwmButtons
+ I Module FvwmTaskBar
# + I Module FvwmPager 0 3

# = = = = =

# Now define some handy complex functions
# 现在开始定义一些复杂的功能以方便用户使用
# 主要是定义鼠标的工作方式，用户可以不理会
# This one moves and then raises the window if you drag the mouse,
# only raises the window if you click, or does a full maximize if
# you double click
AddToFunc Move-or-Raise M Move
+ M Raise
+ C Raise
+ D Maximize 100 100

#
# This one moves and then lowers the window if you drag the mouse,

```

```

# only lowers the window if you click, or does a RaiseLower if you double
# click

AddToFunc Move-or-Lower M Move
+                               M Lower
+                               C Lower
+                               D RaiseLower

#
# This one moves or (de)iconifies:
AddToFunc Move-or-Iconify M Move
+                               D Iconify

#
# This one resizes and then raises the window if you drag the mouse,
# only raises the window if you click, or does a RaiseLower if you double
# click
AddToFunc Resize-or-Raise M Resize
+                               M Raise
+                               C Raise
+                               D RaiseLower

#
# ===== #

#
# Now define the menus - defer bindings until later
# 定义菜单
# This is for the Start menu of the FvwmTaskBar
# 首先定义开始菜单
# AddToMenu 菜单名称 显示的信息 Title
# + 显示讯息 %图标文件名% 执行的相应操作
# 操作包括: Exec 执行程序, 后面跟程序名称及参数
# Popup 弹出新菜单, 后面跟菜单名
# Nop 分割线
# Refresh 重新显示屏幕

AddToMenu StartMenu
+ New shell %mini-sh1.xpm% Exec color-xterm -ls -fb 8x13 &
+ Manual pages %mini-book1.xpm% Exec xman &
+ Magnifying glass %mini-zoom.xpm% Exec xmag &
+ Applications %mini-x2.xpm% Popup Applications
+ Games %mini-happy.xpm% Popup Games
+ Nop
+ Lock Screen %mini-lock.xpm% Exec xlock &

```

|                                        |                   |                                  |                                          |             |
|----------------------------------------|-------------------|----------------------------------|------------------------------------------|-------------|
| +                                      | Refresh Screen    | % mini-ray .xpm%                 | Refresh                                  |             |
| +                                      |                   |                                  | Nop                                      |             |
| +                                      | Exit Fvwm         | % mini-stop .xpm%                | Popup                                    | Quit-Verify |
| AddToMenu Shells Shells Title          |                   |                                  |                                          |             |
| +                                      | Xterm             | (7x14 font)% mini-term .xpm%     | Exec xterm -sb -sl 500 -j -ls -fn 7x14 & |             |
| +                                      | Color Xterm       | (7x14 font)% mini-term .xpm%     | Exec color-xterm -tb -sl 500 -j -ls -fn  |             |
| 7x14 -fb 7x14bold -title Color xterm & |                   |                                  |                                          |             |
| +                                      | Color Xterm       | (8x13 font)% mini-term .xpm%     | Exec color-xterm -sb -sl 500 -j -ls -fn  |             |
| 8x13 -title Color xterm &              |                   |                                  |                                          |             |
| +                                      |                   |                                  | Nop                                      |             |
| +                                      | Large Xterm       | (10x20 font)% mini-display .xpm% | Exec nxterm -sb -sl 500 -j -ls -fn       |             |
| 10x20 &                                |                   |                                  |                                          |             |
| +                                      | Large Color Xterm | (10x20 font)% mini-display .xpm% | Exec color-xterm -sb -sl 500 -j -ls -fn  |             |
| 10x20 &                                |                   |                                  |                                          |             |
| +                                      |                   |                                  | Nop                                      |             |

```
+ Worm% mini-lball.xpm%      Exec xlock -nolock -nice 0 -mode worm &
+ World% mini-lball.xpm%      Exec xlock -nolock -nice 0 -mode world &
```

# 下面是屏幕锁定程序

AddToMenu Screenlock Lock Screen Title

```
+ Bat% mini-rball.xpm%      Exec xlock -nice 0 -mode bat &
+ Blank% mini-rball.xpm%     Exec xlock -nice 0 -mode blank &
+ Blot% mini-rball.xpm%      Exec xlock -nice 0 -mode blot &
+ Bounce% mini-rball.xpm%    Exec xlock -nice 0 -mode bounce &
+ Flame% mini-rball.xpm%     Exec xlock -nice 0 -mode flame &
+ Galaxy% mini-rball.xpm%    Exec xlock -nice 0 -mode galaxy &
+ Grav% mini-rball.xpm%      Exec xlock -nice 0 -mode grav &
+ Helix% mini-rball.xpm%     Exec xlock -nice 0 -mode helix &
+ Hop% mini-rball.xpm%       Exec xlock -nice 0 -mode hop &
+ Hyper% mini-rball.xpm%     Exec xlock -nice 0 -mode hyper &
+ Kaleid% mini-rball.xpm%    Exec xlock -nice 0 -mode kaleid &
+ Life% mini-rball.xpm%      Exec xlock -nice 0 -mode life &
+ Life3d% mini-rball.xpm%    Exec xlock -nice 0 -mode life3d &
+ Maze% mini-rball.xpm%      Exec xlock -nice 0 -mode maze &
+ Pyro% mini-rball.xpm%      Exec xlock -nice 0 -mode pyro &
+ Qix% mini-rball.xpm%       Exec xlock -nice 0 -mode qix &
+ Random% mini-rball.xpm%    Exec xlock -nice 0 -mode random &
+ Rect% mini-rball.xpm%      Exec xlock -nice 0 -mode rect &
+ Rock% mini-rball.xpm%      Exec xlock -nice 0 -mode rock &
+ Rotor% mini-rball.xpm%     Exec xlock -nice 0 -mode rotor &
+ Sphere% mini-rball.xpm%    Exec xlock -nice 0 -mode sphere &
+ Spline% mini-rball.xpm%    Exec xlock -nice 0 -mode spline &
+ Swarm% mini-rball.xpm%     Exec xlock -nice 0 -mode swarm &
+ Water% mini-rball.xpm%     Exec xlock -nice 0 -mode wator &
+ Worm% mini-rball.xpm%      Exec xlock -nice 0 -mode worm &
+ World% mini-rball.xpm%     Exec xlock -nice 0 -mode world &
```

# 下面是游戏程序

AddToMenu Games Games Title

```
+ Abuse% %                  Exec abuse &
+ Maze% mini-maze.xpm%      Exec maze &
+ Spide% mini-exp.xpm        Exec exec spider &
+ Xplayed% mini-cdlabel.xpm% Exec exec xplayed &
+ Xboard% mini-slom.xpm%     Exec xboard &
+ Xeyes% mini-eyes.xpm%      Exec xeyes &
+ Xhextris% mini-hextris.xpm Exec exec xhextris &
# +                          Xlander Exec exec xlander &
+ Xlogo% mini-bx2.xpm%       Exec xlogo &
```

```

+ Xmahjong% mini-mahjongg.xpm%      Exec exec xmahjongg &
# + Xroach% mini-roach.xpm%          Exec exec xroach &
+ Xtetris% mini-tetris.xpm%          Exec xtetris &
# + Xvier Exec exec xvier &
+ Xbill% mini-cross.xpm%             Exec xbill &
+ Xpcman% mini-question.xpm          Exec xchomp &
+ Bomb% mini-bomb.xpm%               Exec xdemicur &
+ Jewel% mini-question.xpm           Exec xjewel &
+ Xsnow                               Exec xsnow &
+ XEarth% mini-x2.xpm                Exec xearth &

# 下面是应用程序
AddToMenu Applications Applications Title
+ Netscape% mini-nscape.xpm%         Exec netscape &
+ Ghostview% mini-gv.xpm%            Exec ghostview &
+ Xview% mini-xv.xpm%                Exec xv &
+ Xedit% mini-edit.xpm%              Exec xedit -geometry 630x490 + 95 + 5 &
+ Xedit% mini-edit.xpm%              Exec xedit &
# + Xfilemanager% mini-filemgr.xpm%   Exec xfilemanager &
+ Xfm% mini-filemgr.xpm%             Exec xfm &
# + Xfractint% mini-fractal.xpm%       Exec xvt -font 7x14 -e xfractint map=chroma &
+ Xpaint% mini-paint.xpm%            Exec xpaint &
+ Xgrab% mini-camera.xpm%            Exec xgrab &
+ Xcolormap% mini-colors.xpm%         Exec xcmmap &
+ Xgdb% mini-bug2.xpm%               Exec xgdb &
+ gdb% mini-bug1.xpm%                Exec gdb &
+ Xwpe% mini-x.xpm%                  Exec xwpe &

#
# This menu is invoked as a sub-menu - it allows you to quit,
# restart, or switch to another WM.
# 下面是退出菜单
AddToMenu Quit-Verify Really Quit Fvwm? Title
+ Restart% mini-turn.xpm%             Restart fvwm95-2
+
+ Start Fvwm% mini.fvwm.xpm%           Restart fvwm
+ Start olvwm% mini-olvwm.xpm%         Restart /usr/openwin/bin/olvwm
+ Start olwm% mini-olwm.xpm%          Restart /usr/openwin/bin/olwm
+
+ Yes, Really Quit% mini-exclam.xpm%   Quit
+ No, Don't Quit% mini-cross.xpm%     Nop

#

```

```
# Provides a list of modules to fire off
# 直接运行模块的菜单
AddToMenu Module-Popup Modules Title
+ Buttons                               Module FwrmButtons
+ Audio% mini-audiovol.xpm%           Module FwrmAudio
+ Clean-Up% mini-pencil.xpm%          Module FwrmClean
+ Task bar                             Module FwrmTaskBar
+ Identify% mini-question.xpm%        Module FwrmIdent
+ Save Desktop                         Module FwrmSave
+ Debug% mini-bug2.xpm%               Module FwrmDebug
+ Pager                                Module FwrmPager 0 3
+ Window List% mini-windows.xpm%      Module FwrmWinList
+ Talk                                 Module FwrmTalk
+ Configure% mini-hammer.xpm%         Module FwrmConfig

#
# This menu will fire up some very common utilities
# 常用工具
AddToMenu Utilities Utilities Title
+ Top% mini-run.xpm%                  Exec rxvt -T Top -n Top -e top &
+ Calculator% mini-calc.xpm%          Exec xcalc &
+ Manual pages% mini-book1.xpm%       Exec xman &
+ Magnifying glass% mini-zoom.xpm%    Exec xmag &
+ Oclock% mini-oclock.xpm%           Exec o'clock &
+                                     Noop
+ Applications% mini-x2.xpm%          Popup Applications
+ Shells% mini-term.xpm               Popup Shells
+ Games% mini-happy.xpm%              Popup Games
+                                     Noop
+ Screensaver% mini-display.xpm%      Popup Screensaver
+ Lock Screen% mini-lock.xpm%         Popup Screenlock
+                                     Noop
+ Modules% mini-modules.xpm%          Popup Module-Popup
+                                     Noop
+ Refresh Screen% mini-ray.xpm%       Refresh
+                                     Noop
+ Exit Fwrm% mini-stop.xpm%           Popup Quit-Verify

#
# This defines the most common window operations
# 窗口操作菜单
AddToMenu Window Window Ops Title
- Move                                Function Move-or-Raise
```

```

+ Resize                Function Resize-or-Raise
+ Raise% mini-raise.xpm% Raise
+ Lower% mini-lower.xpm% Lower
+ (De)Iconify           Iconify
+ { Un}Stick            Stick
+ { Un}Maximize         Maximize 100 100
+                        Nop
+ Kill                  Destroy
+ Close                 Delete
+                        Nop
+ Switch to...          Window List
+ Refresh Screen% mini-ray.xpm% Refresh

#
# A trimmed down version of Window Ops, good for binding to decorations
# 窗口操作菜单
AddToMenu Window-Ops2 Restore      Iconify -1
+                                Move      Move-or-Raise
+                                Size      Resize-or-Raise
+                                Minimize  Iconify 1
+                                Maximize  Maximize t00 100
+                                Nop
+                                Kill      Destroy
+                                Close Alt + F4 Delete

# = = = = =

#
# One more complex function - couldn't be defined earlier because it used
# pop-up menus
#
# This creates the system menu for the title-bar window-ops
# pop-up
#
AddToFunc window_ops_func C PopUp Window-Ops2
+      M PopUp Window-Ops2
+      D Delete

# = = = = =

#
# Mouse bindings
# 设定鼠标功能

```

```
# First, for the mouse in the root window
# Button 1 gives the Utilities menu
# Button 2 gives the Window Ops menu
# Button 3 gives the WindowList (like TwmWindows)
# I use the AnyModifier (A) option for the modifier field, so you can hold down
# any shift-control-whatever combination you want!
# 首先, 确定鼠标在 root window 时(左, 中, 右 = 1,2,3 ,0 = any)
# 按左键会弹出 Utility 菜单
# 按中键会弹出 Window Ops 菜单
# 按右键会弹出 WindowList(窗口列表)
# Context : 描述在何处使用这种鼠标功能.
#           R : root window
#           W : 任何应用程序的窗口
#           T : 标题栏
#           S : 窗口的边界
#           F : 窗口的四个角
#           I : 图标
#           0 - 9 : 标题栏上的按钮
#           A : 除了标题栏上的按钮以外的上述各项
# Modifier : 需要按下的键盘上的控制键
#           N : 不配合任何键
#           C : Control 键
#           S : Shift 键
#           M : Meta 键
#           A : C,S,M 任意一个
# Function : Fvwm95 的内部功能
```

| # | Button    | Context | Modifi | Function                             |
|---|-----------|---------|--------|--------------------------------------|
|   | Mouse 1   | R       | A      | Menu Utilities Nop                   |
|   | Mouse 2   | R       | A      | Menu Window Ops Nop                  |
|   | Mouse 3   | R       | A      | WindowList                           |
|   | # Mouse 3 | R       | A      | Module winlist FvwmWinList transient |

```
# Now the title bar buttons
# Any button in the left title-bar button gives the window ops menu
# Any button in the first right title-bar button Iconifies the window
# Any button in the second title-bar button full-maximizes
# Any button in the third (rightmost) title-bar button closes
# Note the use of Mouse 0 for AnyButton.
# 现在定义鼠标在标题栏时的功能
# Button Context Modif Function
Mouse 0 I A Function window_ops_func
```

```

Mouse 1      2      A      Delete
Mouse 0      4      A      Maximize 100 100
Mouse 0      6      A      Iconify

```

```

#
# Now the rest of the frame
# Here I invoke my complex functions for Move-or-lower, Move-or-raise,
# and Resize-or-Raise.
# 现在开始定义鼠标在边界时的功能
# Button 1 in the corner pieces, with any modifiers, gives resize or raise
Mouse 1      F      A      Function Resize-or-Raise
# Button 1 in the title, sides, or icon, w/ any modifiers, gives move or raise
Mouse 1      TS     A      Function Move-or-Raise

```

```

# Button 1 in an icons gives move for a drag, de-iconify for a double-click,
# nothing for a single click
# Button 2 in an icon, w/ any modifiers, gives de-iconify
Mouse 1      I      A      Function Move-or-Iconify
Mouse 2      I      A      Iconify

```

```

# Button 2 in the corners, sides, or title-bar gives the window ops menu
Mouse 2      FST    A      Function window_ops_func

```

```

# Button 3 anywhere in the decoration (except the title-bar buttons)
# does a raise-lower
Mouse 3      TSIF   A      RaiseLower
# Button 3 in the window, with the Modifier-1 key (usually alt or diamond)
# gives Raise-Lower. Used to use control here, but that interferes with xterm
Mouse 3      W      M      RaiseLower

```

```

# =====

```

```

# Now some keyboard shortcuts.
# 现在设定键盘
# Arrow Keys
# press arrow + control anywhere, and scroll by 1 page
# Ctrl+方向键 = 向方向键的方向移动一个虚拟桌面
Key Left     A      C      Scroll -100 0
Key Right    A      C      Scroll +100 +0
Key Up       A      C      Scroll +0 -100
Key Down     A      C      Scroll +0 +100

```

# press arrow + meta key, and scroll by 1/10 of a page

# Alt + 方向键 = 向方向键的方向移动 1/10 个虚拟桌面

|           |   |   |               |
|-----------|---|---|---------------|
| Key Left  | A | M | Scroll -10 +0 |
| Key Right | A | M | Scroll +10 +0 |
| Key Up    | A | M | Scroll +0 -10 |
| Key Down  | A | M | Scroll +0 +10 |

# press shift arrow + control anywhere, and move the pointer by 1% of a page

# Shift + Ctrl + 方向键 = 移动鼠标光标 1% 个虚拟桌面

|           |   |    |                  |
|-----------|---|----|------------------|
| Key Left  | A | SC | CursorMove -1 0  |
| Key Right | A | SC | CursorMove +1 +0 |
| Key Up    | A | SC | CursorMove +0 -1 |
| Key Down  | A | SC | CursorMove +0 +1 |

# press shift arrow + meta key, and move the pointer by 1/10 of a page

# Shift + Alt + 方向键 = 移动鼠标光标 10% 个虚拟桌面

|           |   |    |                   |
|-----------|---|----|-------------------|
| Key Left  | A | SM | CursorMove -10 +0 |
| Key Right | A | SM | CursorMove +10 +0 |
| Key Up    | A | SM | CursorMove +0 -10 |
| Key Down  | A | SM | CursorMove +0 +10 |

# Keyboard accelerators

# 设定功能键

|        |   |   |                                        |
|--------|---|---|----------------------------------------|
| Key F1 | A | M | Popup Utilities 弹出工具菜单                 |
| Key F2 | A | M | Popup Window Ops 弹出窗口操作菜单              |
| Key F3 | A | M | Module WindowList FvwmWinList 弹出窗口列表菜单 |
| Key F4 | A | M | Iconify 窗口最小化                          |
| Key F5 | A | M | Move 移动窗口                              |
| Key F6 | A | M | Resize 改变窗口大小                          |

# Page Up/Page Down keys are used to scroll by one desktop page

# in any context, press page up/down + control

# in root context, just pressing page up/down is OK

#

# I prefer the non-wrapping scroll. These are for example purposes only

|             |   |   |                  |
|-------------|---|---|------------------|
| # Key Next  | A | C | Scroll 100000 0  |
| # Key Next  | R | N | Scroll 100000 0  |
| # Key Prior | A | C | Scroll -100000 0 |
| # Key Prior | R | N | Scroll -100000 0 |

|            |   |    |                  |
|------------|---|----|------------------|
| Key Tab    | A | M  | Prev Focus 前一个窗口 |
| Key Tab    | A | MS | Next Focus 下一个窗口 |
| Key Escape | A | C  | WindowList 窗口列表  |

```

# ===== #
# Definitions used by the modules
# 模块的设置定义
# ----- Pager 的部分
* FvwmPagerBack blue
#     背景颜色
* FvwmPagerFore red
#     前景颜色
* FvwmPagerFont -adobe-helvetica-bold-r *- *-10- *- *- *- *- *- *
#     字体
* FvwmPagerHilight #c0c0c0
#     高亮颜色
* FvwmPagerGeometry -1500-1500
#     位置
* FvwmPagerSmallFont 5x8
#     小字体
# * FvwmPagerRows 3
#     行数
# * FvwmPagerColumns 2
#     列数
# * FvwmPagerLabel 0 Desktop
#     第 n 个桌面的名称

# ----- FvwmButtons 的部分 (from 2.0.42)
* FvwmButtonsFore Black
#     前景颜色
* FvwmButtonsBack yellow
#     背景颜色
* FvwmButtonsFont -adobe-helvetica-bold-r *- *-10- *- *- *- *- *- *
#     字体
# Geometry - really likes to pick its own size, but giving a position is OK
* FvwmButtonsGeometry -5-34
#     位置
# Layout: specify rows or columns, not both
* FvwmButtonsRows 1
#     列数
# Define the buttons to use.....
# 定义桌面上常用的按钮
* FvwmButtons(Title xclock, Icon clock.xpm, \
    Swallow(UseOld) xclock 'Exec xclock -bg \ yellow \
    -padding 0 -geometry -1500-1500 &')
* FvwmButtons(2x1, Title load, \

```

```

Swallow(UseOld) xload 'Exec xload \
    -nolabel -bg yellow -update 5 -geometry -1500-1500 &')
# * FvwmButtons(Title xterm, Icon xterm.xpm, \
    Action 'Exec XTerm color-xterm \
        -ls -sb -fn 8x13 -title Color xterm &')
# * FvwmButtons(Title xfm, Icon FileManager.xpm, \
    Action 'Exec Xfm xfm -title File Manager &')
# * FvwmButtons(Title textedit, Icon TextEditor.xpm, \
    Action 'Exec Textedit xedit &')
# * FvwmButtons(Title xcalc, Icon Calculator.xpm, \
    Action 'Exec Calculator xcalc &')
# * FvwmButtons(Title xview, Icon xv.color.xpm, \
    Action 'Exec xv xv &')
# * FvwmButtons(Title gview, Icon gv.xpm, \
    Action 'Exec ghostview ghostview &')
# * FvwmButtons(Title xcin, Icon chinese.xpm, \
    Action 'Exec xcin xcin -fab5 etl5 -fn vga8x15 &')
# * FvwmButtons(Title netscape, Icon netscape_3d.xpm, \
    Action 'Exec Netscape netscape -geometry 920x622+90+5 &')
* FvwmButtons(Title xkill, Icon bomb.xpm, Action Destroy)
* FvwmButtons(2x1, Swallow(UseOld) Desktop 'FvwmPager 0 0')

# ----- 设定窗口信息的颜色和字体
* FvwmIdentBack #000080
#     背景颜色
* FvwmIdentFore Yellow
#     前景颜色
* FvwmIdentFont -adobe-helvetica-medium-r-*-*12-*-*-*-*-*
#     字体

# ----- 设定窗口列表的颜色和字体
* FvwmWinListBack #c0c0c0
#     背景颜色
* FvwmWinListFore Black
#     前景颜色
* FvwmWinListFont -adobe-helvetica-bold-r-*-*10-*-*-*-*-*
#     字体
* FvwmWinListAction Click1 Iconify -1,Focus
#     mouse 左键为跳至此窗口
* FvwmWinListAction Click2 Iconify
#     mouse 中键为最小化
* FvwmWinListAction Click3 Module FvwmIdent FvwmIdent
#     mouse 右键为显示其窗口信息

```

```
* FvwmWinListUseSkipList
# 不显示忽略的窗口
* FvwmWinListGeometry + 0-1
# 窗口列表菜单的位置
* FvwmWinListShowCurrentDesk
# 显示目前桌面上的窗口
```

```
#----- 设定任务栏的参数
```

```
Style FvwmTaskBar NoTitle, BorderWidth 4, HandleWidth 4, Sticky, StaysOnTop, WindowListSkip,
```

```
CirculateSkip
```

```
* FvwmTaskBarGeometry + 0-0
# 位置
* FvwmTaskBarFore Black
# 前景颜色
* FvwmTaskBarBack # c0c0c0
# 背景颜色
* FvwmTaskBarTipsFore black
# 提示的前景颜色
* FvwmTaskBarTipsBack bisque
# 提示的背景颜色
* FvwmTaskBarFont -adobe-helvetica-medium-i-*-*-*120-*-*-*-*-*
# 字体
* FvwmTaskBarSelFont -adobe-helvetica-bold-r-*-*-*120-*-*-*-*-*
# 选择到的窗口的字体
* FvwmTaskBarAction Click1 Iconify -1, Raise, Focus
# 左键 ---- 切换至此窗口
* FvwmTaskBarAction Click2 Iconify
# 中键 ---- 将该窗口最小化
* FvwmTaskBarAction Click3 Module FvwmIdent FvwmIdent
# 右键 ---- 显示该窗口的信息
* FvwmTaskBarUseSkipList
# 不显示忽略的窗口
* FvwmTaskBarAutoStick
# 显示在确定的位置
* FvwmTaskBarStartName Start
# 开始菜单的名称
* FvwmTaskBarStartMenu StartMenu
# 开始菜单的菜单项
* FvwmTaskBarStartIcon mini-start.xpm
# 开始菜单的图标
* FvwmTaskBarShowTips
# 是否显示提示
```

```
# * FvwmTaskBarShowTransients
# * FvwmTaskBarClockFormat %l:%M%p
    任务栏的时间格式
# * FvwmTaskBarHighlightFocus
# FvwmTaskBarAutoHide
# 是否自动隐藏
# FvwmTaskBarMailCommand Exec color-xterm -T Mail -ls -fn 8x13 -c pine -i
# 有 E-mail 来的时候启动的 mail 服务程序。
```

以上就是 fvwm95 的设定文件的格式,用户可以自行对照并设定自己的 fvwm95,另外如果用户希望使用中文的 fvwm95,也有于明俭(yumj@sun.ihep.ac.cn)先生汉化的 cfvwm95,在本书的配套光盘中有该程序,该 fvwm95 的全部菜单都是中文的,如下图:

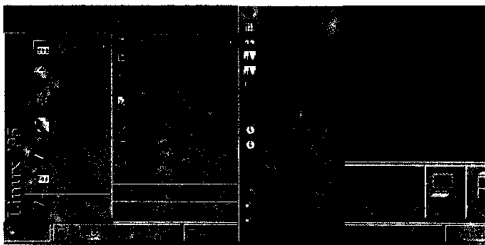


图 3.14 中文 fvwm95

对于那些喜欢尝新鲜的用户来说,fvwm95 可能太普通了,下文就将向读者介绍两个最酷的窗口管理器——AfterStep 和 enlightenment。

AfterStep 是从一个叫 BowMan 的窗口管理器演变而来的模拟 NextStep(一种非常漂亮的图形工作站,是由 Steve Jobs 的公司出品的)的一个窗口管理器,其漂亮程度的确不是 fvwm95 能望其项背的。同 fvwm95 比较 AfterStep 有下列优点:

(1)华丽的图标和界面,所有的图标都是 3D 的立体图像,连窗口的标题栏也是渐变色的。不过用户的显存没有 2M 是别想玩 AfterStep 的,一定要将 X Window 至少设成 16bit 色彩才行。不过 AfterStep 提供两套图标,一套是 65536 色的,一套是 256 色的,用户可以根据需要选择使用。

(2)可以自由组合屏幕上的图标,并且可以将所有虚拟窗口合成一个真正的大窗口。

(3)也有事件音效,并且可以用菜单设定屏幕的背景,最有趣的功能就是屏幕背景可以是动画效果,这是笔者所看到的所有窗口管理器中唯一支持这种功能。

(4)设定文件格式简单,Afterstep 的设定文件 .steprc 的格式比 fvwm 的 .fvwm2rc95 的个是要简单得多,对于初学者来说更容易入门一些。

下面笔者简单介绍下如何取得并安装最新版的 AfterStep(在本书所附光盘中带有 AfterStep 的源代码),请到 <http://mango.sfasu.edu/~frank/afterstep/> 这个地址去下载最

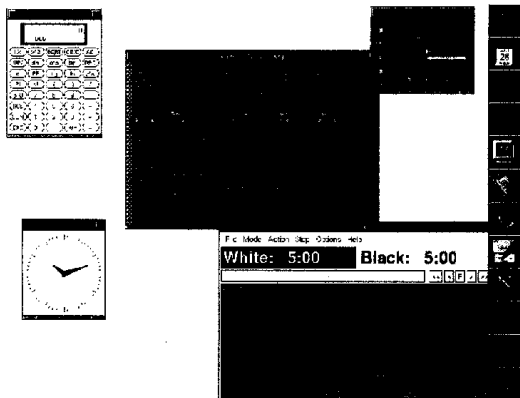


图 3.15 AfterStep 的画面

新的 AfterStep 和它的补丁程序。安装 AfterStep 也十分简单,将取回的文件在一个目录中展开,以超级用户身份登录,然后运行 Installme,就会出现下面的文字:

```
#####
# AfterStep v1.0 Installation #
#####
This Installation must be run as root...
If you are not root, press control-C and
either login as root or su to root.
Otherwise, press < Enter > and sit back
while your system is being upgraded to
AfterStep! < Enter >
Are you using an 8bpp display? (Y/N) -Answer Y if uncertain.
```

询问用户是否使用 65536 色的显示模式,如果用户不确定,回答 N,回答 N 就只能使用 256 色的图标。不过这会节省 AfterStep 消耗的系统资源。

```
Enter up to ten users who will be using AfterStep (eg, frank dan root):
```

```
----- NoTE -----
```

```
Each will receive new .xsession, .xinitrc, .Xdefaults, and .steprc
files. Existing files will be saved and appended with a .before
extension. Press < Enter > if you would like to finish without
configuring any users.
```

```
root lenx
```

这里要让用户选择是否让更多的用户使用 AfterStep 做窗口管理器,用户最多可以输入十个用户名,中间用空格隔开。

输入用户列表完毕后按回车系统就开始编译并自动安装 AfterStep,如果没有发现出错信息,用户就可以使用 AfterStep 了,如果有错误,用户就需要手工修改 Makefile,直到能自行执行 make 不出错时再重新运行 Installme 就可以了。

如果用户要给 AfterStep 加上事件音效,编辑用户根目录下的 .steprc 文件,找到

```
# * AudioPlayCmd /usr/bin/showaudio
```

这一行,将这一行的 # 号除去,并且设置可以播放 au 文件的播放器的正确路径,缺省是 /usr/bin/showaudio。如果用户的 Linux 系统已经支持声霸卡并且已经在 /dev 目录下制造出 /dev/audio 这个设备,那么直接编辑一个 shell 文件 showaudio 就可以了:

```
# ! /bin/sh
```

```
cat $1 > /dev/audio
```

然后编辑下面这些设定(将 # 号除去)

```
# * Audio startup gong.au
# * Audio shutdown clink.au
# * Audio add_window bong.au
# * Audio raise_window drip.au
# * Audio lower_window drip.au
# * Audio focus_change drip.au
# * Audio destroy_window crash.au
# * Audio iconify ploop.au
# * Audio deiconify ploop.au
# * Audio toggle_paging fwop.au
# * Audio new_page beaming.au
# * Audio new_desk beam_trek.au
# * Audio configure_window hoh.au
# * Audio window_name bong.au
# * Audio icon_name beep.au
# * Audio res_class bong.au
# * Audio res_name beep.au
# * Audio end_windowlist twang.au
```

这样用户的 AfterStep 就会有音效了。

如果把漂亮的 AfterStep 比作天使的话,那么 Enlightenment 就是窗口管理器中的魔鬼,其阴森恐怖的风格,绝对让那些喜欢奇异风格的用户大呼过瘾。下面就是它的画面(见图 3.16)。

目前比较热门的窗口管理器还有 KDE, WindowMaker, ICEWM, QVWM, XFCE 等。

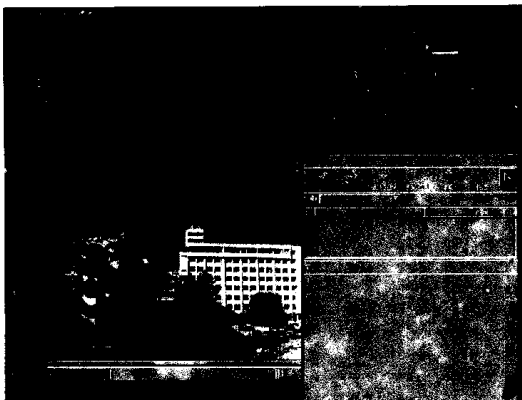


图 3.16 Enlightenment 的画面

### 3.3 /etc 下的配置文件剖析

下面是 Linux 文件系统的目录结构：

```

/_/_/_bin
|_boot
|_dev
|_cdrom
|_etc
|_home_/_/_/_ftp
|_/_/_fangh
|_lib
|_lost+found
|_mnt
|_proc
|_root
|_sbin
|_shlib
|_tmp
|_usr_/_/_/_X11R6
|_/_bin
|_/_sbin

```

```

|   |_ etc
|   |_ doc
|   |_ include
|   |_ openwin
|   |_ lib
|   |_ local_ _ _ _ bin
|   |   |_ include
|   |   |_ lib
|   |_ man
|   |_ src_ _ _ _ linux
|_ var_ _ _ _ log
    |_ spool_ _ _ _ mail
        |_ lpd

```

Linux 的文件系统条理分明,但是初学者往往不知道它们的用途,虽说不影响到普通的操作,用户要想得心应手地使用 Linux 就必须了解各个重要目录的用途。下面就简单介绍一下 Linux 的目录结构和/etc 下的配置文件。

Linux 是由一个内核(vmlinuz)加上各个大小功能不同的目录和应用程序组合起来的,整个文件系统是在一个根目录下延伸出来的,这个根叫作“/”同 DOS 下的根目录是“\”正好相反。上图就是一个简略的目录树,对于 Linux 的不同发行套件,这些目录也有一些细微的差别,但是主要的目录结构是一样的。下面就把主要的文件和目录介绍一下:  
/vmlinuz

这个文件就是系统内核。系统启动时加载进内存,整个系统运行于其上。另外在安装 LILO 时有用,用户在自己重新编译内核之后要让新的核心起作用就必须重新运行 lilo。  
/bin

bin 是 binary 的缩写,在传统的 UNIX 系统中这个目录存放了使用者最经常使用的命令,如 cp,ls,mv 等等。  
/boot

这是存放给 lilo 使用的一些文件。

/dev

dev 是 device(设备)的缩写,这个目录包含了所有 Linux 的外部设备,Linux 对待外部设备同 MS-DOS 使用 .sys 的驱动程序,Windows 95 使用 .vxd,Windows NT 使用 .vdd 不一样,所有外部设备的驱动程序都是由系统提供的,一般用户可以像访问文件一样轻松访问各种各样的外部设备。例如:

```
/dev/cdrom --> 光驱
```

```
/dev/mouse --> 鼠标
```

/dev 下有一个很特殊的文件/dev/null,这个文件就像是宇宙中的黑洞,所有送进去的数据都无影无踪了,它的作用就是让一些输出永不出现,例如

```
tar cvf bak.tar ./bak > /dev/null &
```

就不会产生任何输出。

/cdrom

这个目录是空的,惯例上是让用户将光盘文件系统挂到这个目录下。

/etc

这是 *et cetera* 的缩写,这个目录包含了系统管理所需要的配置文件和子目录,是系统中最重要目录之一,这个目录中下面将要详细介绍。

/sbin

这个目录是用来存放 Super User(系统管理员)的系统管理程序,如 *fsck*、*mount* 等。

/home

这个目录是用来存放用户的主目录的地方,一般说来“/home/用户名”就是该用户的主目录。/home 目录下有一个 *ftp* 目录是给提供 *ftp* 服务的 *wu-ftp* 使用的。

/lib

*lib* 是 *library* 的缩写,这个目录是用来存放系统最基本的动态链接共享库的,如 *ld*、*so*,这个目录如果被删除,系统就肯定不能启动了。几乎所有的应用程序都需要用到这个目录下的共享库。

/lost + found

这个目录下一般是空的,但是当文件系统发生故障,例如系统掉电造成不正常停机,在机器启动的时候,有些文件找不到应该存放的地方,那些无家可归的文件就会放到这个目录下,有些类似于 MS-DOS 修复文件系统时产生的 \* .CHK 文件,这个目录是 *mkfs* 的时候自动产生的。

/mnt

这个目录是空的,系统提供这个目录让用户临时挂载别的文件系统。

/proc

这个目录是 Linux 提供的一个虚拟系统,是由系统在系统启动的时候在内存中产生的,用户可以直接通过访问这些文件来获得系统信息,例如 */proc/kcore* 就是系统运行的时候内存的映像文件。

/root

这个目录是超级用户的主目录。

/shlib

这个目录是为了使用运行 SCO UNIX 程序的 IBCS2 而设的。

/tmp

这是 *temporary* 的缩写,用来存放不同程序执行时所产生的临时文件,类似于 DOS/Windows 95 下所设的 *temp* 变量,这是除 */usr/local* 目录以外一般用户可以使用的一个目录(不幸的是,这个目录也是黑客最常使用的一个目录),这个目录在启动时系统并不自动删除。如果把 */tmp* 目录单独放在一个盘上,注意它不能太小,因为有一些应用程序会产生一些比较大的临时文件,例如 GCC 在编译时有可能要使用十几兆的临时文件,另外如果用户不当关机,重新进入系统时有些应用程序不能正确运行,例如 X 服务器,那么把这个目录下的所有文件都删除也许能解决问题。

/usr

这是 *user* 的缩写,Linux 系统中占地最大的目录,用户的很多应用程序和文件几乎都存放在这个目录下面,比较重要的目录有下面这些目录:

/usr/X11R6

存放 XFree86 的目录,所有同 X Window 相关的程序都存放在这个目录下面。

/usr/bin

用户用到的大多数应用程序都存放在这个目录下面。

/usr/sbin

给超级用户使用的一些比较高级的管理程序和系统守护程序。

/usr/doc

Linux 下的文档的大本营,所有的文档都存放在这个目录下面,是初学者需要好好阅读的地方。

/usr/include

存放了 Linux 下开发或编译应用程序需要使用的头文件,有编程经验的用户应该并不陌生。

/usr/openwin

存放了 SUN 的 OpenWin。

/usr/lib

存放常用的动态链接共享库和静态档案库。一般来说这是在系统安装时候就已经安装好的。

/usr/local

这个目录是用来让普通用户或超级用户安装新软件使用的,它的目录结构同/usr 相似,但允许设成普通用户可写。

/usr/man

这个目录用来存放系统的在线使用手册(online help)。

/usr/src

这个目录是用来存放系统的源代码,缺省时有一个/usr/src/linux 目录,是用来存放系统内核的源代码,用户重新编译内核就是在这个目录下进行的,有些应用程序也会使用这个目录下的头文件,所以用户一般也不要随便删除。

上面简单介绍了 Linux 的目录结构,下面就要向读者介绍如何配置/etc 下的系统设定文件。

/etc 目录

/etc/DIR\_COLORS

用来设定在用 ls 列目录时,各种不同类型文件的颜色,注意不能用管道重定向输出,这样颜色就没有了,缺省时的目录是蓝色,可执行文件是绿色,压缩文件是蓝色等。用户如果要自行设置的话,需要参考该文件中的 ANSI 色彩控制码。

/etc/HOSTNAME

设定用户的节点名。

/etc/NETWORKING

里面只有一行 Yes,是标明网络存在的,没有任何作用。

/etc/host.conf

这个文件说明用户的系统会如何查询节点名,它应该包含至少以下两行:

order hosts,bind

multi on

这两行通知先检查 /etc/hosts 文件, 然后去查名字服务器, 并且可以使用多个名字服务器。

/etc/hosts

设定用户自己的 IP 与名字的对应表。

/etc/hosts.allow

设置允许使用 inetd 的机器, 例如

ALL.202.38.128. 就是 202.38.128. 上的所有机器都可以使用。

/etc/hosts.deny

设置不允许使用 inetd 的机器。

/etc/hosts.equiv

在里面可以设定一些远端机器用 rsh 或 rlogin 登录这台机器而不用密码, 类似于 rhosts 的作用, 不过这东西最好不要乱设, 以可相信的节点为主, 才不会造成安全问题。

/etc/inetd.conf

设定系统的网络守护进程 inetd 的配置, 格式如下:

<service\_name> <sock\_type> <proto> <flags> <user> <server\_path> <args>

服务名称          包类型          协议          参数          用户          服务器路径          参数

例如:

```
shell stream tcp nowait root /usr/sbin/tcpd in.rshd -L
login stream tcp nowait root /usr/sbin/tcpd in.rlogind
# execstream tcp nowait root /usr/sbin/tcpd in.rexecd
# talk dgram udp wait root /usr/sbin/tcpd in.talkd
ntalk dgram udp wait root /usr/sbin/tcpd in.talkd
```

这是安装了 tcp\_wrapper 之后的 inetd.conf。

/etc/inetd.pid

inetd 这个进程的进程 id。

/etc/hosts.lpd

设定远端有哪些节点可以使用本机器的打印机。

/etc/gateways

设定路由器:

<net | host> name1 gateway name2 metric value <passive | active | external>

当 routed 启动时, 将读取 /etc/gateways 这个配置文件。如果一个路由不做路由信息交换就被标成 passive, 反之就是 active, net 或 host 这个关键字指出路由是连接到网络上还是连接到一台特定的机器上。name1 就是目的网络或目的机器的名字。name2 就是路由信息将被送往的路由的名字或 IP 地址。

/etc/protocols

设定系统支持的协议, 用户也可以自行增加, 格式如下:

| 协议名  | 代码 | 别名   | 注解                                          |
|------|----|------|---------------------------------------------|
| ip   | 0  | IP   | # Internet protocol, pseudo protocol number |
| icmp | 1  | ICMP | # Internet control message protocol         |

|      |     |      |                                     |
|------|-----|------|-------------------------------------|
| igmp | 2   | IGMP | # Internet group multicast protocol |
| ggp  | 3   | GCP  | # gateway-gateway protocol          |
| tcp  | 6   | TCP  | # transmission control protocol     |
| pup  | 12  | PUP  | # PARC universal packet protocol    |
| udp  | 17  | UDP  | # user datagram protocol            |
| idp  | 22  | IDP  | # WhatsThis?                        |
| raw  | 255 | RAW  | # RAW IP interface                  |

/etc/named.boot

设定本机为名字服务器的配置文件。

/etc/named.pid

本机上运行的名字服务器的进程 id。

/etc/networks

设定网络的配置文件。可参见前面的章节。

/etc/resolv.conf

这个文件实际上是设定系统的名字服务器。

/etc/services

设定系统的端口、协议类型和提供的服务。

/etc/exports

这个文件是给设定 NFS 文件系统用的,该文件中设定将哪些目录设定为可被挂载的 NFS 文件系统。例如用户要将/home/ftp/pub/linux/slackware 设定给科学院网络上的用户使用,并且是只读的:

```
/home/ftp/pub/linux/slackware *.*.ac.cn(ro)
```

然后查看/etc/rc.d/rc/inet2 文件中是否已经启动了 rpc.nfsd 和 rpc.mountd,这样这个目录就可以在远端被挂载了。

/etc/NNTP\_INEWS\_DOMAIN

设置新闻服务器的配置文件。

/etc/nntpserver

设置用户使用的新闻服务器的地址。

/etc/XF86Config

X Window 的配置文件。

/etc/hostid

系统独有的一个硬件 id,许多商业程序利用它来做 license 的管理,在其他工作站上这个参数是无法更改的,只有 Linux 可以用 hostid 这个参数来更改。

/etc/at.deny

设置哪些用户不能使用 at 这个命令,系统检查的顺序是首先检查 at.deny 是不是空的,如果不空,并且 at.allow 存在的话,就查看 at.allow 中有哪些用户,不在 at.allow 中的用户都不能使用 at 命令。

/etc/bootptab

这是用来设置无盘工作站的远端守护进程/user/net/in.bootpd 的,用户不需要设置。

/etc/devinfo

用来给 MAKEDEV 这个程序设定各种不同设备驱动文件的格式,用户不需要设置。MAKEDEV 是在 `/dev` 目录下创建设备文件的程序。

`/etc/makedev.cfg`

同 DEVINFO 一样是给 MAKEDEV 使用的设置文件,用户不需要设置。

`/etc/diphosts`

用来设置拨号服务器的用户名和密码。

`/etc/slip.hosts`

`/etc/slip.login`

设定 SLIP 的配置文件。

`/etc/fastboot`

这个文件是使用 `shutdown -f` 所产生的, `-f` 就是快速启动,在重新启动之后,系统会去检查这个文件是否存在,以决定是否运行磁盘检查程序 `fsck`。

`/etc/fstab`

记录开机要 mount 的文件系统,在 `/etc/rc.d/rc.s` 中的 `/sbin/mount -avt nonfs` 这一行,就是用来自动 mount `/etc/fstab` 中记录的文件系统,它的格式如下,被忽略的两个选项是 `dump-freq` 一般为 0,另外一个 `pass-number` 指定 `fsck` 顺序的。如

| 文件系统的设备名               | 加载地点                | 文件系统格式               | 读写权限                  |
|------------------------|---------------------|----------------------|-----------------------|
| <code>/dev/hdb2</code> | <code>swap</code>   | <code>swap</code>    | <code>defaults</code> |
| <code>/dev/hdb1</code> | <code>/</code>      | <code>ext2</code>    | <code>defaults</code> |
| <code>/dev/hsc</code>  | <code>/cdrom</code> | <code>iso9660</code> | <code>ro</code>       |
| <code>/dev/hda1</code> | <code>/dos</code>   | <code>msdos</code>   | <code>rw</code>       |
| <code>/dev/hda5</code> | <code>/dosd</code>  | <code>vfat</code>    | <code>rw</code>       |
| <code>none</code>      | <code>/proc</code>  | <code>proc</code>    | <code>defaults</code> |

`/etc/mtab`

系统在启动时创建的信息文件,里面记载了系统已经 mount 的文件系统。这个文件是动态、更新的,可以参见 `/proc/mounts` 这个文件。

`/etc/ftppass`

设定 ftp 服务器的一些配置,包括匿名服务器的设置也是在这个文件中配置。

`/etc/ftpconversions`

设定在 ftp 时使用的过滤器位置。例如“`get 目录名.tar`”就可以将 ftp 服务器上的目录先 tar 再传回来,这个文件就是设定这些打包或压缩程序的目录和参数的。

`/etc/ftpusers`

设定不能使用 ftp 服务的用户,缺省为 `root`、`uucp` 和 `news` 等。这是为了系统安全而设定的。

`/etc/inittab`

设定系统启动时 `init` 进程把系统设置成什么样的 `runlevel`,用户不需要设置。

`/etc/ld.so.cache`

查找系统动态链接共享库的缓存,如果损坏用 `ldconfig` 可以重新产生。

`/etc/ld.so.conf`

系统动态链接共享库的路径,应用程序从这个文件去查找相应的库文件,例如:

```
/usr/local/lib
/usr/X11R6/lib
/usr/i486-linuxaout/lib
/usr/openwin/lib
```

/etc/lilo.conf

配置多重启动程序 lilo 的配置文件,每次更改之后一定要重新运行 lilo 才有效。

/etc/magic

这个文件是用来给 Linux 下的 file 命令使用的, file 命令是用来让用户查看一个文件是什么类型的文件,如是文本文件还是压缩文件,是 DOS 程序还是 Linux 程序,而 file 使用的数据库就是这个 magic 文件。里面记录了各种文件的头信息,用户不需要设置。

/etc/aliases

这个文件是给 sendmail 使用的设置别名的文件,用户不需要设置。

/etc/mail.rc

/etc/mailcap

/etc/sendmail.cf

/etc/sendmail.st

以上的文件都是为了设置 sendmail 的,用户不需要设置。

/etc/issue

这个文件记录用户登录前显示的信息,特别要注意的是,如果用户要加入自己的信息,那么就一定要将 /etc/rc.d/rc.S 这个文件中下面几行标记起来:

```
# echo > /etc/issue
# echo Welcome to Linux /bin/uname -a | /bin/cut -d \ -f3. > > /etc/issue
# echo > > /etc/issue
# echo "/bin/uname -a | /bin/cut -d \ -f3. '." > /etc/motd
```

否则每次开机,系统都会重新创建/etc/issue 和/etc/motd 这两个文件。

/etc/motd

是超级用户用来发布给普通用户的一些通知的地方,即 Message Of the Day 的意思,参见 issue。

/etc/organization

存放用户的名字和组织,没有什么用处,就像 Windows 95 常要用户输入的用户名和工作单位一样。

/etc/group

用来设定用户的组名与相关信息,超级用户可以使用下列命令来设置:

```
groupadd [-g gid [-o]] 组名
groupdel 组名
groupmod [-g gid [-o]] [-n 新组名] 组名
```

/etc/passwd

这是系统最重要的用户密码文件,格式如下:

Login Name:Encrypted Password:UID:GID:GCOS:Home Directory:Login Shell

例如:

```
halt:x:7:0:halt:/sbin:/sbin/halt
operator:x:11:0:operator:/root:/bin/bash
root:x:0:0:root:of all evil,,,:/root:/bin/tcsh
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
sync:x:5:0:sync:/sbin:/bin/sync
bin:x:1:1:bin:/bin:
ftp:x:404:1::/home/ftp:/bin/bash
daemon:x:2:2:daemon:/sbin:
adm:x:3:4:adm:/var/adm:
lp:x:4:7:lp:/var/spool/lpd:
mail:x:8:12:mail:/var/spool/mail:
postmaster:x:14:12:postmaster:/var/spool/mail:/bin/bash
news:x:9:13:news:/usr/lib/news:
uucp:x:10:14:uucp:/var/spool/uucppublic:
man:x:13:15:man:/usr/man:
games:x:12:100:games:/usr/games:
guest:x:405:100:guest:/dev/null:/dev/null
nobody:x:65534:100:nobody:/dev/null:
dfbb:x:0:100:test:/home/dfbb:/bin/tcsh
```

Login Name: 用户名,必须唯一,且小于 8 个字符,可以大小写混合,一般都用小写,上面的 halt,operator,...,nobody 都是系统设定的,并不存在这些用户,如果这个文件里有一些不认识的用户名,就有可能系统被入侵的表现。

Encrypted Password: 用 DES 或 MD5 加过密的密码,如果是 \* 就会无法登录,但是如果安装了 shadow 就会都是 x,如果没有就不要密码。

UID: user id, 用户的 id, 每个人都不一样,100 以前保留为系统使用,root 总是 0。

GID: group id, 用户的组 id。

GCOS: 用户的个人信息,随便填,用“,”作分割符,用户可以用 chfn 来改变。

Home Directory: 用户的主目录。

Login Shell: 用户的缺省 shell, 用户可以用 chsh 来挑选 shells 文件中所列的 shell。

系统管理员同黑客斗争最激烈的恐怕就是这个 passwd 文件了,一般来说,黑客只要能获得 passwd 文件就可以用 crackjack, john 等解密程序加上一个大密码字典扫描出部分用户的密码,从而攻破系统。常见的多是利用 httpd 中 cgi 的 bug, 如 pht 攻击, 或者利用 sendmail 的 bug, 或者是 ftpd 的 bug, 传回这个文件, 要不就利用系统程序的 bug 将 root 的密码栏写为空白, 所以系统管理员最好安装 shadow, shadow 文件对一般用户是不可读的, 加密后的密码就放在这个文件中, 对安全性有很大帮助, Slackware 和 Redhat 都缺省安装了 shadow。

/etc/shadow

见/etc/passwd。

/etc/pnpdevices

列出了支持的 Plug & Play 设备。

/etc/sudoers

如果系统管理员用 `ttysnoops` 替换了 `login` 这个程序,就可以用 `snooptty` 这个程序来监控用户的屏幕,这个文件就是能监听的终端列表。

/etc/sudoers

可以 `sudo` 命令的配置文件,下面是个简单的例子:

```
# Cmnd alias specification
Cmnd_Alias SHUTDOWN = /sbin/halt, /sbin/shutdown
# User specification
root    ALL = ALL
chenxy  ALL = ALL
fangh   ALL = ALL, ! SHUTDOWN
```

在这个系统中, `chenxy` 同 `root` 一样可以用 `sudo` 执行任何命令,而 `fangh` 只能执行除 `SHUTDOWN` 之外的所有命令, `SHUTDOWN` 是在上面定义的命令列表,编辑这个文件有一个专门的编辑器 - `visudo`,所有使用 `sudo` 的记录都在 `/var/adm/sudo.log` 中。

/etc/syslog.conf

系统记录程序 `syslogd` 的配置文件,格式如下:

```
*.info;*.notice /usr/adm/messages
*.debug         /usr/adm/debug
*.warn         /usr/adm/syslog
```

即所有的 `info` 和 `notice` 信息都存在 `/usr/adm/messages` 下, `debug` 信息在 `/usr/adm/debug` 下,警告信息在 `/usr/adm/syslog` 下。

/etc/utmp

这是系统目前正在使用的用户信息,用户退出后系统就会在这个文件中删除相应信息。`utmp` 的结构如下:

```
struct utmp {
    short ut_type;           /* 登录类型 */
    pid_t ut_pid;           /* 进程 id */
    char ut_line[12];        /* 使用终端 */
    char ut_id[2];          /* 初始化 id */
    time_t ut_time;         /* 登录实际 */
    char ut_user[8];         /* 用户名 */
    char ut_host[16];        /* 远程登录机器名 */
    long ut_addr;           /* 远程登录机器 IP 地址 */
};
```

/etc/wtmp

同 `utmp` 差不多,只是它是不断累加的,是所有用户登录的数据库,其中还有系统 `shutdown` 和 `reboot` 的记录,像 `who` 和 `last` 这些程序都会用到这个文件,如果这个文件损坏那么有可能 `who` 和 `last` 会出错,系统管理员可以隔一段时间就删除这个文件,此后再用 `touch` 来产生。

/etc/nologin

系统在要 shutdown 时不希望有用户登录进来,就产生这个文件,里面写着系统正在 shutdown 之类的话,用户这时就没有办法登录,万一系统在这时候丢掉,那下次就谁也登录不进来了,如果系统管理员在维护系统时自己产生这个文件,就可以防止用户登录。

/etc/securetty

设定有哪些终端可以让 root 登录,为了系统安全,可以设定成只有 console 上的用户才能用 root 登录。

/etc/termcap

这可用来设置系统的终端信息,因为各种终端的制造商不同或者规格不同造成的不兼容,而 termcap 就是一个各种终端的大数据库,对于终端上的某项功能定义一个实际上对于终端操作的控制码或者是属性设定,这样就可以让不同的终端都可经过 termcap 的翻译使用相同的输入输出控制,下面是一个 termcap 的例子:

```
lx|linux|console|con80x25|Linux System Console: \
:do=^J;co#80;li#25:cl=^E[H^E[J;sf=^E[D;sb=^E[M; \
:le=^H;bs:am;cm=^E[%i%d;%dH;nd=^E[C;up=^E[A; \
:ce=^E[K;ed=^E[J;so=^E[7m;se=^E[27m;us=^E[36m;ue=^E[m; \
:md=^E[1m;mr=^E[7m;mb=^E[5m;me=^E[m;is=^E[1;25r^E[25;1H; \
:ll=^E[1;25r^E[25;1H;al=^E[L;dc=^E[P;dl=^E[M; \
:it#8;ku=^E[A;kd=^E[B;kr=^E[C;kl=^E[D;kh=^H;ti=^E[r^E[H; \
:ho=^E[H;kP=^E[5^;kN=^E[6^;kH=^E[4^;kh=^E[1^;kD=^E[3^;kI \
=^E[2^; \
:kl=^E[[A;k2=^E[[B;k3=^E[[C;k4=^E[[D;k5=^E[[E;k6=^E[17^; \
:k7=^E[18^;k8=^E[19^;k9=^E[20^;k0=^E[21^;K1=^E[1^;K2=^E \
[5^; \
:K4=^E[4^;K5=^E[6^; \
:pt:sr=^E[M;vt#3:xn;km;bl=^G;vi=^E[?25l;ve=^E[?25h;vs=^E[?25h; \
:sc=^E7;rc=^E8;cs=^E[%i%d;%dR; \
:rl=^Ee;r2=^Ee;r3=^Ee;
```

它包含三种定义方式:

- (1)直接写上功能名称如 xon 就代表使用 xon/xoff 传输协议。
- (2)功能名称后加上 '#' 号再加一个数值;如 col # 80 代表 80 列。
- (3)功能名称后加 '-' 号再加一描述,如 cub1 = ^H 就是 Ctrl + H 是退一格。

/etc/tty

设定系统的终端类型。如:

```
console tty1
console tty2
console tty3
console tty4
console tty5
console tty6
vt100 tty0
vt100 tty1
```

```
vt100 tty2
```

```
vt100 tty3
```

/etc/gettydefs

getty\_ps 的定义文件。

/etc/yp.conf

NIS 的配置文件。

/etc/mtools.conf

设定 mtools 程序的参数, mtools 是在 Linux 下的 dos 命令集合。

/etc/fdprm

设定格式化软盘的参数, 用户不需要设置, 除非用户要使用像 HD-COPY 之类的软件格式化出来的大容量软盘。

/etc/shells

记录了用户可以使用的 shell 列表, 实际上如果强行直接修改 passwd 文件, 也可以使用不在列表中的 shell 程序, 可是在 ftp 时, 就无法登录。

/etc/login.access

控制用户登录权限的设定文件, 格式如下

```
permission : users : origins
```

其中 permission 是 '-' 或 '+', 即否定或允许。第二栏是用户名, 可以用空格隔开的用户名, 也可以是组名, 如果是 ALL 就指所有用户, 第三栏是具体限制登录地点, 首先如 LOCAL 代表本地, console 是控制台, .ihcp.ac.cn 代表所有 .ihcp.ac.cn 上的机器, 加上 EXCEPT 就是除了的意思。例如:

```
--:wheel:ALL EXCEPT LOCAL .ihcp.ac.cn
```

```
--:wabsarco wbssecr wbspac wbsym wscosor wstaiwdc:ALL
```

/etc/login.defs

这是所有用户登录时的缺省配置文件, 这个文件中有大量的定义, 许多原先在 profile 和 login 文件中的定义都挪到这里。如用户设置缺省路径、登录时间限制和最多错误登录次数等。这是系统管理员应该常常修改的地方。

```
/etc/csh.cshrc
```

```
/etc/csh.login
```

```
/etc/profile
```

```
/etc/zprofile
```

不管用户的 login shell 是那种 shell, 每一个 shell 都有它自己的设置文件, 我们设定理想环境时所需使用的指令与设定的变量, 都可以写在这些设置中, 在登录时执行, 这些设置文件的名称分别是:

csh

启动(依照所列顺序):

.cshrc——每次都会读取。

.login——是登录 shell 的话才会读取。

退出前执行的命令:

.logout——是登录 shell 的话才执行。

其他:

.history——储存命令的历史记录(根据 \$ savehist 的值决定大小)。

tcsh

启动(依照所列顺序):

/etc/csh.cshrc——每次都会读取。

/etc/csh.login——是登录 shell 的话才会读取。

.tcshrc——每次都会读取。

.cshrc——假如没有 .tcshrc 存在的话才读取。

.login——是登录 shell 的话才会读取。

退出前执行的命令:

.logout——是登录 shell 的话才会读取。

其他:

.history——保存命令的历史记录(根据 \$ savehist 的值决定大小)。

.cshdirs——保存目录栈。

sh

启动(依照所列顺序):

/etc/profile——是登录 shell 的话才会读取。

.profile——是登录 shell 的话才会读取。

退出前执行的命令:

用“trap 命令 0”指定的命令。

ksh

启动(依照所列顺序):

/etc/profile——是登录 shell 的话才会读取。

.profile——是登录 shell 的话才会读取。

\$ ENV——设定后就读取。

bash

启动(依照所列顺序):

/etc/profile——是登录 shell 的话才会读取。

.bash\_profile——是登录 shell 的话才会读取。

.profile——如果没有 .bash\_profile,则会在 login 时读取。

.bashrc——交互式非登录 shell 的话才会读取。

\$ ENV——如果设定了则读取。

退出前执行的命令:

.bash\_logout——是登录 shell 的话才读取。

其他:

.inputrc——Readline 初始化之时读取。

zsh

启动(依照所列顺序):

.zshenv——如果无 -f 就会读取。

- .zprofile ——是登录 shell 的话才会读取。
- .zshrc ——交互 shell 如果无 -f 就会读取。
- .zlogin ——是登录 shell 的话才读取。

退出前执行的命令：

- .zlogout ——是登录 shell 的话退出前执行的命令。

如果这些文件都存在,它们的执行顺序将是:

(1)sh 系列: /etc/profile --> \$HOME/.profile

(2) csh 系列: /etc/csh.cshrc->/etc/csh.login-> \$HOME/.cshrc --> \$HOME/.login

在这些 start files 中, /etc 下的 profile 是系统管理者为大家设定的, 一般用户无法做任何改动, 如果对系统管理者的设定觉得不够或不符合需要, 则可在自己的帐号下的 .profile, .cshrc 和 .login 进行增加或修正, 下面是常见的设置:

让 Backspace 键能有作用

在 \$HOME/.profile 或 \$HOME/.cshrc 中加上  
stty erase ^H

可以输入汉字

在 \$HOME/.profile(sh 系列)中加上  
stty -istrip cs8  
LC\_CTYPE = ISO\_8859\_1  
export LC\_CTYPE

在 \$HOME/.cshrc(csh 系列)中加上  
stty -istrip cs8  
setenv LC\_CTYPE iso\_8859\_1

让提示符能随着路径的变动而改变

在 sh 及 csh 中并无法做到这个功能, 而在 bash 及 tcsh 中的设定则分别是:

bash: 在 \$HOME/.profile 中加上

PS1 = "PWD" > '

或 PS1 = \w >

tcsh: 在 \$HOME/.cshrc 中加上

set prompt = '% ' >

mesg y (mesg n)

如果在工作时不希望被 talk 等信息所打扰, 可用 'mesg n' 指令, 以避免外来信息中断正要进行的工作('mesg y'则是恢复与外界的通讯)。

umask [nnn]

设定新增文件或目录的保护模式(mode), 新文件或新目录的保护模式为系统管理员用 create 所设定的属性减去 nnn, 如系统管理员所设定的文件缺省读写权限是 777, 而使用者又自行设定了 umask 022, 则这个用户在他个人帐号下的每一个新产生的档案的读写权限将会是 755(777-022)。

Function 与 Alias

对于一连串或某一个指令, 我们可以用函数(function)的方式(在 sh、ksh、bash 中)或

alias 指令(在 csh,tcsh,ksh,bash 中),用--简短字符串来代替一个或一连串的指令,下面是函数的用法:

```
vi() {
    msg n
    /usr/ucb/vi $ *
    msg y
}
```

**注意:**函数名不能与 shell 的内建命令重复,否则就会失败。

同样的功能,alias 的用法如下:

```
alias vi 'msg n; /usr/ucb/vi \! *; msg y'
```

用“'”将命令序列包起来,用“;”来分割。

如果同时使用 alias,和 function 定义一个别名,则先执行 alias 定义的别名。

另外建议大家定义 alias rm 'rm -i',防止误操作删除文件。

常用变量设定

在 UNIX 环境中的变量可分为两类,一是环境变量(Enviroment Variable),一是 shell 变量(Shell Variable),二者之间的不同之处在于:环境变量不会因为改变 shell 而失去这个变量的作用,如 PATH 是一个环境变量,因此,即使用户在设定好 PATH 之后再执行另一个 shell(如原来是用 sh,而后在 sh 环境中再执行 csh),PATH 这个环境变量仍能继续作用,而 shell 变量(如 csh 的 cwd)则只有在特定的 shell 下才有作用,一执行其他 shell 就失去作用。

在 sh 及 csh 二系列各有其不同的设定变量的方式,命令如下:

(1) Bourne Shell 系列

```
[VAR] = [STRING or NUMBER]
export [VAR]
```

如果当我们离开 shell,进入另一个程序(如 tin)之后,需要使用这个变量,则除了设定这个变量的值外,还需要以 export 指令加以定义,如设定屏幕的变量 TERM 就需以 export 来定义:

```
TERM = vt100
export TERM
```

(2) C Shell 系列

```
set [var] = [STRING or NUMBER]
setenv [var] [STRING or NUMBER]
```

其中 set 是单纯设定变量值,setenv 则除了设定变量值外,还包含了类似 sh 中 export 的功能。

而无论是 Bourne Shell 或 C Shell,它们都有一些基本的、共同的常用变量,如:

(1) PATH

设定可执行文件的查找路径,若当前目录也要包含在寻找的路径中,则需在设定路径时给予“.”,如:

```
PATH = .:/bin:/usr/local/bin:/usr/bin
```

或

```
set path = ( . /bin /usr/local/bin /usr/bin)
```

不过不建议 root 增加这一条,会对安全有影响。

#### (2)HOME

标明用户主目录的所在位置,通常在登录时系统自动设定。

#### (3)SHELL

标明用户使用的 shell 名,通常在登录时系统自动设定。

上面是一些常用变量,如果用户不清楚,直接运行 `set` 或 `setenv` 就可以观察目前所有变量的内容,如果不清楚某一变量的设定内容,可用“`echo $[VAR]`”来观察变量的内容,如:

```
echo $PATH
```

另外用户编辑完自己的配置文件并不会立即起作用,除非用

```
source 配置文件名
```

才可以让配置文件立刻起作用。

到此为止,/etc 下的文件配置就告一段落了,实际上只是一个非常简单的介绍,更多的还是要用户在实践中去摸索。

### 3.4 其他设备的配置(光驱、声卡、打印机)

#### 1. 如何在 Linux 下配置光驱

在 Linux 下配置光驱实际上是一件很轻松的工作,因为绝大多数用户的光驱是 ATAPI 接口的,而 Linux 系统核心内置的 ATAPI 兼容的驱动程序,使得无需任何多余的手续就可以支持所有的 ATAPI 接口的光驱。只要用户的光驱是最近购买的、四倍速以上的光驱就都是 ATAPI 接口的光驱。另外用户可能会购买 SCSI 接口的光驱,这种光驱只要用户使用的 SCSI 卡能够在 Linux 下正常工作,并且块(block)大小是 512 比特的光驱就都可以在 Linux 下使用。下面就简单介绍一下在 Linux 下安装光驱的步骤。主要以 ATAPI 接口的光驱为例子向读者介绍。

(1)首先,用户在启动 Linux 机器的时候查看一下启动信息,如果没看清的话就用 `dmesg` 命令来查看,如果发现在 Linux 辨认出硬盘之后有这样一句类似的信息:

```
hdc: AZTECH CD-ROM, ATAPI CDROM drive
```

其中 `hdc` 就是用户的 CDROM 所在的 IDE 口,那么用户的光驱就已经可以使用了,请用用户以 root 身份登录,使用下面的命令加载光驱(确定已经将数据光盘插入光驱):

```
# mount /dev/hdc /cdrom
```

这时候,在 `/cdrom` 目录下应该就是光驱上的文件系统了,用户还应该作一下符号链接:

```
# ln -s /dev/hdc /dev/cdrom
```

在光驱加载后,如果要取出光盘,就不能像在 DOS/Windows 下一样直接去按光驱弹出键了,按了之后一般是没有反应的,因为 Linux 把光驱给锁住了,必须先离开 `/cdrom` 目录,下这样的命令:

```
# umount /cdrom
```

这时候才能取出光盘。用户可以使用一个更好的管理软件 Supermount:

```
ftp://sunsite.unc.edu/pub/linux/patches/diskdrives/)
```

用户可以在 `/etc/fstab` 文件中加上这么一行:

```
/dev/hdc      /cdrom  iso9660  ro,user,exec,noauto,nohide
```

这样用户以后就可以简单地用下列命令挂载 CD-ROM 了:

```
# mount /cdrom
```

(2) 如果用户发现 Linux 不支持自己的光驱,那么就可以尝试下面的步骤:

首先如果 ATAPI 光驱是插在一个单独的 IDE 口上,将跳线改为 MASTER,不要设置成 SLAVE。

另一种方法就是重新编译内核:

ATAPI 的光驱应该选择下面的选项:

```
Enhanced IDE/MFM/RLL disk/cdrom/tape support (CONFIG_BLK_DEV_IDE) [Y/n/?]
```

```
Include IDE/ATAPI CDROM support (CONFIG_BLK_DEV_IDECD) [Y/n/?]
```

如果是特定接口的光驱,如插在声卡上的光驱等就要选择相应的选项,如果不确定,干脆全部选中。

SCSI 的光驱应该选择下面的选项:

```
SCSI support (CONFIG_SCSI) [Y/n/m/?]
```

```
SCSI CD-ROM support (CONFIG_BLK_DEV_SR) [Y/n/m/?]
```

还要加入对 SCSI 卡的支持,如这行提示:

```
Add ptec AHA152X support (CONFIG_SCSI_AHA152X) [Y/n/m/?]
```

常见的光盘都使用 ISO-9660 文件系统,所以也要选择下面这一行:

```
ISO9660 cdrom filesystem support (CONFIG_ISO9660_FS) [Y/n/m/?]
```

重新编译内核,再用 lilo 安装内核,用新的内核启动,查看是否已经辨认出光驱,如果还是不行,那么就要在系统启动时利用命令行参数来进行配置:

Sbpcd 光驱

LILO 后面的参数:

```
sbpcd = < io-address > , < interface-type >
```

第一个参数是 I/O 地址(如 0x230)。interface-type 可以是“SoundBlaster”、“LaserMate”或“SPEA”。手工建立设备文件:

```
# mknod /dev/sbpcd b 25 0
```

Sonycd535 光驱

LILO 后面的参数:

```
sonycd535 = < io-address >
```

io-address 是 I/O 地址(如 0x320)。不然用户也可以写在 `sonycd535.h` 里编译。手工建立设备文件:

```
# mknod /dev/sonycd535 b 24 0
```

Cdu31a 光驱

LILO 后面的参数:

```
cdu31a = < io-address > , < interrupt > , PAS
```

第一个参数是 I/O 地址(如 0x340),第二个是中断号(0 代表不用中断,用于查询)。

第三个参数只有在连接 Pro Audio Spectrum 16 声卡时要加, 固定为 PAS。io-address 是必要的, 而 interrupt 则可有可无。可手工建立设备文件:

```
# mknod /dev/cdu31a b 15 0
```

Aztc 光驱

LILO 后面的参数:

```
aztc = < io-address >
```

唯一的参数是 I/O 地址(如 0x340)。手工建立设备文件:

```
# mknod /dev/aztc0 b 29 0
```

Gscd 光驱

LILO 后面的参数:

```
gscd = < io-address >
```

它用 I/O 地址作为其参数(如 0x340)。手工建立设备文件:

```
# mknod /dev/gscd0 b 16 0
```

Mcd 光驱

LILO 后面的参数:

```
mcd = < io-address > , < irq >
```

参数分别是接口卡的 I/O 地址(如 0x340)及中断号。手工建立设备文件:

```
# mknod /dev/mcd b 23 0
```

Mcdx 光驱

LILO 后面的参数:

```
mcdx = < io-address > , < irq >
```

参数分别是接口卡 I/O 地址(如 0x340)及中断号。手工建立设备文件:

```
# mknod /dev/mcdx0 b 20 0
```

Cm206 光驱

LILO 后面的参数:

```
cm206 = < io-address > , < interrupt >
```

第一个是 I/O 地址(如 0x340), 第二个是中断号。手工建立设备文件:

```
# mknod /dev/cm206cd b 32 0
```

Optcd 光驱

LILO 后面跟的参数:

```
optcd = < io-address >
```

参数是接口卡 I/O 地址。手工建立设备文件:

```
# mknod /dev/optcd0 b 17 0
```

Sjcd 光驱

LILO 后面的参数:

```
sjcd = < io-address > , < interrupt > , < dma >
```

指出 I/O 地址、中断及 DMA 号码(如 sjcd = 0x340, 10, 5)。手工建立设备文件:

```
# mknod /dev/sjcd b 18 0
```

Bpcd 光驱

LILO 后面的参数:

```
bpcd = < io-address >
```

参数是使用的 I/O 地址 (如 `bpcd = 0x3bc`)。手工建立设备文件:

```
# mknod /dev/bpcd b 41 0
```

### SCSI CD-ROM 光驱

查看 SCSI-HOWTO 检查是否能使用 SCSI 卡。手工建立设备文件:

```
# mknod /dev/scd0 b 11 0
```

```
# mknod /dev/scd1 b 11 1
```

### IDECD 光驱 (ATAPI 光驱)

LILO 后面跟的参数:

```
hdx = cyls, heads, sects, wpcorn, irq 或 hdx = cdrom
```

这里的 `hdx` 可以是 `{hda, hdb, hdc, hdd}`, 或只有 `hd`, 表示下一台机器。只有前三个参数是必要的 (`cyls, heads, sects`), 例如:

```
hdc = 1050, 32, 64 hdd = cdrom
```

`/usr/src/linux/Documentation/ide-cd` 有详细介绍, 请仔细阅读。

然后再看看能否让 Linux 找到光驱, 如果还是不行, 查看一下 `/proc/devices` 核心编译了哪些驱动程序, 查看一下 `/proc/filesystems` 有没有 `iso9660` 文件系统, 看看 `/proc/ioports` 有没有给光驱分配 io 端口。再重复上面的步骤或者询问别的有经验的用户。如果有些光驱需要使用厂家提供的特定驱动程序才能工作, 那么试一下下面的方法, 在 DOS 下启动, 加载厂家的驱动程序, 然后热启动机器 (或者用 `loadlin` 启动 Linux) 进入 Linux。看看光驱能否工作。

## 2. 在 Linux 下配置声卡

首先对于没有声卡又想用金山影霸一样使用 PC 喇叭模拟声卡的用户, 请看下面的节点: <ftp://ftp.informatik.hu-berlin.de/pub/os/linux/hu-sound/>。

在 Linux 下配置声卡之前首先确定用户的声卡在 DOS/Windows 下可以使用, 最好是与 Sound Blaster 或 Sound Blaster Pro 兼容的, 如果是 ALS007 之类用软件来模拟 SB 的声卡就先在 DOS 下运行他们的 `init` 程序, 然后热启动或用 `loadlin` 启动 Linux, 就可以像配置 Sound Blaster 一样配置这些声卡了。另外最好将声卡设置成最通用的设置: IO 地址为 `0x220`, 中断为 `5`, DMA 为 `1`, 然后启动 Linux, 查看启动信息 (或者用 `dmesg`), 看看有没有类似下面的信息, 如果有的话, 用户的声卡就已经被 Linux 支持了:

```
Sound initialization started
< Sound Blaster 16 (4.13) > at 0x220 irq 5 dma 1,5
< Sound Blaster 16 > at 0x330 irq 5 dma 0
< Yamaha OPL3 FM > at 0x388
Sound initialization complete
```

查看一下 `/etc/sndstat` 的信息, 如果文件存在并且有类似下面的信息:

```
% cat /dev/sndstat

Sound Driver: 3.5.4-960630 (Sat Jan 4 23:56:57 EST 1997 root,
Linux fizzbin 2.0.27 #48 Thu Dec 5 18:24:45 EST 1996 i586)
Kernel: Linux fizzbin 2.0.27 #48 Thu Dec 5 18:24:45 EST 1996 i586
```

Config options: 0

Installed drivers:

Type 1: OPL-2/OPL-3 FM

Type 2: Sound Blaster

Type 7: SB MPU-401

Card config:

Sound Blaster at 0x220 irq 5 drq 1,5

SB MPU-401 at 0x330 irq 5 drq 0

OPL-2/OPL-3 FM at 0x388 drq 0

Audio devices:

0: Sound Blaster 16 (4.13)

Synth devices:

0: Yamaha OPL-3

Midi devices:

0: Sound Blaster 16

Timers:

0: System clock

Mixers:

0: Sound Blaster

那么用户的声卡就可以正常工作了。另外查看一下有没有 `/etc/audio /etc/dsp /etc/mixer` 这些文件,如果有,立刻就可以用下面的命令测试一下用户的声卡:

```
cat /vmlinuz > /dev/audio
```

如果用户听到用户的声卡怪声四起,就用 `Ctrl + C` 来终止,用户下面的工作就是去下载 `l3dec`, `mpeg123`, `midplay`, `playmod`, `realplayer` 之类的放音程序或者给 `fvwm95` 加上音效...

如果不行,那么用户就只有去重新编译内核或者可加载模块了,在 `make config` 时,对下面的选项:

Sound card support (CONFIG\_SOUND) [M/n/y/?]

回答 Yes, 开始设置声卡。

Old configuration exists in /etc/soundconf. Use it Y/n/?

是否使用以前的设置,一般回答 N 继续下面的设置。

ProAudioSpectrum 16support Y/n/?

用户声卡是 PAS16 的卡就选 Yes。

SoundBlaster support Y/n/?

用户声卡是 Sound Blaster 或兼容卡就回答 Yes。

Gravis Ultrasound support Y/n/?

一般选 No。

MPU-401 support (NoT for SB16) Y/n/?

如果是声霸卡就选 N。

6850 UART Midi support Y/n/?

最好回答'N'。6850 UART 现在已经没有多少人用了。

PSS (ECHO-AD12111) support Y/n/?

16 bit sampling option of GUS (not GUS MAX) Y/n/?

GUS MAX support Y/n/?

Microsoft Sound System support Y/n/?

Ensoniq Soundscape support Y/n/?

MediaTriX AudioTriX Pro support Y/n/?

Support for MAD16 and/or Mozart based cards?

Support for Crystal CS4232 based (PnP) cards Y/n/?

Support for Turtle Beach Wave Front (Maui, Tropez) synthesizersY/n/?

如果用户有上面列出的声卡就选 Yes,否则就选 No。

SoundBlaster Pro support Y/n/?

选择 Yes。

SoundBlaster 16 support Y/n/?

如果用户声卡是 Sound Blasrer 16 或 AWE32 就选 Yes。

Audio Excel DSP 16 initialization support Y/n/?

如果有这种卡就选 Yes。

/dev/dsp and /dev/audio support (usually required) Y/n/?

MIDI interface support Y/n/?

FM synthesizer (YM3812/OPL-3) support Y/n/?

/dev/sequencer support Y/n/?

上面是用来产生/dev 下面的设备的,统统选 Yes。

Do you want support for the mixer of SG NX Pro ?

Do you want support for the MV Jazz16 (ProSonic etc.) ?

Do you have a Logitech SoundMan Games Y/n/?

上面请用户根据需要选择。

下面会询问用户声卡的 I/O 地址、中断及 DMA 号。用户如果不知道,回到 DOS 下用声卡的设置程序查看,或者到 Windows 95 的控制台/系统/声卡中去查看。

The sound driver is now configured.

Save copy of this configuration to /etc/soundconf [Y/n/?]

最后回答 Yes 就可以存到/etc/soundconf 中,下面可编译并安装内核,如果用户想用可加载模块就到/usr/src/linux/drivers/sound 下执行:

```
make module
```

此时可以编译可加载模块,最后就是用 insmod 来加载声卡模块。如果用户还是没有能找到声卡,那么去阅读一下 SOUND HOWTO 也许会有帮助,或者自己去阅读/usr/src/linux/drivers/sound 下面的文件。

### 3. 在 Linux 下配置打印机

如果系统内核在编译的时候已经加入了对 lp 设备的支持,用户可以查看 `/proc/device` 目录下有没有 lp 设备,在 `/dev` 下应该有 `lp0`、`lp1`、`lp2` 等设备,如果用户发现不能使用打印机,可以在 `lilo` 命令行加上下面的参数:

格式: `lp=port0[,irq0[,port1[,irq1[,port2[,irq2]]]]]`

例如: `lp=0x378,0` 或 `lp=0x278,5,0x378,7`

使用可加载模块的就用 `io=x,y,z` 和 `irq=x,y,z`。

如果是串口打印机,就需要用 `atty` 来设置相应的串口,这需要查阅相应的打印机手册,获得相关的控制流程自行设置。

用户可以用 `cat /etc/lilo.conf >/dev/lp0` 的方法来测试打印机是否已经被加载。

下面介绍一下 Linux 下的打印服务器 `lpd` 的运行方式:

`lpd` 控制打印机的守护进程,每台打印机打印时也有一个 `lpd`。

`lpr` 同 `lpd` 联系在打印队列中加入一个打印任务的命令

`lpq` 列出所有在打印队列中的任务

`lpc` 控制打印队列的命令,可以开始,停止,记录打印任务

`lprm` 从打印队列中删除一个打印任务

用户每用 `lpr` 提交一个打印任务,`lpr` 就通过 `/dev/printer` 与 `lpd` 通讯,`lpd` 就扫描 `/etc/printcap` 文件查询打印机对应的缓存目录,然后由 `lpd` 来控制打印。

设置打印机的第一步就是在 `/etc/printcap` 中设置打印机,如下面的例子:

```
# LOCAL djet500
lp|dj|deskjet: \
:sd=/var/spool/lpd/dj: \
:mx#0: \
:lp=/dev/lp0: \
:sh:
```

此时设置了一台 `deskjet` 打印机,再在 `/var/spool/lpd` 中建立相应的缓存目录。另外如果要正确打印文档,还需要在 `printcap` 中加入相应的过滤器,所有的打印任务都是通过这个过滤器来打印的,如下所示:

```
lp|dj|deskjet: \
:sd=/var/spool/lpd/dj: \
:mx#0: \
:lp=/dev/lp0: \
:if=/var/spool/lpd/dj/filter: \
:sh:
```

用户可以安装 `APS-filter` 就能直接打印 `text`、`PostScript`、`dvi`、`gif` 等格式的文件,网络节点为

`ftp://sunsite.unc.edu/pub/Linux/system/Printing/aps-491.tgz`

其他过滤器有:

`ftp://tsx-11.mit.edu/pub/linux/sources/usr.bin/magic-filter-0.4.tar.gz`

`ftp://sunsite.unc.edu/pub/Linux/system/Printing/magicfilter-1.1b.tar.gz`

如果用户要安装网络上的 UNIX 打印机。应在/etc/printcap 中如下设置：

```
# REMOTE djet500
lp|djl|deskjet: \
    :sd=/var/spool/lpd/dj: \
    :rm=远端机器节点名: \
    :rp=打印机名: \
    :lp=/dev/null: \
    :sh:
```

使用 Windows 95/NT、Lan Manager 或 Samba 共享的打印机：先安装 smbprint，再设置/etc/printcap：

```
lp|远端打印机的全路径: \
    :lp=/dev/null:sh: \
    :sd=/var/spool/lpd/lp: \
    :if=/usr/local/sbin/smbprint:
```

使用 NETWARE 网络打印机：先安装 ncpfs(ftp://linux01.gwdg.de/pub/ncpfs/)，再设置/etc/printcap：

```
sub2|远端 NETWARE 打印机: \
    :lp=/dev/null:sh: \
    :sd=/var/spool/lpd/sub2: \
    :if=/var/spool/lpd/nprint-script:
```

具体打印可以使用下面的脚本文件 nprint-script：

```
#!/bin/sh
/usr/local/bin/nprint -S net -U name -P passwd -q printq-q -
```

使用 HP 或其他网络打印服务器：

```
lj-5|远端打印机名: \
    :lp=/dev/null:sh: \
    :sd=/var/spool/lpd/lj-5: \
    :rm=打印机地址:rp=raw:
```

一般说来，Linux 下许多应用程序自己就提供了一些过滤器，如 Ghostscript 让用户可以通过 Ghostview 观看、打印 PS 文件和 PDF 文件，并且现在已经为 Ghostscript 开发了中文(GB)字库，已经可以直接阅读打印中文 PS 文件。

上面就是 Linux 下打印机的设置过程，更详细的请查阅本书的系统管理部分。

## 3.5 常见问题解答

### 1. 如何设置屏幕保护时间？

在控制台(Console)下：

修改/etc/rc.d/rc.M 的内容

```
# Tell the viewers what's going to happen...
echo Going multiuser...
```

```
# Screen blanks after 15 minutes idle time.
```

```
/bin/setterm -blank 15
```

在 X Window 下:

S3 的 X 服务器支持这个功能,在 XF86Config 文件中修改:

```
Section Device
Identifier      s3-Virge
VendorName      Winfast
BoardName       s3-Virge
VideoRam        2048
Option          power_saver
# # # ~~~~~
# # # 加这一行
EndSection
```

```
Section Screen
Driver          accel
Device          s3-Virge
Monitor         mag17
```

```
BlankTime 5
SuspendTime 15
OffTime 30
# # # ~~~~~
# # # 加这三行
```

另外用 xset 也可以设置这三个参数,例如

```
xset s 300 5 s blank s power 900 1800
```

## 2. 在 X Window 下如何使用 xconsole?

修改/etc/syslog.conf,在

```
*.=info;*.=notice           /usr/adm/messages
*.=debug                     /usr/adm/debug
*.warn                       /usr/adm/syslog
```

后加一行

```
*.*                          /dev/console
```

将 /usr/X11R6/bin/startx 这个文件的最后一行

```
xinit $ ..... $ .....
```

改成

```
xinit $ ..... $ ..... > /dev/console
```

如此可以把信息显示在 xconsole 中。

## 3. 在 Linux 下如何录音?

如果用户的 Linux 系统支持声卡并且有/dev/audio 这个设备文件,用下面的方法:

```
% dd bs=8k count=4 </dev/audio > sample.au # 录四秒种的声音
4+0 records in
4+0 records out
% cat sample.au >/dev/audio # 放音

测试,或者:
cat /dev/audio > file.au

录完按 Ctrl+C。
```

#### 4. 在 Linux 下如何将音乐 CD 转成 WAV 文件再压缩成 mp3 文件?

站点:

<http://www.tardis.ed.ac.uk/~psyche/pc/cdrom/CDDA.html>

使用 `cdda2wav.tar.gz` 这个程序包来抓音轨,用 `l3enc` 这个软件来压成 mp3,用 `l3dec/mpeg123/x11amp` 来播放 mp3。

#### 5. 如何用 EPSON 系列彩色打印机打印彩色文件?

使用 Ghostscript 5.01 其中有一个叫 `stcolor` 的驱动程序,如果是 Stylus COLOR 或 Stylus COLOR II 等,试试在 `apsfilter` 的设定文件中使用以下参数:

```
/usr/bin/gs -q -dSAFER -sDEVICE=stcolor \
-r360x360 -sDithering=fscmyk -sOutputFile=- stcolor.ps -
```

相关网址:

Michael Holve - The Epson Color Stylus and Linux

<http://www.fammed.sunysb.edu/holve/epson.html>

STCOLOR FAQ

<http://www.isisnet.com/bdavidson/gs.stc.FAQ.html>

如果用户使用的是 Stylus COLOR 500 或 600/800 彩色打印机,请参阅下列网址:

Epson Color Stylus 500, Ghostscript, and Linux

<http://www.pe.net/~williams/Stylus/Stylus.html>

#### 6. 如何加速 X Window 的运行?

(1)使用 `rxvt` 来取代 `xterm`, `rxvt` 占用的内存较少。

(2)尽量使用同一种字体(例如全用 `8x16`)。

(3)使用 `xf86` 来作字体缩放的工作:在 X Window 中,如果并应用程序要求一个不存在的字体, X server 会从硬盘加载一个同类但不同大小的字体,并再做缩放的工作,这时候 X Window 会特别慢,启动 `xf86` 后 X Window 的速度会大大提高, `xf86` 的设置见后面的章节。

#### 7. 如何在 Linux 下看 Windows 95 的长文件名?

确定核心支持 `vfat` 文件系统,然后改 `/etc/fstab` 中的选项,例如:

```
mount -t vfat /dev/hdal /dos
```

这样就可以看 win95 的长文件名了。

## 8. 如何在 Linux 下观看 VCD?

使用 2.1.xx 的系统核心, 下载 xreadvd 和 mtv 这两个程序(光盘中有附)。然后使用

```
% xreadvd | mtv
```

播放就可以了。

## 9. ET6000 使用 16 比特真彩的方法?

(1) 确定为 XFree86 3.2 以上版本。

(2) 用 X32\_SVGA 做 X server。

(3) 显示卡选 140 (et6000 generic)。

(4) ramdac 选 9。

(5) clock 选 8。

(6) 编辑用户的 /etc/XF86Config 中 ET6000 Chip 设定的 section:

加入 Option linear。

(7) 输入 startx——bpp16 就可以测试了。

## 10. 如何 Netscape 中听音乐?

确定已经安装声卡, 再安装 xanim 和 playmidi 这两个程序。配置 netscape, 以 netscape 3.01 Gold 为例, 选择 Option / Gernal / helper:

|             |             |
|-------------|-------------|
| audio/basic | xanim %s    |
| audio/x-wav | xanim %s    |
| audio/x-mid | playmidi %s |

# 4 Linux 使用入门

## 4.1 Linux 基本命令的使用

本节主要讲解 Linux 下基本命令的使用,由于篇幅的关系,无法列出所有 Linux 命令的使用说明。读者在使用没有列出的命令时,可以使用“--help”参数查询该命令所附的在线帮助,或者利用 man 来查询更详细的使用说明。

### 4.1.1 ash

#### 1. 简介

ash——a shell

这是由 Kenneth Almquist 在 1989 年编写的,ash 是 Linux 下许多命令解释器中的一个它的许多特性接近于 SYSTEM V 的 shell。

#### 2. 部分参数说明

ash [ -efijnosz ] [ +efijnosz ] [ -c 命令 ] [ 参数 ]

-c : 若用 -c 参数,则 ash 执行完这条命令后退出。

-s : 若用 -s 参数,则 ash 从标准输入中读入命令(在执行完 -c 带的命令之后。),如果不跟 -c -s 参数则 ash 以所跟的第一个参数为文件名,从此文件中读入命令。如无参数则 ash 缺省设定 -s 参数,从标准输入中读入命令,直到输入 exit。

如果参数 0 的第一个字母是 '-' 则 ash 确定为 login shell,ash 将从 /etc/profile 或用户的根目录中的 .profile 读入相应的设置和环境变量。

-e : 若用 -e 参数,则 ash 执行命令后返回值为非零值时则退出 ash。

-f : 若用 -f 参数,则 ash 关闭自动产生文件名功能。

-j : 打开伯克利 UNIX 风格的工作控制。

-n : 读入命令但是并不执行。

#### 3. 范例

ash -c ls

ash 执行 ls 这个命令后退回原先的目录和 shell。

ash -s

ash 执行一个新的 shell,现在可以在这个 shell 中工作,按 (Ctrl-D) 或输入 exit 后,则退回原先的目录和 shell,所设定的环境变量返回原先的值。

## 4.1.2 at

### 1. 简介

at, batch, atq, atrm:安排、检查及删除队列中的工作。

由 Thomas Koenig 编写。

### 2. 部分参数说明

- at [-V] [-q 队列] [-f 文件名] [-mldbv] 时间  
at -c 作业 [作业...]  
atq [-V] [-q 队列] [-v]  
atrm [-V] 作业 [作业...]  
batch [-V] [-q 队列] [-f 文件名] [-mv] [时间]
- at 在设定的时间执行作业。
- atq 列出用户排在队列中的作业,如果是超级用户,则列出队列中的所有工作。
- atrm 删除队列中的作业。
- batch 用低优先级运行作业,只要系统的 loadavg(系统平均负载) < 1.5(或者在 atrun 中设定的值),它就开始执行作业。
- V : 若用-V 参数,则显示版本号到标准错误输出。
- q 队列:若用-q 参数,则指定可选队列名称,队列名称可以是 a 到 z 或 A 到 Z 之间的任意字母。at 的缺省队列名是 c, batch 的缺省队列名是 E, 队列的字母顺序越高,则队列的优先级越低。如果是大写字母的话则提交给 batch, 如果 atq 使用-q 参数,则只显示这个队列中的作业。
- m : 执行完作业后立即使此作业并没有输也给提交作业的用户发送提示 mail。
- f 文件名 : 从文件中读取作业。
- l : 等于 atq。
- d : 等于 atrm。
- b : 等于 batch。

时间 : 这是用户设定的作业开始执行的时间。时间的格式分成三个部分:时间、日期及偏移量。可接受的时间形式是 HHMM 或 HH:MM, 在一天中指定的时间运行,如果时间过去了就在第二天执行。可以在时间后加入 AM 或 PM 使其在上午或下午运行,也可以指定在哪一天执行,给出日期的格式应为 MMDDYY, MM/DD/YY 或 MM.DD.YY, 也可以给出偏移量:

时间 + 计数 时间间隔

时间间隔可以是 minutes, hours, days, weeks。

也可以指定 today 让作业在今天执行, tomorrow 指定作业在明天执行。

### 3. 注意事项

如果没有指定-f 选项,则 at 从标准输入读入所有的命令,所以可以通管道,重定向,或交互输入来输入命令。

超级用户可以在任何情况下使用 at 系列的命令。一般用户使用 at 系列命令的权利由文件/etc/at.allow 和/etc/at.deny 控制。如果/etc/at.allow 存在,则只有在这个文件中的用户才能使用 at 系列的命令。如果/etc/at.allow 文件不存在,则检查文件/etc/at.deny 这个文件。只要不列在这个文件中的用户都可以使用 at 系列的命令。缺省的配置是/etc/at.deny 为一个空文件,这表明所有的用户都可以使用 at 系列的命令。

#### 4. 范例

```
at -f work 4pm + 3 days
```

在三天后下午 4 点执行文件 work 中的作业。

```
at -f work 10am Jul 31
```

在七月 31 日上午 10 点执行文件 work 中的作业。

### 4.1.3 banner

#### 1. 简介

banner - 打印大标题。

banner 在标准输出上打印高质量的标题,如果没有输入要打印的文字,则其等待从标准输入输入一行文字。其标题由 “\*” 组成。

由 Mark Horton 编写。

#### 2. 部分参数说明

```
/usr/bin/banner | -wn | 信息
```

-w 宽度 输出宽度从 132 到 n,n 缺省为 80 列。

#### 3. 注意事项

不能打印字符: <, >, |, ., \, ^, \_ , !, :, ;, l, ~。信息长度为 10 个英文字符。如果用空格分开字符串,还是连续打印字符串,两个字符串可以被括在引号(“)中。这使 banner 把这些字符串置于同一行中。

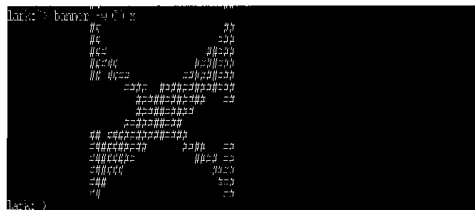


图 4.1 banner 输出

#### 4. 范例

如图4.1。

#### 4.1.4 bash

##### 1. 简介

bash - GNU Bourne-Again Shell

自由软件基金会(Free Software Foundation Inc),拥有 bash 版权。bash 是 Linux 下的许多命令解释器中的一个,同 sh 兼容,并且包含了 ksh 和 csh 中一些有用的特性。遵从 IEEE Posix Shell and Tools specification (IEEE Working Group 1003.2)。

##### 2. 部分参数说明

bash [ 参数 ] [ 文件名 ]

- c 字符串 : 若用 -c 参数,则 bash 从字符串中读入命令,如果字符串后还有变量就被设定为从 \$0 开始的位置参数。
- i : 若用 -i 参数,则 bash 是交互的。
- s : 若用 -s 参数,则 bash 从标准输入中读入命令(在执行完 -c 带的命令之后),直到输入 exit。
- : 单一的“-”符号表明参数执行完毕,并且屏蔽此后所跟参数,后面的所有变量都被看作是文件名。
- norc : 如果 bash 是交互的,则不执行个人初始化文件: ~/.bashrc,如果 bash 作为 sh 来运行,这个参数缺省是关闭的。
- nopprofile : 不执行系统范围内的启动文件 /etc/profile,也不执行个人的启动文件 ~/.bash\_profile、~/.bash\_login 或 ~/.profile,缺省情况下, bash 作为登录的 shell 时以这些文件作为启动文件。
- rcfile 文件名 : 如果 bash 是交互的,则以此文件作为 bash 的启动文件,替代 ~/.bashrc。
- version : 在 bash 开始时显示此 bash 的版本号。
- quiet : 不显示版本号和其他信息,这是缺省值。
- login : 激活 bash,伪装为登录 shell。
- nobraceexpansion : 不执行大括号扩展。
- nolinediting : 在交互状态下不使用 GNU 的 readline 库去读取命令,即取消了命令行编辑功能。
- posix : 改变 bash 的行为,使其符合 Posix 1003.2 规定的标准。

##### 3. 注意事项

bash 过于庞大,运行速度也太慢,并且不符合 Posix 标准。在某些情况下 aliases 模糊不清。

##### 4. 范例

```
lark:~ > bash -version
GNU bash, version 1.14.7(1)
```

```
bash$
```

显示此 `bash` 的版本号,并进入一个交互的 `shell`。

#### 4.1.5 bc

##### 1. 简介

`bc` - 一个简单的计算器。

作者为 Philip A. Nelson., `bc` 是一个支持交互式的表达式计算的语言,其格式类似于 C 语言,在命令行上还可以加载一个标准的数学库。

##### 2. 部分参数说明

- `bc [-lwsqv] [长选项] [文件名 ...]`
- `-l` 和 `--mathlib` : 定义标准数学库。
- `-w` 和 `--warn` : 给出对不符合 Posix 标准的扩展功能的警告。
- `-s` 和 `--standard` : 变成符合 Posix 标准的 `bc`。
- `-q` 和 `--quiet` : 不输出 GNU `bc` 的欢迎信息。
- `-v` 和 `--version` : 输出版本号 and 版权信息后退出。

##### 3. 范例

```
lark:~> bc
bc 1.04
Copyright (C) 1991, 1992, 1993, 1994, 1997 Free Software Foundation, Inc.
This is free software with ABSOLUTELY No WARRANTY.
For details type 'warranty'.
1 + 2 * 3.1415926
7.2831852
quit
lark:~>
```

#### 4.1.6 cal

##### 1. 简介

`cal` - 显示一个日历。

最早出现在 Version 6 AT&T UNIX。`cal` 在标准输出上显示日历。对于查询某年某月某日的用户来说是十分有用的。

##### 2. 部分参数说明

- `cal [-jy] [月 [年]]`
- `-j` : 显示 julian 日期(日子是从 1 月 1 日开始累加的)
- `-y` : 显示今年的日历

年份必须写全,“cal 89”是错误的,应该为“cal 1989”。月份必须是在 1~12 之间的数字,或英文的全称(过少无法区分)。不加参数则输出当前月的日历。

### 3. 范例

```
lark:~ > cal 3 1998
      March 1998
Su Mo Tu We Th Fr Sa
 1  2  3  4  5  6  7
 8  9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31
```

```
lark:~ >
```

但是 1752 年 9 月的月历就非常奇怪,少了 13 天。

```
lark:~ > cal 3 1998
      September 1752
Su Mo Tu We Th Fr Sa
      1  2 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
```

这是因为这个月开始整个大不列颠帝国(美国那时还是英国殖民地),从凯撒历更换成现行的公历。

#### 4.1.7 cat

##### 1. 简介

cat - 连接文件并打印到标准输出。

cat 是 CATenate 的缩写,常用来显示文件,类似于 DOS 下的 TYPE 命令。

##### 2. 部分参数说明

```
cat [-benstuvAET] [-number] [-number-nonblank]
    [-squeeze-blank] [-show-nonprinting] [-show-ends]
    [-show-tabs] [-show-all] [-help] [-version] [文件名...]
```

-b, -number-nonblank : 计算所有非空输出行,从第一行开始计数。

-e : 作用等于 -vE。

-n, -number : 计算所有输出行,从第一行开始计数。

-s, -squeeze-blank : 将所有连续的多行空行替换为一个空行。

-t : 作用等于 -vT。

-u : 没有作用,只是为了某些 UNIX 的兼容性而设。

-v, -show-nonprinting : 显示除换行符和 TAB 以外的所有控制符。使用'作标志,并在字符的高位放置 Escape 控制符。

-A, -show-all : 作用等于 -vET。

-E, -show-ends : 在每一行的结尾加上字符 "\$"。

-T, -show-tabs : 显示控制符 TAB 为'T'。

- help: 输出帮助信息并退出(返回值为非零值)。
- version: 输出版本信息并退出。

### 3. 范例

```
lark: ~ > cat -b -E .lessrc          显示文件 .lessrc 的内容
1      back-line      $ $
3      forw-scroll $
4      back-scroll $
5      goto-line $
6      goto-end $

lark: ~ > cat myfile1 myfile2 > tmp  将文件 myfile1,myfile2 连接起来输出到文件 tmp。
```

## 4.1.8 cd

### 1. 简介

cd - 改变当前目录。

### 2. 部分参数说明

cd 目录名

如果不加目录名,则回到用户的根目录,此外用户必须拥有读该目录的权限。进入另外一个用户的目录只要 cd ~用户名即可。

### 3. 范例

```
lark:/tmp> cd id1          进入 id1 目录
lark:/tmp/id1> cd ~chenxy  进入用户 chenxy 的目录
lark: ~chenxy>
```

## 4.1.9 chgrp

### 1. 简介

chgrp - 改变文件的组。

### 2. 部分参数说明

```
chgrp [-Rcfv] [-recursive] [-changes] [-silent]
        [-quiet] [-verbose] [-help] [-version] 组 文件名...
```

-c —changes: 只有在文件的组确实改变时才进行详细说明。

-f —quiet —silent: 不输出组不能改变的文件的错误信息。

-v —verbose: 详细说明组的变化。

-R —recursive: 改变本目录及其所有子目录的文件的组。

—help: 在标准输出上输出帮助信息并退出。

—version: 在标准输出上输出版本信息并退出。

组: 要改变到的组可以是组号对应的数字,也可以是/etc/group 文件中的组名。

文件名：以空格分开的要改变组所有权的文件列表，支持通配符。  
如果用户不是该文件的属主或超级用户，则不能改变该文件的组。

### 3. 范例

```
lark:~ > chgrp -R book /opt/local/book/*.*
```

改变/opt/local/book/及其子目录下的所有文件的组为 book。

## 4.1.10 chmod

### 1. 简介

chmod -改变文件保护，文件保护控制用户对文件的访问权，有三个安全级别：所有者级别，组访问级别和其他用户访问级别。在这三个级别中，又有三种权限：读(r)、写(w)与执行(x)。(用户可以 ls -lg 来观看某一文件的所属的 group)对于文件来说读权限意味着可以看文件的内容，写文件权可以修改或删除文件，执行权限则可以执行它(类似于 DOS 下的 EXE, COM, BAT 文件)。对于目录来说，读权限意味着可以查看目录下的内容，写权限意味着能在目录下建立新文件，并可以从目录中删除文件，执行权限意味着可以从一个目录转到另一个目录。

### 2. 部分参数说明

```
chmod [-Rcfv] [-recursive] [-changes] [-silent]
      [-quiet] [-verbose] [-help] [-version] 保护权限 文件名...
```

-c —changes：只有在文件的权限确实改变时才进行详细说明。

-f —silent -quiet：不输出权限不能改变的文件的错误信息。

-v —verbose：详细说明权限的变化。

-R —recursive：改变本目录及其所有子目录的文件的权限。

—help：在标准输出上输出帮助信息并退出。

—version：在标准输出上输出版本信息并退出。

保护权限：格式为[ugoa...][[+-=][rwxXstugo...]]...[,...]

“ugoa”控制哪些用户对该文件的权限将被改变：(u)文件的所有者，(g)与文件所有者同组的用户，(o)其他组的用户，(a)所有用户。操作符“+”使得用户选择的权限被追加到每个目标文件，操作符“-”使得这些权限被撤销，“=”使得目标文件只具有这些权限。“rwxXstugo”选择新的属性，(r)读权限，(w)写权限，(x)执行权，(或对目录的访问权)，(X)只有目标文件对某些用户是可执行的或该目标文件是目录时才追加 x 属性，(s)同时设定用户或组 ID，(t)保存程序的文本到交换设备上，(u)目标文件属主，(g)目标文件属主所在的组，(o)其他用户。如果用数字来表示属性，则(0)没有权限，(1)执行权，(2)读权，(4)写权，然后将其相加，所以数字属性的格式应为 3 个从 0 到 7 的八进制数其顺序是 (u)(g)(o)。

文件名：以空格分开的要改变权限的文件列表，支持通配符。

### 3. 范例

```
lark~ > chmod a+x destfile 使所有用户对文件 destfile 有读写执行权。
```

`lark ~ > chmod 644 destfile` 使所有用户可以读文件 `destfile` 只有属主才能改变。

#### 4.1.11 chown

##### 1. 简介

`chown` - 改变文件的属主和组。

##### 2. 部分参数说明

`chown [-Rcfr] [-recursive] [-changes] [-help] [-version] [-silent] [-quiet] [-verbose]  
[用户] [: [组]] 文件名...`

`-c` `--changes` : 只有在文件的属主确实改变时才进行详细说明。

`-f` `--silent` `--quiet` : 不输出属主不能改变的文件的错误信息。

`-v` `--verbose` : 详细说明属主的变化。

`-R` `--recursive` : 改变本目录及其所有子目录的文件属主。

`--help` : 在标准输出上输出帮助信息并退出。

`--version` : 在标准输出上输出版本信息并退出。

用户: 可以是用户名或用户 `id`。

组: 可以是组名或组的 `id`。

文件名: 以空格分开的要改变权限的文件列表, 支持通配符。

##### 3. 范例

`lark ~ > chown dfbb:book destfile` 将文件 `destfile` 的属主改成 `dfbb` 组改成 `book`。

#### 4.1.12 chsh

##### 1. 简介

改变用户登录的 `shell`。

作者 Julianne Frances Haugh

##### 2. 部分参数说明

`chsh [-s 登录的 shell] [用户名]`

普通用户可以改变自己的登录 `shell`, 超级用户可以改变其他用户的登录 `shell`, 所有的 `shell` 必须是在 `/etc/shells` 文件中列出的 `shell`, 但是超级用户不受这个限制。如果用户的 `shell` 是一个受限制的 `shell`, 则用户不能改变其登录 `shell`。如果不加 `-s` 参数则 `chsh` 输出用户的当前登录 `shell` 并提示用户输入新 `shell` 的路径。改变 `shell` 时都需要输入用户的密码。

##### 3. 范例

`lark: ~ > chsh`

Password:

Changing the login shell for dfbb

Enter the new value, or press return for the default

```
Login Shell [/bin/csh]: /bin/tesh
```

```
lark:~ >
```

#### 4.1.13 clear

##### 1. 简介

clear - 清除屏幕(类似于 DOS 的 cls)。

##### 2. 范例

lark~ > clear 清除屏幕,提示符被移动到左上角。

#### 4.1.14 compress

##### 1. 简介

compress, uncompress, zcat - 压缩,展开文件。

compress 利用 LZW 原理来压缩文件,原文件将被替代为扩张名为 .Z 的新文件。

uncompress 和 zcat 则被用来展开压缩文件。

##### 2. 部分参数说明

```
compress [-f] [-v] [-c] [-V] [-b bits] [文件名 ...]
uncompress [-f] [-v] [-c] [-V] [文件名 ...]
zcat [-V] [文件名 ...]
```

-f: 强制替代所有原文件,如果不加此参数且 compress 工作在后台则 compress 将提示用户,该参数对于压缩整个目录的文件特别有用。

-v: 在压缩过程中输出每个文件的压缩比。

-c: 将压缩结果输出到标准输出。

-V: 输出版权信息。

-b bits: bits 为 9~16 的数字,用来控制 LZW 的压缩深度。

文件名:以空格分开的要压缩的文件列表,支持通配符。

##### 3. 范例

```
lark:~ > compress -f -v -V destfile 压缩文件 destfile
Based on compress.c, v 4.0 85/07/30 12:50:00 joe Release
Options: BITS = 16
destfile: Compression: 5.55% — replaced with destfile.Z
lark:~ > uncompress destfile.Z      还原文件 destfile
lark:~ >
```

#### 4.1.15 cp

##### 1. 简介

cp - 拷贝文件。

## 2. 部分参数说明

cp [options] 源文件 目标文件

cp [options] 源文件... 目标目录

Options:

```
[--abdfilprsvxPH] [-S backup-suffix] [-V {numbered,existing,simple}] [--backup]
[--no-dereference] [--force] [--interactive] [--one-file-system] [--preserve] [--recursive]
[--update] [--verbose] [--suffix = backup-suffix] [--version-control = {numbered,existing,simple}]
[--archive] [--parents] [--link] [--symbolic-link] [--help] [--version]
```

-a, --archive : 在拷贝过程中保留尽可能多的源文件的结构和属性,同-dpR 作用相同。

-b, --backup : 给将要覆盖和删除的文件作备份。

-d, --no-dereference : 只拷贝符号连接,而不拷贝其指向的文件并在拷贝中保留原有的连接关系。

-f, --force : 删除已存在的目标文件。

-i, --interactive : 在删除已存在的目标文件时给出提示。

-l, --link : 用硬连接来代替非目录文件的拷贝。

-P, --parents : 将给出路径的源文件连路径一起拷贝。

```
cp --parents a/b/c exist-ing_dir' copies
```

将文件 a/b/c 拷贝到 exist-ing\_dir/a/b/c,

-p, --preserve : 保留源文件的属主、组、权限和时间标志。

-r : 整个目录拷贝。

-s, --symbolic-link : 用建立符号连接来代替非目录文件的拷贝,除非目标文件也在当前目录不然所有的源文件都是绝对路径(从“/”开始),在不支持符号连接的系统上将给出错误信息。

-u, --update : 如果非目录目标文件已存在且具有相同或更新的修改时间则不拷贝。

-v, --verbose : 在拷贝时输出每个文件的名称。

-x, --one-file-system : 从开始拷贝的文件跳过在不同文件系统上的子目录。

-R, --recursive : 整目录拷贝。

--help : 在标准输出上输出帮助信息并退出。

--version : 在标准输出上输出版本信息并退出。

-S, --suffixbackup-suffix : 用于产生备份文件的后缀,此选项修改由环境变量 SIMPLE\_BACKUP\_SUFFIX 设定的后缀,如果都没有定义则缺省是“~”。

-V, --version-control {numbered,existing,simple} : 备份产生的类型。此选项修改由环境变量 VERSION\_CONTROL 设定备份产生的类型,如果都没有定义则缺省是“existing”其有效选项是:

't' or 'numbered' 总是建立编号备份。

'nil' or 'existing' 只为已经有编号备份的文件建立编号备份,其他文件作单--备份。

'never' or 'simple' 只建立单一备份。

### 3. 范例

```
lark: ~ > cp sourcefile destfile      拷贝文件 sourcefile 到文件 destfile。
lark: ~ > cp * /tmp                    拷贝当前目录下所有文件到/tmp 目录。
lark: ~ >
```

## 4.1.16 cpio

### 1. 简介

cpio - 向存档中拷贝文件或从存档中拷贝文件。

### 2. 部分参数说明

```
cpio {-o|-create} [-0acvABLV] [-C bytes] [-H format] [-M
message] [-G [[user@]host:]archive] [-F
[[user@]host:]archive] [--file=[[user@]host:]archive]
[--format=format] [--message=message] [--null] [--reset-
access-time] [--verbose] [--dot] [--append] [--block-
size=blocks] [--dereference] [--io-size=bytes] [--quiet]
[--force-local] [--help] [--version] < name-list {>
archive]
```

```
cpio {-i|-extract} [-bcdfmnrtsuvBSV] [-C bytes] [-E file]
[-H format] [-M message] [-R [user][:][group]] [-I
[[user@]host:]archive] [-F [[user@]host:]archive]
[--file=[[user@]host:]archive] [--make-directories]
[--nonmatching] [--preserve-modification-time] [--numeric-
uid-gid] [--rename] [--list] [--swap-bytes] [--swap]
[--dot] [--unconditional] [--verbose] [--block-
size=blocks] [--swap-halfwords] [--io-size=bytes] [--pat-
tern-file=file] [--format=format]
[--owner=[user][:][group]] [--no-preserve-owner] [--mes-
sage=message] [--force-local] [--no-absolute-filenames]
[--sparse] [--only-verify-crc] [--quiet] [--help] [--ver-
sion] [pattern...] [< archive]
```

```
cpio {-p|-pass-through} [-0adlmuvLV] [-R
[user][:][group]] [--null] [--reset-access-time] [--make-
directories] [--link] [--quiet] [--preserve-modification-
time] [--unconditional] [--verbose] [--dot] [--derefer-
ence] [--owner=[user][:][group]] [--no-preserve-owner]
[--sparse] [--help] [--version] destination-directory <
name-list
```

cpio 有三种模式:

(1).copy-out 模式:向存档文件中拷贝文件,从标准输入给出的文件列表(这个文件列表可以用 find 命令来产生)读取源文件名,向标准输出写入存档文件。

(2).copy-in 模式:从存档文件中拷贝文件,从标准输入读入存档文件,可以用通配符来确定要拷贝的文件名,如果不给出文件名则拷贝所有文件。

(3).copy-pass 模式:将文件从一个目录拷贝到另外一个目录,实际上是把 copy-out 和 copy-in 模式结合起来,只不过不只用一个存档文件,从标准输入给出的文件列表读取源文件名,目标目录作为一个不带“-”的参数给出。

cpio 支持下列文件格式:

binary, old ASCII, new ASCII, crc, HPUX binary, HPUX old ASCII, old tar, and POSIX.1 tar。

缺省的 cpio 产生 binary 格式的存档文件。

-O, -null:在 copy-out 和 copy-pass 模式,文件列表由空字符结束而不是由换行结束。

-a, -reset-access-time:在读取文件时重置文件的时间。

-A, -append:附加到已有的存档文件后,只工作在 copy-out 模式,存档文件必须是由 -O, -F 指定的磁盘文件。

-b, -swap:在 copy-in 模式既交换 bytes 又交换 halfwords 等于 -sS。

-B:将 I/O 的块大小改变为 5120bytes,缺省是 512bytes。

-bblock-size = BLOCK-SIZE:将 I/O 的块大小改变为 BLOCK-SIZE\*512bytes。

-c:使用老的 ASCII 存档格式。

-C IO-SIZE, -io-size = IO-SIZE:将 I/O 的块大小改变为 IO-SIZE bytes。

-d, -make-directories:如果需要就创建目录。

-E FILE, -pattern-file = FILE:在 copy-in 模式从文件 FILE 中读出要解开的文件列表。

-f, -nonmatching:只拷贝文件列表中没有的文件。

-F, -file = archive:存档文件的文件名,如果目标设备是另外一台机器上的磁带机则用“HOSTNAME:文件名”来处理,cpio 自动加上 username@(用户必须在那台机器的 ~/.rhosts 文件中添加本机器)。

-force-local:在使用 -F, -I, -O 选项时即使文件名中有“:”号也把它作为一个本地文件来处理。

-H FORMAT, -format = FORMAT:存档文件使用 FORMAT(bin,odc,newc,crc,tar,ustar,hpbm,hpode)格式。

-i, -extract:使用 copy-in 模式。

-I archive:等同于 -F, -file = archive。

-k:无用(为某些兼容性而设)。

-l, -link:如果可能用连接文件代替拷贝文件。

-L, -dereference:对于文件连接直接拷贝文件而不是连接。

-m, -preserve-modification-time:在产生文件时恢复文件的修改时间。

- M MESSAGE, --message = MESSAGE : 在备份媒体用完时给出提示信息 MESSAGE(中间用%d给出卷标)。
- n, --numeric-uid-gid : 在文件列表时用数字的 UID 和 GID 代替名字。
- no-absolute-filenames : 在 copy-in 模式用相对路径代替绝对路径。
- no-preserve-owner : 在 copy-in 和 copy-pass 模式,在恢复时不改变文件的属主恢复这些属主,本参数对于普通用户是缺省选项。
- o, --create : 使用 copy-out 模式。
- O archive : 等同于 -F, --file = archive。
- only-verify-crc : 在读 CRC 格式的存档文件时只检查每个文件的 CRC 而不解开发件。
- p, --pass-through : 使用 copy-pass 模式。
- quiet : 不输出已拷贝的块的数目。
- r, --rename : 交互式地给文件改名。
- R [user[:.][group], --owner [user[:.][group] : 在 copy-out 和 copy-pass 模式将所有文件的属主和组改变成设定的用户和组(只有超级用户才能使用)。
- sparse : 在 copy-out 和 copy-pass 模式将小文件写入一个大块时用零来填补剩余部分。
- s, --swap-bytes : 在 copy-in 模式交换 bytes。
- S, --swap-halfwords : 在 copy-in 模式交换 halfwords。
- t, --list : s'输出输入的文件列表。
- u, --unconditional : 替代所有同名的现存文件而不给出提示。
- v, --verbose : 列出正在处理的文件给出 "ls -l" 格式的文件列表。
- V --dot : 在处理每个文件时输出 "."。
- version : 输出版权信息并退出。

### 3. 范例

```
lark:~> cpio -o < filelist > d
cpio: sourcefile: truncating inode number
1 block
lark:~>
```

从文件 filelist 中读出源文件列表,读入源文件并写入存档文件 d

```
lark:~> cpio -i -v -F d
sourcefile
1 block
lark:~>
```

从存档文件 d 中拷贝出文件

#### 4.1.17 crontab

##### 1. 简介

crontab - 操作每个用户的守护程序和该执行的时间表。

作者 Matthew Dillon。

## 2. 部分参数说明

`crontab file [-u user]` - 用指定的文件替代目前的 `crontab`。

`crontab - [-u user]` - 用标准输入替代目前的 `crontab`。

`crontab -l [user]` - 列出用户目前的 `crontab`。

`crontab -e [user]` - 编辑用户目前的 `crontab`。

`crontab -d [user]` - 删除用户目前的 `crontab`。

`crontab -c dir` - 指定 `crontab` 的目录。

`crontab` 文件的格式: `M H D m d cmd`。

`M` : 分钟(0-59)。

`H` : 小时(0-23)。

`D` : 天(1-31)。

`m` : 月(1-12)。

`d` : --星期内的天(0~6, 0 为星期天)。

`cmd` 要运行的程序, 程序被送入 `sh` 执行, 这个 `shell` 只有 `USER`, `HOME`, `SHELL` 这三个环境变量。

下面是一个例子文件:

```
# MIN HOUR DAY MONTH DAYOFWEEK COMMAND
# 每天早上 6 点
10 6 * * * date

# 每两个小时
0 */2 * * * date

# 晚上 11 点到早上 8 点之间每两个小时, 早上 8 点
0 23-7/2,8 * * * date

# 每个月的 4 号和每个礼拜的礼拜一到礼拜三的早上 11 点
0 11 4 * mon-wed date

# 1 月份日早上 4 点
0 4 1 jan * date
```

## 3. 范例

```
lark:~ > crontab -l          列出用户目前的 crontab。
# MIN HOUR DAY MONTH DAYOFWEEK COMMAND
10 6 * * * date
0 */2 * * * date
0 23-7/2,8 * * * date

lark:~ >
```

#### 4.1.18 csh

##### 1. 简介

见 tcsb

#### 4.1.19 cut

##### 1. 简介

cut - 将文件中的每一行都去掉一个域。

##### 2. 部分参数说明

```
cut  {-b  byte-list,  --bytes = byte-list}  [-n]  [--help]
      [--version] [file...]
```

```
cut  {-c  character-list,  --characters = character-list}
      [--help] [--version] [file...]
```

```
cut  {-f  field-list,  --fields = field-list} [-d delim] [-s]
      [--delimiter = delim]  [--only-delimited]  [--help]  [--ver-
      sion] [file...]
```

byte-list、character-list 和 field-list 是数字或一个区间,最小是 1,区间格式为“n-m”,n 为开始的数字,m 为结束的数字,“n-”意味着到行尾。

-b, --bytes byte-list : 只输出在 byte-list 区间的 bytes, TABs 和 backspaces 都占 1 byte。

-c, --characters character-list : 只输出在区间 character-list 内的字符 TABs 和 backspaces 都占一个字符。

-f, --fields field-list : 只输出在 field-list 内的域。

-d, --delimiter delim : 指定在 -f 参数中的 field-list 的分割符(为 delim 中的第一个字符,缺省为 TAB)。

-n : 不分割多 byte 的字符。

-s, --only-delimited : 在 -f 参数中不输出没有域分割符的行。

--help : 在标准输出上输出帮助信息并退出。

--version : 在标准输出上输出版本信息并退出。

##### 3. 范例

```
lark:~ > cat sourcefile
total 0
-rw-r--r--  1 fangh  users      0 Aug 23 21:46 destfile
lark:~ > cut -c 5-20 sourcefile
10
-rw-r--r--  1 fangh
```

lark: ~ >

#### 4.1.20 date

##### 1. 简介

date - 打印或设置系统日期和时间。

##### 2. 部分参数说明

```
date [-u] [-d datestr] [-s datestr] [--utc] [--universal]
    [--date = datestr] [--set = datestr] [--help] [--version]
    [+FORMAT] [MMDDhhmm[[CC]YY][.ss]]
```

+FORMAT :控制如何显示日期。

指令:% : %号;n : 新行;t : TAB。

时间域:

```
%H    小时 (00..23);
%I    小时 (01..12);
%k    小时 ( 0..23);
%l    小时 ( 1..12);
%M    分钟 (00..59);
%p    AM 或 PM;
%r    12 小时 (hh:mm:ss [AP]M);
%s    从 1970-01-01 00:00:00 UTC 开始的时间;
%S    秒 (00..61);
%T    24 小时 (hh:mm:ss);
%X    时间表示 (%H;%M;%S);
%Z    时区(如 E)T)如没有指定时区则不显示。
```

日期域:

```
%a    缩写形式的星期名 (Sun..Sat);
%A    星期名 (Sunday..Saturday);
%b    缩写形式的月名 (Jan..Dec);
%B    月名 (January..December);
%c    日期和时间 (Sat Nov 04 12:02:33 EST 1989);
%d    每个月的第几天 (01..31);
%D    日期 (mm/dd/yy);
%h    等同于 %b;
%j    一年的第几天 (001..366);
%m    月份 (01..12);
%U    一年中的第几个星期,星期天为星期的第一天 (00..53);
%w    每个星期的第几天(0..6) 0 为星期天;
%W    一年中的第几个星期,星期一为星期的第一天 (00..53);
```

%x 日期 (mm/dd/yy);  
 %y 年份的后两个数字 (00..99);  
 %Y 年份 (1970...).  
 [MMDDhhmm[[CC]YY][.ss]  
 MM 月份;  
 DD 日期;  
 hh 小时;  
 mm 分钟;  
 CC 年份的前两位 (可选项);  
 YY 年份的后两位 (可选项);  
 ss 秒 (可选项)。

只有超级用户才有权限设置系统时间(启动时在 CMOS 中读出)。

-d datestr, --date dates : 显示 datestr 中指定的日期和时间。

-s datestr, --set datestr : 设定 datestr 中指定的日期和时间。

-u, --universal : 显示或设定格林威治时间(缺省为本地时间)。

--help : 在标准输出上输出帮助信息并退出。

--version : 在标准输出上输出版本信息并退出。

### 3. 范例

打印两天前的时间:

```
lark: ~ > date --date '2 days ago'
Fri Aug 22 20:20:08 CDT 1997
lark: ~ >
```

打印月份和日期:

```
lark: ~ > date '+ %B %d'
August 24
lark: ~ >
```

## 4.1.21 dd

### 1. 简介

dd - 拷贝一个文件(并可以同时转化它)

### 2. 部分参数说明

```
dd [-help] [--version] [if = file] [of = file] [ibs = bytes]
[obs = bytes] [bs = bytes] [cbs = bytes] [skip = blocks]
[seek = blocks] [count = blocks]
[conv = |ascii,ebcdic,ibm,block,unblock,lcase,ucase,swab,noerror,notrunc, sync]]
```

if = file : 以 file 作为源文件以代替标准输入。

of = file : 以 file 作为目标文件以代替标准输出。

ibs = bytes : 一次读 bytes 个 byte。

obs = bytes : 一次写 bytes 个 byte。

cbs = bytes : 一次读写 bytes 个 byte, 这个参数将改写 ibs 和 obs 的值。

skip = blocks : 从开头忽略 blocks 个 ibs 大小的块。

seek = blocks : 从开头忽略 blocks 个 obs 大小的块。

count = blocks : 只拷贝 blocks 个 ibs 大小的块。

conv = conversion[, conversion...]: 转换文件, 可转换的格式计有: ascii, ebcdic, ibm, block, unblock, lcase, ucase, swab, noerror, notrunc, sync

—help : 在标准输出上输出帮助信息并退出。

—version : 在标准输出上输出版本信息并退出。

### 3. 范例

将文件 sourcefile 拷贝到文件 destfile。

```
lark ~ > dd if= sourcefile of= destfile
0+1 records in
0+1 records out
lark: ~ >
```

## 4.1.22 df

### 1. 简介

df - 报告磁盘剩余空间。

### 2. 部分参数说明

```
df [-aikPv] [-t fstype] [-x fstype] [--all] [--inodes]
[--type = fstype] [--exclude-type = fstype] [--kilobytes]
[--portability] [--print-type] [--help] [--version] [file-
name...]
```

-a, --all : 列出 BLOCK 为零的文件系统缺省是不列出。

-i, --inodes : 用 inode 的使用状况来代替 block 的使用状况。

-k, --kilobytes : 用 1K 为单位来输出 block (缺省是 512bytes)。

-P, --portability : 使用 POSIX 格式输出。

-T, --print-type : 输出每个文件系统的类型。

-t, --type = fstype : 只输出列在 fstype 中的类型的文件系统。

-x, --exclude-type = fstype : 只输出不在 fstype 中的类型的文件系统。

-v : 无用, 只是为了同 SYSTEM V 版本的 df 兼容。

—help : 在标准输出上输出帮助信息并退出。

—version : 在标准输出上输出版本信息并退出。

### 3. 范例

```
lark: ~ > df -a -T
```

```

Filesystem      Type 1024-blocks  Used Available Capacity Mounted on
/dev/hda1      ext2  497667      335214    136751      71% /
none           proc      0            0          0          0% /proc
/dev/hda3      ext2  2454879     105267    2222699      5% /home
/dev/hda2      ext2  995147      49603     894136      5% /usr/local
lark: ~ >

```

#### 4.1.23 diff

##### 1. 简介

diff - 比较两个文件的不同。

##### 2. 部分参数说明

diff [选项] 源文件 目标文件

- a : 将所有文件当作文本文件来处理。
  - b : 忽略空格造成的不同。
  - B : 忽略空行造成的不同。
  - q --brief : 只报告何处不同, 不报告具体信息。
  - c : 使用纲要输出格式。
  - e --ed : 输出 ed 可以编辑的格式。
  - f : 输出同-e 类似的反序格式。
  - H : 利用试探法加速对大文件的搜索。
  - i : 忽略大小写的变化。
  - l --paginate : 用 pr 对输出分页。
  - n --rcs : 输出 RCS 格式。
  - r --recursive : 比较目录时比较所有子目录。
  - report-identical-files -s : 两文件相同才报告。
  - v --version : 在标准输出上输出版本信息并退出。
- 常常利用 diff 来产生补丁程序(patch)。

##### 3. 范例

比较 destfile 和 sourcefile

```

lark: ~ > diff sourcefile destfile
1,2c1
< total 0
< -rw-r--r-- 1 faangh  users          0 Aug 23 21:46 filelist
> ../sourcefile
lark: ~ >

```

#### 4.1.24 dialog

##### 1. 简介

用来在 shell 编程中产生对话框。



```
15293 .
lark:/tmp>
```

#### 4.1.26 echo

##### 1. 简介

echo - 显示一行文本。

##### 2. 部分参数说明

```
echo [-ne] [字符串 ...]
echo {-help,--version}
-n : 不输出新行。
-e : 使 echo 可以输出下列 Esc 字符:
\ a   警告(响铃)。
\ b   退格。
\ c   在行尾不另起一行。
\ f   换页。
\ n   换行。
\ r   回车。
\ t   制表符。
\ V   垂直制表符号。
\ \   反斜杠。
\ nnn 字符的八进制码是 nnn。
```

--help : 在标准输出上输出帮助信息并退出。

--version : 在标准输出上输出版本信息并退出。

注意: 大多数 shell 内建 echo 命令, 上述参数只对 /usr/bin/echo 起作用。

##### 3. 范例

```
lark:~> /bin/echo -e just \ \ wait
just \ wait
lark:~>
```

#### 4.1.27 ed

##### 1. 简介

ed red - 文本行编辑器。

##### 2. 部分参数说明

```
ed [-f] [-Gs] [-p 字符串] [文件名]
red [-f] [-Gs] [-p 字符串] [文件名]
- : 在使用 e, r, w, q 和 ! 命令时不输出信息。
-G : 禁止向后一致性, 作用于命令 'G', 'V', 'f', 'I', 'm', 't' 和 '!'
```

-s : 禁止诊断,一般当 ed 的输入是一个脚本文件时使用该选项。  
-p 字符串 : 以该字符串为提示符。  
文件名 : 要编辑的文件名。

### 3. 范例

```
lark:~ > ed .lessrc
114
wq
114
lark:~ >
```

## 4.1.28 egrep

### 1. 简介

见 grep

## 4.1.29 env

### 1. 简介

env - 在修改过的环境中运行程序,运行完后不影响原环境设置。

### 2. 部分参数说明

