



Linux 系统管理专家系列

# Red Hat Linux 9

## 系统管理员完全学习手册



北京希望电子出版社 总策划  
姜中华 师鸣若 王大印 编 著



科学出版社  
[www.sciencep.com](http://www.sciencep.com)



Linux 系统管理专家系列

# Red Hat Linux 9

## 系统管理员完全学习手册



北京希望电子出版社 总策划  
姜中华 师鸣若 王大印 编 著



科学出版社  
[www.sciencep.com](http://www.sciencep.com)

责任编辑：刘海芳  
封面设计：梁运丽

# Red Hat Linux 9

系统管理员完全学习手册

## Linux 系统管理专家系列

《Linux 服务器安全手册》

《Linux 系统管理员手册》

《Linux 服务器配置手册》

《Red Hat Linux 9 系统管理员完全学习手册》

## 本书特点

- 结构清晰
- 内容实用
- 使有一定经验的用户迅速提高应用能力

## 本书适用于

- Linux 初学者
- 大专院校学生
- 广大 Linux 爱好者

需要本书或技术支持的读者，请与北京中关村 083 信箱（邮编 100080）发行部联系

市场部电话：010-82702660, 62978181

技术支持电话：010-62978181 转 528

010-62978181 转 516 (综合编室)

传 真：010-82702696

E-mail: tbd@bhp.com.cn

投 稿: textbook@bhp.com.cn

网 址: www.bhp.com.cn

ISBN 7-03-015300-6



9 787030 153005 >

ISBN 7-03-015300-6

定价：36.00 元






Linux 系统管理专家系列

# Red Hat Linux 9

## 系统管理员完全学习手册



北京希望电子出版社 总策划  
姜中华 师鸣若 王大印 编 著

 科学出版社  
[www.sciencep.com](http://www.sciencep.com)



## 内 容 简 介

Linux 是多用户分时操作系统, 是自由软件, 具有 Unix 的全部功能, 得到多数硬件厂商的支持, 而且 Linux 操作系统本身附带了许多有用的应用软件。在众多 Linux 发行版本中, Red Hat Linux 是最常用、最容易使用的版本之一。

全书讲述了 Red Hat Linux 9 的安装、Linux 的基础命令(文件、shell、网络配置和软件安装)、桌面应用知识(GNOME、KDE 和常用应用软件)和企业应用(各种服务器的配置和使用)方面的内容。本书始终以实用为原则, 通过丰富的实例说明问题, 既能使新手轻松入门, 又能使有一定经验的用户迅速提高应用能力。

本书适合 Linux 初学者、大专院校学生以及广大 Linux 爱好者使用。

需要本书或需要得到技术支持的读者, 请与北京中关村 083 信箱(邮编: 100080)发行部联系, 电话: 010-82702660, 010-82702658, 010-62978181 转 103 或 238, 传真: 010-82702698, E-mail: tbd@bhp.com.cn。

### 图书在版编目(CIP)数据

Red Hat Linux 9 系统管理员完全学习手册/姜中华, 师鸣若, 王人印编著. —北京: 科学出版社, 2005.7

(Linux 系统管理专家)

ISBN 7-03-015300-6

I. L… II. ①姜…②师…③王… III. Linux 操作系统—技术手册 IV. TP316.89-62

中国版本图书馆 CIP 数据核字(2005)第 026897 号

责任编辑: 刘海芳 / 责任校对: 叶 子

责任印刷: 东 升 / 封面设计: 梁运丽

科 学 出 版 社 出 版

·北京东黄根大街 16 号

邮政编码: 100017

<http://www.sciencep.com>

北京市东升印刷厂印刷

科学出版社发行 各地新华书店经销

\*

2005 年 7 月第 一 版 开本: 787×1092 1/16

2005 年 7 月第一次印刷 印张: 27 1/2

印数: 1—5 000 字数: 630 000

定价: 36.00 元

## 前 言

一般用户可能会认为 Linux 与 UNIX 一样是专家操作系统，只有少数计算机精英能够使用。这种想法产生的原因有二，其一是 Linux 发展太快，许多新功能在 Linux 的老版本中没有，其二是许多 Linux 图书的读者对象是有经验的用户，作者编写时注重系统的原理而不注重实用，从而使用户产生畏惧之感。

本书介绍了 Red Hat Linux 最新版——Red Hat Linux 9。该版本在操作系统内核、图形界面、系统与网络管理方面都比其早期版本功能强大，并且有了很大的突破，对中文的支持方面更加突出。它容易使用，安装方便，许多方面与 Windows 操作系统类似。

全书讲述了 Red Hat Linux 9 的安装、Linux 的基础命令（文件、shell、网络配置和软件安装）、桌面应用知识（GNOME、KDE 和常用应用软件）和企业应用（各种服务器的配置和使用）方面的内容。

本书使用的约定如下：

（1）Linux 命令前加入了提示符，如#或者\$（#表示超级用户执行的命令，\$表示普通用户执行的命令）。本书以实例为基础，采用了 Linux 完整的提示符（如[`globus@wang globus`]\$）。在用户自己安装的 Linux 系统中，提示符可能与本书示例的提示符不同，这不会影响用户的使用。

（2）井号（#）也表示配置文件和 shell 脚本的说明语句，应根据上下文区分具体含义。

（3）加号（+）表示使用组合键，如 Ctrl+C。

（4）在介绍 Linux GUI 时，使用“|”表示菜单或者子菜单选项。例如，“主菜单”|“系统设置”|“网络”表示先单击 KDE（或者 GNOME）的“主菜单”，然后单击“系统设置”子菜单，再单击“系统设置”子菜单下的“网络”菜单项。

（5）有关说明、提示和警告都采用“注意”形式表示。

本书的读者对象为初、中级桌面用户和企业用户。本书透彻地讲述了 Red Hat Linux 9 的常用功能及系统管理方面的内容，既能使新手快速入门和进阶，又能使有一定经验的用户迅速提高应用能力。

本书得到以下项目资助：

国家高技术研究发展计划（863）（项目编号：2003AA144030）

国家自然科学基金（项目编号：90204016）

国家 973 项目（项目编号：2004CB318004）

本书主要由姜中华、师鸣若、王火印共同编著。另外参与编著、整理和排版的人员还有徐津、刘在强、袁磊、闻金川、张增华、宝力高、赵树林、程斌、徐志飞、靳梅、程斌、钟仕增、杨宁、付磊、王飞、马喜、王霞、王利明等。在本书的编写过程中得到了信息安全国家重点实验室的林东岱研究员和西安理工大学计算机学院的崔杜武教授的大力帮助。此外，本书的出版还得益于中国科学院信息安全国家重点实验室的各位同志的大力协助。在此一并表示感谢。写作过程中作者尽了最大努力，希望将多年使用 Linux 的经验奉献给广大读者。

作 者

# 目 录

第1章 安装 Red Hat Linux 9.....	1
1.1 Linux 操作系统简介.....	1
1.1.1 Linux 的历史.....	1
1.1.2 Linux 操作系统的特点.....	1
1.1.3 Linux 操作系统的应用领域.....	3
1.1.4 常见的 Linux 发行版.....	3
1.2 准备安装 Red Hat Linux 9.....	6
1.2.1 选择硬件平台.....	6
1.2.2 Red Hat 的版本.....	6
1.2.3 检查 Red Hat Linux 9 的硬件需求.....	6
1.2.4 准备安装 Red Hat Linux 9.....	7
1.3 安装 Red Hat Linux 9.....	9
1.3.1 选择安装模式.....	9
1.3.2 测试安装介质.....	10
1.3.3 选择语种、键盘和安装途径.....	11
1.3.4 选择安装类型.....	12
1.3.5 磁盘分区.....	13
1.3.6 配置引导装载程序.....	16
1.3.7 配置 TCP/IP 网络.....	20
1.3.8 设置主机名.....	21
1.3.9 防火墙配置.....	21
1.3.10 语言支持的选择.....	24
1.3.11 时区配置.....	24
1.3.12 设置根口令.....	25
1.3.13 验证配置.....	26
1.3.14 软件包组的选择.....	28
1.3.15 准备安装.....	30
1.3.16 安装软件包.....	31
1.3.17 创建引导盘.....	31
1.3.18 显卡配置.....	32
1.3.19 显示器和定制.....	33
1.3.20 安装完成.....	34
1.4 LILO 和 GRUB 配置.....	34

1.4.1 LILO 的配置实例.....	34
1.4.2 LILO 的常用参数.....	36
1.4.3 使用 LILO 时遇到的问题及对策.....	37
1.4.4 GRUB 的配置行.....	38
1.4.5 GRUB 的配置实例.....	38
1.5 使用 Windows 的引导管理器引导 Linux.....	39
思考与练习.....	42
第2章 初次使用 Linux.....	43
2.1 系统引导和用户登录.....	43
2.2 X Window 使用基础.....	46
2.2.1 桌面环境的基本操作.....	46
2.2.2 在桌面系统中使用 X 终端.....	48
2.3 用户注销.....	51
2.4 系统关闭和重启.....	51
2.4.1 文字模式下的系统关闭.....	51
2.4.2 图形模式下的系统关闭.....	52
2.5 使用 Linux 上网指南.....	52
2.5.1 图形化浏览器的使用.....	52
2.5.2 文件下载.....	55
2.5.3 图形化电子邮件客户.....	61
2.6 如何获得在线帮助.....	67
2.6.1 手册页.....	67
2.6.2 包的文档.....	70
2.6.3 HOWTO 和 FAQ.....	71
2.6.4 locate 命令.....	72
2.6.5 info 页.....	72
思考与练习.....	73
第3章 使用 Linux 的基本命令.....	74
3.1 如何在 Linux 中运行程序.....	74
3.1.1 登录、注销 Linux.....	74
3.1.2 Linux 中的 shell 简介.....	75
3.1.3 shell 的工作原理和过程.....	78



3.1.4	使用 Windows 的超级终端 远程登录到 Linux .....	79	5.2.3	从双向继续查找指定的 字符串 .....	164
3.1.5	Linux 系统的最基本命令 .....	83	5.2.4	以单词为单位移动光标 .....	164
3.2	Linux 中的目录操作 .....	93	5.2.5	行内的快速光标移动 .....	165
3.2.1	使用目录管理文件 .....	93	5.2.6	使用行号定位行 .....	165
3.2.2	常用的目录操作 .....	94	5.2.7	将光标移动到当前屏幕的 某些位置 .....	166
3.2.3	路径和当前目录 .....	100	5.2.8	将光标返回到原来位置 .....	167
3.3	Linux 的文件操作 .....	103	5.2.9	让屏幕显示文件的指定位置 .....	167
3.3.1	基本的文件管理 .....	103	5.2.10	光标和屏幕定位小结 .....	167
3.3.2	文件和目录权限操作 .....	123	5.3	vi 的文本删除 .....	168
3.3.3	文件比较、查找和定位操作 .....	132	5.3.1	删除字符 .....	168
3.3.4	模拟 DOS 文件和目录命令 .....	147	5.3.2	删除行 .....	169
	思考与练习 .....	152	5.3.3	删除单词 .....	169
第 4 章	安装和使用汉字终端 .....	153	5.3.4	删除一行的一部分内容 .....	169
4.1	Linux 控制台中文文化 .....	153	5.3.5	删除光标至某一个字符 之间的文本 .....	169
4.1.1	Linux 控制台中文文化的现状 .....	153	5.3.6	使用行号删除文本行 .....	170
4.1.2	Linux 控制台基本工作原理 .....	153	5.4	vi 的文本添加 .....	170
4.2	zhcon 简介 .....	154	5.4.1	在光标的左侧添加文本 .....	170
4.2.1	什么是 zhcon .....	154	5.4.2	在当前行的上面或者下面 插入新行 .....	171
4.2.2	zhcon 的开发工作 .....	154	5.4.3	在行首和行末添加文本 .....	172
4.2.3	zhcon 的特点 .....	154	5.4.4	添加命令小结 .....	172
4.3	zhcon 的安装和使用 .....	155	5.5	vi 的撤销和重复命令 .....	173
4.3.1	zhcon 的下载以及其运行 环境 .....	155	5.5.1	多次撤销命令 .....	173
4.3.2	安装 zhcon .....	155	5.5.2	撤销一行上的所有修改 .....	173
4.3.3	启动和退出 zhcon .....	156	5.5.3	重复最后一次文本修改命令 .....	174
4.3.4	zhcon 的中文输入法 .....	156	5.6	vi 的文本替换 .....	175
4.3.5	使用 zhcon 的常见问题 .....	158	5.6.1	替换字符 .....	175
	思考与练习 .....	159	5.6.2	逐字符替换文本 .....	175
第 5 章	全屏幕编辑器 vi .....	160	5.6.3	多个字符替换单个字符 .....	175
5.1	全屏幕编辑器 vi 简介 .....	160	5.6.4	用多个单词替换单个单词 .....	176
5.1.1	Linux 编辑器简介 .....	160	5.6.5	行替换 .....	176
5.1.2	进入和退出 vi 编辑器 .....	161	5.6.6	修改行的剩余部分 .....	176
5.1.3	进入输入模式并进行删除 和修改操作 .....	162	5.7	全局替换 .....	177
5.1.4	简单的光标移动 .....	162	5.7.1	搜索当前行的第一个实例 .....	177
5.2	在文件中快速移动光标 .....	163	5.7.2	搜索指定行的第一个实例 .....	177
5.2.1	方向键的扩充 .....	163	5.7.3	设置过滤条件进行搜索 .....	177
5.2.2	查找指定的字符串 .....	163			

5.7.4	在同一行中替换所有实例 .....	178
5.8	移动和复制文本 .....	178
5.8.1	复制和粘贴行 .....	179
5.8.2	剪切和粘贴行 .....	180
5.8.3	复制字符和单词 .....	180
5.8.4	复制文本块 .....	181
5.8.5	移动文本块 .....	182
5.9	保存文件 .....	183
5.9.1	保存当前编辑的文件 .....	183
5.9.2	取消对文件的修改 .....	183
5.9.3	保存文本块 .....	184
5.9.4	覆盖已有文件 .....	184
5.9.5	向文件追加文本 .....	184
5.10	vi 编辑器与 shell 的交互 .....	185
5.10.1	读入文本文件 .....	185
5.10.2	在 vi 中运行子 shell .....	185
5.10.3	读入 shell 命令行的输出 .....	186
	思考与练习 .....	187
第 6 章	shell 的使用 .....	188
6.1	shell 基础 .....	188
6.1.1	什么是 shell .....	188
6.1.2	几种流行的 shell .....	188
6.2	shell 程序设计基础 .....	189
6.2.1	shell 基本语法 .....	189
6.2.2	shell 程序的变量和参数 .....	190
6.3	shell 编程的流程控制 .....	192
6.3.1	shell 程序设计的流程控制 .....	192
6.3.2	运行 shell 程序的方法 .....	196
6.4	bash 程序的调试 .....	197
6.4.1	shell 程序的调试 .....	197
6.4.2	shell 程序的跟踪 .....	197
6.5	bash 的内部命令 .....	198
6.6	bash 的其他命令 .....	199
6.6.1	简化命令输入 .....	199
6.6.2	重复执行命令 .....	200
6.6.3	定义命令的别名 .....	202
6.6.4	改变 bash 提示符 .....	204
6.6.5	作业控制 .....	205
6.6.6	定制 bash .....	206

6.6.7	bash 的作业 (job) 控制 .....	208
	思考与练习 .....	209
第 7 章	常用网络命令 .....	210
7.1	网络诊断和设置命令 .....	210
7.1.1	检测主机的连通性 —— ping 命令 .....	210
7.1.2	显示网络状态 —— netstat 命令 .....	212
7.1.3	显示和配置网络接口 —— ifconfig 命令 .....	216
7.1.4	网络环境配置工具 —— netconfig 命令 .....	220
7.1.5	显示数据包到主机间的路径 —— traceroute 命令 .....	221
7.1.6	与 Internet 域名服务器交互查询 —— nslookup 命令 .....	222
7.1.7	显示和操作 IP 路由表 —— route 命令 .....	224
7.1.8	网络配置综合练习 .....	226
7.2	常用网络工具 .....	228
7.2.1	文本模式 Web 浏览器 —— lynx 命令 .....	228
7.2.2	远程登录 —— telnet 命令 .....	229
7.2.3	远程文件传输 —— ftp 命令 .....	231
7.2.4	简单文本电子邮件客户 —— mutt 命令 .....	233
7.2.5	安全 shell —— ssh 命令 .....	235
7.2.6	安全的文件传输 —— sftp 命令 .....	237
	思考与练习 .....	238
第 8 章	文件系统管理 .....	239
8.1	Linux 文件系统简介 .....	239
8.1.1	Linux 文件系统概述 .....	239
8.1.2	Linux 文件系统的重要目录 .....	239
8.1.3	Linux 专用的文件系统 .....	241
8.2	Linux 磁盘管理 .....	242
8.2.1	显示与设定硬盘的参数 —— hdparm 命令 .....	242
8.2.2	Linux 下的硬盘分区工具 .....	

—— fdisk 命令 .....	243	10.2.2 手工启动进程 .....	274
8.2.3 文件系统创建工具		10.2.3 调度启动进程 .....	274
—— mkfs 命令 .....	245	10.2.4 进程的挂起及恢复命令	
8.2.4 文件系统挂接工具		—— bg、fg .....	279
—— mount 命令 .....	246	10.3 进程查看 .....	279
8.2.5 磁盘空间管理工具		10.3.1 who 命令 .....	279
—— df 和 du 命令 .....	253	10.3.2 w 命令 .....	279
8.2.6 检查与修复 Linux 文件系统		10.3.3 ps 命令 .....	280
—— fsck 命令 .....	255	10.3.4 top 命令 .....	281
8.2.7 调整 Linux 文件系统参数		10.3.5 进程调度 .....	282
—— tune2fs 命令 .....	257	思考与练习 .....	282
8.2.8 同步文件系统		<b>第 11 章 内核管理和模块管理</b> .....	283
—— sync 命令 .....	258	11.1 Linux 内核和模块简介 .....	283
8.2.9 启用/停用交换分区		11.1.1 Linux 内核的发展 .....	283
—— swapon 命令 .....	258	11.1.2 Linux 内核的组成 .....	284
8.2.10 /etc/fstab 文件 .....	258	11.1.3 Linux 内核子系统之间的	
8.3 磁盘复制命令 .....	260	关系 .....	284
思考与练习 .....	261	11.1.4 确定 Linux 内核的版本 .....	285
<b>第 9 章 用户管理</b> .....	262	11.2 升级 Red Hat Linux 定制的内核 .....	286
9.1 用户管理的基本操作 .....	262	11.2.1 Red Hat Linux 的定制	
9.1.1 增加用户 .....	262	内核的特点 .....	286
9.1.2 修改用户账号 .....	264	11.2.2 准备升级 Red Hat Linux	
9.1.3 删除或禁止用户账号 .....	266	的定制内核 .....	286
9.2 超级用户授权和创建多个超级用户 .....	266	11.2.3 下载升级的内核 .....	287
9.2.1 建立多个超级用户 .....	266	11.2.4 执行升级 .....	287
9.2.2 为普通用户分配特权 .....	267	11.2.5 校验初始 RAM 磁盘映像 .....	288
思考与练习 .....	268	11.3 从源代码升级 Linux 内核 .....	288
<b>第 10 章 Linux 进程管理</b> .....	269	11.3.1 内核编译模式 .....	289
10.1 内核对进程的管理 .....	269	11.3.2 新版本内核的获取 .....	289
10.1.1 linux 的进程 .....	269	11.3.3 升级 module-init-tools	
10.1.2 linux 的进程调度 .....	270	软件包 .....	289
10.1.3 虚拟内存 .....	271	11.3.4 配置、编译和安装 linux-2.6.x	
10.1.4 进程创建 .....	271	内核 .....	290
10.1.5 时钟和定时器 .....	272	11.3.5 引导测试 .....	292
10.1.6 程序执行 .....	272	11.3.6 make menuconfig 选项说明 .....	292
10.1.7 ELF 可执行文件格式 .....	273	11.4 Linux 模块管理 .....	299
10.1.8 ELF 共享库 .....	273	11.4.1 模块的概念 .....	299
10.2 Linux 进程管理命令 .....	273	11.4.2 加载模块 .....	299
10.2.1 进程及作业 .....	273	11.4.3 与模块操作有关的命令 .....	299



11.4.4	手工加载模块 .....	300	14.2.1	显示主机名 .....	320
11.4.5	从内存中卸载模块 .....	301	14.2.2	临时更改主机名 .....	321
	思考与练习 .....	301	14.2.3	永久更改主机名 .....	321
<b>第 12 章</b>	<b>压缩和备份 .....</b>	<b>302</b>	14.3	管理 hosts 文件 .....	321
12.1	备份的重要性 .....	302	14.4	配置文件 inittab .....	322
12.2	归档(备份)命令 .....	302	14.5	rc 启动脚本 .....	323
12.2.1	tar 命令 (tape archive) .....	302	14.6	管理系统服务 .....	324
12.2.2	dump 命令 .....	304	14.6.1	使用 ntsysv 管理服务 .....	324
12.2.3	ar 命令 .....	304	14.6.2	使用 chkconfig 管理服务 .....	325
12.2.4	cpio(copy in/out)命令 .....	305	14.6.3	手工启动和停止系统服务 .....	325
12.3	压缩命令 .....	306	14.6.4	创建特定级别的新服务 .....	326
12.3.1	compress 命令 .....	306		思考与练习 .....	329
12.3.2	gzip 命令 .....	307	<b>第 15 章</b>	<b>X Windows 系统的配置和使用 .....</b>	<b>330</b>
12.3.3	gunzip 命令 .....	308	15.1	安装和配置 XFree86 .....	330
12.3.4	zip 命令 .....	308	15.1.1	安装 XFree86 .....	330
12.3.5	unzip 命令 .....	308	15.1.2	配置 XFree86 .....	330
12.3.6	bzip2 命令 .....	309	15.1.3	使用 redhat-config-xfree86 配置 XFree86 .....	331
12.3.7	bunzip2 命令 .....	310	15.1.4	手工修改 XFree86 配置文件 .....	333
12.4	综合实例 .....	310	15.2	使用图形化桌面 .....	336
12.4.1	各种与压缩和归档相关 的文件名及其功能 .....	310	15.2.1	使用桌面 .....	336
12.4.2	常见文件格式操作实例 .....	310	15.2.2	使用面板 .....	337
	思考与练习 .....	311	15.2.3	使用小程序 .....	338
<b>第 13 章</b>	<b>应用软件安装方法 .....</b>	<b>312</b>	15.2.4	使用通知区域 .....	338
13.1	源代码安装 .....	312	15.2.5	在面板上添加图标和小程序 .....	339
13.2	RPM 包在文本环境的安装 .....	313	15.2.6	配置桌面面板 .....	339
13.2.1	使用 RPM .....	313		思考与练习 .....	340
13.2.2	安装和升级选项 .....	313	<b>第 16 章</b>	<b>安装和使用常用应用软件 .....</b>	<b>341</b>
13.2.3	查询选项 .....	314	16.1	安装和使用 PDF 阅读器 .....	341
13.2.4	验证选项 .....	314	16.1.1	Linux 提供的 PDF 阅读器 .....	341
13.2.5	删除选项 .....	315	16.1.2	安装 Acrobat Reader .....	341
13.2.6	RPM 使用举例 .....	315	16.1.3	使用 Acrobat Reader .....	342
13.3	RPM 包在 GUI 环境中的安装 .....	316	16.2	安装和使用多媒体播放器 MPlayer .....	343
13.3.1	使用文件管理器 Nautilus .....	316	16.2.1	下载 MPlayer .....	343
13.3.2	使用 Nautilus 安装 RPM 包 .....	317	16.2.2	安装 MPlayer .....	344
	思考与练习 .....	317	16.2.3	使用 MPlayer .....	345
<b>第 14 章</b>	<b>系统全局设置 .....</b>	<b>318</b>	16.3	安装和使用 Gaim .....	346
14.1	获得 Linux 系统信息 .....	318	16.3.1	下载 Gaim .....	346
14.2	设置主机名 .....	320	16.3.2	安装 Gaim .....	346

16.3.3	使用 Gaim.....	347
16.4	安装和使用 Flash 播放器.....	348
16.4.1	下载 Flash 播放器.....	348
16.4.2	安装 Flash 播放器.....	348
16.4.3	使用 Flash 播放器.....	349
16.5	安装和使用新的中文字体.....	349
16.5.1	准备要安装的字体.....	350
16.5.2	生成字体信息.....	350
16.5.3	将字体目录加入字体 服务器路径中.....	350
16.5.4	重新启动 xfs 字体服务器.....	350
16.5.5	使用新添加的字体.....	350
16.6	支持 NTFS 文件系统.....	351
16.6.1	下载 NTFS 的 RPM 包.....	351
16.6.2	安装 NTFS 的 RPM 包.....	352
16.6.3	挂接 NTFS 的分区.....	352
	思考与练习.....	352
第 17 章	OpenOffice 套件.....	353
17.1	OpenOffice.org 办公套件的功能.....	353
17.2	使用 OpenOffice 的字 处理程序 Writer.....	354
17.3	使用 OpenOffice 的电子表格 处理程序 Calc.....	356
17.4	使用 OpenOffice 的演示文稿 创建程序 Impress.....	357
17.5	使用 OpenOffice 的图形 处理程序 Draw.....	359
	思考与练习.....	359
第 18 章	排版软件 LaTeX 的配置和使用.....	360
18.1	安装 LaTeX 套件.....	360
18.2	简单使用 LaTeX.....	361
18.2.1	LaTeX 文档的编写.....	361
18.2.2	编译 LaTeX 文档.....	365
18.3	LaTeX 中文化.....	368
18.3.1	下载和安装 CJK4.5.2 宏包.....	368
18.3.2	安装和配置中文字体.....	369
18.3.3	在 LaTeX 中使用中文.....	369
	思考与练习.....	370
第 19 章	架设 Web 服务器.....	371

19.1	安装和使用 Apache 服务器.....	371
19.1.1	安装 Apache 服务器.....	371
19.1.2	启动 Apache 服务器.....	372
19.1.3	测试 Apache 服务器.....	372
19.2	配置 Apache 服务器的过程.....	372
19.3	对 Apache 进行基本设置.....	373
19.4	对 Apache 进行默认项设置.....	374
19.4.1	站点配置.....	374
19.4.2	记录日志设置.....	375
19.4.3	环境变量.....	376
19.4.4	目录.....	376
19.5	虚拟主机设置.....	377
19.5.1	添加和编辑虚拟主机.....	377
19.5.2	SSL 支持.....	378
19.6	服务器设置.....	379
19.7	调整性能.....	380
19.8	保存设置.....	380
	思考与练习.....	381
第 20 章	架设域名服务器.....	382
20.1	DNS 和 BIND.....	382
20.1.1	DNS 原理简介.....	382
20.1.2	BIND 简介.....	383
20.2	安装和配置 BIND.....	383
20.2.1	安装 BIND.....	383
20.2.2	启动和停止 BIND.....	384
20.2.3	配置 BIND.....	384
20.3	测试 DNS 服务器.....	390
20.3.1	DNS 客户端端的改动.....	390
20.3.2	使用 nslookup 测试 DNS 服务器.....	390
	思考与练习.....	391
第 21 章	多种平台之间的文件共享.....	392
21.1	与 Windows 系统共享文件: Samba.....	392
21.1.1	配置 Samba 服务器.....	392
21.1.2	连接 Samba 共享.....	396
21.2	与 Unix/Linux 共享文件: NFS.....	397
21.2.1	导出 NFS 文件系统.....	398
21.2.2	启动和停止 NFS 服务器.....	401
21.2.3	在客户端挂接 NFS 文件.....	

系统 .....	401	23.1.2 安装 vsftp .....	410
思考与练习 .....	402	23.1.3 启动和停止 vsftpd .....	410
<b>第 22 章 架设邮件服务器 .....</b>	<b>403</b>	<b>23.2 使用和设置 vsftp .....</b>	<b>410</b>
22.1 电子邮件简介 .....	403	23.2.1 vsftp 服务器的默认设置 .....	410
22.1.1 电子邮件系统的工作原理 .....	403	23.2.2 vsftp 服务器的基本配置 .....	412
22.1.2 电子邮件的标准 .....	403	23.2.3 vsftp 服务器的匿名用户	
22.1.3 电子邮件系统的组成 .....	404	设置 .....	413
22.2 安装和启动 Sendmail .....	404	23.2.4 vsftp 服务器的高级配置 .....	414
22.2.1 安装 Sendmail 软件 .....	404	23.2.5 vsftp 服务器配置实例 .....	415
22.2.2 启动/重新启动/停止		思考与练习 .....	416
Sendmail 服务 .....	405	<b>第 24 章 构建防火墙 .....</b>	<b>417</b>
22.3 配置 Sendmail .....	405	24.1 安装 iptables .....	417
22.3.1 配置 Sendmail 的一般步骤 .....	405	24.1.1 防火墙简介 .....	417
22.3.2 根据情况修改 sendmail.mc		24.1.2 安装和启动防火墙 .....	417
文件 .....	406	24.2 Netfilter 防火墙系统框架 .....	418
22.3.3 为新用户开设电子邮件账号 .....	406	24.2.1 Netfilter 防火墙的特点 .....	418
22.3.4 为电子邮件账户设置别名 .....	407	24.2.2 Netfilter 防火墙在 IPv4 中	
22.3.5 指定邮箱容量限制 .....	407	实现的原理和结构 .....	419
22.3.6 支持 POP 和 IMAP 功能 .....	407	24.3 理解 Netfilter 防火墙技术 .....	420
思考与练习 .....	408	24.3.1 建立规则和链 .....	420
<b>第 23 章 架设 FTP 服务器 .....</b>	<b>409</b>	24.3.2 netfilter/iptables 系统的优点 .....	423
23.1 安装 vsftp 服务器软件 .....	409	24.3.3 防火墙实例 .....	423
23.1.1 vsftp 简介 .....	409	思考与练习 .....	425



# 第1章 安装 Red Hat Linux 9

Linux 是一种可以运行在 PC 机上的免费的 UNIX 操作系统。从 Linux 问世以来，一直受到业界的青睐，极具吸引力。Linux 不仅在高端的服务器市场占有很大的份额，而且在桌面和嵌入式领域也得到了广泛应用。目前虽然有多种版本的 Linux 系统存在，但是相对而言，Red Hat Linux 不管是从在用户数量、软硬件支持，还是性能和易用性上都占有优势。

本章将通过实例介绍 Red Hat Linux 的安装步骤以及相关的安装经验，使读者快速进入 Linux 世界，避免出现一些常见的错误。通过本章的学习，读者能够了解 Linux 的历史，并以 Red Hat Linux 9 为例介绍如何根据实际情况灵活安装 Red Hat Linux 操作系统。

## 1.1 Linux 操作系统简介

### 1.1.1 Linux 的历史

Linux 是一种可以运行在 PC 机上的免费的 UNIX 操作系统。Linux 最早是由计算机业余爱好者 Linus Torvalds（当时他是芬兰赫尔辛基大学的学生）在 1991 年开发出来的。他的目的是想设计一个代替 Minix（是由一位名叫 Andrew Tanenbaum 的计算机教授编写的一个操作系统演示程序）的操作系统，这个操作系统可用于 x86 系列 CPU 的个人计算机上，并且具有 Unix 操作系统的全部功能。Linux 的吉祥物是 Linux 企鹅，它是由 Linus Torvalds 挑选出来的，用于代表它所创立的 Linux 操作系统。

Linus Torvalds 把 Linux 的源程序在 Internet 上公开，其目的是建立不受任何商品化软件版权制约的、全世界都能自由使用的、与 Unix 兼容的操作系统。世界各地的编程爱好者自发组织起来对 Linux 进行改进，并且编写各种应用程序，今天 Linux 已发展成一个功能强大的操作系统，成为操作系统领域最耀眼的明星。

Linux 的开发以及它的源代码是在 GNU（General Public License，通用公共许可证）的保护之下的，每个人都可以完全免费获取、使用、修改和重新发布它们。但是这并不意味着 Linux 和它的一些周边软件发行版本也是免费的。Linux 有着广泛的用途，包括网络、软件开发、用户平台等等。Linux 被认为是一种高性能，低开支的可以替换其他昂贵操作系统的系统。

### 1.1.2 Linux 操作系统的特点

Linux 能够在 PC 计算机上实现 Unix 的全部特性，具有多任务、多用户和分时的能力。Linux 是在 GNU 公共许可权限下免费获得的，是一个符合 POSIX 标准（Portable Operating System Interface，可移植操作系统接口）的操作系统。Linux 操作系统软件包不仅包括完整的 Linux 操作系统，而且还包括了文本编辑器、高级语言编译器等应用软件；它还包括带有多个窗口管理器的 X-Windows 图形用户接口，允许用户使用窗口、图标和菜单对系统进行操作。

Linux 之所以受到广大计算机爱好者的喜爱，主要原因有：

- 它属于自由软件。用户不必支付任何费用，就可以获得它及其源代码，并可以依据自己的需要对其进行必要的修改，用户不仅可以无偿地使用，而且可以对其进行自由的传播。
- 它具有 Unix 的全部功能。任何使用 Unix 操作系统，或想要学习 Unix 操作系统的人都可以从 Linux 中获益。
- Linux 得到多种硬件厂商的支持。它支持各种硬件，大多数可以在 Windows 中使用的设备，都可以在 Linux 中使用，只有少数设备需要另外安装驱动程序。
- 具有众多的应用软件。Linux 的发行版中不仅包含了许多常用的应用软件，而且世界上还有许多可用于 Linux 的开放源码的免费软件，另外，世界大多数软件公司的商业软件也都支持 Linux 操作系统。

由于 Linux 是一套具有 Unix 的全部功能的免费操作系统，所以它在众多软件中占有很大的优势，为广大计算机爱好者提供了学习、探索以及修改计算机操作系统内核的机会。

操作系统是一台计算机必不可少的系统软件，是整个计算机系统的灵魂。一个操作系统是一个复杂的计算机程序集，它提供操作过程的协议或行为准则。没有操作系统，计算机就无法工作，就不能解释和执行用户输入的命令或运行简单的程序。大多数操作系统都是由一些主要的软件公司支持的商品化程序，用户只能有偿使用。如果用户购买了一个操作系统，他就必须满足供货商所要求的一切条件。因为操作系统是系统程序，所以用户不能擅自修改或试验操作系统的内核。这对于广大计算机爱好者来说无疑是一种束缚。

要想充分发挥计算机的作用，仅有操作系统是远远不够的，用户还需要各种应用程序的支持。应用程序是用于处理某些工作（如字处理）的软件包，通常情况下它也只能有偿使用。每个应用程序的软件包都为特定的操作系统和机器编写。使用者无权修改这些应用程序，而使用 Linux 可以克服以上弊端，将操作系统变成一种操作环境。

由于 Linux 是一套自由软件，所以用户可以无偿地得到它及其源代码，可以无偿地获得大量的应用程序，而且可以任意修改和补充它们。这对用户学习、了解 Unix 操作系统的内核是非常有益的。学习和使用 Linux，能为用户节省一笔可观的开支。Linux 是目前惟一可免费获得的、为 PC 机平台上的多个用户提供多任务、多进程功能的操作系统，这是人们要使用它的主要原因。就 PC 机平台而言，Linux 提供了比其他任何操作系统都要强大的功能，Linux 还可以使用户远离各种商品化软件提供者促销广告的诱惑，再也不用承受每过一段时间就升级之苦，因此，可以节省大量用于购买或升级应用程序的资金。

Linux 不仅为用户提供了强大的操作系统功能，而且还提供了丰富的应用软件。用户不但可以从 Internet 上下载 Linux 及其源代码，而且还可以从 Internet 上下载许多 Linux 的应用程序。可以说，Linux 本身包含的应用程序以及移植到 Linux 上的应用程序包罗万象，任何一位用户都能从有关 Linux 的网站上找到适合自己特殊需要的应用程序及其源代码，这样，用户就可以根据自己的需要下载源代码，以便修改和扩充操作系统或应用程序的功能。

Linux 为广大用户提供了一个 PC 机上学习和使用 Unix 操作系统的机会。尽管 Linux 是由计算机爱好者们自主开发的，但迄今为止，它已经是非常稳定的操作系统，从而为用户学习和使用目前世界上最流行的 Unix 操作系统提供了廉价的机会，现在已经有许多软件公司（如 Red Hat 和 Turbo Linux）发布新的 Linux 操作系统发行版本，从而更加方便 Linux

的使用。目前, Linux 已成为 Unix 系统在个人计算机上的一个替代品, 并能用于替代那些较为昂贵的操作系统。因此, 如果一个用户在公司上班的时候在 Unix 系统上编程, 或者在工作中是一位 Unix 的系统管理员, 他就可以在家里安装一套 Unix 的兼容系统, 即 Linux 系统, 在家中使用 Linux 就能够完成一些工作任务。

### 1.1.3 Linux 操作系统的应用领域

由于 Linux 操作系统具有以上的特点, 使得 Linux 操作系统的应用日趋广泛。目前, Linux 主要应用在以下 3 个领域:

(1) 高端服务器领域。Linux 内核具有稳定性、开放源码等特点, 加上 Linux 强调开放源代码, 从而不必支付大笔使用权费用, 所以 Linux 近年来已获得 IBM、戴尔、康柏、SUN 等世界著名厂商的支持。当今常用的服务器操作系统有 Unix、Linux 和 Windows, 根据调查, Linux 操作系统在服务器市场的占有率已达 25%。由于 Linux 可以提供企业网络环境所需的各种网络服务, 加上 Linux 服务器也可以用来提供虚拟私有专用网 (VPN) 或充当路由器 (Router) 与网关 (Gateway), 因此在不同操作系统争战的情况下, 企业只要掌握 Linux 技术配合系统整合与网络等技术, 便能享有低成本、可靠的网络环境。

(2) 桌面应用领域。众所周知, Windows 在桌面领域一直占据绝对的优势。但是随着 Linux 操作系统在图形用户接口方面和桌面应用软件的发展, Linux 在桌面应用方面也得到了显著的提高, 越来越多的桌面用户转而使用 Linux。事实证明, Linux 已经能够满足办公、娱乐和信息交流的需求。

(3) 嵌入式应用领域。通常情况下, 嵌入式及信息家电的操作系统支持所有的运算功能, 但是需要根据实际的应用对其内核进行定制和裁减, 以便为专用的硬件提供驱动程序, 并且在此基础上进行应用开发。目前, 能够支持嵌入式的常见操作系统有 Palm OS、嵌入式 Linux 和 Windows CE。虽然 Linux 在这个领域还是初生之犊, 但是以 Linux 的特性正好符合 IA (基于 Intel 架构) 产品的操作系统小、稳定、实时与多任务等需求, 而且 Linux 开放源代码、不必支付许可证费用, 许多世界知名厂商包括新力、IBM 等纷纷在其 IA 中采用 Linux, 开发视频电话和数字监视系统等多种应用。

### 1.1.4 常见的 Linux 发行版

用户能接触到的 Linux 发行版本主要包括: Slackware, Red Hat, Debian, S.u.S.E。下面简要介绍一下它们的特色。

#### 1. Walnut Creek Slackware

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。网址为 <http://www.slackware.com>, 并且可以从 <http://www.slackware.com/getslack> 所列的镜像站点列表中选择下载。

Slackware Linux 是最早出现的 Linux 发行套件之一, 它是由 Patrick Volkerding 制作的。一开始它的载体是软盘, 因此它一直保留着安装时的简单目录结构, 以软盘为单位的安装系列, 对于没有光驱的用户, 它也许是惟一的选择。Slackware 的特点就是安装简单, 目录结构清楚, 版本更新快, 在 1997 年一年就推出了好几个版本。其缺点是软件种类不如 Red



Hat 和 Debian 多, 并且其安装不如 Red Hat 快速、简洁、直观。其升级方式也不如 Red Hat 和 Debian 简单, 同 RPM (Red Hat Package Manager) 相比, Slackware 只有一个相对简陋的包安装工具 `pkgtool`, 经常会出现卸载软件后, 其他软件无法使用的故障。但是 Slackware Linux 最大的特点为: 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。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。

## 2. Red Hat Linux

Red Hat Linux 是由 Red Hat software 公司发布的。其网址为 <http://www.Red Hat.com>, ftp 站点为 <ftp://ftp.redhat.com>, 可以从该站点下载它的发行版。

Red Hat 的问世比 Slackware 和 Debian 都要晚, 但却后来居上 (Red Hat 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 Red Hat Linux (昵称为小红帽 Linux, 其吉祥物是一个头带红帽的人)。Red Hat Linux 的优点如下:

- **支持硬件平台多。**Red Hat Linux 从 4.0 版本开始, 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。为电影 Titanic 制作计算机特技效果的公司制作过程中就使用了 105 台运行 Red Hat Linux 的 Alpha 计算机。现在 Red Hat Linux 的发行版能支持 7 种系统结构。
- **优秀的安装接口。**从 4.x 版开始, Red Hat 公司对 Red Hat Linux 的安装接口进行了大手术, 由原先 3.x 版本的在大批 `Img` 映像文件中去寻找启动盘, 变成只须制作一张启动盘就可以进行 CD-ROM 方式的安装工作。整个安装过程非常简单明了, 用户只须选择很少的选项就可以开始安装, 与 Slackware 相比要简单得多。Red Hat Linux 为用户提供了文本和图形两种安装接口, 它可以支持光盘启动安装、硬盘方式安装、ftp 方式安装、http 服务器方式安装、NFS 方式安装和 ISO 方式安装六种安装方式。
- **独特的 RPM 升级方式。**Red Hat 所有的软件包都是以 RPM (Red Hat Package Manager) 方式包装的, 这种包装方式让用户可以轻松地软件升级, 彻底卸载 (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 Red Hat Linux 之后, 就不需要再重新安装, 只需要不断升级就可以了。这使得 RPM 成为 Red Hat Linux 最大的卖点。并且 Red Hat 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 方式带来的便利之处, 免去手工编译之苦。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。**Red Hat 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共享软件。这些软件都经过 Red Hat 公司技术人员的认真调试和配置, 让一个普通用户安装完 Red Hat Linux 之后立刻就能享受配置完整的 Web 服务器和 Samba 服务器等多种软件包, 而不需要用户再花费大量

时间和精力去编译和安装这些软件。

- **安全性能好。**Red Hat 默认配置下的系统安全性能已经很高，同时它还提供 PAM (Pluggable Authentication Modules, 可插入认证模块) 以加强系统安全性能和系统管理的扩充性。如果用户计划提高系统的安全性，就要安装更多的安全软件，例如 TCP wrapper，有关这部分的详细信息，可参阅相关的用户指南的说明。Red Hat 提供快速的系统安全补丁建议，Red Hat 公司的有关安全方面的邮件列表 (Mailing List) 是最权威的 Linux 安全方面的消息来源之一。如果用户对于安全方面十分在意的话，就需要订阅 Red Hat 公司的 Mailing List，另外还可以经常浏览它们的主页，然后使用 RPM 来更新软件或对软件打补丁，那么系统安全的维护工作就会轻松许多。
- **方便的系统管理接口。**Red Hat 提供了一套 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、安装打印机等系统管理方面的工作，与 UNIX 通常采用的字符方式的接口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 目录及其子目录中收录了完整的 HOWTO、LDP 和 FAQ 系列说明文件，还有 Red Hat 独有的长达 250 多页的用户指南，详细说明各种软件安装、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，Red Hat Linux 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，Red Hat 可以让用户很快享受到 Linux 的强大功能，而免去繁琐的安装与设置工作。但是它的目录结构复杂，而且系统配置文件也比较凌乱，对于初学者来说对系统的配置会比较麻烦。

### 3. Debian Linux

这是由 GNU 发行的 Linux 发行套件，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、升级方便、软件之间的关联性强，并且拥有开放式的开发环境。

Debian 的网站为 <http://www.debian.org>，可以从 <ftp://ftp.debian.org/debian> 下载它的发行版。

Debian 是独一无二的，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，一个久经考验的错误跟踪系统允许用户迅速报告它们发现的臭虫 (Bug) 和安全问题，然后由 Debian 的开发人员迅速进行维护工作。一般说来，很快就会有该软件包的一个新修正版本放置到相应的 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 Red Hat 的 RPM 方式类似的方式来进行软件升级，因而在 Debian Linux 环境下安装软件也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。

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, 只是我国用户要获得 Debian Linux 相对要困难一些, 国内只有少数 FTP 站点收集了 Debian Linux, 其中有 [ftp.sepc.ac.cn](http://ftp.sepc.ac.cn)。

## 1.2 准备安装 Red Hat Linux 9

### 1.2.1 选择硬件平台

Red Hat Linux 可以在许多硬件平台上运行, 如 x86 (PC) 系列、Alpha 以及 Sparc 等。本节将主要讲述基于 x86 系列的安装过程, 因为该系列是最主要的 Linux 平台。其他平台上的安装过程大致相同, 具体细节可以参考 Red Hat 的网站。

### 1.2.2 Red Hat 的版本

Red Hat Linux 从发行版 1.0 开始到现在的 9.0 的版本, 其中 Red Hat Linux 6.0 已经比较完善, 但是还存在中文支持不够好和系统配置不一致的问题。Red Hat Linux 7.3 的中文支持已经比较完善, Red Hat Linux 9 是 Red Hat Linux 系列中最高的版本, 它也为 Red Hat Enterprise Linux 和 Fedora 计划奠定了坚定的基础。Red Hat Linux 9 可以从 [ftp://ftp.redhat.com](http://ftp.redhat.com) 站点免费下载。

在 Red Hat Linux 9 之后, Red Hat 公司提出了 Red Hat Enterprise Linux 和 Fedora 计划, Red Hat Enterprise Linux 是专门为企业应用而设计和开发的 Linux 版本, 不过这些版本是收费的, 不能免费下载。Fedora 计划的 Fedora 则是一个新的免费 Linux 版本, 它是专门为开发人员和新技术狂热的人设计的, 它不能用于关键的计算环境。使用者可以从 [ftp://ftp.redhat.com](http://ftp.redhat.com) 和它的镜像站点免费下载 Fedora。

本书专门讲述 Red Hat Linux 9 的安装和使用, 原因是 Red Hat Linux 9 不仅支持企业计算, 也支持桌面应用, 同时 Red Hat Linux 9 也可以容易得到。而且本书所讲的绝大多数内容也适用 Red Hat Enterprise Linux 和 Fedora。

### 1.2.3 检查 Red Hat Linux 9 的硬件需求

从最开始到现在, Linux 就规定了正常运行的最小硬件需求, 而且该要求至今也没有变化。Red Hat Linux 对硬件的要求很低, 但是如果要使 Linux 支持服务器级的应用, 则对硬件的要求可能会提高。要查看 Red Hat Linux 是否支持机器的硬件配置, 请查阅网页 [http://www.redhat.com/software/rhl9\\_hcl.html](http://www.redhat.com/software/rhl9_hcl.html), 该网页包含了 CPU、显卡、SCSI 控制器、IDE 控制器、网卡、Modem 和声卡的兼容列表。

当然, 为了提高 Red Hat Linux 系统的整体性能, 使用者需要使用性能更高的硬件作为支撑。以下是建立一个强大的 Red Hat Linux 系统所遵循的基本策略:

- CPU。CPU 是系统关键的计算和控制部件, 处理器越快越好。当然, 并不一定要使用当前最高性能的处理器。一般来说, 计算密集性的应用, 需要一个很高性能的处理器, 例如网络计算中处理器的性能越高, 效果就越好, 比如 Pentium IV 2.8GHz。但是如果仅仅在 Red Hat Linux 上运行 Apache 来创建一个 Web 服务器,

那么考虑选择中档，甚至低档的处理器，可以选择 Pentium II 的处理器。

- **RAM。**RAM 也是影响系统性能最关键的因素之一，内存越大越好。一般，对一个 Web 服务器来说，如果需要每秒钟响应的次数越多时，需要的内存数量就越多。
- **硬盘。**服务器需要多大硬盘取决于它完成的工作。如果想让服务器为 Web 站点或者局域网存储大量文件，那么要确定容量需求，并购买适当大小的硬盘。现在大容量硬盘价格已经非常便宜，硬盘的大小并不是一个关键问题，同时还要注意选择硬盘的种类。目前，有两类选择：IDE/EIDE 或者 SCSI。如果服务器对整个局域网或者 Internet 都很关键，比如要提供 Web、FTP 或邮件服务，那么请使用多个 SCSI 硬盘，它的速度会更快；另外，使用多个 SCSI 硬盘，可以减少驱动子系统的等待时间，这样会提高服务器的整体性能。
- **网卡。**为了提高服务器和计算机的性能，需要高性能的网络适配卡（NIC）。例如，可以选择 3Com 和 Intel 公司生产的高性能 NIC。现在的 Red Hat Linux 能很好地支持即插即用（PNP，Plug and Play）功能。
- **其他硬件。**其他重要的硬件包括 CD-ROM 驱动器、DVD 驱动器或软盘驱动器。一般来说，服务器上的 CD-ROM 可以是 IDE 接口，或者为 SCSI 接口，但其速度并不要求是最快的。除此之外，对于重要的服务器，还须考虑是否配置服务器级的机箱、电源和 UPS 等设备。

Red Hat Linux 9 的多种安装方式的最小配置参见表 1-1。当硬件安排就绪之后，就可以准备安装 Red Hat Linux 了。

表 1-1 Red Hat Linux 9 安装时的最小配置

	CPU	内存	硬盘
文本模式	200MHz	64MB	
图形模式	400MHz	192MB	
定制（最小）			475MB
定制（最大）			5GB
服务器			850MB
个人桌面			1.7GB

#### 1.2.4 准备安装 Red Hat Linux 9

##### 1. 获得 Red Hat Linux 9 安装光盘

首先从 Red Hat 站点 [http://ftp.redhat.com/pub/Red\\_Hat/linux/9/en/iso/i386](http://ftp.redhat.com/pub/Red_Hat/linux/9/en/iso/i386) 下载 Red Hat Linux 9 的光盘映像文件 shrike-i386-disc1.iso、shrike-i386-disc2.iso 和 shrike-i386-disc3.iso。然后使用 Nero 等刻录软件将它们刻录成光盘，或者可以直接购买 Red Hat Linux 9 的安装光盘。Red Hat Linux 9 的安装盘共有 3 张。

**注意：**当将 Red Hat CD-ROM 放在一台远程计算机上时，可以通过远程计算机提供的 FTP、NFS 或者 SMB 服务安装 Red Hat Linux。这些方法比较容易出错而且有时不能正常运行，所以笔者强烈建议使用本地的 CD-ROM 驱动器，并且在本地完成

所有安装任务, 这样既省时又省力。

如果必须通过 FTP、NFS 或者 SMB 服务安装系统, 请确保本地的计算机能够提供所需服务, 而且远程的 Red Hat 服务器已将 CD-ROM 挂接并导出。也可以将整个 Red Hat CD-ROM 拷贝到硬盘以加快安装速度。

基于本地 CD-ROM 安装过程最简单, 可以省去许多麻烦, 本章下面的内容将采用这种方式安装。

较新的主板的 BIOS 都支持从 CD-ROM 启动系统, 这样就可以直接从光盘进行安装, 就没有必要制作引导盘。在这种情况下, 只需重新设置 BIOS, 使之能够从光盘安装启动系统。

**注意:** 在安装结束后还需要将 BIOS 的设置恢复成硬盘启动。

大多数较新的 BIOS 版本允许用户选择一个引导序列, 如 CD-ROM, LAN, C, A。这样 BIOS 首先试图从光盘启动, 如果光盘不能启动, 然后从网络, 接着从硬盘, 最后从软盘引导。这样就不需要在安装完成后, 再恢复设置, 只要注意每次引导前将可以引导的光盘从光盘驱动器中取出 CD-ROM 光盘, 就可以从硬盘启动系统了。

## 2. 在 Windows 9x/XP/2000/2000 下创建引导盘

按以下步骤在 Windows 9x/XP/2000/2000 下创建引导盘:

(1) 假定 CD-ROM 驱动器是 E, 而且假定 Red Hat Linux 9 的第一张安装盘已经放入到驱动器, 而且当前盘为 E 盘, 请在 MD-DOS 窗口中输入如下内容 (黑体为输入内容):

```
E:\>e:\dosutils\rawrite
```

(2) rawrite 程序将会显示如下提示信息:

```
Enter disk image source file name:
```

(3) 输入映射文件名 images\bootdisk.img。

rawrite 程序将会显示如下提示信息:

```
Enter disk image source file name: \images\bootdisk.img
```

(4) 输入正确的驱动器名。一般使用 A 盘, 则键入 a: 作为目标驱动器。然后将一张已格式化的空软盘放入驱动器, 出现如下提示:

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

并且按 Enter (回车) 键继续。

(5) rawrite 程序将 bootdisk.img 文件映像写入软盘。这个过程大约要花 1~2 分钟。这样就完成了软盘的制作。

(6) 可以使用列目录命令, 显示 A 盘的目录, 目录如下:

```
Volume in drive A has no label.
```

```
Volume Serial Number is 3F83-3D78
```

```
Directory of A:\
```

```
2003-10-07  18:26                7,856 ldlinux.sys
2003-10-07  18:26                435 SYSLINUX.CFG
2003-10-07  18:26                364 BOOT.MSG
```

```

2003-10-07 18:26          534,021 INITRD.IMG
2003-10-07 18:26          888,750 VMLINUZ
2003-10-07 18:26           6,056 SPLASH.LSS
2003-10-07 18:26           1,026 GENERAL.MSG
2003-10-07 18:26           660 OPTIONS.MSG
2003-10-07 18:26           869 PARAM.MSG
2003-10-07 18:26           557 RESCUE.MSG
2003-10-07 18:26           549 SNAKE.MSG
      11 File(s)      1,441,143 bytes
      0 Dir(s)        19,968 bytes free

```

### 3. 在 Linux 创建启动盘

如果没有安装 MS-Windows 系统, 可以通过另外一台装有 Linux 系统的计算机创建启动盘, 步骤如下:

(1) 将 Red Hat Linux 9 安装盘的第一张盘放入光盘, 并将它挂接到 `/mnt/cdrom` 目录。执行如下命令:

```
# mount -t iso9660 /dev/cdrom /mnt/cdrom
```

(2) 将当前目录转到引导映像文件所在的目录 `/mnt/cdrom/images`, 命令如下:

```
# cd /mnt/cdrom/images
```

(3) 假定软盘驱动器为 `/dev/fd0` (默认软盘驱动器), 并插入一张 1.44MB 的软盘 (3.5 英寸高密), 执行如下命令:

```
# dd if=bootdisk.img of=/dev/fd0 bs=1440k
```

这样就创建了所需的启动盘。启动盘创建完毕, 就可以从 CD-ROM 安装 Red Hat Linux 9 了。

## 1.3 安装 Red Hat Linux 9

如果从光盘引导系统, 先将 Red Hat Linux 9 的第一张安装盘放入光盘驱动器, 然后重新启动系统。如果使用软盘引导系统, 先将刚才制作的启动盘插入机器的 A 盘驱动器, 并将第一张安装盘插入光盘驱动器, 然后再重新启动系统。

### 1.3.1 选择安装模式

系统启动之后, 用户将看到图 1-1 所示的欢迎屏幕, 而且可以看到如下的提示符:

```
boot:
```



图 1-1 Red Hat Linux 9 安装的第一个欢迎屏幕

这时用户有 3 种选择。

- 直接按 Enter 键，开始标准安装或升级过程。
- 在 boot: 提示符后输入 `linux export` 以专家模式安装或者升级。在专家模式中，Red Hat Linux 的安装过程不会试图自动监测硬件，只允许用户手工选择各种硬件的型号。只有当用户确实是专家，或者在某个硬件的自动监测有问题的情况下才使用专家模式。
- 在 boot: 提示符后输入 `linux rescue` 进入应急模式。当试图在某次事故之后恢复系统时可选择此项。

在这 3 种方式下都可以输入内核参数，以便根据需求来安装系统。如果用户第一次安装 Red Hat Linux，最好的选择就是按 Enter 键开始安装过程。

**注意：**许多系统管理员更倾向于使用自动化的安装方法，在他们的机器上安装 Red Hat Linux。为满足这种需要，Red Hat 开创了 kickstart 安装方法。使用 kickstart，系统管理员可以创建单个文件，该文件包括对典型 Red Hat Linux 安装中所询问的问题的回答。

kickstart 文件可以被保留在单个服务器系统上，并可以被计算机在安装过程中读取。这种安装方法能够支持使用单个 kickstart 文件来在多台机器上安装 Red Hat Linux，从而成为网络和系统管理员的理想选择。

kickstart 让用户自动化大部分 Red Hat Linux 的安装任务。

### 1.3.2 测试安装介质

接下来就出现测试安装介质屏幕，检测光盘是否完整，是否能够进行安装，如 1-2 所示。

因为安装盘的光盘介质有 3 张，任何一张出现问题，都会导致安装失败，因此 Red Hat Linux 的安装程序在安装一开始就提供了介质测试，来判断这三张安装盘是否完好无损。当第一张检测完毕后，安装程序就提示你插入下一张盘，按 Enter 键开始对这张盘测试，直

到第3张测试完毕，方可继续安装。如果在这个过程中，任何一张盘出现问题，则需要重新刻录对应的光盘，然后进行测试。如果读者能确信光盘是完好无损的，就可以单击 Skip 按钮跳过检测过程，这样就能够节省一定的时间。



图 1-2 检测安装介质的对话框

**注意：**在安装过程中，可以按 Tab 键在不同的按钮、输入框和其他控件间改变输入焦点。如果是文本方式，也可以使用向左和向右的方向键。

在图形安装界面中，可以使用鼠标进行选择和操作。除此之外，也可以使用 Tab 键切换输入焦点，当切换到希望的按钮上时，可以按 Enter 键来执行该按钮的相关操作。

### 1.3.3 选择语种、键盘和安装途径

在出现开始的欢迎屏幕之后，用户需要选择安装过程中使用的语言，默认为英国英语，如图 1-3 所示。在列表中有多种语言可进行选择，本过程选择“简体中文”（Chinese (Simplified)（简体中文））。

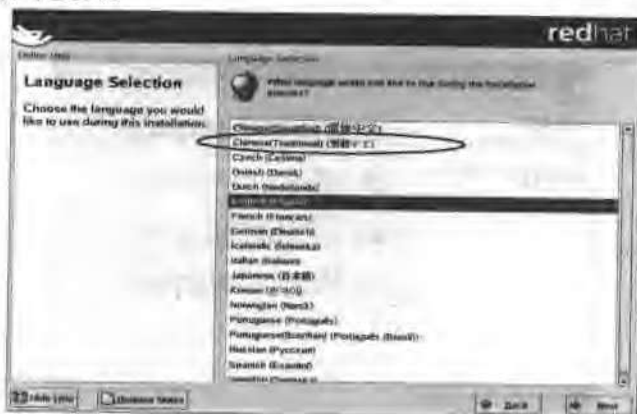


图 1-3 选择安装过程中所使用的语言



单击 Next 按钮到下一页, 选择键盘类型, 按 Enter 键使用默认的 U.S. English 键盘布局。接下来要选择安装的鼠标类型, 如图 1-4 所示。

注意: 对于笔记本电脑, 一般可以选择 ALPS GlidePoint (PS/2), 也可以选择 3 Button Mouse (PS/2)。

如果错误的选择鼠标, 安装完成后将会导致鼠标操作的严重问题。

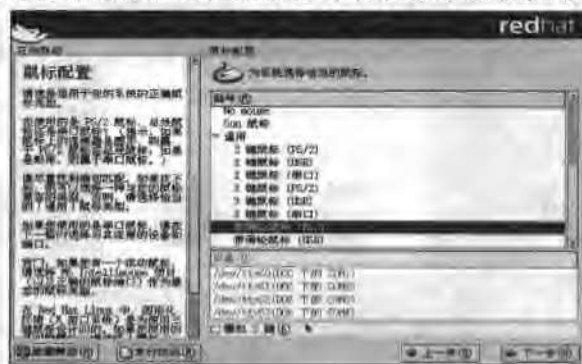


图 1-4 选择鼠标类型

#### 1.3.4 选择安装类型

当在图 1-4 中按回车键之后, 会让用户选择安装类型, 如图 1-5 所示。安装程序提供了 4 种安装类型。

- 个人桌面。适合便携式计算机, 具有图形化桌面, 但是缺乏软件开发和管理工具的图形化接口。特点是占用的硬盘空间小。
- 工作站。适合于不需要提供网络服务的工作站, 而且提供图形化接口、软件开发和管理工具。
- 服务器。可以提供多种网络服务, 如文件共享、打印共享、WWW 服务、ftp 服务和域名服务等多种网络服务。同时也可以选择安装图形桌面环境。
- 定制。用户可以根据需要选择各种所需的组件。这种方法特别适合高级用户安装, 因为用户可以完全控制安装的进程。

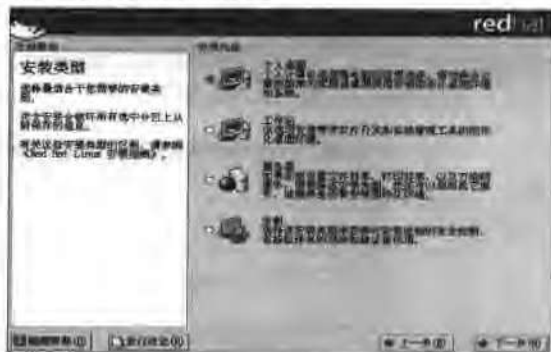


图 1-5 选择安装类型

### 1.3.5 磁盘分区

磁盘分区是一个非常重要的步骤，硬盘分区后才能使用。安装程序提示用户进行磁盘分区。首先选择分区方式，如图 1-6 所示。Red Hat Linux 9 安装程序提供了两种分区方式，一种是自动分区方式，另一种为手工分区方式（使用 Disk Druid 工具）。

安装 Linux 的时候需要进行的硬盘分区对很多初学者来说，都是一件很头痛的事情。因为稍有不慎就会把硬盘的所有数据弄丢了。不过，Red Hat Linux 9 的分区确实非常容易。使用 Red Hat Linux 9 提供的 Disk Druid 硬盘分区工具就能使用图形化的操作向导进行磁盘分区了。

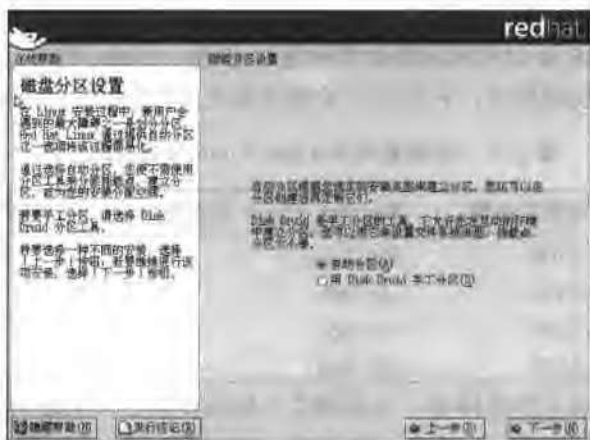


图 1-6 选择磁盘分区方法

**注意：**要谨慎使用“自动分区”进行自动分区模式，自动分区如果使用不当，就会破坏硬盘的已存在的 Windows 分区，造成数据丢失。

在此我们选择“用 Disk Druid 手工分区”模式。手动分区的操作窗口如图 1-7 所示。

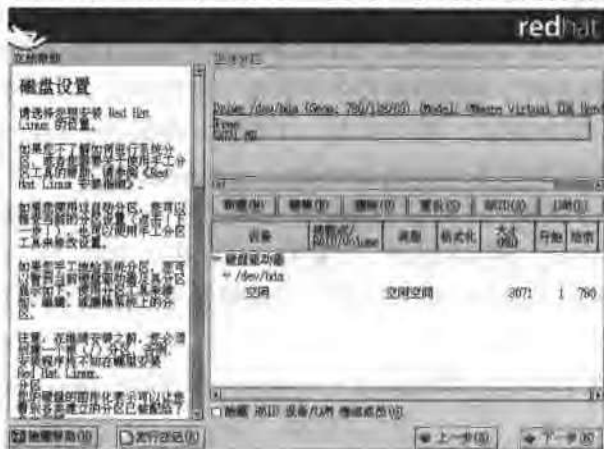


图 1-7 手工分区

硬盘分为 IDE 硬盘和 SCSI 硬盘两种，PC 机通常使用 IDE 硬盘，因为它的价格相对便

宜。服务器通常使用 SCSI 硬盘，它的价格相对昂贵，但是它的读写速度比 IDE 硬盘的速度要快。对这两种硬盘的比较如表 1-2 所示。

表 1-2 IDE 硬盘与 SCSI 硬盘的比较

硬盘类型	速度	价格	机器可用的数量	使用场合	是否需要另加接口
IDE	慢	低	不多于4块（包括CDROM）	PC机	不需要
SCSI	快	高	可以更多，如16块	服务器	需加SCSI适配卡

一台计算机通常提供 4 个 IDE 接口，最多可以安装 4 块硬盘。通常 CD-ROM 驱动器需要占用一个 IDE 接口，因此，一台计算机一般最多可以安装 3 块硬盘。而事实上，最常见的情况是一台计算机安装一块硬盘。