```
env [-] [-i] [-u name] [--ignore-environment]
[--unset = name] [--help] [--version] [name = value]... [命令
[参数...]]
```

-u, --unset name : 从原环境中删除变量(name 为变量名)。  
-, -i, --ignore-environment : 开始一个空环境,忽略原环境的所有设置。  
--help : 在标准输出上输出帮助信息并退出。  
--version : 在标准输出上输出版本信息并退出。

### 3. 范例

```
lark:~ > env - ls
destfil      destfile     sourcefile
lark:~ >
```

## 4.1.30 expand

### 1. 简介

expand - 将文件中的 TAB 转换为空格并输出到标准输出。

### 2. 部分参数说明

```
expand [-tab1[,tab2[,...]]] [-t tab1[,tab2[,...]]] [-i]
[--tabs = tab1[,tab2[,...]]] [--initial] [--help] [--ver-
```

sion} [文件名...]

-, -t, --tabs tab1[,tab2[,...]] : 如果只有一个 tab1,则将 TAB 转化为 tab1 个空白 (缺省为 8)如果有 tab2,tab3...则将第一个 TAB 转化为 tab1 个空格,第二个 TAB 转化为 tab2 个空格,没有指出的全部转化为一个空格。

-i, --initial : 只转化用来分开两个非空格和 TAB 字符的 TAB 为空格。

--help : 在标准输出上输出帮助信息并退出。

--version : 在标准输出上输出版本信息并退出。

### 3. 范例

```
lark: ~ > expand test
abc
lark: ~ >
```

## 4.1.31 fdformat

### 1. 简介

fdformat - 低级格式化一张软盘。

作者: Werner Almesberger.

### 2. 部分参数说明

fdformat [-n] 设备

支持的设备格式有:

```
/dev/fd0d360 {minor = 4}
/dev/fd0h1200 {minor = 8}
/dev/fd0D360 {minor = 12}
/dev/fd0H360 {minor = 12}
/dev/fd0D720 {minor = 16}
/dev/fd0H720 {minor = 16}
/dev/fd0h360 {minor = 20}
/dev/fd0h720 {minor = 24}
/dev/fd0H1440 {minor = 28}

/dev/fd1d360 {minor = 5}
/dev/fd1h1200 {minor = 9}
/dev/fd1D360 {minor = 13}
/dev/fd1H360 {minor = 13}
/dev/fd1D720 {minor = 17}
/dev/fd1H720 {minor = 17}
/dev/fd1h360 {minor = 21}
/dev/fd1h720 {minor = 25}
/dev/fd1H1440 {minor = 29}
```

## 2. 部分参数说明

-n : 在格式化时不校验

## 3. 范例

```
lark:~ > fdformat /dev/fd0H1440
Double-sided, 80 tracks, 18 sec/track. Total capacity 1440 kB.
Formatting ... done
Verifying ... read: Success
```

### 4.1.32 fgrep

#### 1. 简介

见 grep

### 4.1.33 file

#### 1. 简介

file - 探测文件类型。

## 2. 部分参数说明

```
file [-vczL] [-f namefile] [-m magicfiles] 文件名 ...
```

-v : 在标准输出上输出版本信息并退出。

-m magicfiles : 指定用来代替缺省幻数文件(/etc/magic)的新幻数文件。

-z : 探测压缩过的文件。

-c : 输出正在处理的幻数文件的分析表格,常同-m 参数联合使用来测试新幻数文件。

-f namefile : 从文件 namefile 中读取要分析的文件名列表。

-L : 允许符号连接。

文件名 : 要分析的文件名。

## 3. 范例

```
lark:/tmp> file *
destfile:      ASCII text
elm.rc.OLD:    English text
portnum:       empty
rc.inet1.OLD:  Bourne shell script text
lark:/tmp>
```

### 4.1.34 find

#### 1. 简介

find - 用来在大量目录中搜寻特定文件的强有力的工具。

## 2. 部分参数说明

`find [路径...] [匹配表达式]`

路径... : 要搜寻的目录(可以是多个目录,用空格分开)。

匹配表达式 : 要搜寻的文件匹配标准或说明。

`-name` 文件名 : 告诉 FIND 要找什么文件;要找的文件包括在引号中(支持通配符 \* 和?)。

`-perm` 模式 : 匹配所有符合指定数字模式值的文件,如果模式前面是“-”号则搜寻所有除这个模式以外的所有模式。

`-size n` 匹配大小为 `n` 个 block 的文件(若 `n` 后有 `k` 则为 `nK` 大小的文件)。

`-user` 用户名 : 搜寻所有属主为用户名(名称和 ID 都可以)的文件。

`-group` 组名 : 搜寻所有组为组名的文件。

`-atime n` : 搜寻在前 `n` 天访问过的文件。

`-mtime n` : 搜寻在前 `n` 天修改过的文件。

`-exec` 命令 : 对于每个匹配文件执行命令,标志 `||` 用于指定命令执行时文件名出现的地方,命令必须中止于符号“`\;`”。

`-print` : 输出搜寻结果到标准输出。

## 3. 范例

```
lark: /> find ./ -name "passwd" -print
./usr/bin/passwd
./home/ftp/etc/passwd
./etc/passwd
lark: />
```

### 4.1.35 fortune

#### 1. 简介

fortune - 输出一段有趣的谚语或格言。

作者 : Ken Arnold。

#### 2. 范例

```
lark: /> fortune
"I must have a prodigious quantity of mind; it takes me as much as a
week sometimes to make it up."
    — Mark Twain, "The Innocents Abroad"
lark: />
```

### 4.1.36 grep

#### 1. 简介

grep, egrep, fgrep - 在文件中搜寻匹配的行并输出。

## 2. 部分参数说明

```
grep [-[AB]] num [-[CEFGVBchilnavwx]] [-e ] 正则表达式
-l -file [ 文件名... ]
```

-A num : 输出匹配行后的 num 行。

-B num : 输出匹配行前的 num 行。

-V : 输出版权信息。

-f file : 从文件中读入表达式。

-q : 不输出。

-s : 不输出错误信息。

grep = Global Regular Expression Print, grep 来自 ed 的搜寻所有符合某表达式的命令: “g/re/p”, 其中 re 代表规则表达式。

fgrep 是 fixed grep, 只能查询固定串。

egrep 是 extended grep 扩展查询, 支持某些增强的表达式。

## 3. 范例

在文件 services 中查找含有 ftp 的行。

```
lark:/etc> grep ftp services
ftp          21/tcp
tftp         69/udp
sftp         115/tcp
lark:/etc>
```

### 4.1.37 gzexe

#### 1. 简介

gzexe - 将可执行文件压缩成较小的可执行文件(类似于 DOS 下的 PKLITE, LZEXE)。

## 2. 部分参数说明

```
gzexe [ 文件名... ]
```

-d: 解压缩。

本命令对那些磁盘空间较小的用户特别有用。

## 3. 范例

```
lark:~> ls -la cat
-rwxr-xr-x 1 fangh users 20916 Aug 25 21:33 cat *
lark:~> gzexe cat
cat: 48.9%
lark:~> ls -la cat
-rwxr-xr-x 1 fangh users 11283 Aug 25 21:33 cat *
```

```
lark: ~ >
```

#### 4.1.38 gzip

##### 1. 简介

gzip, gunzip, zcat - 压缩或展开文件。

##### 2. 部分参数说明

```
gzip [ -acfhLnNrtvV19 ] [ -S suffix ] [ 文件名 ... ]
```

```
gunzip [ -acfhLnNrtvV ] [ -S suffix ] [ 文件名 ... ]
```

```
zcat [ -fhLV ] [ 文件名 ... ]
```

-d —decompress —uncompress : 解压。

-h —help : 显示帮助信息并退出。

-t —test : 检验压缩文件。

-V —version : 显示版权信息并退出。

-v : 输出压缩信息。

-# —fast —best : # 为 1(fast)-9(best) 数字越小速度越快, 但压缩比越小, 缺省为 6。

-c —stdout —to-stdout : 压缩结果输出到标准输出, 如不加 -c 参数, 则 gzip 将源文件压缩为 .gz 文件并删除源文件。

##### 3. 范例

```
lark: ~ > gzip -v sourcefile
```

```
sourcefile:          15.2% — replaced with sourcefile.gz
```

```
lark: ~ >
```

#### 4.1.39 head

##### 1. 简介

head - 显示文件的前几行。

##### 2. 部分参数说明

```
head [-c N[bkm]] [-n N] [-qv] [--bytes = N[bkm]] [--lines = N]
```

```
 [--quiet] [--silent] [--verbose] [--help] [--version]
```

```
 [文件名...]
```

```
head [-Nbcklmqv] [文件名...]
```

-c N, —bytes N : 输出文件的前 N 个字节。

-n N, —lines N : 输出文件的前 N 行。

-q, —quiet, —silent : 不输出文件名的信息。

-v, —verbose : 输出文件名的信息。

—help : 在标准输出上输出帮助信息并退出。

—version : 在标准输出上输出版本信息并退出。

### 3. 范例

输出文件 .lessrc 的第一行。

```
lark: ~ > head -n 1 .lessrc
forw-line
lark: ~ >
```

## 4.1.40 hostid

### 1. 简介

hostid - 显示或设置系统的 HOSTID。

作者 : Mitch DSouza。

### 2. 部分参数说明

hostid [-v] [十六进制的 ID]

-v : 给出十进制和十六进制的 ID。

对于每一台机器,其 HOSTID 是唯一的,只有超级用户才能更改。

### 3. 范例

```
lark: ~ > hostid -v
Hostid is 650780800 {0x26ca2080}
lark: ~ >
```

## 4.1.41 id

### 1. 简介

id - 显示当前用户的用户名和组。

### 2. 部分参数说明

id [-a]

-a: 用在支持一用户多组的系统上。

id 在取得用户的 user id 和 group id 后从文件 /etc/passwd、/etc/group 中查出用户的用户名和组名。

### 3. 范例

```
lark: ~ > id
uid = 10000(fangh) gid = 100(users) groups = 100(users)
lark: ~ >
```

## 4.1.42 join

### 1. 简介

join - 从两个文件中取出相同的行,按照指定的关键字为匹配的两行连接成一行。

## 2. 部分参数说明

```
join [-a 1|2] [-v 1|2] [-e empty-string] [-o field-
list...] [-t char] [-j 1|2] field [-1 field] [-2 field]
文件 1 文件 2
join |—help, —version|
```

-a file-number : 文件 1 或 2 中不匹配的行也输出, file-number 可以是 1 或 2。  
 -e string : 用字符串 string 代替输出的空域。  
 -1, -j1 field : 在文件 1 的第 field 个域连接两个文件。  
 -2, -j2 field : 在文件 2 的第 field 个域连接两个文件。  
 -j field : 等于是 -1 field -2 field。  
 -o field-list... : 按照 field-list 中的格式构造输出行。  
 -t char : 用字符 char 作为输入输出域的分割符。  
 -v file-number : 为文件 file-number 的每一个不匹配的行打印一行代替通常的输出。  
 —help : 在标准输出上输出帮助信息并退出。  
 —version : 在标准输出上输出版本信息并退出。

## 3. 范例

```
lark:~ > join testfile1 testfile2
this is testfile is testfile
lark:~ >
```

## 4.1.43 kill

### 1. 简介

kill - 中止一个进程

### 2. 部分参数说明

```
kill [-s 信号 | -p] [-a] 进程号 ...
kill -l [ 信号 ]
```

kill 向指定的进程发出特定的信号, 如果没有指定信号则送出 TERM 信号, TERM 信号将杀死没有捕捉到这个信号的进程。对于某些进程可能要使用 KILL (9) 信号强制杀死。例如: kill -9 11721, 将强制杀死进程 11721。大多数 SHELL 内建 kill 命令。

进程号... : 指定要杀死的进程列表, 每个进程号可以是一个进程的 id 或进程名。

-s : 指定要送出的信号, 信号可以由数字给出, 也可以是信号名。常用的信号有 KILL(不能被进程忽略)和 HUP。

-p : 只打印进程的 ID 不送出信号。

-l : 打印所有的信号列表(在 /usr/include/linux/signal.h 中定义)。

### 3. 范例

杀掉进程 11721

```
lark:~ > ps
PID TTY STAT TIME COMMAND
11668 p1 S   0:00 -tcsh
11721 p1 T   0:00 cat
11737 p1 R   0:00 ps
lark:~ > kill 11721

[1]      Terminated                  cat
lark:~ >
```

#### 4.1.44 last

##### 1. 简介

last - 显示过去多少个用户或终端登录到本机器。

##### 2. 部分参数说明

last [-数目] [-f 文件名] [-t tty] [-h 节点名] [-i IP 地址] [-l] [-y] [用户名...]

last 从文件/var/log/wtmp 中读出用户登录和离开的所有记录,并且可以查询某个用户登录的情况、某个节点登录的情况及某个 tty 登录的情况。例如“last root -t console”将列出所有 root 用户在控制台上登录的情况,如果不加参数则 last 将显示出所有用户登录的情况。

- 数目: 确定显示多少行。
- f 文件名: 用指定的 log 文件代替缺省的/var/log/wtmp 文件。
- h 节点名: 只显示从某个节点登录的用户。
- i IP 地址: 只显示从某个 IP 地址登录的用户。
- l: 用 IP 地址显示远端地址。
- t tty: 只显示在特定 tty 上登录的用户。
- y: 输出日期时加上年份。

##### 3. 范例

显示过去 3 次用户 fangh 登录的情况:

```
lark:~ > last -3 fangh
fangh      ttyt1      csun01.ihep.ac.c Tue Aug 26 18:46   still logged in
fangh      ttyt2      csun01.ihep.ac.c Mon Aug 25 22:32 - 23:14   (00:41)
fangh      ttyt2      csun01.ihep.ac.c Mon Aug 25 19:58 - 21:59   (02:01)
lark:~ >
```

#### 4.1.45 less

##### 1. 简介

less - 相对于 more,用来按页显示文件。

作者: Mark Nudelman。

## 2. 部分参数说明

```
less -?
less --help
less -V
less --version
less [-[ + ]aBcCdeEfGgIImMnNqQrsSuUVwX]
      [-b bufs] [-h lines] [-j line] [-k keyfile]
      [-loO] logfile [-p pattern] [-P prompt] [-t tag]
      [-T tagfile] [-x tab] [-y lines] [-[z] lines]
      [+ [+ ]cmd] [--] [文件名]...
```

less 类似于 more,但是 less 允许向前或向后浏览文件。less 并不全部读入文件后才显示,所以在显示大文件时它的显示速度比 vi 之类的编辑器要快。less 采用 termcap,所以它支持的终端类型非常广泛。

- \ ? --help : 列出 less 的所有操作命令。
- a : 查询时从当前屏幕的最后一行之后开始查询。
- bn : 指定 less 为每一个文件开 nK 的缓冲区。
- B : 当 less 的输入是从管道中来的时候,其缓冲区的大小是字典设置的,如果加上本选项,则使用由 -bn 指定大小的缓冲区,但这有可能丢失前面输入的内容。
- c : 从上往下刷屏,缺省的是从下往上刷屏。
- C : 同 -c,只是刷屏之前先清屏。
- d : less 在终端上要输出错误信息,此参数强制 less 继续执行刷屏滚屏之类的命令。
- e : 当 less 第二次到达文件末尾时退出,缺省的退出方式是命令“q”。
- E : 当 less 第一次到达文件末尾时退出。
- f : 打开非正常文件(一个目录,或一个设备文件)。
- hn : 指定回滚的最大行数。
- i : 搜寻时忽略大小写。
- m : 显示已经显示的文件百分比。
- q -Q : 不输出响铃符号。
- r : 显示控制符。
- version -V : 显示版权信息。

## 3. 范例

显示 test 文件

```
lark: ~ > less test
```

### 4.1.46 ln

#### 1. 简介

ln - 在文件间建立连接

## 2. 部分参数说明

ln [参数] 源文件 [目标文件]

ln [参数] 源文件... directory

参数:

[-bdfinsvF] [-S backup-suffix] [-V {numbered,existing,simple}]  
 [--version-control={numbered,existing,simple!}]  
 [--backup] [--directory] [--force] [--interactive]  
 [--no-dereference] [--symbolic] [--verbose] [--suffix=backup-suffix] [--help] [--version]

-b, --backup: 为删除的文件建立备份。

-d, -F, --directory: 允许超级用户建立硬连接。

-f, --force: 删除目标文件。

-i, --interactive: 在删除文件时给出提示。

-n, --no-dereference: 如果目标文件是一个一个目录的符号连接则替代此符号连接,而不是在这个目录中作连接。

-s, --symbolic: 用符号连接代替硬连接。

-v, --verbose: 输出每个要连接的文件的文件名。

--help: 在标准输出上输出帮助信息并退出。

--version: 在标准输出上输出版本信息并退出。

另外对链接文件做改变属性的动作是没有意义的,因为只有它们链接到的文件的属性才是文件的真正属性。

## 3. 范例

将文件 sourcefile 连接到文件 test。

```
lark:~ > ln -s sourcefile test
```

```
lark:~ > ls -la test
```

```
lrwxrwxrwx 1 fangh users 10 Aug 26 20:36 test -> sourcefile
```

```
lark:~ >
```

### 4.1.47 logname

#### 1. 简介

logname - 输出用户的登录时的用户名。

#### 2. 范例

```
lark:~ > logname
```

```
fangh
```

```
lark:~ >
```

### 4.1.48 ls

#### 1. 简介

ls, dir, vdir - 列出目录下的文件 (类似于 DOS 下的 DIR 命令)。

## 2. 部分参数说明

```
ls [-abcdefgiklmnopqrstuxABCFGILNQRSUXI78] [-w cols] [-T
cols] [-l pattern] [--all] [--escape] [--directory]
[--inode] [--kilobytes] [--numeric-uid-gid] [--no-group]
[--hide-control-chars] [--reverse] [--size] [--width = cols]
[--tabsize = cols] [--almost-all] [--ignore-backups]
[--classify] [--file-type] [--full-time] [--ignore = pat-
tern] [--dereference] [--literal] [--quote-name] [--recur-
sive] [--sort = {none,time,size,extension}] [--for-
mat = {long,verbose,commas,across,vertical,single-column}]
[--time = {atime,access,use,ctime,status}]
[--color[ = {yes,no,tty}]] [--colour[ = {yes,no,tty}]]
[--8bit] [--8bit] [--help] [--version] [name...]
```

- a, --all : 列出目录系的所有文件,包括以“.”开头的隐含文件。
- c, --time = ctime, --time = status : 输出文件 i 节点的修改时间,并按此排序。
- d, --directory : 将目录当作文件显示,而不是显示其下的文件。
- e, --full-time : 输出时间的全部信息,而不是输出简略信息。
- f -U, --sort = none: 对目录下的文件不排序。
- g : 无用。
- i, --inode : 输出文件 i 节点的索引信息。
- k, --kilobytes : 如果输出文件大小,则按 K 的形式输出。
- l, --format = long, --format = verbose : 输出文件的详细信息。
- m, --format = commas : 横向输出文件名,以“,”作分割符。
- n, --numeric-uid-gid : 用数字的 UID、GID 代替名称。
- o, --color, --colour, --color = yes, --colour = yes : 输出彩色文件名。
- p -F, --classify : 在每个文件名后附上一个字符以说明该文件的类型。
- q, --hide-control-chars : 用?代替不可输出的字符。
- r, --reverse : 对目录反向排序。
- s, --size : 在每个文件名后输出该文件的大小。
- t, --sort = time : 用时间对目录排序。
- u, --time = atime, --time = access, --time = use: 用文件上次被访问的时间来排序。
- x, --format = across, --format = horizontal : 按列输出,横向排序。
- A, --almost-all : 输出除“.”“..”以外的所有文件。
- B, --ignore-backups : 不输出以“-”结尾的备份文件。
- C, --format = vertical : 按列输出,纵向排序。
- G, --no-group : 输出文件组的信息。
- L, --dereference : 列出连接到的文件。
- N, --literal : 不限制文件的长度。^
- R, --recursive : 列出所有子目录下的文件。

- S, --sort = size : 用文件大小排序。
- X, --sort = extension : 用文件扩展名(最后一个"."后的字符)的字符顺序排序。
- l, --format = single-column : 一行只输出一个文件。
- 7, --7bit : 不输出 SCLl (ISO 646) (0x20-0x7E)以外的字符。
- 8, --8bit : 输出所有 8-bit ISO 8859 (0x20-0x7E, 0xA1-0xFF)字符。
- w, --width cols : 设定输出宽度为 cols 列。
- T, --tabsize cols : 设定每个 TAB 宽度为 cols 列。
- color = no, --colour = no : 不显示彩色文件名。
- help : 在标准输出上输出帮助信息并退出。
- version : 在标准输出上输出版本信息并退出。

### 3. 范例

列出当前目录下的所有文件。

```
lark:~ > ls -la
total 6
drwxr-xr-x  2 fangh  users    1024 Aug 26 20:52 ./
drwxr-xr-x 19 root   root     1024 Aug 26 21:09 ../
-rw-r--r-   1 fangh  users       15 Aug 21 21:57 .bash_history
-rw-r--r-   1 fangh  users       34 Aug 20 20:41 .less
-rw-r--r-   1 fangh  users    114 Aug 25 19:58 .lessrc
-rw-r--r-   1 fangh  users       72 Aug 24 18:43 sourcefile
lrwxrwxrwx   1 fangh  users       10 Aug 26 20:36 test -> sourcefile
lark:~ >
```

## 4.1.49 man

### 1. 简介

man - 显示具有一定格式的在线手册。

### 2. 部分参数说明

- man [-adfhkKtW] [-m system] [-p string] [-C config\_ file]
- [-M path] [-P pager] [-S section\_ list] [section] 查询名 ...
- man 搜寻的路径由 MANPATH 和(MAN)PAGER 这两个环境变量设置。
- C config\_ file: 指定 man.config 这个配置文件的位置, 缺省在 /usr/lib/man.config。
- M path : 指定 man 搜寻存放在线手册的路径。
- P pager : 指定用来显示手册的程序, 缺省为 /usr/bin/less。
- S section\_ list : 选择手册的章节。
- k 关键字: 查询包含该关键字的所有 man page。
- man 对于新手和老手来说都是非常有用的一个工具, 用来快速查询命令和程序的使用方法与参数。编程人员也可以用来查询 C 函数的用法。对于 'rn(1)' 或 'ctime(3)' 这样的输出, 其中括弧中的数字是指 UNIX 手册中该文件所在的章节。当你打 man 3 ctime

时,表示是要查阅在第3节中的 etime 的内容。下面是常见的 UNIX 手册的分类:

- 1 用户命令。
- 2 系统调用。
- 3 库函数。
- 4 设备和设备驱动程序。
- 5 文件格式。
- 6 游戏。
- 7 有用的杂类,如宏命令包。
- 8 系统维护和管理命令。

### 3. 范例

查询 ls 的用法

```
lark ~ > man ls
```

## 4.1.50 mtools

### 1. 简介

mtools - 用来在 UNIX 下访问 DOS 下磁盘的工具包。

### 2. 工具介绍

mc - 进入 DOS 子目录。

```
mc [DOS 下的目录名]
```

mcopy - 拷贝 DOS 文件。

```
mcopy [-tvmOsSrRA] 源文件 目标文件
```

```
mcopy [-tvmOsSrRA] 源文件 [源文件列表] 目标目录
```

```
mcopy [-tvm] MSDOS 源文件
```

t: 将文本文件中的回车和换行符转化为换行符。

n: 覆盖已有文件时并不给出提示信息。

mdel - 删除 DOS 文件。

```
mdel [-v] msdos 文件 [msdos 文件列表]
```

mdeltree - 删除 DOS 目录。

```
mdeltree [-v] msdos 目录 [msdos 目录列表]
```

mdir - 查看 DOS 目录。

```
mdir [-w] msdos 目录
```

```
mdir [-f] [-w] [-a] msdos 文件 [msdos 文件列表]
```

w: 输出宽行文件列表(不包括时间信息)同 DOS 下的 dir/w。

a: 列出隐藏文件。

f: 不计算磁盘剩余空间。

mformat - 格式化 DOS 磁盘。

```
mformat [-t 磁道] [-h 磁头] [-s 扇区] [-i 卷标]
```

```
[-F] [-l 文件系统版本] [-S 大小代码] [-2 0 磁道上的扇区]
```

```

[-M 软件扇区大小] [-a] [-X] [-C]
[-H 隐藏扇区] [-r 根扇区] 驱动器:
mkmanifest - 创建一个恢复 UNIX 格式的文件名的 SHELL。
mkmanifest [文件名]
mlabel - 给 DOS 磁盘加卷标。
mlabel [-vcs] 驱动器:[新卷标]
c: 清除原有卷标。
s: 显示原有卷标。
mmd - 创建 DOS 目录。
mmd [-voOsSrRA] msdos 目录 [ msdos 目录列表 ]
mmove - 移动 DOS 下的文件。
mmove [-voOsSrRA] 源文件 目标文件
mmove [-voOsSrRA] 源文件 [ 源文件列表 ] 目标目录
mmount - mount DOS 磁盘。
mmount msdos 驱动器 [转载参数]
mtype - 显示 DOS 下的文件。
mtype [-ts] msdos 文件 [ msdos 文件列表 ]
t 查看文本文件。
s 略去高位字符。
mrd - 删除 DOS 下的目录。
mrd [-v] msdos 目录 [ msdos 目录列表 ]
mren - 将 DOS 下的文件改名。
mren [-voOsSrRA] 源文件 目标文件

```

#### 4.1.51 mesg

##### 1. 简介

mesg - 选择接收或者不接收其他用户发来的信息。

##### 2. 部分参数说明

```
mesg [ n ] [ y ]
```

此命令用来控制是否接受其他用户利用 talk 或 write 向用户发出的信息(缺省是接受), 如果用户不想被打扰的话(例如用户正在编辑文件时)使用 mesg n 就可以不接受别的用户向用户的终端发过来的信息, 不过这样有可能会错过别的用户发出的谈话请求, 如果用户想取消这一设置使用 mesg y 就可以恢复接受其他用户发来的信息, mesg 不跟参数则显示用户目前的设置情况。

##### 3. 范例

```

lark: ~ > mesg n
lark: ~ > mesg
is n

```

```
lark:~ >
```

#### 4.1.52 mkdir

##### 1. 简介

mkdir - 建立目录(同 DOS 下的 md)。

##### 2. 部分参数说明

```
mkdir [-p] [-m 权限] [--parents] [--mode = 权限] [--help]
      [--version] 目录列表
```

m, --mode 权限：给予建立的目录以设定的权限(缺省为 drwxr-xr-x)。

p, --parents：确定输入的每一层目录都存在, 建立所有不存在的目录, 例如 mkdir -p test/book, 如果 test 目录不存在则建立 test 目录。

help：在标准输出上输出帮助信息并退出。

version：在标准输出上输出版本信息并退出。

#### 4.1.53 mknod

##### 1. 简介

mknod - 建立特殊文件。

##### 2. 部分参数说明

```
mknod [选项] 文件名 {bcu} major minor
```

```
mknod [选项] 文件名 p
```

Options:

```
[-m 权限] [--mode = 权限] [--help] [--version]
```

mknod 用来建立 FIFO(管道文件)文件, 特殊字符文件, 特殊块文件。

p 建立 FIFO 文件。

b 建立块文件(有缓存)。

c, u 建立字符文件(无缓存)。

**注意:** 当建立块文件和字符文件时必须给出设备的 major 和 minor 数, mknod 常常被用来建立设备文件(/dev 目录下都是设备文件)。

-m, --mode 权限：给予建立的文件以设定的权限(缺省为 0666)。

--help：在标准输出上输出帮助信息并退出。

--version：在标准输出上输出版本信息并退出。

##### 3. 范例

```
lark:~ > mknod tt p
```

```
lark:~ > ls -la tt
```

```
prw-r--r- 1 fangh users 0 Sep 2 20:33 tt
```

```
lark:~ >
```

#### 4.1.54 more

##### 1. 简介

more - 在终端上按页观看文件的过滤器。

more 的功能没有 less 那么强大,而且 less 还提供了对 more 的模拟,不过一般用户可能更习惯于使用同 DOS 环境下相似的 more。

##### 2. 部分参数说明

- more [-dlfpsu] [-num] [+ / 要搜寻的字符串] [+ 行数] [文件列表]
- num : 设定屏幕的大小(行数)。
- d : more 给出提示 “[Press space to continue, 'q' to quit.]” 在输入错误的命令时给出 “[Press 'h' for instructions.]”来代替响铃。
- l : 通常 more 将^L 看成是一个特殊字符,如果文件中含有这个字符则 more 会在一行停下来,此参数屏蔽这一特性。
- f : 让 more 逻辑计算文件的行数,缺省是按显示在屏幕上的行数来计算的。
- p : 不滚屏,而是在清屏后显示下一屏。
- c : 不滚屏,而是从上到下显示下一屏。
- s : 将多个连续的空行转化成一行。
- u : 屏蔽下划线。
- + / : 设定要搜寻的字符串。
- + 行数 : 从文件的第几行开始显示。

##### 3. 范例

显示文件/etc/group 并搜寻字符串 bbs。

```
lark:~> more +/bbs /etc/group

...skipping
users:100:games
nogroup:-2:
bbs:x:99:bbs,bbsroot,bbsuser
lark:~>
```

#### 4.1.55 mv

##### 1. 简介

mv - 将文件改名。

##### 2. 部分参数说明

- mv [参数] 源文件 目标文件
- mv [参数] 源文件列表(支持通配符) 目标目录
- Options:

```
[ -bfuv ] [-S backup-suffix] [-V {numbered,existing,simple}]
[ -b ] [ -f ] [ -i ] [ -u ] [ -v ] [ -h ] [ -v ]
[ -b ] [ -f ] [ -i ] [ -u ] [ -v ] [ -h ] [ -v ]
[ -b ] [ -f ] [ -i ] [ -u ] [ -v ] [ -h ] [ -v ]
```

mv 将文件从一个文件名更改到另一个文件名,或者将一批文件从一个目录移动到另外一个目录。

**注意:**在不同的文件系统间移动文件时,只能移动普通文件。

- b, --backup : 为要移动的文件制作备份。
- f, --force : 强制覆盖已有的文件。
- i, --interactive : 在覆盖已有文件前给出提示让用户选择是否覆盖。
- u, --update : 在目标文件的时间比原文件新时不覆盖目标文件。
- v, --verbose : 在移动每个文件时输出相应信息。
- help : 在标准输出上输出帮助信息并退出。
- version : 在标准输出上输出版本信息并退出。

### 3. 范例

```
lark:~ > mv -v sourcefile destfile
sourcefile -> destfile
lark:~ >
```

## 4.1.56 newgrp

### 1. 简介

newgrp - 改变用户的组 ID。

sg - 以用户的另一个组 ID 运行命令。

### 2. 部分参数说明

```
newgrp [-] [组]
sg [-] [组 [-c 命令]]
```

newgrp 这条命令用来在用户登录后改变用户的组 ID,如果不跟“-”参数,则所有环境设置将被继承下来,反之则重新初始化,使用这条命令时如果不跟参数,则将用户的组改变为/etc/passwd 文件中的缺省组,在输入命令后会提示用户输入密码。如果用户不是要改变到的组的成员或该组的密码部分为空,则该用户将被屏蔽。sg 命令同 newgrp 类似,但是该命令不改变用户的 shell。在运行完给定的命令后退回原先的组 ID 和 shell。

### 3. 范例

改变用户的组 ID 到 sys

```
bash # newgrp sys
# exit
exit
bash #
```

#### 4.1.57 nice

##### 1. 简介

nice - 以设定的优先权来运行程序。

##### 2. 部分参数说明

```
nice [-n adjustment] [-s adjustment] [--adjustment=adjustment]
      [--help] [--version] [命令 [参数列表]]
```

-n adjustment, -adjustment, --adjustment=adjustment : adjustment 为要调整的新优先权,缺省时为 10,最高为-20(优先权最高),最低为 19(优先级最低)。

--help : 在标准输出上输出帮助信息并退出。

--version : 在标准输出上输出版本信息并退出。

##### 3. 范例

以优先权利-19 运行程序 mpg123

```
lark:~> nice -19 /usr/local/bin/mpg123 music1.mp3
```

#### 4.1.58 passwd

##### 1. 简介

passwd - 设置用户的密码。

##### 2. 部分参数说明

```
passwd [-fl-s] [用户名]
passwd [-g] [-r|R] 组名
passwd [-x max] [-n min] [-w warn] [-i inact] 用户名
passwd [-ll-ul-dl-S] 用户名
```

用户可以用 passwd 这个命令更改自己的登录密码,一般用户只能更改他自己的密码,超级用户可以更改其他所有用户的密码,超级用户和组的管理者可以更改组的密码,还可以用这个命令来更改用户的其他信息,如用户的全名、用户的登录 shell、用户的密码失效的时间间隔等等。用户首先被提示输入旧密码,程序接受这个密码并将其加密后同文件/etc/passwd(或/etc/shadow 如果有 shadow 的话)中存储的旧密码进行比较,用户只有一次机会输入正确的密码,但是超级用户不必输入,因为这样才可以替那些忘记密码的用户更改密码。

用户的密码可以由大小写字母和阿拉伯数字及标点符号组成,最长 8 位,最短 5 位。如果用户输入的密码符合条件,则 passwd 提示用户再输入一遍以便确认,在两次输入密码相互吻合之后就将用户的密码改变。如果使用选项-g,则是让超级用户和组的管理者更改组的密码,如果不加组名则使用当前组。在使用-g 选项的同时使用-r 选项则解除组的密码,使得任何一个用户都可以具有访问这个组的权力,反之使用-R 选项则让这个组对任何用户都是不可访问的。超级用户可以用选项-x, -n, -w, -i 选项来设定用户密码的

使用期限。-x 选项用来设定用户密码有效的最长期限,在 max 天后用户需要改变密码,而-n 选项则设定用户只有在 min 天之后才能改变密码,-w 选项用来设置警告用户的天数,-i 选项用来设置在 inact 天后将用户的帐号屏蔽掉,在 inact 天后用户将不能登录入自己的帐号。使用选项-l 可以屏蔽一个用户的帐号(将用户的密码改成一个不可能解密的值),使用选项则恢复这个用户的帐号,密码被更改到上一次的值,即-l 选项的逆过程。使用-S 选项则输出用户帐号的状态,输出的信息由下列部分组成:第一部分是表明用户的帐号是屏蔽(L)、无密码(NP)及有可使用的密码(P),第二部分是最后一次改变密码的时间。下面 4 个部分分别是最小使用密码时间,最长使用密码时间、警告时间和无效时间。即上文所述-n -x -w -i 选项所设置的时间。

下面向读者介绍一下密码设置应注意的一些问题,对于 UNIX 系统来说,其安全性同每个用户的密码都有关系,一个设置得很好的密码会让攻击系统的黑客花费很长的时间而一无所得,而一个简单的密码会让一个精心设置的系统的安全毁于一旦。因而仔细地挑选密码对于每个用户来说都是一件很重要的工作。下面给出设置密码的一些建议:不要使用自己或家人名字的汉语拼音,生日、门牌号码电话号码来组成密码,也不要使用一个字典中能查得到的英文单词作为密码,这些都属于容易攻击的密码范围,也不要将密码记在一个别人看得见的地方。另外为了使密码易记,可以使用下面的方法,以一句英文句子的第一个字母或最后一个字母来作为密码,例如:

Do you still love me tommorrow ?

密码为 Dyslmt?

这就是一个黑客们不大可能加入字典的一个密码,其安全性显然较好。另外密码中最好包含数字和大小写,这样安全性会有所提高。

### 3. 范例

#### 更改密码

```
lark:~ > passwd
Changing password for fangh
Old password:oldpass          密码并不显示出来
Enter the new password (minimum of 5, maximum of 8 characters)
Please use a combination of upper and lower case letters and numbers.
New password:newpass
Re-enter new password:newpass
Password changed.
lark:~ > passwd -S          显示用户情况
fangh P 09/05/97 0 99999 7 -1
lark:~ >
```

## 4.1.59 paste

### 1. 简介

paste - 将不同文件的行连接起来。

## 2. 部分参数说明

```
paste [-s] [-d delim-list] [--serial] [--delimiters = delim-  
list] [--help] [--version] [文件列表...]
```

-s, --serial : 一次只从一个文件中取一行,而不是从每个文件中取一行。

-d, --delimiters delim-list : 使用 delim-list 中的符号来代替缺省的 TAB 键为行分割符。

--help : 在标准输出上输出帮助信息并退出。

--version : 在标准输出上输出版本信息并退出。

缺省的 paste 将输出到标准输出上。

## 3. 范例

将文件 source1 source2 输出到文件 dest 中。

```
lask~> paste source1 source2 > dest
```

## 4.1.60 pr

### 1. 简介

pr - 处理文件以便打印,输出到标准输出。

### 2. 部分参数说明

```
pr [+ PAGE] [-COLUMN] [-abcdffmrtv] [-e[in-tab-char[in-tab-  
width]] [-h header] [-i[out-tab-char[out-tab-width]]] [-l  
page-length] [-n[number-separator[digits]]] [-o left-mar-  
gin] [-s[column-separator]] [-w page-width] [--help]  
[--version] [文件列表...]
```

+ PAGE : 从第 PAGE 页开始打印。

-COLUMN : 输出 COLUMN 栏。

-e : 以 “^G” 这种形式来输出控制符。

-d : 将输出的每一行后加一新的空白行。

-e[in-tab-char[in-tab-width]] 将 TAB 扩展为空格, in-tab-char 为设定的 TAB 符(缺省就是 TAB), in-tab-width, 为每个 TAB 扩展为几个空格, 缺省为 8 个。

-f, -F : 在页面使用换页符, 缺省使用新行填满剩余部分。

-h header : 用 header 字符串来代替缺省的文件名作为输出的标题。

-i[out-tab-char[out-tab-width]] : 将空格缩写成 TAB, out-tab-char 为设定的 TAB 符(缺省就是 TAB), out-tab-width 为几个空格缩写为 TAB, 缺省为 8 个。

-l page-length : 设置输出的每页有 page-length 行。

-m : 将所有文件同时并行输出到几个栏位内。

-n[number-separator[digits]] : 输出行号, number-separator 为行号后的字符。缺省为 TAB, digits 为行号的位数, 缺省为 5。

-o left-margin : 设定左边界为 left-margin 个空格。

-r : 打不开文件时不输出错误信息。

-s[column-separator] : 用单个字符 column-separator 分栏,缺省为 TAB。

-w page-width : 设定页宽为 page-width 列,缺省为 72。

--help : 在标准输出上输出帮助信息并退出。

--version : 在标准输出上输出版本信息并退出。

### 3. 范例

对文件 destfile 进行处理

```
lark:~ > pr -f destfile -n
```

|   |                          | destfile                | Page | 1 |
|---|--------------------------|-------------------------|------|---|
| 1 | total 9                  |                         |      |   |
| 2 | -rw-r--r-- 1 fangh users | 172 Sep 6 17:09 aaa     |      |   |
| 3 | -rw-r--r-- 1 fangh users | 0 Sep 8 21:15 destfile  |      |   |
| 4 | -rw-r--r-- 1 fangh users | 99 Sep 6 21:08 destfile |      |   |
| 5 | -rw-r--r-- 1 fangh users | 28 Sep 7 14:28 list     |      |   |
| 6 | drwx----- 2 fangh users  | 1024 Sep 6 17:06 mail/  |      |   |
| 7 | -rw----- 1 fangh users   | 3652 Sep 8 15:45 mbox   |      |   |
| 8 | drwxr-xr-x 6 fangh users | 1024 Sep 6 16:57 mp3/   |      |   |

```
lark:~ >
```

## 4.1.61 ps

### 1. 简介

ps - 查看进程状态。

### 2. 部分参数说明

```
ps [-] [lujsvmaxScwehnrn] [ttx] [0[+|-]k1[[+|-]k2...]
--sortX[+|-]key[.[+|-]key[,...]] --help --version [pids]
```

-l : 长列表。

-u : 用户格式——给出用户名和开始的时间。

-j : 作业格式——给出 pgid,sid。

-s : 信号格式。

-v : 虚拟存储器格式。

-m : 显示内存信息。

-f : 给出树状格式。

-a : 显示其他用户的进程。

-x : 显示没有控制台的进程。

-c : 从 task\_struct 中给出进程名。

-e : 显示环境。

- w : 宽行输出, 不自动换行。
- h : 无标题。
- r : 只显示正在运行的程序。
- n : 用数字来输出 USER 和 WCHAN。
- txx: 只显示受 tty.xx 控制的进程。
- pids : 只显示特定的进程。
- help : 在标准输出上输出帮助信息并退出。
- version : 在标准输出上输出版本信息并退出。

下面解释一下各栏的意义:

- FLAGS : 长格式的 F 域。
- UID : 用户的 ID。
- PID : 进程的 ID。
- PPID : 父进程的 ID。
- PRI : 进程优先级。
- NI : 标准 UNIX 的优先级。
- SIZE : 虚拟内存的大小。
- RSS : 驻留空间的大小。
- WCHAN : 进程等待的内核事件。
- STAT : 进程状态——代码如下。
- R : 正在运行。
- S : 睡眠。
- D : 不可打断的睡眠。
- T : 停止或跟踪。
- Z : 僵尸进程。
- W : 进程没有驻留页。
- N : 进程有一负 nice 值。
- TTY : 进程的控制台。
- PAGEIN : 主内存页失败的数目(页失败造成页从磁盘或 CACHE 中读取)。
- TRS : 文本驻留大小。
- SHARE : 共享内存。
- SWAP : 交换设备上的 K 字节数。

### 3. 范例

显示当前进程:

```
lark:~ > ps -c
      PID TTY STAT  TIME COMMAND
 8724 p4 S    0:00 tesh
 8876 p6 R    0:00 ps
lark:~ >
```

#### 4.1.62 pwd

##### 1. 简介

pwd - 显示正在工作或当前的目录名。

## 2. 部分参数说明

`pwd [-help, -version]`

`--help` : 在标准输出上输出帮助信息并退出。

`--version` : 在标准输出上输出版本信息并退出。

**注意:** `pwd` 输出的是真实路径名, 它不会输出符号连接的目录名。

大多数 UNIX shell 内建 `pwd` 命令, 这样用户运行的就有可能不是 `/bin/pwd` 这个命令。

## 3. 范例

列出当前工作目录

```
lark: ~ > pwd
/home/fangh
lark: ~ >
```

### 4.1.63 quota

#### 1. 简介

`quota` - 显示磁盘使用情况和限制情况。

## 2. 部分参数说明

`quota [-guv | q]`

`quota [-uv | q]` 用户名

`quota [-gv | q]` 组名

`-g` : 显示用户所在组的磁盘限制。

`-u` : 显示用户的磁盘限制(缺省设置)。

`-v` : 显示没有分配空间的文件系统的分配情况。

`-q` : 显示简洁的信息, 除了超过限额的情况一般不显示。

**注意:** Linux 缺省是不设置 `quota` 的, 如果用户要设置 `quota` 请阅系统管理部分。

## 3. 范例

```
lark: ~ > quota
Disk quotas for user fangh (uid 10000): none
lark: ~ >
```

### 4.1.64 rcp

#### 1. 简介

`rcp` - 在计算机间拷贝文件。

## 2. 部分参数说明

`rcp [-px] [-k realm] 源文件名 目标文件名`

- rep [-px] [-r] [-k realm] 源文件列表 目标目录
- r : 如果源文件名是目录,则拷贝所有子目录,目标文件必须也是目录。
  - p : 尽量保持文件的修改时间和访问权限。
  - k : 要求 rep 获得外部许可。
  - x : 给拷贝的数据流用 DES 方法加密。

**注意:**要成功地拷贝文件,用户必须在目标机器上具有运行 rsh 的权利。参见 rsh 和 rlogin。

### 3. 范例

从机器 siskin 上拷贝文件:

```
lark ~ > rep siskin:/home/fangh/mbox ./mbox
lark ~ >
```

## 4.1.65 reset

### 1. 简介

reset - 将终端复位。

在使用终端的过程中,有时会发现屏幕的字符“花”掉了,这时使用 reset 就可以恢复,例如在运行完电路分析软件 PSPICE 后,就会发现屏幕出现错乱字符,使用 reset 就可恢复正常。reset 调用 tput 函数,向终端发出复位信号。如果屏幕被 Ctrl+S 锁住了,可以用 Ctrl+Q 来解除锁定。

### 2. 范例

将终端复位:

```
lark: ~ > reset
```

## 4.1.66 rm

### 1. 简介

rm - 删除文件或目录。

### 2. 部分参数说明

- ```
rm [-dfrvR] [-directory] [-force] [-interactive]
[-recursive] [-help] [-version] [-verbose] 文件名列表...
```
- d, -directory : 本选项将用 unlink 来替代 rmdir 删除目录,只有超级用户有权使用。这种删除方法不用使目录为空就可删除,但本选项在目录非空时会造成目录中的文件簇丢失,在执行这个命令后可以用 fsck 来修复文件系统。
  - f, -force : 忽略不存在的文件,并且不给用户任何提示。
  - i, -interactive : 在删除每个文件时提示用户选择 “y” 或 “n”。
  - r, -R, -recursive : 将整个目录连带所有子目录删除。
  - v, -verbose : 删除每个文件时输出文件信息。

—help：在标准输出上输出帮助信息并退出。

—version：在标准输出上输出版本信息并退出。

**注意：**在 Linux 下如果产生类似于“-f”这种文件名的文件，删除是先跟“-”选项。这表明后面所跟的“-f”不是选项而是文件名，例如“rm -file”是删除文件“-file”，在删除文件名中包含特殊字符的文件时，可以用“\ + 特殊字符”，或用“”将文件名括起来，例如要删除文件“this is a test”可以用下列命令：

```
rm this \ is \ a \ test
```

或者 rm “this is a test”。

**特别注意：**使用 rm -rf \* 这个命令，如果用户是超级用户，并且在 root 下使用此命令，那么系统的所有文件将被删除。而在 UNIX 下恢复文件几乎是一件不可能的事情。因而要谨慎对待这个命令。另外建议用户将 rm 定义为“rm -i”，并加入到用户的登录文件中。

### 3. 范例

删除文件 destfile。

```
lark: ~ > rm -v destfile
```

```
destfile
```

```
lark: ~ >
```

## 4.1.67 rmdir

### 1. 简介

rmdir - 删除空的目录。

### 2. 部分参数说明

rmdir [-p] [--parents] [--help] [--version] 目录列表...

-p --parents：例如 rm -p dir1/dir2，如果删除 dir2 后 dir1 已经为空目录，则删除目录 dir1。

—help：在标准输出上输出帮助信息并退出。

—version：在标准输出上输出版本信息并退出。

### 3. 范例

删除目录 test。

```
lark: ~ > rmdir test
```

```
lark: ~ >
```

## 4.1.68 set

### 1. 简介

set - 显示或设置环境变量。

unset - 删除环境变量。

## 2. 部分参数说明

set 变量名 = 变量值

如不跟参数则显示当前设置:

unset 变量名

这两个命令都是 shell 内建的命令。

## 3. 范例

lark: ~ > set term = vt100            设置变量 term 为 vt100

lark: ~ > unset term                删除变量 term

### 4.1.69 sh

#### 1. 简介

见 bash。

### 4.1.70 sleep

#### 1. 简介

sleep : 延迟设定的时间。

## 2. 部分参数说明

sleep [-help] [-version] number[smhd]

sleep 延迟由 number + smhd 给出的时间。

smhd 的含义是:s 秒;m 分钟;h 小时;d 天。

-help : 在标准输出上输出帮助信息并退出。

-version : 在标准输出上输出版本信息并退出。

## 3. 范例

lark: ~ > sleep 1m        等待 1 分钟

lark: ~ >

### 4.1.71 sort

#### 1. 简介

sort : 将文本文件排序。

## 2. 部分参数说明

sort [-cmus] [-t 间隔符] [-o 输出文件名] [-T 临时目录]

[-bdfiMnr] [+ POS1 [-POS2]] [-k POS1[,POS2]] [文件名列表...]

sort {-help, -version}

sort 有三种使用方式:缺省的是将文本文件排序,还可以合并文件。

-c : 检查文件是否已经被排序,如果没有则给出错误信息并退出。

- m : 将文件列表中的文件作为一个组来派序,并合并起来,每个文件必须先各自排序。
- b : 排序时忽略行首的空白。
- d : 按电话本的目录方式排序,忽略除了字母、数字和空白以外的所有字符。
- f : 将小写字母当作大写字母来排序,例如“b”作为“B”来排序。
- i : 忽略 ASCII 字符内码不在范围 040 ~ 0176 之间的字符。
- M : 将每一行开头如果是月份的英文缩写的前三个字母的话就按月份大小排序。
- n : 按数字大小而不是字母顺序来排序。
- r : 反向排序。
- o 输出文件名 : 将输出重定向到文件中取代缺省的标准输出。
- t 间隔符 : 用设定的间隔符取代缺省的空格符。
- + POS1 [-POS2] : 用每一行的第 POS1 列到第 POS2 列来排序,如不给出 POS2 则到行尾。
- k POS1[,POS2]; 同上。
- help : 在标准输出上输出帮助信息并退出。
- version : 在标准输出上输出版本信息并退出。

### 3. 范例

列出文件 testfile 的内容:

```
lark:~ > cat testfile
```

```
231
this is a test
1231
do you know
321
let's go
```

```
lark:~ > sort testfile
```

将文件 testfile 排序

```
1231
231
321
do you know
let's go
```

```
this is a test
```

```
lark:~ > sort -n testfile
```

将文件 testfile 按数字大小排序

```
231
321
1231
do you know
let's go
this is a test
```

```
lark~ >
```

#### 4.1.72 split

##### 1. 简介

split - 分割文件。

##### 2. 部分参数说明

```
split [-lines] [-l lines] [-b bytes[bkm]] [-C bytes[bkm]]  
[—lines = lines] [—bytes = bytes[bkm]] [—line-  
bytes = bytes[bkm]] [—help] [—version] [输入文件 [输出文件]]
```

输出文件名为输出文件的名称,程序自动在其后加上 aa,ab,ac,...,缺省时为 x。

-lines, -l lines, —lines = lines :从输入文件中读出 lines 行,然后写入一个新的输出文件。

-b bytes[bkm], —bytes = bytes[bkm] :从输入文件中读出 bytes(b - 512 比特, k - 1 K, m - 1 兆)大小的数据,然后写入一个新的输出文件。

-C bytes[bkm], —line-bytes = bytes[bkm] :写入每个输出文件一行的大小被限制为 bytes 大小。

—help : 在标准输出上输出帮助信息并退出。

—version : 在标准输出上输出版本信息并退出。

##### 3. 范例

分割文件 tesh.txt。

```
lark: ~ > split tesh.txt tesh.txt.  
lark: ~ > ls -la tesh *  
-rw-r--r-- 1 fangh users 252409 Sep 19 13:30 tesh.txt  
-rw-r--r-- 1 fangh users 48074 Sep 19 13:30 tesh.txt.aa  
-rw-r--r-- 1 fangh users 46000 Sep 19 13:30 tesh.txt.ab  
-rw-r--r-- 1 fangh users 46097 Sep 19 13:30 tesh.txt.ac  
-rw-r--r-- 1 fangh users 47867 Sep 19 13:30 tesh.txt.ad  
-rw-r--r-- 1 fangh users 43427 Sep 19 13:30 tesh.txt.ae  
-rw-r--r-- 1 fangh users 20944 Sep 19 13:30 tesh.txt.af  
lark: ~ >
```

#### 4.1.73 stty

##### 1. 简介

stty - 设置或显示终端行数的设置情况。

##### 2. 部分参数说明

```
stty [设置...]  
stty [-a,-all,-g,-help,-save,-version]
```

stty 如果不跟参数,则输出当前的终端设置情况。如跟参数则用“-”表示关闭该设

定。

控制设定：

[-]parenb：输出时产生奇偶校验位，并且必须在输入时也带奇偶校验位。

[-]parodd：设置为奇校验。

cs5 cs6 cs7 cs8：设置字符大小为 5、6、7 或 8 比特。

[-]hupcl [-]hup：当本终端最后一个进程结束时送出 hangup 信号。

[-]estopb：每个字符用两个停止位，加“-”号时为一个。

[-]cread：允许接受输入。

[-]clocal：禁止调制解调器控制信号。

[-]crtsets (np)：使能 RTS/CTS 握手信号。

输入设定：

[-]ignbrk：忽略 break 字符。

[-]brkint：break 字符产生一个中断。

[-]ignpar：忽略奇偶校验错的字符。

[-]parmrk：标志奇偶校验错的字符。

[-]inpek：使能对输入的奇偶校验检查。

[-]istrip：对输入字符的高八位清零。

[-]inlcr：将换行符转换为回车符。

[-]igncr：忽略回车。

[-]icrnl：将回车符转换为换行符。

[-]ixon：使能 XON/XOFF 流控制。

[-]ixoff [-]tandem：如果输入缓冲满则输出停止字符，当输入缓冲空时输出开始字符。

[-]iucL (np)：将大写字符转换为小写字符。

[-]ixany (np)：允许任何字符重置输出。

[-]imaxbel (np)：如果新的字符来到时输入缓冲满则响铃，但是并不清缓冲。

[-]olcuc (np)：将小写字母转变为大写字母。

[-]oocrnl (np)：将回车符转变为新的换行符。

[-]onlcr (np)：将换行符转变为回车加换行符。

[-]onoer (np)：不在第一列输出回车。

[-]ofill (np)：用填充字符来代替时间延迟。

[-]ofdel (np)：使用删除符代替空白符为填充时使用的字符。

nll nl0 (np)：换行延迟类型。

cr3 cr2 cr1 cr0 (np)：回车延迟类型。

tab3 tab2 tab1 tab0 (np)：水平制表延迟类型。

bs1 bs0 (np)：退格延迟类型。

vt1 vt0 (np)：垂直制表延迟类型。

ff1 ff0 (np)：表格延迟类型。

本地设置:

- `[-]isig`: 使能中断,退出,挂起等特殊字符。
- `[-]icanon`: 使能删除,杀死等特殊字符。
- `[-]ixtext`: 使能非 POSIX 标准的特殊字符。
- `[-]echo`: 输入回显。
- `[-]echoe`, `[-]crterase`: 将删除符回显为退格-空格-退格。
- `[-]echok`: 在杀死符后回显新行。
- `[-]echonl`: 没有回显其他字符时也回显新行。
- `[-]noflsh`: 在中断和退出等特殊字符后禁止清空缓冲。
- `[-]xcase (np)`: 在设置了 `icanon` 的情况下用“\”加小写字母来输入和显示相应的大写字母。
- `[-]tostop (np)`: 停止在后台的作业向终端输出。
- `[-]echoprt [-]prterase (np)`: 在符号‘\’和‘/’之间将删除符回显为后退符。
- `[-]echoctl [-]ctlecho (np)`: 用‘c’的形式来输出控制符。
- `[-]evenp [-]parity`: 同 `parenb -parodd cs7`。加‘-’后,同 `-parenb cs8`。
- `[-]oddp` 同 `parenb parodd cs7`。加‘-’后,同 `-parenb cs8`。
- `[-]nl` 同 `icrnl -onlcr`。加‘-’后,同 `icrnl`。

合并设定:

- `ek`: 将删除和杀死符重置到缺省值。
- `sane`: 将所有设置重置到缺省值。
- `[-]pass8`: 同 `-parenb -istrip cs8`。加‘-’后,同 `parenb istrip cs7`。
- `[-]litout`: 同 `-parenb -istrip -opost cs8`。加‘-’后,同 `parenb istrip opost cs7`。

特殊字符:

- `intr`: 送出中断信号。
- `quit`: 送出退出信号。
- `erase`: 删除输入的最后一个字符。
- `kill`: 删除当前行。
- `eof`: 送出文件结束符。
- `eol`: 行结束符。
- `eol2 (np)`: 交替的行结束符。
- `switch (np)`: 不同 SHELL 之间切换。
- `start`: 重新开始输出。
- `stop`: 停止输出。
- `susp`: 送出终端停止符。
- `dsusp (np)`: 在清空输入缓冲后送出终端停止符。
- `rprnl (np)`: 重绘当前行。
- `werase (np)`: 删除输入的最后一个单词。

特殊设定:

- `min N`: 当设定 `-icanon` 后,在特定时间内最少要输入 N 个字符。

time N : 设定上一选项的特定时间。

ispeed N : 设定输入速度为 N

ospeed N : 设定输出速度为 N。

rows N (np) : 通知系统核心终端有 N 列。

cols N columns N (np) : 通知系统核心终端有 N 行。

size (np) : 输出系统核心已知的终端的行列数。

line N (np) : 使用行设定 N。

speed 输出终端的速度。

N : 设定输入和输出的速度为 N, N 可以为:

0 50 75 110 134 134.5 150 200 300 600 1200 1800 2400 4800 9600 19200 38400 exta

extb。exta 同 19200; extb 同 38400。如果设定了 0 又设定了 -clocal 则终端将被挂起。

选项:

-a, --all : 输出所有设置。

-g, --save : 输出所有设置, 输出格式为另一个 stty 可以恢复当前设置的格式。

-help : 在标准输出上输出帮助信息并退出。

--version : 在标准输出上输出版本信息并退出。

### 3. 范例

输出当前设置:

```
lark:~ > stty -a
speed 9600 baud; rows 0; columns 0; line = 0;
intr = ^C; quit = ^\ ; erase = ^?; kill = ^U; eof = ^D; eol = <undef>;
eol2 = <undef>; start = ^Q; stop = ^S; susp = ^Z; rprnt = ^R; werase = ^W;
lnext = ^V; flush = ^O; min = 1; time = 0;
-parenb -parodd cs8 -hupcl -cstoph cread -clocal -crtscts
-ignbrk -brkint -ignpar -parmrk -inpcr -istrip -inlcr -igncr icrnl ixon
-ixoff
-iuclc -ixany -imaxbel
opost -olcuc -ocrn1 onlcr -onocr -onlret -ofill -ofdel nl0 cr0 tab0 bs0
vt0 ffo
isig icanon ixten echo echoe echok -echonl -noflsh -xcase -tostop
-echoprnt
echoctl echoke
lark:~ >
```

## 4.1.74 sync

### 1. 简介

sync - 清空 Linux 文件系统的缓冲区。

在系统的 CPU 停止工作以前必须调用该命令来将系统的缓冲区写入硬盘。  
REBOOT 和 HAlt 指令都调用该命令来清空缓冲区。Linux 的优点之一就是磁盘存取速

度快,这是因为它将不用的内存都拿来作缓冲,这带来的问题就是在系统关机前必须将缓冲清空,否则就会丢失信息和数据。请看 Linux 本人是如何说的:“sync 这个命令只保证在一定时间内将数据写入硬盘,这需要一定的时间写完所有的数据,如果用户在执行这条命令后马上关机的话,请稍等片刻。”(但是 REBOOT 和 Halt 命令会代替用户作这些工作。)

update - 周期性地清空 Linux 文件系统的缓冲区

它有两种工作模式,一种是缺省地每 5 秒种写一些块(不调用 sync),一种是每 30 秒调用一次 sync。在系统启动时 update 作业应该越早运行越好,在 SLACKWARE 版本中在/etc/rc.d/rc.S 中的第二条命令就是运行 update。它的进程号一般是 13。

## 2. 部分参数说明

- update [ 选项 ]
- S : 只使用 sync 工作。
- s 秒: 设定调用 sync 的时间间隔,缺省是 30 秒。
- f 秒: 设定缺省工作方式的时间间隔,缺省是 5 秒。

### 4.1.75 su

#### 1. 简介

su - 改变用户的 ID 或成为超级用户。

作者为 Julianne Frances Haugh (jfh@tab.com)。

su 可让用户在一个登录的 shell 中不用退出就可以变成另外一个用户,如果后面不跟参数,则 su 缺省地变成超级用户。执行 su 命令时,程序会让用户输入密码,如果密码不正确,则程序将给出错误信息后退出,所有 su 的动作都会被系统记录在一个 log 文件中,以便探测谁在恶意攻击系统。su 之后,当前的所有环境变量都会被传递过去,但是路径值 \$PATH 将被重置为/bin:/usr/bin(对一般用户)或/sbin:/bin:/usr/sbin:/usr/bin(对超级用户),这两个缺省值可以通过改变文件/etc/login.defs 中的 ENV\_PATH 值和 ENV\_SUPATH 的值来改变。

## 2. 部分参数说明

- su [-] [用户名 [参数]]
- su 后面可以跟参数,例如 su 用户名 -c 命令,就会在改变用户 ID 后执行相应的命令并退出。

## 3. 范例

变更为用户 fangh。

```
lark: ~ > su fangh
Password:
> whoami
fangh
```

#### 4.1.76 sudo

##### 1. 简介

`sudo` - 以超级用户的身份执行某些命令。

`visudo` - 编辑可以使用 `sudo` 命令的用户列表。

在系统管理中,有时需要给某些用户的部分系统一些特权以便执行某些系统命令。例如 `shutdown`, `halt`, `lpc` 等命令,但是系统管理员又不愿意给予他们全部的系统特权,这时 `sudo` 命令就非常有用了。`Sudo` 允许经过同意的用户以超级用户的身分执行指令。`Sudo` 用 `/etc/sudoers` 这个文件来判定谁是被授权的用户。`Sudo` 将会提示用户输入密码开始一段 `N` 分钟的可以使用的时间(其中 `N` 是在安装的时候定义的且缺省值为 5 分钟)结束后自动退出。

`Sudoers` 这个文件是由一个选择性的主机别名(`host alias`)域,一个选择性的指令别名(`command alias`)域以及用户说明(`user specification`)域所组成的。所有的指令别名或主机别名必须以他们自己的关键字作为开始(`Host_Alias/Cmnd_Alias`)。

**注意:**只有第一次使用的用户(在用户说明域里有记录的用户)使用时会有说明。

用户说明域格式:

用户 有权组 [ ; 有权组 ] ...

有权组 ::= 主机类别 = [ `op` ] 指令类别 [ , [ `op` ] 指令类别 ] ...

主机类别 ::= 小写的主机名或主机别名。

指令类别 ::= 指令或指令别名。

`op` ::= '!' 否定符。

主机别名域格式:

`Host_Alias` 主机别名 = 主机列表

`Host_Alias` ::= 一个关键字。

主机别名 ::= 一个大写的别名。

主机列表 ::= 以逗号间隔的一些主机名

指令别名域格式:

`Cmnd_Alias` 指令别名 = 指令列表

`Cmnd_Alias` ::= 一个关键字。

指令别名 ::= 一个大写的别名。

指令列表 ::= 以逗号间隔的一些指令。

在 '#' 符号后面是注解。

太长的行可以使用 '\ ' 来分成新的行。保留的别名 'ALL' 在 {`Host`, `Cmnd`}\_Alias' 里都可以使用。不要使用关键字 'ALL' 来定义一个别名,这个别名无效。注意到 'ALL' 代表全部的主机跟指令。用户可以使用这个语法从所有定义中去掉一些项目;

```
user host = ALL, ! ALIAS1, ! /etc/halt...
```

`sudoers` 文件的范例文件:

```
# Host alias specification

Host_Alias HUB = houdini.rootgroup.com: \
    REMOTE = merlin,kodiakthorn,spirit
Host_Alias MACHINES = kalkan,alpo,milkbones
Host_Alias SERVERS = houdini,merlin,kodiakthorn,spirit

# Command alias specification

Cmdnd_Alias LPCS = /usr/etc/lpc,/usr/uecb/lprm
Cmdnd_Alias SHELLS = /bin/sh,/bin/csh,/bin/tcsh
Cmdnd_Alias MISC = /bin/rm,/bin/cat: \
    SHUTDOWN = /etc/halt,/etc/shutdown

# User specification

britt REMOTE = SHUTDOWN:ALL = LPCS
robb ALL = ALL,! SHELLS
nieusma SERVERS = SHUTDOWN,/etc/reboot: \
    HUB = ALL,! SHELLS
jill houdini.rootgroup.com = /etc/shutdown,MISC
markm HUB = ALL,! MISC,! /etc/shutdown,! /etc/halt
billp ALL = /usr/local/bin/top:MACHINES = SHELLS
davehieb merlin = ALL:SERVERS = /etc/halt: \
    kodiakthorn = ALL
```

上面的 `sudoers` 说明文件是由 4 个主机别名说明, 4 个指令别名说明以及 7 个用户说明所组成。Britt 被允许在远端机器 (merlin, kodiakthorn, 还有 spirit) 上执行 `/etc/halt`, `/etc/shutdown`, `/usr/etc/lpc` 以及 `/usr/uecb/lprm`。Rohn 被允许在任何机器上执行除 `SHELL` 指令以外的任何指令。Jill 被允许在机器 houdini 上执行 `/etc/shutdown`、`/bin/rm` 以及 `/bin/cat`。Davehieb 可以在机器 merlin 以及 kodiakthor 上执行任何指令并且可以 `halt` 服务器。

`Sudoers` 文件应该用 `visudo` 编辑, 它会锁住该文件并且检查语法, 以避免产生错误的文件。`Sudo` 被设计成用 4.3 BSD 的 `syslogging` 来记录, 但是也可以使用用户自己的 `log` 进程来记录。如果一个没有被授权的用户执行了 `sudo` 的话, 将会有一封 `mail` 从该用户处寄送到当地的管理员处 (Linux 缺省是 `root`)。所有的设定都是在安装时定义的由 `sudo`、`h` 及 `Makefile` 定义, 但是 Linux 缺省的是安装好的执行文件, 用户如果要更改的话, 需要重新编译与安装。

允许巢状的主机以及指令别名。

允许在 `sudoers` 文件中使用 `host specifier`

以便使用全局符号 (`user ALL,! SERVERS, ... = commands`)。

允许在 `sudores` 文件里的用户别名 (就如同主机/指令别名一样)。

## 2. 部分参数说明

`sudo` 命令

### 3. 范例

用 `sudo` 来 `halt` 机器

```
lark: ~ > sudo halt
```

## 4.1.77 tail

### 1. 简介

`tail` - 显示文件的最后一部分。

### 2. 部分参数说明

```
tail [-c [+ ]N[bkm]] [-n [+ ]N] [-fqv] [--bytes=[+ ]N[bkm]]
      [--lines=[+ ]N] [--follow] [--quiet] [--silent] [--verbose]
      [--help] [--version] [文件名列表]
```

`tail` 缺省显示文件列表中每个文件的最后十行, 如果没有文件名或文件名为 “-” 则其从标准输入中读取文件, 如果有多个文件则其会在文件前面加上 “= = > 文件名 < = =” 以便区分。

-c N, --bytes N : 显示文件后部的 N 比特大小的部分, N 后面可以跟 bkm 参数。

b : 512 比特的块。

k : 1 k 的块。

m : 1 M 的块。

-f, --follow : 如果文件大小在增长的话, `tail` 将跟随文件增长而显示。

-l, -n N, --lines N : 显示文件末尾的 N 行。

-v, --verbose : 一直输出 “= = > 文件名 < = =” 形式的文件名。

--help : 在标准输出上输出帮助信息并退出。

--version : 在标准输出上输出版本信息并退出。

### 3. 范例

显示文件 `/etc/DIR_COLORS` 的最后三行:

```
lark: ~ > tail -v -n 3 /etc/DIR_COLORS
= = > /etc/DIR_COLORS < = =
.xbm 01;35
.xpm 01;35
.tif 01;35
lark: ~ >
```

## 4.1.78 tar

### 1. 简介

`tar` - GNU 版的文件打包备份的工具。

### 2. 部分参数说明

```
tar [ - ] A --catenate --concatenate | c --create | d
```

```

--diff --compare ! r --append ! t --list ! u --update ! x
--extract --get [ --atime-preserve ] [ -b, --block-size N ]
[ -B, --read-full-blocks ] [ -C, --directory DIR ] [
--checkpoint      ] [ -f, --file [HOSTNAME:]F ] [ --force-
local      ] [ -F, --info-script F --new-volume-script F ] [
-G, --incremental ] [ -g, --listed-incremental F ] [ -h,
--dereference ] [ -i, --ignore-zeros ] [ --ignore-failed-
read ] [ -k, --keep-old-files ] [ -K, --starting-file F ]
[ -l, --one-file-system ] [ -L, --tape-length N ] [ -m,
--modification-time ] [ -M, --multi-volume ] [ -N,
--after-date DATE, --newer DATE ] [ -o, --old-archive,
--portability ] [ -O, --to-stdout ] [ -p, --same-permis-
sions, --preserve-permissions ] [ -P, --absolute-paths ] [
--preserve      ] [ -R, --record-number ] [ --remove-files
] [ -s, --same-order, --preserve-order ] [ --same-owner ]
[ -S, --sparse ] [ -T, --files-from F ] [ --null      ] [
--totals ] [ -v, --verbose ] [ -V, --label NAME ] [
--version ] [ -w, --interactive, --confirmation ] [ -W,
--verify ] [ --exclude FILE ] [ -X, --exclude-from FILE
] [ -Z, --compress, --uncompress ] [ -z, --gzip,
--ungzip ] [ --use-compress-program PROG ] [ --block-
compress ] [ --[0-7][lmh] ]

```

[文件名列表]

[目录名列表]

tar 是对一般用户最易用和最可靠的实用程序,从归档介质(可以是一个磁带机,也可以是普通文件)上保留或恢复文件,也能从多个文件归档中抽取单个文件。

-A, --catenate, --concatenate : 向归档文件中追加另一个 tar 文件。

-c, --create : 建立新的归档文件。

-d, --diff, --compare : 将归档文件和文件系统上的文件作比较。

--delete : 从归档文件中删除文件(不能够用在磁带上)。

-r, --append : 向归档文件末尾追加文件。

-t, --list : 对归档文件列目录。

-u, --update : 只追加比归档文件中的文件更新的文件。

-x, --extract, --get : 从归档文件中解出文件。

--atime-preserve : 不改变文件的访问次数。

-b, --block-size N : 块大小为 N 乘以 512 比特(N 缺省为 20)。

-B, --read-full-blocks : 为兼容 4.2BSD 而设。

-C, --directory DIR : 将目录名改为 DIR。

--checkpoint : 在读归档文件时输出文件名。

-f, --file [HOSTNAME:]F : 使用设备 F 来储存归档文件。

- force-local : 强制设定归档文件为本地文件。
- F, —info-script F —new-volume-script F 为在每盘磁带的结尾执行设定的脚本文件。
- G, —incremental : 用老的 GNU 格式来打包/列出/解开归档文件。
- g, —listed-incremental F : 用新的 GNU 格式来打包/列出/解开归档文件。
- h, —dereference : 不保存符号连接而保存其指向的文件。
- i, —ignore-zeros : 忽略零 block 的文件。
- ignore-failed-read : 在文件不可读时退出。
- k, —keep-old-files : 在从归档文件中释放文件时不改写现有的文件。
- K, —starting-file F : 从归档文件的第 F 个文件开始。
- l, —one-file-system : 只在本地文件系统产生归档文件。
- L, —tape-length N : 在写完  $N * 1024\text{byte}$  后换磁带。
- m, —modification-time : 归档时不改变文件的修改时间。
- M, —multi-volume : 存放/列出/解开多卷的归档文件。
- N, —after-date DATE, —newer DATE : 只存放比日期 DATE 更新的文件。
- o, —old-archive, —portability : 用旧的 V7 格式存放(缺省为 ANSI 格式)。
- O, —to-stdout : 将文件解开到标准输出。
- p, —same-permissions, —preserve-permissions : 解开文件时恢复所有访问限制。
- P, —absolute-paths : 解开文件时使用绝对路径。
- preserve : 同 -p -s。
- R, —record-number : 显示归档文件内部的所有记录数。
- remove-files : 将文件归档后删除源文件。
- s, —same-order, —preserve-order : 显示同归档文件中顺序一样的文件名。
- same-owner : 解开文件时恢复原所有者。
- S, —sparse : 有效存放稀疏文件。
- T, —files-from F : 从文件 F 中读取要释放或解开的文件名。
- null : 使能-T 参数可以读取有空格的文件名,屏蔽参数-C。
- totals : 在创建文件时显示总字节数。
- v, —verbose : 处理过程中输出相关信息。
- V, —label NAME : 用卷标名 NAME 来命名归档文件。
- version : 输出版本号。
- w, —interactive, —confirmation : 进行每一步操作前要确认。
- W, —verify : 在打包文件后进行校验。
- exclude FILE : 不将文件 FILE 归档。
- X, —exclude-from FILE : 不将文件 FILE 中列出的文件归档。
- Z, —compress, —uncompress : 用 compress 来压缩归档文件。
- z, —gzip, —ungzip : 用 gzip 来压缩归档文件。
- use-compress-program PROG : 用自定义的压缩程序 PROG 来压缩归档文件。
- block-compress : 将压缩的归档文件做成块,以便磁带机使用。
- [0-7][lmbh] : 设定磁带驱动器和密度。

### 3. 范例

将当前目录下所有 .txt 文件打包并压缩到归档文件 bak.tar.gz

```
lark:~> tar czvf bak.tar.gz ./*.txt
./tssh.txt
```

将目录 ./sec 打包到归档文件 sec.tar.gz

```
lark:~> tar czvf sec.tar.gz ./sec
./sec/
./sec/advance.edu.tgz
./sec/email
./sec/doc.tar
lark:~>
```

## 4.1.79 tssh

### 1. 简介

tssh - 一个带有文件名自动补充和行编辑功能的 C shell。

tssh 是著名的伯克利 UNIX 的 C SHELL 的增强版本,它既可以作为一个交互式的登录 SHELL 又可以作为脚本文件的处理器。

### 2. 部分参数说明

-b : 强迫 tssh 中止对参数的处理,在此选项之后的参数都作为传递给 tssh 的命令来处理。

-c : 若用 -c 参数,则 tssh 执行完后面所跟的命令后退出。

-d : 从文件 ~/.cshdirs 中读入目录堆栈。

-Dname[=value] : 设置环境变量 name 的值为 value。

-e : tssh 执行命令后返回值为非零时退出 tssh。

-f : tssh 忽略配置文件 ~/.tsshrc,这样启动 tssh 的过程会加快。

-F : tssh 使用 fork 来代替 cfork 以产生新进程。

-i : 即使标准输入不为终端,也成为交互式的 shell。

-l : 表明此 shell 为登录 shell。

-m : tssh 从当前目录下读取 ~/.tsshrc 即使此目录不为当前用户的根目录。

-n : 读入命令但是并不执行。

-q : tssh 接受 SIGQUIT 信号。

-s : 从标准输入接受命令。

-t : 使 tssh 只处理单行命令,当一行写不下的时候,用 “\” 可以在下一行接着输入。

tssh 在作为登录 shell 启动时,首先读取系统配置文件 /etc/csh.cshrc 和 /etc/csh.login,接着到用户的根目录下去读取 ~/.tsshrc,如果没有 ~/.tsshrc 则去读取 ~/.cshrc,接着读取 ~/.history, ~/.login,最后是 ~/.cshdirs。非登录 shell 启动时只读取 /etc/csh.cshrc 和 ~/.tsshrc。

命令行编辑器:

使用方向键来编辑,上下为在历史记录中选择命令,左右键为在命令行上前后移动进行编辑。

命令补充和列表:

在用户键入命令时,经常不能记住命令或目录的全名,这时 `tcsh` 可以帮助用户将剩余部分补全,例如用户键入 `"/usr/lost"` 但是忘记后面的部分,这时候用户只需要键入 `TAB` 键, `tcsh` 就会自动将剩余的部分补全——`"/usr/lost + found/"`,当用户给出的部分有多个匹配的情况下例如 `"/usr/l"` 则 `TAB` 将无法补全,这时候用户只须键入 `"Ctrl + D"` 则 `tcsh` 将替用户列出所有匹配的情况: `"lib/local@ lost + found/"`,用户就可以自己选择了。

拼写检查:

在设置变量 `correct` 的值为 `cmd` 后,则 `tcsh` 会对用户输入错误的命令给出提示

```
> set correct = cmd
> lz /usr/bin
CORRECT> ls /usr/bin (y|n|e|a)?
```

如果回答 `yes` 就会将 `lz` 更正为 `ls`

此外, `tcsh` 所占内存比 `ash` 要大 400k, 为 1.2M 左右,如果用户内存不足的话可以换用其他 `shell`。

以上的介绍只是一些粗略的介绍, `TCSH` 的强大功能有待于用户在实践中去发掘,或使用 `man tcsh` 去查看 `tcsh` 的用户手册。

#### 4.1.80 tee

##### 1. 简介

`tee` - 从标准输入中读取,向标准输出或其他文件中写入。

##### 2. 部分参数说明

```
tee [-ai] [--append] [--ignore-interrupts] [--help]
    [--version] [文件名列表]
```

`-a`, `--append`: 将标准输入追加到输出文件的后面而不是缺省的覆盖它。

`-i`, `--ignore-interrupts`: 忽略中断信号。

`--help`: 在标准输出上输出帮助信息并退出。

`--version`: 在标准输出上输出版本信息并退出。

##### 3. 范例

将输入重定向到标准输出和文件 `out` 中:

```
lark:~> tee out
this is a test
this is a test

lark:~> cat out
this is a test
```

```
fish:~ >
```

#### 4.1.81 test

##### 1. 简介

test - 比较文件类型并返回相应的值。

##### 2. 部分参数说明

```
test [表达式]
```

```
test [-help, -version]
```

test 在表达式为真的时候返回 0, 在表达式为假的时候返回 1, 可行的表达式如下:

-b 文件名: 如果文件存在且为块文件则为真。

-c 文件名: 如果文件存在且为字符文件则为真。

-d 文件名: 如果文件存在且为目录则为真。

-e 文件名: 如果文件存在则为真。

-f 文件名: 如果文件存在且为普通文件则为真。

-g 文件名: 如果文件存在且为设置组则为真。

-k 文件名: 如果文件是真实文件则为真。

-L 文件名: 如果文件存在且为符号连接文件则为真。

-p 文件名: 如果文件存在且为管道文件则为真。

-r 文件名: 如果文件存在且可读则为真。

-s 文件名: 如果文件存在且比上一个文件大则为真。

-S 文件名: 如果文件存在且为 socket 文件则为真。

-t [fd]: 如果 fd 在一个终端上打开则为真。

-u 文件名: 如果文件存在且其设置用户 ID 的位为 1 则为真。

-w 文件名: 如果文件存在且可写则为真。

-x 文件名: 如果文件存在且可执行则为真。

-O 文件名: 如果文件存在且其属主为有效用户则为真。

-G 文件名: 如果文件存在且其属组为有效组则为真。

文件 1 -nt 文件 2: 如果文件 1 比文件 2 新则为真。

文件 1 -ot 文件 2: 如果文件 1 比文件 2 旧则为真。

文件 1 -ef 文件 2: 如果文件 1 与文件在相同的设备上并且有相同的 inode 则为真。

-z 字符串: 如果字符串长度为 0 则为真。

-n 字符串: 如果字符串长度不为 0 则为真。

字符串 1 = 字符串 2: 如果两字符串相等则为真。

字符串 1 != 字符串 2: 如果两字符串不相等则为真。

! 表达式: 如果表达式为假则为真

表达式 1 -a 表达式 2: 表达式 1 和表达式 2 都为真则返回真。

表达式 1 -o 表达式 2: 表达式 1 或表达式 2 只要有一个为真就返回真。

参数 1 OP 参数 2: OP 为 -eq, -ne, -lt, -le, -gt, 或 -ge。

此命令一般是为 shell 编程使用。

#### 4.1.82 time

##### 1. 简介

time - 取得以秒为单位的当前时间。

此条命令取得从 00:00:00 GMT, January 1, 1970 开始的总时间。

##### 2. 范例

```
lark: ~ > time
0.170u 0.400s 12:48.52 0.0%    0 + 0k 0 + 0io 2229pf + 0w
lark: ~ >
```

#### 4.1.83 top

##### 1. 简介

top - 显示系统的最高进程。

top 这个命令可以即时显示当前系统占用 CPU 时间的进程,它同时提供一个交互的界面让用户可以观察系统进程情况,并按照 CPU 使用情况、占内存大小、运行的时间来对进程排序,这是系统管理的一项必不可少的工具。

##### 2. 部分参数说明

```
top [-] [d delay] [q] [c] [S] [s] [i]
```

d : 设定刷新屏幕的时间间隔。

q : 此选项使 top 刷新时间为零,如果是超级用户使用这个选项的话,则 top 将以最高优先级运行。

S : 设定累积模式,即一个进程的 CPU 时间包括它的所有子进程消耗的时间。

s : 使 top 以安全方式运行,即对外界输入不作反应。

i : 忽略死进程和僵尸进程。

c : 显示命令行。

在运行 top 时按 h 键就可以观看 top 交互命令的帮助。

##### 3. 范例

```
lark ~ > top

10:16pm up 2 days, 7:53, 2 users, load average: 0.14, 0.08, 0.06
66 processes: 65 sleeping, 1 running, 0 zombie, 0 stopped
CPU states: 4.7% user, 1.3% system, 5.6% nice, 94.4% idle
Mem: 63104K av, 60608K used, 2496K free, 35332K shrd, 9836K buff
Swap: 128516K av, 292K used, 128224K free, 19508K cached

  PID USER      PRI  NI  SIZE  RSS SHARE STAT   LIB % CPU % MEM   TIME
COMMAND
```

```

5968 root      15   0 11672 11672 1520 S      0  4.1 18.4   3:41 X
10944 fangh     17   0  460  460  344 R      0  1.1  0.7    0:00 top
10356 bbs       1   0  384  384  292 S      0  0.1  0.6    0:00 bbsnnp
10359 bbs       1   0  396  396  288 S      0  0.1  0.6    0:00 bbsnnp
9526 root       1   0  500  500  388 S      0  0.1  0.7    0:00 in.telnetd
7823 chenxy     1   0 2144 2144 1232 S      0  0.1  3.3    0:00 cxterm
   1 root       0   0  320  320  252 S      0  0.0  0.5    0:05 init
   2 root       0   0    0    0    0 SW      0  0.0  0.0    0:00 kflushd
   3 root      -12 -12    0    0    0 SW <    0  0.0  0.0    0:00 kswapd
   4 root       0   0    0    0    0 SW      0  0.0  0.0    0:00 nfsiod
   5 root       0   0    0    0    0 SW      0  0.0  0.0    0:00 nfsiod
   6 root       0   0    0    0    0 SW      0  0.0  0.0    0:00 nfsiod
   7 root       0   0    0    0    0 SW      0  0.0  0.0    0:00 nfsiod
9502 root       0   0  292  292  236 S      0  0.0  0.4    0:00 agetty
120 root       0   0  544  544  400 S      0  0.0  0.8    0:12 nmbd
14 root       0   0  292  284  228 S      0  0.0  0.4    0:00 kerneld

```

#### 4.1.84 touch

##### 1. 简介

touch - 改变文件的时间参数。

touch 将文件的访问时间、修改时间设置为系统的当前时间,如果该文件不存在则建立一个空的新文件。

##### 2. 部分参数说明

```

touch [-acfm] [-r reference-file] [-t MMD-
Dhhmm[[CC]YY][.ss]] [-d time]
[—time = |atime,access,use,mtime,modify}] [—date = time]
[—file = reference-file] [—no-create] [—help] [—version]

```

文件名列表

- a, —time = atime, —time = access, —time = use : 只改变访问时间。
- c, —no-create : 如果目标文件不存在,不建立空文件。
- d, —date time 是: 使用 time 来代替目前时间。
- f : 为了同 BSD 版本的 touch 兼容。
- m, —time = mtime, —time = modify : 只改变修改时间。
- r, —file reference-file : 用文件 reference-file 的时间来代替当前时间。
- t MMDhhmm[[CC]YY][.ss] : 使用此种格式时间来代替当前的时间。
- help : 在标准输出上输出帮助信息并退出。
- version : 在标准输出上输出版本信息并退出。

##### 3. 范例

将当前目录下的文件时间参数修改为当前时间:

```
lark: ~ > touch *
lark: ~ > ls -l
total 1883
drwx----- 2 fangh users 1024 Sep 28 21:07 Mail/
-rw-r--r-- 1 fangh users 71209 Sep 28 21:07 bak.tar.gz
drwx----- 7 fangh users 1024 Sep 28 21:07 circle/
-rw-rw-r-- 1 fangh users 1208006 Sep 28 21:07 circle.tgz
-rw-r--r-- 1 fangh users 378880 Sep 28 21:07 sec.tar.gz
-rw-r--r-- 1 fangh users 252409 Sep 28 21:07 tesh.txt
lark: ~ >
```

#### 4.1.85 tr

##### 1. 简介

tr - 替换或删除字符。

从标准输入中读取输入,经处理后输出到标准输出。

##### 2. 部分参数说明

```
tr [-cst] [-complement] [-squeeze-repeats] [--truncate-set1] 字符串 1 字符串 2
tr {-s,-squeeze-repeats} [-c] [-complement] 字符串 1
tr {-d,-delete} [-c] 字符串 1
tr {-d,-delete} {-s,-squeeze-repeats} [-c] [-complement] 字符串 1 字符串 2
```

-c, --complement : 补足字符串 1。

-d, --delete : 删除字符串 1 而不是替换。

-s, --squeeze-repeats : 将字符串的顺序替换为一个顺序。

-t, --truncate-set1 : 将字符串 1 比字符串 2 长的地方截断。

--help : 在标准输出上输出帮助信息并退出。

--version : 在标准输出上输出版本信息并退出。

##### 3. 范例

将输入中的 abc 替换为 ddd:

```
lark: ~ > tr abc ddd
I love abc
I love ddd
```

#### 4.1.86 tty

##### 1. 简介

tty - 显示目前终端所用的 tty。

tty 的返回值:

0: 标准输入是一个 tty。

- 1: 标准输入不是一个 tty。
- 2: 给出错误参数。
- 3: 发生写错误。

## 2. 部分参数说明

`tty [-s] [-silent] [-quiet] [-help] [-version]`  
-s, -silent, -quiet: 不输出任何信息, 只给出返回值。  
-help: 在标准输出上输出帮助信息并退出。  
-version: 在标准输出上输出版本信息并退出。

## 3. 范例

显示目前终端所用的 tty。

```
lark:~> tty
/dev/ttypl
lark:~>
```

## 4.1.87 umount, mount

### 1. 简介

mount - 装载一个文件系统。  
umount - 卸下一个文件系统。

### 2. 部分参数说明

`mount [-hV]`  
  
`mount -a [-ffnrw] [-t vfstype]`  
`mount [-fnrw] [-o options [...]] device | dir`  
`mount [-fnrw] [-t vfstype] [-o options] device dir`  
  
`umount [-hV]`  
`umount -a [-n] [-v] [-t vfstypes]`  
`umount [-n] [-v] special | node...`  
-V: 输出版权信息。  
-h: 输出帮助信息。  
-v: 校验模式。  
-a: mount 在文件/etc/fstab 中的所有文件系统。  
-F: 同-a 联用, 并行 mount 所有文件系统。  
-f: 同-v 联用, 不作真实 mount 过程只是检查输出。  
-n: mount 文件系统后不写入文件/etc/mtab。  
-r: 将文件系统 mount 为只读模式。  
-w: 将文件系统 mount 为读写模式。

-t vfstype : 指定文件系统类型。

现有可 mount 的系统类型在 /usr/src/linux/fs/filesystems.c 中定义：

minix, ext, ext2, xiafs, hpfs, msdos, umsdos, vfat, proc,  
nfs, iso9660, smbfs, ncpfs, affs, ufs, romfs, sysv, xenix

-o 选项：设定 mount 的一些参数：

async : 文件系统必须被异步写入。

atime : 每次访问时将 i 节点的访问时间更新。

auto : 自动选择文件系统类型。

defaults : 使用缺省选项 suid, dev, exec, auto, nouser 与 async 等。

dev : 解释文件系统上的特殊设备文件。

exec : 允许运行文件系统上的可执行文件。

noatime : 每次访问时不将 i 节点的访问时间更新。

noauto : 禁止自动选择文件系统类型。

nodev : 禁止解释文件系统上的特殊设备文件。

noexec : 禁止运行文件系统上的可执行文件。

nosuid : 禁止设定用户 ID 位和设定组位起作用。

nouser : 禁止非超级用户使用。

remount : 重新 mount 一个已经 mount 的文件系统。

ro : 将文件系统 mount 为只读模式。

rw : 将文件系统 mount 为读写模式。

suid : 允许设定用户 ID 位和设定组位起作用。

sync : 文件系统必须被同步写入。

user : 允许非超级用户使用。

不加参数则列出系统目前 mount 的文件系统。

### 3. 范例

列出系统目前 mount 的文件系统：

```
lark:~ > mount
/dev/hda1 on / type ext2 (rw)
none on /proc type proc (rw)
/dev/hda3 on /home type ext2 (rw)
/dev/hda2 on /usr/local type ext2 (rw)
lark:~ >
加载光驱:
lark:~ > mount /dev/hdb /mnt
```

## 4.1.88 uname

### 1. 简介

uname - 显示系统信息。

## 2. 部分参数说明

- ```
uname [-snrvma] [--sysname] [--nodename] [--release]
      [--machine] [--all] [--help] [--version]
```
- m, --machine : 显示机器硬件种类。
  - n, --nodename : 显示机器网络节点名。
  - r, --release : 显示操作系统发布版本。
  - s, --sysname : 显示操作系统名称。
  - v : 显示操作系统版本。
  - a, --all : 显示上面的所有信息。
  - help : 在标准输出上输出帮助信息并退出。
  - version : 在标准输出上输出版本信息并退出。

## 3. 范例

显示机器 lark 的全部信息:

```
lark: ~ > uname -a
Linux lark 2.0.30 #3 Tue Jun 24 03:49:52 CDT 1997 i586 unknown
lark: ~ >
```

### 4.1.89 unarj, unzip, lha

#### 1. 简介

在 DOS 下最常用的压缩软件有 arj, pkzip 和早期的 lha, 他们产生的压缩文件如何在 Linux 下展开呢? 这就要用到 unarj, unzip, lha 这三个工具软件了, 其用法同其在 dos 下的同类软件用法几乎一样。只是请注意 unarj 和 lha 都是版权软件, 是没有源码的, 不属于 GNU, 且 unarj 和 unzip 只能解压不能压缩。

## 2. 部分参数说明

- unarj l 压缩文件名 查看压缩文件目录。
- unarj x 压缩文件名 解开压缩文件。
- unzip -v 压缩文件名 查看压缩文件目录。
- unzip -d 压缩文件名 解开压缩文件。
- lha l 压缩文件名 查看压缩文件目录。
- lha x 压缩文件名 解开压缩文件。
- lha a 压缩文件名 源文件名 压缩新文件。

### 4.1.90 uniq

#### 1. 简介

uniq - 从一个排好序的文件中删除重复的行。

## 2. 部分参数说明

- ```
uniq [-cdu] [-f skip-fields] [-s skip-chars] [-w check-
```

```
chars] [-# skip-fields] [+ # skip-chars] [-count]
[-repeated] [-unique] [-skip-fields = skip-fields]
[-skip-chars = skip-chars] [-check-chars = check-chars]
[-help] [-version] [输入文件] [输出文件]
```

-u, -unique : 只输出独一无二的行。

-d, -repeated : 只输出重复的行。

-c, -count : 输出重复行的重复次数。

-number, -f, -skip-fields = number : 选择开始比较的域, number 为忽略掉的域, 各域之间是以 TAB 或空格键分开的。

+ number, -s, -skip-chars = number : 选择开始比较的字符, number 为忽略掉的字符。

-w, -check-chars = number : 选择比较多少个字符。

-help : 在标准输出上输出帮助信息并退出。

-version : 在标准输出上输出版本信息并退出。

如不指定输入文件则以标准输入代替, 如不指定输出文件则以标准输出代替。

### 3. 范例

处理文件 sourcefile, 输出到标准输出:

```
lark:~ > cat sourcefile
abcd
abcd
abcde123
abcd

lark:~ > uniq -c sourcefile
  2 abcd
  1 abcde123
  1 abcd

lark:~ >
```

## 4.1.91 uuencode, uencode

### 1. 简介

uuencode - 将一个二进制文件编码为一个文本文件。

uudecode - 将 uuencode 产生的文件复原。

在 Internet 尚未十分发达的时候, 很多用户只能使用 E-mail, 要交换软件是一件十分麻烦的事情, 因为不能用 E-mail 直接传送二进制文件, 于是 uuencode 和 uudecode 就成为十分有用的两个工具。uuencode 产生的文本文件以 begin 644 为开始标志, 以 end 为结束标志在这中间的就是编码过的文件。编码后的文件将比源文件大 35% 左右。uuencode 产生的文件将输出到标准输出, 如要存成文件, 请使用管道功能将其重定向到一个磁盘文件中。

## 2. 部分参数说明

uuencode 源文件名 编码时使用的文件名

vudecode 要解码的文件名

## 3. 范例

对文件 sourcefile 进行编码,

```
lark:~ > uuencode sourcefile sf
```

```
begin 644 sf
```

```
886)C9'IA8F-D'F%B8V1E,3(S'F%B8V0 *
```

```
,
```

```
end
```

```
lark:~ >
```

将目录 temp 打包,压缩,编码后 mail 给用户 fangh:

```
lark:~ > tar cvf - ./temp | gzip | uuencode temp.tar.gz | mail fangh
```

```
./temp/
```

```
./temp/sourcefile
```

```
lark:~ >
```

## 4.1.92 vi

### 1. 简介

vi - 功能强大的 UNIX 编辑器。

vi 是 UNIX 世界里最通用的全屏编辑器,所有的 UNIX 机器都提供本编辑器 Linux 里提供的是 vi 加强版-vim,但是是同 vi 完全兼容。vi 的原意是“visual”即可视编辑器,用户键入时立即被显示出来。而且其强大的编辑功能可以同任何一个最新的编辑器相媲美,而且学会 vi 可以让用户在任何一台 UNIX 机器无论是 SUN, HP, AIX, SGI, 还是 Linux 或 FreeBSD 上都可以得心应手地编辑文件。

## 2. 部分参数说明

只要在命令行上键入 vi 就可以进入 vi 的编辑环境。

vi 有两种状态,输入状态以及指令状态。用户在输入状态下可以输入文字资料,指令状态是用来执行打开文件、存档或离开 vi 等操作命令,执行 vi 后首先进入指令状态,此时输入的任何字符都作为指令来处理。输入“vi 文件名”则 vi 自动装入文件或开始一个新文件,vi 屏幕的左方会出现波浪号“~”,代表本行为空行。

要如何进入输入状态呢? 以下为相应的指令:

追加 (append)

a 从光标所在位置后面开始追加文字,光标后的文字随追加的文字向后移动。

A 从光标所在列最后面的地方开始追加文字。

插入 (insert)

i 从光标所在位置前面开始插入文字,光标后的文字随追加的文字向后移动。

I 从光标所在列的第一个非空白字符前面开始插入文字。

开始 (open)

o 在光标所在列下方新增一列并进入输入状态。

O 在光标所在列上方新增一列并进入输入状态。

修改与删除

x 删除光标所在的字符。

dd 删除光标所在的列。

r 修改光标所在字符接下来要修正的字符。

R 进入改写状态,新增文字会覆盖原先文字

s 删除光标所在字符,并进入输入状态。

S 删除光标所在的列,并进入输入状态。

在 vi 下如何移动光标

| 指令        | 说明                                       | 功能键        |
|-----------|------------------------------------------|------------|
| 0         | 移动到光标所在列的最前面                             | [Home]     |
| \$        | 移动到光标所在列的最后面                             | [End]      |
| [Ctrl][d] | 向下半页                                     |            |
| [Ctrl][:] | 向下一页                                     | [PageDown] |
| [Ctrl][u] | 向上半页                                     |            |
| [Ctrl][b] | 向上一页                                     | [PageUp]   |
| H         | 移动到屏幕的第一列                                |            |
| M         | 移动到屏幕的中间列                                |            |
| L         | 移动到屏幕的最后列                                |            |
| b         | 移动到下一个字的第一个字母                            |            |
| w         | 移动到上个字的第一个字母                             |            |
| e         | 移动到下个字的最后一个字母                            |            |
| ^         | 移动到光标所在列的第一个非空白字符                        |            |
| n-        | 减号移动到上一列的第一个非空白字符<br>前面加上数字可以指定移动到以上 n 列 |            |
| n+        | 加号移动到下一列的第一个非空白字符<br>前面加上数字可以指定移动到以下 n 列 |            |
| nG        | 直接用数字 n 加上大写 G 移动到第 n 列                  |            |
| fx        | 往右移动到第 x 个字符上                            |            |
| Fx        | 往左移动到第 x 个字符上                            |            |
| tx        | 往右移动到第 x 个字符前                            |            |

续表

| 指令       | 说明                                     | 功能键                       |
|----------|----------------------------------------|---------------------------|
| Tx       | 往左移动到第 x 个字符前                          |                           |
| ;        | 配合 f&t 使用,重复一次                         |                           |
| ,        | 配合 f&t 使用,反方向重复一次                      |                           |
| /string  | 往右移动到有 string 的地方                      |                           |
| ? string | 往左移动到有 string 的地方                      |                           |
| n        | 配合 /&? 使用,重复一次                         |                           |
| N        | 配合 /&? 使用,反方向重复一次                      |                           |
| n(       | 左括号移动到句子的最前面<br>前面加上数字可以指定往前移动 n 个句子   | 句子是以<br>! , ? 三种符号来<br>分割 |
| n)       | 右括号移动到下个句子的最前面<br>前面加上数字可以指定往后移动 n 个句子 |                           |
| n{       | 左括弧移动到段落的最前面<br>前面加上数字可以指定往前移动 n 个段落   | 段落是以<br>段落间的空白列<br>来分割    |
| n}       | 右括弧移动到下个段落的最前面<br>前面加上数字可以指定往后移动 n 个段落 |                           |

指令列表如下:

- d 删除(delete)。
- y 复制(yank)。
- p 放置(put)。
- c 修改(change)。

范围可以是下列几个:

- e 光标所在位置到该字的最后一个字母。
- w 光标所在位置到下个字的第一个字母。
- b 光标所在位置到上个字的第一个字母。
- \$ 光标所在位置到该列的最后一个字母。
- O 光标所在位置到该列的第一个字母。
- ) 光标所在位置到下个句子的第一个字母。
- ( 光标所在位置到该句子的第一个字母。
- | 光标所在位置到该段落的最后一个字母。
- | 光标所在位置到该段落第一个字母。

其他:

- :q 结束编辑(quit)  
如果不想存档而要放弃编辑过的文件则用 :q! 强制离开。
- :w 存档(write)

其后可加所要存档的名。

:wq 即存档后离开。

zz 功能与 :wq 相同。

#### 4.1.93 wall

##### 1. 简介

wall - 向所有用户广播信息。

##### 2. 部分参数说明

wall [文件名]

wall 将给定文件名输出到所有登录用户的屏幕上,如果不给出文件名则以标准输入作为输入,以 ctrl + D 作为文件结束符。只有超级用户才能向那些设为屏蔽信息状态的用户屏幕上输出信息。

##### 3. 范例

向所有用户广播信息:

```
bash # wall
system will be shutdown in 5 min!!!
Broadcast Message from root@lark
(/dev/tty5) at 20:08 ...

system will be shutdown in 5 min!!!
```

bash #

#### 4.1.94 wc

##### 1. 简介

wc - 统计一个文件中的字节数、单词数与行数。

##### 2. 部分参数说明

wc [-clw] [-bytes] [-chars] [--lines] [--words] [--help]  
[--version] [文件名列表]

-c, --bytes, --chars : 只输出字节数。

-w, --words : 只输出单词数。

-l, --lines : 只输出行数。

--help : 在标准输出上输出帮助信息并退出。

--version : 在标准输出上输出版本信息并退出。

##### 3. 范例

```
lark: ~ > wc -c tesh.txt
```

```
252409 tesh.txt
lark: ~ > wc -w tesh.txt
27584 tesh.txt
lark: ~ > wc -l tesh.txt
5478 tesh.txt
lark: ~ >
```

#### 4.1.95 whereis

##### 1. 简介

whereis - 定位可执行文件、帮助文件、源代码文件。

缺省时 whereis 在下列目录搜寻文件。

|                          |                           |
|--------------------------|---------------------------|
| /bin                     | /usr/interviews/bin/LINUX |
| /usr/bin                 | /usr/bin/X11              |
| /etc                     | /usr/X11/bin              |
| /usr/etc                 | /usr/X11R5/bin            |
| /sbin                    | /usr/X11R6/bin            |
| /usr/sbin                | /usr/X386/bin             |
| /usr/games               | /usr/local/bin            |
| /usr/games/bin           | /usr/local/etc            |
| /usr/emacs/etc           | /usr/local/sbin           |
| /usr/lib/emacs/19.22/etc | /usr/local/games          |
| /usr/lib/emacs/19.23/etc | /usr/local/games/bin      |
| /usr/lib/emacs/19.24/etc | /usr/local/emacs/etc      |
| /usr/lib/emacs/19.25/etc | /usr/local/TeX/bin        |
| /usr/lib/emacs/19.26/etc | /usr/local/tex/bin        |
| /usr/lib/emacs/19.27/etc | /usr/local/bin/X11        |
| /usr/lib/emacs/19.28/etc | /usr/contrib              |
| /usr/lib/emacs/19.29/etc | /usr/hosts                |
| /usr/lib/emacs/19.30/etc | /usr/include              |
| /usr/TeX/bin             | /usr/g++-include          |
| /usr/tex/bin             |                           |

##### 2. 部分参数说明

whereis [-bmsu] [-BMS 目录名列表 -f] 文件名列表

- b : 只搜寻可执行文件。
- m : 只搜寻 man 帮助文件。
- s : 只搜寻源代码文件。
- u : 搜寻上述路径中不属于三种文件中任何一种文件。
- B : 改变搜寻可执行文件的路径。
- M : 改变搜寻 man 帮助文件的路径。
- S : 改变搜寻源代码文件的路径。

### 3. 范例

查询 whereis 所在路径。

```
lark:~ > whereis whereis
whereis: /usr/bin/whereis
lark:~ >
```

## 4.1.96 who

### 1. 简介

who - 查看其他登录的用户。

### 2. 部分参数说明

```
who [-inqsuwHT] [-count] [-idle] [-heading] [-help]
    [-message] [-mesg] [--version] [--writable] [file] [ami]
```

-m : 同 whoami 和 “who am i”

-q, -count : 只输出用户的登录名和正在使用的用户数日。

-i, -u, -idle : 输出用户发呆的时间, “.”表明用户没有发呆, “old”表明用户发呆超过 24 小时。

-H, -heading : 输出表头。

-w, -T, -mesg, -message, -writable : 输出用户的信息屏蔽状态。

+ : 可写。

- : 不可写。

? : 不能发现终端设备。

-help : 在标准输出上输出帮助信息并退出。

-version : 在标准输出上输出版本信息并退出。

### 3. 范例

列出系统当前用户:

```
lark:~ > who -H
USER      LINE      LOGIN-TIME  FROM
chenxy    ttyt2     Sep 29 20:37 (hpte2.ihep.ac.cn)
liubx     ttyt3     Sep 29 15:59 (:0.0)
fangh     ttyt4     Sep 29 19:19 (csun01.ihep.ac.c)
```

```
lark:~ >
```

## 4.1.97 which

### 1. 简介

which - 显示命令的全路径。

### 2. 部分参数说明

which 程序名

### 3. 范例

查询 kill 和 who 的路径。

```
lark: ~ > which kill who
kill: shell built-in command.
/usr/bin/who
lark: ~ >
```

## 4.1.98 write

### 1. 简介

write - 向另外一个用户发信息。

如果用户不是超级用户,则不能对设定 message n 的用户发信息,输入完成后,使用 ctrl-D 结束。

### 2. 部分参数说明

```
write user[ @host ] [ ttyname ]
```

## 4.1.99 zoo

### 1. 简介

zoo - 使用 Lz 方式压缩文件并打包。

### 2. 部分参数说明

```
zoo { acfDeghHILPTuUvVx } [ aAcCdEfghImMnNoOpPqSul : / . @ n + - = ]
```

打包文件名 文件名列表

zoo -command 打包文件名 文件名列表

zoo h

| 选项       | 描述                | 专家选项  |
|----------|-------------------|-------|
| -add     | 向打包文件中增加文件        | aP:   |
| -extract | 从打包文件中解开文件        | x     |
| -move    | 移动文件到打包文件中        | aMP:  |
| -test    | 测试打包文件            | xNd   |
| -print   | 从打包文件中解开文件输出到标准输出 | xp    |
| -delete  | 从打包文件中删除文件        | DP    |
| -list    | 列出打包文件的目录         | VC    |
| -update  | 更新打包文件中的文件        | aunP: |
| -freshen | 增加打包文件中的文件        | auP:  |
| -comment | 给文件加注解            | c     |

### 3. 范例

将目前目录下所有文件打包到文件 bak.zoo 中:

```
lark:~ > zoo a bak ./ *
Zoo: ./bak.tar — (24%) added
Zoo: ./circle — (31%) added
Zoo: ./sec.tar — (20%) added
Zoo: ./sourcefile — (13%) added
lark:~ >
```

## 4.2 Emacs 的使用

### 4.2.1 功能强大的编辑器 Emacs 介绍

Emacs 大概是世界上使用最广泛的编辑器了,不仅可以在 UNIX 平台上使用,还可以在 DOS、VMS、OS/2 和 Windows NT 上使用,这几乎覆盖了 90% 以上的系统平台,而且其强劲的功能更让用户爱不释手,在 Emacs 中用户可以调试程序(使用 gdb)、收发 E-mail、阅读网络上的新闻组、作计算器、写 latex 文件、进行拼写检查,并且提供详尽的在线帮助和文档,可以由用户自行定制环境,其内建的 lisp 语言更为其提供了强大的可扩充性能。可以说“足不出 Emacs 就可做任何事”。因而 Emacs 也成为与 GCC 齐名的 GNU 的拳头产品。

Emacs 的名称是由 Editor MACroS 缩写而来,因为它是在 1976 年由 GNU 的创始人 Stallman 在美国麻省理工学院的 PDP-10 机器上为一个文本编辑器 TECO 所编写的一套宏命令。目前最新的 Emacs 版本是 20.2,在这一版中,Emacs 将强大的多国语言处理工具 mule(由日本 ETL 在 Emacs 的基础上编写的, MULTilingual Enhancement to GNU Emacs)集成到官方版本中,使其对于我们中国计算机用户来说也成为在 UNIX 环境下编辑中文文件的一个强有力的工具。

下面就是 Emacs 在 X Window 下的标准界面(见图 4.2)。

### 4.2.2 Emacs 安装简介

Slackware Linux 和 RedHat Linux 都缺省将 Emacs 作为标准配制,在 Slackware Linux 中 Emacs 是作为 E 包来安装的。目前的 20.2 版如果不包括其国际字库的部分全部安装的话,需要 35M 硬盘空间。如果用户要自行安装 Emacs 则需要至少 8M 以上的虚拟内存,否则在编译过程中会出错,并且需要 80M 左右的硬盘空间以便编译使用。如果安装国际字库包的话还需要另外的 20M 空间。

下面以 Emacs 20.2 为例讲解一下安装的全过程:

(1) 取得 Emacs 的源程序(在任何 sunsite 的 mirror 节点上)

|            |                   |                |
|------------|-------------------|----------------|
| Emacs 的源文件 | emacs-20.2.tar.gz | 13051804 bytes |
| 多国文字输入法    | leim.20.2.tar.gz  | 3303280 bytes  |
| 国际字库包      | intlfonts-1.0.tar | 18022400 bytes |

(2) 选定一个临时目录展开打包文件



图 4.2 Emacs 的界面

```
tar xzvf emacs-20.2.tar.gz
```

```
tar xzvf leim.20.2.tar.gz
```

会产生一个目录 `emacs-20.2`，其中 `leim.20.2.tar.gz` 展开后在其子目录 `leim` 下，如不安装 `leim` 包，则 Emacs 中就没有中文输入法，当然也就无法输入汉字了。

### (3) 编译并安装 Emacs

进入目录 `emacs-20.2`，依次输入下列命令

```
./configure      进行自动配置
./make           进行编译
./make install   进行安装
```

首先 `./configure` 将进行对系统环境的自动探测，产生下列文件

```
./Makefile
./lib-src/Makefile.c
./oldXMenu/Makefile
./man/Makefile
./lwlw/Makefile
./src/Makefile.c
./leim/Makefile
./src/config.h
./src/paths.h
./lib-src/Makefile
./src/Makefile
```

一般说来用户不用干预这一过程，但是如果出现问题的话，则可以阅读文件

./INSTALL来查询。输入./make后,接下来的编译过程耗时较长,依不同的机器而定,大约从十几分钟到半个小时不等。如果一切顺利的话就可以进行下一步的安装了。输入./make install 就会进行 Emacs 的安装,缺省时将安装到/usr/local 目录下(注意,Slackware 和 RedHat 缺省的将 Emacs 安装到/usr 目录下),如果要改变目录,请使用 make install prefix = /usr/来将可执行文件安装到/usr 目录下。现在 Emacs 已经被安装到用户的系统之中了,但是如果要使用中文的话用户还要安装国际字库包—intlfonts-1.0.tar,将此包解开到目录/usr/X11R6/lib/X11/fonts 下:

```
cd /usr/X11R6/lib/X11/fonts
tar xvf /字库包所在路径/intlfonts-1.0.tar
```

这会产生相应的字库文件,运行 unpackfont 这个文件,解开相应的包对于中文使用者来说,一般只需要展开下列包:Asian,Chinese-BIG, Misc,Chinese 就可以了,相应的命令是:

```
./unpackfont Asian Chinese-BIG Misc Chinese
```

其会建立相应的目录并且展开字库文件,建立 fonts.dir 和 fonts.alias 这两个文件,并且将这些目录加入系统目前的 X 字型的路径中,但是要想下次开机时自动加入的话,请去/etc(在 RedHat 下是/etc/X11)目录下修改 XFree86 的配置文件:XF86Config,在设置 FontPath 的地方加入下列行:

```
FontPath "/usr/X11R6/lib/X11/fonts/Asian/"
FontPath "/usr/X11R6/lib/X11/fonts/Chinese/"
FontPath "/usr/X11R6/lib/X11/fonts/Chinese-BIG/"
FontPath "/usr/X11R6/lib/X11/fonts/Misc/"
```

这样系统下次启动 X Window 时就会自动加载这些字型。用户就可以在 Emacs 中使用中文了。

#### 4.2.3 进入 Emacs 的世界

在命令行上输入 emacs 就可以启动 Emacs 了,如果用户没有启动 X Window 环境,则出现的将是 Emacs 的文本环境,这其实是 Emacs 使用时间最长也是最经典的界面——文本界面,但是其操作的热键和命令同 X Window 下的是一致的,本书不再作重复介绍,其界面如下图(见图 4.3)。

Emacs 同 vi 不一样,没有编辑状态和指令状态之分,其最重要的概念是其独特的缓冲区,Emacs 编辑的所有文件都是放在缓冲区中的,Emacs 支持同时编辑多个缓冲区,它可以将一个文件在多个缓冲区中打开不同的拷贝,甚至其所有的在线帮助和文档以及出错信息都是作为一个缓冲区来显示,当然这些缓冲区是不可写的,用户可以在这些缓冲区之间拷贝或粘贴文本。并且一般所有的缓冲区在硬盘上都有一个以“#”开头的备份文件,这样在系统突然崩溃的时候可以即时将用户的工作进行备份。

在用户编辑文件时,常见的菜单界面如下,除非用户在编辑一些特殊类型的文件,例如当用户编辑扩展名为.c 的 C 语言文件时,Emacs 会产生菜单选项 C 向用户提供一些针对编辑 C 程序特别有用的一些命令。当用户编辑扩展名为.tex 的文件则会多出菜单选项 tex,让用户在编辑完 tex 文件后可以即时地观看输出并打印。下面主要介绍一下常见菜

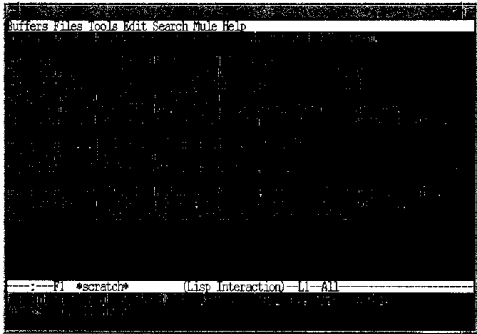


图 4.3 字符界面的 Emacs

单和热键的使用。首先我们介绍一下在 Emacs 的帮助中最常见的一些缩略符:所有 Emacs 的热键都是由 CONTROL 键(一般是键盘上的 Ctl 键)和 META 键(一般是键盘上的 Alt 键)加上一些键的组合组成的,如果没有 Alt 键,则可以用输入一个 Esc,再输入相应的键来代替。

例子: C-<键> 表示按住 ctl 键并输入相应的键, C-f 表示按住 ctl 键并输入 f 键

M-<键> 同 C-<键>

下面是简单的菜单介绍:

Buffers:

当前缓冲区列表

List All Buffers (C-x C-b)

此菜单项类似于 WINWORD 中的选择窗口命令,可以在不同的窗口之间切换。这在编辑多个文件时是很有用的。

Files :

此菜单项是有关文件操作和窗口操作的命令部分。

Open File... (C-x C-f)

打开一个文件,如果输入的文件名不存在则开辟一个新文件。

Open Directory (C-x C-d)

打开一个目录,将文件排列成表,用户可以移动光标或鼠标到相应的文件下,再输入回车就会打开该文件,相当于 windows 下的文件列表框。

Save Buffer (C-x C-s)

将当前正在编辑的文件存盘。

Save Buffer As... (C-x C-w)

将当前正在编辑的文件另存为新的文件。

**Revert Buffer**

从磁盘中重新打开正在编辑的文件, 这样会销毁用户做的任何更改。

**Insert File...** (C-x i)

在光标处插入文件。

**Kill Current Buffer**

关闭目前正在编辑的文件。

**Make NewFrame** (C-x 5 2)

打开一个新的窗口, 窗口中的文件就是用户正在编辑的文件。

**Open New Display...**

在另外一个 X 显示上打开一个新的窗口。

**Delete Frame** (C-x 5 0)

关闭光标所在的窗口。

**Split Window** (C-x 2)

拆分窗口为两个窗口。

**One Window** (C-x 1)

将光标所在窗口最大化为一个窗口。

**Exit Emacs** (C-x C-c)

退出 Emacs。

**Tools :**

此菜单项是一些常用工具的集合。

**Print Buffer**

打印当前正在编辑的文件。

**Print Region**

打印被选中的部分。

**Postscript Print Buffer**

用 Postscript 过滤器来打印当前文件。

**Postscript Print Region**

用 Postscript 过滤器来打印被选中的部分。

**Compare**

比较文件。

**Merge**

合并文件。

**Apply Patch**

准备给一个文件打补丁(编程者使用)。

**Ediff Miscellanea**

Ediff 功能(编程者使用)。

**Version Control**

版本控制(编程者使用)。

**Read NetNews**

阅读网络新闻组。

Read Mail

阅读电子邮件。

Send Mail

发送电子邮件。

Display Calendar

显示一个日历。

Search Files...

搜寻文件。

Compile...

编译(编程者使用)。

Debugger...

调试(编程者使用)。

Edit:

这就是常用的编辑命令:

Undo (C + \_)

撤消上一次操作。

Cut (cut)

剪贴。

Copy (copy)

拷贝。

Select and Paste

选择并粘贴。

Paste (paste)

粘贴。

Clear

清除。

Text Properties

设置属性,包括显示字型、颜色或其他一些属性。

Fill

填充。

Spell

拼写检查,可以选择多种语言的字典。

Search:

搜寻菜单。

Search...

向下查找。

Regexp Search...

向下表达式查找。

Search Backwards...

向上查找。

Regexp Search Backwards...

向上表达式查找。

Repeat Search

重复向下查找。

Repeat Regexp

重复向下表达式查找。

Repeat Backwards

重复向上查找。

Repeat Regexp Backwards

重复向上表达式查找。

Bookmarks

书笈。

Find Tag... (M\_)

寻找标记。

Query Replace (M\_%)

查询替换。

Query Replace Regexp...

表达式替换。

#### 4.2.4 获得帮助

想快速了解 Emacs 的人,可以参考 Emacs 的自学教材(tutorial)。使用 Emacs 自学教材的方法很简单,只要输入“Ctrl-h t”(help-with-tutorial)即可进入 Emacs 的自学的状态了。Emacs 的自学教材可分成以下几部分:

1. 介绍 Emacs 目录的输入方法,即介绍

Ctrl 与 Meta 键。

2. viewing screenfuls。

3. 光标控制。

4. Ctrl-g 的用法。

5. 窗口的使用。

6. 插入和删除。

7. 取消操作。

8. Emacs 文件的处理。

9. Emacs 的缓冲区。

10. 扩展命令。

11. 简介 Emacs 的 mode line 与 echo area。

12. 查询和替换。

13. recursive editing levels。

14. 获得更多帮助。

15. 离开 Emacs。

16. 有关 Emacs 的版权问题。

用户在使用 Emacs 之前最好仔细阅读这些文档。

Emacs 除了自学教材可供参考外,它还提供了其他的在线帮助,让使用者可以随时查阅需要的相关信息。

Emacs 的在线帮助都是以“Ctrl-h”为开端,其种类有以下二种:

Ctrl-h

Ctrl-h Ctrl-h

输入 Ctrl-h (help-command), 屏幕的最下端会出现如下的信息:

```
C-h (Type ? for further options)-
```

此时的“Ctrl-h”只是用做前置字(prefix key),它是用来等候使用者输入其他的命令。如果输入“?”,则屏幕的下方会出现所有可供使用的选择,使用者可根据需要来选择合适的选项。

如果输入两次的“Ctrl-h”(help-for-help),屏幕下方会出现:

```
type one of the options listed or Space to scroll
```

此时 Emacs 会另外开启一个窗口,将所有与求助的选项都列出来,且会做一简要的说明,要滚动此窗口则输入 Space。此新开启的窗口共有二十一个选项,包括:

```
a b e f C-f i k C-k l m n p s t v w C-c C-d C-n C-p C-w
```

使用这二十一个选项的方法非常简单,只要在“Ctrl-h”之后输入任一选择就可以了。例如,要选择“a”,则执行“Ctrl-h a”即可。

现在将 Emacs 常用的在线帮助一一作解释。Emacs 常用的在线帮助有:

```
Ctrl-h c
```

```
Ctrl-h a
```

```
Ctrl-h k
```

```
Ctrl-h v
```

```
Ctrl-h w
```

```
Ctrl-h i
```

“Ctrl-h c”与“Ctrl-h k”的功能相似,二者都是在寻求与热键有关的信息。二者唯一的差别,就在于对命令的解释详细与否而已。“Ctrl-h c”是简述与热键连结的命令,而“Ctrl-h k”则详述连结热键的命令。二者都有一个缺省的前提,那就是都先知道热键是什么键,而想进一步知道此热键所使用命令的名称。现在举“Ctrl-x Ctrl-c”的例子来说明二者的差别。

输入“Ctrl-h c RET”(RET,即键盘上的回车键。任何一个命令输入完毕时,必需紧跟着一个 Enter。它的作用是用来告诉系统,命令输入已经结束,可以开始执行相关的动作了),则屏幕下方会出现:

```
Describe key briefly: -
```

在“:”的后面输入“Ctrl-x Ctrl-c”,则在原先出现“Describe key briefly: -”的地方出现:

```
C-x C-c runs the command save-buffers-kill-emacs
```

输入“Ctrl-h k RET”,则屏幕下方出现

```
Describe key: -
```

在“:”的后面输入“Ctrl-x Ctrl-c”,则 Emacs 会另以一个窗口显示如下的信息:

```
save-buffers-kill-emacs:
```

```
Offer to save each buffer, then kill this emacs process.
```

```
With prefix arg, silently save all file-visiting buffers, then kill.
```

此信息的第一行是热键所连结命令的全名,以后的行数则是对此命令的详细说明。

“Ctrl-h w”(where-is)的用法与“Ctrl-h c”和“Ctrl-h k”正好相反。“Ctrl-h w”是在知道 Emacs 的命令而想知道是否有相对应的热键时所使用的。例如输入“Ctrl-h w”,屏幕的下方会出现:

```
where is command:
```

在“:”之后输入“save-buffers-kill-emacs”,则在原处会出现:

save-buffers-kill-emacs is on C-x C-c

想知道命令是否有相对应的热键,可以用这个方法查询。“Ctrl-h w”可以使用 Emacs 的自动补充功能,但不方便的地方就是必需给予命令的第一个字符。如“save-buffers-kill-emacs”,必需先给予以 s 开头的子字符串,才能逐步使用 Emacs 的自动补充功能。如果不输入 s 开头的字符串,而输入其他字符串,则无法找到相对应的命令,所以使用“Ctrl-h w”必需要记住命令的第一个字。除此之外,Emacs 的在线帮助还提供了另一种帮助,可让使用者输入任意的子字符串,都可以找到相对应的命令,那就是:

Ctrl-h a

输入“Ctrl-h a”则屏幕的下方会出现

command-apropos (regex):

此时只要给予与命令相关的任一子字符串或“regular expression”)再按下 RET, Emacs 会另开一个窗口,将所有包含此子字符串或 regular expression 的命令全列出来。“Ctrl-h a”主要的目的是当使用者无法正确地输入命令的第一个字符时,可以将所有包含使用者记得的部分子字符串的命令都列举出来。

Emacs 除了以上几个常用的在线帮助之外,还有一个非常实用的资料查阅中心,那就是“Ctrl-h i”。“Ctrl-h i”执行“Info program”,它主要是用来浏览已建立成树状结构的文本文件,在 Emacs 的在线帮助功能中,目前所有与 GNU 有关的文件都可透过 Info 来阅读。

info 对于文件的编排,是以超文本的方法来处理所有的相关文件。当输入“Ctrl-h i”或“Esc-x info”后,会先进入 info 树状(tree)结构的最顶端。如下就是进入 info 时的第一个画面:

```

- * - Text - * -

```

```

This is the file .../info/dir, which contains the topmost node of the Info
hierarchy. The first time you invoke Info you start off looking at that
node, which is (dir)Top.

```

```

File: dir      Node: Top      This is the top of the INFO tree

```

```

This (the Directory node) gives a menu of major topics.

```

```

Typing "d" returns here, "q" exits, "?" lists all INFO commands, "h"
gives a primer for first-timers, "mTexinfo" visits Texinfo topic,
etc.

```

```

-- PLEASE ADD DOCUMENTATION TO THIS TREE. (See INFO topic first.) --

```

```

* Menu: The list of major topics begins on the next line.

```

```

* Info: (info). Documentation browsing system.

```

```

* Emacs: (emacs). The extensible self-documenting text editor.

```

```

* VIP: (vip). A VI-emulation for Emacs.

```

```

* Texinfo: (texi.info).

```

```

    With one source file, make either a printed manual
    (through TeX) or an Info file (through texinfo).

```

```

    Full documentation in this menu item.

```

```

* Termcap: (termcap).

```

The termcap library, which enables application programs to handle all types of character-display terminals.

\* Regex: (regex).

The GNU regular expression library.

A

\* Cpp: (cpp.info).

C pre-processor.

\* Gcc: (gcc.info).

GNU C Compiler — an ANSI C Compiler developed by FSF.

\* Gzip: (gzip.info).

GNU zip program — an compress package developed by FSF.

\* Ispell: (ispell.info).

A spelling checker.

\* Libg++: (libg++.info).

C++ libraries.

\* Gmake: (make.info).

A make utility developed by FSF.

\* Bison: (bison.info).

GNU Yacc.

\* Gawk: (gawk.info).

GNU awk — pattern scanning and processing language.

\* Gdb: (gdb.info).

GNU debugger.

\* Info-std: (info-std.info).

Stand alone GNU info.

\* Makeinfo: (makeinfo.info).

Program for producing \*.info file from \*.texi file.

\* Graphics: (graphics.info).

A set programs for producing plot files and display them on Tektronix 4010, PostScript, and X window system compatible output devices.

\* m4: (m4.info).

m4 is macro processor, in the sense that it copies its input to the output, expending macros as it goes. GNU m4 is mostly compatible with system V, Release 3 version.

\* Hyperbole: (hyph.info).

GNU Emacs-based everyday information management system.

Use `3C-h h d d` for a demo. Include Smart Key context-sensitive mouse or keyboard key support, a powerful rolodex, and extensible hypertext facilities including hyper-links in mail and news messages.

\* Standards Coding Style: (standards.info).

GNU Coding Style.

下面是 info 的常用热键:

h

不知如何使用 info 模式者,可在进入 info 模式后,使用 info 的在线帮助。要使用 info 的在线帮助,只要在进入 info 模式后,输入“h”即可。此不只是一个在线帮助,还是一个简单的教程。阅读它的说明,即可了解如何使用 info。

d

“Ctrl-h i”命令,会先进入 info 树状结构的根部。任何情况下,可输入“d”回到此根部。

?

想知所有与 info 有关的命令,只要输入“?”,就可以得知所有命令的全貌。

q

要离开 info,只要输入“q”,就可以离开 info 而回到先前的缓冲区。

上面就是如何使用 Emacs 的帮助。

#### 4.2.5 Emacs 使用技巧

##### 1. 如何在 Emacs 中执行 Shell

在 Emacs 中有两种执行 shell 的方法:一种是进入 shell command mode,另一种是进入 shell mode。二者都可以执行 shell,其最大不同之处是,进入 shell mode 状态执行 shell 的同时,仍可以切换到其他模式处理别的工作,但如果使用 shell command mode,就必须等 shell 执行完后才可以做其他的事。

使用 shell command mode 时,使用者在屏幕的最下方输入要执行的 shell 命令,Emacs 会开启一个名为“\* Shell command output \*”的窗口,将 shell 命令执行的结果显示在此窗口中。shell mode 则是执行一个子 shell,其输入与输出都是透过同一个缓冲区,所以输入与输出是在同一个地方,它不似 shell command mode,命令输入与结果的显示在不同的地方。

##### (1) shell command mode

Esc-! (shell-command)启动 shell command mode

Esc- (shell-command-on-region)

##### (2) shell mode

Esc-x shell \ index Esc-x shell 是启动 shell mode 的命令。

##### 2. 程序的编辑、编译与测试

Emacs 是一个综合的环境,在提供程序编辑的同时,自然会提供一个可供程序执行的环境。以下就谈谈 Emacs 可以为程序开发者提供那些服务。Emacs 针对不同的语言提供不同的编辑模式。Emacs 提供的服务有程序缩进的安排、括号对应的提示、程序注解的安排、光标移动的方式与程序的删除等。基本上 Emacs 是提供一个编写程序的格式,只是此格式可根据使用者的需要而自行设计。Emacs 选择适合的语言模式,并根据所编辑的文件并扩展名来判断。像上面提到的那样,如果用户编辑扩展名为 .c 的 C 语言程序,Emacs 会自动给予 C 语言模式,而不需使用者自行处理。Emacs 提供的程序语言模式有 LISP, SCHEME, C, C++, FORTRAN, MAKEFILE, AWK, PERL, ICON 与 MUDDLE

等。

编辑好的程序可以直接进入 Emacs 编译模式,不需离开 Emacs 到 UNIX 的 shell 下进行编译动作。进入 Emacs 的编译模式很简单,只要输入“Esc-x compile”即可。Emacs 缺省的编译命令是 make,执行 Esc-x compile 命令的结果如下所示:

```
compile command: make -k
```

如果要使用其他的编译器,只需在“compile command:”的后面加上对应的编译命令即可,此命令与在 UNIX shell 下使用编译的方法完全相同。

除了编辑、编译之外,程序开发者还需要的功能是调试器,Emacs 也提供了在编辑器内部调试程序的功能。Emacs 提供了四种调试器,分别为 gdb,dbx,xdb 与 sdb,使用者可根据需要来选择合适的调试器。下面是使用调试器的命令:

```
Esc-x gdb RET file RET
```

```
Esc-x dbx RET file RET
```

```
Esc-x xdb RET file RET
```

```
Esc-x sdb RET file RET
```

### 3. 如何在 Emacs 中打印文件

除了以上的功能外,Emacs 还提供了打印功能。Emacs 可针对整个缓冲区或某部分的区域打印,其相关的命令如下:

Esc-x print-buffer

打印整个缓冲区的内容。Emacs 处理此工作的方法是先使用 shell 的 pr 命令,然后再使用 shell 的 lpr 打印命令。

Esc-x lpr-buffer

此命令与上一个命令相似,只是不通过 pr 而直接使用 lpr。

Esc-x print-region

与 Esc-x print-buffer 相似,唯一不同之处在于,此命令只打印部分的区域。

Esc-x lpr-region

与 Esc-x lpr-buffer 相似,唯一不同之处在于,此命令只打印部分的区域。

### 4. 在 Emacs 中如何收发信件

在 Emacs 众多的综合功能中,信件收发的功能自然是不可或缺的。Emacs 所提供的电子邮件系统分为两个部分,一部分发送信件(mail),另一部分为收取信件(rmail)。Emacs 对于发送信件与读取信件提供了许多方便的操作功能,读者可自行参考手册,下面只告诉读者如何进入送信件与读信件的模式:

```
Esc-x mail (发送信件)
```

```
Esc-x rmail (读取信件)
```

### 5. 其他 Emacs 工具

日历

Emacs 的 Calendar 与一般的日历功能相似。Diary 还可以提醒使用者该注意的事情。

进入 Calendar 的方法如下:

`Esc-x calendar`

Diary 的使用可以在进入 Calendar 的模式后,再来设定相关的配置。

阅读新闻组

Emacs 中阅读新闻组的子程序称为 GNUS。Emacs 的 GNUS 根据 “.newsrc”去下载相应的新闻组的信件,它的内容包括所有被订阅的(subscribe)的新闻组,以及未被阅读的文章。在 GNUS 中还可以看到或隐藏未被订阅的新闻组,可以再订阅未订阅的新闻组或取消订阅某一个新闻组。使用 GNUS 这个子系统只需输入如下的命令即可:

`Esc-x gnus RET`

版本控制(Version Control)

管理源代码(source file)也是 Emacs 所提供的服务项目之一。VC(版本控制)是一个套装软件,它可以记录源代码所有改变的版本,它保留所有改变的记录且存放于一个文件中,对于每一版本重复的部分只会保留一份记录。VC 也会记录每一版本被创建、谁创建等的相关资料。目前 Emacs 是通过 VC 来使用操作系统所提供的 RCS 或 SCCS 的版本控制软件。

下面就来简介 Emacs 版本控制的功能。Emacs 的 VC 提供的功能如下:

- (1)将文件注册于 VC 之下。
- (2)可将注册的文件从 VC 的控制中取出与放入。
- (3)放入 VC 的每一个版本都可以随时取出。
- (4)可比较任一版本间的异同。
- (5)可将一组相关的文件,置于 VC 之下。
- (6)可自行设计版头(version header),此标题可置于 VC 下的文件中。

## 6. 在 Emacs 中设定变量

在 Emacs 中任何设定变量的方法都是以 “Meta-x set-variable(或 Esc-x set-variable)”来完成变量的设定。用户可以只设定真假值变量、数值变量或字符串变量。

如果只是设定真假值变量,Emacs 有一个缺省的规则,Emacs 中以任何 “non-nil” 的值来代表真,习惯上是以 “t” 来表示真;而以 “nil” 来代表假。

在设定新的变量值之前,如果想知道目前变量的值,Emacs 可以 “Ctrl-h v”(describe-variable)来查阅变量的值。下面是简单的例子:

以 “Ctrl-h v” 查阅 auto-save-visited-file-name 的变量的值

- (1)echo area 处会出现 Describe variable。
- (2)在 Describe variable: 后输入 auto-save-visited-file-name。
- (3)屏幕上会另开一个窗口,显示如下的信息:

```
auto-save-visited-file-name's value is
nil
```

Documentation:

\* Non-nil says auto-save a buffer in the file it is visiting, when practical.

Normally auto-save files are written under other names.

以“Meta-x set-variable”来设定变量

- (1) echo area 处会出现 Set variable。
- (2) 在 Set variable: 后输入 auto-save-visited-file-name RET。
- (3) echo area 处会出现 Set auto-save-visited-file-name to value。
- (4) 此时可以利用在“:”之后的 minibuffer, 输入变量的值。此变量目前的值是“nil”, 要改其值为肯定的可以输入“t”。

以“Ctrl-h v”查询 auto-save-visited-file-name 的变量

在 Emacs 执行过程中所设定的变量值, 只对日前所执行的 Emacs 有用, 一旦离开此 Emacs, 所有的设定就恢复成原来的缺省值。要想永久保留此设定的变量值, 就必需将所设定的变量值储存在文件名为“.emacs”。

.emacs 文件为 Emacs 的起始文件, 进入 Emacs 时会先执行此文件内的命令, Emacs 的设定也要根据此文件。因为启动 Emacs 时, Emacs 会先执行 .emacs 文件, 所有存于此文件的变量会被重新设定一次。

在 .emacs 文件中设定变量“auto-save-visited-file-name”与“auto-save-interval”的方法如下所示:

```
(setq auto-save-visited-file-name t)
(setq auto-save-interval 350)
```

## 4.3 RPM 的使用

### 4.3.1 什么是 RPM

RPM 是 Red Hat Package Manager 的缩写, 是由 Red Hat 公司根据 BOGUS (PM 与 PMS) 发展而来的软件包的安装与管理程序, 就像 Windows 平台上的 Uninstaller 和 Cleansweep 这些软件一样, 是让用户用来自行安装和管理 Linux 上的应用软件包的管理程序。其最大的好处就是让用户可以直接以 Binary 方式安装软件包, 免去了自行编译之苦, 并且可以帮用户查询是否已经安装有相关的运行库, 在卸除这些软件包的时候, 它又会很聪明地不会像笨笨的 Slackware 的 pkgtool 那样把其他公用的程序运行库删除。在利用 rpm 升级软件的时候, rpm 会保留原先的配置文件, 这样用户就不用重新手工设定新的软件了。同时, rpm 管理一个数据库, 里面包含了所有的软件包的文件资料, 通过这份数据库, 用户可以进行软件包的查询工作。RPM 主要是给 Linux 使用的, 但对于其他 UNIX 系统的兼容性也相当不错, 事实上, 它已经移植到 SunOS, Solaris, AIX, Irix, AmigaOS 及其他系统上。但是不同 UNIX 操作系统所编译出来的 binary 软件包是互不兼容的。Red Hat 公司鼓励其他 Linux 厂商在它们的 distribution 当中使用 RPM (不过目前好象还没有什么消息, UNIX 上传统的地方割据, 各自为政在这儿又得到了体现)。RPM 是一个开放的系统, 操作非常简单, 并且提供自行扩充的接口。它的开发过程完全公开, 而且遵循 GPL 宣言, 用户可以在符合 GPL 的宣言条件下自由地使用及传播 RPM。

RPM 设计的目的有下面几点:

- (1) 方便的升级功能: rpm 让用户不用重新安装整个系统就可以对单个软件包进行升

级,当一个新的发行版本问世的时候,用户也不用重新安装,rpm 会替用户全面、自动、智能地升级系统,并且保留用户原先的配置文件,这就大大减小了用户维护系统的工作量。

(2)强大的查询功能:用户可以针对整个软件包的数据,或是某些特定的文件进行查询,也可以轻松地查出某个文件是属于哪个软件包,或是从哪里来的。RPM 文件本身是经过压缩的,但用户还是可以很容易地快速查询每个软件包的内容,因为在 rpm 软件包里,已经加入一段特殊的 binary header,它记录了全部查询时所需要的数据,这一点大大加快了查询速度。

(3)系统校验:当用户不小心删除了某个重要的文件,但是又不知道是哪些软件包需要这个文件,这时候就可以用 rpm 来查询已经安装的软件包中缺了哪些文件,是否需要重新安装。并且用户可以校验出安装的软件包是否已经被别人更改过。

(4)允许用户能够使用“纯净”的源代码:让用户取得“未经处理过的源代码”,同时再附上一份“补丁”程序,用户可凭借这些来完成程序的编译工作。这样的做法带来不少好处。举个例子,如果某个程序的新版本问世了,用户可能没有必要再重头开始做全部的编译工作,先观察“补丁”程序的内容,看看有哪些部分才是用户需要做的。这样就能让用户更清楚地知道新版本有哪些改进的地方。

#### 4.3.2 安装和使用 RPM

在 Redhat Linux 上是不用用户安装的,系统缺省就已经安装了 RPM,在别的发行版本上要安装 rpm 请执行下面的步骤:

首先下载 rpm 与相关的软件包,rpm 可从下列地址获取:

可执行文件: <ftp://ftp.redhat.com/pub/code/rpm/rpm-2.3.11.tar.gz>

源代码文件: <ftp://ftp.redhat.com/pub/code/rpm/rpm-2.3.11.src.tar.gz>

下列这两个软件包只有在由源代码文件安装 rpm 时才需要(请先检查是否已经安装了这两个软件包,不过 Slackware 缺省已经安装了这两个软件包):

zlib: <http://www.mathematik.th-darmstadt.de/~lehmann/png/zlib-1.0.4.tar.gz>

cpio: <ftp://prep.ai.mit.edu/pub/gnu/cpio-2.4.2.tar.gz>

##### 1. 由可执行文件包安装 rpm

(1)cd /

(2)tar xzvf rpm-2.3.11.tar.gz

(3)mkdir /var/lib/rpm /var/local /var/local/lib /var/local/lib/rpm

##### 2. 由源代码文件安装 rpm

(1)tar xvfz rpm-2.3.11.src.tar.gz

(2)cd rpm-2.3.11

./configure

如果这时候警告用户安装 zlib,用户就请先自行编译安装 zlib,再来执行 ./configure。

(3)编译程序,执行:

make

如果不出什么错误的话就开始安装(如果出错,请检查-L选项是否已经包括zlib所在路径):

```
make install
```

```
(4)mkdir /var/local /var/local/lib /var/local/lib/rpm /var/lib/rpm
```

这就安装完rpm了。下面简单介绍一下rpm的使用。

1. 用RPM安装软件包,最简单的命令如下:

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

下面rpm就会输出该软件包的名称,并且显示一个状态条。安装软件很简单,但是有时会给出一些出错消息:

#### • “Package Already Installed”

如果该软件包已经安装,用户就会看到下面的出错信息:

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

如果用户想强行安装就加上选项--replacepkgs这个选项强迫rpm重新安装这个软件。

#### • “Conflicting Files”

如果该软件包包含其他软件包安装过的文件,用户就会看到下面的出错信息:

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

如果用户想覆盖原先的文件,使用--replacefiles选项就可以。

#### • “Unresolved Dependency”

rpm的软件包能够查询该软件的“依赖”关系,即正确运行该软件需要其他哪些软件包,所以rpm在安装之前会先查询,如果系统没有安装需要的软件包,用户就会看到下面的出错信息:

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

要继续安装,用户就必须先安装相应的软件包,在上例中就是foo这个软件包,要不然即使--nodeps选项强行安装,不过笔者并不推荐这样做,这样安装的结果十有八九不能运行。

## 2. 用rpm反安装软件包

要反安装软件包foo,只需要下面这行命令:

```
$ rpm -e foo
```

注意:软件包名是foo,而不是rpm文件名“foo-1.0-1.i386.rpm”。

如果该软件包是别的软件包运行时候需要的,用户就会看到下面的出错信息:

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

如果用户要强行反安装,就使用 `--nodeps` 选项,不过这样相关的软件包就不能运行了。

### 3. 用 rpm 升级软件

升级软件类似于安装软件:

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

用户值得注意的是 rpm 会自动反安装相应软件包的老版本。如果老版本软件的配置文件同新版本的不兼容, rpm 会自动将其保存为另外一个文件,用户会看到下面的信息:

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

这样用户就可以自己手工去更改相应的配置文件。

另外如果用户要安装老版本的软件,用户就会看到下面的出错信息:

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

如果用户要强行安装就使用 `--oldpackage` 参数。

### 4. 查询软件包

用户可以用 `rpm -q` 在 rpm 的数据库中查询相应的软件, rpm 会给出软件包的名称, 版本, 发布版本号, 例如:

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

下面是查询时可以使用到的特定参数:

- a 查询目前系统安装的所有软件包。
- f 文件名 查询包括该文件的软件包。
- F 问-f 参数, 只是输入是标准输入(例如 `find /usr/bin | rpm -qF`)。
- p 软件包名 查询该软件包
- P 问-p 参数, 只是输入是标准输入(例如 `find /mnt/cdrom/RedHat/RPMS | rpm -qP`)。

下面是输出时的格式选择:

- i 显示软件包的名称、描述、发行、大小、编译日期、安装日期、开发人员等信息
- l 显示软件包包含的文件
- s 显示软件包包含的文件目前的状态, 只有两种状态: `normal` 和 `missing`。
- d 显示软件包中的文档(如 `man`, `info`, `README` 等等)。
- c 显示软件包中的配置文件, 这些文件一般是安装后需要用户手工修改的, 例如: `sendmail.cf`, `passwd`, `inittab` 等。

如果用 `-v` 参数就可以得到类似于 `ls -l` 的输出。

### 5. 用 rpm 校验软件包

用户可以用 rpm 来校验已经安装的软件包, rpm 可以校验文件大小、MD5 校验码、文件权限、类型和属主等信息。

下面是校验时的参数说明:

- a. 简单校验软件包中的文件是否存在:

```
rpm -V foo
```

- b. 校验软件包中的单个文件:

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

- c. 校验安装的所有软件包:

```
rpm -Va
```

- d. 比较一个源软件包和已经安装的软件包:

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

如果校验通过就没有任何出错信息, 如果有出错信息, 它的格式如下: 出错信息是一个 8 位的字符串, 如果是一个 “.”, 就代表没有问题, 下面是相应的字符的意义:

- S MD5 校验错
- S 文件大小错
- L 符号连接错
- T 文件修改时间错
- D 设备文件错
- U 用户名错
- G 组名错
- M 属性(包括读写权和文件类型)错

如果用户看到这些信息, 最好重新安装或者手工修复。

### 6. 实际使用技巧

(1) 用户可以通过 FTP 来安装软件包。如果用户能够连上网络, 想安装某个新的软件包时, 可以直接用它的 URI 地址来安装如:

```
rpm -i ftp://ftp.pht.com/pub/linux/redhat/rh-2.0-beta/RPMS/foobar-1.0-1.i386.rpm
```

也可以用网络来查询。

(2) 假如用户不小心误删了几个文件, 但不确定到底是哪些文件, 想对整个系统进行校验, 以了解哪些部分可能已经损坏, 可用:

```
rpm -Va
```

(3) 如果用户碰到一个认不出来的文件, 想要知道它是属于哪一个软件包的话, 可以这样做:

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

结果会得到下面的信息:

```
xjewel-1.6-1
```

(4) 如果用户得到一个 RPM 文件, 却不清楚它的内容, 可以这样做:

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

结果会得到下面的信息：

```

Name       : koules                      Distribution: Red Hat Linux Colgate
Version    : 1.2                        Vendor: Red Hat Software
Release    : 2                          Compile Date: Mon Sep 02 11:59:12 1996
Install date: (none)                   Compile Host: porky.redhat.com
Group      : Games                      Source RPM: koules-1.2-2.src.rpm
Size       : 614939
Summary    : SVGAlib action game with multiplayer, network, and sound support
Description:
This arcade-style game is novel in conception and excellent in execution.
No shooting, no blood, no guts, no gore. The play is simple, but you
still must develop skill to play. This version uses SVGAlib to
run on a graphics console.
```

(5)如果用户想了解某个 RPM 软件包会在系统里安装哪些文件,可以这样做:

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

结果会得到下面的信息：

```

/usr/doc/koules                               /usr/doc/koules/Icon2.xpm
/usr/doc/koules/ANNOUNCE                     /usr/doc/koules/Koules.FAQ
/usr/doc/koules/BUGS                         /usr/doc/koules/Koules.xpm
/usr/doc/koules/COMPILE.0S2                  /usr/doc/koules/README
/usr/doc/koules/COPYING                      /usr/doc/koules/TODO
/usr/doc/koules/Card                         /usr/games/koules
/usr/doc/koules/ChangeLog                    /usr/games/koules.svga
/usr/doc/koules/INSTALLATION                 /usr/games/koules.tcl
/usr/doc/koules/Icon.xpm                     /usr/man/man6/koules.svga.6
```

最后给出 rpm 的英文参数表：

RPM version 2.3.9

Copyright (C) 1997 - Red Hat Software

This may be freely redistributed under the terms of the GNU Public License

usage: rpm {--help}

rpm {--version}

rpm {--initdb: [-] [--dbpath <dir>]}

```

rpm {--install -i: [-v] [--hash -h] [--percent] [--force] [--test]
      [--replacepkgs] [--replacefiles] [--root <dir>]
      [--excludedocs] [--includedocs] [--noscripts]
      [--refile <file>] [--ignorearch] [--dbpath <dir>]
      [--prefix <dir>] [--ignoreos] [--nodeps]
      [--ftp proxy <host>] [--ftpport <port>]}
      file1.rpm ... fileN.rpm
```

rpm {--upgrade -U: [-v] [--hash -h] [--percent] [--force] [--test]}

```

[--oldpackage] [--root <dir>] [--noscripts]
[--excludedocs] [--includedocs] [--refile <file>]
[--ignorearch] [--dbpath <dir>] [--prefix <dir>]
[--ftp proxy <host>] [--ftpport <port>]
[--ignorecos] [--nodeps] file1.rpm ... fileN.rpm

rpm [--query -q] [--afpg] [-i] [-l] [-s] [-d] [-c] [-v] [-R]
[--scripts] [--root <dir>] [--refile <file>]
[--whatprovides] [--whatrequires] [--requires]
[--ftpuseport] [--ftp proxy <host>] [--ftpport <port>]
[--provides] [--dump] [--dbpath <dir>] [targets]

rpm [--verify -V -y] [--afpg] [--root <dir>] [--refile <file>]
[--dbpath <dir>] [--nodeps] [--nofiles] [--noscripts]
[--comd5] [targets]

rpm [--setperns] [--afpg] [target]
rpm [--setugids] [--afpg] [target]
rpm [--erase -e] [--root <dir>] [--noscripts] [--refile <file>]
[--dbpath <dir>] [--nodeps] [allmatches]
package1 ... packageN

rpm [--bld] [pleiba] [-v] [--short-circuit] [--clean] [--refile <file>]
[--sign] [--test] [--timecheck <s>] specfile

rpm [--rebuild] [--refile <file>] [-v] source1.rpm ... sourceN.rpm
rpm [--recompile] [--refile <file>] [-v] source1.rpm ... sourceN.rpm
rpm [--resign] [--refile <file>] package1 package2 ... packageN
rpm [--addsign] [--refile <file>] package1 package2 ... packageN
rpm [--checksig -K] [--nopgp] [--nomd5] [--refile <file>]
package1 ... packageN

rpm [--rebuilddb] [--refile <file>] [--dbpath <dir>]
rpm [--querytags]

```

用户可以在 RPM 的使用手册 `man` 里找到更详细的说明。

## 4.4 常见问题解答

### 1. 如何将 `man` 的内容存成文本文件？

以 `man tcsh` 为例：

- (1) `man tcsh | col -b > tcsh.txt`
- (2) `nroff -man tcsh.man | col -b > tcsh.txt`

将 `info` 变成文本文件, 以 `make` 为例：

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

### 2. 怎样给一批文件改名？

以改 `*.foo` 到 `*.bar` 为例

**C Shell:**

```
foreach f ( * .foo )
    set base = 'basename $f .foo'
    mv $f $base.bar
end
```

**Bourne Shell:**

```
for f in * .foo; do
    base = 'basename $f .foo'
    mv $f $base.bar
done
```

**C Shell:**

```
foreach i ( * .c )
    set j = 'echo $i | sed 's/\ .foo $/.bar/g''
    mv $i $j
end
```

**Korn Shell:**

```
for f in * .foo; do
    mv $f ${f%foo}.bar
done
```

**3. 如何将大写文件名变为小写文件名?****C Shell:**

```
foreach f ( * )
    mv $f 'echo $f | tr '[A-Z]' '[a-z]''
end
```

**Bourne Shell:**

```
for f in * ; do
    mv $f 'echo $f | tr '[A-Z]' '[a-z]''
done
```

**Korn Shell:**

```
typeset -ll
for f in * ; do
    l = $f
    mv $f $l
done
```

**4. 如何在 shell 脚本文件中让终端发声?**

使用下面的命令

```
echo -n '^G'
```

^G 是表示 ASCII 中的响铃字符。在 Emacs 中可以用 Ctrl-Q Ctrl-G 输入, 在 vi 中则用 Ctrl-V Ctrl-G 输入。

或者使用

```
echo '\007 \e'
```

## 5. 怎样寻找正在运行的程序的进程 ID?

使用下面的命令:

```
ps ux | awk '/程序名/ ! /awk/ {print $2}'
```

## 6. 如何查看用户有哪些后台作业?

```
lark ~ > jobs          查看用户有哪些后台作业
[1]-  Stopped          vi thesis.txt
[2]+  Stopped          joe work.c
[3]   Stopped          tetris
[4]   Stopped          telnet 159.226.43.21

lark ~ > %4            跳到编号为 4 的那个后台作业,即 telnet 159.226.43.21
(若改打 fg 则跳到编号为 2 的那个后台作业)
```

## 第 二 部 分

# Linux 系统管理与中文环境

---

- 5 Linux 系统管理
- 6 Linux 编程简介
- 7 Linux 网络管理与服务
- 8 Linux 上的中文环境
- 9 Linux 系统安全概述



# 5

## Linux 系统管理

### 5.1 root 账号

作为任何一种 UNIX 系统的管理员都需要有一定程度的责任心,对于 Linux 也是一样,即使这个系统只有一个用户。

系统管理的很多工作只有登录到 root 帐号里才能完成。在 UNIX 系统里,root 帐号很特殊,通常的文件保护和其他安全机制对 root 都不适用。root 可以获得和修改系统上的任何文件,不管文件的属主是谁。一般用户不能破坏文件系统或损坏他人的文件,但 root 没有这些限制。

为什么 UNIX 系统的安全性是首要特征呢?首先,UNIX 允许用户设定他们的文件能被别人利用到何种程度。用户可以用 chmod 命令设置文件的保护,使文件对同组的用户是可读的、可写的、可执行的,或使文件被彻底保护,不允许别人读、写或执行。这保证了数据的隐蔽性和完整性。例如,用户不愿意别人读他的私人邮件,或未经允许编辑他的一个非常重要的程序的源代码。

UNIX 的安全机制也防止了用户破坏系统。系统限制了对很多原始设备文件的操作,这些设备文件(在/dev 目录下)对应于系统的硬件,如硬盘等。如果普通用户直接读写磁盘驱动设备,他们的操作可能会带来灾难,即整个盘的内容被覆盖掉。相反,系统要求普通用户通过文件系统来进行磁盘操作,这样安全性通过文件保护得到了加强。

必须注意到并非所有对系统的“破坏”都是故意的。系统安全更大意义上是保护用户不受由他们自己的失误带来的损失,而不是充当系统警察。事实上,在很多系统上,安全性是很松的。UNIX 的安全设计可以促进为同一个项目工作的用户之间的数据共享。系统可以把这些用户安排到同一个用户组中,文件保护可以针对整个组的要求设置。例如,一个开发计划可能允许所有开发组成员对一系列文件有读写权利,但禁止其他用户修改这些文件。对每个用户个人的文件,用户自己可以决定文件的访问保护设成公开的还是隐蔽的。

UNIX 的安全机制还可以防止普通用户执行一些特定的操作,如在程序中使用一些特定的系统调用。例如,在 shutdown 等命令中使用一个系统调用可以使系统挂机。如果普通用户可以在他们的程序中调用这个函数,他们可能在任何时候不小心(或故意)使系统挂机。

很多情形下系统维护和升级操作需要超越 UNIX 的安全机制。这就是需要 root 帐号的原因。因为 root 账号不受以上种种限制,一个有经验的系统管理员不必担心文件保护或其他限制而可以顺利工作。

通常登录为 root 的方法是使用 su 命令。用户使用 su 命令可以成为另外一个用户的身分,如

```
su wang
```

将提示输入 wang 的口令,如果口令正确,则用户的 ID 被设成 wang 的 ID。单用 su 命令,系统将提示输入 root 的口令,如果输入口令正确,则用户的 ID 将被设成 root,用户将获得 root 的全权。一旦用户使用 root 账号完毕,可以用通常的方法退出,并回到用户真实的身分。

为什么不简单地从登录提示符直接登录成 root? 这在某些情况下是必需的。但在大多数情况下,最好是先登录进普通账号再使用 su 命令。在一个多用户系统上, su 命令的使用情况通常在系统日志文件(如/var/log/messages)中有一条记录:

```
Aug 22 08:55:32 lark su[8112]: + tty1 wang-root
```

这条信息表示用户 wang 曾成功地(“+”号,“-”号则表示登录失败)使用 su 命令登录到 root 账号。从登录提示符直接登录成 root 在日志文件中则没有记录,系统管理员无从知道哪个用户曾使用或试图使用 root 账号。一个系统有多个管理员时这一点很重要,这时经常有必要知道谁何时使用过 su 命令。

root 账号可以被看作一把双刃剑,有用也有潜在的危險。不恰当的使用可能会对系统造成不可恢复的破坏。最简单地,用 root 不小心执行 rm -fr /,将删除系统中所有的文件。例如,想删除一个无用的目录/usr/src/tmp 时,不小心在第一个斜线后多打进一个空格,则

```
rm -rf /usr/src/tmp
```

将产生灾难性的后果。

另一个常见错误是错用了某些命令的参数。例如,dd 命令通常用来把大块数据从一个地方拷贝到另一个地方。如果要把设备/dev/hda 上最靠前的 1024 字节的数据(通常是硬盘的分区表和引导记录)拷贝到文件/tmp/stuff 中,可以用

```
dd if=/dev/hda of=/tmp/stuff bs=1k count=1
```

但是,如果不小心把以上命令中 if 与 of 写颠倒了,文件/tmp/stuff 的内容将被拷贝到/dev/hda 的顶端。硬盘的分区表和引导记录将被一堆乱七八糟的东西覆盖,后果可想而知。

所以,在用 root 执行命令时,敲 Enter 之前都必须仔细审查输入的命令以确保正确。如果对命令的参数或定义不大明白,必须马上查找手册或在一个安全的环境里试一试。

通常 root 账号的提示符与普通账号不同。一般约定 root 账号的提示符包含字母“#”,普通账号的提示符包含字母“\$”或“%”。尽管提示符可以提示用户是否在使用 root 账号,但不小心把命令敲到错误的窗口或虚拟控制台中还是经常发生的。

像其他强有力的工具一样,root 账号也会被盗用。所以,系统管理员必须谨慎保护 root 的口令,如果必须告诉别人,只告诉值得信任的或对自己的行为负责的人。更重要的是如果一个系统只有一个人可以使用 root 账号,他可以清楚地知道系统是如何配置的,如果别的人可以修改重要的系统文件,系统管理员将无从知道系统配置的一些改变,系统上有些事情将不像他预想的一样工作。所以一个系统只有一个可以决定系统配置的管理员意味着他总可以掌握系统是如何工作的。

另外,让别人知道 root 的口令意味着很有可能其他人用 root 账号时会出现某种错误,尽管我们可以充分相信每个对使用 root 账号有经验的人,但任何人都可能会犯错误。如果只有一个系统管理员,使用 root 时犯错误的几率就会小些。

## 5.2 引导系统

引导 Linux 系统有几种方法。最通常的方法是从硬盘或软盘引导。一般情况下,系统安装程序已经配置了上述一种或两种引导方法。了解怎样配置系统引导方法对管理员来说是很重要的。

### 5.2.1 从软盘引导

Linux 引导软盘通常只包含一个内核映像,当用软盘引导系统时,这个内核映像被装载进内存。内核映像通常是用与 gzip 压缩程序相同的算法压缩过的,这使得一般要超过 1M 字节大小的内核只占几百 k 字节的磁盘空间。内核代码的一部分没有压缩;这部分包含把内核从磁盘映像解压缩并装载进内存的程序。所以实际上在引导时内核是自解压进内存的。

内核映像中存有一些参数,其中有一旦系统引导后作为根文件系统的设备名称。另一个参数是用作系统控制台的文本模式。所有这些参数可以用 rdev 命令修改。

内核启动后,它尝试挂在在内核映像中指定设备上的文件系统。这将作为根文件系统,即文件系统 /。当内核不能挂在指定设备上的文件系统时,它将停止运行,并产生一个致命错误信息。

内核映像中存储的根设备是硬盘上的根文件系统。这意味着一旦内核引导起来,它将挂上一个硬盘分区作为根文件系统,然后将所有的控制转给硬盘。一旦内核被装载进内存,它就停留在内存里,引导软盘就不再需要了(除非重启系统)。

很多 Linux 发行套件都会在安装系统时提示做一张引导软盘。当机器上有几种操作系统时,从硬盘引导有时是一件头痛的事(如设置 OS/2 或 Windows NT 的引导管理器可以引导 Linux 有些困难),这时从软盘引导是比较容易的方式。

从内核映像可以方便地制作引导软盘。在很多 Linux 系统上,内核存储在文件 /vmlinuz 中。但并不总是这样;有些 Linux 系统可能把内核存在 /boot/vmlinuz、/vmlinux 或 /image。(如果一个系统有多个内核映像,可以用 LILO 选择其中之一引导,见下节。)有些新安装的 Linux 系统甚至在硬盘上没有内核映像,而只是制作一张引导软盘。系统管理员总是可以编译自己的内核,定制内核只包含对应于自己现有硬件设备的驱动。

假设内核映像 in /vmlinuz,为制作引导软盘,第一步,用 rdev 命令设置根设备为 Linux 系统的根文件系统。在 root 账号下,用 rdev -h 可输出 rdev 的用法信息,有很多支持的选项,如指定根设备、交换设备和虚拟盘大小等。

rdev /vmlinuz 将给出内核中指定的根设备:

```
lark # rdev /vmlinuz
Root device /dev/hda1
```

如果不正确, Linux 的根文件系统实际在 /dev/hda3, 可以用以下命令:

```
lark # rdev /vmlinuz /dev/hda3
```

```
lark #
```

再执行一遍 `rdev /vmlinuz` 以确定已设置的正确性。

现在拿一张新的,已格式化过的软盘(可在 MS-DOS 中格式化或在 Linux 中用 `fdformat` 格式化)。用 `dd` 命令把内核映像拷贝到软盘上:

```
lark # dd if=/vmlinuz of=/dev/fd0 bs=8192
```

简单地说,这条命令拷贝输入文件(`if` 选项)/`vmlinuz` 到输出文件(`of` 选项)/`dev/fd0` (第一个软盘设备,使用块大小(`bs`)为 8192 字节。

这样引导软盘已经做好。关闭系统后重新用软盘引导。如果一切正常,则 Linux 系统将像往常一样引导并运行。

### 5.2.2 用 LILO 引导

LILO 是一个通用的引导管理器,可引导一台机器上安装的几种操作系统,包括 Linux。配制 LILO 的方法有很多种。这里我们将介绍两种最常用的方法:把 LILO 安装在硬盘的主引导记录里,及把 LILO 安装成只引导 Linux 的二级引导器。

LILO 是从硬盘引导 Linux 的最通常的方法。(“从硬盘引导”的意思是:内核本身储存在硬盘上,不需要引导软盘。但请注意,即使用软盘引导,在内核被装载进内存后,控制权也就转到了硬盘。)如果 LILO 安装在硬盘的主引导区,即 MBR 上,当硬盘引导时,它是最先被执行的代码。然后 LILO 就可以引导其他操作系统,如 Linux 或 MS-DOS,并允许用户在引导时选择其一。

但是,OS/2 和 Windows NT 都有它们自己的引导管理器,而且也占用着主引导区。如果机器上有这两种操作系统之一,为了从硬盘引导 Linux,必须把 LILO 安装成只引导 Linux 的二级引导器。在这种情况下,LILO 被安装在作为 Linux 根文件系统的分区的引导记录里,引导管理器软件(OS/2 的或 Windows NT 的)在用户需要引导 Linux 时执行 LILO。

我们将要看到,OS/2 和 Windows NT 的引导管理器在引导 Linux 时某种程度上是不合作的,这是它们设计的缺陷。所以,当用户绝对需要使用这两种引导管理器时,从软盘引导 Linux 会更方便一些。

另一方面,LILO 与 Windows 95 共同使用时要简单些。只要像设置引导 MS-DOS 一样设置引导 Windows 95 就可以了。但是如果在安装 LILO 后安装 Windows 95,Windows 95 的安装程序将覆盖第一个硬盘的主引导区,所以必须用 Linux 引导软盘来引导 Linux 以重新安装 LILO。

许多 Linux 发行套件在安装 Linux 系统时都有配置和安装 LILO 这一步。但是,用户最好亲手配置一下 LILO 以确保所有设置都是正确的。

#### 1. /etc/LILO.conf 文件

配置 LILO 的第一步是建立 LILO 配置文件,通常是 `/etc/LILO.conf`。(在某些系统中,可能叫 `/boot/LILO.conf` 或 `/etc/LILO/config`。)

下面看一个 LILO.conf 文件的例子,设置一些基本参数:

```
boot = /dev/hda
compact
delay = 50
install = /boot/boot.b
map = /boot/map
```

boot 行设置 LILO 应该把自己安装哪个设备的引导记录里。这里我们将把 LILO 安装在第一个非 SCSI 硬盘/dev/hda 的主引导记录里。如果从 SCSI 硬盘引导,设备名应类似/dev/sda。如果 boot 行给出的是一个分区名,如/dev/hda2,则 LILO 将作为一个二级引导器安装在这个分区。稍后将讨论这一问题。

compact 行告诉 LILO 要进行某种优化。LILO.conf 中一般应该包含这一行。同样,一般也应该包含 install 行和 map 行。install 行指定的文件将作为引导扇区在安装 LILO 时被写进主引导区,map 给出安装 LILO 时产生的“map 文件”,这些文件应该在/boot 目录中,有些系统可能把它们放在/etc/LILO 目录里。/boot/map 只在第一次安装 LILO 后才产生。delay 则定义了 LILO 在引导缺省的操作系统(/etc/LILO.conf 中的第一个操作系统段)前将等待多长时间,在这段时间内用户可以选择引导其他操作系统,50 表示 5 秒钟。

现在,对每一个想通过 LILO 引导的操作系统,在/etc/LILO.conf 中加上一段文本。有关 Linux 的一段将类似:

```
# Linux: 根分区/dev/hda2
image = /vmlinuz          # 内核
label = Linux             # OS 的名称(用在 LILO 引导菜单)
root = /dev/hda2          # 根分区
vga = ask                 # 引导时询问 VGA 的文本模式
```

image 行给出内核映像的名称,其子项包括:label,给出这个段的名称,它将用在 LILO 引导菜单中;root,指定 Linux 的根分区;vga,指定系统控制台所使用的 VGA 文本模式。

vga 的有效模式包括:normal(标准 80 × 25 显示)、extended(加长文本模式,132 × 44 或 132 × 60 等)、ask(引导时提示用户输入模式)、或一个整数(如 1,2,3)。整数对应当使用 ask 时可选择的模式序号。有哪些适用模式取决于显示卡,可用 vga = ask 来得到一个列表。

如果用户希望引导几种 Linux 内核,他可以为每一个内核增加一段 image...,这些 image 段唯一要求的子项是 label。如果在 image 段中没有指定 root 或 vga,用 rdev 写进核心映像的缺省值将起作用。如果指定了 root 或 vga,它们将覆盖内核中的缺省值。所以,在使用 LILO 引导 Linux 时,没必要用 rdev,LILO 配置文件将设置这些引导参数。

引导 MS-DOS 或 Windows 95 的一段类似:

```
# MSDOS/Windows 95: 分区在/dev/hda1
other = /dev/hda1         # 分区
table = /dev/hda          # 含分区表的设备
label = msdos             # OS 的名称(用在 LILO 引导菜单)
```

从 LILO 引导 OS/2 的一段与上段几乎相同(当然,必须用另一个 label)。

如果 MS-DOS 或 Windows 95 分区在第二个硬盘上,则必须在 MS-DOS 的 other...

一段中加上:

```
loader = /boot/any.d.b
```

对在第二个硬盘上的 OS/2 分区,必须加上:

```
loader = /boot/os2.d.b
```

还有很多 LILO 配置的选项,但以上介绍对大多数系统来说是足够的。

当/etc/LILO.conf 文件已经设置好后,使用 root 账号执行命令/sbin/LILO 将在主引导区安装 LILO。显示信息如下:

```
lark # /sbin/LILO
Added Linux
Added msdos
lark #
```

在出错时用 LILO 的 -v 选项可输出更多的诊断信息;用 -C 选项可以指定使用另外一个配置文件,而不用缺省的/etc/LILO.conf。

现在就可以关闭系统,重新引导,试一试新安装的引导管理器了。系统将首先输出“LILO”然后等待一段时间。缺省时将引导/etc/LILO.conf 中的第一个操作系统段。若要在/etc/LILO.conf 中的其他操作系统或内核,则在 LILO 等待时间内,敲 Control 或 Shift 或 Alt 键,在“LILO”后将弹出“boot:”,这时可敲入其他操作系统的 label,或敲 Tab 键,LILO 将输出/etc/LILO.conf 中定义的所有操作系统的 label,敲入其中之一,LILO 将引导所敲入的 label 对应的操作系统。

## 2. LILO 作为二级引导器

如果一台机器在使用 OS/2 或 Windows NT 的引导管理器,用户又不想让 LILO 占用主引导区,那么可以把 LILO 配置成二级引导器,即把它放在 Linux 根分区的引导记录里。这只要把/etc/LILO.conf 中的 boot...行改成 Linux 的根分区就可以做到。例如

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

将把 LILO 安装在/dev/hda2 的引导记录里,LILO 将只引导 Linux。

**注意:**这只对硬盘的主分区(不是扩展或逻辑分区)有效。

用这种方式引导 Linux, Linux 的根分区必须在分区表中被标记为“active”(用 Linux 或 MS-DOS 的 fdisk 命令)。引导系统时, BIOS 会读进“active”分区的引导记录以启动 Linux。

如果使用 OS/2 或 Windows NT 的引导管理器,就必须以这种方法安装 LILO,然后告诉引导管理器从硬盘的哪个分区引导 Linux。

LILO 可以与 OS/2 的引导管理器一起工作,但设置不太容易。因为 OS/2 的引导管理器不识别用 Linux 的 fdisk 创建的分区。解决办法是用 OS/2 的 fdisk 给 Linux 分区(用 Linux 的 fdisk 创建)一个名字。另一种方法是用 OS/2 的 fdisk 创建 Linux 分区,先格式化成 OS/2 可以识别的分区类型,如 MS-DOS FAT 分区;然后用 Linux 的 fdisk 设置分区类型为 Linux native 和 Linux swap,再在这些分区上安装 Linux,在 Linux 根分区的引导记录上安装 LILO。

解决 OS/2 的引导管理器问题的更好的方法是不用它,而用装在主引导区的 LILO 去

引导 OS/2。

### 3. 指定引导选项

用软盘引导安装 Linux 时,在引导提示符后可以输入很多引导参数,如:

```
hdx = cylinders, heads, sectors
```

给出了硬盘的几何参数。在某些机器的硬件配置下,可能在每次引导 Linux 时,为使硬件能被正确探测到,都必须给出这些参数。

如果用 LILO 从硬盘引导 Linux,可以在/etc/LILO.conf 中指定这些参数,而不必每次引导时都在提示符后输入它们。在 LILO.conf 文件的 Linux 段中,加上一行:

```
append = "hdx = 683,16,38"
```

这使系统工作和在引导提示符后输入 hdx = 683,16,38 一样。如果要指定多个引导参数,用一个 append 行就可以:

```
append = "hda = 683,16,38 hdb = 64,32,202"
```

这里我们分别指定了第一和第二个硬盘的几何参数。

**注意:**只有在引导时,内核没有探测到某些硬件才有必要使用这些引导选项。

其他一些可用的引导选项大都是用来处理硬件探测的。但以下选项有时可能也用得上:

single

以单用户模式引导系统,跳过所有的系统配置而直接在控制台上启动一个 root shell。

root = partition

尝试挂上分区 partition 作为 Linux 的根分区。将覆盖/etc/LILO.conf 中给出的值。

ro

以只读方式挂上根系统。通常是为了运行 fsck。

ramdisk = size

指定虚拟盘设备的大小为 size 字节,将覆盖/etc/LILO.conf 中给出的值。大多数用户不必关心如何使用虚拟盘,它一般是用在安装过程中。

vga = mode

设置 VGA 显示模式。将覆盖/etc/LILO.conf 中给出的值。可用模式是 normal, extended, ask 或一个整数。等价于 LILO.conf 中用的 vga = 值。

以上这些选项都可以在引导提示符后输入,或在/etc/LILO.conf 中用 append 选项指定。

LILO 有完整的文档描述所有配置选项。在很多 Linux 系统中,可以在/usr/doc/LILO 或/usr/src/LILO 中找到这些文档。

### 4. 删除 LILO

把主引导区上的 LILO 删除的最简单的方法是用 MS-DOS FDISK。命令

```
FDISK /MBR
```

将用一个有效的 MS-DOS 引导记录重写主引导区。

LILO 把系统最初的引导记录备份在文件/boot/boot.0300(对 IDE 设备)和/boot/boot.0800(对 SCSI 设备)。这些文件中包含安装 LILO 前的主引导区数据,可以用 dd 命

令替换相应硬盘的引导记录。如:

```
dd if=/boot/boot.0300 of=/dev/hda bs=446 count=1
```

拷贝文件/boot/boot.0300的头446个字节到/dev/hda。尽管文件大小是512字节,但只能拷贝头446字节到主引导区。

**注意:**必须谨慎使用这条命令,只有在不得已时才使用它,而且一定要确信/boot/boot.0300和/boot/boot.0800中存放的是所需要的引导记录。很多Linux发行套件可能在安装时会包含这些文件,如果是这样,必须在安装LILO前删除它们。

### 5.2.3 用Loadlin引导

Loadlin是一个DOS程序,用它可以从DOS提示符引导Linux内核。当用户不想从软件引导Linux,也不想把LILO安装到硬盘的引导区,并与其他操作系统的引导记录冲突时,可以采用在DOS下安装Loadlin程序,从DOS引导Linux。

另一种情况,Loadlin也是很有用的。如果系统上有一些特定的硬件,必须依靠DOS驱动把它设置到一个特定的状态时才能使用(如某些需要用DOS驱动设置成Sound Blaster兼容的声卡)时,用户可以先引导DOS,再加载声卡的DOS驱动,然后用Loadlin从DOS引导Linux,这样可以避免声卡在系统重起时被重新设置,声卡仍保持Sound Blaster兼容模式从而可以在Linux中使用。

Loadlin包含在Slackware的A系列中,在slackware/a4目录下有loadlin.tgz,将它展开在DOS分区的某个目录下,可以得到LOADLIN.EXE和其他文件。在RedHat中,LOADLIN.EXE放在./dosutils目录下。

除LOADLIN.EXE外,为从DOS引导Linux,用户还需要拷贝一个内核到DOS分区下。对于Slackware,用户可以在kernels/目录下各种内核的子目录中拷贝一个适当的zImage文件到DOS分区下;对于RedHat,用户可以把dosutils/autoboot/vmlinuz拷贝到DOS分区下。当然,用户也可以启动Linux,挂上DOS分区,将系统当前的内核文件(通常是/vmlinuz或zImage)拷贝到DOS分区下。

在DOS提示符下,我们可以用如下的方法来引导Linux(假设Linux内核放在C:\LOADLIN目录下):

```
C:\> loadlin c:\loadlin\vmLinux root=/dev/hda2 ro
```

root参数指定Linux的根文件系统的分区。如果用一个UMSDOS分区作为root,用:

```
C:\> loadlin c:\loadlin\vmLinux root=/dev/hda1 rw
```

如果使用放在A:驱动器内的Slackware安装用root软盘,用命令:

```
C:\> loadlin c:\loadlin\vmLinux root=/dev/fd0 rw load_ramdisk=1
```

## 5.3 系统启动和初始化

这一节我们将讨论当系统引导时都做了些什么。了解这个过程及涉及的文件对进行了解各种系统的配置非常重要。

### 5.3.1 内核引导信息

当我们从软盘或硬盘引导Linux时,内核在系统控制台上打印很多信息,这些信息在

系统启动完成后将分散存储在文件 `/var/log/syslog` 和 `/var/log/messages` 文件中：

```
LILO Loading Linux.....
Uncompressing Linux...done.
Now booting the kernel
Console: 16 point font, 400 scans
Console: colour VGA + 80x25, 1 virtual console (max 63)
pcibios_init : BIOS32 Service Directory structure at 0x000f9860
pcibios_init : BIOS32 Service Directory entry at 0xf0530
pcibios_init : PCI BIOS revision 2.10 entry at 0xf0560
Probing PCI hardware.
Calibrating delay loop.. ok - 333.41 BogoMIPS
Memory: 63104k/65536k available (788k kernel code, 384k reserved, 1260k data)
Swansea University Computer Society TCP/IP for NET3.034
IP Protocols: ICMP, ICMP, UDP, TCP
VFS: Diskquotas version dquot_5.6.0 initialized
Checking 386/387 coupling... Ok, fpu using exception 16 error reporting.
Checking 'hlt' instruction... Ok.
Linux version 2.0.30 (root@darkstar) (gcc version 2.7.2.2) # 3 Tue Jun 24 03:49:52 CDT 1997
Serial driver version 4.13 with no serial options enabled
tty00 at 0x03f8 (irq = 4) is a 16550A
tty01 at 0x02f8 (irq = 3) is a 16550A
Ramdisk driver initialized : 16 ramdisks of 4096K size
hda: QUANTUM FIREBALL ST4.3A, 4110MB w/81kB Cache, LBA, CHS = 524/255/63
hdb: GCD-R580B, ATAPI CDROM drive
ide0 at 0x1f0-0x1f7,0x3f6 on irq 14
Floppy drive(s): fd0 is 1.44M
Started kswapd v 1.4.2.2
FDC 0 is a post-1991 82077
md driver 0.35 MAX_MD_DEV = 4, MAX_REAL = 8
scsi : 0 hosts.
scsi : detected total.
Partition check:
hda: hda1 hda2 hda3 hda4
JAVA Binary support v1.01 for Linux 1.3.98 (C)1996 Brian A. Lantz
VFS: Mounted root (ext2 filesystem) readonly.
INIT: version 2.69 booting
Adding Swap: 128516k swap-space (priority -1)
Starting kerneld, version 2.0.0 (pid 14)
/etc/rc.d/rc.S: Testing filesystem status: Read-only file system
Parallelizing fsck version 1.10 (24-Apr-97)
/dev/hda1: clean, 29510/128520 files, 352255/514048 blocks
/dev/hda2: clean, 390/257040 files, 82613/1028160 blocks
/dev/hda3: clean, 3335/634880 files, 188649/2538270 blocks
```

```
Remounting root device with read-write enabled.
none on /proc type proc (rw)
/dev/hda3 on /home type ext2 (rw)
/dev/hda2 on /usr/local type ext2 (rw)
Updating module dependencies for Linux 2.0.30:
lpl at 0x0378. (polling)
CSLIP: code copyright 1989 Regents of the University of California
SLIP: version 0.8.4-NET3.019-NEWTTY-MODULAR (dynamic channels, max = 256).
PPP: version 2.2.0 (dynamic channel allocation)
PPP Dynamic channel allocation code copyright 1995 Caldera, Inc.
PPP line discipline registered.
loading device 'eth0'...
ne.c:v1.10 9/23/94 Donald Becker (becker@cesdis.gsfc.nasa.gov)
NE * 000 ethercard probe at 0x300: 00 40 05 4e dd d3
eth0: NE2000 found at 0x300, using IRQ 3.
INIT: Entering runlevel: 3
Going multiuser...
Swansea University Computer Society IPX 0.34 for NET3.035
IPX Portions Copyright (c) 1995 Caldera, Inc.
Appletalk 0.17 for Linux NET3.035
Mounting remote file systems...
Starting daemons: syslogd klogd portmap inetd lpd mountd nfsd
Starting sendmail daemon (/usr/sbin/sendmail -bd -q15m)...
Loading the iBCS module...
Running gpm...

Welcome to Linux 2.0.30.
lark login:
```

到“INIT: version 2.69 booting”一行前的信息都来自内核本身,打印哪些信息取决于系统上有什么硬件及内核中编译了哪些设备驱动。

首先 LILO 加载和解压缩内核并从内核引导系统。内核将探测控制台显示器类型、初始化 PCI BIOS、探测系统 PCI 设备(如显卡)及对 CPU 速度用 BogoMIPS 进行估算等。

接着,内核打印了系统内存信息:共 65536k(64M),剩余 63104k,内核本身占用约 2432k。

下一步,内核加载 TCP/IP 网络支持、磁盘空间限量支持、检查数学协处理器和打印内核版本信息。

现在内核开始探测各种硬件设备:串行口设备、硬盘、软盘驱动器和 SCSI 设备等。对探测到的设备打印 I/O 端口、中断、容量等信息,并打印硬盘分区情况。

在内核引导的最后,以只读方式挂上系统的根文件系统,并启动 init 程序。“INIT: version 2.69 booting”一行及以下的信息都是 init 程序调用系统启动文件时的运行信息。

### 5.3.2 init 和 inittab

init 程序通常放在 `/etc/bin` 或 `/sbin` 下,在大多数系统上是 `/sbin/init`。

init 是一个通用程序,它可产生新进程,并且在某些程序退出时能重起它们。如每一个虚控制台由一个 `getty` 程序运行。如果从某个虚控制台的登录进程退出, `getty` 进程也将退出, `init` 将重起一个新的 `getty`,使用户仍可从那个虚控制台登录。

init 还负责在系统启动时运行一系列程序和脚本文件。init 做的每件事情都由 `/etc/inittab` 文件控制。这个文件每一行的格式是:

```
code:runlevels:action:command
```

code 是用以鉴别这一行的单个或两个字符的序列,在此文件中应是唯一的。文件中的某些记录应该使用特定的 code 才能使系统工作正常。

runlevels 是此记录应被执行的运行级别的列表。运行级别是一个给出当前系统状态的数字或字母。例如,当系统的运行级别改变为 3, `/etc/inittab` 文件的记录中所有在 runlevels 域含数字 3 的记录将被执行。运行级别是对 `/etc/inittab` 文件的记录归类的好方法。例如,我们可能希望运行级别 1 是启动系统的最小配置。运行级别 2 除启动所有运行级别 1 的配置外,还启动网络配置。运行级别 3 除启动所有运行级别 1 和 2 的配置外,还启动拨号登录配置等。

command 域对应这个记录,init 要执行的命令。action 域是 init 执行 command 的方式,如:是只执行 command 一次,还是在它退出时重起它。

现在我们看一个 Slackware 的 `/etc/inittab` 文件的例子:

```
# Default runlevel.
id:3:initdefault:

# System initialization (runs when system boots).
si:S:sysinit:/etc/rc.d/rc.S

# Script to run when going single user (runlevel 1).
su:1S:wait:/etc/rc.d/rc.K

# Script to run when going multi user.
rc:2345:wait:/etc/rc.d/rc.M

# What to do at the "Three Finger Salute".
ca::ctrlaltdel:/sbin/shutdown -t5 -rfn now

# Runlevel 0 halts the system.
l0:0:wait:/etc/rc.d/rc.0

# Runlevel 6 reboots the system.
l6:6:wait:/etc/rc.d/rc.6
```

```
# Startagetty for virtual consoles 1 through 6
c1:1235:respawn:/sbin/agetty 38400 tty1 Linux
c2:1235:respawn:/sbin/agetty 38400 tty2 Linux
c3:1235:respawn:/sbin/agetty 38400 tty3 Linux
c4:1235:respawn:/sbin/agetty 38400 tty4 Linux
c5:1235:respawn:/sbin/agetty 38400 tty5 Linux
c6:12345:respawn:/sbin/agetty 38400 tty6 Linux

# Startup X Window System: xdm
x1:4:wait:/etc/rc.d/rc.4
```

首先我们看到,缺省的运行级别被设为 3。这条记录的 action 域是 initdefault。这是系统启动正常运行时的运行级别。

下一条记录告诉 init 系统启动时应当运行 /etc/rc.d/rc.S。这条记录的 action 域是 sysinit,说明这条记录将在系统启动后 init 开始运行时执行。/etc/rc.d/rc.S 是个 shell 脚本文件,它包含着基本的系统初始化命令,如激活交换空间、启动 kernel、检查(修补)并挂上文件系统或使系统钟与 CMOS 同步等,各种以可加载模块形式编译的设备驱动也在这时被加载进系统。如在上节的系统引导信息里,我们可以看到,并行口设备对 SLIP 和 PPP 协议的支持,以太网卡驱动等都被加载进系统等。

下一条记录告诉 init 当进入单用户模式时运行 /etc/rc.d/rc.K,它将杀死所有守护程序,并将系统置于单用户状态,但文件系统保持挂上状态。

接着我们看到,在多用户模式,即运行级别 2,3,4,5 时,init 都将执行 /etc/rc.d/rc.M。这个 shell 脚本文件执行所有在多用户模式下需要运行的程序。如:设置节点名、配置网络、启动各种网络守护进程、启动 syslogd,lpd,crond 等守护进程。rc.M 也将调用 /etc/rc.d/rc.local 文件,用户有时需要使某些程序在系统引导时运行,可以把运行这些程序的命令放入 rc.local 文件。

下一个记录,action 域是 Ctrlaltdel,表示如果用户使用 Ctrl + Alt + Del 组合键,系统将执行命令:

```
/sbin/shutdown -t5 -rfn now
```

“安全”关闭并重新引导。

标记为 c1 到 c6 的记录在第一到第六个虚控制台上运行 /sbin/agetty 程序。agetty 程序是 getty 程序的一个变种,它打开一个终端设备,为终端驱动设置各种参数,运行 /bin/login 程序并初始化一个登录进程。这六条记录的 action 域为 respawn,即在 agetty 进程退出后(用户退出登录),init 程序将重起它。

/etc/inittab 文件的最后一条记录是在运行级别 4 时,启动 X Window 系统的 X 显示管理器程序。在控制台上将是 X Window 系统的登录界面。用户不必在虚控制台上手动执行“startx”命令启动 X Window 系统。

**注意:**在运行级别 4 下,init 程序只在 tty6 上运行一个 agetty,用户可以用“Alt + F6”转换到 tty6 上使用这个虚控制台。

以上是 Slackware 下 /etc/inittab 的格式。但在别的 Linux 发行套件里,运行级别的机制可能不同。RedHat 就使用了不同的一套机制。RedHat 的 /etc/inittab 文件里有这么几

行:

```
10:0:wait:/etc/rc.d/rc 0
11:1:wait:/etc/rc.d/rc 1
12:2:wait:/etc/rc.d/rc 2
13:3:wait:/etc/rc.d/rc 3
14:4:wait:/etc/rc.d/rc 4
15:5:wait:/etc/rc.d/rc 5
16:6:wait:/etc/rc.d/rc 6
```

即对每个运行级别 N, init 执行“/etc/rc.d/rc N”, /etc/rc.d/rc 本身是个 shell 脚本文件, “/etc/rc.d/rc N”的意义是: 执行/etc/rc.d/rcN.d 目录下的一系列脚本文件。如“/etc/rc.d/rc 3”意味着执行/etc/rc.d/rc3.d 目录下的一系列脚本文件。而在/etc/rc.d/rcN.d 目录下的脚本文件的文件名格式是 Snnxxxx 或 Knnxxxx, nn 是在 00 和 99 之间的一个数, xxxx 是某种系统服务的名称。nn 决定了这些脚本文件被执行的顺序, nn 小的文件先执行, nn 大的文件后执行。以“S”开头的文件用来启动系统服务, 以“K”开头的文件用来杀死(或停止)系统服务。如 S69inet 用以配置网络, 而 K74syslog 则用以系统日志守护进程。系统启动时, 进入运行模式 3 时, 在/etc/rc.d/rc3.d 目录中所有以“S”开头的文件将被依序执行; 系统关闭时, 离开运行模式 3 前, 在/etc/rc.d/rc3.d 目录中所有以“K”开头的文件将被依序执行。

## 5.4 单用户模式

大多数情况下, Linux 运行在多用户模式下, 并允许用户登录。但有一种特殊的状态叫单用户模式, 在这种模式下, UNIX 在运行但没有登录提示符。单用户模式时的用户只能是超级用户(root)。事实上, 在安装系统时, 如果出现错误, 机器就进入单用户模式。单用户模式对某些常规的系统管理工作也很重要, 如检查损坏的文件系统。

在单用户模式下, 系统几乎是无用的; 很少的配置, 文件系统也没有挂上等。这是把系统从某些错误中恢复回来是很必要的。

**注意:** UNIX 在单用户模式下也是一个多进程的系统。很多程序可以同时运行。服务器可以在后台运行, 所以某些特别的功能如网络等可以工作。但如果系统带着不止一个终端, 只有控制台可以使用。X Window 系统也不能在单用户模式下运行。

## 5.5 关闭系统

关闭 Linux 系统比引导和启动系统要简单的多。但它并不是简单地按 reset 开关。像所有 UNIX 系统一样, Linux 在内存中缓冲了磁盘读写。这意味着除非在绝对必要时, 写盘才是被延迟的, 对磁盘的同一个块的多次读出可能实际上是从 RAM 中读的。这极大地提高了系统性能, 因为盘操作相对于 CPU 来说实在是太慢了。

问题是, 如果系统突然被去电或重启(如 reset), 则内存缓冲区的内容将不能写进硬盘, 数据将会丢失或损坏。在大多数系统上, 为防止系统崩溃所造成的严重破坏, 由/etc/

rc.d/rc.S 或 rc.sysinit 启动的 /sbin/update 程序,每 30 秒钟把缓冲区的内容写入磁盘(称为“同步”(sync))。但是,为彻底安全,系统在重新引导前必须经过一个“安全的”关闭。这不仅能保证磁盘缓冲区正确地同步,而且可以让所有正在执行的进程干净地退出。

shutdown 是中止或重启系统的一般命令。作为 root,执行:

```
/sbin/shutdown -r +10
```

将使系统在 10 分钟后重新启动。-r 开关表示系统在关闭后将重新启动,+10 是在关闭前等待的时间(分钟)。系统将向所有可用的终端发出一条警告信息,然后等待直到关闭时间的到来。如果管理员想在命令行加上自己的警告信息,可用

```
/sbin/shutdown -r +10 "Rebooting to try new kernel"
```

也可以指定一个绝对的关闭时间:

```
/sbin/shutdown -r 13:00
```

将在下午 1:00 重启系统。而:

```
/sbin/shutdown -r now
```

将立即重启系统(当然,在安全的关闭过程之后)。

用-h 开关代替-r 开关,系统将在关闭后中止而不重新启动,管理员就可以关闭电源而不必担心丢失数据。

正如我们在“init 和 inittab”一节里所说,可在/etc/inittab 里加上一条 ctrlaltdel 的记录,使 init 能捕捉组合键 Ctrl + Alt + Del 而执行 shutdown 命令。

**注意:**绝对不要用按系统电源开关或机器面板上的 reset 开关来重启系统。除非系统完全死机,一般都应该用 shutdown。一个多进程系统的巨大优点是,当一个程序死了时,用户仍可以转到其他的窗口或虚拟控制台去恢复。

shutdown 还有一些其他的选项。-c 开关将取消最近一次运行的 shutdown。(当然,也可以用 kill 命令杀掉那个进程,不过 shutdown -c 容易些。)-k 开关将只发出警告信息但并不真正关闭系统。

## 5.6 管理用户账号

在 UNIX 系统上,用户账号的概念有几个方面的意义。其中最主要的,因身份鉴别和安全原因,系统必须区别使用机器的不同个人,账号概念给系统提供了一种方法。每个用户有一个个人账号,每个账号有不同的用户名和口令。用户可以为自己的文件设置保护、允许或限制别人访问它们。系统中每一个文件都“属于”一个特定的用户,他可以设置这个文件的保护。用户账号被用来鉴定对系统的使用;只有有账号的那些人才可以使用机器。另外,账号被用来确认用户、保持系统日志和用发送者的名字标记电子邮件等。

除了个人账号外,系统上还有提供管理功能的用户。如系统管理员用 root 账号进行系统维护。这样的账号可以在进入个人账号后,用 su 命令登录。

系统中的另外一些账号不能被人交互使用。这些账号通常由系统 daemon 使用,它们经常必须以除 root 及个人账号外一个特殊用户的 ID 来访问某些文件。

系统管理员的工作之一就是为机器上的所有用户(真实的或虚拟的)创建和管理账号。这通常是一项轻松的工作,但懂得怎么去做也是很重要的。

### 5.6.1 passwd 文件

系统的每一个账号在 `/etc/passwd` 文件中有一行记录。这条记录给出了每个账号的一些属性,如用户名、真实名字和口令等。`/etc/passwd` 是 ASCII 文件,它必须是普通用户可读的,但只有 `root` 可写。

最初 `/etc/passwd` 这种一般可读性并没有什么大问题,每个人都可以读到加密的口令,但是,要想解开一个好的口令,现有的硬件太慢了。随着软硬件的发展,现在很多人不这么认为,于是产生了更严密的方法来保存口令文件,即 `shadow` 口令。文件 `/etc/passwd` 中原来放口令的地方放上一个“x”,而加密的口令放在另一个只有 `root` 才能读的文件 `/etc/shadow` 中。

`/etc/passwd` 文件中的每一条记录有如下格式:

```
login_name:passwd:UID:GID:user_name:directory:shell
```

`login_name` 用来区别不同用户,在同一系统中应是唯一的。对于个人账号,这就是用户的登录名。在很多系统上,它限制在 8 个字母或数字之内。例如: `wang,guest01`。

`passwd` 是用户的加密口令。在使用 `shadow` 口令的系统中,这一项放上一个“x”,而真正的加密口令放在 `/etc/shadow` 里。如果 `passwd` 域的第一个字母是“\*”,那么,这个账号就被查封了,系统不允许这个用户登录。

`UID` 是用户 ID,用来区别不同的用户,在同一系统中是唯一的。系统在内部处理进程和文件保护时使用 `UID` 域,处理数字比处理字符串更容易,也更紧凑。`login_name` 和 `UID` 都可用来识别用户,对系统来说 `UID` 更重要,对用户来说 `login_name` 更方便。

`GID` 是组 ID,表示用户缺省组别的一个整数。组文件是 `/etc/group`,参见下节。

`user_name` 包含有关用户的一些信息,如真实姓名、办公室地址和电话等。`mail` 和 `finger` 等程序用这些信息来标识系统的用户。

`directory` 是用户的主目录,由用户个人使用。当用户登录时,他的 `shell` 把 `directory` 作为用户的工作目录。

`shell` 是当用户登录时运行的程序名称,通常是一个 `shell` 的全路径名,如 `/bin/bash` 或 `/bin/tcsh`。

以上域中有几个是任选的。必需的是 `login_name`, `UID`, `GID` 和 `directory`。

下面看一看 `/etc/passwd` 中的两个记录:

```
root:x:0:0:/:root:/bin/bash
```

```
wang:x:1001:100:Wang Jun,Room 215,4321567,4321765:/home/wang:/bin/tcsh
```

第一条记录是 `root` 账号。首先,注意 `root` 的 `UID` 是 0。系统认为 `UID` 为 0 的用户是特殊的,不受通常的安全限制。`root` 的 `GID` 也是 0,这是一个惯例,表示 `root` 组。系统上很多文件属于 `root` 账号和 `root` 组。

在很多系统上,`root` 的主目录是 `/root`,或是 `/`。而且通常用 `Bourne-shell` 的变种(在本例中为 `/bin/bash`)作为 `root` 的 `shell`,当然也可以用 `C-shell`,但 `Bourne-shell` 和 `C-shell` 的定义不同,在它们之间切换使用时可能会出现混淆而导致错误。

第二个记录是一个普通的个人用户,用户名是 `wang`,`UID` 是 1001,`GID` 是 100,说明 `wang` 的组在 `/etc/group` 中编号为 100。在 `user_name` 一项中几个用“,”隔开的字符串或

数字的意义,我们可以用 `finger` 命令看一看:

```
lark% finger wang
Login: wang                      Name: Wang Jun
Directory: /home/wang           Shell: /bin/tcsh
Office: Room 215, 432-1567       Home Phone: 432-1765
On since Mon Aug 25 15:09 (CDT) on ttyt1 from siskin
No mail.
```

可以看出, `user_name` 中包含有用户的真实姓名、办公室地点、办公电话和家庭电话。

用户本人可以用 `chfn` 命令修改 `user_name` 项:

```
lark% chfn
Password:
Changing the user information for chenxy
Enter the new value, or press return for the default
```

```
Full Name [Wang Jun]:
Room Number [Room 215]:
Work Phone [4321567]: 64321567
Home Phone [4321765]: 64321765
```

lark%

个人用户的主目录一般在 `/home` 中,目录名与用户名相同。这是一个有用的约定,避免在寻找一个特定用户的主目录时遇到困难,但技术上可把主目录放在任何地方。但一般应遵循系统的目录格局。

**注意:**作为一个系统管理员,一般没必要直接去修改 `/etc/passwd` 文件。有一些程序可以帮助管理员创建和维护用户账号。

### 5.6.2 shadow 文件

`/etc/shadow` 文件包含用户账号的加密口令信息和口令的有效期限信息。每个用户有一条记录,每个记录用“:”隔开为 9 个域,分别是:

```
登录名
加密口令
口令上次更改时距 1970 年 1 月 1 日的天数
口令更改后不可以更改的天数
口令更改后必须再更改的天数(有效期)
口令失效前警告用户的天数
口令失效后距账号被查封的天数
账号被封时距 1970 年 1 月 1 日的天数
保留未用
```

加密口令域必须填上。加密口令由 13~24 个字符组成,所用字符在 64 个字符范围内(a 到 z, A 到 Z, 0 到 9, . 和 /)。参考 `crypt(3)` 查看如何解释这个字符串。口令域的第一个字符是“\*”,表示账号不能登录。

**注意:**如果口令更改后不可以更改的天数大于口令的有效期限,则用户不能更改自己的口

令。

/etc/shadow 文件中的信息覆盖在/etc/passwd 文件中的有关口令和口令有效期的所有信息中。

下面给出/etc/shadow 文件中的几行记录的例子：

```
root:Io22AJsAc48Bc:10090:0:::
ftp:*:9797:0:::
wanz:gERo4lrBzleUc:10088:0:180:7:::
```

对于 root 账号,没有给出口令的有效期,所以 root 账号不会因一段时间没有登录而被查封,但系统管理员要经常注意更新 root 的口令。ftp 账号的口令域是“\*”,表示不能登录为 ftp。wang 的口令的有效期是 180 天,警告期是 7 天,表示 173 天后,系统将提示用户 wang 修改口令,如果 7 天内不修改,系统在有效期过后将立即查封 wang 的账号。

用户修改口令用 `passwd` 命令,这个命令一般在 `/usr/bin` 里。普通用户只能修改自己的口令,而且必须回答老的口令。`root` 可以修改任何用户的口令,且系统不问老的口令。

### 5.6.3 group 文件

用户组是逻辑地组织用户账号集合的方便途径,它允许用户在组内共享文件。系统上的每一个文件都有一个用户和一个组的属主。使用 `ls -l` 命令可以看到每一个文件的属主和组,如:

```
lark% ls -l
-rwxrwx-r-- 1 wang  users    711 Aug 19 15:45 article.tex
lark%
```

这个文件属于用户 wang 和组 users。从文件保护可以看出, wang 对文件 article.tex 有读、写和执行的权力;任何 users 组的用户都有读和写的权力;其他用户只有读的权力。这并不意味着 wang 就在 users 组,它只表示 users 组的任何人按上列保护码都可以访问该文件。

每个用户至少属于一个组,即/etc/passwd 文件中的 GID 域。但一个用户可以属于多个组。类似/etc/passwd 文件,对系统上的每个组,在/etc/group 文件中有一行记录,记录的格式是:

group\_name:passwd:GID:user\_list

group\_name 是组的名字。passwd 是组的密码,允许不在这个组的用户用 newgrp 命令访问这个组。GID 是组系统用来区分不同组的 ID,在/etc/passwd 中的 GID 域用这个数来指定用户的缺省组。user\_list 是用“,”分开的用户名,列出是这个组的成员,但其在文件/etc/passwd 中的缺省 GID 不是这个组的用户。即缺省是这个组的用户不必列在这里。

一个/etc/group 文件的例子:

```
root:0:root
bin:1:root,bin,daemon
users:100:
friend:150:wang,lixin,chenx
```

root 和 bin 组是管理组,在本质上类似系统上使用的“虚拟”账号,系统上的很多文件属于组 root 和 bin。其他两个组是用户账号组。

passwd 域不经常用,但它与 newgrp 程序联系起来用时,允许不是特定组的用户获得那个组的 ID,如果那个用户输入了正确的口令。例如,使用命令:

```
lark% newgrp friend
Password: friend 组的口令
lark%
```

启动了一个组 ID 为 friend 的新 shell。如果 passwd 域是空的,或第一个字符是“\*”,用户如果试图用 newgrp 到那个组会得到一个 permission denied 错误。

但是,passwd 域很少使用,而且实际上是没必要的。事实上,很多系统没有提供设置组口令的工具。相反,要使一个用户成为很多组的成员,可以把用户名加到每一个附加的组的 user\_list 域。在以上的例子中,wang,lixin,chenx 除了是/etc/passwd 文件中指定的组的成员外,还是 friend 组的成员。

groups 命令可列出当前用户所属组的名称,如在 wang 的账号里。

```
lark% groups
users friend
```

表明 wang 属于 users 和 friend 两个组。

用户登录时,被自动地赋予/etc/passwd 中的组 ID,也自动成为/etc/group 中列出用户的组的成员。这意味着用户对系统上任何属于这些组的文件有“组的访问权”。即,这些文件的组保护位(用 chmod g+... 设置)适用于他(除非他又是文件的属主,这时属主保护位适用于他)。

那么,作为一个系统管理员,怎样设计用户的组呢?这实际上是一个风格的问题,取决于系统将如何使用。对只有少量用户的系统,可以设置所有个人用户的账号都属于单一的组,比如说 users。

**注意:**所有的系统组(系统安装时/etc/groups 文件中就带有的那些组)必须予以保留,各种守护进程和程序要用到它们。

如果系统上有很多用户,有很多种方法为他们编组。例如,教育机构可以把用户分成 students, faculty 和 staff 等组。软件公司可能把每个设计组分成一个组。

把用户加到一个附加的组中通常需要系统管理员去编辑/etc/group 文件。

#### 5.6.4 创建账号

创建一个用户账号需要几个步骤:加一个记录到/etc/passwd,创建用户的主目录,在用户主目录中设置用户的缺省配置文件(如.bashrc)。幸运的是,我们不必手动执行这些步骤,几乎所有的 Linux 系统都包含了一个程序 adduser 来完成用户账号的创建。

用 root 账号运行 adduser。只要在提示符后输入所要求的信息,很多提示都有合理的缺省值,简单地敲回车表示接收缺省值。

```
lark# adduser
```

```
Login name for new user (8 characters or less) []: wang
```

User id for test [ defaults to next available]:

Initial group for test [users]:

Additional groups for test []:

test's home directory [/home/wang]:

test's shell [/bin/bash]: /bin/tcsh

test's account expiry date (MM/DD/YY) []:

OK, I'm about to make a new account. Here's what you entered so far:

New login name: wang

New UID: [Next available]

Initial group: users

Additional groups: [none]

Home directory: /home/wang

Shell: /bin/tcsh

Expiry date: [no expiration]

This is it... if you want to bail out, hit Control-C. Otherwise, press  
ENTER to go ahead and make the account.

Making new account...

Changing the user information for test

Enter the new value, or press return for the default

Full Name []: Wang Jun

Room Number []:

Work Phone []:

Home Phone []:

Other []:

Changing password for wang

Enter the new password (minimum of 5, maximum of 8 characters)

Please use a combination of upper and lower case letters and numbers.

New password:

Re-enter new password:

Password changed.

```
Done...
```

```
lark #
```

新账号创建后, /etc/skel 目录中的文件被拷贝到用户的主目录中。/etc/skel 目录中含有为新账号准备的基本文件, 它们是缺省的配置文件。系统管理员可把其他一些认为新账号必须拥有的文件拷贝到/etc/skel 中。

现在新账号可以使用了。用户 wang 可以用 adduser 设定的口令登录。为保证安全性, 新账号在第一次登录后应立即用 passwd 修改口令。

### 5.6.5 删除和查封账号

删除一个用户账号比创建要容易些。要删除一个账号, 必须删除/etc/passwd 文件中此用户的记录, 删除/etc/group 文件中提及的此用户, 删除用户的主目录及其他由此用户创建或属于此用户的文件。例如, 如果此用户在/var/spool/mail 中有一个邮箱, 也要删除它。

userdel 命令将删除一个账号和账号的主目录。例如:

```
userdel -r wang
```

将删除账号 wang。-r 选项将删除用户的主目录。其他与此用户相关的文件, 如邮箱和 crontab 文件, 必须手工删除。寻找与某用户相关的文件的一个快速的方法是用命令:

```
find / -user username -ls
```

它将用 ls -l 列出所有属主是 username 的文件。当然, 在用这条命令时, /etc/passwd 文件中必须仍有 username 的记录。如果已删除了用户 username, 则必须用 -uid num 选项代替 -user username 选项, num 是已删除的用户的 UID。

因某种原因临时查封一个用户账号就更简单了。可把用户的记录从/etc/passwd 文件中去掉(保留主目录和其他文件不变), 或在/etc/passwd(或/etc/shadow)文件中在 passwd 域的第一个字符前加上一个 "\*", 如:

```
wang: *gERc4lxBzIeUc:10088:0:180:7:::
```

则禁止了此用户账号的登录。

### 5.6.6 修改用户账号

修改用户账号和组的属性比较简单的方法通常是直接修改/etc/passwd 和/etc/group 文件。很多系统提供了 usermod 和 groupmod 命令, 但直接手工修改文件似乎更容易些。

改变用户的口令用 passwd 命令, 它将提示输入一个口令, 加密后存在/etc/passwd 文件中。

如果要修改一个账号的 UID, 可以先修改这个账号在/etc/passwd 文件中记录的 UID 域, 然后把这个用户的所有文件改成新的 UID。如:

```
lark # chown -R wang /home/wang
```

将设置 wang 主目录下的所有文件为 wang 的新 UID。如果 ls -l 命令显示某些文件的属主是数字的 UID, 而不是用户名, 则表明此 UID 对应的用户不存在, 必须用 chown 命令作修正。

## 5.7 文件系统

对于 UNIX 系统,文件系统指的是某个格式化后用于存储文件的设备(如硬盘、软盘或 CD-ROM)。文件系统可以在很多允许随机访问的存储介质上建立。(注意:磁带只允许顺序访问,所以不能包含一个文件系统。)

文件存储的精确格式和手段并不重要,系统对它能识别的所有文件系统类型提供了一个通用界面。在 Linux 下,文件系统类型包括:二级扩展文件系统或 ext2fs,用于存储 Linux 文件;MS-DOS 文件系统,允许 Linux 访问 MS-DOS 分区和软盘上的文件;其他文件系统,包括 CD-ROM 使用的 ISO 9660 文件系统。

每种文件系统类型存储数据的基本格式都是不一样的。但是,在 Linux 下访问任何文件系统时,系统都把数据整理成在一个目录树下的文件,连同我们熟悉的文件的属主、组 ID、保护位和其他特征。

事实上,属主、保护等信息只有那些能存储 Linux 文件的文件系统类型才能提供。对于没有存储这些信息的文件系统类型,用来访问这些文件系统的内核驱动程序会“伪造”这些信息。例如,MS-DOS 文件系统没有文件属主的概念,但所有文件都显示成属主是 root。用这种方法,在一定层次上,所有文件系统类型都很相似,每个文件都有一定的属性。至于这些属性是否真的在文件系统底层被使用是另外一回事了。

**注意:**作为系统管理员,应该知道怎样在软盘上创建文件系统以存储 Linux 文件和怎样在硬盘上添加新的文件系统,应该知道一旦数据遭到破坏,怎样用各种工具检查和恢复文件系统,还应该知道访问在软盘和 CD-ROM 上的文件系统所要用的命令和有关文件。

### 5.7.1 文件系统类型

下面将介绍 Linux 内核现在支持的文件系统类型。新的文件系统类型仍在不断地被加入系统。我们可以查出系统的内核现在能支持哪几种文件系统。在内核源程序树中,目录/usr/src/Linux/fs中的每一个子目录名都是内核支持的文件系统类型。我们可以在编译内核时可选择支持哪些文件系统类型。

现在我们对当前 Linux 内核支持的文件系统类型逐一简单介绍。

#### minix

是在 Minix 操作系统中使用的文件系统。Linux 最初就是在 Minix 下交叉编译的。但它有一些缺点:最大只能使用 64M 的分区、短文件名和单一时间标签等。现在还可用在软盘和内存虚拟盘上。

#### ext

是 minix 文件系统中一个精心设计的扩展,但现在已经被扩展的文件系统的第二版(ext2)取代,并将最终从内核中去掉。

#### ext2

是 Linux 使用的高性能磁盘文件系统,既用于固定盘,又用于可拆装介质。二级扩展文件系统是扩展文件系统(ext1)的扩展。ext2 在速度和 CPU 使用上是 Linux 支持的最高

效的文件系统。支持 256 字符的文件名,最大 4T 字节的文件系统大小。

xiafs

最初被设计为对 Minix 文件系统的、稳定的、安全的扩展。它在没有增加过度的复杂性的情况下提供了最基本的和最常用的特征。xia 文件系统现在已不再被开发和维护,也很少使用了。

msdos

是 DOS、Windows 和某些 OS/2 计算机使用的文件系统。msdos 文件名不能长于 8 个字符的基本名加上一个“.”和 3 个字符的扩展名。

umsdos

是 Linux 使用的扩展的 DOS 文件系统。它在 DOS 文件系统的基础上增加了对长文件名、UID/GID、POSIX 文件保护和特殊文件(设备、命名管道等)的支持,但又保持与 DOS 的兼容。被用于在一个 MS-DOS 分区的目录中安装 Linux。这对想试一试 Linux 的新用户是很有用的,因为不必重新分区。

vfat

是微软的 Windows 95 和 Windows NT 使用的扩展的 DOS 文件系统。VFAT 在 MSDOS 文件系统下增加了对长文件名的支持。

proc

是一种伪文件系统,被用来作为连接内核数据结构的界面(而不是读取和解释/dev/kmem)。特别是,它不占磁盘空间。如果我们看一下/proc 目录,就会发现其中有一些“文件”和“目录”,其内容随时间不断变化。内核通过/proc 文件系统提供系统统计数据和进程信息,当用户试图访问/proc 中的任何文件时,内核可以识别并产生满足用户读请求的某些数据。这些“文件”和“目录”都不是放在磁盘上的,它们是内核制作的,以方便 ps 和 top 等程序访问这些数据。

iso9660

是符合 ISO 9660 标准的 CD-ROM 文件系统类型。有几种类型:

(1)High Sierra:ISO 9660 成为标准前 CD-ROM 文件系统的先驱。在 Linux 支持的 iso9660 文件系统内,它自动被识别。像 MS-DOS 一样,这种文件系统限制了文件名长度,只存储了有限的文件信息;

(2)Rock Ridge:Linux 也支持由 Rock Ridge 交换协议给出的系统使用共享协议记录,它们向 UNIX 节点进一步描述 iso9660 文件的文件,提供了诸如长文件名、UID/GID、POSIX 保护和设备等信息。结果是,在 MS-DOS 下访问 ISO 9660 CD-ROM 时显示 8.3 格式的文件名,在 Linux 下将给出“真正的”,完整的文件名。在 Linux 支持的 iso9660 文件系统内,Rock Ridge 自动被识别。

hpfs

是 OS/2 使用的高性能文件系统。由于缺乏应有的文档,在 Linux 下这种文件系统是只读的。

sysv

是 SystemV/Coherent 文件系统 in Linux 上的实现,实现了 Xenix FS、SystemV/386 FS 和 Coherent FS。这三个文件系统实际上是用同一个内核驱动程序处理的,只不过使

用的参数略有不同。

nfs

是用以访问远程机器上的磁盘的网络文件系统。

smb

是支持 Windows for Workgroups, Windows NT 和 Lan Manager 使用的 SMB 协议的网络文件系统。使用 smb 文件系统需要一个特殊的 mount 程序, 在 ksmbfs 包中, 可从以下地址下载:

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

对于 2.1.70 以后版本的内核, 可以使用 Samba 软件包中的 smbmount 和 smbmount 程序在 Linux 中挂卸 smb 文件系统。参见 7.7 节“Linux 和 Windows 95/NT 的文件和打印机共享”。

ncpfs

是支持 Novell NetWare 使用的 NCP 协议的网络文件系统。使用 ncpfs 文件系统需要特殊的程序, 可从以下地址下载:

`ftp://Linux01.gwdg.de/pub/ncpfs`

### 5.7.2 挂卸文件系统

在 Linux 下, 要访问任何文件系统, 必须先把它挂在一个确定的目录上。这使系统的文件看起来像是那个目录里, 我们就可以按通常的方法访问它们。

mount 命令用来挂上文件系统, 通常必须由 root 来执行。这条命令的格式是:

```
mount -t type device mount-point
```

type 是上节给出的文件系统类型的一种, device 是文件系统所在的物理设备(/dev 中的设备文件), mount-point 是挂上这个文件系统的目录。在执行 mount 之前必须先创建这个目录。

例如, 在 /dev/hda2 上有一个二级扩展文件系统, 把它挂到目录 /mnt 上用命令:

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

如果一切顺利, 就可以在目录 /mnt 下访问这个文件系统。

同样, 挂在软盘上的文件系统用命令:

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

这使使得在 /mnt 下可以访问 MS-DOS 格式的软盘上的文件。

有许多 mount 命令的选项可以用 -o 开关指定。例如, MS-DOS 和 ISO 9660 文件系统支持把文本文件从 MS-DOS 格式(每行尾有 CR-LF)转换到 UNIX 格式(每行尾只有一个换行符)的“自动转换”功能。用命令:

```
mount -o conv=auto -t msdos /dev/fd0 /mnt
```

对不含有代表可执行代码的扩展名(如 .exe 和 .bin 等)的文件自动执行这种转换。

mount 一个常用的选项是 -o ro(或等价的 -r), 这个选项将把文件系统挂为只读的。所有试图对这个文件系统进行的写操作都会得到一个“permission denied”的错误。对某些介质, 把文件系统挂成只读是必要的。如 CD-ROM 是不可写的, 如果不用 -r 选项挂一个 CD-ROM 时, 系统会把 CD-ROM 挂上, 但会给出如下信息:

```
mount: block device /dev/cdrom is write-protected,mounting read-only
```

挂一个写保护的软盘时也会出现类似信息。

卸下一个文件系统有两方面的结果：一是使系统缓冲区的内容与磁盘上文件系统的实际内容同步；二是使文件系统从作为 mount-point 的目录脱离，文件系统将不能再使用。这时就可以在同一 mount-point 挂上别的文件系统。

卸下文件系统用 umount 命令，如用：

```
umount /dev/fd0
```

卸下/dev/fd0(软盘)上的文件系统。同样，卸下任何一个现在挂在特定目录上的文件系统都使用类似：

```
umount /mnt
```

的命令。

**注意：**可拆装的介质，如软盘、CD-ROM 等，在它们挂上的时候不能从驱动器上拿走或换成另外一张。否则将使关于这些设备的系统信息与实际情况不再同步，会带来无穷无尽的麻烦。所以，当需要换软盘或 CD-ROM 时，要先用 umount 命令卸掉它，然后再把盘从驱动设备上拿走。

从软盘上的文件系统读或写也像硬盘一样利用内存的缓冲区。这意味着当从软盘上读或写时，不一定马上就有盘操作。系统非同步地处理软盘上的 I/O，只是在绝对必需的时候才读写数据。所以，当往软盘上拷贝一个小文件时，如果驱动器的灯没亮，不用担心，数据最终会写到盘上。用：

```
sync
```

命令会迫使系统立刻把所有文件系统缓冲区的内容写到盘上。卸下一个文件系统时会自动这样做。

如果系统管理员希望让普通用户也可以挂上或卸下某些设备，有两种方法。一种方法是，在/etc/fstab 文件中给某个设备增加 user 选项，允许指定的用户对指定设备使用 mount 和 umount 命令。另一种方法是，使用 Linux 上的 mount 前端程序，这些程序在运行时有效 UID 为 root，这样普通用户可以挂上特定设备。一般来说，不应该允许普通用户去挂上或卸下硬盘分区，但 CD-ROM 和软盘应该是普通用户可以使用的。

尝试挂上一个文件系统时，可能会出现若干种错误。但不幸的是，mount 命令对很多问题会给出同样的错误信息：

```
mount: wrong fs type,/dev/cdrom already mounted,/mnt busy,or other error
```

wrong fs type 很明显，用户可能指定了一种错误的文件系统类型。缺省是 iso9660，如果用户没有指定 -t 选项，或指定 auto 类型，则系统将探测 superblock 以确定文件系统类型。

device already mounted 意思是这个设备已被挂在另一个目录上了。用 mount 命令不带选项可以查看当前哪些设备被挂在哪儿：

```
lark # mount
/dev/hda1 on /msdos type msdos (rw)
none on /proc type proc (rw)
/dev/hda2 on / type ext2 (rw)
/dev/hda3 on /usr/local type ext2 (rw)
```

```
/dev/cdrom on /cdrom type iso9660 (ro)
```

我们看到系统挂上了三个硬盘分区,其中两个的类型是 ext2,一个是 msdos;还有挂在 /cdrom 的一个 CD-ROM 和 /proc 文件系统。每一行的最后一项表示文件系统挂上的选项,如 (rw) 表示读写模式, (ro) 表示只读模式。

mount-point busy 错误是比较奇特的。它一般表示在有些操作发生在 mount-point 目录下阻碍了在那儿挂上文件系统。如在这个目录下有一个打开的文件,或某进程的当前工作目录在 mount-point 下。使用 mount 命令前,要确定 root 的当前工作目录不在 mount-point 下;先用 cd / 到最上层目录去再做 mount。mount-point busy 错误也可能是因为已经有一个文件系统挂在 mount-point 下,用 mount 命令不带选项查看一下。

other error 没什么用。mount 失败还有其他一些原因。如果文件系统有某些数据或介质错误, mount 可能会报告说它不能读文件系统的 superblock, superblock 在类 UNIX 文件系统内是存储文件信息和文件系统整体性质的部分。如果试图挂上一个 CD-ROM 或软盘,但驱动器里没有 CD-ROM 或软盘,则将会有如下错误信息:

```
mount: /dev/cdrom is not a valid block device
```

如果当挂上或访问某文件系统时遇到问题,也可能是文件系统上的数据损坏了。在 Linux 里有一些工具可以帮助修补某些文件系统类型。参见 5.7.4 节“检查和修补文件系统”。

系统在引导时会自动挂上一些文件系统。在 /etc/fstab 文件里列出了在引导时必须挂上的文件系统名称。这个文件中的每一行的格式如下:

```
device mount-point type options
```

这里, device, mount-point 和 type 与它们在 mount 命令中的意义相同, options 是用在 mount 命令的 -o 开关后的选项,各选项之间用“,”隔开。

下面是一个 /etc/fstab 文件的例子:

| # device  | directory  | type    | options  |
|-----------|------------|---------|----------|
| /dev/hda1 | /msdos     | msdos   | defaults |
| none      | /proc      | proc    | defaults |
| /dev/hda2 | /          | ext2    | defaults |
| /dev/hda3 | /usr/local | ext2    | defaults |
| /dev/hdb  | /cdrom     | iso9660 | ro,user  |
| /dev/hda4 | swap       | swap    | defaults |

上面文件的最后一行是一个交换分区,将在“交换空间”一节里介绍。/dev/cdrom 一行中的 user 选项表示除 root 外的其他用户也可以挂上这个文件系统。这个选项如果存在,一个用户可以执行命令:

```
mount /cdrom
```

挂上这个设备。注意如果在使用 mount 命令时只给出了 device 和 mount-point 两个参数之一, mount 将在 /etc/fstab 中搜寻相应的 device 或 mount-point,用 /etc/fstab 中列出的参数来挂上文件系统。这使挂在 /etc/fstab 文件中列出的文件系统较为简单。

defaults 选项应当用于大多数文件系统。它包含了其他一些选项,如 rw(读写), async(在文件系统和内存缓冲区之间非同步的 I/O),等等。除非有特别的需要而必须修改这些参数,否则应对大多数文件系统使用 defaults,对只读的设备如 CD-ROM 使用 ro。

`mount -a` 命令将挂上所有在 `/etc/fstab` 文件中列出的文件系统。这个命令在系统启动时由 `/etc/rc.d` 目录中的一个描述文件(一般是 `rc.s` 或 `rc.sysinit`)执行。这样,当系统启动完成后,所有在 `/etc/fstab` 中的文件系统,如硬盘分区、CD-ROM 等都已挂上并可以使用了。

这里有一个例外:根文件系统。挂在 `/` 上的根文件系统通常包含文件 `/etc/fstab` 和 `/etc/rc.d` 下的描述文件。为了能访问这些文件,内核本身在引导时必须直接挂上根文件系统。包含根文件的设备已被做进内核映像中并且可以用 `rdev` 命令改变。引导时,内核将尝试几种文件系统类型(先是 `minix`,接着是 `ext2`,...),把内核中指定的设备挂成根文件系统。如果引导时,内核打印出如下错误信息:

```
VFS: Unable to mount root fs
```

则表明可能是:内核中的根设备不正确;或,内核内没有编译进支持根设备的文件系统类型的代码,这通常只发生在用户用自己编译的内核启动时;或,根设备在某些方面被破坏了。在以上情形下,内核将不能引导下去。参见本章其他节以解决这些问题。

想挂上一个文件系统并不需要把它列入 `/etc/fstab` 文件,但如果想在引导时就挂上或使用 `user` 选项,就必须把它列入 `/etc/fstab` 文件。

### 5.7.3 创建文件系统

文件系统可以用 `mkfs` 命令创建。创建文件系统类似于“格式化”一个分区或软盘,之后就可以存储文件。

每个文件系统类型都有与自己相联系的 `mkfs` 命令,例如, `mkfs.msdos` 用于创建 MS-DOS 文件系统, `mkfs.ext2` 用于创建二级扩展文件系统。 `mkfs` 程序是一个前端程序,用它调用各种类型的 `mkfs` 来创建相应类型的文件系统。

**注意:**另有一些命令如 `mke2fs` 和 `mkdosfs`,它们分别等价于 `mkfs.ext2` 和 `mkfs.msdos`。事实上, `mkfs.ext2` 和 `mkfs.msdos` 分别是指向 `mke2fs` 和 `mkdosfs` 的符号连接。用 `mkfs.fs-type` 类型的命令只是使前端程序 `mkfs` 执行创建各种类型文件系统的程序更简单些。

用 `mkfs` 命令创建文件系统的格式是:

```
mkfs -t type device blocks
```

`type` 是要创建的文件系统的类型, `device` 是要创建文件系统的设备(如 `/dev/fd0`, 对软盘来说), `blocks` 是文件系统的大小,以 1024 字节为单位(1 块)。

例如,在软盘上创建一个 `ext2` 文件系统:

```
mkfs -t ext2 /dev/fd0 1440
```

这里, `blocks` 是 1440, 指定了一个 1.44MB 的高密 3.5 英寸软盘。用 `-t msdos` 创建 MS-DOS 软盘。

现在,我们就可以挂上这张软盘,拷贝数据到上面,等等。注意在把软盘从软驱上拿走前,要用 `umount` 命令卸下它。

创建文件系统将删除对应物理设备(软盘、硬盘分区等)上的所有数据。 `mkfs` 一般在创建文件系统之前不提示用户,所以在创建一个新的文件系统前要确定自己在做什么。

在硬盘分区上创建文件系统与上面完全类似,但要记住以分区名,如 `/dev/hda2` 为

device,而不要在类似/dev/hda的设备上创建文件系统,这是指整个盘,而不是硬盘上的一个分区。创建分区用命令 fdisk。

在硬盘分区上创建文件系统时要明确给出的 device 和 size 是正确的。如果 device 错了,可能会破坏现有文件系统的数据。如果 size 错了,可能会覆盖其他分区上的数据。size 应该与 Linux 的 fdisk 给出的分区大小相同。

在软盘上创建文件系统时,最好先做一下低级格式化。这将把扇区和轨道信息写入软盘,使用设备/dev/fd0或/dev/fd1就可以自动探测到软盘的容量大小。做低级格式化的一种方法是用 MS-DOS 的命令 format,另一种方法是用 Linux 的命令 fdformat。例如,格式化第一个软驱内的软盘用以下命令:

```
lark # fdformat /dev/fd0h1440
Double-sided, 80 tracks, 18 sec/track. Total capacity 1440 kB.
Formatting ... done
Verifying ... done
```

fdformat命令用-n选项将跳过验证这一步。

每个特定文件系统类型的 mkfs 命令都支持一些有用的选项。-c 选项使在创建文件系统时检查物理介质损坏的块。如果发现损坏的块,它们被标记出来,这样当往文件系统上写文件时将不使用这些块。要使用对每个特定文件系统类型的选项,把它们放在 mkfs 命令的 -t type 选项后,如:

```
mkfs -t type -c device blocks
```

参考对各种文件系统类型的 mkfs 命令的在线手册查看可使用的选项,一般它们都支持 -c 选项用于检查坏块,-v 选项用于长输出。

如果系统上没有安装 mkfs.type,则使用 mkfs 创建 type 型的文件系统通常会失败。事实上,几乎所有 Linux 支持的文件系统类型都有一个对应的 mkfs.type 可以使用。

如果在使用 mkfs 时遇到问题,多半是因为 Linux 在访问物理设备时存在问题。对于软盘,可能因为软盘坏了。对于硬盘,问题就严重了。例如,在内核里的硬盘设备驱动程序可能在读硬盘时遇到了问题。这可能是硬件问题,也可能只是因为硬盘的几何参数设置错误。

### 5.7.4 检查和修补文件系统

有时有必要检查 Linux 文件系统的一致性,如果有错误或数据丢失要加以修补。这些错误通常来自系统崩溃或突然停电,这时内核不能使文件系统缓冲区的内容与磁盘的内容同步。在大多数情况下,这种错误是比较小的。但如果系统崩溃时正在写一个大文件,这个文件将丢失,没有对应于它的文件记录,但分配给它的块却仍标记为“使用中”。另一种情形是,无意中往硬盘驱动设备(如/dev/hda)或某一分区直接写入数据也会导致错误。

fsck 程序的功能是检查文件系统并修正错误。像 mkfs 一样,fsck 也是指定文件系统类型的 fsck.type(如 fsck.ext2,fsck.msdos 等)程序的前端程序。(类似 mkfs.ext2,fsck.ext2 也是指向 e2fsck 的一个符号连接,当没有安装 fsck 前端程序的时候两个都可以使用。)

fsck 的使用非常简单,命令格式为:

```
fsck -t type device
```

这里,type 是要修补的文件系统的类型,device 是文件系统所在的设备(硬盘分区或软盘)。

例如,要检查在/dev/hda2 上的一个 ext2 文件系统,我们可以这样做:

```
lark# fsck -t ext2 /dev/hda2
Parallelizing fsck version 1.10 (24-Apr-97)
e2fsck 1.10, 24-Apr-97 for EXT2 FS 0.5b, 95/08/09
/dev/hda2 is mounted. Do you really want to continue (y/n)? Yes

/dev/hda2 was not cleanly unmounted, check forced.
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
Fix summary information < y > ? Yes

Block bitmap differences: + 500101 -500102. FIXED
Inode bitmap differences: + 124444 -124487. FIXED
```

```
/dev/hda2: * * * * * FILE SYSTEM WAS MODIFIED * * * * *
```

```
/dev/hda2: 5833/257040 files (0.5% non-contiguous), 162846/1028160 blocks
```

首先应注意到系统在检查一个挂上的文件系统前要请求确认。如果文件系统是挂上的,fsck 发现并修正了任何错误,系统就必须重新引导。这是因为 fsck 所做的改变可能并没有传回系统关于文件系统结构的内部信息中去。一般说来,最好不要去检查挂上的文件系统,而要把它先卸下再检查。

但是,根文件系统却不能先卸下再检查。检查根文件系统的一种方法是,用软盘引导,并使用软盘上的根文件系统,例如用许多 Linux 发行套件的安装软盘(boot/root 盘)。这样,硬盘上的根文件系统并没有挂上,这时可以检查它。

另一种检查根文件系统的方法是把它挂成只读的,即可以在 LILO 引导提示符后用 ro 选项。但是,系统配置的其他部分(如引导时/etc/init 执行的程序)需要根文件系统是可写的。所以不能按正常的方式引导系统,否则这些程序将执行失败。必须使系统引导成单用户模式(使用引导选项 single),这避免了引导时附加的系统配置。在检查过根文件系统后就可以按正常方式引导系统。

为使根文件系统挂成只读方式,除了用引导选项 ro 外,也可以用 rdev 命令在内核映像中直接设置只读标志。

很多 Linux 系统在启动时自动检查文件系统。这通常是在/etc/rc.d/rc.S 文件中执行 fsck。fsck 完成后,rc.S 将执行命令

```
mount -w -n -o remount /
```

-o remount 选项使重新用新的参数挂上指定文件系统,-w(等价于-o rw)选项把文件系统

挂成读写模式。以上命令的效果就是:把根文件系统重新挂成读写模式。

每个特定文件系统类型的 fsck 命令都支持一些有用的选项。-a 选项自动确认 fsck 显示的所有提示;-c 选项做坏块检查;-v 选项使在检查过程中输出更多的信息。这些选项应放在 fsck 的 -t type 选项后,如

```
fsck -t type -v device
```

**注意:**fsck 并不能检查出和修补文件系统的所有错误,但对于最常见的问题它都可以处理。如果用户不小心删除了一个重要的文件,现在还没有简单的方法来恢复它。现在,有些人正在努力做一个适用于二级扩展文件系统的“undelete”工具。我们还是提醒用户做备份,或适用 rm -i 命令在删除任何一个文件之前都提示用户。

## 5.8 交换空间

### 5.8.1 交换空间

交换空间是一般性术语,它表示用于在系统上增加表现的内存数量的磁盘空间。在 Linux 下,交换空间用于实现页面管理(paging),即当物理内存低的时候,把内存页面(一个页面通常是 4 096 字节)写到磁盘上,当需要的时候再读入物理内存的过程。进行页面管理的进程对特定的情形做了优化。Linux 上的虚内存子系统允许内存页面在运行的程序之间共享。例如,如果系统上同时有好几个 emacs 进程在运行,只有一个 emacs 的代码被加载进内存。又如,文本页面(包含程序代码,而不是数据)通常是只读的,所以当从交换区移出时并不写到盘上,这些页面直接从内存中释放出来,再用时再从原始的可执行文件中读出来。

当然,交换空间并不能从根本上解决物理内存的缺乏。盘存取比 RAM 存取慢几个数量级。所以,当同时运行很多程序,而它们不能同时都装载进内存时,交换是一种很有效的手段。如果用户在这些程序之间快速切换,就会发现有一个延迟,这是因为内存页面要在物理内存与磁盘之间交换。

Linux 支持两种形式的交换空间:独立的磁盘分区和已存在的 Linux 文件系统上的一个文件。一个系统最多可以有 16 个交换区,每个交换区最大可以是一个 128MB 的磁盘文件或分区。所以 Linux 最大允许 2G 字节的交换空间。

注意使用交换分区比使用磁盘文件效率要高。因为独立分区保证了磁盘块的连续,而交换文件的磁盘块可能分散在整个文件系统中,这在某些情形下效率非常低。很多人在必须临时增加额外的交换空间时使用交换文件,例如,当系统因物理内存和交换空间缺乏而奄奄一息时。交换文件对在需要时临时增加交换空间是一种有效的方法。

几乎所有的 Linux 系统都要利用某种形式的交换空间,通常是一个单独的交换分区。我们已介绍过在 Linux 安装过程中如何创建交换分区。本节我们介绍如何增加和删除交换分区或文件。

首先,要了解系统当前有多少交换空间。free 可以就系统内存使用情况作出报告:

```
lark # free
```

|      | total | used  | free | shared | buffers | cached |
|------|-------|-------|------|--------|---------|--------|
| Mem: | 63104 | 61536 | 1568 | 15912  | 13312   | 38480  |

```
-/ + buffers:      9744      53360
Swap:  128516      4      128512
```

这里的所有数字是以 1024 字节的块为单位。我们看到,系统有 6 3104 块(约 63MB)的物理内存,有 61 536 块(约 61MB)正在使用。注意系统实际上的物理内存比这里的“total”一栏中给出的多一些,这个数字不包括内核自己的各种需要所使用的部分。

“shared”一栏列出在多个进程间共享的物理内存的数量。这里我们看到,大约有 16MB 页面在共享使用。“buffers”和“cached”栏是用来作内核缓冲区的内存的数量。缓冲区用来加速磁盘操作,允许内存直接为磁盘读写服务。缓冲区大小将随系统内存使用状况而增大或减小;如果应用程序需要会占用这部分内存。所以,尽管我们看到有约 62MB 的内存存在使用,但应用程序正在使用的并没有这么多。第二行“-/+ buffers”在 used 项中减去了缓冲区所占内存,而在 free 项中加上了这部分内存,列出了应用程序实际可以使用的系统内存数量。

第三行是交换空间使用状况。总的交换空间是约 128MB。在本例中,系统几乎没有使用交换空间,因为物理内存很充沛。如果有新的应用程序启动,将首先使用一部分缓冲区内存。只有在系统不能用别的方法收集到物理内存时才使用缓冲区。

**注意:**free 所报告的交换空间的大小要比系统上交换分区和文件的总和小。这是因为每个交换区域都有一些块用来存储交换区域的每个块的使用状况。但这部分的花费是很小的,通常对每个交换区域只有几 K 字节。

如果用户感觉有必要创建一个交换文件,应该先用 df 命令看一看系统上各个文件系统上的剩余空间还有多少。这个命令显示所有(或用户指定的)文件系统的大小和已使用的百分比。

### 5.8.2 创建交换空间

增加新的交换空间的第一步是创建作为交换区域的文件或分区。如果要创建新的交换分区,可以用 fdisk 命令创建分区。如果要创建交换文件,就必须创建一个相当于要增加的交换空间大小的文件。我们可以用 dd 命令很方便地实现这一点。例如,要创建一个 8MB 大小的交换文件,可以用命令

```
dd if=/dev/zero of=/swap bs=1024 count=8192
```

从/dev/zero 读出 8192 块(8MB)数据写到文件/swap 中(/dev/zero 是一个特殊的设备,对它的读操作总是返回零字节,类似/dev/null 的逆向设备)。

在创建好交换文件或分区后,要用 mkswap 命令去“格式化”交换区域,其格式是:

```
mkswap -c device size
```

这里,device 是交换分区或文件的名称,size 是交换区域的大小(以块为单位,一块即 1K 字节)。`-c` 开关是可选的,它使 mkswap 在格式化交换区域时检查损坏的块。

例如,对上面产生的交换文件,可用命令

```
mkswap -c /swap 8192
```

如果交换区域是分区,上面命令中就必须以分区名(如/dev/hda4)代替文件名。

对一个交换文件做过 mkswap 后,要用 sync 命令确保格式信息已经在物理上被写进交换文件。对格式化一个交换分区,没必要用 sync 命令。

### 5.8.3 激活交换空间

新的交换空间在使用之前必须用 `swapon` 命令激活。例如,以上在创建交换文件和使用 `mkswap` 和 `sync` 命令后,必须用命令

```
swapon /swap
```

把新的交换区域加到可用的交换区域的总量中去;可用 `free` 命令验证一下。如果要使用一个新的交换分区,用命令

```
swapon /dev/hda4
```

激活交换分区 `/dev/hda4`。

像文件系统一样,交换区域也是在系统启动时从系统的一个启动文件(通常是 `/etc/rc.d/rc.S`)中调用 `swapon -a` 而自动激活的。这个命令查找 `/etc/fstab` 文件,所有 `/etc/fstab` 文件中, `type` 一栏为 `swap` (或 `options` 一栏为 `sw`)的记录对应的分区或文件被 `swapon -a` 激活。

例如,如果 `/etc/fstab` 文件包含

| # device  | directory | type | options     |
|-----------|-----------|------|-------------|
| /dev/hda4 | swap      | swap | default 1 1 |
| /swap     | none      | swap | sw          |

则两个交换区域 `/dev/hda4` 和 `/swap` 将在引导时被激活。用户在增加了新的交换区域后,必须在 `/etc/fstab` 中加上一条记录。

### 5.8.4 关闭交换空间

关闭交换空间用命令

```
swapoff device
```

这里 `device` 是要关闭的交换分区或文件的名字。例如,关闭交换分区 `/dev/hda4` 用命令

```
swapoff /dev/hda4
```

关闭交换分区后,这个分区就可以自由使用了。关闭交换文件后,只要简单地用 `rm` 命令删除对应的交换文件就可以回收磁盘空间。注意,绝对不要在关闭交换文件之前就删除它,否则会有灾难性后果。

最后,删除在 `/etc/fstab` 中对应已关闭的交换空间的记录,否则,在下次引导系统时会有找不到交换区的错误。

## 5.9 设备文件

设备文件使用户程序可以通过内核访问系统上的硬件设备。它们实际上不是“文件”,但从应用程序看像文件:程序可以对它们进行读写和 `mmap()` 等操作。当用户访问一个设备“文件”时,内核会识别这个 I/O 请求,并把它传递给一个设备驱动程序,设备驱动程序再进行一些操作,如从串行口读数据,或发送数据到声卡。

设备文件为程序员提供了一种访问系统资源的方便的方法,它不要求程序员了解其背后的硬件设备是如何工作的。在 Linux 下,像大多数 UNIX 系统一样,设备驱动程序本

身是内核的一部分。在“编译内核”一节里,我们将介绍在编译内核时,如何只包括进对应特定系统硬件的驱动程序。

在很多类 UNIX 的操作系统里,设备文件在目录/dev 下。系统上的每个设备都应在/dev 下有一个对应的设备文件。例如,/dev/ttyS0 对应于第一个串行口,在 MS-DOS 下叫 COM1;/dev/hda2 对应于第一个 IDE 硬盘的第二个分区。事实上,在/dev 中还有许多设备文件可能在系统上没有对应的硬件。设备文件是在系统安装时创建的,包括了每一个可能的设备驱动。它们不必在系统上都有对应的硬件。

在/dev 目录中还有一些不对应于任何真实设备的伪设备文件。如,/dev/null 是一个“下水道”,所有对/dev/null 的写请求都将成功,但写的数据将被忽略。类似地,上节我们用/dev/zero 创建了交换文件,任何对/dev/zero 的写请求将简单地返回零字节。

当用 ls -l 命令列出/dev 目录中的设备文件时,我们将看到每行都类似下面一行:

```
brw-r----- 1 root disk 3, 0 Apr 28 1995 hda
```

/dev/hda 对应于第一个 IDE 硬盘。首先,注意文件保护一栏的第一个字母是 b,意味着这是一个块设备文件。设备文件或者用 b 表示块设备,或者用 c 表示字符设备。块设备通常是一个类似硬盘的设备:对这个设备的读写是以整块为单位的(块的大小由设备决定,并不一定是我们在 Linux 下叫做“块”的 1024 字节),而且设备的访问是随机的。相反,字符设备通常是顺序读写的,I/O 可以是单个字节为单位。比如,串行口就是一个字符设备。

**注意:**上面一行中通常表示文件尺寸的一栏现在是用“,”隔开的两个数。第一个数是主设备号,第二个是子设备号。当程序访问设备文件时,内核是以主设备号和子设备号的形式接收 I/O 请求的。主设备号通常指定内核中的一个驱动程序,子设备号指定这个驱动程序所处理的设备。例如,所有的串行口设备有相同的主设备号和不同的子设备号。内核利用主设备号把 I/O 请求指向特定的驱动程序,驱动程序用子设备号来确定要访问哪个设备。

/dev 目录下文件的命名约定是相当混乱的。内核并不关心在/dev 下用的是什么文件名,它只关心主设备号和子设备号。Linux 发行套件的维护者,应用程序的设计者和设备驱动的作者可以自由地选择设备文件的名称。通常是由设备驱动程序的作者为设备推荐一个名字,后来对类似设备沿用了这个命名规则。这可能会随着系统的发展导致误解和不一致。

在用户拿到的 Linux 发行套件中的设备文件,应当是与发行套件的内核版本和设备驱动程序相一致的。当用户升级内核,或增加额外的设备驱动时,也许需要用 mknod 命令增加一个新的设备文件。mknod 命令的格式是

```
mknod -m permission name type major minor
```

这里,name 是要创建的设备文件的全路径名,如/dev/rft0;type 是 c(字符设备)或 b(块设备);major 是主设备数;minor 是子设备数;-m permission 是可选项,设置新的设备文件的保护位为 permission。

例如,在加一个新的设备驱动到内核时,要创建一个块设备/dev/bogus,主设备号是 42,子设备号是 0,要用如下命令:

```
mknod /dev/bogus b 42 0
```

如果不指定 `-m permission` 选项,新设备的保护用当前的 `umask` 来设定,通常是 0644。如果想设置 `/dev/hogus` 的保护是 0666,用

```
mknod -m 666 /dev/hogus b 42 0
```

当然,也可以用 `chmod` 命令在创建设备文件后修改保护。

设备文件的保护控制着谁可以怎样访问物理设备。例如,对上面的 `/dev/hda`,它的保护是 0640,意味着只有属主 `root` 可以读写,只有 `disk` 组可以读。一般对特定的设备不能给所有的用户以直接读和写的权限,特别是对应于磁盘和分区的设备文件。否则,任何用户都可以对一个磁盘分区使用 `mkfs` 命令而完全破坏系统的数据。

对于磁盘和分区来说,用这种方法破坏数据必须有写的权限,但是读的权限可能也会带来安全问题。如果一个用户对一个对应于磁盘分区的设备文件有读的权限,他就可以偷看别人的文件。同样, `/dev/mem` 对应着系统的物理内存,如果一个聪明的用户对它有读的权限,他就可以在别人登录的时候偷到别人(甚至 `root`)的口令。

所以,在给新增加的设备文件设置保护时,应该明确用户将怎样使用这个设备。并行口、声卡和虚控制台等设备允许普通用户使用一般是安全的。但系统上其他大多数设备应该限定为 `root`(或 `setuid` 为 `root` 的应用程序)使用。

在 `/dev` 目录下的很多文件实际上是到另一个设备文件的符号连接。这些连接使得用户可以用一些较常用的名字来使用某些特定的设备。例如,串行鼠标可能使用设备文件 `/dev/cua0`, `/dev/cua1`, `/dev/cua2`, `/dev/cua3` 之一,取决于鼠标连于哪个串行口上。很多人把 `/dev/mouse` 连接到鼠标所对应的串行口设备文件,如

```
ln -s /dev/cua2 /dev/mouse
```

这样,我们就可以从 `/dev/mouse` 使用鼠标,而不必记住鼠标在哪个串行口上。这种约定同样用在 `/dev/cdrom` 和 `/dev/modem` 等设备上。这些文件通常是连接到 `/dev` 目录中 `CD-ROM` 或调制解调器对应的真实设备文件上。

删除设备文件用 `rm` 命令:

```
rm /dev/hogus
```

删除设备文件并没有从内核或内存中删除对应的设备驱动程序;它只是使用户无法与对应的设备驱动程序对话。同样,增加一个设备文件并没有给系统增加一个设备驱动程序,事实上,用户可以为并不存在的驱动增加一个设备文件。设备文件只是简单地提供了连接内核中特定驱动程序(如果它存在的话)的桥梁。

## 5.10 系统的备份与恢复

系统备份是保护用户不受数据损坏或丢失之苦的一种非常重要的手段。如果系统的硬件出了问题,或用户不小心删掉了重要的文件,都有可能造成数据损坏或丢失。经常性的系统备份可以使损失减小到最低程度。用户在使用 Linux 的过程中,可能对系统设定做了一系列的改变,有些改变不能简单地用从原始安装介质上重新安装而得到恢复。不过,如果用户在手边保存着 Linux 的原始软盘或 CD-ROM,也许备份下整个系统是不必要的,原始安装介质本身就是一个很好的备份。

在 Linux 下,像其他 UNIX 系统一样,用户在登录进 `root` 时可能会犯错误而使系统无

法引导或登录。有些新手在这种情况下可能会重新安装一遍整个系统,这通常是不必要的。我们将在稍后介绍如何处理这种情形。

有些数据丢失后可以用文件系统维护工具来恢复。但与其他操作系统不同的是,在 UNIX 上一般不可能“undelete”用 rm 命令删除的或用 cp 及 mv 命令不小心覆盖掉的文件。在这些极端情形下,备份对数据恢复是极其重要的。

备份通常使用磁带或软盘。若长期使用,磁带一般比软盘更可靠,但没有什么介质是 100% 可靠的。有很多工具可以帮助系统管理员做备份。最简单的可能是用 gzip 和 tar 命令把文件从硬盘上备份到软盘或磁带上。当备份不是太频繁时,例如一个月不超过一次时,这是最好的方法。

如果系统上有很多用户,或系统配置频繁地被改变,这时就有必要实行增量备份方案。在这种方案下,大约每个月做一次系统的“完全备份”。然后,每个星期仅对过去的一个星期内改变了的文件进行一次备份。同样,每天夜里,仅对过去 24 小时内改变的文件做一次备份。

增量备份是基于这样一种思想:小步伐的备份更加经济;备份需要更少的软盘或磁带,每周和每夜的备份工作量较小,更容易完成。我们手头的备份最多是一天前的。如果我们不小心删除了整个系统,可以用下列方式从备份恢复它

(1) 先恢复最近一个月的备份。假如我们是在 7 月 17 日删除了系统,我们将恢复 7 月 1 日的完全备份。这时系统的文件是 7 月 1 日做备份时的状态。

(2) 再恢复本月到目前为止每周的备份。在我们的例子中,我们要恢复 7 月 7 日和 14 日的两个周备份。恢复每个周备份使那一周更改过的文件得到更新。

(3) 恢复从上一次周备份后的每一个日备份。在我们的例子中,我们要恢复 7 月 15 日和 16 日的日备份。现在系统看起来和 7 月 16 日做日备份时一样了;我们至多丢失了一天的文件。

一个完全的月备份可能会有几百 MB 的数据量,这取决于系统的大小。这完全可以存在一个磁带上,却是上百片软盘的容量。但是周备份和日备份通常需要少得多的存储空间。

备份应该具有的一个重要特征是可以从备份中选择恢复单个文件。这样,如果用户不小心删除了一个或几个文件,他就可以简单地只恢复这几个文件,而不必做整个系统的恢复。但是,这取决于备份的方法,有时候做起来会很容易,有时候会极其困难。

在本节里,我们将介绍使用 tar, gzip 和其他一些相关的工具做软盘和磁带备份。我们甚至也介绍软盘和磁带机的使用。我们可以通过写一些 shell 描述文件使备份过程自动化,甚至可以用 cron 把备份定时在夜间进行。还有一些软件包可以提供方便的菜单界面,包含备份、恢复指定文件等功能。它们中大部分是 tar 和 gzip 的前端程序。用户可以选择适合自己的系统的备份软件。

### 5.10.1 简单备份

做备份的最简单的方法是用 tar 命令把系统上所有的文件或一些指定目录下的文件打包存档。但在做备份之前,首先必须明确哪些文件需要备份。在绝大多数情况下是没有必要备份系统上的全部文件的,特别是如果手头还有原始安装软盘或 CD-ROM 时。如

果用户已经对系统作了重要的改变,但除此以外其他一切还跟原始安装介质上没什么区别的话,就可以只备份更改过的文件。但是,跟踪这些改变是很困难的。

一般来说,用户会改变在/etc目录中的系统配置文件。其他一些目录中还包含一些重要的配置文件,如/usr/X11R6/lib/X11中的XFree86配置文件。这些目录和文件都是需要备份的。

如果用户已经重新编译和更新过自己的内核,那么就必须备份在/usr/src/Linux下的内核源程序。

建议用户在使用Linux的过程中,记录下自己曾经对系统的什么特征作了改变,这样在备份的时候会据此作出明智的选择。如果用户实在不放心,也可以把整个系统备份下来,但这可能会耗费很多磁带。

当然,需要备份的东西还有系统上每个用户的主目录(通常在/home目录下)。如果系统接收电子邮件,还要备份用来接收用户邮件的收件箱。很多用户习惯把老的邮件放在收件箱里,而收件箱有可能因邮递程序等的错误而遭到破坏。用户的收件箱通常放在/var/spool/mail目录里。

### 1. 用磁带备份

明确了哪些文件或目录需要备份后,我们就可以开始备份了。我们可以直接使用tar命令做备份:

```
tar cvf /dev/rft0 /usr/src /etc /home
```

以上命令把/usr/src,/etc和/home目录下的所有文件备份到/dev/rft0上。/dev/rft0是第一个“软磁带”设备,即挂在软盘控制线上的磁带机。很多PC机上使用的磁带机都使用这种接口。对于SCSI磁带机,设备名是/dev/st0,/dev/st1等。其他接口的磁带机各有对应的设备名,用户可以去查找与内核中设备驱动程序有关的文档。

从磁带上的档案中读取文件用命令

```
tar xvf /dev/rft0
```

这与处理磁盘上的tar文件完全类似。

使用磁带机时,磁带被用作只能单向读写的流设备。当tar完成后,磁带设备将被关闭,磁带将被回卷。不能在磁带上创建文件系统,不能像磁盘一样挂上磁带或以文件方式访问磁带上的数据。磁带设备只能作为单个“文件”创建和从中抽取档案。

磁带在使用之前应该格式化。这保证了开始标记和坏块信息被写进磁带。但到本书写作时,还没有在Linux下格式化QIC-80磁带(用作软磁带设备)的工具。所以必须在DOS下格式化磁带或购买预先格式化的磁带。

当档案只须占用磁带容量的一小部分时,每个磁带只创建一个tar文件是很浪费的。为使一个磁带能存放更多的文件,首先必须避免磁带在每次使用后回卷,还必须使磁带在创建和抽取时都能走到下一个“文件标志符”。

解决的方法是使用不回卷磁带设备,其名字对于软磁带机是/dev/nrft0,/dev/nrft1等,对于SCSI磁带是/dev/nst0,/dev/nst1等。当使用这些设备读写时,一旦tar执行完毕,设备将关闭,但磁带不回卷。所以就可以接着用tar命令往磁带上添加新的档案。磁带上的两个tar文件彼此互不相关。当然,如果以后重写磁带而覆盖了第一个tar文件,

也可能覆盖了第二个 tar 文件或在第一个和第二个 tar 文件之间留下了一段没用的空隙(可视为垃圾)。一般地,不要试图更新有很多文件的磁带上的单独一个文件。

使用不回卷磁带设备,只要空间允许,我们就可以往磁带上增加很多文件。磁带使用后回卷用命令 `mt`。`mt` 是对磁带机进行各种操作的通用命令。

例如,命令

```
mt /dev/nrft0 rewind
```

回卷在第一个软磁带设备中的磁带。类似地,

```
mt /dev/nrft0 reten
```

把磁带卷到尾部再回卷以拉紧磁带。

当从一个多文件的磁带上读文件时,必须在 `tar` 命令中使用不回卷磁带设备,必须用 `mt` 命令使磁带的当前位置指向适当的文件。

例如,跳到磁带上的下一个文件,用命令

```
mt /dev/nrft0 fsf 1
```

这是跳过一个文件。跳过两个文件用

```
mt /dev/nrft0 fsf 2
```

注意在以上命令中使用合适的不同卷磁带设备。这条命令并不是将磁带卷到第二个文件的位置,而是从当前位置跳过两个文件。在记不清磁带的当前位置的情况下,用 `mt` 命令把磁带回卷。也可以用 `mt` 命令跳回,参考 `mt` 的在线手册可获得完整的选项解释。

每次读取多文件磁带都必须用 `mt`。连续使用两次 `tar` 命令读磁带上两个档案文件通常是不能奏效的,这是因为 `tar` 不能识别在两个文件之间的文件标志符。第一个 `tar` 命令完成后,磁带的位置是文件标志符的开始。这时立刻使用 `tar` 命令会给出一个错误信息,因为 `tar` 命令试图读文件标志符。在从磁带上读取一个文件后,要用命令

```
mt device fsf 1
```

使磁带移到下一个文件。

## 2. 用软盘备份

类似上节所述,命令

```
tar cvf /dev/fd0 /usr/src/etc /home
```

把目录 `/usr/src/etc` 和 `/home` 备份到第一个软盘设备。读取备份用命令

```
tar xvf /dev/fd0
```

同样把软盘看作磁盘上的一个文件。用户没有软盘上创建文件系统和试图挂上它。

因为软盘的存储能力很有限,GNU `tar` 命令允许创建“多卷”档案。使用这个特征时,`tar` 在读写完一张软盘时会提示用户插入新的一张软盘。`tar` 的 `M` 选项使用这个特征,如

```
tar cvMf /dev/fd0 /usr/src/etc /home
```

注意:要在每张软盘上写上标记,并在恢复档案时不要把原来的顺序搞乱了。

### 5.10.2 增量备份

增量备份是保持系统及时备份的一种好方法。例如,每日备份过去 24 小时内更改的文件,每周备份过去一周更改的文件,每月备份整个系统。

创建增量备份的工具具有 `tar`, `gzip`, `cpio` 等等。创建增量备份的第一步是产生一个在过

去一段时间里更改过的文件的列表。这可以用 `find` 命令很容易地做到。

例如,产生在过去 24 小时内更改过的文件的列表用命令

```
find / -mtime -1 \! -type d -print > /tmp/filelist.daily
```

`find` 的第一个参量是起始目录,在这里是/,根目录。`-mtime -1` 选项告诉 `find` 命令找出所有在过去 24 小时内更改过的文件。`\! -type d` 比较复杂,它的含义是从输出中去除某些不必要的东西,它告诉 `find` 命令从文件列表中去掉目录名。`!` 是否定算符(`-type d` 的相反,即去掉类型 `d` 的文件名),它前面的反斜线是必要的,否则 `shell` 将把它解释为特殊字符。

`-print` 把所有符合查找要求的文件名打印到标准输出上。这里我们把标准输出重定向到一个文件中。

类似地,找出在过去一周内更改过的所有文件用:

```
find / -mtime -7 -print > /tmp/filelist.weekly
```

注意如果这样用 `find` 命令,它将详查所有挂上的文件系统。如果系统上碰巧挂上一个 CD-ROM, `find` 也将搜寻 CD-ROM,而我们绝无必要备份 CD-ROM。我们可以给定除/外的其他起始目录,来列出在那个目录下要备份的文件。

现在我们已经产生了要备份的文件的列表。以前我们用 `tar` 的时候是在命令行指定要存档的文件。但现在的这个文件列表放在一个命令行里太长了(一个命令行的长度通常限制在 2 048 个字符以内),而且这个列表本身是放在一个文件里的。

我们可以用 `tar` 的 `-T` 选项指定一个文件,它包含要备份的文件列表。要使用这个选项必须用 `tar` 命令的另一种定义,即在所有的选项前加上破折号“-”。例如,把列在文件 `/tmp/filelist.daily` 中的所有文件备份到设备 `/dev/rft0` 上,用命令:

```
tar -cv -T /tmp/filelist.daily -f /dev/rft0
```

用户可以写一个简短的 `shell` 代码自动产生文件列表并用 `tar` 备份,然后用 `cron` 命令在夜间确定的时刻执行它。用户所要注意的就是在磁带机里要有磁带。用户可以为周备份和月备份写类似的 `shell` 代码。我们将在下节里介绍 `cron`。

## 5.11 用 `cron` 制订作业日程

在 UNIX 系统上,有些系统维护的工作需要定时完成;有时候,用户可能需要在特定时间完成一些特定工作。有些工作可能不便于在指定时刻由用户亲手输入命令来完成。在这种情况下,用户可以用 `cron` 机制来制定作业日程,使系统在规定的时间内完成特定的工作。

简单地讲,用户可以运行 `crontab` 命令,用 `cron` 识别的特定格式输入一些记录,每一行记录指定了要运行的命令及运行的时间。

`crontab` 命令把用户输入的记录存入 `/var/spool/cron/crontabs` 目录中一个以用户名命名的文件中(如用户 `wang` 的 `crontab` 文件将是 `/var/spool/cron/crontabs/wang`)。守护程序 `crond` 定期读入这个文件,并在适当的时候执行其中的命令。在系统引导时,一个 `rc` 文件启动 `crond`。实际上并没有 `cron` 这个命令,只有 `crontab` 和 `crond` 程序。

在大多数系统上,每个用户都可以使用 `cron`;但在某些系统上,只有 `root` 可以使用

cron。下面我们以一个例子说明如何添加一条 crontab 记录。

在 Linux 系统上,有一个 updatedb 命令,它搜索指定的文件系统,把文件系统上所有文件的名字存放在一个文件名数据库里,用户可以用 locate 命令查询系统上是否有某个或某些文件。如

```
lark# updatedb
```

命令将缺省地搜索所有在/etc/mtab 文件里列出的文件系统,并把所有文件的全路径名存储在文件名数据库/var/spool/locate/locatedb 中。而下列命令将会搜索文件名数据库,找出所有含字符串“myreport”的文件名(包括路径):

```
lark% locate myreport
/home/wang/myreport
/home/wang/myreport/1.txt
/home/wang/myreport/letter.doc
/home/wang/myreport/myreport.txt
/home/wang/myreport/myreport.txt
/home/wang/myreport/zebra.txt
/home/zhang/myreport.doc
```

因为文件系统中每天都有大量的文件被产生或删除,所以我们希望每天更新文件名数据库。我们希望这种更新在没有用户登录时,即在夜间完成。我们可以把 updatedb 语句加进 root 的 crontab 记录。以 root 登录,执行 crontab 命令

```
lark# crontab -e
```

则 crontab 命令让用户用目前缺省的编辑器编辑 root 的 crontab 文件。在文件最后附加上下面一行:

```
40 04 * * *      updatedb 1> /dev/null 2> /dev/null
```

然后存盘退出。crontab 命令将更新 root 的 crontab 文件,并及时通知 crond 进程。crond 将在每天凌晨 04:40 执行更新文件名数据库的任务。

每条 crontab 记录由六个域组成:M H D m d cmd。

M :分钟(0~59)

H :小时(0~23)

D :天(1~31)

m :月(1~12)

d :一星期内的天(0~6,0 为星期天)

cmd:要运行的程序,程序被送入 sh 执行,sh 只有 USER,HOME,SHELL 这三个环境变量,详见本书“Linux 使用入门”一章中对 crontab 命令的介绍。

## 5.12 管理打印服务

Linux 上的打印系统允许很多用户同时发送打印请求,打印系统按顺序处理要打印的文件,使之正确地打印在以不同方式接在系统上的各种打印机上。一个文件还可以通过网络打印到另一台计算机控制的打印机上。

当用户在一个应用程序中用鼠标按下“Print”按钮,或在命令行发出一个打印命令

(如 `lpr`)时,要打印的文件并没有直接输出到打印机上,因为打印机可能正在使用。文件首先在一个称为打印机 `spool` 的目录中被存储成一个临时文件。一旦打印机空闲,临时文件一个接一个地从 `spool` 目录中被发送到打印机上。每个打印机有自己单独的 `spool` 目录。

Linux 启动时建立一个打印机守护进程 `lpd`。这个进程始终在后台检查每一个打印机 `spool` 目录中是否有文件等待打印。一旦某个 `spool` 目录中有等待打印的文件,`lpd` 就作自我复制。新的 `lpd` 进程接管了这个 `spool` 目录,把文件放到打印队列上。等到打印机上所有文件打印完毕,`lpd` 就把队列上的文件发送到打印机上。一旦 `spool` 目录中没有等待打印的文件,对应的 `lpd` 进程就完成了任务而退出了。对于每个打印机,只要它的 `spool` 目录下有等待打印的文件,原始的 `lpd` 进程就会为它复制一个 `lpd`。

Linux 在启动时会把系统的打印机端口连接到 `/dev` 目录下的一个设备文件。下一步是在打印机设置文件 `/etc/printcap` 中把设备与将在打印命令中使用的打印机名联系起来。

在打印管理中,另一项非常重要的工作是为 `lpd` 准备打印过滤器(filter),打印过滤器也是在 `/etc/printcap` 文件中指定的,用来格式化将要打印的文件。

现有的几种 Linux 打印系统的软件包中,大多数 Linux 发行套件使用的是源于 BSD 的软件包,其中有 `lpd` 打印机守护程序。这些软件包包括在 Linux 上支持传统的 UNIX 风格的打印的一些命令和帮助文档。BSD 打印系统不如基于 System V UNIX 的打印管理系统(其中使用了 `lpsched` 守护程序)有更多的管理工具,但是每个用户都可以控制他们发送到打印机上的文件。我们在这一节里将介绍安装和配置 BSD 打印支持系统的方法。

某些 Linux 发行套件,如 RedHat,提供了一个打印机管理工具,通过一个图形界面简化了打印机安装和管理。但它们并不总是工作得很正常。如果用户想对一些打印配置进行修改,或想改进打印性能,就必须了解如何设置打印系统。本节将对此作简单介绍。

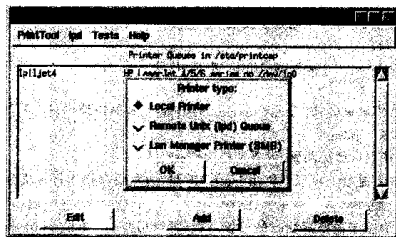


图 5.1 RedHat 的打印机管理工具 `printtool`

### 5.12.1 检查打印机硬件

在设置打印系统之前,首先应安装好打印设备。把打印机正确地接在机器的并行口或串行口上。Linux 系统的内核在启动过程中一般会正确地探测到机器的并行口和串行

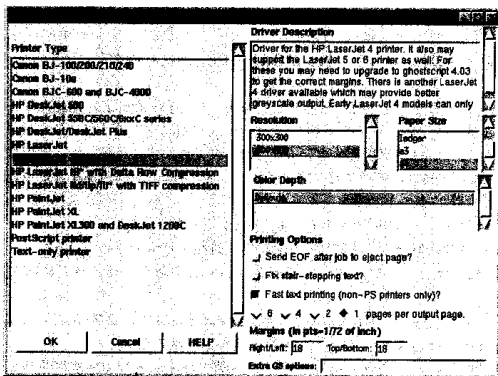


图 5.2 printtool 中配置打印过滤器

口,并行口设备名一般是/dev/lp0或/dev/lp1等,串行口设备名一般为/dev/ttyS0,/dev/ttyS1等。一般Linux发行套件的内核或本身支持并行口打印,或把并行口打印支持编译成可加载模块。如果是后者,用户必须保证在/etc/rc.d/rc.modules文件中用/sbin/modprobe命令把并行口打印模块加载进内核,如对2.0版本的内核是下面一行:

```
/sbin/modprobe lp
```

在测试版内核2.1.xx中则是下面三行

```
/sbin/modprobe parport
```

```
/sbin/modprobe parport_pc io=0x378,0x278 irq=7,5
```

```
/sbin/modprobe lp
```

在系统启动信息里(同时写进/var/log/messages),我们应该可以看到类似

```
lp1 at 0x0378, (polling)
```

的信息,这表明并行口已探测到,设备文件为/dev/lp1。

超级用户root是唯一可以不使用lpd打印守护进程而直接往打印机对应的输出设备写数据的用户。假如我们在第一个并行口上接了一个打印机,它的设备名可能是/dev/lp0或/dev/lp1,取决于内核探测时的设置。我们可以用/usr/sbin/lptest直接往打印机上发送一个ASCII文件,以检测我们是否已正确地安装了打印机(如果是PostScript打印机,我们可能需要发送一个小的PostScript文件)。lptest程序将产生一个测试文件,它包含全部96个可打印的ASCII字符,每行缺省79个字符,逐行递进,如波浪状,缺省共200行,加换行符共16000个字符。它的两个选项分别表示行的长度和行数,如

```
lptest 35 6
```

将输出如下结果

```
!"# $ % &'()*+,-./0123456789;:<=>? @ ABC
"# $ % &'()*+,-./0123456789;:<=>? @ ABCD
# $ % &'()*+,-./0123456789;:<=>? @ ABCDE
$ % &'()*+,-./0123456789;:<=>? @ ABCDEF
% &'()*+,-./0123456789;:<=>? @ ABCDEFG
&'()*+,-./0123456789;:<=>? @ ABCDEFGH
```

下面的命令

```
lpctest > /dev/lp1
```

将把测试文件发往并行口设备 lp1,如果打印机正确输出了上述结果,说明打印机连接正确。

我们也可以使用 cat 命令把一个文件发往打印机。如发送一个检测性 PostScript 文件到一个 PostScript 打印机上,用命令

```
cat testfile.ps > /dev/lp1
```

如果系统上连接的是串行口打印机,试着把文件直接发往连接打印机的串行口设备。如果打印机连在第二个串行口上,用

```
lpctest > /dev/ttyS1
```

如果用户使用的是一个连续纸打印机,文件打印完后不能自动换页,用户可以在 /etc/printcap 文件中设置强制换页,或在打印过滤器中把换页符附加到文件尾部。我们将在以后几节中讨论到这个问题。

如果测试结果是阶梯状文字,类似如下

```
!"# $ % &'()*+,-./0123456789;:<=>? @ ABC
                                     "# $ % &'()*+,-./0123456789;:<=>? @ ABCD
                                     # 8
```

这是因为,UNIX 文本文件的每一行的结尾处只有一个换行符(ASCII 码是 10),而 DOS 文本文件则在行尾有换行符和回车符。打印机当前是被设置成 DOS 模式。为了能在 UNIX 下打印文本文件,或安装一个支持 UNIX 的换行设置的打印过滤器,或重新配置打印机,使之在遇到换行符时另起一行。对于第一种方法,我们将在下面加以讨论;对于第二种方法,请参考打印机的硬件说明。

### 5.12.2 选择打印软件

在 Linux 系统下打印,首先必须安装 BSD 打印系统。事实上,大多数 Linux 发行套件中都包括 BSD 打印系统。为了使它支持现代的打印机,我们还必须安装一些其他的辅助软件包。一般来说,至少要求系统上有如下软件包:Ghostscript, nenscript。大多数 Linux 发行套件中都包括了这些软件包和其他排版和打印工具。如果用户所使用的发行套件中缺少以上软件包,可从 Linux 的 FTP 节点或别的发行套件的 CD 中取来。

当前打印技术的趋势是采用页面描述语言(PDL, page-description language),它提供了复杂的图形和字库控制。目前最流行的 PDL 是 PostScript,它已经被 UNIX 操作系统和 Internet 社会普遍接受。它被接受的一个重要原因是 Ghostscript 的出现,它是 Aladdin 公司的 PostScript 实现软件。它在 FSF 的 GPL 版权声明下发布,并带一个大的字库集。Ghostscript 实现了 PostScript 语言的几乎全部指令,并支持浏览工具,可使 PostScript 文

件在 X Window 下显示。它还可以把 PostScript 转换成其他打印语言,如 HP 的 PCL 等,从而可以产生能直接发送到打印机上的输出格式。如果用户想打印除字符外的任何文件,Ghostscript 都是不可缺少的。Ghostscript 包支持 Adobe 的 Type 1 和 Type 3 的 PostScript 字库,并提供了一系列图形格式转换和过滤的工具。

但光有 Ghostscript 是不够的,因为它不能对普通文本文件进行处理。对很多系统来说,nenscript 满足简单、灵活、可靠的要求,所以我们可以选用它。它可以将输入的普通文本文件转换为 PostScript 文件,而对输入的 PostScript 文件则不作改变。所以可以用 nenscript 放在 Ghostscript 的前面,从而使传递给 Ghostscript 的都是 PostScript 文件。

Ghostscript 一般都包含在 Linux 的发行套件里。如果用户需要更新的版本,可从

`ftp://ftp.cs.wisc.edu/ghost/aladdin/gs # # # /`

取来 Ghostscript 的源程序包进行编译。

nenscript 包含在 RedHat Linux 里,但在 Slackware 中并没有包含 nenscript。用户可从

`ftp://sunsite.unc.edu/pub/Linux/system/printing/`

或其镜像节点取来 nenscript 在系统上进行编译安装。

### 5.12.3 配置 printcap 文件

BSD 打印系统的打印机配置文件是 `/etc/printcap`。printcap 文件中的每条记录对应系统的一个打印机队列。一个 printcap 记录的例子是:

```
ljet4 |lp|ps|PostScript; \
:lp = /dev/lp1; \
:sd = /var/spool/lpd/ljet4:mx # 0:mc # 0; \
:lf = /var/log/lpd-errs;if = /var/spool/lpd/ljet4/filter;
```

printcap 记录的格式约定是:

- (1)注释行以“#”开始;
- (2)续行号是“\”,注意“\”前不能有空格;
- (3)一行和它所有的续行定义了一个打印机队列;
- (4)每行里每个域之间用“:”隔开,最后一个域后必须有“:”,但第一个域前无“:”;
- (5)第一个域定义打印机队列名,每个打印机队列可有多个名字,以“|”隔开;有“lp”作为名字的打印机队列为缺省打印机队列;

(6)每个本地打印机队列都必须有一个“lp”域定义打印机连接在哪个端口。如上例中的“lp = /dev/lp1”。

printcap 记录中的各主要域意义如下:

sd

(spool directory) 打印缓存目录。一般是 `/var/spool/lpd` 目录的子目录,目录名应与第一个打印机队列名一致。如在上例中,第一个打印机队列名是 ljet4,则打印缓存目录被定义为 `/var/spool/lpd/ljet4`。

lp

(local printer) 本地打印机设备。如 `/dev/lp0`、`/dev/lp1`、`/dev/ttyS1` 等。不能与 rp (远

程打印机)同时设定。如果设置为串口设备,必须用 `br` 设定波特率。

`lf`

(`log file`) 存储错误信息的记录文件。

`mx`

(`maximum size`) 可打印的文件的最大尺寸。缺省是 `mx # 0`,即无限制;其他值设置的是最大文件的块(`block`)数。

`if`

(`input filter`) 输入过滤器。如果没有指定输入(`if`)或输出(`of`)过滤器,使用缺省的 `/usr/sbin/lpf` 过滤器。

`of`

(`output filter`) 输出过滤器。不太经常使用。

`br`

(`"baud" rate`) 串口波特率,与串口打印机配合使用。

`rp`

(`remote printer`) 远程打印机队列名。不能与 `lp` 同时使用。但必须用 `rm` 指定远程机器。在使用远程打印机时,一般只须设 `sd` 和 `lf` 域。

`rm`

(`remote machine`) 控制远程打印机的远程系统。

`sh`

(`suppress header`) 不打印标题页。一般应设 `sh` 域。

`mc`

(`maximum copies`) 最大打印份数。缺省是 `mc # 0`,即无限制。

一个使用远程打印机的 `printcap` 记录的例子是:

```
stcolor l Epson!EPSON Stylus Color: \
    :mx # 0:rm = angel.ihep.ac.cn:rp = lp: \
    :sd = /var/spool/lpd/stcolor: \
    :lf = /var/log/lpd-errs:
```

#### 5.12.4 打印过滤器

`printcap` 文件中 `if` 域指定的是打印过滤器,所有的文件都要先经过打印过滤器再送到打印机上。有些 Linux 发行套件包含打印过滤器,如 RedHat 的 `printtool` 包。网络上有各种各样的打印过滤器软件,有些打印机的生产商也会提供打印过滤器软件。但我们也可以自己编写利用已有软件的打印过滤器。

下面是一个利用 `Ghostscript` 和 `nenscript` 的简单却有效的打印过滤器。假设我们使用的是 HP LaserJet 4 系列的打印机。我们在 `/etc/printcap` 文件中使用了如下的 `printcap` 记录建立打印机队列:

```
ljet4!lp: \
    :sd = /var/spool/lpd/ljet4: \
    :mx # 0: \
    :sh: \
```

```

:lp = /dev/lp1: \
:if = /var/spool/lpd/ljet4/filter;
printcap 记录中指定的输入过滤器是/var/spool/lpd/ljet4/filter 文件,我们创建这个文件如下:

```

```

# ! /bin/sh

DEVICE = ljet4
RESOLUTION = 600x600
PAPERSIZE = a4
SENDEOF =

nenscript -TUS -ZB -p- l
if [ "$DEVICE" = "PostScript" ]; then
    cat -
else
    gs -q -sDEVICE = $DEVICE \
    -r $RESOLUTION \
    -sPAPERSIZE = $PAPERSIZE \
    -dNOPAUSE \
    -dSAFER \
    -sOutputFile = -
fi

if [ "$SENDEOF" != "" ]; then
    printf "\n"
fi

exit 0

```

这个过滤器没有什么难懂之处。首先它设置了作为 Ghostscript 命令的选项的变量。接着,输入文件经 nenscript 传递到 gs 命令。如果某种打印机打印完后不会自动换页,可设置 SENDEOF 变量,如:

```
SENDEOF = '\f'
```

将在打印完后发给打印机一个换页字符。

用户必须根据自己的打印机型号正确设置 filter 文件中的 DEVICE 变量。DEVICE 是传递给 gs 命令作为“-sDEVICE = \$DEVICE”选项的,它必须正确代表用户所使用的打印机型号。用户可以用“gs -help”命令看一看系统上的 Ghostscript 支持哪些设备:

```

lark% gs -help
Aladdin Ghostscript 5.10 (1997-11-23)
Copyright (C) 1997 Aladdin Enterprises, Menlo Park, CA. All rights reserved.
Usage: gs [switches] [file1.ps file2.ps ...]
Most frequently used switches: (you can use # in place of =)
    -dNOPAUSE      no pause after page    l -q 'quiet', fewer messages

```

```
-g < width > x < height >      page size in pixels   | -r < res > pixels/inch resolution
-sDEVICE = < devname >         select device           | -dBATCH exit after last file
-sOutputFile = < file >        select output file; - for stdout, |command for pipe,
                                embed %d or %ld for page #
```

Input formats: PostScript PostScriptLevel1 PostScriptLevel2 PDF

Available devices:

```
x11 x11alpha x11cmymk x11gray2 x11mono doskjet djet500 laserjet ljetplus
ljet2p ljet3 ljet4 edeskjet edcolor edjmono edj550 pj pjxl pjxl300
uniprint hj10e bj200 bjc600 bjc800 faxg3 faxg32d faxg4 pexmono pcxgray
pcx16 pcx256 pcx24b pcxcmymk pbm pbmraw pgm pgmraw pgm pgmraw pnm pnmraw
ppm ppmraw tiffle tiffg3 tiffg32d tiffg4 tiffbw tiffpack tiff12ne
tiff24ne psmmono psgray bit bitrgb bitcmymk pgmmono paggray png16 png256
png16m jpeg jpeggray pdfwrite pswrite epswrite pxlmono pxcolor nullpage
```

Search path:

```
. : /usr/local/share/ghostscript/5.10 ;
   /usr/local/share/ghostscript/fonts
```

For more information, see /usr/local/share/ghostscript/5.10/doc/use.txt.

Report bugs to ghost@aladdin.com; use the form in bug-form.txt.

用户可以在“Available devices:”一栏里寻找对应自己的打印机的设备名,把它用作 DEVICE 变量。如果找不到自己的打印机的设备名,就必须重新编译 Ghostscript,将相应的设备驱动加进 Ghostscript。

在设置完/etc/printcap 文件和打印过滤器后,我们必须重新启动 lpd,这可以用如下命令来做(以 root 登录):

```
lark # lpc restart all
```

lpc 杀死系统上所有的 lpd 进程,并重新启动 lpd。

现在我们就可用 lpr 命令直接打印文本文件和 PostScript 文件了:

```
lark % lpr report.txt
lark % lpr resume.ps
```

## 5.13 进程记账

有些时候,系统管理员可能需要记录系统上每个用户对系统资源的使用状况,如用户的联机时间、CPU 使用量、具体到每一条命令所用系统资源的统计等。这个过程称为进程记账(process accounting)。进程记账对任何互联网服务提供商(ISP, Internet Service Provider)都是一项很重要的工作,因为他们根据记账信息来确定向用户的收费金额。对于一般公用系统的管理员来说,也可以利用记账软件提供的信息来统计系统的使用状况,为系统优化提供信息。

进程记账支持从 Linux 的 1.3.73 版开始已经被集成进内核。所以,当前的 Linux 内核都已有记账支持。对于先于 1.3.73 版的内核,可以从下列地址取来一些“补丁”文件:

```
ftp://iguana.hut.fi/pub/Linux/Kernel/process_accounting
```

修正内核使之支持记账功能。但推荐升级到 2.0 版以上的内核。

统计记账信息需要进程记账软件。有些 Linux 的发行套件可能已经包含了进程记账软件(可查看系统上是否有 `accton`, `sa`, `ac` 等命令)。如果系统上没有记账软件,可使用 GNU 的 `acct` 包,可从 `ftp://prep.ai.mit.edu/pub/gnu` 目录或 GNU 的任一镜像节点下载。目前的版本是 6.3,文件名是 `acct-6.3.tar.gz`。

`acct` 软件包的安装过程很简单,首先把软件包打开在某个工作目录下,然后执行下列命令:

```
lark% cd acct-6.3
lark% ./configure
lark% make
lark% su
lark # cp ac accton dump-acct dump-utmp last lastcomm sa /sbin
lark # cp ac.1 last.1 lastcomm.1 /usr/man/man1
lark # cp accton.8 sa.8 /usr/man/man8
```

`/sbin/accton` 是启动和关闭进程记账的命令。如果不带参数运行 `/sbin/accton`,它缺省是关闭进程记账。启动进程记账的命令一般放在系统的某个启动文件里,如在 Slackware 里,用户可以在 `/etc/rc.d/rc.local` 文件里加上以下几行:

```
# Start process accounting
if [ -x /sbin/accton ]
then
    echo "Turning on accounting..."
    /sbin/accton /var/account/pacct
fi
```

下次重新启动系统时,系统将启动进程记账。`/var/account/pacct` 是系统的进程记账文件,如果不存在,可用下面的命令创建:

```
lark # touch /var/account/pacct
lark # chown root /var/account/pacct
lark # chmod 0644 /var/account/pacct
```

下面介绍几个进程记账命令。

`ac`:

`ac` 根据 `/var/log/wtmp` 文件打印用户联机时间的统计。`/var/log/wtmp` 由系统的 `init` 和 `login` 进程管理,是用户登录记录文件。`ac` 命令不带选项打印自从有 `wtmp` 记录以来所有用户登录时间的总和。下面是 `ac` 命令的一些主要选项:

`-d, --daily-totals`

逐日统计,输出类似如下的结果:

```
Nov 26 total 115.42
Nov 27 total 75.63
Nov 28 total 42.81
Nov 29 total 22.43
Nov 30 total 39.09
Dec 1 total 20.52
Dec 2 total 44.87
```

```
Today total 56.45
```

```
-p, --individual-totals
```

按用户名统计。ac -p 输出每个用户联机总时间,如下:

```
li 5.53
liu 52.52
root 15.74
wang 28.14
yang 14.91
zhang 0.75
zhao 75.90
total 193.49
```

-p 后列出用空格分开的若干用户名,表示只统计列出的用户的联机时间。

-d 和 -p 联合起来使用则给出每个用户逐日的统计。

last

last 根据/var/log/wtmp 文件打印用户每次登录的联机时间。last 按时间倒序输出用户每次登录的起止时刻和停留时间。last 命令后可跟用户名、终端设备名(如 tty4 或简单地 p4)选项,表示仅列出指定的用户名或指定终端设备上的联机时间。-num 选项表示只列出 wtmp 中最近的 num 条记录。如:

```
lark% last -6
zheng tty2 0:0 Wed Dec 3 20:45 still logged in
zheng tty3 hpws3.ihep.ac.cn Wed Dec 3 20:35 - 20:45 (00:10)
bbs tty4 hpar1.ihep.ac.cn Wed Dec 3 20:17 still logged in
yuan flp fish.ihep.ac.cn Wed Dec 3 20:03 - 20:03 (00:00)
bbs tty4 sun.ihep.ac.cn Wed Dec 3 19:55 - 20:17 (00:22)
bbs tty4 hpws2.ihep.ac.cn Wed Dec 3 19:48 - 19:55 (00:06)
```

sa

sa 命令对/var/account/pacct 文件中记录的有关系统上执行过的命令的信息进行统计。它还可以把这些信息写进一个总结文件(通常是/var/account/sapacct),其中包含某条命令执行的次数和所用系统资源。sa 也可以把这些信息按用户进行统计,并可把结果写进文件/var/account/usracct。如果不指定任何选项,sa 命令将输出在/var/account/pacct 文件中关于所有命令的信息:

```
lark% sa
444079 2340.09re 352.27cp
383 66.44re 65.42cp in.telnetd
132946 1.27re 65.13cp grep
15 127.14re 60.98cp X
135772 4.41re 36.76cp sh
126124 185.88re 24.66cp ping
5 2.56re 19.96cp swriter3
19 0.31re 15.14cp find
```

|       |          |         |          |
|-------|----------|---------|----------|
| 60    | 39.60re  | 13.19cp | netscape |
| 34    | 0.10re   | 5.92cp  | geip     |
| 17418 | 0.14re   | 4.68cp  | lqq      |
| 218   | 38.26re  | 2.71cp  | bbs      |
| 65    | 0.66re   | 2.50cp  | gs       |
| 1688  | 2.61re   | 2.48cp  | cat      |
| 334   | 0.18re   | 0.93cp  | top      |
| 199   | 144.49re | 0.83cp  | smbd     |
| 35    | 42.71re  | 0.81cp  | extern   |
| 188   | 103.78re | 0.71cp  | tcsh     |
| 131   | 0.30re   | 0.66cp  | sort     |
| 482   | 0.37re   | 0.65cp  | sendmail |

最右边是所统计的命令,左边第一栏是命令执行次数,第二栏是命令所耗“real”时间的总和,第三栏是命令所耗用户 CPU 时间 + 系统时间的总和,以分钟为单位。上面输出的第一行是所有命令的总和。输出结果缺省按第三栏的用户 CPU 时间 + 系统时间排序。下面是 sa 命令的一些选项:

-b, --sort-sys-user-div-calls

输出按用户 CPU 时间 + 系统时间的总和除以命令执行次数排序。

-c, --percentages

对统计数据同时输出所占百分比。

-j, --print-seconds

对每项时间统计,以命令执行次数作平均,并以秒为单位。

-l, --separate-times

把用户 CPU 时间和系统时间分别列出,而不是把它们加起来。

-m, --user-summary

对每个月户统计进程数和总的 CPU 时间,输出如下:

|             |        |           |          |
|-------------|--------|-----------|----------|
| lark% sa -m | 444510 | 2344.12re | 354.19cp |
| root        | 15757  | 979.62re  | 149.86cp |
| fang        | 396481 | 338.33re  | 128.42cp |
| chen        | 7409   | 429.77re  | 53.87cp  |
| bbs         | 3541   | 340.31re  | 6.40cp   |
| yuan        | 234    | 0.58re    | 5.48cp   |
| guest       | 16887  | 0.14re    | 4.51cp   |
| bin         | 1247   | 4.16re    | 3.53cp   |
| zheng       | 1181   | 52.55re   | 0.98cp   |
| liu         | 836    | 81.08re   | 0.51cp   |
| mei         | 324    | 32.06re   | 0.31cp   |
| yu          | 123    | 70.42re   | 0.07cp   |
| bbsroot     | 59     | 2.04re    | 0.07cp   |
| yang        | 108    | 12.55re   | 0.06cp   |

|          |     |        |        |
|----------|-----|--------|--------|
| an       | 73  | 0.08re | 0.04cp |
| xu       | 46  | 0.41re | 0.03cp |
| postmast | 122 | 0.02re | 0.02cp |
| nobody   | 82  | 0.01re | 0.02cp |

输出的第一行是所有用户的总和。

`-n, --sort=num-calls`

按命令执行次数排序。

`-r, --reverse-sort`

按反序输出。

`-u, --print-users`

对记账文件中的每个命令,打印用户名,用户 CPU 时间+系统时间,命令名。注意:这个选项覆盖所有其他选项。

`--sort=real-time`

按“real”时间域排序。

**lastcomm**

打印/var/account/pacct 文件中记录的系统中执行过的命令的信息。如果加上命令名、用户名或终端设备名等选项,则只打印包含任一给定选项的记录。如查找哪些用户使用“netscape”命令和哪些用户在 tty6 上登录过,用命令

```
lastcomm netscape tty6
```

将输出记账文件中任何一个域(命令、用户名、终端设备)是 netscape 或 tty6 的命令记录。lastcomm 命令的主要选项为:

`--user name`

列出记账文件中用户名为 name 的记录。

`--command name`

列出记账文件中命令为 name 的记录。

`--tty name`

列出记账文件中终端设备为 name 的记录。

`--strict-match`

列出符合 lastcomm 命令的所有选项的记录。如

```
lastcomm --strict-match --user wang --command ping --tty tty4
```

将列出用户 wang 在 tty4 终端上使用 ping 命令的记录。

上面我们介绍了基本的记账命令。一个实用的记账系统需要用户根据实际需要写出一些处理程序(如一些 Shell 代码),综合以上命令给出的信息,并给出正确的、易读的结果。

## 5.14 软件升级

Linux 是一个快速发展的系统。Linux 计划的协作性的本质使得新软件层出不穷。现有软件则不断有新的版本问世,对 Linux 内核来说尤其如此,内核有时候甚至每天都有

新的版本出现。用户很难做到始终把握软件的最新发展。那么,何时和怎样对系统上的重要软件进行更新呢?

一般来说,应该在确实有更新的需要时才升级系统的一部分软件。如,某个应用程序的新版本修正了当前版本的一些严重的“臭虫”,或提供了对一些很有用的新特性的支持,或比当前版本在运行速度上有明显提高,这时就有必要对这一软件进行更新。单纯为了系统上有某软件的最新版本而进行更新是不合适的。

有时候升级某个软件可能会要求系统上的编译器、函数库或其他软件为最新版本,这样我们就可以了解到系统上有些软件需要更新,这有利于保持系统软件的及时升级。

我们在哪里可以找到 Linux 上软件的新版本呢?首先,每个 Linux 的发行套件,如 Slackware, RedHat 等,都是在不断更新的。这些更新都被做成编译后可直接安装的包,并最先被放置在它们的 FTP 节点上。其次,很多 Linux 软件库,如 sunsite.unc.edu, tsx-11.mit.edu 等,都收集了很多可在 Linux 上编译和运行的软件,这些软件大都以源程序包的形式存放,需要用户在自己的系统上编译安装。另外,越来越多的最初在其他操作系统上开发的软件现在也加进了对 Linux 的支持,它们散布在全世界的 FTP 或 HTTP 服务器上,用户可以用网络上的各种搜索引擎来获取自己需要的软件。最后,周期性地出版的各种有关 Linux 的 CD-ROM 一般都尽量包含出版前的最近的更新。CD-ROM 使用方便,但从出版到用户使用可能已有一段时间,所以有些软件可能已不是最新的版本。

那么,最好的升级方法是什么呢?有些人可能会在他们喜欢的一种 Linux 发行套件有了新版本之后,整个地重装系统。用这种方法升级系统不必担心各种版本的软件混到一起工作。对于那些不能访问 Internet 的用户,这可能是最简单的方法;因为如果用户每两个月才能收到一张新的 CD-ROM,那么系统上的很多软件在他们收到新的 CD-ROM 时已经过时了。

但是,重装系统不是一种好的升级方法。因为现在流行的几种 Linux 发行套件都不是以这种方式升级的,全部重装系统可能是复杂的和耗时的。并且,这种方式的升级将使用户失去已对系统所做的修改和配置,用户必须对任何可能在重新安装过程中被删除的文件做备份。很多新手选择这种升级途径是因为它简单易学。实际上,每次发行套件所作的更新是有限的,所以完全重装通常是不必要的。只要了解一些系统升级的知识就可以轻松升级。

#### 5.14.1 用 Linux 发行套件升级

每个 Linux 发行套件中都有—些文件用来记录更新状况。如在 Slackware 中有文件 slackware/ChangeLog.txt,按时间倒序记录着 Slackware 的更新状况,下面是某一时间 ChangeLog.txt 的头几行:

```
Tue Nov 11 20:33:35 CST 1997
slackware/n6/tcpip.tgz: Upgraded to use pop3d-1.005l, fetchmail-4.3.2,
netkit-rusers-0.11, net-tools-1.432,
wu-ftpd-2.4.2-beta-15.
rootdsk/umsdos.gz: Fixed a bug where the rootdisk would hang after
mounting.
```

```

-----
Mon Nov 3 21:38:35 CST 1997
slackware/a6/ps.tgz: Upgraded to procps-1.2.2.
slackware/a6/diska6: edited.
-----
Fri Oct 31 22:37:18 CST 1997
slackware/a8/shadow.tgz: Upgraded to shadow-971001.tar.gz.
-----
Tue Oct 14 19:32:02 PDT 1997
Slackware 3.4.0 is released.
*****
Tue Oct 14 16:38:44 PDT 1997
slackware/n1/apache.tgz: Upgraded to apache-1.2.4.
slackware/n1/diskn1: edited.
slackware/n6/tcpip.tgz: Upgraded to tcp-wrappers-7.6.

Switched to a working version of ytalk.
Switched to slattach-1.2.0.
-----

```

我们可以看出,1997年10月14日,Slackware 3.4.0版发布。10月31日,shadow包升级为971001版。11月3日,procps包升级为1.2.2版。11月11日,若干TCP/IP程序包得到升级,umsdos中的一个“臭虫”得到修正。如果用户想把自己系统上的以上软件升级,首先把所要的软件包从存放Slackware的某一节点下载到自己的系统上,然后在root账号里用installpkg程序安装,如:

```

lark# installpkg tcpip.tgz
Installing package tcpip.tgz...
Executing install script for tcpip.tgz...

```

表示TCP/IP包已正确安装,pop3d,fetchmail,netkit-rusers,net-tools,wu-ftpd等软件得到升级。

Slackware中提供了一个软件包管理程序pkgtool。它以运行在终端上的菜单的形式提供了在系统上安装、删除软件包的方便界面(图5.3)。

在RedHat中,redhat-4.2/updates/|i386, sparc, alpha|目录中有对RedHat 4.2发行套件中部分软件的升级。redhat-4.2/updates/README.errata文件描述了软件包修正和升级的信息,文中“Overview”部分给出了已修正或升级的软件的列表,如:

#### Overview

- \* 15-May-1997: elm
- \* 21-May-1997: lpr
- \* 05-Jun-1997: mkinitrd
- \* 05-Jun-1997: initscripts
- \* 05-Jun-1997: kaffe
- \* 05-Jun-1997: kernel
- \* 20-Jun-1997: pwdb

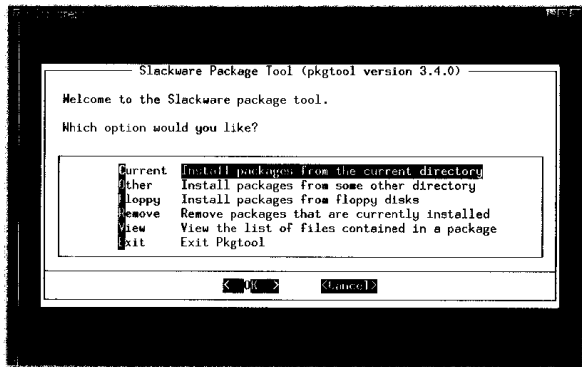


图 5.3 用 pkgtool 安装删除软件包

- \* 20-Jun-1997: XFree86, X11R6.1
- \* 09-Jul-1997: db
- \* 18-Jul-1997: ld.so
- \* 21-Jul-1997: bind
- \* 11-Aug-1997: kernelcfg, pythonlib
- \* 23-Sep-1997: NetKit, traceroute, wu-ftpd, man
- \* 26-Sep-1997: samba

其后的“Detailed Errata”部分简要地说明了对上述软件所做的修正或升级。在 i386 等子目录中,也有一个 OOREADME.errata 文件,是有关对特定平台(如 i386)的修正或升级。用户应仔细阅读这两个文件,以决定是否需要做对应升级。例如,因为新版本的 wu-ftpd 在安全性能上有所改进,用户要升级 FTP 服务器 wu-ftpd,则先到存放 RedHat 的 FTP 节点取来软件包 redhat-4.2/updates/i386/wu-ftpd-2.4.2b15-1.i386.rpm,然后在 root 账号下用

```
lark # rpm -U wu-ftpd-2.4.2b15-1.i386.rpm
```

来对系统上的软件进行升级。

另外,有一个程序 rpm2targz 可以把 RedHat 使用的 .rpm 形式的软件包转换成 .tar.gz 格式,以便于在没有安装 rpm 的系统上安装这些软件包。

### 5.14.2 升级函数库

UNIX 系统上的库文件有两种类型,即档案库(archive)和共享库(shared library)。当编译应用程序时,应用程序可与档案库或共享库或同时与两种类型的库作连接以产生可

执行代码。与档案库的连接是静态连接,即档案库中程序调用的函数的目标代码被拷贝进可执行文件中。与共享库的连接是动态连接,即共享库中的函数并没有被拷贝进可执行代码中,只是在可执行代码中做了一些标记,程序执行时才从共享库中调用相应的函数。动态连接节省了大量盘空间,因为对于每一个库函数,硬盘上只在共享库中存储了一个拷贝。所以现在的 UNIX 系统上大量的程序都是动态连接的。

档案库的文件名格式一般为 `libx.a`,共享库的文件名一般为 `libx.so.version`,如 `libc.a`,`libc.so.5.4.38`,`libX11.a`,`libX11.so.6.1`。对同一个库,系统上一般同时有档案库和共享库。

动态连接的程序在运行时需要在硬盘上寻找所需的共享库。我们可以用 `ldd` 命令来看一个程序需要哪些共享库:

```
lark% ldd /usr/X11R6/bin/xterm
libXaw.so.6 => /usr/X11R6/lib/libXaw.so.6 (0x4000a000)
libXmu.so.6 => /usr/X11R6/lib/libXmu.so.6 (0x40042000)
libXt.so.6 => /usr/X11R6/lib/libXt.so.6 (0x40054000)
libSM.so.6 => /usr/X11R6/lib/libSM.so.6 (0x40092000)
libICE.so.6 => /usr/X11R6/lib/libICE.so.6 (0x4009b000)
libXext.so.6 => /usr/X11R6/lib/libXext.so.6 (0x400b0000)
libX11.so.6 => /usr/X11R6/lib/libX11.so.6 (0x400b0000)
libc.so.5 => /lib/libc.so.5 (0x4014c000)
```

我们看到, `xterm` 程序需要 8 个共享库,前 7 个是 X Window 系统的函数库,最后一个是标准 C 函数库。左边是程序所需共享库的文件名,右边表明在硬盘某目录下找到了所需的库。如果某个共享库没有找到,则这个库名的右边将会出现“not found”的字样。

那么,程序是如何在硬盘上找到共享库的呢?下面作一简单介绍。

用共享库连接一个程序时,程序中被加进一段代码,这段代码用来在程序运行时执行动态连接器 `ld.so`。`ld.so` 负责寻找程序所需要的共享库,并把库函数目标代码加载进内存。动态连接的程序中对用到的共享库中的函数都做了标记,`ld.so` 把标记换成在共享库中对应函数的代码。

`ld.so` 在 `/etc/ld.so.conf` 文件列出的目录中搜寻共享库文件。当前的 Linux 版本的 `/etc/ld.so.conf` 文件一般包含下面几行:

```
/usr/local/lib
/usr/X11R6/lib
/usr/i486-Linuxaout/lib
/usr/openwin/lib
```

`ld.so` 总是搜寻 `/lib` 和 `/usr/lib` 目录。一般没有必要修改此文件。如果想在搜索路径表上增加其他目录,可设置环境变量 `LD_LIBRARY_PATH`(如系统上有些用户可能会在自己的某个目录中创建一些个人用的共享库)。如确实修改了 `/etc/ld.so.conf`,必须用 `ldconfig` 命令重新产生共享库缓存文件 `/etc/ld.so.cache`。`ld.so` 将首先搜寻缓存文件。

当初创建共享库时,一般会给连接命令 `ld` 一个选项“-soname name”,`ld` 将会在共享库中设置 `DT_SONAME`(共享名)域的值为 `name`。如:

```
lark% ld -shared -lm -t -soname libtest.so.0 -o libtest.so.0.1 \
```

```
? test1.o test2.o test3.o test4.o
ld: mode elf i386
test1.o
test2.o
test3.o
test4.o
lark%
```

上例中,创建一个共享库文件 `libtest.so.0.1`,而设置共享名为 `libtest.so.0`,只使用了主版本号,忽略了次版本号,也可以不用“-soname name”指定共享名,这样 `ld.so` 将只识别 `libtest.so`,无法区分版本,下面将会进一步说明这一点。

当一个应用程序与共享库动态连接时,例如:

```
lark% gcc -o main -L/usr/local/lib -ltest main.o
```

`ld` 会在 `/usr/local/lib` 目录下寻找 `-ltest` 指定的共享库 `libtest.so`,所以我们总需要在安装共享库的目录下做一个从 `.so` 文件到 `.so.version` 文件的符号连接。如:

```
lark# ln -s libtest.so.0.1 libtest.so
```

否则 `ld` 在找不到 `libtest.so` 的情形下会寻找 `libtest.a` 并与之静态连接,如果找不到 `libtest.a` 会报告:

```
/usr/some-dir/bin/ld: cannot open -ltest: No such file or directory
```

连接时,应用程序记录的是共享库的共享名 `libtest.so.0`,或在共享库没设定共享名时,记录 `libtest.so`,而非共享库的文件名。这就导致了在程序运行时,`ld.so` 需要找的是硬盘上以共享名命名的文件(或在共享库没设定共享名时,寻找 `libtest.so`),即如果一个程序与上述 `libtest.so.0.1` 动态连接,它在执行时,`ld.so` 却需要找 `libtest.so.0`。所以必须在安装 `libtest.so.0.1` 的目录下(这里是 `/usr/local/lib`)创建 `libtest.so.0` 文件,使它作为到真实的共享库文件 `libtest.so.0.1` 的符号连接。

这样,为使程序能正确地连接和使用共享库 `libtest.so.0.1`,在它的安装目录里必须同时有三个文件:(1) 共享库本身, `libtest.so.0.1`; (2) 共享名指向共享库文件的符号连接, `libtest.so.0`; (3) 指向共享库的符号连接 `libtest.so`。

让我们看一下 `/usr/lib` 目录下的一些库文件:

```
-rw-r--r-- 1 root 88488 Mar 13 1996 libdb.a
lrwxrwxrwx 1 root 10 Nov 30 10:52 libdb.so -> libdb.so.1
lrwxrwxrwx 1 root 15 Nov 30 10:52 libdb.so.1 -> libdb.so.1.85.4
-rwxr-xr-x 1 root 67432 Mar 13 1996 libdb.so.1.85.4
```

我们可以看出,由于共享库 `libdb.so.1.85.4` 的共享名是 `libdb.so.1`,文件 `libdb.so.1` 是指向 `libdb.so.1.85.4` 的符号连接,而 `libdb.so` 则指向 `libdb.so.1`(从而最终指向共享库 `libdb.so.1.85.4`)。

升级函数库时必须用新版本的 `.a` 和 `.so.version` 文件替换老版本的相应文件。对于 `.a` 文件这很容易,只要把新版本的 `.a` 文件拷贝并覆盖旧的版本就可以了。但对于共享库文件 `.so.version` 就复杂得多。因为系统上已有的很多程序可能在运行时要与它动态连接,所以不能简单地删除它或重新命名,必须保证 `ld.so` 能根据程序中存放的共享名找到所需的共享库。假设我们要把上述函数库升级到 1.86.1 版,首先要把新的档案库 `libdb`

.a拷贝到/usr/lib以覆盖老的 libdb.a:

```
lark # cp libdb.a /usr/lib
```

接着拷贝 libdb.so.1.86.1:

```
lark # cp libdb.so.1.86.1 /usr/lib
```

现在必须改变相关的符号连接。libdb.so.1.86.1 的主版本号仍是 1, 其共享名是 libdb.so.1。通常我们不能简单地删除 libdb.so.1, 然后再用 ln -s 命令建立符号连接 libdb.so.1 指向新的共享库 libdb.so.1.86.1。因为系统上可能正运行着某个程序用着 libdb.so.1, 删除 libdb.so.1 可能会导致程序运行失败(对于像 libc 这样的函数库尤其是这样, 因为几乎系统上所有的程序都要用到 libc, 删除它的共享名文件后甚至 ln 命令都无法运行)。我们必须用 ln -sf 命令使改变符号连接的过程一次完成:

```
lark # ln -sf /usr/lib/libdb.so.1.86.1 /usr/lib/libdb.so.1
```

现在我们看到, 由于主版本号没改变, 原来的 libdb.so 还是经由 libdb.so.1 指向新的共享库文件 libdb.so.1.86.1, 所以无须改变 libdb.so。以上几步完成后, 我们就可以安全地删除老的共享库文件 libdb.so.1.85.4 了。但是, 如果是升级到 libdb.so.2.0.5, 由于新共享库的共享名是 libdb.so.2, 我们就必须改变 libdb.so 的连接了:

```
lark # cp libdb.a /usr/lib
```

```
lark # cp libdb.so.2.0.5 /usr/lib
```

```
lark # ln -s /usr/lib/libdb.so.2.0.5 /usr/lib/libdb.so.2
```

```
lark # ln -sf /usr/lib/libdb.so.2 /usr/lib/libdb.so
```

现在我们再列一下 /usr/lib 目录中的 libdb 文件:

```
-rw-r--r-- 1 root 88488 Mar 13 1996 libdb.a
lrwxrwxrwx 1 root 10 Nov 30 10:52 libdb.so -> libdb.so.2
lrwxrwxrwx 1 root 15 Nov 30 10:52 libdb.so.1 -> libdb.so.1.85.4
-rwxr-xr-x 1 root 67432 Mar 13 1996 libdb.so.1.85.4
lrwxrwxrwx 1 root 14 Aug 15 21:40 libdb.so.2 -> libdb.so.2.0.5
-rwxr-xr-x 1 root 68158 May 18 1995 libdb.so.2.0.5
```

这样以前编译的程序仍可以动态连接老的共享库 /usr/lib/libdb.so.1 而正确运行, 而以后编译的程序将连接新的共享库 /usr/lib/libdb.so.2。

如果我们认为 2.0.5 版本有足够好的向下兼容性, 用 1.85.4 版连接的程序能在 2.0.5 版下正常运行, 我们可以用下列命令使 libdb.so.1 也指向 libdb.so.2.0.5:

```
lark # ln -sf /usr/lib/libdb.so.2.0.5 libdb.so.1
```

这样, 以前编译的程序在运行时也动态连接新版本的共享库 libdb.so.2.0.5。而老的共享库可以删除。

那么在哪里可以找到函数库的新版本呢? 除了各种 Linux 发行套件的经常性升级外, 一些基本的系统函数库(如标准 C 函数库 libc, 数学函数库 libm 等)可以在 sunsite.unc.edu 的 /pub/Linux/GCC 目录中找到。那里包含着 Linux 版本的 gcc 编译器、函数库、头文件和其他有用的工具。用户可参见相关的 README 或 release 文件了解怎样安装。

### 5.14.3 升级编译器

系统上另一个重要的必须经常升级的部分是 C 编译器及相关工具。这包括 gcc

(GNU C 和 C++ 编译器)、连接器、汇编器、C 预处理器和编译器用的各种头文件和函数库。当前版本的 gcc 软件包可从很多 FTP 节点,如 sunsite.unc.edu 的/pub/Linux/GCC 目录中取来。通常,升级编译器软件包只要简单地在 root 用户下拆开几个 tar 文件包就可以了。

查看系统当前的 gcc 版本用命令:

```
gcc -v
```

它通常显示类似下面两行信息:

```
Reading specs from /usr/lib/gcc-lib/i486-unknown-linux-gnu/lib1/2.7.2.3/specs
gcc version 2.7.2.3
```

gcc 一般放在/usr/bin下,它本身是一个调用真实的编译器和代码产生工具的前端程序,后者放在以下格式的目录下:

```
/usr/lib/gcc-lib/machine/version
```

如对于上面的例子,真实的编译器和代码产生工具放在

```
/usr/lib/gcc-lib/i486-unknown-linux-gnu/lib1/2.7.2.3
```

目录下。

#### 5.14.4 升级其他软件

除了上述软件以外,系统上其他一些软件也需要周期性地更新。当然,我们最好而且只须更新我们得到的软件。

更新软件的第一步是获取软件的最新版本,这通常是一个用 tar 打包 gzip 压缩的 .tar.gz 文件,当然也可能是 .tar.Z(用 compress 压缩,可用 uncompress 或 gzip 解压缩)等格式的文件。软件包的形式可能是已编译好的二进制码及相关文件,这种包在系统上打开后就可直接安装;也可能是源程序包,其中提供了软件的全部或部分源程序,用户必须用适当的命令去编译和安装。一般免费的软件都提供完整的源程序包。这不仅使用户可以针对自己系统的情况来编译软件,而且为有兴趣的用户研究和进一步开发软件提供了可能。

一般来说,用户在取来软件包后,应该先用 tar tvf(或 tar ztvf)看一看 .tar 或 .tar.gz(.Z)文件中究竟放了些什么文件,判断是源程序包还是二进制码包,是否有 README 或 INSTALL 等安装说明文件。对于二进制包,大部分情况下可以直接把包打开在/或/usr目录下,然后删除同一程序老版本的残留文件,老版本的文件可能有些并没有被新版本覆盖掉,如果不删除,有可能会影响新版本程序的运行。有些二进制包可能需要用包里提供的安装程序来安装,这时先把二进制包打开在一个临时目录里,然后遵循安装说明里的信息和步骤进行安装。

对于源程序包来说,编译和安装过程稍微复杂一些。首先,把源程序包打开在某个工作目录下。一般软件的源程序包里总有名字类似 README,INSTALL 的说明文件,通常放在软件包的第一层目录下。这些文件非常重要,说明了软件的适用平台,编译它对系统和其他软件的要求,编译和安装的步骤及其他注意事项。用户应首先仔细阅读这些文件。现在越来越多的免费软件包遵循类似的配置、编译和安装约定。用户只须了解一般的编译安装步骤就可以顺利安装大部分的软件包。

一般来说,编译安装一个软件包的基本步骤如下:

### 1. 产生 Makefile 文件

很多遵循 GNU 的规范软件包总在源程序目录里提供一个 Shell 命令文件 `configure`, 执行这个文件可以对 UNIX 系统的软件配置环境进行检测, 产生一些头文件, 设置一些反映系统配置的宏定义, 并从 `Makefile.in` 文件产生正确的 `Makefile` 文件。如果软件包含有这么一个文件, 首先执行它:

```
./configure 或 sh configure
```

`configure` 的一个重要选项是 `--prefix`, 它指定编译完成后的安装目录, 并设置相应的编译选项。因为有些程序需要在编译时知道它以后运行时从哪里找到它用到的文件。如:

```
./configure --prefix = /usr/local
```

指定软件安装在 `/usr/local` 下的各目录下, 如可执行文件进入 `/usr/local/bin`, 函数库进入 `/usr/local/lib`, 头文件进入 `/usr/local/include` 等等。

对于 X Window 系统的应用程序, 有时需要用 `"xmkmf"` 命令从 `Imakefile` 文件产生 `Makefile` 文件, 这将在有关 X Window 系统的一章中加以说明。

### 2. 检查 Makefile 和 config.h

在软件包没有提供 `configure` 或 `configure` 产生的 `Makefile` 没有反映系统的实际情况下, 检查 `Makefile` 文件, 它控制编译器编译软件包。有时候可能需要对 `Makefile` 作适当的修改, 哪些地方要作修改在 `Makefile` 文件本身或软件安装说明文件中应该有介绍。有些软件还要求用户检查和修改 `config.h` 文件。一般用 `configure` 产生的 `Makefile` 和 `config.h`, 很少需要手动修改。

### 3. 编译

通常用 `make` 命令实现, `make` 命令读进 `Makefile` 文件, 并用其中设定的编译命令及选项来编译源程序包。`make` 命令的可用“目标”在 `Makefile` 文件中有设定, 通常 `make all` 表示编译 `Makefile` 中列出的所有目标(例如, 软件包包含几个程序, 但依据 `Makefile`, `make` 命令只缺省编译一个)。

### 4. 安装

通常用 `make install` 命令实现。它把可执行文件、库文件、头文件、信息文件及程序运行时要用到的文件安装到 `Makefile` 指定的目录中。因为 `--prefix` 缺省是 `/usr` 或 `/usr/local` 目录, 软件一般都被安装进系统目录, 所以 `make install` 一般需要在 `root` 账号下进行。

编译和安装一个新软件时, 有可能会遇到各种各样的问题, 特别是对于还没有在 Linux 上测试过的软件。碰到问题时要耐心, 细致, 仔细阅读软件包的各个说明文件, 检查有关的 `Makefile`, 看是否有定义不符合系统, 如编译器的选项等。

## 5.15 编译新内核

编译新内核对于一个喜欢钻研操作系统的 Linux 爱好者(我们常称之为 hacker,但并不带贬义)来说是一种非常重要的技巧,但任何系统管理员都必须对其有所了解。

在什么情况下我们有必要在系统上建立一个新的内核呢?想删除内核中本系统用不上的设备驱动程序就必须重建内核,这将减少内核本身占用的内存数量,因为内核是常驻内存的,而且内核占用的内存存在可用内存缺乏时并不能为其他程序所利用。如果现有核心不支持或没有把我们的系统上某些硬件设备的驱动程序编译进去,我们就必须把内核升级到较新的版本。比较新的内核版本一般来说会提供更多的设备驱动程序。新的版本的内核通常对旧版本的一些错误有重要的修正。它们可能比旧版本效率更高,更稳定。某些应用程序可能需要较新的内核版本支持才能运行。对于那些开发内核代码的人来说,使自己使用的机器的内核与最新的发展同步就更重要了,这样他们才可能得知自己正在参与开发的代码现在有什么新的改变。当然,大部分人是因为他们需要一些新的设备驱动程序以及错误修正的部分而更新内核。

我们可以用 `uname -a` 命令来查看系统当前的内核版本:

```
lark% uname -a
```

```
Linux lark 2.1.57 #2 Sun Oct 12 10:26:31 CST 1997 i586 unknown
```

我们看到系统的内核版本是 2.1.57,第二次(#2)编译于 1997 年 10 月 12 日 10:26:31,星期天,节点名是 lark,CPU 类型是 Pentium。

Linux 内核是很多人共同工作的结晶。每个小组开发其中的一小块,代码的某些部分是不同设计思想的集合体。但总的来说,内核代码是干净和统一的,有兴趣开发它的人应该不会遇到太大的麻烦。但是,由于有很多人同时在对内核做大量的开发工作,核心的更新是很快的,有时候甚至每天都有新的版本发布。这是因为很多设备驱动程序包含在内核代码里,每次有人升级了一个驱动,就必须发布一个新的版本。现在的趋势是越来越多地使用可装载设备驱动,这样这些驱动的维护者就可以独立于主要的内核来发布它们,这就缓和了以前飞快的升级速度。

现在,Linux Torvalds 维护着“官方”的内核发布版。若要对某个“bug”进行修正或在内核中加进新的特性,只要把它发给 Linux 进行整理。

内核版本号遵循以下约定:

```
major.minor.patchlevel
```

major 是主版本号,它很少改变。minor 是次版本号,major.minor 就是当前内核的版本号。patchlevel 是对当前内核的修正次数。如 2.2.6 表示对内核 2.2 版的第 6 次修正版。

遵循约定,偶数的内核版本(如 1.2,2.0,2.2 等)是“稳定”的发布版,对它的修正通常是去除程序中的小毛病,即所谓“臭虫”,而没有新的特性加入。奇数的内核版本(如 1.3,2.1 等)是“开发”版,对它的修正包括开发者想加进的新代码及对原代码臭虫的修正。当一个开发版的内核已经足够成熟和稳定,可以提供给广大用户使用,它就被命名成下一个偶数版,而开发版也跟着命名为下一个奇数版。

例如,内核版本 1.0 和 1.1 是并行存在的。对 1.0 版的修正仅仅是对已存在的代码

的臭虫的修正。对 1.1 版的修正包括对臭虫的修正和很多新代码,如新的设备驱动、新的特性等等。当内核版本 1.1 已足够稳定(大约是 1.1.65 版)时,它就被改成 1.2 版,并做了一个拷贝命名为 1.3 版。1.2 版和 1.3 版又是并行开发的,1.2 是稳定的内核,而 1.3 是开发版,将又很多新的特性被加进去。当 1.3 版开发到一定程度时,实际上是推出了 2.0 版,而不是 1.4 版。在写作本书的时候,稳定版本即是 2.2 版,而开发版本是 2.3 版。  
**注意:**以上约定只适用于 1.0 版以后的 Linux,在 1.0 版之前是只有一个“当前”版本的。

后来发现对于用户和内核开发者来说,有两个并行的版本既利于稳定的使用,又利于活跃的开发,才改用现行约定。但是,开发版肯定是有许多臭虫并且不稳定的,所以一般用户应使用稳定的发布版,只有当需要开发版的新特性并愿意承受因此带来的问题时,才推荐编译和使用开发版的内核。

当前内核的源程序放在 `/usr/src/Linux` 目录里。如果用户想从当前内核的源程序重建内核,则不需要获取别的文件或适用任何修正。如果用户想把内核升级到一个新的版本,则须按下节的指导获取内核的源代码和“补丁”文件。

### 5.15.1 获取内核源程序

正式发布的 Linux 内核源程序是以 `.tar.gz` (或 `.tar.bz2`) 格式打包并压缩存储的。对每一个版本有一个名为 `Linux-x.x.xx.tar.gz` 的文件,如对 2.2.6 版有 `Linux-2.2.6.tar.gz`。这个文件包含 2.2.6 版 Linux 内核的全部源代码。同时对每一个版本还有一个对一个版本进行修正的“补丁”文件,文件名格式为 `patch-x.x.xx.gz`,如对 2.2.6 版有 `patch-2.2.6.gz`,这个文件是用 `diff` 命令比较 2.2.5 和 2.2.6 版源程序发布包的所有文件所得结果压缩后得到的。这意味着如果有了 `Linux-2.2.5.tar.gz` 和 `patch-2.2.6.gz`,我们就可以用 `patch` 命令和 `patch-2.2.6.gz` 来修改 2.2.5 版的源程序而得到 2.2.6 版的源程序。类似地,从 `Linux-2.2.4.tar.gz`,`patch-2.2.5.gz` 和 `patch-2.2.6.gz`,我们同样可以得到 2.2.6 版的源代码。所以,只要我们有一个版本的内核的完整的源程序包和以后各个版本的“补丁”文件,我们就可以得到最新版本的内核源代码。

例如,假设系统当前的内核版本是 2.2.1,要升级到 2.2.6 版。若用户手头没有 2.2.1 版的源程序包,就必须直接取来 `Linux-2.2.6.tar.gz`;若用户已有 2.2.1 版的源程序包,只需取来 `patch-2.2.2.gz` 直到 `patch-2.2.6.gz`。一般“补丁”文件的体积要小得多,所以对需要从网络上下载源程序的用户来说要经济得多。

Linux 内核的源程序包可以通过匿名 `ftp` 从节点 `ftp.kernel.org` 的 `/pub/Linux/kernel` 目录下载。但从它的镜像节点或遍布全球的存有 Linux 内核的其他节点下载也许更快。`ftp.kernel.org` 在各地的正式镜像节点都有统一格式的节点名,如在美国的镜像节点为 `ftp.us.kernel.org`,在英国的镜像节点为 `ftp.uk.kernel.org`,在中国的镜像节点为 `ftp.cn.kernel.org`。其他一些存有 Linux 内核的节点及目录如下:

|      |                                                                             |
|------|-----------------------------------------------------------------------------|
| 芬兰:  | <code>ftp.funet.fi:/pub/Linux/PEOPLE/Linus</code>                           |
| 美国:  | <code>sunsite.unc.edu:/pub/Linux/kernel</code>                              |
| 美国:  | <code>tsx-11.mit.edu:/pub/Linux/sources/system</code>                       |
| 英国:  | <code>sunsite.doc.ic.ac.uk:/pub/UNIX/Linux/sunsite.unc-mirror/kernel</code> |
| 奥地利: | <code>ftp.univie.ac.at:/systems/Linux/sunsite/kernel</code>                 |

```

德国:      ftp.Germany.EU.net:/pub/os/Linux/Local.EU.net/Kernel/Linux
德国:      sunsite.informatik.rwth-aachen.de:/pub/Linux/PEOPLE/Linux
法国:      ftp.ibp.fr:/pub/Linux/sources/system/patches
澳大利亚:  sunsite.uns.edu.au:/pub/Linux/kernel
日本:      sunsite.sut.ac.jp:/pub/archives/Linux/kernel

```

用户应该从离自己的网络最近的节点下载。一般来说, sunsite.uns.edu 在世界各地的镜像节点是一种很好的选择, sunsite.uns.edu:/pub/Linux/MIRRORS 文件列出了它的镜像节点。

Linux 内核源程序在各节点上一般是按版本分目录存放, 如 v2.1/子目录里存放 2.1.xx 版本的内核及“补丁”文件, v2.2/子目录里存放 2.2.xx 版本的内核及“补丁”文件。

对不能使用 ftp 的用户, 有个存放 Linux 的 BBS 系统列表会定期刊登在新闻组 comp.os.linux.announce 上, 试着到那里去寻求帮助。

### 5.15.2 解开源程序包

在解拆源程序包之前, 首先要明确编译 Linux 内核所需要的盘空间。压缩后的 Linux 内核的全部源代码在 2.0.30 版时约占 6MB, 在 2.2.6 版时则已有 13MB, 解压后则分别有 25MB 和近 54MB; 这还不包括编译所产生的所有文件, 编译完成后总共大约要占 70~90MB 的盘空间, 这取决于用户的选择和配置, 如选择编译的驱动程序的数量。所以, 一般来说, 如果系统上有许多种硬件设备, 要准备足够的硬盘空间来容纳这个配置。

以 root 登录, cd 到 /usr/src 目录(以下的叙述如无特殊说明都相对于这个目录)。这个目录下应该有一个叫 Linux 的子目录, 其中存放着系统当前版本的内核的源代码。如果系统在安装时没有选择安装内核的源代码(Slackware 中的 K 系列), 则这个目录中至少应该有 Linux 内核的头文件(在 Slackware 的 D 系列中), 这些头文件在 Linux/include 子目录下, 它们在编译大多数 Linux 下的系统软件时是需要的。应该保留 Linux/include 目录下的所有文件, 以便在新的内核没有编译成功或不能稳定运行时可以恢复系统以前的内核配置。但将要编译的新内核的源程序也将存放在 Linux 子目录中, 为了不致混淆, 必须在解开新的内核的源程序包前, 先把 Linux 目录改成其他的名字, 最好是根据系统目前使用的内核版本来修改这个子目录的名字。我们可以用“uname -r”命令来显示当前内核的版本号。如果“uname -r”显示“2.0.30”, 我们就可以把 Linux 子目录改为 Linux-2.0.30。如果用户保存有当前内核的源程序包, 为节省盘空间, 可把除 Linux/include 子目录外的其他子目录都删掉。总之要确定在解开新内核的源程序包之前, /usr/src 目录下没有“Linux”这个子目录。

现在我们就可以解开新内核的源程序包了。在 /usr/src 目录下, 用:

```
lark # zcat Linux-x.x.xx.tar.gz | tar xvf -
```

或

```
lark # tar zxvf Linux-x.x.xx.tar.gz
```

来解开内核源程序包。如果用户拿到的是 .tar 文件而不是 .tar.gz 文件, 就用

```
lark # tar xvf Linux-x.x.xx.tar
```

这里文件名中的 x.x.xx 应换成真正的版本号。我们将会看到屏幕上闪过所有源程序的名字及所在目录。当以上命令完成以后, 应该会出现一个新的 Linux 子目录, 其中包含

x.x.xx 版的 Linux 内核的全部源程序。

如果我们需要对 x.x.xx 版的源程序作修正以得到更新版本的内核源程序,我们必须用 patch 命令把“补丁”文件所作的修改加进 Linux 目录中的 x.x.xx 版的源程序中。

下面举例说明:假如我们要编译 2.0.31 版的内核,我们有 Linux-2.0.30.tar.gz 和 patch-2.0.31.gz,从上面的叙述可知,这两个文件可产生 2.0.31 版的内核的全部源程序(相当于 Linux-2.0.31.tar.gz)。在/usr/src 目录下,我们可用如下步骤生成 2.0.31 版的源程序:

```
lark # tar zxvf Linux-2.0.30.tar.gz
lark # gzip -d patch-2.0.31.gz
lark # patch -p0 -s < patch-2.0.31
```

最后一条命令把 patch-2.0.31 文件中包含的 2.0.31 版对 2.0.30 版所作的修改加进 Linux 目录下 2.0.30 版的源程序。又如,我们要编译 2.2.6 版的内核,我们有以下文件:Linux-2.2.3.tar.gz, patch-2.2.4.gz, patch-2.2.5.gz, patch-2.2.6.gz。在/usr/src 目录下,我们可用如下步骤生成 2.2.6 版的源程序:

```
lark # tar zxvf Linux-2.2.3.tar.gz
lark # gzip -d patch-2.2.4.gz patch-2.2.5.gz patch-2.2.6.gz
lark # patch -p0 -s < patch-2.2.4
lark # patch -p0 -s < patch-2.2.5
lark # patch -p0 -s < patch-2.2.6
```

这里,patch 命令加-s 选项意思是只有在 patch 出错时才给出屏幕信息。注意使用 patch 时要从低版本到高版本逐个修正。

### 5.15.3 编译前的准备工作

首先我们要注意编译新的内核对系统上其他软件版本的要求。在 Linux/Documentation 目录里,有一个文件 Changes,列出了正确编译内核和支持内核正确运行所需要的各种软件版本说明。用户不一定需要最新版本的软件才能正确编译和运行新内核,但每个版本的新内核对一些软件版本总有最低要求。例如,2.0.31 版本的内核对其他软件版本的要求如下:

|                        |                              |
|------------------------|------------------------------|
| Kernel modules         | 2.0.0                        |
| PPP daemon             | 2.2.0f                       |
| Dynamic linker (ld.so) | 1.7.14                       |
| GNU CC                 | 2.7.2.1                      |
| Binutils               | 2.7.0.3                      |
| Linux C Library        | Stable; 5.2.18, Beta; 5.4.33 |
| Linux C++ Library      | 2.7.2.1                      |
| Termcap                | 2.0.8                        |
| Procps                 | 1.0f                         |
| Gpm                    | 1.10                         |
| SysVinit               | 2.64                         |
| Util-Linux             | 2.5                          |
| Mount                  | 2.5p                         |

|                               |            |                              |
|-------------------------------|------------|------------------------------|
| Net-tools                     | 1.32-alpha |                              |
| Kbd                           | 0.91       |                              |
| 右侧所列版本号是最低要求。而 2.2.6 版的内核则要求: |            |                              |
| Kernel modules                | 2.1.121    | ;insmod -V                   |
| Gnu C                         | 2.7.2.3    | ;gcc --version               |
| Binutils                      | 2.8.1.0.23 | ;ld -v                       |
| Linux libc5 C Library         | 5.4.46     | ;ls -l /lib/libc.so.*        |
| Linux libc6 C Library         | 2.0.7pre6  | ;ls -l /lib/libc.so.*        |
| Dynamic Linker (ld.so)        | 1.9.9      | ;ldd -v                      |
| Linux C++ Library             | 2.7.2.8    | ;ls -l /usr/lib/libg++*.so.* |
| Procps                        | 1.2.9      | ;ps --version                |
| Procinfo                      | 15         | ;procinfo -v                 |
| Psmisc                        | 17         | ;pstree -V                   |
| Net-tools                     | 1.49       | ;hostname -V                 |
| Loadlin                       | 1.6a       |                              |
| Sh-utils                      | 1.16       | ;basename --v                |
| Autofs                        | 3.1.1      | ;automount --version         |
| NFS                           | 2.2beta40  | ;showmount --version         |
| Bash                          | 1.14.7     | ;bash --version              |
| Ncpfs                         | 2.2.0      | ;ncpmount -v                 |
| Pcmcia-cs                     | 3.0.7      | ;cardmgr -V                  |
| PPP                           | 2.3.5      | ;pppd -v                     |
| Util-Linux                    | 2.9g       | ;chsh -v                     |

要升级内核的用户并不一定需要上面列出的所有软件。这取决于用户将编译哪些驱动。注意上面对 2.2.6 版的列表的第三栏给出了列出当前系统各种软件版本的命令。如对于 Gnu C,

```
lark# gcc --version
2.7.2.3
```

表示当前版本是 2.7.2.3。对于 Binutils,

```
lark# ld -v
GNU ld version 2.8.1 (with BFD Linux-2.8.1.0.1)
```

表明当前版本是 2.8.1.0.1。如果系统上有一些编译和运行新内核必需的软件版本较低,应在编译前先予以更新。

其次,检查一下 `/usr/include/asm`, `/usr/include/Linux`, `/usr/include/scsi` 是不是正确地符号连接到 `/usr/src/Linux/include` 下的对应目录。如果不是,可以这样做:

```
lark# cd /usr/include
lark# rm -fr asm Linux scsi
lark# ln -s /usr/src/Linux/include/asm-i386 asm
lark# ln -s /usr/src/Linux/include/Linux Linux
lark# ln -s /usr/src/Linux/include/scsi scsi
```

上面 `asm-i386` 目录对于其他体系结构,如 `sparc` 或 `alpha`,则对应地换成 `asm-sparc` 或 `asm-alpha`。

最后,为确保源程序目录树里没有残留的.o文件和其他从属文件,使用如下命令彻底清理一下:

```
lark # cd /usr/src/Linux
```

```
lark # make mrproper
```

到了现在,我们可以进入配置和编译内核的阶段了。

#### 5.15.4 配置内核

在Linux目录下执行“make config”命令来配置内核。所谓配置内核就是选择内核支持哪些特性和编译哪些设备驱动。“make config”需要bash才能工作,它按\$BATH, /bin/bash, /bin/sh的顺序来搜寻bash。与“make config”等效的配置命令有“make menuconfig”和“make xconfig”,分别用菜单和X Window窗口的形式更方便地实现内核的配置。

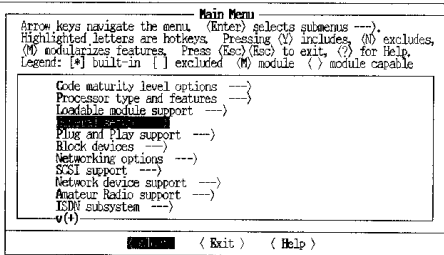


图 5.4 Linux 内核的菜单配置界面 (V2.2.1)

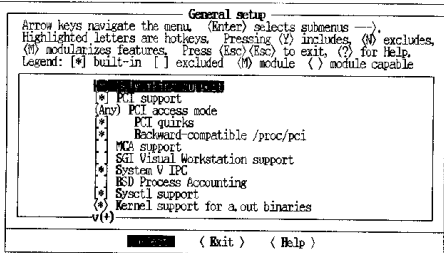


图 5.5 General setup 菜单 (V2.2.1)

那么,配置内核主要是做了哪些选择呢?用“make menuconfig”或“make xconfig”命令,对于内核 2.2.6 版,我们可以较清楚地看到,内核的配置主要分为以下几类:

|                                                  |                                     |
|--------------------------------------------------|-------------------------------------|
| Code maturity level options                      | (代码成熟程度选项)                          |
| Processor type and features                      | (处理器类型和特性)                          |
| Loadable module support                          | (可加载模块支持)                           |
| General setup                                    | (一般设置)                              |
| Plug and Play support                            | (即插即用支持)                            |
| Block devices                                    | (块设备)                               |
| Networking options                               | (网络选项)                              |
| SCSI support                                     | (SCSI 支持)                           |
| Network device support                           | (网络设备支持)                            |
| Amateur Radio support                            | (业余收音机支持)                           |
| ISDN subsystem                                   | (ISDN 子系统)                          |
| CD-ROM drivers(not for SCSI or IDE/ATAPI drives) | (CD-ROM 驱动(不含 SCSI 和 IDE/ATAPI 驱动)) |
| Filesystems                                      | (文件系统)                              |
| Console drivers                                  | (控制台驱动)                             |
| Character devices                                | (字符设备)                              |
| Sound                                            | (声音)                                |
| Kernel hacking                                   | (内核监视)                              |

对每一类都有一些问题有待用户做出选择。如列入“General setup”一类的有以下问题:

|                                        |
|----------------------------------------|
| Networking support                     |
| PCI support                            |
| MCA support                            |
| SGI Visual Workstation support         |
| System V IPC                           |
| BSD Process Accounting                 |
| Sysctl support                         |
| Kernel support for a.out binaries      |
| Kernel support for ELF binaries        |
| Kernel support for MISC binaries       |
| Parallel port support                  |
| Advanced Power Management BIOS support |

对每一个问题一般有“y”和“n”两种选择,对某些问题还有第三种选择“m”。选择“y”表示对相应特性的支持或相应的设备驱动将被编译进内核,选择“n”则表示不把对相应特性的支持或相应的设备驱动编译进内核,即内核将不支持相应特性或设备。选择“m”则表示把对相应特性的支持或相应的设备驱动编译成可加载模块。

什么是可加载模块?可加载模块是内核的一部分(通常是设备驱动程序),但是并没有编译到内核里面去。它们被分别编译,然后几乎可以在任何时候插入运行中的内核或从中取出。这已经成为一种增加东西到内核里去的较好方式,因为这种方法明显地更为模块化。许多常用的设备驱动程序,例如网卡、声卡驱动程序以及磁带机的驱动程序就可以编译成可加载模块。

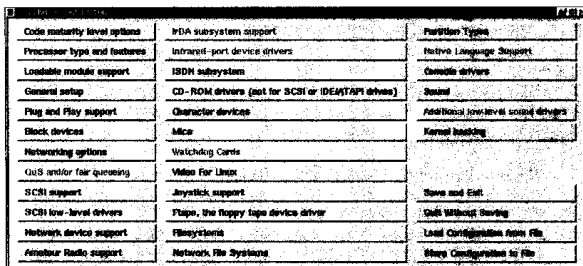


图 5.6 Linux 内核的 X Window 配置界面

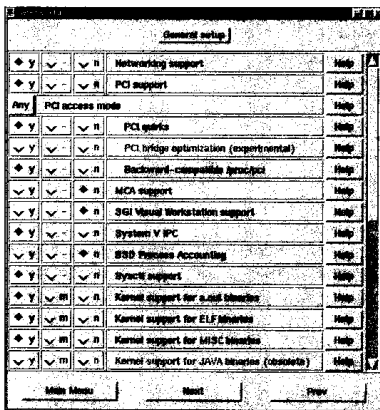


图 5.7 General setup 窗口

下面我们分类介绍一些重要的配置选项。在每个选项前的“[ ]”表示有“y”和“n”两种选择，“< >”表示有“y”，“n”和“m”三种选择。有一些很明显或是不重要的选项不在此说明。如果用户在配置时有不明确的地方，请使用配置时的帮助文件推荐的选择。

### 1. Code maturity level options (代码成熟程度选项)

#### ☐ Prompt for development and/or incomplete code/drivers

某些代码或驱动尚属测试阶段,尚不稳定,是否提供选择?使用这些代码或驱动可能会遇到些问题,但如愿意参与测试或确实需要其中的某些新特性,回答“y”,则在以下各类问题中将提供有关这些测试性代码或驱动的选择。否则回答“n”。

### 2. Processor type and features(处理器类型和特性)

#### (386, 486, Pentium, PPro) Processor family

选择 CPU 类型,用于优化目的。选择 386 优化的内核将不能在 486 以上机型上运行。一般根据实际情况选择即可。不推荐往高或往低选。但如果使用低于 2.7 版本的 gcc,即使用 Pentium 或 Pentium Pro,也只能选择 486。

#### ☐ Kernel math emulation

是否模拟数学协处理器?386 和 486SX 没有内置的数学协处理器(除非增加了 487SX 或 387),必须回答“y”。而 486DX 和 Pentium 系列都有内置的数学协处理器,可以回答“n”,不过回答“y”也无关紧要,只是使内核稍大一些而已,内核在引导时若检测到数学协处理器,会直接使用它而忽略模拟。

### 3. Loadable module support (可加载模块支持)

#### ☐ Enable loadable module support

是否支持使用可加载模块?现在越来越多的驱动倾向于使用可加载模块。如果此处回答“n”,则以下所有特性或驱动不能编译成可加载模块。应回答“y”。

#### ☐ Kernel module loader

是否支持内核守护程序?如果用户选择把某些驱动或文件系统编译成可加载模块,则用户在使用相关功能前应加载(用 insmod 或 modprobe)相应模块。如果此处回答“y”,则内核和用户级守护程序“kerneld”配合将会在适当的时候自动加载和卸载各种模块。如不确定,回答“y”。

### 4. General setup (一般设置)

#### ☐ Networking support

是否支持网络?一般应回答“y”,因为即使机器不与任何网络联结,有些应用程序也总是需要网络。

#### ☐ PCI support

对于主板使用了 PCI 总线的机器,回答“y”。否则回答“n”。

#### ☐ System V IPC

是否支持 System V 的进程间通讯?进程间通讯是一类库函数和系统调用,它使进程同步并可交换信息。很多程序需要它才能运行,如 Linux 下的 DOS 模拟 dosemu。应回答“y”。

< > Kernel support for a.out binaries

内核是否支持 a.out 格式的运行时库和执行代码?

< > Kernel support for ELF binaries

内核是否支持 ELF 格式的运行时库和执行代码?

a.out (Assembler. OUTput) 是早期的 UNIX 的运行时库和执行代码使用的一类格式。Linux 曾使用其中的 QMAGIC 和 ZMAGIC, 后来渐渐地被 ELF 格式取代。但现在 Linux 上有些程序仍使用 a.out 格式, 所以应保留对它的支持。回答“y”。ELF (Executable and Linkable Format) 现在被广泛地使用在不同体系结构和操作系统上。现在很多执行代码只用 ELF 格式发布。绝对回答“y”。

< > Kernel Support for MISC binaries

内核是否支持加速直接运行一些二进制代码的支持? 在 Linux 上, 有些代码需要专用的解释器才能执行, 如 Java, Python 和 Emacs-Lisp 等; 如果内核加速此项支持, 我们在运行它们时只需要打入它们的名字, Linux 的内核会负责寻找相应的解释器来执行它们。本配置项使上面“Kernel support for JAVA binaries”的配置成了过时的和没必要的。

< > Parallel port support

< > PC-style hardware

如果用户想使用并行口设备(如打印机等)就必须选择上面两项。它们提供并行口支持和 PC 并行口支持。也可以将它们编译成可加载模块, 分别产生 parport.o 和 parport\_pc.o 文件。

[ ] Advanced Power Management BIOS support

是否支持具有高级电源管理性能的 BIOS? 一般用于笔记本电脑。如果用户要编译将用于笔记本电脑的内核, 应当回答“y”。

## 5. Block devices (块设备)

< > Normal PC floppy disk support

[ ] Enhanced IDE/MFM/RLL disk/cdrom/tape/floppy support

[ ] Include IDE/ATAPI CDROM support

以上三项分别是当前 PC 上最常用的软盘支持、IDE 接口的设备支持和用 ATAPI 协议的 CD-ROM 支持, 只要系统上有软驱、IDE 硬盘和 CD-ROM 驱动器, 都应该回答“y”。如果系统上有其他 IDE/ATAPI 设备, 如磁带机或软驱, 可选择对应的设备驱动。

[ ] CMD640 chipset bugfix/support

[ ] RZ1000 chipset bugfix/support

CMD640 和 RZ1000 普遍用在很多 486 和 Pentium 主板上, 但它们有一些重要的设计缺陷会导致系统运行失败, 一般以上两项都应回答“y”。

< > Loopback device support

这项支持使用户可以把一个普通文件用作块设备。用户可以像用硬盘分区、光盘等真实的块设备一样在这个文件上创建文件系统, 把它挂到系统上等等。这对在烧一张光盘前检查 ISO 9660 文件系统映像文件非常有用。一般可把它编译成模块, 即选“m”。

## 6. Networking options (网络选项)

如果用户在上面 General setup (一般设置) 中选择支持网络, 则可在本类和下面的 Network device support (网络设备支持) 类问题中进一步选择。本类问题是关于各类网络协议和服务。一般应至少选择下面一项:

### [ ] TCP/IP networking

它是 Internet 和大多数局域以太网上使用的协议簇。某些程序如 X Window 系统总是需要 TCP/IP 支持。即使系统没有与任何别的计算机相连, 也可以有叫做 loop-back 的网络形式使本机内部实现网络功能, 如使用 ping, telnet 等。回答“y”有益无害。

如果用户希望加进对其他协议或服务的支持, 可选择对应的选项, 如选择支持 IPX 协议或 AppleTalk DDP 协议等。

## 7. SCSI support (SCSI 支持)

如果系统上有 SCSI 设备, 如硬盘、磁带机或 CD-ROM 驱动器, 选择总项

### < > SCSI support

并在“SCSI low-level drivers”中选择自己要用的 SCSI 卡的驱动。可以编译成模块。

## 8. Network device support (网络设备支持)

此类选项提供内核对网络设备 (如网卡) 和其他非以太网连接所使用的网络协议的支持。若下列一项回答“y”, 则在其下的问题中选择系统所使用的网络设备或协议。

### [ ] Network device support

对于以太网连接, 选择:

#### [ ] Ethernet (10 or 100Mbit)

然后选择系统所用网卡, 回答“y”或“m”。如对于 3c509 网卡, 对:

#### [ ] 3COM cards

回答“y”, 然后对其子项:

#### < > 3c509/3c579 support

回答“y”或“m”。对于 NE2000 兼容网卡, 选择:

#### [ ] Other ISA cards

在其子项中选择

#### < > NE2000/NE1000 support

回答“y”或“m”。

对于调制解调器拨号上网, 如果希望使用 PPP 或 SLIP 协议, 选择以下选项:

#### < > PPP (point-to-point) support

#### < > SLIP (serial line) support

## 9. ISDN subsystem (ISDN 子系统)

### < > ISDN support

如果用户需要 ISDN 支持, 回答“y”, 并在其下列出的 ISDN 卡和协议中选择自己需

要的驱动。

#### 10. CD-ROM drivers(not for SCSI or IDE/ATAPI drives)

(CD-ROM 驱动(不含 SCSI 和 IDE/ATAPI 驱动))

##### [ ] Support non-SCSI/IDE/ATAPI CDR0M drives

如果用户的 CD-ROM 驱动器不是 SCSI 或 IDE/ATAPI 接口,回答“y”,并在其下列出的各种驱动中选择对应自己的 CD-ROM 驱动器的一种。

#### 11. Filesystems (文件系统)

这里是有关各种文件系统及其特性的选项。用户可以选择让自己的内核支持何种文件系统。但有些文件系统类型应该总是支持的,如:

##### < > Minix fs support

虽然 Minix 文件系统已经被 ext2 所取代,但它仍被用在 root/boot 盘和 RAM 虚拟盘中。所以用户应在这里回答“y”或“m”,以使系统可以读这些通用的软盘格式。

##### < > Second extended fs support

ext2 是 Linux 的标准文件系统,这里当然应该回答“y”,除非准备把 Linux 作为在一个 DOS 分区中的 umsdos 文件系统而运行。

##### < > DOS FAT fs support

如果用户准备在 Linux 中使用基于 FAT 的文件系统,如 MS-DOS, VFAT(Windows 95)和 UMSDOS 文件系统,这里应该回答“y”。它本身不是一种文件系统,但为内核提供 FAT 支持。在下列两个选项里回答“y”,以使 Linux 可以挂上和使用硬盘上的 DOS 分区和 Windows 95 的分区:

##### < > MSDOS fs support

##### < > VFAT (Windows-95) fs support

##### [ ] /proc filesystem support

/proc 是提供系统运行状态信息的虚拟文件系统,它保证系统的正常运行。每个人都应该回答“y”。

##### < > ISO9660 cdrom filesystem support

这是 CD-ROM 使用的标准文件系统。没有对 ISO9660 文件系统的支持,系统将无法读取光盘上的文件。有光驱的系统都应当回答“y”。

其他一些选项不会影响系统的正常运行,用户可以根据自己的需要来选择。如:

##### [ ] Quota support

系统管理员如果想给用户设置磁盘定额,回答“y”。

##### < > NFS filesystem support

用户如果想用 NFS 协议挂上和使用远程系统的磁盘分区,回答“y”或“m”。

##### < > SMB filesystem support (to mount WfW shares etc..)

如果用户需要系统对 SMB 文件系统的支持,必须选择上面的选项。SMB(Server Message Buffer)是 Windows for Workgroup(WfW), Windows 95, Windows NT 和 Lan Manager 等用来实现在局域网上传文件和打印机共享的协议。用户选择此项后可以用特殊的程序挂卸局域网上的 SMB 文件系统,像使用 NFS 文件系统一样方便地使用 Windows

95/NT 等操作系统的共享目录。参见 7.7 节“Linux 和 Windows 95/NT 的文件和打印机共享”。

#### [ ] SMB Win95 bug work-around

选择此项将修正 Windows 95 的 SMB 服务器的某些缺陷。如果用户想使用 Windows 95 上的共享文件系统,必须选择此项。

如果用户想让自己的内核支持其他操作系统上的文件系统,如 OS/2 HPFS, UFS 或 NCP 等文件系统,可选择对应的选项回答“y”或“m”。

### 12. Character devices (字符设备)

#### < > Standard/generic serial support

是否支持标准的串行口设备? 对于大多数系统,在串行口上会使用鼠标、调制解调器及其他设备,应回答“y”或“m”。

#### < > Parallel printer support

是否支持并行口打印机? 如果准备使用并行口打印机,应回答“y”或“m”。如果选择编译成可加载模块,将产生 lp.o 文件。

在 Character devices (字符设备) 一类问题中,还有其他一些选项,对应一些特定的字符设备,如磁带机、非串口鼠标,如果系统上有这样的设备,可对相应的问题回答“y”或“m”。

### 13. Sound (声音)

#### < > Sound card support

是否支持声卡? 如果机器上有个声卡,要在 Linux 上使用它,回答“y”。注意要了解声卡的各项参数(I/O 端口,中断,DMA 通道)。如果要把声卡驱动编译为可加载模块,回答“m”。如果不准备使用声卡,回答“n”,则下面有关声卡的问题都无须回答了。

如果选择支持声卡,必须在 Linux 支持的声卡或芯片中选择与机器上的声卡相符合的一种或几种(较新的内核版本支持较多种类的声卡)。例如对于 Sound Blaster 系列,如 AWE32,除选择“Sound Blaster (SB, SBPro, SB16, clones) support”外,还可以选择“Generic OPL2/OPL3 FM synthesizer support”。注意选择声卡或芯片后,有些应该在其后的相关问题中配置声卡的各种参数,如对于 AWE32 要求给出 I/O 基址,DMA,16bit DMA,MPU401 I/O 基址等参数。

### 14. Kernel hacking (内核监视)

#### [ ] Kernel profiling support

是否对内核行为作记录? 如回答“y”,内核中各种过程所耗时间的信息将被记录在文件/proc/profile 中,这个文件必须用 readprofile 包(可从 sunsite.unc.edu 取到)来读。一般用户回答“n”。

### 15. 注意事项

内核的配置是编译内核过程中极其重要的一步,关系到编译的成败。要适当选择编

译哪些模块和驱动。一般来说,编译进内核的模块和驱动越多,内核支持的特性和硬件就越多,但内核的体积就越大,系统运行时所占的内存也越大。而且支持某些功能会使内核的速度变慢。有些处于测试阶段的模块或驱动有时可能还会对系统的正常运行造成影响。所以,要谨慎地配置内核,一般只选择那些必需的模块或驱动。

配置内核时要即时查看帮助文件和 Linux/Documentation/目录下有关的文档。在对某个配置不太明确时,尽量使用帮助文件中的推荐选择。

机器上的某些插卡对应的驱动建议编译为可加载模块,如网卡和声卡等,这样可减少内核的体积和所占的内存。某些不常用的驱动可在必要的时候再用 `insmod`, `modprobe` 或 `kernel` 加载进系统。

对于测试版的内核,如 2.1.xx 版等,其中可能有些代码并未完成或未与整体达成统一。如果在配置后的编译过程中,发现在某个模块中出错,建议重新配置,不要选择对应模块。

### 5.15.5 编译内核和用新内核引导

内核配置完成后,在 Linux 目录下执行“`make dep`”,以正确地建立所有的从属文件。从属文件中包含着每个 .c 文件中引用的头文件(.h)的全路径名。

下面,用“`make zImage`”编译压缩的内核映像文件。如果希望做一张不含根文件系统和 LILO 的引导盘,插一张软盘到 A: 驱,做“`make zdisk`”。“`zdisk`”对于测试新内核很方便,如果它完全不能工作,只要把软盘拿掉,再用旧的内核启动即可。

编译一个内核大约需要多长时间?这主要取决于机器 CPU 的速度以及系统的内存数量。

如果一切顺利,“`make zImage`”将会产生一个压缩的内核映像文件 `/usr/src/Linux/arch/i386/boot/zImage`。如果“`make zImage`”产生的内核太大,编译过程的最后将提示用“`make bzImage`”做一个更小的内核。

如果用户已经配置内核的任何部分为可加载模块,这时必须做“`make modules`”和“`make modules_install`”,以编译和安装可加载模块。可加载模块将被安装到类似 `/lib/modules/2.2.6/` 的目录下。参考 `Documentation/modules.txt` 以获得关于怎样使用可加载模块的更详细的信息。

现在必须备份老的内核,以便在新的内核不能正常运行时能恢复老的内核。如把老内核 `/vmlinuz` 备份为 `/vmlinuz.old`。备份后,把新的内核 `/usr/src/Linux/arch/i386/boot/zImage` 拷贝到系统放置老的内核的地方,以覆盖老的内核,如拷贝为 `/vmlinuz`。然后,编辑 `/etc/LILO.conf` 文件,在后面加进用老内核引导的一段,如:

```
# For old kernel
image = /vmlinuz.old      # 老内核的文件名
root = /dev/hda1          # 根文件系统
label = old               # 引导老内核所用标记
read-only
```

而原来的一段,如

```
image = /vmlinuz
```

```
root = /dev/hda1
label = Linux
read-only
```

已经变为引导新的内核了。然后重新安装 lilo, 通常是运行命令 `/sbin/LILO`。注意, 每次安装新的内核后都必须重新运行 LILO, 即使并没有改变 `/etc/LILO.conf` 文件。

现在, 新的内核已安装完毕, 可以关机, 试着用新内核引导了。如果新内核无法引导或引导后无法正常和稳定运行, 重新用老的内核引导(在上面的例子中, 在 `lilo boot:` 后打入 `old`)后, 检查新内核的配置, 尽量减少不必要的模块和驱动, 重新编译后再作尝试。当新的内核能正常运行后, 再逐渐恢复预期的内核配置。

# 6

## Linux 编程简介

### 6.1 GNU C 和 GNU Fortran 的安装和使用

本节将介绍如何在 Linux 系统上安装 GNU C 和 Fortran 编译器及软件开发环境。实际上,在各种 Linux 的发行套件里都包含预编译的软件。但如果用户想对 GNU C 和 Fortran 的版本进行升级,或想多了解一些有关 Linux 上编程环境的知识,本节将是有益的参考。

本节还将对在 Linux 下如何编译、连接 C 和 Fortran 程序及程序的运行和调试作一概括的介绍。

#### 6.1.1 安装 gcc

GNU C 的编译器叫 gcc。gcc 软件包支持 C, C++, Objective C 语言;并可加进对 Fortran 语言的支持。

从任一存放 GNU 软件的 FTP 站点取来 gcc-2.7.2.3.tar.gz。如果需要对 Fortran 语言的支持,还必须取来 g77-0.5.21.tar.gz。

到某一工作目录,打开 gcc-2.7.2.3.tar.gz:

```
% tar zxvf gcc-2.7.2.3.tar.gz,
```

将会产生一个目录 gcc-2.7.2.3/。

如果需要对 Fortran 语言的支持,在同一工作目录下打开 g77-0.5.21.tar.gz,把 g77 软件包并入 gcc 软件包,并用 g77 软件包中的“补丁”文件来修改 gcc 软件包中的一些文件:

```
% tar zxvf g77-0.5.21.tar.gz
% ln -s gcc-2.7.2.3 gcc
% ln -s g77-0.5.21 g77
% mv -i g77/* gcc
% patch -p1 -V t -d gcc < gcc/t/gbe/2.7.2.3.diff
% cd gcc
% touch f77-install-ok
% touch f2c-install-ok
```

现在,在 gcc-2.7.2.3/目录下,执行如下命令编译和安装 gcc:

```
% ./configure
% make bootstrap
% make compare
```

```
# make -k install
```

最后一步如果是安装到系统目录,应在 root 账号里完成。注意,gcc 的标准安装目录是 /usr/local 下的各子目录,如 /usr/local/bin, /usr/local/lib, /usr/local/include 等。如果用户希望将 gcc 安装到别的目录下,应在 configure 一步中指定 --prefix,如:

```
% ./configure --prefix = /usr
```

表明将 gcc 安装到 /usr/bin, /usr/lib, /usr/include 等目录里。

gcc + g77 软件包将在系统上安装如下文件(假设安装目录是 /usr/local):

```
/usr/local/bin:
    gcc, i586-unknown-linux-gnulibc1-gcc, unprotoize, protoize,
    c + +, g + +, f77, g77

/usr/local/lib:
    libf2c.a,
    gcc-lib/i586-unknown-linux-gnulibc1/2.7.2.3.f.1/
        SYSCALLS.c.X, ce1, ce1obj, ce1plus, cpp,
        crtbegin.o, crtbeginS.o, crtend.o, crtendS.o,
        f771, libf2c.a, libgcc.a, libobjc.a, specs,
    include/
        * .h.
    objc/
        * .h

/usr/local/info:
    ccp.info *, g77.info *, fcc.info *

/usr/local/man/man1:
    cccp.1, f77.1, g + +.1, g77.1, gcc.1
```

安装完毕后,用户可以用“gcc -v”和“g77 -v”来检验 gcc 和 g77 是否正确安装。

现在用户就可以用 gcc 和 g77 来编译 C 和 Fortran 程序了。但是,如果用户想使用 C + + (G + +) 来编译 C + + 程序,还必须编译安装 GNU 的 C + + 类库。

从 GNU 软件的 FTP 站点取来 libg + + -2.7.2.tar.gz,依照下面的步骤编译和安装 GNU 的 C + + 类库:

```
% tar zxvf libg + + -2.7.2.tar.gz
% cd libg + + -2.7.2
% ./configure
% make all
# make install
```

**注意:**如果在编译 gcc 时,configure 一步使用了 --prefix 参数,则在编译 libg + + 时也要使用相同的 --prefix 参数。最后一步如果是安装到系统目录,应在 root 账号里完成。

libg + + 软件包将在系统上安装如下文件(假设安装目录是 /usr/local):

```
/usr/local/bin:
    gperf, genclass

/usr/local/lib:
    libiberty.a, libg + + .a, libstdc + + .a,
```

```

g + + -include/
    头文件,
    gen/
        头文件,
    std/
        头文件
/usr/local/i586-unknown-Linux/include:
    _G_config.h
/usr/local/man/man1:
    gperf.1, configure.1

```

有的用户希望将 libg + + 和 libstdc + + 编译成共享库。用如下的 configure 命令:

```
% ./configure --enable-shared
```

再编译安装,这将在 /usr/local 目录下安装如下文件:

```

/usr/local/lib:
    libg + + .so.27.2.0,
    libg + + .so.27 -> libg + + .so.27.2.0,
    libg + + .so -> libg + + .so.27.2.0,
    libstdc + + .so.27.2.0,
    libstdc + + .so.27 -> libstdc + + .so.27.2.0,
    libstdc + + .so -> libstdc + + .so.27.2.0
/usr/local/i586-unknown-Linux/include:
    assert.h

```

### 6.1.2 使用 gcc 编译和连接

用 gcc 编译 C 程序生成可执行文件,有时候看起来像是一步就完成了,但实际上它要经历如下的四个步骤:

预处理(Preprocessing)

这一步需要分析各种命令,如 # define, # include, # if 等。gcc 调用 cpp 程序来进行预处理工作。

编译(Compilation)

这一阶段根据输入文件产生汇编语言,由于通常是立即调用汇编程序,所以其输出一般不保存在文件中。gcc 调用 ccl 进行编译工作。

汇编(Assembly)

这一步将汇编语言用作输入,产生有 .o 扩展名的目标文件。gcc 调用 as 进行汇编工作。

连接(Linking)

这一阶段中,各目标文件 .o 被放在可执行文件的适当位置上,该程序引用的函数也放在可执行文件中(对使用共享库的程序稍有不同)。gcc 调用连接程序 ld 来完成最终的任务。

gcc 的基本用法是:

```
% gcc -o prog main.c subr1.c subr2.c subr3.c
```

“-o prog”指定输出的可执行文件名为 prog。如果设有指定-o 参数, gcc 就使用缺省的可执行文件名 a.out。如果想单独编译每一个源文件, 最后再进行连接, 进行如下

```
% gcc -c main.c
% gcc -c subrl.c
% gcc -c subrl2.c
% gcc -c subrl3.c
% gcc -o prog main.o subrl.o subrl2.o subrl3.o
```

其中的-c 选项表示编译产生目标文件, 但不连接。最后将所有目标文件组合在一起, 构成可执行文件。由于最后一个命令的输入都是目标文件, 不需要编译和汇编, 所以 gcc 就只调用连接程序。

gcc 如何处理出现在命令行的文件取决于文件的名字, 确切地说, 取决于文件的扩展名。在每一种情况下, gcc 都要将该文件送到相应的程序去进行预处理, 编译或汇编, 然后将产生的所有目标模块连接在一起, 产生一个可执行文件。下面是 gcc 如何处理不同类型文件的列表

|           |                                   |
|-----------|-----------------------------------|
| file.c:   | C 源文件, 被 gcc 预处理和编译               |
| file.C:   | C++ 源文件, 被 gcc 预处理和编译             |
| file.cc:  | C++ 源文件, 被 gcc 预处理和编译             |
| file.cxx: | C++ 源文件, 被 gcc 预处理和编译             |
| file.m:   | Objective C 源文件, 被 gcc 预处理, 编译和汇编 |
| file.i:   | 预处理后的 C 源文件, 被 gcc 编译             |
| file.ii:  | 预处理后的 C++ 源文件, 被 gcc 编译           |
| file.s:   | 汇编语言源文件, 被 as 汇编                  |
| file.S:   | 汇编语言源文件, 被 as 预处理和汇编              |
| file.o:   | 编译后的目标文件, 传送给 ld                  |
| file.a:   | 目标文件库, 传送给 ld                     |

gcc 在命令行上经常使用的几个选项是:

-c

只预处理, 编译和汇编源程序, 不进行连接。编译器对每一个源程序产生一个目标文件。

-o file

确定输出文件为 file。如果没有用-o 选项, 缺省的可执行文件的输出是 a.out, 目标文件和汇编文件的输出对 source.suffix 分别是 source.o 和 source.s, 预处理的 C 源程序的输出是标准输出 stdout。

-Dmacro

-Dmacro = defn

其作用类似于源程序里的 #define。例如

```
% gcc -c -DHAVE_GDBM -DHELP_FILE = \"help\" edict.c
```

其中第一个-D 选项定义宏 HAVE\_GDBM, 在程序里可以用 #ifdef 去检查它是否被设置。第二个-D 选项将宏 HELP\_FILE 定义为字符串“help”(由于反斜线的作用, 引号实际上已成为该宏定义的一部分), 这对于控制程序打开哪个文件是很有用的。

#### -Umacro

某些宏是被编译程序自动定义的。这些宏通常可以指定在其中进行编译的计算机系统类型的符号,用户可以在编译某程序时加上-v 选项以查看 gcc 缺省定义了哪些宏。如果用户想取消其中某个宏定义,用-Umacro 选项,这相当于把 # undef macro 放在要编译的源文件的开头。

#### -ldir

将 dir 目录加到搜寻头文件的目录列表中去,并优先于在 gcc 缺省的搜索目录。在多个-I 选项的情况下,按命令行上-I 选项的前后顺序搜索。dir 可使用相对路径。如-I. ./inc 等。

#### -O

对程序编译进行优化,编译程序试图减少被编译程序的长度和执行时间,但其编译速度比不作优化时慢,而且要求较多的内存。

#### -O2

允许比-O 更好的优化,编译速度较慢,但结果程序的执行较快。

#### -g

产生一张用于调试和排错的扩展符号表。-g 选项使程序可以用 GNU 的调试程序 gdb 进行调试。优化和调试通常不兼容的,同时使用-g 和-O(-O2)选项经常会使程序产生奇怪的运行结果。所以不要同时使用-g 和-O(-O2)选项。

#### -fPIC

产生位置无关的目标代码,可用于构造共享函数库。

以上是 gcc 的编译选项。gcc 的命令行上还可以使用连接选项。事实上,gcc 将所有不能识别的选项传递给连接程序 ld。连接程序 ld 将几个目标文件和库程序组合成一个可执行文件,它要解决对外部变量、外部过程、库程序等的引用。但我们永远不必要显式地调用 ld。利用 gcc 命令去连接各个文件是很简单的,即使在命令行里没有列出库程序,gcc 也能保证某些库程序以正确的次序出现。

gcc 的常用连接选项有下列几个

#### -Ldir

将 dir 目录加到搜寻-l 选项指定的函数库文件的目录列表中去,并优先于 gcc 缺省的搜索目录。在有多-l 选项的情况下,按命令行上-l 选项的前后顺序搜索。dir 可使用相对路径。如-L. ./lib 等。

#### -lname

在连接时使用函数库 libname.a,连接程序在-Ldir 选项指定的目录下和/lib, /usr/lib 目录下寻找该库文件。在没有使用-static 选项时,如果发现共享函数库 libname.so,则使用 libname.so 进行动态连接。

#### -static

禁止与共享函数库连接。

#### -shared

尽量与共享函数库连接。这是 Linux 上连接程序的缺省选项。

下面是一个使用 gcc 进行连接的例子

```
% gcc -o prog main.o subr.o -L.../lib -lany -lm
```

### 6.1.3 创建函数库

有时候,用户会感到有必要创建自己的函数库。`ar` 命令可用于创建和更新函数库。

创建一个新的函数库,可用如下命令

```
% ar rs lib-name list-of-files
```

`r` 选项表示 `ar` 命令应把在列表 `list-of-files` 中的目标文件增加到 `lib-name` 库中,如果 `lib-name` 库不存在,就创建一个。`s` 选项表示为该档案库产生一个索引(从而不必要执行 `ranlib` 命令了)。

要更新一个库,用

```
% ar rus lib-name list-of-files
```

该命令将 `list-of-files` 列出的任何文件的日期与库中该文件的版本进行比较,如果 `list-of-files` 中的文件比库中的版本更新,`ar` 就用新版本替代老版本。`s` 选项可更新此函数库的索引。

从函数库中删除一个或多个文件,可使用如下命令

```
% ar ds lib-name list-of-files
```

该命令可删除 `list-of-files` 列出的并在 `lib-name` 库中的所有文件。

从一个函数库中提取一个或多个文件,可用

```
% ar x lib-name list-of-files
```

该命令并不修改函数库文件本身,而是从库中提取 `list-of-files` 列出的所有文件。通常,被提取文件的时间被标记成当前时间。但如果用 `xo` 选项代替 `x` 选项,则被提取文件的时间被标记成它们进入档案库的时间。

### 6.1.4 利用 make 和 Makefile 实现自动编译

UNIX 系统上的很多软件包都是使用 `make` 程序和 `Makefile` 文件来实现自动编译的。`make` 程序的目的是自动确定一个软件包的哪些部分需要重新编译,并用特定的命令去编译它们。准确使用 `make` 可以大大减少编译程序所花费的时间,因为它可以消除不必要的再编译。

为使用 `make`,我们必须编写一个叫 `makefile` 的文件,它描述了软件包中各个文件之间的关系,提供了更新每个文件的命令。在一个软件包里,通常是可执行文件由连接目标文件而更新,而目标文件由编译源文件而更新。

当一个适当的 `makefile` 存在时,每次我们改变某些源文件,简单的 `shell` 命令:

```
% make
```

将足以完成所有必需的重新编译。`make` 程序利用 `makefile` 的数据和每个文件最新一次更改的时间来确定哪些文件需要更新;对每个需要更新的文件,`make` 程序使用 `makefile` 中定义的命令来更新它。

如果 `make` 程序没有使用 `-f` 选项指定一个 `makefile`,`make` 将在当前目录下按顺序寻找下列文件: `GNUmakefile`, `makefile`, `Makefile`。我们推荐使用 `Makefile`。因为它的第一个字母是大写,通常被列在一个目录的文件列表的前面。

要想使用 make,就必须首先创建 Makefile 文件。下面说明编写 Makefile 文件的规则。

一个 Makefile 中包含着一些目标(target),对每一个目标,提供了与这个目标有相关性(dependency)的其他目标或文件的名字及实现这个目标的一组命令。所谓目标,就是 make 程序要完成的一项任务,目标常常是文件名,也可以不是;所谓相关性,即某目标的完成依赖于其他一些目标或文件。下面举一个简单的 Makefile 的例子说明这个问题:

```
# 一个简单的 Makefile 例子
# 以 # 开头的为注释行
prog:    prog.o subr.o
        gcc -o prog prog.o subr.o
prog.o:  prog.c prog.h
        gcc -c -I. -o prog.o prog.c
subr.o:  subr.c
        gcc -c -o subr.o subr.c
clean:
        rm -f prog *.o
```

上面的 Makefile 中定义了四个目标:prog,prog.o,subr.o 和 clean。目标从每行的最左边开始写,后面跟一个冒号(:),如果有与这个目标有相关性的其他目标或文件,把它们列在冒号后面,并以空格隔开。然后另起一行开始写实现这个目标的一组 shell 命令。shell 命令可以有若干行。注意,每条 shell 命令行的第一个字符必须是 <tab> 键。它不能用一个或一串空格开头,否则 make 就会显示一条错误信息指出 Makefile 中写错的那一行,并退出,如

```
Makefile:2: *** missing separator. Stop.
```

一般情况下,调用 make 命令可输入:

```
% make target,
```

target 是 Makefile 文件中定义的目标中的一个,如果省略 target,make 就将生成 Makefile 文件中定义的第一个目标。对于上面 Makefile 的例子,单独的一个“make”命令等价于

```
% make prog,
```

因为 prog 是 Makefile 中定义的第一个目标。

make 为它执行的每一条命令行生成一个新的 shell。其结果是,被该 shell 执行的命令只在单一的命令行里有效。特别是 cd 命令,它只能影响它所在的命令行。如在下列命令行中:

```
cd ../lib
gcc -c -o subr.o subr.c
```

对第二行,第一行的 cd 命令是无效的。要想在编译 subr.c 之前进入 ../lib 目录,可使用如下命令:

```
cd ../lib; gcc -c -o subr.o subr.c
```

在 Makefile 中,可使用续行号(\)将一个单独的命令行延续成几行。但要注意在续行号(\)后面不能跟任何字符(包括空格和 <tab> 键)。

make 在检查一个目标是否已经过并需要更新时,采用的是按相关性递归的方法。

make 在构建一个目标之前要生成该目标依赖的所有文件,并递归地前进,从而确保这些文件是最新的。make 采取如下步骤去生成一个目标:

(1)如果一个目标 task 不是作为一个文件而存在,它就是过时了。命令 make task 总是执行该任务。

(2)make 检查所有与 task 目标有相关性的目标。对于不是 Makefile 中定义的任务,而只是文件的相关目标,则检查文件是否比 task 文件更新,文件中有一个更新则 task 就过时了。对于是 Makefile 中定义的任务的相关目标,按同样的方法递归检查其是否过时,如果其中有任何一个过时了,task 也就过时了。

(3)从递归的底层向上,对所有已过时的目标进行更新;只有当一个目标所依赖的所有目标都已是最新的时,这个目标才可以进行更新。

我们通过上面 Makefile 的例子来看目标更新的过程。现在假设我们修改了文件 subr.o。我们用如下命令更新目标 prog,即重新编译可执行文件 prog:

```
% make prog
```

由于目标 prog 依赖于目标 prog.o 和 subr.o。我们必须检查目标 prog.o 和 subr.o 是否过时。目标 prog.o 依赖于文件 prog.c 和 prog.h,我们检查目标文件 prog.o 和源文件 prog.c 和 prog.h 的日期,发现 prog.o 比它所依赖的源文件要新,即并不过时。再检查目标 subr.o,它依赖于文件 subr.c。由于我们编辑了 subr.c,它的日期比目标文件 subr.o 的日期要新,即 subr.o 过时了。从而依赖于 subr.o 的所有目标都过时了。我们用定义目标 subr.o 的 shell 命令来更新 subr.o:

```
gcc -c -o subr.o subr.c
```

由于目标 subr.o 过时并更新,导致目标 prog 已经过时,我们必须用定义 prog 的一组 shell 命令来更新它:

```
gcc -o prog prog.o subr.o
```

从而完成了“make prog”的任务。

如果我们是第一次编译上面这个软件,则因为 prog,prog.o,subr.o 等目标文件都不存在,按照约定,所有目标 prog,prog.o,subr.o 都是过时的,都必须更新,即必须从底向上执行定义这些目标的所有命令。

在上面 Makefile 的例子中,还定义了一个目标 clean,我们输入 make clean 的命令将执行:

```
rm -f prog *.o
```

clean 目标是 Makefile 中常用的一种专用目标,即删除所有的目标模块。输入 make clean 命令时,make 就查看一个名为 clean 的文件,如果该文件不存在(我们约定永远不在软件目录中使用具有这一名字的文件),make 就执行定义该目标的所有命令。

另一个经常用到的专用目标是 install。它通常是将编译完成的可执行文件和程序运行所需的其他文件拷贝到指定的安装目录中,并设置相应的保护。

为了简化命令的书写,在 Makefile 中可以定义一些宏(macro)和使用几个预先定义的缩写。下面是两个很有用的缩写:

\$@ 代表该目标的全名

\$\* 代表已经删除了后缀的目标名

\$ < 代表该目标的第一个相关目标名

按照这样的缩写,前面 Makefile 的例子可改写成:

```
# 一个简单的 Makefile 的例子
# 以 # 开头的为注释行

prog:      prog.o subr.o
           gcc -o $@ prog.o subr.o

prog.o:    prog.c prog.h
           gcc -c -I. -o $@ $<

subr.o:    subr.c
           gcc -c -o $@ $* .c

clean:
           rm -f prog *.o
```

这类缩写对编写缺省的编译规则时很有用。

一个宏定义从一行的最左边开始书写,具有如下格式:

```
macro-name = macro-body
```

当 make 在处理该 Makefile 时,就用 macro-body 替代 \$(macro-name) 串。如上面的 Makefile 的例子可用定义宏的方法使之更清晰:

```
# 一个简单的 Makefile 的例子
# 以 # 开头的为注释行

DEPENDS   = prog.o subr.o

prog:      $(DEPENDS)
           gcc -o $@ $(DEPENDS)

prog.o:    prog.c prog.h
           gcc -c -I. -o $@ $* .c

subr.o:    subr.c
           gcc -c -o $@ $* .c

clean:
           rm -f prog *.o
```

在定义一组编译选项时,宏定义也是很有用的。如宏定义:

```
CFLAGS = -DDEBUG -g
```

中包含了两个编译选项。我们可以在 Makefile 中用下列命令编译用于调试的目标文件:

```
gcc -c $(CFLAGS) -o list.o list.c
```

如果在 Makefile 文件中没有给出从某一目标的相关目标构造这一目标的命令,make 程序将适用隐含规则。

make 程序预定义了一些隐含(implicit)规则,每个隐含规则适用于一个目标类型和它的相关类型的组合。对应同一目标类型的相关类型可能不止一个,例如,有多个规则可产生目标类型为 .o 的文件,如从 .c 文件用 C 编译器编译得到,或从 .f 文件用 Fortran 编译器得到。那么,如何确定究竟应该适用哪条隐含规则?

隐含规则是通过后缀(suffix)规则来实现的。带有哪些后缀的目标适用隐含规则是由 make 程序的内置目标 .SUFFIXES 的相关目标列表定义的。缺省的后缀列表是:  
.out, .a, .ln, .o, .c, .cc, .C, .p, .f, .F, .r, .y, .l, .s, .S, .mod, .sym, .def, .h, .info, .dvi, .

tex, .texinfo, .texi, .txinfo, .w, .ch, .web, .sh, .etc, .el。在所有隐含规则中允许的目标或相关类型必须是上列类型中的一种。并且这个列表中,位置靠前的类型具有更高的优先级。即如果有数条隐含规则的目标类型一样,make 程序按上面列表的顺序适用第一个其相关类型存在或可以构造的隐含规则。

例如,对下面的 Makefile:

```
# 一个简单的 Makefile 的例子
# 以 # 开头的为注释行
prog:      prog.o subr.o
           gcc -o prog prog.o subr.o

clean:
           rm -f prog *.o
```

prog 的相关目标 prog.o(和 subr.o)的构造规则没有定义,make 程序适用隐含规则。缺省的隐含规则中可生成目标类型为 .o 文件的相关类型有很多种:.c,.cc,.C,.cp,.f,...。make 程序将按顺序找出第一个存在的或可以构造的相关类型,如果它找到了 prog.c 文件,它就适用从 .c 文件生成 .o 文件的隐含规则。如果 prog.c,prog.cc,prog.C 和 prog.p 文件都没有找到或不可构造,但找到了 prog.f,它就适用从 .f 文件生成 .o 文件的隐含规则,依此类推。

make 程序的一条隐含规则可以用如下的语句表示:

```
.c.o:
$(CC) -c $(CPPFLAGS) $(CFLAGS) $<
```

这是一条目标类,型是 .o 而相关类型是 .c 的隐含规则。它确定了如何从 .c 文件生成 .o 文件。其中用到的宏 CC, CPPFLAGS, CFLAGS 在 make 程序里都有缺省的定义。关于 make 程序定义的所有隐含规则的信息请参考 GNU 的 make 软件包中携带的文档。

如果我们不想使用 make 程序定义的缺省规则,我们可以定义自己的后缀规则,如:

```
.c.o:
gcc -c -g -DDEBUG -DHELP_FILE = \ "help \" -o $* .o $<
```

我们也可以 .SUFFIXES: 清除缺省后缀列表或往后缀列表表中增加新的相关类型。如:

```
# 首先清除后缀列表,所有的后缀规则(包括缺省规则)都失效
.SUFFIXES:
# 增加 .c, .o 到后缀列表里,以后所有的后缀规则可涉及 .c 和 .o 文件
.SUFFIXES: .c .o

.c.o:
gcc -c -g -DDEBUG -DHELP_FILE = \ "help \" -o $* .o $<
```

后缀规则适用于所有 UNIX 版本的 make 程序,GNU 的 make 程序提供了一种更方便的规则定义方式,称为模式(pattern)规则。模式规则的格式类似 Makefile 中一般的目标定义,但它使用百分号(%)作为通配符,如:

```
% .o: % .c
gcc $(CFLAGS) -c $<
```

表示从 .c 文件生成 .o 文件的规则。

模式规则不依赖于后缀列表。它与后缀规则的另一个不同之处可用下面的例子说

明。后缀规则：

```
.c.o:
    $(CC) -c $(CFLAGS) $(CPPFLAGS) -o $@ $<
```

说明了从 .c 文件产生 .o 文件的规则。但后缀规则不能有自己的相关目标。如下面的规则：

```
.c.o: foo.h
    $(CC) -c $(CFLAGS) $(CPPFLAGS) -o $@ $<
```

要求从相关文件 foo.h 编译产生文件“.c.o”，而完全不同于后缀规则的定义。而模式规则却可以有自己的相关目标，如：

```
%o: %.c foo.h
    $(CC) -c $(CFLAGS) $(CPPFLAGS) -o $@ $<
```

定义了从 .c 文件生成 .o 文件的规则，而且使所有用这条规则生成的 .o 文件都与 foo.h 相关。

## 6.2 C Shell 编程

C Shell 是由加利福尼亚大学伯克利分校的 Bill Joy 开发的一个命令解释器，它使用了类似 C 语言的语法。它提供了在 Bourne Shell 中不存在的很多更方便交互使用的特征，如文件名补全、命令别名、历史命令替换、作业控制和一些内置命令。当然，像 Bourne Shell 一样，它也支持变量、命令和文件名替换。启动 C Shell 的命令是 csh，通常在 /bin 或 /usr/bin 下。

tcsh 是 C Shell 的增强，但又与 C Shell 完全兼容。tcsh 包括了命令行编辑器、可编程单词补全、拼写校正、历史命令替换机制、作业控制和类似 C 语言的语法。

Linux 上使用的是 /bin/tcsh，/bin/csh 只是指向 /bin/tcsh 的一个符号连接。

本节并不是要详细介绍 csh 和 tcsh 的用法，而是就如何编写 C Shell 脚本文件作一个简单的说明。

C Shell 脚本文件包含一系列 shell 命令，C Shell 把它们作为一个整体来执行。用户主目录下的 .cshrc，.login，.logout 等就是 C Shell 脚本文件。

### 6.2.1 运行脚本文件

一个 C Shell 脚本文件可以用如下命令来运行：

```
% csh script_file arg_1 arg_2 ...
```

其中 script\_file 是要执行的脚本文件的名称。arg\_1, arg\_2 ... 是脚本文件运行时可能要求的一些输入参量。C Shell 把这些参量放在 shell 变量组 argv 中，分别作为 argv[1], argv[2], 等等。没有 argv[0]，C Shell 用 \$0 代替 argv[0]。在这个例子中，\$0 等于 script\_file。C Shell 在读完输入参量后，开始顺序执行 script\_file 文件中的命令。

如果用户想直接运行脚本文件，不想在命令行使用 csh 命令，编辑脚本文件第一行是：

```
#!/bin/csh
```

(“#”号在脚本文件中也用以表示注释行)然后用 `chmod` 命令使脚本文件成为可执行文件:

```
% chmod 755 script_file
```

现在用户就可以在命令行直接敲脚本文件名:

```
% script_file
```

C Shell 就会自动执行脚本文件 `script_file` 了。如果脚本文件的第一行不是 `#!/bin/csh`, 上一行命令的结果是 Bourne Shell (通常是 `/bin/sh`) 来执行这个脚本文件, 而不是 C Shell!

### 6.2.2 脚本的执行

C Shell 对每一个 shell 脚本本行进行分析: 鉴别命令和进行变量代换。每一个以“\$”开头的变量被替换为它们的值。例如脚本行:

```
echo $sum1
```

打印变量 `sum1` 的当前值到 shell 脚本的标准输出文件。如果 `sum1` 没有定义则会出错。

为鉴别某变量当前是否已指定了一个值, 用下列记号:

```
$? sum1
```

如果变量 `sum1` 当前已指定了一个值, 问号(?)使上述表达式返回值 1, 否则返回值 0。这是访问一个未定义的变量而不产生错误的唯一方法。

为确定某变量有多少个(以空格隔开的)组成部分, 用记号:

```
$ #sum1
```

“#”号指定变量的组成部分数。如:

```
set sum1 = (a b c)
echo $? sum1
1
echo $ #sum1
3
unset sum1
echo $? sum1
0
echo $ #sum1
sum1: Undefined variable.
```

我们可以访问每个变量的各个组成部分的值:

```
echo $sum1[1]
```

将返回变量 `sum1` 的第一个部分, 在上例中为“a”。类似地,

```
echo $sum1[ $ #sum1]
```

将返回变量 `sum1` 的最后一个变量, “c”。而

```
echo $sum1[1-2]
```

将返回变量 `sum1` 的头两个部分, 即“a b”。

在 shell 脚本中, 其他常用到的记号有:

```
$n
```

(这里 `n` 是一个数), 它是:

```
$argv[n]
```

的缩写,返回 `argv` 变量的第 `n` 个部分。如 `$2` 等价于 `$argv2`,等等。而:

```
$*
```

等价于 `$argv`。注意 `$n` 和 `$argv[n]` 有个微小的区别。如果 `n` 不是 1 和 `#argv` 之间的某个值, `$argv[n]` 将产生错误,而 `$n` 永远不会产生下标越界错误。

为避免这种错误的发生,可使用 `n`-形式的下标范围,当指定变量的组成成分数小于 `n` 时,会返回一个空值。如上例中 `$sum1[5-]` 将返回空值,而 `$sum1[3-]` 则同 `$sum1[3]`。`m-n` 式的下标范围在 `m` 大于变量的组成成分数,而 `n` 在变量的下标范围内时,也将返回一个空值,不会产生错误。

下列记号:

```
$$
```

返回当前 shell 的进程号。由于每个进程都是唯一的,所以进程号可用来产生唯一的临时文件名。

下列记号:

```
<
```

将被从 shell 的标准输入读取的下一行输入取代,而不是使用正在处理的脚本的下一行。这在编写交互式的 shell 脚本是很有用。如:

```
echo 'Yes or No?'
set a=( $< )
```

将打印“`Yes or No?`”到 shell 的标准输出设备,并从 shell 的标准输入设备读取回答放入变量 `a`。

**注意:**在 `echo` 语句中我们必须使用单引号,否则 `No?` 将返回当前目录下以 `No` 开头的三个字母的文件名。如在当前目录下有名为 `now` 的文件时,下列命令:

```
echo Yes or No?
的结果是:
Yes or now
```

### 6.2.3 C Shell 脚本表达式

C Shell 脚本中经常含有利用某些变量的当前值的表达式。事实上,C 语言中大多数的算符都可在 C Shell 脚本中以相同的格式和优先级运用。这些算符包括:

```
|| && | ^ & = = ! = = ~ ! ~ < = > =
< > << >> + - * / % ! { }
```

这里优先级从左往右增高。‘=’、‘!’、‘=’、‘~’和‘!~’,‘<’、‘>’、‘<’和‘>’,‘<<’和‘>>’,‘+’和‘-’,‘\*’、‘/’和‘%’,每组间具有相同的优先级。‘=’、‘!’、‘=’、‘~’和‘!~’

将其左右的变量看作字符串,其他则将其左右的变量看作数字。‘=’、‘~’和‘!~’类似‘=’和‘!’、不同的是‘=’、‘~’和‘!~’的右边可以有通配符(如 `*`、`?` 或 `[ ]`)。

表达式的各部分间以空格隔开。

表达式里空的或缺少的变量被认为是‘0’。所有表达式的结果是字符串,可表示十进制数。

shell 还允许如下格式的文件查询:

- r filename 如用户可读 filename, 返回 true 或 1, 否则返回 false 或 0
- w filename 如用户可写 filename, 返回 true 或 1, 否则返回 false 或 0
- x filename 如用户可执行 filename, 返回 true 或 1, 否则返回 false 或 0
- e filename 如 filename 存在, 返回 true 或 1, 否则返回 false 或 0
- o filename 如用户是 filename 的属主, 返回 true 或 1, 否则返回 false 或 0
- z filename 如 filename 长度为零(空文件), 返回 true 或 1, 否则返回 false 或 0
- f filename 如 filename 是普通文件, 返回 true 或 1, 否则返回 false 或 0
- d filename 如 filename 是目录, 返回 true 或 1, 否则返回 false 或 0

用户可以把一条命令放在一对花括号{ }里以判断命令的执行是否正常结束:

```
} command }
```

**注意:**在 command 两边要有空格。如果命令以退出值 0 正常结束, 上述记号将返回 1; 如果命令以非 0 的退出值非正常结束, 上述记号将返回 0。如果用户需要某命令具体的退出状态, 可以在这条命令执行后紧接着在下一条命令中检查系统变量 \$status。但注意, 每条命令都重新设置 \$status 变量。

例如下面的 shell 脚本利用了正常退出条件:

```
# ! /bin/csh
if { date } then          打印日期
    echo OK               打印“OK”
endif
```

#### 6.2.4 C Shell 脚本控制结构

C Shell 允许的控制结构与 C 语言非常相似。

注解( # )

在每个注释行或不想执行的命令行前加上“#”号表示此行是注解。

foreach 命令

这个语句的语法是:

```
foreach index_variable ( loop_count_value_list )
    command_1
    command_2
    .
    .
end
```

对于 loop\_count\_value\_list 中的每一个值, 将执行“foreach”行和它对应的“end”行之间的所有命令, 变量 index\_variable 依序设为 loop\_count\_value\_list 的每一个值。

在这个循环里, 可用 break 命令停止并跳出循环执行, 用 continue 命令中止对循环变量 index\_variable 的当前值的循环, 而开始对 loop\_count\_value\_list 中下一个值的循环。在 foreach 循环执行完毕后, 循环变量 index\_variable 保持它的最后一个值。

例:

```
foreach ddd ( *.txt )
    if ( $ddd = ccc.txt ) continue
```

```
echo $ddd
rm -f tempfile
sed -e s/"Wang X.M."/"/Wang Xiaomei"/g $ddd > tempfile
mv tempfile $ddd
end
```

if-then-endif 命令

这个语句的语法是：

```
if ( expression ) then
    command_1
    command_2
    .
    .
    .
endif
```

我们可以用关键词 else 来嵌套这些语句如下：

```
if ( expression ) then
    command_1
    command_2
    .
    .
    .
else if ( expression ) then
    command_A
    command_B
    .
    .
    .
else
    command_X
    command_Y
    .
    .
    .
endif
```

**注意：**在整个结构的结尾只使用了一个 endif。

C Shell 中 if 语句的另一种格式是：

```
if ( expression ) command
```

或：

```
if ( expression ) \
    command
```

这用于只执行一条命令时。“\”是续行号。

while 语句

while 结构与 C 语言类似:

```
while ( expression )
    command_1
    command_2
.
.
.
end
```

switch 语句

switch 结构与 C 语言类似:

```
switch { word }
case str1:
    commands
.
.
.
breaksw
case strn:
    commands
.
.
.
breaksw
default:
    commands
.
.
.
breaksw
endsw
```

goto 语句

C Shell 也允许 goto 语句和标号,与 C 类似:

```
loop:
    command_1
    command_2
.
.
.
goto loop
```

### 6.2.5 为命令提供输入

C Shell 脚本中的命令使用运行它的 shell 的标准输入。这使得 C Shell 脚本可以完全

参与命令管道,但我们需要使用一个中间记号来向 shell 脚本中的命令提供行间数据。例如,下面的脚本编辑变量表列出的每一个文件,把每一行前的空格删除(< space> 和 < tab> 分别表示空格和跳格字符):

```
# debblank -- remove leading blanks
foreach i ( $argv )
ed - $i << 'EOF'
1,$ s/^[ <space> <tab> ] * //
w
q
'EOF'
end
```

记号 << 'EOF' 表示 ed 命令的标准输入,从 shell 脚本文件的下一行开始直到只含 'EOF' 的一行前为止。用单引号把 EOF 括起来的目的是,是不让 shell 对下面直到 'EOF' 前的文本行作变量代换。事实上,只要 < 后用以中止文本输入的字符串(这里是 EOF)的任何一部分被单引号括起来(如 E'OF', 'E'OF 等),shell 都将不对文本行进行变量代换。在这个例子中,因为我们用到了 1, \$ 的编辑符,我们必须确保 shell 不对 \$ ... 进行变量代换。当然,我们也可以在 \$ 前插入一个反斜线(\)来禁止变量代换,如:

```
1,\ $ s/^[ <space> <tab> ] * //
```

但使用单引号的 'EOF' 更清晰可靠。但注意两者不能同时使用。

### 6.2.6 捕获中断

如果我们的 shell 脚本创建临时文件,我们希望能捕获 shell 脚本的中断,这样我们可以清除这些临时文件。我们可以在 shell 脚本的开头使用下面的语句:

```
onintr label
```

这里 label 是一个程序标号,标记处理中断的代码。当 shell 接到一个中断信号后, C Shell 将自动执行:

```
goto label
```

命令,并执行 label 段脚本所包含的命令。这时如果我们希望脚本以非 0 值退出,在 label 段代码里加上一句:

```
exit 1
```

注意:下面的语句

```
onintr -
```

将使 shell 脚本忽略所有的中断信号。

### 6.2.7 一个 C Shell 脚本的例子

下面这段 C Shell 脚本用以对一些 C 程序的备份,但只备份那些与上次备份的版本不相同的文件。备份的文件存在个人主目录下的 backup/子目录下:

```
#! /bin/csh
foreach i ( $argv )
if ( $i != * .c ) then
```

```
echo $i is not a .c program
continue
else
echo $i is a .c program
endif
echo check file /backup/$i:t
if (! -r /backup/$i:t) then
echo $i:t not in backup...not cp\'ed
continue
endif
echo compare two files $i and /backup/$i:t
cmp -s $i /backup/$i:t
if ($ status != 0) then
echo making new backup of $i
cp $i /backup/$i:t
endif
end
```

这段 C Shell 脚本可存成文件 bkup,并用下面的格式运行:

```
% ./bkup * .c
```

### 6.3 AWK 简介

AWK 是一种程序设计语言。它具有一般程序设计语言常见的功能。因为 AWK 语言具有某些特点,如:解释执行不需先编译;变量无类型的区分(Typeless),可使用字符串作为数组的下标(Associative Array)……等,所以,使用 AWK 编写程序比起使用其他语言更简洁便利且节省时间。AWK 还具有一些内部功能,使得 AWK 善于处理记录(Record),域(Field)形式的数据;除此之外,AWK 内部有管道的功能,可将处理中的数据传送给外部的 shell 命令加以处理,再将 shell 命令处理后的数据返回 AWK 程序,这个特点也使得 AWK 程序很容易使用系统资源。

由于 AWK 具有上述特色,在问题处理的过程,可轻易使用 AWK 来编写一些小工具;这些小工具并非用来解决整个大问题,它们只个别扮演解决问题过程中的某些角色,可凭借由 shell 所提供的管道,将数据按需要传送给不同的小工具进行处理,以解决整个大问题。这种解题方式,使得这些小工具可因不同需要而被重复组合及使用(reuse);也可凭借这种方式来先行测试大程序原型的可行性与正确性,将来如果需要较高的执行速度时再用 C 语言来改写。这是 AWK 最常被应用的地方。如果能常常这样处理问题,读者就会以更高的角度思考抽象的问题,而不会拘泥于细节。这正是 UNIX 的精髓所在。

为便于解释 AWK 程序结构及有关术语(terminology),先以一个员工工资文件(emp.dat)为例:

|      |       |     |     |
|------|-------|-----|-----|
| A125 | Jenny | 100 | 210 |
| A341 | Dan   | 110 | 215 |
| P158 | Max   | 130 | 209 |

P148 John 125 220

A123 Linda 95 210

文件中各域依次为:员工 ID,姓名,工资,及实际工时。ID 中的第一码为部门识别码。“A”,“P”分别表示“组装”及“包装”部门。下面着重于说明 AWK 程序的主要结构及工作原理,并对一些重要的名词辅以必要的解释。由这部分内容,读者可体会出 AWK 语言的主要精神及 AWK 与其他程序设计语言的差异。

### 6.3.1 名词定义

(1)记录(record):AWK 从数据文件上读取数据的基本单位。以上列文件 emp.dat 为例,AWK 读入的:

第一条记录是“A125 Jenny 100 210”

第二条记录是“A341 Dan 110 215”

一般而言,一条记录相当于数据文件上的一行数据。

(2)域(Field):为记录上被分隔开的子字符串。以记录“A125 Jenny 100 210”为例,

| 第一栏  | 第二栏   | 第三栏 | 第四栏 |
|------|-------|-----|-----|
| A125 | Jenny | 100 | 210 |

一般是以空白字符来分隔相邻的域。

### 6.3.2 如何执行 AWK

在命令行上输入:

```
awk 'AWK 程序' 数据文件文件名
```

则 AWK 会先编译该 AWK 程序,然后执行该程序来处理所指定的数据文件。

### 6.3.3 AWK 程序的主要结构

AWK 程序中主要语法是表达式 {动作},故常见的 AWK 程序形式如下:

表达式 1 |动作 1 |

表达式 2 |动作 2 |

.....

表达式 n |动作 n |

AWK 可接受许多不同形式的表达式。一般常使用“关系判断式”(Relational expression)来当成表达式。例如: $x > 34$  是一个表达式,判断变量  $x$  与 34 是否存在大于的关系。 $x = y$  是一个表达式,判断变量  $x$  与变量  $y$  是否存在等于的关系。

AWK 提供 C 语言中常见的关系运算符(Relational Operators),如:

$>$ ,  $<$ ,  $>=$ ,  $<=$ ,  $=$ ,  $!=$ .

除此之外,AWK 还提供  $\sim$  (符合) 及  $!\sim$  (不符合) 二个关系运算符。例如:

```
"banana" ~an/
```

整个是一个表达式,因为“banana”中含有可符合  $/an/$  的子字符串,故这种关系式成立(true),整个表达式的值也是 true。

动作是由许多 AWK 指令构成。AWK 的指令与 C 语言中的指令十分类似。例如:

```
AWK 的 I/O 指令 : print, printf( ), getline..
```

AWK 的流程控制指令: `if(...) {...} else {...}, while(...) {...} ...`

AWK 会先判断(Evaluate)表达式的值,如果表达式判断后的值为 `true`(或不为 0 的数字,或不是空的字符串),则 AWK 将执行该表达式所对应的动作。相反的,如果表达式的值不为 `true`,则 AWK 将不执行该表达式所对应的动作。

例如,如果 AWK 程序中有下列两指令:

```
50 > 23           | print "Hello! The word!!" ;
"banana" /123/    | print "Good morning !" ;
```

AWK 会先判断 `50 > 23` 是否成立。因为该式成立,所以 AWK 将输出 "Hello! The word!!"。而另一表达式为 `"banana" /123/`,因为 "banana" 内未含有任何子字符串可符合 `/123/`,该表达式的值为 `false`,故 AWK 将不会输出 "Good morning !"

### 6.3.4 AWK 的域变量

AWK 内部的域变量及其含义如下:

| 域变量 | 含 义                      |
|-----|--------------------------|
| \$0 | 为一字符串,其内容为目前 AWK 所读入的记录。 |
| \$1 | 代表 \$0 上第一个域的数据。         |
| \$2 | 代表 \$0 上第二个域的数据。         |
| ... | 其余类推                     |

读入记录时,AWK 是按下面的规则修正这些内部的域变量:

- (1) 当 AWK 从数据文件中读取一条记录时,AWK 会使用内部变量 `$0` 予以记录。
- (2) 每当 `$0` 被调用时(例如:读入新的记录 或自动变更 `$0,...`)AWK 会立刻重新分析 `$0` 的域情况,并将 `$0` 上各域的数据用 `$1, $2, ...` 予以记录。

AWK 提供了许多内部变量,使用者于程序中可使用这些变量来取得相关信息。常见的内部变量有:

| 内变量      | 含 义                                                    |
|----------|--------------------------------------------------------|
| NF       | (Number of Fields)为一整数,其值表示 <code>\$0</code> 上所存在的域数目。 |
| NR       | (Number of Records)为一整数,其值表示 AWK 已读入的记录数目。             |
| FILENAME | 正在处理的数据文件文件名。                                          |

例如,AWK 从数据文件 `emp.dat` 中读入第一条记录 "A125 Jenny 100 210" 之后,程序中:

```
$0 的值将是 "A125  Jenny 100 210"
$1 的值为 "A125"      $2 的值为 "Jenny"
$3 的值为 100         $4 的值为 210
NF 的值为 4           $NF  的值为 210
NR 的值为 1           FILENAME 的值为 "emp.dat"
```

### 6.3.5 AWK 的工作流程

执行 AWK 时,它会反复进行下列四步骤:

- (1) 自动从指定的数据文件中读取一条记录;

- (2)自动更新(Update)相关的内部变量的值。如：NF,NR,\$0...
- (3)逐次执行程序中的所有表达式|动作|指令；
- (4)当执行完程序中所有表达式|动作|时，如果数据文件中还有未读取的数据，则反复执行步骤1到步骤4。

AWK 会自动重复进行上述 4 个步骤，使用者不需要在程序中编写这个循环 (Loop)。

AWK 处理数据时，它会自动从数据文件中一次读取一条记录，并将该数据切分成一个个的域；程序中可使用 \$1,\$2,... 直接取得各个域的内容。这个特色让使用者易于用 AWK 编写处理和改变数据格式的程序。

以文件 emp.dat 为例，计算每人应发工资并输出报表。执行如下命令：

```
awk '! print $2, $3 * $4' emp.dat
```

执行结果输出如下：

```
Jenny 21000
Dan 23650
Max 27170
John 27500
Linda 19950
```

AWK 会自行一次读入一列数据，故程序中仅需告诉 AWK 如何处理所读入的记录。

几点说明：

- (1)Linux 命令行上，执行 AWK 的语法为：

```
awk 'AWK 程序' 欲处理的数据文件文件名
```

本范例中的程序部分为 {print \$2,\$3 \* \$4}。把程序置于命令行时，程序的前后需要以单引号括住。

- (2)emp.dat 为指定给该程序处理的数据文件文件名。

- (3)本程序中使用的表达式|动作|语法：

```
表达式      动作
| print $2, $3 * $4 |
```

表达式部分被省略，表示无任何限制条件。故 AWK 读入每条记录后都将无条件执行这个动作。

(4)print 为 AWK 所提供的输出指令，会将数据输出到 stdout(屏幕)。print 的参数间以“,”隔开，输出数据时会以空白隔开。

(5)将上述的程序部分储存于文件 pay1.awk 中。执行命令时再指定 AWK 程序文件的文件名。这是执行 AWK 的另一种方式，特别适用于程序较大的情况，其命令格式如下：

```
awk -f AWK 程序文件名 数据文件文件名
```

故执行下列两命令，将产生同样的结果：

```
awk -f pay1.awk emp.dat
awk '{ print $2, $3 * $4 }' emp.dat
```

用户可使用“-f”参数，让 AWK 主程序使其调用其他函数。其命令格式如下：

```
awk -f AWK 主程序文件名 -f AWK 函数文件名 数据文件文件名
```

- (6)AWK 中也提供与 C 语言中类似用法的 printf() 函数。使用该函数可进一步控

制数据的输出格式。例如,编辑另一个 AWK 程序如下,并取名为 `pay2.awk`:

```
! printf("\ %6s Work hours: %3d Pay: %5d\n", $2, \ $3, $3 * $4) }
```

执行下列命令

```
awk -f pay2.awk emp.dat
```

执行结果屏幕出现:

```
Jenny Work hours: 100 Pay: 21000
Dan Work hours: 110 Pay: 23650
Max Work hours: 130 Pay: 27170
John Work hours: 125 Pay: 27500
Linda Work hours: 95 Pay: 19950
```

### 6.3.6 输出合乎指定条件的记录

利用 `~` (符合) 和 `!~` (不符合) 二个关系运算符,可判断某字符串是否包含能符合所指定规范表达式的子字符串。所以很容易使用 AWK 来编写需要字符串比较,判断的程序。

例如,对于如下问题:

1. 组装部门员工调工资 5%, (组装部门员工的 ID 系以 "A" 开头)
2. 所有员工最后的工资如果仍低于 100, 则以 100 计。
3. 编写 AWK 程序输出新的员工工资报表。

这个程序需要先判断所读入的记录是否合乎指定条件,再进行某些动作。AWK 中表达式 `!动作!` 的语法已包含这种 `"if (条件) {动作}"` 的结构。编写如下的程序,并取名 `adjust1.awk`:

```
$1 ~ ^A. * / { $3 * = 1.05 } $3 < 100 { $3 = 100 }
! printf("%s %8s %d\n", $1, $2, $3);
```

然后使用如下命令:

```
awk -f adjust1.awk emp.dat
```

结果如下:

```
A125 Jenny 105
A341 Dan 115
P158 Max 130
P148 John 125
A123 Linda 100
```

几点说明:

(1) AWK 的工作程序是:从数据文件中每次读入一条记录,依序执行完程序中所有的表达式 `!动作!` 指令,再从数据文件中读进下一条记录继续进行处理:

| 表达式                        | 动作                                                   |
|----------------------------|------------------------------------------------------|
| <code>\$1 ~ ^A. * /</code> | <code>{ \$3 * = 1.05 }</code>                        |
| <code>\$3 &lt; 100</code>  | <code>{ \$3 = 100 }</code>                           |
|                            | <code>! printf("%s %8s %d\n", \$1, \$2, \$3);</code> |

(2) 第一个表达式 `!动作!` 是:

```
$1 ~ ^A. * / { $3 * = 1.05 }
```

`$1 ~ ^A. * /` 用来判断该条记录的第一栏是否包含以 "A" 开头的子字符串。其中

`/^A.*` 是一个规范表达式,用以表示任何以“A”开头的字符串。动作部分为 `$3 * = 1.05`,与 `$3 = $3 * 1.05` 意义相同。运算符字符串,“`*`”的用法则与 C 语言中一样。以后遇到这种与 C 语言中用法相同的运算符字符串或语法将不予赘述。

(3)第二个表达式{动作}是:

```
$3 < 100 { $3 = 100 }
```

如果第三栏的数据内容(工资)小于 100,则调整为 100。

(4)第三个表达式{动作}是:

```
{ print("%s %-8s %d\n", $1, $2, $3) }
```

省略了表达式(无条件执行动作),故所有记录调整后的数据都将被输出。

### 6.3.7 AWK 中数组的特色

AWK 程序中允许使用字符串当作数组的下标(index)。利用这个特色十分有助于数据统计工作。(使用字符串当下标的数组称为 Associative Array) 下面举例说明。

首先建立一个数据文件,并取名为 `reg.dat`。这是学生注册的数据文件;第一栏为学生姓名,其后为该生所上课程:

|      |      |      |      |
|------|------|------|------|
| Ma   | 操作系统 | 软件工程 | 体育   |
| Song | 政治   | 硬件结构 | 软件工程 |
| Wang | 体育   | 图像处理 | 操作系统 |
| Lisa | 图像处理 | 人工智能 |      |
| Lily | 体育   | 硬件结构 |      |

AWK 中,使用数组前不需要定义数组名称及其大小。例如,希望用数组来记录 `reg.dat` 中各门课程的上课人数情况,有二项信息必须要储存:

- 课程名称,如:“操作系统”、“软件工程”等,共有那些课程事前并不明确。
- 各课程的上课人数。如:有几个人修“操作系统”。

在 AWK 中只要用一个数组就可同时记录上列信息。其方法如下:使用一个数组 `Number[ ]`,以课程名称当 `Number[ ]` 的下标,以 `Number[ ]` 中不同下标所对应的元素的值代表上课人数。

例如,有 2 个学生修“操作系统”,则以

```
Number["操作系统"] = 2
```

表示。如果修“操作系统”的人数增加一人,则

```
Number["操作系统"] = Number["操作系统"] + 1
```

或

```
Number["操作系统"]++
```

那么如何取出数组中储存的信息?回想一下 C 语言,定义 `int Arr[100]` 后,如果想得知 `Arr[ ]` 中所储存的数据,只需要用一个循环,如:

```
for(i=0; i<100; i++) printf("%d\n", Arr[i]);
```

即可。上式中:

数组 `Arr[ ]` 的下标:0,1,2,...,99

数组 `Arr[ ]` 中各下标所对应的值: `Arr[0]`, `Arr[1]`, ..., `Arr[99]`

但 AWK 中使用数组并不需要事先定义。以刚才使用的 `Number[ ]` 而言,程序执行

前,并不知将来有哪些课程名称可能被当成 Number[ ] 的下标。AWK 提供了一个指令,凭借该指令 AWK 会自动找寻数组中使用过的所有下标。以 Number[ ] 为例,AWK 将会找到“操作系统”、“软件工程”等等。使用该指令时,需要指定所要找寻的数组及一个变量,AWK 会使用该变量来记录从数组中找到每一个下标。例如:

```
for(course in Number){...}
```

指定用 course 来记录 AWK 从 Number[ ] 中所找到的下标。AWK 每找到一个下标时,就用 course 记录该下标的值且执行 {...} 中的指令。凭借这个方式便可取出数组中储存的信息。

下面举个例子。问题是:统计各科上课人数,并输出结果。

建立如下程序,并取名为 course.awk:

```
{for(i=2;i<=NF;i++) Number["$i"]++}
END {for(course in Number)
    printf("%-10s %d\n", course, Number[course])}
```

执行下列命令:

```
awk -f course.awk reg.dat
```

执行结果如下:

```
政治      1
软件工程   2
人工智能   1
硬件结构   2
图像处理   2
操作系统   2
体育       3
```

几点说明:

(1) 这程序包含两个表达式:动作|指令。

```
表达式      动作
              {for(i=2;i<=NF;i++) Number["$i"]++}
END          {for(course in Number)
              printf("%-10s %d\n",course,Number[course])}
```

(2) 第一个表达式|动作|指令中省略了表达式部分。故随着每条记录的读入其动作部分将逐次无条件被执行。以 AWK 读入第一条数据“Ma 操作系统软件工程体育”为例,因为该条数据 NF = 4(有 4 个域),故该动作的 for 循环中 i = 2,3,4:

| i | \$i    | 最初 Number["\$i"]               | Number["\$i"]++ |
|---|--------|--------------------------------|-----------------|
| 2 | “操作系统” | AWK default Number["操作系统"] = 0 | 1               |
| 3 | “软件工程” | AWK default Number["软件工程"] = 0 | 1               |
| 4 | “体育”   | AWK default Number["体育"] = 0   | 1               |

(3) 第二个表达式|动作|指令中,END 为 AWK 的保留字,为表达式的一种。END 成立(其值为 true)的条件是:AWK 处理完所有数据,即将离开程序时。平常读入记录时,END 并不成立,故其后的动作并不被执行;只有当 AWK 读完所有数据时,该动作才会被执行(注意,不管记录有多少条,END 仅在最后才成立,故该动作仅被执行一次。)BEGIN 与 END 有点类似,是 AWK 中另一个保留的表达式。惟一不同的是:以 BEGIN 为表达

式的动作于程序一开始执行时,被执行一次。

(4)NF 为 AWK 的内部变量,用以表示 AWK 正处理的数据中所包含域的个数。

AWK 程序中如果含有以 \$ 开头的自定义变量,都将以如下方式解释:以 i = 2 为例, \$i = \$2 表示第二个域数据。(实际上, \$ 在 AWK 中为一运算符,用以取得域数据。)

### 6.3.8 AWK 程序中使用 Shell 命令

AWK 程序中允许呼叫 Shell 指令,并提供管道解决 AWK 与系统间数据传递的问题。所以 AWK 很容易使用系统资源。读者可利用这个特色来编写某些适用的系统工具。

下面是一个例子:写一 AWK 程序来输出登录人数。

将下列程序建文件,命名为 count.awk:

```
BEGIN
    while ("who" | getline) n++
    print n
}
```

并执行下列命令:

```
awk -f count.awk
```

执行结果将会输出目前登录人数。

几点说明:

(1)AWK 程序并不一定要处理数据文件。以本例而言,仅输入程序文件 count.awk,未输入任何数据文件。

(2)BEGIN 和 END 同为 AWK 中的一种表达式。以 BEGIN 为表达式的动作,只有在 AWK 开始执行程序,尚未打开任何输入文件前,被执行一次。(注意,只被执行一次)

(3)"|" 为 AWK 中表示管道的符号。AWK 把管道前的字符串"who"当成 shell 上的命令,并将该命令送往 shell 执行,执行的结果(原先应于屏幕输出)则凭借管道送进 AWK 程序中。

(4)getline 为 AWK 所提供的输入指令。其语法如下:

| 语法                 | 由何处读取数据   | 数据读入后置于                  |
|--------------------|-----------|--------------------------|
| getline var < file | 所指定的 file | 变量 var(var 省略时,表示置于 \$0) |
| getline var        | 管道        | 变量 var(var 省略时,表示置于 \$0) |
| getline var        | 标准输入      | 变量 var(var 省略时,表示置于 \$0) |

当表达式为 BEGIN 或 END 时,getline 将由 stdin 读取数据,否则由 AWK 正处理的数据文件上读取数据。getline 一次读取一行数据,它的返回值由下面的规则确定:

|               |       |
|---------------|-------|
| 如果读取成功        | 返回 1  |
| 如果读取失败        | 返回 -1 |
| 如果遇到文件结束(EOF) | 返回 0  |

本程序使用 getline 所返回的数据来作为 while 判断循环停止的条件。

### 6.3.9 如何读取命令行上的参数

大部分的应用程序都允许使用者在命令后增加一些选择性的参数。执行 AWK 命令时这些参数大部分用于指定数据文件文件名,有时希望在程序中能从命令行上得到一些

其他用途的数据。本小节中将叙述如何在 AWK 程序中取用这些参数。

建立文件如下,命名为 `see_arg` :

```
awk '
BEGIN {
  for( i=0; i<ARGC; i++ )
    print ARGV[i] # 依次输出 AWK 所记录的参数
}
' $*
```

执行如下命令 :

```
% see_arg first-arg second-arg
```

结果屏幕出现 :

```
awk
first-arg
second-arg
```

逐个打印出 `awk` 的命令行参数。

几点说明 :

(1) `ARGC`, `ARGV` 为 AWK 所提供的内部变量。`ARGC` 为一整数。代表命令行上,除了选项 `-v`, `-f` 及其对应的参数的外所有参数的数目。`ARGV` 为一字符串数组。`ARGV[0]`, `ARGV[1]`, ..., `ARGV[ARGC-1]` 分别代表命令行上相对应的参数。

例如,当命令行为:

```
% awk -vx = 36 -f program1 data1 data2
```

或:

```
% awk '{ print $1, $2 }' data1 data2
```

其 `ARGC` 和 `ARGV` 的值分别为:

```
ARGC      的值为    3
ARGV[0]   的值为    "awk"
ARGV[1]   的值为    "data1"
ARGV[2]   的值为    "data2"
```

命令行上的 `"-f program1"`, `"-vx = 36"`, 或程序部分 `"{ print $1, $2 }"` 都不会列入 `ARGC` 及 `ARGV` 中。

(2) AWK 利用 `ARGC` 来判断应打开的数据文件个数。但使用者可强行改变 `ARGC`; 当 `ARGC` 的值被使用者设为 1 时, AWK 将被蒙骗, 误以为为命令行上并无数据文件文件名, 故不会以 `ARGV[1]`, `ARGV[2]`, ... 为文件名来开文件读取数据; 但于程序中仍可凭借 `ARGV[1]`, `ARGV[2]`, ... 来取得命令行上的数据。

某一程序 `test1.awk` 如下 :

```
BEGIN:
  number = ARGC # 先用 number 记住实际的参数个数
  ARGC = 2 # 自行更改 ARGC=2, AWK 将以为只有一个数据文件
              # 仍可凭借 ARGV[] 取得命令行上的数据
  for( i=2; i< number; i++ ) data[i] = ARGV[i]
  ;
```

于命令行上输入:

```
% awk -f test1.awk data_file apple orange
```

执行时 AWK 会打开数据文件 `data_file` 以进行处理。不会打开以 `apple,orange` 为文件名的文件(因为 ARGV 被改成 2)。但仍可凭借 ARGV[2],ARGV[3]取得命令行上的参数 `apple,orange`。

(3)可以用下列命令来达到上例的效果:

```
% awk -f test2.awk -v data[2]="apple" -v data[3]="orange" data_file
```

### 6.3.10 AWK 的内部变量( Built-in Variables )

ARGC

表示命令行上除了选项 -F,-v,-f 及其所对应的参数外的所有参数的个数。

ARGV

一个数组用以记录命令行上的参数。

FILENAME:

用以表示目前正在处理的数据文件的文件名。

FS

域分隔字符。

\$0

表示目前 AWK 所读入的记录。

\$1, \$2, ...

分别表示所读入的记录的第一栏,第二栏。

NR

从 AWK 开始执行该程序后所读取的记录数。

FNR

与 NR 功能类似。不同的是 AWK 每打开一个新的数据文件,FNR 便从 0 重新累计 NF 表目前的记录所在的域数。AWK 每读入一条数据后,在程序中可以从 NF 来得知该条数据包含的域个数。在下一条数据被读入之前,NF 并不会改变。但使用者如果自行使用 \$0 来记录数据,例如使用 `getline`,这时 NF 将代表新的 \$0 上所记载的数据的域个数。

OFS

输出时的域分隔字符。缺省值 " "(一个空白),详见下面说明。

ORS

输出时记录的分隔字符。缺省值 "\n"(跳行),见下面说明。

OFMT

数值数据的输出格式。缺省值 "%.6g"(如果需要时最多输出 6 位小数)。

用 `print` 指令一次输出多项数据。例如:

```
print $1, $2
```

输出数据时,AWK 会自动在 \$1 与 \$2 的之间补上一个 OFS 的值(缺省值为一个空白)。每次使用 print 输出数据后,AWK 会自动补上 ORS 的值(缺省值为跳行)。使用 print 输出数值数据时,AWK 将采用 OFMT 的值为输出格式。例如:

```
print 2/3
```

AWK 将会输出 0.666667。程序中可通过改变这些变量的值,来改变指令 print 的输出格式。

#### RS( Record Separator)

AWK 从数据文件上读取数据时,依 RS 的定义把数据切割成许多记录,而 AWK 一次仅读入一个记录以进行处理。RS 的缺省值是“\n”。所以一般 AWK 一次仅读入一行数据。有时一个记录包括了几列数据(Multi-line Record)。这种情况下不能再以“\n”来分隔并邻的记录,可改用空白行来分隔。在 AWK 程序中,令 RS = “ ”表示以空白行来分隔并邻的记录。

#### RSTART

与使用字符串函数 match() 有关的变量,详见下面说明。

#### RLENGTH

与使用字符串函数 match() 有关的变量。用户使用 match(...) 函数后,AWK 会将 match(...) 执行的结果以 RSTART,RLENGTH 记录。

#### SUBSEP(Subscript Separator)

数组中下标的分隔字符,缺省值为“\034”。实际上,AWK 中的数组只接受字符串中的下标,如:Arr["John"]。但使用者在 AWK 中仍可使用数字作为数组的下标,甚至可使用多维的数组(Multi-dimentional Array)如:Arr[2,79]。事实上,AWK 在接受 Arr[2,79]之前,就已先把其下标转换成字符串“2\03479”,其后便以 Arr["2\03479"]代替 Arr[2,79]。可参考下例:

```
awk 'BEGIN { Arr["2,79"] = 78
        print Arr["2,79"]
        print Arr["2 \ 79 "]
        print Arr["2 \ 03479"]
        idx = 2 SUBSEP 79
        print Arr[idx]
    }
    $ x
```

执行结果输出:

```
78
78
78
78
```

### 6.3.11 AWK 的内部函数 (Built-in Functions)

#### 1. 字符串函数

**index:** 原字符串, 找寻的子字符串

如果原字符串中含有欲找寻的子字符串, 则返回该子字符串在原字符串中第一次出现的位置, 如果未曾出现该子字符串则返回 0。例如执行:

```
awk 'BEGIN| print index("8-12-94","-") |'
```

结果输出 2。

**length:** 字符串

返回该字符串的长度。例如执行:

```
awk 'BEGIN | print length("John") |'
```

结果输出 4。

**match**(原字符串, 用以找寻比较规范的表达式)

AWK 会在原字符串中找寻合乎指定的规范表达式的子字符串。如果合乎条件的子字符串有多个, 则以原字符串中最左方的子字符串为准。AWK 找到该字符串后会以这种字符串为依据进行下列动作:

(1) 设定 AWK 内部变量 **RSTART**, **RLENGTH** :

**RSTART** &= 符合条件的子字符串在原字符串中的位置。

&= 0 ; 如果未找到符合条件的子字符串。

**RLENGTH** &= 符合条件的子字符串长度。

&= -1 ; 如果未找到符合条件的子字符串。

(2) 返回 **RSTART** 的值。

例如执行:

```
awk ' BEGIN |
    match( "bansna", /(an)/ )
    print RSTART, RLENGTH
    ;'
```

执行结果输出 2 4。

**split**(原字符串, 数组名称, 分隔字符)

AWK 将依所指定的分隔字符 (field separator) 来分隔原字符串成一个个的域 (field), 并以指定的数组记录各个被分隔的域。例如:

```
ArgList = "SP12p89"
split( ArgList, Arr, /Pp/ )
```

执行后 Arr[1] = 5, Arr[2] = 12, Arr[3] = 89。

**sprintf**(输出的格式, 输出的数据, 输出的数据, ...)

该函数的用法与 AWK 或 C 的输出函数 **printf**() 相同。所不同的是 **sprintf**() 会将要求输出的结果当成一个字符串返回。一般最常使用 **sprintf**() 来改变数据格式。如, x 为一数值数据, 如果欲将其变成一个含二位小数的数据, 可执行如下指令:

```
x = 28
x = sprintf("%f", x)
```

执行后  $x = "28.00"$ 。

sub( 比较用的规范表达式, 将替换的新字符串)

将原字符串中第一个(最左边)合乎所指定的规范表达式的子字符串改以新字符串。

注意:

(1) 第二个参数“将替换的新字符串”中可用“&”来代表“合乎条件的子字符串”。

如, 执行下列指令:

```
A = "a6 b12anan212.45an6a"
sub(/(an)+[0-9]*\./, "&|", A)
```

结果输出: a6b12|anan212|.45an6a

(2) sub() 不仅可执行替换的功能, 当第二个参数为空字符串("") 时, sub() 所执行的是“删除指定字符串”的功能。

(3) sub() 与 match() 的搭配使用, 可逐次取出原字符串中合乎指定条件的所有子字符串。例如执行下列程序:

```
awk '
BEGIN {
    data = "p12-P34 P56-p61"
    while( match( data, /[0-9]+/ ) ) {
        print substr(data, RSTART, RLENGTH)
        sub(/ [0-9]+/, "", data)
    }
}
```

结果输出:

```
12
34
56
61
```

(4) sub() 中第三个参数(原字符串)如果未指定, 则其缺省值为 \$0。可用 sub(/ [0-9]+/, "", "digital") 表示 sub(/ [0-9]+/, "", "digital", \$0)

gsub: 比较用的规范表达式, 将替换的新字符串, 原字符串

这个函数与 sub() 一样, 同样也是进行字符串替换的函数。不同点是:

(1) gsub() 会替换所有符合条件的子字符串。

(2) gsub() 会返回被替换的子字符串。

substr: 字符串, 起始位置[, 长度]

返回从起始位置起, 指定长度的子字符串。如果未指定长度, 则返回起始位置到字符串末尾的子字符串。例如:

```
awk '{ BEGIN { print substr( "User: Wei-Lin Liu", 6) }
```

结果输出:

Wei-Lin Liu

## 2. 数学函数

`int(x)`:

返回  $x$  的整数部分(去掉小数)。例如：

`int(7.8)` 将返回 7

`int(-7.8)` 将返回 -7

`sqrt(x)`:

返回  $x$  的平方根。例如：

`sqrt(9)` 将返回 3

如果  $x$  为负数,则执行 `sqrt(x)`时将会造成 Run Time Error

`exp(x)`:

返回  $e$  的  $x$  次方。例如：

`exp(1)` 将返回 2.71828

`log(x)`:

返回  $x$  以  $e$  为底的对数值。例如：

`log(e) = 1`

如果  $x < 0$ ,则执行 `sqrt(x)`时将会造成 Run Time Error。

`sin(x)`:

$x$  需要以弧度为单位,`sin(x)`将返回  $x$  的  $\sin$  函数值。

`cos(x)`:

$x$  需要以弧度为单位,`cos(x)`将返回  $x$  的  $\cos$  函数值

`atan2(y, x)`:

返回  $y/x$  的  $\tan$  反函数的值,返回值系以弧度为单位。

`rand()`:

返回介于 0 与 1 之间的(近似)随机数; $0 < \text{rand}() < 1$ 。除非使用者自行指定 `rand()` 函数起始的 seed,否则每次执行 AWK 程序时,`rand()`函数都将使用同一个内定的种子,来产生随机数。

`srand(x)`:

指定以  $x$  为 `rand()` 函数起始的 seed。如果省略了  $x$ ,则 AWK 会以执行时的日期与时间为 `rand()` 函数起始的 seed。

# 7

## Linux 网络管理与服务

### 7.1 TCP/IP 网络配置

#### 7.1.1 TCP/IP 协议集

TCP/IP, 即传输控制协议/互联网协议 (Transmission Control Protocol/Internet Protocol), 实际上是一个由多种协议组成的协议集, 它定义了计算机通过网络互相通讯及协议集各层次之间通讯的规范。

TCP/IP 最初在支持军事和计算机科学研究在美国高等研究计划署的网络 ARPAnet 上发展起来, 所以 TCP/IP 有时候也被称为“DARPA 互联网协议”。从那以后, 许多其他的 TCP/IP 网络也相继投入使用, 如美国国家科学基金会的 NSFNET 和全世界数以千计的局域网或区域网, 所有这些网络被联结成一个巨大的联合体——因特网 (Internet)。

在一个 TCP/IP 网络上, 每个机器有一个 IP 地址, 这是一个 32-bit 的数字, 唯一地确定了一台机器。IP 地址通常由用“.”隔开的四个十进制数表示, 如 IP 地址 0x81124c15 (16 进制) 通常写成 129.18.76.21。

IP 地址由两部分组成: 网络地址和节点地址。网络地址由 IP 地址的高位组成, 节点地址由低位组成 (一般来说, 每个节点是网络上的一台机器)。这两个部分的大小取决于网络的类型。如在一个 B 类网络 (IP 地址的第一个字节在 128 到 191 之间) 上, IP 地址的头两个字节是网络地址, 后两个字节是节点地址。如 IP 地址为 129.18.76.21 的机器是网络 129.18 上的节点 76.21。

IP 地址的节点部分可以分出子网地址。利用子网技术, 大的网络可以被分为若干个小个子网, 每一个都可以独立维护。例如, 一个 B 类网络可以提供两个字节的节点信息, 最多可容纳 65534 个节点。如果指定节点地址的第一个字节 (整个 IP 地址的第三个字节) 作为子网地址, 第二个字节则是相应子网的节点地址。在这种情况下, IP 地址 129.18.76.21 表示网络 129.18 的子网 76 的节点 21。

同一或不同机器上希望通过 TCP/IP 进行通讯的进程通常要指定目的机器的 IP 地址和端口 (port) 地址。IP 地址给出数据从源地址传到目的地址的路径。端口地址是一个 16-bit 的数字, 它指定了目标机器上应该接收此项数据的一个特定的服务或应用。例如, telnet 允许用户从一台机器上启动另一台机器上的登录进程。在远程机器上, 通常有一个 telnetd 进程, 它监听着一个特定的端口 (通常是 23) 以等待连接请求。当用户执行 telnet 并给出将要登录的机器的地址后, telnet 程序尝试建立与远程机器端口 23 的连接。如果成功, telnet 和 telnetd 就能够互相通讯, 为用户提供远程登录。

TCP/IP 协议集由一系列协议组成。传输控制协议 (TCP) 负责提供两个进程 (可能

运行在网络的不同机器上)间可靠的、面向连接的通讯。用户数据报协议(UDP)类似 TCP,但它是无连接的、不可靠的服务。必要的话,使用 UDP 的进程必须提供它们自己的应答和同步程序。

TCP 和 UDP 以包(packet)为单位发送和接收数据。每个包除了包括要发送的信息外,还有一个包含源和目的地的端口地址的头。

互联网协议(IP)位于 TCP 和 UDP 所在的协议层之上,它负责选径和发送 TCP 或 UDP 包。IP 把每个 TCP 或 UDP 包封装在另一个包(称为 IP 数据包)里,它包含了源和目的地的 IP 地址以及路径信息。

**注意:** IP 对端口地址一无所知,端口只与 TCP 和 UDP 相关。同样,TCP 和 UDP 也不处理 IP 地址,IP 地址只与 IP 相关。

网络节点除有 IP 地址以外,通常还有一个名字。域名服务(Domain Name Service, DNS)把节点名翻译成 IP 地址,或把 IP 地址翻译成节点名,并在整个因特网上发布名字-IP 地址数据库。使用节点名使得一台机器的 IP 地址改变(例如这台机器被移到了另外一个网络上)时,不必担心别人在网络上找不到这台机器。这台机器的 DNS 记录只是更新了 IP 地址,所有用节点名对这台机器的参照继续有效。DNS 是一个巨大的、世界范围的分布式数据库,每个组织维护其中的一小片,列出本组织的机器。

现在,让我们讨论一下 TCP/IP 网络的基本模型。一个网络是一些机器通过物理网络介质(如以太网或串口线)连接起来的集合,每个网络都有自己内部处理选径和包传递的方法。

网络之间通过网关(gateway)连接起来。网关是一台同两个或更多个网络有直接连接的节点,网关可以在网络之间交换信息,把包从一个网络传递到另一个网络。例如,网关可能是一个有两个或多个以太网接口的工作站,每个接口连接到一个不同的网络,操作系统利用这种连接使这个工作站发挥网关的作用。网关在与它有直接连接的每个网络上都有一个 IP 地址。

IP 使用 IP 地址的网络部分确定包在机器之间传递的路线。网络上的每个机器都有一个路由表(routing table),它列出一些网络和相应的网关机器。为选定包到一个特定机器的传递路径,IP 首先查看目的地址的网络部分。如果路由表中有那个网络的记录,IP 就选定包通过相应的网关传递出去;否则,IP 指定包通过路由表中“缺省”的网关传递。

路由表中除对网络外,还可以有对指定机器的记录。另外,每个机器的路由表中都有它自己的一条记录。

我们可以用 netstat 命令来查看某机器当前的路由表,其输出大致类似下面的例子(假设本机是 129.18.76.21):

```
lark% netstat -rn
Kernel routing table
```

| Destination net/address | Gateway address | Flags | RefCnt | Use   | Interface |
|-------------------------|-----------------|-------|--------|-------|-----------|
| 129.18.76.0             | 129.18.76.21    | UN    | 0      | 23442 | eth0      |
| default                 | 129.18.76.99    | UGN   | 0      | 23442 | eth0      |
| 127.0.0.1               | 127.0.0.1       | UH    | 0      | 232   | lo        |
| 129.18.76.21            | 127.0.0.1       | UH    | 0      | 232   | lo        |

第一栏是路由表包括的目标网络或节点的地址。路由表第一条是对网络 129.18.76,这是本机所在网络。任何本机发到这个网络的包必须通过 129.18.76.21,即本机的 IP 地址。一般一台机器到自己网络的路由总是通过它自己。

Flags 栏给出目标地址的信息, U 表示此路由“up”, N 表示目标是一个网络,等等。RefCnt 和 Use 栏给出这条路由使用统计。Iface 栏列出路由使用的网络设备, eth0 表示以太网接口, lo 表示 loopback 设备。

路由表中第二条是缺省路由,适用于所有目的网络或节点地址不在路由表中的包。本例中, 129.18.76.99 可看作通向外界的门户,所有 129.18.76 子网的机器必须通过 129.18.76.99 与其他网络通讯。

路由表中第三条是对地址 127.0.0.1,这是 loopback 地址。当机器想与自己建立 TCP/IP 连接时适用这个地址。它使用 lo 作为接口设备,避免了 loopback 连接使用以太网接口 (eth0),这样网络带宽不会因机器与自己对话而浪费。

路由表中最后一条是对地址 129.18.76.21,这是本机的 IP 地址。如上述,它利用 127.0.0.1 作为自己的网关。

连接不同网络的网关的路由表通常类似下面的例子(假设这个网关在两个子网的地址分别为 129.18.76.99 和 129.18.113.4):

| Destination net/address | Gateway address | Flags | RefCnt | Use   | Iface |
|-------------------------|-----------------|-------|--------|-------|-------|
| 129.18.76.0             | 129.18.76.99    | UN    | 0      | 23442 | eth0  |
| 129.18.113.0            | 129.18.113.4    | UN    | 0      | 23442 | eth1  |
| default                 | 129.18.113.41   | UGN   | 0      | 23442 | eth1  |
| 127.0.0.1               | 127.0.0.1       | UH    | 0      | 232   | lo    |
| 129.18.76.99            | 127.0.0.1       | UH    | 0      | 232   | lo    |

本网关通过 eth0 设备与 129.18.76 网络连接,通过 eth1 设备与 129.18.113 网络连接,缺省路由是 129.18.113.41。如果 129.18.76 网络的机器想同外界通讯,它将把包发往它的网关 129.18.76.99,129.18.76.99 将把包发往它的缺省路由,即网关 129.18.113.41。如此下去,包从一个网关传递到下一个网关,直到抵达目的网络。这就是因特网所基于的基本结构:一个通过网关连接起来的、看起来无穷无尽的网络链条。

### 7.1.2 在以太网上设置 TCP/IP

事实上,没有任何网络硬件也可以使用 TCP/IP,把 TCP/IP 配置成“loopback”模式就可以使一台机器与自己交谈。

我们要介绍的是当在一个基于以太网的 TCP/IP 网络上增加一台机器时,如何配置它与网络的 TCP/IP 连接。关于如何建立一个完整的局域网不在我们的讨论范围内。

在设置 TCP/IP 之前,首先应该知道一台机器网络设置的如下信息:

IP 地址 (IP address)

不与任何其他机器重复的、用“.”隔开的一串数字,例如 129.18.76.99。对于 loopback 模式,机器的 IP 地址是 127.0.0.1。

子网掩码 (subnetwork mask)

格式类似 IP 地址的一串数字,当与机器的 IP 地址作逐位“与”运算时,得出机器所

在的子网地址。例如子网掩码是 255.255.255.0 时,如果 IP 地址是 129.18.76.21,则此 IP 地址的子网部分为 129.18.76。

**注意:**子网地址和网络地址是有区别的。对于一个 B 类网络,IP 地址的头两个字节(这里是 129.18)是网络地址,后两个是节点地址。但是,如果子网掩码设为 255.255.255.0,则 129.18.76 是子网地址(网络 129.18 的子网 76),而 20 是节点地址。

对于 loopback 设备来说,因为 loopback 地址总是 127.0.0.1,所以这个设备的掩码总是 255.0.0.0。

子网地址(subnetwork address)

IP 地址的由子网掩码决定的子网部分。如上所述,如果子网掩码设为 255.255.255.0,IP 地址是 129.18.76.21,则 129.18.76.0 是子网地址。

**注意:**只有 loopback 的系统没有子网地址。

广播地址(broadcast address)

这个地址用来向本子网的每一台机器发布广播包。一般来说,这等于在子网地址中以 255 代替节点地址。对于子网地址 129.18.76.0,广播地址是 129.18.76.255;而对于子网地址 129.18.0.0,广播地址是 129.18.255.255。只有 loopback 的系统没有广播地址。

网关的 IP 地址

作为通向外界的缺省路由的机器的 IP 地址。只有 loopback 的系统和孤立网络没有网关地址。

名字服务器(name server)的 IP 地址

用来处理节点名-地址转换的机器的 IP 地址。只有 loopback 的系统没有名字服务器地址。

rc 文件是系统在引导时由 init 进程执行的系统资源配置文件,它们运行基本的系统守护进程(daemon),如 sendmail 和 crond 等,也被用来配置网络参数。rc 文件通常在/etc/rc.d 或 /etc/rc\*.d 目录中。

现在我们以 Linux 为例,介绍一下基本的 TCP/IP 网络配置。在 Slackware 中,有关网络配置的 rc 文件有 rc.inet1 和 rc.inet2,rc.inet1 用来设置基本网络参数(如 IP 地址和路由信息),rc.inet2 启动基本的 TCP/IP 守护进程(telnetd,ftpd,...),在 rc.inet1 中使用了 ifconfig 和 route 命令。注意在其他操作系统中,配制网络的方式可能各不相同,但 ifconfig 和 route 是配制 TCP/IP 网络的基本命令。

ifconfig 用指定的参数,如 IP 地址、子网掩码、广播地址等配置网络设备接口。route 用来产生和修改路由表。

下面是 rc.inet1 文件的例子:

```
# ! /bin/sh
#
# rc.inet1      This shell script boots up the base INET system.
#
# Version:      @( # )/etc/rc.d/rc.inet1 1.01 05/27/93
#
```

```

HOSTNAME='cat /etc/HOSTNAME'

# Attach the loopback device.
/sbin/ifconfig lo 127.0.0.1
/sbin/route add -net 127.0.0.0 netmask 255.0.0.0 lo

# IF YOU HAVE AN ETHERNET CONNECTION, use these lines below to
# configure the eth0 interface. If you're only using loopback or SLIP,
# don't include the rest of the lines in this file.

# Edit for your setup.
IPADDR="129.18.76.21"      # REPLACE with YOUR IP address!
NETMASK="255.255.255.0"   # REPLACE with YOUR netmask!
NETWORK="129.18.76.0"     # REPLACE with YOUR network address!
BROADCAST="129.18.76.255" # REPLACE with YOUR broadcast address, if you
                           # have one. If not, leave blank and edit below.
GATEWAY="129.18.76.99"    # REPLACE with YOUR gateway address!

# Uncomment the line below to configure your ethernet card.
/sbin/ifconfig eth0 $ ${IPADDR} broadcast $ ${BROADCAST} netmask $ ${NETMASK}

# If the line above is uncommented, the code below can also be
# uncommented. It sets if the ethernet was properly initialized, and
# gives the admin some hints about what to do if it wasn't.
if [ ! $ ? = 0 ]; then
    cat << END
Your ethernet card was not initialized properly. Here are some
reasons why this may have happened, and the solutions:
1. Your kernel does not contain support for your card. Including all the network drivers in a Linux
   kernel can make it too large to even boot, and sometimes including extra drivers can cause system
   hangs. To support your ethernet, either edit /etc/rc.d/rc.modules to load the support at
   boottime, or compile and install a kernel that contains support.
2. You don't have an ethernet card, in which case you should comment out this section of /etc/rc.d/
   rc.inet1. (Unless you don't mind seeing this error...)
END
fi

# Uncomment these to set up your IP routing table.
/sbin/route add -net $ ${NETWORK} netmask $ ${NETMASK} eth0
if [ ! " $ GATEWAY" = "" ]; then
/sbin/route add default gw $ ${GATEWAY} netmask 0.0.0.0 metric 1
fi

```

```
# End of rc.inet1
```

ifconfig 命令的格式是:

```
ifconfig interface options
```

例如:

```
ifconfig lo 127.0.0.1
```

指定 lo (loopback) 设备的 IP 地址 127.0.0.1。而:

```
ifconfig eth0 129.18.76.21
```

指定 eth0 (第一个以太网) 设备的地址为 129.18.76.21。除了给出地址外,以太网设备通常要求用 netmask 选项设置子网掩码,用 broadcast 选项设置广播地址。

route 命令的格式是:

```
route add [ -net | -host ] destination [ gw gateway ]  
[ metric metric ] options
```

destination 是这条路由的目标地址 (或关键词 default), gateway 是这条路由的网关 IP 地址, metric 是这条路由的 metric 数。

route 命令用来往路由表里增加路由。我们至少应该给 loopback 设备、本网络和缺省网关添加路由。如我们的缺省路由是 129.18.76.99 时,应该用以下命令:

```
route add default gw 129.18.76.99
```

在 destination 之前使用 -net 或 -host 表示 destination 分别是一个网络或指定节点。(一般来说,路由是指向网络,但有时可能会有一个独立的机器需要自己的路由。这种情况下,必须用 -host 选项添加路由。)

metric 选项指定这条路由的 metric 值。当到一个指定的地方有一条以上的路由时,系统必须决定用哪一条,这时 metric 值低的路由优先级高。缺省路由的 metric 值设为 1,使缺省路由的优先级最高。

到一个指定的地方有一条以上的路由是怎么出现的呢? 首先,可能是在 rc.inet1 文件中对一个指定地址用了多个 route 语句。但是,如果系统运行了 routed 进程,路由表会动态地增加路由,因为其他机器可能在网络上广播路由信息,致使本机的路由表中增加额外的路由。设置缺省路由的 metric 值为 1,使任何路由表中新增加的路由的优先级都不可能高于缺省路由。

现在让我们来看一看 rc.inet2 文件。rc.inet2 运行 TCP/IP 的各种守护进程,这并不是使机器可以与网络交谈所必需的,所以放在另外一个文件里。我们总是必须先配置 rc.inet1 文件,以保证系统能从网络上接收包并能向网络发送包,然后再配置 rc.inet2 文件。

rc.inet2 运行的主要守护进程有 inetd, syslogd 和 routed, 其他服务也可以从 rc.inet2 启动。最重要的是 inetd, 它是其他系统守护进程的管理者。它运行在后台, 监听着一些特定的网络端口以等待连接请求。当一个连接建立时, inetd 产生一个对应于连接端口的守护进程。例如, 当一个到来的 telnet 连接一经建立, inetd 产生 in.telnetd, 它接管这个 telnet 连接, 这样, 网络守护进程只在需要时才被启动, 从而更简单、更经济。

syslogd 是系统记录守护进程, 它收集应用程序的记录信息并把它们存储在 /etc/syslogd.conf 指定的记录文件中。

routed 管理动态路由信息。当系统要往别的网络发送包时可能需要额外的路由，routed 可在不须用户介入的情况下处理路由表。

下面是一个 rc.inet2 的例子，它启动了 syslogd, inetd 和 routed。

```
# ! /bin/sh
# Sample /etc/rc.d/rc.inet2

# Start syslogd
if [ -f /usr/sbin/syslogd ]
then
    /usr/sbin/syslogd
fi

# Start inetd
if [ -f /usr/sbin/inetd ]
then
    /usr/sbin/inetd
fi

# Start routed
if [ -f /usr/sbin/routed ]
then
    /usr/sbin/routed -q
fi
```

下面介绍其他几个有关系统网络信息的文件。

/etc/hosts 包含一个 IP 地址和对应的节点名的列表。一般来说，/etc/hosts 只包含本机的地址。本地的名字服务器负责其他机器的地址-名字转换。

例如，如果本机的名字是 lark.ccc.com, IP 地址是 129.18.76.21, 则 /etc/hosts 类似

```
127.0.0.1      localhost
129.18.76.21   lark.ccc.com lark
```

在只有 loopback 的情况下，/etc/hosts 文件只有 127.0.0.1 一行。

/etc/resolv.conf

这个文件设置名字服务器，给出名字服务器的地址和本网络的 DNS 域名。域名是本地完整的节点名去掉第一部分。如对于 lark.ccc.com, 域名是 ccc.com。lark.ccc.com 如用名字服务器 129.18.76.56, 则 /etc/resolv.conf 将有下面两行

```
domain        ccc.com
nameserver     129.18.76.56
```

可以在 /etc/resolv.conf 中指定不止一个名字服务器，每个都必须以 nameserver 开头并独占一行。

设置节点名

一般应在系统启动文件中早于 rc.inet1 与 rc.inet2 的位置用 hostname 命令设置系

统的节点名。如 `/bin/hostname lark.ccc.com` 设置系统完整的节点名为 `lark.ccc.com`

一旦系统的所有网络配置文件都已修改好,就可以重新引导系统,启动网络了。

测试网络连接的最简单的方法就是使用 `telnet` 远程登录到其他系统。首先试着连接本地网络的系统,接着试着连接别的网络的系统。后者可以检测本机通过网关与外部世界的连接。

当通过网关连接远程系统成功、但连接本地网的机器却失败时,可能是子网掩码或路由表中本地网的路由设置错误。

当使用 IP 地址连接时成功、但使用节点名失败时,可能是名字服务器设置错误(检查 `/etc/resolv.conf`) 或到名字服务器的路由设置错误。

网络问题的一个最常见的来源是路由表设置错误。可以用

```
netstat -rn
```

命令显示路由表,不用 `-n` 选项显示节点或网络的名字,而不是地址。使用 `route` 增加或删除路由(用 `route del`)。

## 7.2 万维网(WWW)服务器的安装和配置

万维网(WWW, World Wide Web)是因特网(Internet)上广泛使用的一种信息发布方式。常用的免费万维网服务器有 CERN httpd, NCSA httpd 和 Apache。常用的万维网客户端程序(浏览器)有 Netscape Navigator(Communicator), lynx, hotjava, arena 等。

下面以 Apache HTTP 服务器为例,介绍万维网服务器的安装和配置。

### 7.2.1 安装

首先,从“Apache HTTP Server Project”的 FTP 目录

```
ftp://www.apache.org/apache/dist/
```

或 `http://www.apache.org/dist/`

或遍布全球的镜像节点上下载最新版本(写作本书时为 1.3.6)的 Apache HTTP 服务器的源程序包,其文件名格式是 `apache_1.3.6.tar.gz` 或 `apache_1.3.6.tar.Z`。把源程序包打开在一个工作目录下:

```
% tar zxvf apache_1.3.6.tar.gz
```

进入 `apache_1.3.6` 目录,先运行 “`./configure --help`”,看一看可配置的编译参数:

```
# ./configure --help
Usage: configure [options]
Options: [defaults in brackets after descriptions]
General options:
--quiet, --silent           do not print messages
--verbose, -v               print even more messages
--shadow[ = DIR]           switch to a shadow tree (under DIR) for building
```

Stand-alone options:

```
--help, -h          print this message
--show-layout        print installation path layout (check and debug)
```

#### Installation layout options:

```
--with-layout = [ F ; ] ID    use installation path layout ID (from file F)
--target = TARGET            install name-associated files using basename TARGET
--prefix = PREFIX            install architecture-independent files in PREFIX
--exec-prefix = EPREFIX      install architecture-dependent files in EPREFIX
--bindir = DIR               install user executables in DIR
--sbindir = DIR              install sysadmin executables in DIR
--libexecdir = DIR           install program executables in DIR
--mandir = DIR               install manual pages in DIR
--sysconfdir = DIR           install configuration files in DIR
--datadir = DIR              install read-only data files in DIR
--includedir = DIR           install includes files in DIR
--localstatedir = DIR        install modifiable data files in DIR
--runtimedir = DIR           install runtime data in DIR
--logfiledir = DIR           install logfile data in DIR
--proxycachedir = DIR        install proxy cache data in DIR
```

#### Configuration options:

```
--enable-rule = NAME        enable a particular Rule named 'NAME'
--disable-rule = NAME        disable a particular Rule named 'NAME'
                                [ IRIXN32 = yes IRIXNJS = no PARANOID = no ]
                                [ SHARED_CHAIN = default SHARED_CORE = default SOCKS4 ]
                                [ SOCKSS = no WANTHSREGEX = default ]

--add-module = FILE          on-the-fly copy & activate a 3rd-party Module
--activate-module = FILE      on-the-fly activate existing 3rd-party Module
--permute-module = N1 ; N2    on-the-fly permute module 'N1' with module 'N2'
--enable-module = NAME        enable a particular Module named 'NAME'
--disable-module = NAME        disable a particular Module named 'NAME'

[ access = yes      actions = yes      alias = yes      ]
[ asis = yes        auth = yes          auth_anon = no   ]
[ auth_db = no      auth_dbm = no       autoindex = yes  ]
[ cern_meta = no    cgi = yes           digest = no     ]
[ dir = yes         env = yes           example = no     ]
[ expires = no      headers = no        imap = yes      ]
[ include = yes     info = no           log_agent = no   ]
[ log_config = yes  log_referer = no    mime = yes      ]
[ mime_magic = no   mmap_static = no    negotiation = yes ]
[ proxy = no        rewrite = no        setenvif = yes   ]
[ so = no           spelling = no       status = yes     ]
[ unique_id = no    userdir = yes       usertrack = no   ]
```

|                         |                                                     |
|-------------------------|-----------------------------------------------------|
| --enable-shared = NAME  | enable build of Module named 'NAME' as a DSO        |
| --disable-shared = NAME | disable build of Module named 'NAME' as a DSO       |
| --with-perl = FILE      | path to the optional Perl interpreter               |
| --without-support       | disable the build and installation of support tools |
| --without-confadjust    | disable the user/situation adjustments in config    |

#### suEXEC options:

|                           |                                                           |
|---------------------------|-----------------------------------------------------------|
| --enable-suexec           | enable the suEXEC feature                                 |
| --suexec-caller = NAME    | set the suEXEC username of the allowed caller [ www ]     |
| --suexec-docroot = DIR    | set the suEXEC root directory [ PREFIX/share/htdocs ]     |
| --suexec-logfile = FILE   | set the suEXEC logfile [ PREFIX/var/log/suexec_log ]      |
| --suexec-userdir = DIR    | set the suEXEC user subdirectory [ public_html ]          |
| --suexec-uidmin = UID     | set the suEXEC minimal allowed UID [ 100 ]                |
| --suexec-gidmin = GID     | set the suEXEC minimal allowed GID [ 100 ]                |
| --suexec-safeopath = PATH | set the suEXEC safe PATH [ /usr/local/bin:/usr/bin:/bin ] |

#### Deprecated options:

|          |                                                |
|----------|------------------------------------------------|
| --layout | backward compat only: use --show-layout        |
| --compat | backward compat only: use --with-layout=Apache |

上述列表中有许多可配置参数,大多我们可以暂时不去管它们的意义,但有些必须现在就设置,如“--prefix”指定 Apache HTTP 服务器的安装目录,缺省的安装目录是 /usr/local/apache。如果用户希望安装到其他目录,就必须在运行“./configure”时在命令行指定。另外,用户需要设置的参数是有关模块的选项:

```
--enable-module = NAME
--disable-module = NAME
```

Apache 的设计使用了模块方式,需要的功能必须编译进 HTTP 服务器。上面列表中列出了哪些模块是缺省编译的,哪些不是。用户如果想添加或去掉一些模块,可以在运行“./configure”时用“--enable-module = NAME”或“--disable-module = NAME”设置。例如,如果用户想使用 Apache HTTP 服务器的代理功能,就可以在“./configure”行加上

```
--enable-module = proxy
```

可以使用数个“--enable-module = NAME”或“--disable-module = NAME”以设置很多模块。关于所有模块的具体说明,请参阅 Apache 软件包中自带的说明文档。

现在,我们可以按照下面的步骤编译 Apache HTTP 服务器:

```
# cd apache_1.3.6
# ./configure
(如果用户要指定参数,用下面的方式:
./configure --prefix = /home/httpd --enable-module = proxy ...)
# make
# make install
```

安装程序将把所有文件安装到缺省的 (/usr/local/apache) 或用户指定的 Apache HTTP 服务器目录中。用户可以马上用下面的命令启动 Apache HTTP 服务器,测试一下

是否安装正确：

```
# /usr/local/apache/bin/apachectl start
```

然后用浏览器打开地址 `http://localhost:8080/`，如果能看到 Apache HTTP 服务器的测试页面(“It Worked! The Apache Web Server ...”)，说明安装过程正确。

下一步是修改服务器的配置文件，以适合用户的个人要求。在 `/usr/local/apache/conf` 目录里有安装过程根据用户的配置参数设定的初始配置文件：`httpd.conf`、`srml.conf` 和 `access.conf`。以前这三个配置文件都要用到，现在的 Apache HTTP 服务器版本可以把所有的配置都放进 `httpd.conf` 文件。

`httpd.conf` 文件配置服务器的各项参数，如端口号、以哪个用户身份运行、文件树的根目录等。我们需要改动的可能有下面几个指令：`Group`、`ServerAdmin`、`ServerName` 等。`Group` 指 `httpd` 以哪个组成员的身份运行；`ServerAdmin` 是服务器管理员的电子邮件地址；`ServerName` 则是运行服务器的机器的节点名，如 `lark.ihep.ac.cn`。

改完 `httpd.conf` 文件的配置后，用户可以用下面的命令杀死旧的 `httpd` 进程，并用新的配置启动 `httpd`：

```
# /usr/local/apache/bin/apachectl restart
```

Apache 的所有在线手册存放在 `/usr/local/apache/htdocs/manual` 目录下，可用类似

```
http://myserver/manual/
```

的 URL 查看。

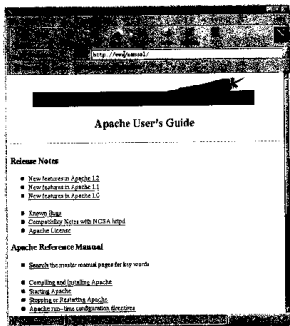


图 7.1 Apache 的在线手册

## 7.2.2 配置指令

万维网服务器的行为取决于 `/usr/local/apache/conf` 目录下配置文件 `httpd.conf` 的设置。配置语句称为指令(directive)，下面介绍一些常用的指令。设置代理服务的指令请参看下一节“代理服务器的设置”。

**ServerType**

定义:ServerType type

缺省:ServerType standalone

设置系统将以什么方式运行 httpd。type 可以为:

**inetd**

从系统进程 inetd 运行 httpd,启动 httpd 的命令加进/etc/inetd.conf。

**standalone**

httpd 以独立的守护进程运行,启动 httpd 的命令加进系统启动脚本文件(/etc/rc.d/rc.local 或/etc/rc.d/rc3.d/... )。

inetd 方法较少使用。对每个 HTTP 连接,inetd 将启动一个新的 httpd,连接完毕后,httpd 进程退出。用这种方法启动 httpd 较慢,但较安全,且少占系统资源。standalone 最常用使用,因为它相应较快,对于一个较忙的万维网站点,standalone 可能是唯一的选择。

**Port**

定义:Port number

缺省:Port 80

httpd 占用的端口号。number 可从 0 到 65535,但有些(特别是小于 1024 的)端口号可能为特定的协议保留,请查看/etc/services 文件。小于 1024 的端口号只能为 root 使用。

**User**

定义:User UNIX-userid

缺省:User # -1

**Group**

定义:Group UNIX-group

缺省:Group # -1

User 指令后的用户名(或 UID)和 Group 指令后的组(或 GID)定义 httpd 作为什么用户和什么组运行。为使用这条指令,standalone 服务器必须最初以 root 启动。为安全目的,应该以 nobody,nogroup 这样的用户和组运行 httpd。不要把 httpd 设成以 root 用户运行。

**ServerAdmin**

定义:ServerAdmin email-address

万维网站点的管理员的电子邮件地址

**ServerRoot**

定义:ServerRoot directory-filename

缺省:ServerRoot /usr/local/apache

设置服务器存放的目录,应包括 conf/和 logs/等子目录。其他配置文件的相对目录是相对于此目录的。

**ErrorLog**

定义>ErrorLog filename

缺省:ErrorLog logs/error\_log

设置错误信息输出文件。如果 filename 不是以斜线 (/) 开头,则是相对于 ServerRoot。

#### TransferLog

定义: TransferLog filename

缺省: TransferLog logs/access\_log

设置传输记录文件。如果 filename 不是以斜线 (/) 开头,则是相对于 ServerRoot。

#### ServerName

定义: ServerName fully-qualified domain name

定义服务器节点名,必须为服务器节点的合法 DNS 名字。有时候不定义这个名称 httpd 可能无法正常启动。

#### MinSpareServers

定义: MinSpareServers number

缺省: MinSpareServers 5

#### MaxSpareServers

定义: MaxSpareServers number

缺省: MaxSpareServers 10

定义最少和最多的空闲子服务器进程数。如果当前空闲子服务器进程数多于 MaxSpareServers,则父进程将杀死一些子进程;如果当前空闲子服务器进程数少于 MinSpareServers,父进程会产生一些子进程。

#### StartServers

定义: StartServers number

缺省: StartServers 5

httpd 启动时的子服务器进程数。由于子服务器进程数由实际使用状况自动控制,故没有必要改动这个数字。

#### Listen

定义: Listen [IP address:] port number

使 httpd 除缺省外在多个 IP 地址或端口上等候连接。缺省是 httpd 对发往机器的所有 IP 地址的请求作出反应,但只响应发往 Port 指令指定的端口的请求。

#### < VirtualHost >

定义: < VirtualHost addr[:port] ... > ... < /VirtualHost >

< VirtualHost > 和 < /VirtualHost > 用于包含一系列只适用于某个特定的虚拟节点的指令。当服务器接收到对某个虚拟节点请求,它使用在 < VirtualHost > 和 < /VirtualHost > 之中的配置指令。addr 可以是虚拟节点的 IP 地址或对应其 IP 地址的完整节点名。例如:

```
< VirtualHost 10.1.2.3 >
ServerAdmin webmaster@host.foo.com
DocumentRoot /www/docs/host.foo.com
ServerName host.foo.com
ErrorLog logs/host.foo.com-error_log
```

```
TransferLog logs/host.foo.com-access_log
< /VirtualHost >
```

每一个 VirtualHost 必须对应于服务器的一个不同的 IP 地址或不同的节点名。可以用“ifconfig alias”命令将服务器所在机器配置成可以多个地址接收 IP 包。

#### DocumentRoot

定义: DocumentRoot directory-filename

缺省: DocumentRoot /usr/local/apache/htdocs

确定 httpd 发布文件的缺省根目录。除了符合别名的以外,服务器将从 URL 中取出的所有路径前加上 DocumentRoot,以得到真实的本地路径。例如访问

```
http://www.ccc.com/test/index.html
```

将对应于本地文件 /usr/local/apache/htdocs/test/index.html。

#### UserDir

定义: UserDir directory/filename

缺省: UserDir public\_html

定义用户主目录下的一个目录,对某个用户的文件的请求都相对于这个目录。如对

```
http://www.foo.com/~bob/one/two.html
```

将被服务器翻译成:

```
UserDir public_html -> ~bob/public_html/one/two.html
UserDir /usr/web -> /usr/web/bob/one/two.html
UserDir /home/* /www -> /home/bob/www/one/two.html
```

下面的指令将向客户软件发送转移地址:

```
UserDir http://www.foo.com/users -> http://www.foo.com/users/bob/one/two.html
UserDir http://www.foo.com/* /usr -> http://www.foo.com/bob/usr/one/two.html
UserDir http://www.foo.com/~* / -> http://www.foo.com/~bob/one/two.html
```

**注意:**“UserDir ./”将映射“/~root”到“/”,请不要使用。

#### DirectoryIndex

定义: DirectoryIndex local-url local-url ...

缺省: DirectoryIndex index.html

当客户请求是一个目录索引(在目录名后加“/”)时,提供给此指令指定的资源通常是此目录下的一个文件。如果提供几个 local-url,则使用找到的第一个;如果一个也没找到,则给出此目录文件的列表。如:

```
DirectoryIndex index.html
```

将使 http://myserver/docs/ 映射成 http://myserver/docs/index.html。local-url 不必都相对于请求的目录,如:

```
DirectoryIndex index.html index.txt /cgi-bin/index.pl
```

使在请求目录下 index.html 和 index.txt 都没有时执行 CGI 脚本文件 /cgi-bin/index.pl。

#### Alias

定义: Alias url-path directory-filename

URL 的路径部分若以 url-path 开头则被映射到以 directory-filename 开头的本地文件,从

而允许文件存储在本地文件系统除 DocumentRoot 以外别的地方。如：

```
Alias /image /ftp/pub/image
```

所有对 `http://myserver/image/foo.gif` 的请求将使服务器返回文件 `/ftp/pub/image/foo.gif`。但注意如果在 `url-path` 的结尾使用 `"/`，如：

```
Alias /icons/ /usr/local/apache/icons/
```

则服务器只对 `/icons/` 应用别名，直不对 `/icons` 应用别名。

#### ScriptAlias

定义：`ScriptAlias url-path directory-filename`

类似 `Alias`，但它表示目的目录包含 CGI 程序。用户应使用下行：

```
ScriptAlias /cgi-bin/ /usr/local/apache/cgi-bin/
```

使 `http://myserver/cgi-bin/foo` 的请求能正确执行 CGI 程序 `/usr/local/apache/cgi-bin/foo`。

#### AccessFileName

定义：`AccessFileName filename`

缺省：`AccessFileName .htaccess`

当向客户端程序返回一个文件时，服务器在文件路径上所有加了访问控制的目录上搜寻此指令定义的文件。例如

```
AccessFileName .acl
```

在返回文件 `/usr/local/web/index.html` 前，服务器将读取 `/.acl`、`/usr/.acl`、`/usr/local/.acl` 和 `/usr/local/web/.acl` 中的指令，除非它们被如下指令禁止：

```
<Directory />
AllowOverride None
</Directory>
```

#### Directory

定义：`<Directory directory> ... </Directory>`

`<Directory>` 和 `</Directory>` 包含一组只对指定目录及其子目录有效的指令。

`directory` 可以是某个目录的全名，也可以是包含通配符的字符串。例如

```
<Directory /usr/local/etc/httpd/htdocs>
Options Indexes FollowSymLinks
AllowOverride None
order allow,deny
allow from all
</Directory>

<Directory /usr/local/etc/httpd/cgi-bin>
AllowOverride None
Options None
</Directory>
```

#### Options

定义：`Options [+|-]option [+|-]option ...`

控制对一个特定目录哪些服务器特性有效。option 可为 None, All, Indexes, ExecCGI, FollowSymLinks...。Indexes 表示如果某 URL 映射为某目录,而在此目录中没有 DirectoryIndex 指定的文件,则返回此目录的文件列表。ExecCGI 表示允许执行 CGI 文件。FollowSymLinks 表示服务器将遵循此目录中的符号连接。参见下面 <Directory> 指令。

#### AllowOverride

定义: AllowOverride override override ...

缺省: AllowOverride All

当服务器发现 .htaccess 文件,它必须知道文件中的哪些指令可以覆盖以前的限制信息。当设为 None 时服务器将不读 .htaccess 文件,设为 All 时服务器将允许所有的指令。可设为:

AuthConfig 允许使用注册指令

FileInfo 允许使用文件类型控制指令

Indexes 允许使用目录索引指令

Limit 允许使用节点限制指令

Options 允许使用指定目录特性指令

请参考 Apache 的在线手册,查看每一类指令的范围。

#### allow

定义: allow from host host ...

影响哪些节点可访问某特定目录。host 可以为:

all 允许所有节点访问

(部分)域名 以这个字符串结尾的节点可以访问

完整的 IP 地址 允许访问节点的 IP 地址

部分 IP 地址 IP 地址的前三个字节,允许相应子网的节点访问

#### deny

定义: deny from host host ...

影响哪些节点可访问某特定目录。host 可以为:

all 禁止所有节点访问

(部分)域名 禁止以这个字符串结尾的节点访问

完整的 IP 地址 禁止访问节点的 IP 地址

部分 IP 地址 IP 地址的前三个字节,禁止相应子网的节点访问

#### order

定义: order ordering

缺省: order deny, allow

定义 deny 和 allow 指令被鉴定的顺序。ordering 可以为

deny, allow deny 指令在 allow 指令前被鉴定(初始状态是 OK)

allow, deny allow 指令在 deny 指令前被鉴定(初始状态是 FORBIDDEN)

mutual-failure 只有在 allow 的列表上但不在 deny 的列表上的节点可访问(与初始状态无关)

例如:

```
order deny,allow
deny from all
allow from .nasa.uiuc.edu
```

允许在 .nasa.uiuc.edu 域的节点访问,禁止其他所有节点访问。

## 7.3 代理服务器的设置

### 7.3.1 代理服务器

代理服务器(Proxy Server),一般是指在防火墙上的应用程序,是把防火墙内的 Intranet和防火墙外广阔的 Internet 连接起来的管道。它除了可以监控网络和记录传输信息外,还可以提供企业级的文件缓存、复制和地址过滤服务。任何应用程序如需要与外界通讯必须首先与代理服务器交谈,以代理服务器为中介出入防火墙。

目前常用的一种代理服务器是带文件缓存(cache)的 HTTP 代理服务器。当一个客户程序被设置成使用局域网上的 HTTP 代理服务器时,它的 HTTP(或 FTP)请求将被发往代理服务器,而不是文件实际上的源地址。当代理服务器接到这一请求后,首先在自己的缓存区里寻找所要求的文件,如果缓存区里没有,则从文件的源地址处下载(并存入缓存区),再把结果发回客户程序。以后同样的请求将从缓存区得到解答,文件经局域网从代理服务器传到客户程序。每个文件只从广域网下载一次,避免了同一文件的重复传输,节省了网络带宽和客户程序下载文件的时间。缓存区会及时清除过期的文件,并避免缓存那些实际上是由程序产生的应答(如 CGI 程序的输出,对每次请求其结果都可能不同)。

一些常用的 HTTP 服务器,如 CERN HTTPD、Apache 可以被设置成带缓存的代理服务器。另有一些是被专门设计成代理服务器或与现存的 HTTP 服务器共同使用,可能支持更多的协议类型,并附加有其他功能,如自动翻译等,常用的有 DeleGate、Squid 等。

使用 HTTP 代理服务器需要:① 安装、设置、运行 HTTP 代理服务器;② 设置客户程序,一般是 WWW 浏览器。

下面以 Apache HTTP 服务器为例,说明如何安装、设置、运行 HTTP 代理服务器,以 Netscape Communicator 为例,说明如何设置浏览器使用代理服务器。

### 7.3.2 设置 Apache HTTP 服务器为代理服务器

Apache HTTP 服务器的详细介绍和安装指南见上一节,这里只着重介绍与代理服务器相关的内容。

Apache HTTP 服务器只在 1.1 或更高的版本中含有代理模块,它实现了对 FTP、(SSL 的)CONNECT、HTTP/0.9 和 HTTP/1.0 等协议请求的代理/缓存功能。在缺省情况下,代理模块没有被编译进服务器的可执行文件 httpd,如想使用 Apache 的代理功能,必须在编译 httpd 前改变有关设置,加进代理模块。下面就介绍如何编译有代理功能的 Apache HTTP 服务器。

首先,从“Apache HTTP Server Project”的任一镜像节点下载最新版本(写作本书时为

1.3.6)的 Apache HTTP 服务器的源程序包,其文件名应类似 `apache_1.3.6.tar.gz` 或 `apache_1.3.6.tar.Z`。把源程序包打开在某个工作目录下,应产生 `apache_1.3.6/` 目录。在运行“configure”程序时要加上“`--enable-module=proxy`”选项:

```
# cd apache_1.3.6
# ./configure --enable-module=proxy
# make
# make install
```

其次,对 Apache HTTP 服务器 `conf/` 目录下的服务器配置文件 `httpd.conf` 作适当的改动,增加与代理/缓存有关的指令,使 HTTP 服务器工作在带缓存的代理服务器状态。

`httpd.conf` 中与代理/缓存有关的 Apache 指令(directive)有如下几条:

```
ProxyRequests
ProxyRemote
ProxyPass
ProxyBlock
CacheRoot
CacheSize
CacheMaxExpire
CacheDefaultExpire
CacheLastModifiedFactor
CacheGcInterval
CacheDirLevels
CacheDirLength
NoCache
```

下面逐条加以介绍。

#### ProxyRequests

定义: `ProxyRequests on/off`

缺省: `ProxyRequests Off`

允许或禁止使用 Apache 的代理功能。注意设置 `ProxyRequests` 为“off”并没有使 `ProxyPass` 指令失效。

#### ProxyRemote

定义: `ProxyRemote <match> <remote-server>`

定义此代理服务器的远程代理。`<match>` 可以是远程服务器支持的 URL 形式,表示对于此 URL,使用远程代理服务器;可以是 URL 的一部分,只要完整的 URL 中含有这一部分,就使用远程服务器;也可以是“\*”,表示对所有的请求都使用远程服务器。`<remote-server>` 是远程服务器 URL 的一部分,定义为

```
<remote-server> = <protocol> ://<hostname>[:port]
```

`<protocol>` 是用来与远程服务器通讯的协议;Apache 代理模块只支持“http”。下面是几个例子:

```
ProxyRemote http://goodguys.com/ http://mirrorguys.com:8000
ProxyRemote * http://cleversite.com
```

ProxyRemote ftp http://ftpproxy.mydomain.com:8080

在最后一个例子中,代理服务器将把所有的 FTP 请求封装成一个 HTTP 代理请求,并把它传递给另一个可以处理它的远程代理服务器。

#### ProxyPass

定义:ProxyPass <path> <url>

这条指令允许远程服务器被映射到本地服务器中;本地服务器这时不是作为通常意义的代理,而是看起来更像远程服务器的镜像。<path>是本地的一个虚拟路径;<url>是远程服务器 URL 的一部分。假设本地服务器的地址为 http://wibble.org,那么

ProxyPass /mirror/foo http://foo.com

将把一个本地请求 http://wibble.org/mirror/foo/bar 隐含地转换成一个到 http://foo.com/bar 的代理请求。

#### ProxyBlock

定义:ProxyBlock <word/host/domain list>

这条指令给出了一个由空格分开的词语、节点和域名的列表。含有列表中词语、节点或域名的 HTTP、HTTPS 和 FTP 文件请求被代理服务器阻断。代理模块在启动时将试图决定列表中可能的节点名的 IP 地址,并把它缓存起来以备将来做符合检验。例如:

ProxyBlock joes\_garage.com some\_host.co.uk rocky.wotsamattau.edu

'rocky.wotsamattau.edu' 的 IP 地址如出现也符合阻断条件。注意 'wotsamattau' 已经足够符合 'wotsamattau.edu'。而:

ProxyBlock \*

将阻断到所有节点的连接。

#### CacheRoot

定义:CacheRoot <directory>

设置存放缓存文件的目录;注意这个目录必须是 httpd 服务器可写的。

#### CacheSize

定义:CacheSize <size>

缺省:CacheSize 5

设置缓存区大小,以 KB (1024 bytes) 为单位。尽管实际占用的空间可能增长而超过这个上限,但垃圾收集机制将删除文件直到占用的空间等于或小于这个上限。

#### CacheGcInterval

定义:CacheGcInterval <time>

每隔 <time> 小时检查缓存区,如果占用的空间已经超过 CacheSize 设置的上限就删除文件。

#### CacheMaxExpire

定义:CacheMaxExpire <time>

缺省:CacheMaxExpire 24

在没有检查原始服务器之前被缓存的 HTTP 文件将最多保存 <time> 小时,因此文件最多可能过期 <time> 小时。即使随文件提供了一个失效期,这个限制也将被执行。

### CacheLastModifiedFactor

定义:CacheLastModifiedFactor < factor >

缺省:CacheLastModifiedFactor 0.1

如果原始 HTTP 服务器没有提供文件的失效期,依照如下公式估计失效期:

失效期 = 离最新一次修改的时间 \* < factor >

例如,如果文件最新一次修改是 10 小时前,且 < factor > 是 0.1,则失效期将被设为  $10 * 0.1 = 1$  小时。如果失效期比 CacheMaxExpire 设置的时间长,则后者优先。

### CacheDirLevels

定义:CacheDirLevels < levels >

缺省:CacheDirLevels 3

设置缓存区中子目录的层数。被缓存的数据将在 CacheRoot 下存储所设置的子目录层数。

### CacheDirLength

定义:CacheDirLength < length >

缺省:CacheDirLength 1

设置代理缓存子目录名的字母数。

### CacheDefaultExpire

定义:CacheDefaultExpire < time >

缺省:CacheDefaultExpire 1

如果文件是通过一个不支持失效期的协议获取的,则使用 < time > 小时作为失效期。CacheMaxExpire 不能覆盖这个设置。

### NoCache

定义:NoCache < word/host/domain list >

给出了一个由空格分开的词语、节点和域名的列表。符合列表中词语、节点或域名的 HTTP 和不用口令的 FTP 文件不被代理服务器缓存。代理模块在启动时将试图决定列表中可能的节点名的 IP 地址,并把它们缓存起来以备将来做符合检验。例如:

```
NoCache joes_garage.com some_host.co.uk bullwinkle.wotsamattau.edu
```

'bullwinkle.wotsamattau.edu' 的 IP 地址如出现也符合不缓存条件。注意 'wotsamattau' 已经足够符合 'wotsamattau.edu'。而:

```
NoCache *
```

将彻底禁止缓存。

下面就一些一般性设置问题做些说明。

#### (1) 控制使用代理

可以通过通常的 <Directory> 控制块来控制对代理服务器的使用。例如:

```
<Directory proxy: * >
```

```
<Limit GET PUT POST DELETE CONNECT OPTIONS>
```

```
order deny,allow
```

```
deny from [不允许使用此代理服务器的节点名或 IP 地址]
```

allow from [允许使用此代理服务器的节点名或 IP 地址]

</Limit>

</Directory>

<Files> 块也可以用来控制对代理服务器的使用。

(2) 为什么文件类型 xxx 不能用 FTP 下载

很可能是因为 xxx 类型的文件在代理服务器的 mime.types 配置文件中没有被定义为 application/octet-stream, 可以加上这么一句:

```
application/octet-stream    bin dms lha lzh exe class tgz tar
```

(3) 为什么 Apache 使用代理模块时启动较慢

如果使用了 ProxyBlock 或 NoCache 指令, 服务器在启动时将查找节点名的 IP 地址, 并把它们缓存起来以备将来做符合检验。查找节点名的 IP 地址将需要几秒钟或更长的时间。

设置好 httpd.conf 文件后, 在启动代理服务器之前一般应产生 CacheRoot 指定的目录。注意这个目录的 owner 和 group 应该与 httpd.conf 中指定的服务器运行时的 owner 和 group 相同。

现在就可以按通常启动 httpd 的方法来启动代理服务器了。

### 7.3.3 设置浏览器使用代理服务器

几乎所有常用的 WWW 浏览器都可以设置成使用代理服务器。

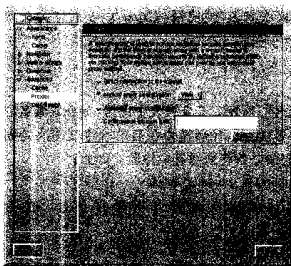


图 7.2 在 Netscape Communicator 中  
设置代理服务器

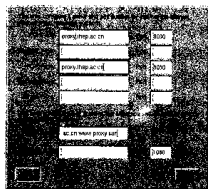


图 7.3 手工设置代理服务器

设置 Netscape Communicator 使用代理服务器的方法非常简单。运行 netscape, 选择菜单项 Edit → Preferences..., netscape 将弹出 Preferences 窗口; 在左侧 Category 中, 选择 Advanced → Proxies; 在 Proxies 的三个选项中, 缺省是“Direct connection to the internet”, 即不使用代理服务器; 若要使用代理服务器, 一般应选“Manual proxy configuration”, 并按右边的“View...”, 则“View Manual Proxy Configuration”窗口弹出。假设代理服务器节点名是 proxy.ihep.ac.cn, 端口号是 8080, 则在“FTP Proxy:”项中填

“proxy.ihep.ac.cn”,右侧“Port:”项中填“8080”;在“HTTP Proxy:”项中填“proxy.ihep.ac.cn”,右侧“Port:”项中填“8080”。可能有一些域或节点不需要使用代理去访问(如局域网的节点),把这些域名或节点名填在“No Proxy for:”项中,Netscape 将不通过代理服务器而直接访问它们。当然,也可以对不同的协议使用不同的代理服务器。

## 7.4 FTP 服务器的设置和维护

### 7.4.1 wu-ftp 和匿名 FTP 服务的配置

现在最常用的免费 FTP 服务器软件是 wu-ftp(Washington University FTP Server)。wu-ftp 目前的版本是 2.4.2,软件包文件名是 wu-ftp-2.4.2.tar.Z,可从全世界众多的 FTP 节点取来。

Linux 的发行套件中有 wu-ftp 的某一版本,所以升级到新版本的安装步骤很简单:

```
lark # tar xzvf wu-ftp-2.4.2.tar.Z
lark # cd wu-ftp-2.4.2
lark # ./build lnx          # 编译 Linux 版本的可执行文件
lark # make install
```

然后编辑/etc/inetd.conf 指向新的 ftpd。通常无须做这一步,因为安装过程会自动用新的 ftpd 覆盖旧的版本。最后重新启动 inetd。

用户可以将自己的 FTP 服务器设置为可用 anonymous 或 ftp 用户连接的匿名 FTP 服务器。

在设置匿名 FTP 服务的过程中,要注意系统安全的要求,以免以后有人利用匿名 FTP 对系统进行侵犯。

首先,在系统上建立一个新的用户 ftp 和一个新的用户组 ftp。用户组 ftp 的 GID 应该用一个远离其他用户组的数字,如 30000。用户 ftp 的 UID 应该用一个远离实际用户的数字,例如 30000。ftp 用户应属于 ftp 组,且 ftp 组应除 ftp 外不含其他用户。为 ftp 用户选定一个方便管理的主目录,如/ftp;匿名 FTP 服务的所有文件都将存放在这个目录下。ftp 用户的 shell 应该是一个不存在的 shell,如/nosuchshell 等。这样我们不必用 useradd 命令,而可以直接在/etc/passwd 文件下增加一行:

```
ftp:x:30000:30000:Anonymous FTP:/ftp:/nosuchshell
```

在/etc/group 文件里增加一行:

```
ftp::30000:
```

在/etc/shadow 文件中增加一行:

```
ftp:*::::
```

为便于管理及安全起见,我们可以用通常的方法建立一个新的 FTP 管理员帐号,如 ftpadmin。以后对 FTP 服务器中向外发布的文件(在~ftp/pub 中)的管理应尽量用此帐号完成。

现在我们建立匿名 FTP 目录结构。我们要建立如下目录:

```
~ftp
```

这个目录的属主应当是 root,并且是任何人不可写的(0555)。

~ftp/bin

这个目录存放 FTP 服务器要用到的一些命令。其属主应当是 root,并且是任何人不可写的(0111)。拷贝/bin(或/usr/bin)下的ls,tar,gzip,compress等程序到这个目录下,这几个文件的保护应该是0111。

~ftp/lib

这个目录的属主应当是 root,并且是任何人不可写的(0111)。拷贝到~ftp/bin目录下的几个程序可能都是动态连接的,所以它们运行时需要共享库,它们用到的共享库放在这个目录下。如用“ldd /bin/ls”命令可能会得到下面的输出:

```
lark% ldd /bin/ls
libc.so.6 => /lib/libc.so.6 (0x40003000)
/lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x00000000)
```

所以必须拷贝/lib/libc.so.6和/lib/ld-linux.so.2到~ftp/lib目录下。这些库文件应该是可读可执行的(0555)。

~ftp/etc

拷贝/etc/passwd和/etc/group文件到这个目录下,并从中删除任何个人用户和个人用户组的信息,但应保留ftp用户和ftp组。这两个文件应该是可读的(0444)。

~ftp/pub

这个目录中存放匿名 FTP 服务器上要发布的所有文件及目录结构。这个目录的属主和组应该是 FTP 服务器管理员(如ftpadmin)的属主和组,其保护应该是0755,即 FTP 管理员可全权操作。

~ftp/incoming

这个目录中存放由 FTP 服务器的使用者上载到 FTP 服务器上的文件。很多人可能希望把自己编制软件或收集的好软件放到一个公共场所供更多人使用。匿名 FTP 服务器可以配置成使用者可以在~ftp/incoming目录下放些文件。当然 FTP 服务器管理员应当经常整理这个目录,把有用的软件移到~ftp/pub目录中适当的子目录下,并删除某些没用的或不适当的东西,特别要留意其中是否有可疑的程序(如危害系统安全的“黑客”程序等)。这个目录的属主和组都应当是ftp,其保护可以是0755或0775。

创建完匿名 FTP 服务需要的目录结构后,我们必须配置 FTP 服务器的行为。在wu-ftpd软件包的doc/examples目录下有几个配置文件的例子:ftpaccess,ftpusers,ftpconversions,ftphosts和ftpgroups,它们应该被放置在/etc目录下,但用户可能需要对其中的一些加以改动。

/etc/ftpconversions

这个文件定义文件的“飞行”转换规则。所谓“飞行”转换指文件或目录在(使用RETRIEVE命令)传递过程中进行压缩(解压缩)和打包。我们可以不加改动地把doc/examples/ftpconversions拷贝到/etc下。它的内容如下:

```
:.Z : : /bin/compress -d -c %s;T_REG|T_ASCII;O_UNCOMPRESS;UNCOMPRESS
: : : .Z:/bin/compress -c %s;T_REG;O_COMPRESS;COMPRESS
:.gz : : /bin/gzip -cd %s;T_REG|T_ASCII;O_UNCOMPRESS;GUNZIP
```

```

:      : .gz:/bin/gzip -9 -c %s;T_REG;O_COMPRESS;GZIP
:      : .tar:/bin/tar -c -f - %s;T_REG|T_DIR;O_TAR;TAR
:      : .tar.Z:/bin/tar -u -Z -f - %s;T_REG|T_DIR;O_COMPRESS|O_TAR;TAR +
COMPRESS
:      : .tar.gz:/bin/tar -c -z -f - %s;T_REG|T_DIR;O_COMPRESS|O_TAR;TAR + GZIP

```

以上几句的意思可以列表如下:

标 准 表

| 真实文件名          | 结果文件名               | 处理措施                     |
|----------------|---------------------|--------------------------|
| <filename> .Z  | <filename>          | 传递前用 compress -d 命令解压缩   |
| <filename> .gz | <filename>          | 传递前用 gzip -d 命令解压缩       |
| <filename>     | <filename> .Z       | 传递前用 compress 命令压缩       |
| <filename>     | <filename> .gz      | 传递前用 gzip -9 命令压缩        |
| <filename>     | <filename> .tar     | 传递前用 tar -cf 命令打包        |
| <filename>     | <filename> .tar.Z   | 传递前用 tar -cZ 命令打包并压缩     |
| <filename>     | <filename> .tar.gz  | 传递前用 tar -czf 命令打包并压缩    |
| <directory>    | <directory> .tar.Z  | 传递前用 tar -cZ 命令对目录打包并压缩  |
| <directory>    | <directory> .tar.gz | 传递前用 tar -czf 命令对目录打包并压缩 |

“飞行”转换的主要作用是压缩传递的数据量,减少传递时间;另外可以将一个目录下的所有文件打包后一次传递过来,不用费时耗力地一个个单独传递。

/etc/ftpusers

本地用户如果其用户名出现在 ftpusers 文件中,将不被允许使用此 FTP 服务器,我们应当将 root 及其他系统帐号(如 bin,sys)等加进 ftpusers 文件,并把它拷贝到/etc 下。

/etc/ftpgroups

ftpgroups 文件并不是不允许哪个组的本地用户使用 FTP 服务的意思,它只有在在使用 SITE GROUP 功能时才用到。请参考 wu-ftp 的有关说明文件,一般不必使用 ftpgroups。

/etc/ftphosts

设置允许或拒绝某些特定节点(或域)以某用户名使用 FTP 服务。如下面的 ftphosts 文件设置 FTP 服务器拒绝 202.122.35 子网的所有节点的匿名登录,但单独允许 202.122.35.32 和 202.122.35.33 节点的匿名登录:

```

allow ftp 202.122.35.32 202.122.35.33
deny ftp 202.122.35.*

```

注意:若这两行颠倒顺序(deny 放在前面),则 allow 一行将不起作用。又如

```

allow ftp *.edu.cn *.ihep.ac.cn ftp.ccc.ac.cn
deny ftp *

```

允许中国教育网.edu.cn的所有机器、科研网中 ihep.ac.cn的所有机器及科研网中另一台机器 ftp.ccc.ac.cn 用匿名 FTP 登录,但拒绝所有其他匿名 FTP 登录。

/etc/ftppass

它是 FTP 服务器的主要配置文件。doc/examples/ftppass.heavy 是一个稍加修改

就可适用于大多数 FTP 服务器的例子。下面逐句分析一下这个例子：

```
login:fail 2
```

2 次登录错误后,打印“repeated login failures”信息并退出连接。缺省值是 5。

```
class local real,guest,anonymous * .domain 0.0.0.0
```

```
class remote real,guest,anonymous *
```

定义用户类别。格式为

```
class < class > < typelist > < addrglob > [ < addrglob > ... ]
```

< typelist > 为用户类型,可以为 real(真实用户),guest(属于客人组(用 guestgroup 定义)的真实用户),anonymous(匿名 FTP 用户)。class 语句定义从各 < addrglob > 地址来、用 < typelist > 登录的用户为 < class > 类。如上例中,本机和来自 .domain 网的连接被归入 local 组,其他用户被归入 remote 组。

```
limit local 20 Any /etc/mmsgs/msg.toomany
```

```
limit remote 100 SaSu!Any1800-0600 /etc/mmsgs/msg.toomany
```

```
limit remote 60 Any /etc/mmsgs/msg.toomany
```

限制各类用户在某段时间的数量,如果超过限制,打印文件/etc/mmsgs/msg.toomany 并拒绝登录。如上句限制 local 组用户在任何时间不能超过 20 人,remote 组用户在周六、周日的 18:00 到 06:00 不能超过 100 人,平常不能超过 60 人。

```
readme README * login
```

```
readme README * cwd = *
```

当用户登录时或第一次进入某目录时,FTP 服务器提示用户阅读的文件。如:

```
ftp> cd slackware
```

```
250-Please read the file README34.TXT
```

```
250- it was last modified on Mon Nov 3 18:48:00 1997 - 116 days ago
```

```
250 CWD command successful.
```

```
ftp>
```

```
message /welcome.msg login
```

```
message .message cwd = *
```

当用户登录时或第一次进入某目录时,FTP 服务器在屏幕上打印的信息文件。

```
compress yes local remote
```

```
tar yes local remote
```

对某类用户是否支持适用 compress 或 tar 功能。

```
# allow use of private file for SITE GROUP and SITE GPASS?
```

```
private yes
```

是否允许通过 SITE GROUP 和 SITE GPASS 命令适用秘密文件。

```
# passwd-check < none|trivial|rfc822 > [ < enforce|warn > ]
```

```
passwd-check rfc822 warn
```

口令检验方式。none 表示不进行口令检查;trivial 表示口令必须含“@”;rfc822 表示

口令必须是满足 rfc822 规定的地址。当口令不符合要求时, warn 表示警告用户但允许登录; enforce 表示警告用户且使用户退出。

```
log commands real
log transfers anonymous,real inbound,outbound
```

log commands <typelist> 记录 <typelist> 类型(可以为 anonymous, guest, real) 用户使用的命令。log transfers <typelist> <directions> 记录 <typelist> 类型的用户做的 <directions> (可以为 inbound 传进服务器、outbound 传出服务器) 方向的文件传输。

```
shutdown /etc/shutmsg
```

如果 shutdown <path> 指定的文件存在, 服务器将定期检查 <path> 文件以查看服务器是否预定关闭。<path> 文件的格式为

```
<year> <month> <day> <hour> <minute> <deny_offset> <disc_offset>
<text>
```

<deny\_offset> 和 <disc\_offset> 的意思是在服务器关闭前多长时间新的连接将被拒绝和现存的连接将被中断。<text> 是对拒绝连接的用户的一段信息。如

```
1998 01 27 22 05 0010 0005
System shutdown at %s
```

表示 1998 年 2 月 27 日 22:05 关闭 FTP 服务器, 10 分钟前拒绝新的连接, 5 分钟前中断正在进行的 FTP 连接。外部程序 ftpshut 可用来产生 <path> 文件。如上面的文件可用下列命令产生:

```
lark # ftpshut 2205
```

```
# all the following default to "yes" for everybody
delete          no      guest,anonymous      # delete permission?
overwrite       no      guest,anonymous      # overwrite permission?
rename          no      guest,anonymous      # rename permission?
chmod           no      anonymous              # chmod permission?
umask           no      anonymous              # umask permission?
```

对哪一类型的用户哪种命令禁止使用。

```
# specify the upload directory information
upload /var/ftp * no
upload /var/ftp /incoming yes root daemon 0600 dirs
upload /var/ftp /bin no
upload /var/ftp /etc no
```

哪些目录可用以上载文件。上例中 ~ftp(/var/ftp) 目录下/incoming 目录可用来上载文件, 上载的文件的属主是 root, 组别是 daemon, 保护是 0600; dirs 表示可在/incoming 目录下创建子目录, nodirs 表示不可创建子目录。下例:

```
upload /var/ftp /incoming * yes ftpadmin ftp 0644 dirs
```

表示可在/incoming 目录下上载文件、创建子目录, 并可在子目录中上载新文件等等。

```
# directory aliases... [note, the ";" is not required]
```

```
alias inc: /incoming
```

给/incoming目录创建别名inc:,用户在任何时候只要用命令“cd inc:”就可到达/incoming目录。只对cd命令有效。

```
# cdpath
cdpath /incoming
cdpath /pub
cdpath /
```

与csh中的cdpath意义相同。

```
# path-filter...
path-filter anonymous /etc/pathmsg ^[-A-Za-z0-9_\.] * $ ^\ . -
path-filter guest /etc/pathmsg ^[-A-Za-z0-9_\.] * $ ^\ . -
```

格式是

```
path-filter <typelist> <msg> <allowed_charset> | <disallowed_reg-exp> ...}
```

定义所有<typelist>类型用户上传的文件的名字必须由允许的字符<allowed\_charset>组成,不允许有<disallowed\_reg-exp>等规定的表达式,否则打印信息<msg>并拒绝接收。如上例中,匿名用户上载的文件名称必须由A-Z,a-z,0-9和“\_”、“.”等字符组成,禁止有“.”和“-”开头的文件名,否则拒绝接收,并打印“ftp/etc/pathmsg”文件。

```
# specify which group of users will be treated as "guests".
guestgroup ftponly
```

定义哪个组的真实用户属于客人组。

```
email user@hostname
```

FTP服务器管理员的电子邮件地址。

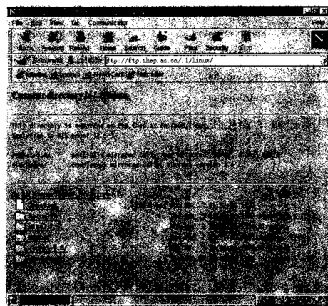


图 7.4 用 Netscape 浏览 FTP 目录

(目录列表的上面显示的文本即是 .message 文件的内容)

### 7.4.2 FTP 镜像节点的维护

Linux 的各种发行套件和应用软件总是在不断的更新过程中。有些用户希望随时拥有最新版本的软件,一般的方法是经常检查存放自己感兴趣的软件的 FTP 节点,如果有新的版本发布,就把它下载到本地计算机上。这种方法在关注的软件不太多、软件更新速度不太快、软件套件的文件数目不太多时还可以应付,但如果要跟踪 Linux 的各种发行套件和 GNU 软件集这样软件数目很大、文件数目很多、且更新速度很快的软件集的变化,用上述的方法就非常耗时耗力,且难免出现错误。这时我们可以对存放某常用软件的 FTP 节点的相应目录在本地 FTP 服务器上建立镜像目录,用一些 FTP 镜像软件在后台定期对远程节点进行检查,如果发现远程目录下有文件已更新,立即对本地目录下的相应文件进行更新,但远程目录中未作更新的文件本地节点维持不变,这就避免了不必要的重复传输。

目前有几种常用的 FTP 镜像软件,如 mirror,ftpmirror 等。FTP 站点的管理员可以利用 ftpmirror 或 mirror 维护自己的 FTP 站点上多种软件和资料的镜像目录。

下面就介绍如何利用 ftpmirror 软件 FTP 镜像节点。

#### 1. ftpmirror 的安装

ftpmirror 可以用来通过 FTP 方式拷贝目录树(即所谓的 mirror)。ftpmirror 是 perl 脚本文件,它需要 perl-5.003 或以后的版本。ftpmirror 的最新版本可从

<http://ftp.intec.co.jp/pub/utis/>

取来。目前的版本是 ftpmirror-1.21.tar.gz。ftpmirror 的安装步骤如下:

- (1) 将软件包打开在 /usr/local/lib 下。

```
lark# cd /usr/local/lib
lark# tar zxvf ftpmirror-1.21.tar.gz
```

如果用户将软件包打开在别的地方,必须修改 ftpmirror, mirror.sh, mkdirinfo 和 param.pl 文件中的“/usr/local/lib”字符串。

- (2) 如果有必要,修改 ftpmirror 和 mkdirinfo 文件的第一行,使之指向 perl 命令。
- (3) 到 config/目录下创建缺省配置文件。

```
lark# cd /usr/local/lib/ftpmirror/config
lark# cp default-sample default
lark# vi default
```

需要改动的可能有 ftp-pass 和 home-directory。

- (4) 根据需要创建软件包的配置文件(参考本节下面的内容)。

```
lark# cp GNU-sample.gnu
lark# vi gnu
```

最简单的配置文件包含三行记录:ftp-server, remote-directory 和 local-directory。例如 gnu 文件:

```
ftp-server = ftp.u-aizu.ac.jp
```

```
remote-directory = /pub/gnu/prop
local-directory = ~/pub/gnu
```

(5) 用 `ftpmirror` 命令后加软件包的名称。`ftpmirror` 寻找当前目录或 `config/` 目录下与软件包名称相同的配置文件。

```
lark # /usr/local/lib/ftpmirror/ftpmirror.gnu
```

(6) 文件传递的任何信息输出在 `stdout`。

## 2. `ftpmirror` 的参数定义

`ftpmirror` 的参数分四类:系统和行为、FTP 进程、软件包、镜像拷贝。`ftpmirror` 的各种参数的缺省值定义在文件 `param.pl` 中。

用户可以在缺省配置文件 `default`, 软件包配置文件和命令行上配置 `ftpmirror` 的参数。其中命令行参数优先级最高, 其次是软件包配置文件, 再次是缺省配置文件 `default`, 最低是参数的缺省值文件 `param.pl`。

`ftpmirror` 的系统和行为类参数有:

`local-hostname` (缺省: "hostname" 命令的输出) 本机节点名。

`debug` (缺省: 0)

向后兼容。`debug = 1` 等同于 `log-priority = 6`。

`log-priority` (缺省: 5)

信息输出级别。目前支持的选择(高级别的输出含低级别的输出)有:

- 0: EMERG, 紧急输出, 目前是无输出
- 4: WARNING, 警告输出
- 5: NOTICE, 输出文件传输和删除(unlink)信息
- 6: INFO, 输出文件比较信息
- 7: DEBUG, 更详细的输出

`test-mode` (缺省: 0)

如果此值为 1, `ftpmirror` 打印要传输或删除的文件名, 但实际什么也不做。

`ftpmirror` 的 FTP 进程类参数有:

`ftp-server` (缺省: 无)

要连接的 FTP 服务器, 对每个软件包都应指定此参数。

`ftp-gateway` (缺省: 无)

如果使用 FTP 代理网关(如 TIS Internet Firewall Toolkit), 可设此参数。

`ftp-user` (缺省: anonymous)

FTP 进程的缺省登录名。缺省是使用匿名 FTP 登录。

`ftp-pass` (缺省: ftp-admin@'hostname')

FTP 进程的口令。

`ftp-group` (缺省: 无)

如果想在登录后改变 GID, 设置组的名字。必须与 `ftp-gpass` 同时使用。

`ftp-gpass` (缺省: 无)

如果想在登录后改变 GID, 设置改变组别要用到的口令。必须与 `ftp-group` 同时使

用。

`ftp-passive` (缺省:0)

如果被要求使用被动连接(FTP的PASV命令),设置此值为1。它可以帮助ftpmirror通过一个SYN/ACK类的防火墙。

`ftp-idle` (缺省:0)

如果此参数不为0,ftpmirror将设置FTP进程的空闲时间计时器为此值。FTP服务器必须支持SITE IDLE命令。

`ftp-max-idle` (缺省:0)

如果此参数不为0,ftpmirror将设置FTP进程的空闲时间计时器为最大值。FTP服务器必须支持SITE IDLE命令。

`ftp-list-method` (缺省:stat)

可决定用STAT还是LIST命令获取FTP服务器的目录内容。STAT命令通过控制连接返回服务器的目录内容,它比LIST稍快。但某些类型的FTP服务器不能返回ftpmirror所需要的目录内容,这种情况下必须设置`ftp-list-method=list`。

ftpmirror的软件包类参数有:

`package` (缺省:无)

指定软件包名。当要在一个配置文件里配置好几个软件包时需要此参数。在package语句后定义的任何参数属于package语句定义的软件包。

`server` (缺省:无)

用来定义特定FTP服务器的参数。这个参数后的任何参数属于server语句定义的特定服务器,直到碰到package语句或另一个server语句。可定义`ftp-gateway`,`ftp-passive`,`remote-timezone`等参数作为服务器参数。如:

```
# 在此例的缺省设置里,绝大多数FTP服务器需要代理(proxy)
ftp-gateway = proxy.intec.co.jp
# 局域网
server = ftp.intec.co.jp
undef ftp-gateway
# 使用另外一个代理服务网关,设置时差
server = ftp.triton.ad.jp
ftp-gateway = proxy2.intec.co.jp
remote-timezone = +0900
# 其他软件包
package = ...
```

`load-package`

如果对某个软件包定义了这个参数,则当前的软件包从load-package指定的软件包配置中继承所有参数。注意,load-package指定的软件包必须在前面定义过。参见下面的参数`alternate-package`。

`alternate-package`

当对一个软件包FTP登录进程失败后,用另一个替代的镜像进程。用此参数设置替代进程的替代软件包名。如:

```
package = FreeBSD
ftp-server = ftp.freebsd.org
remote-directory = /pub/FreeBSD
local-directory = ^/pub/FreeBSD
lsR-file = /pub/FreeBSD/ls-lR.gz
alternate-package = FreeBSD-triton
package = FreeBSD-triton
load-package = FreeBSD
ftp-server = ftp.triton.ad.jp
```

在软件包 FreeBSD 的配置中指定替代进程软件包名是 FreeBSD-triton。如果 FTP 登录 ftp.freebsd.org 失败,则转向软件包 FreeBSD-triton。这个软件包的配置参数与 FreeBSD 软件包一样,但指定了另一个 FTP 服务器 ftp.triton.ad.jp。

ftpmirror 的镜像拷贝类参数有:

unlink (缺省:yes)

如果想删除(unlink)本地存在但在远程服务器上已不存在的文件,设置此参数为“yes”。如设置此参数为“rename”,则本地的这类文件被改名为原文件名后加“-”。

unlink-limit (缺省:0)

不删除(unlink)体积大于 unlink-limit 指定值(如果不为 0)的文件或目录。可以防止因远程 FTP 服务器的问题意外删除本地大文件或目录。如 unlink-limit = 30M 表示大于 30M 的文件或目录将不删除,而 unlink-limit = 100 则表示不删除含有 100 个以上文件的目录。

compare-stat (缺省:1)

在比较程序里,ftpmirror 可使用 STAT/LIST 或 SIZE/MDTM 的结果。如果此值设为 1,则 ftpmirror 使用 STAT/LIST 的结果来比较本地/远程文件。请用 list-method 设置使用 STAT 和 LIST 中哪一个。

check-mtime (缺省:1)

ftpmirror 缺省要比较本地/远程文件的修改时间,如果时间不一致,就重新传递文件。如果此值设为 0,ftpmirror 在本地/远程文件的修改时间不一致时,如果文件大小一致,只依照远程文件的时间修改本地文件的时间,并不重传。

list-method (缺省:STAT)

缺省时,ftpmirror 发送 STAT 命令去 FTP 服务器取得某目录的内容。但有时使用 STAT 存在问题,这时可设 list-method 为“LIST”。

remote-timezone (缺省:无)

指定远程 FTP 服务器的时区。因为 MDTM 返回的时间是 GMT,此参数只对 compare-stat = 1 时有效。在没有指定 remote-timezone 时,ftpmirror 将尝试计算 FTP 服务器的时区。例如,如果远程 FTP 服务器在日本,则设 remote-timezone = +0900。不支持夏时制。

lock-directory (缺省:/tmp)

ftpmirror 使用锁定机制以避免同一时间对同一软件包存在多个镜像拷贝进程。此参数定义存放锁定文件的目录。

temp-directory (缺省: /tmp)

存放临时文件(如 ls-lR, dirinfo)的目录。

home-directory (缺省: ~ftp)

为简便起见,我们可以在 local-directory 参数中使用波浪符(~),波浪符将替换为这里定义的 home-directory。

remote-directory (缺省:无)

将要镜像拷贝的远程 FTP 服务器的目录。镜像拷贝将对此目录向下递规进行。所有软件包配置必须指定 remote-directory。

local-directory (缺省:无)

本地节点上与 remote-directory 匹配的目录。所有软件包配置必须指定 local-directory。下面的例子说明镜像拷贝远程 FTP 服务器上/pub/FreeBSD 目录到本地的/ftp/pub/FreeBSD 目录下:

```
remote-directory = /pub/FreeBSD
local-directory = /ftp/pub/FreeBSD
```

如果定义了 home-directory(或 ftp 用户的主目录存在),我们可以这样写:

```
remote-directory = /pub/FreeBSD
local-directory = ~/pub/FreeBSD
```

ls-lR-file (缺省:无)

ftpmirror 可使用 FTP 服务器上的 ls-lR(或类似的)文件,而不去用命令 STAT/LIST 扫描目录。本参数指定远程服务器上的 ls-lR 文件名。如果 ls-lR-file 是 .gz 或 .Z 形式的,将自动使用 gunzip 或 uncompress 命令去解压缩。例如,用 ls-lR.gz 文件去镜像拷贝 ftp.freebsd.org:/pub/FreeBSD,可配置如下:

```
ftp-server = ftp.freebsd.org
remote-directory = /pub/FreeBSD
local-directory = ~/pub/FreeBSD
ls-lR-file = /pub/FreeBSD/ls-lR.gz
```

ls-lR-map (缺省:无)

当使用 ls-lR-file 时,远程服务器上的 ls-lR-file 文件中的列表(ls-lR 命令的输出)可能用了无效的顶级目录,因为服务器上使用 ls-lR 命令时的目录可能不是当前目录。如 ftp.ij.ad.jp:/pub/GNU/ls-lR.Z 文件中的记录类似 GNU/gcc-2.7.2.tar.gz,但为镜像拷贝这个目录树,“GNU”应当为“.”。这时我们应该定义 ls-lR-map:

```
ls-lR-file = /pub/GNU/ls-lR.Z
ls-lR-map = /~GNU/./
```

transfer-file-regexp (缺省:符合所有文件名)

perl 的规范表达式,ftpmirror 只传输文件名与这个表达式符合的文件。在远程 FTP 服务器上的任何文件名都是用以“./”开头的相对路径(相对于 remote-directory)表示的。如果 remote-directory 是“/pub/FreeBSD”,服务器上文件“/pub/FreeBSD/2.1.5-RELEASE/README.TXT”被表示为“./2.1.5-RELEASE/README.TXT”。例如:

```
transfer-file-regexp + = ! /~ $/
transfer-file-regexp + = ! /\ \ \ .in/
transfer-file-regexp + = ! /\ \ \ .nfs/
```

在这种配置下,ftpmirror 不传递文件名以“~”结尾或以“.in”和“.nfs”开头的文件。又如

```
transfer-file-regexp + = /\ \ /bash-/
transfer-file-regexp + = /\ \ /gcc-/
transfer-file-regexp + = !
```

表示 ftpmirror 只传递 bash-\* 和 gcc-\* 文件,但不传递其他所有文件。上例中最后一行表示缺省是不传递所有文件。

transfer-directory-regexp (缺省:符合所有目录名)

perl 的规范表达式,ftpmirror 只扫描目录名与这个表达式符合的目录。在远程 FTP 服务器上的任何目录的表达式的结尾都有一个“/”。下面的例子表示 ftpmirror 将不去扫描(不传递)目录名是“lost + found”或以“~”结尾的目录:

```
transfer-directory-regexp + = ! /\ \ /lost + found \ \ $/
transfer-directory-regexp + = ! /~ \ \ $/
```

override-file-regexp (缺省:符合所有文件名)

perl 的规范表达式,定义本地目录下可以覆盖的文件。在实际的镜像拷贝进程中,ftpmirror 将传递同时与 transfer-file-regexp 和 override-file-regexp 符合的文件。任何不与 override-file-regexp 符合的本地文件将不被 ftpmirror 修改。这与 transfer-file-regexp 不同,任何不与 transfer-file-regexp 符合的本地文件将被删除。这对把从几个远程 FTP 服务器取来的文件放在同一个本地目录时特别有用。如下例中,ftpmirror 将只传递 bash-\* 文件,而不对其他文件进行修改。

```
transfer-file-regexp + = /\ \ /bash-/
transfer-file-regexp + = !
override-file-regexp + = /\ \ /bash-/
override-file-regexp + = !
```

override-directory-regexp (缺省:符合所有目录名)

利用这个参数,ftpmirror 可以把从几个远程 FTP 服务器取来的文件放在同一个本地目录里。下面的例子显示如何把 FreeBSD-NonUS 软件包传递到 FreeBSD 目录里:

```
package = FreeBSD
ftp-server = ftp.tokyonet.ad.jp
remote-directory = /pub/FreeBSD
local-directory = ~/pub/FreeBSD
override-directory-regexp + = ! /\ \ \ /FreeBSD-NonUS \ \ //
```

```
package = FreeBSD-nonUS
ftp-server = ftp.waseda.ac.jp
remote-directory = /pub/FreeBSD-nonUS
local-directory = ~/pub/FreeBSD/FreeBSD-nonUS
```

override-file-uid (缺省:0)

指定本地文件或目录的 UID,所有符合 override- \* -regexp 的文件或目录的 UID 都被设为这个值。如果 ftpmirror 由非特权用户执行,这个参数将被忽略。

override-file-gid (缺省:0)

同 override-file-uid,但是指定 GID。

**注意:**如果定义了 override-file-uid 或 override-file-gid,所有符合 override- \* -regexp 的文件(即使不是传过来的)将被用户 chown 命令改变属主或组。

override-file-mode (缺省:root -> 0444,其他 -> 0644)

指定本地文件保护。如果定义了此参数,ftpmirror 将对所有符合 override- \* -regexp 的文件使用 chmod 命令。

override-directory-mode (缺省:0755)

同 override-file-mode,但指目录的保护。

default-file-uid (缺省:0)

取消 override-file-gid 的设置,并设本地文件的缺省 UID。ftpmirror 按如下顺序确定文件属主:override-file-uid、文件在 FTP 服务器上的属主、default-file-uid。

default-file-gid (缺省:0)

同 default-file-uid,但指 GID。

default-file-mode (缺省:root -> 0444,其他 -> 0644)

取消 override-file-mode 的设置,并设本地文件的缺省保护。ftpmirror 按如下顺序确定文件保护:override-file-mode、文件在 FTP 服务器上的保护、default-file-mode。注意,由于文件保护模式掩码是 0777,不能对文件设置 setuid/setgid 模式。

default-directory-mode (缺省:0755)

同 default-file-mode,但指目录的保护。

用户可以在命令行给 ftpmirror 指定参数。命令行的任何参数必须以"--"开头,如

```
lark # ftpmirror --test-mode=1 FreeBSD
```

使 ftpmirror 运行在测试模式(不进行实际传输)下。

### 3. 目录信息

ftpmirror-1.2 以后的版本可以产生或使用目录信息(directory information),目录信息存在每个目录下的 .dirinfo 文件中,ftpmirror 在目录比较阶段使用它们。使用目录信息可以使对目录树的镜像拷贝更快、更完整。

例如,一个目录信息文件 .dirinfo 可能包含下面的内容:

```
file = LICENSE
md5checksum = 83cdabcad3c9c3480faa6a75c47c7207
modified = 800142639
size = 4631
owner = root
group = wheel
directory = OLD
```

```
begin-update = 842336105
end-update = 842336105
owner = root
group = wheel
symlink = Welcome.html
linkto = ../index.html
owner = root
group = wheel
directory = archive
begin-update = 842336105
end-update = 842336105
owner = root
group = wheel
file = index
md5checksum = b5a6c559ec486a0f12493ce52121619
modified = 801170780
size = 799
owner = root
group = wheel
....
```

在目录信息文件里,任何文件都必须有一个文件类型记录。目前有三种文件类型:

file = file-name .. 通常的文件

symlink = link-file .. 符号连接

directory = dir-name .. 目录

每个文件类型可以跟随着其他一些参数:

file file-type

size 文件大小  
modified 上次修改日期  
md5checksum MD5 校验和  
owner 属主  
group 组别

symlink file-type

linkto 连接到(某文件或目录)  
owner 属主  
group 组别

directory file-type

begin-update 上次更新的起始时间  
end-update 上次更新的结束时间  
owner 属主  
group 组别

ftpmirror 尝试比较本地和远程服务器上的同一文件的这些参数。特别地,比较目录更新

时间是很有效的,因为这表明在该目录中没有文件更新,从而 ftpmirror 可以跳过对这个目录的扫描。

ftpmirror 的各个配置文件中可以使用如下有关目录信息的参数:

md5-program (缺省:系统上 md5 程序(如果有的话))的路径名)

当使用目录信息时,用来计算 MD5 校验和的程序。

load-local-dirinfo (缺省:0)

如果此参数为 1,ftpmirror 将使用本地目录信息文件 .dirinfo。

store-local-dirinfo (缺省:0)

如果此参数为 1,则在每个镜像进程后将在本地目录下产生 .dirinfo 文件。

load-remote-dirinfo (缺省:0)

如果此参数为 1,ftpmirror 将使用储存在远程 FTP 服务器上的目录信息文件 .dirinfo。

#### 4. ftpmirror 的自动镜像拷贝脚本 mirror.sh

ftpmirror 软件包中的 shell 脚本文件 mirror.sh 可以用来(通过使用 cron)自动镜像拷贝,用户可以用它来自动执行每天的、每周的、每月的镜像拷贝任务。下面是使用 mirror.sh 的方法:

① 在 /usr/local/lib/ftpmirror 目录下创建文件 list.xxx,这个文件里是要进行镜像拷贝的软件包的名称的列表,每行一个,空行或以“#”开头的注释行被忽略。

② 执行 mirror.sh。如果用户创建了 /usr/local/lib/ftpmirror/list.daily,则可使用下面的命令

```
lark # /usr/local/lib/ftpmirror/mirror.sh daily
```

对 list.daily 中列出的软件包进行镜像拷贝。

③ mirror.sh 的缺省输出文件是 /var/log/mirror.xxx。mirror.sh 用 rotate 程序对这些输出文件进行轮换式的更名。rotate 程序可从下列地址取来:

```
ftp://ftp.intec.co.jp/pub/utlis/rotate-1.1.tar.gz
```

## 7.5 BBS 服务器安装说明

在 Linux 上架设 BBS(电子公告牌)其实是一件很容易的事情,下面就简单介绍一下如何在 Linux 上架设一个 Firebird BBS (这是在大陆和台湾使用人数最多的一种 BBS,如国内最早的曙光 BBS,十分兴旺的水木清华、饮水思源等站都是从这个系统上改编而来的。其他 BBS 如 Mapple BBS,PowerBBS 请读者自行查询相关资料)。本章是参考了 FirebirdBBS 的 README 编写的。

### 7.5.1 安装 BBS 服务器

#### 1. 取得 Firebird BBS 的源代码

用户可以从本书所附的光盘中找到 Firebird 的源程序,或者去网络上下载,目前最新的版本是 2.5。如果用户是从台湾下载的 Big5 码的源程序,可以用一个叫 HC 的小软件

进行转码：

```
gzip -d FirebirdBBS2.5.tgz
b2g < FirebirdBBS2.5.tar > FirebirdBBS2.5gb.tar
```

或者将解压后的 FirebirdBBS2.5.tar 的文件用 richwin 或者双桥中文视窗带的内码转换器将其转成 GB 码即可。

## 2. 设定 BBS 的帐号

首先用 adduser 加 bbs 和 bbsroot 这两个帐号,然后去修改/etc/passwd 文件编辑相应栏如下(增加 bbsuser 这个用户,实际上就是 bbs 这个用户,只不过 shell 不一样,其中 bbsroot 是用来修改 /home/bbsroot/bbs\_src 中的 BBS 源程序)：

```
bbsroot: *:9990:99:BBS Adminster:/home/bbsroot:/bin/csh
bbs: 9999:99:Normal BBS user :/home/bbs:/home/bbs/bin/bbsrf
bbsuser: *:9999:99:Manage BBS user:/home/bbs:/bin/csh
```

其中 \* 号应该是加密后的密码,用户可以自行设置。

编辑/etc/group 文件,加入 bbs 管理组：

```
bbs: *:99:bbs,bbsuser,bbsroot
```

方便用户管理目录的读写权限。

## 3. 解开源代码

用 bbsroot 登录,解开 FirebirdBBS2.5.tgz

```
tar xzvf FirebirdBBS2.5.tgz
```

这时候 bbsroot 目录下就多了个 bbs\_src 目录,所有的 BBS 源程序就放在这个目录下。

## 4. 编译 BBS 源程序

以 bbsroot 身份进入 bbs\_src 目录,先执行自动配置程序：

```
cd bbs_src
cd CONFIG
./configure
```

在 linux 的系统下,屏幕上会有下面的输出：

```
creating cache ./config.cache
.....
... testing C Compiler ...
.....
checking for gcc... gcc
checking whether we are using GNU C... yes
checking whether gcc accepts -g... yes
.....
... testing system type ...
```

```

.....
checking host system type... i486-linux
checking host system type... i486-linux
checking target system type... i486-linux
checking build system type... i486-linux
.....
Please fill data for your BBS 下面让用户输入 bbs 的相关资料,回车就使用缺省选项
.....
Home directory of BBS -> [/home/bbs]      BBS 根目录,按回车使用缺省
UID of BBS -> [9999]                    BBS 的 user id,按回车使用缺省
The ID of your site -> |NoName|          BBS 的名称,用英文
The name of your site -> [Never Land BBS] BBS 的名称,用中文
The name of your site -> [some.where.on.earth] BBS 的节点名
Maximum number of users -> [20236]      BBS 最多登记用户数目
Maximum number of boards -> [256]      BBS 最大版面数
Maximum number of on-line users -> [256] 同时上线人数限制
Show idle time of BBS users ? -> |N|    是否显示发呆实际
.....
updating cache ./config.cache
creating ./config.status
creating Makefile
creating config.h
creating Install.sh
产生 Makefile 和安装程序
.....
You may review the Makefile now. Then, type 'make' to build your system.
.....

```

现在 bbs\_src 目录下就有 Makefile 文件了,用户可以开始编译程序:

```
make
```

如果不出错的话,现在 bbs\_src 下应该有这三个执行程序 bbs, bbs.chatd 以及 bbsrf。下面请用户以 root 身份登录,到 /home/bbsroot/bbs\_src 下执行安装程序(一定要用 root 登录,否则无法正确安装!):

```
make install
```

现在 BBS 系统就已经安装完毕,下面就开始配置系统了。

## 5. 登录 SYSOP 并配置 BBS

现在以 bbs 这个用户登录就可以进入 bbs 了。首先输入 new,注册一个叫 SYSOP 的用户(一定要大写),这个用户就是 bbs 的系统管理帐号。如果用户想提供 guest 帐号,就再开一个 guest 帐号,然后限制 guest 帐号只能看文章,不能发文章,不能聊天和发 mail。

如果用户不能登录,那么用户先用 bbsuser 登录,在 /home/bbs 下执行 ./bin/bbs h localhost。如果有一个无名站的画面,那就是 ./bin/bbsrf 有问题,用户将其属性应该改为

4755,且 owner 是 root。

#### 6.增加 sysop 讨论区(全部小写)

讨论区名称: sysop

讨论区说明: 0[系统] ○ 系统管理区 <-这儿一定要注意格式,中间要留空格,因为前面的 0[系统] ○全是控制码,用来控制这个讨论区的目录和属性。

讨论区管理员: SYSOP

是否限制存取权力 (Y/N)? [N]:

是否加入匿名板 (Y/N)? [N]:

#### 7.自行编辑文件

|             |                    |               |
|-------------|--------------------|---------------|
| 第一次进站的欢迎画面  | ~/Welcome          |               |
| 要求身份确认      | ~/etc/emailpost    |               |
| 进站失败画面      | ~/etc/goodbye      |               |
| 进站欢迎画面      | ~/etc/issue        |               |
| 上线人数过载画面    | ~/etc/loginfull    |               |
| 菜单设定文件      | ~/etc/menu.ini     |               |
| 新注册者使用须知    | ~/etc/newregister  |               |
| 要求输入个人资料    | ~/etc/preach       |               |
| 注册说明        | ~/etc/register     |               |
| 系统设定文件      | ~/etc/sysconf.ini  |               |
| 注册人数满载画面    | ~/etc/user_full    |               |
| 活动看板        | ~/etc/movie        |               |
| 身份确认信       | ~/etc/mailcheck    |               |
| 留言板         | ~/etc/checknotepad | ~/etc/notepad |
| 离站画面        | ~/etc/logout       |               |
| 注册成功/失败信件   | ~/etc/smail        | ~/etc/fmail   |
| 填注册单成功/失败信件 | ~/etc/s_fill       | ~/etc/f_fill  |
| 站长列表        | ~/etc/sysops       |               |

上面的文件修改完毕后,应该删除/home/bbs/sysconf.img 这个文件才能起作用。另外 bbs 目录下的文件列表如下:

|               |             |
|---------------|-------------|
| .passwd       | 用户的密码文件     |
| bbslog        | bbs 帐号的一些信息 |
| register.list | 注册用户的注册单    |
| trace         | 系统运行信息记录    |
| use_board     | 版面使用信息记录    |
| usies         | 用户使用记录      |
| login.bad     | 错误登录记录      |
| 0Announce     | 精华区目录       |

|        |            |
|--------|------------|
| bin    | 可执行文件目录    |
| boards | 版面目录       |
| etc    | 设定目录       |
| help   | 帮助文档目录     |
| home   | 用户目录       |
| mail   | 用户 mail 目录 |
| tmp    | 暂存目录       |
| vote   | 投票数据存放目录   |

## 7.5.2 使用 bbs 的一些技巧

### 1. 直接 telnet 到这个节点就以 bbs 登录

有两种方法,一种是去下载 stand alone telnetd,替换系统原有的 telnetd;另外一种非常简单,编辑/etc/rc.d/rc.local 文件加入下面几行:

```
/usr/sbin/telnetd bbs 23
/usr/sbin/telnetd XXXX 用 xxxx 端口进行正常登录
```

### 2. 扩充 tty

要增加 Linux 上的虚拟终端的数目,由于系统核心已经支持 256tty,所以只需要在/dev 下面作出相应的 tty 设备文件,并使用支持 256tty 的 telnetd 和 rlogind 就可以了。下面是增加 tty 的脚本文件 maketty:

```
#!/bin/bash
if [ `echo ${0}cut -c1-2` = './' ]; then
    proname='echo ${0}cut -c3-'
else
    proname=$0
fi

makedev () { # usage: makedev name [bcu] major minor owner:group mode
    rm -f $1
    mknod $1 $2 $3 $4 &&
    chown $5 $1 &&
    chmod $6 $1
}

usage () {
    echo
    echo "Usage : $proname <pty major>"
    echo
    echo "eg: (please refer to your include/linux/major.h)"
    echo
    echo "$proname 2 # make ptyXX for pty. master major = 2"
```

```

echo " $ progname 49 # make ptyXX for pty master major = 49"
echo
exit 0
|

if [ -z "$ 1" ] ; then
    usage
fi

if [ "$ 1 + 0" != "$ 1" ] ; then
    usage
else
    major = $ 1
fi

echo Creating ptyp0 / tty0 to ptyEf / ttyEf
minor = 0
major_slave = 'expr $ major + 1'
while [ $ minor -le 255 ] ; do
    p1 = 'expr $ minor / 16 + 1'
    p2 = 'expr $ minor % 16 + 1'
    newname = 'echo pqrstuvwxyzabcdef| cut -c $ p1'echo 0123456789abcdef| cut -c $ p2'
    echo -n mknod pty $ newname / tty $ newname
    makedev /dev/pty $ newname c $ major $ minor root:root 666
    makedev /dev/tty $ newname c $ major_slave $ minor root:root 666
    minor = 'expr $ minor + 1'
done
echo "done."
exit 0

```

运行时输入 `maketty 2` 就可以了。

### 3. 如何在 bbs 系统下安装转信系统

BBS 上的转信系统是用来将 bbs 上的文章同网络上的新闻组进行交换的工具, 这样用户就可以在自己的 bbs 系统上同新闻组的所有用户进行讨论了。

安装简介:

(1) 在安装转信系统 `innbbsd` 之前请大家一定要备份自己的 bbs 系统, 因为如果在测试过程中出了什么问题, 那就会造成大量数据的丢失。

#### (2) 取得 `innbbsd`

`ftp://ftp.csie.nctu.edu.tw/pub/bbs/misc/innbbsd/innbbsd-?? .tar.gz`

(3) 将其在 `/home/bbsroot/bbs_src` 下解开后进入 `innbbsd` 目录, 设定 `newsfeeds.bbs`,

nodelist.bbs 和 bbsname.bbs 三个文件。

- 设定 bbsname.bbs, 就是用户 bbs 的名称
- 设定 nodelist.bbs, 例如:

# 节点代号

```
# -----
newserver      news.xxx.xxx.cn      POST(119)
mypbbs         mybbs.xxx.xxx.cn      I HAVE(7777) # 用户自己的 bbs
otherbbs       otherbbs.xxx.xxx.cn  I HAVE(7777)
```

其中 POST(119) 就是 news 服务器开放的端口, I HAVE(7777) 就是 bbs 之间互相转信。

- 设定 newsfeeds.bbs, 例如:

```
# This is a sample of newsfeeds.bbs
# 新闻组          BBS 上的看版          对应的新闻服务器或 bbs 服务器
# -----
cn.bbs.comp.test    test          newserver
cn.bbs.comp.linux   linux         newserver
cn.bbs.comp.linux   linux         otherbbs
```

上面设定用户需要转的新闻组和对应的 BBS 或服务器的代号(在 nodelist.bbs 中定义 #)

(4) 编辑 Makefile, 改动下面几行

```
BBSHOME = ???
BBS_SRC = ???
```

到

```
BBSHOME = /tmp/bbs
BBS_SRC = /home/bbs
```

(5) 执行编译程序 make, 产生 innbbsd, bbsnrp, batplink 和 ctlinnbbsd。

(6) 建测试目录

```
mkdir /tmp/bbs/boards /tmp/bbs/boards/test /tmp/bbs/innd
```

将 newsfeeds.bbs, nodelist.bbs 和 bbsname.bbs 拷贝到 /tmp/bbs/innd 下。

(7) 编辑 innbbsd/newserver.active。

```
cn.bbs.test 0000000000 0000000001 y
```

(8) 运行命令

```
innbbsd
bbsnrp 新闻服务器 innd/newserver.active
```

(9) 看看 /tmp/bbs/boards/test 下有没有文章, 如果有就一切 OK。

(10) 以上测试如果成功的话, 修改 Makefile:

```
BBSHOME = /home/bbs
BBS_SRC = /home/bbsroot/bbs_src
```

这一步是设定成正确的 bbs 和 bbsrc 的目录, 接着重新编译:

```
make clean
```

```
make
```

再杀掉前面的 innbbsd 进程

```
kill `cat /usr/tmp/innbbsd-7777.pid`
```

su 成 bbsuser (和 bbs 同 uid), 运行 innbbsd

innbbsd 端口号 (default = 7777)

运行 bbsnnrp 取信:

- 每个 bbsnnrp 必须用不同的 active file。当此 active file 正在被 bbsnnrp 使用时, 请勿修改。修改前先杀掉 bbsnnrp 进程。

- active file 格式如下:

新闻组名称 高位 低位 y

例如:

```
cn.bbs.test 0000000000 0000000001 y
```

- 假设 news server 是 news.xxx.xxx.cn, active file 在 ~bbs/innd/news1.active

```
bbsnnrp news.xxx.xxx.cn ~bbs/innd/news1.active
```

如果第一次运行 bbsnnrp, 建议不要取旧文章, 只复位 active file, 用

```
bbsnnrp -c 新闻服务器 active-file
```

就取得最新的高低位数

- 将以上操作加到 crontab 定时转信

```
crontab -e
```

例如, 每十分钟取一次就输入

```
10,20,30,40,50 * * * * /home/bbs/innd/bbsnnrp news-server \
/home/bbs/innd/active-file > /dev/null 2 > &1
```

或用

```
bbsnnrp -w 等待时间?..... &
```

则 bbsnnrp 每一段时间转一次。

- BBS 站不通过新闻服务器互转

在 nodelist.bbs 加入

```
bbs-server1 bbs-server1-host-name IHAVE(7777)
```

```
bbs-server2 bbs-server2-host-name IHAVE(7777)
```

在 newsfeeds.bbs 中加入

```
newsgroups board-name name1 name2 ...
```

```
cn.bbs.test test bbs-server1 bbs-server2
```

(11) 送信程序 bntplink 安装设定

bbsnnrp 只用取 news, 转 news 出去还要另外运行 “bntplink”。第一次运行 bntplink 前先执行 (假设 BBS home 在 /home/bbs/)

```
% bbspost visit /home/bbs # 第一次执行会非常久
```

New article listed:

```
test M.860912324.A SYSOP      测试文章 ...
....                          # 列出所有未送信文章
```

接下来到 test 版 post 一篇测试文章,然后执行下列命令

```
% batplink /home/bbs
```

看看新闻组里有没有这篇文章。

(12) Server gateway innbbsd 控制程序 (ctlinnbbsd) 介绍

ctlinnbbsd reload 重新装入 innbbsd 的数据文件

ctlinnbbsd shutdown 关闭 innbbsd

ctlinnbbsd mode 检查 innbbsd 的模式

ctlinnbbsd addhist < mid > path 增加历史记录

ctlinnbbsd grephist < mid > 查询历史记录

ctlinnbbsd hismaint 维护历史记录

ctlinnbbsd getrusage 获取资源使用状况(-DCETRUSAGE)

如果要关闭 innbbsd,最好使用 ctlinnbbsd shutdown。另外在 bbsnarp 运行时千万不要改 active file,不然后果自负!

另外要转 big5 内码的信件就在编译时加上 hztty 的支持,然后编辑/home/bbs/innd 下面的 filter.ctl 文件,加入下面一行

```
tw.bbs.* ,hk.* :big2gb:gb2big
```

就可以直接转 big5 内码的信件了。

## 7.6 安装新闻服务器

新闻服务器分 innd 和 cnews 两种,不过现在使用 innd 的更多一些,下面就简单介绍如何安装 innd。

(1) 到下面的节点下载最新的 innd:

```
ftp://ftp.vix.com/pub/innd/;
```

(2) 解开 innd \*.tgz;

(3) 执行 ./configure 进行系统检测;

(4) 执行 make 编译程序;

(5) 执行 make install 安装程序;

(6) 修改一些系统的设定

编辑下面的文件:

```
/etc/passwd:
```

```
news:*:9:13:news:/usr/lib/news:
└──────────┬──────────┘
            |            |
            |            |gid 13
            |            |
            └──────────┘
            |
            |uid 9
```

```

/etc/group:

news:13:news
      |
      +--- gid 13

/usr/lib/aliases:

news: root
usenct: root

```

更改此文件后请用户以 `newaliases` 命令更新 dbm aliases database, 再执行下面的命令:

```

cd /usr/bin
ln -sf ../lib/news/inews inews
ln -sf ../lib/news/rnews rnews

```

请用户在 `/usr/lib/news` 目录内建立以下各空文件:

```

touch history
touch history.dir
touch history.pag
touch errlog
touch log

```

并将权限和拥有人改为 `news` 所有。请用户注意 `inn news server` 对权限的要求, 如用户设的权限错误, 就算用户是 `root` 照样无法让它正常工作。

```

chmod 664 {history *, log, errlog}
chown news.news {history *, log, errlog}

```

然后是修改 `/etc/syslog.conf` 加入

```

* .warn                                /usr/adm/syslog
* .err                                /usr/adm/syslog

```

之后请用户确定在 `/var/spool/news/` 下是否有 `junk`, `control`, `in.coming` 和 `out.going`。如果没有的话请用户以手动开启, 并将其拥有人改为 `news`, 接着请用户编辑它的各设定文件。

```

/usr/lib/news/inn.conf:

fromhost:                test.xxx.xxx.cn
pathhost:                 test.xxx.xxx.cn
organization:            TEST Work groups
server:                  test.xxx.xxx.cn
domain:                  xxx.xxx.cn

/usr/lib/news/hosts.nttp:

# 设定本主机可由哪一台主机传送给讨论信给用户
# 如果愿意将讨论信传送给用户的站名
# 或者 IP 地址不列在此文件的话, 它将无法得到传输的权力

```

```
test:xxx.xxx.cn:
les(x:xxx.xxx.cn:
127.0.0.3:

/usr/lib/news/nntp.access:

# nntp.access 基本有这三种设定方法,详细的请用户自行参阅 man.
#:no:~no:~! *      ——关闭所有读取和发表功能
test:xxx.xxx.cn:READ POST:::~ *      ——只允许 test.xxx.xxx.cn 读取和发表
~:Read Post:::~ *      ——所有人都可读取和发表

/usr/lib/news/nntp.send.ctf:

# 此文件记录的是用户想传 news 给哪些主机
# 假如有新建立的新闻主机要求用户这么传送的话
# 此文件必须相对应于 /usr/lib/news/newsfeeds 文件内
testx:testx.xxx.xxx.cn::

/usr/lib/news/expire.ctf:

# 这是记录讨论信多久过期,如果全设为 14 天就象下面这样
/remember/:14
*:A:14:14:14

/usr/lib/news/newsfeeds:

# 这是定义用户所接收的讨论区及传给下一个新闻服务器哪些讨论区
ME:cn.*:: ——此行是必须的,指定用户所接收的讨论区为 cn.*

testx \ ——
:cn.* \ | ——这是定义用户也传递 cn.* 给 testx.xxx.xxx.cn
:Tf,Wnm: ——
```

至此 INN 新闻服务器已大致设定完成了,笔者认为设定比较简单,管理的话用户就得花些心思研究随 inn 而来的文档了。全部设定完后请用户取得 active newsgroups 这两个文件,具体操作如下:

```
su - news
telnet 上级新闻服务器 nntp > /tmp/active
list active
quit
telnet 上级新闻服务器 nntp > /tmp/newsgroups
list newsgroups
quit
```

如此便取回这两个文件。然后用用户编辑这两个文件,将一些标题删除并且把用户所不想接收的讨论区删除,然后用下列命令将 active 内的记录置零:

```
awk 'f; $2 = 0000000000; $3 = 0000000001; print f' \
/tmp/active > /usr/lib/news/active
```

如此即可完成置零的动作,然后用用户把刚刚传回来的 newsgroups 文件也拷贝至 /usr/

lib/news,至此需要的文件就大致齐全了。

#### (7) 启动新闻服务器

接下来便是启动 INN。不过别急,因为 news 所用的 port 119 目前是 Inet 这个守护进程所监管的,用户必须把它 mark 掉并重新启动 inetd,如此 inn news 才可取得 port 119 的控制权。

启动前请用户先用 inncheck 查查看,是不是所有的设定文件都正确了呢?