一台计算机的 IDE 接口对应的硬盘名称与 Red Hat Linux 的设备名有表 1-2 的对应关系。如果系统多于一块硬盘时，它们的引导顺序如表 1-3 的第 3 列所示。

表 1-3 IDE 硬盘与 Red Hat Linux 设备名对应表

IDE硬盘的位置	Red Hat Linux 的设备名	引导顺序
第1主IDE	/dev/hda	1
第1从IDE	/dev/hdb	3
第2主IDE	/dev/hdc	2
第2从IDE	/dev/hdd	4

例如一台计算机装有两块硬盘，这块硬盘分别装在第 2 主 IDE 和第 1 从 IDE 接口处，这是系统将会从第 2 主对应的硬盘引导系统。装在第二主 IDE 接口的硬盘的设备名为 /dev/hda。

图 1-7 中显示的硬盘类型为 IDE 硬盘，容量大小为 3071MB，全部空闲。它被接在第 1 主 IDE 接口上，因此它的设备名称为 /dev/hda。笔者将以此为例讲述创建分区的过程。

SCSI 硬盘的设备名是 /dev/sd[a-z] 形式，/dev/sda 是第一块 SCSI 硬盘，/dev/sdc 是第 3 块 SCSI 硬盘，以此类推。

常见的硬盘分区类型是 Windows 的 NTFS 和 FAT，以及 Linux 的 ext3 和 swap 分区。对于一块硬盘分区的设备名为该硬盘的设备名后加一个数字表示。例如 IDE 硬盘 /dev/had 的分区名称具有 /dev/hda[N] 形式，其中 N 是一个数字，例如 /dev/hda2 和 /dev/hda3。同样，SCSI 的硬盘 /dev/sda 分区的设备名分别 /dev/sda1，/dev/sda2 等等。

Disk Druid 是一个提供了友好接口的分区工具，用户可以利用安装程序提供的 Disk Druid 工具进行分区的创建、删除和修改。如果用户已经在以前的 OS（如 Windows 2000/XP）分过区，可以在此删除这些分区，或者使用未用的磁盘空间建立双重引导的 Red Hat Linux 系统。

用户应仔细考虑如何分区。通常情况下，需要 3 个分区：根分区、交换分区和引导分区。

- （根）分区。存放系统中除内核文件以外的所有的文件，包括系统文件，所有的应用程序和用户的主目录。容量越大越好，建议一般不小于 3GB，分区类型为 ext3。
- 交换分区。Linux 系统在运行时，当内存容量不能满足需要时，可将内存的一些

数据缓存在这个分区中,以便使内存供其他进程使用,其大小为物理内存的两倍,分区类型为 Linux swap,如果运行大型系统,则需要的交换分区的容量会更大。例如安装和运行 Oracle 数据库,交换分区的最小需求是 1GB。

- 引导分区。存放 Linux 内核和一些启动的配置文件,这个分区并不需要太大,只需要 100MB 就足够了。

**注意:** 在 Linux 系统中,要使用一个分区,除了 swap 分区以外,必须要有一个挂载点(也称 mount point)。一般将挂载点对应的分区称作“xxx 分区”,其中 xxx 为挂载点的名称。例如挂载点/对应的分区称为根分区, /boot 对应的分区被称为 boot 分区。

在图 1-7 所示的 Disk Druid 图形接口上,可以单击“新建”按钮新建一个分区。单击“编辑”按钮来更改一个已经创建的分区设置,例如容量,分区类型和挂载点。单击“删除”按钮,删除一个已经存在的分区,释放该分区的磁盘空间。

假定要创建一个大小为 80MB 的 /boot 分区,首先单击“新建”按钮,将会弹出一个“添加分区”对话框。然后从“挂载点”下拉菜单选择 /boot,保留系统默认的 ext3 的文件系统,在“大小框”中输入 80,如图 1-8 所示。然后单击“确定”按钮。

这样 /boot 分区就创建了,如图 1-9 所示。

使用上述方法创建/(根)分区和 swap 分区,其中 swap 分区的大小为 256MB,根分区的大小为 2737MB (=3071-80-256MB)。建好的分区如图 1-10 所示。

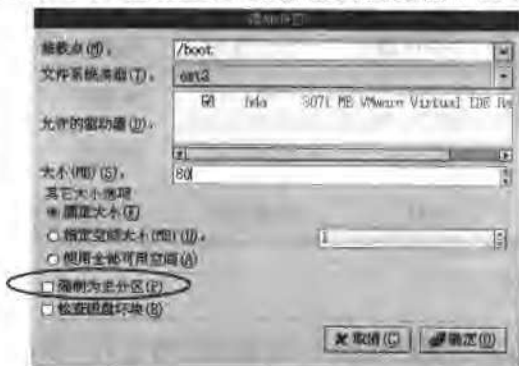


图 1-8 创建 /boot 分区



图 1-9 创建 /boot 分区后的结果

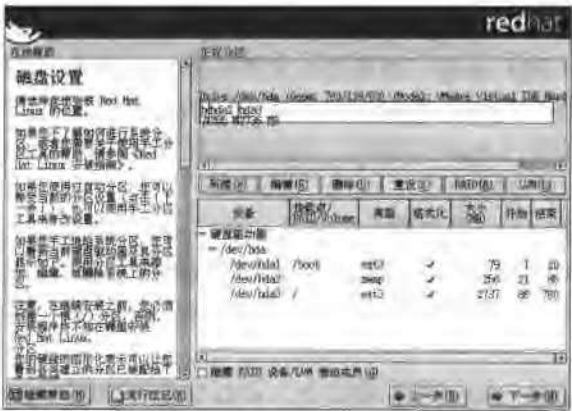


图 1-10 创建的 3 个分区

注意：新创建的分区需要格式化才能被 Linux 使用。从图 1-10 可以看出这三个分区的“格式化”标志都已选中，表明都将被格式化。

现在将创建的 3 个分区的信息归纳为表 1-4。该表分别说明了分区的名、挂接点、分区类型、功能、大小和分区的设备名。在图 1-10 的界面中单击“下一步”按钮，进入到“引导装载程序配置”页。

表 1-4 创建的分区的信息汇总

分区名	挂接点	分区类型	存放信息	大小	分区的设备名
引导分区	/boot	ext3	引导文件	80MB	/dev/hda1
交换分区		Swap	虚拟存储	256MB	/dev/hda2
根分区	/	ext3	各种文件	2737MB	/dev/hda3

1.3.6 配置引导装载程序

Red Hat Linux 除了可以引导自身之外，还提供了两套引导装载程序，它们分别是 LILO 和 GRUB，它们不仅能够引导 Linux 操作系统，还可以引导 Windows 和其他操作系统，这对于在同一台机器中同时安装多个操作系统，或者同一个 Linux 系统，同时使用多个版本的内核都是非常有用的。

1. LILO 多重引导管理器

Linux 引导可以采用多种方式，其加载器 LILO (Linux Loader, Linux 装载器) 是一个灵活的引导加载程序，与其他常用的引导加载程序相比，LILO 引导方式显得更具有艺术性，对其深入的理解，将有助于我们方便地处理多重系统、网络引导、大硬盘及大内存等诸多棘手的问题。

LILO 的引导机制。众所周知，计算机的最初启动是由 BIOS 控制的，在对一些硬件 (如内存、键盘等) 初始化之后，它会试图加载硬盘的主引导记录 (MBR，硬盘的第一个扇区) 或软盘的引导扇区。MBR 可通过两种方式运行，其一是定位到活动分区并加载相应的引导

扇区, 然后由引导扇区完成该分区内操作系统的基本组件的加载; 其二是直接从一指定分区中加载信息, 并通过它装入任一分区的操作系统, 诸如 LILO、OS/2 boot loader 及 Partition Magic 等引导加载程序都可以配置成这种方式。

软盘的引导扇区 (位于软盘上的第一个分区) 相当于硬盘活动分区的引导扇区, 它通常用于装入软盘上的操作系统。由此可见, 只要把 LILO 安装在 MBR、活动分区或者引导软盘上, 就能接管计算机的控制权, 然后由 LILO 完成后继的引导过程。LILO 中建立一个引导表地址编码, 借此它的引导程序就能定位到 Linux 的内核文件, 这种地址编码既可以按照柱面/磁头/扇区模式, 又可以采用 LBA 的线性块号模式, 因此, 即使对大多数 SCSI 控制硬盘, LILO 也能运转良好。

当 LILO 定位到配置文件后, 经过预引导过程, 系统将允许选择引导不同的操作系统或者不同的内核配置, 如果选择了引导 Linux, 还可以直接传递参数到系统内核。

- 与其他系统的引导加载程序相比, LILO 具有更大的灵活性, 其引导方式也更丰富;
- 当 LILO 被安装在硬盘的 MBR、活动分区或引导软盘上时, LILO 作为原引导程序的替身, 它能引导任一硬盘任一分区上的 Linux 和其他操作系统;
- 除了引导扇区, 它没有任何隐含文件, 也不需要使用特定的分区, 它的配置文件可以存放在任何分区、甚至可以存放在与 Linux 毫不相干的 DOS 分区的某个子目录下;
- 它能引导几个不同的内核配置, 甚至是几个不同的内核;
- 它能引导同一机程序上的多个 Linux 版本;
- 它能从网络上引导 Linux。

## 2. GRUB 多重启动管理器

GRUB 是一个多重启动管理器。GRUB 是 GRand Unified Bootloader 的缩写, 它可以在多个操作系统共存时, 选择引导哪个系统。它可以引导的操作系统包括 Linux, FreeBSD, Solaris, NetBSD, BeOSi, OS/2, Windows95/98, Windows NT 和 Windows2000/2003。它可以载入操作系统的内核, 并且初始化操作系统 (如 Linux, FreeBSD), 或者把引导权交给操作系统 (如 Windows 98) 来完成引导。

GRUB 可以代替 LILO 来完成对 Linux 的引导, 特别适用于 linux 与其他操作系统共存的情况, GRUB 与 LILO 相比, GRUB 有以下几个特点:

- **支持大硬盘。**现在大多数 Linux 发行版本的 LILO 都有同样的一个问题: 根分区 (/boot 分区) 不能分在超过 1024 柱面的地方, 一般位于 8.4G 左右的地方, 否则 LILO 不能被安装, 或者安装后不能正确引导系统。GRUB 不会出现这种情况, 只要安装时大硬盘是在 LBA 模式下, GRUB 就可以引导根分区在 8GB 空间以外分区上的 Linux。
- **支持开机画面。**GRUB 支持在引导开机的同时显示一个开机画面。这样可以制作自己的个性化开机画面; 对于 PC 厂商, 这样可以在开机时显示电脑的一些信息和厂商的标志等。GRUB 支持 640×480, 800×600, 1024×768 三种模式的开机画面, 而且可以自动选择最佳模式, 与 Windows 操作系统的 320×400 分辨率的开机画面不可同日而语。

- **两种执行模式。**GRUB 不但可以通过配置文件进行例行的引导,还可以在引导时动态改变引导时的参数,还支持动态加载各种设备的驱动程序。例如用户在 Linux 下编译了一个新的内核,但不能确定它是否能够工作,就可以在引导时动态地改变 GRUB 的参数,尝试装载这个新的核心。GRUB 的命令行有非常强大的功能,而且支持如 `bash` 或 `doskey` 一样的历史功能,可以用上下方向键来寻找以前使用过的命令。
- **菜单式选择。**GRUB 使用一个菜单来选择不同的系统进行引导。用户还可以自己来配置各种参数,如延迟时间、默认操作系统等。
- **分区位置改变后不必重新配置。**LILO 是通过读取硬盘上的绝对扇区来装入操作系统的,因此每次分区改变都必须重新配置 LILO,例如你用 PQ Magic 调整了分区的大小,那么在重新配置好之前,LILO 就不能引导这个分区的操作系统了。而 GRUB 是通过文件系统直接把核心读取到内存,因此只要操作系统核心的路径没有改变,GRUB 就可以来引导系统。除此之外,GRUB 还有许多非常强大的功能,例如支持多种外部设备,动态装载操作系统内核,甚至可以通过网络装载操作系统。
- **系统内核。**GRUB 支持多种文件系统,支持多种可执行文件格式,支持自动解压,还可以引导不支持多重引导的操作系统。

由以上的分析可知,LILO 虽然也比较灵活,但是使用 GRUB 引导系统会更加方便,提供的功能也更多,而且配置起来更加方便。因此笔者建议读者在安装 Linux 时选择 GRUB 作为多重引导管理器。

**注意:** 笔者建议读者使用 GRUB,因为它更容易配置和使用,而且它的功能十分强大。它是 Red Hat Linux 9 安装的默认引导装载程序。

Red Hat Linux 9 的引导装载程序的配置如图 1-11 所示。在图 1-11 所示的画面上,通过单击“改变引导装载程序”按钮来改变 Linux 使用的引导装载程序,弹出的对话框如图 1-12 所示。

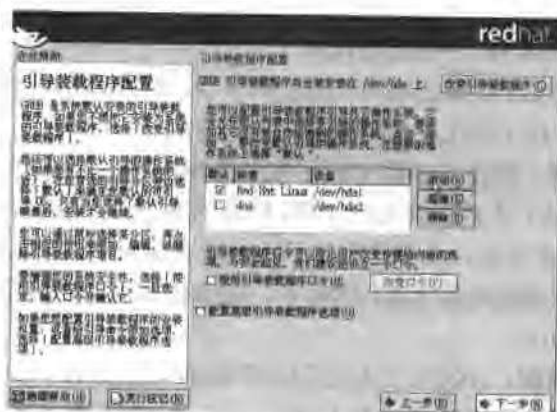


图 1-11 Red Hat Linux 引导装载程序配置

图 1-12 中有 3 种选项,如果系统中有多多个操作系统,或者需要使用不同的 Linux 内核



时，在安装时最好选择一个引导装载程序。如果你需要使用其他操作系统或者第三方提供的引导装载程序，你也可以选择“不要安装引导装载程序”。在此使用默认选项“以 GRUB 为引导装载程序”，然后单击“确定”按钮。

注意：笔者建议即使计划使用非 Red Hat Linux 9 提供的引导装载程序，在此也选择 GRUB 为引导装载程序，因为这样会使系统具有更大的灵活性。

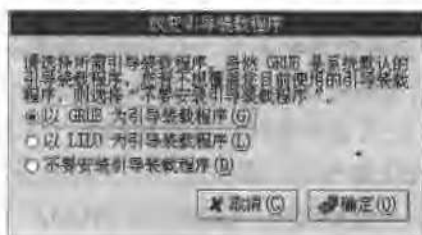


图 1-12 改变引导装载程序

图 1-11 列出了引导装载程序将要引导的操作系统列表。这个列表中的每一个引导项包含 3 个部分：“默认”栏表示哪一项是默认启动的；“标签”栏表示该启动项在启动菜单中显示的提示，用户可以根据自己的爱好随便修改；“设备”栏表示了该启动项从哪个分区获得引导数据。用户可以通过单击“添加”按钮来添加一个新的引导项，用于引导一个新的操作系统。通过单击“编辑”按钮来修改某一个项目的设置，在这里要改变的主要是默认引导标志和引导菜单所显示的“标签”。用户也可以通过单击“删除”按钮来删除选中的引导项。

选中图 1-11 中的“使用引导装载程序口令”复选框，可以给引导装载程序设置口令。

图 1-11 有一个“配置高级引导装载选项”复选框，如果选中该复选框，下一页将会出现图 1-13 的“高级引导程序配置”页。

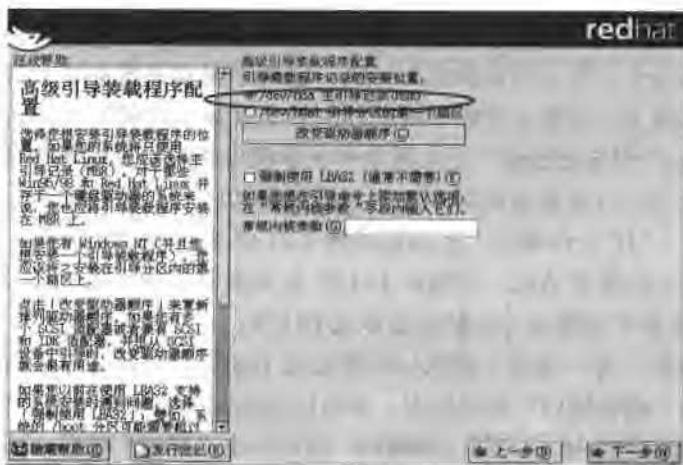


图 1-13 高级引导装载程序配置

可以将引导装载程序安装在主引导记录 (MBR) 上，也可以将它放在 Red Hat Linux 9



引导分区的第一扇区上。默认选择被安装在主引导记录。一般情况下, 不要改动这个设置, 否则可能导致安装完成后无法引导 Linux。

**注意:** 如果将引导程序安装在主引导记录, 之后又需要安装低版本的 Windows 操作系统 (例如 Windows 98/Windows NT), 这些 Windows 操作系统将会重写硬盘的主引导记录, 使系统使用 Windows 的引导装载程序, 或者需要频繁安装多个操作系统。这样将会导致无法引导 Linux, 为了避免这种情况发生, 可以将 Red Hat Linux 9 的引导装载程序安装在“引导分区的第一个扇区中”, 如果先择了此选项, 还需要在前面的分区进行一定的限制, 同时 Linux 安装完成后, 也不能直接引导 Linux, 还需要对系统进一步的配置。有关使用 Windows 的引导管理程序引导 Linux 的详细内容, 请参阅 1.5 节的内容。

除非有特殊需要, 否则不需要选中“强制使用 LBA32”复选框, 也不需要“添加内核参数”。

当以上设置完成后, 单击“下一步”按钮, 进入“网络设置”页。

### 1.3.7 配置 TCP/IP 网络

当引导装载程序设置完成之后, 单击“下一步”按钮, 将会进入如图 1-14 所示的“网络设置”页进行网络设置。网络设置也是安装中很重要的一步, 否则需要在系统安装完成后再进行设置。如果系统中没有安装网络设备, 将不会出现图 1-14 所示的网络设置屏幕。

安装程序会自动检测到系统安装的任何网络设备, 并把它们显示在“网络设备”列表中。选定网络设备后, 单击“编辑”选项卡。从弹出的“编辑接口”屏幕上, 可以选择通过 DHCP 来配置网络设备的 IP 地址和子网掩码 (若没有选择 DHCP 则需要手工配置), 通常选择“引导时激活”该设备。如果选择了“引导时激活”, 系统中的网络接口就会在引导时被启动。如果你没有 DHCP 客户的访问权, 或者不能肯定这里该提供什么信息, 请与网络管理员联系。

从图 1-14 可以看出, 椭圆框中所列出的是这台计算机上可用的网络设备, 本例的网卡列表只包含一块网卡, 网卡的设备名为 `eth0`, 它表示是该系统的第一块网卡。如果系统有两块和两块以上网卡, 该网卡列表中将会列出多个设备。分别是 `eth0`, `eth1`, `eth2` 等。对每一个网络设备, “引导时激活”栏表示是否在系统启动时启用这块网卡。如果选中该复选框, 表示系统引导时自动激活该网卡, 否则系统启动之后, 不会激活, 用户可以根据需要手工激活该网卡。“IP/子网掩码”栏列出该网卡的 IP 地址和子网掩码。如果该网卡被设置成 DHCP (动态主机配置协议, 它能从 DHCP 服务器中自动或者一个 IP 地址), 在这里不需要指定 IP 地址和子网掩码, 在此将显示成 DHCP。

在系统安装时, 每一块网卡被默认配置均为 DHCP, 在此可以单击“编辑”按钮进行修改, 在弹出的“编辑接口”对话框中, 你可以将该网卡设置为静态 IP 地址, 这时你需要输入与该网卡对应的 IP 地址和子网掩码, 然后单击“确定”按钮; 或者设置启动时是否激活该接口, 如图 1-15 所示。

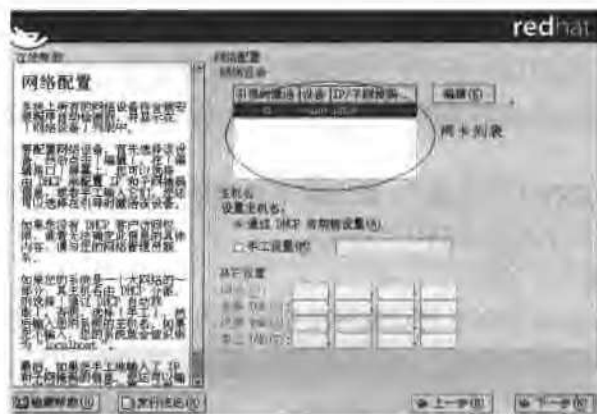


图 1-14 网络设置

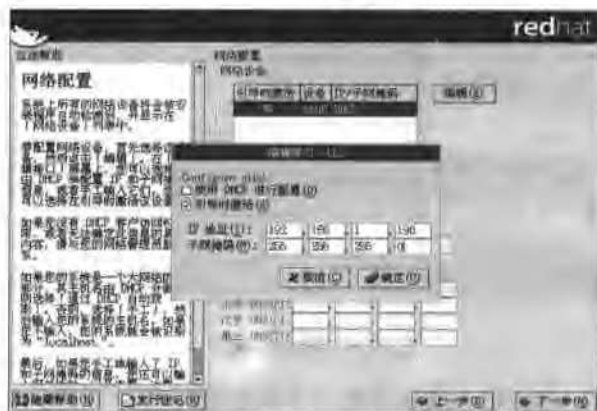


图 1-15 网络接口的编辑

### 1.3.8 设置主机名

如果主机的网络设备有一个全限定的主机名，则可以选择使 DHCP（动态主机配置协议）自动检测它，或者在提供的字段中直接进行手工输入。如果采用的是手工输入 IP 地址和子网掩码信息，并且需要通过网络访问该子网以外的主机时，用户还必须提供网关、主要、次要和第三 DNS 服务器的 IP 地址。其中次要和第三 DNS 是可选的。

**注意：**即使你的计算机不属于任何网络，你也可以为你的系统输入一个主机名。如果你不借助这个机会输入一个名称，你的系统将会被称为 localhost。要在安装结束后改变你的网络配置，使用“网络管理工具”。在 shell 提示符下键入 `redhat-config-network` 命令来启动“网络管理工具”。如果你不是根用户，它会提示你输入根口令后再继续。

### 1.3.9 防火墙配置

Red Hat Linux 9 为增加系统安全性提供了防火墙保护。防火墙存在于你的计算机和网络之间，用来判定网络中的远程用户是否有权来访问你的计算机上的哪些资源。一个正确

配置的防火墙可以极大地增强系统的安全性。

如图 1-16 所示是 Red Hat Linux 9 安装时防火墙的配置页。

你需要根据实际情况为你的系统选择恰当的安全级别。Red Hat Linux 提供了 3 个安全级别。

(1) 高级。如果选择了“高级”，系统就不会接受那些没有被你具体指定的连接（除了默认设置外）。只有以下连接是默认允许的：

- DNS（域名服务）回应。
- DHCP。任何使用 DHCP 协议的网络接口都可以被相应地配置。

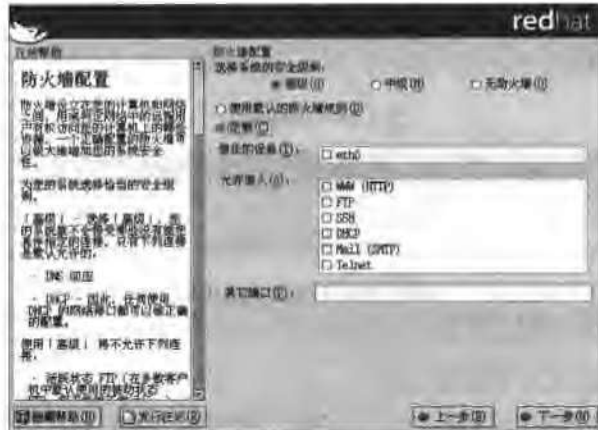


图 1-16 防火墙配置

如果选择“高级”，你的防火墙将不允许下列连接：

- 活跃状态 FTP（在多数客户机中默认使用的被动状态，FTP 应该能够正常运行）。
- RealAudio。
- 远程 X 窗口系统（使用 X 协议）的客户机。

如果要把系统连接到互联网上，但是并不打算运行服务器，这是最安全的选择。如果需要额外的服务，你可以选择“定制”来具体指定允许通过防火墙的服务。

**注意：**如果你在安装中选择设置了中级或高级防火墙，网络验证方法（NIS 和 LDAP）将行不通。

(2) 中级。如果你选择了“中级”，你的防火墙将不准你的系统访问某些资源。访问下列资源是默认不允许的：

- 低于 1023 的端口。这些端口主要被一些系统服务所使用，如 FTP、SSH、telnet、HTTP、和 NIS。
- NFS 服务器端口（2049）。在远程服务器和本地客户机上，NFS 都已被禁用。
- 为远程 X 客户机设立的本地 X 窗口系统显示。
- X 字体服务器端口（xfs 不在网络中监听；它在字体服务器中被默认禁用）。

如果用户希望允许提供 RealAudio 之类网络服务，但仍要堵塞到普通系统服务的访问，选择“中级”。可以选择“定制”来具体指定哪些服务能穿过防火墙。

**注意：**如果你在安装中选择设置了中级或高级防火墙，网络验证方法（NIS 和 LDAP）

将不可用。

(3) 无防火墙。无防火墙给予完全访问权，并不做任何安全检查，安全检查是对某些服务的禁用。建议只有在一个可信任的网络（非互联网）中运行时，或者想稍后再进行详细的防火墙配置时才选此项。

选择“定制”单选按钮来添加信任的设备或允许其他的进入接口。

(4) 信任的设备。选择“信任的设备”中的任何一个，将会允许你的系统接受来自这一设备的全部通信；它不受防火墙规则的限制。例如，如果机器在一个局域网中，但是通过 PPP 协议（一种 Modem 使用的拨号协议）拨号连接到互联网，这时可以选择 eth0 为信任设备，而后所有来自局域网的通信将会被允许。把 eth0 选为“信任的”意味着所有这个以太网内的通信都是被允许的，但是 ppp0 接口仍旧有防火墙限制。如果想限制某一接口上的通信，请不要选择它。

建议你不要将连接到互联网的公共网络上的设备设置为“信任的设备”。

(5) 允许进入。启用这些选项将允许指定的服务穿过防火墙。这些服务包括：

注意：在工作站类型安装中，大多数这类服务在系统内没有被安装。

- DHCP（动态主机配置协议）。如果允许提供 DHCP 查询和回应服务，将会允许任何使用 DHCP 来判定其 IP 地址的网络接口。
- SSH。SSH（Secure SHell，安全 shell）是用来在远程机器上登录以及执行命令的一组工具。如果你打算使用 SSH 工具通过防火墙来访问你的机器，就启用该选项。这时需要安装 openssh server 软件包，以便使用 SSH 工具来远程访问你的机器。
- Telnet。Telnet 是用来在远程机器上登录的协议。Telnet 通信是不加密的，几乎没有提供任何防止来自“网络嗅探”之类的安全措施。建议不要允许进入 Telnet 访问。如果你想允许进入的 Telnet 访问，需要安装 telnet server 软件包。
- WWW (HTTP)。HTTP 协议被 Apache（以及其他万维网服务器）用来进行网页服务。如果打算向公众开放你的万维网服务器，请启用该选项。如果仅仅查看本地网页或开发网页，则不需要启用该选项。如果打算提供网页服务的话，则需要在该机器上安装 httpd（Apache Web Server）软件包。启用 WWW (HTTP)将不会为 HTTPS（安全的 HTTP 协议）打开一个端口。要启用 HTTPS，需要在“其他端口”字段内注明。
- 邮件（SMTP）。如果需要允许远程主机直接连接到你的机器来发送邮件，则启用该选项。如果想从你的 ISP 服务器中收取 POP3 或 IMAP 邮件，或者使用诸如 fetchmail 之类的工具，则不需要启用该选项。如果不正确配置 SMTP 服务器，会导致远程机器使用你的服务器发送垃圾邮件。
- FTP。FTP 协议是用于在网络机器间传输文件的协议。如果你打算使你的 FTP 服务器可被公开使用，则启用该选项。你需要安装 vsftpd 软件包才能利用该选项。
- 其他端口。用户可以允许到这里没有列出需要其他机器访问本机器的其他端口，方法是在“其他端口”字段内把它们列出。格式为：端口:协议。例如，如果你想允许 IMAP 通过你的防火墙，则可以指定 imap:tcp。用户还可以具体指定端口，要允许 UDP 包在端口 57 通过防火墙，输入 57:udp。要指定多个端口，用逗号将它们隔开。

注意：要在安装完毕后改变你的安全级别配置，使用“安全级别配置工具”。在 shell 提示下键入 `redhat-config-securitylevel` 命令来启动“安全级别配置工具”。如果你不是根用户，它会提示你输入根口令后再继续。

### 1.3.10 语言支持的选择

在 Linux 系统上可以同时安装和支持多种语言。用户必须选择一种语言作为默认语言。当安装结束后，系统中将会使用默认语言。如果用户选择并安装了其他语言，则可以在安装后改变其默认语言。

如果只打算在系统上使用一种语言，只选择那种语言将会节省大量磁盘空间。默认语言是在安装中选择要使用的语言。

语言选择如图 1-17 所示。在图中选择了简体中文作为默认的语言，而且没有选择安装其他语言。

注意：如果你只选择了一种语言，当安装完成后，你将只能使用该种语言。



图 1-17 语言支持选择

要在系统上使用多种语言，需要具体指定要安装的语言，或者选择在 Red Hat Linux 系统上安装所有可用的语言。使用“重设”按钮来取消选择。重设会还原到默认设置，只有被选中的语言才会被安装到系统。

注意：在 shell 提示下键入 `redhat-config-language` 命令来启动“语言配置工具”。如果你不是根用户，它会提示你输入根口令后再继续。

### 1.3.11 时区配置

你可以通过选择你的计算机的物理位置，或者指定你的时区和通用协调时间（UTC）的偏移（UTC 偏移）来设置时区，如图 1-18 所示。

请注意屏幕上端的两个标签。第一个标签允许用户按位置来配置时区。

在互动地图上，用户可以单击一个用黄点标记的指定城市，一个红色的 X 会出现来标明用户的选择。除此之外，用户也可以在一个列表中滚动并选择一个时区。

第二个标签允许用户指定 UTC 偏移。这个标签显示了一个可从中选择的偏移列表，以及设置夏时制的选项。



图 1-18 配置时区

在“位置”和“UTC 偏移”这两个标签上，都可以选择“系统时钟使用 UTC”。如果用户知道系统应该设置到 UTC，则选择此复选框。

**注意：**要在安装完成后改变时区配置，请使用“时间和日期属性工具”。

在 shell 提示下输入 `redhat-config-date` 命令来启动“时间和日期属性工具”。

如果你不是根用户，它会提示你输入根口令后再继续。

要作为基于文本的程序来运行“时间和日期属性工具”，使用 `timeconfig` 命令。

### 1.3.12 设置根口令

设置根账号口令是安装过程中最重要的步骤之一，如图 1-19 所示。Linux 系统中的根账号与用 Windows NT 机器上的管理员账号 (administrator) 类似。根账号被用来安装软件包，升级 RPM，以及执行系统维护工作。作为根用户登录，可使用户对系统有完全的控制权。

**注意：**根用户（又称超级用户）对整个系统有完全的控制权；正因为如此，请最好只有在执行系统维护或管理时才以根用户身份登录。



图 1-19 设置根口令

通常情况下，只有在进行系统管理时才使用根账号，而创建一个非根账号来做日常工作。若用户需要快速管理某项事务时，用 `su -` 命令暂时登录为根用户，管理结束后，再使用 `exit` 命令返回到普通用户身份。遵循这些最基本的原则，将会减少因输入错误或不正确的命令而破坏 Linux 系统的机会。

**注意：**要临时成为根用户，在终端机窗口的 shell 提示下输入 `su -`，并按 Enter 键，然后，输入根口令并按 Enter 键。

安装程序会提示为系统设置一个根口令。必须输入一个根口令。如果根口令没有被输入，安装程序将不会继续安装。

根口令至少包括 6 个字符，输入的口令不会在屏幕上显示。你必须重复输入两次口令，如果两个口令不匹配，安装程序将会让你重新输入根用户口令。

根用户口令一般被设为可以记住，但又不容易被别人猜到的组合。你的名字、电话号码、password、root、123456、以及 111111 等都是典型的不安全口令。好的口令混合使用数字、大小写字母和一些特殊符号，并且不包含任何词典中的现成词汇，如 Aard387vark 或 xyl&\*c,L]。请记住，口令是区分大小写的。如果用笔记下你的根用户口令，请将它保存在一个安全的地方。然而，笔者建议不要记录创建的任何用户口令。而且口令长度至少为 12 个字符。

**注意：**不要使用本书提供的任何示例口令。使用其中任何一个，都可能导致安全风险。

要在安装后改变你的根口令，使用“根口令配置工具”。在 shell 提示下键入 `redhat-config-rootpassword` 命令来启动“根口令配置工具”。如果你不是根用户，它会提示你输入根口令后再继续。

根口令是用于 Red Hat Linux 系统的管理员口令。应该只在需要进行系统维护时才登录为根用户。根账号不受任何对普通用户限制的影响，因此，作为根用户所做的任何改变将会影响到整个系统。

### 1.3.13 验证配置

如果不必设置网络口令的话，可以跳过这一部分。如果不知道是否需要设置，请向系统管理员咨询。

除非正在设置 NIS 验证，用户会注意到只有 MD5 和“屏蔽”口令可以被选（参见图 1-20）。笔者推荐同时选中 MD5 和“屏蔽”这两个选项，以便使该系统更加安全。

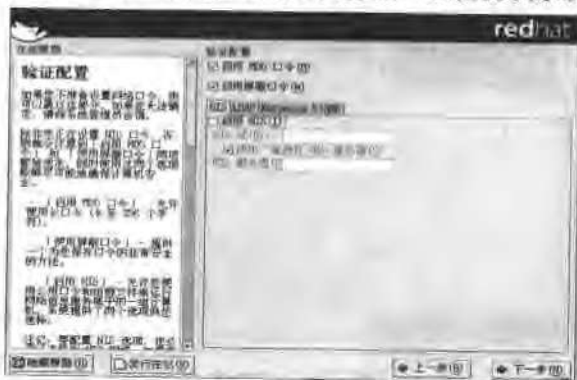


图 1-20 验证配置



要配置 NIS 选项，必须确认已连接到 NIS 网络。如果不能肯定是否已连接到 NIS 网络，请向系统管理员咨询。

- “启用 MD5 口令”。允许使用长口令（长至 256 个字符），而不是标准的 8 个字符或更少。
- “启用屏蔽口令”。提供一个保存口令的安全方法。口令被贮存在只能够被根用户读取的/etc/shadow 文件中。
- “启用 NIS”。允许系统使用一个公用口令和组文件来运行位于同一网络信息服务域内的一组计算机。你可以从下列选项中选择：
  - NIS 域。允许用户指定系统所属的域或计算机组。
  - 使用广播来寻找 NIS 服务器。允许系统在所处的局域网中广播消息来寻找一个可用的 NIS 服务器。
  - NIS 服务器。使计算机使用一个指定的 NIS 服务器，而不是在局域网中广播请求任何可用的服务器。

**注意：**如果在安装过程中选择了中级或高级防火墙配置，网络验证方法（NIS 和 LDAP）将不可用。

- “启用 LDAP”。告诉计算机使用 LDAP 做部分或全部验证。LDAP 合并你的机构中的某类信息。例如，你的机构中所有不同的用户列表可以被并入一个 LDAP 目录。你可以从下列选项中进行选择：
  - “LDAP 服务器”。允许访问一个运行 LDAP 协议的指定服务器（通过提供 IP 地址来完成）。
  - LDAP DN。允许你用识别名称（DN）来查找用户信息。
  - “使用 TLS（传输层安全）查寻”。该选项允许 LDAP 在验证前向 LDAP 服务器发送加密的用户名及口令。
- “启用 Kerberos”。Kerberos 是提供网络验证服务的一种安全系统。可供选择的选项有 3 个：
  - “领域”。该选项允许系统访问一个使用 Kerberos 的网络，这个网络可以由一个或多个服务器（又称 KDC），以及客户机（可能很多）组成。
  - KDC。该选项允许系统访问密钥分发中心（KDC）。KDC 是一个分配 Kerberos 门票的机器（有时也称“门票许可服务器”或 TGS）。
  - “管理服务器”。该选项允许 Linux 系统访问一个运行 kadmind 的服务器。
- “启用 SMB 验证”。用来设置 PAM 以便使用 SMB 服务器来验证用户。用户必须在此提供如下的两项信息：
  - “SMB 服务器”。指定你的工作站为了验证所要连接的 SMB 服务器。
  - “SMB 工作组”。指定配置的 SMB 服务器所在的工作组。

**注意：**要在安装结束后改变你的验证配置，请使用 authconfig 命令。

在 shell 提示下键入 authconfig 命令。如果你不是根用户，它会提示你输入根口令后再继续。



### 1.3.14 软件包组的选择

当分区被选定并按配置格式化后，用户便可以选择要安装的软件包了。

除非选择的是定制安装，安装程序将会自动为你选择大多数软件包。例如，如果用户要执行个人桌面安装，它将自动安装 GNOME 桌面、办公套件 OpenOffice、万维网浏览器 Mozilla、电子邮件套件 Evolution、即时消息、声音、视频程序和游戏。

要选择单个软件包，选择“定制要安装的软件包集合”复选框。可以选择根据功能归类的软件包组（譬如，“X 窗口系统”、“编辑器”）、单个软件包或者两者的组合。要选择一个软件包组，单击它旁边的复选框（参见图 1-21）。

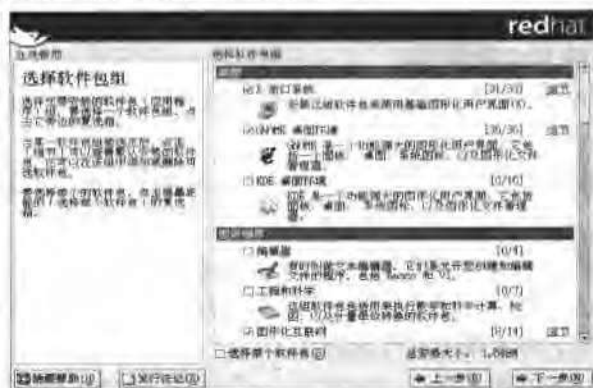


图 1-21 选择软件包组

在图 1-21 中选择想要安装每个组件。在定制安装中选择“全部”检查框（在组件列表的结尾处），会安装包括在 Red Hat Linux 中的所有软件包。

选定了软件包组后，单击该软件包组右边蓝色的“细节”链接，可以查看哪些软件包会被默认安装，还可以在该组中添加或删除可选软件包。一般来说，一个软件包组包含了一些必选的软件包，还包含了一些可选的软件包，用户可以根据需要选择这些可选的软件包。如图 1-22 中所示的“KDE 桌面环境”的细节对话框显示了包括 switchdesk 和 xinetd 在内的必选软件包。还包括 autorun、kdeaddons 和 kdesadmin 在内的可选软件包。



图 1-22 查看软件包细节和选择可选的软件包

要选择单个软件包，就要选中图 1-21 屏幕底部的“选择单个软件包”复选框。

### 1. 选择单个软件包

如图 1-23 所示，当选定了想要安装的软件包组后，用户可以用鼠标选择或取消选择单个软件包。



图 1-23 选择单个软件包

用户可以在“树形视图”或“平面视图”中查看单个软件包。

- “树形视图”。允许用户查看按照应用程序类型分类的软件包。
- “平面视图”。允许用户在屏幕右侧查看按字母顺序列表的全部软件包。

使用“树形视图”，会看到一个软件包组的列表。当扩展这个列表（通过双击软件包组名称旁的文件夹箭头）来挑选一个组时，该组内的软件包列表会出现在右侧的面板内。

“平面视图”允许用户在屏幕右侧看到按字母顺序列表的全部软件包。要按字母顺序给软件包排序，单击“软件包”标签。要按大小给软件包排序，单击“大小(MB)”标签。

要选择单个软件包，双击该软件包名称旁的复选框。

要了解某一指定软件包的说明信息，单击该软件包的名称。有关它的信息将会在屏幕底部出现。

用户可以选择或取消选择在某一组内列出的所有软件包，方法是，单击“选择全组”或“取消选择全组”按钮。

**注意：**某些软件包（如内核和某些库）是每个系统所必须的，它们不能被选择或被取消选择。这些基础软件包是被默认选定的。

### 2. 未解决的依赖关系

为了能够正常运行，许多软件包都要依赖安装在系统上的其他软件包。例如，许多图形化 Red Hat 系统管理工具需要 python 和 pythonlib 软件包。要确定系统已安装了所有需要的软件包以便正常运行，安装程序在每次安装或删除软件包时，都会检查这些软件包的依赖关系（dependencies）。

如果任何软件包需要一个没有选择安装的软件包，安装程序会把这些未解决的依赖关

系列出，从而给用户提供一个解决问题的机会（参见图 1-24）。

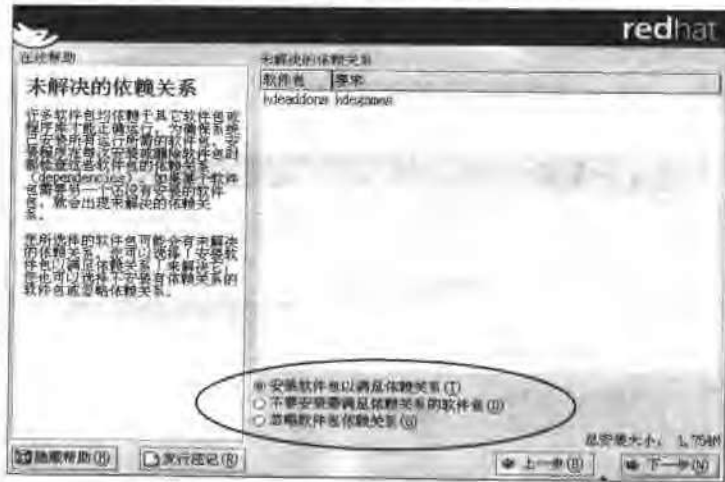


图 1-24 未解决的依赖关系

只有在已选择的软件包缺少其所需软件包时，“未解决的依赖关系”屏幕才会出现。在屏幕底部所缺软件包的列表之下，一个“安装软件包以满足依赖关系”复选框默认被选中。如果保持复选框的被选状态，安装程序将会把所有需要的软件包添加到已选软件包列表中，从而自动解决依赖关系。

如果不想安装支撑这些软件包的其他软件包，选择“不要安装需满足依赖关系的软件包”单选按钮。

如果只安装选择的软件包，而不解决依赖关系，选择“忽略软件包依赖关系”单选按钮。

**注意：**要在安装结束后，安装或删除软件包，使用“软件包管理工具”。

在 shell 提示下键入 `redhat-config-packages` 命令来启动“软件包管理工具”。

如果你不是根用户，它会提示你输入根口令后再继续。

### 1.3.15 准备安装

单击图 1-25 所示的“下一步”按钮，将会看到一个“即将安装”屏幕。

如果单击“下一步”按钮，将会开始真正的安装过程。当安装完成，并且重新引导系统后，一份完整的安装日志可在 `/root/install.log` 中找到，以备今后参考。单击“上一步”按钮，将可以逐步取消安装过程。

**注意：**如果由于某种原因，用户可能不希望继续安装进程，这是安全取消安装，并重新引导系统的最后机会。一旦单击“下一步”按钮，分区信息将会被写入，软件包将会被安装。如果想中止安装，用户现在就应该在任何信息被写入硬盘驱动器之前重新引导系统。



图 1-25 准备安装

如果真正的安装过程开始之后，又要取消安装进程，用户需要按计算机上的 Reset 按钮，或者使用 Ctrl+Alt+Del 组合键来重新启动机器。

### 1.3.16 安装软件包

到了这一步，在所有软件包被安装之前，用户不必进行任何操作（参见图 1-26）。安装的速度需要根据所选择的软件包的数量和计算机的速度而定。



图 1-26 软件包的安装过程

**注意：**由于 Red Hat Linux 9 的安装光盘为 3 张，因此如果使用光盘安装，在安装过程中会弹出一个对话框，提示你插入第二张或者第三张光盘，并且自动将前一张光盘从光驱中弹出，这时你将对应的安装盘插入光驱，并在弹出的对话框中单击“确定”按钮，继续安装过程。

### 1.3.17 创建引导盘

当所有的软件包安装结束之后，安装程序提示用户是否创建引导盘。如果希望创建一张引导盘，则需要在软盘驱动器内插入一张空白的、已格式化的磁盘（请参见图 1-27），然后单击“下一步”按钮。

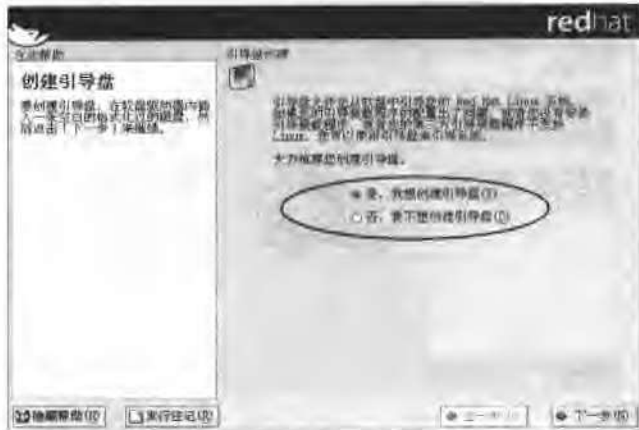


图 1-27 创建引导盘

强烈建议用户创建一张引导盘。如果由于某种原因，系统无法使用 GRUB、LILO、或第三方的引导装载程序来正确引导，那么引导盘将使用和能够正确地引导 Red Hat Linux 系统。

选择“是，我想创建引导盘”单选按钮，稍候片刻后，引导盘将会被创建，从磁盘驱动器中把它取出，并给它注以明确的标签，以备以后使用。

注意：用户也可以在安装结束后，使用 `mkbootdisk` 命令创建引导盘。有关 `mkbootdisk` 命令的详细信息，可以在 shell 提示符后执行 `man mkbootdisk` 命令。

如果不想创建引导盘，请在单击“下一步”按钮前选择“否，我不想创建引导盘”单选按钮。

如果使用引导盘来引导系统（而不是 GRUB 或 LILO），对内核做了改变（包括安装一个新内核）之后，都应该确保创建一张新的引导盘。

### 1.3.18 显卡配置

安装程序现在将会为用户提供一个视频卡列表，以供选择。

如果用户已经安装了 X 窗口系统软件包，那么现在就可以为系统配置一个 X 服务器。如果没有选择要安装 X 窗口系统软件包，请跳过本节。

如果视频卡没有出现在列表中（参见图 1-28），这说明 X 可能不支持它。然而，如果用户对视频卡有技术上的了解，可以选择“没列出的卡”，然后试图通过匹配视频卡的视频芯片与 X 服务器来配置它。

下一步，输入在视频卡上安装的视频内存数量。如果不能肯定显存的数量，请阅读视频卡的说明文档。选择多于可用的内存不会损坏视频卡，但是 X 服务器可能不会正确启动。

注意：Red Hat Linux 安装程序往往能检测显卡型号，用户通常可以接受其检测结果。

但是检测的显卡内存往往不是很准确，通常比实际的显示内存要少，为了更好的显示效果，用户需要在图 1-28 的“视频卡内存”列表中选择正确的值。

如果用户确信可以断定选择的数值是不正确的，可以单击“恢复原值”按钮来返回到建议的设置。



图 1-28 显卡配置

如果希望安装后再配置 X 或干脆不安装，还可以选择“跳过 X 配置”复选框。

如果在安装结束后改变你的 X 配置，需要使用 X 配置工具。

**注意：**在 shell 提示符下输入 `redhat-config-xfree86` 命令来启动“X 配置工具”。如果你不是根用户，它会提示你输入根口令后再继续。

### 1.3.19 显示器和定制

为了完成 X 配置，用户必须配置显示器并定制 X 设置。

#### 1. 配置显示器

安装程序会为用户提供一个显示器列表。用户可以使用安装程序检测到的显示器型号，也可以在显示器列表中选择。

**注意：**如果在一个带有 LCD 屏幕的便携电脑上安装 Red Hat Linux，用户应该选择最恰当的“通用”型号。

如果用户的显示器型号没有出现在列表中，那么可以选择最恰当的“通用”或者“检测过的显示器”型号。若选择“通用”的显示器，或者“检测过的显示器”，安装程序会建议一个水平和垂直频率范围。这些数值通常在用户的显示器说明文档中可以找到，也可从显示器的销售商或制造商处得到。请核对显示器文档，以确保这些数值已经被正确设置了。

**注意：**对于一个支持 75Hz 以上刷新频率和支持 1024 × 768 的显示器（无论是 CRT 还是 LCD 显示器），而 Red Hat Linux 又无法检测出它的型号，那么可以在“水平频率范围”输入框中输入 31.5 ~ 100，在“垂直频率范围”输入框中输入 50 ~ 100，这将会得到很好的显示效果。较新的显示器的刷新频率都大于或者等于 75Hz，分辨率至少为 1024 × 768。

除非你能确定你的显示器参数，否则不要选择与你的显示器相似的显示器。这么做可能会使你的显示器超频，而导致损坏。

如果用户的显示器在 Windows 操作系统中能支持 24 位真彩色，那么可以将“颜色深



度”设置成“真色（24 位）”，如果分辨率能达到  $1024 \times 768$ ，则将“屏幕分辨率”设置为  $1024 \times 768$ 。



图 1-29 设置显示器

**注意：**如果由于未能正确地设置显示器，使其分辨率和色彩深度太低，将会使得 X 系统的外观很难看，有时显示器显示出的图形会变形。

### 1.3.20 安装完成

当安装结束后，安装程序会提示用户做好重新引导系统的准备。如果安装介质（磁盘驱动器内的磁盘或光盘驱动器内的光盘）在重新引导时没有被自动弹出，请记住取出它们。

## 1.4 LILO 和 GRUB 配置

有关 LILO 和 GRUB 的基本原理请参见 1.3.6 节的描述。本节将主要讲述 LILO 和 GRUB 引导装载程序的配置文件。这是实际使用 LILO 和 GRUB 时应该着重考虑的问题。

### 1.4.1 LILO 的配置实例

如果 Linux 系统使用了 LILO 引导管理程序，那么使用 LILO 配置文件是使用和管理 Red Hat Linux 系统的一个部分。

配置 LILO 是通过编辑 `/etc/lilo.conf` 文件来实现的，这个文件的规则很简单，举例来说，使用 `cat` 命令查看 `/etc/lilo.conf`。

```
[globus@jiang /etc]$ cat /etc/lilo.conf
```

/etc/lilo.conf 文件的内容如下：

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

default=smp
```

```

image=/boot/bzImage
label=smp
read-only
root=/dev/hda1

image=/boot/vmlinuz-2.4.20-8
label=linux
read-only
root=/dev/hda1

other = /dev/hda3
label=dos
table=/dev/hda

```

首先是 `boot=/dev/hda`，这表示启动磁盘是第一张 IDE 硬盘（由于这个原因，通常不能从软盘来启动 Linux）。

接下来，`install=/boot/boot.b` 表示要将 `/boot/boot.b` 文件的内容写到引导记录中。这个文件在安装 Linux 的时候就已经创建。

```

prompt
timeout=50
default=smp

```

这三行表示系统启动时将会显示一个 lilo: 提示信息，并等待 5 秒钟，注意 `timeout` 的单位是 0.1 秒。如果 5 秒钟之内没有输入系统的名字，那么将使用默认引导 `default=smp`，引导标号为 `smp` 的系统。

接下来定义的是各个引导项目。首先定义了两个 Linux 引导项目。`image=/root/bzImage` 这行定义了启动时使用的内核映像，`label=smp` 定义了这个启动选择项的名称是 `smp`。在下面的几行是用 `bzImage` 启动系统时的一些附加参数，`read-only`，使用只读方式连接文件系统，由于系统启动的时候，要检测文件系统的完整性，因此要使用这个选项。当系统启动后，启动脚本会自动将它切换为 `read-write` 模式。

可以通过将 `default=smp` 中的 `smp` 修改成其他引导项的名称，来引导其他系统或者其他内核。在本例中可以使用 `linux` 和 `dos`。其中选择 `linux`，可以默认的启动 `vmlinuz-2.4.20-8` 版本的内核，使用 `dos`，可以默认的启动系统中的 Windows 系统。

紧接着定义了一个名叫 `linux` 的启动项，这个启动项用来启动位于 `/dev/hda1` 上的 Linux 内核。

```

image=/boot/vmlinuz-2.4.20-8
label=linux
read-only
root=/dev/hda1

```

接着，可以再定义一个名叫 `dos` 的启动项，这个启动项用来启动位于 `/dev/hda3` 上的



MS-DOS 分区的 Windows 系统。

```
other=/dev/hda3
label=dos
table=/dev/hda
```

注意：如果引导 Linux 的内核，要使用 image=内核文件名的方式定义引导项目，而需要引导 DOS 分区时，需要使用 other=分区名的方式定义项目。

其中在 image=启动项中，使用“root=分区名”的形式指定与内核对应的根分区的分区名。由于 DOS 不使用内核映像而是利用引导记录，系统必须能够找到分区表信息以便定位引导记录，在“other=分区名”启动项中，使用“table=分区表的位置”指定分区表的位置。table=/dev/hda 表示系统的 DOS 类型的分区表在/dev/hda 上。

在上述的例子中，第一和第二个引导项目都使用了 Linux 的内核，因此使用 image=方式定义，第三个引导项目，引导 DOS 分区，因此使用了 other=方式定义。

当完成了 lilo.conf 的设置工作后，为了使它的设置生效，需要执行/sbin/lilo 命令。会出现以下的信息：

```
[openlab] # /sbin/lilo
Added smp *
Added linux
Added dos
```

注意：用户可以模仿上述实例为新的内核添加引导项，也可以为多个 DOS 分区设置引导项。

1.4.2 LILO 的常用参数

LILO 的引导参数很多，在此只对一些比较重要的参数作一介绍，这些参数如表 1-5 所示。

表 1-5 LILO 重要的引导参数及其说明

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

续表

参数	说明
nfsaddr=	设定网络通讯所需的各种网络界面地址, 如无此参数, 则内核会试图用反向地址解析协议(RARP)或启动协定(BOOTP)找出这些参数, 其格式为: nfsaddr=〈客户端IP〉:〈服务端IP〉:〈网关IP〉:〈子网屏蔽〉:〈客户端名称〉:〈网络设备名〉:〈auto〉
image=	指定Linux的内核文件
delay=	设定引导第一个映像前的等待时间, 以0.1秒
Disk=	此参数为某一特殊的硬盘定义非标准参数
Append=	为内核传递一个可选的参数行, 其典型的应用是: 为不能完全由系统自动识别的硬盘指定参数, 如append="hd=64,32,202"
Label=	此参数为每个映像指定一个名字, 以供引导时选择
read-only	设定以只读方式挂入根文件系统, 用于文件系统一致性检查(fsck)
install=	安装一个指定文件作为新的引导扇区, 默认为/boot/boot.b
loader=	说明所使用的链加载程序(chain loader), 默认为/boot/chain.b, 如果不是从首硬盘或软盘启动, 那么此选项必须说明
table=	说明包含分区表的设备名, 如果此参数忽略, 引导加载程序将不能传递分区信息到已引导的操作系统。当此参数指向的分区表被修改时, 必须重新运行/sbin/lilo
init=	内核初始化时执行的程序, 通常过程为init、getty、rc和sh, 版本1.3.43以来的Linux内核能够执行/sbin/init说明的命令行, 若在引导过程中出现问题, 则可设置init=/bin/sh直接跳到Shell
ramdisk_start=	由于内核不能放在压缩的内存文件系统映像内, 为使内核映像能够和压缩的内存映像放在一张软盘内, 加入ramdisk_start=〈offset〉, 这样内核才开始执行
vga=	设置显示模式, 如80×50、132×44等
Other=	开始以提示启动其他独立的系统

### 1.4.3 使用 LILO 时遇到的问题及对策

安装和使用 LILO 通常会遇到以下问题:

- 将 LILO 安装在 MBR (主引导记录) 可能会导致一些问题。
- 建议把 LILO 安装在主分区上, 当然, 这时 linux 不能被安装在扩展分区内。这样, 重装 Windows 98 后就不会再毁坏 LILO。只需要设置一下活动(active)分区, 使用常见的 fdisk 即可使用。
- 建议把根分区分在硬盘的前面。如果有许多 linux 分区, 那么应该把根分区或者 /boot 分区放在硬盘的前面。否则 lilo 引导时就会受到 1024 柱面的限制。
- 安装 lilo 时不要先 mount 非 ext3 的分区。有一些硬盘可能还未被格式化(分好区后没有生成文件系统), 那么这时挂接上这些分区后, 再写 lilo 时就会失败。

注意: 将 LILO 安装在 MBR 后, 再安装低版本的 Windows 系统(如 Windows 98)将会破坏 LILO。必须重新配置 LILO, 否则就无法使用 LILO 和 Linux。因此最好将 LILO 安装在主分区上。

对 LILO 的每一次修改, 都不会自动生效, 为了使之生效, 必须以根用户身份

执行/sbin/lilo 命令。

使用 GRUB 作为引导装载程序或者利用 Windows 的引导装载程序会更方便，对 GRUB 的任何修改只需要编辑 GRUB 的配置文件即可，它会自动生效。如果使用 Windows 的装载程序，在重新安装了 Windows 系统后，不需要破坏 Linux 系统，这样就可以使这两种操作系统保持最大的独立性。

1.4.4 GRUB 的配置行

在 Red Hat Linux 9 中 grub 是默认安装的。

安装了 grub，开机后会出现一个菜单，列出所有的启动选项。如果设置了启动画面则会显示启动画面，按 Esc 键则可以取消启动画面显示菜单选项。此时按 e 键，可以编辑所选菜单项的启动命令，按 c 键进入命令行模式，用上下键选择菜单项，按回车键启动所选菜单项。

在命令行模式下，可以打入命令直接执行，例如可以敲入 poweroff 关闭计算机。按 Tab 键可以列出所有支持的命令。

GRUB 启动时会在/boot/grub/中寻找一个名为个 grub.conf 的配置文件，如果它找不到此文件，则直接进入命令行模式，而不进入菜单模式。

grub.conf 是一个文本文件，用户可以用任何文本编辑器来打开它。每一行代表一个配置命令，如果一行以“#”字符开始，则表明这一行为注释行，可以通过简单地增加或减少注释行来改变配置。表 1-6 所示为 GRUB 配置文件的配置行及其说明。

表 1-6 GRUB 配置文件的配置行及其说明

参数	说明
Timeout <i>second</i>	设定在等待 <code>second</code> 秒之后引导默认的操作系统。如果设置成-1，则 grub 会一直等待直到用户选择一个引导项为止
default <i>num</i>	默认启动第 <code>num+1</code> 行选项，也就说 <code>default=0</code> 则默认启动菜单第一行的操作系统， <code>default=1</code> 则启动第2行的操作系统，如此类推
splash <i>pathname/filename</i>	指定开机画面的文件所存放的路径和文件名，如 splash boot/logo/800×600×8.img 是指用在/boot/logo路径下的800×600.img文件作为开机画面
title <i>OSname_title</i>	后面的字符就是你在菜单项上所看见的选项，你可以写上操作系统的名字和描述，如用title Red Hat Linux 9, Single Mode 描述了这一选项是以单用户模式引导Red Hat Linux 9

1.4.5 GRUB 的配置实例

下面结合两个实例来说明 GRUB 引导选项的意义，以及如何修改配置文件以满足系统的引导需要。

实例 1

```
title Red Hat Linux, Default Mode
root (hd0,1) #hd0 是指第一张硬盘(主硬盘)
#(hd0,1) 是指第一张硬盘的第二个分区。
```

```
kernel /boot/vmlinuz vga=auto root=/dev/hda2
#kernel /boot/vmlinuz 是指 Linux 核心的路径在 /boot/vmlinuz 中。
#vga=auto 是设定显示模式。
#root=/dev/hda2 是指把第一张硬盘的第二个分区作为根挂载点 ("/")。
title Microsoft Windows
root (hd1,0) #root (hd1,0) 这指的是第二个硬盘 (从硬盘) 上第一个分区
chainloader (hd1,0)+1 #chainloader (hd1,0)+1 指首先装入一个扇区的数据
#然后把引导权交给它。
```

## 实例 2

```
default=0 #默认启动 Red Hat Linux 9
timeout=0 #立即启动系统, 不等待
splashimage=(hd0,2)/boot/grub/splash.xpm.gz
title Red Hat Linux 9 (2.4.20-8) #启动项显示标题
root (hd0,2) #根挂载点挂载的硬盘分区为 /dev/hda2
#即第一张硬盘的第三个分区
kernel /boot/vmlinuz-2.4.20-8 ro root=LABEL=/
#内核文件名为 /boot/ vmlinuz-2.4.20-8。
initrd /boot/initrd-2.4.20-8.img
title Windows XP
root noverify (hd0,0) # root (hd0,0) 这指的是第一张硬盘 (主硬盘) 上第一个分区
chainloader +1 #首先装入一个扇区的数据
#然后把引导权交给它。
```

## 1.5 使用 Windows 的引导管理器引导 Linux

对于桌面用户, 用户常常在一台 PC 机上同时安装 Windows 和 Linux 系统。用户可能根据需要会升级和重新安装 Windows 和 Linux 系统。使用 Linux 的引导装载程序存在以下不足:

- 如果使用 Linux 提供的 LILO 和 GRUB, 可能由于重装 Windows 系统时, Windows 重写硬盘的主引导记录 (MBR), 导致破坏 Linux 生成的主引导记录。从而无法重新启动 Linux 系统, 必须重新设置 Linux 的引导装载程序。
- 如果在 Windows 中使用 PQ Magic 等分区管理工具来修改分区, 如移动分区和改变分区的大小。这时也可能导致 Linux 的 LILO 和 GRUB 不能正常引导。
- 一些用户已经用惯了 Windows 系统, 对 Windows 的引导装载程序有偏爱, 而不习惯 Linux 的引导装载程序。
- LILO 有 1024 柱面的限制, 因此安装 Linux 操作系统时, 往往需要移动 Windows 分区, 或者改变 Windows 分区的大小, 以便将 Linux 移至 1024 柱面以内。一般的分区工具 (如 PQ Magic) 都容易导致分区的破坏, 从而丢失 Windows 分区的重要数据。

考虑到 Linux 和 Windows 将会长期共存一台计算机上的事实。为了保证这两个系统的相对独立性,又能够顺利启动这两个操作系统。用户可以使用 Windows 的引导装载器(OS Loader)引导 Windows 和 Linux 系统。

为了用 Windows 的引导装载器来引导 Linux, 用户安装 Linux 时的步骤如下:

- 首先安装 Windows 操作系统, 然后再安装 Linux。这样 Windows 的 OS Loader 就会安装到 Windows 的主分区。
- 将 Linux 的根分区或者/boot 分区设置成主分区。在安装 Red Hat Linux 9 的过程中, 将 Linux 的根分区或者/boot 分区设置成主分区。为了将根分区设置成主分区, 需要在图 1-8 中选中“强制为主分区”复选框。

**注意:** 在同一张 IDE 硬盘上, 主分区的数量不能超过 4 个, 一般来说, Windows 操作系统会占用两个主分区, 一个是主 DOS 分区, 用于引导 Windows 系统, 另一个是扩展 DOS 分区, 用于管理 Windows 使用的其他逻辑 DOS 盘。

这样 Linux 系统最多能建立两个主分区, 建议不要创建/boot 分区, 而将 Linux 的根分区和交换 (swap) 分区设置成主分区。这样一块硬盘上刚好会有 4 个主分区。

Linux 分区可以在 DOS 扩展分区中创建, 但是这样无法将 LILO 或者 GRUB 安装在这些分区的主引导记录上, 因为它无法被 BIOS 正常引导。

- 将 LILO 或者 GRUB 安装到主分区上, 而不是主引导记录上, 或者干脆不安装 Linux 的引导装载程序。用户可以在 1.3.6 节的图 1-12 中选择“不安装引导装载程序”单选按钮, 或者在图 1-13 中选中“引导分区的第一个扇区”复选框。

**注意:** 如果不安装任何 Linux 引导装载程序, 那么用户就只能使用 Windows 的引导装载程序。如果选择安装 GRUB, 并将 GRUB 安装在“引导分区的第一个分区”上, 那么通过适当的设置, 用户不仅可以使 Windows 的引导装载器引导每一个操作系统, 而且还能通过 Linux 的 GRUB 引导装载程序来引导它们。

- 当 Linux 安装完成后, 使用 Red Hat Linux 安装光盘以“救急模式”启动系统。将 Red Hat Linux 9 的第一张光盘放入 CD-ROM 驱动器, 重新启动系统, 使得用 Linux 光盘引导系统。

**注意:** 当以这种方式安装了 Red Hat Linux 之后, Linux 并没有对主引导记录作任何修改, 因此系统无法以正常的方式引导 Linux, 然而用户可以以原有的方式引导 Windows 操作系统。

这时必须以“应急模式”引导进入 Linux 系统, 以便操纵硬盘上已安装的 Linux 系统。

- 在弹出的安装方式界面中 (最开始的一屏), 按键盘的功能键 F5, 了解有关“救急模式”的帮助, 然后输入:

```
linux rescue
```

接着按回车键, 进入“救急模式”。

- 根据安装提示界面一步一步进行操作, 直到启动“应急模式”。
- 挂接硬盘上所有的 Linux 分区, 使得这些分区有正确的挂接点。在命令提示符后面输入:

```
chroot /mnt/sysimage
```

然后按回车键，这样就完成了 Linux 分区的正常挂接。

**注意：**这时用户可以使用 Linux 的各种命令，查阅系统的帮助和操纵各种分区（包括 DOS 分区）的数据。

- 建立一个挂接点目录，用于挂接一个 DOS 分区。执行以下命令：

```
mkdir /mnt/dos
```

- 查看可用的 DOS 分区。在 Linux 中使用 fdisk 命令，判断哪些分区是 DOS 分区，包括主 DOS 分区和扩展 DOS 分区。输入如下命令：

```
fdisk /dev/hda
```

其中/dev/hda 是硬盘设备名，如果硬盘不是第一张 IDE 硬盘，用户可以根据硬盘的位置替换成相应的硬盘设备名，然后按回车键，启动 fdisk，进入 fdisk 的环境。输入 p，然后按回车键，将会列出该硬盘的分区列表，选择其中的一个 DOS 分区（只能是主 DOS 分区，或者是逻辑 DOS 盘，不能是扩展 DOS 分区），记住该分区的设备名，假定目前选定的是/dev/hda3。

**注意：**用户不能选择 NTFS 格式的分区，因为刚装好的系统不支持 NTFS 格式的分区，而且在 Linux 下操纵 NTFS 分区是不安全的。因此用户的硬盘上应该至少有一个 FAT 或者 FAT32 格式的 DOS 分区，否则可能只有使用网络或者是软盘来代替硬盘作后续处理，有关如何使用软盘或者网络的信息，请参阅本书的后续章节。

- 挂接你选择的 DOS 分区。输入如下命令：

```
mount /dev/hda3 /mnt/dos
```

这时就可以使用 ls 命令验证刚你挂接的分区是不是你期望选定的 DOS 分区。

- 将 Linux 引导分区的第一个扇区以一个文件的形式保存在你挂接的 DOS 分区上。这里假定 Linux 的引导分区的设备名为/dev/hda2，使用如下命令：

```
dd if=/dev/hda2 of=/mnt/dos/redhat.lnx bs=512 count=1
```

这一条 Linux 命令将 Linux 引导分区的第一个扇区以 redhat.lnx 文件保存到/mnt/dos 上，即保存到/dev/hda3 对应的 DOS 分区的根目录上。

- 退出 Linux 系统，重新启动系统。执行如下命令：

```
exit
```

- 在 Windows 的引导装载器中加入 Linux 引导项。首先将上一步生成的 redhat.lnx 文件复制到 Windows 的引导分区根目录下，假定 Windows 引导分区的盘符为 C:，然后在 Windows 的引导分区根目录下的引导配置文件 boot.ini 的最后加入如下的一行：

```
c:\Red Hat.lnx="Red Hat Linux 9"
```

修改后的 Windows 引导配置文件内容如下：

```
[boot loader]
```

```
timeout=30
```

```
default=multi(0)disk(0)rdisk(0)partition(1)\WINDOWS
```

```
[operating systems]
```

```
multi(0)disk(0)rdisk(0)partition(1)\WINDOWS="Windows Server 2003,
```

```
Enterprise" /fastdetect
```

```
c:\Red Hat.lnx="Red Hat Linux 9"
```

**注意:** boot.ini 文件默认具有系统和隐含属性, 为了编辑这个文件需要使用如下的 DOS 命令变成普通文件:

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

当修改 boot.ini 之后, 为了避免误删除 boot.ini 和 redhat.lnx 文件, 最好使用以下的命令给这两个文件加入系统和隐含属性:

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

```
attrib +h +s +r c:\redhat.lnx
```

另外, 在 boot.ini 文件中输入的 Red Hat Linux 9 表示引导菜单的提示文字, 用户可以换成喜欢的其他文字, 而且这些文字两边需要各加一个半角的引号, 而不是全角的引号。

如果 boot.ini 中的 timeout 设置为 0, 它不会等待用户去选择, 直接引导默认的引导项, 因此你需要将 timeout 设置成大于 0 的值, 单位为秒。

- 重新启动系统, 用户将会看到引导菜单中多了一个引导项。选中 Red Hat Linux 9 的项, 当选中, 并按回车键, 将会启动 Linux 系统。

**注意:** 如果用户在 Linux 的引导分区的第一个分区安装了诸如 GRUB 的 Linux 引导装载程序, 那么在选定了 Linux 之后, Linux 还会弹出一个 GRUB 引导菜单, 这时再选择你将要引导的项目, 然后才能引导对应的项目。为了简化引导过程, 你可以在 GRUB 的配置文件中将 Linux 设置成默认引导, 并且将它的超时项设置为 0, 使之能够直接引导 Linux。

## 思考与练习

- (1) Linux 有哪些特点? 有哪些常见的 Linux 发行版?
- (2) 如何制作 Linux 启动软盘?
- (3) Linux 安装的常见步骤是什么?
- (4) 使用 LILO 和 Grub 时, 将这些引导加载程序放在主引导记录 and 分区引导扇区有什么区别?

## 第2章 初次使用Linux

本章我们将通过实例介绍初步使用 Red Hat Linux 9 的步骤以及相关经验,使读者能够快速进入 Linux 世界,避免出现一些常见的错误。通过本章的学习,读者能够初步使用 Red Hat Linux 9,以便为进一步使用和管理 Linux 奠定基础。

### 2.1 系统引导和用户登录

计算机正常的加电序列完成后,用户将会看到图形化的引导装载程序提示。GRUB 引导装载程序的引导菜单如图 2-1 所示。该菜单显示,系统中安装了 Windows 2003 和 Red Hat Linux 9。

**注意:** GRUB 引导菜单的底部将会出现有关使用的帮助信息。用户可以通过上下光标键选定所需要的选项,此时,这些选项将以高亮度显示,按回车键引导用户所选定的操作系统,按 e 键在启动之前编辑命令,按 a 键可以在引导之前修改内核参数,按 c 键进入命令行方式。

如果使用 LILO 引导装载器,将会出现类似的菜单界面。



图 2-1 GRUB 引导菜单

在这里,用户可以做以下的事情:

- 按 Enter 键。引导默认引导项目。
- 选择一个引导标签,并按 Enter 键。导致引导装载程序引导与该引导标签相对应的操作系统。
- 等待。引导装载程序的超时时间过后(默认为 5 秒钟),引导装载程序将会自动引导默认的引导项目。

如果在引导菜单中选择引导 Red Hat Linux (2.4.20-8)。用户会看到一行一行如图 2-2 所示的向上滚动的信息。



```

[LI]: version 2.04 booting
Setting default font (lat0-sun16): [ OK ]

Welcome to Linux
Press 'I' to enter interactive startup.

Mounting proc filesystem: [ OK ]
Unmounting initrd: [ OK ]
Configuring kernel parameters: [ OK ]
Setting clock (localtime): Thu Apr 29 07:02:52 CEST 2004 [ OK ]
Loading default keymap (us): [ OK ]
Setting hostname john [ OK ]
Your system appears to have shut down uncleanly.
Press Y within 1 second to force file system integrity check...
Checking root filesystem
/ : clean, 98321/460208 files, 462146/219712 blocks

Remounting root filesystem in read-write mode: [ OK ]
Activating swap partitions: [ OK ]
Finding module dependencies: [ OK ]
Checking filesystems [ OK ]
Mounting local filesystems: [ OK ]
Enabling local filesystem quotas: [ OK ]
Enabling swap space: [ OK ]

```

图 2-2 Red Hat Linux 启动过程中的滚动信息

最后，根据用户在图 1-29 安装时选择的登录类型，会出现不同的登录界面。如果用户在图 1-29 中选择“文本登录”方式，会出现一个 login: 登录提示，在此输入用户名和密码后，便可以进入系统。如果用户在安装时选择“图形化登录”方式，系统将会出现如图 2-3 所示的图形化登录屏幕。



图 2-3 图形化登录屏幕

在图形化登录界面中，首先需要输入用户名，并且以回车键结束，然后系统会提示输入该用户账号对应的口令，口令也以回车键为结束标志。如果用户名和口令验证得到验证后，则进入 X 系统，如图 2-6 所示。

**注意：**在第一次启动了安装有 Red Hat Linux 的机器时，用户会看到“欢迎”屏幕。

用户可以在它的引导下进行 Red Hat Linux 配置。使用这个工具，用户可以设置系统时间和日期，新建常规的个人用户账号，安装软件，在 Red Hat 网络上注册系统，额外光盘和结束设置等等。这样的设计可以让用户能够从一开始就配置环境，即使是那些 Red Hat Linux 的初学者也可以得心应手地使用 Red Hat Linux 系统。

在屏幕的底端有一些按钮，通过这些按钮，用户可以设置本次会话所使用语言和会话

类型，当然，用户也可以重新启动系统和关闭计算机。

### 1. 选择会话语言

在登录时用户可以为会话选择一种语言。用鼠标单击图 2-3 的左下部的“语言”按钮，会弹出如图 2-4 所示的对话框。用户可以利用此对话框在每次启动时改变语言的配置，因为第 1 章安装的语言是简体中文，因此默认的语言是“中文（简体）”。这个对话框的最上面有一个“最后”选项，提示用户可以使用最后一次使用的语言。



图 2-4 为会话选择一种语言

**注意：**如果想使用默认语言以外的其他语言，那么用户在安装 Red Hat Linux 9 时必须选择安装该种语言。否则，由于没有安装对应的字体，将会导致一些文字无法正常显示，相应地会在界面上显示一些方框（每一个汉字对应一个方框）。

### 2. 会话选择

用鼠标单击图 2-3 登录界面左下部的“会话”按钮，会弹出如图 2-5 所示的会话框。用户可以有以下 5 种选择：

- 最后。使用最后一次登录时使用的 X 桌面类型。
- Failsafe。
- 默认。用户所设置的默认窗口系统，默认值为 GNOME 或者 KDE。
- GNOME。使用 GNOME 桌面环境。
- KDE。使用 KDE 桌面环境。

**注意：**如果用户已经习惯了某一种桌面环境，用户可以将用户喜欢的桌面环境设置成默认值。这样用户就不需要每次在图形化登录画面中选择会话的桌面类型。



图 2-5 选择桌面类型

## 2.2 X Window 使用基础

### 2.2.1 桌面环境的基本操作

当用户登录成功后，Linux 将会对桌面系统进行一系列的初始化工作，最后启动相应的桌面环境。例如，启动到 KDE 桌面，如图 2-6 所示。使用 KDE 和 GNOME 与使用微软的 Windows 界面非常相似。屏幕底端是面板，面板上会有一些图标，它与 Windows 系统的任务栏类似。面板的最左部（屏幕的左下角）的“红帽子”图标与 Windows 系统的“开始”菜单类似。用户可以单击该图标启动该菜单，菜单中包括了 Linux 所能提供的大多数功能。

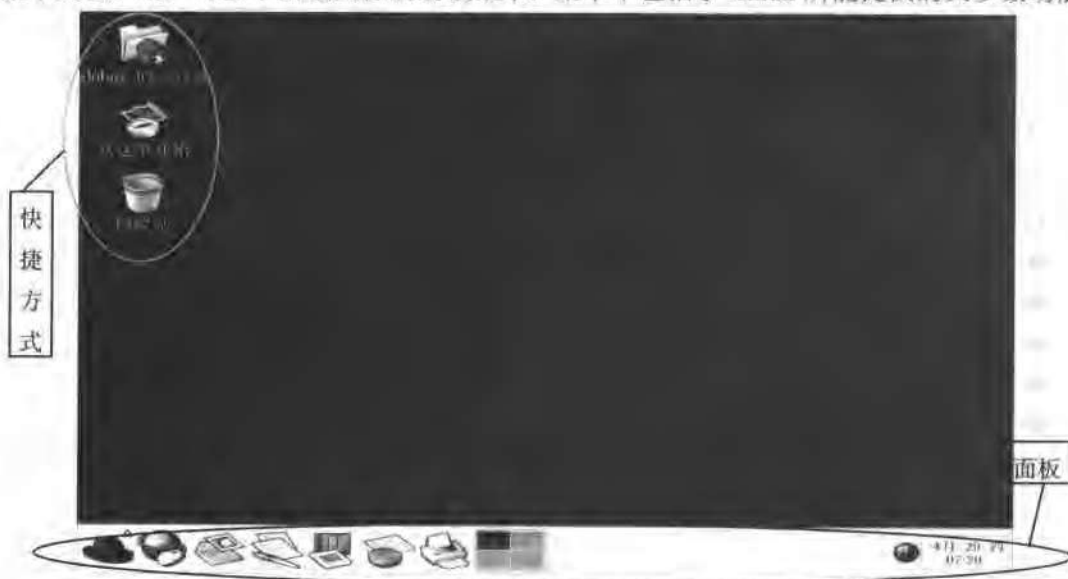


图 2-6 Linux 的 KDE 桌面

KDE 的主菜单如图 2-7 所示。主菜单包括“互联网”、“办公”、“图形”、“声音和视频”、“游戏”、“系统工具”、“系统设置”、“附件”、“首选项”、“主文件夹”、“帮助”、“网络服务器”、“运行文件”、“查找文件”、“打开最近的文件”、“打开最后的文件”、“锁住屏幕”和“注销”，共 17 个选项。



图 2-7 KDE 的菜单系统

若一个菜单的右边有一个形如“}”的符号，则表示该菜单还有子菜单，如图 2-7 中的“系统工具”菜单就有一个子菜单。

操作 KDE 和 GNOME 开始菜单的方法与操作 Windows 主菜单的方法相同，单击“主菜单”（或者称为开始菜单），将会弹出主菜单，在主菜单中选定用户希望操作的菜单项，然后单击该菜单项。如果该菜单项没有子菜单，就会执行该菜单指定的程序，否则它将打开它所对应的子菜单。

本书将使用“主菜单”|“系统工具”|“终端”表示：首先单击“主菜单”，然后选中“系统工具”，单击该菜单项，在弹出的子菜单中选定并单击“终端”菜单项。

KDE 桌面上可以摆放一些常用项目的快捷方式，默认有当前用户的主目录和回收站等内容。

**注意：**如果用户使用文本方式登录进入 Linux，用户可以执行 `startx` 命令切换到图形化桌面系统。这时系统会直接进入默认的桌面环境。

GNOME 与 KDE 的外观类似，同样包含桌面、面板和开始菜单组成。GNOME 开始菜单的菜单项也与 KDE 的菜单项相同。

每次执行一个程序都会在桌面上弹出一个窗口。

KDE 有些常见的快捷键，如表 2-1 所示，它们可以代替一些常见的鼠标操作，来加速 KDE 的操作。

**注意：**笔者在此只是简要介绍 KDE 和 GNOME 的使用方法，关于 KDE 和 GNOME 的配置、管理和详细使用，参见第 15 章 XWindow 的配置和使用。

表 2-1 KDE 的快捷键

热键	对应功能
Alt+Tab	切换至目前工作桌面上的任何一个已经开启的应用程序
Alt+F1	弹出KDE菜单
Alt+F2	弹出输入条, 执行输入的程序
Alt+F3	弹出当前窗口的管理菜单 (控制菜单)
Alt+F4	关闭当前窗口
Ctrl+F1...F8	切换工作空间
Ctrl+Alt+Backspace	强制退出X窗口系统

### 2.2.2 在桌面系统使用 X 终端

在 Linux 中，用户需要打开一个终端（term），以便执行 Linux 命令。

如果用户以文本方式登录系统时，将进入虚拟的文字终端，这时用户可以在虚拟终端中输入和执行 Linux 命令，并且将结果返回到终端上，供用户查看。

在终端中输入 `exit` 或者 `quit` 命令，可以退出图形化终端。也可以通过单击终端窗口左上角的 X 按钮关闭终端窗口。

**注意:** 如果 Red Hat Linux 9 的默认语言为简体中文, 那么在虚拟终端输入 Shell 命令时, 都使用中文显示执行结果, 但是基于文本的虚拟终端不是汉字终端, 执行结果的显示是一些乱码, 为了让它支持中文显示, 用户需要安装一个汉字终端软件, 汉字终端软件很多, 例如可以下载和安装 zhcon, 来支持中文显示和输入。

有关 zhcon 的安装和使用, 请参阅本书的第 4 章: 安装和使用汉字终端。

如果用户进入图形化桌面环境，那么用户可以通过执行“主菜单”|“系统工具”|“终端”命令打开一个终端窗口，它是一个图形化的终端。在 GNOME 中可以使用 GNOME 图形化终端，在 KDE 中可以使用被称为 Konsole 的图形化终端。

默认设置的 GNOME 终端如图 2-8 所示。默认设置的 KDE 终端如图 2-9 所示。



图 2-8 默认设置的 GNOME 终端



图 2-9 默认设置的 Konsole 终端

细心的读者从图 2-8 和图 2-9 可以看出，这两种图形化终端显示的文字都有重影，这严重影响了用户在终端上查看信息。产生“重影”的原因是终端使用的默认字体为非等宽字体，如果 Konsole 使用一种等宽字体，就可以避免这个问题。Red Hat Linux 9 提供了 Courier 等宽字体。下面将讲述将 Konsole 中的字体设置成 courier 的过程。

首先单击 KDE 的“主菜单”|“系统工具”|“终端”菜单项，启动 Konsole，再选中 Konsole 的菜单项“设置”|“字体”|“定制”之后，会弹出一个如图 2-10 所示的“选择字体”对话框。在该对话框的“字体”列表中选择 Courier，在“大小”列表中选择 13。最后单击“确定”按钮，设置后的效果如图 2-11 所示。

为了让设置的字体对以后启动的所有 Konsole 终端都起作用，用户需要在 Konsole 中单击“设置”|“保存设置”菜单项，来保存用户的字体设置。



图 2-10 Konsole 的字体设置对话框

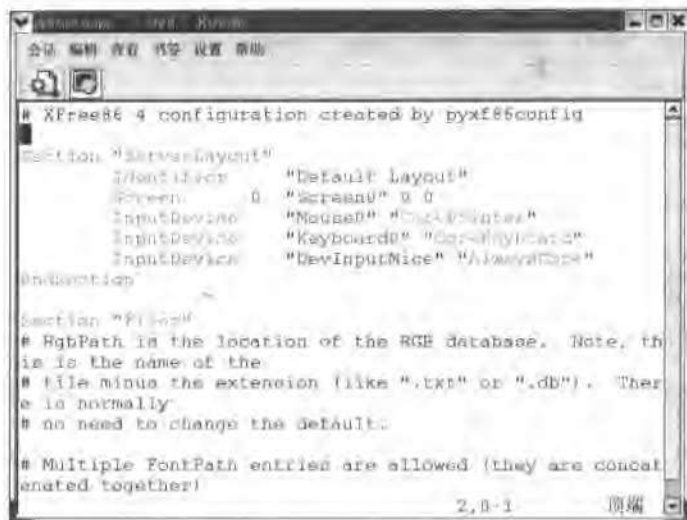



图 2-11 修改字体后的显示效果

**注意:** GNOME 终端也可以用与设置 Konsole 相似的办法设置它的显示字体,只是在 GNOME 终端中,用户需要单击“编辑”|“配置文件”菜单打开“编辑配置文件”对话框。在该对话框,选中 default 配置文件,然后单击“编辑”按钮,在弹出的“编辑文件 Default”对话框中,取消选中“与其他应用程序使用相同的字体大小”复选框,再在弹出的“选择终端字体”对话框的“字体簇”列表选中 courier,“大小”中选择 13,最后单击“确定”按钮关闭“选择终端字体”对话框,然后单击“关闭”按钮关闭“编辑文件 Default”对话框,再单击“关闭”按钮关闭“编辑配置文件”对话框,用户将会看到终端中的文字“重影”消失。

**注意:** 在 Konsole 终端中可以在通过单击“设置”|“方案”菜单弹出的子菜单中,选择一种用户喜欢的方案,其中有 18 种方案可供选择,例如笔者偏爱方案列表中的“Linux 颜色”方案。这种方案的背景为黑色,文字颜色会根据内容的不同而不同,它实际上是对 Linux 的文本终端的模拟。

如果用户的设置不仅仅对当前的终端有效,记住在完成设置后,单击“设置”|“保存设置”菜单项保存设置。

对于 Konsole 终端用户可以在一个 Konsole 窗口中同时打开几个终端子窗口,如果用户需要同时处理多项任务,这个功能是很方便的。用户可以单击“会话”|“新建 Shell”,或者单击工具栏上的  按钮,新建一个会话子窗口。有多个子窗口的 Konsole 如图 2-12 所示。


对于 Konsole 终端中的多个子窗口,用户可以单击工具栏上的  按钮来切换,用户也可以使用 Shift+左/右方向键来加速切换。输入 exit 命令关闭当前终端子窗口。





图 2-12 打开多个终端窗口的 Konsole

## 2.3 用户注销

在文字终端中，可以通过如下的两种方式注销：

- 输入 `exit` 或者 `logout` 命令
- 按 `Ctrl+D` 组合键

注销之后，Red Hat Linux 9 会出现登录提示 (`login:`)，等待一个新用户的登录。

在图形桌面环境中，可以单击“主菜单”|“注销”菜单项完成注销，当单击该菜单项之后，会弹出一个询问是否结束当前会话的对话框。在 GNOME 的询问中，用户还可以选中“保存当前会话”以便下次登录后恢复最后一次会话，最后单击“确定”按钮实施注销过程。文字终端类似，它会弹出如图 2-3 所示的图形登录界面，等待用户登录。

## 2.4 系统关闭和重启

### 2.4.1 文字模式下的系统关闭

用户必须有超级用户权限，才能在文字模式下关闭系统。可以有如下的 4 条命令关闭系统：

- **init 0**。这条指令告诉系统将运行级别切换成 0。该命令立即关闭系统，如果系统正在多用户模式下工作，该命令不会通知其他用户系统将要关闭。该命令与 Windows 系统的关闭计算机功能相同，对于 ATX 电源（较新的计算机都是 ATX 电源），它将关闭电源。
- **init 1**。系统切换到单用户模式，用户可以在单用户模式下，输入 `init 3` 或者 `init 5` 继续启动到多用户模式，输入 `init 0` 关闭系统。



- **halt**。也与 **init 0** 类似，但是它不会关闭电源，用户可以复位计算机，使之重新启动，用户也可以按主机的 **Power** 键关闭系统的电源。
- **shutdown**。可以给用户一个提示，并给用户保存数据的时间，然后彻底关闭系统。

**shutdown +10**

表示输入该命令 10 分钟之后关闭系统，它是一个相对时间，关闭的效果与 **init 1** 相同。

**shutdown 23:20**

该命令表示将在 23 点 20 分关闭系统，关机的时间是绝对的时间。

**shutdown -c**

该命令取消关机事件。

该命令可以通过参数执行 **init 0** 的功能，把系统的运行级别变为 0，立即彻底关闭系统。例如

**shutdown -h now**

等价于 **init 0**。

## 2.4.2 图形模式下的系统关闭

在 KDE 或者 GNOME 的图形桌面环境中，用户除了可以在 X 终端中以根用户身份使用 2.3 节介绍的方法关闭系统之外，用户还可以首先单击桌面的“主菜单”|“注销”菜单项注销当前用户，然后单击图形化登录屏幕的下方的“关机”按钮，关闭系统。该方式等同于用户在终端中输入 **init 0** 命令，如果可能，它会关闭电源。

## 2.5 使用 Linux 上网指南

对于初学者，他们往往需要上网查阅使用 Linux 的问题解答、下载软件或者升级内核。为此，本书将先简述在 Linux 的上网方法。本节首先介绍 Red Hat Linux 提供的图形化浏览器的使用方法，然后介绍文件下载的方法，最后介绍如何使用 Red Hat Linux 提供的图形化电子邮件客户的方法。

### 2.5.1 图形化浏览器的使用

Red Hat Linux 9 提供了两种 Web 浏览器，它们分别是 Mozilla 和 Konqueror。无论是 KDE 还是 GNOME，都可以通过单击“主菜单”|“互联网”|“Mozilla 万维网浏览器”菜单项，启动 Mozilla 浏览器。通过单击“主菜单”|“互联网”|“更多互联网应用程序”|“Konqueror Web 浏览器”菜单项启动 Konqueror 浏览器。打开的 Mozilla 和 Konqueror 浏览器的外观分别如图 2-13 和 2-14 所示。

在 Mozilla 中，单击工具栏上的 **Back** 按钮可以回退到前一个打开的网页，单击 **Forward** 按钮具有与 **Back** 按钮相反的功能，单击 **Reload** 按钮刷新当前网页，单击 **Stop** 按钮可以终止当前网页的打开，工具栏中间的文本框用于输入和显示当前打开的网页的 URL（通用资源定位器）。单击 **View>Show/Hide/Personal Toolbar** 菜单可以显示或者隐藏个人工具栏，所谓的个人工具栏就是图 2-13 中包含 **Home** 和 **Bookmarks** 按钮的那一行工具栏。



图 2-13 Mozilla 的界面

图 2-13 和图 2-14 表明，它们本身都支持中文，因此可以不设置它们的语言和字符集。当网页本身没有设定正确的字符集时，浏览器可能会显示乱码。这是用户需要从 Mozilla 浏览器窗口的 View|Character Coding|Auto Detect 菜单中选择一种字符编码。一般情况下，当用户选择几次后，用户就可以猜出正确的编码，从而使网页正确显示。



图 2-14 Konqueror 浏览器界面

如果用户的局域网设置了代理服务器，如果不是透明代理，用户就需要在浏览器中设置代理服务器的 IP 地址和端口号。在 Mozilla 浏览器中设置代理的步骤如下：

- (1) 单击 Edit|Preferences 菜单项，将会弹出一个 Preferences 对话框，如图 2-15 所示。
- (2) 在 Preferences (参数选择) 对话框的 Category (类别) 列表中，选择 Advanced (高级) 项目，单击该项目展开该节点。
- (3) 在该节点的子节点中，单击 Proxies (代理)，在对话框的右边将会出现 Configure Proxies to Access the Internet (配置访问 Internet 的代理) 框架。
- (4) 右边的框架是由 3 个单选按钮组成，默认选项为 Direct connection to the Internet (直接连接到 Internet)，这是表明用户的计算机不需要设置代理。

如果用户所在的局域网配置了非透明的代理，那么用户需要选中第二个单选按钮 **Manual proxy configuration**（手工配置代理）。用户需要分别为 HTTP、SSL、FTP 和 Gopher 网络服务设置代理，分别输入这几种代理的 IP 地址和端口号（port）。在图 2-15 中设置的代理的 IP 地址和端口号都相同，IP 地址为 192.168.39.254，端口号为 3128。如果代理服务器提供 SOCKS 代理，用户设置 SOCKS 主机的 IP 地址和端口号，这时用户还要了解用户的代理服务器的 SOCKS 版本号，根据需要选择 **SOCKS v4** 或者 **SOCKS v5** 单选按钮。

如果有可用的自动代理配置 URL，用户可以在 **Automatic proxy configuration URL** 文字框中输入对应的 URL。

(5) 当设置完成后，单击 **OK**（确定）按钮。

**注意：**根据代理服务器的设置，不同协议的代理端口很可能是不同的，同时用户也可以对不同的协议使用不同的代理服务器，这样不同协议的 IP 地址文字框里输入的 IP 地址会不相同。另外，代理服务器可能并没有代理某一种协议，因此这时用户不需要输入对应的 IP 地址和端口号，例如图 2-15 中的 Gopher 协议就没有输入代理的 IP 地址和端口号。

有关代理服务器到底代理了哪些服务，请与网络管理员联系。

用户也可以根据需要在参数设置对话框中设置其他属性。

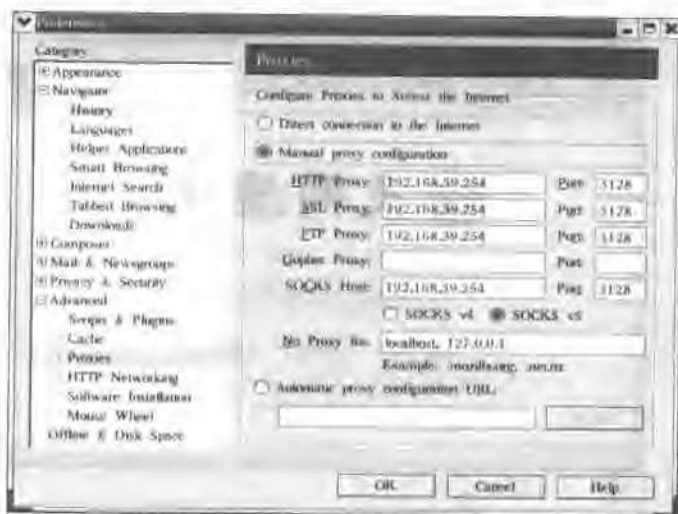


图 2-15 在 Mozilla 中配置代理服务器

**注意：**笔者认为 Mozilla 浏览器是 Red Hat Linux 中非常好用的一款浏览器，但是在默认安装中，它不提供 SSL (Secure Socket Layer, 安全套接字层)、Flash Player (Flash 动画播放器)。

如何配置 Mozilla 浏览器，使之支持 SSL 和 Flash，请参阅第 17 章的有关的内容。

对于 Konqueror 浏览器，用户可以通过单击主菜单的“设置”|“配置 Konqueror”菜单项打开“设置”对话框，并在左边列表中选择“代理”，并在右边弹出的“配置所用的代

理服务器”框中设置代理，设置方法类似，在此不再详述。

在浏览器中，用户可以浏览网页，打开 ftp 站点，下载文件，也可以访问基于 Web 的邮件服务器。

注意：Red Hat Linux 9 也提供了一个文本模式 Web 浏览器 lynx，有关 lynx 的信息，请参阅第 7 章的介绍。

## 2.5.2 文件下载

从网上下载文件是用户上网的一种基本需求。在互联网上支持文件下载的协议主要包括使用 ftp 协议、http 协议和 ssh 协议。本节将讲述在 Red Hat Linux 9 提供的文件下载工具，并且讲述这些工具所支持的协议。

在 Red Hat Linux 中，用户可以使用下述 3 种方法：

- 从浏览器中下载。
- 使用 wget 下载文件。
- 使用图形化下载工具 gftp。

下面将分别介绍这 3 种方法。

### 1. 从浏览器下载

用户可以从 Mozilla 或者 Konqueror 中下载文件。首先打开对应的网址，然后显示出指定的文件。对于 Mozilla，用户需要右击用户想下载的文件链接，然后在弹出的菜单中，选择并单击 Save Link Target As（连接另存为）菜单项，然后会弹出一个 Save As（另存为）对话框，“文件名”文本框里提供一个默认的文件名，用户可以修改这个文件名，用户需要在 Look in（保存位置）列表框中选择保存文件的路径。然后单击 Save（保存）按钮，开始下载过程。Mozilla 默认提供了一个下载管理器，用于管理文件的下载，包括文件下载的速度和进度。

Konqueror 浏览器的文件下载与 Mozilla 浏览器的文件下载类似，只是 Konqueror 浏览器默认使用中文，并且不提供下载管理器。

注意：Mozilla 支持 http 和 ftp 协议的文件下载，Konqueror 不仅支持 http 和 ftp 文件的下载，而且还支持 sftp 协议的文件下载。有关 sftp 的更多相关信息，请参阅本书的第 7 章。

### 2. 使用 wget 的文件下载

wget 是一个 Linux 环境下用于从 Web 上提取文件的工具，它是一个 GPL 许可证下的自由软件，作者为 Hrvoje Niksic。

wget 支持 HTTP、HTTPS 和 FTP 协议，支持代理服务器和断点续传功能，能够自动递归远程主机的目录，找到合乎条件的文件并将其下载到本地硬盘上。如果必要，wget 将恰当地转换页面中的超级连接以便在本地生成可浏览的镜像。由于没有交互式界面，wget 可在后台运行，捕获并忽略 HANGUP 信号，因此即使在用户退出登录以后，仍可继续运行。通常 wget 用于批量地下载 Internet 网站上的文件，或制作远程网站的镜像。它的语法如下：

```
wget [options] URL列表
```

URL 地址格式说明, 可以使用如下格式的 URL:

http://host[:port]/path 或者是 ftp://host[:port]/path

例: 如从 IP 地址为 192.168.1.104 的 ftp 服务器上下载 401.jpg 文件, 它的 URL 是:

ftp://192.168.1.104/WhosWho/401.jpg

则使用 wget 下载的命令是:

```
$ get ftp://192.168.1.104/WhosWho/401.jpg
```

文件下载过程如图 2-16 所示。窗口中的第一行是用 wget 下载 401.jpg 的命令, 中间的方框中的信息是该命令的文件下载过程。这些信息包括与服务器的连接信息, 文件下载的进度, 下载速度, 在本地保存的文件名。

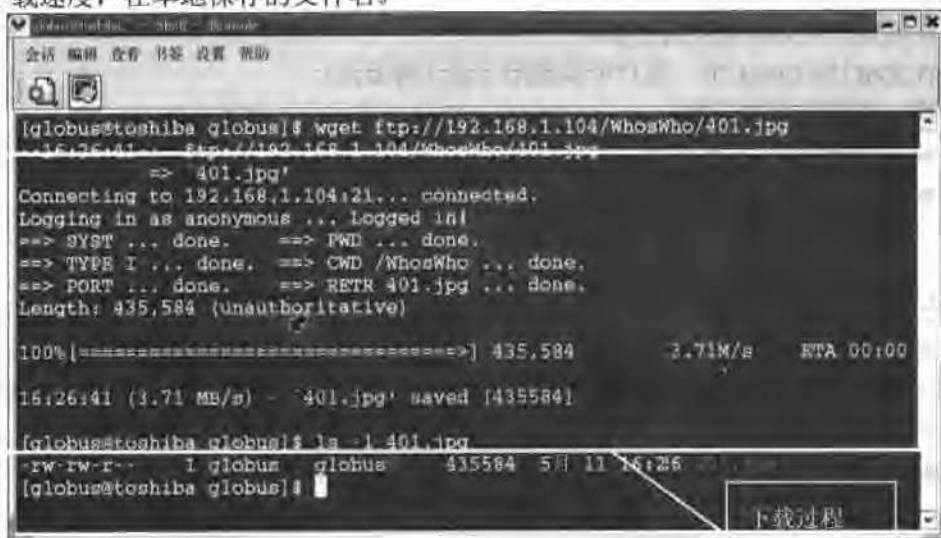


图 2-16 使用 wget 下载文件的示例

wget 的选项较多, 但大部分应用只需要几个常用的选项。

#### (1) 常用选项。

- **-r** 递归。对于 HTTP 主机, wget 首先下载 URL 指定的文件, 然后 (如果该文件是一个 HTML 文档的话) 递归下载该文件所引用 (超级连接) 的所有文件 (递归深度由参数 **-l** 指定)。对于 FTP 主机, 该参数意味着要下载 URL 指定的目录中的所有文件, 递归方法与 HTTP 主机类似。
- **-N** 时间戳。该参数指定 wget 只下载更新的文件, 也就是说, 与本地目录中的对应文件的长度和最后修改日期一样的文件将不被下载。
- **-m** 镜像。相当于同时使用 **-r** 和 **-N** 参数。
- **-l** 设置递归级数。默认为 5。-l1 相当于不递归, -l0 为无穷递归。注意, 当递归深度增加时, 文件数量将呈指数级增长。
- **-t** 设置重试次数。当连接中断 (或超时) 时, wget 将试图重新连接。如果指定 **-t0**, 则重试次数设为无穷多。
- **-c** 指定断点续传功能。实际上, wget 默认具有断点续传功能, 只有当用户使用别的 ftp 工具下载了某一文件的一部分, 并希望 wget 接着完成此工作的时候, 才需

要指定此参数。

例：将在本地硬盘建立 `http://website.com.cn/` 的镜像。命令如下：

```
# wget -m -l4 -t0 http://website.com.cn/
```

镜像文件存入当前目录下名为 `website.com.cn` 的子目录中(用户也可以使用 `-nH` 参数指定不建立该子目录，而直接在当前目录下建立镜像的目录结构)，递归深度为 4，重试次数为无穷。

(2) 使用频率稍低的选项。

- `-A acclist / -R rejlist`，这两个参数用于指定 `wget` 接受或排除的文件扩展名，多个名称之间用逗号隔开。例如，假设不想下载 MPEG 视频影像文件和 AU 声音文件，可使用如下参数：

```
-R mpg.mpeg.au
```

- `-L` 只扩展相对连接，该参数对于抓取指定站点很有用，可以避免向宿主主机的其他目录扩散。例如，某个人网站地址为：`http://www.pub.org/~jzhl/`，使用如下命令行：

```
$ wget -L http://www.pub.org/~jzhl/
```

则只提取该个人网站，而不涉及主机 `www.pub.org` 上的其他目录。

- `-k` 转换连接，HTML 文件存盘时，将其中的非相对连接转换成为相对连接。
- `-X` 在下载 FTP 主机上的文件时，排除若干指定的目录。

(3) 用于设置 `wget` 的工作界面的选项。

- `-v` 设置 `wget` 输出详细的工作信息。
- `-q` 设置 `wget` 不输出任何信息。
- `-i filename`。如果已经在一个 HTML 文档(或普通文本文档)中存储了所要提取的文件的链接信息，可以让 `wget` 直接从该文件中提取信息，而不用在命令行中提供 URL 地址。地址文件也可以不是 HTML 文档，例如，一个普通的文本文件，其中有需要下载的 URL 列表即可。

用户可以用以下技巧提高下载速度：由于 Linux 是一个多任务系统，用户可以同时运行多个 `wget` 进程以提高下载速度，例如，先下载某网站主页 (`index.html`)，然后将该文件所列出的所有地址分别用一个独立的 `wget` 进程进行下载。

注意：当前的 `wget` 不支持 SSH (sftp) 协议的文件下载。

### 3. 图形化下载工具 FTP

gFTP 是 X Window 下的一个用 Gtk 开发的多线程 FTP 客户端工具。它与 Microsoft Windows 下运行的 CuteFTP 等 FTP 工具极为类似，Red Hat Linux 9 提供的 gFTP 的版本为 2.0.14，它主要有以下特性：

- 并行下载
- 断点续传
- 传输任务队列
- 全目录下载
- ftp/http 代理传输支持

- 支持 http、ftp、本地和 SSH2 协议
- 远程目录缓存
- 被动/非被动文件传输
- 文件拖放 (Drag and drop)
- 书签
- 传输中断

如果系统中安装了 GNOME，默认情况下会自动安装 gFTP 2.0.14。gftp 的命令语法如下：

```
gftp [--help] [-h] [--version] [-v] [[ftp://|http://|sftp://][user:pass@]site[:port][directories]]
```

进入 gFTP 的画面如图 2-17 所示。

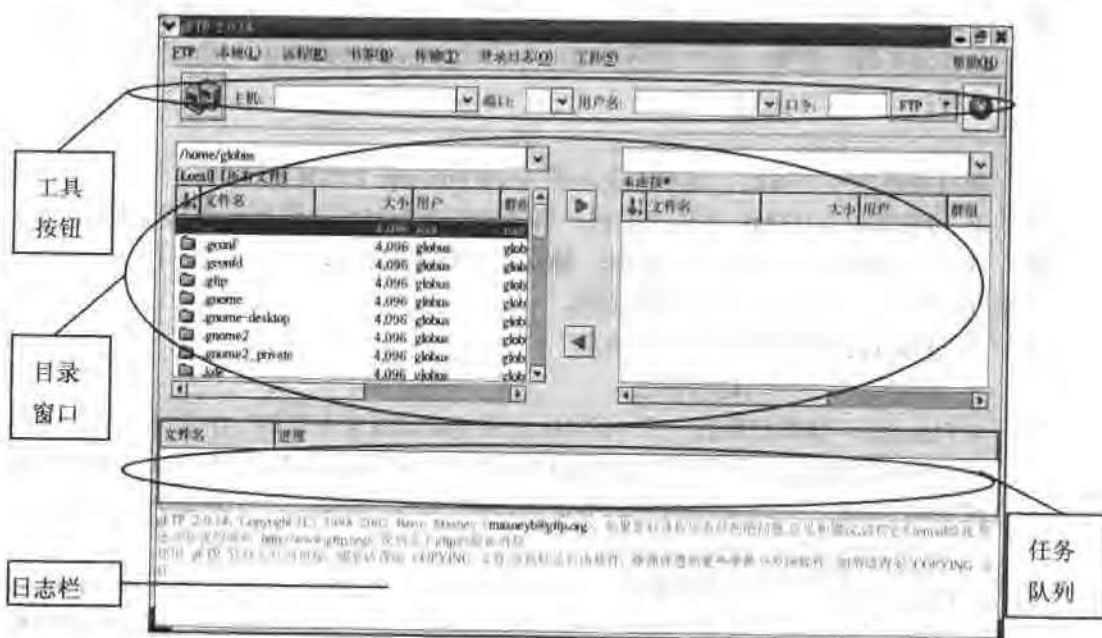


图 2-17 gFTP 的界面

整个窗口的布局分为 5 部分：菜单栏、工具按钮、目录窗口、任务队列、日志栏。

(1) 菜单条的使用。

菜单分为 8 项，分别如下：

- **FTP**。FTP 菜单是 gFTP 的一些整体设置项目。

第 1 栏，当前子窗口。Window 1 指目录窗口中左边一个子窗口，Window 2 指右边一个。只能选定一个子窗口，选中的子窗口上会出现一个\*号。远程连接操作成功后，gFTP 会把远程主机的文件列表放在选中的那个窗口里。

第 2 栏，传输方式。按钮按下去表示选中。包括“文本模式”和“二进制”模式，二者选其一，单击对应的菜单项，选对应的传输方式。

第3栏, 选项。弹出如图 2-18 所示的“选项”对话框。

“常规”选项卡中各选项的功能为“电子邮箱”是匿名登录者用该邮件地址作为匿名 FTP 服务器的认证口令;“查看程序”是用于快速查看目录窗口中所列文件项的程序;“编辑程序”是用于编辑目录窗口中所列文件项的程序;“开始目录”连接成功后显示的第一个目录, 默认为根目录。该页还有一些选项, 在此不再详述, 请参阅相关的文档。

“网络”选项卡中主要设置与网络相关的选项, 包括网络超时秒数(默认 120 秒), 连接重试次数(默认 3 次), 重试前的等待时间(默认 30 秒), 最大传输速率(默认不限制), 默认的文件传输协议(默认为 FTP, 在此可以选择 FTP、HTTP、本地和 SSH2)。

FTP、HTTP 和 SSH 选项卡分别设置与各自协议相关的选项, 包括代理设置, 主动和被动传输模式, 协议的版本等。

“本地主机”选项卡对本地文件进行设置。

第4栏, 退出。






图 2-18 gFTP 的“选项”对话框

- “本地”菜单。是对 Window1 进行操作的菜单项, 它们一般都可以用鼠标和快捷键代替, 在此就不介绍了。
- “远程”菜单。它是用来对 Window2 进行操作的菜单项, 内容与“本地菜单”一致。
- “书签”菜单。gFTP 用收藏夹的方式快速访问定制的站点。即使对 gFTP 进行升级, 收藏夹中的内容仍能够继续使用。用户可以通过该菜单来添加和编辑书签。
- “传输”菜单。用来管理传输, 主要包括上传、下载、开始、停止行或者取消一个传输, 改变队列的顺序。
- “登录日志”菜单。用来管理登录日志。可以查看、清除, 保存日志信息。
- “工具”菜单。可用于对比 Window1 和 Window2 的内容, 避免重复下载, 方便用户使用。



## (2) 工具按钮的使用。

多种工具按钮的设置使连接和操作速度加快。是最左边的按钮，用于与远端服务器建立连接；是最右边的按钮，用于断开连接，“主机”文本框用于输入远程服务器的主机名；“端口”文本框输入与远程主机的端口号，通常情况下可以有所用的网络协议自动决定，因此用户通常不需要输入；“用户名”文本框接收用户输入与远程主机登录时的用户账号；“口令”文本框接收与远程用户账号对应的口令。如果用户使用的是匿名用户，那么用户可以不输入用户名和口令。“协议”下拉列表框用于选择与远程服务器的通信协议。

当按钮呈明亮的红色时，表示当前执行的传输可以停止。

当用户在上述对应的文本框里输入主机名（IP 地址亦可）、端口号、用户名、口令和传输协议之后，单击“连接”按钮，与远程主机进行连接。当连接成功后，远程主机的目录和文件列表将会显示在右边的目录子窗口。

注意：上述几种传输协议中，SSH 需要在本机安装 sftp Server ([HTTP://WWW.XBILL.ORG/SFTP](http://www.xbill.org/sftp))。通过给 Window1 和 Window2 分别指定不同的 URL，该方式可实现两台 FTP 服务器之间的文件互操作。

## (3) 目录窗口的操作。

目录窗口的内容与 CuteFTP 等软件的目录窗口极为类似。单击鼠标右键可实现 Local 和 Remote 菜单中的操作。对于三键鼠标，用第三键可以实现拖放；对于两键鼠标，在“模拟三键”的情况下，可以用两键同时按下实现拖放。用 Shift 键或 Ctrl 键可实现多选。可选择目录或文件。

## (4) 文件下载举例。

假定用户在北京大学的 FTP 网站上，下载名为 `cxterm5.0.p3.tar.gz` 的文件，它的 URL 为 `ftp://ftp.oku.edu.cn/pub/chinese/cxterm5.0.p3.tar.gz`。操作步骤：

1) 首先在“主机”文本框中输入北京大学的主机名 `ftp.pku.edu.cn`，并在协议下拉列表中选择 FTP，然后单击连接按钮，开始连接过程，在日志里面显示连接的详细信息。当连接成功后，右边的目录子窗口将会列出 `ftp` 服务器根目录的文件和子目录列表。由于没有输入端口号，用户名和密码，因此 gFTP 会使用 FTP 协议默认的端口号 21，匿名用户名 `anonymous`，和设置的默认口令 `globus@wang`。

2) 在右边的目录子窗口中，通过鼠标操作将 `/pub/chinese` 变为当前目录。

3) 选定文件 `cxterm5.0.p3.tar.gz`，并将它拖至左边的目录子窗口。

4) 文件下载过程开始，任务队列窗口出现了文件 `cxterm5.0.p3.tar.gz` 的下载任务。并且提示当前的下载速度和估计剩余时间。日志窗口中，也会显示与服务器交互过程。当下载完成后，该下载任务将从下载任务列表中删除。传输过程如图 2-19 所示。

注意：用户可以用同样的方法下载 HTTP 服务器上的文件，以及使用 SSH 协议的文件。

用户也可以上传文件，不过使用的用户在远程服务器上必须具有上传文件的权限。

用户可以使用个 FTP 同时上传和下载多个任务。

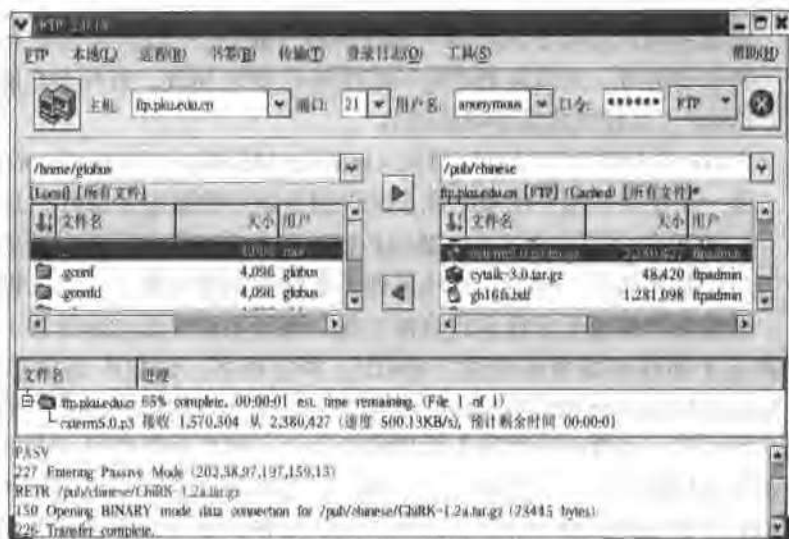


图 2-19 使用 gFTP 下载文件的实例

### 2.5.3 图形化电子邮件客户

电子邮件是一种在互联网上与其他人通讯的流行方式。用户可以通过电子邮件客户 (email client) 来使用电子邮件。电子邮件客户是一种理解各种电子邮件传输标准, 并允许用户收发和阅读电子邮件的应用程序。

Red Hat Linux 包括了好几种电子邮件应用程序, 其中有 Evolution 和 Mozilla Mail 之类的图形化电子邮件客户, 也有 mutt 之类的基于文本的电子邮件客户。由于所有的电子邮件客户都执行同样的基本任务 (收发邮件), 所有这些客户程序又是为满足某类用户的需要而设计的, 因此, 用户可以选择带有最能满足用户个人需求功能的那一个。

本节主要是讲述如何使用包括在 Red Hat Linux 中的一些流行的图形化电子邮件应用程序。

- Mozilla Mail。
- Evaluation 电子邮件。
- KMail。

在用户启动电子邮件客户之前, 用户应该具备从用户的互联网服务提供商处获得的信息, 只有这样用户才可以正确地配置电子邮件客户程序。下面列出了用户可能需要了解的信息:

- 用户的电子邮件地址和口令。用户用来收发邮件的电子邮件地址。它的格式通常是 yourname@yourisp.com, 每一个邮件地址都有一个口令与之对应, 电子邮件收发通常都有需要邮件账号的口令。
- 接收电子邮件的服务器类型 (POP 或 IMAP) 和主机名。为了能接收邮件, 用户必须知道用户的网络管理员或 ISP 使用的是哪类服务器。POP 或 IMAP 地址的格式通常是 mail.someisp.com。
  - POP 是邮局协议 (Post Office Protocol) 的简写。它用来把邮件从邮件服务器

发送到用户的电子邮件客户的收件箱 (inbox)。收件箱是贮存接收到的电子邮件的地方。大多数 ISP 电子邮件使用 POP 协议, 有些 ISP 能够使用较新的 IMAP (互联网信息存取协议)。

- IMAP 是互联网信息存取协议 (Internet Message Access Protocol) 的简写。它是一个用来从用户的 ISP 电子邮件服务器中检索电子邮件消息的协议。IMAP 与 POP 的不同之处在于, IMAP 服务器上的邮件被贮存在服务器上, 即使用户下载并阅读了它们, 这些邮件仍被保留在服务器上。默认设置下, POP 邮件被用户的电子邮件客户直接下载, 而且不被保留在服务器上。
- 寄发电子邮件的服务器类型 (SMTP) 和主机名。SMTP 是简单邮件传送协议 (Simple Mail Transfer Protocol) 的简写。它是一个用来在服务器间发送电子邮件消息的协议。多数在互联网上发送邮件的电子邮件系统使用 SMTP 来把消息从一个服务器传输到另一个服务器中。然后, 这些消息便可由电子邮件客户使用 POP 或 IMAP 服务器来检索。SMTP 还被用来把消息从电子邮件客户发送到邮件服务器内。这就是用户在配置电子邮件程序时需要指定 POP 或 IMAP 服务器以及 SMTP 服务器的 IP 地址或者 URL 的原因。

例如, 笔者的电子邮件地址为 jzh\_smr@sohu.com, 该邮件的 ISP 是搜狐, ISP 规定在发送邮件时使用 SMTP 协议, 收取邮件时使用 POP3 协议, 邮件发送服务器的主机名为 SMTP.SOHU.COM, 接收邮件的服务器主机名为 POP3.SOHU.COM。如果使用搜狐提供的邮件发送和邮件接收服务器, 都必须使用对应的电子邮件地址的密码, 为了保证电子邮件的安全, 电子邮件地址的密码是保密的, 只有邮件地址的所有者知道。

**注意:** 有些邮件是基于 Web 的电子邮件, 可能并不提供邮件发送服务器和邮件接收服务器, 那么用户就不能使用本节所讲的图形化电子邮件客户, 这时只能通过 ISP 提供的 Web 邮件网页来收/发电子邮件。然而一些 ISP 对同一个邮件地址, 既提供 Web 邮件方式访问, 又提供邮件收/发服务器。例如笔者的邮件地址 jzh\_smr@sohu.com 除提供收发邮件的服务器之外, 也提供基于 Web 的访问方法。本节所讲的所有图形化电子邮件客户都不支持微软的 hotmail 的收发。目前在 Red Hat Linux 9 中只能使用 Web 方式访问该种邮件。Hotmail 需要 SSL 的支持, 默认安装中, Red Hat Linux 9 没有安装 SSL (PSM, 个人安全管理器)。

如果用户对所需信息有疑问, 请联系用户的 ISP 或网络管理员。如果没有正确的配置, 用户将无法全面利用本节讨论的电子邮件客户。

这些图形化电子邮件客户程序如下:

### 1. Evolution

Evolution 不仅仅是一个电子邮件程序。它提供了所有标准的电子邮件客户功能, 包括功能强大的邮箱管理、用户定义的过滤器、以及快速搜索。除此之外, 它还具备灵活的日历 (调度器) 功能, 该功能允许用户在线地创建和确认组群会议和特别事件。Evolution 是用于 Linux 和基于 UNIX 的系统的功能完善的个人和工作组信息管理工具, 它还是 Red Hat Linux 的默认电子邮件客户。

用户可以通过单击面板上的“主菜单”|“互联网”|“Evolution 电子邮件”从桌面上

启动 Evolution。

如果是第一次启动 Evolution, 它会提供一个向导, 帮助用户设置用户的电子邮件信息。这个向导首先出现一个欢迎页, 在此单击“下一步”按钮, 接着会出现“标识”页, 在“必需的信息”组中, 需要填入用户的邮件标示(全名)和邮件地址(Email Address), “全名”会包含在用户发送的邮件中, 收件人可以看到这个名称。对于笔者的搜狐邮箱, 在“全名”文本框中输入“姜中华”或者 Zhonghua Jiang, 在邮件地址中输入 jzh\_smr@sohu.com, 单击“下一步”按钮。

在出现的“正在接收电子邮件”页中, 需要用户选择“服务器类型”, 它表示接收邮件的服务器类型, 这里有一个类型列表可供用户选择。用户需要根据 ISP 提供的信息选择, 通常情况下选择 POP, 有些会选 IMAP。对于笔者的搜狐邮箱, 在此选 POP。单击“下一步”按钮。在出现的下一页“正在接收的电子邮件”页中选择关于接收邮件的服务器具体信息, 对于不同类型的接收邮件服务器, 设置会各不相同。对于 POP 协议, 用户需要设置 POP 服务器的“主机”名、邮件地址的“用户名”、Use secure connection (SSL) (是否使用 SSL)、身份验证类型, 以及是否记住密码。对于是否 SSL, 主要由 ISP 决定。身份验证类型也有多种方式, 但是最常用的是“密码”。对于笔者的搜狐邮件地址, 在“主机”文本框中输入 POP3.SOHU.COM, Username 文本框处输入 jzh\_smr, 在 Use secure connection (SSL) 下拉列表中选择 Never, 在验证类型下拉列表中选择“密码”, 选中“记住该密码”复选框, 以避免下次再次输入密码。单击“下一步”按钮, 在第3屏“正在接收电子邮件”页中设置两个接收邮件的选项。“自动检查新邮件”复选框设置是否自动监测新邮件, 邮件存储的“在服务器上保留邮件”复选框设定是否在服务器上保留邮件, 默认情况下, 当用户接收邮件后, POP 服务器将删除该邮件, 在许多情况下, 选中保留邮件选项, 用户可避免邮件丢失的损失。笔者选中“在服务器上保留邮件”复选框。单击“下一步”按钮。

在出现的“正在发送电子邮件”页中, 选择发送邮件的服务器类型、主机名, 是否使用 SSL 协议、服务器是否需要认证, 以及认证类型。服务器类型可以选择 SMTP 和 Sendmail, 常用的服务器认证类型为 PLAIN 和“登录”两种。对于笔者的搜狐邮件地址, 在“服务器类型”下拉列表中选择 SMTP, 在“主机”文本框中输入 SMTP.SOHU.COM, 在 Use secure connection (SSL) 下拉列表中选择 Never, 选择“服务器需要认证”, 在“验证类型”下拉列表中选择“登录”, 选中 Remember this password 复选框记住口令。单击“下一步”按钮。

在出现的“账号管理”页中, 设置一个管理用的“名称”, 这个名称可以任意取, 默认是该邮件地址。选中 Make this my default account 复选框可以将该账号设置为默认账号。单击“下一步”按钮。

在接下来出现的“时区”页中, 用鼠标左键在给出的地图上选择, 或者在“选择”下拉列表中选择用户所在的时区。笔者选择“亚洲/上海”。单击“下一步”按钮。

在出现的“完成”页中, 单击“结束”按钮, 完成对该账号的设置。

最终将会出现 Evolution 的界面, 用户可以在工具栏的“新建”下拉列表中选择新建 Appointment (约会), Contact List (联系人列表), Post Message (发送消息), 联系人, 任务和文件夹等项目, 用户可以单击工具栏的“发送和接收”按钮来发送和接收邮件。

**注意：**在第一次发送和收取邮件时，Evolution 会提醒用户输入密码，并且提醒用户是否记住该密码。

如果用户选择了记住密码，那么以后再收发邮件时就不需要再次输入密码了。

单击“收件箱”可以查看用户的收件箱里的邮件。查看收件箱的实例如图 2-20 所示。一旦消息已编写完毕，并且输入了要发送到的电子邮件地址后，在工具栏中单击“发送”按钮。

虽然 Evolution 所能进行的不仅仅是收发电子邮件，本小节只集中于它的电子邮件方面的功能。如果用户想了解关于 Evolution 的其他功能的使用信息（如日历（调度）和组群消息收发），单击工具栏上的“帮助”按钮，然后选择用户希望了解的组件。



图 2-20 Evolution 的收件箱实例

要编写一封邮件，在工具栏上单击“新建”按钮可创建新的邮件，在工具栏上单击“回复”按钮，可以回复当前邮件。一些回复邮件的实例如图 2-21 所示。

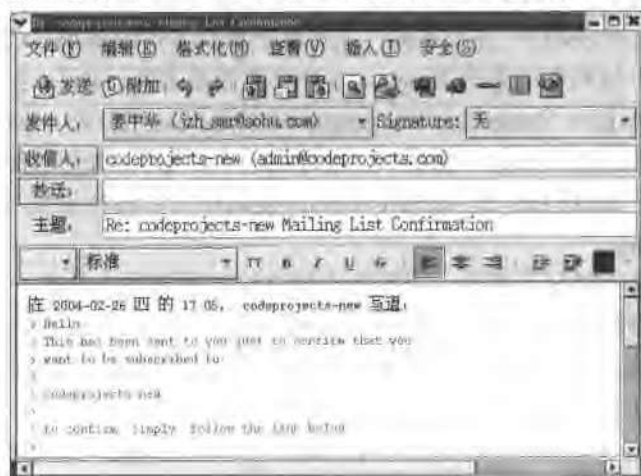



图 2-21 Evolution 回复邮件实例

## 2. Mozilla Mail

下面将简单地介绍使用 Mozilla 来收发邮件的基本步骤。如果用户需要关于使用 Mozilla Mail 的进一步信息, 请阅读 Mozilla 主菜单的 Help | Help Contents 菜单项对应的内容。

要启动 Mozilla Mail, 可以有以下 3 种方式:

- 选择“主菜单”|“互联网”|“更多互联网应用程序”| Mozilla Mail。
- 在 Mozilla 浏览器中, 单击 Mozilla 屏幕左下角的邮件和新闻组图标 。
- 在 Mozilla 浏览器中, 选择 Mozilla 的主菜单 Window | Mail & Newsgroups。

Mozilla Mail 的界面如图 2-22 所示。与邮件账号有关的配置与 Evolution 中的配置类似, 在次不再重述。如果用户希望获得更详细的信息, 请参阅联机文档。

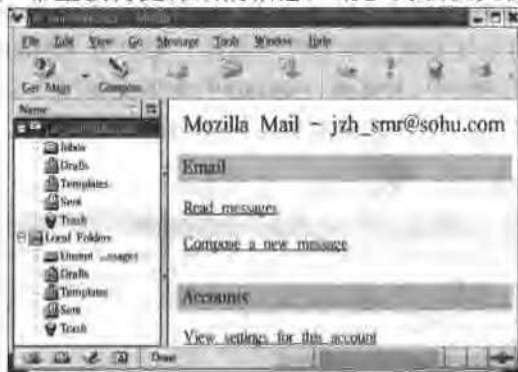


图 2-22 Mozilla Mail 的主界面

如果通过 Mozilla Mail 撰写一封邮件, 单击工具栏上的 Compose 按钮, 或者单击主菜单 Message | New Message 菜单项。它将会弹出如图 2-23 所示的 Compose (撰写) 窗口。用户需要在 To 文本框中输入收件人地址, 也可以单击工具栏的 Address 按钮选择收件人; 在 Subject 文本框中输入邮件的主题, 其中 From 下拉列表框中默认会用当前操作的邮件账号作为发件人, 如果用户同时配置了多个邮箱, 可以从列表中选择其他的邮箱名作为发件人; 如果需要附件, 单击工具栏上的 Attach 按钮, 在弹出的 Enter file to attach 对话框选择要附件的文件名。



图 2-23 Mozilla Mail 中撰写邮件的界面

在邮件正文文本框中输入邮件正文后，可以单击工具栏上的 Send 按钮，或者单击 File | Send Now 或者 Send Later 菜单项发送邮件。

在 Mozilla Mail 的工具栏上单击 Get Msgs，可以收取电子邮件。


要阅读电子邮件，在 Mozilla Mail 窗口中单击用户为自己创建的邮件文件夹，来查看消息列表。然后，单击用户想阅读的消息。

一旦用户读过了消息，可以删除它，也可以把它存到另一个文件夹中。

**注意：**与 Evolution 相比，Mozilla Mail 的界面和提供的功能简单实用，能使用户关注到最主要的事情上，而不会被一些繁琐的细节所干扰。

### 3. Kmail

KMail 是 KDE 的电子邮件工具。它有一个与 Evolution 很相似的视觉化的图形界面。这个界面允许用户使用图形化界面来收发电子邮件。要打开 KMail，选择“主菜单”|“互联网”|“更多互联网应用程序”|KMail。KMail 的主界面如图 2-24 所示。

在能够使用 KMail 之前，用户必须配置有关邮件帐户的信息。要运行配置工具，需要选择 KMail 工具栏上的“配置”工具，或者选择“设置”|“配置 KMail”菜单。

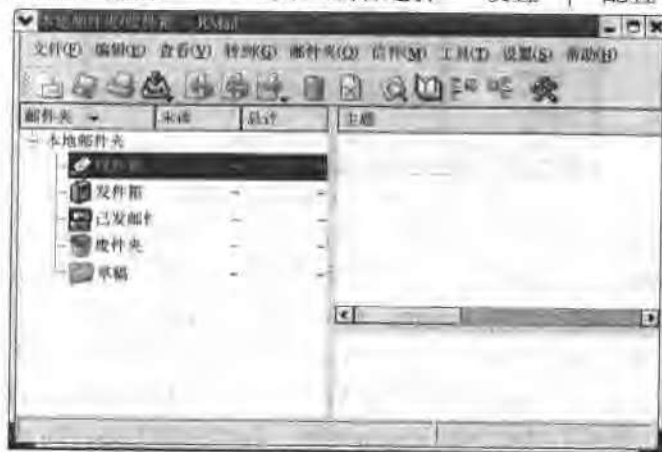


图 2-24 KMail 邮件客户的主界面

“配置 KMail”窗口如图 2-25 所示，它包括以下几个部分：“身份标识”、“网络”、“外观”、“编写器”、“安全”和“邮件夹”。要开始收发消息，用户必须改变“身份标识”和“网络”活页标签。准备好从用户的服务提供者或管理员处获得的电子邮件信息，因此用户可以填充所需信息，以便开始使用 KMail。要获取附加信息，请参阅 KMail 用户手册（“帮助”|“KMail 手册”）或访问 KMail 的主页。

**注意：**在 Red Hat Linux 9 中，除了图形化的邮件客户以外，还有基于文本的邮件客户 Mutt 等，它们的功能非常强大，可以完成用户的绝大多数邮件处理任务，有关 mutt 的使用，请参阅第 7 章。





图 2-25 KMail 的“配置”对话框

## 2.6 如何获得在线帮助

在 Linux 操作系统中，与其他 Unix 系统类似，有很多 Shell 命令。这些命令一般都包含很多参数，很多 Windows 用户往往不熟悉这些命令，系统必须为用户提供联机帮助，除了这些帮助之外，用户还可以通过网络在互联网上获得在线帮助。

如果用户正在寻找命令和出错信息，最好的地方就在用户的系统中查找。用户有如下的几种帮助方式可用：

- 手册页 (Man Pages)。它们是有关 Shell 命令、文件格式和系统调用的参考资料。
- RPM 包文档。许多包都包含附加的文档，RPM 能帮助找到这些文档。
- HOWTO 和 FAQ。从 Linux 文档计划中获得帮助信息。
- slocate 命令。它是一个文件搜索命令。
- info 页。超文本文件 (非 HTML 文件)。

本节的后续部分将介绍如何使用这 5 种获得帮助的方式。

### 2.6.1 手册页

在 Red Hat Linux 系统中，系统中的每条命令都有相关的手册页 (man page)。这里的手册页是指有关 Shell 命令的一种联机帮助文档。当使用这些命令遇到问题或困难时，用户可以立刻得到这个帮助手册。

用户要得到有关某条命令的手册页，在命令提示符下，执行如下命令：

```
man shell_command
```

其中 shell\_command 指的是用户需要得到帮助的某条 Shell 命令。当这条命令执行时，它将会弹出一个文本的窗口，这个窗口没有边框，窗口中显示了帮助文档的一部分。



注意：使用 `man page` 往往需要了解一条命令的功能，这条命令的某个参数的意义，或者有某种需求，需要查看实现这种功能所对应的参数。

例：假定用户在使用 `wget` 命令时遇到困难，用户可以在 Shell 提示符后输入 `man wget` 得到帮助信息。获得 `wget` 使用手册的窗口如图 2-26 所示。

```
NAME
  wget - GNU Wget Manual

SYNOPSIS
  wget [option]... [URL]...