```
/usr/lib/news/bin/inncheck -v
Looking at /usr/lib/news/active
Looking at /usr/lib/news/control.ctl
Looking at /usr/lib/news/expire.ctl    440    一以下这些号码是这些文件的
Looking at /usr/lib/news/hosts.nntp    440    权限设定,
Looking at /usr/lib/news/inn.conf      444
Looking at /usr/lib/news/moderators    444
Looking at /usr/lib/news/newsfeeds     444
ME, testx, /usr/lib/news/newsfeeds:0;
ME entry accepts all incoming article distributions
done
Looking at /usr/lib/news/nntp.access    440
Looking at /usr/lib/news/nntp.send.ctl  440
Looking at /usr/lib/news/overview.fmt  444
Looking at /usr/lib/news/passwd.nntp
```

如果有以上的字出现就代表正常,可以启动新闻服务器。

请注意除了启动 server 执行 /usr/lib/news/etc/rc.news 需要 root 的身份外,其他管理请以 news 身份进行,不然系统会很不正常。现在用户可以启动服务器,再以 ps ax 看有没有 /usr/lib/news/etc/innd -p4 -i0 这一行。如用户看到了,就可以用 tin 或者其他新闻阅读器去发表文章;如没成功的话请用户看看 /var/adm/syslog 有没有什错误讯息,然后再试试看。

用户的新闻服务器已经正常运行后,用户可以将这些加入用户的/etc/rc.d/rc.local 文件中。

```
if [ -f /usr/local/etc/rc.news ]; then
    /usr/lib/news/rc.news ;
    echo Starting INN news service
fi
```

以 root 身份使用/bin/crontab -e news 加入下面这两行:

```
40 23 * * * /usr/local/news/bin/news.daily delayrm
0,10,20,30,40,50 * * * /usr/lib/news/send-nntp testx testx.xxx.xxx.cn
```

下面是示范如何增加新闻组。新增一个 cn.bbs.comp.Linux 的新闻组:

```
ctlinnd newgroup cn.bbs.comp.Linux
```

编辑 ~news/lib/newsgroups,加入一行

cn.bbs.comp.linux 中文 Linux

就可以了。

## 7.7 Linux 和 Windows 95/NT 的文件和打印机共享

SMB(Server Message Buffer)是 Windows for Workgroup(WfW), Windows 95, Windows NT 和 Lan Manager 等用来实现在局域网文件共享和打印机共享的协议。在使用 Windows 95/NT 时,我们经常通过网络邻居浏览其他机器上共享目录里的文件,或把文件输出到联结在另一台机器上的共享打印机上。Linux 不仅可以使 NFS 和 UNIX 打印队列等方法与其他 UNIX 系统实现文件和打印机共享,也可以使用 SMB 协议实现与 Windows 95/NT 等系统的文件和打印机共享。

Samba 是 SMB 客户程序/服务器软件包,它主要包含以下程序:

(1) SMB 服务器 `smbd` 为 SMB 客户机如 Windows 95/NT、Warp 服务器等提供 Windows NT 和 LAN Manager 风格的文件和打印服务。

(2) Netbios(rfc1001/1002)名字服务器 `nmbd` 可以提供浏览支持,用户甚至可以用 Samba 作为局域网的主浏览服务器。

(3) SMB 客户程序 `smbclient` 类似 `ftp` 程序,用以从 UNIX、Netware 和其他操作系统上访问 SMB 服务器上的资源(文件和打印机)。

(4) SMB 客户程序的 `tar` 扩展 `smbtar` 用以方便地拷贝 SMB 服务器上的文件。

在 Linux 上, Samba 还提供了挂卸 SMB 文件系统的工具程序 `smbmount` 和 `smbumount`。下面就介绍 Samba 在 Linux 上的安装和使用。

### 7.7.1 Samba 的安装和配置

最新的 Samba 软件包可从下列地址

`ftp://ftp.samba.org/pub/samba/`

或其镜像节点取来。目前的版本是 `samba-2.0.3.tar.gz`。

将 `samba-2.0.3.tar.gz` 展开在某个工作目录下,应产生 `/samba-2.0.3` 目录,到其下的 `source/` 子目录下,编辑文件 `Makefile`:

```
% cd samba-2.0.3/source
```

```
% ./configure
```

用户可以在运行“`./configure`”之前用“`./configure --help`”看一看可能的配置选项,如 Samba 的安装目录(缺省为 `/usr/local/samba`,用“`--prefix`”配置)等。有一些选项可以在安装完成后在配置文件 `smb.conf` 中设定,无须在 `Makefile` 中更改。

然后,在 `/source` 目录下编译 Samba 并安装到指定目录中:

```
% make
```

```
# make install
```

注意安装时需要使用 `root` 帐号。

安装过程将在系统上创建如下目录和文件(假设安装目录为 `/usr/local/samba`):

```

/usr/local/samba/bin:
    addtosmbpass,      nmblookup,      smbmount,      smbtar,
    make_printerdef,   smbclient,      smbpasswd,      smbmount,
    make_smbcodepage,  smbd,           smbrun,        testparm,
    nmbd,              smbstatus,      testprns,      convert_smbpasswd

/usr/local/samba/lib/codepages:
    codepage.437,codepage.850,codepage.861,codepage.932,codepage.949,
    codepage.737,codepage.852,codepage.866,codepage.936,codepage.950

/usr/local/samba/var:

```

安装完毕后,我们必须创建 Samba 配置文件 `smb.conf`。在软件包的 `examples/` 目录下有一个缺省的配置文件 `smb.conf.default`,我们可以对它做适当修改后拷贝到 `$BASEDIR/lib` 目录下,命名为 `smb.conf`。注意“make install”并不安装 `smb.conf` 文件,用户必须自己在 `$BASEDIR/lib` 目录(或用户在 `Makefile` 文件里定义的其他目录)下创建 `smb.conf` 文件。

`smb.conf` 的格式有点像 Windows 95 里的 .INI 文件。文件分成若干节。每节以放在方括号里的节的名字开始,然后直到下一节开始为止。每节里是若干参量设置行,其格式为“参量名 = 参量值”。节和参量名不区分大小写。每个参量行只有第一个“=”号有意义,其两边的空格被忽略。节和参量名两头和中间的空格也被忽略。参量值两头的空格被忽略,但中间的空格被保持原样。以井号( # )和分号( ; )开始的行和空行被忽略。以外斜线( \ )结束的行将在下一行继续,与一般 UNIX 的规则一样。参量值是字符串(无须引号)或布尔值(Yes/No,0/1 或 true/false)。

`smb.conf` 文件的每一节描述一种(共享)服务。节的名字是服务名,节的参量定义服务的属性。有三个特殊的节: `[global]`, `[homes]` 和 `[printers]`,它们有特殊的属性,分别定义了一些全局性的参量、用户的主目录文件服务和打印机服务。

每种服务的定义由用以访问这种服务的目录、用户的访问权限和其他一些设置组成。所有服务的目录(由 `path` 参量定义)必须事先存在。服务按其类型可分为文件服务和打印服务,按其用户可分为客人( `guest` )服务(不需口令)和用户服务(需要口令)。

当配置文件里有 `[printers]` 一节时,客户程序可以连接本地 `printcap` 文件中定义的任何打印机。服务器将为本地 `/etc/printcap` 文件里的每一个打印机创建一个同名的共享服务。

当配置文件里有 `[homes]` 一节时,服务器可以自动创建将客户连接到他们的主目录的服务。如果用户请求连接“//server/service”,服务器将搜寻现存的服务中是否含有名为 `service` 的服务。如果没有,服务器将认为 `service` 是一个用户名,而在本地口令文件中搜索用户 `service`;如果用户 `service` 存在且用户输入了正确的口令,则服务器将按 `[homes]` 复制一个名为 `service` 的服务,其路径(`path` 参量)为用户的主目录。注意:如果用户名与某一打印机同名,其主目录服务将不被创建。

关于所有可定义的参量的说明请参考 Samba 软件包的在线帮助 `smb.conf(5)`,我们在这里只通过例子简单介绍最常用的一些参量的定义。

下面是一个简单的 `smb.conf` 文件的例子,使用了一些最常用的参量,我们来行说明每一参数的意义。

```
# ——全局参量节——
[global]

# workgroup = NT 域名或 Workgroup 名
workgroup = MYGROUP

# server string 等价于 NT 描述域 (Description field)
server string = Samba Server

# 将连接请求限制在局域网上的某些机器。如下例允许两个 C 类网络和
# “loopback”界面的连接请求,而禁止来自其他机器的连接请求。
; hosts allow = 192.168.1. 192.168.2. 127.

# 自动加载打印机列表,而不是单独设置每一个打印机
load printers = yes

# 如果需要使用另一个 printcap 文件,则定义此参量
; printcap name = /etc/printcap

# 如果希望建立客人帐号,定义此参量。同时必须把这帐号加进/etc/passwd,
# 否则使用用户“nobody”作为客人帐号
; guest account = penguin

# 对从每个机器过来的连接使用一个单独的日志文件
log file = /usr/local/samba/var/log.%m

# 限制日志文件的最大长度(Kb)
max log size = 50

# 安全模式。很多人希望使用用户水平的安全模式
# 参考 security_level.txt 文件
security = user

# 只当 security = server 时使用口令服务器选项
; password server = < NT-Server-Name >

# 输入口令应与真实口令前 password level 个字符符合
; password level = 8

# 用户也许希望使用口令加密。在仔细阅读 ENCRYPTION.txt, Win95.txt
# 和 WinNT.txt 文件之前不要使用该参量
; encrypt passwords = yes

# UNIX 用户可以映射成不同的 SMB 用户名
```

```
; username map = /etc/smbusers

# 对每一个不同的机器使用不同的配置文件
# %m 将替换成请求连接机器的 netbios 名字
; include = /usr/local/samba/lib/smb.conf.%m

# 这个选项将使服务器提供更高的性能,特别是在使用 smbmount
# 挂卸 SMB 文件系统时。参考 speed.txt 文件
socket options = TCP_NODELAY

# 配置 Samba 使用多个网络界面
# 如果机器使用多个网络界面,必须在这里列出
; interfaces = 192.168.12.2/24 192.168.13.2/24

# 保持大小写。系统缺省是"No"
# 注意;可对每个共享服务单独设置此参量
; preserve case = No
; short preserve case = No
# 对于 DOS 文件缺省是使用大写字母
; default case = lower
# 谨慎使用区分大小写参量。它可能会导致错误
; case sensitive = No

# ——共享服务定义——
[homes]
# 说明文字,当一个客户程序列出本服务器的共享服务时,comment 后的
# 文字将出现在共享名旁边
comment = Home Directories
# 当一个客户程序以客人用户列出本服务器的共享服务时,homes 服务将
# 不出现在列表中。但为本机的真实用户创建的主目录服务继承[global]区设
# 定的 browseable(缺省是 browseable = Yes)
browseable = No
# 客户程序对服务目录有写的权限,将被为真实用户创建的主目录服务继承
writable = Yes

# 定义打印机共享。使用 BSD 风格的打印系统不必单独定义每一个打印机
[printers]
comment = All Printers
# path 指定的目录必须事先创建,对其他服务也是这样,否则服务将不能使用
path = /tmp
browseable = No
# 设置 public = Yes 允许客人( guest account)使用打印机。
# guest ok 与 public 同义,即设置为 Yes 时使用该项服务无须口令
```

```
guest ok = No
# printable 服务总是允许向服务目录中写文件,但只能通过打印缓存操作实现
writable = No
printable = Yes

# 设置网络用户共享文件的临时区域,每个人都可在这里存放文件
# 供别人使用
[tmp]
; comment = Temporary file space
; path = /tmp
; read only = No
; public = Yes

# 一个只读的共享目录,但在"staff"组的用户有写的权限
[public]
; comment = Public Stuff
; path = /home/samba
; public = Yes
; writable = Yes
; printable = No
# "@"后是用户组的名称。注意:使用 write list 必须设定 writable = Yes
; write list = @staff

# 一个私人打印机,只提供给 fred 使用。打印缓存目录设为 fred 的主目录。
# 当然 fred 必须有往缓存目录写的权限
[fredsprn]
; comment = Fred's Printer
; valid users = fred
; path = /home/fred
# 打印机名
; printer = fred's_printer
; public = No
; writable = No
; printable = Yes

# 一个私人目录,只允许 fred 使用。注意:fred 对此目录需要有写的权限
[fredsdire]
; comment = Fred's Service
; path = /usr/somewhere/private
; valid users = fred
; public = No
; writable = Yes
; printable = No
```

```
# 一个对每个连接的机器有不同的目录的服务。这使得我们可以对不同的
# 机器使用不同的配置。我们也可以用“%L”对不同的用户使用不同的配置
# “%m”在连接时将替换成客户机器名
;[pcthome]
; comment = PC Directories
; path = /usr/po/%m
; public = No
; writable = Yes

# 一个公共目录,所有用户有读写权限。注意用户在这个目录下创建的所有
# 文件的属主都将为缺省用户(guest account),所以任何用户都可以删除
# 别的用户在此目录下放置的文件。显然缺省用户必须对此目录有写的权限
# 当然也可以指定别的用户,这时所有的文件的属主将为此用
;[public]
; path = /usr/somewhere/else/public
; public = Yes
; only guest = Yes
; writable = Yes
; printable = No

# 共享一个目录,使两个用户可以在这个目录下放置文件,文件的属主
# 是各自的用户。在这个设置里,共享目录应该是两个用户都可写的。当然
# 文件的保护应该设置恰当。这个设置可以扩展到多个用户的情况
;[myshare]
; comment = Mary's and Fred's stuff
; path = /usr/somewhere/shared
; valid users = mary fred
; public = No
; writable = Yes
; printable = No
; create mask = 0765
```

编译好适当的 smb.conf 文件后,把它放置在 \$BASEDIR/lib 目录下。这个文件可在以后任何时候根据需要修改。

现在用“testparm”程序检验 smb.conf 文件的设置。如果一切正常,它将打印出已定义的服务的信息;否则它将会打印错误信息,用户就需要修改 smb.conf 文件。

启动 smbd 和 nmbd 可以选择使用守护进程的方式还是用 inetd 启动。如果选择使用守护进程,可以把启动 smbd 和 nmbd 的命令加进/etc/rc.d/rc.local 文件(或其他文件)中。但更方便的是修改/etc/inetd.conf 和/etc/services,使得在必要的时候由 inetd 启动 smbd 和 nmbd。

现在我们介绍如何使用 inetd 自动启动 smbd 和 nmbd。首先编辑/etc/services 文件,

将端口 139/tcp 和 137/udp 定义如下：

```
netbios-ssn 139/tcp
netbios-ns 137/udp
```

如果/etc/services 文件中已有定义,依照上面的定义进行适当的修改。

其次,修改/etc/inetd.conf 文件,加进下面两行:

```
netbios-ssn stream tcp Nowait root /usr/local/samba/bin/smbd smbd
netbios-ns dgram udp wait root /usr/local/samba/bin/nmbd nmbd
```

其中的命令路径要指向 Samba 的安装目录。

用“kill -HUP”命令重新启动 inetd 进程。如果系统上有老的版本的 nmbd 正在运行,要把它杀死。

现在我们就可以使用 Linux 上的 SMB 服务器和各种客户程序了。

### 7.7.2 Samba 的使用要点

当我们在 Linux 上安装了 SMB 服务后,就可以在 Windows 95/NT 的网络邻居里发现我们的计算机,并可使用我们的共享服务。

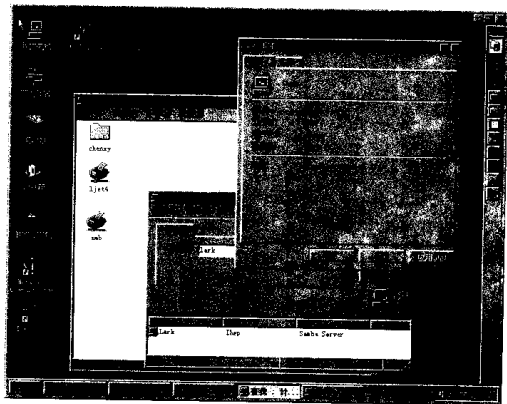


图 7.5 在 Windows 95 里使用 Linux 上 Samba 提供的共享服务

在 Linux 上,我们可以用 smbclient 命令访问局域网上的 SMB 服务器,如 Samba 服务器 smbd、Windows 95/NT 的共享目录和打印机等。例如,对于我们刚安装配置好的 Linux 上的 SMB 服务器,我们可以用“smbclient -l server”命令来查看它的共享服务列表。这里 server 是 SMB 服务器的 netbios 名。SMB 服务器会询问口令,即服务器上与请求连接用户同名的真实用户的口令。如果口令回答正确,将以此用户建立连接;如果不同答口令,将

以客人用户(guest account)建立连接 然后 SMB 服务器会打印出服务器的共享服务列表,浏览列表和工作组列表:

```
% smbclient -L lark
Added interface ip = 202.38.128.32 boad = 202.38.128.255 nmask = 255.255.255.0
Server time is Tue Mar 17 10:31:35 1998
Timezone is UTC + 8.0
Password:
Domain = [IHEP] OS = [UNIX] Server = [Samba 1.9.18p3]
security = user
```

```
Server = [LARK] User = [chenxy] Workgroup = [IHEP] Domain = [IHEP]
```

| Sharename | Type    | Comment                    |
|-----------|---------|----------------------------|
| -----     | ---     | -----                      |
| chenxy    | Disk    | Home Directories           |
| IPC\$     | IPC     | IPC Service (Samba Server) |
| ljet4     | Printer | lp                         |
| smb       | Printer |                            |

This machine has a browse list:

| Server | Comment        |
|--------|----------------|
| -----  | -----          |
| HPTC2  | Samba 1.9.17p2 |
| LARK   | Samba Server   |

This machine has a workgroup list:

| Workgroup | Master |
|-----------|--------|
| -----     | -----  |
| IHEP      | HPTC2  |

smbclient 的用法定义如下:

```
% smbclient
Usage: smbclient service <password> [-p port] [-d debuglevel] [-l log]
Version 1.9.18p3
-p port          connect to the specified port
-d debuglevel    set the debuglevel
-l log basename. Basename for log/debug files
-n netbios name. Use this name as my netbios name
-N              don't ask for a password
-P              connect to service as a printer
-M host         send a winpopup message to the host
```

|                   |                                                 |
|-------------------|-------------------------------------------------|
| -m max protocol   | set the max protocol level                      |
| -L host           | get a list of shares available on a host        |
| -I dest IP        | use this IP to connect to                       |
| -E                | write messages to stderr instead of stdout      |
| -U username       | set the network username                        |
| -W workgroup      | set the workgroup name                          |
| -c command string | execute semicolon separated commands            |
| -t terminal code  | terminal i/o code [sjjs euc jis7 jis8 jnet hex] |
| -T <clx> 1XgbNa   | command line tar                                |
| -D directory      | start from directory                            |

下面是 smbclient 使用的例子:

```
smbclient -l server 浏览 SMB 服务器 server 的共享服务列表
smbclient //server/service 连接 SMB 服务器 server 上的服务 service
smbclient //server/service -N 不在命令行询问口令(service 支持客人用户访问)
smbclient //server/service -P 以打印机方式连接 server 上的打印机服务 service
smbclient //server/service -U user 设置网络用户名为 user
```

使用 smbclient 建立某一服务连接后,将进入一个类似 ftp 命令的进程,可以在 smbclient 的提示符下使用很多熟悉的命令,必要时用“?”命令查询可用的命令。如下面是一个用 smbclient 连接一个 Window 95 的共享目录服务的例子:

```
lark[287]% smbclient //siskin/book
Added interface ip = 202.38.128.32 bcast = 202.38.128.255 nmask = 255.255.255.0
Server time is Tue Mar 17 10:50:08 1998
Timezone is UTC + 8.0
Password:
security = share
smb: \ > dir
.                D                0    Sat Mar 14 09:56:56 1998
..               D                0    Sat Mar 14 09:56:56 1998
ACROREAD        D                0    Sat Mar 14 09:56:56 1998
PDF1            D                0    Sat Mar 14 09:57:02 1998
PDF2            D                0    Sat Mar 14 10:00:16 1998
PDF2TXT         D                0    Sat Mar 14 10:06:24 1998

63989 blocks of size 16384. 18596 blocks available
smb: \ > cd acroread
smb: \acroread \ > dir
.                D                0    Sat Mar 14 09:56:56 1998
..               D                0    Sat Mar 14 09:56:56 1998
ABCPYDOC.INI    AR            14979  Sat Jan 10 11:09:50 1998
ARCHIVE.Z       AR            3152851 Sat Jan 10 11:09:54 1998
README.TXT      AR            39617  Sat Jan 10 11:09:54 1998
```

```

SETUP.EXE      AR      44064   Sat Jan 10 11:09:54 1998
SETUP.INS      AR      283452  Sat Jan 10 11:09:54 1998
SETUP.PKG      AR       1265   Sat Jan 10 11:09:54 1998
 _INST321.EX_  AR      323612  Sat Jan 10 11:09:50 1998
 _ISDEL.EXE    AR       8192   Sat Jan 10 11:09:50 1998
 _SETUP.DLL    AR       5984   Sat Jan 10 11:09:50 1998
 _SETUP.LIB    AR      247179  Sat Jan 10 11:09:50 1998

```

63989 blocks of size 16384, 18596 blocks available

```

smb: \acoread\ > prompt
prompting is Now off
smb: \acoread\ > mget *
getting file \acoread\ ABCPYDOC.INI of size 14979 bytes as ABCPYDOC.INI
(365.697 kb/s) (average 365.698 kb/s)
getting file \acoread\ ARCHIVE.Z of size 3152851 bytes as ARCHIVE.Z
(736.416 kb/s) (average 732.903 kb/s)
getting file \acoread\ README.TXT of size 39617 bytes as README.TXT
(118.313 kb/s) (average 688.714 kb/s)
getting file \acoread\ SETUP.EXE of size 44064 bytes as SETUP.EXE
(632.812 kb/s) (average 687.891 kb/s)
getting file \acoread\ SETUP.INS of size 283452 bytes as SETUP.INS
(846.509 kb/s) (average 698.384 kb/s)
getting file \acoread\ SETUP.PKG of size 1265 bytes as SETUP.PKG
(247.065 kb/s) average 697.928 kb/s)
getting file \acoread\ _INST321.EX_ of size 323612 bytes as _INST321.EX_
(892.732 kb/s) (average 710.935 kb/s)
getting file \acoread\ _ISDEL.EXE of size 8192 bytes as _ISDEL.EXE
(615.38 kb/s) (average 710.701 kb/s)
getting file \acoread\ _SETUP.DLL of size 5984 bytes as _SETUP.DLL
(531.245 kb/s) (average 710.33 kb/s)
getting file \acoread\ _SETUP.LIB of size 247179 bytes as _SETUP.LIB
(768.744 kb/s) (average 713.582 kb/s)
smb: \acoread\ > quit
lark[288]%

```

smbclient 更详细的用法请参照 Samba 软件包里的说明文件。

如果用户的系统内核支持 SMB 文件系统,就可以用 `smbmount` 和 `smbumount` 命令挂卸局域网上的其他 SMB 服务器上的共享目录,像使用 NFS 文件系统一样在 Linux 里方便地使用 SMB 文件系统。为使普通用户也可以使用这一功能,必须把 `smbmount` 和 `smbumount` 程序安装成 `setuid` 的,例如:

```

# chmod 4755 /usr/local/samba/bin/smbmount
# chmod 4755 /usr/local/samba/bin/smbumount

```

假设我们需要把运行 Windows 95 的机器 `siskin` 上的共享目录 `book` 挂到 Linux 上,可

使用如下命令：

```
% smbmount //siskin/book /home/chenxy/mnt
```

我们可以用“df”命令显示一下系统的文件系统：

```
% df
Filesystem            1024-blocks    Used    Available    Capacity    Mounted on
/dev/hda2              1181489      800454      319988        71%      /
/dev/hda4              2454879     1982457      345509        85%      /home
//siskin/book          1023824      726288      297536        71%      /home/chenxy/mnt
```

现在我们可以像使用本地文件系统一样使用 SMB 文件系统了：

```
% cd /home/chenxy/mnt
% ls -lAF
total 4
drwxr-xr-x 1 chenxy users 512 Mar 14 09:56 acroread/
drwxr-xr-x 1 chenxy users 512 Mar 14 09:57 pdf1/
drwxr-xr-x 1 chenxy users 512 Mar 14 10:00 pdf2/
drwxr-xr-x 1 chenxy users 512 Mar 14 10:06 pdf2txt/
```

当我们想卸下某个 SMB 文件系统时,使用命令 `smbumount`：

```
% smbumount /home/chenxy/mnt
```

### 7.7.3 Linux 和 Windows 95/NT 的打印机共享

Samba 使 Linux 上的打印机可以成为共享资源,也使得 Linux 可以使用 Windows 95/NT 等 SMB 服务器上的共享打印机。下面介绍 Linux 和 Windows 95/NT 怎样使用对方的共享打印机。

#### 1. 在 Linux 上使用 Windows 95/NT 的打印机

在 Samba 软件包里提供了一个辅助打印的 shell 脚本文件 `smbprint`,它利用 `smbclient` 命令将打印发送到网络打印机上。我们可以利用它建立网络打印机队列。

首先,在 Windows 95/NT 里将打印机设为共享,最好设置共享口令,以避免对打印机无节制地使用。

在 Linux 系统的 `/etc/printcap` 文件里为网络打印机建立一个新的队列,其定义大致如下：

```
smb: \
:sd = /var/spool/lpd/smb: \
:mx # 0: \
:sh: \
:lp = /dev/null: \
:af = /var/spool/lpd/smb/acct: \
:if = /var/spool/lpd/smb/filter:
```

注意：

- ① 打印机设备名定义: `lp = /dev/null`,因为网络打印机不与本地任何设备文件相联

系。

② 定义记帐文件: `af = /var/spool/lpd/smb/acct`, 实际上并不一定实施记帐, 这条记录提供的目录将被 `smbprint` 程序使用。

将 Samba 软件包提供的 `smbprint` 程序(在 `examples/printing/` 目录下)拷贝到 Samba 的安装目录下 `/bin` 子目录里(如 `/usr/local/samba/bin`)。注意设置成可执行模式(755)。

创建打印机缓存目录 `/var/spool/lpd/smb`, 在其中建立打印过滤器文件 `filter`。我们可以把在第五章 5.12.4 节“打印过滤器”里介绍的简单打印过滤器文件稍作修改, 产生如下的网络打印机过滤器文件 `filter`:

```
# ! /bin/sh

DEVICE = ljet4
RESOLUTION = 600x600
PAPERSIZE = a4
SENDEOF =

nenscript -TUS-ZB -p -l
if [ "$ DEVICE" = "PostScript" ]; then
    cat - | /usr/local/samba/bin/smbprint -
else
    gs -q -sDEVICE = $ DEVICE \
        -r $ RESOLUTION \
        -sPAPERSIZE = $ PAPERSIZE \
        -dNOPAUSE \
        -dSAFER \
        -sOutputFile = - - | /usr/local/samba/bin/smbprint -
fi

if [ "$ SENDEOF" = "" ]; then
    printf ""
fi

exit 0
```

这里我们假设网络打印机是 HP LaserJet 4L 系列, 对于其他类型的打印机, 只要更改上面 `filter` 文件中的“`DEVICE = ljet4`”一行就可以了(参考 5.12.4 节)。

`smbprint` 运行时需要读取一个网络打印机信息文件 `.config` 中的信息, 这个信息文件放置在 `printcap` 中定义的记帐文件所在目录。对于本例, 记帐文件为 `af = /var/spool/lpd/smb/acct`, 其目录为 `/var/spool/lpd/smb`, 所以我们必须在这个目录下创建 `.config` 文件如下:

```
server = SISKIN
service = LJ6L
password = "hplj6l"
```

其中给出打印机服务器的名称,网络打印机的共享名及共享口令。为保密起见,将`.config`的属主设为`bin`,保护设为`0400`。

现在我们就可以用下列命令重新启动打印管理器(使用`root`帐号),把网络打印队列激活:

```
# lpc restart all
```

然后用如下命令打印到网络打印机上:

```
% lpr -P smb my_file
```

**注意:** `smbprint`使用的缺省日志文件是`/tmp/smb-print.log`。在第一次使用网络打印队列前,必须确保在`/tmp`目录下无此文件。`smbprint`第一次运行时会创建这个文件,给它设置正确的文件保护。如果`smbprint`运行时无法写入此文件,将导致严重的后果(不断重新启动打印队列, `/usr/adm/messages`文件急剧膨胀)。

如果用户使用的是 RedHat 的打印管理器 `printtool`,可以直接在 `printtool` 窗口里增加网络打印机,而不必手工进行以上的配置。

## 2. 在 Windows 95/NT 上使用 Linux 的打印机

Linux 上的 Samba 服务器可以将 Linux 的打印机设为共享资源。那么,怎样在 Windows 95/NT 中设置使用 Linux 的共享打印机呢?这分为两种情况,即取决于在 Windows 95/NT 看来, Linux 上的打印机是 PCL 打印机还是 PostScript 打印机。

对于在 Linux 中用 RedHat 的 `printtool` 产生的打印队列,由于 `printtool` 的打印过滤器会自动区分 PCL 文件和其他格式的文件,所以从外界看来,它是 PCL 打印机。所以在 Windows 95/NT 中,我们可以按照通常安装网络打印机的方法安装真实打印机的驱动程序,如对于 Linux 上的 HP LaserJet 4L 系列打印机,在 Windows 95/NT 中安装网络打印机时,我们使用 HP LaserJet 4L 的驱动程序。

对于在 Linux 中使用本书第五章 5.12.4 节“打印过滤器”中的简单打印过滤器设置的打印机队列,由于它没有区分普通文本文件和 PCL 文件,且使用了 Ghostscript 作为打印过滤程序的一部分,所以从外界看来,它是一个 PostScript 打印机。我们在 Windows 95/NT 中安装网络打印机时,必须使用某种 PostScript 打印机的驱动程序,如使用 HP LaserJet 4P/4MP PostScript 的驱动(实际上,使用哪种 PostScript 打印机的驱动无关紧要),还必须将打印机属性中的“PostScript 输出格式”改成“存档格式”。下面说明在这种情况下网络打印机的安装步骤:

(1) 在 Windows 95/NT 中选择“开始”→“设置”→“打印机”→“添加打印机”;在“添加打印机”向导里选择“网络打印机”;在“网络路径或队列名”里填写网络打印机名称,如“\\lark \ ljet4”(如图 7.6),这是一台在 Linux 上用简单过滤器建立的打印机队列。

(2) 选择“HP LaserJet 4P/4MP PostScript”作为网络打印机的驱动程序。安装驱动程序(如图 7.7)。

(3) 驱动程序安装完毕后,用鼠标右键点击刚生成的网络打印机图标,选择“属性”,在打印机属性窗口里,改“PostScript”项中“PostScript 输出格式”为“存档格式”(如图 7.8)。

如此安装之后,在 Windows 95/NT 中,所有的打印作业将首先被 PostScript 打印机驱动程序转换为 PostScript 文件,然后发往在 Linux 上的网络打印机打印输出。

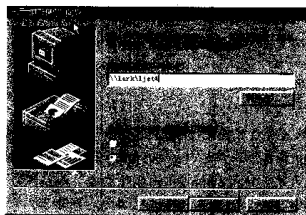


图 7.6 在 Windows 95 中安装 Linux 的共享打印机:填写网络打印机名称

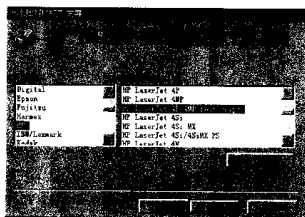


图 7.7 在 Windows 95 中安装 Linux 的共享打印机:选择打印机驱动程序

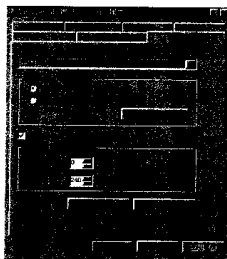


图 7.8 在 Windows 95 中安装 Linux 的共享打印机:修改 PostScript 输出格式

# 8

## Linux 上的中文环境

一种操作系统在中国的推广成效很大程度上取决于运行于它上面的软件对中文的处理能力。Linux 诞生于国外,其历史不到十年,但由于它作为最流行的免费 UNIX 操作系统的独特魅力,很多其他 UNIX 系统上的中文软件已经被移植到 Linux 上,一些专门为 Linux 设计的中文软件也纷纷出现,Linux 上的中文环境正在一步步完善。

本章我们将介绍 Linux 上常用的几种中文软件。Linux 上的中文软件大致分为两种,中文平台和支持使用中文的应用程序。我们将介绍的中文平台及相关软件,包括字符环境下的中文终端,X Window 系统下的中文终端,中文输入服务器,X Window 系统的中文文化包装,X Window 系统中文字库的基本知识等。在支持中文的应用程序中,我们将介绍几种常见的中文编辑器和中文文本处理软件,Netscape 的中文设置等。我们还将介绍中文打印的基本知识以及 X Window 系统中文化的基本原则。最后我们将介绍 Linux 上的外挂式中文平台 ZWinPro。

### 8.1 互联网上的 Linux 中文资源

#### 8.1.1 中文软件资源

IFCSS 是互联网上收集免费中文软件最多最全的地方,它的网址是

<ftp://ftp.ifcss.org/pub/software/>

IFCSS 的中文软件资源在世界各地的镜像节点有:

中国台湾

<ftp://ftp.edu.tw/Chinese/ifcss/software> [全部]

<ftp://ftp.ntu.edu.tw/Chinese/ifcss/software> [全部]

<ftp://iiif.iii.org.tw/Chinese/ifcss/software> [全部]

中国大陆

[北京] <ftp://info.bta.net.cn/pub/software> [部分]

[广东] <ftp://info.gz.gdpta.net.cn/pub/mirror/chnssoftware> [部分]

[上海] <ftp://info.sta.net.cn/pub/software> [部分]

中国香港

<ftp://ftp.cuhk.hk/pub/chinese/ifcss/software> [全部]

新加坡

<ftp://ftp.technet.sg/pub/chinese> [全部]

日本

<ftp://ftp.fedu.uec.ac.jp/pub/china/software/ifess> [全部]

美国

<ftp://cnd.org/pub/software> [全部]

<ftp://phaikon.nmsu.edu/pub/software> [全部]

北京拓林思软件有限公司(TurboLinux, Inc.)的 FTP 服务器是 TurboLinux 简体中文版和繁体中文版的官方节点。它还收集了较多的 UNIX 系统下的中文软件,特别是支持中文 GB 码(和 GBK 码)的软件。它还是存放本书作者及周围的一些致力于 Linux 中文化的朋友开发或改编的中文软件的“大本营”。这个 FTP 服务器的地址是

<ftp://ftp.turboLinux.com.cn/pub/chinese/>

它实际上是本书第一版中提到的站点 <ftp://ftp.ihep.ac.cn/pub/chinese/> 的延续。这个 FTP 服务器还是 gtk, gimp, GNOME, KDE, Slackware, RedHat, GNU, X11R6.4, RFC 等软件的镜像节点。它们的镜像目录放置在

<ftp://ftp.turboLinux.com.cn/mirrors/>

### 8.1.2 中文 Linux 计划

随着 Linux 在中文世界的广泛使用和 Linux 中文化的发展,日前在互联网上已经出现了一些由对 Linux 中文化感兴趣的网友自发建立的与 Linux 中文化相关的计划(project)。

#### 1. 中文 Linux 计划(Chinese Linux Project)(中国大陆)

中文 Linux 计划最初的设想是在清华大学 BBS 由 homer 提出的,又经 olly, macaw, eric, cc, hai 和 dfbb 等的共同倡议,于 1997 年 5 月 5 日正式成立。homer 建立起了中文 Linux 计划的“官方”节点 [cLinux.ml.org](http://cLinux.ml.org), 渐渐地建立起了 HTTP 服务器, 邮递列表(Mailing List), 新闻组(NNTP)服务器和 BBS 等。目前中文 Linux 计划在中文 Linux 世界已经具有一定的影响力。

中文 Linux 计划有一个邮递列表是

[majordemo@clinux.ml.org](mailto:majordemo@clinux.ml.org)

#### 2. Linux 中文文件计划(中国台湾)

Linux 中文文件计划(CLDP)目前正全力进行于 Linux HOWTOs 文件的中文翻译工作,它的网址是

<http://www.Linux.org.tw/CLDP/>

CLDP 的邮递列表是 [ldp-trans@Linux.org.tw](mailto:ldp-trans@Linux.org.tw)。

#### 3. Linux 中文化计划(中国台湾)

这是为建立一个完整 Linux 中文化作业环境为目标的计划。由陈永升发起,日前还在草创阶段,有待大家共同努力。本计划的网页是

<http://chinese.Linux.org.tw/>

负责正式对外发布公告,并提供本计划执行时问题讨论、批评建议与成果发表等工

作

### 8.1.3 Linux 的中文站点

目前在大陆的 Linux 中文站点还不太多,希望更多的朋友为 Linux 在中国的普及和 Linux 中文文化事业的发展作出奉献。

<http://cldp.turboLinux.com.cn/>

这是目前在中国大陆收集有关 Linux 的中文资料最多最全的一个站点。本节点现存各类中文资料超过 100 兆字节,资料收集和维护工作由本书作者之一方汉承担,它是本书第一版中提到的站点 <http://csun01.ihep.ac.cn/Linux/> 的延续(如图 8.1 所示)。



图 8.1 中国大陆最早的中文 Linux 计划主页

<http://cLinux.ml.org/>

这是中文 Linux 计划(中国大陆)的“官方”节点。

另外,中国台湾有很多网站收集了大量的中文(Big5 码)资料。下面是其中的一些网站的地址:

<http://www.Linux.org.tw/>

<http://chinese.Linux.org.tw/>

<http://Linux.cis.nctu.edu.tw/>

<http://Linux.ntic.edu.tw/>

<http://freebsd.ee.ntu.edu.tw/bbs/6/index.html>

[http://cc.shu.edu.tw/~rick/wwwguide/c\\_Linux\\_hopenet.html](http://cc.shu.edu.tw/~rick/wwwguide/c_Linux_hopenet.html)

<http://henry.cis.nctu.edu.tw/~Linuxwww/>

<http://marr.dorm9.nccu.edu.tw/~marr/Comp/PC-UNIX/index.html>

<http://www.phys.ntu.edu.tw/~cwhuang/pub/os/Linux/>

### 8.1.4 开设 Linux 的中文讨论区的 BBS 站点

目前在大陆和台湾都有很多中文 BBS 站点开设了 Linux 的中文讨论区,下面列出其中一些(事实上,几乎每一个高等院校的 BBS 站点上都有 Linux 讨论区):

清华大学水木清华站:bbs.net.tsinghua.edu.cn (202.112.58.200)

中国科学院曙光站:bbs.ncic.ac.cn (159.226.43.21)

上海交通大学饮水思源站:bbs.sjtu.edu.cn (202.120.2.114)

暨南大学华南木棉站:melon.gznet.edu.cn (202.112.17.37)

中国科技大学站:bbs.ustc.edu.cn (202.38.64.3)

广州电信局蓝天站:bbs.gznet.com (202.96.151.222)

物理学会 physbbs.iam.s.sinica.edu.tw

东吴计中 bbs.scu.edu.tw

中央电机 bbs.cc.eccu.edu.tw

交大资料 bbs.cis.nctu.edu.tw;9001-9009

清大资料 bbs.cs.nthu.edu.tw

清大电机 bbs.ee.nthu.edu.tw

云技计中 bbs.yuntech.edu.tw

成大计中 bbs.ncku.edu.tw

中山计中 bbs.nsysu.edu.tw

SOB 电机分站:freebsd.ee.ntu.edu.tw

### 8.1.5 开设 Linux 的中文新闻组的新闻服务器

大陆地区有

news.clinux.ml.org;cn.bbs.comp.Linux

news.freessoft.cei.gov.cn;chinese.comp.Linux

台湾地区有

news.csie.nctu.edu.tw;tw.bbs.comp.Linux

## 8.2 中文字符集,编码和编码转换程序

### 8.2.1 中文字符集及编码

目前在大陆和香港台湾等地使用着不同的中文字符集标准。中国大陆最常用的基本中文字符集是 GB 2312-80。GB 指“国标”,即中国的国家标准。这个中文(简体)字符集标准包括 7445 个字符,其中 6763 个是汉字。汉字分为两级,第一级汉字按音序排列,第二级汉字先按部首后按剩余笔画排列。它的字符排列如下:

1 区:94 个符号

2 区:72 个数字

3 区:94 个满宽度 GB 1988-89 字符(对应于可打印的 ASCII 字符)

4 区:83 个日文平假名

- 5 区:86 个日文片假名
- 6 区:48 个大写和小写希腊字母
- 7 区:66 个大写和小写斯拉夫(俄语)字母
- 8 区:26 个拼音和 37 个注音字符
- 9 区:76 个制表符(09-04 到 09-79)
- 16 区 — 55 区:3755 个汉字(一级汉字;最后 1 个是 55-89)
- 56 区 — 87 区:3008 个汉字(二级汉字;最后 1 个是 87-94)

字符集 GB/T 12345-90 在字符的数量和安排上几乎等同于 GB 2312-80,但所有的简体字都用对应的繁体字替代了。

字符集 GB 1988-89 对应于 ASCII 字符或 ISO 646,但现金符号(0x24)用中国的人民币符号取代了 ASCII 字符集中的美元符号(\$)。

中文字符集的编码一般是双字节码,对于每个字符集标准都可能几种编码体系。对于 GB 2312-80 字符集,常用的编码有 EUC 和 HZ。EUC 表示“Extended UNIX Code”,是在 ISO 2022-1993 中定义的一种编码体系,用来处理大的字符集。它在各种 UNIX 系统上被广泛地用来对各种语言的字符集进行编码。EUC 包括 4 个代码集(code set),编号为 0 到 3。EUC 在中国大陆字符集的实现如下:

| EUC 代码集                    | 编码范围          |
|----------------------------|---------------|
| 代码集 0 (ASCII 或 GB 1988-89) | 0xA1-0x7E     |
| 代码集 1 (GB 2312-80)         | 0xA1A1-0xFFFE |
| 代码集 2                      | 未使用           |
| 代码集 3                      | 未使用           |

我们经常把 GB 2312-80 字符集的 EUC 编码称为国标码或 GB 码。我们看到,GB 码的第一个字节是 0xA1-0xFE,其中 0xA1-0xF7 对应 GB 2312-80 字符集的 87 个区,0xF8-0xFE 没有定义;第二个字节是 0xA1-0xFE,对应每个区最多 94 个字符。GB 码的两个字节都是 8 位码(>0x7F),不会同 ASCII 码(≤0x7E)相混淆,这一点对中文处理非常有利,如像本文这样 ASCII 字符和中文字符混排的文本,我们很容易根据 8 位码的出现确定它们之间的界限,而后用不同的方法处理 ASCII 字符串和中文字符串。

但是,有时候我们可能需要用明显的界限把 ASCII 字符串和中文字符串分开,如发送邮件或向 USENET 的中文新闻组(如 alt.chinese.text)投递稿件时。因为有些电子邮件服务器和 NNTP 服务器在接收和处理 8 位码时可能会有困难。现在有多种方法可以解决使用网络发送和接收 8 位码的问题,对于中文字符来说其中有一种方法就是使用 HZ (HZ-GB-2312)编码体系。在 HZ 码中,GB 码的两个字节的最高位都被置为 0,中文字符编码的第一个字节被限制在 0x21-0x77,第二个字节的范围是 0x21-0x7E;在中文字符串与 ASCII 字符串之间用转换序列隔开。HZ 码使用由两个可打印字符组成的简单字符串作为转换序列:

单字节字符集

转换序列

十六进制代码

|            |       |        |
|------------|-------|--------|
| ASCII      | ~     | 0x7E7D |
| 双字节字符集     | 转换序列  | 十六进制代码 |
| -----      | ----- | -----  |
| GB 2312-80 | ~     | 0x7E7B |

在 HZ 码中,波浪符(0x7E)被解释为逃逸字符,所以它有特殊的意义。如果在单字节字符模式下要使用波浪符,必须连用 2 个,即~~实际表示~。这意味着在 HZ 码中有 3 个逃逸序列:

| 逃逸序列 | 十六进制代码 | 意 义         |
|------|--------|-------------|
| ~~   | 0x7E7E | 单字节字符模式下的 ~ |
| ~    | 0x7E7D | 转换到单字节字符模式  |
| ~    | 0x7E7B | 转换到双字节字符模式  |

实际上还有第 4 个逃逸序列,~加上一个换行符(~\n),它是个简单的续行标志,不产生输出。因为这几个逃逸序列的编码不对应于 GB 2312-80 字符集中已定义的任何双字节字符,所以使用它们不会遇到问题。

另一个广为人知的中文字符集是 GB 13000.1-93,它实际上是 ISO 10646-1:1993 (Unicode Version 1.1) 的中文翻版。我们经常用到的是它的一个中文特定子集 GBK。

GBK (“K”是“扩展”的汉语拼音的第一个字母),是 GB 2312-80 的一个扩展,它同时容纳了 ISO 10646-1:1993 (GB 13000.1-93) 中其他的中文字符。GBK 由以下字符组成:

GB 2312-80(包含 GB 6345.1-86 中指定的修正和增加部分)

GB/T 12345-90 中的非汉字部分

14,240 个增加的汉字

166 个增加的符号

GBK 在逻辑上可以分为 5 级

| 级别    | 字符数   | 内 容                                      |
|-------|-------|------------------------------------------|
| GBK/1 | 717   | GB 2312-80 和 GB/T 12345-90 中的非汉字字符       |
| GBK/2 | 6,763 | GB 2312-80 汉字                            |
| GBK/3 | 6,080 | ISO 10646-1:1993 中的汉字                    |
| GBK/4 | 8,160 | ISO 10646-1:1993 中的 8,059 个汉字和 101 个其他汉字 |
| GBK/5 | 166   | Big Five 中的非汉字字符,其他字符                    |

从 GBK/2 到 GBK/4 共有 21,003 个汉字。整个 GBK 中文字符集共有 21,886 个字符。

GBK 的编码是对 GB 2312-80 的 EUC 编码的扩展

| 级别    | 编码范围            | 代码点数  | 编码字符数 |
|-------|-----------------|-------|-------|
| GBK/1 | 0xA1A1 — 0xA9FE | 846   | 717   |
| GBK/2 | 0xB0A1 — 0xF7FE | 6,768 | 6,762 |
| GBK/3 | 0x8140 — 0xA0FE | 6,080 | 6,080 |
| GBK/4 | 0xA440 — 0xFE40 | 8,160 | 8,160 |
| GBK/5 | 0xA840 — 0xA940 | 192   | 166   |

大致地说,GBK 的编码范围是:

第一个字节: 0x81 — 0xFE

第二个字节: 0x40 — 0x7E, 0x80 — 0xFE

GB 2312-80 的 EUC 编码(0xA1A1-0xFEFE)实际上是 GBK 编码的一个子集。

注意 GB 2312-80 的 EUC 编码(GB 码)和 GBK 的编码(GBK 码)的一个最显著的不同是,GB 码的两个字节都是 0x80 以上的 8 位码,而 GBK 码的第二个字节可以是 0x80 以上的 8 位码,也可以是小于是 0x80 的 7 位码。另外,GBK 码的第一个字节范围(0x81 — 0xFE)比 GB 码(0xA1 — 0xFE)宽。在中文应用程序中处理 GBK 码必须采用与只处理 GB 码的应用程序不太相同的方法。

日前在台湾等地常用的中文字符集有大五(Big Five)字符集和 CNS 11643-1992。我们在这里只介绍大五字符集。大五字符集由 94 个区组成,每个区有 157 个字符,分为两组,分别有 63 个和 94 个字符。下面是它的字符分类

1 区:157 个符号

2 区:157 个符号

3 区:94 个符号

4 区 — 38 区:5401 个汉字(一级汉字;最后一个 38-63)

41 区 — 89 区:7652 个汉字(二级汉字;最后一个 89-116)

大五字符集的编码体系一般称为大五码或 Big5 码。它的编码规则是:

| 双字节字符 | 编码范围                 |
|-------|----------------------|
| ----- | -----                |
| 第一个字节 | 0xA1-0xFE            |
| 第二个字节 | 0x40-0x7E, 0xA1-0xFE |
| 单字节字符 | 编码范围                 |
| ----- | -----                |
| ASCII | 0x21-0x7E            |

Big5 码的第一个字节中从 0xA1-0xF9 对应大五字符集的 89 个区,第二个字节 0x40-0x7E 和 0xA1-0xFE 分别对应每个区的两组字符。我们看到,Big5 码的第一个字节都是 8 位码,第二个字节中第二组是 8 位码,而第一组却是 7 位码,与 ASCII 码重合。这与 GBK 码相似。这就要求软件在处理 Big5 码时要比处理 GB 码有额外的判断来区分中文字符串和 ASCII 字符串。

现在我们看一个例子。对于字符串“1 尺 3 寸”,它在 GB, HZ 和 Big5 3 种编码体系中

的编码如下(字符编码用 16 进制表示,但逃逸序列为清晰起见用可打印字符表示):

| 编码体系 | 编码     |          |        |          |
|------|--------|----------|--------|----------|
| GB   | 0x31   | 0xB3DF   | 0x33   | 0xB4E7   |
| HZ   | 0x31 ~ | 0x335F ~ | 0x33 ~ | 0x3467 ~ |
| Big5 | 0x31   | 0xA4D8   | 0x33   | 0xA46F   |

### 8.2.2 编码转换程序

由于在中文世界同时使用着多种字符集和多种编码体系,我们经常会需要对文本进行编码转换。如把 GB 码的文本转换为 HZ 码,或 HZ 码转换为 GB 码;把 GB 码文本转换为对应的繁体字的 Big5 码,或作相反的转换,等等。网络上有些免费软件可以进行这种转换。编码转换的原理大都是先区分界定中文字符串和 ASCII 字符串,ASCII 字符串保持不变,对中文字符串将其编码作对应的替换,必要时加上或去掉逃逸序列。下面介绍的软件都可从 [ftp.ifess.org](http://ftp.ifess.org) 或其镜像 FTP 节点的 `./UNIX/convert` 目录或从

[ftp://ftp.turbolinux.com.cn/pub/chinese/packages/](http://ftp.turbolinux.com.cn/pub/chinese/packages/)

目录下载。另外,Unicode 与 GB 码,GBK 码,Big5 码的转换对应表可从 IFCSS 节点的 `./data` 目录下载。

`HZ-2.0.tar.gz`

这个软件包中包含 GB 码和 HZ 码相互转换的程序 `gb2hz` 和 `hz2gb`。它们的使用方法非常简单,输入和输出分别是 `stdin` 和 `stdout`:

```
lark% gb2hz < GB 码文件 > HZ 码文件
lark% hz2gb < HZ 码文件 > GB 码文件
```

`hc-30.tar.gz`

这个软件包中包含 GB 码和 Big5 码相互转换的程序 `hc`。GB 码与 Big5 码的相互转换通常是建立一个汉字的 GB 码和 Big5 码的对照表,转换程序通过查找对照表把汉字的 GB 码转换为 Big5 码,或 Big5 码转换为 GB 码。软件包中的 `hc.tab` 就是这样一个对照表。软件编译后可产生两个 shell 脚本 `g2b` 和 `b2g`,其输入和输出分别是 `stdin` 和 `stdout`:

```
lark% g2b < GB 码文件 > Big5 码文件
lark% b2g < Big5 码文件 > GB 码文件
```

`hztty-2.0.tar.gz`

这是一个把中文终端的输入输出从一种编码转换成另外一种编码的工具。它用户可以在支持一种编码的终端上以另一种编码格式读写中文。如下文中将要介绍的字符环境下的中文终端 `cee` 只支持 GB 码,如果我们想阅读及编辑 Big5 码的文件,就可以使用 `hztty` 对 `cee` 的输入输出进行转换:

```
lark% cce                                # 启动中文环境 cce
lark% hztty -I gb2big -O big2gb
[hztty started] [using /dev/tty3]
```

```

# -l gb2big:把 GB 码输入转换为 Big5 码
# -O big2gb:把 Big5 码输出转换为 GB 码
# 编辑 Big5 码的文件
# 退出 hztty

lark% vi big5.txt
lark% exit
exit

[hztty exited]
lark%

```

hztty 实际上是在原 shell 下重起了一个新的 shell, 这个新的 shell 完成输入输出的编码转换。hztty 可支持下列编码之间的输入输出转换: GB, HZ, Big5, UTF-8, UTF-7, Unicode(16 位)。

## 8.3 字符环境下的中文模拟终端——chdrv, cce, kon

### 8.3.1 chdrv

chdrv 是运行在 VGA/SVGA 控制台上的中文终端模拟程序。chdrv 由台湾的王佑中 (wycc@iis.sinica.edu.tw) 维护, 最新的版本可从

<ftp://Linux.cis.nctu.edu.tw/packages/chinese/chdrv/>

下载, 源程序 chdrv-x.x.xx.tar.gz (x.x.xx 是版本号), 已编译的二进制执行文件 chdrvbin-x.x.xx.tar.gz, 以及中文字库 chdrvfont.tar.gz。

chdrv 因为直接读写 tty 设备, 所以必须由 root 来启动。chdrv 启动后在 tty7 (用 Alt-F7 或 Ctrl-Alt-F7) 建立一个虚拟终端, 用户在 tty7 上登录进系统后就可以使用 chdrv 提供的中文支持。而 tty1-tty6 和 X Window 系统不受 chdrv 的影响。

root 可以在任何时候用 “chdrv -k” 命令使已运行的 chdrv 退出。

chdrv 支持 GB 码, GB 码环境使用 HBF 字库, 用户参照 chdrv 软件包中的有关说明在 Big5 码和 GB 码环境间转换。本书著者之一(方汉)曾将 chdrv 修改为专门支持 GB 码的版本, 并改进了其中文 GB 码的输入法。用户可从

<ftp://ftp.turboLinux.com.cn/pub/chinese/system/>

目录取得这个软件。

### 8.3.2 cce

Linux 终端中文环境 cce(Console Chinese Environment)运行在 VGA/SVGA 控制台的虚屏上, 它将当前的虚屏变成一个中文字符彩色模拟终端, 用户可以在其中进行中文信息处理。

cce 由 raner(herui@info.cs.tsinghua.edu.cn)参考 yact, bes16, chdrv 等中文控制台程序开发, 用户可从

<ftp://ftp.turboLinux.com.cn/pub/chinese/system/>

目录取得这个软件。编译安装后, 以 “cce” 命令启动中文环境 (如图 8.2 所示)。

在 cce 下, 用 Control-Space 切换中英文输入状态, 用 Control-Alt-0 到 Control-Alt-9 切

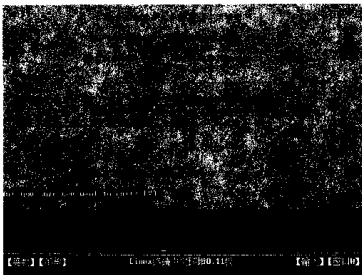


图 8.2 Linux 终端中文环境 cce

换中文输入法。cce 缺省将内码、全拼和五笔输入法分别加载为输入法 0、输入法 1 和输入法 2。cce 还带有双拼和自然码输入法,每种输入法都支持联想功能。用户可以在 cce 启动时在命令行加载自己需要的输入法,如:

```
lark% cce -in3 shuangpin -in4 ziranma
```

用户也可以方便地增加新的输入法,只要造一个文本方式的 .cin 文件,并用 cce 提供的 cin2tab 程序将它转换成 cce 可以识别的 .tab 格式。tab2cin 程序可以完成 .tab 到 .cin 的转换, cce 的一个有趣的特征是,它支持多个虚拟控制台,如:

```
Control-Alt-A  增加一个虚拟控制台
Control-Alt-K  退出当前的虚拟控制台
Control-Alt-N  转到下一个虚拟控制台
Control-Alt-P  转到上一个虚拟控制台
```

### 8.3.3 kon

kon 是运行在 VGA/SVGA 上的终端模拟程序。kon 由 Takashi MANABE 开发,由本书的著者之一(方汉)移植到中文,并加进对中文输入法的支持。kon 的中文版(kongb)的最新版本可从

```
ftp://ftp.turboLinux.com.cn/pub/chinese/system/
```

目录取到。

kon 较之其他中文模拟终端的优点在于它的显示速度非常快。kon 的另外一个特点是它支持鼠标。

kon 的中文版支持内码、全拼、五笔、双拼、自然码等输入法。它的输入法文件格式与 cce 相同,用户也可以自己增加新的输入法。用户在 kon 里激活和改变输入法的快捷键也与 cce 相同。

## 8.4 X Window 系统下的中文终端——CXterm, crxvt

### 8.4.1 CXterm

CXterm 可能是 X Window 系统下最常用的中文模拟终端。它是从 X Window 系统的标准应用程序 xterm 修改而来,但增加了显示和输入中文的功能。它支持中文的 GB 码和 Big5 码,日文的 JIS 码和韩文的 KS 码,并可通过 hztty 的转换而支持 HZ 码等其他中文编码。

下面介绍 CXterm 在 Linux 下的编译和安装,对用户在安装和使用 CXterm 时经常会碰到的问题作了着重介绍。

#### 1. 编译安装

为保证 CXterm 的正确顺利安装,建议以下步骤应该在 X 界面下的 xterm 中完成,因为 CXterm 在配置和安装过程中,会通过向 X 服务器查询来获得系统的有关信息,特别是关于系统中文字库的信息。用户可以先用一个小的 X 程序试试程序是否可以打开 DISPLAY。如:

```
lark% xset q
```

如果用户需要把 CXterm 的执行文件及相关文件安装到系统目录下,必须在 root 帐号下编译安装。

首先,从 IFCSS 或其镜像节点的 ./x-win/cxterm/目录下取来 CXterm 的源程序包。需要的文件有:

```
cxterm5.0.p3.tar.gz  
cxterm-5.0.p3-color.patch.gz
```

第一个文件是 CXterm 的原始软件包,它实现了单色的中文模拟终端;第二个文件是对 CXterm 的彩色修正的“补丁”文件,它使 CXterm 可以显示各种色彩,如用 ls 命令列出文件名时。特别是当我们用 CXterm 连接中文 BBS 站点时,各种显示色彩可以使界面更生动,可读性更强。

在某个工作目录下,展开软件包:

```
lark # tar zxvf cxterm5.0.p3.tar.gz
```

软件将展开在子目录 cxterm-5.0/下。

接着,在同一工作目录下,用“补丁”文件对源程序目录树作彩色修正:

```
lark # gzip -cd cxterm-5.0.p3-color.patch.gz | patch -p0 -s
```

上述命令也可以用下面两条命令代替:

```
lark # gzip -d cxterm-5.0.p3-color.patch.gz  
lark # patch -p0 -s < cxterm-5.0.p3-color.patch
```

patch 的“-s”选项表示除非出错,否则 patch 命令将不输出屏幕信息。

确定 X Window 系统的可执行文件的目录在命令搜索路径上,即环境变量 PATH 中包含 /usr/X11/bin(或 /usr/X11R6/bin),CXterm 的安装过程要用到 xmkmf, bdf2pcf 等命令。然后进入 cxterm-5.0/目录,执行 config.sh,屏幕输出如下:

```
lark# cd xterm-5.0
lark# ./config.sh
```

CXTERM VERSION 5.0 by Yongguang Zhang (C) 1995

Please read the COPYRIGHT file on the copyright and permission Notices. If you do not agree the terms, stop by pressing < Ctrl-C > .

To make things smooth, you'd better run this configuration program under X Window. You may type < Ctrl-C > any time to stop the configuration.

You need to answer several questions, so don't run this program in background or redirect the output. Multiple choices are listed in parentheses (), and default or suggested answers are given in brackets []. The error messages during the make will be saved automatically in "/install.log".

Continue with config.sh (y/n) [y] ? y

-----  
--- BASIC MENU ---

0. Read COPYRIGHT Notice

1. Compile, Install, and Configure "CXTERM 5.0" in One Step

--- OPTION MENU ---

2. Compile xterm (Not to install)

3. Install xterm (after successful compilation in 2)

4. Install additional Chinese font(s) for your X window

5. Configure your account for using xterm (after installation in 3)

x. Exit

-----  
Please choose (0/1/2/3/4/5/x):

强烈建议用户选择菜单中 2-3-5 的步骤,如果选择把编译安装配置在一步完成(选项 1),则当其中某一步出现错误时很难查找原因。

在上面 "Please choose (0/1/2/3/4/5/x) :" 后键入 "2", 则 config.sh 执行如下命令

```
make -f Makefile.ini World 2 > &1 | tee Install.log
```

上述命令按如下步骤编译 CXterm:

```
xmkmf
make xrelease
make Makefiles
```

```
make includes
make
```

除“make xrelease”外,以上是 X 应用程序编译的一般步骤,“make xrelease”这里用来确定系统使用的 X Window 系统的版本号(X11R5 或 X11R6)。“make”将编译可执行文件 `exterm` 和工具程序 `hzimctrl`, `tit2cit`, `cit2tit`, 并对字典文件和字库进行格式转换。编译完成后, `config.sh` 程序将回到菜单选项。

如果一切正确,可执行步骤 3 来安装编译好的程序:

```
Please choose (0/1/2/3/4/5/x) :3
```

You may install `exterm` in the default X window directory or a directory of your own choice. If you are a system manager and wish to make `exterm` available to all the users, you may choose to install `exterm` in the X window system directory, along with other X applications. Or, if you do Not have the privilege or wish to place it in a separately place, you may install `exterm` to any directory of your choice.

1. Install in directories of your choice.
2. Install in the system X Window directories (may need "root" access).
0. Exit `exterm` installation.

```
Please choose (1,2,0) [1] :
```

这里, `config.sh` 提供了两种选择:安装到用户指定的目录中或安装到 X Window 系统的缺省目录中。由于 `CXterm` 安装程序本身的缺陷,选择第二种方法将导致安装过程中的一些麻烦而可能影响 `CXterm` 的正确使用。因此,我们建议使用第一种安装方法,即使用户决定把 `CXterm` 安装到系统范围的目录中。

例如,我们要把 `CXterm` 安装到 `/usr/local` 下各个子目录中,可以进行如下操作:

```
Please choose (1,2,0) [1] : 1
```

Where do you want to put the executable binary files for `exterm`?

I would suggest `/root/bin`. You may choose any directory.

Enter the full path name [`/root/bin`] : `/usr/local/bin`

Where do you want to put the dictionary files for `exterm`?

I would suggest `/root/dict`. You may choose any directory.

Enter the full path name [`/root/dict`] : `/usr/local/lib/exterm.dic`

Where do you want to put the `exterm` manual pages?

I would suggest `/root/man`. You may choose any directory.

Enter the full path name [`/root/man`] : `/usr/local/man`

Where do you want to put the xterm documentation?

I would suggest /root/doc/xterm. You may choose any directory.

Enter the full path name [/root/doc/xterm] : /usr/local/doc/xterm

接下来,安装程序将检测系统上是否已经安装中文 GB 码和 Big5 码的字库,并建议用户是否安装相应字库。然后是文件安装过程。可执行文件、字典文件、在线帮助文件(man 页面)和文档将被安装到用户指定的目录中。

安装程序将产生一个 shell 脚本文件 CXterm,并把它安装到/usr/local/bin 目录中,CXterm 文件中设置了正确运行可执行文件 xterm 所必需的 X 资源并最终调用 xterm 运行中文模拟终端。所以用户应该运行脚本文件 CXterm 而不要直接运行 xterm,否则 xterm 将无法正确设定 X 资源而无法输入或显示中文。

config.sh 的菜单中步骤 5 是设置用户帐号使用 xterm。它实际上只是把字典目录(在上面的例子中是/usr/local/lib/cxterm.dic)中的资源文件 CXterm.ad 附加到当前用户(即执行 config.sh 的用户)根目录下的.Xdefaults 文件后。如果用户的.Xdefaults 文件中包含了 CXterm.ad 文件,用户就无须通过 shell 脚本文件 CXterm 来运行 xterm,而可以在命令行直接输入命令 xterm。但是,运行脚本文件 CXterm 可能更加方便一些,因为它不要求每个用户都必须把 CXterm.ad 加到自己的.Xdefaults 文件中。

用户当然可以选择除/usr/local/...外的其他目录来安装 CXterm,无论如何,最终应该把脚本文件 CXterm(或执行文件 xterm)所在的目录加进命令搜索路径环境变量 PATH 中去。

CXterm 的运行示例如图 8.3 和 8.4 所示。

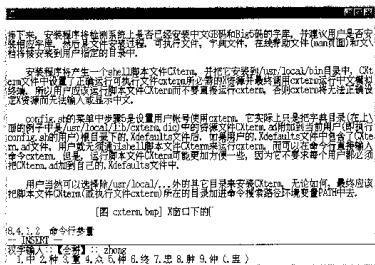


图 8.3 X Window 下的中文模拟终端 CXterm

## 2. 命令行参量

CXterm(xterm)的缺省资源存放在文件 CXterm.ad 中,在 CXterm 启动时自动加载。CXterm(xterm)同样也可以在命令行重新定义一些资源。标准的 X Toolkit 命令行参量和 xterm 的命令行参量同样适用于 xterm。CXterm(xterm)特有的或常用的命令行参量有以下一些:

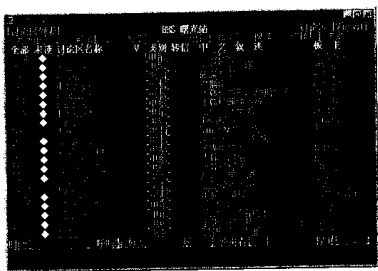


图 8.4 在 Cxterm 中登录 BBS

-fu < 英文字库 >

指定显示用英文字库

-fh < 中文字库 >

指定显示用中文字库,其字符宽度应为英文字符的两倍,高度应与英文字符相当

-fnb < 中文字库 >

指定显示用黑体中文字库,其宽度和高度应与“-fh”指定的标准体中文字库一致。如果只指定了标准体和黑体中文字库中的一种,则它将被用作标准体字库,而黑体字符将用标准体重复打印的方法实现:

-hm < 输入法 >

指定 exterm 启动时的初始输入法。缺省是“ASCII”,即 exterm 以英文模式启动

-hz < 编码 >

指定 exterm 的编码体系,GB 或 BIG5。缺省是 GB

-GB

表示 exterm 使用 GB 码。同“-hz GB”

-BIG5

表示 exterm 使用 Big5 码。同“-hz BIG5”

-hid < 汉字输入法目录 >

指定包含汉字输入法的目录的搜索路径,多个目录之间用“:”隔开

-hls < 行间距 >

指定 Cxterm 窗口文本区邻近两行的间距(以像素为单位)

### 3. 输入法

Cxterm 的窗口除包含有通常 Xterm 的文本区外,在窗口的底部还有一个输入区,用以显示中文输入信息。Cxterm 对每一种编码都提供了几种输入法,如对 GB 码有无调拼音、带调拼音、缩写拼音、五笔字型、区位码、内码、英汉等十多种输入法,每种输入法都可

以用热键 F1 到 F11 或 Shift-F1 到 Shift-F11 激活,如 F6 激活无调拼音、F8 激活五笔字型、Shift-F2 激活区位码,而 F1 总可在当前输入法和 ASCII 码状态之间转换。

CXterm 的输入法储存在外部文件中,在需要时由 `cxterm` 加载进内存。输入法文件有两种格式,文本格式和编译格式。文本格式的文件扩展名是 `.tit`(Text Input Table),是用户可编辑的;编译格式的文件扩展名是 `.cit`(Compiled Input Table),由 `cxterm` 在运行时加载。`tit2cit` 和 `cit2tit` 两个工具可以对输入法文件进行这两种格式之间的相互转换。

CXterm 开放式的输入法使得我们可以比较方便地构造我们喜欢使用的输入法的 `tit` 文件。例如,由于 UCdos 的输入法是结合中国人使用习惯,经过不断改进的一种优秀的输入法,我们希望在 CXterm 中使用 UCdos 的输入法。我们可以用 UCdos 5.0 以上版本提供的 `imdmng` 程序来帮助生成 `tit` 文件。以 UCdos 的拼音输入法 `py` 为例:

(1) 首先在 DOS 下,到 UCdos \ DRV 目录下运行 `imdmng`:

```
C:\UCDOS\DRV>IMGMNG PY.IMD PY.TXT
```

(2) 编辑 `PY.TXT`,删除前 34 行注解,加入以下内容,存成文件 `upy.tit`:

```
# $Id: upy.tit,v 1.3 1995/01/02 06:07:48 ygz Exp $
# HANZI input table for cxterm
# To be used by cxterm, convert me to .cit format first
# .cit version 2
ENCODE: GB
PROMPT: 汉字输入::UCDOS 拼音:: \ 040
AUTOSELECT:      NEVER
#
COMMENT {源于 UCdos}
# input key definitions
VALIDINPUTKEY:    abcdefghijklmNopqrstuvwxyz
WILDCARDKEY:      *
WILDCHARKEY:      ?
ASSOCIATIONKEY:   -
# choice list keys
SELECTKEY:        1 \ 040
SELECTKEY:        2
SELECTKEY:        3
SELECTKEY:        4
SELECTKEY:        5
SELECTKEY:        6
SELECTKEY:        7
SELECTKEY:        8
SELECTKEY:        9
SELECTKEY:        0
MOVERIGHT:        . > = +
MOVELEFT:         , < - _
# pre-editing keys
```

```

BACKSPACE:      \ 010 \ 177      # Ctrl-H, DEL
DELETEALL:      \ 015 \ 025      # RETURN, Ctrl-U
CURSOR-BEGIN:   \ 001            # Ctrl-A
CURSOR-END:     \ 005            # Ctrl-E
CURSOR-FORW:    \ 006            # Ctrl-F
CURSOR-BACK:    \ 002            # Ctrl-B
CURSOR-ERASE:   \ 004            # Ctrl-D
REPEATKEY:      \ 020 \ 022      # Ctrl-P, Ctrl-R
# the following line must Not be removed
BEGINPHRASE
#

```

(3) 拷贝 `upy.tit` 到 Linux 下,用 `tit2cit` 将其转换成 `CXterm` 可以使用的输入码表:

```
lark% tit2cit upy.txt > upy.cit
```

把 `upy.cit` 拷贝到 `CXterm` 的字典目录下的 `gb/` 子目录中,然后修改 `CXterm.ad`,定义激活 `upy` 输入法的热键就可以了。如果想用 `upy.cit` 替换 `CXterm` 中自带的 `PY.cit`,只须把 `upy.cit` 拷贝到 `gb/` 目录下覆盖 `PY.cit` 就可以了,而不必修改 `CXterm.ad`。

#### 4. shell 脚本文件 `CXterm` 和资源文件 `CXterm.ad`

shell 脚本文件 `CXterm` 是启动中文模拟终端 `cxterm` 的重要文件,它先根据资源文件 `CXterm.ad` 设置 `cxterm` 的 X Window 资源,然后调用 `cxterm` 启动中文模拟终端。用户在使用 `CXterm` 时经常遇到的很多问题就源于这两个文件在 `CXterm` 安装过程中没有设置正确。

本节我们对 shell 脚本文件 `CXterm` 和资源文件 `CXterm.ad` 作一些简单的分析,以帮助用户理解 `CXterm` 的运行过程和使用方法。

我们先看 shell 脚本文件 `CXterm`。它首先设置了几个环境变量:

```

#! /bin/sh
BINDIR = /usr/local/bin
XFONTDIR = /home/chenxy/xfonts
CXTERM_AD = /usr/local/lib/cxterm.dic/CXterm.ad
# ENCODE_FONTS:      automatically search for a font from the list
GB_FONTS = "hanzigb16st cclib16st beijing16"
BIG5_FONTS = "hku16st taipei16 taipei15"

```

其中 `CXTERM_AD` 很关键,它给出了资源文件 `CXterm.ad` 的位置;如果在 `CXTERM_AD` 指定的位置没有这个文件,`cxterm` 启动时将无法设置输入法字典目录和热键等资源,用户将无法使用各种输入法;这与直接运行 `cxterm` 而没有把 `CXterm.ad` 加进用户根目录下的 `Xdefaults` 文件的错误是一样的,如用户按 `F6` 键想激活“无调拼音”输入法时 `cxterm` 将无法理解用户 `F6` 键的意思。

接着,`CXterm` 定义了一个命令行参量。用户可以用这个参量来确定 `cxterm` 运行在哪种编码模式下(缺省是中文 `GB` 码)。如“`CXterm -b5`”表示 `cxterm` 使用 `Big5` 码,而“`CXterm-jis`”则使 `cxterm` 运行于日文 `JIS` 码模式:

```
NAME = cxterm
case "$1" in
    -gb * | -GB * )      NAME = cxterm; shift 1 ;;
    -big5 | -Big5 | -BIG5 | -b5 | -B5 )  NAME = cxterm; shift 1 ;;
    -jis | -JIS )        NAME = cxtermjis; shift 1 ;;
    -ks * | -KS * )      NAME = cxtermks; shift 1 ;;
esac
```

下面的一大段是设置 cxterm 使用的中文(或日文,韩文)字库。check\_use\_font() 是检查系统是否存在某字库的小程序:

```
check_use_font ()
{
    for f in "$@" ; do
        if [ "$(xlsfonts $f 2>/dev/null" = "$f" ] ; then
            echo $f
            return 0
        fi
    done
    xset fp+ "$XFONDIR"
    for f in "$@" ; do
        if [ "$(xlsfonts $f 2>/dev/null" = "$f" ] ; then
            echo $f
            return 0
        fi
    done
    return 1
}
```

如果用户用“-fh”选项给出了字库名,cxterm 将使用用户指定的字库:

```
CF=""
FH=0
for i in "$@" ; do
    if [ "$i" = "-fh" ] ; then
        FH=1; continue;
    fi
    if [ "$FH" = "eq 1" ] ; then
        CF='check_use_font $i'
    fi
    if [ "$CF" = "" ] ; then
        echo "Specified font doesn't exist in the X Window system."
        exit 1
    fi
done
break
fi
FH=0
```

done

如果用户没有用“-fh”选项指定字库名,而用户指定使用 GB 码或 Big5 码,程序将检查系统中是否有上面定义的变量 GB\_FONTS 或 BIG5\_FONTS 中指定的字库,并使用最先找到的一种;如果所有的字库都没有找到,程序将退出:

```
if [ "$CF" = "" ]; then
    if [ "$NAME" = "extern" ]; then
        CF='check_use_font $GB_FONTS'
    if [ "$CF" = "" ]; then
        echo "CanNot identify a chinese font in the X Window system."
        exit 1
    fi
    FH_OPT="-fh $CF"
    elif [ "$NAME" = "externb5" ]; then
        CF='check_use_font $BIG5_FONTS'
    if [ "$CF" = "" ]; then
        echo "CanNot identify a chinese font in the X Window system."
        exit 1
    fi
    FH_OPT="-fh $CF"
fi
fi
```

如果用户没有在命令行指定字库,或指定的字库没有找到,且用户使用的是日文 JIS 码或韩文 KS 码,则 extern 将使用在 CXterm.ad 文件中“\* hanziFont”资源给出的字库。

由于 extern 程序本身的缺陷,它不能识别在命令行参量中给出的含有“-”字符的长字库名,如若在“-fn”参量中使用

```
-cclib-song-medium-r-Normal--16-160-75-75-c-160-gb2312.1980-0
song-medium16
```

等字库名,cxterm 会说

```
cxterm: unable to locate a suitable font
```

而退出。如果我们想使用这些字库,可以先在相应的字库目录中的 fonts.alias 文件里给它们加上别名,如

```
ccs16 -cclib-song-medium-r-Normal--16-160-75-75-c-160-gb2312.1980-0
```

执行“xset fp rehash”命令重新读入字库数据库,然后就可以使用 ccs16 字库了:

```
CXterm -fn ccs16
```

在 shell 脚本文件 CXterm 的最后,设置了环境变量 XENVIRONMENT(X Window 系统的应用程序从这个变量指定的文件中读取 X 资源)并执行 cxterm:

```
XENVIRONMENT= $CXTERM_AD
export XENVIRONMENT
exec $CXTERM -name $NAME $FH_OPT $*
```

现在我们来分析资源文件 CXterm.ad。CXterm.ad 文件中定义了 cxterm 窗口的各

种属性,如大小、颜色、键定义、字库等。它是从 xterm 的资源文件修改得来的。我们这里只分析 xterm 特有的部分。

CXterm.ad 首先定义了窗口的背景和前景颜色,用户可以修改成自己喜欢的颜色。然后对 xterm 支持的每一种编码,CXterm.ad 用单独的一段来定义各种字库、输入法字典文件目录和激活输入法的热键。下面分析对应 GB 码的一段:

```
!
! resource for GB xterm (Simplified Chinese as used in Mainland China)
!
xterm * HanziEncoding:          GB
xterm * hanziInputDir:          /usr/local/lib/cxterm.dic/gb
```

xterm \* hanziInputDir 定义了输入法字典目录的位置,如果它指向了一个错误的目录或目录中不存在某一输入法的字典文件,当用户用对应热键想激活这种输入法时,在 xterm 窗口下部的输入区会有类似下面的语句出现:

```
Unable to locate the input table for PY
```

对应输入法无法激活。下面一句

```
xterm * hanziAssociation:      simple.lx
```

定义联想字典(在 xterm \* hanziInputDir 目录中)。下面的语句定义字库:

```
xterm * hanziFont:             hanzigb16st
xterm * hanziBoldFont:
xterm * font:                  8x16
xterm * boldFont:
```

xterm \* hanziFont 和 xterm \* font 用以指定缺省的中文和西文字库,这里分别是 hanzigb16st 和 8x16,中文字符的宽度是西文字符的二倍。下面一段定义了字体选择菜单的资源:

```
xterm * fontMenu.Label:      VT Fonts
xterm * fontMenu * fontdefault * Label:  Default
xterm * fontMenu * font1 * Label:  Normal Ascii
xterm * VT100 * font1:      8x16
xterm * fontMenu * font2 * Label:  Large Ascii
xterm * VT100 * font2:      12x24
xterm * fontMenu * font3 * Label:  Normal Hanzi
xterm * VT100 * font3:      hanzigb16st
xterm * fontMenu * font4 * Label:  Thin Hanzi
xterm * VT100 * font4:      hanzigb16fs
xterm * fontMenu * font5 * Label:  Large Hanzi
xterm * VT100 * font5:      hanzigb24st
xterm * fontMenu * font6 * Label:  Normal Hanzi (GR)
xterm * VT100 * font6:      cclib16st
xterm * fontMenu * fontEscape * Label:  Escape Sequence
xterm * fontMenu * fontsel * Label:  Selection
```

用户启动 `xterm` 后,可以在任何时候在 `xterm` 窗口的文本区按住 `Ctrl`-鼠标右键激活字体选择菜单并选择所希望的中西文字体。如选择“Large Ascii”和“Large Hanzi”将使 `xterm` 的西文字体为 12x24,中文字体为 `hanzigb24st`。下面一段定义了激活各种输入法的热键:

```
xterm * VT100. Translations: # override \
~ Shift < KeyPress > F1: set-HZ-parameter(input-conv = toggle) \ n \
Shift < KeyPress > F1: switch-HZ-mode(ASCII) \ n \
~ Shift < KeyPress > F2: switch-HZ-mode(IG) \ n \
Shift < KeyPress > F2: switch-HZ-mode(QW) \ n \
< KeyPress > F3: popup-panel(config) \ n \
< KeyPress > F4: switch-HZ-mode(TONEPY) \ n \
~ Shift < KeyPress > F5: switch-HZ-mode(QJ) \ n \
Shift < KeyPress > F5: switch-HZ-mode(Punct) \ n \
~ Shift < KeyPress > F6: switch-HZ-mode(PY) \ n \
Shift < KeyPress > F6: switch-HZ-mode(CCDOSPY) \ n \
~ Shift < KeyPress > F7: switch-HZ-mode(WuBi) \ n \
Shift < KeyPress > F7: switch-HZ-mode(CangJie) \ n \
~ Shift < KeyPress > F8: switch-HZ-mode(English) \ n \
Shift < KeyPress > F8: switch-HZ-mode(CTLau) \ n \
~ Shift < KeyPress > F9: switch-HZ-mode(SW) \ n \
Shift < KeyPress > F9: switch-HZ-mode(QianMa) \ n \
< KeyPress > F10: switch-HZ-mode(TeleCode) \ n \
~ Shift < KeyPress > F11: switch-HZ-mode(HIRAGANA) \ n \
Shift < KeyPress > F11: switch-HZ-mode(KATAKANA) \ n \
~ Meta < Ctrl < Btn1Down > : select-start() click-HZ-area() \ n \
~ Meta < KeyPress > Escape: insert() set-HZ-parameter(input-conv = off)
```

如 F6 激活无调拼音, F7 激活五笔字型, Shift-F2 激活区位码。F3 键将弹出输入法配置窗口,用户可以在窗口内设置输入法字典目录,选择输入法及是否使用自动选择,自动分段和联想等功能。

#### 8.4.2 crxvt

`crxvt` 是中文输入服务器 `Xcin` 的客户程序,它本身只具备显示中文的能力。中文输入靠与中文输入服务器 `Xcin` 的通讯完成。参见 8.6 节“中文输入服务器——`Xcin`, `Chinput`”。

### 8.5 X Window 系统的中文字库

X Window 系统下的中文软件使用的中文字库大致有以下几类:

- BDF: Bitmap Distribution Format, X Window 系统字库的位图格式
- HBF: Hanzi Bitmap Font, 汉字位图字库, 每个汉字的位图占同样大的空间

- PCF: Portable Compiled Format, X Window 系统显示字库的标准格式, 可跨平台使用

- TTF: TrueType Font, 多在 Windows 下使用的轮廓线字库

这四种字库文件的扩展名通常是 .bdf, .hbf, .pcf, .ttr。但要注意, \* .hbf 文件只是字库描述文件, 汉字位图则存放在另外的文件里。

四种字库文件相互转换的程序有:

- bdf2pcf: X Window 系统自带

- hbftobdf bdftohbf:

`ftp://ftp.ifcss.org/pub/software/fonts/utills/hbf.tar.gz`

- gethbf: 从 X 服务器中取出某显示字库, 形成 BDF 文件

- ttf2bdf: 在 freetype 软件包里

`ftp://ftp.physiol.med.tu-muenchen.de/pub/freetype/` 或

`ftp://sunsite.unc.edu/pub/Linux/X11/fonts/`

### 8.5.1 在 X Window 中使用中文点阵字库

免费的中文 HBF 和 BDF 字库可从 `ftp://ftp.ifcss.org/pub/software/fonts/` 取得, 本书作者收集并整理的中文 PCF 字库可从下列地址获得:

`ftp://ftp.turboLinux.com.cn/pub/chinese/fonts/pcf/`

如果想使用中文字库作为 X Window 系统显示字库, 必须把 HBF, BDF, TTF 等格式的字库转换成 PCF 格式。但有些中文软件中的中文显示和打印可直接处理使用 HBF, BDF, TTF 等格式的字库。

IFCSS 存放有中文 GB 码的 16x16, 24x24, 48x48 和 56x56 等几种点阵的 HBF 字库。16 点阵的字库有宋体, 24 和 48 点阵的字库有宋体、黑体、楷体 and 仿宋体, 56 点阵的只有仿宋体。16 和 24 点阵的字库还有繁体。

从 IFCSS 取来的 HBF 字库 (.hbf 文件及相应的位图文件) 可直接用 hbftobdf 和 bdf2pcf 生成 PCF 字库 (但要注意 IFCSS 存放的 cc48s.hbf, cc48k.hbf, cc48h.hbf, cc48fs.hbf 四个文件中的位图文件名有误, 应加以更改)。不过, IFCSS 的 HBF 文件多用字库的简称或别名作为字库名 (HBF 文件中的“FONT”项)。但很多软件需要用到中文字库完整地符合 XLF 规则的字库名, 如 Netscape 根据字库名的“CHARSET\_REGISTRY”是否为“gb2312.1980”来搜索 X 服务器上的中文字库。另外用字库的简称或别名命名的字库不能用作可缩放 (scalable) 字库 (见下)。所以有必要把 HBF 文件中的“FONT”项作修改。如对 cc48s.hbf, 把下行:

```
FONT cc48s
```

改为:

```
FONT -CCLib-Song-medium-r-Normal-jiantizi-48-480-75-75-c-480-GB2312.1980-1
```

另外, 还可以把不同点阵和不同字体的中文 HBF 字库的“FOUNDRY”项改成统一的名称, 如“cclib”或其他, 以纳入一套显示字库的体系。这样很多软件用同样的索引就可以找到所有点阵所有字体的中文 X 字库。现在, 我们就可以用下列的命令产生 PCF 字库:

```
# hbftobdf cc48s.hbf > cc48s.1.bdf
```

```
# bdf2pcf cc48s.1.bdf > cc48s.1.pcf
```

IFC8S 的 GB 码的 HBF 文件使用标准的 EUC 编码(0xA1A1—0xF7FE),即中文字符的两个字节的最高位都为 1,所以在上面的命令中,我们在产生的 PCF 文件名中用“\_1”表示这一点。目前仍有很多软件使用两个字节的最高位都为 0 的编码(0x2121—0x777E)去索引 PCF 字库,所以我们有必要产生编码范围为 0x2121—0x777E 的 PCF 字库。这可以通过对各个 HBF 文件中的编码范围作修改来做到。如 cc48s.hbf 中如下几行:

```
DEFAULT_CHAR 0xA1A1
.....
HBF_BYTE_2_RANGE 0xA1-0xFE
.....
HBF_CODE_RANGE 0xA1A1-0xA9FE cc48.sym 0
HBF_CODE_RANGE 0xB0A1-0xD7FE cc48s.1 0
HBF_CODE_RANGE 0xD8A1-0xF7FE cc48s.2 0
```

可修改成:

```
DEFAULT_CHAR 0x2121
.....
HBF_BYTE_2_RANGE 0x21-0x7E
.....
HBF_CODE_RANGE 0x2121-0x297E cc48.sym 0
HBF_CODE_RANGE 0x3021-0x577E cc48s.1 0
HBF_CODE_RANGE 0x5821-0x777E cc48s.2 0
```

并把下行:

```
CHARSET_ENCODING "1"
```

改为:

```
CHARSET_ENCODING "0"
```

并把字库名从:

```
FONT -CClib-Song-medium-r-normal-jiantizi-48-480-75-75-c-480-GB2312.1980-1
```

改为:

```
FONT -CClib-Song-medium-r-normal-jiantizi-48-480-75-75-c-480-GB2312.1980-0
```

然后,用下列的命令产生 PCF 字库:

```
# hbftobdf cc48s.hbf > cc48s.bdf
# bdf2pcf cc48s.bdf > cc48s.pcf
```

我们依照上面的说明整理出一套中文 X 字库后,可把 PCF 文件用 compress 压缩成 .pcf.Z 文件,或用 gzip 压缩成 .pcf.gz 文件,以节省磁盘空间。Linux 上 X Window 系统的字库存放在 /usr/X11R6/lib/X11/fonts 目录的各子目录下,如 100dpi/, 75dpi/, Type1/, misc/等。我们可以把中文 X 字库的 PCF 文件(\*.pcf.Z)拷贝到 misc/目录下,然后执行如下命令:

```
lark# cd /usr/X11R6/lib/X11/fonts/misc
```

```
lark# /usr/X11R6/bin/mkfontdir
```

mkfontdir 将搜索 misc/ 目录下的所有字库并更新字库目录文件 fonts.dir。fonts.dir 文件中现在已增加了所有中文字库的列表, 类似下面的几行:

```
ecs16.pcf.Z -c-elib-song-medium-r-normal--16-160-75-75-c-160-gb2312.1980-0
ecs16f.pcf.Z -c-elibf-song-medium-r-normal--16-160-75-75-c-160-gb12345.1990-0
ecs24.pcf.Z -c-elib-song-medium-r-normal--24-240-75-75-c-240-gb2312.1980-0
ech24.pcf.Z -c-elib-hei-medium-r-normal--24-240-75-75-c-240-gb2312.1980-0
eck24.pcf.Z -c-elib-kai-medium-r-normal--24-240-75-75-c-240-gb2312.1980-0
ecs24.pcf.Z -c-elib-fangsong-medium-r-normal--24-240-75-75-c-240-gb2312.1980-0
ecs24f.pcf.Z -c-elibf-song-medium-r-normal--24-240-75-75-c-240-gb12345.1990-0
ec48s.pcf.Z -c-elib-song-medium-r-normal--48-480-75-75-c-480-gb2312.1980-0
ec48h.pcf.Z -c-elib-hei-medium-r-normal--48-480-75-75-c-480-gb2312.1980-0
ec48k.pcf.Z -c-elib-kai-medium-r-normal--48-480-75-75-c-480-gb2312.1980-0
ec48fs.pcf.Z -c-elib-fangsong-medium-r-normal--48-480-75-75-c-480-gb2312.1980-0
ecs16_1.pcf.Z -c-elib-song-medium-r-normal--16-160-75-75-c-160-gb2312.1980-1
ecs16f_1.pcf.Z -c-elibf-song-medium-r-normal--16-160-75-75-c-160-gb12345.1990-1
ecs24_1.pcf.Z -c-elib-song-medium-r-normal--24-240-75-75-c-240-gb2312.1980-1
ech24_1.pcf.Z -c-elib-hei-medium-r-normal--24-240-75-75-c-240-gb2312.1980-1
eck24_1.pcf.Z -c-elib-kai-medium-r-normal--24-240-75-75-c-240-gb2312.1980-1
ecs24_1.pcf.Z -c-elib-fangsong-medium-r-normal--24-240-75-75-c-240-gb2312.1980-1
ecs24f_1.pcf.Z -c-elibf-song-medium-r-normal--24-240-75-75-c-240-gb12345.1990-1
ec48s_1.pcf.Z -c-elib-song-medium-r-normal--48-480-75-75-c-480-gb2312.1980-1
ec48h_1.pcf.Z -c-elib-hei-medium-r-normal--48-480-75-75-c-480-gb2312.1980-1
ec48k_1.pcf.Z -c-elib-kai-medium-r-normal--48-480-75-75-c-480-gb2312.1980-1
ec48fs_1.pcf.Z -c-elib-fangsong-medium-r-normal--48-480-75-75-c-480-gb2312.1980-1
```

有些程序(如 CXterm)只能识别短的字库名。所以我们必须给上面的中文 X 字库加上别名。我们可以根据上面 fonts.dir 中的几行增加中文字库的别名, 即把字符串“.pcf.Z”去掉, 得到:

```
ecs16 -c-elib-song-medium-r-normal--16-160-75-75-c-160-gb2312.1980-0
ecs16f -c-elibf-song-medium-r-normal--16-160-75-75-c-160-gb12345.1990-0
ecs24 -c-elib-song-medium-r-normal--24-240-75-75-c-240-gb2312.1980-0
ech24 -c-elib-hei-medium-r-normal--24-240-75-75-c-240-gb2312.1980-0
eck24 -c-elib-kai-medium-r-normal--24-240-75-75-c-240-gb2312.1980-0
ecs24 -c-elib-fangsong-medium-r-normal--24-240-75-75-c-240-gb2312.1980-0
ecs24f -c-elibf-song-medium-r-normal--24-240-75-75-c-240-gb12345.1990-0
ec48s -c-elib-song-medium-r-normal--48-480-75-75-c-480-gb2312.1980-0
ec48h -c-elib-hei-medium-r-normal--48-480-75-75-c-480-gb2312.1980-0
ec48k -c-elib-kai-medium-r-normal--48-480-75-75-c-480-gb2312.1980-0
ec48fs -c-elib-fangsong-medium-r-normal--48-480-75-75-c-480-gb2312.1980-0
ecs16_1 -c-elib-song-medium-r-normal--16-160-75-75-c-160-gb2312.1980-1
ecs16f_1 -c-elibf-song-medium-r-normal--16-160-75-75-c-160-gb12345.1990-1
```

```
ecs24_1 -cclib-song-medium-r-normal--24-240-75-75-c-240-gb2312.1980-1
ccn24_1 -cclib-hei-medium-r-normal--24-240-75-75-c-240-gb2312.1980-1
ock24_1 -cclib-kai-medium-r-normal--24-240-75-75-c-240-gb2312.1980-1
ccfs24_1 -cclib-fangsong-medium-r-normal--24-240-75-75-c-240-gb2312.1980-1
ocs24f_1 -cclibf-song-medium-r-normal--24-240-75-75-c-240-gb12345.1990-1
cc48s_1 -cclib-song-medium-r-normal--48-480-75-75-c-480-gb2312.1980-1
cc48h_1 -cclib-hei-medium-r-normal--48-480-75-75-c-480-gb2312.1980-1
cc48k_1 -cclib-kai-medium-r-normal--48-480-75-75-c-480-gb2312.1980-1
cc48fs_1 -cclib-fangsong-medium-r-normal--48-480-75-75-c-480-gb2312.1980-1
```

把上列字库别名加进 misc/目录下的 fonts.alias 文件中。中文字库的安装即告完成。我们可以用 xset 命令让 X 服务器立即重读所有字库目录,更新 X 服务器中存放的字库信息:

```
lark% xset fp rehash
```

现在我们就可以使用新安装的中文 X 字库了。下次启动 X 服务器的时候,我们无需使用 xset 命令,因为 /usr/X11R6/lib/X11/fonts/misc 在 X 服务器缺省的字库搜索路径上,X 服务器在启动时已搜索所有在缺省的字库搜索路径上的目录并建立了所有字库的信息(如图 8.5 和 8.6 所示)。

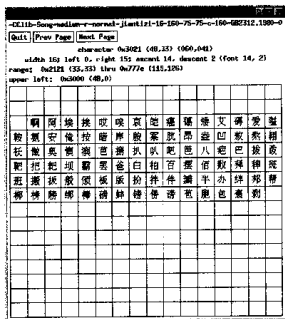


图 8.5 中文 GB 码 X 字库的一个区

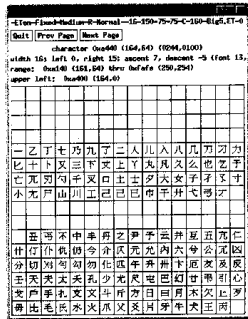


图 8.6 中文 Big5 码 X 字库的一个区

如果我们希望把中文 X 字库放到一个单独的地方,可以在 /usr/X11R6/lib/X11/fonts 目录下创建一个目录如 /Chinese。把中文 X 字库的 PCF 文件(\*.pcf.Z)拷贝到 /Chinese 目录下,用 mkfontdir 命令创建 fonts.dir 文件,把中文字库的别名存放在 fonts.alias 文件中拷贝到 /Chinese 下。然后用命令

```
lark% xset fp + /usr/X11R6/lib/X11/fonts/Chinese/
```

把 /Chinese 目录加到当前字库搜索路径上并更新 X 服务器中存放的字库信息,我们就可以用新安装的中文 X 字库了。但是 /Chinese 目录尚不在 X 服务器的缺省字库搜索路径

上。为把 /Chinese 加到 X 服务器的缺省字库搜索路径上,使得每次启动 X 服务器不必手动执行 xset 命令,我们可以修改 /etc/X11/XF86Config 文件,搜寻定义 FontPath 的语句,把 /Chinese 目录加进 FontPath 变量:

```
FontPath "/usr/X11R6/lib/X11/fonts/Chinese/"
```

如果一个无特权的用户想建立自己的 X 字库目录,如 \$HOME/xfonts,可依上面的说明拷贝字库文件到此目录中,创建 fonts.dir 和 fonts.alias 文件,执行:

```
lark% xset fp + $HOME/xfonts
```

并把这条命令加进 \$HOME/.xsession 文件,则以后这个用户再使用 X 服务器的时候,自己的字库目录 \$HOME/xfonts 会被自动加进当前字库搜索路径上。

下面介绍如何使用可缩放中文字库。一个符合 XLFD 规则的 X Window 系统字库名共有 14 个域(field),如:

```
-cclib-song-medium-r-normal-jiantizi-24-240-75-75-c-240-gb2312.1980-0
```

其中第 7,8,9,10,12 五个域的意义分别是:

```
PIXEL_SIZE POINT_SIZE RESOLUTION_X RESOLUTION_Y AVERAGE_WIDTH
```

这五个数值中只有三个是相互独立的,另外两个可以根据这三个计算得到。它们之间的相互关系为:

$$\text{AVERAGE\_WIDTH} = \text{RESOLUTION\_X} * \text{POINT\_SIZE} / 72.27$$

$$\text{PIXEL\_SIZE} = \text{RESOLUTION\_Y} * \text{POINT\_SIZE} / 72.27$$

一个 X 显示字库的名字(而不是别名)如果符合 XLFD 规则,则在它被装入 X 服务器的时候就成为可缩放字库。我们可以指定上述五个域中互相独立的三个,另外两个用“0”或“\*”来代替,由 X 服务器来决定装入 X 服务器的显示字体的宽度和高度。如我们在程序中用下面的名字请求 X 服务器装入中文字库:

```
-cclib-song-medium-r-normal-jiantizi-* -260-75-75-c-* -gb2312.1980-0
```

将得到近似 26x26 点阵的简体宋体字。每个 X 服务器启动时都有缺省的分辨率和点距。如果我们在字库名中没有指定 X 或 Y 方向的分辨率或点距,X 服务器将使用它们的缺省值。如我们要求装入下列字库:

```
-cclib-song-medium-r-normal-jiantizi-26-0-0-0-c-0-gb2312.1980-0
```

X 服务器将可能装入下列简体宋体字库:

```
-cclib-song-medium-r-Normal-jiantizi-26-251-75-75-c-250-gb2312.1980-0
```

由原字型通过缩放得到的字型显示效果通常较原字型粗糙(如图 8.7 所示)。

建立 X 字库服务器可以把本机上的 X 字库提供给网络上其他计算机使用。对于中文 X 字库,有时可能有必要建立 X 字库服务器,因为中文 X 字库一般文件都比较大,使用 X 字库服务器可避免每台机器上都保存一套字库,占用很多磁盘空间。而且通常由一个人维护更新中文 X 字库会更方便有效。

首先,创建字库服务器配置文件,如 fs7100.conf,其主要内容如下:

```
port = 7100      # X 字库服务器占据端口号
client-limit = 10 # 此字库服务器最多允许客户连接数
clone-self = on  # 当一个字库服务器达到最多连接数时,启动新服务器
catalogue = /usr/X11R6/lib/X11/fonts/Type1,
```

```

/usr/X11R6/lib/X11/fonts/Speedo,
/usr/X11R6/lib/X11/fonts/misc,
/usr/X11R6/lib/X11/fonts/75dpi,
/usr/X11R6/lib/X11/fonts/100dpi,
/usr/X11R6/lib/X11/fonts/Chinese

```

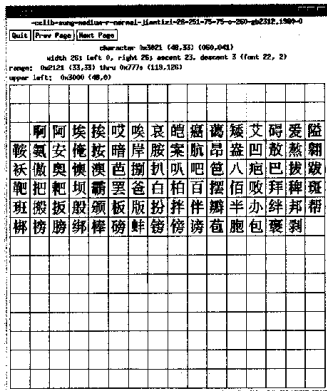


图 8.7 通过字型缩放得到的 26x26 点阵的简体宋体字

# 字库目录

```
default-point-size = 120 # 缺省点数 x 10
```

```
default-resolutions = 100,100,75,75 # 缺省分辨率 100x100 和 75x75
```

然后启动 X 字库服务器:

```
lark% xfs -cf fs7100.conf &
```

在客户机上用“xset fp+ tcp/server\_name:port”命令把 X 字库服务器加到当前字库搜索路径上,如:

```
apple% xset fp+ tcp/lark:7100
```

### 8.5.2 在 X Window 中使用中文 TrueType 字库

最近,随着 UNIX 上免费的 TrueType 字库引擎的发展,用户已经可以把以前只在 Windows 或 Macintosh 上使用的 TrueType 字库用作 X Window 系统下的显示字体。在 X Window 下使用中文 TrueType 字库的优点是,用户可以按自己的要求得到多种字形的正体、斜体、粗体和可无限缩放的中文显示字体。

为了能在 X Window 下使用 TrueType 字库,用户需要在系统上安装如下软件:

- TrueType 字库引擎 FreeType

- 支持 TrueType 字库的 X 服务器或 X 字库服务器(xfs)  
并适当地设置 TrueType 字库目录。

### 1. 安装 TrueType 字库引擎 FreeType

由 FreeType 开发小组开发的免费的 TrueType 字库引擎 FreeType 是一套接口函数, 用户可以在程序中利用 FreeType 函数从 TrueType 字库中提取字库的总体信息和单个字形的信息, 从而按自己的要求产生显示字体或打印字体。用户可以从

<http://www.freetype.org>

获得有关 FreeType 的信息和最新版本的 FreeType。FreeType 目前的版本是 1.2。

FreeType 的安装非常简单。从上述主页指引的地址下载 freetype-1.2.tar.gz, 然后按下面的步骤编译安装 FreeType:

```
# tar xzvf freetype-1.2.tar.gz
# cd freetype-1.2
# ./configure --prefix=/usr
# make all
# make install
```

安装过程将在 /usr/lib 目录下安装以下开发函数库:

```
libttf.a
libttf.so → libttf.so.2.1.0
libttf.so.2 → libttf.so.2.1.0
libttf.so.2.1.0
```

在 /usr/include 下安装开发头文件:

```
fterrid.h      ftxcmap.h      ftxgasp.h      ftxpost.h
ftnameid.h     ftxerr18.h     ftxkern.h      ftxwidth.h
```

在 /usr/bin 下安装以下演示应用程序:

```
ftdump      fterror      ftlint      ftstring
fstrpnm     ftimer       ftview      ftzoom
```

其中 ftdump 命令用以显示一个 TrueType 字库文件的总体信息, 如

```
# ftdump htsong.ttf
```

```
font name table entries
```

```
-----
SongTi - 2.5
```

```
Copyright of Hunan Huatian Information Industry Co., Ltd, 1999
```

```
Gezujin
```

```
-----
character map encodings
```

```
-----  
There are 2 encodings:
```

```
encoding 0: Apple Roman  
encoding 1: Windows Unicode
```

```
-----  
ftxcmmap test
```

```
-----  
There are 2 encodings:
```

```
encoding 0:
```

```
first: 0
```

```
next: 2
```

```
last: 255
```

```
encoding 1:
```

```
first: 32
```

```
next: 32
```

```
last: 65535
```

```
-----  
Memory footprint statistics:
```

```
-----  
135124 bytes ( 132kByte): face object
```

```
9001 bytes ( 9kByte): glyph object
```

```
6532 bytes ( 7kByte): instance object
```

```
31419 bytes ( 31kByte): exec. context object
```

```
-----  
182076 bytes ( 178kByte): total memory usage
```

我们可以看出,这个字库含有“Apple Roman”和“Windows Unicode”两种编码。这个信息对我们很有用。

ftview 命令用来浏览一个 TrueType 字库中的所有字形(glyph)。如用户可以用

```
# ftview 50 htsong.ttf
```

以每个字形 50 个点的大小来浏览 TrueType 字库 htsong.ttf 中的所有字形。ftview 有很多热键来实现放大缩小字形,向前或向后跳转 1 个、10 个、100 个或 1000 个字形等。用户可以用 ftview 命令来检查 TrueType 字库中字形的质量(如图 8.8 所示)。

## 2. 安装支持 TrueType 字库的 X 服务器或 X 字库服务器(xfs)

日本的 X TrueType Server 计划开发的 X TrueType Server(简称 X-TT)利用 TrueType 字库引擎 FreeType,对 XFree86 的源代码进行了修改。修改后的源代码编译出

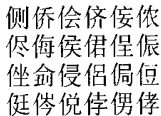


图 8.8 用 ftview 程序检查  
中文 TrueType 字库

的 X 服务器或 X 字库服务器可以支持在 X Window 系统中使用 TrueType 字库。

下面的网址可以指引用户下载最新的 X-TT:

<http://hawk.ise.chuo-u.ac.jp/student/person/tshiozak/X-TT/>

X-TT 目前的版本是 1.1。

为编译安装支持 TrueType 字库的 X 服务器或 X 字库服务器,用户必须下载以下软件包:

- freetype-1.2.tar.gz

用户必须首先编译安装 FreeType。

- xtt-1.1.tar.gz

xtt11-to-pl00.diff.gz

xtt11pl00-to-pl01.diff.gz

xtt11pl01-to-pl02.diff.gz

xtt11pl02-to-pl03.diff.gz

xtt11pl03-to-pl04.diff.gz

xtt11pl04-to-pl05.diff.gz

X-TT 1.1 的源代码包和小的修正,含对 XFree86 源代码的修正和一些文档。

- X333src-1.tgz

XFree86 源代码包。如果用户只希望编译支持 TrueType 字库的 X 服务器,而不希望编译支持 TrueType 字库的 X 字库服务器,用户可以只下载 X333servonly.tgz。

- xtt-1.1-gb13000.diff

本书著者之一(陈向阳)对 X-TT 的修正,加进对 GBK 码的支持和对编码为“Windows GB2312”的 TrueType 字库的支持。用户可以从

<ftp://ftp.turboLinux.com.cn/pub/chinese/system/>

下载这个修正。

首先,用户要展开 X-TT 1.1 的源代码包:

```
# tar xzvf xtt-1.1.tar.gz
```

然后,用户要展开 XFree86 的源代码包。如果用户想编译安装所有 XFree86 3.3.3

文件,用下面的命令:

```
# tar zxvf X333src-1.1.gz
```

如果用户只想编译安装 X 服务器,用下面的命令:

```
# tar zxvf X333serveronly.1.gz
```

如果用户只想编译安装 X 字库服务器,用下面的命令:

```
# xtt-1.1/ext-xf86.sh X333src-1.1.gz
```

下面,用户要对 XFree86 的源代码加进 X-TT 的修正:

```
# patch -p0 -t < xtt-1.1/xtt-1.1.diff
# gzip -cd xtt11-to-pl00.diff.gz | patch -p0 -t
# gzip -cd xtt11pl00-to-pl01.diff.gz | patch -p0 -t
# gzip -cd xtt11pl01-to-pl02.diff.gz | patch -p0 -t
# gzip -cd xtt11pl02-to-pl03.diff.gz | patch -p0 -t
# gzip -cd xtt11pl03-to-pl04.diff.gz | patch -p0 -t
# gzip -cd xtt11pl04-to-pl05.diff.gz | patch -p0 -t
# patch -p0 -t < xtt-1.1-gb13000.diff
```

下面是编译安装过程。对于要安装所有 XFree86 3.3.3 文件或只安装 X 服务器的情形,用户在适当地设置 xf86site.def 文件后,执行以下命令:

```
# cd xc
# make World
# make install
```

对于只想安装 X 字库服务器的情形下,执行以下命令:

```
# cd xc
# make xfs
# make install
```

### 3. 设置 TrueType 字库目录

为了使 X Window 系统能使用中文 TrueType 字库,用户必须把中文 TrueType 字库文件放置到一个字库目录下,并正确书写 fonts.dir 和 fonts.alias 文件,以使 X 服务器能用标准的 XLFD 字库名来调用 TrueType 字库。

X-TT 支持各种西方语言和中日韩语等多种字符集。我们在这里只就与中文 TrueType 字库有关的问题进行说明。对于其他语言,用户可以参考这里的说明并根据软件包中的说明文件进行设置。

X-TT 可以使用的中文 TrueType 字库文件必须是下面几种编码之一:

```
Windows Unicode
Windows GB2312
Windows Big 5
```

(用户可以用 `ftdump` 命令来查看一个中文 TrueType 字库文件的编码。)用户必须根据中文 TrueType 字库所包含的中文字符集和所使用的编码,来决定如何去写 `fonts.dir` 文件。

X-TT 可以使用的中文 TrueType 字库文件的字符集,编码和 X 字库名可能的组合如

下:

| 字符集    | 编码              | X 字库名                                                                             |
|--------|-----------------|-----------------------------------------------------------------------------------|
| GBK    | Unicode         | * -gb13000.1993-1<br>* -gb2312.1980-0<br>* -big5-0, * -big5.et-0, * -big5.et-en-0 |
| GB2312 | Unicode, GB2312 | * -gb2312.1980-0                                                                  |
| Big5   | Unicode, Big 5  | * -big5-0, * -big5.et-0, * -big5.et-en-0                                          |

例如,如果用户有一个含 GBK 字符集,以 Unicode 编码的中文 TrueType 字库文件 song.ttf,用户可以建立一个目录

```
/usr/X11R6/lib/X11/fonts/tt/
```

并将 song.ttf 文件拷贝到这个目录下,然后在此目录下创建 fonts.dir 文件,内容如下:

```
16
song.ttf -xtt-song-medium-r-Normal--0-0-0-c-0-gb2312.1980-0
song.ttf -xtt-song-medium-r-Normal--0-0-0-c-0-gb13000.1993-1
song.ttf -xtt-song-medium-r-Normal--0-0-0-c-0-big5-0
song.ttf -xtt-song-medium-r-Normal--0-0-0-c-0-big5.et-0
ab = y: song.ttf -xtt-song-bold-r-Normal--0-0-0-c-0-gb2312.1980-0
ab = y: song.ttf -xtt-song-bold-r-Normal--0-0-0-c-0-gb13000.1993-1
ab = y: song.ttf -xtt-song-bold-r-Normal--0-0-0-c-0-big5-0
ab = y: song.ttf -xtt-song-bold-r-Normal--0-0-0-c-0-big5.et-0
ai = 0.3: song.ttf -xtt-song-medium-i-Normal--0-0-0-c-0-gb2312.1980-0
ai = 0.3: song.ttf -xtt-song-medium-i-Normal--0-0-0-c-0-gb13000.1993-1
ai = 0.3: song.ttf -xtt-song-medium-i-Normal--0-0-0-c-0-big5-0
ai = 0.3: song.ttf -xtt-song-medium-i-Normal--0-0-0-c-0-big5.et-0
ai = 0.3: ab = y: song.ttf -xtt-song-bold-i-Normal--0-0-0-c-0-gb2312.1980-0
ai = 0.3: ab = y: song.ttf -xtt-song-bold-i-Normal--0-0-0-c-0-gb13000.1993-1
ai = 0.3: ab = y: song.ttf -xtt-song-bold-i-Normal--0-0-0-c-0-big5-0
ai = 0.3: ab = y: song.ttf -xtt-song-bold-i-Normal--0-0-0-c-0-big5.et-0
```

第一行的“16”表示此 fonts.dir 文件中的 X 字库总数。下面 16 行左侧表示 TrueType 字库名和修饰,右侧表示 X 字库名。“ab = y”表示粗体,对应 X 字库名中 WEIGHT 项的“bold”,“ai = 0.3”表示斜体,倾斜度为 0.3,对应 X 字库名中 SLANT 项的“i”,“ai = 0.3: ab = y”表示粗斜体,对应 X 字库名中的“bold-i”。

如果用户还有一个含 GB2312 字符集,编码为 GB2312 的中文 TrueType 字库 kai.ttf,可以把字库文件拷贝到/usr/X11R6/lib/X11/fonts/tt/目录中,然后在 fonts.dir 文件中加上下面几行:

```
kai.ttf -xtt-kai-medium-r-Normal--0-0-0-c-0-gb2312.1980-0
ab = y: kai.ttf -xtt-kai-bold-r-Normal--0-0-0-c-0-gb2312.1980-0
ai = 0.3: kai.ttf -xtt-kai-medium-i-Normal--0-0-0-c-0-gb2312.1980-0
ai = 0.3: ab = y: kai.ttf -xtt-kai-bold-i-Normal--0-0-0-c-0-gb2312.1980-0
```



个文件作适当修改。其中最重要的两点是:将中文 TrueType 字库目录 `/usr/X11R6/lib/X11/fonts/tt/` 加进“catalogue”一行和增加一行“deferglyphs = 16”。修改后的 config 文件大致如下:

```
# font server configuration file
# $ XConsortium: config.cpp,v 1.7 91/08/22 11:39:59 rws Exp 8

clone-self = on
use-syslog = off

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/,
            /usr/X11R6/lib/X11/fonts/tt/

deferglyphs = 16
error-file = /usr/X11R6/lib/X11/fs/errors
# in decipoints
default-point-size = 120
default-resolutions = 75,75,100,100
```

用户可在命令行运行下面的命令以启动 X Font Server:

```
# /usr/X11R6/bin/xfs &
```

X 字库服务器将在 7100 端口等待请求。为方便起见,用户最好把这一行加进系统启动文件,如 `/etc/rc.d/rc.local`,以使系统每次启动时都能自动启动 X 字库服务器。

要在普通的 X 服务器下使用中文 TrueType 字库,启动 X 服务器时必须加上选项“-deferglyphs 16”。如果用户设定的系统启动模式是文本模式,必须修改 `/usr/X11R6/bin/startx` 文件,把最后一行改为

```
xinit $ clientargs -deferglyphs 16 $ serverargs
```

如果用户设定的系统启动模式是图形模式,即用 `xdm` 启动 X 服务器,必须修改 `/usr/X11R6/lib/X11/xdm/Xservers` 文件,把最后一行改为

```
:0 local /usr/X11R6/bin/X -deferglyphs 16
```

对于启动为文本模式的系统,用户可以在命令行上用

```
# startx
```

启动 X 服务器。对于启动为图形模式的系统,在 `xdm` 登录窗口登录,`xdm` 程序就会启动 X 服务器。然后用户必须在 X 模式下的终端命令行上输入命令

```
# xset fp+ tcp/localhost:7100
```

以设置 X 服务器的中文 TrueType 字库路径。为避免每次启动 X 服务器都必须手动运行这条命令,用户可以把这条命令加进系统的 `/usr/X11R6/lib/X11/xinit/xinitrc`(对于文本模式)或 `/usr/X11R6/lib/X11/xdm/Xsession`(对于图形模式)文件中,这样 X 服务器在每次启动时将自动执行上面这条命令。

现在,用户就可以在 X Window 下使用中文 TrueType 字库了。

## 8.6 中文输入服务器——Xcin, Chinput

### 8.6.1 Xcin + crxvt

XCIN 是 Xwindow Chinese Input 的缩写。是一个在 X Window 系统模式下运行的中文输入系统,xcin 是利用 X Window 系统的客户机/服务器模式工作的,所以系统上只要启动一个 xcin,便可为许多 crxvt 虚拟终端提供输入服务,因而占用系统资源较少(如图 8.10 所示)。xcin 目前由台湾的居士先生(thhsieh@twclx.phys.ntu.edu.tw)负责维护,用户可从下面的地址获得 xcin 的源程序包:

<ftp://Linux.cis.nctu.edu.tw/packages/chinese/xcin/>

xcin 的安装遵循一般的 GNU 软件包安装的规则。

本书的著者之一(方汉)将 xcin + crxvt 改编成可支持 GB 码的版本,用户可取来

<ftp://ftp.turbolinux.com.cn/pub/chinese/system/xcingb-2.2.tar.gz>

经过 X Window 系统的中文文化包装(WRAP)后,xcin 可为 X Window 系统的很多应用软件提供中文输入服务。参见 8.15 节“X Window 系统的中文文化包装(WRAP)方案”。

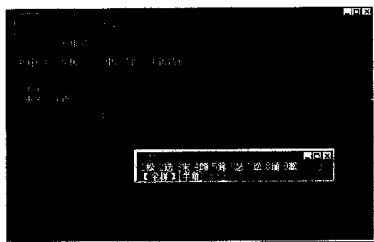


图 8.10 用 Xcin 在 crxvt 中输入中文

### 8.6.2 Chinput

汉字输入服务器 Chinput 是基于 X Window 系统的汉字输入方案。其目的在于为现有软件在汉字输入方面的汉化提供方便。本软件是在 UNIX 上的 Cxterm 的基础上开发的。所以其用户界面和输入方法为大家所熟悉。本软件还在用户函数接口(API)方面提供了更方便的操作。

Chinput 由于明俭(justiny@turboLinux.com.cn)编制,用户可以从下列地址取得 Chinput:

<ftp://ftp.turbolinux.com.cn/pub/chinese/packages/>

目前 Chinput 已被合并到由于明俭、陈向阳和方汉共同开发的外挂式中文平台 ZWinPro 之中,作为它的中文输入服务器。

本软件采用客户间通讯(Inter-Client Communication)方式实现键盘输入的汉字转换。

用户键入的字符输送到输入服务器,服务器负责根据对应的输入方法进行汉字转换。使用客户机/服务器方式进行软件的汉字输入汉化相当容易。

本软件具有良好的用户界面,采用有三维效果的无边框窗口,窗口不接受 FocusIn(除非有意设置,即使已是聚焦窗口,用户仍然不受边界影响),并且永远位于窗口堆栈(stack order)的最上方(与 Windows 系统上的汉字外挂平台外观效果类似)。它使用户的注意力集中在窗口输入区,而不受由窗口聚焦变更带来的影响。

Chinput 输入条分为两部分,左边为输入区,右边为按钮区,有开关按钮,浏览输入按钮,半角/全角切换按钮和中/英文标点符号切换按钮(如图 8.11 所示)。



图 8.11 Chinput 输入条

输入条支持两种显示模式:单行和双行。单行模式如图 8.12 所示。



图 8.12 单行显示模式

全角输入用于把输入的英文字母和数字符号转换为对应的中文字母和数字。全角输入和半角输入的切换可以通过点击输入条上的半角/全角按钮完成。缺省的输入模式为半角输入。

汉字标点符号转用于把输入的英文标点符号转换为汉字标点符号。其切换方法是点击输入条上的标点符号按钮。需要注意的是:

(1) 为了适应汉字对标点符号的要求,一个英文标点符号可能对应多个中文标点符号。下表列出了所有一对多的标点符号。凡注有“循环输入”的标点符号,表明在标点输入时,输入条将按标点的顺序循环返回。

| 中文标点  | 英文标点 | 说 明                       |
|-------|------|---------------------------|
| ,     | ,    | Control + , 或全角 + 中文标点时输入 |
| 。     | .    | Control + . 或全角 + 中文标点时输入 |
| “ ”   | " "  | 循环输入                      |
| ‘ ’   | ' '  | 循环输入                      |
| ¥ \$  | \$   | 循环输入                      |
| { }   | { }  | 循环输入                      |
| [ ]   | [ ]  | 循环输入                      |
| { }   | { }  | 循环输入                      |
| [ ]   | [ ]  | 循环输入                      |
| ——    | -    | 破折号扩展                     |
| ..... | ~    | 省略号扩展                     |
| < >   | < >  | 循环输入                      |
| >>    | >    | 循环输入                      |

(2) 英文标点符号“,”和“.”一般用于输入时向前和向后翻页。只有在选中全角时才以中文标点符号返回,为了方便用户,ZWinPro 还提供了更简单的输入方法,即按

Control + “,” 或 “.” 可以输入相应的中文标点。

(3) 另外, 还增加了两个符号“-”和“~”, 用来输入汉字的破折号和省略号。

Chinput 支持 CXterm 的 20 多种输入法, 并使用与 CXterm 相同的功能键进行输入法切换。如下表:

| 按键       | 输入方法       |
|----------|------------|
| F1       | 允许/禁止 中文输入 |
| F2       | 内码输入       |
| F3       | 智能拼音       |
| F4       | 带调拼音       |
| F6       | 无调拼音       |
| F7       | 五笔字型       |
| F8       | 英汉         |
| F9       | 首尾         |
| F10      | 电报码        |
| Shift-F1 | ASCII      |
| Shift-F2 | 区位码        |
| Shift-F6 | 缩写拼音       |
| Shift-F7 | 新仓颉        |
| Shift-F8 | 刘式粤音       |
| Shift-F9 | 栈码         |

Chinput 新增的智能拼音(由 Li Zhenchun <zhchli@163.net> 提供)输入法由 F3 键激活。按 Control-Space 可隐藏或显示输入条。按下输入条面板上的“开关”按钮便可以退出Chinput。

Chinput 提供了浏览编码并输入到聚焦窗口的机制。浏览输入一般用于输入汉字符号、汉语拼音符号、汉字表格符号等。如图 8.13 所示。

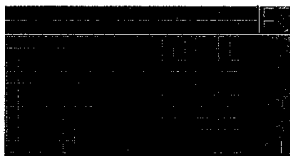


图 8.13 浏览输入

输入条的属性由用户主目录下的文件 .chinput 决定, 或在此文件不存在时由 /usr/lib/ZWinPro/Chinput.ad 决定。在这个文件中可以指定输入条的下面一些属性:

```
!
! Resource file for Chinput-1.5
!
!
```

```

! 显示模式: 单行或两行
!
chinput.mode:                TWO

!
! 初始编码: GB, BIG5, JIS, KS
!
chinput.encoding:            GB

!
! 输入法词典目录, 由 CXterm 安装产生
!
chinput.dictionary:          /usr/dict

!
! 输入方法
! GB 编码 : ASCII, IC, QW, TONEPY, QJ, Punct, PY, ZNPY, CCDOSPY, WuBi,
!           CangJie, English, CTLau, SW, QianMa, TeleCode,
!           HIRAGANA, KATAKANA
! BIG5 编码: ASCII, IC, PY, QJ, Punct, ETZY, ZOZY, Simplex, CangJie,
!           English, 4Corner, HSU, ARRAY30, Boshiamy
! 日文编码: ASCII, IC, ROMKANA, TCODE, QJ, Punct
! 韩文编码: ASCII, IC, HANGUL, HANJA, QJ, Punct
!
chinput.inputmethod:         ZNPY

!
! 联想输入文件名
!
chinput.association:         simple.lx

!
! 智能拼音输入
!
chinput.table:               /usr/lib/ZWinPro/pyinput/table
chinput.libphrase:           /usr/lib/ZWinPro/pyinput/libphrase
chinput.usrphrase:           .pyinput/usrphrase

!
! 16 点阵字体(与 24 点阵字体任选其一)
!
chinput.font:                8x16
chinput.gbfont:              hanzigb16fs

```

```

chinput.big5font:      -eten-fixed-medium-r-Normal-16-150-75-75-c-160-big5.et-0
chinput.jisfont:       jiskan16
chinput.ksfont:        -daewoo-mincho-medium-r-Normal-16-120-100-100-c-160-ksc5601.
1987-0

!
! 24 点阵字体(与 16 点阵字体任选其一)
!
! chinput.font:        12x24
! chinput.gbfont:      ccs24
! chinput.big5font:    -eten-fixed-medium-r-normal-24-230-75-75-c-240-big5.et-0
! chinput.jisfont:     jiskan24
! chinput.ksfont:      -daewoo-mincho-medium-r-normal-24-170-100-100-c-240-ksc5601.
1987-0

!
! 面板颜色
!
chinput.dimcolor:      # 666666
chinput.lightcolor:    # f5f5f5
chinput.panelcolor:    # c0c0c0
chinput.hzcolor:       # 000000
chinput.textcolor:     # 000000

!
! end of resource
!
```

软件设计直接使用 Xlib,处理速度较快,而且便于在各平台间移植。所有以 X11,Xt,Xm 等为基础的软件都可以很方便地使用该服务器。

本软件提供了如下 API:

- HZclientInit() 初始化客户软件
- HZQueryServer() 获取服务器的有关信息,如协议、版本、编码形式、目前状态等
- HZconfigServer() 改变服务器的设置或状态。如编码、面板颜色、输入方法、窗口锁定、窗口嵌入等
- HZsendKey() 输送键盘输入到服务器
- HZprocInput() 处理服务器回送或自身回送

该 API 简化了程序员对键盘输入的处理。即所有键盘输入都向服务器发送,而客户端仅仅处理服务器的回送信息或自身回送。所谓自身回送即如果服务器没有启动或虽然启动但处于“disable”状态,则 HZsendKey 向自身发送信息(ClientMessage)。这样使得服务器与客户程序的设计都比较可靠。如图 8.14 所示为其工作原理。

使用该 API,客户程序不必锁定输入窗口 ID,也不必通知服务器汉字转换开始和结

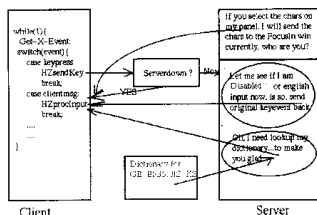


图 8.14 汉字输入服务器工作原理

束,更无与 Xserver 争夺 X 事件(XEvent)的必要。多个客户程序可以同时共享同一服务器。多个程序可以同时控制输入服务器,而不必关心服务器目前的状态。客户程序可以在服务器启动前或启动后运行。客户软件不会因服务器死掉而受影响。

使用该 API 对现有软件在输入方面的汉化相当容易,只需把原来的键盘处理改为对客户信息(Client Message)的处理即可。

本软件使用的中文 X 字库存放在

<ftp://ftp.turbolinux.com.cn/pub/chinese/fonts/pcf/>

目录。参见 8.5 节“X Window 系统的中文字库”。

经过 X Window 系统的中文化包装(WRAP)后,Chinput 可为 X Window 系统的很多应用软件提供中文输入服务。参见 8.15 节“X Window 系统的中文化包装(WRAP)方案”。

## 8.7 中文化的 X Window 编程 Widget 集——EZWGL

EZWGL 是由德克萨斯大学的 Maorong Zou 创建的一套 X Window 系统下的 Widget 集,它以编程方便、功能齐全而深受用户的喜爱。尤其是它的三维画板(3D Canvas) Widget 库更是所有 Widget 集中唯一具有的功能。

EZWGL 的基本组件有(如图 8.15~8.23 所示):



图 8.15 标签(Label)



图 8.16 边框(Frame)

- 标签(Label, FreeLabel, NWLabel)
- 边框(Frame)
- 按钮(Button, Check Button, Radio Button, MenuButton)
- 菜单(Menu, SubMenu)
- 滚动条(Slider)
- 图标(Icon)
- 输入区(Entry, optEntry)
- 列表(ListBox)
- 文本区(Text)
- 笔记本(Notebook)
- 缩放条(PaneHandle)
- 三维画板(3D Canvas)

高级组件有:

- 文件选择器(FileSelector)
- 执行器(Executor)
- 树状列表(ListTree)
- 高级列表(FancyListBox)
- 工作区(WorkArea)



图 8.17 按钮(Button, Check Button, Radio Button)

EZWGL 的最近版本和在线档案可以在 <http://rene.ma.utexas.edu/~mzou/EZWGL/> 取得。



图 8.18 图标(Icon)

于明俭(justiny@turboLinux.com.cn)对 EZWGL 进行了全面汉化,推出了 EZWGL-1.24 的汉化版本。汉化版本支持 CB, BIG5, 日文 EUC, 韩文 EUC 和 ASCII 码混用,并且支持上述四种编码的输入;支持 EZWGL 的所有组件,原软件包中所附的例子都经过汉化测试并已证明稳定。

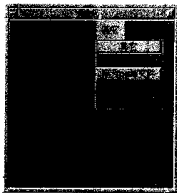


图 8.19 菜单(Menu, SubMenu)



图 8.20 滚动条(Slider)

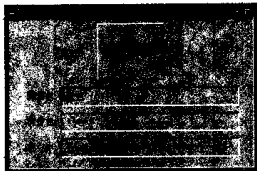


图 8.21 输入区(Entry)

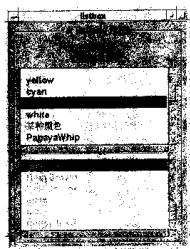


图 8.22 列表(ListBox)

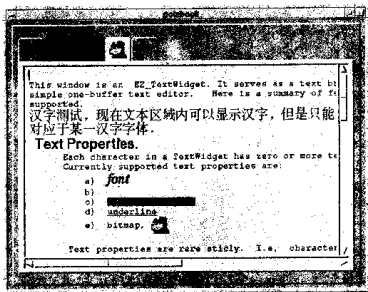


图 8.23 笔记本(Notebook)及其中的文本区(Text)

汉化的 EZWGL 的源代码可以从下列地址下载:

<ftp://ftp.turboLinux.com.cn/pub/chinese/packages/EZWGL-1.24-chinese.tar.gz>

编译时应注意以下几点:

1. Makefile 中的缺省设置为 GB, 如果使用 BIG5, 请改动定义“LANG\_SUPPORT”的值为 BIG5\_SUPPORT。如果使用 JIS 请改动定义“LANG\_SUPPORT”的值为“JIS\_SUPPORT”。如果使用 KS 则改为“KS\_SUPPORT”。
2. 测试目录(test 和 examples)下的程序还使用了非缺省字体, HZ\_SONG24, 它在 include/EZ.h 中的定义为“csc24”(汉字 24 点阵宋体)。如使用 BIG5, 请把它事先设为相应的字体再编译。
3. 在目录 test 和 examples 下的文件为 GB 演示文件。如果使用 BIG5, 请用 hc-30.tar.gz 中的 g2b 转换。
4. 使用 JIS 时, 显示的缺省字体为“jiskan16”。在程序中必须使用 EUC 编码。

目录 test 下有一文本文件 text.txt.euc 可用作 notebook.c 的测试。

#### 5. 使用 KS 时,显示的缺省字体为

```
-daewoo-mincho-medium-r-Normal-16-120-100-100-c-160-ksc5601.1987-0
```

目录 test 下有一文本文件 text.txt.ks 可用作 notebook.c 的测试。

汉化的 EZWGL 保持了与原版本的兼容,以前用 EZWGL 编的程序基本上不用改动就可以编译。而现在的标签和菜单等等都可以进行中西文并用。

汉化版本还附加了对汉字字体的配置项 EZ\_HZFONT\_NAME 和 EZ\_HZFONT\_ID,以方便使用非缺省汉字字体(如图 8.24 所示)。凡使用西文字体的地方可同样设置中文字体。例如:

```
tmp = EZ_CreateLabel(toplevel, "水平和竖直衬边(Padding)");
EZ_ConfigureWidget(tmp,
    EZ_FOREGROUND, "red",
    EZ_TEXT_LINE_LENGTH, 80,
    EZ_HEIGHT, 0,
    EZ_FONT_NAME, "-Adobe-Times-Bold-i-Normal--*-180-*-*-*-*-*-*",
    EZ_HZFONT_NAME, "ccs24",
    0);
```

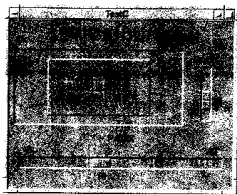


图 8.24 使用非缺省中文字体

文本区装入的注释文件可以使用 .hzfont 来改变汉字的缺省字体。例如:

```
.default
.font "-Adobe-Helvetica-Bold-R-Normal--*-180-*-*-*-*-*-*"
.hzfont "ccs24"
.fig "# ff0000"
Here is 24 point Song Font. 本区域的汉字字体
为 24 点宋体。
```

在文本区设置区域字体的函数有改变,现在的每个字符对应的 Property 都有两种字体 ID,即原来是:

```
EZ_TextSetRegionFont(Widget *textw, char *font);
```

现在是:

```
EZ_TextSetRegionFont(Widget *textw, char *fontname, char *hzfontname);
```

有些内部函数的参量也有改变,请使用时注意。



图 8.25 中文 GB 码输入(Entry)

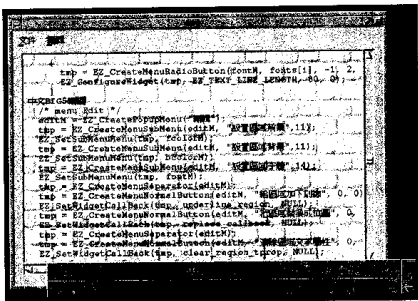


图 8.26 中文 Big5 码输入(TextArea)

## 8.8 中文编辑器——jvim, Emacs(MULE), chpower

Linux 上原有的在终端方式下使用的文本编辑软件有些可以在中文终端内编辑中文文本,如 vi, jed 等。但是由于它们是为单字节的西文字符设计的,用来编辑中文时存在各种缺陷,如光标是按单字节移动,而不是按汉字的双字节移动;删除时可能会删掉半个汉字;显示中文有时出现乱码,等等。而 X Window 下的文本编辑软件原本不可显示和输入中文,即使用本章前面提到的“包装”方法实现中文显示和输入后,由于现在中文“包装”软件的不完善,仍存在一些与终端方式下文本编辑软件类似的缺陷。所以,目前还是很有必要设计支持双字节文字显示和输入的文本编辑软件。现在已经有一些这种类型的文本编辑软件,多数是日本人显示和输入日文设计的,也有中国台湾省 UNIX 爱好者和海外华人专门为中文显示和输入设计的。本节我们介绍几种支持双字节文字输入和显示的文本编辑软件,即中文终端下的编辑器 jvim, X Window 系统的应用程序 Emacs(MULE)和 chpower。

### 8.8.1 jvim

jvim 实际上是日本人对 vim 做的一个双语扩展,而 vim 则是与 UNIX 上的 vi 几乎完

全兼容但可获得源代码的软件。jvim 支持编辑双字节字符和单字节字符混排的文件。jvim 支持几种编码方式,其中包括 EUC 编码,这使得我们可以用它在中文终端下编辑中文文件。

jvim 软件包被日本人列入 JE (Japanese Extensions) 系列,可从日本的 FTP 节点的 Linux 库中 ./JE/sources/base/ 目录下取得,如

```
ftp://ftp.u-aizu.ac.jp
/pub/os/Linux/JE/sources/base/jvim2.0r+onew2.2.10.tar.gz
```

编译 jvim 的大致步骤是:

- (1) 到 ./src 子目录下,拷贝 makefile.junx 到 Makefile。
- (2) 编辑 Makefile,首先需要修改的是 JPDEF 和 JPLIB 的定义,要改成

```
JPDEF =
```

```
JPLIB =
```

即定义为空。其他可能需要修改的变量是 PREFIX,DEFS 等,不修改一般也不会出致命错误。

(3) 用命令“make euc”编译 jvim,必须指定编码“euc”,否则编译出来的 jvim 不能处理 GB 码。

(4) 安装:“make install”和“make install-doc”命令。用户可以用 jvim 来取代系统原来的 vi。

jvim 在显示上支持双宽度光标,可使光标始终覆盖一个完整的汉字,而光标移动到 ASCII 字符上时仍是单宽度。jvim 的光标的起始位置始终在汉字的第一个字节上,一次删除键将删除整个汉字,文本插入不会在汉字的两个字节之间。

jvim 是中文终端下简单易用的中文编辑器。

### 8.8.2 Emacs{MULE}

Emacs 是 GNU 的创始人 Richard Stallman 开发的一种“可扩展的、可定制的、实时显示的编辑器和计算环境”。Emacs 远不只是一个编辑器,它功能全面而强大,是一个完整的计算支持环境。它把 Lisp 紧密集成在编辑器里,用 Lisp 来编写各种扩展和 X 界面。有人称 Emacs 是一种“生活方式”(life style),因为很多 Emacs 的使用者甚至一坐在计算机前就不离开 Emacs,编辑文件、收发邮件、编译程序、执行 shell 命令,等等。

MULE 是 GNU Emacs 的多语种增强(MultiLingual Enhancement),MULE 的文本缓冲区可以同时包含来自多种语言的字符:中文、日文、韩文、越南语、泰语、希腊语、俄罗斯语、阿拉伯语、希伯来语。MULE 还为每种语言都提供了输入法。特别的,它集成了 CXterm 的全部输入法。

Emacs 从 20.1 版本开始,已经把 MULE 作为自己的一部分包含进来了,MULE 本身从此消失。

Emacs 的编译安装过程遵循统一的 GNU 软件的安装规则。即

```
lark% ./configure
lark% make
lark# make install
```

用户在安装时可以不改任何东西。Emacs 的缺省安装目录是 `/usr/local/share/emacs`，可执行文件是 `/usr/local/bin/emacs`。

为使 Emacs 在启动时就自动设置成中文环境，用户可以在自己的主目录下创建一个名为 `.emacs` 的文件。文件 `.emacs` 的内容为 Lisp 语句：

```
(setup-chinese-gb-environment)
(global-set-key [f6] 'toggle-input-method)
```

(第一句设置中文 GB 环境，第二句将 F6 键定义为激活(或取消)缺省中文输入法的快捷键)，并在主目录下的 `.Xdefaults` 文件后附上一句

```
emacs.font:-*-fixed-medium-r-Normal-*-16-*-*-*-*-*
```

它定义了与中文字库配合使用的 ASCII 字库。注意只能使用上面的长字库名，否则中文字库无法自动加载。

各种输入法定义在 Emacs 的安装目录下 `/20.x/leim/leim-list.el` 文件中。缺省的中文输入法定义在 `/20.x/lisp/language/china-util.el` 文件中。用户可在自己的主目录下的 `.emacs` 文件后附加一条语句定义别的输入法为缺省输入法，如

```
(setq default-input-method "chinese-tonepy")
```

将把缺省输入法定义为带调拼音。

用户可在使用 Emacs 的过程中用键序“C-x RET C-\”改变当前输入法。“C-x”表示 Control-x，其他类似；“RET”表示 Return 键。Emacs 窗口的底部的命令显示区会要求用户提供输入法的名称(如图 8.27 所示)。

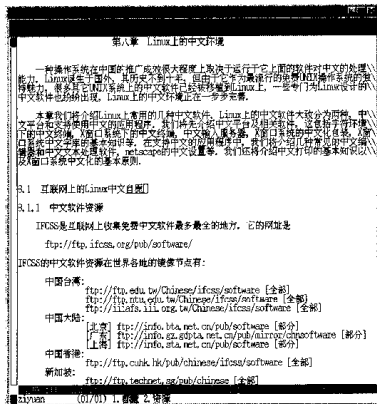


图 8.27 在 Emacs 中显示和输入中文

用户也可以为 Emacs 增加新的输入法。如在“CXterm”一节中,我们从 UC DOS 的输入法生成了 upy.tit 文件。我们可以在 Emacs 中把它转换为 Emacs 可识别的输入法文件。启动 Emacs,在 Emacs 中用键序“M-x titdic-convert upy.tit”。“M-x”表示 Meta-x 或 ESC-x。上述键序生成 upy.el 文件,编辑这个文件,找到下面一行:

```
(quail-define-package "chinese-upy" "Chinese-GB" "汉"
```

把“汉”改成一个合适的两个汉字的输入法提示符,如“全拼”。把下列几行:

```
("." . quail-next-translation)
(">" . quail-next-translation)
("=" . quail-next-translation)
("+ " . quail-next-translation)
("," . quail-prev-translation)
("<" . quail-prev-translation)
(";" . quail-prev-translation)
("_ " . quail-prev-translation))
```

改成:

```
("." . quail-next-translation-block)
(">" . quail-next-translation-block)
("=" . quail-next-translation-block)
("+ " . quail-next-translation-block)
("," . quail-prev-translation-block)
("<" . quail-prev-translation-block)
(";" . quail-prev-translation-block)
("_ " . quail-prev-translation-block)
(" " . quail-select-current))
```

把 upy.el 存盘。然后在 Emacs 中用键序“M-x byte-compile RET”后再敲入文件名 upy.el。Emacs 将把 upy.el 编译成 upy.elc 文件。把 upy.el 和 upy.elc 文件拷贝到 Emacs 的安  
装目录下 ./20.x/leim/quail/ 目录中。在 ./20.x/leim/leim-list.el 文件中定义新的输入法:

```
(register-input-method
 "chinese-upy" "Chinese-GB" 'quail-use-package
 "简拼" "UCDOS 简拼"
 "quail/upy")
```

如果用户没有往 Emacs 的安装目录下写文件的权限,可以把 upy.el 和 upy.elc 拷贝至某个  
人目录下,如 \$HOME/quail 下,再把类似上面的四行加入自己的 .emacs 文件中:

```
(register-input-method
 "chinese-upy" "Chinese-GB" 'quail-use-package
 "简拼" "UCDOS 简拼"
 "$ $HOME/quail/upy")
```

唯一不同的是最后一行的文件目录名。

### 8.8.3 chpower

chpower 全称是 ChinesePower, 是一个 X Window 下双字节的字处理软件(如图 8.28 所示)。它的主要特点是:



图 8.28 在 chpower 中编辑中文文件

- 支持在同一文件中使用 GB, HZ, Big5, EUC-JIS, EUC-KS, UNICODE, UTF7, UTF8 等多种编码
- 支持 GB, Big5, JIS, KS 等编码的输入法
- 横向排版和纵向排版
- 所见即所得, WYSIWYG (What You See is What You Get)
- 支持 TTF (True Type Font) 和 HBF (HZ Bitmap Font) 两种字库
- 图形嵌入
- 同时使用多种字号
- PS 在线打印
- 编码转换

等。它最突出的特点应该是“所见即所得”和“支持 TTF 字库”。中文文字处理软件如果要保证同西文相仿的输出质量, 就必须采用 TTF 字库。

chpower 可从下列地址或其镜像地址取到:

<ftp://ftp.ifess.org/pub/software/x-win/editor/chpower-3.0.tar.gz>

编译安装按 chpower-3.0 README 文件的说明, 不会遇到什么问题。

## 8.9 英汉/汉英辞典——cdict(xcdict)、xdict

很多用户由于英语水平所限,在使用计算机进行工作或浏览互联网页面的时候,希望能有一个在线的英汉/汉英辞典,以便在必要的时候能随时查阅不认识的英文单词或中文词汇翻译成英文。这里介绍的 cdict(xcdict)和 xdict 就是这样的工具。

### 8.9.1 cdict(xcdict)

cdict(xcdict)运行在 UNIX 系统的普通中文终端或 X Window 系统上,可以随时帮助用户进行英汉/汉英对译。

cdict 最初是一个只支持中文 Big5 码的英汉/汉英辞典,它使用 21 世纪字典 5.0 版的字库和倚天 ET2000 所附的英汉字典,所含英文词汇超过十万。本书作者之一(陈向阳)将它改编为支持中文 GB 码,于明俭进一步将它改编成 X Window 系统下的应用程序。

cdict 中文 GB 码文本版和 X Window 版(xcdict)可分别从

<ftp://ftp.turboLinux.com.cn/pub/chinese/packages/>

目录得到。两个软件包的安装不会有什么困难,请参照软件包中的说明文件。

cdict 的文本版的英汉辞典用法如下:

```
% cdict word[ \ * ]
```

即它不仅支持每次查阅一个单词,还支持用通配符一次查阅同一词根的多个单词。

如果安装了 cdict 的 X Window 版,可用“xcdict”命令启动 cdict,并配合中文输入服务器 Chinput 进行中文词汇的输入(如图 8.29 和 8.30 所示)。

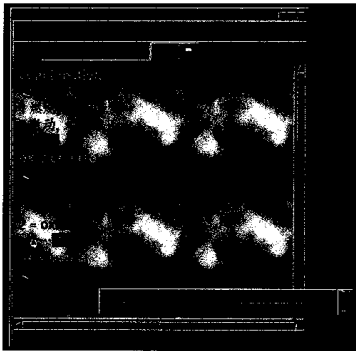


图 8.29 英汉/汉英辞典 cdict 的 GB 码 X Window 版

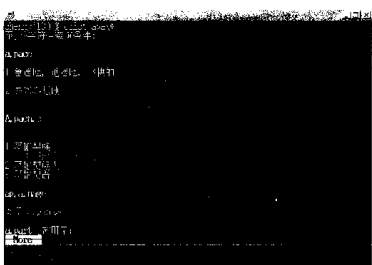


图 8.30 英汉/汉英辞典 edict 的 GB 码文本版

### 8.9.2 xdict

xdict 是一个 X Window 界面的英汉小词典,由伏建军开发。大约有 177000 个词条。它包含了很多最新的科技词汇,特别是计算机类词汇(如图 8.31 所示)。

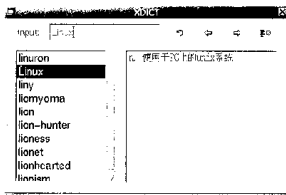


图 8.31 英汉小词典 xdict

用户可以从下列目录下载 xdict:

<http://ftp.turboLinux.com.cn/pub/chinese/packages/>

xdict 的界面用 Qt 设计。所以在安装 xdict 之前要先安装 Qt。请参考下列页面获得有关 Qt 的信息:

<http://www.troll.no>

xdict 的安装请参考软件包中的 README 文件。

## 8.10 汉化的 X 窗口管理器 fvwm95

窗口管理器的汉化实际上并不是特殊的汉化。使用中文化“包装”后它同样可以显示和输入中文。但是,窗口管理器软件是一种特殊软件,它负责其他窗口在屏幕上的行为。其启动也一般在其他软件之前。所以对其进行专门的汉化对系统的稳定性是有益的(如

它可以不受某些动态库的影响,在启动其他软件时不必把“包装”功能强加到它们身上)。

于明俭(justiny@turboLinux.com.cn)对 Linux 系统下较流行的窗口管理器fvwm95-2.0.43a-Autoconf 软件包进行了汉化。经过汉化后,它可以支持 GB,BIG5,日文和韩文EUC 等编码(如图 8.32)。

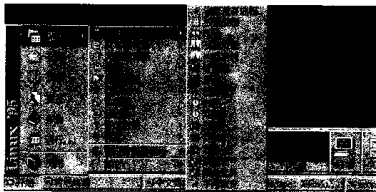


图 8.32 汉化的 fvwm95

汉化的 fvwm95 的源代码 fvwm95-2.0.43a-Autoconf-chinese.tar.gz 可从

<ftp://ftp.turbolinux.com.cn/pub/chinese/packages/>

下载。所用中文字库可从

<ftp://ftp.turbolinux.com.cn/pub/chinese/fonts/pcl/>

目录下取得。编译和安装过程请参照软件包中说明。

很多人都曾经问过这样的问题:“我的 Netscape 窗口顶端的标题栏怎么显示不了中文?”这是因为 X 窗口的标题栏属于 X 窗口管理器管理,如果 X 窗口管理器不支持中文显示,标题栏自然无法显示中文。我们可以使用汉化的 fvwm95,则所有窗口的标题栏就都可以显示中文了。

## 8.11 Big5 码的中文 X 服务器——CXWin

CXWin 是台湾人开发的对 XFree86 的一个修补(patch),它使 Linux 的 X Window 系统可以支持中文显示。目前的 CXWin 仅支持 Big5 编码。CXWin 实现了中文的下拉菜单,中文的窗口标题栏,以及在各种不同的窗口管理器及应用软件中显示中文。

用户可从:

<ftp://Linux.cis.nctu.edu.tw/packages/X/Xserver/CXwin/3.3.1/>

取得 CXWin 3.3.1。如果用户有 XFree86 的源代码包,也可以在上述地址取得 CXWin 对 XFree86 的修补文件并自行编译成 X 服务器。

用户需要安装的实际上只有一个 X 服务器,首先用 gzip 对 XF86\_SVGA.gz 文件解压缩:

```
lark# gzip -d XF86_SVGA.gz
```

然后备份原来的 X 服务器:

```
lark# cd /usr/X11R6/bin
```

```
lark# mv XF86_SVGA XF86_SVGA_BACKUP
```

再把 CXWin 的 X 服务器放到适当的地方：

```
# mv /tmp/XF86_SVGA /usr/X11R6/bin
# chown root.bin XF86_SVGA
# chmod 4755 XF86_SVGA
# ln -sf XF86_SVGA X
```

如果用户取来的是 RPM 包,可以执行如下命令安装 CXWin:

```
# rpm -ivv XFree86-SVGA-3.3.1-1c.i386.rpm
```

CXwin 需要下面 4 种 Big5 字库才能正常使用: taipei15, taipei16, taipei20 与 taipei24。

## 8.12 中文文本处理软件——LaTeX 的 CJK 扩展

TeX/LaTeX 是一套出色的文本处理软件,它因其强大的排版能力和优秀的输出品质很早就为广大用户、特别是学术界的朋友所喜爱和使用。CJK 是 LaTeX 的宏文件包。它使用户可以在 LaTeX 文件中使用 CJK (Chinese/Japanese/Korean) 的文字编码。LaTeX 的 CJK 扩展使用户可以用 LaTeX 中文文本的排版和打印。

安装 CJK 前必须首先安装一种 TeX/LaTeX 套件,teTeX 是目前较好的一种。很多 Linux 的发行套件都包含了 teTeX,如 Slackware 的 T 系列。如果用户需要更新的版本,可从

```
ftp://ftp.tex.ac.uk/pub/tex/systems/UNIX/teTeX/distrib/
```

目录或其镜像地址取来。

CJK 可从:

```
ftp://ftp.ifcss.org/pub/software/tex/
```

或 IFCS 的镜像节点取来,目前的版本是 CJK-4.1.3.src.tar.gz。

CJK 支持中文 GB 码和 Big5 码,用户可以根据自己的需要决定安装对某一种或对两种编码的支持。

用户还需要取来中文字库。CJK 可使用 HBF 和 TTF 字库。用户可从

```
ftp://ftp.ifcss.org/pub/software/fonts/gb/hbf/
```

取来 48x48 点阵的中文 GB 码四种字体(宋体、黑体、楷体、仿宋体)的 HBF 字库,需要的文件有:

```
cc48s.1 cc48s.2 cc48s.hbf
cc48h.1 cc48h.2 cc48h.hbf
cc48k.1 cc48k.2 cc48k.hbf
cc48fs.1 cc48fs.2 cc48fs.hbf
cc48.sym
```

对于 Big5 码,用户可使用免费的 TTF 文件,从下列目录

```
ftp://ftp.ifcss.org/pub/software/fonts/big5/ms-win/
```

取来 ntu\_\*.arj, ntu\_\*.a01, ntu\_\*.a02, ntu\_\*.a03。它们是用 ARJ 压缩的,解压缩后形成 8 个 TTF 文件:

```
ntu_br.ttf      ntu_fs_m.ttf      ntu_kai.ttf      ntu_li_m.ttf
ntu_mb.ttf      ntu_mm.ttf      ntu_mr.ttf      ntu_tw.ttf
```

CJK 的安装步骤较多,用户在安装时务必按照说明谨慎进行。下面说明怎样安装 CJK 的 GB 码支持。安装 CJK 的 Big5 码支持的方法请参看 CJK 软件包的说明文件。

(1) 首先,要了解 `teTeX` 套件安装在系统的哪个目录下,Slackware 是将它安装在 `/usr/lib/teTeX` 目录下,teTeX 套件缺省的安装目录是在 `/usr/local/teTeX`。但用户可以用

```
lark% which latex
```

来确定一下。

假设 `teTeX` 安装在 `/usr/local/teTeX` 目录下,teTeX 定义了一个内部变量 `TEXMF` 指向 `/usr/local/teTeX/texmf` 目录。我们以后的说明经常引用这个变量,为方便起见,用户可以在安装过程中定义这个变量:

```
lark# setenv TEXMF /usr/local/teTeX/texmf
```

但使用 `LaTeX` 不必定义这个变量。

(2) 将 `CJK-4.1.3.src.tar.gz` 打开在某个临时目录下,产生 `CJK/` 目录。`CJK/4.1.3/doc` 目录下有关于 CJK 安装和使用的重要文档,用户务必要认真阅读,特别是 `INSTALL` 文件和 `teTeX/` 子目录下的文件。

(3) 进入 `CJK/4.1.3` 目录,将 `texinput/` 目录名改成 `CJK`,把整个目录拷贝到 `$TEXMF/tex/latex/` 目录下。

(4) 创建 `$TEXMF/fonts/hbf/chinese/` 目录,将中文 HBF 文件拷贝到这个目录下。

(5) 进入 `CJK/4.1.3/utls/hbf2gf/` 目录,编译 `hbf2gf` 程序。拷贝可执行文件到 `/usr/local/teTeX/bin` 目录下。根据 `cfg/` 目录下的 `gsfsl4.cfg` 文件修改创建 `cc48s.cfg`,`cc48h.cfg`,`cc48k.cfg`,`cc48fs.cfg` 这四个文件。下面给出 `cc48s.cfg` 的例子:

```
hbf_header      $TEXMF/fonts/hbf/chinese/cc48s.hbf
mag_x           1
threshold       128
comment         GB SongTi

design_size      12
target_size     12

y_offset        -13

nmb_files       -1

output_name     cc48s

checksum        123456789

dpi_x           300
```

|                |                                          |
|----------------|------------------------------------------|
| pk_files       | No                                       |
| tfm_files      | Yes                                      |
| long_extension | No                                       |
| coding         | codingscheme GuoBiao encoded TeX text    |
| pk_directory   | \$ TEXMF/fonts/pk/modelss/chinese/cc48s/ |
| tfm_directory  | \$ TEXMF/fonts/tfm/chinese/cc48s/        |

hbf\_header 项要与中文 GB 码 HBF 文件存放的地点一致。latex 和 dvips 运行时分别产生的 TFM 和 PK 型字库将存放在 tfm\_directory 和 pk\_directory 里。这两个目录无需手动产生, latex 和 dvips 运行时会自动创建它们。

(6) 创建 \$ TEXMF/hbf2gf/ 目录, 将 cc48s.cfg, cc48h.cfg, cc48k.cfg, cc48fs.cfg 这四个文件拷贝到 \$ TEXMF/hbf2gf/ 下。

编辑 \$ TEXMF/fontname/special.map 文件, 在文件末尾增加四行:

```
cc48s      chinese      cc48s
cc48fs     chinese      cc48fs
cc48k      chinese      cc48k
cc48h      chinese      cc48h
```

并将文件中原有的下面一行

```
cc          public      concrete
```

移至文件的最后(在上面 cc48... 四行后面), 以免其成为搜索中文字库名 cc48.. 的障碍。

(7) 依照 CJK/4\_1.3/doc/teTeX/ 目录中的 \*.diff 文件, 对 /usr/local/teTeX/bin 目录下的相应文件进行修改。由于每个文件的改动都不是太多, 建议用户动手一行行修改, 而不要用 patch 命令, 因为系统使用的 teTeX 的版本可能与 CJK 软件包做 \*.diff 文件时使用的版本有所不同。

(8) 在 \$ TEXMF/tex/latex/CJK/GB 目录下根据 c10fs.fd 的例子编制 c10song.fd, c10hei.fd, c10kai.fd 和 c10fs.fd 文件。可将 c10fs.fd 中的 "gsfs14" 字符串更改为 "cc48fs" 以生成新的 c10fs.fd 文件, 再对新的 c10fs.fd 进行修改(字符串 "C10fs.fd", "fs", "cc48fs") 得到 c10song.fd, c10hei.fd 和 c10kai.fd。

(9) 把 /usr/local/teTeX/bin 加到环境变量 PATH 的最前端, 然后运行

```
lark# texconfig rehash
lark# texconfig hyphen
```

更新 TeX 输入文件数据库 \$ TEXMF/ls-R 文件。

现在, 中文 GB 码 LaTeX 就已安装完毕。我们可以用 CJK/4\_1.3/examples/GB.tex 来做测试:

```
lark% latex GB.tex
```

latex 会根据 GB.tex 使用了哪些汉字来产生一些 TFM 文件, 这些 TFM 文件一旦产生就被放进了 cc48s.cfg 等文件中指定的 tfm\_directory 目录中, 下一次用到同样的汉字时,

latex 就会使用以前产生的 TFM 文件,而不必再产生一次了。

latex 完成后,用 dvips 命令产生 PostScript 文件:

```
lark% dvips GB
```

dvips 会根据 GB.tex 使用了哪些汉字来产生一些 PK 文件,这些 PK 文件一旦产生就被放进了 cc48s.cfg 等文件中指定的 pk\_directory 目录中,下一次用到同样的汉字时,dvips 就会使用以前产生的 PK 文件,而不必再产生一次了。

如果 dvips 顺利完成,将产生 GB.ps,用户可以用 gs 命令读 GB.ps(如图 8.33 所示),也可以用 xdvi 命令来读 GB.dvi。

中文 GB 码的 LaTeX 文件是书写规则很简单,下面是一个简单的例子:

```
\documentclass[12pt]{article}
\usepackage{CJK}
```

```
\begin{document}
```

```
\begin{CJK*}{GB}{song}
```

这一行是中文宋体,These are Chinese Song Ti characters.

```
\CJKfamily{fs}
```

这一行是中文仿宋体,These are Chinese FangSong Ti characters.

```
\end{CJK*}
```

```
\end{document}
```

我们看到:

- (1) 在文件的开头要加上 `\usepackage{CJK}` 一句。
- (2) 所有的中文要放在 `\begin{CJK*}{GB}{...}` 和 `\end{CJK*}` 之间。
- (3) 用 `\CJKfamily{...}` 改变字体。
- (4) 其他与西文 LaTeX 一致。

LaTeX 对西文的处理不因使用 CJK 扩展而改变。

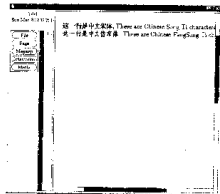


图 8.33 用 Ghostview 浏览 CJK 生成的 PostScript 文件

由于 CJK 的安装比较繁琐,为方便用户,本书著者之一(陈向阳)将 CJK 的 GB 码支持重新打包,提供了一个较为自动化的安装过程。这个打包还提供了 GBK 码的扩展,修正了对使用中文 PostScript 字库的支持。用户可在下列地址下载这个 CJK 的 GB 码支持

打包：

```
ftp://ftp.turboLinux.com.cn/pub/chinese/packages/cjk-4.1.3.gb.tar.gz
```

这个打包对应的 TeTeX 版本是 tetex-0.4 系列,如 0.4pl8 等。用户在安装过 TeTeX 后,可以用如下简单的步骤安装 CJK 的 GB 码部分。

首先将 cjk-4.1.3.gb.tar.gz 展开在某个工作目录下,然后:

```
# cd cjk-4.1.3.gb
# make
# make install
```

安装过程将自动侦测 TeTeX 的安装目录,把 CJK 扩展安装到正确的目录里。用户可以在安装完成后把下列新增加的可执行文件连接到 /usr/bin 目录下:

```
MakeTTFPK, hbf2gf, gbconv, gbklatex, b5latex
```

并执行下列命令以更新 TeTeX 文件名数据库:

```
texconfig rehash
```

现在用户就可以使用中文 GB 码的 LaTeX 了,它使用了中文 GB 码的 HBF 字库。如果用户想产生更清晰更优美的中文输出,就必须使 LaTeX 使用中文 PostScript 字库。用户首先必须按 8.14 节的说明把中文 PFM 文件拷贝到 TeTeX 的 TFM 文件目录中。在书写中文 LaTeX 文件时必须使用字库编码“pmC”,如:

```
\begin{CJK * }{[pmC]}{GB}{Song}
.....
\end{CJK * }
```

或

```
\begin{CJK * }{[pmC]}{Bg5}{Ming}
.....
\end{CJK * }
```

等。在编译 LaTeX 文件时必须使用“gbklatex”和“b5latex”命令来代替“latex”命令:

```
# gbklatex GBdoc.tex
# b5latex Bg5doc.tex
```

在产生 PostScript 文件时可以简单地使用

```
# dvips GBdoc
# dvips Bg5doc
```

## 8.13 Netscape 的中文设置

Netscape Communicator(或 Navigator)是目前最流行的万维网(World Wide Web,简称 WWW)浏览器(Browser)。UNIX 平台上的 Communicator 可以支持多种语言编码显示,包含中文 GB 码和 Big5 码。但有些用户在设置 Communicator 显示中文时还是会碰到一些问题。本节我们介绍 Netscape Communicator 中文设置的一般方法。

Netscape 按一定的规则在 X 服务器上搜寻中文 X 字库。它的缺省规则是:对 GB 码,

优先使用字库名最后两个域是“gb2312.1980-0”的中文字库,如果找不到这种字库,才使用字库名最后两个域是“gb2312.1980-1”的中文字库;对 Big5 码,使用字库名最后两个域是“big5-0”,“big5.et-0”,“big5.et.ext-0”,“big5.etext-0”,“big5.hku-0”,“big5.hku-1”,“big5.pc-0”,“big5.shift-0”,“hpbig5-”的中文字库。所以,如果系统上有中文字库,但是没有以这些字符串结尾的中文字库,Netscape 仍然不能显示中文。在这种情况下,我们可以用设置字库别名的方法来解决这个问题。例如,系统上有中文 GB 码字库 cclib16 和 cclib24,我们可以编辑字库所在目录下的 fonts.alias 文件,给它们加上这样的别名:

```
-cclib-song-medium-r-normal--16-160-75-75-c-160-gb2312.1980-0 cclib16
-cclib-song-medium-r-normal--24-240-75-75-c-240-gb2312.1980-0 cclib24
```

即我们给它们定义以“gb2312.1980-0”作为最后两个域的 XLFD 形式的别名。然后用

```
lark% xset fp rehash
```

让 X 服务器更新字库信息。我们再次启动 Netscape 的时候,就可以使用这些中文字库了。

Netscape 对搜索到的中文字库,按 XLFD 字库名的头两个域分类,如上例中的两个中文字库将被分入 Song(Celib)一类,而:

```
-sun-song-medium-r-Normal--14-120-75-75-c-120-gb2312.1980-0
-sun-song-medium-r-Normal--16-140-75-75-c-140-gb2312.1980-0
```

则被归入 Song(Sun)类。其他还可能有 Kai(Celib),Hei(Celib),Fangsong(Celib)等类。对每一种编码,Netscape 在同一时刻只能使用一类中文字库。我们可以设置 Netscape 使用哪类中文字库。GB 码字体的设置步骤是:启动 Netscape,打开 Edit → Preferences 窗口,选择 Appearance → Fonts。在右边“For the Encoding:”项里,选择“Simplified Chinese(gb.2312-80)”。在其下“Variable Width Font:”和“Fixed Width Font:”里,选择我们希望使用的变化宽度和固定宽度的中文 GB 码字库类及显示尺寸(“Size:”)。“Size:”指的是 HTML 文件缺省的字体的显示尺寸。“Allow scaling”指是否允许使用可缩放字体。可缩放字体可使中文显示字体的大小级别清楚,如在只有 16 和 24 点阵的字库的情形下,可显示中间级别如 20 点阵的字体。但使用可缩放字体减慢 Netscape 的显示速度(如图 8.34 所示)。

设置使用一种 GB 码字体后,可能需要重新启动 Netscape 以使设置生效。

设置好字库后,要使 Netscape 可以显示中文,还必须设定 Netscape 当前使用的编码体系。在 Netscape 窗口中,打开 View → Encoding 菜单,选择 Simplified Chinese (GB2312),即设置 Netscape 目前使用 GB 码。如果想使下次启动 Netscape 时,自动使用 GB 码,可在 View → Encoding 菜单中,选择最底部的 Set Default Encoding,则把当前编码设为缺省编码。

Netscape 本身支持中文显示,但却不支持中文输入,而且如果有些中文页面的 HTML 文件中如果设置 charset 不等于“gb2312”,Netscape 将无法显示这些页面中的中文。我们可以用中文化“包装”解决在 Netscape 中的中文输入和显示。

我们对 X Window 使用中文化“包装”后,也可以解决 Netscape 的菜单汉化问题。实现方法很简单,只要将 Netscape 软件包中的资源文件 Netscape.ad 中的所有定义菜单项的字符串翻译成中文,然后在中文化“包装”后的环境下,运行 Netscape 时,将资源文件指

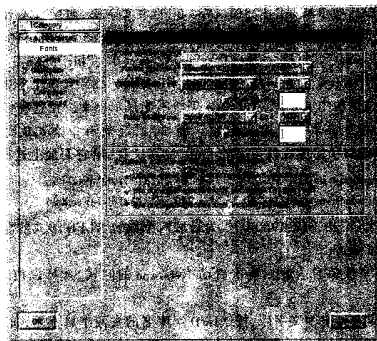


图 8.34 在 Netscape Communicator 中设置中文字库

定为 Netscape.ud,如

```
# sh -c 'export XENVIRONMENT = /usr/lib/netscape/Netscape.ud; netscape'
```

就可以实现 Netscape 菜单中文化了(如图 8.35 所示)。



图 8.35 中文 Netscape Communicator

## 8.14 中文打印和中文 PostScript 字库

中文打印是 Linux 中文化的一项重要内容。很多软件的最终目的是形成硬拷贝,即提供符合各种要求的、良好的打印输出。

目前 Linux 上的打印系统通常是用 nenscript(或 mpage)把非 PostScript 文件转换成 PostScript 文件(对 PostScript 文件则不作改变),由 Ghostscript 对 PostScript 文件进行转换后输送到打印机上。所以中文打印的关键在于如何生成 Ghostscript 可以阅读的中文

PostScript 文件。但由于没有中文 PostScript 字库体系,现有的可形成 PostScript 文件的中文软件(如 chppower 和中文 LaTeX(CJK))大都是把用到的汉字字型以位图或轮廓形式全部存储在 PostScript 文件中,而不是像西文那样,使用外部的 PostScript 字库体系。这就导致 PostScript 文件体积太大,且打印效果欠佳。所以,必须尽快确立中文 PostScript 字库体系,构造中文 PostScript 字库,使中文 PostScript 文件的结构尽可能地接近西文 PostScript 文件,以保证尽可能地与现有的软件兼容。

本书的著者之一(陈向阳)分析了目前各种中文软件对中文字库的处理方法,并参考了日文文字处理软件,总结出一套中文 PostScript 字库,初步建立了构造和使用中文 PostScript 字库的基本原则。这套中文 PostScript 字库通过 Ghostscript 程序使用从中文 TrueType 字库产生的中文 PFB 字库。TrueType 字库在中文打印方面比点阵字库所具有的优势是不言而喻的。所以以后 UNIX 上中文打印的发展方向是利用 TrueType 字库更方便地产生更高质量的中文输出。

中文 PostScript 字库软件包当前版本为 chpfb-1.3.tar.gz,存放地址为:

`ftp://ftp.turboLinux.com.cn/pub/chinese/packages/`

以前的版本曾经叫作 psfonts-gb 和 gbpfb。用户可以随时浏览上述网址以获得软件的最新版本。

#### 8.14.1 安装中文 PostScript 字库

为使 Ghostscript 能显示中文 PostScript 文件,我们必须把中文 PostScript 字库加进 Ghostscript 的字库目录,并在 Fontmap 文件中增加相应的定义。但我们绝对无须修改 Ghostscript 的任何源代码。

安装中文 PostScript 字库并设置 Ghostscript 可分下面几步:

(1) 首先要确定 Ghostscript 相关文件(初始化文件、字库等)安装在哪个目录下。我们可以用“gs -help”命令,其输出应该包含类似下面的两行:

```
Search path:
. : /usr/share/ghostscript/5.10 :
  /usr/share/ghostscript/fonts
```

可以看出,在本例中,Ghostscript 相关文件安装在 /usr/share/ghostscript 及其子目录下。

(2) 安装 FreeType 1.2。chpfb 中的 ttf2pfb.c 程序利用 FreeType 提供的 TrueType 引擎将 TrueType 字库中的字形转换为 PFB 字库中的字形描述。安装 FreeType 的说明请参考 8.5 节。

(3) 将软件包 chpfb-1.3.tar.gz 打开在某个临时目录下。编译和安装一些转换程序:

```
# cd chpfb-1.3
# make prefix=/usr
# make install
```

安装程序将把下列程序安装到 /usr/bin 目录下:

```
ttf2pfb, t1asm, gbpfb, b5pfb,
ns2gbps, ns2b5ps, t1t2gbps, t1t2b5ps, jy2gbps, a2gbps, a2b5ps
```

把下列文件安装到 /usr/lib/chpfb 目录下:

```
GBK_codeRange, BIG5_codeRange, printafm.ps, Fontmap.zh
```

如果用户想安装到别的目录下,在上面的“make prefix = /usr”命令中把“/usr”改成其他目录就可以了,如“make prefix = /home/jack”将把 chpfb 的相关文件安装到/home/jack/bin 和/home/jack/lib/chpfb 目录下。

(4) 将 Fontmap.zh 文件附加到 Ghostscript 原有的 Fontmap 文件后,如:

```
# cat Fontmap.zh > > /usr/share/ghostscript/5.10/Fontmap
```

(5) 将 ./fonts 目录下的所有文件(GB\*.ps,BIG5\*.ps)拷贝到 Ghostscript 的字库目录下:

```
# cd fonts
# cp GB*.ps BIG5*.ps /usr/share/ghostscript/fonts
```

(6) 从中文 TrueType 字库产生中文 PFB 字库并安装到 Ghostscript 的字库目录下  
chpfb 软件包中的 tt2pfb 程序用来从中文 TrueType 字库产生原始的 Type1 型中文 PostScript 字库。可以使用的中文 TrueType 字库的编码必须是 Unicode, GB2312 或 Big5。用户可以用 FreeType 软件包中的“ftdump”程序来获得一个 TrueType 字库的编码。如果 some.ttf 是一个中文 TrueType 字库,在命令

```
# ftdump some.ttf
```

的输出结果中有类似

```
encoding 1: Windows Unicode
```

的一行,表明这个 TrueType 字库使用 Unicode 编码。如果有

```
encoding 0: Windows GB2312
```

表明这个 TrueType 字库使用 GB 码。如果有

```
encoding 2: Windows Big 5
```

的一行,表明这个 TrueType 字库使用 Big5 编码。GB 码的中文 TrueType 字库只包含 GB2312-80 字符集(共约 7000 多个字形),Big5 码的中文 TrueType 字库只包含大五字符集(共约 13000 多个字形)。而 Unicode 编码的中文 TrueType 字库可能只含有 GB2312-80 字符集或大五字符集,也可能包含 GBK 字符集(超过 20000 个字形),GBK 字符集是 GB2312-80 字符集的超集,完全包含 GB2312-80 字符集和大五字符集。

最初的 tt2pfb.c(作者:Chun-Yu Lee <d791013@ce.ntu.edu.tw>)不能从 Unicode 编码的中文 TrueType 字库产生 GBK 或 Big5 编码的 PostScript 字库。陈向阳在程序中加进了 GBK 或 Big5 到 Unicode 的编码转换部分,使程序可以按 GBK 或 Big5 码从 Unicode 编码的中文 TrueType 字库中索引中文字形。

tlasm 用来将原始的 Type1 型中文 PostScript 字库转变为 PFA 或 PFB 格式。

Ghostscript 和 afm2tfm(在 TeTeX 软件包中)用来从 PostScript 字库文件产生 AFM 和 TFM 文件。

chpfb 软件包中提供了两个 shell 脚本文件 ‘gbpfb’ 和 ‘b5pfb’, 它们把从中文 TrueType 字库文件产生中文 PFB 字库文件和相关的 TFM 文件的过程全部自动化了。gbpfb 命令的用法是:

```
gbpfb ttf_file face
```

face 可以为 song/hei/fs/kai, 分别代表“宋/黑/仿宋/楷”4 种字体。b5pfb 命令的用法是:

```
b5pfb ttf file face
```

face 可以为 ming/hei/fs/kai, 分别代表“明(宋)/黑/仿宋/楷”4 种字体。例如, 如果用户有一种含 GBK 字符集的中文宋体 TrueType 字库 song.ttf, 用户可以用如下命令

```
# gbpfb song.ttf song
# b5pfb song.ttf ming
```

产生 CBK 字符集和 Big5 字符集的中文宋体 PostScript 字库。

一个基本的中文 PostScript 字库的集合包含中文 GBK 字符集和 Big5 字符集“宋/黑/仿宋/楷”四种字体的 PostScript 字库。对每种字体, 将有 126 个 GBK 字符的 PFB 文件, 每个 PFB 文件表示 GBK 编码的一个区:

```
gbsong81.pfb  —— gbsongfe.pfb
gbhei81.pfb   —— gbheife.pfb
gbkai81.pfb   —— gbkaife.pfb
gbfs81.pfb    —— gbfsfe.pfb
```

对每种字体, 将有 89 个 Big5 字符的 PFB 文件, 每个 PFB 文件表示 Big5 编码的一个区:

```
b5minga1.pfb —— b5mingf9.pfb
b5heia1.pfb  —— b5heif9.pfb
b5kaia1.pfb  —— b5kaif9.pfb
b5fsa1.pfb   —— b5fsf9.pfb
```

gbpfb 和 b5pfb 程序将在产生这些 PFB 文件后把它们放进当前目录的 fonts/子目录中, 并同时将产生的 TFM 文件放进 tfm/子目录中。用户必须把这 860 个( $126 * 4 + 89 * 4$ )PFB 文件(约 100MB)拷贝到 Ghostscript 的字库目录中, 如/usr/share/ghostscript/fonts/(对于 tfm/子目录中的文件, 它们将用在 LaTeX 的 CJK 扩展中。用户可以把它们拷贝到 teTeX 的 TFM 文件目录中, 如果 teTeX 的安装目录为/usr/lib/teTeX, 则可以把这些 TFM 文件拷贝到

```
/usr/lib/teTeX/texmf/fonts/tfm/chinese/
```

目录中)。

**注意:** 如果用户没有 root 权限, 不能把文件(如步骤 4, 5 和 6 中)放进系统目录, 则可使用个人目录(如 \$HOME/gsfonts)存放这些文件。把 ./fonts 目录下所有字库文件(GB\*.ps, BIG5\*.ps)和所有 PFB 文件拷贝到 \$HOME/gsfonts 目录中, 把 Fontmap.zh 也拷贝到 \$HOME/gsfonts 目录中改成 Fontmap。设置环境变量 GS\_LIB 指向 \$HOME/gsfonts:

```
sh 或 bash:
GS_LIB = $HOME/gsfonts
export GS_LIB

csh 或 tcsh:
setenv GS_LIB $HOME/gsfonts
```

(7) 现在 Ghostscript 已经能显示中文 PostScript 文件了。先试一下 ./examples 目录下的几个例子。demo.gps 和 demo.bps 是直接手工编写的中文 GB 码和 Big5 码的 PostScript 文件, gbk.gps 是 GBK 字符集的全部字符列表, big5.bps 是 Big5 字符集的全部

字符列表(如图 8.36 所示)。



图 8.36 中文 PostScript 字库示例

如果用户的打印机队列使用 Ghostscript 作为输入过滤器(如 RedHat 缺省的打印机队列配置软件 printtool 就是这样),现在就可以打印中文 PostScript 文件了。如果用户尚没有设置这样的打印机队列,可参考第五章“Linux 系统管理”及本软件包的 README 文件进行设置,以实现中文打印的目的。

### 8.14.2 中文 PostScript 字库

中文 PostScript 字库目前包括 GBK 字库和 Big5 字库。

中文(GBK)PostScript 字库目前包含 4 种字体,即宋体、黑体、楷体、仿宋体。对每种字体,有 126 个中文 Type1 PostScript 字库文件,以 PFB(Printer Font Binary)的格式存储,它们分别对应于中文 GBK 编码的 126 个区(0x81-0xFE):

|               |                |   |              |                 |
|---------------|----------------|---|--------------|-----------------|
| GB-Song81     | —— GB-Song     | : | gbsong81.pfb | —— gbsongfe.pfb |
| GB-Hei81      | —— GB-Hei      | : | gbhei81.pfb  | —— gbheife.pfb  |
| GB-Kai81      | —— GB-Kai      | : | gbkai81.pfb  | —— gbkaife.pfb  |
| GB-FangSong81 | —— GB-FangSong | : | gbfs81.pfb   | —— gbfsfe.pfb   |

上面几行中左侧是字库名,右侧是对应的字库文件。

在上述 Type1 中文 PostScript 字库的基础上,定义了 4 种 Type0(复合型)中文(GBK)基本字库:

|             |                         |
|-------------|-------------------------|
| GB-Song     | ( 字库文件 GB-Song.ps )     |
| GB-Hei      | ( 字库文件 GB-Hei.ps )      |
| GB-Kai      | ( 字库文件 GB-Kai.ps )      |
| GB-FangSong | ( 字库文件 GB-FangSong.ps ) |

它们的 FMapType 是 2,即每次从要显示的字符串中读出两个字节,从第 1 个字节(即汉字的高 8 位)确定子字库,用第 2 个字节作为索引从子字库中取出汉字的轮廓字型。如对 GB-Song 字库,若读出的字符串的编码是 0xc8cb,从 0xc8 我们确定子字库 GB-Songc8,从子字库文件 gbsongc8.pfb 中取出对应第 2 个字节 0xcb 的轮廓字型,即“人”字。

在中文(GBK)基本字库和西文 Type1 型 PostScript 字库的基础上,定义了四种 Type0 中文(GBK)正体字库:

|                     |                               |               |   |
|---------------------|-------------------------------|---------------|---|
| GB-Song-Regular     | GB-Song                       | + Times-Roman |   |
|                     | ( 字库文件 GB-Song-Regular.ps     |               | ) |
| GB-Hei-Regular      | GB-Hei                        | + Times-Bold  |   |
|                     | ( 字库文件 GB-Hei-Regular.ps      |               | ) |
| GB-Kai-Regular      | GB-Kai                        | + Helvetica   |   |
|                     | ( 字库文件 GB-Kai-Regular.ps      |               | ) |
| GB-FangSong-Regular | GB-FangSong                   | + Helvetica   |   |
|                     | ( 字库文件 GB-FangSong-Regular.ps |               | ) |

它们的 FMapType 是 4,即从要显示的字符串中读出 1 个字节,根据其最高位是“0”还是“1”确定是西文字符还是中文字符,从而决定查找西文 PostScript 字库还是再取 1 个字节查找中文(GB)基本字库。

同上,定义了 4 种 Type0 中文(GBK)斜体字库:

|                    |                              |                     |   |
|--------------------|------------------------------|---------------------|---|
| GB-Song-Italic     | GB-Song                      | + Times-Italic      |   |
|                    | ( 字库文件 GB-Song-Italic.ps     |                     | ) |
| GB-Hei-Italic      | GB-Hei                       | + Times-BoldItalic  |   |
|                    | ( 字库文件 GB-Hei-Italic.ps      |                     | ) |
| GB-Kai-Italic      | GB-Kai                       | + Helvetica-Oblique |   |
|                    | ( 字库文件 GB-Kai-Italic.ps      |                     | ) |
| GB-FangSong-Italic | GB-FangSong                  | + Helvetica-Oblique |   |
|                    | ( 字库文件 GB-FangSong-Italic.ps |                     | ) |

定义了 4 种 Type0 中文(GBK)固定宽度字库(为使中文字符的宽度是西文字符的 2 倍,将西文字符的宽度缩减 1/6):

|                   |                             |                |   |
|-------------------|-----------------------------|----------------|---|
| GB-Song-Fixed     | GB-Song                     | + Courier      |   |
|                   | ( 字库文件 GB-Song-Fixed.ps     |                | ) |
| GB-Hei-Fixed      | GB-Hei                      | + Courier-Bold |   |
|                   | ( 字库文件 GB-Hei-Fixed.ps      |                | ) |
| GB-Kai-Fixed      | GB-Kai                      | + Courier      |   |
|                   | ( 字库文件 GB-Kai-Fixed.ps      |                | ) |
| GB-FangSong-Fixed | GB-FangSong                 | + Courier      |   |
|                   | ( 字库文件 GB-FangSong-Fixed.ps |                | ) |

定义了 4 种 Type0 中文(GBK)固定宽度斜体字库(为使中文字符的宽度是西文字符的 2 倍,将西文字符的宽度缩减 1/6):

|                     |                               |                   |   |
|---------------------|-------------------------------|-------------------|---|
| GB-Song-FixedItalic | GB-Song                       | + Courier-Oblique |   |
|                     | ( 字库文件 GB-Song-FixedItalic.ps |                   | ) |

|                         |                                     |                       |  |
|-------------------------|-------------------------------------|-----------------------|--|
| GB-Hei-FixedItalic      | GB-Hei                              | + Courier-BoldOblique |  |
|                         | ( 字库文件 GB-Hei-FixedItalic.ps )      |                       |  |
| GB-Kai-FixedItalic      | GB-Kai                              | + Courier-Oblique     |  |
|                         | ( 字库文件 GB-Kai-FixedItalic.ps )      |                       |  |
| GB-FangSong-FixedItalic | GB-FangSong                         | + Courier-Oblique     |  |
|                         | ( 字库文件 GB-FangSong-FixedItalic.ps ) |                       |  |

为与西文 Times, Courier 和 Helvetica 系列字库对应, 通过别名和 Type0 字库文件定义了以下 8 种字库(最后 2 个字库中 GB-Kai 的高度拉长了 1/10):

|                          |                                      |
|--------------------------|--------------------------------------|
| GB-Times-Roman           | ( GB-Song-Regular 的别名 )              |
| GB-Times-Italic          | ( GB-Song-Italic 的别名 )               |
| GB-Times-Bold            | ( GB-Hei-Regular 的别名 )               |
| GB-Times-BoldItalic      | ( GB-Hei-Italic 的别名 )                |
| GB-Courier               | GB-FangSong + Courier                |
|                          | ( 字库文件 GB-Courier.ps )               |
| GB-Courier-Oblique       | GB-FangSong + Courier-Oblique        |
|                          | ( 字库文件 GB-Courier-Oblique.ps )       |
| GB-Courier-Bold          | GB-Kai + Courier-Bold                |
|                          | ( 字库文件 GB-Courier-Bold.ps )          |
| GB-Courier-BoldOblique   | GB-Kai + Courier-BoldOblique         |
|                          | ( 字库文件 GB-Courier-BoldOblique.ps )   |
| GB-Helvetica             | ( GB-FangSong-Regular 的别名 )          |
| GB-Helvetica-Oblique     | ( GB-FangSong-Italic 的别名 )           |
| GB-Helvetica-Bold        | GB-Kai + Helvetica-Bold              |
|                          | ( 字库文件 GB-Helvetica-Bold.ps )        |
| GB-Helvetica-BoldOblique | GB-Kai + Helvetica-BoldOblique       |
|                          | ( 字库文件 GB-Helvetica-BoldOblique.ps ) |

中文(Big5)PostScript 字库的定义与中文(GBK)PostScript 字库几乎完全一致。

中文(Big5)PostScript 字库目前包含 4 种字体, 即明体、黑体、楷体、仿宋体。对每种字体, 有 89 个中文 Type1 PostScript 字库文件, 以 PFB(Printer Font Binary)的格式存储, 它们分别对应于中文 Big5 编码的 89 个区(0xA1—0xF9):

|                 |                    |   |              |                 |
|-----------------|--------------------|---|--------------|-----------------|
| BIG5-Mingai     | —— BIC5-Mingf9     | : | b5mingai.pfb | —— b5mingf9.pfb |
| BIG5-Heiai      | —— BIG5-Heif9      | : | b5heiai.pfb  | —— b5heif9.pfb  |
| BIG5-Kaiai      | —— BIG5-Kaif9      | : | b5kaiai.pfb  | —— b5kaif9.pfb  |
| BIG5-FangSongai | —— BIG5-FangSongf9 | : | b5fsai.pfb   | —— b5fsf9.pfb   |

在上述 Type1 中文 PostScript 字库的基础上, 定义了 4 种 Type0(复合型)中文(Big5)基本字库:

|               |                           |
|---------------|---------------------------|
| BIG5-Ming     | ( 字库文件 BIG5-Ming.ps )     |
| BIG5-Hei      | ( 字库文件 BIG5-Hei.ps )      |
| BIG5-Kai      | ( 字库文件 BIG5-Kai.ps )      |
| BIG5-FangSong | ( 字库文件 BIG5-FangSong.ps ) |

在中文(Big5)基本字库和西文 Type1 型 PostScript 字库的基础上, 定义了 4 种 Type0

中文(Big5)正体字库:

|                       |                                 |               |   |
|-----------------------|---------------------------------|---------------|---|
| BIG5-Ming-Regular     | BIG5-Ming                       | + Times-Roman |   |
|                       | ( 字库文件 BIG5-Ming-Regular.ps     |               | ) |
| BIG5-Hei-Regular      | BIG5-Hei                        | + Times-Bold  |   |
|                       | ( 字库文件 BIG5-Hei-Regular.ps      |               | ) |
| BIG5-Kai-Regular      | BIG5-Kai                        | + Helvetica   |   |
|                       | ( 字库文件 BIG5-Kai-Regular.ps      |               | ) |
| BIG5-FangSong-Regular | BIG5-FangSong                   | + Helvetica   |   |
|                       | ( 字库文件 BIG5-FangSong-Regular.ps |               | ) |

同上,定义了4种Type0中文(Big5)斜体字库:

|                      |                                |                     |   |
|----------------------|--------------------------------|---------------------|---|
| BIG5-Ming-Italic     | BIG5-Ming                      | + Times-Italic      |   |
|                      | ( 字库文件 BIG5-Ming-Italic.ps     |                     | ) |
| BIG5-Hei-Italic      | BIG5-Hei                       | + Times-BoldItalic  |   |
|                      | ( 字库文件 BIG5-Hei-Italic.ps      |                     | ) |
| BIG5-Kai-Italic      | BIG5-Kai                       | + Helvetica-Oblique |   |
|                      | ( 字库文件 BIG5-Kai-Italic.ps      |                     | ) |
| BIG5-FangSong-Italic | BIG5-FangSong                  | + Helvetica-Oblique |   |
|                      | ( 字库文件 BIG5-FangSong-Italic.ps |                     | ) |

定义了4种Type0中文(Big5)固定宽度字库(为使中文字符的宽度是西文字符的2倍,将西文字符的宽度缩减1/6):

|                     |                               |                |   |
|---------------------|-------------------------------|----------------|---|
| BIG5-Ming-Fixed     | BIG5-Ming                     | + Courier      |   |
|                     | ( 字库文件 BIG5-Ming-Fixed.ps     |                | ) |
| BIG5-Hei-Fixed      | BIG5-Hei                      | + Courier-Bold |   |
|                     | ( 字库文件 BIG5-Hei-Fixed.ps      |                | ) |
| BIG5-Kai-Fixed      | BIG5-Kai                      | + Courier      |   |
|                     | ( 字库文件 BIG5-Kai-Fixed.ps      |                | ) |
| BIG5-FangSong-Fixed | BIG5-FangSong                 | + Courier      |   |
|                     | ( 字库文件 BIG5-FangSong-Fixed.ps |                | ) |

定义了4种Type0中文(Big5)固定宽度斜体字库(为使中文字符的宽度是西文字符的2倍,将西文字符的宽度缩减1/6):

|                           |                                     |                       |   |
|---------------------------|-------------------------------------|-----------------------|---|
| BIG5-Ming-FixedItalic     | BIG5-Ming                           | + Courier-Oblique     |   |
|                           | ( 字库文件 BIG5-Ming-FixedItalic.ps     |                       | ) |
| BIG5-Hei-FixedItalic      | BIG5-Hei                            | + Courier-BoldOblique |   |
|                           | ( 字库文件 BIG5-Hei-FixedItalic.ps      |                       | ) |
| BIG5-Kai-FixedItalic      | BIG5-Kai                            | + Courier-Oblique     |   |
|                           | ( 字库文件 BIG5-Kai-FixedItalic.ps      |                       | ) |
| BIG5-FangSong-FixedItalic | BIG5-FangSong                       | + Courier-Oblique     |   |
|                           | ( 字库文件 BIG5-FangSong-FixedItalic.ps |                       | ) |

为与西文Times、Courier和Helvetica系列字库对应,通过别名和Type0字库文件定义了以下8种字库(最后2个字库中BIG5-Kai的高度拉长了1/10):

|                            |                                                                            |
|----------------------------|----------------------------------------------------------------------------|
| BIG5-Times-Roman           | ( BIG5-Ming-Regular 的别名 )                                                  |
| BIG5-Times-Italic          | ( BIG5-Ming-Italic 的别名 )                                                   |
| BIG5-Times-Bold            | ( BIG5-Hei-Regular 的别名 )                                                   |
| BIG5-Times-BoldItalic      | ( BIG5-Hei-Italic 的别名 )                                                    |
| BIG5-Courier               | BIG5-FangSong + Courier<br>( 字库文件 BIG5-Courier.ps )                        |
| BIG5-Courier-Oblique       | BIG5-FangSong + Courier-Oblique<br>( 字库文件 BIG5-Courier-Oblique.ps )        |
| BIG5-Courier-Bold          | BIG5-Kai + Courier-Bold<br>( 字库文件 BIG5-Courier-Bold.ps )                   |
| BIG5-Courier-BoldOblique   | BIG5-Kai + Courier-BoldOblique<br>( 字库文件 BIG5-Courier-BoldOblique.ps )     |
| BIG5-Helvetica             | ( BIG5-FangSong-Regular 的别名 )                                              |
| BIG5-Helvetica-Oblique     | ( BIG5-FangSong-Italic 的别名 )                                               |
| BIG5-Helvetica-Bold        | BIG5-Kai + Helvetica-Bold<br>( 字库文件 BIG5-Helvetica-Bold.ps )               |
| BIG5-Helvetica-BoldOblique | BIG5-Kai + Helvetica-BoldOblique<br>( 字库文件 BIG5-Helvetica-BoldOblique.ps ) |

为支持西文 ASCII 字符和中文字符的混合使用,我们应当使用上述定义的中文正体、斜体、固定宽度、固定宽度斜体 PostScript 字库,而不要直接使用中文基本字库 GB-Song, GB-Hei, GB-Kai, GB-FangSong 和 BIG5-Ming, BIG5-Hei, BIG5-FangSong, BIG5-Kai, 因为中文基本字库没有引用西文 PostScript 字库。

目前中文 PostScript 字库还不够完善,以后肯定会又很多改进,字库定义的方式可能会有改变,但这里确定的字库命名规则应该不会有太大的改变。

### 8.14.3 使用中文 PostScript 字库

#### 1. 编写中文 PostScript 文件的规则

中文 PostScript 字库的设计过程已经尽量考虑与西文 PostScript 文件及软件的编写惯例兼容,但由于西文 PostScript 字库基本上都是 Type1 字库,而中文 PostScript 字库无论如何也不能只使用一个 Type1 字库,同西文字库类比直接引用的应该是 Type0 字库,所以编写中文 PostScript 文件的规则与西文 PostScript 文件略有不同。如在很多软件产生的 PostScript 文件中,经常会有类似下面修改西文字库的 Encoding 项的语句:

```
/F0
/Times-Roman findfont
dup length dict begin
  { 1 index /FID ne {def} {pop pop} ifelse } forall
/Encoding isolatin1encoding def
currentdict end
definefont pop
```

“/Encoding isolatin1encoding def”一句把 Times-Roman 字库的 Encoding 项更改为“isolatin1encoding”并重新定义了字库。“isolatin1encoding”是西文 Type1 字库的一种标准

Encoding。但如果对 GB-Times-Roman 进行这样的操作将会导致 invalidfont 的错误。因为 GB-Times-Roman 是 Type0 字库,其 Encoding 项与 Type1 字库的 Encoding 项有本质上不同的意义。所以在中文 PostScript 文件中不允许出现以上这种类型的语句。

除了上面这种类型的语句外,中文 PostScript 文件的编写规则基本等同于西文 PostScript 文件。下面是一个简单的例子:

```
%! PS-Adobe-3.0
%%
/GB-Song-Regular 24 selectfont
100 600 moveto
(希望 Chinese PostScript Fonts 越来越完善) show
%%
/GB-Courier-Oblique 16 selectfont
100 300 moveto
(\ 322 \ 342 \ 274 \ 373)show (和)show < hda8d2e9 > show
(请发至 seanc@turboLinux.com.cn)show
%%      上面两行是“意见和建议请发至 turbolinux.com.cn”
showpage
```

我们看到,在 show 算符前面的字符串可以是汉字,可以是汉字的八进制代码,也可以是汉字的十六进制编码。

中文 PostScript 字库无条件地允许 ASCII 字符和汉字的混合使用。如果以前使用 Times-Roman 字库,现在简单地用 GB-Times-Roman 或 BIG5-Times-Roman 代替就行了,等等。

当前版本的中文 PostScript 字库定义了与西文 Times 系列、Courier 系列和 Helvetica 系列对应的中文字库。很多软件生成的 PostScript 文件总是使用 Times 系列字库作为非等宽字库,使用 Courier 系列字库作为等宽字库。

固定宽度的中文 PostScript 字库中把西文字符的宽度设计成中文字符宽度的一半,这是为满足中西文文本对齐的需要。因为固定宽度的字库大多用在要求上下对齐的文本输出中。

## 2. 把西文 PostScript 文件转换为中文 PostScript 文件

目前很多软件支持中文屏幕显示,但在输出 PostScript 文件时,却没有区分中文字符和西文字符,把每个中文字符当作两个最高位为“1”的 ASCII 扩展字符。如 Netscape 存储的中文 WWW 页面的 PostScript 文件用 Ghostscript 显示时,所有的中文字符都被显示成两个 ASCII 扩展字符。而且这些软件存储的 PostScript 文件中通常含有上面提到的对字库的 Encoding 项进行修改的语句。所以必须对它们进行适当的修改,以产生可显示和打印的中文 PostScript 文件。如下面这个简短的 shell 脚本文件(名为 ns2gbps)对 netscape 存储的 PostScript 文件进行修改,以产生新的中文 GBK 码 PostScript 文件:

```
#!/bin/sh
if [ $# -ne 1 ]; then
    echo "Usage: ns2gbps ps_file"
```

```

exit
fi
sed -e s/" \ /Encoding isolatin1encoding"/"% \ /Encoding isolatin1encoding"/g \
-e s/" \ /Times-Roman findfont"/"% \ /GB-Times-Roman findfont"/g \
-e s/" \ /Times-Italic findfont"/"% \ /GB-Times-Italic findfont"/g \
-e s/" \ /Times-Bold findfont"/"% \ /GB-Times-Bold findfont"/g \
-e s/" \ /Times-BoldItalic findfont"/"% \ /GB-Times-BoldItalic findfont"/g \
-e s/" \ /Courier findfont"/"% \ /GB-Courier findfont"/g \
-e s/" \ /Courier-Oblique findfont"/"% \ /GB-Courier-Oblique findfont"/g \
-e s/" \ /Courier-Bold findfont"/"% \ /GB-Courier-Bold findfont"/g \
-e s/" \ /Courier-BoldOblique findfont"/"% \ /GB-Courier-BoldOblique findfont"/g \
-e s/"findfont exch scalefont setfont"/"exch selectfont"/g \
-e s/\ ( \ \ \ \ [2-3][0-9][0-9] \ ) \ show/% %/g \
-e s/"pagelevel restore"/"%pagelevel restore"/g \
$!

```

这个 shell 脚本文件用 sed 语句对原来的 PostScript 文件中一些语句进行了修改,即:注释掉会导致 invalidfont 错误的改变 Encoding 项的语句,把所有对 Times 和 Courier 系列西文字库的引用改为引用 GB-Times 和 GB-Courier 系列中文字库。如此简单的改动不会影响西文的显示和打印,却可显示和打印出优美的中西文混排的中文 PostScript 输出。同样,有另外一个类似的 shell 脚本文件 ns2b5ps 对 netscape 存储的 PostScript 文件进行修改,以产生新的中文 Big5 码 PostScript 文件。

如果用户的打印机队列使用 Ghostscript 作为输入过滤器,现在就可以打印经过 ns2gbps(ns2b5ps)转换后的中文 PostScript 文件了。如果用户想直接从 netscape 中打印,可以打开 netscape 的“File → Print...”菜单,设置“Print Command”项为“ns2gbps - l lpr”或“ns2b5ps - l lpr”,就可以从 netscape 中直接打印所有页面,特别是含中文字符的页面。

### 3. 打印中文普通文本文件

很多用户经常有打印普通中文文本文件的需要。软件包提供了 shell 脚本文件 txt2gbps 和 txt2b5ps 来把普通中文文本文件转换成中文 PostScript 文件。它们的用法是

```

txt2gbps [-fn chinese_font] text_file
txt2b5ps [-fn chinese_font] text_file

```

txt2gbps 缺省是使用固定宽度的宋体字库 GB-Song-Fixed,txt2b5ps 缺省是使用固定宽度的明体字库 BIG5-Ming-Fixed,但用户可以用“-fn”选项来指定使用其他字库,如:

```
lark% txt2gbps -fn GB-Kai-Fixed report.txt > report.gps
```

将使用固定宽度的楷体字库 GB-Kai-Fixed 把中文(GB)普通文本文件 report.txt 转换成中文(GB)PostScript 文件 report.gps。然后用户就可以使用:

```
lpr report.gps
```

进行打印了。使用固定宽度的字库是为了使中文 PostScript 文件中的 ASCII 字符和中文字符像普通文本文件里一样保持行列对齐。但如果用户并不要求这样,也可以选用非固定宽度的字库。

日前在 Linux 上使用的将文本文件转换为 PostScript 文件的打印过滤程序主要有 mpage 和 nenscript。对它们的源代码稍作修改,就可以使它们能把中文文本文件转换为中文 PostScript 文件。这样如果用户已经安装了中文 PostScript 字库并用 printtool 或按本书第五章的指导配置了打印队列,可以从:

<ftp://ftp.turbolinux.com.cn/pub/chinese/packages/>

下载陈向阳对 nenscript 和 mpage 程序的修正。编译安装后,用户就可以直接在命令行用:

```
lp gb.txt
```

打印中文文本文件了。

#### 4. 把中文 PDF 文件转换为中文 PostScript 文件

最近有一张很流行的 CD-ROM,包含着很多用中文 PDF 格式存储的名著,如领袖著作、作家全集、今古小说等等。特别是还包括了很多武侠小说迷爱不释手的金庸武侠小说全集。这些 PDF 文件可在中文 Windows 95 等操作系统下用 Adobe 的 Acrobat Reader 阅读,但由于它们使用了 Windows 的系统中文字库,所以在各种 UNIX 上却无法用同样的软件 Acrobat Reader 阅读。这使很多爱读书的用户感到很遗憾。现在用中文 PostScript 字库却可以在 UNIX 上以 PostScript 文件的格式进行阅读!本软件中提供的 shell 脚本文件 jy2gbps 可以利用 Acrobat Reader 将中文 PDF 文件转换成中文 (GB) PostScript 文件, jy2gbps 的内容很简单:

```
#!/bin/csh
if ( $# argv != 1 && $# argv != 2 ) then
    echo "Usage: jy2gbps pdf_file [ps_file]"
    exit
endif
set pdf_file = $1:t
set ps_file = $pdf_file:r.gps
if ( $# argv == 2 ) set ps_file = $2
rm -fr $ps_file
acoread -toPostScript -level2 < $1 | \
    sed -e s/"\ /Encoding . pdfEncodings"/"% \ /Encoding pdfEncodings"/g \
        -e s/"N2 \ /Helvetica"/"N2 \ /GB-Song-Regular"/g \
        -e s/"N3 \ /Helvetica"/"N3 \ /GB-Hei-Regular"/g \
        -e s/"N4 \ /Helvetica"/"N4 \ /GB-Kai-Regular"/g \
        -e s/"N5 \ /Helvetica"/"N5 \ /GB-FangSong-Regular"/g \
        > $ps_file
```

首先,acoread 将中文 PDF 文件转换成普通的 PostScript 文件,然后 sed 命令将文件中修改 PostScript 字库 Encoding 项的语句注释掉;将原文件中都定义为西文 PostScript 字库 Helvetica 的 Windows 系统字库 N2, N3, N4, N5 分别重新定义为正确的中文 (GB) PostScript 字库,从而生成中文 (GB) PostScript 文件。用户就可以用 Ghostscript 阅读并打印这些脍炙人口的著作了。jy2gbps 的用法是:

```
jj2gbps pdf_file [ps_file]
```

缺省的输出文件名是将输入文件的扩展名从 .pdf 改为 .gps。

这里需要提到的是,这张光盘里存储的很多著作包含了一些图片,这些图片是以 JPEG 的格式存储在 PDF 文件里的,从而用 jj2gbps 转换生成的中文 (GB) PostScript 文件中也包含了这些 JPEG 图片。所以必须使用支持 JPEG 格式图形的 Ghostscript 才能阅读这些 PostScript 文件。如果用户使用的是没有 JPEG 支持的 Ghostscript,可从:

```
ftp://ftp.cs.wisc.edu/ghost/aladdin/gs###/
```

(### 是版本号)或其镜像节点取来最新版本的 Aladdin Ghostscript 及 JPEG 库,重新编译 Ghostscript,加进对 JPEG 的支持。

#### 8.14.4 其他中文打印软件

##### 1. gb2ps

gb2ps 可以打印 GB 码和 HZ 码的中文文件。gb2ps 可从:

```
ftp://ftp.ifcss.org/pub/software/UNIX/print/
```

目录取来 gb2ps.2.02.tar.gz,从:

```
ftp://ftp.ifcss.org/pub/software/fonts/gb/misc/
```

取来 gb2ps 用到的中文字库文件 csong24.ccf, cfang24.ccf, ckai24.ccf 和 chei24.ccf。把字库存放在某个目录下,如 /usr/local/lib/gb2ps/cfont。

在编译 gb2ps 之前先更改 Makefile,设置:

```
CFONT = /usr/local/lib/gb2ps/cfont/
```

```
COVERPAGE = /usr/local/lib/gb2ps/cover.ps
```

然后:

```
lark # make
```

```
lark # cp gb2ps /usr/local/bin
```

gb2ps 的特点是它定义了一些标记,用户可以在文本文件中使用这些标记来控制打印输出的效果,如字库选择、字体大小、横排竖排、字体修饰等。这些标记是局部的,即它们可以控制文件中一定范围的字符。

##### 2. cnprint

cnprint 是一个将中日韩文的文本文件转换成 PostScript 文件的软件。它使用 HBF 字库,主要特点是:

- 支持 GB, Hz, zW, BIG5, CNS, JIS, EUC, Shift-JIS, KSC, UTF8 和 UTF7 等多种编码
- 支持竖向排版
- 支持页面分栏
- 可设定字体大小
- 可实现 GB, HZ, Big5 间代码转换
- 支持选页打印

cnprint 可从:

<ftp://ftp.ifess.org/pub/software/UNIX/print>

目录或其镜像节点取得。

## 8.15 X Window 系统的中文化“包装”(WRAP)方案

“包装”(WRAP)方案最近才被用来进行 X Window 系统的中文化,但已显示出其强大的威力。很多原本不支持中文显示和输入的 X Window 系统上的应用程序,即使是没有源代码的商业软件,只要经中文化“包装”后,不经任何修改,不必重新编译,就可显示和输入中文。

“包装”(WRAP)方案是 X Window 系统的中文化传统概念的一次重要提升。

“包装”(WRAP)方案利用了很多 UNIX 系统运行动态连接的程序时的 PRELOAD 机制,对 X Window 系统函数动态库中的某些函数进行了“包装”和替换,使之实现支持中文显示,并可与中文输入服务器连接实现中文输入。

目前 X Window 系统的中文化“包装”程序有 XA(Xcin Anywhere)和 Chinput。

### 8.15.1 PRELOAD 机制

UNIX 系统上广泛使用着共享函数库,应用程序可以与共享函数库动态连接。动态连接的程序没有把调用的函数做进程序中,它们只在运行时从共享函数库中调出所用函数。我们可用 ldd 命令查看程序与哪些共享库连接,如:

```
lark% ldd /usr/X11R6/bin/xterm
libXaw.so.6 => /usr/X11R6/lib/libXaw.so.6 (0x4000a000)
libXmu.so.6 => /usr/X11R6/lib/libXmu.so.6 (0x40042000)
libXt.so.6 => /usr/X11R6/lib/libXt.so.6 (0x40054000)
libSM.so.6 => /usr/X11R6/lib/libSM.so.6 (0x40092000)
libICE.so.6 => /usr/X11R6/lib/libICE.so.6 (0x4009b000)
libXext.so.6 => /usr/X11R6/lib/libXext.so.6 (0x400b0000)
libX11.so.6 => /usr/X11R6/lib/libX11.so.6 (0x400bb000)
libc.so.5 => /lib/libc.so.5 (0x4014c000)
```

在 Linux, Solaris 等操作系统上,如果当前 shell 下定义了一个环境变量 LD\_PRELOAD,它是一些共享目标文件的列表,则程序在执行时将先从 LD\_PRELOAD 指定的共享目标文件中寻找程序用到的所有共享函数,只有在这些共享目标文件中找不到的函数,程序才去与之动态连接的共享库中寻找。这里需要注意的是,如果某个程序是 set-user-ID 或 set-group-ID 的,程序在执行时将只搜索那些在程序连接生成时 LD\_RUN\_PATH 变量或 -R 选项指定的路径上的共享目标,其他共享目标一律被忽略。即,LD\_PRELOAD 变量将不起作用。

### 8.15.2 X 函数的中文“包装”原理

利用动态连接的 PRELOAD 机制,我们可以对 X Window 系统的某些函数进行“包装”,以使原来只是用来处理 ASCII 码的函数可以根据输入情况有区别地处理 ASCII 码和

中文的编码,并把“包装”后的函数做成共享库,利用设置 LD\_PRELOAD 变量使一些 X 应用程序调用“包装”后的 X 函数,而不是直接调用 X 函数库中的函数,从而实现不用更改应用程序的源代码而使它们支持中文处理。

如我们设置 LD\_PRELOAD 变量如下:

```
lark% setenv LD_PRELOAD /usr/local/lib/libwrap.so
```

在 libwrap.so 中有我们重新编写的 X 函数 XNextEvent,其中对与中文处理有关的事件进行了特别处理。xterm 程序运行时,将首先去 libwrap.so 文件中寻找 XNextEvent 函数,而不是在没有 LD\_PRELOAD 变量时直接去 libX11.so.6 中寻找,结果 xterm 在 libwrap.so 中找到了 XNextEvent 函数,它就连接 libwrap.so 中的这个函数而不再去寻找 libX11.so.6 了。从而 xterm 程序运行时就与在没有 LD\_PRELOAD 变量时有很大不同。

由于 X Window 系统的全部源代码是公开的,我们可以很方便地根据原来的 X 函数的代码对它们进行“包装”,使它们支持中文处理。但这种“包装”有个原则,即应不影响对 ASCII 码的处理。

现在举例说明如何对 X 函数 XDrawString 进行中文“包装”。

XDrawString 函数的定义是:

```
XDrawString(display, d, gc, x, y, string, length)
Display * display;
Drawable d;
GC gc;
int x, y;
_Xconst char * string;
int length;
```

其功能是在 Drawable d 上,从位置(x,y)开始,将长度为 length 的字符串 string 用 gc 中指定的字库画出来,简单地说,就是在 X 窗口的某个位置写出 length 个字符。如果 string 中含有中文字符,XDrawString 只会把它们当作 2 个 ASCII 扩展字符画出来。

另有一个函数 XDrawString16,其定义是:

```
XDrawString16(display, d, gc, x, y, string, length)
Display * display;
Drawable d;
GC gc;
int x, y;
_Xconst XChar2b * string;
int length;
```

其功能是在 Drawable d 上,从位置(x,y)开始,将长度为 length 的 16 位编码的字符串 string 用 gc 中指定的字库画出来。如果 string 是一个中文字符串,而 gc 中指定了一种中文 X 字库,我们就可以用这个函数将中文字符串用这种中文字库的图像画在 X 窗口里。

“包装”XDrawString 函数要求,XDrawString 能把一个 ASCII 字符和中文字符混合的字符串分别用合适的 ASCII 字库和中文字库在 X 窗口的正确位置画出来。

我们可以用如下的函数轮廓来建立“包装”后的 XDrawString 函数:

```

XDrawString(display, d, gc, x, y, string, length)
|
handle = dlopen("/usr/X11R6/lib/libX11.so.6", RTLD_LAZY);
realfunc = (int (*)(void))dlsym(handle, "XDrawString");

将 string 分成 ASCII 字符和中文字符分开的若干段(str);
for(每段 str) :
    if(str 是 ASCII 字符串) :
        设置 gc 中字库为 ASCII 字库;
        计算 str 像素长度 len;
        realfunc(display, d, gc, x, y, str, strlen(str));
        # 用 libX11.so.6 中的 XDrawString
    else :
        # str 是中文字符串
        设置 gc 中字库为中文字库;
        计算 str 像素长度 len;
        XDrawString16(display, d, gc, x, y, str, strlen(str)/2);
    :
    x = x + len;
:
:

```

这样,通过对 XDrawString 函数的包装,我们使应用程序中调用 XDrawString 画字符串的语句都能正确地画出中文字符串了。类似地,我们还可以包装其他与字符串和文本输出相关的函数,使它们都能正确处理中文字符,从而使原来不支持中文显示的软件在中文化“包装”后可以显示中文。

使 X Window 系统的软件支持中文输入是通过“包装”XNextEvent, XLookupString 等函数,在这些函数里加进了对与中文输入相关事件的处理,使客户程序能使用中文输入服务器而完成的。如加进了发送键盘事件到中文输入服务器,接收中文输入服务器返回的事件等语句。

### 8.15.3 中文化“包装”软件——XA(Xcin Anywhere)和 Chinput

目前已有的中文化“包装”软件有:台湾人 weijr(weijr@magic.math.ntu.edu.tw)首先编写的 XA(Xcin Anywhere);本书作者根据 XA(Xcin Anywhere)修改的 GB 码版;于明位编写的 Chinput 套件。

XA(Xcin Anywhere)对 XLookupString, XNextEvent, XDrawString, XDrawImageString 等几个 X 函数进行了包装,使用的中文输入服务器是 xcino。XA 基本上可以在 netscape, rxvt, xterm, color\_xterm, cxdterm, xedit, xjed, xpostit, xclipboard, xsgdb, xman 等软件上实现中文显示和输入。XA 可从:

<ftp://magic.math.ntu.edu.tw/pub/XA/>

目录下下载。

本书作者之一(陈向阳)维护的 GB 版 XA 增加了对 XOpenDisplay, XTextExtents 等函数的包装,重新计算中西文字符串的长度,改进了在非等宽 ASCII 字库间中文的显示,

使中西文在netscape等软件中的显示连贯;修改了确定中文字库高度和宽度的规则,使中西文的对比更协调;考虑了X11R6的X服务器和SUN Solaris的OpenWin X服务器对可缩放字库的处理不同,使“包装”在两个操作系统上都可有较好效果;使用了本书作者之一(方汉)改编的支持GB码的xcin作为中文输入服务器。总的来说,GB码的XA在中文显示上进行了较大的改进,特别是灵活的字库缩放规则使中西文显示更加统一。这在使用Netscape时体现的尤其明显。GB码版的XA可从下列地址获取:

<http://ftp.turbolinux.com.cn/pub/chinese/system/>

XA的使用很简单,用户可以在X Window启动文件里设置LD\_PRELOAD变量指向XA的共享目标文件wrap.so,这样所有在X窗口管理器里运行的程序都被中文“包装”了;用户也可以用软件包提供的shell脚本文件xa启动某一个应用程序,如

```
lark$ xa netscape
```

LD\_PRELOAD只对Netscape进程有效,用户就可以有选择地对某些程序进行中文化“包装”(如图8.37所示)。

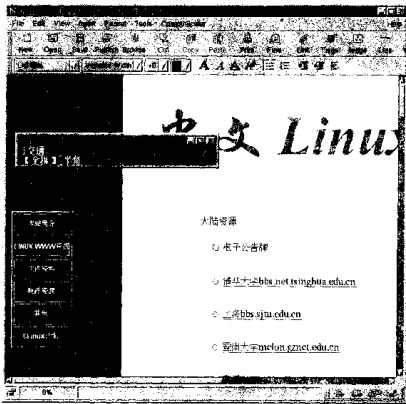


图 8.37 在“包装”后的 Netscape Composer 中输入中文

中文化套件Chinput中对X函数的“包装”进一步深化,增加了对多种事件的处理。它可支持GB、Big5、日文EUC、韩文EUC等多种编码,并可对编码进行实时切换;对半汉字显示和光标位置进行了修正,使移动光标或鼠标刷行时保持汉字输出完整;支持屏幕抓词自动翻译,即与套件中的CDict软件配合可以用鼠标刷词进行英汉对译。

Chinput中文“包装”的使用方法与XA类似。用户可以设置LD\_PRELOAD变量,也可以用shell脚本run运行某个程序(如图8.38所示)

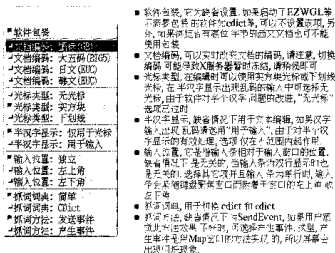


图 8.38 Chinput 软件中的“包装”选项

### 8.16 X Window 系统中文化的基本方法

目前 Linux 上 X Window 系统下的中文软件越来越多,各种软件使用的中文化方法各不相同,各有长短。已经到了有必要总结一下 X Window 系统下软件中文化的经验,归纳一些 X Window 系统中文化的基本方法的时候了。

下面是本书作者和周围的一批正在致力于 Linux 中文化的同仁(以于明俭为代表)对几个中文化基本问题的相近的看法。注意,这里的观点会随时间的推移和经验的进一步积累有所改变。

1. 有必要对 X 服务器进行中文化吗

目前已有的中文化的 X 服务器有 CXWin, 它可以使 X Window 系统的窗口管理器及其他应用程序显示中文, 但不支持中文的输入。

希望通过改动 X 服务器来使应用程序支持中文输入是很困难的。这需要改动 X 服务器的事件(XEvent)分配(Dispatch)。即使如此,有些软件还是不可能支持汉字输入。

## 2. 没有源程序时如何实现中文化

如果软件是采用动态连接的(Dynamic Link),可以改动共享函数库(一般为.so),当然程序员必须熟悉如何替换库中的函数,甚至替换标准库函数。此类方法成功的可能性很小。

“包装”技术是新兴的中文化技术之一。它采用了动态连接程序的 PRELOAD 机制,对一些 X Window 函数进行了“包装”,改变了它们对字符串显示和键盘事件处理,从而可以使应用软件正确地显示和输入中文。如可以使普通的 xterm, netscape, Xform... 等软件显示和输入中文。

如果软件是静态连接的(Static Link),则只能改动可执行文件了。如果该软件支持中文显示,或通过“包装”后可以显示中文,可以试一下编辑该软件(用vi或一些hexedit

软件),改动其中较明显的字符串为中文字符,如“File”改为“文件”。但它仅仅能使软件中的菜单标题等变成中文字符。

### 3. 有源程序时的如何中文化

有源程序时软件的中文化比较容易,程序员要做两件事:(1)中文字符和 ASCII 字符的混合显示;(2)中文输入。

原软件需要改动的地方一般有

- 混合显示,必须使用新的函数取代原来的 XDrawString 和 XDrawImageString
- 窗口尺寸计算,必须使用新的函数取代原来的 XTextWidth 和 XTextExtents
- 编码探测,使用普通手法或 unicode 判断是否中文编码或是中文编码的哪个字节
- 窗口事件的重处理,把功能键和输入键码分开,输入键码作汉字转换
- 汉字接收处理
- 处理字库,GC 等

### 4. 如何解决中文输入问题

X Window 系统最初的设计并没有考虑到中文的输入问题。所谓中文输入实际上是把用户输入的键码(KeyCode)发送到一个用于转换的软件,该软件依据一定的转换方法(即输入法)转成中文编码,然后发送到该键码的窗口中,或把中文编码存储到 Cut 缓冲区内,由用户获取。

所以中文输入问题是十分棘手的问题。首先它需要软件支持汉字显示,如果显示不正确,输入的内容出了问题无从查找。其次它需要软件支持 8 位码的输入,如果软件禁止 8 位码的字符输入,也不可能实现中文输入。鉴于以上两点,一般来说,软件在制作时就应考虑在中文的显示与输入问题。

通过重定义键盘,使键盘按下后可以产生完整的 Latin-1 字符集,则可以有效地避开某些软件对键盘的处理(XLookupString())。关于如何定义键盘可以参考 xkeycaps-2.38.tar.gz,软件 Chinput 则正是利用该方法实现中文输入。

如果上述两个条件都满足,一般地,我们可以用 Cut/Paste 的方法输入汉字。否则必须依据实际情况使用其他方法。

### 5. X Window 中文化目前较好的解决方案是什么

中文化 X 窗口管理器 + X Window 的中文化“包装”软件 + 中文输入服务器

这实际上是一种外挂式中文平台的方案。这种方案的优点是明显的。它不需要对大量的 X 应用程序进行逐个的中文化,它甚至不必知道应用程序的源代码。X Window 的中文化“包装”软件将处理中文显示和输入,在应用程序和中文世界之间搭起一座桥梁。当然,中文化“包装”软件需要对使用它时在中文显示和输入方面碰到的新问题,及时加以解决方法。

前面已经说过,由于 X 窗口管理器的特殊性,我们需要对它进行单独的中文化,使它支持中文显示。这是一个较为简单的工作。

我们需要有一个好的中文输入服务器软件,它必须有方便多样的输入法和简洁的

API。

目前,我们可以使用如下软件组合:

中文化的 fvwm95 + XA(Xcin Anywhere) + Xcin  
中文化的 fvwm95 + xu(XA 的 GB 码版) + xcingb(Xcin 的 GB 码版)  
中文化的 fvwm95 + Chinput 套件

每种方案都可支持 GB 码或 Big5 码或可同时支持两种编码。

## 8.17 ZWinPro——Linux 上的外挂式中文平台

于明俭和本书作者以中文化“包装”方案为基础,开发出在 Linux 上的外挂式中文平台 ZWinPro,并把它作为 TurboLinux 中文版的基础中文平台。

ZWinPro 采用动态载入的方式截取 X Window 的输入输出函数,从而起到动态汉化的作用。其主要组件有:

- 汉字输入服务器及输入条
- 控制条
- 屏幕取词
- 粘贴板
- 用户开发函数库接口

其主要特点有:

- 汉字使用 True Type 字体或缩放的点阵字体显示
- 多语种多编码支持
- 中文输入支持 GB 和 BIG5
- 灵活的输入输出控制
- 完善的用户汉化接口
- 方便地屏幕取词

ZWinPro 可实现单窗口控制。在所有应用程序启动时,ZWinPro 将它设置成缺省的编码和输入法,用户可以用 ZWinPro 控制条对单个窗口的编码和输入法进行设置,而不影响其他窗口的设置。由于使用了中文 TrueType 字体,ZWinPro 可以为应用程序提供多种中文字体的无级缩放的显示,并提供字体的修饰。如对黑体字库,ZWinPro 可以提供各种大小的正体、斜体、粗体、斜粗体显示字体。ZWinPro 缺省对宋体、黑体、仿宋体和楷体字库进行了配置,用户可以按自己的要求增加其他中文字体。

ZWinPro 的汉字输入服务器采用 Chinput,输入条是汉字输入的前端,它使用了 CXterm 的输入方法,支持汉字 GB,汉字 BIG5,日文 EUC 和韩文 EUC 编码,支持 20 多种输入方法(详见 CXterm)。输入条采用有三维效果的无边框窗口,一般情况下不接受聚焦,并且永远位于其他窗口的上面。关于 Chinput 的详细介绍请阅读 8.6 节。

ZWinPro 支持中文 GB/BIG5 编码输入自动切换。ZWinPro 自动识别聚焦窗口正在使用的编码(GB 或 BIG5),根据该窗口的编码把输出的结果转换为该编码。

控制条是 ZWinPro 的控制中心(如图 8.39 所示)。它从左到右每个区域的功能如下:



图 8.39 ZWinPro 控制条

### 1. 控制条的移动

按下控制条最左边的写有“ZWinPro 中文平台”字样的按钮,并拖动鼠标,便可以把控制条移动到屏幕的任何地方。

### 2. 编码及字体设置

控制条上的左起第二个按钮用于控制聚焦窗口的编码和字体。当编码设置为“西文”时,聚焦窗口被设置为非汉化状态。当编码设置为“中文 GB”或其他双字节语种时,会激活子菜单,此时用户可选择相应的字体,聚焦窗口将显示该编码和使用选定字体。同时,按钮上的标签随之改变,表示聚焦窗口的编码和字体(如图 8.40 所示)。



图 8.40 设置聚焦窗口的编码和字体

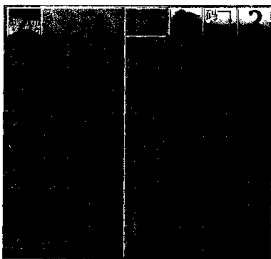


图 8.41 设置聚焦窗口使用的输入法

### 3. 输入方法切换

控制条上的左起第三个按钮用于切换输入条的输入方法,当选择其中汉字输入方法后,按钮的标签被设为“Chinpu”,否则,按钮标签为“西文”(如图 8.41 所示)。

### 4. 窗口刷新

控制条上的“刷新”按钮用于当有些窗口显示乱码时刷新全屏幕。

### 5. 内码转换

控制条上的“码<->码”按钮用于提供汉字内码转换的界面,它使用外部命令实现文件编码转换(如图 8.42 所示)。

ZWinPro 支持“屏幕取词”。所谓“屏幕取词”是指获取鼠标所在位置的英文单词或中文词汇,并进行处理。ZWinPro 支持鼠标所在位置词汇的中英文互译。

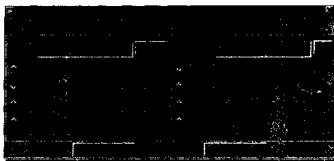


图 8.42 编码转换

屏幕取词服务器 `edict` 在 ZWinPro 启动时已经运行在后台。用户可按如下方法使用屏幕取词功能:按下键盘左边的 `Alt` 键,同时需要在查找的词汇上移动鼠标,则有底色为黄色的浮动条在词汇位置的右下方弹出,并解释词汇的内容。控制取词条的热键为:

- 鼠标左键:使取词条消失
- 鼠标右键:使取词条锁定在屏幕上
- `Control + 鼠标左键`:使取词服务退出

ZWinPro 粘贴板一般用于给用 `Chinput` 不能输入的软件提供汉字输入。粘贴板的使用方法如下:

- (1) 按下 `Control + Shift` 键并点击鼠标中键,则粘贴板在点击位置附近被激活。
- (2) 在粘贴板上输入并编辑单行文本。使用左右箭头按键来移动,使用 `Back Space` 和 `Delete` 删除文本。
- (3) 按回车键(`Return`)选中输入的文本并放在粘贴板的缓冲区中。此后用户可以按应用软件定义的粘贴方法(一般为鼠标中键)粘贴文本(如图 8.43 所示)。



图 8.43 ZWinPro 粘贴板

ZWinPro 提供的用户开发函数库接口有:

- 支持中文输入输出的目标文件
- 仅支持中文显示的目标文件
- 不受环境变量影响的专用目标文件

用户在 X Window 下编写支持汉字的软件时,只需要把上述支持汉字输入和输出的目标文件作为第一个目标文件连接即可。这种方法在不支持 `LD_PRELOAD` 的平台上也可以使用。例如,在 `HPUX` 上稍微改动用户开发函数库并重新连接 `HPUX` 上模拟动态连接的函数库。

ZWinPro 的开发成员正在或将为 ZWinPro 增加以下新的功能:

- 手写输入(使用鼠标或数字化板输入)
- 语音识别与输入(支持语音输入到任何窗口)
- 语音控制(建立完善的语音控制系统)
- 即时语音提示与校正系统

在不久的将来,ZWinPro 将成为 Linux 上的一个多媒体中文平台。

# 9

## Linux 系统安全概述

许多 UNIX 系统都存在着这样或那样的问题,这些问题很容易使系统受到攻击。有时候,系统被攻击了,许多管理员还不知道。即使机器内已经被攻入而且也已发现,只要不影响一般的使用,就不作防范。这种把系统的安全寄托于黑客的“仁慈”的作法是不可取的。现在,攻击者不做破坏可能是因为他的攻击只是出于一种好奇,但这也同时说明了系统的薄弱,如果某个怀有恶意的攻击者将系统毁坏,那时再想作防护就来不及了。所有这些导致不安全因素,可能是系统本身的不完善,也有可能系统设置有问题。系统管理的任务就是维持系统正常运转、升级和正确设置系统。现在,国内的网络管理人员不作防范的另外一个原因就是认为国内的站点都不是很有名,被访问的次数较少,觉得不会有人来攻击,事实上,从最近了解的情况看,国内许多提供服务的 ISP 都受到过攻击,而且有的已被攻入。因为,在互联网上的用户很多,其中有不少黑客。可以这样说,有互联网的地方,就会有“黑客”的存在。

妥善的系统管理可将网络的不安全性降至现有条件下的最低状态。目前,在 UNIX 上发现的大多问题,都是由于编程失误或管理不善。如果每个网络及系统管理员都能注意到以下几点,即可在现有条件下,将网络安全风险降至最低。

本章是 UNIX 系统安全的一些基本常识,不仅适用于 Linux 系统,也适用于其他常用操作系统。

### 9.1 设置安全帐号

设置安全帐号,是配置任务中最重要的一项工作。通常,攻击都是从系统中获得一个帐号开始的。一般,帐号主要包括以下的一些不安全因素:

#### 1. 口令

有多种方法闯入 UNIX 系统,最简单的方法是知道别人的口令后直接以该用户的身份登录。所以,当使用不安全的口令字时 UNIX 是一个不安全的系统。而且,此种攻击也不易得知攻击者的身份。只有通过严禁易猜测使用口令来减少此类攻击的发生。这可以通过使用与入侵者同样的技术,如用 CRACK 之类的口令字破解程序,对系统中的口令进行测试,对口令字选择不当的用户发出警告,让他们更换这些易猜测口令字。从现有的测试结果可知,在不采取强制措施的情况下,加上用户的口令安全意识薄弱,总会有一些用户使用易于猜测的口令。实际上,对于许多很少更新的系统来说,针对该系统的安全漏洞即使已被发现,也很少会有人及时予以防范。这样,只要获得一个帐户,就存在着获得各种特权的可能。一般,加强口令的安全还可以通过设置 shadow 文件来实现。许多

UNIX 系统的加密口令都是保存在 `/etc/passwd` 文件中,而这个文件是对于普通用户是可读的。这样,只要有一个帐号就可以得到该文件,然后使用一些口令测试方法就可以猜测用户口令。目前,系统本身没有加 shadow 的 UNIX 系统可以通过一些程序来设置此功能,将加密口令设置到只有超级用户可以读的 `/etc/shadow` 文件中,可以增加加密口令的难度。

## 2. 长期不更换口令的帐号

再安全的口令经过一段时间后会变得不安全,经常更换口令可以加强系统的安全。但是,用户一般都很少更换口令。因此,可采用设置强制更换口令周期的办法来防止某一口令的长期使用。

## 3. 旧帐号

一些规模庞大的 UNIX 系统可能具有许多旧帐号,可能这些用户已离开该组织或已搬迁到别的地方,帐号长期没有人使用。这类帐号便成为不安全的因素。一般,对于一些临时创建的帐号应该设置截止日期。若发现有超过截止日期的帐号,可以同该用户联系,确定是否删除它。

## 4. 临时不用的帐号

如果用户外出,有一段时间不使用帐号时,可以暂时对帐号加锁。如果不加锁,会给人入侵造成一定的机会,因为此时没有人会发现攻击者用此帐号登录。一般,可在用户保密码口令字段前加一任意标记字符,待他们回来后再删去。

## 5. 客户帐号

客户帐号对安全性是一个严重的威胁,尽量不要使用这种帐号。若必须使用,要注意删除已不用的客户帐号,并确保没有像 `guest` 或 `visitor` 这种简单的口令。客户帐号经常被黑客用来尝试攻击系统。

## 6. 缺省帐号

许多 UNIX 系统中都设置有缺省帐号,也存在缺省口令或者没有口令。在刚安装完系统时,这些缺省帐号的口令没有改变。而国内许多管理员并没有意识到这个问题,加上这种帐号的权限较高,这样会造成很大的漏洞。所以一旦发现这类帐号,需要及时设置或更改口令。

## 7. 组和组帐号

为了共享文件,建立一个工作组帐号可以让几个组员用同一个名字和口令注册。这是最简单的解决方案,但却很危险。若需设置组,可以编辑组文件 `/etc/group` (这个文件的格式是: `groupname:password:groupid:user1,user2,user3,...`), 在 `11` 个字符位置放一个星号并在 `groupid` 段设置 `0~65535` 中一个未用过的数字。对于 `user1`、`user2`、`user3` 等,键入组成员的用户名。然后,使用 `chmod` 和 `chgrp` 命令在文件和目录中设置这一组希望共

享的组允许位即可。

## 8. 网络信息服务(NIS)

NIS 是一个可以很方便地对多系统共享存储在单一节点上的单一口令文件和单一一个组文件进行管理的工具。但同时它又是一个黑客可利用破坏多系统的有力工具。这是因为 NIS 的工作原理为:在客户服务器上,口令文件包括一个“+::0:0::”。当用户从这个系统注册以后,NIS 从 NIS 主机查找用户口令。但不幸的是,这一行完全像一个不需要口令的超级用户的入口。同时,不要不小心删除前面的“+”字符。

## 9.2 确保互联网的安全

UNIX 系统可以互联,且它的设计很容易实现互联。但这也使它们很容易受到黑客们在连接的地方对它们进行的破坏。以下是一些不安全因素:

### 1. 受托节点

在 UNIX 系统中规定有受托节点。与其他主机相比,特殊的地方在于它是可信的,并且在它注册和执行远程命令时不需键入口令。受托节点是 Morris 蠕虫对 UNIX 系统进行攻击的一个特性,由于使用受托节点是一种较方便的方法,所以很多人仍然愿意使用它。若是系统有必要设置受托节点,可采用如下方案:

检查/etc/hosts.equiv 文件

/etc/hosts.equiv 文件中记录所有的受托节点。在这些节点上用 rlogin 注册或用 rsh 执行远程命令的远程用户可以不要口令就能取得本地机的访问权。所以,可通过消除不在可见范围且不能保证为比较安全的节点来减少 hosts.equiv 中的受托节点的数量。有时候,即使主机本身是安全的,但如果不是 24 小时开机,且在该网段内没有设置 IP 地址与硬件地址的严格对应(不幸的是,多数网络均未作此设置),最好将这些机器也从受托节点中移走。(某些 UNIX 版本在 hosts.equiv 中分配一个入口,一个“+”。它允许把所有主机作为受托节点,所以,必须确信在这一文件中没有任何象“+”这样的通配符存在。所有受托节点的定义要明确。)

rhhosts 文件

.rhhosts 文件允许在特殊的节点上的特殊用户有受托访问权(不需口令)。任何用户可以在自己的目录中建立这一文件,并且允许访问帐号不需口令。这便带来用户可能取得系统访问权来访问其他任何用户的脆弱性。确认.rhhosts 文件是否处于非正常使用状态是很困难的,但可以确定的是,.rhhosts 最终将接纳一个不该接纳的客户。所以,对此类问题,解决办法是拒绝赋予用户建立.rhhosts 文件的权利,且定期运行一个程序来访问每一个用户的目录并删除任何一个.rhhosts 文件(这可由 find 命令完成)。

### 2. 安全终端

某些 UNIX 版本可以设置安全终端,它允许超级用户从这些终端注册。根据 UNIX 版本的不同,分别在/etc/ttylab,/etc/tty 或/etc/securetty 文件中设置。在 Linux 上使用

的是/etc/securetty。如果允许超级用户从给定的终端注册,只需简单的把终端设备名(不含/dev/)加在文件/etc/securetty中,如不允许超级用户从某个终端下注册,将其终端设备名从/etc/securetty文件中删除。在设置完成以后,键入命令“kill -HUP 1”强迫系统重读该文件即可。许多 UNIX 系统假设任何事物都是安全的并允许超级用户从任何地方登录。因此,需要检查安全终端文件,并将不安全的终端移走。实际上,高技术的黑客有足够的能力来从一般帐号登录,再通过 su 成为超级用户。但是,如此的预防可以增加攻击的难度。

### 3. 网络文件系统 NFS

NFS 允许两个或更多的节点共享磁盘文件。因此,它允许无盘工作站的用户把文件存储到中心服务器上,通过网络启动,这种功能非常有用。但许多 NFS 产品的缺省设置不包含任何安全特性,使网上任何主机(无论是否值得信任)都可以使用 NFS 来取得服务器上的文件。加强 NFS 的设置,可以减少不必要的麻烦。

一般在/etc/exports文件中,列出了被共享的目录和可以使用这些目录的节点。如 lark 节点的/etc/exports文件中的两行可能是:

```
/home    master(rw) trusty(rw,no_root_squash)
/public  * .local.domain(ro,all_squash)
```

其中第一行中含有的(rw)项是很危险的。事实上,在 Linux 系统上缺省的共享选项就是(rw)。如在上例中, master 节点上除 root 外,所有与 lark 上用户有相同 UID 的用户都可以对 lark 上/home下所有属于这个 UID 的文件进行读写操作。(no\_root\_squash)就更危险了,它取消了把 trusty 上的 root(UID:GID=0:0)用户映射成匿名用户的防护,这样 trusty 节点上的 root 用户就可以对 lark 上的/home下所有文件进行全权操作。上面第二行是在共享文件系统时较安全的设置,它允许本域(local.domain)下的机器对 lark 的/public 目录进行只读操作,并把所有外部用户都映射成匿名用户。所以,对以(rw)方式共享文件系统要格外小心。不要赋予任何不信任的主机以这样的访问权。

### 4. 文件传输协议 FTP

文件传输协议 FTP 由一对程序组成,文件传输服务程序(in.ftpd 或 wu.ftpd 等)和客户程序(ftp)。它能让用户在系统之间传输文件。因为许多旧版本的 FTP 服务器有漏洞,而且一般的 UNIX 系统的缺省设置为提供该服务,所以应查看所使用的 FTP 服务器程序的版本,确定系统正在运行的版本是较新的。往往,一些组织为了增强服务,需要使用匿名登录服务,即在系统中没有帐号的呼叫者仍可以在 FTP 服务器的文件系统中检索特定目录中的文件。这带来一些安全的漏洞,若要提供这种服务,应进行一些安全设置。我们在本书“网络管理与服务”一章“FTP 服务器的设置与维护”一节已有所介绍,此处不再重复。

### 5. 普通文件传输协议 TFTP

普通(trivial)文件传输协议 TFTP 常用于从网络启动 X 终端和路由器等。由于 tftp 在进行文件传输时不需要用户口令,因此会带来许多安全问题。如果这种服务不是必需

的,系统管理员通常将其关闭,即在 `inetd.conf` 中与 `tftp` 相关的一行之前加上注释符“#”即可。如果这种服务是必需的,则必须小心配置 `tftp`。否则,系统将允许传输任何文件!在配置了 `tftp` 后,可用如下方法检查漏洞:

```
% tftp
tftp> connect myhost
tftp> get /etc/motd
```

执行上述命令后,若没有得到“file not found”或“Permission denied”信息,而是得到了一个文件,则证明有漏洞,用户需立即重新配制或安装 `tftp` 程序。

### 6. Sendmail 程序

原码越长,安全漏洞可能就越多,而 `sendmail` 就恰巧应了这句话。`Sendmail` 虽然实现的功能简单,但原码却极其庞大,其本身存在的安全问题,也是如今网络安全专家最头疼的问题。尤其在老版本中的 `sendmail` 中存在着各种各样的问题, `Morris` 的互联网蠕虫便是利用了这些错误中的一个。目前还没有一个十分安全的 `sendmail` 软件,但已经出现了不少针对其安全问题的改进版本,因此建议系统管理员随时注意更换最新版本的 `sendmail` 软件,读有关错误和漏洞的建议。并采取以下措施弥补 `sendmail` 的安全漏洞:

- 从 `/etc/aliases` 或 `/usr/lib/aliases` 中移走译码别名文件。
- 在 `/etc/sendmail.cf` 中撤销引人注目口令。
- 通过运行 `telnet` 并执行 `debug` 命令来检测 `sendmail` 是否支持 `debug`。若没有得到命令不承认的提示,则说明支持 `debug`,需替换 `sendmail` 程序。

### 7. finger 和 fingerd 程序

有了 `finger` 和 `fingerd` 命令,系统外的人可以知道一个用户的全名、个人目录名,最后一次注册时间以及他最后一次检查邮件的时间。但 `fingerd` 中有错误(曾为 `Morris` 的蠕虫病毒利用),后来这些错误得到修补。1988 年 11 月 5 日前的 `fingerd` 都存在这些错误。

### 8. X Window 系统管理

X Window 系统的使用无疑使计算机网络锦上添花。它不但提供友好的图形用户界面,还可以使用户通过网络在本机调用远程节点的 X 程序。也正是由于 X Window 系统的广泛应用,物美价廉的 X 终端(X terminal)才得以倍受欢迎。X Window 系统产生的宗旨原本就是通过客户机/服务器模型,使用户的本地界面可以显示(服务)网络上任何其他系统的 X 客户程序;反过来,本机 X 客户程序也可以在网络上任何其他系统上显示。X Window 系统的设计本身就是针对并构架于计算机网络环境之上的。然而在它给广大用户带来方便的同时,也正是由于它的网络属性带来了不可避免的网络安全问题。

许多用户在醉心于 X Window 系统的网络功能带来便捷的同时,往往意识不到自己为计算机的安全打开一个缺口。在工作间隙,有不少同事开玩笑,在朋友的计算机屏幕上送一个“大钟”(xclock)或某张明星照,然后再跑过去观看他的反应,这些毕竟是善意的。但如果是一些心怀不轨的人,那将是很危险的。如果用户没有将自己的 X 窗口妥善管理好,那么除了屏幕也许会受到不希望的干扰外,更严重的是用户屏幕上的显示及用户的敲

键动作都有可能被远程某个不怀好意的人监视到。对于一个敏感数据或信息,例如口令,在输入时被人偷窥到,那么系统就毫无安全性可言了。作者曾试验过一些用户的计算机安全性,发现由于不少用户由于缺乏这方面的知识或由于系统管理不善,他们的系统安全性能十分脆弱。可以很容易地将其正在使用的屏幕的全部内容捕捉到自己的显示器上,或者在其屏幕上写字,开个善意的玩笑。虽然要保证绝对的安全是不可能的,但下面将推荐几项措施,以保障用户使用 X Window 系统的基本安全:

设置 X 访问控制(Access Control)

正确设置/etc/X0.hosts。这个文件列出了可以访问本 X 服务器(X Server)的一系列节点名。即名单上系统的 X 客户程序可以随时访问本地 X 服务器。这个文件由系统管理员根据实际情况合理设定。从安全角度出发,建议此文件中尽可能地少设入口。

正确使用 xhost 程序(用于交互地设置节点访问控制)。例如允许 angel 上的 X 客户程序访问 lark 上的 X 服务器,则应在 lark 上设置“xhost + angel”,会有如下信息:

```
angel being added to access control list
```

若使用“xhost +”命令则会解除访问控制,允许任何节点的 X 客户程序访问本地 X 服务器。从安全的角度讲,这无疑是大开城门,存在许多潜伏的危险。因此,强烈建议用户不要使用这样的命令。

上面的措施只能限制到节点上,即使所有其他节点均处于访问控制之下,也仍存在着一个很大漏洞,即任何连接到本机节点上的用户都有权访问本机 X 服务器。为了解决这一点,最普遍采用的方法是麻省理工魔术蛋糕(MIT-MAGIC-COOKIE-1)。

#### (1) \$HOME/.Xauthority 文件

如果某系统的 X 服务器提供 MIT-MAGIC-COOKIE-1 服务,用户就会在自己的主目录下发现一个 .Xauthority 文件。这个文件装有机器可识别代码(被称为魔术蛋糕 MAGIC COOKIE),相当于进入 X 服务器的密码。一个 X 客户程序在被允许与 X 服务器联接前,必须首先能够从 .Xauthority 中获得密码,当被 X 服务器核实后,才能获得其访问权。因此,为了确保只有用户自己可以访问其 X 显示,必须将 .Xauthority 文件保管好,只允许用户本人可读或修改此文件,即 .Xauthority 的文件保护应设置为“-rw-----”。

#### (2) xauth 命令

用户通常希望自己在网络上无论从哪个节点帐号的 X 客户程序均可以访问魔术蛋糕,这样,不仅可以很方便地使用 X Window 系统的网络功能,而且又保证了安全性。当然如果用户的 .Xauthority 是由用户的几个节点帐号共享的(如用 NFS 系统),那么用户在这几个节点的 X 客户程序可以方便地访问各 X 服务器。但如果是两个互不关联的系统,这时 xauth 命令可以解决这个问题。xauth 命令是用于将一个节点的魔术蛋糕传送给另一个节点,例如将节点 lark 的魔术蛋糕传送给 angel,则有如下命令:

```
xauth extract -lark:0 -rsh angel xauth merge -
```

这样,只要是用户本人使用这两个节点的 X 资源,就不需再做任何 X 访问控制的设置。保护 Xterm

Xterm 是 X Window 系统下的模拟字符终端,用户可以在命令提示符下键入各种命令。为了防止他人偷窥用户在 Xterm 输入的内容, Xterm 专门提供了 Xterm 保护功能,即在 Xterm 的主菜单下(同时按 Ctrl 和左边的鼠标键)选“Secure Keyboard”一项,这时

Xterm 的底色变黑。如果没有变色,则很有可能某个人正在监视着这个 Xterm。一般建  
议用户在输入敏感信息时,将其 Xterm 设保护。

### 9.3 对安全性进行监控和管理

#### 1. 监控帐号

虽然系统监控是一件十分枯燥的工作,但却很重要。幸运的是,UNIX 提供了允许用  
户全局了解系统上活动的命令。同时,对系统管理员应警觉的事件也作了丰富的日志记  
录,用户可以运行这些命令和查看这些记录。

要想监控用户的工作,可用下面这些方法:

##### (1) 进程状态命令 ps(process status)

ps 命令能给出用户一个在系统中运行的所有进程的完整列表(包括进程 ID、占用  
CPU 的时间、占用内存的数量及哪个用户运行了哪个命令等)。

##### (2) /usr/adm/lastlog 文件

/usr/adm/lastlog 文件能跟踪每一个用户的最后一次注册时间。当某一用户注册后,  
系统自动从这一文件中读取该用户最后一次注册的时间,并在屏幕上显示出来。finger  
也用这一时间。所以,可告诉用户学会用它查看最后一次注册的时间,并确定是否真在这  
一时间注册,若有其他人使用他们的帐号,则显示的时间会比他最后一次注册的时间更晚  
一些(据报道,Dockmaster,一个 NSA 安全网的远程调用者便用这种方法找出不合法访问  
者)。

##### (3) /etc/utmp 文件

/etc/utmp 文件记录了谁正在系统上的信息。用户可用 who 命令来查看这个文件,  
它会列出用户名、终端号、注册时间和远程主机。所以,要经常使用 who 来了解系统上有  
哪些用户。

##### (4) /usr/adm/wtmp 文件

/usr/adm/wtmp 文件记录了每个用户的注册和注销时间。也可用 who 来察看这一  
信息:

```
% who /usr/adm/wtmp
```

我们可以用 last 命令选取我们感兴趣的一部分信息:

```
% last /usr/adm/wtmp username
```

```
% last /usr/adm/wtmp terminal
```

last 把文件中的指定用户根据日期进行排序,为调查提供了方便。

我们认为 who 和 last 都是不完善的审计工具。如果用户有相当有编程的能力,可以  
为自己把信息整理出来,把注意力集中在自己感兴趣的用户、终端或时间范围上。

##### (5) 记帐信息

如果系统上安装并运行了进程记帐软件,用户可以利用分析记帐文件来收集系统上  
的进程信息,以对系统行为进行监控。参加本书 5.14 节“进程记帐”。

#### 2. 监控互联网络安全性

这一工作可能更重要一些,也更困难。以下是一些实现方法。

### (1) 利用系统日志

syslogd 守护进程用于记录各种守护进程和服务产生的错误,它的行为由文件/etc/syslog.conf 来控制,用户可以通过配置这个文件来强制某些命令或进程(如内核、发送邮件、网络服务守护进程、打印进程、用户身份验证程序和其他许多程序)向系统控制台/var/log/syslog 和/usr/adm/messages 这样的日志文件发送信息或出错提示。因此,如果某人注册后,运行 su 取得了超级用户的特权,这一信息会被记录下来,且系统管理员可以扫描信息文件中使用 su 命令的用户。同时,这一信息文件也包含这一尝试是否成功的注释,系统管理员可以密切注视失败的用户,他们很可能在尝试不合法的访问。如果某人用 root 注册,login 将把这一信息写入信息文件。当第三次注册失败后,login 自动记录失败信息。这种失败的原因可能是合法用户忘记了口令(可能这个合法用户在许多系统中都有口令,他忘记在此系统中该用哪一个口令),或者失败的原因是一个入侵者正在试图猜测口令。故要时常查看系统日志文件,查出那些不成功注册。

### (2) showmount 命令

showmount 命令可用来显示所有正在使用 NFS 服务器共享的文件系统的节点(不加选项时,列出节点表;加-a 选项,列出所有节点/目录的集合;加-d 选项,列出当前被使用的共享目录)。故可以用它来确认 NFS 服务器共享目录的使用情况,若发现异常,应及时处理。

## 3. 监控文件安全

下面的命令可帮助用户查出那些可以被不合法用户修改的文件及那些不合法用户访问过的文件和目录:

find 命令

find 命令用来实现浏览文件系统,并根据文件名、类型、模式、所有者或修改时间等条件查找一类文件,用户将得到符合条件的一个文件列表(可将这个文件列表传送到打印机或用其他 UNIX 命令进行处理)。并且可重点查找 setuid、setgid、全可写或没有所有者的文件。

## 9.4 备份系统

当系统丢失或遭到破坏而需彻底恢复时,备份的价值就体现出来了。所以,要经常备份系统,这可由 tar、cpio 来完成。参见 5.10 节“系统的备份与恢复”。

## 9.5 安全建议

知道有人入侵是一种可怕的感觉。虽然大多数入侵者很少或根本不会破坏系统,但用户会感觉到被侵犯或秘密被泄露。按照经验,试图跟踪一个入侵者而导致的恼怒会更强烈。系统和网络管理员应该按以下方法做对此类情况进行强有力的控制:

- 确定是否真正有人入侵
- 关闭入侵者进入的入口

- 使损害最小化
- 找到入侵者

### 9.5.1 系统可能已被入侵的信息

被入侵的系统遭受破坏的可能很显然,或者只是偶然发现,这取决于入侵者的小心程度和用户的谨慎程度。本节列出了一些表明系统可能已被入侵的警告。

#### 1. 新帐户

系统管理员应该翻阅一下系统口令文件,寻找不熟悉的帐号,同时也应该注意那些原来处于不活跃状态,现在突然活跃起来的帐号,以及 UID 为 0 的非 root 帐号。下面的命令自动进行此项检查:

```
% awk -F: '$3 == 0 && $1 != "root" {print}' /etc/passwd
```

如果系统使用 NIS,就应该用以下命令:

```
% (cat /etc/passwd;ypcat passwd)!awk -F: '$3 == 0 && $1 != "root" {print}'
```

#### 2. 不熟悉文件

如果在机器的系统目录中发现不熟悉的、无法解释的文件,则应当提出怀疑。特别要怀疑那些不熟悉的隐含目录,它们通常在名字中含有空格和其他不可见的字符。在这些目录中,可能会发现入侵程序和其他破坏活动的迹象。这可使用下面的命令查看隐含目录:

```
% ls -lab
```

此命令会打印一长串列表,包括以“.”作为文件名开头以及含有不可打印字符(列表中用 8 进制代码 \ddd 表示)的文件和目录。

通常不被注意的双圆点和控制字符的组合导致多数管理者忽视这个假的目录。多数现在的 UNIX 系统允许键入 CTRL+V 键,再键入某控制字符来写这样一个目录的名字。如用户看到目录“.. \b”,在此命名中,用户应该键入:

```
% cd ..\V^H
```

这个输入将使用户进入隐含目录。如果不想这样做,则应当记住现在的大多数的 UNIX 系统的 fsck 程序能把这种隐含目录识别为错误,并询问是否要删除它。

#### 3. 糟糕的执行情况

不正常的慢速系统运行可能是一个警告信号:一个入侵者正在为达到目的而使用系统的 CPU 时间,管理员应该使用 ps 和类似 top 或 vmstat 这样的命令研究系统中可疑的活动。应该特别注意哪些使用了大量 CPU 时间或内存的程序。入侵者经常使用成功入侵的机器的 CPU 来窃取另一站点的口令文件。

#### 4. 系统崩溃

像其他所有操作系统一样,UNIX 有时也会崩溃。当这种事情发生时,它通常在控制台显示混乱的消息,用一些不清楚的、非正式的术语作出解释。如果用户的系统崩溃,就

应该进行检查,以确定是否有硬件错误指示出来。

如果没找到系统崩溃的合理原因,崩溃就可能是入侵者为了掩盖其线索而引发的。

#### 5. 拒绝服务

一些攻击只是想关闭一个节点,手段是使网络接口拥塞、充满硬盘或运行自身复制程序。如果用户认为有此迹象,可相应地用 `netstat`、`df`、`du` 或 `ps` 命令来检查。

#### 6. 特洛伊木马

如果系统上正在运行着 `tripwire`(一种系统监控软件),当任一系统程序被替换时,`tripwire` 就会发出警告。如果这种情况发生,很可能入侵者已把某个或多个系统程序替换为其他程序。即使系统上没有运行 `tripwire`,如果入侵者不是很狡猾,系统管理员仍可通过检查文件的 CTIME 并与其他系统程序作比较来寻找特洛伊木马。

#### 7. 日志文件

虽然入侵者只是从日志文件中消匿他(她)的行踪,但有些入侵者则干脆删除整个日志文件。如果发现日志文件丢失,则要确认它是不是偶然事故或自动日志更换程序造成的。如果已排除了各种可能的原因,则这可能是入侵的迹象。

#### 8. 无法解释的登录

系统管理员应该留意用户的使用习惯,如果看到一个用户在一个不寻常的时间或从不寻常的节点登录,那么他(她)的帐号可能已被入侵。

类似地,如果在系统日志文件中不断出现失败的登录活动,例如登录 `sys`、`uucp`、`bin` 和 `sync`,或者从某个帐号不正确地使用 `su` 命令,这些可能是系统被入侵的迹象。

#### 9. 从其他站点来的抱怨

系统入侵者通常从一个被破坏的站点转到另一个站点,他们这样做既是为了避免被发现,也是为了建立名声。如果另一站点的系统管理员通知我们,他(她)的站点正在被来自我们系统中的一台机器袭击,这一迹象(虽然不完全可靠,每个人都会犯错误)表明我们的站点已被入侵了。

### 9.5.2 反入侵的对策

如果有迹象表明某站点被入侵,系统管理员要保持冷静,必须仔细考虑应采取的适当行动,这一节讨论可能的行动和它们的正反两方面的作用。

#### 1. 进行定期系统检查

系统管理员可利用 `cron` 程序和以上各节中列出的方法,定期收集系统运行的各种信息,及时发现入侵者。

## 2. 制作硬备份

当系统被人入侵后,系统上所有联机的记录信息都可能遭受损害。“打印出来”应该是系统管理员牢记的话。管理员应当把入侵者采取的行动和发现的结果,包括时间和日期都记在笔记本上,这些信息可能随后的彻底调查极为重要,而且将来可能对诉讼法律有用。

## 3. 将问题孤立

应该防止入侵者在网络中从一个机器跳到另一个机器。系统管理员可以用防火墙将被人入侵的系统与网络的剩余部分隔离开来。至少,应该把被人入侵的站点从其他系统中可信主机的列表中删除。

## 4. 对话

如果系统管理员在某一联机时发现了入侵者,可试图用 talk 或 write 命令与他(她)交谈。交谈可采取以下两种方法:可以简单地告诉对方我们不赞赏这种浪费时间的做法并予以阐明,因为许多入侵者只想闯入系统而不是想真正搞破坏,也可以用起诉对方违法的威胁暗示(或明确地告诉)对方。

这两种方法都可能产生适得其反的结果。但另一方面,有时只要让他(她)知道有人发现了就足以吓跑入侵者。任何情况下,如果我们选择“对话”,一定要记录下入侵者说出的每件事,这也许以后会成为有价值的证据。

## 5. 杀死入侵者

如果系统管理员在系统中捕获一个入侵者,则应该改变口令并杀死相应进程。这些手段有双重作用,既修补了漏洞,又将入侵者赶出了系统。

## 6. 杀死计算机

如果系统管理员很担心入侵者正在自己的系统上搞破坏,可以按以下方法加以阻止:关掉网络。如果我们撤除了入侵者正在使用的通信线路,那就消除了对系统安全最直接的影响。

关闭计算机。通常,突然断电对 UNIX 文件系统非常有害,但这是对付极其危险的入侵者的最后一种办法。

## 7. 修补漏洞

如果我们能发现入侵者进入系统的通路,就应该立即消灭它。如果入侵者是通过一个被盗用的帐号而获得访问权,就应该禁止使用该帐号或者改变口令。

## 8. 检查损害情况

当我们确认系统暂时恢复安全后,就需要寻找入侵者对数据的破坏以及埋下的安全隐患:

检查新帐号和 UID 为 0 的帐号。

如果系统正使用 tripwire, 请查找特洛伊木马, 并用干净的软件替换它。

检查系统上的 setuid 或 setgid 程序。

检查有近来的 CTIME、ATIME 和 MTIME 时间标记的文件, 看看是否有被人入侵者替换、检查或修改了的文件, 要记住, 后两种时间标记是容易伪造的。

检查 .rhosts 和 /etc/hosts.equiv 中的变化。

检查隐含文件和目录。

检查 /etc/exports 表, 确认没有将 NFS 文件系统输出给不可信的主机。

使用 showmount 命令, 确认没有未经许可的主机挂上了本机共享的文件系统。

## 9. 从备份中恢复

系统管理员应当经常制作完整的或增量的备份。如果我们每天都更新了备份, 那么总能求助它们修复我们的系统所受的损害。

### 9.5.3 用其他软件加强安全性

我们可以从互联网上下载一些有关系统和网络安全的免费软件来对系统安全进行检测和加强。这里介绍一些目前较常用的此类软件。这些软件基本都可以从下述地址得到:

<ftp://coast.cs.purdue.edu/pub/tools/UNIX/>

#### 1. COPS

COPS 安全检查系统(计算机启示和口令系统)由 shell 脚本文件和程序两部分文件组成, 它具有以下功能:

- (1) 检查易攻破的口令选择。
- (2) 检查正文文件的控制访问(全局可写和分组可写)。
- (3) 检查 setuid 文件的允许权限设置。
- (4) 检查启动文件, profile 和 .eshrc。
- (5) 检查是否有人增加或删除系统文件。

COPS 现在包含 Kuang Expert System, 这个有规则的智能程序可以用来模仿攻击者, 但它并不真把系统搞乱, 而只是报告系统的安全性弱点。当 COPS 运行结束后, 报告被存到盘上或邮寄给系统管理员。

#### 2. tcpdump

网络监控和数据识别工具软件。

#### 3. TCP Wrappers

TCP Wrappers 是一个安全性软件包, 它是由荷兰 Eindhoven 技术大学的 Wietse Venema 编制。该软件的主要部分是 tcpd, 它作为由 inetd 运行的每个网络进程的一个简单包装。用户可以用简单的配置语言设定规则, 依据各种准则接受或否决与任何服务的

连接。而且,用户还可以在否决连接后有一些举措。例如,可以查找一个企图请求非法 rlogin 的主机。

另外,tcpcd 将所有网络连接请求记录到包装服务设备中,这为用户监控破坏系统安全的活动提供了另一层新的监听方式。TCP Wrappers 是一个极好的软件包,我们建议所有对网络安全严肃对待的系统管理员使用此软件包。Slackware Linux 系统就预先配置了 TCP Wrappers。用户需要做的是正确地设置 HOSTS\_ACCESS 文件。

TCP Wrappers 软件包带有一个名为 libwrap.a 的支持库。此库由许多程序使用,为其他服务提供类似包装的防御。尤其特殊的是安全 portmapper 和 rpcbind,它们提供安全的 NIS 和 NFS 服务。同样重要的还有 loginlog,它用于记录所有远程登录。

#### 4. CPM

对网络接口的状态进行检测,发现问题并提出警告。

#### 5. SPI

检查配置文件、访问控制、口令及一些系统事件。

#### 6. Tamu

对帐号计录等系统敏感部件进行检查。

#### 7. Internet Scanner

对 Intranet、Firewall、Web 等进行检测,并对发现的问题提出警告。

#### 8. ifstatus

对网络接口进行检查,判断是否有人对端口进行攻击。

### 9.5.4 一些建议

通向安全和稳固之路应遵循以下告诫:

- (1) 废除系统所有默认的帐号和密码。
- (2) 在用户合法性得到验证前不要显示公司题头、在线帮助以及其他信息。
- (3) 废除“黑客”可以攻击系统的网络服务。
- (4) 不要允许用户以“root”直接登录到主机上。
- (5) 使用 6 到 8 位的字母数字式密码。
- (6) 限制用户尝试登录到系统的次数。
- (7) 记录违反安全性的情况并对安全记录进行复查。
- (8) 对于重要信息,上网传输前要先进行加密。
- (9) 重视专家提出的建议,安装他们推荐的系统“补丁”。
- (10) 限制不需密码即可访问的主机文件。
- (11) 修改网络配置文件,以便将来自外部的 TCP 连接限制到最少数量的端口。不允许诸如 tftp、sunrpc、printer、rlogin 或 rexec 之类的协议。

- 
- (12) 用 `upass` 代替 `sendmail`。`sendmail` 有太多已知漏洞, 很难修补完全。
  - (13) 去掉对操作并非至关重要又极少使用的程序。
  - (14) 使用 `chmod` 将所有系统目录变更为 711 模式。这样, 攻击者们将无法看到它们当中有什么东西, 而用户仍可执行。
  - (15) 只要可能, 就将磁盘安装为只读模式。其实, 仅有少数目录需读写状态。
  - (16) 将系统软件升级为最新版本。老版本可能已被研究并被成功攻击, 最新版本一般包括了这些问题的补救。



# 附录

---

- A GNU 公用许可证 (General Public License)
- B Linux 网络资源
- C Linux 软件资源
- D 本书所附光盘使用说明





## GNU 公用许可证 (General Public License)

### GNU 公用许可证

第 2 版, 1991 年 6 月

Copyright (c) 1989, 1991 Free Software Foundation, Inc.

675 Mass Ave, Cambridge, MA 02139, USA

每个人都可以拷贝和分发这个许可证文档的原封不动的拷贝, 但绝不允许更改它。

### 绪言

绝大多数软件的许可证会剥夺你共享和改变这些软件的自由。相反, GNU 公用许可证致力下保证你共享和改变免费软件的自由——保证该软件对所有用户都是免费的。这种公用许可证适用于绝大多数免费软件基金会的软件和其他作者允许使用的程序 (其他的免费软件基金会的软件受 GNU 库公用许可证的保护), 同样你也可以将它用到你的程序中。

当我们谈及免费软件时, 我们指的是自由而非价格。我们所设计的公用许可证确保你有分发这些免费软件拷贝的自由 (如果你愿意, 也可以为这种服务收费), 确保你收到源代码或能得到它 (如果你想要的话), 确保你能改变它或在新免费软件中使用其中部分代码; 并且还可能确保你知道可做这些事。

为了保护你的权利, 我们需要做出一些限制以避免任何人否认你的这些权利或要求你放弃这些权利。

如果你分发软件拷贝或修改它的话, 这些限制对你来说意味着特定的责任。

例如, 你在分发一个程序的拷贝, 不管是免费的还是收费的, 你必须把你所拥有的所有权利授予接收者。你必须确保他们同样接收或者能够得到源代码, 并且你必须向其展示这些问题以便其了解其权利。

我们用两个步骤保护你的权利:

(1) 给软件赋予版权;

(2) 给你提供这种许可证, 允许你合法地拷贝、分发和/(或)修改软件。

同样, 为了保护每个作者及用户, 并能确认每个人都能理解这种免费软件没有保证的事实。如果软件被其他人修改了并继续传播, 我们需要其接收者知道其拥有的并非源代码。这样其他人造成的任何问题都不会影响源代码作者的声望。

最后, 任何免费软件都受到专利的威胁。我们希望避免一个免费软件的再分发者可以独自获得专利许可证, 避免事实上将程序变为财富的危险。为了防止这种危险, 我们应澄清的是任何专利必须为每个人的自由使用而注册或者根本就不注册。

下面列出有关拷贝、分发和修改的精确条款和条件。

## GNU 公用许可证

## 关于拷贝、分发和修改的条款和条件

这种许可证可应用在任何程序或其他包含有通告的作品上,这种通告由版权拥有者放置,并通知它可在这种公用许可证的条款下分发。下边的“程序”指的是任何一种程序或作品,而“基于程序的作品”指的是受版权法保护的程序或者任何派生的作品,也就是说,一个包含程序或部分程序的作品,既包含有原封不动的拷贝,又包括修改和(/或)翻译到另一种语言的副本(这以后,“翻译”毫无限制地包括在“修改”这个词中),每个许可证都以“你”作为称呼。

除了拷贝、分发和修改之外的活动都不包括在这种许可证之中,它们都超出了范围,运行程序的行为并不受限制,只有在程序输出的内容组成一个基于程序的作品时它才受许可证的限制(与运行程序无关),它是否正确取决于程序完成的功能。

1. 你可以像你接收程序源代码一样在任何媒体上分发它的原件备份,只要你在每个拷贝的显著位置适当地发布版权通告和担保声明,你可以与参考这个许可证并缺乏保证的通知保持联系,并可以随同程序给程序的其他接收者一份许可证的拷贝。

你可以因为传递拷贝的行为收费,并且可以任意提供授权保护做为收费的代价。

2. 合成基于程序的作品时,你可以修改程序拷贝或其中任一部分。并且只要你同样满足以下这些条件,你就可在上边第1条款下拷贝和分发这种修改品或作品:

(a)你必须使已修改的文件带有显著的通知,声明你已改变文件及其日期;

(b)你必须使你分发或出版的作品,不管其整体或部分来源于其他什么程序,都应该整体标注为不收费以供第三方基于本许可证条款来使用;

(c)如果已修改的程序在运行时要正常地交互式阅读命令,在以绝大多数普通方法来开始运行这种交互式使用时,你必须使它打印或显示包括有适当版权通告和无授权担保声明的文档信息(否则,应指出提供了授权担保),同时也应使它声明用户可在这些条件下再分发程序,并告诉用户如何查看本许可证的拷贝(例如:如果程序本身是交互式的但不能正常地打印这种声明,你的基于程序的作品并不要求打印这种声明)。

这些要求做为一个整体应用到已修改的作品上,如果该作品的可定义部分并非来自于程序,并且有理由认为是自身独立的和单独的作品,当你以独立作品分发它们时,那么本许可证及其条款不再适用。但当你以一个基于程序作品的整体或部分分发这些相同部分时,整个作品的分发必须遵守许可证的条款,对于其他许可证的许可将延伸到整体上,并因此对每个部分都不必考虑是谁编写的。

因此,本许可证的目的不在于声明你编写作品的权利,相反,目的只是使用这种权利来控制基于程序的派生或收集作品的分发。

此外,仅仅非基于程序的另一作品,在一定存储或分发媒介上的程序(或基于程序的一件作品)的汇集不构成本许可证范围的另外作品。

3. 你可以(或基于它的作品,在第2节中)基于第1和第2节中的条款,以目标代码或可执行形式拷贝和分发程序,只要你能完成以下任一工作:

(a)伴随程序是完全对应的机器可读的源代码,它必须在一个可为软件交换任意使用的媒介上基于第1和第2的条款下被分发;

(b) 伴随程序的是书面方式提供的完全机器可读的相应源代码的拷贝, 它至少应在 3 年内有效, 应以不超过支付源码分发代价的价格提供给第三方, 它应基于第 1 和第 2 节的条款并在任意使用的媒介上分发;

(c) 伴随程序的是你收到用以分发相应源代码的信息 (这种替代方案只在非商业性销售和只在以接收到目标代码或可执行状态程序时提供, 它应与上述部分一致)。

了解作品的源代码意味着对作品可任意修改, 对于可执行的程序, 完整的源代码意味着其所有模块的全部源代码, 任何有关的接口定义文件, 加上用于控制编译和可执行程序安装的程序。但是, 分发的源代码不需要包括任何随可执行程序运行的基于操作系统的主要部件 (编译器, 内核等等) 一起正式分发的东西 (可以是源代码或二进制形式), 除非这些部件本身有可执行程序。

如果执行程序或目标代码的分发由提供从指定位置拷贝的访问手段来完成, 那么提供从相同地点对源代码拷贝的相同访问手段即与源代码等效分发, 即使第三方并不一定要为目标代码来拷贝源代码。

4. 除非此许可证提供了许可, 否则你不能拷贝、修改、再注册或分发程序。除了拷贝、修改、再注册或分发程序之外的尝试和努力都是不允许的, 否则本许可证将自动终止你的权利。

另外, 从你这里基于本许可证已接收到拷贝或权利的团体将不会终止其许可证, 只要这些团体完全遵守规定。

5. 只要你还没有签署本许可证, 你就不必接受本许可证的规定。但是, 没有其他的東西能赋予你修改或分发程序或作品的权限。如果你不接受该许可证, 你的所有对程序的操作在法律上是被禁止的。因此, 通过对程序 (或任何基于程序的作品) 的修改或分发, 就表明你接受本许可证以及对程序及基于程序的作品关于拷贝、分发或修改的条款和条件。

6. 每次你再分发程序 (或任何基于程序的作品), 接收者将自动从初始注册者那里获得许可证并按照这些条款和条件拷贝、分发或修改程序。此时你不必给接收者对这些权利的使用强加任何更多的限制。你不必对第三方强加对本许可证的服从负责任。

7. 由于法庭判决的结果, 违反专利的控诉或其他任何原因 (不限于专利问题), 如果强加的条件 (不管是法庭判决、协议或其他) 与本许可证的条件矛盾, 他们并不会因该许可证的条件而原谅你。如果你分发程序不能同时满足许可证的条件或其他适当的条件, 那么其后果是你就根本不能分发该程序。例如, 专利许可证就不会允许那些从你这里直接或间接地获得拷贝的人毫不负责地再分发该程序, 而唯一能让你满足它就是完全制止对程序的分发。

本节的目的并非诱使你违反任何专利或其他财产权利, 这一节唯一的目的在于保护免费软件分发系统的完整性, 并通过公共许可证实践来实现。许多人已经通过此系统的持续应用做出了巨大的贡献, 并有利于实现大范围的软件分发, 它取决于著者/制作者决定他或她是否愿意通过其他任何系统分发软件, 而不能强迫他/她作出选择。

本节的目的在于彻底弄清该许可证剩余部分的后果。

8. 如果程序的分发和 (或) 使用通过专利或有版权的接口限制在特定国家内, 将程序置于该许可证之下的初始版权持有者将增加除这些国家之外的明显地理限制, 因此只

允许在那些没被排除在外的国家之中分发。在这种情况下,这个许可证纳入这种限制就如同其编写在许可证的主体之中。

9. 免费软件基金会将随时出版修订后的或新版本的公用许可证,这些新版软件与当前版本的软件在思想上很相似,但可能在处理新问题或所关心的细节上不同。

每个版本都被赋予了一个唯一的版本号。如果程序指定了将应用在自身及其以后版本上该许可证的版本号,你就会有免费软件基金会出版的关于那一版及其后继版的下列条款和条件的选项。如果程序没有指定该许可证的版本号,你可以选择免费软件基金会曾经出版的任何版本。

10. 如果你想把部分程序结合到另外分发条件不同的免费软件里,可以写信给作者征得同意。对于由免费软件基金会拥有版权的软件,可写信给免费软件基金会。有时我们会为此做一些例外。我们的决定取决于由保留所有我们免费软件的免费分发状态和促进软件共享和重用这两大目标。

11. 由于程序注册为不收费,因此对该程序就没有担保达到实际法律允许的程度。否则除非在文字中声明,版本所有者或其他团体提供程序时没有任何种类的担保,既包括表明的或隐含的,也包括对特殊目的的可销售性和适用性的隐含担保,至于程序的质量和性能的整体风险均在于你自己。如果想验证程序的错误,你就得承担所有必要的服务、修改或纠错的费用。

12. 除非法律需要或在文字中声明,否则任何版权所有者或其他在以前允许的修改或再分发程序的团体,将对任何损失负责,包括任何一般的、特殊的、偶然的或由使用该程序引起的损失后果(包括不受限制的数据的丢失或不精确的数据或由你或第三方团体承担的损失或程序与其他程序的失败操作),即使这些版权所有者或其他团体已被建议有这种损失的可能性。

## 条款和条件结束

### 如何将这些条款应用到你的新程序中

如果你开发了一个新程序,并且想让公众最大可能地使用它,达到这一点的最佳途径就是使它成为免费软件,这样每个人都能在这些条款下再分发和修改它。

要这样做,就应把下列通知联到程序上,把它们联到每个源文件的开头以充分表明无担保,这种做法是最安全的。并且每个文件都应至少有“Copyright”行和在何处可发现全部的通告的指示。

< 给出编程者的姓名的行和其完成工作的简要介绍 >

Copyright(C)19yy <作者姓名>

```
This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation; either version 2 of the License, or
(at your option) any later version.
```

This program is distributed in the hope that it will be useful,

but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

同样也应加上如何用电子邮件或信件与你联系的信息。

如果程序是交互式的,在它开始进入交互模式时使其输出一个类似这样的简短通告:

```
Gnomovision version 69, Copyright (C) 19yy name of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type 'show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type 'show c' for details.
```

假设的命令 'show w' 和 'show c' 显示公用许可证的适当部分。当然,你使用的命令可以叫做不同于 'show w' 和 'show c' 的其他名字,甚至可以是鼠标单击或菜单选项,也可以是适合你程序的任何东西。

同样,如果必要,你应请你的雇主(如果你作为一名编程人员开展工作)或你的学校来为程序签署“copyright disclaimer”,这里有一个例子。请改换其名字:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the program
'Gnomovision' (which makes passes at compilers) written by James Hacker.
```

< signature of Ty Coon >, 1 April 1989

Ty Coon, President of Vice

这个公用许可证不允许将你的程序添加到牟利的程序中。如果你的程序是常规库程序,你可以认为允许牟利的应用程序链接到该库程序是很有用的。如果你想这样做,请使用 GNU 库公用许可证而不是这里的公用许可证。

# B

## Linux 网络资源

### 一、Linux 中文资源

中文 Linux 计划的一个主页

<http://eldp.ml.org>

<http://eldp.turbolinux.com.cn>

中文 Linux 计划的主页

<http://clinux.ml.org>

台湾中文 Linux 计划主页

<http://www.linux.org.tw>

交大资料 Linux 主页

<http://linux.cis.nctu.edu.tw>

交大果茶小站站长 Hanky 的 Linux 主页

<http://henry.dorm10.nctu.edu.tw/~linuxwww>

X Window 配置

<http://cnpa.yzit.edu.tw/~garylee>

一些 Linux & X 软件说明

<http://www.oz.nthu.edu.tw/~wohong>

### 二、Linux 正式站点

Linux 官方站点

<http://www.linux.org>

Linux 在英国的官方站点,是 Linux/SMP 和 Linux 8086 的老家

<http://www.uk.linux.org>

Linux 性能测试

<http://www.silkroad.com/bass/linux/bm.html>

Linux 在商业和工业上的用途实例

<http://www.m-tech.ab.ca/linux-biz>

免费散发过期 Linux CD 的地方

<http://emile.math.ucsb.edu:8000/giveaway.html>

Linux 用户登记的地方

<http://domen.uninett.No:29659>

Linux 用户和开发者联系的地方

<http://www.silkroad.com/lds/present.html>

在商业上使用 Linux

<http://linas.org/linux/index.html>

针对初级和中级 Linux 用户的地方

<http://www.lsl.com/forum>

Linux 用户的游戏天堂

<http://www.cs.washington.edu/homes/tlau/linux-game.html>

Linux 的电子杂志

<http://www.redhat.com/linux-info/lg>

系统核心和新的计划以及网络等

<http://www.uwsg.indiana.edu/hypermail>

Linux 新闻和网络资源

<http://www.redhat.com/linux-info/lg/linux.html>

Linux 网络资源大全

<http://www.li.org/linux-int>

Linux 上使用 JAVA

<http://www.parnasse.com/java.shtml>

使你的 Linux 更漂亮

<http://www.eit.com/~kevinh/art/linux>

登记 maillist 的地方

<http://summer.snu.ac.kr/~djshin/linux/mail-list/index.shtml>

每月一次的 email, 内容是所有 Linux 的发布信息

<http://www.all-linux.com/announce.html>

可搜寻 SUNSITE 的软件目录和其他 Linux 信息

<http://www.linuxnow.com>

Linux2.0 的图像

<http://www.isc.tamu.edu/~lewling/linux>

Greg Roelofs 的艺术作品

[http://pobox.com/~newt/greg\\_linuxpics.html](http://pobox.com/~newt/greg_linuxpics.html)

Linux 趣话

<http://www.cs.helsinki.fi/~wirzeniu/linux/linux-quotes.html>

宠物 Linux

<http://penguin.uklinux.org>

大量关于 X Window 的文档

<http://www2.nwpower.net/~badams>

Linux 软件评论

<http://karlsberg.usask.ca/~slg/lsr/lsr.html>

Linux 的小工具

[http://www.redhat.com/linux-info/lg/linux\\_toys.html](http://www.redhat.com/linux-info/lg/linux_toys.html)

在线 Linux 用户俱乐部

<http://www.linuxware.com>

全球 Linux 资源

[http://www-i2.informatik.RWTH-Aachen.de/arnd/lx\\_wwwsites.html](http://www-i2.informatik.RWTH-Aachen.de/arnd/lx_wwwsites.html)

Linux 开发者的 Linux 使用技巧

<http://www.acm.org/~daveotto/linux.html>

### 三、Linux 安装配置信息

如何运行最新和最好的系统核心

<http://www.cviog.uga.edu/Misc/info/LinuxBleed.html>

如何在 Linux 下配置各种各样的硬件

<http://www.hal-pc.org/~davidl/linux/linux.config.html>

如何编译系统核心

[http://math-www.uni-paderborn.de/~axel/config\\_help.html](http://math-www.uni-paderborn.de/~axel/config_help.html)

在笔记本上运行 Linux

<http://www.cs.utexas.edu/users/kharker/linux-laptop/volunteer.html>

如何更好地使用你的系统核心

<http://rsphyl.anu.edu.au/~gpg109/mem.html>

如何在 Linux 下播放 midi 和声音文件

<http://www.digiserve.com/ar/linux-snd/>

如何解决常见 Linux 问题

<http://vortex.cc.missouri.edu/~rhys/linux.html>

支持 Linux 的硬件制造商

<http://linux.crynwr.com/>

未公开的 Linux 秘密

<http://www.hti.net/~moz/linux.htm>

提升用户的系统速度

<http://www.concentric.net/~joat1/lusthome.html>

Linux 核心变化

<http://www.linuxhq.com/>

如何在 Linux 下使用 JDK

<http://www.blackdown.org/java-linux/JDK-1.1.1/Howto-1.1.1/Java-Linux-1.1.1-Howto-4.html>

使用 pgp 加密

<http://www.rust.net/~strix/pgphan11.htm>

如何使用 shadow

<http://www.gaijin.com/linux/shadow.html>

Slackware 的错误修正

<http://cesdis.gsfc.nasa.gov/linux-web/simplefixes/simplefixes.html>

#### 四、Linux 手册

##### ELF HOWTO

<http://www.sjc.ox.ac.uk/users/barlow/elf-howto.html>

##### 如何在 Linux 运行 FORTRAN

<http://studbolt.physast.uga.edu/templon/fortran.html>

##### Linux 常见问题

<http://www.cl.cam.ac.uk/users/iwj10/linux-faq/index.html>

##### Linux 的出版杂志

<http://www.ssc.com/lj/index.html>

##### 系统核心开发者的 maillist

<http://linux.ucs.indiana.edu/hypermil/linux/kernel/index.html>

##### 系统核心变化

<http://www.crynwr.com:80/kchanges>

##### 在笔记本上使用 Linux

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

##### Linux 相关资源

[http://www.redhat.com/linux-info/lq/linux\\_library.html](http://www.redhat.com/linux-info/lq/linux_library.html)

##### 非英语版的帮助文件

<http://wfn-shop.princeton.edu/~angel/manpages>

##### O'Reilly 公司的 Linux 书籍

<http://www.ora.com/catalog/linuxnut>

##### Linux 安全指南

<http://www.aoy.com/Linux/Security>

##### Linux 安全指南

<http://bach.cis.temple.edu/linux/linux-security>

##### 如何阅读 Linux 源代码

<http://sunsite.unc.edu/linux-source>

##### 目前最新的系统核心

<http://www.cs.helsinki.fi/cgi-bin/linuxversion>

##### Linux 下的 shareware

<http://www.users.dircon.co.uk/~trix>

##### 在线 RedHat 帮助

<http://www.redhat.com/support/docs/rhl/manual>

##### RPM 的主页

<http://www.redhat.com/support/docs/rpm>

#### 五、Linux 网络

##### SLIP 指南

<http://www.leland.stanford.edu/~wkn/Linux/slip/slip.html>

ISDN 指南

<http://www.ix.de/ix/linux/linux-isdn.html>

IPMASQ 指南

<http://www.hwy401.com/achau/ipmasq>

网络管理指南

<http://www.uni-tuebingen.de/zdv/projekte/linux/books/nag/nag.html>

PPP 指南

<http://www.cnw.com/~elarsen/linux/pppassis.html>

X.25 指南

<http://linas.org/linux/x.25.html>

在 Slackware 上运行 mgetty, pppd, Auto\_PPP

<http://www.gti.net/tiger/mgetty.html>

用 Linux 当路由器

[http://www.nicom.com/~twallace/linux\\_route](http://www.nicom.com/~twallace/linux_route)

QUE 的 Linux 书

<http://www.mcp.com/que/et/intlinux>

## 六、Linux 计划

自动产生 C/C++ 源代码的分析图

<http://www.fste.cowan.ac.edu.au/~tlittlef>

Debian Linux

<http://www.debian.org>

Linux 上的 JAVA

<http://www.blackdown.org/java-linux.html>

Linux 操作系统

<http://www.10mb.com/linux>

如何开发游戏

<http://www.ultranet.com/~bjhall/GSDK>

更好的编程界面

<http://blank.pages.de/lip>

数据采集和进程控制软件

<http://www.llp.fu-berlin.de>

可移植的 GUI 开发类库

<http://lamicounter.epfl.ch/OPaC>

在 PowerMAC 上运行基于微内核的 Linux

<http://www.mklinux.apple.com>

在 PowerPC 上运行 Linux

<http://www.linuxppc.org>

目前正在运行的 Linux 计划的状态

<http://www.ix.de/ix/linux/Projects-Map.html>

在 MCA 系统上运行的 Linux

<http://glycerine.cetmm.uni.edu/mca>

免费 WWW 浏览器

<http://www.tjhsst.edu/Mnemonic>

用户级的网络通信界面

<http://www2.cs.cornell.edu/U-Net/Default.html>

Linux 下的串口配置

<http://www2.tecnoland.es/~inaky>

Windows 模拟器

<http://dutifp.twi.tudelft.nl:8000/wine>

X.25 设备

<http://www.cyberspace.org/~krbonne/linux-x.25>

## 七、UNIX 资源

Unix 指南

<http://www.eecs.nwu.edu/unix.html>

Unix 使用

<http://wwwhost.cc.utexas.edu/cc/services/unix/index.html>

系统管理员指南

<http://www.polaris.net/ugu>

Unix 系统安全

<http://www.deter.com/unix>

GNU 的 Hurd

<http://www.cs.pdx.edu/~trent/gnu/hurd/index.html>

GNU 的图形界面 GNUSTEP

<http://www.gnustep.org>

另一个免费的 UNIX 操作系统

<http://www.freebsd.org>

## 八、X Window 系统资源

开发系统标准组织

<http://www.xopen.org>

免费的 XFree86 的主页

<http://www.xfree86.org>

X Window 的主页

<http://www.x.org>

X 下的工具,技巧,资源配置

<http://www.gaijin.com/X>

#### X 性能测试

<http://www.goof.com/xbench>

#### Fvwm 使用技巧

<http://mars.superlink.net/user/ekahler/fvwm.html>

#### 学习 X Window

<http://www.cs.curtin.edu.au/units/cg252-502/notes>

#### Kenton Lee 的 X Window 使用文件

<http://www.rahul.net/kenton/bib.html>

#### Motif 的主页

<http://ourworld.compuserve.com:80/homepages/efjohnson/motif.html>

#### X Window 下的应用软件

<http://www.ee.ryerson.ca:8080/~elf/xapps/faq.html>

#### X Window 下的图标

<http://www.inria.fr/koala/lehors/xpm.html>

#### X Window 使用的图片

<http://www.fri.uni-lj.si/~iztok/xwindow.html>

### 九、FTP 资源

#### Caldera 的 FTP 节点, 包括 CND 的镜像

<ftp://ftp.caldera.com/pub>

#### 著名的 Ftp.edrom.com

<ftp://wearchive.edrom.com:/pub/linux>

#### Debian FTP 站点

<ftp://ftp.debian.org>

#### Linux 核心的主 FTP 站点

<ftp://ftp.funet.fi/pub/Linux>

#### gatech FTP 站点

<ftp://ftp.cc.gatech.edu/pub/linux>

#### Pacific Hi-Tech FTP 站点

<ftp://ftp.pht.com/pub/linux>

#### Red Hat FTP 站点

<ftp://ftp.redhat.com/pub>

#### 最重要的 SunSITE FTP 站点

<ftp://sunsite.unc.edu/pub/Linux>

#### Linux 系统核心

<http://ftp.kernel.org/pub>

#### 排行第二的 tsx-ll FTP 站点

<ftp://tsx.mit.edu/pub/linux>

## 十、Linux 发布版本

Slackware 站点

<http://www.crl.com:80/~adras>

Caldera 站点

<http://www.caldera.com>

最便宜的 Linux 光盘

<http://www.cheapbytes.com>

一张软盘的 Linux

<http://www.cosmoseng.com>

Craftwork Linux

<http://www.craftwork.com>

一张软盘的 Linux

<http://www.wu-wien.ac.at/usr/h93/h9301726/dlx.html>

Infomagic 的 Linux 光盘

<http://www.infomagic.com>

用 NFS 安装 Slackware 或 Debian Linux

<http://nfs.internetp.com>

商业版的 Linux-FT 的主页

<http://www.lasermoon.co.uk>

Linux 书籍

<http://www.lsl.com>

Linux 书籍和 CD

<http://www.springer-ny.com:80/catalog/linux.html>

Linux 发行版本

<http://www.pht.com>

RedHat Linux

<http://www.redhat.com>

Linux 和 UNIX 资源

<http://www.ssc.com>

WGS Linux

<http://www.wgs.com>

Plug&Play Linux 和 Linux 圣经

<http://www.yggdrasil.com>

# C Linux 软件资源

## 一、一般节点

Linux 的商业信息指南

(免费) <http://sunsite.unc.edu/LDP/HOWTO/Commercial-HOWTO.html>

Linux 下商业软件列表

(免费) <http://sunsite.unc.edu/LDP/products.html>

Linux 在线手册

(免费) <http://www.ssc.com/linux/resources/man.html>

可搜寻的 Linux 软件包数据库

(免费) <http://www.boutell.com/lsm>

Linux 新软件发布消息

(免费) <http://www.cs.helsinki.fi/~wirzeniu/linux/cola.html>

X Window 软件

(免费) <http://www.ee.ryerson.ca:8080/~elf/xapps/faq.html>

## 二、系统管理工具

权限控制和用户管理

(免费) <http://acua.gist.net.au>

支持 APC 不间断电源

(免费) <http://www.dyer.vanderbilt.edu/server/apcupsd>

PERL script 用来处理系统记录

(免费) <http://www.iaehv.nl/users/grimaldo/chklogs.html>

进程管理工具

(商业) <http://www.activetools.com>

模拟 sun 的 storius 的设备收集机制

(免费) <http://www.xarius.demon.co.uk/software/devalloc>

软件工程

(免费) <http://www.iit.edu/~linjinl/esepl.html>

UNIX 系统配置

(免费) <http://www.foxnet.net/~apenwarz/figurine>

批处理系统

(免费) <http://www.shef.ac.uk/uni/projects/nqs>

Linux 进程的 3D 监视器

(免费) <http://www.actrix.gen.nz/users/michael/giveaways.html>

Linux x86 中断优先级调整程序

(免费) <http://www.best.com/~cae/irqtune>

Zip 和 Jaz 驱动器的驱动程序

(免费) <http://www.scripps.edu/~jsmith/jazip>

UPS 软件提供自动关机

(商业) <http://www.deltecpower.com/soft.html>

久经考验的 Linux 系统管理工具

(免费) <http://www.solucorp.qc.ca/linuxconf>

增强型 Linux 资源监视器

(免费) <http://www.zip.com.au/~bb/linux>

用 Motif 编制的监视用户使用磁盘空间

(免费) [http://www.ensta.fr/internet/unix/sys\\_admin/mfsm.html](http://www.ensta.fr/internet/unix/sys_admin/mfsm.html)

lackware 的 package tool 的追加工具

(免费) <http://www.kalika.demon.co.uk/sdk-software>

用网络上的时间服务器来校对本机器的时间

(免费) <http://www.icce.rug.nl/docs/programs/nist/nist.html>

Linux 身验证系统

(免费) <http://pare.power.net/morgan/Linux-PAM>

简单的系统资源监视器

(免费) <http://www.gedanken.demon.co.uk/procmeter/index.html>

用 TCL 编写的时间校对程序

(免费) <http://www.hsdi.com/qddb/applications/qtime-docs>

多重系统启动程序

(商业) <http://www.v-com.com/syscomm.html>

系统负载监视器

(免费) <http://www.xarius.demon.co.uk/software/xloadtime>

X Window 下的用户管理程序

(免费) <http://www.icce.rug.nl/docs/programs/xuserdoc/xuser.html>

### 三、网络工具

监视多个 UNIX 系统的网络情况,磁盘空间

(免费) <http://www.iti.qc.ca/iti/users/sean/bb-dnld>

Linux 光盘工作站

(免费) <http://www.damtp.cam.ac.uk/linux/bootpc>

用来管理 UNIX 系统上不同协议的脚本程序

(免费) <http://www.iu.hioslo.no/~mark/eftop.html>

ISP 管理用户的工具

(免费) <http://dayworld.net.au/~jsno>

client/server 配置管理系统

(免费) <http://www.cyclic.com>

hifs 的系统监视工具

(免费) <ftp://ftp.stack.nl/pub/linux/hifs>

Linux 上的移动计算

(免费) <http://mip.ee.nus.sg>

Linux 记帐程序

(免费) <http://www.geocities.com/SiliconValley/Lakes/2628/ipacct.html>

允许 Win/DOS 用户存取 UNIX 上的文件

(商业) <http://www.mpas.com>

监视网络堵塞情况并自动产生 HTML 文件

(免费) <http://www.ee.ethz.ch/~oetiker/webtools/mrtg>

Linux 上的 appletalk 协议

(免费) <http://artoo.hitchcock.org/~flowerpt/projects/linux-netatalk>

Linux 网络管理工具

(免费) [http://www.inka.de/sites/lina/linux/NetTools/index\\_en.html](http://www.inka.de/sites/lina/linux/NetTools/index_en.html)

ISP 管理系统

(商业) <http://www.regulus.safe.ca>

远程启动机器的工具

(免费) <http://128.113.163.180/linux/index.html>

显示目前子网上机器的启动时间和顺序

(免费) <http://ca006033.student.utwente.nl/linux/ruplist.html>

支持用户访问微软的文件打印共享服务

(免费) <http://samba.canberra.edu.au/pub/samba>

网络安全测试程序

(免费) <http://www.fish.com/satan>

TCL 编写的 SNMP 网络管理工具

(免费) <http://wwwsnmp.cs.utwente.nl/~schoenw/scotty>

网络管理和诊断程序

(商业) <http://www.MimeStar.com/secmain.htm>

Linux 版网络窃听器

(免费) <http://reptile.rug.ac.be/~coder/sniffit/sniffit.html>

串行口监视器

(免费) <http://www.roestock.demon.co.uk/serialmon>

一个简单的网络节点监视器

(免费) <http://strobe.weeg.uiowa.edu/~edhill>

建立在 scotty 基础上的网络管理程序

(免费) <http://wwwsnmp.cs.utwente.nl/~schoenw/scotty/#TKINED>

以太网阻塞监视程序

(免费) <http://www.ee.up.ac.za/~herman/htmls/publicationsandreleases.html>

#### 四、文件系统

从 AFS 2 发展而来的网络文件系统

(免费) <http://www.coda.cs.cmu.edu>

ext2 文件系统工具

(免费) <http://web.mit.edu/tytso/www/linux/e2fsprogs.html>

将 linux 分区 mount 成 win95 下的一个分区

(免费) <http://www.globalxs.nl/home/p/pvs>

苹果 Macintosh 的 HFS 文件系统

(免费) <http://www-sccm.stanford.edu/~hargrove/HFS>

在 UNIX 下访问 HFS 文件系统

(免费) <http://www.mars.org/~rob/proj/hfs>

不用 mount DOS 磁盘就可以直接访问他们的 UNIX 工具

(免费) <http://linux.wauug.org/pub/knaff/mttools>

在 Linux 下访问 Windows NT 的 NTFS

(免费) <http://www.informatik.hu-berlin.de:80/~loewis/ntfs>

mount 由 WFW, LanManager, NT 共享出来的盘

(免费) <http://www.SerNet.DE/vl/linux-lan>

透明的文件加密系统

(免费) <http://mikenos.dia.unisa.it/tcfs>

在 Linux 下访问 win95 的长文件名

(免费) <http://bmrc.berkeley.edu/people/chaffee/vfat.html>

#### 五、服务器

一个模块化的代理程序

(免费) <http://freenet.msp.mn.us/~olearydr/agentd/agentd.html>

LINUX 下管理音乐 CD 的商业程序

(商业) [http://www.psocd.com/e\\_cdfile.htm](http://www.psocd.com/e_cdfile.htm)

Magma 公司的实时通信服务器

(商业) <http://chat.magmacom.com>

DHCP 服务器, 动态分配 IP 地址

(免费) <http://www.fugue.com/dhcp/dhcp-beta-4.html>

PPP 和 SLIP 拨号服务器

(免费) <http://www.loonie.net/~eschenk/diald.html>

简单的通过电话线匿名下载文件的程序

(免费) <http://www.anders.com/~jeremyw/software.html>

同 FIDO NET 交换文件

(免费) <http://www.fido.de/fidogate>

Troll Tech 公司的 FTP 服务器

(免费) <http://www.troll.no/freebies/ftpd.html>

提供 PC 和 MAC 的打印服务

(商业) <http://www.infomagic.com/infomagic/wgs.html>

小规模 USENET 系统

(免费) <http://www.troll.no/freebies/leafnode.html>

一个主要支持 modem 的 getty 版本

(免费) <http://www.leo.org/~doering/mgetty>

高负载的 FTP 服务器

(商业) <http://www.probc.net/~mgleason/ncftpd/index.html>

getty 的替代程序

(免费) <http://www.cistron.nl/~miquels/radius>

实时性要求高的 Linux 系统核心

(免费) <http://luz.cs.nmt.edu/~rtllinux>

CMU 的 SNMP 的 Linux 版本

(免费) <http://www.gaertner.de/snmp>

基于 SPO256-A12 语音卡的读出文本文件的服务器

(免费) <http://www.tycho.com/packages/speak/index.html>

Xingtech 公司的视频点播服务器

(商业) [http://www.xingtech.com/products/sw\\_server.html](http://www.xingtech.com/products/sw_server.html)

Sybase SQL 服务器的 cgi 接口

(商业) <http://linux.sri.com>

建立在线服务的全套软件

(商业) <http://www.mmb.com>

网络时间协议和校对时间的程序

(免费) <http://www.eecis.udel.edu/~ntp>

NIS 服务器

(免费) <http://www.vt.uni-paderborn.de/~kukuk/linux/nis.html>

## 六、设备驱动程序

不同版本的系统核心中的驱动程序

(免费) <http://cesdis.gsfc.nasa.gov/linux/drivers/vortex.html>

AdvanSys SCSI 卡的驱动程序

(免费) <http://www.advansys.com/linux.html>

PCL-812PG 数据采集卡驱动程序

(免费) <http://dec50.vm.stuba.sk:80/~hruz/soft.html>

喷墨打印机驱动程序

(免费) <http://www.student.informatik.thdarmstadt.de/~wosch/pub/bubbletools/ANNOUNCE.html>

Mylex SCSI 卡驱动程序

(免费) <http://www.dandclion.com/Linux>

CS8900/20 网卡驱动程序

(免费) <http://www.crystal.com/drivers/crysoft.htm>

Matrox PIP-1024/ MuTech MV-200 摄像头的驱动程序

(免费) <http://www.powerup.com.au/~sobeyp>

GS4500 /GS4500A 扫描仪驱动程序

(免费) <http://swt-www.informatik.uni-hamburg.de/~willamo/linux.html>

Hayes ESP 卡驱动程序

(免费) <http://www.nyx.net/~arobinso/esp.html>

并口 Iomega Zip 驱动器驱动程序

(免费) <http://www.torque.net/zip.html>

Jaz 的驱动程序

(免费) <http://www.cnet.com/~bwillmot/jaztool>

CD 刻录格式 joliet

(免费) <http://bmrc.berkeley.edu/people/chaffee/joliet.html>

KS75 数据采集卡驱动程序

(免费) <http://www-control.uni-paderborn.de/linux.html>

Soundblaster AWE32 驱动程序

(免费) <http://bahamut.mm.t.u-tokyo.ac.jp/~iwai/midi.html>

如何编写驱动程序

(免费) <http://www.llp.fu-berlin.de/pool/software/dutil>

Gravis Ultrasound 声卡驱动程序

(免费) <http://www.pf.jcu.cz/%7Fperex/ultra>

Matrox Meteor PCI 摄像头驱动程序

(免费) <ftp://ftp.rwii.com/pub/linux/system/Meteor>

MuTech MV1000 摄像头驱动程序

(免费) <http://wpfd25.physik.uni-wuerzburg.de/~karrer/mv1000/mv1000.html>

mp3 解码器驱动程序

(免费) [http://www.nsm.de/nsm\\_it/e/products/software.htm](http://www.nsm.de/nsm_it/e/products/software.htm)

各种并口设备驱动程序

(免费) <http://www.torque.net/linux-pp.html>

Panasonic/Creative/Matsushita CR-56x 启动程序

(免费) <http://www.tarki.hu/Linux/ped.html>

pentium 芯片性能计数器驱动程序

(免费) <http://www.cs.umd.edu/users/akinlar/driver.html>

Digital 视频系统驱动程序

(免费) <http://www.dvsystems.com/changer>

Connectix QuickCam 摄像头驱动程序

(免费) <http://www.crynwr.com/qcpc>

声卡编程 API

(免费) <http://www.4front-tech.com/ossfree/index.html>

midi 播放器

(商业) <http://www.4front-tech.com/linux.html>

并口 yQuest EZ135/EZFlyer 230 驱动程序

(免费) <http://www.torque.net/ez135.html>

Xidar Technics 的 MicroLan 卡的驱动程序

(免费) <http://www.servtech.com/public/edudley/html/index.html>

MUSTEK Paragon 600/1200 SCSI 扫描仪驱动程序

(免费) <http://www4.informatik.uni-erlangen.de/~aoaugust/mustek/mustek.html>

## 七、X Window 系统

Xinside 的商业版 X server

(商业) <http://www.xig.com>

Metro Link 的商业版 X server

(商业) <http://www.metrolink.com/products/metrox/index.html>

管理图形界面的文档

(免费) <http://www.nijenrode.nl/~steven/ldp/HOWTO/MGR-HOWTO.html>

一个简单的单色窗口管理器

(免费) <http://www.modeemi.cs.tut.fi/~puujalka/wlr2.html>

Xfree86 的大本营

(免费) <http://www.xfree86.org>

## 八、shell 和系统工具

转换 rpm, dpkg, 或者 Slackware 的 tgz 软件包到 rpm 或者 dpkg 格式

(免费) <http://kite.ml.org/programs/alien>

shell 编程指南

(免费) [http://theory.u.winnipeg.ca/unixfiles/Unixhelp/shell\\_overview2.5.html](http://theory.u.winnipeg.ca/unixfiles/Unixhelp/shell_overview2.5.html)

CPU/cache/RAM 性能测试程序

(免费) <http://math.vanderbilt.edu/~mayer/linux/bmark.html>

如何配置 fvwmm

(免费) <http://www.imada.ou.dk/~blackie/dotfile>

ext2 文件系统压缩和解压缩

(免费) <http://netspace.net.au/~reiter/e2compr.html>

配置调试软驱

(免费) [http://linux.wauug.org/pub/knaff/\[dutils](http://linux.wauug.org/pub/knaff/[dutils)

同 X10 模块通讯

(免费) <http://heyu.tanj.com/~dbs>

免费的 PKZIP 解压程序

(免费) <http://quest.jpl.nasa.gov/Info-ZIP>

ISA 总线 PnP 驱动程序

(免费) <http://www.roestock.demon.co.uk/isapnptools>

切换目录的小工具

(免费) <http://www-sef.usc.edu/~lerdsuwa/util>

文件列表小工具

(免费) <http://www.interlog.com/~acli/mfcf/lc>

module 支持工具和文档

(免费) <http://www.pi.se/blox>

一个同 DOS 环境相象的 shell

(免费) <http://www.cs.uct.ac.za/~mwelz/lsh.html>

有人 finger 你的时候提醒用户的工具程序

(免费) <http://www.oneimage.com/~stempel/utlis>

Debian 格式软件包到 RedHat (rpm) 格式软件包的转换程序

(免费) <http://www.people.cornell.edu/pages/rc42/program/martian.html>

C + + 编写的监视调整解调器的程序

(免费) <http://slug.ctv.es/~emclero/momond>

Apple Newton PDA 同 UNIX 交换数据的程序

(免费) <http://privat.swol.de/ReinholdSchoeb/Newton/newtonlink-doc-en.html>

简单字符菜单

(免费) <http://kite.ml.org/programs/pdmenu>

同 RPM 兼容的软件包管理程序

(免费) <http://www.metrotron.com/pub/packages/pkg>

文件管理程序

(商业) <http://www.usregsite.com>

rexec 程序

(免费) <http://www.msc.cornell.edu/~sadd/rexec.html>

RPM 的大本营

(免费) <http://www.rpm.org>

有背景图形的 rxvt

(免费) [http://www.cse.unsw.edu.au/~s2154962/linux\\_rxvt.html](http://www.cse.unsw.edu.au/~s2154962/linux_rxvt.html)

用 zsh 编写的自动追加签名档或 PGP 公开密钥的工具

(免费) <http://www.niweb.com/pgregg/siged>

支持 32 位微内核的 shell

(免费) <http://www.csh.rit.edu/~shaggy/software.html>

一个安全的 shell

(免费) <http://www.cs.hut.fi/ssh>

SVGA 驱动程序

(免费) <http://www.ecc.nottingham.ac.uk/~etzpc/stm.html>

tesh 大本营

(免费) <http://pantheon.cis.yale.edu/~des/tesh.html>

搜寻不安全的符号连接

(免费) <http://www.kalika.demon.co.uk/sdk-software>

X Window 下的时间校准程序

(免费) <http://domen.uninett.no/~hta/titrax/TimeTracker.html>

图形 UNIX 命令界面

(免费) <http://www.phys.ksu.edu/~bailey/tkWorld.html>

图形压缩文件管理器

(免费) <http://www.penet.com/~proteus/TkZip/TkZip.html>

安装 Type1 字库的 perl 程序

(免费) <http://goblet.unu.edu.au/~m9305357/typelinst.html>

dos 下的解 tar 和 gz 文件的程序

(免费) <http://www.isjm.com/tst/untgz.htm>

uuencode, MIME, base64, xxencode 的解码和编码程序

(免费) <http://www.uni-frankfurt.de/~fp/uudeview>

TK 的增强图形窗口

(免费) <http://www.cs.uoregon.edu/research/tcl/script/tkcon>

shell 程序的 widgets

(免费) <http://espar.uah.edu/~mallozzir/html/widtools.html>

在 XDM 上产生漂亮的背景

(免费) <http://chaos.fullerton.edu/XBanner>

基于 Xform 的 module 管理程序

(免费) <http://www.aviso.demon.co.uk/xfmod.html>

用 Motif 编写的 CPU 状态监视器

(免费) <http://www.xs4all.nl/~ripley/RSD/linux.html>

X window 查看系统通知的程序

(免费) <http://www.ee.ryerson.ca:8080/~elf/xmotd.html>

如何加快 X 程序启动时间

(免费) <http://www.hacom.nl/~richard/software/xtoolwait-1.0.README>

另外一种 yacc

(免费) <http://www.thinkage.on.ca/shareware/yay>

Zsh 的大本营

(免费) <http://www.peak.org/zsh>

## 九、桌面和窗口管理器

像 NeXT 的窗口管理器

(免费) <http://afterstep.edoc.com>

像 Amiga Workbench 的窗口管理器

(免费) <http://www.lysator.liu.se/~marcus/amiwm.html>

Linux 下的 CDE

(商业) <http://www.xinside.com/pd/edline.html>

可见部分最多的窗口管理器

(免费) <http://www.rasterman.com>

X Open 桌面

(免费) <http://www.simplicity.net/exode>

商业窗口管理器

(商业) <http://fsw.com/desktop.html>

fvwm 的主页

(免费) <http://www.hpc.uh.edu/fvwm>

自行配置 fvwm

(免费) <http://www.cohaltgroup.com/~roland/fvwm/fvwm.html>

fvwm 上的 CDE 模拟程序

(免费) <http://www.dilnet.upd.edu.ph/~orly/unixcentral/FvwmCDE.html>

fvwm95 的大本营

(免费) <http://www.terraware.net/ftp/pub/Mirrors/FVWM95/fvwm95.html>

fvwm 的工具

(免费) <http://bat710.univ-lyon1.fr/~f-cord96/fvwmserp.htm>

窗口管理器 gwm

(免费) <http://zenon.inria.fr/koala/gwm>

KDE 的主页

(免费) <http://www.kde.org>

像 MAC OS 的窗口管理器

(免费) <http://www.biocle.nuee.nagoya-u.ac.jp/member/tak/mlvwm.html>

模拟 CDE

(免费) <http://www.d.umn.edu/~mbaron/panel/index.html>

drag & drop 的桌面

(免费) <http://leh.net/OffiX>

用 TK 编写的桌面

(免费) <http://people.mainz.netsurf.de/~bolik/tkdesk>

X 程序管理器

(免费) <http://www-personal.umich.edu/~markerim/tkgoodstuff>

类似于 NeXT 的窗口管理器

(免费) <http://www.inf.ufrgs.br/~kojima/wmaker/index.html>

小型窗口管理器

(免费) <http://www.netcomuk.co.uk/~cannam/wm2.html>

X Window 工具栏

(免费) <http://croco.atnet.at/club/spiff/xmtoolbar.html>

## 十、文件和磁盘工具

### 文件管理器

(免费) <http://www-c.informatik.uni-hannover.de/~kaiser/dfm/dfm.html>

### C-ISAM 标准的文件管理器

(免费) <http://www.byted.com/disam96a.html>

### 文件管理器

(免费) <http://www.bongo.demon.co.uk/fileman>

### 快速文件管理器

(免费) <http://www.ed.chalmers.se/~hch/filerunner.html>

### 文件搜寻工具

(免费) <http://glimpse.cs.arizona.edu:1994>

### 文件浏览器支持 HEX

(免费) <http://www.cs.unh.edu/~tudor/git>

### 类似 Windows95 的文件管理器

(免费) <http://jungfrau.ptf.hro.nl/explorer>

### 通过 FTP 的文件管理器

(免费) <http://www.llnl.gov/ia/xdir.html>

### 字符界面类似 Norton Commander 的文件管理器

(免费) <http://mc.blackdown.org/mc>

### 用 Motif 编写的文件管理器

(免费) <http://sugra.desy.de/user/mai/moxfm>

### 查看 ZIP 文件的程序

(商业) [http://www.softwarebuilders.com/SBI\\_Mall/NetZIP\\_Server.html](http://www.softwarebuilders.com/SBI_Mall/NetZIP_Server.html)

### 免费命令管理程序

(免费) <http://home.inreach.com/xpierre/pash>

### 图形 Info 浏览器

(免费) <http://www.math.ucsb.edu/~boldt/tkinfo>

### X Window 下类似于 Norton Commander 的文件管理器

(免费) <http://www.UniX11.com>

### 图形文件管理器

(免费) <http://www.hut.fi/u/mkivinie/X-Files>

### Openlook 标准的文件管理器

(免费) [http://www.info.polymtl.ca/ada2/coyote/www/xvfilemgr\\_main.html](http://www.info.polymtl.ca/ada2/coyote/www/xvfilemgr_main.html)

## 十一、备份和恢复系统

### 大规模网络备份系统

(商业) <http://www.knox-software.com>

## 网络备份工具

(商业) <http://www.estinc.com>

## 强大的商业备份系统

(商业) <http://www.unitrends.com>

## 磁带机驱动程序

(免费) <http://samuel.math.rwth-aachen.de/~LBFM/clauss/ftape/ftape-page.html>

## Unix 备份系统

(免费) <http://www.phy.hw.ac.uk/~karsten/KBackup.html>

## Lone Star 公司的备份软件

(商业) <http://www.cactus.com>

## UNIX/LINUX 备份软件

(商业) <http://www.unisrc.com>

## 图形磁带机备份系统

(免费) <http://www.multiline.com.au/~yusuf>

## 十二、打印工具

### 将文本文件转换为 PS 文件

(免费) <http://www-stud.enst.fr/~demaille/a2ps.html>

### 自动辨认打印文件格式的 Perl 脚本文件

(免费) <http://www.ci.tuwien.ac.at/~leisch/docs/flpr/flpr.html>

### 使用 Linux 字符终端的字库来打印文本文件

(免费) <http://www.glue.umd.edu/~weave/vmwprod/vmwsoft.html>

### 在一页纸上打印两页文件

(免费) <http://www2.awinc.com/users/bvdpoel/#lsoft>

### PS 文件打印过滤器

(免费) <http://ppr-dist.trincoll.edu/printing/pprfaq.html>

## 十三、模拟器和仿真器

### Atari 800, 800XL, 130XE, 5200 模拟器

(免费) <http://www.signus.demon.co.uk/david/atari/atari.html>

### Xwindow 下的 X86 模拟器(可以运行 Win95)

(商业) <http://world.std.com/~bochs>

### Motorola 68000 & HECTOR 1600 模拟器

(免费) <http://www2.ncsu.edu/eos/service/ece/project/bsvc/www>

### Linux 下的 DOS 模拟器

(免费) <http://www.suse.com/~dosemu>

### Oric 模拟器

(免费) [http://www.ensica.fr/~frances/oric/emulate\\_english.html](http://www.ensica.fr/~frances/oric/emulate_english.html)

### Macintosh 模拟器

(商业) <http://www.ardi.com>

FreeBSD 下的 Linux 仿真器

(免费) <http://utgard.bgu.com/handbook/handbook269.html#594>

C64 模拟器

(免费) <http://www.uni-mainz.de/~bauec002/FRMain.html>

任天堂 Nintendo 模拟器

(免费) <http://freeflight.com/fms/iNES>

intel 多处理器模拟器

(免费) <http://galeb.etf.bg.ac.yu/~dav0r/limes/index.html>

SEGA GameGear 模拟器

(免费) <http://freeflight.com/fms/MG>

超级 Nintendo 模拟器

(免费) <http://www.euronet.nl/users/jkoot/index.htm>

Atari 2600 模拟器

(免费) <http://www4.ncsu.edu/~bwmott/www/2600>

Xwindow Atari ST 模拟器

(免费) <http://www.complang.tuwien.ac.at/nino/stonx.html>

Amiga 模拟器

(免费) <http://cestar.technion.ac.il/~reuven/hetz>

在 Linux 下直接运行 VB5 程序

(免费) <http://Softworks1td.com/vbvm.html>

C64, C128, Vic20, PET 模拟器

(免费) <http://fripp.lurz.tu-chemnitz.de/~fachat/vice>

Atari 2600 模拟器

(免费) <http://www.users.zetnet.co.uk/ahornby/v2600.html>

任天堂 Gameboy 模拟器

(免费) <http://www.freeflight.com/fms/VGB>

商业版 linux 下的 windows 模拟器

(商业) <http://www.caldera.com/products/updates/wabi.html>

一个开发中的免费 windows 模拟器

(免费) <http://www.linpro.no/wine>

MultiArcade 模拟器

(免费) <http://www.dit.upm.es/~jantonio/mame>

Sinclair ZX Spectrum 48/128/+3 模拟器

(免费) <http://www.philosys.de/~kunze/xzx>

Sinclair ZX Spectrum 模拟器

(免费) <http://www.comlab.ox.ac.uk/oucl/users/ian.collier/Spectrum>

#### 十四、电子表格

Lotus 1-2-3 兼容的电子表格

(免费) [http://home.t-online.de/home/Axel\\_Reinhold/#mc](http://home.t-online.de/home/Axel_Reinhold/#mc)

网络版电子表格

(商业) <http://www.xess.com/prodinfo.html>

X Window 下的电子表格

(免费) <http://www.edu.stockholm.se/siag>

三维电子表格

(免费) <http://cantor.informatik.rwth-aachen.de/~michael/projects/teapot-en.html>

NEXT 和 Mac 电子表格 Linux 版

(商业) <http://www.wingz.com>

商业版 Linux 下的电子表格

(商业) [http://www.ais.com/linux\\_corner.html](http://www.ais.com/linux_corner.html)

包括科学和金融计算的电子表格

(商业) <http://www.axene.com/english/xquad.html>

#### 十五、数据库

高性能数据库管理程序

(商业) <http://www.softwarcag.com/corporat/solutions/datamanager/default.htm>

免费 client/server 数据库

(免费) <http://tiny.iapnet.com/rob/beagle.html>

应用程序开发和数据库引擎

(免费) [http://www.conetic.com/css\\_freelkit.html](http://www.conetic.com/css_freelkit.html)

兼容 NSI 和 DBC 的 DBMS

(免费) <http://www.cql.com>

多用户的 SQL 服务器

(商业) <http://www.uno.com/dbix.html>

单用户数据库系统

(商业) [http://www.empress.com/store/p\\_emp\\_1/linux1.htm](http://www.empress.com/store/p_emp_1/linux1.htm)

指令简单的数据库引擎

(免费) <http://www.inter-soft.com/html/products/essentia.html>

同 InterSoft SQL 服务器连接的 JAVA 程序

(免费) <http://www.inter-soft.com/html/products/essentia-jdbc.html>

Clipper 5.x 和 Dbase III 的 UNIX 版

(商业) <http://www.wgs.com/fsad.html>

免费的多用户数据库系统

(免费) <http://www.ispras.ru/~kml/gss>

client/server 的数据库系统

- (商业) <http://www.justlogic.com/descr.html>  
商业级高速 client/server 数据库引擎
- (商业) <http://www.kesoftware.com/kesoft/tex.html>  
UNIX 版 RDBMS
- (商业) <http://www.kubl.com/kubl/kubl.htm>  
UNIX 下的一个关系数据库
- (免费) <http://www.dogbert.demon.co.uk/leap.html>  
支持 ANSI SQL 的一个子集的轻量级数据库系统
- (免费) <http://www.Hughes.com.au>  
高速多线程 SQL 服务器
- (免费) <http://www.tcx.se>  
Sybase' 的 Client-Library Linux 版
- (免费) <http://www.sybase.com/products/samples/linux>  
在 Linux 上使用 SCO UNIX 版的 Oracle
- (免费) [http://www.pauck.de/marco/misc/oracle\\_on\\_linux.html](http://www.pauck.de/marco/misc/oracle_on_linux.html)  
面向对象, 事件驱动, 实时的 DBMS 系统
- (商业) <http://www.polyhedra.com/realtime.htm>  
著名的 postgresql
- (免费) <http://www.postgresql.org>  
关系数据库
- (商业) <http://www.hsdi.com/qddb>  
高速关系数据库
- (商业) <http://www.raima.com/rdmplus.htm>  
关系数据库工具
- (商业) <http://www.gnacademy.org:8001/uu-gna/schools/vsliis/rdb>  
Xbase 兼容的多用户数据库
- (商业) <http://www.recital.com/cmr.html>  
SHORE 一个面向对象的数据库
- (免费) [ftp://ftp.cs.wisc.edu/shore/html/index.html](http://ftp.cs.wisc.edu/shore/html/index.html)  
Sybase 的 isql 的替代软件
- (免费) <http://www.voicenet.com/~gray/sqsh.html>  
Solid Tech. 的 client/server DBMS
- (商业) <http://www.solidtech.com>  
SQL 开发工具
- (免费) <http://www.wgs.com/sqlkit/sqlkit.html>  
一个 Linux 下的文本文件数据库
- (商业) <http://www.primenet.com/~keithr/ssfile>  
Xbase 编译器
- (免费) <http://www.on-the-net.com/x2c>

一个简单的图形界面数据库接口

(免费) <http://www.1N-Berlin.DE/User/bitrot/grok.html>

SQL 数据库, 兼容 ANSI X3.135 标准

(商业) <http://www.yard.de/>

## 十六、字处理软件

pdf 文件阅读器

(免费) <http://www.adobe.com/prodindex/acrobat/lindnd.html>

基于 motif 的“所见即所得”文字处理器

(免费) <http://zeus.informatik.uni-frankfurt.de/~weis/cicero.html>

SGML 处理器

(免费) <http://www.art.com/cost>

PS 文件解释器

(免费) <http://www.cs.wisc.edu/~ghost/index.html>

将 ASCII 文件转换为 PS 文件

(免费) <http://www.hut.fi/%7Emtr/genscript>

ghostview, PS/PDF 阅读器

(免费) <http://www.thep.physik.uni-mainz.de/~plass/gv>

文本文件处理器

(免费) <http://pi0959.kub.nl:2080/Paa1/Publick>

著名的 LaTeX 排版软件

(免费) <http://www.tex.ac.uk/CTAN/LaTeX>

LaTeX 的“所见即所得”编辑器

(免费) <http://www.lehigh.edu/~dlj0/LyriX.html>

编辑超文本帮助的编辑器

(免费) [http://public.logica.com/~kempsonc/mh\\_index.htm](http://public.logica.com/~kempsonc/mh_index.htm)

TeX 的一个版本

(免费) <http://web.mathematik.uni-stuttgart.de/ntex>

X11/Motif 的文字处理器

(免费) [http://or.mime.univ-paris8.fr/~loscar/Papyrus/papyrus\\_eng.html](http://or.mime.univ-paris8.fr/~loscar/Papyrus/papyrus_eng.html)

排版软件

(免费) <http://www.cs.berkeley.edu/~ame/Par>

使用 type 1 字库“所见即所得”

(免费) <http://www.uni-paderborn.de/~SciTeX>

SGML 处理工具

(免费) <http://pobox.com/~cg/sgmltools>

“所见即所得”TCL 编辑器

(免费) <http://www.tcltk.com/tclets>

LaTeX 的一种版本——teTeX

(免费) <http://www.tug.org/teTeX>

“所见即所得”文本编辑器

(免费) <http://opera.inrialpes.fr/thot>

将 Tex 文件转换为 HTML 文件

(免费) <http://venus.pfc.mit.edu/tth/tth.html>

超文本转换程序

(免费) <http://members.aol.com/DirkHage/gt/udo6.html>

在线 PS 浏览器

(免费) <http://www.research.digital.com/SRC/virtualpaper>

troff 兼容的处理器

(免费) <http://www2.awinc.com/users/bvdpoel/#lsoft>

字处理软件

(商业) <http://www.caldera.com/products/updates/wmb.html>

Unix 下的 Word Perfect 7

(商业) <http://www.sdcorp.com>

“所见即所得”文本出版软件

(商业) <http://www.axene.com/english/xclamation.html>

TeX 预览软件

(免费) <http://math.berkeley.edu/~vojta/xdvi.html>

X Window 下阅读 PDF 文件

(免费) <http://www.aimnet.com/~derek/xpdf>

Tex/LaTeX 的图形用户界面

(免费) [http://ftp.lrw.uni-bremen.de/xtem/xtem\\_textmenu.html](http://ftp.lrw.uni-bremen.de/xtem/xtem_textmenu.html)

字处理软件

(免费) <http://www.hungry.com/products>

将 YODL 文本转换为其他格式的文件

(免费) <http://www.icce.rug.nl/docs/programs/yodl/yodl.html>

## 十七、群件和办公组件

AUIS 群件

(免费) <http://www.cs.cmu.edu/~AUIS>

办公群件

(商业) <http://www.angoss.com/sw/home.htm>

Linux 的 Applixware

(商业) <http://www.redhat.com/products/product-details.phptml?id=apx>

办公群件

(商业) <http://www.quad.com/linux.htm>

Credit Card 校验系统

(商业) <http://www.hks.net>

Compuwork 市场软件

(商业) <http://www.compuwork.com/linux.html>

UNIX 下的群件

(商业) <http://www.goldmedal.com/gm/index.html>

网络会议系统

(商业) <http://rendezvous.visualtek.com>

著名的 Star Office

(免费) <http://www.stardivision.com/staroffice/linux.html>

实时日程安排软件

(商业) <http://www.crosswind.com/oview.htm>

## 十八、科学与工程数学工具

天文软件

(免费) <http://bima.astro.umd.edu/nemo/linuxastro>

CD-ROM 版天文软件

(商业) <http://www.earthlink.net/~rfactory/lfa.html>

FSM 到 VHDL 转换器

(免费) <http://tech-www.informatik.uni-hamburg.de/vhdl>

热力学计算

(商业) <http://gttserv.lth.rwth-aachen.de/~sp/lt/chemapp/chemapp.htm>

绘制图表的软件

(免费) <http://www.informatik.uni-bremen.de/~davinci>

将 ASCII 中的数据转换为 EPS 格式的文件

(免费) <http://wwwi.wu-wien.ac.at/home/ckarl/ckarl/diag/index.html>

实验物理工控软件

(免费) <http://www.aps.anl.gov/asd/controls/epics>

高能物理数学 1/4 扑闯研 \*

(免费) <http://pcue.hep.phy.cam.ac.uk/ben/foam/foam.html>

画图软件

(免费) <http://www.fmi.uni-passau.de/Graphlet>

地理资源分析软件

(免费) <http://www.cecer.army.mil/grass>

科学绘图软件 GNU plot

(免费) [http://www.es.dartmouth.edu/gnuplot\\_info.html](http://www.es.dartmouth.edu/gnuplot_info.html)

4GL 开发包

(商业) <http://www.rsinc.com>

图像信号处理系统

(商业) <http://www.khorol.com/khoros/whatis.html>

可视数据处理系统

(免费) <http://linkwinds.jpl.nasa.gov>

代数和数值计算软件

(商业) <http://www.maplesoft.com/Products/MapleV/MapleV.html>

有限元方程计算

(商业) <http://www.marc.com>

数学公式分析软件

(商业) <http://www.wri.com/mathematica>

编辑数学论文的软件

(免费) <http://www.win.tuc.nl/win/es/wp/mathspad>

Matlab 到 C + + 的转换器

(商业) <http://www.mathtools.com/matcom.html>

数值计算

(商业) <http://www.mathworks.com/ml5.shtml>

CAD 软件

(商业) <http://www.bentley.com/academic>

数值计算和分布式计算

(免费) <http://www.mupad.de/factsheet.html>

支持 DAQ 的 64 bit 时间保持软件

(免费) <http://www.erols.com/rgerharz/Linux>

神经网络开发和模拟软件

(免费) <http://www.bgif.no/neureka>

数值计算高级语言

(免费) <http://www.che.wisc.edu/octave>

绘图软件

(免费) <http://aixpdslib.seas.ucla.edu/plot-tools/plotmtv4.html>

科学数据绘图软件

(商业) <http://www.powerroar.com/~graph>

有限元软件

(商业) <http://www.comco.com/main/phlex/ProPhlex.html>

信号处理和控制系统

(免费) <http://ptolemy.eecs.berkeley.edu>

LISP 格式的统计语言

(免费) <http://www.stat.auckland.ac.nz/rproj.html>

类似于 matlab

(免费) <http://www.eskimo.com/~ians/r/lab.html>

免费科学数据处理和绘图软件

(免费) <http://heasarc.gsfc.nasa.gov/docs/xte/SOF/robot.html>

面向对象的大规模数据处理系统

(免费) <http://root.cern.ch>

实时跟踪卫星

(商业) <http://www.primenet.com/~bester/sattrack.html>

科学和工程软件

(免费) <http://SAL.KachinaTech.COM>

高级科学和数学计算软件

(免费) <http://www-rocq.inria.fr/scilab>

统计软件

(商业) <http://shazam.econ.ubc.ca>

代数计算

(免费) <http://emmy.math.uni-sb.de/~simath>

神经网络模拟

(免费) <http://www.informatik.uni-stuttgart.de/ipvr/bv/projekte/snns/snns.html>

复杂系统模拟

(免费) <http://www.santafe.edu/projects/swarm>

手写体辨认

(免费) <http://hwr.nici.kun.nl/unipen/uptools3>

CAD 软件

(免费) <http://www.microform.se>

5D 数据处理

(免费) <http://www.ssec.wisc.edu/~billh/vis5d.html>

生物软件

(免费) <http://www.ks.uiuc.edu/Research/vmd>

统计软件

(免费) <http://a42.com/~thor/xldlas>

细胞模拟软件

(免费) <http://penguin.phy.bnl.gov/www/xtoys/xtoys.html>

将分子模型转换为图形

(免费) <http://world.std.com/~wware/xyz2rgb.html>

## 十九、个人软件

个人地址簿

(免费) <http://www.mayr.informatik.tu-muenchen.de/~durka/addressbook>

库存管理, 技术分析软件

(商业) <http://www.falkor.com>

支票管理软件

(免费) <http://www.menet.umn.edu/~clolson/cbb/>

日历和备忘录

(免费) [http://www.research.digital.com/SRC/personal/Sanjay\\_Ghemawat/ical/home.html](http://www.research.digital.com/SRC/personal/Sanjay_Ghemawat/ical/home.html)

日历和行程表

(免费) <http://www.IN-Berlin.DE/User/bitrot/plan.html>

财政管理软件

(免费) <http://www.es.auc.dk/~lupus/xfinans.html>

个人财务管理

(免费) <http://sunsite.auc.dk/xinvest/xinvest.html>

## 二十、网络浏览器和 Plug-In

网络浏览器和 HTML 编辑器

(免费) <http://www.w3.org/pub/WWW/Amaya>

Yggdrasil 提供的免费浏览器

(免费) <http://www.yggdrasil.com/Products/Arena>

WWW 浏览器

(免费) <http://www.candleweb.no>

最早的浏览器之一

(免费) <http://www.unlv.edu/chimera>

用 Emacs lisp 编写的浏览器

(免费) <http://www.cs.indiana.edu/elisp/w3/docs.html>

CNRI 编写的浏览器

(免费) <http://monty.cnri.reston.va.us/grail-0.3>

cookie 过滤软件

(免费) <http://www.junkbusters.com/ht/en/ijb.html>

著名的文本浏览器 lynx

(免费) <http://lynx.browser.org>

如何为 Netscape 编写 Plugin

(免费) <http://www.tcltk.com/tclets/impress>

小型浏览器

(免费) <http://pauillac.inria.fr/~rouaix/mmm>

经典浏览器

(免费) <http://www.ncsa.uiuc.edu/SDG/Software/XMosaic>

Netscape 的 plugin

(免费) <http://mbpc07.cl.ba-karlsruhe.de/netscape-plugins.html>

一个小浏览器

(免费) <http://www.cdm.com.mx/newswire>

Netscape

(商业) <http://www.caldera.com/products/netscape/netscape.html>

communicator4.04

(免费) [http://home.netscape.com/download/client\\_download.html?communicator4.04](http://home.netscape.com/download/client_download.html?communicator4.04)

通过电话浏览网络

(免费) <http://www.rci.rutgers.edu/~phoynet>  
Netscape 的多媒体 plugin  
(免费) <http://www.infovav.se/~hubbe/plugger.html>  
Netscape 的多媒体 plugin  
(免费) <http://home.pages.de/~rasca/pngplugin>  
SGML 浏览器  
(免费) <http://sunsite.auc.dk/qweb>  
用 Tk 编写浏览器  
(免费) <http://tk-www.mit.edu:8001/tk-www/help/overview.html>  
VRML 浏览器  
(免费) <http://hyperg.iicm.tu-graz.ac.at/vrweb>  
支持 MEAPI 的浏览器  
(免费) <http://www.eolas.com/eolas/webrouse>  
离线浏览器  
(免费) <http://www.gedanken.demon.co.uk/wwwoffle/index.html>  
在 WWW 上浏览 info 文件  
(免费) <http://www.fokus.gmd.de/linux/announce.html>

## 二十一、WWW 工具和 HTML 编辑器

### 4GL 处理器

(免费) <http://www.msn.ac.uk/~whaley/ag/appgen.html>

### 简单的 HTML 编辑器

(免费) <http://www.cs.rpi.edu/~puninj/TALK/head.html>

### 文本和 HTML 编辑器

(免费) <http://www.advasoft.com/asWedit.html>

### 商业版 HTML 编辑器

(商业) <http://www.advasoft.com/ASWedit.html>

### WWW BBS

(商业) <http://www.lilikoi.com>

### Empress RDBMS 的 WWW 接口

(商业) <http://www.empress.com/store/dataweb/dataweb1.htm>

### 支持 gifmap 的编辑器

(免费) <http://www.cyberramp.net/~bfriesen/gifmap>

### 数据库 WWW 接口

(免费) <http://www.h-e-i.de/heitml>

### 用 emacs 编辑 HTML

(免费) <http://www.tnt.uni-hannover.de/~muenkel/software/own/hm--html-menus/overview.html>

### 转换 LaTeX 到 HTML 文件

(免费) <http://cbl.leeds.ac.uk/nikos/tex2html/doc/LaTeX2html/LaTeX2html.html>  
WWW 市场调查

(商业) <http://www.lilypad.net>

建立节点导航图

(商业) <http://www.elsop.com/linkscan/index.html>

编辑 gifmap

(商业) <http://www.boutell.com/mapedit>

HTML 编辑器

(免费) <http://www-c.informatik.uni-hannover.de/~dh/mulaw>

如何开发 CGI 程序

(商业) <http://www.nombas.com/sewse/index.htm>

CGI 的 SET UID 包装

(免费) <http://marion.uio.No/~sverrehu/pub-unix>

用 cl/tk 编写的 HTML 编辑器

(免费) <http://www.cohaltgroup.com/~roland/tkHTML/tkHTML.html>

用 perl 编写的 robot

(商业) <http://vancouver-webpages.com/VWbot>

允许用户在线编辑 html 文件

(免费) <http://www.infomedque.ca/~webadmin>

数据库到 Web 的接口

(商业) <http://www.talentsoft.com>

利用 Apache 额 perl 浏览用户目录

(免费) [http://www.esica.fr/public\\_html/Emmanuel.PIERRE/WebLx.en.html](http://www.esica.fr/public_html/Emmanuel.PIERRE/WebLx.en.html)

在网络上收集图像

(免费) <http://www.cyberramp.net/~bfriesen/webmagick>

用 WWW 改用户密码

(免费) <http://www.win.co.nz/web-passwd>

HTML 图形编辑器

(免费) <http://www.sunlabs.com/research/tcl/webtk>

WWW 服务器访问分析软件

(免费) <http://www.webthreads.com>

## 二十二、电子邮件代理软件

mail 代理

(免费) <http://www.bitgate.com/spam>

e-mail 分页工具

(商业) <http://www.sos.com/cyberpage>

ELM

(免费) <http://www.mysa.com/elm.html>

X Window 下的 mail 程序

(免费) <http://www.sunlabs.com/~bwelch/exmh>

支持 POP2, POP3, APOP 和 IMAP 协议的 mail 收发程序

(免费) <http://www.ccil.org/~esr/esr-freeware.html>

pop3 取信程序支持调制解调器

(免费) <http://www.concentric.net/~mrsam/getpop3/index.html>

FTN 到 UNIX 转换软件

(免费) <http://www.averse.org/ifmail>

商业 X Window 下的 mail 软件

(商业) <http://www.ishmail.com>

QWK 离线读信软件

(免费) <http://www.iki.fi/jvuokko/jmr>

maillist 管理软件

(免费) <http://www.greatcircle.com/majordomo>

UNIX 下的 mail 代理软件

(免费) <http://www.ics.uci.edu/~mh>

X 下的 IMAP mail 软件

(免费) <http://camis.Stanford.EDU:80/projects/imap/ml>

离线 mail 阅读器

(免费) <http://goliat.eik.bme.hu/~johnzero/multimail>

小而精悍的 mail 软件

(免费) <http://www.cs.hmc.edu/~me/mutt>

X/Motif 的 mail 软件

(免费) <http://www.ajsoft.demon.co.uk/mxMail.html>

Majordomo 的 Web 接口

(免费) <http://www.ed.umuc.edu/~rshecter/software/pandora.html>

类似于 majordomo 的软件

(免费) <http://www.gmd.de/People/Peter.Simons/private/software/petidomo.html>

PINE,最常用的 mail 软件

(免费) <http://www.cac.washington.edu/pine>

用 PGP 来收发 mail

(免费) <http://www.c2.org/~raph/premail.html>

POP3 代理

(免费) <http://caristudenti.cs.unibo.it/~borgia/homepage/Software/Software.html>

利用 PGP 收发 mail

(免费) <http://www.unicorn.com/privtool/privtool.html>

自动处理软件

(免费) <http://www.ii.com/internet/robots/procmail>

Sendmail 的替代品

(免费) <http://www.qmail.org>

Majordomo 的小工具

(免费) <http://www.ed.umuc.edu/~rshecter/software/ri.html>

最常用的 Mail 代理程序

(免费) <http://www.sendmail.org>

X Window 下的 mail 阅读软件

(免费) <http://www.stud.uni-hannover.de/~kuehnel>

X Window Mail 阅读器

(免费) <http://www.slac.stanford.edu/~raines/tkmail.html>

Mail 阅读器

(免费) <http://www.crocodile.org/UMT/UMT.html>

将 mail 转换为 html 文件

(免费) [http://www.lugs.ch/1.UGS\\_Members/norbert.kuemin/wmf-0.8.html](http://www.lugs.ch/1.UGS_Members/norbert.kuemin/wmf-0.8.html)

X Window 下的 Mail 阅读器

(免费) <http://burka.netvision.net.il/xfmail/xfmail.html>

基于 Motif 的商业 Mail 阅读器

(商业) <http://www.netmanage.com/products/zmail/index.html>

## 二十三、新闻阅读器和网络软件

ircII 2.8 网络聊天程序

(免费) <http://www.sasknet.com/~pionh/index.htm>

新的图形界面 IRC

(免费) <http://www.nijenrode.nl/~ivo/circus>

实时音频通信

(免费) <http://www.wpi.edu/~murti/work.html>

X Window 下配置 PPP 的软件

(免费) <http://www.serv.net/~cameron/ezppp/index.html>

PPP 指南

(免费) <http://www.infomatch.com/~tbradley/go-ppp.txt>

Emacs 中的新闻组阅读器

(免费) <http://www.ifi.uio.no/~larsi/ding.html>

新闻组阅读器

(免费) <http://union.ncsa.uiuc.edu/HyperNews/get/hypernews.html>

client/server 信息系统

(免费) <http://www.virtualbase.com/inet1.htm>

抓网页的程序

(免费) [http://www.isg.de/visualweb/invisible\\_e.html](http://www.isg.de/visualweb/invisible_e.html)

支持线程的新闻组阅读器

(免费) <http://www.nada.kth.se/~su95-kjo/knews.html>

Mud 客户程序

(免费) <http://gramercy.ios.com/~tmtr01/mmucl>

MUD 服务器软件

(免费) <http://mordor.nazgul.com>

著名的 LPMUD 的 MudOS

(免费) <http://www.imaginary.com/LPC/MudOS>

X/Motif 下的 FTP 软件

(免费) <http://www.ajsoft.demon.co.uk/mxFtp.html>

\* 1/4 形新闻组阅读器

(免费) <http://www.general.uwa.edu.au/u/toivo/an-tk>

新闻组下载软件

(免费) <http://www.iquest.net/~jbuchana/computer.html>

PPP 计费软件

(免费) <http://www.isjm.com/tst/pppcosts.htm>

针对 SLIP 用户的 X 新闻组阅读器

(免费) <http://www.shore.net/~pip/qnews.htm>

基于 Ql 库的 IRC

(免费) <http://www.hitgate.com/qtirc>

可编程的 IRC 客户程序

(免费) <http://www.eleves.ens.fr:8080/home/espel/sirc.html>

字符界面 FTP 客户程序

(免费) <http://www.concentric.net/~mrsam/sftp/index.html>

图形离线新闻阅读

(免费) <http://web.inter.nl.net/users/R.W.Pijlman/skim/Skim.html>

基于 NNTP 的新闻组阅读器

(免费) <http://space.mit.edu/~davis/slrm.html>

图形 FTP/TELNET 软件

(免费) <http://jungfrau.ptf.hro.nl/~sprocket>

通过一个普通 UNIX 帐号使用 TCP/IP

(免费) <http://marketplace.com/tia/tiahome.html>

ircII 的图形前端

(免费) <http://home.pages.de/~tkirc>

USENET 到 FIDONET 转换器

(免费) <http://www.aquanet.co.il/shay/usepk1>

支持 Unicode 的终端

(免费) <http://www.cyberatl.net/~johnpaul/uterm.faq>

pppd/chat 的图形界面

(免费) <http://users.hol.gr/~dbouras>

简单的 X Window FTP 客户程序

(免费) <http://www.magg.net/~kaos/html/xmftp.html>

图形 IRC 客户程序

(免费) <http://catless.ncl.ac.uk/Programs/Zircon/README.html>

TCP/IP 音频通信

(免费) <http://www.ssc.com/~roland/work/ztalk.html>

## 二十四、传真电话软件

自动回话系统

(免费) <http://www.icce.rug.nl/docs/programs/callback/callback.html>

Fidonet 的 BBS

(免费) <http://www.vaala.fi/~hydra>

IBM 3270 终端模拟器

(免费) <http://www.sas.com/otherprods/emulus/Home.html>

业余电台软件

(免费) <http://www.acsys.com/radio/radio.html>

如何开发实时会议系统

(免费) <http://www.cpsc.ucalgary.ca/projects/grouplab/projects/groupkit>

FAX 软件

(商业) <http://www.unisrc.com>

Linux 下收发 FAX

(免费) <http://www.vix.com/hylafax>

Kermit

(免费) <http://www.columbia.edu/kermit>

可变 IP

(免费) <http://anchor.cs.binghamton.edu/~mobileip>

调制解调器软件

(免费) <http://adam.wins.uva.nl/~mdejonge/projects/modem>

同 APPLE 的 NEWTON 通信的软件

(免费) <http://www.tcel.com/~aehall/newtl>

BBS 系统

(免费) <http://tycho.com>

## 二十五、WWW 服务器和防火墙

AOL 的 www 服务器

(免费) <http://www.aolserver.com/server/index.html>

WWW 服务器 Apache 的老家

(免费) <http://www.apache.org>

小型 WWW 服务器

(免费) <http://www.boa.org>

## 防火墙 FWTK

(免费) <http://www.tis.com/docs/products/fwtk/index.html>

## MS Frontpage 服务器的 linux 扩展

(免费) <http://www.rtr.com/fpsupport/index.html>

## 配置 proxy 的软件

(免费) [http://www.isg.de/visualweb/invisible\\_e.html](http://www.isg.de/visualweb/invisible_e.html)

## IP 记帐和防火墙软件

(免费) <http://www.xos.nl/linux/ipfwadm>

## Linux 防火墙

(免费) <http://www.els.url.es/~s03786/masq.html>

## NCSA HTTPD

(免费) <http://hoohoo.ncsa.uiuc.edu>

## 面向对象的 Web 服务器

(免费) <http://www.roxen.com>

## 基于 Linux 的 Proxy 服务器

(免费) <http://squid.nlanr.net/Squid>

## SSL 的商业版 Apache

(商业) <http://stronghold.ukweb.com>

## 商业多媒体服务器

(商业) <http://www.vosaic.com/products/servers/MediaServer.html>

## SSL 的网络安全软件

(免费) <http://www.sealabs.com/products.html>

## 使 CERN HTTPD 变成 Proxy

(免费) <http://www.math.ucsb.edu/~boldt/noshit>

## Web 服务器记录文件分析软件

(免费) <http://www.rocksoft.com/webtune>

## WWW 服务器安全问题

(免费) <http://hopf.math.nwu.edu/docs/manual.html>

## CERN 的 WWW 服务器

(免费) <http://www.w3.org/pub/WWW/Daemon>

## 小型的 WWW 服务器

(免费) <http://www.stack.nl/~sven/xs-httpd>

## 商业 WWW 服务器

(商业) <http://www.zeus.co.uk/products/server>

## 二十六、游戏

### 纵横字谜

(免费) <http://www.litsoft.com/across/preview.htm>

### 射击游戏

- (免费) <http://evlweb.eecs.uic.edu/acj/AndyBattalion.html>
- 网络围棋
- (免费) <http://www.hevanet.com/wms/comp/cgoban>
- 多用户的图形 MUD
- (免费) <http://www.ifi.uio.no/~frankj/crossfire/index.html>
- DOOM 的常见问题
- (免费) <http://jcomm.uoregon.edu/~stevev/Linux-DOOM-FAQ.html>
- X Window 下的打坦克
- (免费) <http://www.atomic.com/~ogre/firepower.html>
- X Window 下的文明
- (免费) <http://www.daimi.aau.dk/~allan/freeciv.html>
- GNU 的国际象棋
- (免费) <http://nether.tky.hut.fi/glchess>
- Amiga 上的著名游戏
- (免费) <http://www.kuai.se/~sniemi/gwars.htm>
- 吃豆子
- (免费) <http://www.pharaoh0.demon.co.uk/hatman/index.html>
- 3D 射击游戏
- (免费) <http://www.tu-clausthal.de/student/iMaze>
- 冒险游戏编译器
- (免费) <http://www.cl.cam.ac.uk/users/gdr11/inform>
- 学韩语
- (免费) <http://www.bolthole.com:80/kdrill>
- 多用户的小球游戏
- (免费) <http://www.paru.cas.cz/~hubicka/koules/English/koules.html>
- 防守者
- (免费) <http://www.alp.cistron.nl/~wouters/linuxarc.html>
- 又一个文明
- (免费) <http://www.floot.demon.co.uk/lincity.html>
- X Window 下学外语
- (免费) <http://www.informatik.uni-trier.de/CIP/toeffer>
- 著名的单机 MUD
- (免费) <http://www.win.tue.nl/games/roguelike/nethack>
- Qt 界面的 nethack
- <http://www.uq.edu.au/~cswallis/nhqt/index.html>
- 扑克游戏
- (免费) <http://www4.ncsu.edu/eos/users/b/bwmott/www/cons00>
- 网络文明
- (免费) <http://www.aie.nl/~sreiz/openciv>

Quake

(免费) <http://threewave.planetquake.com/linux>

海战棋

(免费) <http://www.glue.umd.edu/~weave/vmwprod/seabattle.html>

扑克游戏

(免费) <http://marion.uio.no/~sverrehu/pub-unix/html/sol.shtml>

Linux 射击游戏

(免费) <http://huizen.dds.nl/~hosxe/spaceboom.html>

纵版射击游戏

(免费) <http://www.eplab.ph.ed.ac.uk/~p95137>

3D 俄罗斯方块

(免费) <http://www710.univ-lyon1.fr/ftp/xbl/xbl.html>

X Window 下的挖地雷

(免费) <http://www.gedanken.demon.co.uk/xbomb/index.html>

X Window 背景是个会转的大地球

(免费) <http://cag-www.lcs.mit.edu/~tuna/xearth>

Linux 下的大蜜蜂

(免费) <http://www.atomic.com/~ogre/xgal.html>

空间对决游戏

(免费) <http://www.es.uit.No/XPilot>

X Window 上开始下雪

(免费) [http://www.sara.nl/cgi-bin/rick\\_acc\\_xsnow](http://www.sara.nl/cgi-bin/rick_acc_xsnow)

## 二十七、图形图像处理

3D 建模软件

(免费) <http://www.comp.lancs.ac.uk/computing/users/andy/ac3dlinux.html>

3D 建模软件

(商业) <http://www.yonowat.com/Products/Amapi/Amapi.html>

X 下的小图标库

(免费) <http://bsd1.com/icons/ALcons>

3D 渲染软件

(免费) <http://www.ikm.com/9XXWZace/download.html>

Blue Moon 渲染软件

(免费) <http://www.seas.gwu.edu/student/gritz/bmrt.html>

交互式图形软件

(免费) <http://www.cs.cmu.edu/~spot/bomb.html>

Corel Draw Linux 版

(商业) <http://www.caldara.com/solutionsCD/Products/Corel/Info/draw35.htm>

VR 软件

(免费) <http://www.sics.se/dce/dive/dive.html>

做 GIF 动画的软件

(免费) <http://www.iis.ee.ethz.ch/~kiwi/GIFMerge>

Linux 下最好的图形编辑器

(免费) <http://www.gimp.org/>

X Window 下的地图软件

(免费) <http://www.oih.rwth-aachen.de/~paully/ica0.html>

X Window 下的图形软件

(免费) <http://www.wizards.dupont.com/cristy/ImageMagick.html>

矢量图形软件

(商业) <http://www.demon.co.uk/titan>

渲染, 动画, 模拟, VR 图形软件

(免费) <http://www.throedee.com/mhedron/mhedron.htm>

3D 建模和动画软件

(免费) <http://www.ptf.hro.nl/free-d>

光线跟踪软件

(免费) <http://www.povray.org>

3D 渲染和动画软件

(免费) <http://www.radix.net/~dunbar/index.html>

将 2D 图形转换为 AutoCAD 可读的文件

(免费) <http://www.comp.lancs.ac.uk/computing/users/aspinrr/ra-vec.html>

RIPscrip(tm)3.0 多媒体创作软件

(商业) <http://www.telegrafix.com/asr>

X Window 下的 3D 建模软件

(免费) <http://www.cyberus.ca/~denism/sceda/sceda.html>

2-D 绘图软件

(免费) <http://bourbon.cs.ucla.edu:8001/tgif>

X Window 的彩色编辑器

(免费) <http://www.atd.ucar.edu/homes/cook/xact/xact.html>

实时分形软件

(免费) <http://www.paru.cas.cz/~hubicka/XaoS>

光线跟踪软件 POV-Ray 的 X 界面

(免费) <http://cspar.uah.edu/~mallozzir/html/xfpovray.html>

X Window 的小画家

(免费) <http://www.danbbs.dk/~torsten/xpaint/index.html>

显示分形图形

(免费) <http://www.gweep.net/~jer/xtacy.html>

看图软件 Xview

(免费) <http://www.sun.com/sunsoft/catlink/xv/xv.html>

Linux 下的图形处理软件

(免费) <http://www.tummy.com/xvscan>

## 二十八、多媒体软件

VR 图像产生器支持 gif 和 jpeg 标准。

(免费) <http://www.msc.cornell.edu/~houle/vr/freedom>

分布式实时 MPEG 播放器

(免费) <http://ese.ogi.edu/DISC/projects/syntheticx/Player>

支持 UNIX/X11 的多媒体工具和 GUI 开发环境

(免费) <http://www.metacard.com>

支持 MP3 和声音文件

(免费) <http://adam.kaist.ac.kr/~jwj95>

支持 Linux 的实时 MPEG 播放器软件

(免费) <http://www.mpegiv.com>

Quicktime 的转换程序

(商业) <http://www.digigami.com/moviescreamer/about-unix.html>

多格式的自动播放器

(免费) <http://smurfland.cit.buffalo.edu/xanim/home.html>

## 二十九、音乐软件

AWE32 Linux 模式播放器

(免费) <http://xfactor.wpi.edu/private/witek/awe>

Linux 的音量调节器

(免费) <http://jpj.net/~trevor/aumix.html>

基于 X 的音量调节器

(免费) <http://synergy.caltech.edu/~john/programs.html>

多声道的录音机和混合器

(免费) <http://www.ceressoft.com/sstudio/index.html>

用于音乐合成及乐器设计的环境

(免费) <http://www.panix.com/~topper/Cmix/Cmix.html>

利用声卡做示波器

(免费) <http://www.afn.org/~cthugha>

面向对象的语音识别程序

(免费) <http://home.pages.de/~ears>

Emacs 的发音程序

(免费) <http://www.cs.cornell.edu/Info/People/raman/emacsspeak/emacsspeak.html>

多语种声音合成系统

(免费) <http://www.estr.ed.ac.uk/projects/festival/festival.html>

压缩声音文件

(免费) <http://www.miaif.ibp.fr/~tarreau/audio-comp>

MP3 解码器

(商业) <http://www.iis.fhg.de/departs/amm/layer3/index.html>

X Window 下的音量调节

(免费) <http://www.ee.mu.oz.au/staff/pbd/linux/fxvolume>

MIDI 实验系统

(免费) <http://www.nosuch.com/keykit>

X11 下的 MOD 编辑器

(免费) <http://www.cse.unsw.EDU.AU/~conradp/maube>

声音合成器

(免费) <http://tcts.fpms.ac.be/synthesis>

多声道数字声音和音乐系统

(免费) <http://kalahari.ton.tut.fi/s2/midas.html>

Linux 上的 mod 播放器

(免费) <http://www.stack.nl/~mikmak/mikmod.htm>

数字声音编辑处理器

(免费) <http://www.ccmrc.ucsb.edu/~doug/htmls/MiXViews.html>

支持 au, .wav, .mpeg 等多种格式的声音播放器

(免费) <http://www.powerweb.de/mpeg/unix.html>

MPEG 1.0/2.0 声音播放器

(免费) <http://www.sfs.nphil.uni-tuebingen.de/~hipp/mpg123.html>

Linux 上的 4 声道录音机

(免费) <http://rulhmpc38.leidenuniv.nl/private/multitrack/multitrack.html>

不带 DSP 播放 MOD, MTM 和 S3M 音乐文件

(免费) <http://www.jeton.or.jp/users/toru/www/orgsoft/index.html>

通用声音压缩功能

(免费) <http://web.mit.edu/afs/siph/user/xiphmont/OggSquish/html-pages>

使用 Q1 的混合器

(免费) <http://www.coli.uni-sb.de/~clanget/ossmixer.html>

在 Qt GUI Toolkit 中提供简单实用的声音

(免费) <http://student.uq.edu.au/~s002434/qt/QAudio>

实时收听网络上的声音文件

(免费) [http://www.real.com/products/player/download\\_player.html](http://www.real.com/products/player/download_player.html)

音乐编辑软件

(免费) <http://www.bath.ac.uk/~masjpf/rose.html>

在 Linux 上使用声卡的 Morse 格式程序

(免费) <http://livewire.ec.latrobe.edu.au/~sjm/morse/linux.html>

模拟 Commodore 64 的声卡界面

(免费) <http://www.geocities.com/SiliconValley/Lakes/5147/sidplay.html>

录制,回放,剪贴编辑声音文件

(免费) <http://www.eleceng.leeds.ac.uk/staff/een6njb/Software/Studio/screens.html>

声音文件转换器

(免费) <http://www.spies.com/Sox>

MIDI,WAVE 转换器

(免费) <http://www.elinet.fi/~toivonen/timidity>

基于 X 的 tcl/tk 使用 mpeg123 解码的播放器

(免费) <http://www.msc.cornell.edu/~bef2>

多线程的 mpeg 声音播放器

(免费) <http://www.iki.fi/i/tplay.html>

shell 环境下的 wav 播放器

(免费) <http://www.ica.net/pages/wwwg>

带 GUI 前端的 MPEG 播放器

(免费) <http://www.mpeg.org/xaudio>

在 X11 下运行的 CD 播放器

(免费) <http://www.geocities.com/SiliconValley/Pines/1029>

X Window 下的 CD 刻录软件

(免费) <http://www.fh-muenchen.de/rz/xcdroast>

带 X 界面的播放器,要求 GUS 卡和 USS/Lite 驱动器

(免费) <http://www.nyx.net/~arobinso/xgmod.html>

在 X 窗口系统下的使用 Motif 的 CD 播放器

(免费) <http://sunsite.unc.edu/~cddh/xmed>

在 X 窗口系统下的使用 Motif GUI 的混合器

(免费) <http://sunsite.unc.edu/~cddh/xmmix>

支持 GUS 或 AWE 声卡的播放器

(免费) <http://www.mordre.net/~claudio/xmp>

基于 Qt 的 mp3 播放器

(免费) <http://eray.tuug.utu.fi/~phoenix/xmp3player>

基于 X 的 MPEG 播放器和目录浏览器

(免费) <http://www.powerweb.de/mpeg/xmplay.html>

### 三十、电子会议系统

在 internet 上初始化和会议工具

(免费) [http://www.rvs.uni-hannover.de/products/confman/index\\_en.html](http://www.rvs.uni-hannover.de/products/confman/index_en.html)

internet 上的自动会议工具

(免费) <http://www.inria.fr/rodeo/fphone>

跨网络的会议工具

(免费) <http://www.merit.edu/research.and.development/mbone/index/linux.html>

电视会议客户对话协议

(免费) <http://www.pangea.org/~mavilar/qseeme/qseeme.html>

跨网络传输语音

(免费) <http://www.fourmilab.ch/speakfree/unix>

### 三十一、文本编辑器

简洁文本编辑器

(商业) <http://www.crisp.com>

桌面文本编辑系统

(商业) <http://www.zfc.nl>

Unix 标准编辑器 vi

(免费) <ftp://ftp.cs.pdx.edu/pub/elvis/README.html>

源文件输出行编辑器

(免费) <http://cantor.informatik.rwth-aachen.de/~michael/projects/fe.html>

程序员文本编辑器

(免费) <http://ixtas.fer.uni-lj.si/~markom/Re>

Unix 文本编辑器

(免费) <http://space.mit.edu/~davis/jed.html>

用 MGUI 编译的文本编辑器

(免费) [http://www.volfp.vol.it/IT/IT/ITALIANI/MORELLO/medit\\_e.htm](http://www.volfp.vol.it/IT/IT/ITALIANI/MORELLO/medit_e.htm)

用于 X/Motif 系统的 GUI 风格的普通文本编辑器

(免费) <http://www.pat.fnl.gov/nirvana/nedit.html>

GUI 文本编辑器

(免费) <http://www.cs.su.oz.au/~mik/red-manual/red-main-page.html>

独立平台的 GUI 程序员编辑器

(免费) <http://www.inter-soft.com/html/products/scriptum.html>

SEDIT 编辑器

(商业) <http://www.sedit.com/>

基本 X 文本编辑器

(免费) <http://www.cs.helsinki.fi/~wirzeniu/SeX>

可配置的程序员编辑器

(商业) <http://www.slickedit.com/seindex.htm>

全屏字符模式文本编辑器

(免费) <http://www.gu.edu.au/gwis/the/markh.html>

vi 增强编辑器

(免费) <http://www.math.fu-berlin.de/~guckes/vim>

在 Unix/X 环境中提供 Aeme 的编辑器

(免费) <http://www.cs.su.oz.au/~gary/wily>

用于 X-window 的高级文本编辑器

(免费) <http://Pux.hec.fr/xcoral>

X Window 下的 Emacs

(免费) <http://www.xemacs.org>

基于 X 文本编辑器

(免费) <http://www.proximity.com.au/~ben/xenon.html>

## 三十二、Java

Java 的一些图形类库

(免费) <http://www.biss-net.com/biss-awt.html>

带 JDK 库启动 Castanet

(免费) <http://www.blackdown.org/java-linux/Products.html>

带 JTAPl/CORBA/DCOM 的 Java RAD 工具

(免费) <http://www.cerc.wvu.edu/chicory/index.html>

免费,高质量的 Java IDE 开发环境

(免费) <http://www.techno-link.com/clients/ivo/FreeBuilder>

执行 Java(tm)代码的 Just In Time 虚拟机

(免费) <http://www.kaffe.org>

来自 Thought 的用于 Java 的便携式管理库

(免费) <http://www.thoughtinc.com/~nutmeg>

对象数据库管理系统

(商业) [http://www.poet.com/java\\_sdk\\_data.htm](http://www.poet.com/java_sdk_data.htm)

Java-to-C 翻译系统

(免费) <http://www.cs.arizona.edu/sumatra/toba>

为建立 Java 程序的一个完整的开发,应用环境

(商业) <http://www.visix.com>

## 三十三、编程软件

Fortran77 编译器

(商业) [http://www.absoft.com/f77\\_linux.html](http://www.absoft.com/f77_linux.html)

ANSI-85 COBOL 编译器

(商业) <http://www.acucobol.com/Products/ACUCOBOL-85>

用于 MCS-51(tm)家族的交叉汇编器

(免费) [http://www.ece.orst.edu/~paul/8051-goodies/index.html#as31\\_assembler](http://www.ece.orst.edu/~paul/8051-goodies/index.html#as31_assembler)

数据重新格式化的编程语言

(免费) [http://csugrad.cs.vt.edu/manuals/gawk/gawk\\_toc.html](http://csugrad.cs.vt.edu/manuals/gawk/gawk_toc.html)

Unix 下的 Basic

(免费) <http://www.uni-mainz.de/~ihm/basic.html>

面向对象编程软件

(免费) <http://www.mjolner.dk>

提供 DSSSL 支持和唯一编码加密的编译器

(免费) <http://cuiwww.unige.ch/~serrano/bigloo.html>

带编译器的 lisp

(免费) <http://www.mv.com/users/pw/lisp>

在 Linux 上使用的 COBOL 解释器

(免费) <http://www.deskware.com/cobol/cobol.htm>

高级面向对象程序设计语言

(免费) <http://arachnid.cs.cf.ac.uk/CLE>

嵌入式 perl 解释器

(免费) <http://www.engelschall.com/sw/epperl>

符合 ISO 和 ANSI 标准的 Fortran 90 编译器

(商业) <http://www.nag.co.uk/nagware/Fortran90.html>

Pascal 编译器

(免费) <http://www.brain.uni-freiburg.de/~klaus/pascal/fpk-pas>

GNU c++ 编译器

(免费) <http://www.cis.ohio-state.edu/htbin/info/info/g++.info>

GNU C 编译器

(免费) <http://www.cis.ohio-state.edu/htbin/info/info/gcc.info>

类似面向对象, C++ 语言的解释器

(免费) <http://www.zurich.ibm.com/~bba/gomscript.html>

linux 脚本语言

(免费) [http://www.knotwork.com/\(nobg\)/info/comp/freeware/linux/gorby](http://www.knotwork.com/(nobg)/info/comp/freeware/linux/gorby)

由 R4RS 和 IEEE 定义的独立补充方案

(商业) <http://www.munich.net/inlab/scheme>

用于 linux 的 OBeron 编译器

(免费) <http://www.cs.tu-berlin.de/~sepp/jacob.html>

包括 C++ 类库, 支持成员模板

(商业) [http://www.kai.com/C\\_plus\\_plus](http://www.kai.com/C_plus_plus)

quasi-C 解释器可在 X windows 下写 GUI

(免费) <http://spock.ece.drexel.edu/marx>

用户指南

(免费) <http://www.fit.gut.edu.au/CompSci/PLAS/GPM/guide>

Modula-3 的移植

(免费) <http://www.cl.cam.ac.uk/m3doc/linux/m3.html>

类似 VB 的编程语言

(商业) <http://softwork-3.isdn.mcs.net/nbasic.html>

perl 语言

(免费) <http://www.cis.ufl.edu/perl>

优化/并行 ANSI C 编译器

- (商业) [http://www.pgroup.com/ppro\\_cc\\_desc.html](http://www.pgroup.com/ppro_cc_desc.html)  
优化/并行 ANSI C++ 编译器
- (商业) [http://www.pgroup.com/ppro\\_cplus\\_desc.html](http://www.pgroup.com/ppro_cplus_desc.html)  
优化/并行 FORTRAN 77 编译器
- (商业) [http://www.pgroup.com/ppro\\_f77\\_desc.html](http://www.pgroup.com/ppro_f77_desc.html)  
Fortran 编译器
- (商业) [http://www.pgroup.com/ppro\\_hpf\\_desc.html](http://www.pgroup.com/ppro_hpf_desc.html)  
面向对象的书写语言
- (免费) <http://www.python.org>  
设计成简单、高效、灵活性、开放的一个面向对象的语言
- (免费) <http://www.icsi.berkeley.edu/~sather>  
一个高速的功能语言
- (免费) <http://www.llnl.gov/sisal/SisalHomePage.html>  
在 Linux 下执行的 Smalltalk-80
- (免费) <http://www.create.ucsb.edu/squeak>  
IBM's REXX 语言在 Linux 下的移植版本
- (商业) <http://www.scdit.com>  
来自 System Essentials Limited 的编译器
- (免费) <http://www.nai.net/~lerman>  
开发 370/390 汇编程序
- (免费) <http://www.sni.net/tachyon>  
脚本语言和图形用户接口工具
- (免费) <http://sunscript.sun.com>  
32 位 BASIC 编译器和开发环境
- (商业) <http://www.basmark.com>  
类似 C 语言的向量语言
- (免费) <http://zaphod.uni-muenster.de/80/xi>

### 三十四、开发工具

- C++ 可移植用户接口开发环境
- (免费) <http://www.cs.cmu.edu/afs/cs.cmu.edu/project/amulet/www>  
跨平台应用开发
- (免费) <http://www.angoss.com/sw/home.htm>  
用 Tel 写的 make
- (免费) <http://www.iitb.fhg.de/~kir/Bras/brasHome.html>  
并行程序设计库
- (免费) <http://www.bsp-worldwide.org/implmnts/tool06b.htm>  
测试 C/C++ 源码的程序
- (免费) <http://www.fste.ac.cowan.edu.au/~tlittlef>

## 编程环境

(免费) <http://iel.ucdavis.edu/CH>

## C/C++ 代码开发环境

(免费) <http://www.cco.caltech.edu/~jaf/jx/progs.html>

## 对 C/C++ 程序进行代码分析

(免费) <http://www.parasoft.com/wizard/index.html>

## Pascal 到 C/C++ 转换器

(免费) <http://www.ras.ru/moshkow/html-KOI/SOFTWARE>

## 在 C 源码和头文件中建立索引

(免费) <http://fly.hiwaay.net/~darren/ctags.html>

## Perl 5 程序, 将 cvs log 转换成 HTML 文件

(免费) <http://eivind.imm.dtu.dk/cvs2html>

## 从 C 语言产生文本文件 (LaTeX or HTML)

(免费) <http://www.gedanken.demon.co.uk/exref/index.html>

## 图形接口的 GDB 和 DBX

(免费) <http://www.cs.tu-bs.de/softech/ddd>

## X/Motif 应用程序开发

(免费) <http://www.in-tech.com>

## 编写在多台机器上同时运行的程序

(免费) <http://wallybox.csi.net/dipc>

## 基于 QT 的 GUI 对话框编辑器

(免费) <http://www.cis.ohio-state.edu/~jharris/dlgedit>

## 把 C++ 程序转换成 HTML 和 LaTeX 文本的软件

(免费) <http://www.zib.de/Visual/software/doc++/index.html>

## 面向对象到 C 的扩展

(免费) <http://www.edge.net/algorithms>

## 软件开发者的许可证服务器

(免费) <http://www.unicon-ka.de/ifelics.htm>

## 支持 RPC 和 RMI 的接口定义语言编译器

(免费) <http://www.cs.utah.edu/projects/flux/flick>

## fortran 77 工具集合

(免费) <http://www.xs4all.nl/~ripley/RSD/ForUtil.html>

## 基于 X 的应用程序开发系统

(免费) <http://www.agetek.com/genx.htm>

## 程序开发和用法

(免费) <http://www.inter-soft.com/products/ideafix>

## 网络操作系统和程序设计环境

(免费) <http://inferno.lucent.com>

## C/C++ 程序错误检测

(免费) <http://www.parasoft.com/insure++/index.html>

C++ 程序流程图编辑器

(免费) <http://www.vectaport.com/ivtools>

一个类似于 lint 的 C 编辑器

(免费) <http://larch-www.lcs.mit.edu:8001/larch/lclint.html>

程序员工具和编码生成器

(免费) <http://www.imatix.com/html/lrindex.htm>

检测 C 源码, 报告可疑结构

(免费) <http://www.thinkage.on.ca/shareware/lint>

编写 X/Motif 交互图形用户接口

(免费) <http://www.umn.edu/nlhome/g257/park005/lxb/lxb.html>

查找 C 语言内存漏洞

(免费) <http://www.iquest.net/~jbuchana/computer.html>

快速 CUI 设计器

(免费) <http://www.informatik.tu-muenchen.de/~paysan/bigforth.html>

使用 wxWindows 的 GUI 开发环境

(免费) <http://www.cs.rice.edu/~mflatt/mred.html>

C 语法检查

(免费) <http://www.cs.ntu.edu.au/homepages/pjm/nana-home>

将 C++ 程序自动转化为 HTML 文本

(免费) <http://www.obsoft.com/Product/ObjMan.html>

2D, 3D 图形程序开发环境

(免费) <http://www.metrolink.com/products/opengl/index.html>

在开发中验证内存漏洞

(免费) <http://home.earthlink.net/~owenomalley/plumber.html>

多平台 RPC 开发工具

(商业) <http://www.netbula.com/products/powerrpc>

软件项目版本控制系统

(免费) <http://www.XCF.Berkeley.EDU/~jmacd/prcs.html>

Python 编辑器/解释器

(免费) <http://starship.skyport.net/crew/zack/ptui>

为 GUI 程序建立的 python 和 Xforms 的结合

(免费) <http://ultra7.unl.edu.ar:80/pyxforms>

客户-服务器和分布式应用的开发环境

(商业) <http://www.emass.com/reactor>

C 和 C++ 源码分析工具

(商业) <http://207.92.81.101/rsm.htm>

Schema 工具

(免费) <http://www.rgl.polymtl.ca/schema/tool.html>

C, C + +, Java, Fortran, CORBA IDL 的编辑环境

(商业) <http://www.takefive.com>

独立的 Eiffel 程序设计环境和编译器

(免费) <http://swt-www.informatik.uni-hamburg.de/~1willamo/dl.html>

图形界面的 Tcl/tk

(免费) <http://www.sunlabs.com/research/tcl/spectcl>

R4RS Scheme 解释器

(免费) <http://kaolin.unice.fr/html/STk.html>

C/C + + 的快速开发工具

(免费) <http://www.cs.utah.edu/~beazley/SWIG>

GUI 的 Fortran 程序设计工作平台

(免费) <http://perso.club-internet.fr/queutey/TkFPW/index.html>

类似 xcleardiff 的程序

(免费) <http://www.doitnow.com/~quillan/john/tkxcd>

面向对象的软件开发系统

(免费) <http://arachnid.cs.cf.ac.uk/Fower/twrweb/LinuxProducts.html>

跨平台开发和应用 Windows API 程序

(商业) <http://www.willows.com>

用于 UNIX 的基于 Tcl/tk 开发环境

(免费) <http://www.neuron.com/stewart/vtcl>

如何在 X Window 和 Windows 之间移植 C + + GUI 界面

(免费) <http://www.cs.unm.edu/~wampler/vgui/vgui.html>

在 UNIX 上运行 VB 程序

(商业) <http://www.vbix.com>

能产生源代码的 Motif 界面设计工具

(商业) <http://www.bredex.de/EN/vdx>

3D 计算机图形软件

(免费) <http://www.cs.rpi.edu/~martink>

快速开发图形界面

(免费) <http://www.eit.com/software/winterp/winterp.html>

C + +/Java 开发系统

(免费) <http://www.softwarebuero.de/wipeout-eng.html>

编写可移植到 windows 平台上的 python 程序

(免费) <http://bigbear.pc.helsinki.fi/harri/wxpython>

如何开发多平台应用程序

(免费) <http://web.ukonline.co.uk/julian.smart/wxwin>

一个 X Windows 上的图形界面库

(免费) <http://bragg.phys.uwm.edu/xforms>

基于 Modula-2 和 Oberon-2

(免费) <http://www.iis.nsk.su/xtex/xds>

编写字符界面的软件

(免费) <http://www.interbit.ro/~dragos/Xterminal.html>

输入源代码的编辑工具

(免费) [http://WWW.FH-Karlsruhe.de/~somi0011/s\\_soft.html/s\\_soft.html](http://WWW.FH-Karlsruhe.de/~somi0011/s_soft.html/s_soft.html)

类似于 Borland C 的 IDE 开发环境

(免费) <http://www.rpi.edu/~payned/xwpe>

可配置的编程环境

(免费) <http://www.iki.fi/~terop/yade.html>

### 三十五、编程工具库

面向对象的 C/C++ 网络开发工具

(免费) <http://siesta.cs.wustl.edu/~schmidt/ACE.html>

使用 Burrows-Wheeler 的压缩工具

(免费) <http://www.cs.man.ac.uk/arch/people/j-seward/index.html>

编写 CGI 的 C 语言库

(免费) <http://www.eckim.com/software/cgihtml>

画图工具

(免费) <http://www.int.com/linux.html>

并行计算

(免费) <http://www.felinia.com/cpt>

用来显示数据示意图的程序库

(免费) <http://www.mpae.gwdg.de/dislin/dislin.html>

一个类似于 Motif 和 OpenGL 的 widgets 库

(免费) <http://www.ma.utexas.edu/~mzou/EZWGL>

Linux 标准图形界面

(免费) <http://synergy.caltech.edu/~ggi>

OpenGL 图形标准 API

(免费) <http://linas.org/gle/index.html>

用来开发数据库的 C++ 类库

(免费) <http://www.gnustep.org>

浮点运算库

(免费) <http://www.spektracom.de/~arndt/joerg.html#hfloat>

X Window 编程类库

(免费) <http://www.cco.caltech.edu/~jafl/jx>

普通编程类库

(免费) <http://www.atnf.atnf.csiro.au/karma>

处理配置文件的库

(免费) <http://jade.cs.uct.ac.za/kunf>

线性代数 FORTRAN 库

(免费) <http://nils.wustl.edu/schiotz/lapack-linux.html>

类似于 Motif 的免费软件

(免费) <http://www.lesstif.org>

高性能 3D 图形库

(免费) <http://www.ozemail.com.au/~keithw>

GNU C 库

(免费) <http://www.imaxx.net/~thrytis/glibc>

Linux 下的线程编程

(免费) <http://pauillac.inria.fr/~xleroy/linuxthreads>

实时数据压缩/解压软件

(免费) <http://www.infosys.tuwien.ac.at/Staff/lux/marco/lzo.html>

Matlab 兼容的矩阵运算 C++ 类库

(免费) <http://www.mathtools.com>

类似于 OPENG 的 3D 图形库

(免费) <http://www.ssec.wisc.edu/~brianp/Mesa.html>

多媒体开发库

(免费) <http://www.ifi.unizh.ch/groups/mml/projects/met++/met++.html>

编译免费的 Netscape

(免费) [http://www.neurondata.com/Download/mwl\\_dl.html](http://www.neurondata.com/Download/mwl_dl.html)

处理 MIME 的 C++ 类库

(免费) <http://www.fwb.gulf.net/~dwsauder/mimepp.html>

跨平台的图形界面设计软件

(免费) [http://www.volftp.vol.it/IT/ITALIANI/MORELLO/index\\_e.htm](http://www.volftp.vol.it/IT/ITALIANI/MORELLO/index_e.htm)

商业 Motif 库

(商业) <http://www.metrolink.com/products/motif/index.html>

并行计算消息传递接口

(免费) <http://www.mcs.anl.gov/mpi/index.html>

一个类似于 NeXT 的 Widgets

(免费) <http://www.inf.ufrgs.br/~kojima/nextaw/index.html>

开发 Sybase 客户程序

(免费) <http://www.sybase.com/products/samples/linux>

开发基于 POSIX.1c 的多线程程序

(免费) <http://www.aa.net/~mtp/PCthreads.html>

X Window 的绘图工具

(免费) <http://www.shore.net/~pip/qplot.htm>

将网络上一群计算机变成一台计算机

(免费) <http://www.eecg.toronto.edu/~colohan/quarks.html>

提供跨 Win95, NT, UNIX/X 的图形界面编程 API

(免费) <http://www.troll.no/>

OSF/Motif 2.0 开发系统

(商业) <http://www.redhat.com/products/rhm.html>

mod/s3m/mtm/xm 声卡编程 API

(免费) <http://www.egerter.com/seal>

开发面向对象的数据文件

(免费) <http://www.cs.uwo.edu/~haines/research/smart>

基于 Qt (C++, X11) 的移动图像

(免费) <http://student.uq.edu.au/~s002434/qt/SpriteField/doc/html>

跨平台的字符界面开发工具

(免费) <http://www.schwaben.de/home/uz/spunk.eng.html>

基于 C++ 跨平台的界面开发工具

(免费) <http://wild.ch/SpunkCross>

供 g++ 使用的 HP 版本 STL

(免费) <http://www.cs.rpi.edu/~musser/stl.html>

GUI 公司的 OSF/Motif 2.0

(免费) <http://www.lasermoon.co.uk/catalog/guis/swim.htm>

将 adobe 公司的 type 1 字库转换为 bitmap

(免费) <http://www.neuroinformatik.ruhr-uni-bochum.de/ini/PEOPLE/rmz/t1lib.html>

Tcl/Tk 的图形扩展

(免费) <http://www.xpi.com/tix>

基于 Motif 的 MS Windows API

(免费) <http://www.uno.com/faq.html>

一个面向 Motif 和 Xnext 的 C++ Wrapper 库

(免费) <http://soback.kornet.nm.kr/~glgay>

提供类似于 win95 界面的 widgets

(免费) <http://logic.csc.cuhk.edu.hk/~s947547/Xaw95/Xaw95.html>

Xaw 的新版本

(免费) <http://www.zip.com.au/~bb/XawXpm>

Perl 中使用 xforms 库

(免费) <http://SAL.KachinaTech.COM/F/5/XFORMS4PERL.html>

Motif 的编程库

(商业) <http://www.klg.com/products/index.html>

一个 C++ 类库

(免费) <http://www.cs.sc.edu/~sridhar/yacl.html>

在 X Window 下模拟 SGI 的 GL

(免费) <http://WWW.thp.Uni-Duisburg.DE/Ygl/ReadMe.html>

允许程序在执行时解压数据文件

(免费) <http://linux.wanug.org/pub/knaff/zlibc>

# D

## 本书所附光盘介绍

书后所附的光盘是《Linux 实用大全》的配套光盘,本光盘包括下列软件:

(1) TurboLinux 简体中文版 3.0.2 完全收录

(2) 最新的系统内核:2.0.36, 2.2.1

(3) 大量应用软件, 直接光盘 RPM 方式安装, 包括:

窗口管理器: KDE, AfterStep, Enlightenment, WindowMaker, Qvwm

图形多媒体: Gimp, X11amp, Xanim, Mtv, Acrobat Reader, Mesa, Glide

网络: Ezppp, Apache, Squid, Firebird BBS, Socks, Scotty

模拟器: Dosemu, Wine, iBCS, Snes,Bochs

数据库: MySQL, Postgresql

娱乐软件: MUD, DOOM, Quake2(Source Code), QKMJ, OKbrige

编程工具: Gcc, Perl, JDK, Lesstif, EZWGL, Xwpe, Source Navigator

丰富的中文软件: Chinput, XA, Xeingb, Crxvt, Color CXterm, WZCF,

中文 PostScript 字库, 中文 LaTeX, Chpower, Cdict, 中文 Fvwm

最新中文文档: CLDP 计划: WWW.linux.org.tw 和 CLDP.ml.org 的完全镜像, 全为可以在线浏览的 Html 格式

著名的 Linux 七本 GUIDE, Jargon File, 中文版 Unix 入门手册

如果用户在 Windows 95 下要浏览这张光盘, 可以观看作者制作的多媒体光盘介绍程序。在浏览这个程序时, 用户可以用鼠标点击屏幕右下方的控制条来控制浏览过程。在 Linux 下浏览这张光盘需要用 mount 命令将其装载到系统的某一目录下才行。

这张光盘分为下面几个目录: APMKernel, SMPKernel, TurboLinux, bin, customlists, docs, dosutils, images, unsupported, win32utl, Chinese, Document, others。其中 APMKernel, SMPKernel, TurboLinux, bin, customlists, docs, dosutils, images, unsupported, win32utl 目录是 TurboLinux 3.0.2 的安装目录。这里 APMKernel 和 SMPKernel 目录中分别存放的是含高级电源管理和多处理器支持的内核, unsupported 目录中是 2.2.1 版本的内核, docs 目录是 TurboLinux 的中文文档。而 Chinese 目录下存放的是作者精心制作的 Linux 下的中文软件, 用户既可以直接安装, 也可以安装源代码后自行编译。Document 目录下存放的是作者收集的中文 Linux 文档资料, 用户可以用 Netscape 直接在线浏览。others 目录下存放的是作者收集的一些其他软件, 包括一部分中文软件(chinese 目录), 而 emulator 目录下收录了 Apple 和超级任天堂的模拟器, 3dfx 的驱动程序 glide。multimedia 目录下存放的是目前最热门的在 linux 下播放 mp3 和 vcd 的软件 X11amp 和 mtv, 以及著名的 RealAudio。programing 目录下收集了最好的源代码管理软件 source navigator 和最新的浏览器 qtscape。misc 目录下存放了一些小应用软件。

下面特别值得推荐的是 doslinux 目录下的一个小小 Linux 系统。这是笔者制作的一个使用 Dos 分区的 Linux 系统,包括基本的 Linux 系统软件,网络软件和 Xwindow,还有 WZCE 和 CXterm。安装这个 Linux 的步骤非常简单,只需要在用户的 C 盘上运行 doslinux 这个自解压程序,就会在 C 盘上自动生成一个 Linux 目录,用户进入这个目录后运行 linux.bat 批处理文件,就能启动一个 Linux 系统,用户再用 root 登录,就可以进入一个麻雀虽小,但五脏俱全的小小 Linux 环境了,用户只需要配置一下网络设置就可以上网,也可以运行 Xwindow、WZCE、CXterm、Lynx 和 mc 等软件。卸载这个 Linux 系统只需要在 dos 下删除 c:\linux 这个目录。这个系统对于那些想尝试 Linux 又不想对硬盘大动干戈的用户来说是再好不过的试用版。这个 doslinux 的最低系统配置是 386、4M 内存、80M 硬盘,另外 doslinux 目录下还收录了网友 Lenx/raner 制作的 speboot、spcglove 和 tinylinux 这些微型 Linux 系统,用户可以自行阅读相应的 readme 安装。games 目录下是作者收录的一些 Linux 下的游戏。包括 quake、quake2 和 QKMJ 等。

下面作者就简单说明一下光盘上软件的安装方式,首先是安装 TurboLinux 中文版系统,如果用户机器上的主板支持光盘启动的话就可以在 CMOS 中打开用 cdrom 启动的选项,直接用该光盘启动安装 TurboLinux,如果用户主板不支持光盘启动,那么用户也可以按照本书中的步骤自己制作启动盘来安装。

**注意:**本光盘未包含 TurboLinux 要用到的中文 TrueType 字库。用户在安装完 TurboLinux 后,应参考光盘上的 Fonts.txt 文件中的说明安装中文字库,否则在 X 窗口系统下某些汉字可能不能正确显示。

TurboLinux 中文版中已经包含了光盘 Chinese/目录下的部分软件,这个目录中也有些没有放进 TurboLinux 的中文软件包。这些软件包都是用 RPM 制作的,作者已经将各个软件精心配置好了,用户可以免去自行安装软件时繁琐的编译工作,以安装 xcingb-1.0 这个软件包为例:只需要用 `rpm -i xcingb-1.0-1.rpm` 就可以轻轻松松安装相应的可执行的软件包,如果要查看这个软件包的介绍,就用 `rpm -qpl xcingb-1.0-1.rpm` 来查询这个软件包的作用和介绍,如果想查询系统是否已经安装了这个软件包就用 `rpm -qa` 来查看系统当前已经安装的所有 rpm 软件包。卸载软件包用的是 `rpm -e xcingb-1.0`,其他软件包依此类推。需要提醒用户的是,Chinese 目录下 RPMS 目录下存放的是可以直接安装的中文软件的可执行文件,而 SRPMS 目录下存放的是相应软件的源代码包,所以安装软件的话就请进入 RPMS 目录安装相应的 rpm 包,如果要安装源代码就请到 SRPMS 目录下安装相应的 xxx.src.rpm,安装后的源代码的缺省目录是在系统的 /usr/src/redhat/SOURCES 目录下。

以上就是这张光盘的使用简介,欢迎你使用这张 Linux 光盘!

如有问题,请洽:

book@turbolinux.com.cn

请在邮件标题上注明书名。

下面是光盘上收录的中文软件简介,其中标明“\*”的已含于 TurboLinux 中文版: xcingb-1.0

中文 rxvt 和类似于中文之星的输入界面

结合 XA 可以在任意的 X 窗口中输入中文

**hc - 1.0**

内码转换工具,可以在 big5 - gb - hz 之间转换内码

**FirebirdBBS - gb - 2.6**

火鸟 BBS 的最新版本,直接安装,不过安装后用

户还是要做一些设置的,首先以 bbs 登录(无密码)

然后登录 SYSOP 帐号,切记。

**ESII - 1.0**

这是著名的中文 LPMUD 东方故事 II,安装在 /usr/local/games/mud 下

请仔细阅读安装后的使用说明,启动 MUD 命令如下

```
cd /usr/local/games/mud/bin
```

```
startmud ES2 &
```

**edict - 1.0**

英/中字典

**chdrv - 0.23.gb**

文本模式的中文终端,支持 GB 内码

**chinput - 1.4.linux**

由于明俭先生编写的中文集成软件包

包括自己的输入工具、屏幕抓词、自动汉化和中文 EZWGL

**cxterm - 5.0.p3**

彩色中文 CXterm,安装有新的拼音输入法

**fvwm95 - 2.0.43a**

中文 fvwm95

**hztty - 2.0**

多内码终端转换过滤器,可以在 GB 终端上阅读 BIG5 和 HZ 码

**jvim - 2.0r**

可以编辑中文的 Vi

**tcsh - 6.07**

支持中文的 tcsh

**xa - 1.04**

利用 wrap 原理自动汉化 X window 下的英文软件

并且可以在 netscape 中输入汉字

**xemacs - 20.4**

支持中文的 xemacs

**xfonts - chinese**

中文 X 字库

[General Information]

书名=Linux 实用大全（第二版）

作者=陈向阳 方汉

页数=630

SS号=10028824

出版日期=1998年8月第1版