DESCRIPTION
  GNU Wget is a free utility for non-interactive download of files from
  the Web. It supports HTTP, HTTPS, and FTP protocols, as well as
  retrieval through HTTP proxies.

  Wget is non-interactive, meaning that it can work in the background,
  while the user is not logged on. This allows you to start a retrieval
  and disconnect from the system, letting Wget finish the work. By con-
  trast, most of the Web browsers require constant user's presence, which
  can be a great hindrance when transferring a lot of data.

  Wget can follow links in HTML pages and create local versions of remote
  web sites, fully recreating the directory structure of the original
  site. This is sometimes referred to as "recursive downloading".
  While doing that, Wget respects the Robot Exclusion Standard.
```

图 2-26 `man page` 的界面示例

手册页通称由以下几个部分内容组成：

- **NAME**（命令的名称）。本例为 `wget`。
- **SYNOPSIS**（命令的语法格式）。此例为 `wget [option]... [URL]...`
- **DESCRIPTION**（对命令的功能的描述）。主要说明该命令的特征、功能和使用场合。
- **OPTIONS**（选项）。说明该命令提供的所有选项和参数，以及对每一选项和参数的解释说明，文档通常将这些选项分成功能相近的几个大类来分别说明。
- **EXAMAPLES**（举例）。针对该命令的使用举一些例子，以供用户参考。
- **FILES**（文件）。帮助文件的位置默认位置。
- **BUGS**。告诉用户如何将他们遇到的 **BUG** 发送到指定的地方。
- **SEE ALSO**。与该程序相关的其他程序。
- **AUTHOR**。说明软件的作者。
- **COPYRIGHT**。版权信息。

在 `man` 的显示窗口中，在窗口的最下面一行，可以输入一些少量的命令，主要用于信息查找和退出 `man` 手册页。

`man` 程序是基于 `less` 命令的，因此 `less` 程序的操作命令都可以在 `man page` 中使用。常见的操作命令如表 2-2 所示。

表 2-2 `man page` 窗口常用的操作

按键	描述
q	退出
Enter键或者k	向下滚动一行
j	向上滚动一行

续表

按键	描述
SPACE (空格键)	向下滚动一页
CTRL+b	向上滚动一页
g	跳转到手册的第一行
G	跳转到手册的最后一行
/String	在手册中查找第一个string字符串, 后面需要按Enter (回车) 键, String为任意一个字符串
n	发现上一次查找的下一匹配, 可以多次使用
N	相反方向查找的下一匹配, 可以多次使用
h	获得man程序的操作帮助

例: 如果用户希望在如图 2-26 的手册页中查找字符串 `wget`, 那么首先输入 `/wget`, 然后按回车键。匹配的字符串将反色显示, 如图 2-27 所示。

有时纸张阅读更方便。假定用户有一个打印机, 也可以把 `man page` 打印出来。如果用户的打印机没有打印 `Postscript` 的能力, 只想打印 `ASCII` 码, 可用如下命令实现打印:

```
$ man COMMAND | lpr
```

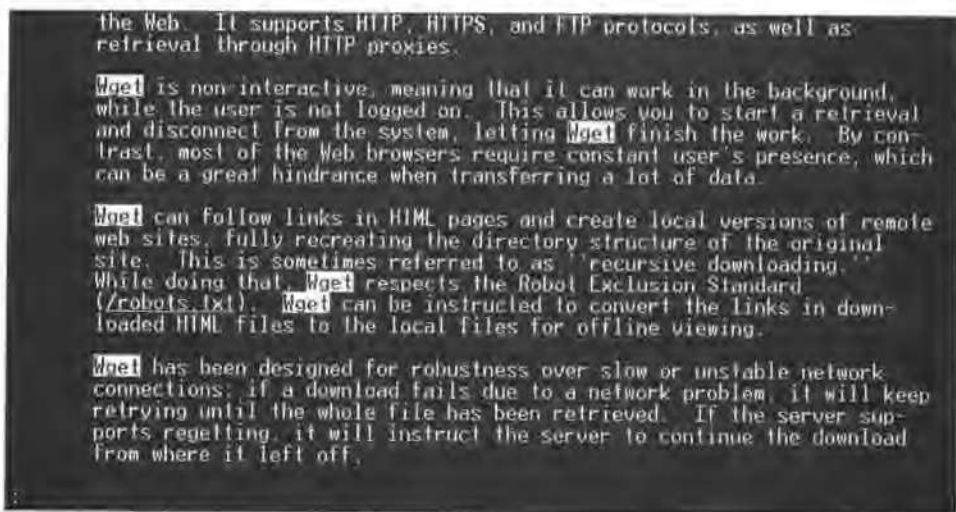


图 2-27 在手册页中查找字符串的实例

如果打印机支持 `postscript`, 则可以用如下命令:

```
$ man -t COMMAND | lpr
```

其中命令中的 `COMMAND` 为 `Shell` 命令的名称。例如对于 `wget`, 可以使用如下的打印命令:

```
$ man wget | lpr
```

注意：可能对于同一个命令有不同的多个手册页，它们分别用不同的节号分开，例如 kill，一个 kill 的 Shell 命令（节号为 1），另一个是 Linux 系统的系统调用（节号为 2），如果用户输入 man kill，实际上看的是第 1 节，即是 kill 命令。如果用户希望查看 kill 系统调用，用户可以输入 man 2 kill。

注意：对于 Linux 的初学者，不能将手册页（man page）当成教程，因为它只是简明的参考资料。

如果刚刚开始学习 Linux，可以使用下面将要介绍的方法获取文档。

## 2.6.2 包的文档

许多软件包都有自述文件（README）和其他与软件相关的文档。Red Hat Linux 的 RPM 包定义了一个存放这些文档标准位置，每个包都把相应的文档放在 /usr/share/doc 下的一个子目录中，因此可以快速检索到这些文档。

子目录名由包的名称和版本号共同决定，如 openssh 包的版本是 3.5p1 版。那么相应的文档就放在 /usr/share/doc/openssh-3.5p1 目录里。

包的文档可能包含多个文件，如 openssh-3.5p1 包含如下的文档：

```
/usr/share/doc/openssh-3.5p1/CREDITS
/usr/share/doc/openssh-3.5p1/ChangeLog
/usr/share/doc/openssh-3.5p1/INSTALL
/usr/share/doc/openssh-3.5p1/LICENCE
/usr/share/doc/openssh-3.5p1/OVERVIEW
/usr/share/doc/openssh-3.5p1/README
/usr/share/doc/openssh-3.5p1/README.privsep
/usr/share/doc/openssh-3.5p1/README.smartcard
/usr/share/doc/openssh-3.5p1/RFC.nroff
/usr/share/doc/openssh-3.5p1/TODO
/usr/share/doc/openssh-3.5p1/WARNING.RNG
/usr/share/man/man1/ssh-keygen.1.gz
/usr/share/man/man8/ssh-keysign.8.gz
```

注意：rpm 是 Red Hat Linux 的包管理命令，它具有包的安装、更新、查询和删除等多种功能。

有关 rpm 的详细信息，请阅读 rpm 的手册页（man page），或者参见本书的第 13 章。

RPM 包的大多数的文档是 ASCII 码格式，用户可以用 more rpmhelpfilename 或者 less rpmhelpfilename 来阅读。例如了解如何安装 openssh-3.5p1，可以使用如下命令来查看：

```
$ more /usr/share/doc/openssh-3.5p1/INSTALL
```

要希望了解某一个包的帮助文件到底有哪些，用户可以使用包查看命令。格式如下：

```
$ rpm -qdi package_name
```

其中 package\_name 是指 RPM 的名称，如对于 openssh-3.5p1，可以输入下述命令，来了解到底有哪些文件：

```
$ rpm -qdi openssl
```

### 2.6.3 HOWTO 和 FAQ

如果用户安装了 HOWTO 和 FAQ, Linux 文档计划的大多数内容将在用户的系统的 `/usr/share/doc` 目录中找到。

这些 HOWTO 分布在 `/usr/share/doc` 目录下的多个子目录里, 文件的格式也是多种多样的, 包括文本文件, pdf 格式的文件和 html 格式的文件, 用户需要 pdf 浏览器和 Web 浏览器来查看这些文件。

输入如下的命令, 查找系统中所有的 HOWTO 文件:

```
$ slocate HOWTO
```

在 Red Hat Linux 9 的默认安装中会有如下的输出:

```
/usr/share/doc/shadow-utils-4.0.3/HOWTO
/usr/share/doc/lvm-1.0.3/LVM-HOWTO
/usr/share/doc/kernel-pcmcia-cs-3.1.31/PCMCIA-HOWTO
/usr/share/doc/logwatch-4.3.1/HOWTO-Make-Filter
/usr/share/doc/stunnel-4.04/VNC_StunnelHOWTO.html
/usr/share/doc/samba-2.2.7a/docs/Samba-HOWTO-Collection.pdf
/usr/share/doc/samba-2.2.7a/docs/htmldocs/OS2-Client-HOWTO.html
/usr/share/doc/samba-2.2.7a/docs/htmldocs/Samba-HOWTO-Collection.html
/usr/share/doc/samba-2.2.7a/docs/htmldocs/Samba-BDC-HOWTO.html
/usr/share/doc/samba-2.2.7a/docs/htmldocs/Samba-LDAP-HOWTO.html
/usr/share/doc/samba-2.2.7a/docs/htmldocs/Samba-PDC-HOWTO.html
/usr/share/doc/cdda2wav-2.0/HOWTOUSE
/usr/share/doc/gaim-0.59.8/PERL-HOWTO
/usr/share/doc/doxygen-1.2.18/LANGUAGE.HOWTO
```

注意: 用户可以使用 `slocate` 命令查找系统中有哪些 HOWTO 可供使用, 以及它们的位置。

用户同样可以使用 `slocate` FAQ 命令查看系统中所有的 FAQ (问题和解答), 除了上述提到的文件格式以外, 还包括使用 `gzip` 的文件, 这样做可以节省系统的磁盘空间。

为了阅读这些文件, 一般情况下在阅读之前需要解压缩, 这样需要不断维护硬盘解压缩的文件。

使用 `zless` 不需要解压缩 HOWTO 和 FAQ 文件, 即可直接阅读这些压缩的文件, 这样可以减少维护硬盘的负担。

笔者在此以 Red Hat Linux 9 上的一个压缩的 FAQ 文件 `/usr/share/man/man3/Net::libnetFAQ.3pm.gz` 为例, 说明如何使用 `zless` 查看 FAQ 和 HOWTO 压缩文件。命令如下:

```
$ zless /usr/share/man/man3/Net::libnetFAQ.3pm.gz
```

`zless` 使用和 `less` 相同, 所以可以在 HOWTO 中容易翻页。

对于文本的 HOWTO 和 FAQ, 可以直接用 `more` 或者 `less` 命令来阅读。

对于文本的 HTML 格式的 HOWTO 和 FAQ, 用户可以使用他们喜欢的浏览器来看。

### 2.6.4 slocate 命令

slocate 提供了一种安全的方法为系统的文件建立索引，同时用户可以通过 slocate 快速搜索到系统中的文件。它使用了与 locate 类似的增量编码方法来压缩它的索引数据库，极大的提高了它的搜索速度，它同时包含了文件的许可权和所有关系，使得用户看不见它们无权访问的文件。

slocate 提供了两大功能：

- 创建文件索引数据库。它会定时或者手工刷新索引数据库。
- 在数据库中搜索指定文件。存在索引数据库，因此它的检索速度非常快，能够满足对所有文件系统快速检索文件的需要。如果不知道一个命令和文件的全名，用户可以使用 slocate 命令来获得全名和位置。

slocate 的常用参数如表 2-3 所示。

表 2-3 slocate 的常用参数

参 数	描 述
-u	从根目录开始建立索引数据库
-U <i>dir</i>	从指定的目录 <i>dir</i> 建立索引数据库
-e <i>dirlist</i>	排除指定的目录列表，以逗号分隔
-f <i>filesystemlist</i>	排除指定的文件系统列表，以逗号分隔

slocate 使用的索引数据库通常由系统定期自动地刷新。但如果用户的计算机没有一直运行 Linux 系统，很可能该索引数据库不是最新的。用户可以以根用户身份，使用以下命令来手工刷新索引数据库：

```
# slocate -u -e /mnt
```

**注意：**这条命令需要花一定的时间，具体的时间要依赖用户的系统的性能和需要建立索引的文件的数量。

如果用户想查找所有的 mkswap 文件，用户可以执行：

```
$ slocate mkswap
```

它的输出可能是：

```
/usr/share/man/man8/mkswap.8.gz
/sbin/mkswap
```

**注意：**locate 不仅返回文件名，而且返回目录名。所以如果用户有一个 /home/ 区一体化 root/swap/ 目录，它将返回这个目录的所有文件。

### 2.6.5 info 页

man 是最普遍的文档格式，而 info 则更有力。它提供了超文本连接使大的文档更易于阅读，同时为文档的作者提供了许多特性。在 Red Hat Linux 的不同方面（特别是 GNU 部分），有一些非常完美的 info 文档。

阅读 info 文档可以使用 info 程序，它不带参数。它将列出一个文档的清单。如果用户没有发现用户所需要的文档，那是因为用户没有安装包含那个文档的软件包。用 RPM 安装

对应的软件包之后再试。

注意：emacs 有一个内置的 info 文档浏览器。使用 [Ctrl-h]+[i] 键就可以进入 info 浏览器。

info 系统是一个超文本系统，任何高亮度显示的文字，都有一个连接导向更多的信息。常用的快捷键的功能如下：

- 使用 Tab 键将光标移到连接。
- 按 Enter 键进入光标所指的连接。
- 按 p 键返回上一页，按 n 键翻到下一页。
- 按 u 键回到文档的上一层。
- 按 d 键返回到主菜单。
- 要退出 info，首先按 Ctrl+X 组合键，然后再按 Ctrl+C 组合键；或者按 q 键退出 info。

学习如何使用 info 的最佳方法是用它阅读 info 文档。如果用户阅读了 info 显示的第一页，将有能力继续阅读所有的文档。

例：用户使用 info 命令查看 man 命令的帮助信息，命令如下：

```
$ info man
```

其结果是如图 2-28 所示的 info 页。

```
File: *manpages*. Node: man, Up: (dir)

man(1) man(1)
NAME
  man - format and display the on-line manual pages
  manpath - determine user's search path for man pages

SYNOPSIS
  man [-andfFhkkltwW] [-path] [-m system] [-p string] [-C config_file]
  [-M pathlist] [-P pager] [-S section_list] (section) name ...

DESCRIPTION
  man formats and displays the on-line manual pages. If you specify sec-
  tion, man only looks in that section of the manual. name is normally
  the name of the manual page, which is typically the name of a command,
  function, or file. However, if name contains a slash (/) then man
  interprets it as a file specification, so that you can do man ./foo.5
  or even man /cd/foo/bar.1.gz.

  See below for a description of where man looks for the manual page
  files.

Info: (*manpages*)man, 346 行 --Top
欢迎使用 info 4.3 版，输入 C-h 以获得帮助，n 将得到菜单。
```

图 2-28 info 使用实例

## 思考与练习

- (1) KDE 有哪些快捷键？
- (2) 在 Konsole 中，汉字有重影，如何解决？
- (3) 关闭系统的方式有哪些？它们的区别和各自的应用场合是什么？
- (4) 如何在 Linux 系统中上网，收发邮件和获得系统帮助？

## 第3章 使用Linux的基本命令

Linux 命令之所以功能强大,除了因为它具有优秀的内核以外,还得益于它拥有丰富的与 UNIX 功能类似的命令。这些命令可以满足 Linux 的绝大部分系统管理和用户应用的需要并能够极大地提高用户的工作效率。由于众多的 Linux 命令是基于文本模式的,因此用户很容易通过各种连接方式远程使用 Linux 系统。

由于 Linux 提供的命令种类繁多,而且大多数命令的参数和选项都比较复杂,对于初学者和有一些经验的用户来说,首先需要去记忆大量的命令参数,这就给学习和使用 Linux 的人带来了困难。

本章将以实例的形式来讲述那些最常见的命令,以及正确使用这些命令时所需要的各种参数,同时略去了那些对用户来说并不是很常用的命令,这样便于用户通过实践循序渐进地掌握 Linux 的精华,有望在短时期内成为 Linux 的高手。

本章讲述的命令主要包括进行文件操作、目录操作、确定系统状态时涉及到的一些常见命令。其余的将在后续的章节中讲述。

### 3.1 如何在 Linux 中运行程序

如果用户想使用 Linux,就必须使用某种方式登录到 Linux 系统,常见的登录方式包括虚拟终端的方式、控制台方式和普通终端方式。普通用户使用的 Linux,大多是基于 PC 机的,并没有安装终端,因此需要在本地或者通过网络登录,采用伪终端方式访问系统。

本节首先讲述了 Linux 的登录方式和工作过程,接着讲述了 Linux 执行用户提交命令的工作原理,并以实例讲述了用户如何登录和使用 Linux,最后介绍了 Linux 提供的一部分基本的实用程序(或者称为命令)。

#### 3.1.1 登录、注销 Linux

出于安全的原因,在系统登录时会提示用户输入用户账号和密码,只有当输入的用户账号和密码通过验证后, Linux 才允许用户进入系统。

用户可以使用图形化的方法来登录和注销系统。笔者将在此讲述用户如何使用字符终端从远程或者本地登录到 Red Hat Linux 9 系统。

在 Red Hat Linux 9 中,用户登录的过程如下:

##### (1) 验证用户账号。

当用户试图从本地或者远程登录系统时,在屏幕左上角会出现如下的一些简单系统说明文本:

```
Red Hat Linux Release 9 (Shrike)
Kernel 2.4.20-8 on an i686
```

它说明该系统是 Red Hat Linux 9,其内核版本为 2.4.20-8。

注意：如果用户升级或者定制过内核，那么内核的版本号可能不是 2.4.20-8。

然后出现系统登录的提示符，举例如下：

```
jiang login:
```

其中，jiang 表示用户所登录的 Red Hat Linux 的主机名，当然，你可以根据显示的主机名来判断你现在正在登录到哪一台计算机。

用户需要在此输入用户账号，通过指定的账号名（即用户登录名，如根用户的账号为 root）来验证身份。在大多数系统上，登录名为小写字母形式。

输入正确的账号名之后，按回车键，完成本步操作。

（2）输入密码。

无论输入的用户账号是否正确，系统都要求用户输入密码，系统显示如下的提示行：

```
Password:
```

用户需要在此输入创建帐户时的密码或者系统管理员提供给用户的密码。系统为了保护密码不被泄漏，当你输入密码时，密码并不显示在屏幕上。这样其他人就无法知道你正在输入的密码是什么，甚至不知道你输入的密码的长度。

输入密码后，按 Enter（回车）键结束。

如果用户正确地输入了用户名和密码，就可以成功登录进入 Linux 系统，然后出现一个 Shell 提示符。否则，系统将会显示如下的错误信息：

```
login incorrect
```

用户可能由于如下的一些常见错误，导致不能正常登录：

- 混淆了数字 1 和字母 l，或者混淆了数字 0（零）和字母 o（或者大写字母 O）。
- 混淆了字母的大小写。
- 记错了用户账号和密码。

（3）如果你认为是输入错误，可以重新输入用户账号和密码。

（4）按 Ctrl+D 键，结束登录过程，登录程序的另一个实例立即启动，显示新的登录提示，用户可以再次登录。

（5）登录成功后，会出现一个字符终端窗口，并出现一个如下所示的 Shell 提示符。

```
[globus@toshiba globus]$
```

为了安全起见，当你完成工作之后，应该及时注销，以避免其他人以该用户账号身份，从而破坏系统，或者避免他人获得系统中的敏感信息。

用户可以使用如下两种方法完成账号注销：

（1）在 shell 提示符下输入 exit，然后按回车键，完成注销。

（2）按 Ctrl+D 键完成注销。

在字符终端下，它将再次出现登录提示，可以重新登录后进入系统。

当注销图形化的终端后，该终端窗口将关闭，如果重新登录，你需要重新启动一个 X 终端。

### 3.1.2 Linux 中的 shell 简介

当用户登录进入 Linux 之后，第一个接触到的便是 shell。笔者在此对它们作一个简单的介绍，以便读者对它有一个初步的认识。



shell 本身是一个用 C 语言编写的程序，它是用户使用 Linux 的桥梁。shell 既是一种命令语言，又是一种程序设计语言。作为命令语言，它交互式地解释和执行用户输入的命令；作为程序设计语言，它定义了各种变量和参数，并提供了许多在高级语言中才具有的控制结构，包括循环和分支。它虽然不是 Linux 系统核心的一部分，但它调用了系统核心的大部分功能来执行程序、建立文件并以并发的方式调度各个程序的运行。因此，对于用户来说，shell 是最重要的实用程序，深入了解和熟练掌握 shell 的特性及其使用方法，是精通 Linux 系统的关键。可以说，shell 使用的熟练程度反映了用户对 Linux 使用的熟练程度。

当用户使用 Linux 时，是通过提交命令来完成所需工作的。一条命令是用户和 shell 之间对话的一个基本单位，它是由多个字符组成并以换行结束的字符串。shell 解释用户输入的命令，就像 DOS 里的 `command.com` 所做的一样。不同的是，在 DOS 中，`command.com` 只有一个，而在 Linux 下比较流行的 shell 有多个，每个 shell 都各有千秋。一般的 Linux 系统都将 `bash` 作为默认的 shell。它也是我们要学习的主要对象。

目前流行的 shell 有：

- `sh` (Bourne shell)
- `csh` (C shell)
- `ksh` (Korn shell, 商业软件)
- `tcsh` (免费)
- `bash` (Bourne Again shell, GNU 的软件之一)

#### 1. `sh`

第一个重要的 shell 是 Bourne shell (如此命名是为了纪念此 shell 的发明者 Steen Bourne)。1979 年，第一个流行的 Unix 版本 7 发行时，开始使用 Bourne shell。Bourne shell 的程序名为 `sh`，因此日后用户便以 `sh` 为 Bourne shell 的主要识别名称。虽然 Unix 上的 shell 有许多种，但 Bourne shell 的地位至今仍然没有改变。许多 Unix 系统中仍然使用 `sh` 作为重要的管理工具。

#### 2. `csh`

C shell 是另一个广为流行的 shell 变种。C shell 主要附在 BSD 版的 Unix 系统中。它的作者是伯克利大学的 Bill Joy。C shell 的语法和 C 语言相类似，因此而得名。这使得 Unix 系统的程式师在学习 C shell 时感到相当方便。

以上这两种 shell 是两大主流，后来的变种 shell 大都吸取这两种 shell 的优点，它们包括 Korn, `tcsh` 及 `bash`。

#### 3. `ksh`

korn shell 是 bourne shell 的增强，它有很多不错的特性，使 shell script 简单高效而且更快。

#### 4. `tcsh`

`tcsh` 是 Linux 比较大的 shell，它由以 William Joy 为代表的共计 47 位作者编成，共有

52 个内部命令。在 Linux 中, `csch` 是指向 `/bin/tcsh` 的一个 shell, 也就是说, `csch` 其实就是 `tcsh`。

在 Linux 系统中, 最常用的 shell 是 `bash`, 其次是 `csch`, 在后续章节中, 笔者将重点讲述这两种 shell。

不管使用什么 shell, 当用户登录完成后, 都会出现一个 shell 提示符, `bash` 的提示符为:

```
[globus@toshiba globus]$
```

```
[root@toshiba /]#
```

`csch` 的提示符为:

```
[globus@toshiba ~]$
```

```
[root@toshiba /]#
```

在这 3 种 shell 提示符中, `$` 表示登录到该终端的用户是普通用户, 而不是根 (`root`) 用户。`#` 表示登录用户是超级用户, 即根 (`root`) 用户。`@` 左边的字符串表示当前登录的用户账号, 对于 `[globus@toshiba globus]$`, 表示当前用户账号是 `globus`, 对于 `[root@toshiba root]#`, 表示当前用户是 `root` (根用户)。`@` 右边的字符串是所登录系统的主机名, 本例的主机名为 `toshiba`, 提示符最右边的字符串表示用户所在的当前目录。对于 `[root@toshiba /]#`, 表示当前目录为 `/` (根目录)。对于 `[globus@toshiba ~]$`, 当前目录为 `~`, 它表示当前目录为用户的主目录。表 3-1 所示为 shell 提示符 `[root@toshiba /]#` 解释。

**注意:** 如果你登录后发现出现 `$` 提示符, 说明你是以普通用户登录的。如果出现 `#` 提示符, 则说明你是以根用户登录的。

表3-1 shell提示符[root@toshiba /]#解释

Shell 提示符的组成	解释
[	shell提示符的左分界号
root	登录的用户账号, 在此为根用户
@	用户账号与主机的分隔符
toshiba	系统的主机名
/	当前目录为根目录
]	shell 提示符的右分界号

用户可以在 shell 提示符下输入各种命令, 以便完成所需要的工作。可以用 `echo $SHELL` 命令来查看用户自己的 shell 类型, 例如执行如下命令:

```
#echo $SHELL
```

其结果如下, 表示当前用户使用的 shell 为 `bash`:

```
bash
```

`$SHELL` 是一个环境变量, 它记录用户所使用的 shell 类型。

**注意:** 如果没有特别的说明, 本书假定使用的 shell 为 `bash`。

## 5. bash

`bash` shell 具有以下功能:

- 与 Bourne shell(`sh`)兼容。
- 包含 C shell 以及 Korn shell 中最有用的功能。

- 具有命令历史记忆能力（与 DOS 的 doskey 类似）。
- 具有工作控制（job control）能力，可控制前台及后台进程。
- 具有 shell 程式设计的能力，可让用户定制 shell，以及进行程序设计和系统管理。

注意：如果你想获得新版的 bash，可以访问 <http://www.gnu.org> 或其 mirror 网站。

例如，可以在 shell 提示符下，输入 ls，将会列出当前目录中的所有文件：

```
[globus@toshiba globus]$ ls
```

其中[globus@toshiba globus]\$为系统提示符，ls 是显示目录文件的命令，执行结果如下：

```
401.jpg          jzh.aux          readme.txt
amsn_received    jzh.dvi          Target
bin              jzh.log          temp.txt
Desktop          jzh.pdf          test.txt
documents        jzh.tex          tools
foxpro8_文件     mbox             w4l-install.log
gbkfonts-linux-0.3 mcc18_文件       whichrpm
intel-537EP_secure-2.27.39.0-R90 missfont.log
jbpproject       netraverse_installer
```

### 3.1.3 shell 的工作原理和过程

Linux 是一个多任务操作系统，可同时运行许多程序，每执行一个命令，就会有一个子进程在运行，系统给它分配内存，定位要运行的代码，并给 CPU 发送请求信号。无论何时请求执行命令，都是在请求 shell 启动一个或者多个进程来运行命令对应的程序代码。

人们在使用 Linux 时，也许认为 shell 本身就能够确定所有已登录的用户，计算系统的日期和日历等。实际上，shell 对这些事情一无所知，当输入 date 命令，并按回车键后，是指示 shell 去运行 date 程序。date 程序是不同于 shell 的独立程序。shell 仅仅知道怎样启动一个新进程，确定用户输入的命令程序的位置，以及如何在新进程中根据用户传递的参数，并执行这个程序。

按回车键表明命令行输入结束，shell 将启动一个新的进程。shell 将命令行中的实用程序名后面的参数传递给新进程，并找到命令的代码，最后让新产生的进程执行该代码。当 shell 启动了 date 程序之后，子进程将按 date 的代码执行。date 程序检查系统的时钟，并将结果格式化后输出，默认情况下，如果没有请求 shell 将结果输出连接到一个文件，或者另一个可执行程序时，date 进程将结果连接到显示器（在显示器显示）。

每一个终端、监视器或者网络都通过一根与计算机相连的电缆连接在计算机上，端口是计算机后面的一个物理位置。每一个端口都有一个名称，后面加一个数字组成。:0 表示与主机相接的第一台显示器。tty1 和 tty2 等表示与主机相接的第一个终端和第二个终端。pts/0、pts/1 和 pts/2 分别表示第 0 个、第 1 个和第 2 个伪终端。伪终端是在本地打开的或者通过网络连接的终端。

date 程序完成它的工作后，它将退出并通知 shell，执行过程一切正常。然后 shell 将显示新的提示符，表示它已经就绪，并等待接收用户输入的下一条命令。

shell 与系统的交互过程如下：

(1) 登录或者启动终端窗口后, shell 显示提示符, 表明 shell 已经就绪, 准备接收命令。

(2) 输入一条命令, 然后按回车键。

(3) 通常情况下, shell 启动一个新的子进程, 在该子进程的环境中, 执行该命令对应的程序代码。

(4) 代码执行结束后, 进程通知 shell, 它的任务已完成, 子进程中止。

(5) shell 一旦接收到退出代码, shell 将显示下一个提示符, 并等待接收下一条命令。

用户经常会遇到输入错误, 当出现输入错误后, shell 应该提供用户改正错误的机会。根据终端的位置的不同, 用户可以采用如下方法改正输入的错误字符或者字符串。

在 Linux 中, 如果在本地的 shell 上出现了输入错误, 而且还没有按回车键确认, 那么用户可以使用 BackSpace (退格键) 或者 Ctrl+H 键来删除光标前的一个字符, 使光标回退。用户也可以使用左和右方向键将光标移到输入错误的字符下面, 然后按 Delete 键删除光标下的一个字符。同样也能在光标的位置插入一个或者是多个字符。


如果使用网络登录到 Linux, 远程终端可能不能正确地使用 BackSpace (退格) 键, 这时用户只能使用 Ctrl+H 键来删除字符, 然后再输入正确的字符。

#### 3.1.4 使用 Windows 的超级终端远程登录到 Linux

笔者在此以 Windows 的超级终端登录到远程的 Linux 为例, 说明远程用户如何使用 Linux。

Windows 系统都提供名为“超级终端”的远程终端程序, 用户可以使用它来登录到远程的 Linux 系统。用户在 Windows 下通过选择“开始菜单”|“程序”|“附件”|“通讯”|“超级终端”菜单项, 启动“超级终端”程序。

假定用户网络使用 TCP/IP 连接远程 Linux 系统, 这台 Linux 主机可以在互联网上, 也可以在局域网上。

超级终端启动后, 出现如图 3-1 所示的“连接描述”对话框, 在对话框的“名称”文本框中输入该连接的描述, 本例在此处输入 192.168.1.19, 在“图标”图形列表框中选择该连接的图标, 在此使用默认的图标。然后单击“确定”按钮继续。

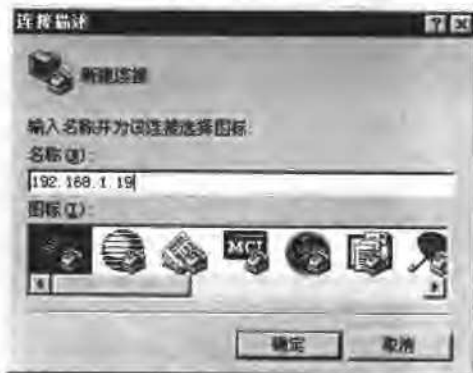


图 3-1 “连接描述”对话框

然后会弹出如图 3-2 所示的“连接到”对话框。在该对话框中，默认的地区为“中华人民共和国 (86)”，要求你输入区号和电话号码，“连接时使用”下拉列表框中，默认的使用调制解调器连接到远程系统，因此这时的电话号码和区号是必须的。你可以根据具体的网络情况选择：调制解调器、com3 串行端口或者 TCP/IP。在此使用局域网连接方式，因此选 TCP/IP (Winsock)。当选中该选项之后，图 3-2 所示的“连接到”对话框变为如图 3-3 所示的对话框。

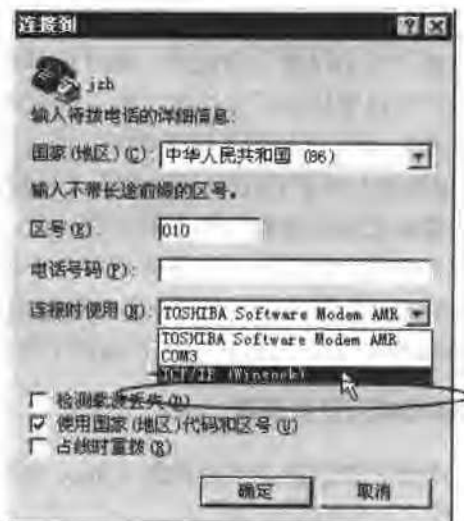


图 3-2 “连接到”对话框 1

在图 3-3 所示的“连接到”对话框中，在“主机地址”文本框中输入你需要登录的远程 Red Hat Linux 9 主机的 IP 地址，在本例中在“主机地址”文本框中输入 192.168.1.19。然后还需要在“端口号”文本框中指定连接到远程主机的 TCP/IP 端口号，它的默认值为 23，使用 Telnet 协议连接远程 Red Hat Linux 主机。本例使用 Telnet 协议，因此采用默认的端口号 (23)。最后单击“确定”按钮，开始远程连接过程。



图 3-3 “连接到”对话框 2

远程连接过程如图 3-4 所示, 首先出现 Red Hat Linux 版本的说明信息, 然后出现登录提示符。在 login: 提示后面输入远程主机中的一个用户账号, 本例输入用户名 globus。然后在 Password: 提示后输入该用户的口令, 在此输入 globus 用户的口令。如果用户账号和口令都正确无误, 登录将会成功, 然后出现 Linux 的 shell 提示符, 如图 3-5 所示。你的 Red Hat Linux 中的用户账号可能没有 globus, 可以使用其他可用普通用户账号。



图 3-4 远程登录过程



图 3-5 远程登录成功后的界面

**注意:** IP 地址为 192.168.1.19 的主机上必须安装和运行 telnet 服务器软件。否则用户无法从任何远程终端上通过网络使用 telnet 协议登录到该 Red Hat Linux 系统上。

出于安全方面的考虑，用户不能从远程终端中以超级用户（root）身份直接登录到远程系统。

如果你必须使用根用户完成特定的工作，你可以首先以一个普通用户登录到远程 Linux 系统，然后输入 `su` 或者 `su -` 切换到根用户。有关 `su` 的详细信息，请参见后续的一些章节的介绍。

也可以在一台运行 Linux 或者 Unix 的系统使用 `telnet` 命令登录到远程的 Linux 系统，其前提是远程系统必须安装了 `telnet` 服务器软件。

还可以使用其他远程登录方式（如 `ssh`），有关 `ssh`，将在后续章节介绍。

从该登录成功后出现的提示符可以判定，当前用户是 `globus`，当前目录也是 `globus`，远程机器的主机名为 `unkown`。

用户可以在该超级终端中，输入 Linux 命令，例如在 `shell` 提示符后输入 Linux 命令 `whoami`，以确定该终端上的用户信息。

运行结果如图 3-6 所示。

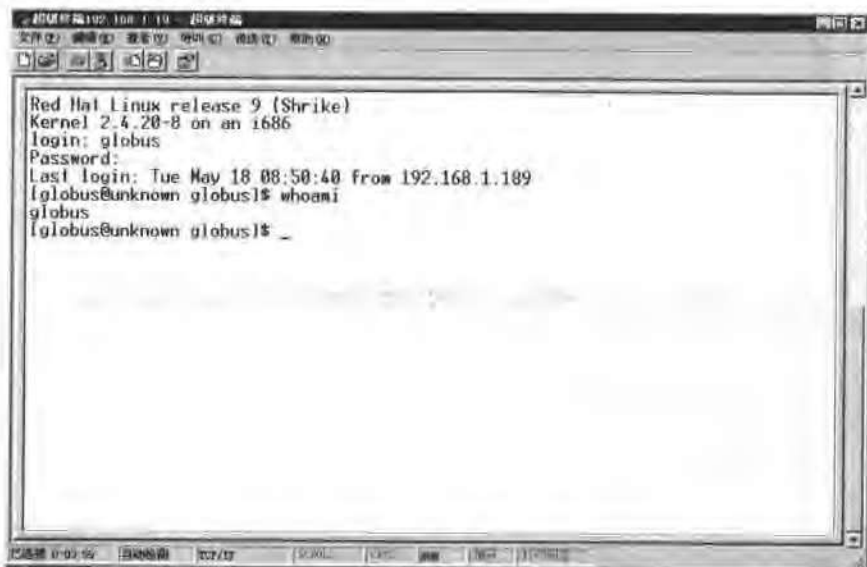


图 3-6 远程终端执行命令实例

系统显示在该终端上登录的用户账号为 `globus`。

用户可以通过“超级终端”在远程 Red Hat Linux 主机上执行多条命令，完成的任务之后，使用 `exit` 或者 `Ctrl+H` 组合键注销。

用户可以对“超级终端”进行设置，单击“超级终端”主菜单的“文件”|“属性”菜单项。在弹出的“属性”对话框中，选择“设置”选项卡，可以对“功能键、箭头键和 `Ctrl` 键用作”进行设置，默认用作“终端键”，用户也可以设置成“Windows 键”。

对于用户在超级终端所按的退格键，超级终端可以以 `Ctrl+H`、`Del` 或者 `Ctrl+H, Space`，`Ctrl+H` 键发送，在“终端仿真”下拉列表中选用不同的仿真终端类型，如图 3-7 所示。



图 3-7 设置超级终端

### 3.1.5 Linux 系统的最基本命令

笔者在此讲述了 Linux 一些最常用的基本命令，目的是让用户熟悉 Linux 命令，并且为后继学习打好基础。这些命令如下：

#### 1. who —— 确定系统中当前用户的信息

Linux 是一个多用户环境，经常需要查看某一用户是否已经登录。

在 shell 提示符后，输入小写字母 who，然后按回车键，将启动 who 程序。

who 程序将显示当前登录用户的列表。举例如下：

```
[globus@jiang globus]$ who
globus  :0          May 17 08:17
globus  pts/0       May 17 08:17
globus  pts/1       May 17 08:18
globus  pts/2       May 17 08:39 (192.168.1.189)
```

刚刚输入的命令指示 shell 执行一个名为 who 的程序（或者称为命令）。程序 who 将确定登录在该系统的所有活动用户，并将那些登录的用户的相关信息格式化输出。在这个例子中，who 的输出如上述结果。结果说明系统上只有一个用户（即 globus）处于活动状态，而且它正在进行 4 个 shell 会话。

识别 who 的输出字段如表 3-2 所示。who 的每一行输出是一个当前用户的一个 shell 会话信息。每一行由多个字段组成，各个字段被一个或者多个空格隔开。

表 3-2 who 输出的第四行的条目组成

字段	描述
Globus	用户登录名
pts/2	访问端口
May 17 08:39	登录的日期和时间
(192.168.1.189)	从远程系统登录的 IP 地址



2. ps —— 列出系统中的进程

输入 ps 命令，按回车键，得到与当前用户相关的进程列表：

```
[globus@jiang globus]$ ps
  PID TTY          TIME CMD
 2872 pts/2    00:00:00 bash
 3148 pts/2    00:00:00 ps
```

输出结果是当前用户账号相关的进程列表，它的每一行是一个进程的信息，每一行中包含 4 个字段。含义如表 3-3 所示。

表3-3 ps命令输出字段的含义

字段	含义
PID	该进程的ID号
TTY	进程隶属的终端号和端口
TIME	进程创建的时间
CMD	该进程执行的命令

每一次执行 ps，它的输出至少有两个进程组成：shell 进程和 ps 进程。在上例中的 shell 进程名为 bash。ps 命令显示的进程都是执行 ps 命令那一刻正在系统中运行的进程。

ps 完成工作之后，它显示进程信息，然后退出。

当你再次运行同一个程序时，一个新进程开始运行。例如每次运行 ps，可以看到 ps 进程的进程 ID 号是不同的，表明这个进程与原来执行相同程序的进程不是同一个进程。

也可以使用 ps 列出系统中所有的进程，在上述的练习中，ps 只列出了与当前账号相关的进程的状态信息。你也可以显示整个系统上所有进程的信息，包括当前用户的所有进程和系统中所有的其他进程。

可以使用下述的命令，显示系统中所有的进程信息：

```
ps -aux
```

或者

```
ps -ef
```

这两条命令都显示一个很长的进程列表，由于进程的数目较多，一部分进程的信息可能会滚出屏幕。

ps 的输出信息包含了许多对大多数用户不需要的多余信息，它是整个系统上每一个正在运行的进程的状态信息表。列表中的许多进程是系统进程，这些进程提供了系统的所有服务，如 http 服务器，ssh 服务器和打印服务器等。这些信息对于系统管理员和其他监视 Linux 的系统的人来说可能是很重要的。尤其是在诊断系统问题时，它对非超级用户来说同样也是很重要的。

3. > —— 将程序的输出结果重定向到文件

当 shell 接到用户执行 who 程序的通知后，who 程序的代码将执行，并输出当前用户的列表。运行 who 程序的进程找到需要的信息，并将这些信息格式化。最后将格式化的信息默认的输出到屏幕。

可以指示 shell 将实用程序的输出从屏幕重定向到一个文件。

在 shell 中执行 `who > login-users`, 执行过程如下:

```
[globus@jiang globus]$ who > login-users
[globus@jiang globus]$
```

执行完该命令之后, 除了出现一个新的提示符以外, 屏幕上没有任何输出信息, 则表明该命令成功执行。该命令行中各个组成部分的意义如表 3-4 所示。

表3-4 shell对who > login-users命令的各部分解释

命令的组成	意义
Who	指示shell启动一个新进程, 以运行程序who的代码
>	指示shell创建一个新文件, 并将进程的输出连接到该文件
login-users	分配给新文件的文件名, 该文件接受who的输出

上述命令指示 shell 创建一个文件名 `login-users` 的新文件, 并将它连接到程序 `who` 的进程的输出。结果 `login-users` 文件的信息与执行 `who` 命令显示在屏幕上的信息相同。

注意: 在 Linux 和 Unix 系统中, 命令执行后不显示任何信息默认表示命令成功执行, 否则会给出错误提示。

#### 4. Date —— 显示和设置系统的当前日期和时间

利用 `date` 命令方便地显示系统日期和时间的程序, 同时它还能设置系统的日期和时间。`date` 命令常用的选项和参数如表 3-5 所示。

表3-5 date命令的常用参数和选项

命令的组成	意义
<code>-d datestr</code> 或者 <code>--date datestr</code>	显示由datestr描述的日期
<code>-s datestr</code> 或者 <code>--set datestr</code>	设置由datestr描述的日期
<code>-u</code> 或者 <code>--universal</code>	显示或设置通用时间
<code>+formatstr</code>	按指定的formatstr字符串显示日期或者时间

注意: 在 Linux 中, 一个程序中功能相同的选项通常由两个表达方法, 其中的一个是短的选项, 通常使用一个“-”后加一个字母表示, 它与 Unix 系统对应命令的对应选项表示方法相同, 另一种表达方式被称为长选项, 它通常使用“--”后加一个英文单词表示。这两种方法的功能是一样的, 用户可根据习惯选用其中的一种。例如在表 3-5 中的第三行, 选项“-u”与选项“--universal”功能完全相同。

例: 使用不带参数的 `date` 命令按默认格式显示系统的当前日期和时间。

```
$date
```

```
一 5月 17 08:27:17 CST 2004
```

结果表明系统当前的日期是 2004 年 5 月 17 日, 星期一, 当前时间为 8 点 27 分 17 秒。按指定的描述字符串显示日期和时间, 而不是系统当前的日期和时间。

```
$date -d "... 5月 17 08:43:18 CST 2004"
```

.. 5月 17 08:43:18 CST 2004

-d 参数用该参数之后的字符串设置的时间。参数中的日期是 2004 年 5 月 17 日星期一，参数中的时间为 8 点 43 分 18 秒。

按特定的格式显示指定的日期和时间，举例如下：

```
$date -d "— 5月 17 08:43:18 CST 2004" +%D
05/17/04
```

+%D 表示按 (mm/dd/yy) 格式显示日期，而不显示其他信息。该例中显示的结果为 05/17/04，表示当前的月份为 5 月，日期为 17 号，年份为 2004 年，其中月日年分别用两位数字表示。

+formatstring 指定了按+后面的 formatstring 字符串指定的格式显示日期。可用格式如表 3-6 所示。

表3-6 date命令中的显示格式字符串的组成元素

日期/时间	格式	意义
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 (00~59)
	%p	显示出AM或PM
	%r	时间 (hh:mm:ss AM或PM)， 12小时
	%s	从1970年1月1日00:00:00到现在经历的秒数
	%S	秒 (00..59)
	%T	时间 (24小时制) (hh:mm:ss)
日 期 域	%X	显示时间的格式 (%H:%M:%S)
	%Z	时区
	%a	星期几的简称 (Sun~Sat)
	%A	星期几的全称 (Sunday~Saturday)
	%b	月的简称 (Jan~Dec)
	%B	月的全称 (January~December)
	%c	日期和时间( Mon Nov 8 14:12:46 CST 1999)
	%d	一个月的第几天 (01~31)
	%D	日期 (mm/dd/yy)
	%h	与%b选项等价
	%j	一年的第几天 (001~366)
	%m	月 (01~12)
	%w	一个星期的第几天 (0代表星期天)
	%W	一年的第几个星期 (00~53, 星期一为第1天)
	%x	显示日期的格式 (mm/dd/yy)
	%y	年的最后两个数字 (2004则是04)
	%Y	年 (如1970, 1996等)

以“XXXX年XX月XX日XX, XX时XX分XX秒”的格式显示系统的当前时间, 命令如下:

```
$date +%Y年%m月%d日, %H时%M分%S秒
2004年05月17, 05时10分12秒
```

将系统的时间设置成9点56秒, 命令如下:

```
# date -s "09:00:56"
— 5月 17 09:00:56 CST 2004
```

当该命令执行完成后, 会显示出你设定的日期和时间。

**注意:** 只有超级用户才有权限使用 date 命令设置系统日期和时间, 一般用户只能使用 date 命令显示时间。否则会出现如下的错误提示:

```
date: cannot set date: 不允许的操作
```

### 5. cal —— 日历查看命令

cal 命令的功能是显示某年某月的日历。

该命令的一般格式为: . cal [选项] [月] [年]

命令中各选项的含义为:

- -j 显示出给定月中的每一天是一年中的第几天 (从1月1日算起)。
- -y 显示出整年的日历。

显示当年当月的日历, 命令如下:

```
$cal
五月 2004
日 一 二 三 四 五 六
                        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
```

该命令显示了当月 (例中当前月份为2004年5月) 的日历, 从该日历可以很容易知道该月的每一天是星期几, 如5月28日是星期五。

如果显示2003年全年的日历, 命令如下:

```
$cal 2003
```

该命令将在终端上依次显示2003年12个月的日历, 排成4行3列的矩阵。

如果显示2003年3月的日历, 命令如下:

```
$cal 3 2003
三月 2003
日 一 二 三 四 五 六
                        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

```

如果 `cal` 带两个参数，则第 1 个参数为月份，第 2 个参数为年份。

例如，显示出 2003 年 3 月中的每一天是一年中的第几天（从 1 月 1 日算起），需要在 `cal` 程序中需要使用 `-j` 参数。命令如下：

```

$cal -j 3 2003
      三月 2003
日 一 二 三 四 五 六
                        60
61 62 63 64 65 66 67
68 69 70 71 72 73 74
75 76 77 78 79 80 81
82 83 84 85 86 87 88
89 90

```

由此可见，2003 年 3 月 1 日为 2003 年的第 60 天，依次类推。

注意：`cal 2003` 与 `cal 03` 的显示结果是完全不同的。`cal 2003` 显示的是 2003 年的日历，而 `cal 03` 则显示的是公元 03 年的日历，读者在使用时务必要注意它们之间的区别。

## 6. echo —— 回显命令

`echo` 命令的功能是在显示器上显示一段文字，一般起一个提示作用。

该命令的一般格式为：`echo [-n] 字符串`

其中选项 `n` 表示输出文字后不换行；字符串可以加引号，也可以不加引号。用 `echo` 命令输出加引号的字符串时，将字符串原样输出；用 `echo` 命令输出不加引号的字符串时，将字符串中的各个单词作为字符串输出，各单词之间用一个空格分隔。

不加引号的 `echo` 举例如下：

```

$echo this is a book.
this is a book.

```

加引号的 `echo` 举例如下：

```

$echo "this is a book."
this is a book.

```

带 `-n` 选项与不带 `-n` 选项的 `echo` 命令比较如下：

如果不带 `-n` 选项参数，执行完 `echo` 命令后，会在显示的结果的下一行出现一个提示符。

```

[globus@jiang globus]$ echo this is a book.
this is a book.
[globus@jiang globus]$

```

带 `-n` 选项的 `echo` 命令，将在 `echo` 的执行结果之后不换行，而直接显示一个 shell 提示

符。

```
$echo -n this is a book.  
this is a book.[globus@jiang globus]$
```

### 7. whoami —— 显示当前用户账号

用户可能在一个系统中不断地使用几个不同的用户账号，进行不同的操作，因此当用户准备执行一条命令时，常常需要确定当前用户账号是什么。那么可以使用 `whoami` 显示当前的用户账号。

在命令提示符中，输入 `whoami` 程序，然后按回车键：

```
[globus@jiang globus]$whoami  
globus
```

结果表明当前用户是 `globus`。

### 8. ls —— 获得目录中的文件列表

用户可以使用 `ls` 来确定指定目录中包含的文件和子目录列表。它可以带参数，而且有多个选项。

执行 `ls` 显示当前目录中的文件列表，执行如下：

```
[globus@unknown globus]$ ls  
[globus@unknown globus]$
```

结果表明，当前目录中没有文件。

使用 `ls /` 显示根目录下的文件列表，执行如下：

```
[globus@unknown globus]$ ls /  
bin  dev  home  lib      misc  opt   root  tftpboot  usr  
boot  etc  initrd  lost+found  mnt   proc  sbin  tmp      var
```

**注意：**在 Linux 的文字终端中，使用 `ls` 命令显示的文件列表中，目录和文件采用了不同的颜色区分。子目录名以蓝色显示，文件白色显示，终端的背景默认颜色为黑色。

### 9. pwd —— 查看当前目录

当目录层次较多，而且有相同的子目录名时，用户可能不知道自己所处的当前目录是什么，`pwd` 命令会显示出你现在所处的目录位置。

执行 `pwd` 命令，显示当前用户的当前工作目录，执行结果如下：

```
[globus@unknown globus]$ pwd  
/home/globus
```

结果显示用户的当前目录为 `/home/globus`，它是从根目录开始的全路径。

### 10. cat —— 显示文件全部内容命令

`cat` 可以将一个指定的文件显示到屏幕上，如果被显示的文件很大，那么它只会将文件的最后一屏显示在屏幕上。在此介绍的是它的最简单用法，有关 `cat` 的详细信息，请参见

本章的后续内容。

使用 `cat /etc/fstab` 显示 `/etc/fstab` 文件的所有内容，执行结果如下：

```
[globus@unknown globus]$ cat /etc/fstab
LABEL=/          /              ext3    defaults    1 1
none             /dev/pts       devpts  gid=5,mode=620 0 0
none             /proc          proc    defaults    0 0
none             /dev/shm       tmpfs   defaults    0 0
/dev/hda2        swap           swap    defaults    0 0
/dev/cdrom       /mnt/cdrom     udf,iso9660 noauto,owner,kudzu,r
o 0 0
/dev/fd0         /mnt/floppy    auto    noauto,owner,kudzu 0 0
```

该命令将 `/etc/fstab` 文件的全部内容都显示在屏幕上。

### 11. bc —— 文字统计程序

`bc` 可以仅仅统计指定文件中所包含的行数、单词数、字节数以及它们的任意组合。默认情况下求出文件中的行数、单词数和字符数。

可用的选项如表 3-7 所示。

表3-7 `wc`命令中的选项

选项	意义
<code>-l</code> 或者 <code>--lines</code>	统计指定文件的行数
<code>-w</code> 或 <code>--words</code>	统计指定文件的单词数
<code>-c</code> 或者 <code>--bytes</code>	统计指定文件的字节数
<code>--help</code>	获得有关 <code>wc</code> 命令的帮助

例：使用 `wc` 命令统计 `/etc/fstab` 文件。

```
[globus@unknown globus]$ wc /etc/fstab
  7      42      545 /etc/fstab
```

结果表明文件 `/etc/fstab` 有 7 行文字，42 个单词，总共包含 545 个字符。

例：使用 `wc -lw /etc/fstab` 只求 `/etc/fstab` 文件包包含的行数和单词数。

```
[globus@unknown globus]$ wc -lw /etc/fstab
  7      42 /etc/fstab
```

### 12. bc —— 计算器工具

`bc` 软件包包括：`bc` 和 `dc` 两个应用程序，`bc` 是一种非常精确的用于处理算术运算的语言。`dc` 是基于堆栈的交互式计算器，是用在文本模式下的。你可以使用 `bc` 进行科学计算。

在命令提示符后输入 `bc`，`bc` 将会显示一段版权信息后进入 `bc` 的界面，光标停留在新行，没有命令提示符。使用程序等待输入。

```
[globus@unknown globus]$ bc
bc 1.06
```

Copyright 1991-1994, 1997, 1998, 2000 Free Software Foundation, Inc.

This is free software with ABSOLUTELY NO WARRANTY.

For details type 'warranty'.

-

可以进行基本的加 (+)、减 (-)、乘 (\*)、除 (/) 和乘方 (^) 运算, 例如在光标处输入 3+5 并按回车键, 屏幕将会显示结果 8。

使用括号能改变运算的次序。

对于除法运算, 如果除数和被除数都是整数, 计算结果也为整数。可以通过输出 `scale=num` 来设置计算的精度, 其中 `num` 为计算结果在小数点后保留的位数。

注意: 在使用 `bc` 程序时输入只能通过键盘、只能输出到屏幕。

以下是使用 `bc` 进行计算的实例, 其中奇数行是输入的计算表达式, 偶数行是计算的结果。

```
3+5
8
144/12
12
144*(4+2)/12
72
scale=2
144*(4+2)/12
72.00
quit
```

注意最后的 `quit`, 指示 `bc` 退出, 返回到提示符。

### 13. `su` —— 以一个替代用户或者组 ID 运行一个 shell

通常情况下进行日常工作时, 建议用户不要以超级用户 (`root` 用户) 登录。由于 `root` 用户具有所有的特权, 因此用户的误操作对系统的影响往往很大, 有时甚至是不可挽回的, 例如进入了根目录, 一个删除操作, 可能会删除文件系统中的所有文件。另外频繁使用 `root` 用户登录, 攻击者攻破 `root` 用户密码的可能性也会增大。

一般的做法是, 一个管理员除了可以使用 `root` 用户账号之外, 还会使用一个具有普通权限的用户账号。除了进行系统管理以外, 管理员都使用普通帐户。为了避免管理员频繁的登录和注销, `Linux` 提供了一个用户切换程序 `su`, `su` 可以在同一系统的任意两个用户之间切换。该命令的实质是改变了对应用户有效的用户和组 ID。

`su` 的语法如下:

```
su [OPTION]... [-] [USER [ARG]...]
```

参数 `USER` 指切换后的用户名, 如果省略 `USER` 参数, 系统就使用 `root` 作为默认的 `USER` 参数。

`su` 常用选项如表 3-8 所示。



表3-8 su命令的常用选项

选项	意义
-i, --login	使这个shell成为登录shell
-c, --command=COMMAND	使用-c或者--command=COMMAND传输一个单 一的命令给shell
-m, -p	不重新设置环境变量
--help	获得有关该命令的帮助

如果在当前用户下，切换到超级用户，进行系统管理，命令如下：

```
[globus@unknown globus]$ su
Password:
[root@unknown globus]# whoami
root
[root@unknown globus]#
```

输入 su 命令之后，系统提示你输入超级用户的口令，当输入的口令验证通过后，它将显示新用户的提示符。在此的用户名显示为 root，可以通过 whoami 命令验证。因为现在的用户是根用户，所以可以输入各种命令，进行系统管理。

如果用户希望退回到原来的账号，用户可以输入 exit 命令。

细心的读者可以从新出现的提示符发现，当前的目录还是 globus，如果希望切换用户之后，新用户的当前目录就是用户的主目录。则可以使用 - 选项。

例：用户从上例中返回到原来的 globus 用户，之后使用 su - 重新以 root 登录到系统。

```
[root@unknown globus]# exit
exit
[globus@unknown globus]$ su -
Password:
[root@unknown root]# pwd
/root
[root@unknown root]#exit
[globus@unknown globus]$
```

从上述的操作过程可知，当使用 su - 切换至 root 用户时，登录后的当前目录也同时切换到了 root 用户的主目录。

使用 su - 命令，当退出时，它将会还原到原来用户的设置，例如当前目录和环境变量。

**注意：**使用 su 命令进行系统管理比注销之后再重新登录更方便。

su 和 su - 命令的差别是很大的，使用 su - 登录成功后不仅将当前目录切换到了用户的主目录，更为重要的是可执行文件的搜索路径也发生了改变，改变成登录后的用户的搜索路径。

建议当需要执行切换后的用户账号搜索路径所指的文件时，使用 su - 命令，否则使用 su 命令。

使用 su 可以切换到任何其他用户。

例：使用 `su` 命令切换至 `henry` 用户。

```
[globus@unknown globus]$ su - henry
Password:
[henry@unknown henry]$ pwd
/home/henry
[henry@unknown henry]$ whoami
henry
[henry@unknown henry]$
```

## 3.2 Linux 中的目录操作

Linux 的文件系统或者目录结构允许通过访问目录树，来创建文件和目录，熟悉 Windows 的用户将会熟悉本节所讲述的目录结构，精心设计文件系统的目录的树状结构能使用户用最少的精力访问他们所需要的信息。

### 3.2.1 使用目录管理文件

目录是 Linux 系统管理文件的一个重要方面，因为在 Linux 系统中几乎所有的东西都是以文件的形式存在，并且所有的文件都是以通过目录访问的。

在现代的文件系统中，一个分区中存放许多不同用途的文件，为了让这些文件不至于相互混淆和容易管理，文件系统都采用分级的文件管理策略，通过在一个分区中创建一个树状目录来实现，在不同的目录中存放不同用途的文件，使每一个目录的文件相对独立，以便用户以目录为单位来处理它。每一个分区在创建时都会产生一个根目录，然后用户或者系统就可以在根目录上创建子目录，在每一个子目录中又可以创建下一级的子目录，这样的目录就构成了一棵目录树，在每一个目录中都可以存放若干的文件。这些文件和目录就构成了文件系统。

文件就是用电磁方法存放在系统硬盘上的信息集合。可以对文件中存储的数据进行访问、修改、复制和删除。实际上，文件并没有实际存放在目录中，它是以数据块的形式存放在硬盘驱动器上，目录仅仅包含每个文件的文件名及其信息节点数。Linux 的信息节点如同图书馆里的“索引卡片”，包含了每个文件所有有用的信息，包括文件在磁盘中的实际存放位置。

在 Linux 和 UNIX 系统中，没有类似于 Windows 盘符的概念，所有的文件都是从/（根目录）开始的。

Red Linux 典型的目录结构如图 3-8 所示。

在图 3-8 左侧的方框里，Linux 系统的根目录下面有多个子目录。这些目录大部分是在安装 Linux 时创建的，这些目录都有专门的用途。

Red Hat Linux 9 根目录下的一级子目录及其作用如表 3-9 所示。

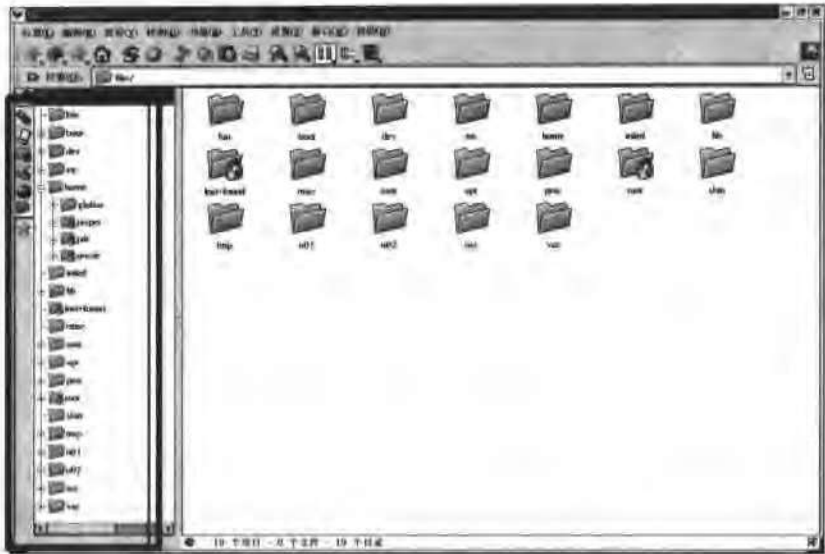


图 3-8 Linux 目录结构举例

表 3-9 Red Hat Linux 9 的一级子目录及其作用

一级子目录	存放的内容
bin	普通用户常用的程序例，如date命令
dev	存放系统中所有设备的设备名
home	存放普通用户的主目录（如globus用户的主目录为/home/globus）
lib	存放内核和应用程序所用的库文件和模块，包括各种硬件的驱动程序
root	根用户的主目录
usr	与用户有关程序，或者用户安装的其他应用程序
boot	存放系统内核和引导管理程序相关的文件
etc	存放与系统硬件和软件有关的所有配置文件
lost+found	根文件系统文件丢失和找到的孤立扇区
mnt	挂接其他分区的挂接点，可能包括光盘（cdrom）、软盘（floppy）和 Windows分区、共享的分区和USB盘分区的挂接点
proc	内存文件系统，对系统的诊断非常重要
sbin	根用户用于系统管理的可执行程序
tmp	系统中各个程序生成的临时文件，在下次启动时可以删除
var	系统运行过程中文件内容会发生变化的目录子树，主要包括mail目录，各种服务器的数据文件，日志文件和假脱机打印系统的打印队列数据

注意：普通用户创建目录和文件主要在他们的主目录下进行，这是因为用户在其他位置没有创建文件和目录的权限。

3.2.2 常用的目录操作

1. 顶层目录结构

用户可以使用ls/目录查看顶层的目录结构，命令如下：

```
[globus@unknown sbin]# ls /
bin  dev  home  lib      misc  opt  root  tftpboot  usr
boot  etc  initrd  lost+found  mnt  proc  sbin  tmp      var
```

显示的结果与表 3-8 的结果比较可知, bin、dev、home、lib、usr、etc、dev、sbin 和 var 都是系统目录。

## 2. 查看当前工作目录 (pwd)

如入 pwd 查看当前的工作目录:

```
[globus@unknown sbin]# pwd
/sbin
```

## 3. 改变目录

如果需要, 你可以改变当前目录。将当前目录改变到/mnt 目录, 输入如下命令:

```
[globus@unknown sbin]# cd /mnt
[globus@unknown mnt]#
```

从提示符可以看出 mnt 成为了当前目录。

也可以输入 pwd 确认当前目录为/mnt:

```
[globus@unknown mnt]# pwd
/mnt
```

输入 ls 命令, 显示/mnt 目录中的文件和目录的列表:

```
[globus@unknown mnt]# ls
cdrom  floppy  winc  wind
```

可以看到这里面有 4 个目录, 其中 cdrom 是 CD-ROM 盘的挂接点, floppy 是软盘的挂接点, 这两个关节点是系统默认生成的。winc 和 wind 是根用户创建的, 分别用于挂接 Windows 的 C 盘和 D 盘对应的分区。

## 4. 返回主目录 (cd)

返回主目录是在目录中移动的一个常用的操作。

输入 cd 命令, 返回用户的主目录。

```
[globus@unknown mnt]# cd
[globus@unknown globus]#
```

在此例中 globus 用户的主目录为/home/globus, 可以使用 pwd 命令验证:

```
[globus@unknown globus]# pwd
/home/globus
```

显示结果为/home/globus, 表明已经回到 globus 用户的主目录。

**注意:** 每一个用户的主目录是不相同的, cd 目录到底转到哪一个目录, 要根据创建该用户时设置的主目录决定。

如果当前用户是 root 用户, 那么它的主目录为/root。

## 5. 创建子目录 (mkdir)

可以使用 `mkdir` 在主目录中创建新的子目录。

在主目录里创建 `newdir1` 子目录：

```
[globus@unknown globus]$ mkdir newdir1
[globus@unknown globus]$
```

输入如下命令，以便确认目录已经创建：

```
[globus@unknown globus]$ ls -F
newdir1/
```

执行完 `mkdir` 命令后，没有任何提示，然后出现 shell 提示符，表明该命令顺利执行，否则会弹出一些错误提示表明无法创建了目录的原因。常见的错误原因如下：

- 目录已存在。错误提示如下：`mkdir: 无法创建目录 newdir: 文件已存在。`
- 你的权限不够。错误提示如下：`mkdir: 无法创建目录 newdir: 权限不够。`

注意默认情况下，除了根用户以外，用户只能在它的主目录（home 目录）和 `/tmp` 目录中创建了目录。

用户可以在一个 `mkdir` 命令中同时创建几个子目录，目录名之间以空格字符分开。

**注意：**从执行结果可以看出，结果只有一项名称为 `newdir1`，它的名字之后有一个 `/`，`ls` 的 `-F` 选项将在所有的目录之后加一个 `/`，在所有的可执行文件之后加一个 `*`，使用 `ls -F` 可以使目录和可执行文件更明显。

读者使用的用户名可能不是 `globus`，而且主目录里也可能有一些其他的文件。

因此 `ls -F` 命令执行后的结果可能不同，但是你仍然会看到 `newdir1` 这一项。

在列文件目录时，Linux 将大写字母开头的文件名列在前面。如果你希望使某一个文件更加容易查找，你可以将文件名的第一个字母设为大写。

用户在主目录中再创建子目录 `newdir2`，命令如下：

```
[globus@unknown globus]$ mkdir newdir2
[globus@unknown globus]$ ls -F
newdir1/ newdir2/
```

结果发现，新的子目录 `newdir2` 已经创建。

将当前目录转向 `newdir2`，输入下列命令：

```
[globus@unknown globus]$ cd newdir2
[globus@unknown newdir2]$
```

使用如下命令在当前目录创建一个名为 `file1.txt` 文件：

```
[globus@unknown newdir2]$ ls -F / > file1.txt
```

然后显示 `file1.txt`，命令如下：

```
[globus@unknown newdir2]$ cat file1.txt
bin/
boot/
dev/
etc/
home/
```

```
initrd/  
lib/  
lost+found/  
misc/  
mnt/  
opt/  
proc/  
root/  
sbin/  
tftpboot/  
tmp/  
usr/  
var/
```

输入 `cd` 命令，回到主目录。

`mkdir` 可以使用 `-p` 选项，使得当创建的子目录已经存在时，不会显示错误信息，同时如果你要创建的子目录的父目录不存在，它会自动创建它。

```
[globus@unknown newdir2]$ cd
```

#### 6. 删除目录 (`rmdir`)

`rmdir` 命令用于删除一个空目录。如果目录里还有文件，`rmdir` 命令将无法删除这个目录。`rmdir` 命令后跟的参数为要删除的目录名。

`rmdir` 的语法如下：

```
rmdir [OPTION]... DIRECTORY...
```

最简单和最常用的命令是在当前目录下删除它的下一级子目录，使用 `rmdir dirname`，其中 `dirname` 为要删除的目录名。你也可以同时删除多个目录，目录名之间用空格隔开。

如删除主目录下的 `newdir1` 子目录，使用下述命令：

```
[globus@unknown globus]$ rmdir newdir1
```

该命令执行完成后，没有任何提示，表明目录 `newdir1` 已被删除，不过你也可以使用 `ls` 命令来查看目录是否被删除。

```
[globus@unknown globus]$ ls -F  
newdir2/
```

执行结果表明，目录 `newdir1` 已被删除。

如果目录非空，试图删除这个目录时，将会出现如下的错误：

```
[globus@unknown globus]$ rmdir newdir2  
rmdir: 'newdir2': 目录非空
```

由于目录 `newdir2` 中存在一个名为 `file1.txt` 的文件，因此不能被删除，试图删除时出现“目录非空”错误。

例：使用 `mkdir` 命令一次创建 3 个子目录，目录分别为 `dir1`，`dir2` 和 `dir3`。然后查看创建的目录名，最后利用 `rmdir` 删除这 3 个文件。命令如下：

```
[globus@unknown globus]$ mkdir dir1 dir2 dir3
[globus@unknown globus]$
[globus@unknown globus]$ ls -F
dir1/ dir2/ dir3/ newdir2/
[globus@unknown globus]$ rmdir dir1 dir2 dir3
[globus@unknown globus]$
```

rmdir 命令的常见的选项如表 3-10 所示。

表 3-10 rmdir 命令的常见参数和选项

选 项	意 义
-p, --parents	试图删除目录路径上的每一个目录 例如：执行rmdir -p dir1/dir2/dir3 那么该命令首先试图删除子目录dir3， 然后试图删除dir2， 最后试图删除dir1 在这个过程中，只要遇到非空目录，命令就终止执行
-v, --verbose	输出处理的每一个目录
--help	显示有关帮助

例如，在主目录中创建 dir1/dir2/dir3 分层的目录结构，然后验证目录已经创建，最后删除你所创建的目录结构，命令如下：

```
[globus@unknown globus]$ mkdir dir1/dir2/dir3
mkdir: 无法创建目录 'dir1/dir2/dir3'：没有那个文件或目录
没有带-p 参数，无法创建/dir1/dir2/dir3，因为 dir1 目录不存在。
[globus@unknown globus]$ mkdir -p dir1/dir2/dir3
目录结构 dir1/dir2/dir3 创建成功。
[globus@unknown globus]$ ls dir1/dir2/
dir3
```

dir3 目录存在，并且在 dir1/dir2 目录之下。

```
[globus@unknown globus]$ rmdir -pv dir1/dir2/dir3
rmdir: 正在删除目录 dir1/dir2/dir3
rmdir: 正在删除目录 dir1/dir2
rmdir: 正在删除目录 dir1
```

使用-p 选项删除多级子目录，使用-v 选项显示 rmdir 的执行过程。

```
[globus@unknown globus]$ ls
newdir2
```

验证表明 dir1/dir2/dir3 多级目录确实被删除。

## 7. 目录更名 (mv)

mv 命令既可以更改文件的名称，也可以更改目录的名称，而且还可以将一个子目录及其该子目录中的文件和下级子目录中的所有文件从原来的位置，移到一个新位置，还可以同时更改目录的名称。

**mv** 命令的基本格式为：

```
mv sourcece dest
```

其中 **source** 为原目录的目录名，**dest** 为更名后的目录名。

将主目录中的 **newdir2** 目录名重命名为 **newdir3**,命令如下：

```
[globus@unknown globus]$ mv newdir2 newdir3
```

```
[globus@unknown globus]$
```

可以使用 **ls** 命令查看目录是否被修改，命令如下：

```
[globus@unknown globus]$ ls
```

```
newdir3
```

```
[globus@unknown globus]$ ls newdir3
```

```
file1.txt
```

结果显示，**newdir3** 的目录里有一个文件 **file1.txt**，这正是原来笔者前面在 **newdir2** 创建的文件。

查看原来的 **newdir2** 目录是否存在，命令如下：

```
[globus@unknow globus]$ ls newdir2
```

```
ls: newdir2: 没有那个文件或目录
```

原来的目录名已更名，所以原来的目录名 **newdir2** 已不存在。

文件目录更名可能会失败，失败的原因主要包括：

- 要更新的目录不存在。
- 目标目录名已经存在。
- 用户无权对文件更名。

**mv** 命令提供了一些有用的选项，其中常见的选项如表 3-11 所示。

表 3-11 mv 命令的常见选项

选项	意义
--backup	如果目标文件和目录存在，将对这些文件做一个备份
-b	与--backup功能相同，但是并不接受一个参数
-f, --force	在覆盖一个已存在的文件和目录之前，并不给用户提示
-i, --interactive	在覆盖一个已存在的文件和目录之前，将给用户一个提示
--reply={yes,no,query}	在覆盖一个已存在的文件和目录之前，应该如何提示用户 yes: 不提示，直接覆盖 no: 不提示，不覆盖 query: 将给用户一个提示，决定是否覆盖它
-u, --update	只有源目录或者文件比目标文件新，或者目标目录或者文件不存在时，才移动该文件
-v, --verbose	显示该命令的所有处理过程
--help	显示该命令的帮助



注意: `mv` 的 `-b` 选项与 `--replay=yes` 功能相同。

`mv` 的 `-i` 选项与 `--replay=query` 功能相同。

### 3.2.3 路径和当前目录

在定位一个文件或者目录时,可以使用绝对(absolute)或相对(relative)路径名。绝对路径从/(根目录)开始,然后逐渐到你所需的目录。相对路径从当前目录开始,当前目录可以在任何目录。

一个简化的 Linux 目录树如图 3-9 所示。在这棵目录树中,笔者列出了 Linux 根目录下的几个重要的子目录,这几个子目录分别是 `/bin`, `/sbin`, `/home` 和 `/etc`。当然还有其他的重要的子目录没有列出,这里只是为了说明问题。

在 `/home` 子目录中,又有两个子目录: `globus` 和 `henry`。它们分别是 `globus` 和 `henry` 的主目录。在 `globus` 的主目录中,笔者使用了 `mv` 命令将目录 `newdir2` 重命名为 `newdir3`。因此在 `globus` 的主目录里,可以看到 `newdir3` 子目录。

根据 3.2.2 节所讲述的内容,在 `newdir3` 中有一个文件 `file1.txt`。在 3.2.2 节中的目录操作都在当前目录中进行的。因此那时没有涉及到路径的细节。

事实上,用户可以处在任意的当前目录中,然后操纵任何位置合法的文件和目录。当然用户在不同的当前目录,对文件和目录的定位是不相同的。

下面介绍两个重要的概念。

- 绝对路径:在指定一个特定的文件时,与当前目录无关,它是从根目录开始的全路径,它能惟一确定一个文件。例如对于在 `newdir3` 目录下的文件 `file1.txt`,它的绝对路径表示应该为 `/home/globus/newdir3`。无论当前路径是什么,都可以使用 `/home/globus/newdir3` 惟一确定 `globus` 用户主目录上的文件 `file1.txt`。
- 相对路径:相对路径是相对于当前目录的。对于任何一个文件和目录,用户也可以使用相对路径表达它,但是对它的表达方式不惟一。例如如果用户处在 `globus` 的主目录,那么 `file1.txt` 的相对路径为: `newdir3/file1.txt`,如果用户的当前目录为 `/home`,那么 `file1.txt` 的相对路径表示为: `home/globus/newdir3/file1.txt`。

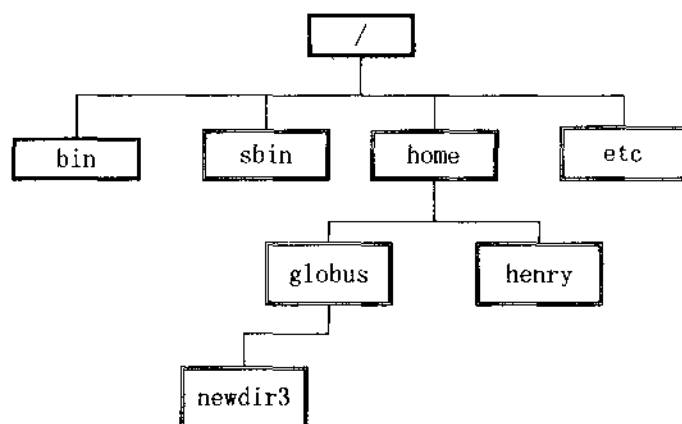


图 3-9 一棵简化的目录树

几个常用的特殊的目录名如表 3-12 所示。

表3-12 常用的几个特殊目录名

选项	意义
/	根目录。所有的文件和子目录都是根目录的子节点
.	当前目录。它指的是用户所处的当前目录，它的路径为pwd命令的执行结果。用它简化文件和目录操作
..	父目录。当前目录的上一级目录。由于Linux的目录结构是树状的，因此一个目录的父目录是为惟一确定的。要注意的是根目录没有父目录
~	当前用户的主目录。它使得无论你处于什么位置，都可以使用~快速定位到用户的主目录

笔者以图 3-9 的目录结构为例，说明当用户出于不同的目录后，如何表达文件 file1.txt 的相对路径何绝对路径，各种情况如表 3-13 所示。

表3-13 以file1.txt为例的相对路径和绝对路径

当前目录	相对路径	绝对路径
/home/globus	newdir3/file1.txt	/home/globus/newdir3/file1
/home	globus/newdir3/file1.txt	/home/globus/newdir3/file1
/	home/globus/Newdir3/file1.txt	/home/globus/newdir3/file1
/home/globus/newdir3	file1.txt	/home/globus/newdir3/file1
/etc	../home/globus/newdir3/file1.txt	/home/globus/newdir3/file1
/home/henry	../globus/newdir3/file1.txt	/home/globus/newdir3/file1
/home/henry	../../home/globus/newdir3/file1.txt	/home/globus/newdir3/file1

从表 3-13 可知，对于一个位置确定的文件或者目录，其绝对路径始终是不变的，除非这个文件被更名，或者被移动。而相对路径会根据当前目录的不同而不同。

**注意：**绝对路径的字符串都是从/字符开始的，而相对路径的字符串一定不是从/字符开始的。为了方便对某个文件和目录进行操作，用户可能会改变当前目录到你操作的文件所在的目录。

在 3.2.2 节中的有关文件名和目录名的参数中都可以使用路径，以便操作不在当前目录中的文件和目录。在以后所讲述的所有有关文件和目录的操作，用户都可以使用路径。

用户到底使用相对路径，还是使用绝对路径来表达文件和目录，主要的依据是哪一个方式更方便，也就是说哪一个表示法更易懂，或者是哪一种表达的路径字符串更短。

下面是一些关于相对路径和绝对路径的操作过程。

- 用户的当前目录在/home/globus，然后列出 newdir3 目录中的 file1.txt 文件：

```
[globus@unknown globus]$ pwd
/home/globus
[globus@unknown globus]$ ls newdir3/file1.txt
newdir3/file1.txt
```

- 将用户的当前目录转到/home，然后再使用相对路径列 file1.txt 文件：

```
[globus@unknown globus]$ cd ..  
[globus@unknown home]$ pwd  
/home  
[globus@unknown home]$ ls globus/newdir3/file1.txt  
globus/newdir3/file1.txt
```

- 将目录转到根目录，然后再使用相对路径列 file1.txt 文件：

```
[globus@unknown /]$ ls home/globus/newdir3/file1.txt  
home/globus/newdir3/file1.txt
```

- 使用绝对路径列 file1.txt 文件：

```
[globus@unknown /]$ ls /home/globus/newdir3/file1.txt  
/home/globus/newdir3/file1.txt
```

- 使用绝对路径将目录转到/etc，然后分别使用相对路径和绝对路径列 file1.txt 文件：

```
[globus@unknown /]$ cd /etc  
[globus@unknown etc]$ ls ../home/globus/newdir3/file1.txt  
../home/globus/newdir3/file1.txt  
[globus@unknown etc]$ ls /home/globus/newdir3/file1.txt  
/home/globus/newdir3/file1.txt
```

- 用相对路径将目录转到 globus 主目录下的 newdir3，然后列当前目录的所有文件：

```
[globus@unknown etc]$ cd ../home/globus/newdir3  
[globus@unknown newdir3]$ ls .  
file1.txt
```

- 在当前目录下，在用户的主目录里创建一个 newdir4 子目录：

```
[globus@unknown newdir3]$ mkdir ~/newdir4  
[globus@unknown newdir3]$ ls ..  
newdir3 newdir4
```

说明已经创建了 newdir4 目录。在此，你可以使用下述的集中方法创建这个目录：

- `mkdir ../newdir4`
- `mkdir /home/globus/newdir4`
- `mkdir ~/newdir4`

将目录转到当前用户的主目录，可以使用下述命令：

- `cd`
- `cd ..`
- `cd /home/globus`

**注意：**在 Linux 中，文件命名可以使用任何可打印字符，有一些字符在 Linux 系统中有特殊用途，因此不能直接使用，例如空格字符，字符\*、!、\$和\。空格字符在 Linux 中用作命令名、选项和参数之间的分隔符。“\”作为命令的的续行符等等。

在 Linux 中，可以间接使用这些字符，有两种方法：

- 在Linux专用的字符前加一个转意字符\。
- 将文件名或者目录名作为一个整体，在两边分别加上双引号“”。

例如，要求在用户的主目录分别创建3个文件abc de、\和～。创建过程如下：

```
[globus@unknown globus]$ mkdir abc\ de
[globus@unknown globus]$ mkdir "\ "
[globus@unknown globus]$ mkdir \~
```

可以使用ls命令来验证，是否成功建立了这些目录：

```
[globus@unknown globus]$ ls
~ \ abc de newdir3 newdir4
```

对于文件abc de，如果不使用abc\ de的形式，也不使用abc de的形式，系统会认为你将要同时创建abc和de两个子目录。

对于文件“\”，如果不使用“\\”，或者“mkdir \\ ”的形式，系统认为该命令还未结束，下一行继续该命令。

对于文件“～”，如果不使用“\～”或者““～””的形式，系统会给出“mkdir: 无法创建目录‘/home/globus’: 文件已存在”的错误信息。

使用另外一种形式，删除创建的3个目录：

```
[globus@unknown globus]$ rmdir "abc de"
[globus@unknown globus]$ rmdir "\\ "
[globus@unknown globus]$ rmdir "\~"
```

**注意：**不建议使用由Linux专用字符创建的文件，但是这对于Linux系统访问由Windows创建的文件十分有用，否则会出现系统无法读取Windows分区的某些文件。

### 3.3 Linux的文件操作

Linux系统使用文件管理磁盘文件和所有外部设备。在Linux系统中，一切操作对象都是文件，无论是普通文件，还是键盘、鼠标、打印机、磁盘、分区，即使是网络连接都是作为文件进行管理的。

#### 3.3.1 基本的文件管理

下面主要讲述如何列出了目录中文件信息（ls）、查看文件内容（cat, more, less）、创建文件（cat, >, <, |, >>, <<）、比较文件（diff）、删除文件（rm）、文件更名（mv）、连接文件（ln），等多项文件管理操作。

##### 1. ls命令

屏幕上会出现已创建的文件的文件名，如果文件不多，列表则会很简明。但是，如果文件很多，列表将会占满整个屏幕，浏览该列表也会变得十分困难。如果设计的目录结构合理，将相关的文件放在不同的目录下，那么每一个目录将会十分简洁。

如果不带任何参数，ls将列出当前目录下所有文件的文件名。

例如，列当前目录的文件列表，你可以使用如下的命令之一：

- `ls`
- `ls .`

可以同时列多个文件和目录，文件之间用空格隔开。

例如，同时列当前目录，和/bin 子目录中的文件，命令为 `ls ./bin`。命令运行结果如下：

```
[globus@unknown globus]$ ls . /bin
.:
newdir3 newdir4

/bin:
arch      cut          gawk      mail      rm         touch
... (省略)
[globus@unknown globus]$
```

`ls` 命令常见的选项如表 3-14 所示。

表3-14 `ls`命令的常见选项

选 项	含 义
<code>-a, --all</code>	除了显示正常文件以外，还显示以“.”打头的文件
<code>-A, --almost-all</code>	与-a参数类似，只是不显示“.”和“..”
<code>-d, --directory</code>	列目录的信息，而不是默认的列目录中的文件和子目录信息
<code>-F, --classify</code>	在每一个项目之后进入类型指示符，/表示目录，*表示是文件
<code>-h, --human-readable</code>	以符合阅读习惯的方式显示文件尺寸，单位为K，M和G
<code>-i, --inode</code>	打印每一个文件的i节点信息，i节点是一串数字
<code>-l</code>	使用长列表的形式显示文件信息
<code>-r, --reverse</code>	以相反的顺序列出文件，默认按照文件名从小到大排序
<code>-S</code>	以文件的大小排序，而不是默认的按照文件名的大小排序
<code>-R</code>	递归的列出所有子目录中的文件信息
<code>--help</code>	获得该命令的帮助信息

注意：“.”打头的文件常常用户的配置文件，通常保存在用户的主目录下，它们对用户使用系统常常起着重要的作用。`ls` 默认不显示它们，这样既可以避免因误操作而破坏这些文件，同时也能使显示目录列表简洁。

使用上述的选项进行如下练习：

列出主目录中的所有文件，包括以“.”开始文件。

```
[globus@unknown globus]$ ls -a
.  .bash_history .bash_profile .canna .gtkrc newdir3 .viminfo
.. .bash_logout .bashrc      .emacs .kde  newdir4 .xemacs
```

在此你可以看到很多以点(.)开头的文件，例如.kde 是当前用户存放 KDE 配置文件的目录。

列出主目录中的所有文件，包括以“.”开始文件，但不包括“.”和“..”。

```
[globus@unknown globus]$ ls -A
```

```
.bash_history .bash_profile .canna .gtkrc newdir3 .viminfo
```

```
.bash_logout .bashrc .emacs .kde newdir4 .xemacs
```

列出 newdir3 本身的信息，而不是 newdir3 目录下文件的信息。

```
[globus@unknown globus]$ ls -d newdir3
```

```
newdir3
```

以分类的形式显示当前目录所有文件的列表。

```
./ .bash_history .bash_profile .canna .gtkrc newdir3/ .viminfo
```

```
../ .bash_logout .bashrc .emacs .kde/ newdir4/ .xemacs/
```

从列表可以看出.kde 是子目录，而.bash\_profile 是文件。

列出当前目录中的文件列表，包括i节点信息。

```
[globus@unknown globus]$ ls -li
```

```
1975988 newdir3 2057062 newdir4
```

从列表中可以看出，文件 newdir3（目录也是文件）的 i 节点为 1975988，文件 newdir4 的 i 节点为 2057062。

对于每一个文件来说，目录中存放的仅仅是文件名和指向的 i 节点号。i 节点也称为信息节点。每一个信息节点仅存放文件的相关信息，这些信息包括文件的所有者、权限，文件的创建时间、文件的最后修改时间和文件的最后阅读时间，以及数据块在驱动器上的实际存放位置。对于目录而言，除了包含文件名、每一个文件和目录的索引节点以外，没有包含任何其他内容。文件和目录是通过信息节点相关联的，目录和文件之间的关系如图 3-10 所示。

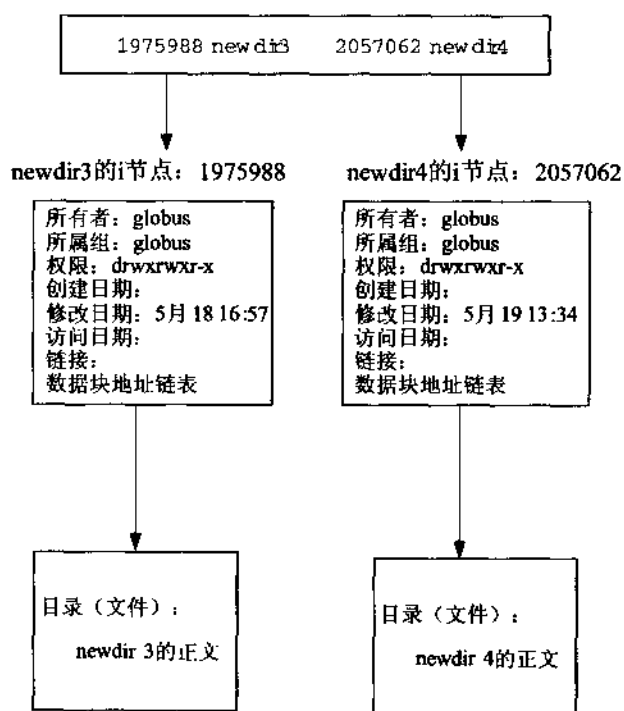


图 3-10 目录和文件之间的关系

注意：每一个用户的主目录就是一个文件，该文件包含了在这个目录下已创建文件的文件名和文件的信息节点数。信息节点包含文件的信息以及定位数据需要的地址。

以长列表的形式显示当前目录中的每一个文件：

```
[globus@unknown globus]$ ls -l
```

总用量 8

```
drwxrwxr-x  2 globus  globus      4096  5月 18 16:57 newdir3
```

```
drwxrwxr-x  2 globus  globus      4096  5月 19 13:34 newdir4
```

这个列表中的每一行是一个文件的信息，以目录 newdir3 为例说明每一个字段的含义，如图 3-15 所示。

表3-15 ls长列表中每一项的含义

字 段	含 义
drwxrwxr-x	文件的属性和权限信息
2	该文件的连接数
globus	该文件属于用户globus所有，即它的属主是globus
globus	该文件所属的组为globus组
4096	目录的大小，4096个字节，以字节为单位
5月 18 16:57	文件最后的修改时间，表明最后修改时间为5月18日，16点47分
newdir3	文件的名称，该目录的目录名为newdir3

递归的显示当前目录以及其子目录中的所有文件：

```
[globus@unknown globus]$ ls -R
```

```
./:
```

```
newdir3 newdir4
```

```
./newdir3:
```

```
file1.txt
```

```
./newdir4:
```

以符合人类习惯的形式显示当前目录的文件的列表：

```
[globus@unknown globus]$ ls -l -h
```

总用量 8.0K

```
drwxrwxr-x  2 globus  globus      4.0K  5月 18 16:57 newdir3
```

```
drwxrwxr-x  2 globus  globus      4.0K  5月 19 13:34 newdir4
```

## 2. 简单的文件创建工具

实用程序 cat 主要有三大功能：

- 一次显示整个文件。
- 从键盘创建一个文件。
- 将几个文件合并成一个文件。

使用 cat 显示文件已经在前面介绍过，在此不再重述。在此主要讲述如何使用 cat 创建

一个文件，以及如何使用 `cat` 将几个文件合并成一个文件。

`cat` 程序默认从键盘的数据，然后逐行的显示到显示器上，如果希望将输入的内容保存到文件，可以使用文件重定向。每输入一行，按回车键确认，输入下一行，当没有按回车键之前，可以使用退格键删除该行的输入错误，使用 `Ctrl+Z` 键结束输入，并且保存文件。

例如，从键盘输入内容，然后将这些信息以文件名 `hello1.txt` 保存到用户的主目录。

```
[globus@unknown globus]$ cat > hello1.txt
```

```
Hello!
```

```
I love Linux.
```

```
Ctrl+D (按Ctrl+D键)
```

显示刚才生成的文件，命令如下：

```
[globus@unknown globus]$ cat hello1.txt
```

```
Hello!
```

```
I love Linux.
```

说明文件已生成，而且确实与刚才的输入相一致。

注意：`cat` 只能产生新文件，不能编辑已存在的文件，因此它是一个很原始的文件生成程序。

可以使用同样的方法产生另一个名为 `hello2.txt` 的文件，文件内容如下：

```
This is file hello2.txt.
```

将这两个文件合并成一个新的文件称之为 `hello3.txt`，使用 `hello1.txt` 内容在新产生文件的前面，你可以使用下列命令：

```
[globus@unknown globus]$ cat hello1.txt hello2.txt > hello3.txt
```

```
[globus@unknown globus]$
```

该命令的执行过程如下，如果没有重定向符“>”存在，那么 `cat` 首先将文件 `hello1.txt` 显示到屏幕上，然后再将文件 `hello2.txt` 显示到屏幕上，最后 `cat` 终止执行。然而由于后面还有重定向符“>”，因此 `cat` 将对文件 `hello1.txt` 和 `hello2.txt` 的显示输入不显示到屏幕，而重定向到当前目录的一个名为 `hello3.txt` 的文件，因此 `hello3.txt` 不存在，所以它将新建这个文件，如果名为 `hello3.txt` 存在，那么 `cat` 程序将覆盖这个文件，得出相同的结果。

新文件已经产生，可以使用 `cat` 命令显示产生的新文件，命令如下：

```
[globus@unknown globus]$ cat hello3.txt
```

```
Hello!
```

```
I love Linux.
```

```
This is file hello2.txt.
```

```
[globus@unknown globus]$
```

关于 `cat` 的几种使用方式的总结如表 3-16 所示。

表3-16 cat常用的使用方法

使用方式	说 明
<code>cat file1 file2 ...</code>	一次显示一个或者多个文件到屏幕
<code>cat &gt; file</code>	将键盘输入保存到一个名为file的文件里
<code>Cat file1 file2 ... &gt; file</code>	将文件file1、file2等多个文件的内容合并成一个新文件file



### 3. 文件通配符

在进行文件和目录操作时，用户可以使用通配符在一个目录里选定文件名具有一个特征的文件和子目录。如果用户适当的使用通配符，将会减少输入量，极大地提高文件和目录操作的效率。

在 Linux 系统中，有两种通配符。

- \*。它通配文件名的 0 个或者是任意多个字符。
- ?。通配文件名中的一个字符。

下面是关于通配符\*和?的使用例子。

abc\*表示以 abc 开头的任何文件，包括文件 abc。如下文件名都是满足该条件：

abc、abcd、abcddd、abcabc、abc123、abc-234和、abcabc

以下是不满足上述筛选条件的文件名：

Abc、acabc、aabc、def和12345

**注意：**在 Linux 系统中文件名和目录名是区分大小写的，即 abc 和 Abc 指的是两个不同的文件，这与 DOS 不区分大小写是不同的。

在 DOS 中通配符\*要么对文件主名有效，要么对扩展名有效，如果要同时对文件名和扩展名都有效需要使用类似“\*.\*”的表示方式。在 Linux 的内核中文件名中的字符“.”没有特殊含义，仅仅作为一个普通字符，因此它没有文件主名和扩展名的概念，因此只用一个\*就可以表示任何文件名。

在 DOS 中通配符\*表示文件名后部的所有字符，例如 a\*和 a\*bcd 没有任何差别，都表示以 a 打头的所有文件。但是在 Linux 中，a\*和 a\*bcd 的意义是完全不同的，a\*表示以 a 打头的所有文件，而 a\*bcd 则表示文件名以 a 打头，并且以 bcd 结束的所有文件。

\*abc\*d 表示前面可以包括任意字符，紧接着是 abc 字符，并且在文件名结尾后至少包含一个字符 d 的所有文件名。下列文件满足筛选条件：

ZZZghjabcd、abcdefghijkl23-d、----abcd、abcd 和 \_\_abc\_\_c\_d

下列文件名不满足筛选条件：

Abcd、abcde、abbccd、d和DEFGabeeeeed

\*表示指定目录中的所有文件和子目录。

abc?表示以 abc 打头的后面只跟一个字符的文件名。下列文件满足筛选条件：

abcd、abc-、abcS、abc3 和abc\_

下列文件名不满足筛选条件：

Abc、abc、abcd、d和DEFGabeeeeed

使用 ls 作下面的练习。

列当前目录的所有文件：

```
[globus@unknown globus]$ ls -F
```

```
hello1.txt hello2.txt hello3.txt hello4.txt newdir3/ newdir4/
```

说明有 3 个文件两个目录。

查找所有包含字符 3 的文件名：

```
[globus@unknown globus]$ ls -d *3*
```

```
hello3.txt newdir3
```

表明当前目录中只有两个满足条件的文件和目录名。

列当前目录的所有文件和目录:

```
[globus@unknown globus]$ ls -d *
```

```
hello1.txt hello2.txt hello3.txt hello4.txt newdir3 newdir4
```

列所有以 **hello** 打头的文件:

```
[globus@unknown globus]$ ls hello*
```

```
hello1.txt hello2.txt hello3.txt hello4.txt
```

列/bin 子目录中文件中包含 **f** 字符的文件:

```
[globus@unknown globus]$ ls /bin/*f*
```

```
/bin/df /bin/false /bin/fgrep /bin/setfont
```

列/bin 子目录中以 **d** 打头, 后面包含一个字符的文件:

```
[globus@unknown globus]$ ls /bin/d?
```

```
/bin/dd /bin/df
```

#### 4. 输入/输出重定向

重定向从来源上可分为输入重定向和输出重定向, 输入重定向是指一个程序不接收默认输入, 而从一个文件中得到数据的输入, 输出重定向是指一个程序不输出到默认的显示器, 而将结果输出到一个文件或者作为另外一个程序的输入。从定向的目标上可分为文件重定向和程序重定向, 文件重定向是指将默认的键盘输入和默认的屏幕输出改变到文件, 程序重定向是指将一个程序的输出作为另外一个程序的输入。程序重定向也称作管道。

将 **ls** 的输出重定向到文件 **hello4.txt**, 命令如下:

```
[globus@unknown globus]$ ls -l > hello4.txt
```

```
[globus@unknown globus]$
```

**ls** 命令的输出没有输出到屏幕, 而重定向到文件 **hello4.txt**。

使用 **cat** 命令验证 **hello4.txt** 的正确性。

```
[globus@unknown globus]$ cat hello4.txt
```

总用量 20

```
-rw-rw-r-- 1 globus globus 21 5月 19 17:49 hello1.txt
-rw-rw-r-- 1 globus globus 25 5月 19 19:07 hello2.txt
-rw-rw-r-- 1 globus globus 46 5月 19 19:07 hello3.txt
-rw-rw-r-- 1 globus globus 0 5月 19 19:43 hello4.txt
drwxrwxr-x 2 globus globus 4096 5月 19 16:13 newdir3
drwxrwxr-x 2 globus globus 4096 5月 19 13:34 newdir4
```

将当前的日期时间追加到 **hello4.txt** 文件中, 命令如下:

```
[globus@unknown globus]$ date >> hello4.txt
```

```
[globus@unknown globus]$
```

显示文件 **hello4.txt** 的内容, 命令如下:

```
[globus@unknown globus]$ cat hello4.txt
总用量 20
-rw-rw-r-- 1 globus globus 21 5月 19 17:49 hello1.txt
-rw-rw-r-- 1 globus globus 25 5月 19 19:07 hello2.txt
-rw-rw-r-- 1 globus globus 46 5月 19 19:07 hello3.txt
-rw-rw-r-- 1 globus globus 0 5月 19 19:43 hello4.txt
drwxrwxr-x 2 globus globus 4096 5月 19 16:13 newdir3
drwxrwxr-x 2 globus globus 4096 5月 19 13:34 newdir4
  5月 19 19:47:27 CST 2004
```

可见当前的日期时间已被追加到文件 `hello4.txt` 中。

使用 “>” 时，如果文件已存在，它将覆盖重定向符后的文件，然后添加内容。而使用 “>>” 时，它在不覆盖原始的文件，而将新内容添加到原始的文件末尾，如果原始文件不存在，则创建新文件。

使用 `wc` 统计文件 `hello4.txt` 的行数，可以用如下的重定向命令完成：

```
[globus@unknown globus]$ wc -l < hello4.txt
8
```

得出文件 `hello4.txt` 的行数为 8，执行过程如下：默认情况下 `wc` 从键盘读入字符串然后进行文本行的统计，使用了 “<” 重定向符号之后，它不从键盘得到，而从文件 `hello4.txt` 中读取内容，而后进行统计。上述的命令与命令 `wc -l hello4.txt` 等价。

用户也可以直接将 `ls` 命令的输出，进行文字统计，命令如下：

```
[globus@unknown globus]$ ls | wc
6      6      60
```

该命令的执行过程如下：`ls` 首先执行 `ls` 命令，列当前目录的文件列表，由于使用了管道符 “|”，因此 `ls` 并没有将目录列表显示到屏幕，而将结果重定向作为 `wc` 程序的输入，然后 `wc` 对这些文件列表进行统计，最后将结果显示在屏幕上。结果是：目录列表总共有 6 行，共 6 个单词，字符总数为 60。

有关重定向的总结如表 3-17 所示。

表3-17 有关重定向的总结

重定向分类	重定向符号	功能说明
文件重定向	>, <, >>, <<	>将左边程序的输出重定向到右边的文件
		<将右边的文件作为左边程序的输入
		>>将左边程序的输出追加到右边的文件
		<<将右边的文件作为左边程序的一部分输入
程序重定向		将左边程序的输出作为右边程序的输入

注意：对于文件重定向，重定向符的左边是程序，重定向符的右边是文件。而对于程序重定向，重定向符的左右两边都是程序。

用户也可以多次使用程序重定向，举例如下：

```
[globus@unknown globus]$ ls | wc | wc
1      3      24
```

命令 `ls | wc | wc` 的执行过程如下：`ls` 首先得到当前目录的文件列表，不输出到屏幕，而把它作为第一个 `wc` 程序的输入，第一个 `wc` 接收 `ls` 的输出，然后进行统计，统计结束后，统计结果并不输出，又作为第二个 `wc` 程序的输入，最后对第一个 `wc` 的统计结果进行再统计，最后输出到屏幕上，结果为 1 行，3 个单词，共 24 个字符。

### 5. 逐屏阅览文件命令——`more`

`more` 是 Linux 的一个快速文件浏览工具，一个文件中可能包含大量数据，有时用户需要查看部分或者全部文件内容。

使用 `more` 命令显示长文件 `/etc/X11/XFree86Config`，命令如下：

```
more /etc/X11/XF86Config
```

它将在屏幕上显示文件 `/etc/X11/XF86Config` 的开始部分。

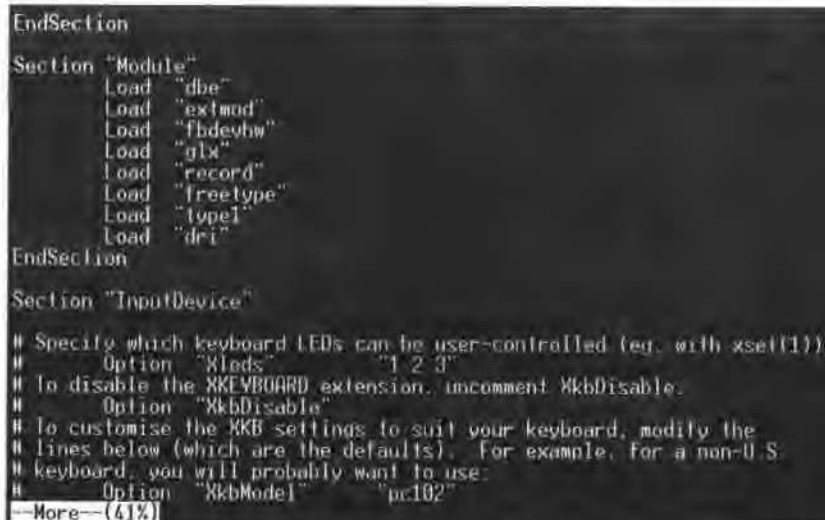
用户可以使用下列的键练习使文件内容向前滚动：

Space (空格)，向前滚动一屏文本  
 Space (空格)  
 Enter (回车)，向前滚动一行文本  
 Enter (回车)

如果用户输入查找字符串，则 `more` 将找到并显示包含目标字符串的第一页的第一个字符串。

作如下练习：

- 假定要在文件 `/etc/X11/XFree86Config` 中查找字符串 `Load`，则输入要查找的字符串 `/Load`，然后按回车键，它将会定位到包含 `Load` 字符串的第 1 页，如图 3-11 所示。



```
EndSection
Section "Module"
    Load "dbe"
    Load "extmod"
    Load "fbdevhw"
    Load "glx"
    Load "record"
    Load "freetype"
    Load "type1"
    Load "dri"
EndSection
Section "InputDevice"
    # Specify which keyboard LEDs can be user-controlled (eg. with xset(1))
    # Option "Xleds" "1 2 3"
    # To disable the XKEYBOARD extension, uncomment XkbDisable.
    # Option "XkbDisable"
    # To customise the XKB settings to suit your keyboard, modify the
    # lines below (which are the defaults). For example, for a non-U.S.
    # keyboard, you will probably want to use:
    # Option "XkbModel" "pc102"
More--(41%)
```

图 3-11 在 `more` 中定位到第一个要查找的字符串

输入 `n` 查找下一个匹配的字符串。

- 通过按 `n` 键，在文件中找下一个目标，并将相应的内容显示在屏幕上：

`n`

n  
n

- 向后回退一屏，输入：

b  
b

- 向前移动几页，输入：

空格键 (Space)  
空格键 (Space)  
空格键 (Space)

- 向下移动一行：

回车键 (Enter)  
回车键 (Enter)  
回车键 (Enter)

- 退出 more 程序，返回到 shell 提示符，输入：

q

more 命令行的各个选项如表 3-18 所示。

表 3-18 more 程序的常见选项

选 项	含 义
空格 (Space)	前进一屏
p	显示下一屏之前先清屏
c	作用同p基本一样
d	在每屏的底部显示更友好的提示信息，如 More (XX%) [Press space to continue, 'q' to quit. ] 而且若用户输入了一个错误命令，则显示出错信息，而不是简单地 响铃报警
l	不处理Ctrl+l（换页符）。如果没有给出这个选项，则more命 令在 显示了一个包含有Ctrl+l字符的行后将暂停显示，并等待接收命令
i	每次显示i行
s	文件中连续的空白行压缩成一个空白行显示

例：显示一个文件的内容，要求每 10 行显示一次，且显示之前先清屏。

```
[globus@unknown globus]$ more -c -10 /etc/X11/XF86Config
# XFree86 4 configuration created by redhat-config-xfree86
```

Section "ServerLayout"

```
Identifier      "Default Layout"
Screen          0  "Screen0" 0 0
InputDevice     "Mouse0" "CorePointer"
InputDevice     "Keyboard0" "CoreKeyboard"
InputDevice     "DevInputMice" "AlwaysCore"
```

EndSection

--More-- (9%)

实用程序 `more` 以每次一屏的方式显示文件内容，并等待用户输入，表 3-19 为 `more` 实用程序常见的操作命令。

表 3-19 `more` 程序的常见操作命令

命令	含义
空格 (Space)	前进一屏
回车 (Enter)	前进一行
b	回退一屏
/string	先前查找第一个含有string字符串的页，并显示这一页
n	在文件中继续向前查找string字符串，并将显示对应页的内容
=	显示当前行号
h	显示各命令的帮助信息
!cmd或:cmd	在子shell中执行cmd命令
q	退出more程序

#### 6. 更高级的文件查看程序——less

`less` 命令的功能几乎和 `more` 命令一样，也是用来按页显示文件，不同之处在于：`less` 命令在显示文件时允许用户既可以向前又可以向后翻阅文件。要移动到用文件的百分比表示的某位置，则直接使用键盘输入一个 0~100 之间的数，并按回车键即可。

`less` 的结合了 `more` 和 `vi`，因此它的操作命令比更丰富。`less` 的一些常用的操作命令如表 3-20 所示。

表 3-20 `less` 程序的常见操作命令

命令	含义
空格 (Space)	前进一屏
或者f	
回车 (Enter)	前进一行
b	回退一屏
/string	先前查找第一个含有string字符串的页，并显示这一页
?string	与/string功能相同，只是查找方向相反
n	在文件中继续向前查找string字符串，并将显示对应页的内容
=	显示当前行号
h	显示各命令的帮助信息
!cmd或:cmd	在子shell中执行cmd命令
j	向上滚动一行
k	向下滚动一行
u	在文件中向上移动半屏文字
d	在文件中向下移动半屏文字
q	退出more

例：需要按页显示/etc/ X11/XF86Config 文件。

```
[globus@unknown globus]$ less /etc/X11/XF86Config
```

## 7. 显示文件的前几行的命令—— head

如果用户希望查看一个文件究竟保存的是什么内容，可以只查看文件的头几行，而不必浏览整个文件。用 **head** 命令只显示文件或标准输入的前几行。

**head** 命令的语法如下：

```
head [-n ] 文件
```

功能：该命令显示每个指定文件的前面 **n** 行。如果没有给出 **n** 值，默认设置为 10。如果没有指定文件，**head** 就从标准输入读取。

例：显示文件/etc/ X11/XF86Config 的前 7 行。

```
[globus@unknown globus]$ head -7 /etc/X11/XF86Config
# XFree86 4 configuration created by redhat-config-xfree86
```

```
Section "ServerLayout"
```

```
Identifier      "Default Layout"
```

```
Screen 0 "Screen0" 0 0
```

```
InputDevice     "Mouse0" "CorePointer"
```

```
InputDevice     "Keyboard0" "CoreKeyboard"
```

```
[globus@unknown globus]$
```

## 8. 显示文件的最后几行的命令——tail

对查看日志文件，往往需要查看文件的最后几行，因为这几行刚好是记录刚发生的事件。与 **head** 命令相反，用户可以使用 **tail** 命令查看文件的尾部。

**tail** 命令的语法如下：

```
tail [+/-num] [参数] 文件
```

功能：该命令显示一个文件的指定内容。它把指定文件的指定显示范围内的内容显示在标准输出上。如果没有给定文件名，则使用标准输入文件。

**tail** 命令中各个选项的含义如表 3-21 所示。

表 3-21 tail 命令的选项

选项	含义
<b>+num</b>	从第num行以后开始显示
<b>-num</b>	从距文件尾num行处开始显示 如果省略num参数，系统默认值为10
<b>-nnum</b>	以文本行为num的计数单位。与参数选项+num或-num选项同时使用时，num表示要显示的文本行行数
<b>-cnum</b>	以字节为num的计数单位。与参数选项+num或-num选项同时使用时，num表示要显示的字符数

例：显示/etc/ X11/XF86Config 文件的最后 6 行文字。

```
[globus@unknown globus]$ tail -6 /etc/X11/XF86Config
```

```
Section "DRI"
```

```
    Group        0
    Mode         0666
```

```
EndSection
```

```
[globus@unknown globus]$
```

**注意：**上述例子中采用的命令与 `tail -n6 /etc/X11/XF86Config` 命令等价。

**例：**显示 `/etc/X11/XF86Config` 文件的最后第 10 个字符。

```
[globus@unknown globus]$ tail -c10 /etc/X11/XF86Config
dSection
```

```
[globus@unknown globus]$
```

### 9. 查看特殊文件命令——`od`

用户通常使用 `od` 命令查看特殊格式的文件内容。通过指定该命令的不同选项可以以十进制、八进制、十六进制和 ASCII 码来显示文件。

语法：`od [选项] 文件...`

`od` 命令中各选项及其含义如表 3-22 所示。

表 3-22 `od` 命令的常见选项

选项	取值	含义
指定偏移量	d	十进制
的基数	o	八进制（默认值）
-A	x	十六进制
	n	不打印位移值
指定数据	c	ASCII 字符或反斜杠序列
显示格式	d	有符号十进制数
-t	f	浮点数
	o	八进制（系统默认值为 02）
	u	无符号十进制数
	x	十六进制数
	n	n 表示每次以 n 个字节为单位显示数据

在表 3-22 中的“-t”选项中，除了取值 c 以外，其他选项后面都可以跟一个十进制数 n，指定每个显示值所包含的字节数。

**注意：**`od` 命令系统默认的显示方式是八进制，这也是该命令的名称由来（Octal Dump），但这不是最有用的显示方式，用 ASCII 码和十六进制组合的方式能提供更有价值的信息输出。

-A 与 `--address-radix=RADIX` 选项等价。



-t 与 -format=TYPE 选项等价。

-t 后的取值 c, d, f, o, u, x 和 n 可以组合使用, 这样可以起到对照的作用, 特别是将 n 与 c, d, f, o, u, x 组合使用, 将能很好地定制显示格式和效果。

下面是有关 od 命令的使用的例子。

例: 假定在当前目录下有一个文本文件, 名为 hello1.txt。

```
[globus@unknown globus]$ cat hello1.txt
```

Hello!

I love Linux.

如果没有上述文件, 可以使用 cat 命令建立。

例: 使用十进制的方式显示地址偏移量, 以文本方式显示文件内容。

```
[globus@unknown globus]$ od -Ad -tc hello1.txt
```

```
0000000 H e l l o ! \n I l o v e L i
```

```
0000016 n u x . \n
```

```
0000021
```

从显示结果可以看出, 显示共 3 行。每一行 16 个字符, 输出的每一行的开始部分表示相对于文件的偏移量, 例如第一行的 0000000 表示从文件的开头开始显示, 第二行的 0000016 表示从文件的开始的第 17 个字符开始显示, 最后一行显示了该文件的大小, 共 21 个字节。偏移量右边字符为当前行需要显示的 16 个字符。各字段的含义如图 3-12 所示。

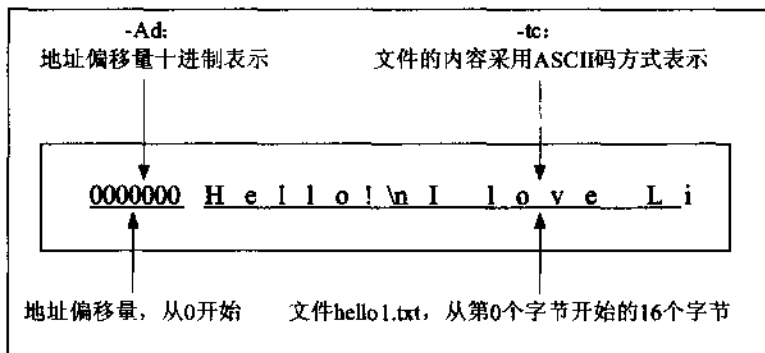


图 3-12 od 命令显示结果中各字段的含义

例: 使用八进制的方式显示地址偏移量, 以十六进制方式显示文件内容。

```
[globus@unknown globus]$ od -Ao -tx hello1.txt
```

```
0000000 6c6c6548 490a216f 766f6c20 694c2065
```

```
0000020 2e78756e 0000000a
```

```
0000025
```

显示结果中, 地址偏移采用八进制, 因此对于偏移 0000020 刚好是十进制的 16, 其他可依次类推。最后一行的 0000025 的十进制表示值为  $2 \times 8 + 5 = 21$ , 与前面的分析相同。

例: 使用十六进制显示偏移量和文件内容。

```
[globus@unknown globus]$ od -Ax -tx hello1.txt
```

```
000000 6c6c6548 490a216f 766f6c20 694c2065
```

```
000010 2e78756e 0000000a
000015
```

在上述的显示中，每4个字节分为一组，作为一个16进制数显示。

用户常常需要以字节为单位以十六进制显示文件的内容，这时可以将`-t`选项中使用多个值：

```
[globus@unknown globus]$ od -Ax -tx1 hello1.txt
000000 48 65 6c 6c 6f 21 0a 49 20 6c 6f 76 65 20 4c 69
000010 6e 75 78 2e 0a
000015
```

上述的结果刚好是每一个字符的ASCII码的十六进制表示，这对于程序设计者进行程序调试是非常有用的。

例：使用十六进制显示偏移量，用以单字节十六进制和以ASCII方式显示文件内容。

对于`hello1.txt`，命令和执行结果如下：

```
[globus@unknown globus]$ od -Ax -tcx1 hello1.txt
000000  H  e  l  l  o  !  \n  I           l  o  v  e           L  i
          48 65 6c 6c 6f 21 0a 49 20 6c 6f 76 65 20 4c 69
000010  n  u  x  .  \n
          6e 75 78 2e 0a
000015
```

10. 文件的复制命令——cp

`cp`命令的功能是将给出的文件或目录拷贝到另一文件或目录中，其功能非常强大。

语法：`cp [选项] 源文件或目录 目标文件或目录`

功能：该命令把指定的源文件复制到目标文件或把多个源文件复制到目标目录中。该命令的常见选项含义如表3-23所示。

表 3-23 cp 命令的常见选项

选项	含义
a	该选项通常在拷贝目录时使用。它保留链接、文件属性，并递归地拷贝目录，其作用等于 <code>dpr</code> 选项的组合
d	拷贝时保留链接
f	删除已经存在的目标文件而不提示
i	和 <code>f</code> 选项相反，在覆盖目标文件之前将给出提示要求用户确认。回答 <code>y</code> 时，目标文件将被覆盖，这是交互式拷贝
p	此时 <code>cp</code> 除复制源文件的内容外，还将把其修改时间和访问权限也复制到新文件中
r	若给出的源文件是一目录文件，此时 <code>cp</code> 将递归复制该目录下所有的子目录和文件。此时目标文件必须为一个目录名
l	不作拷贝，只是链接文件

例：将当前目录中的`hello1.txt`文件复制在当前目录复制一份，取名为`hello1cp.txt`：

```
[globus@unknown globus]$ cp hello1.txt hello1cp.txt
[globus@unknown globus]$
```

没有任何提示，表示文件复制成功。使用如下命令验证文件的内容与源文件的内容是相同的。

```
[globus@unknown globus]$ cat hello1cp.txt
Hello!
I love Linux.
```

例：首先在当前目录下建立一个子目录 `newdir5`，然后将 `newdir3` 及其所有的文件都拷贝到 `newdir5` 中。执行过程如下：

```
[globus@unknown globus]$ mkdir newdir5
[globus@unknown globus]$ cp -r newdir3 newdir5
[globus@unknown globus]$
```

上述命令使用 `-r` 选项，才能保证将 `newdir3` 目录下的文件和子目录递归地拷贝到 `newdir5` 中，否则会出现错误提示：“`cp: 略过目录 'newdir3'`”。

可以验证拷贝成功：

```
[globus@unknown globus]$ ls -l -R newdir5
newdir5:
总用量 4
drwxrwxr-x  2 globus  globus      4096  5月 20 13:46 newdir3

newdir5/newdir3:
总用量 4
-rw-rw-r--  1 globus  globus      111  5月 20 13:46 file1.txt
```

如果上述命令顺利执行，那么 `newdir5` 目录中将会出现 `newdir3` 目录子树。如果再次执行同一条 `cp` 命令时，会出“`cp: 略过目录 'newdir3'`”错误提示，表明目标文件已存在，如果希望覆盖所有已存在的目标文件，而不给用户提示，可以使用 `-f` 选项，命令如下：

```
[globus@unknown globus]$ cp -rf newdir3 newdir5
[globus@unknown globus]$
```

## 11. 文件移动命令——mv

在 3.2 节中，讲述了如何使用 `mv` 进行目录更名。因为目录也是文件的一种，因此 `mv` 可以使用于目录。

用户可以使用 `mv` 命令来为文件或目录改名，或将文件由一个目录移入另一个目录中。该命令如同 DOS 下的 `ren` 和 `move` 的组合。

语法：`mv [选项] 源文件或目录 目标文件或目录`

功能：如果 `mv` 命令中第二个参数类型是目标目录，那么 `mv` 命令将文件重命名并将文件移至一个新的目录中。当第二个参数类型是文件时，`mv` 命令完成文件重命名，此时，源文件只能有一个（也可以是源目录名），它将所给的源文件或目录重命名为给定的目标文件名。当第二个参数是已存在的目录名称时，源文件或目录参数可以有多个，`mv` 命令将各参数指

定的源文件均移至目标目录中。在跨文件系统移动文件时，**mv** 先拷贝源文件，再将原有文件删除，而链接至该文件的链接也将丢失。命令中各选项的含义如 3.2.2 节的表 3-11 所示。

注意：**mv** 与 **cp** 的执行结果不同。**mv** 好象文件“搬家”，文件个数并未增加，而 **cp** 对文件进行复制，文件个数增加了。

## 12. 文件删除命令——rm

在 **linux** 中创建文件很容易，系统中随时会有文件变得过时且毫无用处。用户可以用 **rm** 命令将其删除。该命令的功能为删除一个目录中的一个或多个文件或目录，它也可以将某个目录及其下的所有文件及子目录均删除。对于链接文件，只是删除了链接，原有文件均保持不变。

**rm** 命令的一般形式为：

**rm** [选项] 文件…

如果没有使用 **r** 选项，则 **rm** 不会删除目录。该命令的各选项含义如表 3-24 所示。

表 3-24 rm 命令的常见选项

选项	含义
<b>f</b>	忽略不存在的文件，并且在删除文件和目录前，从不给出提示
<b>r</b>	指示 <b>rm</b> 将参数中列出的全部目录和子目录均递归地删除
<b>i</b>	进行交互式删除

使用 **rm** 命令要格外小心。因为一旦一个文件被删除，它是不能被恢复的。例如，用户在输入 **cp**、**mv** 或其他命令时，不小心输入了 **rm** 命令，当用户按了回车键并认识到自己的错误时，已经为时已晚，文件已经被删除了。为了防止此种情况的发生，可以使用 **rm** 命令中的 **i** 项来确认要删除的每个文件。如果用户输入 **y** 开头的字符串，文件将被删除。如果输入任何其他字符串，文件将被保留。

例：将当前目录的 **hello1cp.txt** 文件更名为 **hello1cpmv.txt**。命令如下：

```
[globus@unknown globus]$ mv hello1cp.txt hello1cpmv.txt
```

验证文件是否被更名，命令如下：

```
[globus@unknown globus]$ ls
```

```
hello1.txt hello3.txt hello1cpmv.txt newdir4
```

```
hello2.txt hello4.txt newdir3 newdir5
```

例：在当前目录交互式删除文件 **hello1cpmv.txt**。命令如下：

```
[globus@unknown globus]$ rm -i hello1cpmv.txt
```

```
rm: 是否删除一般文件 'hello1cpmv.txt' ? y
```

使用 **i** 选项，在询问是否删除 **hello1cpmv.txt** 文件时，须择了 **y**，因此它删除了文件 **hello1cpmv.txt**。使用 **ls** 命令验证如下：

```
[globus@unknown globus]$ ls
```

```
hello1.txt hello2.txt hello3.txt hello4.txt newdir3 newdir4 newdir5
```

例：在当前目录子树递归的删除目录子树 **newdir5**，并且强制删除，不给用户提示。命令如下：

```
[globus@unknown globus]$ rm -rf newdir5
```

上述命令删除了 `newdir5` 及其中的所有文件和下级子目录。使用下述命令来验证其正确性:

```
[globus@unknown globus]$ ls
hello1.txt hello2.txt hello3.txt hello4.txt newdir3 newdir4
```

### 13. 文件链接命令——ln

`ln` 命令在文件之间创建链接。这种操作实际上是给系统中已有的某个文件指定另外一个可用于访问它的名称。对于这个新的文件名,读者可以为其指定不同的访问权限,以便控制对信息的共享和设置权限。如果链接指向目录,便能更加方便用户存取,用户可以利用该链接直接进入被链接的目录,而不需要使用一个很长的路径名。有时系统也需要特定位置的文件和目录名,建立一个链接既快捷,同时也不会影响原有系统的使用。即使用户删除这个链接,它也不会破坏原来的目录。

`ln` 命令的语法如下:

```
ln [选项] 目标 [链接名]
```

```
ln [选项] 目标 目录
```

链接有两种,一种被称为硬链接(Hard Link),另一种被称为符号链接(Symbolic Link)。建立硬链接时,链接文件和被链接文件必须位于同一个文件系统中,并且不能建立指向目录的硬链接。而对符号链接,则不存在这些问题。默认情况下,`ln` 产生硬链接。

在硬链接的情况下,参数中的“目标”被链接至[链接名]。如果[链接名]是一个目录名,系统将在该目录之下建立一个或多个与“目标”同名的链接文件,链接文件和被链接文件的内容完全相同。如果[链接名]为一个文件,系统将会提示该文件已存在,且不进行链接。如果指定了多个“目标”参数,那么最后一个参数必须为目录。

例:在当前目录中,建立 `hello1.txt` 的一个硬连接,连接名为 `hello1ln.txt`。命令如下:

```
[globus@unknown globus]$ ln hello1.txt hello1ln.txt
```

通过 `ls` 命令可知该链接已经建立,而且 `i` 节点的信息都完全一样。例如,修改时间都是 2004 年 5 月 19 日,17 时 49 分,而且它们的引用计数都是 2,而 `hello2.txt` 则是 1。

```
[globus@unknown globus]$ ls -l
```

总用量 28

```
-rw-rw-r-- 2 globus globus 21 5月 19 17:49 hello1ln.txt
-rw-rw-r-- 2 globus globus 21 5月 19 17:49 hello1.txt
-rw-rw-r-- 1 globus globus 25 5月 19 19:07 hello2.txt
-rw-rw-r-- 1 globus globus 46 5月 19 19:07 hello3.txt
-rw-rw-r-- 1 globus globus 441 5月 19 19:47 hello4.txt
drwxrwxr-x 2 globus globus 4096 5月 19 16:13 newdir3
drwxrwxr-x 2 globus globus 4096 5月 19 13:34 newdir4
```

可以进一步使用 `ls -i` 来验证它们的 `i` 节点是完全相同的。

```
[globus@unknown globus]$ ls -i
```

```
2383905 hello1ln.txt 2383906 hello2.txt 2383909 hello4.txt 2057062
newdir4
```

```
2383905 hello1.txt    2383908 hello3.txt    1975988 newdir3
```

可以看出,这两个文件(hello1.txt和hello1ln.txt)的i节点号都是2383905。如果修改了hello1ln.txt文件的内容,那么hello1.txt文件的内容也会随之改变,如果删除hello1.txt,那么hello1ln.txt的引用计数将变为1,这说明这两个文件名所指的文件是同一个,而且文件名hello1ln.txt和文件名hello1.txt的地位是完全一样的。

如果给ln命令加上-s选项,则建立的是符号链接。如果[链接名]已经存在但不是目录,将不建立符号链接。[链接名]可以是任何一个文件名(可包含路径),也可以是一个目录,并且允许它与“目标”不在同一个文件系统中。如果[链接名]是一个已经存在的目录,系统将在该目录下建立一个或多个与“目标”同名的文件,此新建的文件实际上是指向原“目标”的符号链接文件。

给hello1.txt再建立一个符号链接,存放在当前目录下,取名为hello1sln.txt,命令如下:

```
[globus@unknown globus]$ ln -s hello1.txt hello1sln.txt
```

使用ls验证连接已经创建,结果如下:

```
-rw-rw-r--    2 globus  globus      21  5月 19 17:49 hello1ln.txt
lrwxrwxrwx    1 globus  globus      10  5月 20 15:21 hello1sln.txt ->
                                     hello1.txt

-rw-rw-r--    2 globus  globus      21  5月 19 17:49 hello1.txt
-rw-rw-r--    1 globus  globus      25  5月 19 19:07 hello2.txt
-rw-rw-r--    1 globus  globus      46  5月 19 19:07 hello3.txt
-rw-rw-r--    1 globus  globus     441  5月 19 19:47 hello4.txt
drwxrwxr-x    2 globus  globus    4096  5月 19 16:13 newdir3
drwxrwxr-x    2 globus  globus    4096  5月 19 13:34 newdir4
```

其中在结果的第2行“lrwxrwxrwx 1 globus globus 10 5月20 15:21 hello1sln.txt -> hello1.txt”,可以看到hello1sln.txt指向文件hello1.txt。而且其i节的信息是不一样的,验证如下:

```
[globus@unknown globus]$ ls -li
2383905 hello1ln.txt    2383905 hello1.txt    2383908 hello3.txt    1975988
newdir3
2383910 hello1sln.txt  2383906 hello2.txt    2383909 hello4.txt    2057062
newdir4
```

文件hello1.txt和hello1sln.txt的i节点不同:分别是2383905和2383910。

可以通过使用文件名hello1sln.txt来操作文件hello1.txt。

例:显示hello1sln.txt所链接的文件hello1.txt的内容。命令如下:

```
[globus@unknown globus]$ cat hello1sln.txt
Hello!
I love Linux.
```

注意:如果删除符号链接文件所指向的文件,那么符号连接文件将会无意义,因此最好在删除符号链接之后,再删除源文件。

使用rm命令删除符号链接和硬链接。删除符号来连接和硬连接与删除普通文件一样,

都使用 `rm` 命令。

删除符号链接 `hello1sln.txt`，命令如下：

```
[globus@unknown globus]$ rm hello1sln.txt
```

删除硬链接 `hello1ln.txt`，命令如下：

```
[globus@unknown globus]$ rm hello1ln.txt
```

查看 `hello1.txt` 的引用计数，并且检查这些链接是否真的被删除：

```
[globus@unknown globus]$ ls -l
```

总用量 24

```
-rw-rw-r--  1 globus  globus      21  5月 19 17:49 hello1.txt
-rw-rw-r--  1 globus  globus      25  5月 19 19:07 hello2.txt
-rw-rw-r--  1 globus  globus      46  5月 19 19:07 hello3.txt
-rw-rw-r--  1 globus  globus     441  5月 19 19:47 hello4.txt
drwxrwxr-x  2 globus  globus    4096  5月 19 16:13 newdir3
drwxrwxr-x  2 globus  globus    4096  5月 19 13:34 newdir4
```

从结果可知，`hello1.txt` 的引用计数变为 1，而且这些符号链接都已被删除。

#### 14. 创建一个空文件命令 —— `touch`

用户可以使用 `touch` 命令对文件和目录进行管理。`touch` 命令改变文件的时间记录。可以使用 `ls -l` 显示文件的时间记录。

该命令的语法如下：

```
touch [-acfm] [-r 参考文件] [--file=参考文件]
      [-t MMDDhhmm[[CC]YY][.ss]]
      [-d 时间] [--date=时间] [--time={atime,access,use,mtime,modify}]
      [--no-create] [--help] [--version]
```

文件列表

常见的参数如下：

- `a` 改变文件的读取时间记录。
- `m` 改变文件的修改时间记录。
- `c` 假如目标文件不存在，不会建立新的文件。与 `--no-create` 效果相同。
- `f` 仅仅为了与其他 `unix` 系统兼容而保留。
- `r` 使用参考文件的时间记录，与 `--file` 的效果一样。
- `d` 设定时间与日期，可以使用各种不同的格式。
- `t` 设定文件的时间记录，格式与 `date` 指令相同。
- `--no-create` 不会建立新文件。
- `--help` 显示命令帮助。
- `--version` 显示版本信息。

例：最简单的使用方式是将文件的时间记录改为现在的时间。若文件不存在，系统会建立一个新的文件。

```
[globus@wang globus]$ touch test
```

将 test 文件的时间改为 2003 年 5 月 22 日 12 点 22 分。时间的格式可以参考 date 指令，至少需输入 MMDDHHmm（就是月、日、时和分）。

```
[globus@wang globus]$ touch -c -t 200305221222 test
```

检查设置结果：

```
[globus@wang globus]$ ls -l test
```

```
--w--w--w- 1 globus globus 0 2003-05-22 test
```

如果 hello1.txt 存在，则删除该文件：

```
[globus@wang globus]$ rm hello1.txt
```

使用 touch 命令创建一个空的 hello1.txt 文件：

```
[globus@wang globus]$ touch hello1.txt
```

查看文件的修改时间：

```
[globus@wang globus]$ ls -l hello1.txt
```

```
--w--w--w- 1 globus globus 0 5月 23 02:05 hello1.txt
```

修改时间为 2004 年 5 月 23 日，2 时 5 分，而且长度为 0 字节，说明它是一个空文档。

将文件 test 的时间记录设置成与文件 hello1.txt 一样：

```
[globus@wang globus]$ touch -r hello1.txt test
```

检查日期是否相同：

```
[globus@wang globus]$ ls -l hello1.txt test
```

```
--w--w--w- 1 globus globus 0 5月 23 02:05 hello1.txt
```

```
--w--w--w- 1 globus globus 0 5月 23 02:05 test
```

可以看出，现在文件 hello1.txt 和 test 日期和时间是相同的。

### 3.3.2 文件和目录权限操作

Linux 系统将信息都存放在文件中，系统里运行的程序也是以文件形式存储。系统程序的工作模式通常通过与之对应的控制文件来设定。用户能否访问一个文件，以什么方式来访文件是通过文件的权限属性决定的。甚至打印机、终端等设备也是当成文件管理，并和普通文件一样，被赋予一定的权限。

通过修改文件的访问权限，文件的所有者能够决定哪些用户能读取、修改和（或）运行文件。对于系统和用户来说，仔细设置每一个文件和目录的访问权限是 Linux 系统安全的基础。文件能否被访问，以及访问方式是由文件的权限决定的。

#### 1. 文件权限简介

用户为了不同的目的而使用 Linux 系统，但最后归根结底是以文件的形式存取信息。例如用户可能执行命令，输入数据，打印文件和编写程序。所有的活动都是通过以下 3 种方式来访问文件而达到目的的：

- 读取文件（read）。使用文件查看程序（如 cat、more 和 less 等）读取文件的内容。这种方式仅仅读取，而没有对文件进行修改。使用 ls 命令列目录时，实际上 ls 在读取目录文件。
- 修改文件（write）。当使用 cat 从键盘读取输入后，重定向（>）到一个文件时，



则创建了一个文件，如果使用重定向 (>>) 到一个文件时，它会在原有文件的基础上追加内容。如果创建一个新文件，或者删除一个已存在的文件时，其实它也同时修改了目录文件。所有的这些操作都可以归结到对文件的写操作。

- 执行文件 (execute)。执行一个程序 (如 ls、cat 和 ls) 时，系统将创建一个新的进程，来完成该程序的执行过程。这种操作称为执行文件 (执行程序)。

如果拥有一个文件，那么该用户可以启用或者禁用 3 种权限中的一种：读、写或者执行。所有者可以对一个文件给三类用户分别设置权限，这三类用户分别是：文件所有者 (user)、同户所属的组 (group) 和其他所有用户 (other)。other 类的用户是哪些不属于用户所属组的用户。

所谓的文件权限，是指对文件的访问权限，包括对文件的读、写 (包括删除) 和执行 (搜索)。Linux 是一个多用户操作系统，它允许多个用户同时登录和工作。因此 Linux 将一个文件或目录与一个用户和组联系起来。请看下面的例子：

```
drwxr-xr-x 5 root root 1024 Sep 13 03:27 Desktop
```

与文件权限相关联的是第一、第三和第四个域。第三个域是文件的所有者，第四个域是文件的所属组，而第一个域限制了文件的访问权限。在这个例子中，文件的所有者是 root，所属的组是 root，文件的访问权限是 drwxr-xr-x。对于文件和目录来说，每个文件和目录都有一组权限标志和它们结合在一起，在上例中就是第一个域中的内容。下面来仔细分析这个域中各个符号的意义。

该域由 10 个字符组成，可以把它们分为 4 组，具体含义如表 3-25 所示。

表 3-25 文件访问权限字段的分组

权限字段	分组	drwxr-xr-x
文件类型	第一个字符	d (目录)
属主权限标志	2~4 个字符	rwX
属组权限标志	5~7 个字符	r-x
其他用户权限标志	8~10 个字符	r-x

第一个字符为文件类型。由于 Linux 系统对与设备、目录、文件都当作是文件来处理，因此该字符表明此文件的类型，其字符与对应的意义如表 3-26 所示。

表 3-26 文件权限字段的第 1 个标志字符及文件类型的对应关系

文件标志	文件类型	例子
-	普通文件	数据文件、ASCII 纯文本文件或者程序
d	目录	/bin、/mnt 和 /home
b	块设备	/dev/hda (第一个 IDE 硬盘)
c	字符设备	/dev/ttyS0 (与 DOS 的串口 1 等同)
s	套接字	/dev/log
p	命名管道	/dev/initctl (与 “ ” 等同)
l	符号链接	/etc/grub.conf -> ../boot/grub/grub.conf

权限字段的后 9 位是权限标志。

对每个文件或目录都有 4 类不同的用户。每类用户各有一组读、写和执行 (搜索) 文

件的访问权限，这4类用户是：

- root。系统特权用户类，即 UID=0 的用户。
- owner。拥有文件的用户，文件的属主。
- group。共享文件的用户组名称。
- other。不属于上面3类的所有其他用户。

作为 root，它自动拥有了对所有文件和目录的全面的读、写和搜索权限，所以没有必要明确指定它的权限。其他三类用户则可以在单个文件或者目录的基础上分别授予或撤销权限。因此对另外三类用户，一共9个权限位与之对应，分为3组，每组3个，分别用 r、w、x 来表示，分别对应 owner、group 和 other。

注意：在 Linux 下，每个文件属于某一个用户，又同时属于某一个用户组。

权限位对于文件和目录的含义有些不同。每组3个字符对应的含义从左至右的顺序，对于文件来说是：读文件的内容（r）、写数据到文件（w）、作为命令执行该文件（x）。对于目录来说是：读包含在目录中的文件名称（r）、写信息到目录中去（增加和删除索引点的连接）、搜索目录（能用该目录名称作为路径名去访问它所包含的文件或子目录）。具体来说就是：

（1）有只读权限的用户不能用 cd 进入该目录；还必须有执行权限才能进入。

（2）有执行权限的用户只有在知道文件名，并拥有该文件的读权限的情况下才可以访问目录下的文件。

（3）必须有读和执行权限才可以使用 ls 列出目录清单，或使用 cd 命令进入目录。

（4）如用户有目录的写权限，则可以创建、删除或修改目录下的任何文件或子目录，即使该文件或子目录属于其他用户也是如此。

## 2. 改变文件的属主和所属组

由以上的分析可知，对于一个文件都有一个属主（owner）和一个所属组（group）。你可以通过 chown 命令来改变一个文件的属主 owner 和 group。

chown 命令的形式如下：

```
chown [选项] [所有者]:[所属组] 文件列表
```

```
chown [选项] :[所属组] 文件列表
```

例：首先查看当前目录下的文件 hello1.txt 的属主和所属组。命令如下：

```
[globus@unknown globus]$ ls -l hello1.txt
```

```
-rw-rw-r-- 1 globus globus 21 5月 19 17:49 hello1.txt
```

从列表中可知，hello1.txt 属于 globus 用户，属于 globus 组。

注意：同名的用户和组的意义是不同的。

例：以 root 用户身份，将 hello1.txt 文件的属主改成 root。执行过程如下：

首先使用 su 命令切换成 root 用户，需要在输入密码的提示 Password:后，输入 root 用户的口令：

```
[globus@unknown globus]$ su
```

```
Password:
```

```
[root@unknown globus]#
```

然后将 `hello1.txt` 属主改为 `root`，命令如下：

```
[root@unknown globus]# chown root hello1.txt
```

没有出现提示，表明改变属主已成功，使用 `ls` 命令验证如下：

```
[root@unknown globus]# ls -l hello1.txt
```

```
-rw-rw-r-- 1 root globus 21 May 19 17:49 hello1.txt
```

从列表可知 `hello1.txt` 文件的属主已经变成了 `globus`，然后所属组仍然是 `globus`。

使用 `chown` 将 `hello1.txt` 文件的所属组该为 `root` 组，命令如下：

```
[root@unknown globus]# chown .root hello1.txt
```

使用 `ls` 验证 `hello1.txt` 的所属组已经更改为 `globus`，验证如下：

```
[root@unknown globus]# ls -l hello1.txt
```

```
-rw-rw-r-- 1 root root 21 May 19 17:49 hello1.txt
```

`chown` 命令后的 `.root` 参数表式更改 `hello1.txt` 的所属组。如果将 `.root` 改为 `root`，表明改变文件的属主。因此，它们的区别在于：如果 `chown` 的第一个参数前面没有 `.`，表明更改的是文件的属主，否则更改的是文件的所属组。

注意：“`chown .root 文件名`”与“`chown :root 文件名`”两条命令是等价的。

如果同时设置文件的属主和所属组，`chown` 的第一个参数使用“新属主名.新所属组名”或者“新属主名:新所属组名”的形式。

例：使用一条命令将文件的属主更改为 `globus` 用户，将其所属组更改为 `globus` 组。命令如下：

```
[root@unknown globus]# chown globus.globus hello1.txt
```

使用 `ls` 命令验证如下：

```
[root@unknown globus]# ls -l hello1.txt
```

```
-rw-rw-r-- 1 globus globus 21 May 19 17:49 hello1.txt
```

列表表明，文件 `hello1.txt` 的属主和所属组均更改为 `globus`。

你也可以使用 `chgrp` 命令，来改变文件的所属组。

`Chgrp` 命令的语法为：

`chgrp 组名 文件名列表`

例：将 `hello1.txt` 的所属组再改变成 `root`。命令如下：

```
[root@unknown globus]# chgrp root hello1.txt
```

使用 `ls` 命令验证如下：

```
[root@unknown globus]# ls -l hello1.txt
```

```
-rw-rw-r-- 1 globus root 21 May 19 17:49 hello1.txt
```

结果表明，文件 `hello1.txt` 的所属组已经更改成 `root` 组。

注意：在命令 `chown` 中，除了使用用户名和组名之外，你还可以使用用户名和组名对应的用户 ID 和组 ID。

如下 3 条命令是等价的：

`chown :group 文件列表`

`chown .group 文件列表`

`chgrp group 文件列表`

例：root 用户的组 ID 为 0，root 组的组 ID 也为 0，那么如下两条命令是等价的。

```
[root@unknown globus]# chown root.root hello1.txt
```

```
[root@unknown globus]# chown 0.0 hello1.txt
```

同理，下述两条命令也是等价的：

```
[root@unknown globus]# chgrp root hello1.txt
```

```
[root@unknown globus]# chgrp 0 hello1.txt
```

chown 和 chgrp 的选项如表 3-27 所示。

表 3-27 chown 和 chgrp 的常用选项

选 项	含 义
-R, --recursive	递归的处理文件和目录子树里的所有文件和目录
-v, --verbose	显示对每一个文件的处理过程
--help	显示本命令的帮助

例：递归的将当前目录下的 newdir3 子目录下的文件以及下级子目录递归将属主更改为 root，并且显示对每一个文件和目录的处理过程。命令如下：

```
[root@unknown globus]# chown -v -R root newdir3
```

```
changed ownership of 'newdir3' to root
```

```
changed ownership of 'newdir3/file1.txt' to root
```

由此可见，子目录 newdir3 和 newdir3 子目录下的文件 file1.txt 的文件属主都更改为 root。如果 newdir3 中有下级子目录，那么这些子目录和目录中的文件的属主都将更改为 root。

**注意：**一个命令的选项可以分开，也可以合并在一起，例如下面两条命令是等价的：

```
chown -v -R root newdir3
```

```
chown -vR root newdir3
```

还原当前目录里的文件 hello1.txt 和 newdir3 及其子目录里的文件的属主和所属组，命令如下：

```
[root@unknown globus]# chown -vR globus.globus hello1.txt newdir3
```

```
changed ownership of 'hello1.txt' to globus:globus
```

```
changed ownership of 'newdir3' to globus:globus
```

```
changed ownership of 'newdir3/file1.txt' to globus:globus
```

将当前用户还原成 globus，命令如下：

```
[root@unknown globus]# exit
```

```
exit
```

```
[globus@unknown globus]$
```

### 3. 设置文件和目录的访问权限

确定了一个文件的访问权限后，用户可以利用 Linux 系统提供的 chmod 命令来重新设定不同的访问权限。

chmod 命令是非常重要的，用于改变文件或目录的访问权限。用户用它控制文件或目

录的访问权限。

该命令有两种用法。一种是包含字母和操作符表达式的文字设定法；另一种是包含数字的数字设定法。

chmod 命令的文字设定法：

chmod [who] [+ | - | =] [mode] 文件名列表

命令中各选项的含义为：

操作对象 who 可以是如表 3-28 所示的任一个或者它们的组合。

表 3-28 chmod 的操作对象 who 的可选项

who	含 义
u	表示“属主（user）”，即文件或目录的所有者
g	表示“所属组（group）”里的用户，即组名或者组ID与文件中的组名或者组ID相同的所有用户
o	表示既不属于属主（user），也不属于group的“其他（others）”用户
a	表示“所有（all）用户”。它是系统默认值

可用的操作符及其含义如表 3-29 所示。

表 3-29 chmod 命令可用的操作符

操作符	含义
+	添加某个权限
-	取消某个权限
=	赋予给定权限，并取消文件和目录原有的权限设置

设置 mode 所表示的权限可用下如表 3-30 所列字母的任意组合。

表 3-30 chmod 命令可用的操作符

mode	含义
r	可读
w	可写
x	可执行
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	在文件执行时把进程的属主或组ID置为该文件的属主。方式“u+s”设置文件的用户ID位，“g+s”设置组ID位
t	保存程序的文本到交换设备上
u	与文件属主拥有一样的权限
g	与文件属主同组的用户拥有一样的权限
o	与其他用户拥有一样的权限

文件名列表是要改变权限的以空格分开的文件列表，支持通配符。

在一个命令行中可给出多个权限方式，其间用逗号隔开。

例：chmod g+r, o+r hello1.txt

使所属组里的用户和其他用户对文件 hello1.txt 有读权限。

例: `chmod +r hello1.txt`

使所有用户对文件 `hello1.txt` 有读权限。

例: `chmod -r hello1.txt`

```
[globus@unknown globus]$ chmod -r hello1.txt
```

```
[globus@unknown globus]$ ls -l hello1.txt
```

```
--w--w---- 1 globus globus      21  5月 19 17:49 hello1.txt
```

使除 `root` 用户以外的所有用户对文件 `hello1.txt` 都没有读权限。示例如下:

```
[globus@unknown globus]$ su
```

```
Password:
```

```
[root@unknown globus]# cat hello1.txt
```

```
Hello!
```

```
I love Linux.
```

```
[root@unknown globus]# exit
```

```
exit
```

```
[globus@unknown globus]$ cat hello1.txt
```

```
cat: hello1.txt: 权限不够
```

尽管 `hello1.txt` 对任何用户都没有读权限, 但是 `root` 用户默认对任何文件都有读权限。当文件的属主 `globus` 来读 `hello1.txt` 文件时, 由于 `globus` 没有读权限, 因此给用户提示“权限不够”的错误。

例: 设置 `hello1.txt` 的权限字段, 使属主可以读写文件, 所属组可以读文件, 其他用户没有任何权限。命令如下:

```
[globus@unknown globus]$ chmod u=rw,g=r,o= hello1.txt
```

可以使用 `ls` 验证, 命令如下:

```
[globus@unknown globus]$ ls -l hello1.txt
```

```
-rw-r----- 1 globus globus      21  5月 19 17:49 hello1.txt
```

从结果看出, 设置已完成, `globus` 用户可以读和修改该文件, 属于 `globus` 组的用户可以读该文件, 其他用户既不能读, 也不能修改该文件。该文件也没有执行权限。

**chmod 命令的数字设定法:**

读者必须首先需要了解用数字表示属性的含义: 0 表示没有权限, 1 表示可执行权限, 2 表示可写权限, 4 表示可读权限。然后将其相加, 因此数字属性的格式应为 3 个 0~7 的八进制数, 其顺序是 (u) (g) (o)。

所有可能的的基本权限组合如表 3-31 所示。

表 3-31 用数字表示的基本权限

数字	代表的权限
4	可读
2	可写
1	可执行

例如, 如果想让某个文件的属主有“读/写”二种权限, 需要对应的权限的数字表示法

为：4（读）+2（写）=6（读/写）。

数字设定法的一般形式为：

`chmod [mode] 文件名列表`

例：执行下面的命令。

`chmod 754 hello1.txt`

执行过程如下：

`[globus@unknown globus]$ chmod 754 hello1.txt`

`[globus@unknown globus]$ ls -l hello1.txt`

`-rwxr-xr-- 1 globus globus 21 5月 19 17:49 hello1.txt`

查验设置的权限字段的结果为：

`-rwxr-xr--`

上述执行的 `chmod` 命令中：

- 7 将所有者的权限设置为 `rwx`，使属主具有读、写和执行权限（ $7=4+2+1$ ）。
- 5 将所属组用户的权限设置为 `r-x`，使所属组里的用户具有读和执行的权限（ $5=4+1$ ）。
- 4 将其他用户的权限设置为 `r--`，使其他的用户只具有读权限（4）。

在使用 `chmod` 命令时，可以使用如表 3-32 所示的常用选项。

表 3-32 `chmod` 的常用选项

选项	含义
<code>-R, --recursive</code>	递归的处理文件和目录子树里的所有文件和目录
<code>-v, --verbose</code>	显示对每一个文件的处理过程
<code>--help</code>	显示本命令的帮助

#### 4. 设定默认权限命令 —— `umask`

在本章前面的例子里，当用户使用重定向（>）建立一个文件（如 `hello1.txt`）时，操作系统默认给该文件的文件属主和所属组用户赋予了读和写权限，给其他用户赋予了读权限。

创建文件时的默认权限设置是由 `umask` 的值决定的。`umask` 的值决定在创建文件时默认屏蔽（mask）那些权限。改变 `umask` 值不影响现有的文件。`umask` 的初始值是由操作系统决定的，但是用户可以根据需要在 `shell` 命令行中，或者用户的启动文件中进行修改。

要确定在当前环境的 `umask` 的设置值，可以使用 Linux 提供的 `umask` 命令。在结果中如果以数字方式显示，每一个数字以 8 进制表示，从左向右的第一个 0 表示 8 进制的开始，第二位、第三位和第四位分别表示文件属主、所属组用户和其他用户创建文件时默认的访问权限的屏蔽值。

例：以数字形式确定当前的 `umask` 的设置值。命令如下：

`[globus@wang globus]$ umask`

`0002`

该命令的结果中以 8 进制显示了 `0002`，此例第二位和第三位均为 0（二进制的 `000B`，

表示没有屏蔽任何一种权限),表示创建文件时文件主和所属组默认具有读、写和执行权限,第四位为 2 (二进制的 010B,表示屏蔽了写权限),表示其他用户默认具有读和执行权限。

例:在同一个 shell 窗口中,以符号形式显示当前的 umask 的设置值。命令如下:

```
[globus@wang globus]$ umask -S
```

```
u=rwx,g=rwx,o=rx
```

此结果与使用不带选项-S 的 umask 命令的结果相同,只是表述方式不同。

umask 命令用于显示和设置 umask 的值。

umask 命令的语法如下:

```
umask [-p] [-S] [屏蔽值]
```

如果 umask 不带参数“屏蔽值”,则 umask 显示当前 shell 的 umask 值设置。否则设置当前 shell 的 umask 值。

可用的选项:

- -S 用符号的表达屏蔽值。
- -p 如果在省略“屏蔽值”参数时使用,则 umask 命令显示的结果可以被 umask 重新用于设置 umask 值。

注意:umask 值中每一位八进制数与文件中权限字段的八进制值表示的意义相反,也就是说,如果 umask 的文件主屏蔽位为 5 (八进制),则文件创建的默认权限为 2 (7-5),即文件主将具有写权限 (2),而屏蔽了读和执行的权限 (5=7-2)。

对 mask 值为 233 和 022 时的默认权限分析如表 3-33 所示。

表 3-33 umask 的默认权限举例

umask	所有者 (rwx)	所属组 (rwx)	其他用户 (rwx)	权限
233	101	100	100	屏蔽了所有者的写权限,屏蔽了所属组和其他用户的写和执行权限
	5=7-2	4=7-3	4=7-3	
022	111	101	101	没有屏蔽所有者的任何权限,屏蔽了所属组和其他用户的写权限
	7=7-0	5=7-2	5=7-2	

例:将当前 shell 的 umask 值设置成 555。

```
[globus@wang globus]$ umask 555
```

验证你的设置。

```
[globus@wang globus]$ umask
```

```
0555
```

umask 值 0555 说明在以后创建文件时,默认的权限为所有用户 (文件主、所属组和其他用户) 都具有写权限 (2=7-5)。

使用带-p 和-S 参数显示对 umask 的设置。

```
[globus@wang globus]$ umask -S -p
```

```
umask -S u=w,g=w,o=w
```

其显示结果 (umask -S u=w,g=w,o=w) 可以重新用来设置 umask 值,因为它是一条完



整的 `umask` 命令，而且与当前的设置相同。

在当前目录创建一个名为 `test` 的文件。

```
[globus@wang globus]$ touch test
```

使用 `ls` 命令来验证刚创建的 `test` 文件。

```
[globus@wang globus]$ ls -l test
```

```
--w--w--w-    1 globus  globus          0  5月 23 01:26 test
```

例：创建一个新文件 `newfile`，文件主具有读和写权限（6），所属组具有写权限（2），其他用户不具有任何执行权限（0）。由于文件主具有所有权限，因此对应的屏蔽值为  $1=7-6$ ，所属组对应的屏蔽值为  $5=7-2$ ，其他用户的屏蔽值为  $7=7-0$ ，即应将 `umask` 的值设置成 157。执行的命令如下：

设置 `umask` 的值。

```
[globus@wang globus]$ umask 157
```

创建文件。

```
[globus@wang globus]$ ls > newfile
```

查看权限字段。

```
[globus@wang globus]$ ls -l newfile
```

```
-rw--w-----    1 globus  globus        251  5月 23 02:25 newfile
```

从列表中可以看出，结果与要求相一致。

### 3.3.3 文件比较、查找和定位操作

Linux 系统提供了丰富的文件比较、查找和定位命令，用来大大简化用户搜索工作。为了找出两个文件的差别，用户可以使用 `diff` 命令对这两个文件进行比较。当用户知道系统上存在某一个文件，但是不知道这个文件到底在哪个文件夹里，可以使用文件定位命令 `slocate` (`locate`) 或者 `find` 命令。如果用户有一个文件存在，需要查看文件里是否包含某一字符串，进一步知道这个字符串在文件的所有位置，需要使用文件查找命令 `grep`。对于一个文件，如果需要知道这个文件的类型，可以使用文件类型查看命令 `file`。当用户希望查看某一个命令是否存在可执行文件搜索路径中，可以使用 `which` 命令。同时也可以使用文件排序命令 `sort` 排序文件。本小节还介绍了与之相关的其他命令。

#### 1. 文件比较命令——`diff`

`diff` 命令的功能为逐行比较两个文本文件，列出其不同之处。它比 `comm` 命令功能更强。它对给出的文件进行系统的检查，并显示出两个文件中所有不同的行，这些行在各自文件中的位置，以及要将一个文件转换为另一个文件需要对那些行进行添加、修改或者修改动作，它不要求事先对文件进行排序。

`diff` 命令的语法为：

```
diff [选项] file1 file2
```

其中 `file1` 和 `file2` 是要比较的文件。

`diff` 各选项及其含义如表 3-34 所示。

表 3-34 diff 命令的常用选项

选项	含义
-b	忽略行尾的空格，而字符串中的一个或多个空格符都视为相等，如 I like it 与 I like it 被视为相同的字符串
-c	采用上下文输出格式（提供3行上下文）
-Cn	采用上下文输出格式（提供n行上下文）
-r	当file1和file2是目录时，递归作用到各文件和目录上
-i	比较时忽略大小写
--help	显示本命令的帮助

例：假设存在两个文件 diff1.txt 和 diff2.txt，使用 cat 命令显示内容。

```
[globus@wang globus]$ cat diff1.txt
```

```
Larry King is a broadcaster.
```

```
He is good sense of humor.
```

```
He tells good stories.
```

```
King is virtually inescapable.
```

```
[globus@wang globus]$ cat diff2.txt
```

```
Larry King is a broadcaster.
```

```
He is a humorous man.
```

```
He tells good stories.
```

```
King is really inescapable.
```

```
He never prepares for an interview in advance.
```

使用 diff 命令查找它们之间的不同之处：

```
[globus@wang globus]$ diff diff1.txt diff2.txt
```

```
2c2 ----- 第一个文件的第2行与第二个文件的第2行不同
```

```
< He is good sense of humor. 第一个文件的第2行内容
```

```
---
```

```
> He is a humorous man. 第二个文件的第2行内容
```

```
4c4,5 ----- 第一个文件的第4行与第二个文件的第4~5行不同
```

```
< King is virtually inescapable. 第一个文件的第4行内容
```

```
---
```

```
> King is really inescapable. } 第二个文件的第4~5行内容
```

```
> He never prepares for an interview in advance.
```

在这个例子里，diff 输出的第一行为 2c2，表明第一个文件的第 2 行与第二个文件的第 2 行不同，在紧接着的三行输出中，以“<”开头的行表示是第一个文件的内容，以“>”开头的行则表示是第二个文件的内容。内容为“---”的行表示第一个文件中的内容与第二个文件中不同的内容之间的分隔行。在 diff 输出实例中，第 2~4 行表示第一个文件的第 2 行内容为“He is good sense of humor.”，而第二个文件的内容则为“He is a humorous man.”。

## 2. 排序命令 —— sort

sort 命令的功能是对文件中的各行进行排序。sort 命令有许多非常实用的选项，这些选项最初是用来对数据库格式的文件内容进行各种排序操作的。实际上，sort 命令可以认为是一个非常强大的数据管理工具。

sort 命令将逐行对文件中的内容进行排序，如果两行的首字符相同，该命令将继续比较这两行的下一字符，如果还相同，将继续进行比较，直到这两行比较完毕，或者遇到不相同的字符，或者一行比较结束，而另一行还有剩余字符。

语法：

sort [选项] 文件

sort 命令对指定文件中所有的行进行排序，并将结果显示在标准输出上。如不指定输入文件或使用“”，则表示排序内容来自标准输入。sort 排序是根据从输入行抽取的一个或多个关键字进行比较。排序关键字定义了用来排序的最小的字符序列。默认情况下它以整行为关键字按 ASCII 字符顺序进行排序。

在当前目录使用 cat 命令创建一个名为 to-sort.txt 的文本文件。笔者将以该文件为例来讲述 sort 命令。

使用 cat 命令显示 to-sort.txt 文件的内容：

```
[globus@wang globus]$ cat to-sort.txt
```

```
2345
```

```
Book
```

```
  space
```

space 前有两个空格字符

```
*123
```

```
+OPQ
```

```
zip
```

```
1234
```

```
45
```

```
Keep back
```

```
=1234
```

由两个空格字符组成的行

```
]!!!
```

```
abc
```

```
%opq
```

由一个空格字符组成的行

```
opq
```

例：使用默认选项排序当前目录下的文件 to-sort.txt。执行过程如下：

```
[globus@wang globus]$ sort to-sort.txt
```

```
*123
```

```

1234
=1234
2345
45
abc
Book
ljjj
Keep back
opq
%opq
+OPQ
    space
zip

```

在Linux中，`sort`命令默认情况下按字典序排序。按字典顺序排序时，比较时仅字母、数字、空格和制表符有意义，忽略文件中其他的字符（如%，#，<，）等字符），并且不区分大小写，数字在前，字母在后。

改变默认设置的常见选项如表3-35所示。

表 3-35 `sort` 命令的常用选项

选项	含义
<code>-m</code>	合并给定的已排好序的文件
<code>-c</code>	检查给定文件是否已排好序，如果它们没有都排好序，则打印一个出错信息，并以状态值1退出
<code>-u</code>	对排序后认为相同的行，只保留其中一行
<code>-o</code>	将排序结果输出到输出文件，而不是标准输出，如果输出文件是输入文件之一， <code>sort</code> 先将该文件的内容写入一个临时文件，然后再排序和写输出结果

改变默认排序规则的主要选项如表3-36所示。

表 3-36 `sort` 命令的常用排序规则

排序规则	含义
<code>-d</code>	按字典顺序排序，比较时仅字母、数字、空格和制表符有意义
<code>-f</code>	将小写字母与大写字母同等对待
<code>-i</code>	忽略非打印字符
<code>-M</code>	作为月份比较：“JAN” “FEB” “DEC”
<code>-r</code>	按逆序输出排序结果
<code>+pos1 pos2</code>	指定一个或几个字段作为排序关键字，字段位置从 <code>pos1</code> 开始，到 <code>pos2</code> 为止（包括 <code>pos1</code> ，但不包括 <code>pos2</code> ）。如不指定 <code>pos2</code> ，则关键字为从 <code>pos1</code> 到行尾。字段和字符的位置从0开始
<code>-b</code>	在每行中寻找排序关键字时，忽略前导的空白（空格和制表符）
<code>-t separator</code>	指定字符 <code>separator</code> 作为字段分隔符

如果希望 `sort` 命令按照 ASCII 码字符集的顺序排序, 那么用户必须设置 `LC_ALL` 环境变量, 将该环境变量设置为 `POSIX`。

要设置该环境变量, 在 `bash` 或者 `ksh shell` 中, 执行:

```
export LC_ALL="POSIX"
```

而对于 `csch` 或者 `tcsh`, 执行:

```
setenv LC_ALL="POSIX"
```

例: 按 ASCII 码字符集的顺序排序当前目录的 `to-sort.txt` 文件。执行过程如下:

```
[globus@wang globus]$ export LC_ALL="POSIX"
```

```
[globus@wang globus]$ sort to-sort.txt
```

```
space
%opq
*123
+OPQ
1234
2345
45
.=1234
Book
Keep back
ljjj
abc
opq
zip
```

要查看 ASCII 字符集的顺序, 使用命令 `man ascii`, 示例如下:

```
[globus@wang globus]$ man ascii
```

NAME

```
ascii - the ASCII character set encoded in octal, decimal, and hexadecimal
```

DESCRIPTION

```
ASCII is the American Standard Code for Information Interchange. It is
a 7-bit code. Many 8-bit codes (such as ISO 8859-1, the Linux default
character set) contain ASCII as their lower half. The international
counterpart of ASCII is known as ISO 646.
```

```
The following table contains the 128 ASCII characters.
```

```
C program '\X' escapes are noted.
```

```
Oct  Dec  Hex  Char          Oct  Dec  Hex  Char
```

```
-----
```

000	0	00	NUL '\0'	100	64	40	@
001	1	01	SOH	101	65	41	A
002	2	02	STX	102	66	42	B
003	3	03	ETX	103	67	43	C
004	4	04	EOT	104	68	44	D

:

事实上是 `man` 帮助窗口，因此可以使用 `man` 的操作命令查看 ASCII 码字符的顺序。

例：不区分大小写（使用 `-f` 选项）。

```
[globus@wang globus]$ sort -f to-sort.txt
```

```
space
%opq
*123
+OPQ
1234
2345
45
=1234
abc
Book
Keep back
opq
zip
ljjj
```

字母排序中忽略了大小写

对于数字按字典序和 ASCII 码顺序排序。对于 1234 和 234 而言，排序的结果应该为 1234 在前，234 在后（原因是 1 的 ASCII 码的值小于 2 的 ASCII 码）；但是如果使用 `-n` 选项，则 `sort` 排序是按数值的值排序，对于 1234 和 234 而言，排序结果为 234 在前，1234 在后（1234>234）。

例：对文件 `to-sort.txt`，按数值大小排序。执行如下：

```
[globus@wang globus]$ sort -n to-sort.txt
```

```
space
%opq
*123
+OPQ
=1234
Book
```

```
Keep back
```

```
ljjj
```

```
abc
```

```
opq
```

```
zip
```

```
45
```

```
1234
```

```
2345
```



按数值排序

例：对/etc/passwd 文件以“:”作为分隔符，对文件的第一个字段进行排序，并且将排序的结果保存在当前用户的主目录下的 passwd-sort.txt 文件中。命令如下：

```
[globus@wang globus]$ sort +0 -1 -t: /etc/passwd -o ~/passwd-sort.txt
```

命令执行分析：文件/etc/passwd 中包含了系统中所有用户的重要信息。查看文件的内容：

```
[globus@wang globus]$ head -2 /etc/passwd
```

```
root:x:0:0:root:/root:/bin/bash
```

```
bin:x:1:1:bin:/bin:/sbin/nologin
```

可见它的每一行由若干个冒号(:)分隔的字段组成。其中每一行的第一个字段表示系统中的一个用户的用户名，因此在命令中需要使用“-t:”选项，表示将分号作为分隔符。又因需要对第一个字段排序，所以使用“+0 -1”选项将取第 0 个分隔符与第一个分隔符之间的字段（即第一个字段）作为排序依据。“-o ~/passwd-sort.txt”选项将文件保存到主目录的 passwd-sort.txt 文件。

例：对文件 to-sort.txt 进行逆向排序。执行过程如下：

```
[globus@wang globus]$ sort -r to-sort.txt
```

```
zip
```

```
opq
```

```
abc
```

```
ljjj
```

```
Keep back
```

```
Book
```

```
=1234
```

```
45
```

```
2345
```

```
1234
```

```
+OPQ
```

```
*123
```

```
%opq
```

```
space
```

### 3. 识别和删除重复行命令 —— uniq

文件经过处理后在它的输出文件中可能会出现重复的行。例如，使用 `cat` 命令将两个文件合并后，再使用 `sort` 命令进行排序，就可能出现重复行。这时可以用 `uniq` 命令将这些重复行从输出文件中删除，只留下每条记录的惟一记录。

`uniq` 命令的语法如下：

`uniq [选项] 文件`

该命令读取输入文件，并比较相邻的行。在正常情况下，第二个及更多个重复行将被删去，行比较是根据所用字符集的排序序列进行的。该命令加工后的结果写到输出文件中。输入文件和输出文件不能相同。如果输入文件用 “-” 表示，则从标准输入读取。该命令各选项含义如表 3-37 所示。

注意：`uniq` 读取输入文件，并比较相邻的行。因此如果要求删除所有的重复行，可以先对文件进行排序。

表 3-37 `uniq` 命令的常用选项

排序规则	含义
-c	显示输出中，在每行行首加上本行在文件中出现的次数。它可取代 <u>u</u> 和 <u>d</u> 选项
-d	只显示重复行
-u	只显示文件中不重复的各行
-n	前n个字段与每个字段前的空白一起被忽略。一个字段是一个非空格、非制表符的字符串，彼此由制表符和空格隔开（字段从0开始编号）
+n	前n个字符被忽略（字符从0开始编号）
-fn	n是字段
-sn	与+n相同，这里n是字符数

例：使用 `cat` 命令查看文件 `diff1.txt` 和 `diff2.txt`。命令如下：

```
[globus@wang globus]$ cat diff1.txt
```

显示结果如下：

```
Larry King is a broadcaster.
```

```
He is good sense of humor.
```

```
He tells good stories_____
```

重复行

```
King is virtually inescapable.
```

```
[globus@wang globus]$ cat diff2.txt
```

显示结果如下：

```
Larry King is a broadcaster.
```

```
He is a humorous man.
```

```
He tells good stories_____
```

重复行

```
King is really inescapable.
```

```
He never prepares for an interview in advance.
```

删除这两个文件合并后的重复行。

```
[globus@wang globus]$ sort diff1.txt diff2.txt | uniq
```



执行结果如下：

```
He is a humorous man.
He is good sense of humor.
He never prepares for an interview in advance.
He tells good stories.
King is really inescapable.
King is virtually inescapable.
Larry King is a broadcaster.
只显示这两个文件合并后的重复行。
[globus@wang globus]$ sort diff1.txt diff2.txt | uniq -d
```

执行结果如下：

```
He tells good stories.
Larry King is a broadcaster.
```

4. 文件查找命令 —— find

find 命令的功能是在目录中搜索文件，并执行指定的操作。因为该命令提供了相当多的查找条件，所以其功能非常强大。

find 命令语法的语法如下：

```
find 起始目录 寻找条件 操作
```

说明：find 命令从指定的起始目录开始，递归地搜索其各个子目录，查找满足寻找条件的文件，并对其采取相关的操作。

该命令提供的寻找条件可以是一个用逻辑运算符 not、and 或者 or 组成的复合条件。逻辑运算符 and、or 或者 not 的含义如表 3-38 所示。

表 3-38 find 命令的逻辑运算符及其含义

运算符	命令中的表示	含义
and	-a	它是系统默认的选项，表示只有当所给的条件都满足时，才认为满足寻找条件
or	-o	该运算符表示只要所给的条件中有一个满足时，就认为满足寻找条件
not	!	该运算符表示查找不满足所给条件的文件

在当前目录及其子目录中查找同时满足如下条件的文件：

- 文件名以 hell 打头。
- 文件的属主为 globus。

find 命令及其执行过程如下：

```
[globus@wang globus]$ find -name hell* -user 'globus'
./hello1.txt
```

在当前目录及其子目录查找以 hell 打头或者以 diff 打头的文件：

```
[globus@wang globus]$ find -name 'hell*' -o -name 'diff*'
```

执行结果如下：

```
./diff2.txt
./diff1.txt
./hello1.txt
```

在当前目录及其子目录中查找所有不以 `hell` 打头的文件:

```
[globus@wang globus]$ find ! -name 'hell*'
```

笔者在此忽略查找结果。

**注意:** 当使用很多的逻辑选项时, 可以使用括号把这些选项括起来。为了避免 `shell` 本身对括号误解, 用户需要在括号前加转义字符 `\` 来去除括号的意义。

例如, 对于要查找的字符串 `"(55$)"`, 则在查找条件中应该表达为:

```
\(55$\)
```

查找条件包含如表 3-39 所示的常见选项。

表 3-39 Find 命令的查找条件中可以使用的常见选项

分类	查找选项	含 义
n	<code>+n</code>	表示大于n的值, n表示一个数值, 如203
	<code>-n</code>	表示小于n的值
	<code>n</code>	表示正好是n
以 文 件 名 称 和 文 件 属 性 查 找	<code>-name '字符串'</code>	查找文件名与所给字符串匹配的所有文件, 字符串内可用通配符 <code>*</code> 、 <code>?</code> 和 <code>[]</code>
	<code>-lname '字符串'</code>	查找文件名与所给字符串匹配的所有符号链接文件, 字符串内可用通配符 <code>*</code> 、 <code>?</code> 和 <code>[]</code>
	<code>-gid n</code>	查找属于ID号为n的用户组的所有文件
	<code>-uid n</code>	查找属于ID号为n的用户的所有文件
	<code>-group '字符串'</code>	查找属于组名为所给字符串的所有的文件
	<code>-user '字符串'</code>	查找属于用户名为所给字符串的所有的文件
	<code>-empty</code>	查找大小为0的目录或文件
	<code>-path</code>	查找与所给字符串匹配的路径名中包含的所有文件, 字符串内可用通配符 <code>*</code> 、 <code>?</code> 和 <code>[]</code>
	<code>-perm '权限'</code>	查找具有指定权限的文件和目录, 权限的表示可以是数字, 如755和777
	<code>-size n[bckw]</code>	查找指定文件大小的文件, n后面的字符表示单位, 默认为b, 代表512字节的块
	<code>-type x</code>	查找类型为x的文件, x的可能取值如下: b 块设备文件 c 字符设备文件 d 目录文件 p 命名管道 (FIFO) f 普通文件 l 符号链接文件 (symbolic links) s socket文件
	<code>-xtype x</code>	与-type基本相同, 但只查找符号链接文件

续表

分类	查找选项	含 义
以 时 间 为 条 件 查 找	-amin n	查找n分钟以前被访问过的所有文件
	-atime n	查找n天以前被访问过的所有文件
	-cmin n	查找n分钟以前文件状态被修改过的所有文件
	-ctime n	查找n天以前文件状态被修改过的所有文件
	-mmin n	查找n分钟以前文件内容被修改过的所有文件
	-mtime n	查找n天以前文件内容被修改过的所有文件
可 执 行 的 操 作	-exec 命令名称	对符合条件的文件执行所给的Linux命令，而不询问用户是否需要执行该命令。表示命令的参数即为所找到的文件；命令的末尾必须以“\;”结束
	-ok 命令名称	对符合条件的文件执行所给的Linux命令，与exec不同的是，它会询问用户是否需要执行该命令
	-fls 文件名	详细列出所找到的所有文件，并写入文件
	-fprintf 文件名	将找到的文件名写入指定文件
	-print	在标准输出设备上显示查找出的文件名
	-printf 格式	格式的写法请参考有关C语言语法的书

注意：find 命令的第一个以“-”、“(”、“)”、“,”或者“!”符号之一开始的参数表明表达式的开始，表达式之前的参数是搜索路径。

例：在当前目录查找以 hello 打头的文件，并详细列出找到的所有文件。

```
[globus@wang globus]$ find . -name 'diff*' -exec ls -l {} \;
-rw-rw-r-- 1 globus globus 149 5月 21 22:41 ./diff2.txt
-rw-rw-r-- 1 globus globus 110 5月 21 22:36 ./diff1.txt
-rw-rw-r-- 1 globus globus 110 5月 23 04:11 ./temp/diff3.txt
```

例：删除从根目录开始的以及递归的所有子目录下所有 10 天之内没有被访问过的 a.out 或\*.o 文件。

```
[globus@wang globus]$ find / \( -name 'a.out' -o -name '*.o' \) \
    -mtime +10 -exec rm {} \;
```

## 5. 数据库支持的文件查找命令 —— locate

locate 命令（在 Linux 中，事实上是 slocate 命令）用于查找文件，它比 find 命令的搜索速度快。它需要一个数据库，这个数据库由每天的例行工作（crontab）程序来更新的。当建立好这个数据库后，就可以方便地来搜寻所需文件了。

该命令的一般形式为：

locate 文件名模式

例：查找文件名中包含 hello 的文件。

```
[globus@wang globus]$ locate hello
/usr/bin/bonobo-sample-hello
/usr/lib/python2.2/__phello__.foo.py
```

```

/usr/lib/python2.2/__phello__.foo.pyc
/usr/lib/python2.2/__phello__.foo.pyo
/usr/share/doc/bash-2.05b/loadables/hello.c
/usr/share/doc/pygtk2-1.99.14/examples/simple/hello.py
/usr/share/doc/gnome-python2-bonobo-1.99.14/bonobo/bonoboui/hello.py
/usr/share/locale/zh_CN/LC_MESSAGES/katehelloworld.mo
/usr/share/locale/zh_TW/LC_MESSAGES/katehelloworld.mo

```

**注意：**locate 比 find 查找快，但是 locate 的检索条件比 find 少，它仅能给出文件的特征这一惟一的检索条件；而且对于刚刚新建的文件，可能无法被检索出来。

**例：**建立一个新文件 hello7.txt，然后使用 locate 检索文件名中包含 hello 的文件。

```

[glorus@wang glorus]$ touch hello7.txt
[glorus@wang glorus]$ locate hello
/usr/bin/bonobo-sample-hello
/usr/lib/python2.2/__phello__.foo.py
/usr/lib/python2.2/__phello__.foo.pyc
/usr/lib/python2.2/__phello__.foo.pyo
/usr/share/doc/bash-2.05b/loadables/hello.c
/usr/share/doc/pygtk2-1.99.14/examples/simple/hello.py
/usr/share/doc/gnome-python2-bonobo-1.99.14/bonobo/bonoboui/hello.py
/usr/share/locale/zh_CN/LC_MESSAGES/katehelloworld.mo
/usr/share/locale/zh_TW/LC_MESSAGES/katehelloworld.mo

```

从结果中可以看出，刚刚建立的文件 hello7.txt 没有被检索出来。

有关该命令的详细信息，请参考本书的第 2 章，或者 locate 的 man 手册页。

## 6. 文件内容查询命令 —— grep、fgrep 和 egrep

这三个命令以指定模式搜索文件，并通知用户在什么文件中搜索到与指定的模式匹配的字符串，并打印出所有包含该字符串的文本行，在该文本行的最前面是该行所在的文件名。grep 命令一次只能搜索一个指定的模式；egrep 命令检索扩展的正则表达式（包括表达式组和可选项）；fgrep 命令检索固定字符串，但它不识别正则表达式，是快速的搜索命令。这三个程序在搜索与定位文件中特定的主题方面非常有用。要搜索的模式可以被认为是一些关键词，它们可以用来搜索文件中包含的这些关键词。例如当用户编写程序时，可以用它们来寻找某一个函数，或者一些相关的词。grep 命令的搜索功能比 fgrep 强大，因为 grep 命令的搜索模式可以是正则表达式，而 fgrep 却不能。

该组命令中的每一个命令都有一组选项，利用这些选项可以改变其输出方式。例如它们可以在搜索到的文本行上加行号，或者只输出文本行的行号，或者输出所有与搜索模式不匹配的文本行，或只简单地输出已搜索到指定模式的文件名，并且可以指定在查找模式时忽略大小写。

这组命令在指定的输入文件中查找与模式匹配的行。如果没有指定文件，则从标准输入中读取。默认情况下，每个匹配的行显示到标准输出。如果想知道已查到串包含在哪个

文件，则可以使它在每一行输出之前加上对应的文件名。

这组命令的语法如下：

`grep [选项] [查找模式] [文件列表]`

`egrep [选项] [查找模式] [文件列表]`

`fgrep [选项] [查找模式] [文件列表]`

`grep` 命令常见的选项如表 3-40 所示。

表 3-40 `grep` 命令常见的选项

查找选项	含义
E	每个模式作为一个扩展的正则表达式对待
F	每个模式作为一组固定字符串对待（以新行分隔），而不作为正则表达式
b	在输出的每一行前显示包含匹配字符串的行在文件中的字节偏移量
c	只显示匹配行的数量
i	比较时不区分大小写
h	在查找多个文件时，指示 <code>grep</code> 不要将文件名加入到输出之前
l	显示首次匹配串所在的文件名，并用换行符将其隔开。当在某文件中多次出现匹配串时，不重复显示此文件名
n	在输出前加上匹配串所在行的行号（文件首行行号为1）
v	只显示不包含匹配串的行
x	只显示整行严格匹配的行
e 表达式	指定检索使用的模式。用于防止以“-”开头的模式被解释为命令选项
f 表达式文件	从文件中获取要搜索的模式，一个模式占一行

**注意：**在使用这组命令时，首先在命令后输入搜索的模式，再输入要搜索的文件。其中，文件名列表也可以使用特殊字符，如“\*”可以生成全部文件的文件名列表。如果想在搜索模式中包含有空格的字符串，可以使用单引号把要搜索的模式括起来，用来表明搜索的模式是由包含空格的字符串组成。否则 shell 将把空格认为是命令行参数的定界符，而 `grep` 命令将把搜索模式中的单词解释为文件名列表中的一部分。

在下面的例子中，`grep` 命令在文件 `diff1.txt` 和 `diff2.txt` 中搜索指定内容。使用 `cat` 命令显示文件 `diff1.txt` 的内容：

```
[globus@toshiba globus]$ cat diff1.txt
```

```
Larry King is a broadcaster.
```

```
He is good sense of humor.
```

```
He tells good stories.
```

```
King is virtually inescapable.
```

使用 `cat` 命令显示文件 `diff2.txt` 的内容：

```
[globus@toshiba globus]$ cat diff2.txt
```

```
Larry King is a broadcaster.
```

```
He is a humorous man.
```

```
He tells good stories.
```

```
King is really inescapable.
He never prepares for an interview in advance.
```

使用 **grep** 搜索文件并选择目标字符串的行。

例：分别在 `diff1.txt` 和 `diff2.txt` 文件中查找包含模式 **King** 的行。

```
[globus@toshiba globus]$ grep King diff1.txt
Larry King is a broadcaster.
King is virtually inescapable.
[globus@toshiba globus]$ grep King diff2.txt
Larry King is a broadcaster.
King is really inescapable.
```

**注意：****grep** 在文件中搜索包含模式（目标字符串）的行，默认输出到显示器，输出中包含的目标字符串可能在行的任何位置。

例：可以在多个文件中搜索模式，例如在 `diff1.txt` 和 `diff2.txt` 文件搜索目标字符串 **King**。

执行过程如下：

```
[globus@toshiba globus]$ grep King diff1.txt diff2.txt
diff1.txt:Larry King is a broadcaster.
diff1.txt:King is virtually inescapable.
diff2.txt:Larry King is a broadcaster.
diff2.txt:King is really inescapable.
```

**注意：**用户可以在搜索文件名中使用通配符\*和?。例如上例也可以使用“**grep King diff\***”命令，在以 **diff** 打头的文件里搜索字符串 **King**。

当在多个文件里搜索一个模式时，可以使用 **-n** 选项指明该行是来自哪一个文件，在每一行的开头显示它所在的文件名，并且显示所在行行号。

```
[globus@toshiba globus]$ grep -n King diff1.txt diff2.txt
diff1.txt:1:Larry King is a broadcaster.
diff1.txt:4:King is virtually inescapable.
diff2.txt:1:Larry King is a broadcaster.
diff2.txt:4:King is really inescapable.
```

例：使用 **-r** 选项在当前目录树下的所有文件搜索字符串 **King**。

```
[globus@toshiba globus]$ grep -r -n King .
```

使用 **-l** 选项只列出目录下包含匹配文件的文件名，而不显示所有匹配的行。

例：在文件 `diff1.txt` 和 `diff2.txt` 中查找字符串 **King**，只显示包含 **King** 字符串的文件名。

```
[globus@toshiba globus]$ grep -l King diff1.txt diff2.txt
diff1.txt
diff2.txt
```

有时确定每一个文件中匹配次数是很重要的，这时可以使用 **-c** 选项。选项 **-c** 使 **grep** 的输出只包含两个字段：所有被检查的文件名和对应文件中匹配的次数。

例：在 `diff1.txt` 和 `diff2.txt` 文件中分别查看包含字符 **o**（小写字母 **o**）的个数。

```
[globus@toshiba globus]$ grep -c o diff1.txt diff2.txt
```

```
diff1.txt:3
```

```
diff2.txt:4
```

可以使用-x 选项进行整行匹配。

例：在 diff1.txt 和 diff2.txt 文件中分别查看包含 He tells good stories. 的行。

```
[globus@toshiba globus]$ grep -x "He tells good stories." diff1.txt diff2.txt
```

```
diff1.txt:He tells good stories.
```

```
diff2.txt:He tells good stories.
```

也可以使用-v 选项让 grep 选择与目标不匹配的行。

例：在 diff1.txt 中查找不包含字符串 is 的行。

```
[globus@toshiba globus]$ grep -v is diff1.txt
```

```
He tells good stories.
```

如果使用搜索条件中包含 grep 使用的特殊符号，如 grep 使用的选项，会引起解释错误，在这种情况下，可以在模式前使用-e 选项，避免 grep 去解释这些符号。

例：在当前目录及其子树里的所有文件中搜索字符串“-c”。

```
[globus@toshiba globus]$ grep -e -c *
```

使用基本的元字符搜索行。在字符串搜索中，grep 可以将目标字符串的第一个字符、最后一个字符和其他的特定字符解释为特殊含义，而不是字符本身的含义。这些特殊的字符叫做元字符（metacharacter），用这些字符串表示特定的字符串规则。

注意：在这些元字符中，“^”表示文本行的开头，“\$”表示文本行的结束，“.”表示任何一个字符。转意字符“\”将元字符转化为普通字符。“\<word>”表示一个由 word 组成的单词。用[]表示搜索可选择的字符。

例：在文件 diff1.txt 中搜索以字符 H 打头的行。

```
[globus@toshiba globus]$ grep '^H' diff1.txt
```

```
He is good sense of humor.
```

```
He tells good stories.
```

例：在文件 diff1.txt 中搜索不以字符串 He 打头的行。

```
[globus@toshiba globus]$ grep -v '^He' diff1.txt
```

```
Larry King is a broadcaster.
```

```
King is virtually inescapable.
```

例：在文件 diff1.txt 中搜索以“r”和任意一个其他字符结尾的行。

```
[globus@toshiba globus]$ grep 'r.$' diff1.txt
```

```
Larry King is a broadcaster.
```

```
He is good sense of humor.
```

例：在文件 diff1.txt 中搜索只有字符串 He is 结尾的行。

```
[globus@toshiba globus]$ grep '^He$' diff1.txt
```

该命令没有查到所需要的记录。

例：在文件 diff1.txt 中搜索只包含字符串 in 和 ie 的行。

```
[globus@toshiba globus]$ grep 'i(en)' diff1.txt
```

```
Larry King is a broadcaster.
```

He tells good stories.

King is virtually inescapable.

使用-i选项忽略搜索字符串的大小写。

例：在文件 diff1.txt 中搜索包含字母 L 或者字母 l（L 小写形式）的行。

```
[globus@toshiba globus]$ grep '[Ll]' diff1.txt
```

Larry King is a broadcaster.

He tells good stories.

King is virtually inescapable.

或者

```
[globus@toshiba globus]$ grep -i 'L' diff1.txt
```

Larry King is a broadcaster.

He tells good stories.

King is virtually inescapable.

例：在文件 diff1.txt 中搜索至少包含一个字符的行。

```
[globus@toshiba globus]$ grep '.' diff1.txt
```

Larry King is a broadcaster.

He is good sense of humor.

He tells good stories.

King is virtually inescapable.

例：在文件 diff1.txt 中搜索以 e 加任意个字符，再加一个 “.” 字符结束的行。

```
[globus@toshiba globus]$ grep 'e\.' diff1.txt
```

Larry King is a broadcaster.

He tells good stories.

例：在文件 diff1.txt 中搜索以单词 a 组成的行。

```
[globus@toshiba globus]$ grep '\<a\>' diff1.txt
```

Larry King is a broadcaster.

例：在文件 diff1.txt 中搜索以字母 o 后跟 a、b、c 和 d 字符之一组成的行。

```
[globus@toshiba globus]$ grep 'o[a-d]' diff1.txt
```

Larry King is a broadcaster.

He is good sense of humor.

He tells good stories.

或者

```
. [globus@toshiba globus]$ grep 'o[abcd]' diff1.txt
```

Larry King is a broadcaster.

He is good sense of humor.

He tells good stories.

### 3.3.4 模拟 DOS 文件和目录命令

Linux 系统提供了一组称为 mtools 的可移植工具，它们可以让用户轻松地从事标准的 DOS



软盘上读、写文件和目录。这对 DOS 和 Linux 环境之间的交换文件非常有用。它们是不同的文件系统格式的系统之间交换文件的有力手段。对于一个 MS-DOS 格式的软盘，只要把软盘放在驱动器中，就可以利用 mtools 提供的命令来访问软盘上的文件。

注意：在使用 mtools 工具时，用户不必使用 mount 命令将软盘挂接，成为文件系统的一部分。

mtools 的主要命令如表 3-41 所示。

表 3-41 mtools 的常见命令

mtools 命令	功 能
mcd 目录名	改变MSDOS软盘的当前目录
mcopy 源文件 目标文件	在MSDOS和Unix之间复制文件
mdel 文件名	删除MSDOS软盘中文件
mdir 目录名	显示MSDOS软盘的目录
mformat 驱动器号	在低级格式化的软盘上创建MSDOS文件系统
rnlabel 驱动器号	产生MSDOS软盘的卷标
mmkd 目录名	建立MSDOS目录
mrkd 目录名	删除MSDOS目录
mren 源文件 目标文件	重新命名已存在的MSDOS文件
mtype 文件名	显示MSDOS文件的内容

这些命令和 MSDOS 中对应的不加 m 的 MSDOS 命令的功能非常相似。

1. mdir

mdir 与 DOS 的 dir 类似，它的格式如下：

mdir [-/] [-f] [-w] [-a] [-b] msdos 文件列表

其中各选项的功能如表 3-42 所示。

表 3-42 mdir 常见的选项

选 项	功 能
-/	与DOS的dir命令的-s选项类似，它递归的输出快速列目录，并不计算磁盘上的自由空间，对于大FAT格式磁盘，
-f	计算磁盘上的自由空间将要花费很长时间，-f选项可以绕过这一步。对于FAT32文件系统，自由空间将会存储在分区中
-a	同时列出隐含文件
-b	简单形式，只列出文件和目录名
-w	宽行输出

例：列出 A: 的目录。命令如下：

```
[globus@wang globus]$ mdir
```

结果如下：

```
Volume in drive A has no label
Directory for A: /
```

```

LFORMAT  EXE      64888 02-12-1998 23:44
HELP      HLP      8818 01-22-2002 11:49
diskgen   exe      141187 03-03-2002 23:01
LICENSE   TXT      1451 01-22-2002 16:41
UPGRADE   TXT      2107 03-03-2002 23:12
LOOKBACK  DAT      20 11-17-2003 9:12
        6 files          218 471 bytes
                        1 237 504 bytes free

```

从结果可以看出，A：盘的根目录有6个文件，而且没有下级子目录。

例：使用简单形式列出A：的目录。命令如下：

```
[globus@wang globus]$ mdir -b
```

结果如下：

```

A:/LFORMAT.EXE
A:/HELP.HLP
A:/diskgen.exe
A:/LICENSE.TXT
A:/UPGRADE.TXT
A:/LOOKBACK.DAT

```

例：宽行形式列出A：的目录，并且显示隐含文件。命令如下。

```
[globus@wang globus]$ mdir -w -a
```

结果如下：

```

Volume in drive A has no label
Directory for A:/

```

```

LFORMAT.EXE  HELP.HLP  diskgen.exe  LICENSE.TXT  UPGRADE.TXT
LOOKBACK.DAT
        6 files          218 471 bytes
                        1 237 504 bytes free

```

## 2. mmd

在A：盘上创建一个DOS子目录：

```
[globus@wang globus]$ mmd DOS
```

显示DOS子目录中的信息：

```

[globus@wang globus]$ mdir DOS
Volume in drive A has no label
Directory for A:/DOS

```

```

.          <DIR>      05-26-2004  2:18
..         <DIR>      05-26-2004  2:18

```

```

2 files
0 bytes
1 236 992 bytes free

```

这个新建的目录为空。

### 3. mcd

它与 DOS 的 `cd` 命令类似。如果 `mcd` 不带参数，它将显示 DOS 的当前目录。

例：转向 A: 盘的 DOS 子目录。命令如下：

```
[globus@wang globus]$ mcd DOS
```

显示 A: 盘的当前目录：

```
[globus@wang globus]$ mcd
```

```
A:/DOS
```

当前驱动器的当前目录为 A:/DOS。

### 4. Mcopy

在 Linux 和 DOS 之间复制文件。

例：将 A:/下的 LICENSE.TXT 拷贝到当前目录。

```
[globus@wang globus]$ mcopy A:/LICENSE.TXT .
```

列目录验证：

```
[globus@wang globus]$ ls -l LICENSE.TXT
```

```
-rw-rw-r-- 1 globus globus 1451 5月 26 02:31 LICENSE.TXT
```

例：将 A:/下的 LICENSE.TXT 拷贝到 A:/DOS 子目录。

```
[globus@wang globus]$ mcopy A:/LICENSE.TXT A:/DOS
```

使用 `mdir` 验证如下：

```
[globus@wang globus]$ mdir -wf A:/DOS
```

```
Volume in drive A has no label
```

```
Directory for A:/DOS
```

```
[.]          [...]          LICENSE.TXT
```

```
3 files          1 451 bytes
```

### 5. Mtype

显示文件内容。

例：显示 A:/DOS/LICENSE.TXT 文件的内容。命令如下：

```
[globus@wang globus]$ mtype A:/DOS/LICENSE.TXT
```

### 6. Mren

文件更名。

例：将 A:/DOS/LICENSE.TXT 文件更名为 A:/DOS/L.TXT。命令如下：

```
[globus@wang globus]$ mren A:/DOS/LICENSE.TXT A:/DOS/L.TXT
```

## 7. Mdel

删除文件。

例：删除 A:/DOS/L.TXT 文件。命令如下：

```
[globus@wang globus]$ mdel A:/DOS/L.TXT
```

使用 **mdir** 验证文件是否被删除，命令如下：

```
[globus@wang globus]$ mdir A:/DOS/L.TXT
```

```
mdir: File "A:/DOS/L.TXT" not found
```

没有找到文件 A:/DOS/L.TXT，因此说明文件已被删除。

## 8. mrd

删除子目录，子目录必须为空，否则无法删除。

例：删除 A: 盘里的 DOS 目录。命令如下：

```
[globus@wang globus]$ mrd A:/DOS
```

验证目录是否被删除，命令如下：

```
[globus@wang globus]$ mdir -w -f
```

```
Volume in drive A has no label
```

```
Directory for A:/
```

```
LFORMAT.EXE HELP.HLP diskgen.exe LICENSE.TXT UPGRADE.TXT
```

```
LOOKBACK.DAT
```

```
6 files 218 471 bytes
```

没有找到 DOS 子目录，因此表明目录已被删除。

## 9. mformat

格式化 DOS 盘，将格式化为 FAT，并产生根目录，**mformat** 有多个选项，鉴于目前很少用到这些选项，所以在此省略。有关它的参数，请参阅 **mformat** 的手册页（**man page**）。

格式 A: 盘，命令如下：

```
[globus@wang globus]$ mformat A:
```

如果成功，它将不给出任何提示信息。

使用 **mdir** 列目录，验证 A: 已经完成格式化。

```
[globus@wang globus]$ mdir A:
```

```
Volume in drive A has no label
```

```
Volume Serial Number is 4ED0-679B
```

```
Directory for A:/
```

```
No files
```

```
1 457 664 bytes free
```

## 10. Mlabel

**mlabel** 首先显示指定盘的当前卷标，然后提示输入新的卷标，当输入完成后，按回车

键结束。如果你没有输入任何信息，直接按回车键，那么它将会删除指定盘的卷标。

例：将 A: 盘的卷标设置成 DOS1。输入如下命令：

```
[globus@wang globus]$ mlabel A:
```

```
Volume has no label
```

```
Enter the new volume label : DOS1
```

现在的卷标已经改为 DOS1 了。可以使用 `mdir` 命令来验证：

```
[globus@wang globus]$ mdir
```

```
Volume in drive A is DOS1
```

```
Volume Serial Number is 4ED0-679B
```

```
Directory for A: /
```

```
No files
```

```
1 457 664 bytes free
```

从 `mdir` 的显示可知，A: 盘的卷标已经更改为 DOS1。

## 思考与练习

- (1) 如何使用 Windows 的超级终端连接到 Redhat Linux 9?
- (2) 建立目录、改变当前目录、删除子目录更名的命令分别是什么？使用时应该注意什么？
- (3) 显示文件的常见命令有哪些？它们的区别和联系是什么？
- (4) 文件和文件的访问权限有哪些？如何修改文件和目录的权限？

## 第 4 章 安装和使用汉字终端

Linux 是一个功能强大、性能出众、稳定可靠的操作系统，但 Linux 设计之初并未考虑到非英语国家使用者的需要，从而造成在 Linux 上使用中文的困难，并极大制约了 Linux 在国内的应用和普及。

Red Hat Linux 9 基于字符界面的控制台默认不支持中文的显示、输入和打印。虽然当前有众多的终端中文化的方案。笔者在此介绍安装和使用简便，功能也比较强大的 zhcon 支持 CJK 字符集的终端软件。

### 4.1 Linux 控制台中文文化

#### 4.1.1 Linux 控制台中文文化的现状

Linux 是一个功能强大、性能出众、稳定可靠的操作系统，但 Linux 设计之初并未考虑到非英语国家使用者的需要，从而造成在 Linux 上使用中文的困难，并极大制约了 Linux 在中国的应用和普及。随着 Linux 的普及，一些人已经在中文化课题上展开了研究，从早期的 chdrv、yact、wzce 和 cce，到后来的蓝点和 TurboLinux 汉化内核，许多自由软件的爱好者为 Linux 中文化作出了贡献。

基于字符界面的控制台本身并不支持点阵字符输出，要想使之支持中文，必须将控制台切换到图形模式并加入中文解码支持。这些控制台中文解决方案使用的技术不尽相同，基本上可以分为外挂式和内核汉化两种方式。外挂式解决方案使用伪终端技术来挂接中文平台，优点是不用重新编译内核，缺点是兼容性不太好，特别是控制台的 escape 处理（需要重写 VT Parser），相当困难。

现有的中文平台在易用性、扩展性、应用程序兼容性方面还存在着一些不足，zhcon 就是在这样的背景下诞生的。Zhcon 是一种高效的中/日/韩多内码控制台。zhcon 的目标是为中日/韩 Linux 使用者提供一个便捷易用的中文环境，特别适用于初学者，以及需要开发 linux 中文应用程序的开发者。

#### 4.1.2 Linux 控制台基本工作原理

通常，Linux 的控制台工作在文本模式下，要想在屏幕上正确显示，汉字必须将屏幕切换到图形模式，这可以通过调用内核 FrameBuffer 驱动程序来实现。此外，还要能正确识别系统输出到控制台的汉字信息，并调用汉字显示模块将其输出到屏幕。

一种方法是像 DOS 下的汉字系统所做的那样：利用系统时钟中断定时监视显存地址 B800:0000 处的显示缓冲区，动态识别缓冲区中的字符信息。这种方法要求修改内核中断和 TTY 驱动程序，实现起来比较困难，而且需要直接操纵硬件视频缓冲区，大大影响了系统的可移植性和稳定性。

另一种方法就是 UNIX 下多数中文平台采用的基于伪终端（Pseudo-Terminals）的外挂

式解决方案。伪终端（Pseudo-Terminal）是一种类似于终端的特殊的进程间通信通道（channel）。通道的一端被称为主设备（master pseudo-terminal device），另一端被称为从设备。写入主设备的数据被发送到从设备，而写入从设备的数据也可从主设备读出，对用户来说就好像自己实际上连接到了真正的计算机终端之上。简而言之，伪终端是位于虚拟终端和最终的终端设备之间的一种承担着输入输出转换功能的设备。

伪终端诞生之初就得到了广泛应用，典型的例子是 Telnet 服务程序。Telnet 是一种远程登录服务，用户使用 Telnet 客户端程序通过网络登录到远程主机之上进行各种操作。Telnet 服务程序就是一个伪终端，一端连接到 Telnet 客户端程序，另一端连接到主机应用程序，客户和应用程序之间通过伪终端进行对话。

当应用程序需要从输入设备（键盘）读入数据时，它向控制台设备（/dev/console）发出系统调用 read()，接着内核 console 驱动响应该系统请求，从输入设备驱动程序（通常为键盘驱动 keyboard.c）获得输入数据，并将之返回给应用程序。而当应用程序需要想控制台输出数据时，它通过 write() 系统调用将数据发送至 console 设备，再由内核 console 驱动转发到真正的输出设备（终端、打印机等）上去。

通过以上分析可以发现，如果在应用程序从控制台设备（/dev/console）读入数据之前截获键盘输入信息，并提交输入法模块处理，再将处理后得到的中英文信息发送至应用程序即可完成中文输入。而在应用程序将输出数据写入 /dev/console 设备之前截获输出并交由汉字识别模块处理，最终由汉字显示模块输出至屏幕，即可实现中英文输出。这就是 zhcon 之类的中文控制台的基本工作原理。

## 4.2 zhcon 简介

### 4.2.1 什么是 zhcon

zhcon 是工作在 Linux 控制台下的高效双字节中/日/韩（CJK）虚拟终端，就像 DOS 环境中的 UC DOS 一样，为控制台（console）环境提供完整的双字节语言环境。

### 4.2.2 zhcon 的开发工作

编写 zhcon 的初衷是为广大 Linux 爱好者提供一个便捷的控制台中文使用环境目前 zhcon 仍不十分完美，它还在不断改进之中。zhcon 是开放源代码的自由软件，在 GPL 版权下发行，对 Linux 中文开发有兴趣的任何人都可以加入开发队伍。

zhcon 的开发计划在 <http://sourceforge.net/projects/zhcon>，通过该页面可以得到更多的开发信息。

对开发有兴趣的用户可以加入开发邮件列表 zhcon-devel，用户遇到任何使用、开发方面的问题都可以直接发信到：zhcon-devel@lists.sourceforge.net。

### 4.2.3 zhcon 的特点

zhcon 具有如下特点：

- zhcon 不适用于 X Window 的终端，在 X Window 中可以选择 cterm 或 rxvt 作为

中文终端。

- zhcon 是外挂式平台。与内核汉化不同，不需要对 linux 的 kernel 打补丁。如同普通的 linux 软件，可以方便地直接安装 zhcon，在主控制台上执行 zhcon 后，就可以进入中文平台，也可以随时退出返回到西文状态。
- zhcon 是遵循 GPL 原则的自由软件。因为任何人都没有理由为使用自己的母语而付费。
- 广泛的图形驱动支持。支持 FrameBuffer/VGA/libggi 三种显示驱动方式，不重新编译内核就可以使用。
- 多语言环境。支持汉字 GB2312/GBK/BIG5，日文 JIS，韩文 KSCM 等多种双字节内码。支持 12~24 点阵的字体，充分发挥 17 吋大显示屏，1280×1024 高分辨率的威力。
- 多种输入法和输入风格。支持 Windows98 和 UCdos 中的码表输入法（自带 14 种），以及 TLC 的 unicon 输入法模块，与 Chinput 共享。
- 正确识别中文制表符，支持 Emacs 的多种光标类型，可用 linux 的标准热键浏览历史屏幕。

## 4.3 zhcon 的安装和使用

### 4.3.1 zhcon 的下载以及其运行环境

zhcon 的最新版本为 Version 0.2.3。使用 zhcon 虚拟控制台的用户可以从 [http://zhcon.sourceforge.net/develop\\_cn.html](http://zhcon.sourceforge.net/develop_cn.html) 网页下载 zhcon。对于 Red Hat Linux 9，用户可以使用 RPM 预编译包 zhcon-0.2.3-1.i386.rpm（网址为 <http://prdownloads.sourceforge.net/zhcon/zhcon-0.2.3-1.i386.rpm?download>）。

在 Linux 下可以使用如下命令下载 zhcon：

```
wget \
http://keihanna.dl.sourceforge.net/sourceforge/zhcon/zhcon-0.2.3-1.i386.
rpm
```

zhcon 的运行环境为：

- GNU/Linux (Redhat 7.x/8.x/9.x, Debian, Mandrake 8.x)
- FreeBSD

### 4.3.2 安装 zhcon

以根用户身份在 shell 提示符后输入如下命令，执行安装过程：

```
[root@unknown globus]# rpm -ivh zhcon-0.2.3-1.i386.rpm
[root@john globus]# rpm -ivh zhcon-0.2.3-1.rh9.i386.rpm
Preparing...      ##### [100%]
 1:zhcon          ##### [100%]
```



### 4.3.3 启动和退出 zhcon

在 shell 提示符中输入 zhcon，进入 zhcon 的汉字终端，界面如图 4-1 所示。

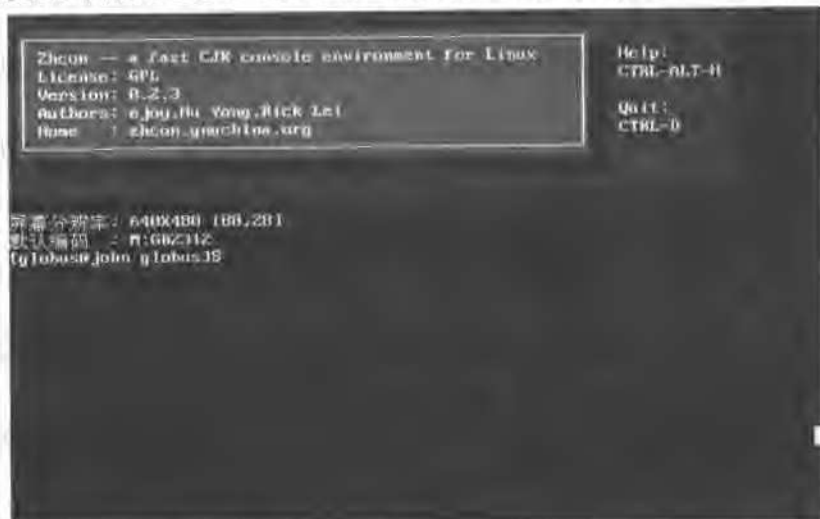


图 4-1 zhcon 启动后的界面

在 shell 提示符中输入 exit，退出 zhcon 的汉字终端环境，返回到默认的英文终端环境。

### 4.3.4 zhcon 的中文输入法

#### 1. 输入法开关

##### 左 Ctrl+空格键

说明：按此组合键来开关输入法，当关闭输入法时，输入状态条同时关闭，输入状态恢复为英文键盘状态。再按此键则打开状态条。当需要长期输入英文文本，或者用户的应用程序和输入法热键冲突时，也可以按此键暂时关闭输入法。

#### 2. 中英文输入状态的切换

##### Shift+空格键

说明：反复按此键可以在中英文之间切换。在频繁切换中英文输入时，这是非常有效的。

注意：如果您打开了全角方式，则此切换是在中文和全角字符之间切换，如图 4-2 所示。

【半角..】全拼 jiang      1.将 2.讲 3.江 4.奖 5.降 6.浆 7.僵 8.姜

图 4-2 zhcon 的输入法提示条

#### 3. 切换中文输入法

在中文输入状态下，按 Ctrl+Shift 键切换输入法，zhcon 包含的主要输入法如下：

- 全拼
- 五笔
- 简拼
- 双拼
- 注音
- 五笔划

切换到“五笔”输入法的输入条，如图 4-3 所示。

【半角.,】五笔 ugü 1.姜 2.美好b 3.姜f 4.美妙i 5.凄v

图 4-3 切换到五笔输入法

#### 4. 全角、半角切换

##### Ctrl+, (逗号键)

可见提示条上最左边的提示由半角变成了全角，如图 4-4 所示。

【全角.,】全拼

图 4-4 切换到全角方式

#### 5. 中英文标点的切换

##### Ctrl+ (句号键)

说明：按此键可以在中英文标点之间切换。在西文输入状态时，标点状态在输入条上显示的标点分别为“.,”。在中文输入状态时，输入法的标点显示为“。,”，其中“。”表示当前为中文标点状态，如图 4-5 所示。

注意：在中文和英文输入状态下，标点状态的设置是分开的。例如，在中文输入时，标点默认可以设置为中文标点，英文输入状态时，标点可以设置为英文标点，而不用反复进行切换。

【全角。】全拼

图 4-5 中文标点方式

#### 6. 特殊功能设定

##### 左 Ctrl + F10 (功能键)

说明：按此键进入系统菜单，用来进行中文系统设定。

#### 7. 清除当前状态条上的候选字（词）输入

##### 按 ESC 键

说明：按此键将清除当前输入条的待选字（词）。

注意：仅当输入条存在候选字时，此键才有效，否则 ESC 键将被直接送到应用程序。

#### 8. 其他功能键

- +/- 候选字前后翻页。
- Alt+Space 打开/关闭 CJK 方式。

- Shift+Pageup/Pagedown/Up/Down 浏览历史窗口。
- Ctrl+F1 GB2312 编码。
- Ctrl+F2 GBK 编码。
- Ctrl+F3 BIG5 编码。
- Ctrl+F4 JIS 编码。
- Ctrl+F5 KSC 编码。
- Ctrl+F9 切换自动编码方式。
- Ctrl+F10 菜单。
- Ctrl+D 退出 zhcon。

#### 4.3.5 使用 zhcon 的常见问题

使用 zhcon 的常见问题如下：

- 如何每次 login 完后自动运行 zhcon?

在.bashrc 文件中，加上:zhcon >/dev/null 2>&1

- 非 root 如何使用 zhcon?

zhcon 是个 suid 程序，需要有 root 权限才能运行，非 root 用户需要以 root 身份做以下工作：

```
chmod 4775 /usr/local/bin/zhcon
```

或

```
chmod +s /usr/local/bin/zhcon
```

不过 Red Hat Linux 9.0 的 RPM 包在安装时已经自动做这个工作。

- 如何在 zhcon 中使用 unicon/chinput 的输入法?

zhcon 内置的输入法只提供基本的输入功能，不过 zhcon 提供了对 TurboLinux 的 unicon 输入法模块的支持，可以很方便地在 zhcon 中使用 unicon(chinput)的输入法。你必须使用源码编译后再安装。

unicon 的输入法是由 libimclient.a 和 libim\_server.so 两个库组成。在安装 unicon 的时候还要安装 GNU pth。

目前，configure 脚本只能自动检测这两个文件被安装到/usr/lib 下的情况，如果您的 unicon 中这两个文件是在其他目录，需要拷贝到/usr/lib。再 ldconfig /usr/lib（当然如果在/usr/local/lib/下也可以（确认/etc/ld.so.conf 中有/usr/local/lib 一行，如果没有，请加上再 ldconfig））总之，只要这两个文件在系统运行时可以找到的地方就行。

然后 rm config.cache，再使用 ./configure，如果没有问题的话就会提示 unicon support: yes。

再 make;make install 后修改/etc/zhcon.conf 中 uniconpath 为 unicon 输入法表路径：

```
uniconpath=/usr/local/unicon
```

或

```
uniconpath=/usr/unicon
```

然后运行 zhcon，就可以使用 unicon 中的输入法了。

- 成功安装了 zhcon，如果 zhcon 运行得很慢，如何解决？  
zhcon 支持 3 种不同的显示驱动：**framebuffer**（最快），VGA（慢）和 libggi（慢）。在启动 zhcon 的时候，它会自动选择最合适的显示驱动。如果当前内核不支持 framebuffer 设备，zhcon 将会使用 VGA 驱动，这时 zhcon 工作在 640×480×16 色模式下，运行起来会比较慢。  
为了达到最好的显示效果，建议使用 framebuffer 驱动并工作在（8bit）256 色的模式下。你会发现运行速度可以和西文显示相差无几。

## 思考与练习

- （1）在 Redhat Linux 9 中，为什么要汉化终端？
- （2）如何安装 zhcon？
- （3）zhcon 有哪些功能？
- （4）zhcon 有哪些快捷键？它们的作用是什么？

## 第 5 章 全屏幕编辑器 vi

在编写文本或者计算机程序时，需要创建文件、插入新行、重排新行、修改内容以及作一些必要的其他修改。Linux 可视化编辑器 vi 是一个功能强大的、快速命令驱动的全屏幕编辑器，它通过按组合键向编辑器发送所有指令。

本章将介绍使用可视化编辑器访问文件、将光标移到文本中的各种位置，进行编辑、修改、替代和删除操作，以及读入其他文件和各个部分的重排文档。

### 5.1 全屏幕编辑器 vi 简介

#### 5.1.1 Linux 编辑器简介

编辑器是使用计算机的重要工具之一，在各种操作系统中，编辑器都是必不可少的部件。Unix 及其相似的操作系统系列中，为方便各种用户在各个不同的环境中使用，提供了一系列的 ex 编辑器，包括 ex, edit, ed 和 vi。其中 ex, edit, ed 都是行编辑器，现在已很少有人使用，Unix 提供它们的原因是考虑到满足各种用户特别是某些终端用户的需要。

值得庆幸的是，Unix 提供了全屏幕 Vi 编辑器，这使用户的工作更轻松。不少 DOS 用户抱怨 vi 编辑器不象 DOS 下的编辑器（如 edit）那么好用，这是因为 vi 考虑到各种用户的需要，没有使用某些通用的编辑键（在各个不同的终端机上它们的定义是不同的，在某些终端机上甚至没有这些键），而是采用状态切换的方法，但这只是习惯的问题，一旦你熟练了 vi，你就会觉得它其实也很好用。

虽然 vi 采用了状态切换的方法，但计算机硬件和操作系统多种多样，某些计算机的键盘上没有特定的某些功能键。这样就可能使一些程序不能在所有的计算机上运行。在 Unix 系统上也存在同样的问题，几乎各大计算机厂商都有自己的 Unix 系统，而 vi 的操作方法也会随之有点出入。

**注意：**本书采用 PC 的键盘来说明 vi 的操作，但在其他系统环境中还需要参考相应的资料。

vi 是 Visual interface 的简称，它在 Linux 上的地位就仿佛 Edit 程序在 DOS 上的地位一样。它可以执行输出、删除、查找、替换、块操作等众多文本操作，而且用户可以根据自己的需要对其进行定制，这是其他编辑程序所没有的。Vi 不是一个排版软件，它不象 Word 或 WPS 那样可以对字体、格式、段落等其他属性进行编排，它只是一个文本编辑程序。

vi 作为 Linux 系统中一种编辑器，它的使用方法和界面与 Unix 平台十分相似。在 Linux 中有很多功能强大，界面友好的编辑器，但是 vi 还是值得用户学习它的使用方法。初步接触 vi，以后会感觉它的界面不太友好，不容易掌握，但是一旦你掌握了 vi 的命令，你会逐渐的感觉到它强大的功能与很高的效率。vi 相对来说较小，用户使用任何 Linux 系统，总是可以使用 vi。在许多系统中，可能只有 vi 供你选择。

广泛使用的 vi 有两个版本。它们是标准的 UNIX 的版本 vi 和 Linux 提供的 vim。vim

与 vi 兼容，同时也对 vi 进行了一些扩充，在 Red Hat Linux 9 中使用的是 vim，vi 是 vim 的一个符号链接。

### 5.1.2 进入和退出 vi 编辑器

#### 1. 进入 vi

要进入 vi 可以直接在系统提示字符下键入：

vi 文件名称

例如：打开当前目录中的 diff1.txt 文件。命令如下：

```
[globus@wang globus]$ vi diff1.txt
```

屏幕如下：

```
He is good sense of humor.
```

```
He tells good stories.
```

```
King is virtually inEscapable.
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
"diff1.txt" [已转换] 4L, 110C
```

```
4,1
```

```
全部
```

进入 vi 之后，首先进入的就是命令模式，也就是说等待命令输入而不是文本输入。这时输入的字母都将作为命令来解释。光标停在屏幕第一行首字符上(用\_表示)。由于 diff1.txt 文件已在系统中存在，则输入上述命令后，在屏幕上显示出该文件的内容，并且光标停在第一行的第一列，在状态行显示出该文件的文件名、行数和字符数，此例中表示打开的文件为 diff1.txt，4 行 (4L)，共 110 个字符 (110C)，其余各行行首均有一个“~”(波浪符号)符号，表示该行为空行。如果你打开的文件为空文件，各行行首均是一个“~”，最后一行也称状态行，显示出当前正在编辑的文件名以及其状态。

**注意：**在这里为了节省篇幅起见，屏幕只显示 9 行。事实上，初始的显示行数与用户所用终端有关，一般的 CRT 终端可显示 25 行。在窗口系统中，显示行数与运行 vi 的那个窗口有关。然而用户还可以对显示行数进行设置。

当用 vi 建立一个新文件时，在进入 vi 的命令中也可以不给出文件名，当编辑完文件需要保存数据时，再由用户指定文件名。

#### 2. 退出 vi

要离开 vi 可以在指令模式下键入:q, :wq 指令则是存盘后再退出(注意有冒号)。要切换到指令模式下则是用 Esc 键，如果用户不知道现在是处于什么模式，你可以多按几次 Esc 键，系统会发出响铃以确认已经进入指令模式。

vi 存在两种模式：指令模式和输入模式。在指令模式下输入的按键将作为指令来处理，

如输入 `a`, `vi` 即认为是在当前位置插入字符。在输入模式下, `vi` 则把输入的按键当作插入的字符来处理。指令模式切换到输入模式只需键入相应的输入命令即可 (如 `a,A`), 而要从输入模式切换到指令模式, 则需在输入模式下键入 `Esc` 键, 如果不知道现在是处于什么模式, 可以多按几次按 `Esc` 键, 系统会发出嘟嘟声, 表示 `vi` 已处于指令模式下了。

### 5.1.3 进入输入模式并进行删除和修改操作

#### 1. 切换到 `vi` 的输入模式

用户有多种方法进入输入模式:

- 新增 (`append`)

按 `a` 键。从光标所在位置后面开始输入新文本, 光标后的文本随新增文本向后移动。

按 `A` 键。从光标所在行的行尾开始输入新的文本。

- 插入 (`insert`)

按 `i` 键。从光标所在位置前面开始插入数据, 光标后的数据随新增文本向后移动。

按 `I` 键。从光标行的第一个非空格字符前面开始插入文本。

- 开始 (`open`)

按 `o` 键。在光标所在行下新增一行并进入输入模式。

按 `O` 键。在光标所在行上方新增一行并进入输入模式。

#### 2. 删除与修改

编辑是文字的新增修改、删除, 甚至包括文字区块的搬移、复制等等。在此首先介绍如何在 `vi` 中进行文本的删除与修改。

**注意:** 在 `vi` 的原始观念里, 输入和编辑是两回事。编辑是在指令模式下操作的, 先利用指令移动光标来定位要进行编辑的地方, 然后才下指令模式下进行编辑。

在命令模式下, `vi` 的删除和修改命令如下:

- 按 `x` 键。删除光标处的一个字符。

- 按 `dd` 键。删除光标所在的行。

- 按 `r` 键。修改光标所在字符, `r` 后紧接着要修正的字符。

- 按 `R` 键。进入替代状态, 输入的文本会覆盖原来的数据, 直到按 `Esc` 键回到指令模式下为止。

- 按 `s` 键。删除光标所在字符, 并进入输入模式。

- 按 `S` 键。删除光标所在的行, 并进入输入模式。

### 5.1.4 简单的光标移动

在 `vi` 的命令模式里, 用户可以使用几个简单的光标移动键。可以使用方向键在文本中移动光标, 以便对光标处的文本进行编辑。

常见的光标移动键如表 5-1 所示。

表 5-1 vi 命令模式中常见的光标移动键

光标移动键	解释
h (或者向左的方向键)	向左移动一列
j (或者向下的方向键)	向下移动一行
k (或者向上的方向键)	向上移动一行
l (或者向右的方向键)	向右移动一行

## 5.2 在文件中快速移动光标

前一节中介绍使用方向键在文件中逐行和逐列移动光标。本节将介绍一些功能更强的光标移动命令，这样会极大地提高编辑的速度和效率。

### 5.2.1 方向键的扩充

在前一节使用方向键的基础上，在这些方向键的前面加入一些数字，可以加快光标的移动。

(1) 例如：输入下列命令，将文件/etc/passwd 拷贝到用户的主目录。取名为 vitest。

```
[globus@wang globus]$ cp /etc/passwd ~/vittest
```

(2) 用 vi 打开 vitest 文件，准备进行编辑：

```
[globus@wang globus]$ vi ~/vittest
```

将光标向下移动 5 行，输入：

5j 或者 5↓

注意：5j 表示输入 5 之后，再次输入 j。而 5↓ 表示输入 5 之后，再按向下的方向键。

当输入完成之后，不需要按回车键，即可立即生效。

你可以使用类似的方法，快速移动光标，举例如表 5-2 所示。

表 5-2 vi 命令模式中扩展的光标移动举例

光标移动键	解释
3j	向下移动3行
6k	向上移动6行
5→	向右移动5列
2←	向左移动2列
10l	向右移动10列
3h	向左移动3列
10↓	向下移动10行

### 5.2.2 查找指定的字符串

用户可以在 vi 中查找指定的字符串（它不仅仅定位单词）。查找方向可以是向前查找（从文件头到文件尾的方向），也可以是向后查找（从文件尾到文件头的方向）。

例如：向前查找第一个字符串 roo，在命令模式中输入：



/root

输入完成后，需要再按回车键。它可以找到 root 等单词。

用户也可以使用向前查找命令查找包含有空格在内字符串。

例：查找字符串 Jiang Zhonghu。命令如下：

/Jiang Zhonghu

输入完成后，需要再按回车键。如果系统上存在一个用户，他的全名为 Jiang Zhonghua，则上述的查找命令将能找到 Jiang Zhonghua 等单词。

### 5.2.3 从双向继续查找指定的字符串

输入 “/:” 命令，将从向前查找第一个字符 “:”，你也可以使用下述命令向后查找第一个字符 “:”：

?:

输入下列命令将继续查找刚才查找的单词：

N —— 沿当前查找相反方向查找下一个单词

n

n —— 沿当前查找相同方向查找下一个单词

N

小写字母 n 将光标沿着当前的查找方向查找下一个字符串，大写字母 N 将沿着与当前查找方向相反的方向查找下一个字符串。

注意：/string 指示编辑器向前查找字符串 string；?string 指示编辑器向后查找字符串 string。无论你当前沿着那个方向查找，使用 n 命令将光标移动到同一方向的下一个目标，使用 N 命令将沿着相反方向查找下一个目标。

### 5.2.4 以单词为单位移动光标

到目前为止，用户已经可以使用 h、j、k、l 或者光标键，或者在前面加一些数字，来移动光标键。除此之外，用户也可在文件中向前或者向后每次以一个单词位为单位移动光标。

以单词为单位的光标移动命令如表 5-3 所示。

表 5-3 vi 的快速光标移动命令及其解释

命令	描述
w	将光标移到下一个单词的开头（单词是由空格和标点符号分隔的）
b	将光标向后移动一个单词的开头。它的功能与w命令相反
e	将光标移到下一个单词的末尾（最后一个字符处）
4w	将光标移动到下一个单词的开头
5b	将光标向后移动五个单词，并定位到第五个单词的开头。它的功能与5w命令相反
2e	将光标移动到下一个单词，并定位到单词的最后一个字符

注意：nw、nb 和 ne 指将光标分别向前或者向后移动 n 个单词。w、b 和 e 分别等价于

lw、lb 和 le。

### 5.2.5 行内的快速光标移动

可以将光标移到当前行的指定位置。命令如表 5-4 所示。

表 5-4 在行内的光标移动

命令	描述
\$	将光标移动到当前行的行末
^或者0	将光标移动到当前行的行首
n	将光标移动到当前行的第n个字符上
fm	将光标移动到文本的下一个字符m

例如：打开主目录中的文件 vitest。

执行如下命令：

```
10j      —— 向下移动10行
13|      —— 将光标移动到当前行的第13个字符
10|      —— 将光标移动到当前行的第10个字符
fu       —— 将光标移动到下一个字符u
$        —— 将光标移动到当前行的最后一个字符
0        —— 将光标移动到当前行的第一个字符
```

### 5.2.6 使用行号定位行

当 vi 打开一个文件时，它对文件中的所有行从 1 开始编号，并且 vi 提供了使用行号定位行的命令。

常用的行号定位命令如表 5-5 所示。

表 5-5 vi 的行号定位行的命令

命令	描述
nG	将光标移动到第n行
G	将光标移动到最后一行
:n回车	将光标移动到第n行
:\$回车	将光标移动到文件的最后一行
:set number回车	使编辑器显示行号
:set nonumber回车	使编辑器不显示行号
-	光标移到当前行的上1行
+	光标移到当前行的下1行

注意：凡是以冒号(:)开头的命令必须使用回车键结束，比如“:\$回车”表示首先输入冒号(:)，然后输入美元符号(\$)，最后按回车键，对于冒号(:)开头的命令，在没有按回车键之前，命令不会执行，因此用户可以使用退格键(BACKSPACE 键)删除输入错误。对于其他的 vi 命令，则不需要输入回车键，一旦输入，立即执行。

为了表述方便,在以后的命令中,如果冒号开头的命令,在表示时将省掉回车键。

: $\$$ 和 $\$$ 两条命令的功能是完全不同的。 $\$$ 表示将光标移动到当前行的行尾,而: $\$$ 则表示将光标移动到文件的最后1行。

例如:打开~/vittest 文件。命令序列如下:

```
10G      —— 将光标移到第10行
G        —— 将光标移到文件的最后一行
:8回车   —— 将光标移到第8行
+        —— 将光标下移1行
-        —— 将光标上移1行
:set number —— vi编辑器显示文件的行号
```

使用:set number 命令后,vi 在每一行的开头显示相对于文件开头的行号,文件的第1行的编号为1,效果如图5-1所示。

```
1 root:x:0:0:root:/root:/bin/bash
2 bin:x:1:1:bin:/bin:/sbin/nologin
3 daemon:x:2:2:daemon:/sbin:/sbin/nologin
4 adm:x:3:4:adm:/var/adm:/sbin/nologin
5 lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
6 sync:x:5:0:sync:/sbin:/bin/sync
7 shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
8 halt:x:7:0:halt:/sbin:/sbin/halt
9 mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
10 news:x:9:13:news:/etc/news:
11 uucp:x:10:14:uucp:/var/spool/uucp:/sbin/nologin
12 operator:x:11:0:operator:/root:/sbin/nologin
13 games:x:12:100:games:/usr/games:/sbin/nologin
14 gopher:x:13:30:gopher:/var/gopher:/sbin/nologin
15 ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
16 nobody:x:99:99:Nobody:/:/sbin/nologin
17 rpm:x:37:37:/var/lib/rpm:/bin/bash
18 vesa:x:69:69:virtual console memory owner:/dev:/sbin/nologin
19 nscd:x:20:20:NSCD Daemon:/:/sbin/nologin
20 sshd:x:74:74:Privilege-separated SSH:/var/empty/ssh:/sbin/nologin
21 rpc:x:32:32:Portmapper RPC user:/:/sbin/nologin
22 rpcuser:x:29:29:RPC Service User:/var/lib/rpc:/sbin/nologin
23 nfsnobody:x:65534:65534:nomymous NFS User:/var/lib/nfs:/sbin/nologin
24 mailnull:x:47:47:/var/spool/queue:/sbin/nologin
25 /bin/bash: line 1: indent: command not found
```

图 5-1 vi 显示行号后的外观

### 5.2.7 将光标移动到当前屏幕的某些位置

vi 也提供了将光标移动到屏幕指定位置的命令,如表5-6所示。

表 5-6 vi 的屏幕定位命令

命令	描述
L	将光标移动到屏幕的最下面1行
M	将光标移动到屏幕的中间1行
H	将光标移动到屏幕的最上面1行

### 5.2.8 将光标返回到原来位置

在编辑文件的过程中，经常需要在文件中将光标从当前位置移到很远的地方，进行必要的操作之后，再返回到操作之前的位置。

“命令将光标返回到文件原来的位置，而不管光标在什么位置，即使在新位置对文件做了修改。

注意：“不是一个双引号，而是两个单引号。

### 5.2.9 让屏幕显示文件的指定位置

光标移到命令虽然能够光标重定位到文件中的一个新位置，但是屏幕显示页随之改变，这些命令是基于行和文本的。用户也可以在文件中向前或者向后移动屏幕显示内容，而迫使光标跟着移动。

在命令模式中，屏幕切换命令有 Ctrl+d、Ctrl+u、Ctrl+f 和 Ctrl+b，每一个方向上有两条显示调整命令。表 5-7 中的命令用于调整工作站的屏幕显示，向前或者向后移动到不同的文本段或者文本块，而不管它的前后关联关系。

表 5-7 屏幕定位命令的功能

命令	描述
Ctrl+d	屏幕定位到文件的下半屏文本
Ctrl+u	屏幕定位到文件的上半屏文本
Ctrl+f	屏幕定位到文件的下一屏文本
Ctrl+b	屏幕定位到文件的上一屏文本

### 5.2.10 光标和屏幕定位小结

本小节介绍了在文件下定位的两种方式：

- 使光标定位到文件的一个特定位置，光标的移动可能会导致屏幕的重定位。
- 屏幕重定位，可能使光标也随之重定位。

表 5-8 列出了命令模式中移动光标的命令。

表 5-8 命令模式定位命令总结

命令	描述
h	向左移动一个字符
j	向上移动一行
k	向下移动一行
l	向右移动一个字符
w	将光标移到下一个单词的开头（单词是由空格和标点符号分隔的）
b	将光标向后移动一个单词的开头。它的功能与w命令相反
e	将光标移到下一个单词的末尾（最后一个字符处）
nw	将光标移动到下一个单词的开头

续表

命令	描述
<code>nb</code>	将光标向后移动n个单词，并定位到第n个单词的开头。它的功能与 <code>nw</code> 命令相反
<code>2e</code>	将光标移动到下一个单词，并定位到单词的最后一个字符
<code>\$</code>	将光标移动到当前行的行末
<code>^</code> 或者 <code>0</code>	将光标移动到当前行的行首
<code>n </code>	将光标移动到当前行的第n个字符上
<code>fm</code>	将光标移动到文本的下一个字符m
<code>nG</code>	将光标移动到第n行
<code>G</code>	将光标移动到最后一行
<code>:n</code> 回车	将光标移动到第n行
<code>:\$</code> 回车	将光标移动到文件的最后一行
<code>:set number</code> 回车	使编辑器显示行号
<code>:set nonumber</code> 回车	使编辑器不显示行号
<code>-</code>	光标移到当前行的上一行
<code>+</code>	光标移到当前行的下一行
<code>L</code>	将光标移动到屏幕的最下面一行
<code>M</code>	将光标移动到屏幕的中间1行
<code>H</code>	将光标移动到屏幕的最上面一行
<code>Ctrl+d</code>	屏幕定位到文件的下半屏文本
<code>Ctrl+u</code>	屏幕定位到文件的上半屏文本
<code>Ctrl+f</code>	屏幕定位到文件的下一屏文本
<code>Ctrl+b</code>	屏幕定位到文件的上一屏文本

## 5.3 vi 的文本删除

### 5.3.1 删除字符

在 vi 的命令模式中，可以使用 `x` 命令来删除单个字符。也可以使用数字来扩充 `x` 命令，可以删除光标处的多个字符。

`u` 命令可以撤销前一次的删除操作。

例如：输入

<code>x</code>	—— 删除光标处的一个字符
<code>4w</code>	—— 光标移到下4个单词的第一个字符处
<code>7x</code>	—— 删除从光标开始的7个字符
<code>w</code>	—— 光标移到下一个单词的位置
<code>3x</code>	—— 删除从光标开始的3个字符
<code>u</code>	—— 撤销最后删除的3个字符
<code>u</code>	—— 撤销使用 <code>7x</code> 命令删除的7个字符

### 5.3.2 删除行

在 vi 的命令模式中, 可以使用 **dd** 命令来删除当前行。也可以使用数字来扩充 **dd** 命令, 可以删除从当前行开始的多行文字。

**u** 命令可以撤销前一次的删除操作。

- dd**                   —— 删除当前行, 光标移到下一行, 并成为当前行
- 10dd**               —— 删除当前行开始的10行, 光标移到下一行, 并成为当前行
- u**                   —— 撤销最后一次删除行的操作
- u**                   —— 撤销倒数第二次删除行的操作

### 5.3.3 删除单词

vi 也提供了单词删除命令, 这在文件编辑和程序设计中是非常有用的。用户可以使用 **dw** 命令删除光标处的一个单词, 也可以使用数字来扩充 **dw** 命令, 可以删除从当前光标开始的多个单词。

例: 输入下列 vi 命令。

- 3w**                   —— 将光标移动到下一个单词的位置
- dw**                  —— 删除光标处的1个单词
- 4dw**               —— 删除从光标开始的4个单词

注意: 如果将光标定位一个单词的中间某个字符上, 使用 **dw** 命令之后, vi 将删除从光标开始和空格字符结束的所有字符。

在 vi 中, 所有除空格以外的非字母数字字符都将视为一个单词。

**dw** 命令具有跨行删除功能, 也就是说, 如果当前行的当前光标到行末的单词数不够删除时, vi 将删除下一行的单词。

### 5.3.4 删除一行的一部分内容

除了删除一整行、一个单词和一个字符, vi 还可以删除一行中的一部分。如果将光标置于一行的中间, 输入 **D** 命令, 则 vi 将删除从光标开始到行末的所有文字。

### 5.3.5 删除光标至某一个字符之间的文本

由上一节的内容可知, 使用 **f<sub>m</sub>** 命令可以将光标从当前位置向后移动到字符 **m** 处, 并成为新的光标位置。

vi 提供了 **df<sub>m</sub>** 命令, 它指示 vi 删除从光标到字符 **m** 之间的所有文本。在此的 **m** 可以为任意字符。

例: 在 **vitest** 的第 10 行, 删除从第一个冒号字符到字符 **/** 之间的所有字符。然后在撤销被删除的内容。

首先将光标定位到第 10 行:

```
10G
```

然后, 将光标定位到当前行的第一个冒号 (:) 处:

```
f:
```

删除从光标到第一个字符/的所有字符：

`df/`

最后，在撤销被删除的内容。

`u`

5.3.6 使用行号删除文本行

从前面的知识可知，可以将光标移动到特定的行（如 5G、:10 和:\$）分别将光标移至第 5 行、第 10 行和文件的最后一行。

除了定位光标以外，用户还可以使用行号进行删除操作。

例：利用行号的删除操作。

删除第 5 行的文本：

`:5d`

删除第 5~10 行的文本：

`:5,10d`

删除从光标开始的 3 行内容：

`3dd`

删除从第 10 行开始到行末的内容：

`:10,$d`

撤销最后两次的删除操作：

`u`

`u`

关于删除命令的总结如表 5-9 所示。

表 5-9 删除命令小结

命令	描述
x	删除字符
dw	删除单词
dd	删除行
D	删除从光标到行末的所有字符
dfm	删除从光标开始到第一个字符m之间的所有文本

5.4 vi 的文本添加

vi 编辑器提供了多条文本添加命令，在实际使用时，用户需要根据在文件中添加文本的位置，文本的长度以及当前光标的位置共同决定选用哪一条命令。

5.4.1 在光标的左侧添加文本

在命令模式中，使用 `a` 命令将使用户在光标的右侧添加文本。使用 `i` 命令在光标的左侧添加文本。当使用了 `a` 或者是 `i` 命令后，vi 将由命令模式切换到编辑模式。

例如：文件 temp.txt 有 4 行文本。内容如下：

```
The commands to paste are p and P.
The differ in the position relative to the cursor where they paste.
p pastes the specified or general buffer after the cursor position.
Specifying count before the paste command pastes text the specified number
of times.
```

使用 vi 打开该文件之后，进行了一系列的光标移动，光标的当前位置在第二行单词 differ 的字母 d 处。

如果输入 i 命令，并输入 only，则使用 i 命令后文件的内容将变成：

```
The commands to paste are p and P.
The only differ in the position relative to the cursor where they paste.
p pastes the specified or general buffer after the cursor position.
Specifying count before the paste command pastes text the specified number
of times.
```

而且光标仍处于字母 d 处。

如果对源文件使用 a 命令，重复上述操作，结果如下：

```
The commands to paste are p and P.
The only affer in the position relative to the cursor where they paste.
p pastes the specified or general buffer after the cursor position.
Specifying count before the paste command pastes text the specified number
of times.
```

按 Esc 键返回命令模式。

注意：a 命令和 i 命令使用户从命令模式切换到插入模式。在 i 命令后面输入的任何字符都作为文本从光标的左侧插入到文件，而 a 命令则将这些文件从光标的右侧追加到文件。

#### 5.4.2 在当前行的上面或者下面插入新行

O 命令在光标所在行的上面插入一个新行，而 o 命令则从光标所在行的下面插入一个新行。这两条命令都将光标移动到插入的新行的行首。

例如：文件 temp.txt 有 4 行文本。内容如下：

```
The commands to paste are p and P.
The differ in the position relative to the cursor where they paste.
p pastes the specified or general buffer after the cursor position.
Specifying count before the paste command pastes text the specified number
of times.
```

当前光标的位置在第二行的 differ 单词的字母 d 处。

输入命令 O，将在当前行的上面插入一个新行。

然后输入如下一行文本：

```
The VI editor has features to help programmers format their code neatly.
```



结果如下:

The commands to paste are p and P.

The VI editor has features to help programmers format their code neatly.

The differ in the position relative to the cursor where they paste.

p pastes the specified or general buffer after the cursor position.

Specifying count before the paste command pastes text the specified number of times.

如果是输入 d 命令,它将在当前行的下面插入一个新行。

重复上述的输入操作,其结果为:

The commands to paste are p and P.

The differ in the position relative to the cursor where they paste.

The VI editor has features to help programmers format their code neatly.

p pastes the specified or general buffer after the cursor position.

Specifying count before the paste command pastes text the specified number of times.

按 Esc 键返回命令模式。

输入下述命令,保存文件,并返回 shell:

:x

注意: :x 命令、:wq 命令和 ZZ 命令三者的功能相同,都是保存文件后,退出 vi。

### 5.4.3 在行首和行末添加文本

用户如果希望在行首和行末添加文本时,光标的当前位置可能在文件中间的某一个位置。用户往往需要多个命令才能在行首或者行末添加文本。

I 命令将光标移到当前行的行首,并切换到添加模式。A 命令将光标移到当前行的行尾,并切换到添加模式。

当用户需要在行首添加文本时,按下述步骤进行:

输入行首插入命令:

I

输入要添加的文本,可能是多行:

Show the current filename and the status.

Clear and redraw the screen.

Redraw the screen removing false lines.

Escape key. Cancels partially formed command

按 Esc 键返回到命令模式。

### 5.4.4 添加命令小结

本节讲述了 vi 提供的在 4 个方向上的文本添加命令。对这 4 个命令的总结如表 5-10 所示。

表 5-10 文本添加命令小结

命令	描述
i	在光标的左侧添加文本
a	在光标的右侧添加文本
I	将光标移到当前行的行首，并切换到添加模式
A	将光标移到当前行的行尾，并切换到添加模式
O	在光标的上方插入一个新行
o	在光标的下方插入一个新行

## 5.5 vi 的撤销和重复命令

### 5.5.1 多次撤销命令

可视化的编辑器 vi 不仅具有强大的文字编辑和光标移动功能，而且还提供了强有力的重复和撤销功能。这两条命令的作用是撤销最后一次或者倒数第几次的操作，或者重复最后一次操作，因此它们依赖于你最后所作的部分操作。

注意：对于 vi 编辑器来说，它只能撤销最后一次操作，而 vim 是 vi 的增强，因此它可以进行多次撤销，这与微软字处理软件 Word 的撤销功能是相似的。在 Linux 中使用的是 vim，因此它支持多次撤销。

使用 u 命令撤销最后一次更改操作。

例：打开主目录中的 vitest 文件。然后输入以下命令：

```
10G      —— 光标移动第10行
dd       —— 删除当前行
:13      —— 光标移动第13行
2dd      —— 删除当前，以及当前行的下一行
:20,$d   —— 删除20行及其之后的所有行
dd       —— 删除当前行
dd       —— 删除当前行
```

执行如下撤销命令：

```
u        —— 撤销最后一次删除的一行
u        —— 撤销倒数第二次删除的一行
u        —— 撤销倒数删除第20行以后的内容
u        —— 撤销删除的两行
u        —— 撤销第一次删除的一行
u        —— 不起作用，因此所有的删除都已撤销
```

### 5.5.2 撤销一行上的所有修改

U 命令可以撤销对当前行进行的所有修改。

例：打开主目录中的 `vitest` 文件。然后输入以下命令：

```
10G      —— 光标移动第10行
w        —— 光标移到下一个单词
w        —— 光标移到下一个单词
x        —— 删除光标处的一个字符
dw       —— 删除光标处的一个单词
U        —— 对10行的所有修改
j        —— 光标移到下一行
3dw      —— 删除光标处的3个单词
x        —— 删除光标处的一个字符
x        —— 删除光标处的一个字符
u        —— 撤销最后一次删除的一个字符
U        —— 撤销倒数第二次删除的一字符，以及
           一次删除的3个单词
```

`vi` 的撤销命令如表 5-11 所示。

表 5-11 撤销命令小结

命令	描述
<code>u</code>	撤销最后一次修改
<code>U</code>	撤销对当前行的所有修改，前提是光标没有移出当前行

### 5.5.3 重复最后一次文本修改命令

用户经常要对一个文件的多个地方进行相同的修改。`vi` 提供的重复命令能使最后的命令重复执行。

`vi` 提供的重复命令是：`.`（英文的句点）。它指示 `vi` 编辑器重复执行最后一次文本修改命令（任何文本修改命令）。

例：由于文件中的每一行都是顶格输入文字的，要求在每一行的开头空出四个空格字符。命令序列如下：

打开所需修改的文件，如 `vitest`：

```
[globus@unknown globus]$ vi vitest
```

进入 `vi` 环境。

转入第一行：

```
1G
```

切换至插入模式：

```
I
```

然后按 4 次空格键，即输入了 4 个空格。

按 `Esc` 键返回到命令模式。

转入下一行：

```
j
```

重复执行 (.) 在行首插入 4 个空格的命令:

依次使用 j 命令, 转到下一行, 然后执行重做命令 (.), 以便插入 4 个空格。直到文件尾。

## 5.6 vi 的文本替换

任何编辑工作都可以使用添加和删除文本来完成, 但是这并不是高效的方法。为了提高效率, 避免用户的疏忽, 可以使用 vi 提供的文本替换功能, 该功能包括: 替换字符、单词和行。

### 5.6.1 替换字符

用户可以在不切换到编辑模式的情况下, 删除一个字符并用另一个字符替换它。

在 vi 的命令模式下, 通过光标移动命令将光标移动到你所需要修改的字符上, 使用 r 命令 (替换命令), 当输入 r 命令后, 紧接着输入你将用于替换的字符。

例: 光标处于 g 字符上, 希望将它替换成 h。

首先输入替换命令:

r

然后输入替换的目标字符:

h

替换完成后, vi 还处在命令模式。

(1) 断开长文本行。

替换字符的一个重要应用是将文本行断成两个文本行。在要断行的两个单词之间的空格, 执行 r 命令之后, 按回车键后就把文本分成两行。

(2) 连接两行文本。

有时需要将两行文本连接成一行, 在文件中选择两个相邻的文本行, 并将光标放在第 1 行的任意位置, 然后输出大写字母:

J

当前行和下一行的文本将连接在一起。

### 5.6.2 逐字符替换文本

r 命令能替换光标处的一个字符。其实逐个替换多个字符有时也是很方便的。它比多次使用 r 命令方便得多。

使用 R 命令, 可以逐字符替换光标后面的文本。

在命令模式下, 在光标处输入 R 后, 随后每次输入的字符将会替换光标处的字符, 当替换完成后, 按 Esc 键返回到命令模式。

### 5.6.3 多个字符替换单个字符

一些用户 (例如程序员) 可能需要删除一个字符, 用多个字符替换这个字符。这不能

通过 `r` 和 `R` 命令来实现。

用户可以使用 `s` 命令，实现删除光标处的一个字符，然后可以添加多个字符，甚至是几行和几页。

当输入完成后，按 `Esc` 键返回到命令模式。

#### 5.6.4 用多个单词替换单个单词

使用 `cw` 命令可以删除光标处的一个单词，并进入添加模式，当输入完需要的文本之后，按 `Esc` 键返回到命令模式，完成修改。

#### 5.6.5 行替换

使用 `r` 替换单个字符，使用 `R` 命令逐字替换光标处的字符，使用 `s` 命令用多个字符替换单个单符。用户也可以使用 `vi` 提供的 `cc` 命令替换当前行。

当执行了 `cc` 命令之后，`vi` 将删除当前行的所有字符，然后切换至添加模式，你可以添加任何文本（甚至是多行文本），例如输入：

```
This option sets the editor so that lines following an indented line will
have the same indentation as the previous line. If you want to back over this
indentation, you can type ^D at the very first character position.
```

This `^D` works in the insert mode, and not in command mode. Also, the width of the indentations can be set with `shiftwidth`, explained below.

按 `Esc` 键返回到命令模式，完成修改。

#### 5.6.6 修改行的剩余部分

用户可以使用 `vi` 的 `C` 命令删除从光标开始到行末的所有字符，然后进入添加模式。然后输入希望添加的内容，可以是多行文字。当输入完成后，按 `Esc` 键返回命令模式，完成修改。

表 5-12 对照了删除命令和替换命令。

表 5-12 删除和替换命令的对照

命令类型	删除	替换
字符	<code>x</code>	<code>r</code>
单词	<code>dw</code>	<code>cw</code>
行	<code>dd</code>	<code>cc</code>
从光标到行尾之间的部分	<code>D</code>	<code>C</code>
从光标到字符 <code>m</code> 之间的部分	<code>dfm</code>	<code>cfm</code>

## 5.7 全局替换

本节介绍的查找和替换都是全局的，首先用户给出模式，然后让编辑器进行搜索，然后进行定位，如果修改替换，使用目标模式进行替换。这里的查找和替换与一般的编辑器的查找和替换类似。

### 5.7.1 搜索当前行的第一个实例

使用:s 命令替换文件中的指定单词。

例：将当前行的第一个单词 **bash** 替换成 **csh**。步骤如下：

(1) 首先将光标移到一个包含单词 **bash** 的行。

(2) 输入下述命令，将 **bash** 改为 **csh**：

```
:s/bash/csh
```

当前行的第一个 **bash**，被替换为 **csh**。

:s 替换命令使用了两个斜杠 (/) 隔开被替换单词和换成的单词。如果找到:s 命令左边的单词，就用命令右边的单词替换。在这个例子中，使用了 **csh** 替换当前行的第一个 **bash**。用户可以使用 **u** 命令撤销刚才进行的替换。

### 5.7.2 搜索指定行的第一个实例

可以在:s 命令的冒号后加入行号来指定需要替换的行的范围。也可以搜索整个文件中的所有行。

将文件 **vitest** 的 1~10 行的第一个 **sbin** 单词替换成 **bin**。

首先打开 **vitest** 文件：

```
[globus@unknown globus]$ vi ~/vitest
```

然后执行如下命令：

```
:s1,10s/sbin/bin
```

当替换了每行的第一个单词 **sbin**，后还可以再次使用这条命令替换每行的第二个 **sbin**。命令如下：

```
:s1,10s/sbin/bin
```

将文件 **vitest** 的 15~行末的第一个 **sbin** 单词替换成 **bin**，命令如下：

```
:s1,$s/sbin/bin
```

### 5.7.3 设置过滤条件进行搜索

如果需要在包含特定字符串的行上搜索指定的字符串，然后替换成另一个字符串，可以使用:g 命令。

:g 命令的格式如下：

:g/过滤条件/s/搜索单词/替换的内容

例如：在文件 **vitest** 中，将包含 **var** 行中的第一个 **sbin** 单词替换成 **bin**。命令如下：

```
:g/var/s/sbin/bin
```

例如：在文件 vitest 中，将包含 var 行中的第一个 var 单词替换成 bin。命令如下：

```
:g/var/s/var/bin
```

这条命令中的第二个 var 也可以省略，因为它与:g/后的串相同。它可以写成下列两种形式：

```
:g/var/s//bin
```

或者

```
:1,$s/var/bin
```

### 5.7.4 在同一行中替换所有实例

前面的命令仅仅替换一行中的第一个实例，如果同一行有多个匹配的目标，你可以使用如下命令：

```
:s/搜索单词/替换的内容/g
```

例如：将文件 vitest 的第 2 行的所有串 bin 替换成 BIN。步骤如下：

(1) 将光标移到第 2 行。

(2) 输入如下命令：

```
:s/bin/BIN/g
```

用户也可以使用行号，搜索指定范围的行。

例：将整个文件中的 bin 替换成 BIN。命令如下：

```
:1,$s/bin/BIN/g
```

该命令的含义如表 5-13 所示。

表 5-13 :1,\$s/bin/BIN/g 命令的含义

命令	含义
1,\$	文件中的所有行
s	进行替换
bin	需要查找的字符串
BIN	替换字符串
/g	全部替换

**注意：**替换命令末尾的 g 称为标志。它与命令开头的 g 不同，命令开头的 g 是文件中所有行的默认声明，即范围为 (1, \$)。替换命令末尾的 g 标志则指示替换命令在指定行中对目标字符串的所有实例进行替换。

## 5.8 移动和复制文本

vi 提供了文本移动和复制命令，用户利用这些指令可以方便地复制行、单词和文本块。

### 5.8.1 复制和粘贴行

本节的复制指的是将一部分文本复制到缓冲区，而不是复制到目标区域。

使用 `yy` 命令可以将当前行复制到 `vi` 的缓冲区，然后将光标定位到需要粘贴文本的地方，最后使用 `p`（小写字母 `p`）命令将最后一次复制的文本，粘贴到光标放在光标之后。

也可以在 `yy` 命令之前加入数字 `n`，将从光标开始的 `n` 行文字复制到缓冲区。

多次使用 `p` 命令，可以将复制的文字粘贴到多个位置。

注意：`vi` 也提供了 `P`（大写字母 `P`）命令，它与 `p` 命令类似，只是它将复制的行粘贴到光标位置的上一行，而 `p` 命令则将复制的行粘贴到光标位置的下一行。

例：打开文件 `vitest`。命令如下：

```
[globus@unknown globus]$ vi ~/vitest
```

将光标移到第 5 行：

```
5G
```

将当前行复制到 `vi` 的缓冲区：输入复制命令：

```
yy
```

执行该命令之后，屏幕没有变化，但是 `vi` 已经将当前行复制到了缓冲区，使用 `yy` 命令并不删除被复制的行，也就是说使用 `yy` 命令对当前文件没有影响。

将光标定位到第 10 行上，然后执行粘贴命令：

```
10G
```

```
p
```

刚才使用 `yy` 命令复制的第 5 行粘贴到光标下面的一个新行上。

将光标定位到第 15 行，然后执行粘贴命令：

```
:15
```

```
p
```

刚才使用 `yy` 命令复制的第 5 行再次粘贴到光标下面的一个新行上。

将光标定位到第 5 行，复制第 5~8 行文字：

```
:5
```

```
4yy
```

将光标定位到你想要粘贴的位置，如当前行的上两行，命令如下：

```
k
```

```
k
```

将复制的 4 行文字粘贴到光标的后面：

```
p
```

将光标定位到文件尾：

```
:$
```

将复制的文本粘贴到倒数第 1 行的上面，命令如下：

```
P
```



### 5.8.2 剪切和粘贴行

当删除文本行之后，也可以使用 `p` 或者 `P` 粘贴文本。

例如：光标定位到当前文件的第 17 行。

`17G`

删除当前行：

`dd`

将光标定位到第 20 行：

`:20`

在光标后面粘贴刚才删除的行：

`p`

注意：用户可以使用删除和粘贴实现文本移动。

### 5.8.3 复制字符和单词

用户也可以使用 `vi` 提供的复制单词和字符命令。

(1) 复制单词。

`yw` 命令把从当前光标到该单词结束的文字复制到缓冲区。

例如：假定 `vi` 已经打开一个文件，处于命令状态。操作如下：

将光标定位到某一个单词的开头。

将光标处的一个单词复制到缓冲区：

`yw`

将光标定位到其他的位置。

将复制的单词粘贴到当前位置：

`p`

将光标定位到另一个的位置。

将复制的单词再次粘贴：

`p`

注意：用户也可以使用 `dw` 命令删除一个单词，然后将光标定位一个希望的位置，然后使用 `p` 命令或者 `P` 命令对单词进行粘贴。

(2) 复制字符。

`yl` 命令把当前光标处的文字复制到缓冲区。

例如：假定 `vi` 已经打开一个文件，处于命令状态。操作如下：

将光标定位到某一个字符。

将光标处的一个字符复制到缓冲区：

`yl`

将光标定位到其他的位置。

将复制的字符粘贴到当前位置：

`p`

将光标定位到另一个的位置。

将复制的字符再次粘贴:

p

注意: 用户也可以使用 x 命令删除当前字符, 然后将光标定位一个位置, 然后使用 p 命令或者 P 命令对字符进行粘贴。

(3) 复制部分文字。

yfm 命令复制当前光标处到第一个字符 m 处的文字到缓冲区, 这里的 m 可以为其他字符。

例如: 假定 vi 已经打开一个文件, 处于命令状态。操作如下:

将光标定位到某一个字符。

将从光标处到第一个字符 z 之间文字复制到缓冲区:

yfz

将光标定位到其他的位置。

将复制的文字粘贴到光标的后面:

p

将光标定位到另一个的位置。

将复制的文字粘贴到光标的前面:

P

注意: 用户也可以使用 dfm 命令删除当前光标处到第一个字符 m 之间的文字, 并保存到缓冲区, 然后将光标定位一个位置, 然后使用 p 命令或者 P 命令对文字进行粘贴。

(4) 对文字的复制、删除和粘贴的总结如表 5-14 所示。

表 5-14 vi 的复制、删除和粘贴命令

操作对象	删除	复制	修改
行	dd	yy	cc
行的剩余部分	d\$或者D	y\$	c\$或者C
当前行从光标到第一个字符m	dfm	yfm	cfm
单词	dw	yw	cw
字符	x	yl	s

#### 5.8.4 复制文本块

vi 提供了复制一行和多行文本的命令。这些命令都是基于行号的, 因此在进行文本复制时, 可打开行号显示功能, 以便能对行号精确控制。

文本复制命令的格式如下:

: 行号 [, 行号] copy 行号

命令中的第一个“行号”是要复制的文本行的行号。第二个行号是可选的, 如果没有这个行号, 表示只复制一行, 否则表示将文本复制到第三个行号的所在行的后面。这些文

本在复制时，作为一个整体复制。

例：打开文件 vitest。命令如下：

```
[globus@unknown globus]$ vi ~/vittest
```

在屏幕上显示行号：

```
:set number
```

将光标移到第 5 行：

```
5G
```

将第 7 行的文本复制到第 10 行后面，命令如下：

```
:7 copy 10
```

将第 7~9 行的文本复制到文件尾，命令如下：

```
:7,9 copy $
```

将第 7~9 行的文本复制到文件开头，命令如下：

```
:7,9 copy 0
```

将当前行至第 15 行的文本复制到第 3 行的后面，命令如下：

```
::,15 copy 3
```

注意：“.”（点）表示当前行，“\$”表示文件尾，“0”表示文件头。

### 5.8.5 移动文本块

vi 提供了移动一行和多行文本的命令。这些命令也是基于行号的，因此在进行文本移动时，可打开行号显示功能，以便能对行号精确控制。

文本移动命令的格式如下：

```
: 行号 [, 行号] move 行号
```

命令中的第一个“行号”是要移动的文本行的行号。第二个行号是可选的，如果没有这个行号，表示只移动一行，否则表示将文本移动到第三个行号的所在行的后面。这些文本在移动时，作为一个整体进行。

例：打开文件 vitest。命令如下：

```
[globus@unknown globus]$ vi ~/vittest
```

在屏幕上显示行号：

```
:set number
```

将光标移到第 5 行：

```
5G
```

将第 7 行的文本移动到第 10 行后面，命令如下：

```
:7 move 10
```

将第 7~9 行的文本移动到文件尾，命令如下：

```
:7,9 move $
```

将第 7~9 行的文本移动到文件开头，命令如下：

```
:7,9 move 0
```

将当前行至第 15 行的文本移动到第 3 行的后面，命令如下：

```
::,15 move 3
```

注意：在 copy 命令中，“.”（点）表示当前行，“\$”表示文件尾，“0”表示文件头。

## 5.9 保存文件

vi 全屏幕编辑器不仅提供了文件存盘命令，用户也可以使用 vi 的放弃文件保存命令，或者在已有的文件尾添加内容的命令。

### 5.9.1 保存当前编辑的文件

为了避免因网络中断，网络故障或者 Linux 主机的故障而导致对文件修改的丢失，或者由于用户自身的误操作，用户需要及时的保存对文件的修改。

（1）保存对文件的修改。

在 vi 命令模式中，输入：

```
:w
```

可以保存对文件的修改。

（2）保存对文件的修改后退出编辑器。

在 vi 命令模式中，输入：

```
:x
```

或者

```
:wq
```

可以保存对文件的修改，然后退出编辑器，返回到 shell 提示符。

（3）保存新版本。

在有些情况下，用户可能希望保持原文件不变，而将修改后的文件保存到一个新文件，要保存文件的新版本，使用带参数的:w 命令，它的参数是要保存的文件名。

例：将当前修改的文件，保存一个新版本，新版本的文件名为 testvinew。命令如下：

```
:w testvinew
```

这里的 testvinew 是任意的文件名。

屏幕的底端提示行将会出现如下类似的信息：

```
"testvinew" [已转换][新] 51L, 2291C 已写入          1,1          顶端
```

这说明 vi 已经将编辑的文件的内容另存成文件名为 testvinew 的副本，由于:w testvinew 命令中的文件名没有带路径，因此新文件 testvinew 保存的位置为当前目录。

### 5.9.2 取消对文件的修改

使用 vi 编辑器对文件进行编辑，如果用户发现原有的文件很重要，不应该修改，或者发现大当前的修改不仅破坏了原有文件，而且对文件的修改不是你所希望的。这时需要放弃对文件的修改，而保持原文件的内容。在 vi 中，为了放弃对文件的修改，必须使用命令退出编辑器，如果需要进一步修改，以后可以再次进入 vi 编辑器环境。

如果你没有对文件进行修改，或者修改的副本已经保存到另一个文件，用户可以放弃当前的修改，返回到 shell 提示符。命令如下：

```
:q
```

如果对文件进行了修改，而且没有保存到另一个文件，当使用:q 命令时，vi 会在屏幕底端出现如下的错误提示：

```
E37: No write since last change (add ! to override)
```

如果以后确实想放弃对当前文件修改，那么输入如下命令：

```
:q!
```

这样就会不保存修改，而返回到 shell 提示符。

注意：如果用户仅做了几处修改，又不希望退出 vi 编辑器，这时用户可以连续使用多次 u 命令撤销原来的修改，直到恢复到用户希望的位置为止。

### 5.9.3 保存文本块

用户经常抽取正在编辑的文件的一部分内容，并将这部分内容保存到一个新文件。要将当前编辑的文件的一部分文本保存到文件，需要如下两方面的信息：

- 要保存文本的开始行号和结束行号。
- 用来保存文本的新文件名。

将当前编辑的文件中的 10~15 行的文本，保存到新文件 vitestpart 中，命令如下：

```
:10,15 write vitestpart
```

编辑器 vi 的屏幕底端将会出现如下的保存提示：

```
"vitestpart" [已转换][新] 6L, 259C 已写入 16,8 顶端
```

通过行号 10, 15 将当前文件的 10~15 行，共 6 行文本保存到新文件 vitestpart。从输出提示可以看出，新文件中共有 259 个字符。

可以在 shell 提示符中查看新产生的文件的内容，命令如下：

```
[globus@unknown globus]$ less vitestpart
```

### 5.9.4 覆盖已有文件

在使用:w filename 命令时，如果新文件在系统中已经存在，为了不丢失信息，vi 将不能覆盖这个文件，再次执行：

```
:10,15 write vitestpart
```

并出现如下的提示信息：

```
E13: File exists (add ! to override) 16,1 顶端
```

它提示用户如果真想覆盖文件 vitestpart，可以在 write 之后加一个感叹号来实现，名列如下：

```
:10,15 write! vitestpart
```

编辑器 vi 的屏幕底端将会出现如下的保存提示：

```
"vitestpart" [已转换][新] 6L, 259C 已写入 16,8 顶端
```

### 5.9.5 向文件追加文本

vi 也提供了向文件中追加文件的命令。

例：将当前文件的 10~15 内容追加到文件 vitestpart 的尾部。命令如下：

```
:10,15 write >> vitestpart
```

编辑器 vi 的屏幕底端将会出现如下的追加提示:

"vittestpart" [已转换] 6L, 259C 已附加

16,1

顶端

## 5.10 vi 编辑器与 shell 的交互

一般来说,使用 vi 编辑器进行文件编辑时,用户只需要使用 vi 的命令即可完成编辑任务,但是在一些特殊的情况下,也可以指示 vi 传递命令给一个子 shell,以便读入其他文件的内容,或者执行 shell 命令。

### 5.10.1 读入文本文件

在文件编辑时,经常需要在编辑一个文件时读入另外一个文件的内容。一些已有的文本文件中包含当前被编辑的文件所需的信息,可以在需要时将它们读入正在编辑的文件中,这样减少重复劳动,这时读入的内容仅是文件的一个副本,原始文件的内容并不会发生任何改变。

从其他文件读入文本,需要提供如下信息:

- 要复制到当前文件的文本文件名。
- 新文件的内容在该文件出现的行号。

例:将 vittestpart 的内容读入到当前文件的第 1~2 行之间。命令如下:

```
:1 read vittestpart
```

该命令读取文件 vittestpart,并将该文件的所有文本行放在第 1~2 行之间。如果在 read 命令中没有包含行号信息,读入的文件将放到光标所有行之后。

注意:read 命令可以缩写为 r。

### 5.10.2 在 vi 中运行子 shell

在 vi 编辑器中,可以让 shell 执行命令。

在 vi 的命令模式中,输入如下命令:

```
:!ls
```

ls 命令的结果如下:

[更新后尚未保存]"

行 6

```
hello1.txt hello3.txt newdir3 testvinew vittestpart
```

```
hello2.txt hello4.txt newdir4 vittest
```

请按 ENTER 或其他命令继续

!命令启动一个子 shell,并运行!之后的命令。

ls 的输出显示在屏幕上,并提示用户按 ENTER 键或者其他键继续 vi 的编辑。当按回车键之后,屏幕被清空,并显示当前的编辑会话。

输入如下命令,运行 who 程序:

```
:!who
```

who 命令的结果如下:

[更新后尚未保存]

```

root      :0          May 24 09:06
root      pts/3        May 30 11:09
globus    pts/4        Jun  1 10:51 (192.168.1.188)
henry     pts/2        Jun  1 16:58 (192.168.1.168)

```

请按 ENTER 或其他命令继续

### 5.10.3 读入 shell 命令行的输出

在 5.10.2 中, 运行了 shell 命令, 并将结果输出到屏幕上, 但是这并不影响当前被编辑的文件, 也可以将 shell 命令的输出读入到当前被编辑的文件中。

编辑器 vi 将 read 命令和执行子 shell 命令结合起来, 可以完成读入 shell 命令的输出的功能。

例: 将列命令 ls 的输出读入到当前光标所在的行之后。命令如下:

```
:read !ls
```

假定读入 shell 命令结果之前光标处在当前文件的第 6 行, 那么读入 ls 命令的输出之后, 被编辑的文件的结果如下:

```

root:x:0:0:root:/root:/bin/bash
news:x:9:13:news:/etc/news:
uucp:x:10:14:uucp:/var/spool/uucp:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
gopher:x:13:30:gopher:/var/gopher:/sbin/nologin
hello1.txt
hello2.txt
hello3.txt
hello4.txt
newdir3
newdir4
testvinew
vitest
vitestpart
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
news:x:9:13:news:/etc/news:
uucp:x:10:14:uucp:/var/spool/uucp:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
gopher:x:13:30:gopher:/var/gopher:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
bin:x:1:1:bin:/bin:/sbin/nologin

```

还有 9 行

15,1

顶端

上述具有下划线的行是使用 ls 命令的输出。最下面一行是提示行。

例：将 who 命令的输出读入到文件的开头。命令如下：

```
:0 read !who
```

读入 who 命令的输出之后，被编辑的文件的结果如下：

```
root      :0      May 24 09:06
root      pts/3    May 30 11:09
globus    pts/4     Jun  1 10:51 (192.168.1.188)
henry     pts/2     Jun  1 16:58 (192.168.1.168)
root:x:0:0:root:/root:/bin/bash
news:x:9:13:news:/etc/news:
uucp:x:10:14:uucp:/var/spool/uucp:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
gopher:x:13:30:gopher:/var/gopher:/sbin/nologin
hello1.txt
hello2.txt
hello3.txt
hello4.txt
newdir3
newdir4
testvinew
vitest
vitestpart
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
news:x:9:13:news:/etc/news:
uucp:x:10:14:uucp:/var/spool/uucp:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
```

还有 4 行

4,1

顶端

上述具有下划线的行是使用 who 命令的输出。最下面一行是提示行。

## 思考与练习

- (1) Linux 下的全屏编辑器 vi 与 Windows 下的 notepad 有什么区别？
- (2) 在 vi 中，有哪些移动光标的命令？它们的作用分别是什么？
- (3) 在 vi 中如何替换文本？
- (4) 在 vi 中如何撤销以前的操作？



## 第 6 章 shell 的使用

shell 本身是一个用 C 语言编写的程序，它是用户使用 Linux 的桥梁。Shell 既是一种命令语言，又是一种程序设计语言。作为命令语言，它交互式地解释和执行用户输入的命令；作为程序设计语言，它定义了各种变量和参数，并提供了许多在高级语言中才具有的控制结构，包括循环和分支。它虽然不是 Linux 系统核心的一部分，但它调用了系统核心的大部分功能来执行程序、建立文件并以并行的方式协调各个程序的运行。因此，对于用户来说，shell 是最重要的实用程序，深入了解和熟练掌握 shell 的特性及其使用方法，是用好 Linux 系统的关键。可以说，shell 使用的熟练程度反映了用户对 Linux 使用的熟练程度。

本章内容主要包括 shell 简介、shell 程序设计基础、流程控制、bash 程序的调试、bash 的内部命令以及使用 bash 的一些技巧。

### 6.1 shell 基础

#### 6.1.1 什么是 shell

当一个用户登录到 Linux 系统之后，系统初始化程序 init 就为每一个用户运行一个称为 shell（外壳）的程序。那么 shell 是什么呢？确切地说，shell 就是一个命令行解释器，它为用户提供了一个向 Linux 内核发送请求以便运行程序的界面系统级程序，用户可以用 shell 来启动、挂起、停止甚至是编写一些程序。

当用户使用 Linux 时是通过命令来完成所需工作的。一个命令就是用户和 shell 之间对话的一个基本单位，它是由多个字符组成并以换行结束的字符串。shell 解释用户输入的命令，就象 DOS 里的 command.com 所做的一样，所不同的是，在 DOS 中，command.com 只有一个，而在 Linux 下比较流行的 shell 有好几个，每个 shell 都各有千秋。一般的 Linux 系统都将 bash 作为默认的 shell。

#### 6.1.2 几种流行的 shell

目前流行的 shell 有 sh、csh、ksh、tcsh 和 bash 等，用户可以用下面的命令来查看你自己的 shell 类型：

```
#echo $SHELL
```

注意：#echo \$SHELL 命令前面的#是 shell 提示符，它表示系统的当前用户为 root，

而\$echo \$SHELL 命令中的\$也是 shell 提示符，它表示当前用户为普通用户。

这两条命令的意思分别是：以 root 用户，或者以普通用户身份，在 shell 提示符后输入 echo \$SHELL 命令，并且按回车键。

\$SHELL 是一个环境变量，它记录用户所使用的 shell 类型。可以用命令：

```
#shell-name
```

来转换到其他的 shell，这里 shell-name 是你想要尝试使用的 shell 的名称，如 ash, csh

等。这个命令为用户又启动了一个 shell，这个 shell 在最初登录的那个 shell 之后，称为下级的 shell 或子 shell。

如果想退出这个子 shell，使用如下命令：

```
$exit
```

有关这些常用 shell 的特征，请参阅 3.1.2 介绍。

## 6.2 shell 程序设计基础

作为命令语言交互式地解释和执行用户输入的命令只是 shell 功能的一个方面，shell 还可以用来进行程序设计，它提供了定义变量和参数的手段以及丰富的程序控制结构。使用 shell 编程类似于 DOS 中的批处理文件，称为 shell 脚本(shell script)，shell 程序或者 shell 命令文件。

### 6.2.1 shell 基本语法

shell 的基本语法主要就是如何输入命令来运行程序以及如何在程序之间通过 shell 的一些参数提供便利手段来进行通讯。

#### 1. 输入输出重定向

在 Linux 中，每一个进程都有 3 个特殊的文件描述指针：标准输入（standard input，文件描述指针为 0）、标准输出（standard output，文件描述指针为 1）、标准错误输出（standard error，文件描述指针为 2）。这 3 个特殊的文件描述指针使进程在一般情况下接收标准输入终端的输入，同时由标准终端来显示输出，Linux 也可以使用普通的文件或管道来取代这些标准输入输出设备。在 shell 中，使用者可以利用“>”和“<”来进行输入输出重定向。

- 命令>文件：将命令的输出结果重定向到一个文件。
- 命令>&文件：将命令的标准错误输出一起重定向到一个文件。
- 命令>>文件：将标准输出的结果追加到文件中。
- 命令>>&文件：将标准输出和标准错误输出的结果都追加到文件中。

#### 2. 管道 (pipe)

pipe 同样可以在标准输入输出和标准错误输出间做代替工作，这样一来，可以将某一个程序的输出送到另一个程序的输入，其语法如下：

```
命令1 | 命令2 [| 命令3...]
```

也可以连同标准错误输出一一起送入管道：

```
命令1 | &命令2 [| &命令3...]
```

#### 3. 前台和后台

在 shell 下面，一个新产生的进程可以通过用命令后面的符号“;”和“&”来分别以前台和后台的方式来执行，语法如下：

```
命令
```

执行“命令”，产生一个前台的进程，下一个命令须等该命令运行结束后才能输入。

命令 &

产生一个后台的进程，此进程在后台运行的同时，可以输入其他的命令。

### 6.2.2 shell 程序的变量和参数

与高级程序设计语言一样，shell 也提供声明和使用变量的功能。对 shell 来讲，所有变量的取值都是一个字符串，shell 程序采用 \$var 的形式来引用名为 var 的变量的值。

shell 有以下几种基本类型的变量：

#### 1. shell 定义的环境变量

shell 在开始执行时就已经定义了一些和系统的工作环境有关的变量，这些变量用户还可以重新定义，常用的 shell 环境变量如下所示。

- HOME：用于保存注册目录的完全路径名。
- PATH：用于保存用冒号分隔的目录路径名，shell 将按 PATH 变量中给出的顺序搜索这些目录，找到的第一个与命令名称一致的可执行文件将被执行。
- TERM：终端的类型。
- UID：当前用户的标识符，取值是由数字构成的字符串。
- PWD：当前工作目录的绝对路径名，该变量的取值随 cd 命令的使用而变化。
- PS1：主提示符，在特权用户下，默认的主提示符是“#”，在普通用户下，默认的主提示符是“\$”。
- PS2：在 shell 接收用户输入命令的过程中，如果用户在输入行的末尾输入“\”然后回车，或者当用户按回车键时 shell 判断出用户输入的命令没有结束时，显示这个辅助提示符，提示用户继续输入命令的其余部分，默认的辅助提示符是“>”。

#### 2. 用户定义的变量

用户可以按照下面的语法规则定义自己的变量：

变量名=变量值

**注意：**在定义变量时，变量名前不应加符号“\$”，在引用变量的内容时则应在变量名前加“\$”；在给变量赋值时，等号两边一定不能留空格，如果变量中本身就包含了空格，则整个字符串都要用双引号括起来。

在编写 shell 程序时，为了使变量名和命令名相区别，建议所有的变量名都用大写字母来表示。

有时用户想要在说明一个变量并对它设定为一个特定值后就不再改变它的值，这可以用下面的命令来保证一个变量的只读性：

readonly 变量名

在任何时候，建立的变量都只是当前 shell 的局部变量，所以不能被 shell 运行的其他命令或 shell 程序所利用，export 命令可以将一局部变量提供给 shell 执行的其他命令使用，其格式为：

export 变量名

也可以在给变量赋值的同时使用 `export` 命令：

```
export 变量名=变量值
```

使用 `export` 说明的变量，在 `shell` 以后运行的所有命令或程序中都可以访问到。

### 3. 位置参数

位置参数是一种在调用 `shell` 程序的命令行中按照各自的位置决定的变量，它是在程序名之后输入的参数。位置参数之间用空格分隔，`shell` 取第一个位置参数替换程序文件中的 `$1`，第二个替换 `$2`，依次类推。`$0` 是一个特殊的变量，它的内容是当前这个 `shell` 程序的文件名，所以 `$0` 不是一个位置参数，在显示当前所有的位置参数时是不包括 `$0` 的。

### 4. 预定义变量

预定义变量与环境变量类似，也是在 `shell` 一开始时就定义了的变量。不同的是，用户只能根据 `shell` 的定义来使用这些变量，而不能重定义它。所有预定义变量都是由 `$` 符和另一个符号组成的，常用的 `shell` 预定义变量如下所示。

- `$#`：位置参数的数量。
- `$*`：所有位置参数的内容。
- `$?`：命令执行后返回的状态。
- `$$_`：当前进程的进程号。
- `$_`：后台运行的最后一个进程号。
- `$0`：当前执行的进程名。

其中，“`$?`”用于检查上一个命令执行是否正确。

注意：在 Linux 中，命令退出状态为 0 表示该命令正确执行，任何非 0 值表示命令出错。

“`$_`”变量最常见的用途是用作临时文件的名字以保证临时文件不会重复。

### 5. 参数置换的变量

`shell` 提供了参数置换能力，以使用户可以根据不同的条件来给变量赋不同的值。参数置换的变量有 4 种，这些变量通常与某一个位置参数相联系，根据指定的位置参数是否已经设置，来决定变量的取值，它们的语法和功能分别如下：

- 变量=`${参数-word}`。如果设置了参数，则用参数的值置换变量的值，否则用 `word` 置换，即这种变量的值等于某一个参数的值。如果该参数没有设置，则变量就等于 `word` 的值。
- 变量=`${参数=word}`。如果设置了参数，则用参数的值置换变量的值，否则把变量设置成 `word` 然后再用 `word` 替换参数的值。注意，位置参数不能用于这种方式，因为在 `shell` 程序中不能为位置参数赋值。
- 变量=`${参数? word}`。如果设置了参数，则用参数的值置换变量的值，否则就显示 `word` 并从 `shell` 中退出，如果省略了 `word`，则显示标准信息。这种变量要求一定等于某一个参数的值，如果该参数没有设置，则显示一个信息，然后退出，因此这种方式常用于出错指示。

- 变量=\${参数+word}。如果设置了参数，则用 word 置换变量，否则不进行置换。
- 注意：所有这 4 种形式中的“参数”既可以是位置参数，也可以是另一个变量，只是用位置参数的情况比较多。

## 6.3 shell 编程的流程控制

本节以 bash 为例向读者介绍 shell 程序设计的高级部分：shell 编程的流程控制、调试方法及 shell 程序的运行方法，同时也向读者介绍一下 bash 的内部命令。

### 6.3.1 shell 程序设计的流程控制

和其他高级程序设计语言一样，shell 提供了用来控制程序执行流程的命令，包括条件、分支和循环结构，用户可以用这些命令建立非常复杂的程序。

注意：与传统的语言不同的是，shell 用于指定条件值的不是布尔表达式，而是命令和字符串。

#### 1. test 测试命令

test 命令用于检查某个条件是否成立，它可以进行数值、字符和文件 3 个方面的测试，其测试符和相应的功能分别如下：

- 数值测试。
  - eq: 等于则为真
  - ne: 不等于则为真
  - gt: 大于则为真
  - ge: 大于等于则为真
  - lt: 小于则为真
  - le: 小于等于则为真
- 字符串测试。
  - =: 等于则为真
  - !=: 不相等则为真
  - z 字符串: 字符串长度为零则为真
  - n 字符串: 字符串长度不为零则为真
- 文件测试。
  - e 文件名: 如果文件存在则为真
  - r 文件名: 如果文件存在且可读则为真
  - 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
  - s 文件名: 如果文件存在且至少有一个字符则为真
  - d 文件名: 如果文件存在且为目录则为真
  - f 文件名: 如果文件存在且为普通文件则为真
  - c 文件名: 如果文件存在且为字符特殊文件则为真

**-b** 文件名：如果文件存在且为块特殊文件则为真

另外，Linux 还提供了与 (“!”)、或 (“-o”)、非 (“-a”) 3 个逻辑操作符用于将测试条件连接起来，其优先级为：“!” 最高，“-a” 次之，“-o” 最低。

同时，bash 也能完成简单的算术运算，格式如下：

```
$[表达式]
```

执行下列运算：

```
v1=10
```

```
v2=$((v1*v1+5))
```

则 v2 的值将为 105。

## 2. if 条件语句

shell 程序中的条件分支是通过 if 条件语句来实现的，其一般格式为：

```
if 条件命令串
```

```
then
```

```
    条件为真时的命令串
```

```
else
```

```
    条件为假时的命令串
```

```
fi
```

## 3. for 循环

for 循环对一个变量每一个可能的值都执行一个命令序列。赋给变量的几个数值既可以在程序内以数值列表的形式提供，也可以在程序以外以位置参数的形式提供。for 循环的一般格式为：

```
for 变量名
```

```
[in 数值列表]
```

```
do
```

```
    若干个命令组成的命令列表
```

```
done
```

变量名可以是用户选择的任何字符串。如果变量名是 v，则在 in 之后给出的数值将顺序替换循环命令列表中的 \$v。如果省略了 in，则变量的取值将是位置参数。对变量的每一个可能的赋值都将执行 do 和 done 之间的命令列表。

## 4. while 和 until 循环

while 和 until 命令都是用命令的返回状态值来控制循环的。while 循环的一般格式为：

```
while
```

```
    若干个命令行1
```

```
do
```

```
    若干个命令行2
```

```
done
```

只要 `while` 的“若干个命令行 1”中最后一个命令的返回状态为真，`while` 循环就继续执行 `do` 和 `done` 之间的“若干个命令行 2”。

`until` 命令是另一种循环结构，它和 `while` 命令相似，其格式如下：

```
until
    若干个命令行1
do
    若干个命令行2
done
```

`until` 循环和 `while` 循环的区别在于：`while` 循环在条件为真时继续执行循环，而 `until` 则是在条件为假时继续执行循环。

shell 还提供了 `true` 和 `false` 两条命令用于建立无限循环结构的需要，它们的返回状态分别是总为 0 或总为非 0

### 5. case 条件选择

`if` 条件语句用于在两个选项选定一项，而 `case` 条件选择为用户提供了根据字符串或变量的值从多个选项中选择一项的方法，其格式如下：

```
case string in
    exp-1)
        若干个命令行1
    ;;
    exp-2)
        若干个命令行2
    ;;
    ...
    *)
        其他命令行
esac
```

shell 通过计算字符串 `string` 的值，将其结果依次与表达式 `exp-1`、`exp-2` 等进行比较，直到找到一个匹配的表达式为止，如果找到了匹配项则执行它下面的命令直到遇到一对分号（`;;`）为止。

**注意：**在 `case` 表达式中也可以使用 shell 的通配符（`*`、`?`、`[]`）。通常用 `*` 作为 `case` 命令的最后表达式以便使在前面找不到任何相应的匹配项时执行“其他命令行”的命令。

### 6. 无条件控制语句 `break` 和 `continue`

`break` 用于立即终止当前循环的执行，而 `continue` 用于不执行循环中后面的语句而立即开始下一个循环的执行。这两个语句只有放在 `do` 和 `done` 之间才有效。

## 7. 函数定义

在 shell 中还可以定义函数。函数实际上也是由若干条 shell 命令组成的, 因此它与 shell 程序形式上是相似的, 不同的是它不是一个单独的进程, 而是 shell 程序的一部分。函数定义的基本格式为:

```
functionname
{
  若干命令行
}
```

调用函数的格式为:

```
functionname param1 param2 ...
```

shell 函数可以完成某些例行工作, 而且还可以有自己的退出状态, 因此函数也可以作为 if、while 等控制结构的条件。

在函数定义时不用带参数说明, 但在调用函数时可以带有参数, 此时 shell 将把这些参数分别赋予相应的位置参数 \$1, \$2, ..., \$\*。

## 8. 命令分组

在 shell 中有两种命令分组的方法: “()” 和 “{}”。前者当 shell 执行()中的命令时将再创建一个新的子进程, 然后这个子进程去执行圆括弧中的命令。当用户在执行某个命令时, 不想让命令运行时对状态集合(如位置参数、环境变量、当前工作目录等)的改变影响到下面语句的执行时, 就应该把这些命令放在圆括弧中, 这样就能保证所有的改变只对子进程产生影响, 而父进程不受任何干扰。{} 用于将顺序执行的命令的输出结果用于另一个命令的输入(管道方式)。当用户要真正使用圆括弧和花括弧时(如计算表达式的优先级), 则需要在其前面加上转义符 (\), 以便让 shell 知道它们不是命令执行控制所用。

## 9. 信号

trap 命令用于在 shell 程序中捕捉到信号, 之后可以有 3 种反应方式:

- 执行一段程序来处理这一信号。
- 接受信号的默认操作。
- 忽视这一信号。

trap 对上面 3 种方式提供了 3 种基本形式:

第一种形式的 trap 命令在 shell 接收到 signal list 清单中数值相同的信号时, 将执行双引号中的命令串。

```
trap 'commands' signal-list
trap "commands" signal-list
```

为了恢复信号的默认操作, 使用第二种形式的 trap 命令:

```
trap signal-list
```

第三种形式的 trap 命令允许忽视信号:

```
trap " " signal-list
```

注意: 对信号 11 (段违例) 不能捕捉, 因为 shell 本身需要捕捉该信号去进行内存的



转储。

在 `trap` 中可以定义对信号 0 的处理（实际上没有这个信号），shell 程序在其终止（如执行 `exit` 语句）时发出该信号。

在捕捉到 `signal-list` 中指定的信号并执行了相应的命令之后，如果这些命令没有将 shell 程序终止的话，shell 程序将继续执行收到信号时所执行的命令后面的命令，这样将很容易导致 shell 程序无法终止。

另外，在 `trap` 语句中，单引号和双引号是不同的，当 shell 程序第一次碰到 `trap` 语句时，将把 `commands` 中的命令扫描一遍。此时若 `commands` 是用单引号括起来的话，那么 shell 不会对 `commands` 中的变量和命令进行替换，否则 `commands` 中的变量和命令将用当前具体的值来替换。

### 6.3.2 运行 shell 程序的方法

用户可以用任何编辑程序来编写 shell 程序（shell 脚本）。因为 shell 程序是解释执行的，所以不需要编译装配成目标程序，按照 shell 编程的惯例，以 `bash` 为例，程序的第一行一般为 `#!/bin/bash`，其中 `#` 表示该行是注释，叹号（`!`）告诉 shell 运行叹号之后的命令并用文件的其余部分作为输入，也就是运行 `/bin/bash`，并让 `/bin/bash` 去执行 shell 程序的内容。

执行 shell 脚本的方法有 3 种。

（1）`bash` shell 脚本名。

这种方法的命令格式为：

```
bash shell脚本名
```

这实际上是调用一个新的 `bash` 命令解释程序，而把 shell 程序文件名作为参数传递给它。新启动的 shell 将去读指定的文件，执行文件中列出的命令，当所有的命令都执行完结束。该方法的优点是可以利用 shell 调试功能。

（2）`sh` shell 脚本名。

这种方法的格式为：

```
sh shell脚本名
```

这种方式就是利用输入重定向，使 shell 命令解释程序的输入取自指定的程序文件。

（3）用 `chmod` 命令赋予 shell 脚本的可执行权限。

一个文件能否运行取决于该文件是否具有执行权。对于 shell 程序，当用编辑器生成一个文件时，系统赋予的许可权限都是 `644(rw-r--)`，因此，当用户需要运行这个文件时，需要将该文件变为可执行的，命令如下：

```
chmod +x shell脚本名
```

然后只需要直接键入文件名即可执行。

在这 3 种运行 shell 程序的方法中，最好按下面的方式选择：当刚建立一个 shell 程序，对它的正确性还没有把握时，应当使用第一种方式进行调试。当一个 shell 程序已经调试好时，应使用第三种方式把它固定下来，使用 `chmod` 变为可执行文件，以后只要键入相应的文件名即可，并可被另一个程序所调用。

## 6.4 bash 程序的调试

在编程过程中难免会出错，通常调试程序比编写程序花费的时间还要多得多，shell 程序同样如此。

### 6.4.1 shell 程序的调试

shell 程序的调试主要是利用 bash 命令解释程序的选项。调用 bash 的形式是：

```
bash -选项 shell程序文件名
```

几个常用的选项是

- **-e**: 如果一个命令失败就立即退出。
- **-n**: 读入命令但是不执行它们。
- **-u**: 置换时把未设置的变量看作出错。
- **-v**: 当读入 shell 输入行时把它们显示出来。
- **-x**: 执行该命令时把命令和它们的参数显示出来。

上面的所有选项也可以在 shell 程序内部用“set -选项”的形式引用，而“set +选项”则将禁止该选项起作用。如果只想对程序的某一部分使用某些选项时，则可以将该部分用上面两个语句包围起来。

未设置变量退出特性（-u 选项）允许用户对所有变量进行检查，如果引用了一个未赋值的变量，就终止 shell 程序的执行。shell 通常允许未设置变量的使用，在这种情况下，变量的值为空。如果设置了未置变量退出选项，则一旦使用了未设置变量就显示错误信息，并终止程序的运行。未置变量退出选项为-u。

当 shell 运行时，若遇到不存在或不可执行的命令、重定向失败或命令异常中止等情况时，如果未经重新定向，该出错信息会打印在终端屏幕上，而 shell 程序仍将继续执行。要想在错误发生时迫使 shell 程序立即结束，可以使用-e 选项，立即终止 shell 程序的执行。

### 6.4.2 shell 程序的跟踪

调试 shell 程序的主要方法是利用 shell 命令解释程序的-v 或-x 选项来跟踪程序的执行。-v 选择项使 shell 在执行程序的过程中把它读入的每一个命令行都显示出来，而-x 选择项使 shell 在执行程序的过程中把它执行的每一个命令在行首用一个“+”加上命令名显示出来，并把每一个变量和该变量所取的值也显示出来。因此，它们的主要区别在于：在执行命令之前无-v 则打印出命令行的原始内容，而有-v 则打印出经过替换后的命令行的内容。

除了使用 shell 的-v 和-x 选择项以外，还可以在 shell 程序内部采取一些辅助调试的措施。例如，可以在 shell 程序的一些关键地方使用 echo 命令把必要的信息显示出来，它的作用相当于 C 语言中的 printf 语句，这样就可以知道程序运行到什么地方及程序目前的状态。

## 6.5 bash 的内部命令

bash 命令解释程序包含了一些内部命令。内部命令在目录列表时是看不见的，它们由 shell 本身提供。常用的内部命令有：echo、eval、exec、export、readonly、read、shift、wait 和点 (.)。下面简单介绍其命令格式和功能。

### (1) echo

命令格式：

echo arg

功能：在屏幕上打印出由 arg 指定的字符串。

### (2) eval

命令格式：

eval args

功能：当 shell 程序执行到 eval 语句时，shell 读入参数 args，并将它们组合成一个新的命令，然后执行。

### (3) exec

命令格式：

exec 命令参数

功能：当 shell 执行到 exec 语句时，不会去创建新的子进程，而是转去执行指定的命令，当指定的命令执行完时，该进程（也就是最初的 shell）就终止了，所以 shell 程序中 exec 后面的语句将不再被执行。

### (4) export

命令格式：

export 变量名

或

export 变量名=变量值

功能：shell 可以用 export 把它的变量向下带入子 shell，从而让子进程继承父进程中的环境变量。但子 shell 不能用 export 把它的变量向上带入父 shell。

注意：不带任何变量名的 export 语句将显示出当前所有的 export 变量。

### (5) readonly

命令格式：

readonly 变量名

功能：将一个用户定义的 shell 变量标识为不可变（只读）的。不带任何参数的 readonly 命令将显示出所有只读的 shell 变量。

### (6) read

命令格式：

read 变量名表

功能：从标准输入设备读入一行，分解成若干个单词，然后赋值给 shell 程序内部定义的变量。

### (7) shift 语句

功能: shift 语句按如下方式重新命名所有的位置参数变量: \$2 成为 \$1, \$3 成为 \$2, ..., 在程序中每使用一次 shift 语句, 都使所有的位置参数依次向左移动一个位置, 并使位置参数 "\$#" 减一, 直到减到 0。

### (8) wait

功能: 使 shell 等待在后台启动的所有子进程结束。wait 的返回值总是真。

### (9) exit

功能: 退出 shell 程序。在 exit 之后可有选择地指定一个数字作为返回状态。

### (10) "." (点)

命令格式:

. shell脚本文件名

功能: 使 shell 读入指定的 shell 程序文件并依次执行文件中的所有语句。

## 6.6 bash 的其他命令

### 6.6.1 简化命令输入

当用户向 bash 或其他 shell 输入一条命令时, 没有必要输入完整的命令。

例如, 假设当前工作目录包含下列文件和子目录:

```
[globus@unknown globus]$ ls -F
hello1.txt* hello3.txt newdir3/ testvinew vitestpart
hello2.txt hello4.txt newdir4/ vitest
```

如果用户想从当前工作目录转到子目录 newdir3, 可以输入:

```
#cd newdir3
```

尽管这条命令可以达到预期的目的, 但还有更简便的方法。因为 newdir3 是当前目录下以 n 开头的两个目录之一, 所以如果首先只键入下面的命令:

```
[globus@unknown globus]$ cd n
```

然后按 Tab 键, bash 将补全这两个目录名前面共同的部分, 并发出警报声, 提醒用户目录名并没有完全补齐, 原因是以 n 开头的目录名不惟一, 结果如下:

```
[globus@unknown globus]$ cd newdir
```

如果你不知道 newdir 后应该输入那一个字符, 则可以再次按 Tab 键, 它将会列出所有满足 newdir 打头的目录名, 结果如下:

```
[globus@unknown globus]$ cd newdir
newdir3 newdir4
[globus@unknown globus]$ cd newdir
```

它提示当前目录有两个目录都是以 newdir 打头的, 分别为 newdir3 和 newdir4。

把它显示在屏幕上。这时只要按 Enter 键就可以执行这条命令了。这时用户需要手工输入 3, 以便将 newdir 补齐为 newdir3。然后按回车键结束。

对于像这样的简单命令, 也许看不到使用 Tab 键有多大的价值, 在使用短小命令时,

这样做甚至会降低输入速度。但当习惯使用它以后，用户就会发现它非常有用。

**注意：**在输入命令的任何时刻，都可以按 Tab 键。这时 bash 将会试图补全此时键入的部分命令。如果不能补全，bash 就会用声音提醒用户它需要更多的信息。

### 6.6.2 重复执行命令

bash 还支持历史命令 (history)，它可以保留一定数量的、曾经送入 shell 的命令。在 Linux 系统上工作时，经常会重复执行同一组命令。例如，在开发程序和查错时，将重复进行相同的循环。编辑源程序、编译源程序、运行可执行文件来测试它的性能，然后再从头开始。为了避免重复输入同样的命令，bash 在送入命令时将它们保存起来，需要时可重复使用这些命令。bash 用命令记录列表 (history list) 保存这些命令。命令记录列表一般可保留 1000 行命令。每当用户退出系统时，bash 会自动将当前命令记录列表保存到一个文件中。默认的文件名称是 .bash\_history，它位于用户的主目录下，在下次登录时，bash 自动将历史文件的内容加载到用户的命令记录列表中。

#### 1. 使用方向键使用命令记录列表中的命令

存入命令记录列表的命令并非都是有用的，所以 bash 提供了几种方法来调用它们。最简单的是使用向上和向下方向键，将逐个列出使用过的命令，用向上的方向键会使最新敲入的命令重新显示在命令提示符后，再用向上方向键就可以得到倒数第二次输入的命令，以此类推。

如果需要的话，命令记录列表中的命令是可以编辑的。用户可以使用向上或者向下的方向键来得到曾经使用过的命令行。可以在命令行中任意一点插入一条命令，也可以用退格键 (Backspace) 或者删除键 (Delete) 删除一条命令。大多数用户都能发现这些简单有效的编辑方法。

#### 2. 使用 history 命令操作命令记录列表

另一种使用历史文件的方法是用 history 命令，history 命令有两种使用方法。第一种是使用命令：

```
# history [n]
```

如果 history 命令后不带任何参数，那么整个命令记录列表的内容都会显示在屏幕上。在 history 命令后跟上参数 n，会使命令记录列表中最后 n 条命令被显示出来。

**例：**显示命令记录列表中最后 15 条命令。命令如下：

```
[globus@unknown globus]$ history 15
```

显示结果如下：

```
529  ls -F
530  cd newdir
531  ls -a .*
532  ls -D -a .*
533  ls -dD -a .*
534  ls -d -a .*
```

```

535 ls -d -a .* -l
536 less .bash_history
537 echo $$
538 ps | grep 12451
539 ps | grep `echo $$`
540 ps | grep `echo $$`
541 ps | grep `$$`
542 ps | grep `echo $$`
543 history 15

```

history 的第二种使用方法是使用下面的格式改变历史文件或命令记录列表的内容:

```
# history [-r | w | a | n] [filename]
```

- -r 是告诉 history 命令读取历史文件的内容, 并把它们作为当前的命令记录列表。
- -w 是告诉 history 命令把当前命令记录列表写入历史文件 (覆盖当前历史文件)。
- -a 参数把当前命令记录列表添加到历史文件的尾部。
- -n 参数把历史文件中的最后 n 行读取到当前命令记录列表中。

history 命令的这些参数的执行结果都会被送入其后的[filename]中, 并把该文件作为历史文件。如果没有定义任何文件名, 执行结果就会送入 shell 变量 HISTFILE。

### 3. 重复最后使用的命令

使用!!将重复使用最后一条命令。

例: 查看当前日期。命令如下:

```
[globus@unknown globus]$ date
```

```
三 6月 2 14:36:32 CST 2004
```

重复执行最后一条命令:

```
[globus@unknown globus]$ !!
```

```
date
```

```
三 6月 2 14:37:11 CST 2004
```

### 4. 通过历史编号运行命令

当使用 history 命令时, 显示在终端屏幕上的每行左边的数字是命令编号, 命令编号的大小由最近输入的指定命令决定。用户为了使用正确的命令编号, 首先运行 history 命令确定命令编号。

除了使用方向键和!!重复执行命令之外, 可以使用另外两种方法。

- !命令编号: 重复执行编号的命令编号的命令。
- !字符串: 重复执行以“字符串”打头的上一条命令。

例: 重复执行以 d 打头的上一条命令。执行过程如下:

```
[globus@unknown globus]$ !d
```

```
date
```

```
三 6月 2 14:50:33 CST 2004
```

重复执行以 **da** 打头的上一条命令，执行过程如下：

```
[globus@unknown globus]$ !da
date
```

```
三  6月  2 14:55:20 CST 2004
```

重复执行以 **do** 打头的上一条命令，执行过程如下：

```
[globus@unknown globus]$ !do
dos2unix
```

使用 **history** 命令列出命令记录列表：

```
[globus@unknown globus]$ history
```

结果只能显示最后的 23 条命令记录：

```
557 history | grep ^d
558 history | grep "d"
559 history | grep -v ^d
560 history | grep -v '^d'
561 history | grep -v '^d'
562 history | grep -v '^da'
563 history | grep -v '^da'
564 vi vitest
565 grep -v '^r' vitest
566 grep '^r' vitest
567 grep v '^r' vitest
568 grep -v'^r' vitest
569 grep -v'^r' vitest
570 grep -v '^r' vitest
571 grep '^r' vitest
572 grep '^r' vitest
573 history | grep '^d'
574 history | grep '^d'
575 history
576 history | grep ' r'
577 history | grep ' d'
578 dos2unix
579 history
```

执行命令记录中的第 564 条记录，该命令为 **vi**，命令如下：

```
[globus@unknown globus]$ !564
```

结果执行 **vi vitest**，编辑 **vitest** 文件。

### 6.6.3 定义命令的别名

命令的别名对定义那些用户经常要使用而又不愿花费精力记住里面的细节的命令是非

常有用的。它对用一组有用的工具来优化命令也很有用。

例如，如果用户经常使用下面命令：

```
$ ls -l -d -i -a
```

那么就可以为该命令定义一个别名，如 `lll`。用为它定义的别名代替每次输入一长串选项。为了定义这样的别名，用户必需使用 `bash` 的 `alias` 命令。

命令的语法如下：

```
alias 别名='命令'
```

例：将命令 `ls -l -d -i -a` 取一个别名 `lll`。操作如下：

```
[globus@unknown globus]$ alias lll='ls -l -d -i -a'
```

每当用户输入 `lll` 命令时，它就会代替原来很长的命令而且能得到和从前一样的结果。如果想删除一条不想要的别名命令，可以使用 `bash` 的 `unalias` 命令。

执行 `lll` 命令，过程如下：

```
[globus@unknown globus]$ lll
2383896 drwx----- 6 globus globus 4096 6月 2 15:09
```

查看 `lll` 命令的细节，命令如下：

```
[globus@unknown globus]$ which lll
```

结果如下：

```
alias lll='ls -l -d -i -a'
/bin/ls
```

删除别名 `lll`，命令如下：

```
[globus@unknown globus]$ unalias lll
```

有很多别名对多数用户来说是很有用的。它们在用户登录时被写入执行文件，每次使用它们以后都会被保存起来。下面是一些也许是用户想要定义的别名：

```
alias ll='ls -l'
alias ls='ls -F'
```

如果用户是一个 `DOS` 用户，而且习惯于使用 `DOS` 文件命令，可以用 `alias` 命令定义如下的别名，使 `Linux` 使用起来就像 `DOS` 一样：

```
alias dir='ls'
alias copy='cp'
alias rename='mv'
alias md='mkdir'
alias rd='rmdir'
```

如果用户输入 `alias` 命令不带任何参数，`bash` 将为用户在屏幕上列出所有已定义的别名。下面是使用不带任何参数的 `alias` 命令得到的结果：

```
[globus@unknown globus]$ alias
alias l.='ls -d .* --color=tty'
alias ll='ls -l --color=tty'
alias ls='ls --color=tty'
alias mc='./usr/share/mc/bin/mc-wrapper.sh'
```



```
alias vi='vim'
alias which='alias | /usr/bin/which --tty-only --read-alias --show-dot
--show-tilde'
```

#### 6.6.4 改变 bash 提示符

bash 有两级提示符。第一级是 bash 在等待输入命令时用户看到的提示符，这也是通常所看到的提示符。

对于普通用户，bash 默认的第一级提示符是“\$”。如果不喜欢用“\$”作为提示符，用户可以用 bash 变量 PS1 来定制自己的提示符。

例：查看当前的提示符设置。

```
[globus@unknown globus]$ echo $PS1
[\u@\h \w]\$
```

例：如果将提示符设置成\$。命令如下：

```
[globus@unknown globus]$ PS1='\$'
```

```
$
```

可见提示符已经变为了\$符号，执行 ls 命令：

```
$ls
hello1.txt hello3.txt newdir3 testvinew vitestpart
hello2.txt hello4.txt newdir4 vitest
$
```

第二级提示符是当 shell 发现用户的命令不全，告诉用户还需要更多输入时的提示符。默认的第二级提示符是“>”。可以用 bash 变量 PS2 来定制用户的第二级提示符。

例：将二级提示符定制为 more info。命令如下：

```
$PS2="more info"
```

可以在命令提示符中显示系统制定好的特殊字符，这些特殊的字符可以在提示符中显示诸如时间、当前用户名、路径等信息。表 6-1 列出了常用的特殊字符。

表6-1 最常用的特殊字符

字符	含义
\l	显示该命令的使用次数
\#	显示当前命令使用的次数
\\$	使用\$作为提示符，如果您是root，那就显示#
\	显示\
\d	显示当前日期
\h	显示在shell上运行的主机名
\s	显示正在运行的shell名
\t	显示当前时间
\u	显示当前使用者的用户姓名
\W	显示当前工作目录
\w	显示当前工作路径

这些特殊字符可以组合起来形成一些有用的提示符，提供给用户一些有用的信息。下面是使用 PS1 改变提示符的几个例子。

```
$PS1="\d"
```

这条命令将在提示符中显示日期。提示符变为：

```
三 6月 02
```

这个例子就说明了引号的重要性。提示字符为：

```
三 6月 02PS1="\t\\"
```

将产生下面的提示符：

```
15:41:25\
```

使用系统默认的提示符，执行如下命令：

```
PS1="\u@\h \W]\$"
```

```
[globus@unknown globus]$
```

### 6.6.5 作业控制

作业控制是指 bash 控制正在运行进程的能力。bash 可以挂起某个进程，以后再恢复该进程的执行。bash 保存当前执行的作业清单。它可以用 jobs 命令进行显示。

为了挂起一个正在运行的进程，可以用 Ctrl+Z 键。使用 bg 命令可以恢复一个被挂起的后台进程，而 fg 命令可以恢复一个被挂起的前台进程。

当用户想在后台执行某条命令，却无意中使它在前台执行时，这些命令就很有用了。当命令在前台执行时，在该进程结束以前用户都无法从键盘输入任何命令。通常大多数命令的执行只需要几秒钟。如果命令执行时间过长，就最好使它进入后台作业以便能在前台继续向 bash 输入指令。

例如，想运行命令 slocate -u，该命令在前台执行，更新搜索数据库，根据文件系统的大小中文件的多少和系统中用户的多少，将会花去几分钟到几十分钟不等。如果用户已在前台执行这条命令，并想让它转入后台以便能在前台继续输入命令，可以用 Ctrl+Z 键将进程挂起，再在提示符下输入 bg 将 slocate 转入后台执行，执行过程如下：

如果当前用户不是超级用户，切换至超级用户。命令如下：

```
[globus@unknown globus]$su
```

```
Password:
```

```
[root@unknown globus]#
```

在前台更新搜索数据库。命令如下：

```
[root@unknown globus]# slocate -u
```

按 Ctrl+Z 键，挂起正在运行的程序，出现提示符：

```
[root@unknown globus]# slocate -u
```

```
[1]+ Stopped slocate -u
```

将数据库更新进程转入后台运行：

```
[root@unknown globus]# bg
```

```
[1]+ slocate -u &
```

用户就可以继续使用 bash 了。

可以使用 `jobs` 命令查看后台运行的进程:

```
[root@unknown globus]# jobs
[1]+  Running                  slocate -u &
```

当 `slocate` 执行完毕时, 会给出如下提示:

```
[1]+  Done                    slocate -u
```

### 6.6.6 定制 bash

在本章已经描述了很多定制 `bash` 的方法。但是用户所做的改变只对当前 `shell` 有效, 用户一旦退出系统, 所做的修改都将丢失。用户可以在 `bash` 的一个初始文件中储存它们, 使这些改变长期有效。

可以把每次进入 `bash` 都要执行的命令放在该文件中, 该文件中包含的内容通常是命令别名和初始变量。`bash` 的全局初始文件名为 `profile`, 该文件对系统中所有的用户都有效。每一个用户在其主目录下都有一个 `.bash_profile` 文件, 每次起动 `bash` 时, `bash` 都要从中读取信息, 文件中所有的命令都将被执行。下面这个文件是系统默认的 `profile` 文件, 该文件位于目录 `/etc` 下, 如果想向 `bash` 添加当前用户的定制信息, 必须把定制信息放在用户主目录的 `.bash_profile` 文件中。

```
# /etc/profile

# System wide environment and startup programs, for login setup

# Functions and aliases go in /etc/bashrc

pathmunge () {
    if ! echo $PATH | /bin/egrep -q "^(|:)$1($|:)" ; then
        if [ "$2" = "after" ] ; then
            PATH=$PATH:$1
        else
            PATH=$1:$PATH
        fi
    fi
}

# Path manipulation
if [ `id -u` = 0 ] ; then
    pathmunge /sbin
    pathmunge /usr/sbin
    pathmunge /usr/local/sbin
fi

pathmunge /usr/X11R6/bin after
unset pathmunge

# No core files by default
ulimit -S -c 0 > /dev/null 2>&1
```

```

USER="`id -un`"                                (^|:)$1($|:)" ; then
LOGNAME=$USER
MAIL="/var/spool/mail/$USER"
HOSTNAME=`/bin/hostname`
HISTSIZ=1000

if [ -z "$INPUTRC" -a ! -f "$HOME/.inputrc" ] ; then
    INPUTRC=/etc/inputrc
fi

export PATH USER LOGNAME MAIL HOSTNAME HI
for i in /etc/profile.d/*.sh ; do
    if [ -r "$i" ] ; then
        . $i
    fi
done

unset i

```

bash 的常用命令如表 6-2 所示。

表6-2 常用bash命令

命令	含义
alias	定义命令别名
bg	后台操作命令，强迫一个被挂起的进程在后台执行
cd	改变工作路径
exit	退出登录
fc	编辑当前命令记录列表，作用同history
fg	前台操作命令。强迫一个被挂起的进程在前台执行
help	提供bash各种命令使用的帮助信息
history	显示最近使用过的n条命令
kill	终止制定进程的执行
pwd	显示工作路径
unalias	删除命令的别名

上表列出的仅是 bash 最常使用的命令，为获取更多的信息请参看 bash 的 man page。

bash 有很多环境变量，表 6-3 列出了部分常用的环境变量及其含义：

表6-3 bash的环境变量

变量	含义
EDITOR, FCEDIT	默认的fc命令编辑器
HISTFILE	该文件用来存放历史命令
HISTSIZ	命令记录列表的大小

续表

变量	含义
HOME	使用者起始目录的绝对路径名
OLDPWD	上一次操作的绝对路径名
PATH	bash查找命令文件的目录列表
PS1	保存bash命令行提示符
PS2	保存bash的第二提示符
PWD	保存当前工作目录的绝对路径名
SECONDS	bash启动所需的秒数

### 6.6.7 bash 的作业(job)控制

bash 把每次执行的命令当作一个作业，并为它们分配作业号，如果有管道，则将这些命令认为是一个作业。bash 可以用作业号来控制作业的运行、挂起、结束。需注意的是作业不等于进程。有关作业控制的符号和命令如下所示。

- **&:** 它用在命令行的最后，作用是将命令放到后台继续运行，立即返回 bash 的提示符。例如

```
[root@unknown /]# ls -lR / &> lsfile &
```

```
[1] 13565
```

```
[root@unknown /]#
```

这行命令是将把/以下的所有目录中的文件送到lsfile，因为权限问题所以会产生错误信息，故用&>。在最后添加了&，所以不等这条命令结束，就返回提示符。输出[1] 13565，其中[]中的1是作业号，后面的数字13565是进程号。

如果没有在命令后面加&，要想在命令执行中使用提示符，可以用Ctrl+Z键先将命令挂起，等以后再用bg转入后台执行。

- **jobs 命令:** 这条命令的作用是显示系统中所有的作业，如首先在后台执行ls -lR / &> lsfile &:

```
[root@unknown /]# ls -lR / &> lsfile &
```

```
[1] 13578
```

从结果可知，作业号为1，进程号为13578。

使用jobs命令查看。命令如下：

```
[root@unknown /]# jobs
```

```
[1]+  Running                  ls --color=tty -lR / >&lsfile &
```

在执行另一个后台作业locate -u &:

```
[root@unknown /]# locate -u &
```

```
[2] 13579
```

从结果可知，作业号为2，进程号为13579。

使用jobs命令查看。命令如下：

```
[root@unknown /]# jobs
```

```
[1]-  Running                  ls --color=tty -lR / >&lsfile &
```

```
[2]+  Running                  locate -u &
```

可以看出有两个作业在运行，作业号分别为 1 和 2。

- **fg** 命令的作用是将作业送到前台。

**fg** 的语法为：

**fg** 作业号

例如：在后台运行 `ls -lR / &> lsfile &`。

```
[root@unknown /]# ls -lR / &> lsfile &
```

```
[1] 13637
```

从结果可知，该作业的作业号为 1。

将作业 1 转到前台：

```
[root@unknown /]# fg 1
```

```
ls --color=tty -lR / >&lsfile
```

可见用户看不见提示符。

运行一段时间之后，作业结束，再次出现提示符。

```
[root@unknown /]#
```

- **bg** 命令的作用是将作业送到后台。

- **kill** 命令是将作业立即结束。

用法是：

**kill** %作业号。

例：

```
[root@unknown /]# cat > mytext
```

```
[1]+ Stopped cat >mytext
```

```
< Ctrl+z>
```

```
$jobs
```

```
[root@unknown /]# jobs
```

```
[1]+ Stopped cat >mytext
```

```
[root@unknown /]# bg
```

```
[1]+ cat >mytext &
```

```
[root@unknown /]# jobs
```

```
[1]+ Stopped cat >mytext
```

```
[root@unknown /]# kill %1
```

```
[1]+ Stopped cat >mytext
```

上面的例子中，**bg** 将作业 1 发送到后台执行（不是挂起），但因为 `cat > text` 需要从键盘上获得输入，放到后台无法运行，因此被自动挂起。

## 思考与练习

- (1) 在 Linux 中，有哪几种常见的 shell？
- (2) 在 Linux shell 中，有哪几种重定向符号，它们的功能分别是什么？
- (3) Linux shell 有哪几种预定义变量，它们的含义是什么？
- (4) **bash** 的作业有哪几个状态，如何切换它们的状态？

# 第 7 章 常用网络命令

Linux 操作系统是与生俱来的网络操作系统，它的网络功能十分强大。Red Hat Linux 9 提供了一套简单高效的网络配置工具，这些工具可以用来设置网卡的 IP 地址、MAC 地址、DNS 服务器地址、内核 IP 路由表、网络状态检测和路由跟踪命令。同时也提供了许多简单有效的基于文本的网络客户端程序供用户使用。这些客户端程序主要包括 http、ftp、telnet、邮件、安全远程登录和安全文件传输的客户端程序。

本章将以实例的形式讲述 Linux 提供基于文本的网络设置和跟踪程序的使用，而且还介绍了常用的网络服务的客户端程序，这将有利于读者进一步熟悉 Linux 的网络操作，并且为后继章节的学习奠定一个良好的基础。

## 7.1 网络诊断和设置命令

Linux 不仅内置了 TCP/IP 网络的所有协议和应用程序，而且提供了一套高效的网络诊断和设置命令，这些命令分别是：ping、ifconfig、ifup、ifdown、netstat、route、traceroute、host 和 nslookup。如果用户掌握了这些命令，那么就能够很轻松地管理网络。

### 7.1.1 检测主机的连通性 —— ping 命令

功能说明：ping 指令会使用 ICMP 传输协议，发出要求回应的信息，若远端主机的网络没有问题，就会回应该信息，因而能得知该主机运作正常。

例如：有时用户想从网络上的某台主机上下载文件，但是又不知道那台主机是否开着，就可以使用 ping 命令检测。

ping 命令的语法为：

ping [-dfnqrRv][[-c<完成次数>][[-i<间隔秒数>][[-I<网络接口>][[-l<前置载入>][[-p<范本样式>][[-s<数据包大小>][[-t<存活数值>][主机名称或 IP 地址]

ping 的选项如表 7-1 所示。

表 7-1 ping 命令的选项

选项	描述
-d	使用Socket的SO_DEBUG功能
-c<完成次数>	设置完成要求回应的次数
-f	极限检测
-i<间隔秒数>	指定收发信息的间隔时间
-I<网络接口>	使用指定的网络接口送出数据包
-l<前置载入>	设置在送出要求信息之前，事先发出的数据包
-n	只输出数值
-p<范本样式>	设置填满数据包的范本样式
-q	不显示指令执行过程，开头和结尾的相关信息除外

续表

选项	描述
-r	忽略普通的Routing Table, 直接将数据包送到远端主机上
-R	记录路由过程
-s<数据包大小>	设置数据包的大小
-t<存活数值>	设置存活数值TTL的大小
-v	详细显示指令的执行过程

例：使用 ping 命令判断 IP 地址为 210.77.7.131 是否已经打开。命令如下：

```
[globus@wang globus]$ ping 210.77.7.131
```

执行结果如下：

```
PING 210.77.7.131 (210.77.7.131) 56(84) bytes of data.
64 bytes from 210.77.7.131: icmp_seq=1 ttl=128 time=0.561 ms
64 bytes from 210.77.7.131: icmp_seq=2 ttl=128 time=0.555 ms
64 bytes from 210.77.7.131: icmp_seq=3 ttl=128 time=0.536 ms
64 bytes from 210.77.7.131: icmp_seq=4 ttl=128 time=0.577 ms
64 bytes from 210.77.7.131: icmp_seq=5 ttl=128 time=0.690 ms
64 bytes from 210.77.7.131: icmp_seq=6 ttl=128 time=0.576 ms
...
```

该命令默认不停地对 IP 地址 (210.77.7.131) 发出 ICMP 请求, 结果显示每一次回应的时间, 例如第 1 次为 0.561 ms, 第 2 次为 0.555 ms。用户可以按 Ctrl+c 键终止 ping 程序的执行。

**注意：**使用 ping 程序的输出中可能会出现包丢失现象, 如果包丢失的比例很大, 说明网络的连通性差, 如果没有丢包现象, 说明网络的连通性好。

在 ping 的输出中, 每一行的 time 值越小, 说明本主机与远程主机的传输速率越高。

当用户对他们自己的计算机进行了网卡、IP 地址、DNS 或者网关设置之后, 通常也需要使用 ping 来验证设置的正确性, 首先 ping 本机的 IP 地址, 如果有回应, 表明对网卡设置正确。然后再 ping 同一子网的其他机器, 如果有回应, 表明局域网的连接是没有异常。再 ping 不同子网的某台计算机的 IP 地址, 如果有回应, 表明网关设置正确。最后 ping 不同子网的某台计算机的域名, 如果有回应, 表明 DNS 的设置正确。否则可能有各种导致 ping 失败的原因, 用户需要根据实际情况逐步发现潜在的问题, 然后解决这些问题, 最后再用 ping 命令进行连通性测试。

例：使用 ping 命令判断 IP 地址为 210.77.7.131 是否已经打开, 仅作三次回应。命令如下：

```
[globus@wang globus]$ ping -c3 210.77.7.131
```

结果如下：

```
PING 210.77.7.131 (210.77.7.131) 56(84) bytes of data.
64 bytes from 210.77.7.131: icmp_seq=1 ttl=128 time=0.554 ms
64 bytes from 210.77.7.131: icmp_seq=2 ttl=128 time=0.576 ms
```



```

64 bytes from 210.77.7.131: icmp_seq=3 ttl=128 time=0.581 ms

--- 210.77.7.131 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1998ms
rtt min/avg/max/mdev = 0.554/0.570/0.581/0.022 ms

```

可以看到, ping 首先显示了 3 次回应, 然后给出了 ping 的统计信息。在这个例子中, 共传输了 3 个包, 接收了 3 个回应, 没有丢失任何包, 共花费 1998ms 时间。

### 7.1.2 显示网络状态 —— netstat 命令

语 法: netstat [-acCeFghilMnNoprstuvVwx][[-A<网络类型>]][--ip]

功能说明: 利用 netstat 指令可让你得知整个 Linux 系统的网络使用情况。netstat 命令是一个监控 TCP/IP 网络的非常有用的工具, 它可以显示路由表、实际的网络连接以及每一个网络接口设备的状态信息。

netstat 命令的选项如表 7-2 所示。

表 7-2 netstat 命令的选项

选项	描述
-a或--all	显示所有连接的Socket
-A<网络类型>或--<网络类型>	列出该网络类型连接中的相关地址
-c或--continuous	持续列出网络状态
-C或--cache	显示路由器配置的cache信息
-e或--extend	显示网络其他相关信息
-F或--fib	显示FIB
-g或--groups	显示多重广播功能群组组员名单
-h或--help	在线帮助
-i或--interfaces	显示网络接口信息表单
-l或--listening	显示监控中的服务器的Socket
-M或--masquerade	显示伪装的网络连线
-n或--numeric	直接使用IP地址, 而不使用域名
-N或--netlink或--symbolic	显示网络硬件外围设备的符号连接名称
-o或--timers	显示计时器
-p或--programs	显示正在使用socket的程序识别码和程序名称
-r或--route	显示系统的路由表
-s或--statistic	显示网络工作信息统计表
-t或--tcp	显示TCP传输协议的连接状况
-u或--udp	显示UDP传输协议的连接状况
-v或--verbose	显示指令执行过程
-V或--version	显示版本信息
-w或--raw	显示RAW传输协议的连接状况
-x或--unix	此选项显示UNIX网络协议, 其效果和指定“-A unix”选项相同
--ip或--inet	此选项显示互联网协议 (TCP/IP), 其效果和指定“-A inet”选项相同

## 1. 使用 netstat 显示网络连接信息

显示当前系统的网络连接情况，命令如下：

```
[globus@wang globus]$ netstat
```

输出结果为：

Active Internet connections (w/o servers)

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State
tcp	0	2	210.77.7.97:telnet	210.77.7.131:4876	ESTABLISHED
tcp	0	1	210.77.7.97:32771	61.135.132.100:smtp	SYN_SENT
tcp	0	0	wang:32770	wang:smtp	TIME_WAIT

Active UNIX domain sockets (w/o servers)

Proto	RefCnt	Flags	Type	State	I-Node	Path
unix	11	[ ]	DGRAM		2129	/dev/log
unix	2	[ ]	DGRAM		2957	
unix	2	[ ]	DGRAM		2918	
unix	2	[ ]	DGRAM		2693	
unix	2	[ ]	DGRAM		2653	
unix	2	[ ]	DGRAM		2639	
unix	2	[ ]	DGRAM		2584	
unix	2	[ ]	DGRAM		2321	
unix	2	[ ]	DGRAM		2179	
unix	2	[ ]	DGRAM		2137	

总的来说，netstat 的输出结果可以分为两个部分，一个是 Active Internet connections，称为有源 TCP 连接，另一个是 Active UNIX domain sockets，称为有源 Unix 域套接口。

在上面的输出结果中，第一部分有 3 个输出结果，显示有源 TCP 连接的情况，其中 Proto 显示连接使用的协议；Recv-Q 和 Send-Q 分别表示接收和发送请求的次数；Local Address 和 Foreign Address 分别表示该连接的本地和远程 IP 地址；State 分别表示该连接的当前状态。

第二部分的输出结果显示的是 Unix 域套接口的连接情况。Proto 显示连接使用的协议；RefCnt 表示连接到本套接口上的进程号；Types 显示套接口的类型；State 显示套接口当前的状态；Path 表示连接到套接口的其他进程使用的路径名。

## 2. 使用 netstat -r 显示系统的路由表

例：列出当前主机的路由表信息。执行过程如下：

```
[globus@wang globus]$ netstat -r
```

结果如下：

Kernel IP routing table

Destination	Gateway	Genmask	Flags	MSS Window	irtt	Iface
210.77.6.0	*	255.255.254.0	U	0 0	0	eth0
169.254.0.0	*	255.255.0.0	U	0 0	0	eth0

```
127.0.0.0      *                255.0.0.0      U          0 0          0 lo
default        210.77.7.254    0.0.0.0      UG         0 0          0 eth0
```

注意：-n 选项令 netstat 命令以点分四段式的形式输出 IP 地址，而不是使用主机名和域名。如果想避免通过网络查找地址（比如避开 DNS 或 NIS 服务器），这一选项特别有用。

netstat 输出结果中，第 1 列是所要访问的目标地址，如果它的值为 default 或者是 0.0.0.0 时，表示该条目为默认路由。第 2 列表示要到达第一列的目标地址，对应的网关地址，如果没有使用网关，就会出现一个星号（\*）或者 0.0.0.0 表示。第 3 列展示路由的子网掩码，在为具体的 IP 地址找出最恰当的路由时，内核将查看路由表内的所有条目，在对找到的路由与目标路由比较之前，将对 IP 地址和 genmask 进行按位“与”计算。第 4 列显示了不同的路由标记。第 7 列表示要到达目标 IP 地址时，应通过哪一个网络接口，如 eth0 表示用第一张网卡，eth1 表示用第二张网卡，lo 表示环回接口。

第 4 列的标记说明如表 7-3 所示。

表 7-3 netstat 命令输出中的路由标记

选项	描述
G	该条目为网关
U	准备使用的接口处于“活动”状态
H	通过该路由，只能到达一台主机
D	如果路由表的条目是由 ICMP 重定向消息生成的，就会设置这个标记
M	如果路由表条目已被 ICMP 重定向消息修改，就会设置这个标记

如果使用加上 -n 选项，显示结果如下：

```
[globus@wang globus]$ netstat -rn
Kernel IP routing table
Destination      Gateway          Genmask         Flags   MSS Window  irtt Iface
210.77.6.0       0.0.0.0         255.255.254.0   U        0 0        0 eth0
169.254.0.0      0.0.0.0         255.255.0.0     U        0 0        0 eth0
127.0.0.0        0.0.0.0         255.0.0.0       U        0 0        0 lo
0.0.0.0          210.77.7.254   0.0.0.0         UG       0 0        0 eth0
```

### 3. 使用 netstat -i 显示接口特性

在随 -i 标记一起调用时，netstat 将显示网络接口的当前配置特性。除此以外，如果调用时还带上 -a 选项，它还将输出内核中所有接口，并不只是当前配置的接口。

例：显示系统中所有网络接口的配置信息。命令如下：

```
[globus@wang globus]$ netstat -i
输出结果如下：
Kernel Interface table
Iface MTU Met RX-OK RX-ERR RX-DRP RX-OVR TX-OK TX-ERR TX-DRP TX-OVR Flg
eth0 1500 0 5652 0 0 0 128 0 0 0 BM RU
lo 16436 0 1152 0 0 0 1152 0 0 0 LR U
```

MTU 和 Met 字段表示的是接口的最大传输单元和度量值；RX 和 TX 两列表示已经准确无误地收发了多少数据包 (RX-OK/TX-OK)、产生了多少错误 (RX-ERR/TX-ERR)、丢弃了多少包 (RX-DRP/TX-DRP)，由于误差而遗失了多少包 (RX-OVR/TX-OVR)；最后一列显示了这个接口设置的标记，在利用 ifconfig 显示接口配置时，这些标记都采用一个字母。

netstat -i 输出中 Flg 列的可能取值如表 7-4 所示。

表 7-4 netstat-i 输出中的 Flg 列的取值

Flg取值	描述
B	已经设置了一个广播地址
L	该接口是一个回送设备
M	接收所有数据包 (混杂模式)
N	避免跟踪
O	在该接口上，禁用ARP
P	这是一个点到点链接
R	接口正在运行
U	接口处于“活动”状态

#### 4. 显示链接信息

netstat 支持用于显示活动或非活动套接字的选项集。选项 -t、-u、-w 和 -x 分别表示 TCP、UDP、RAW 和 UNIX 套接字连接。如果使用 -a 标记，还会显示出等待连接 (也就是说处于监听模式) 的套接字。这样就可以得到一份服务器清单，当前所有运行于系统中的所有服务器都会被显示出来。

例：显示系统中活动的 TCP 连接。命令如下：

```
[globus@wang globus]$ netstat -t
```

显示结果如下：

Active Internet connections (w/o servers)

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State
tcp	0	2	210.77.7.97:telnet	210.77.7.131:4876	ESTABLISHED

结果表明，只有一个来自 210.77.7.131 的 telnet 连接处于活动状态。

例：显示系统中所有的 TCP 连接，命令如下：

```
[globus@wang globus]$ netstat -ta
```

显示结果如下：

Active Internet connections (servers and established)

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State
tcp	0	0	*:32768	*:*	LISTEN
tcp	0	0	wang:32769	*:*	LISTEN
tcp	0	0	*:sunrpc	*:*	LISTEN
tcp	0	0	*:ssh	*:*	LISTEN
tcp	0	0	wang:ipp	*:*	LISTEN

```
tcp      0      0 *:telnet          *:*          LISTEN
tcp      0      0 wang:smtp         *:*          LISTEN
tcp      0      2 210.77.7.97:telnet 210.77.7.131:4876 ESTABLISHED
```

上面的输出表明部分服务器处于等待监听状态。利用-a 选项，netstat 还会显示出所有的套接字。注意，根据端口号可以判断出一条连接是否是外出连接。对呼叫方主机来说，列出的端口号应该一直是一个整数，而对众所周知服务（well known service）端口，且正在使用中的被呼叫方来说，netstat 采用的则是取自/etc/services 文件的象征性服务名。

7.1.3 显示和配置网络接口 —— ifconfig 命令

无论是 Linux 自动安装还是手工安装，Linux 都会向你询问有关网络的问题并配置相关的软件。这个用于配置网卡的基本命令就是 ifconfig。

在执行 ifconfig 命令后，系统将在内核表中设置必要的参数，这样 Linux 就知道如何与网卡通信。

语 法：ifconfig [网络接口][down up -allmulti -arp -promisc][add<地址>][del<地址>][<硬件地址>][io\_addr][irq][media<网络媒介类型>][mem\_start<内存地址>][metric<数目>][mtu<字节>][netmask<子网掩码>][tunnel<地址>][-broadcast<地址>][-pointopoint<地址>][IP 地址]

功能说明：ifconfig 可设置网络接口的状态，或是显示网络接口当前的设置。  
ifcong 常见的选项如表 7-5 所示。

表 7-5 ifcong 命令的常见选型

选项	描述
-a	显示所有网络接口的状态信息
add <地址>	设置网络接口IPv6的IP地址
del <地址>	删除网络接口IPv6的IP地址
down	关闭指定的网络接口
<硬件地址>	设置网络接口的类型与硬件地址
io_addr	设置网络接口的I/O地址
irq	设置网络接口的IRQ（中断号）
media <网络媒介类型>	设置网络接口的媒介类型
mem_start<内存地址>	设置网络接口在内存所占用的起始地址
metric <数目>	指定在计算数据包的转送次数时，所要加上的数目
mtu <字节>	设置网络接口的MTU
netmask <子网掩码>	设置网络接口的子网掩码
tunnel <地址>	建立IPv4与IPv6之间的隧道通信地址
up	启动指定的网络接口
-broadcast<地址>	将要送往指定地址的数据包当成广播数据包来处理
-pointopoint <地址>	与指定地址的网络接口建立直接连线，此模式具有保密功能
-promisc	关闭或启动指定网络接口的promiscuous模式
[IP地址]	指定网络接口的IP地址
[网络接口]	指定网络接口的名称

## 1. 显示网络接口的状态

如果 `ifconfig` 不带参数，它将会显示当前活动接口的状态。如果它仅带一个“网络接口”参数，它仅显示指定接口的状态。如果仅带一个 `-a` 选项，它将显示包括非活动接口在内的所有网络接口的信息。

例：显示系统中所有活动接口的状态信息。命令如下：

```
[globus@wang globus]$ /sbin/ifconfig
```

结果如下：

```
eth0      Link encap:Ethernet  HWaddr 52:54:AB:B0:1F:1A
          inet addr:210.77.7.97  Bcast:210.77.7.255  Mask:255.255.254.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:12092 errors:0 dropped:0 overruns:0 frame:0
          TX packets:835 errors:0 dropped:0 overruns:0 carrier:0
          collisions:149 txqueuelen:100
          RX bytes:1298403 (1.2 Mb)  TX bytes:105014 (102.5 Kb)
          Interrupt:11 Base address:0xc000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:1152 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1152 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:7632092 (7.2 Mb)  TX bytes:7632092 (7.2 Mb)
```

从结果可以看出，系统中有两个活动的网络接口，以 `eth0` 为例说明显示各字段的含义。

- `eth0`：表示该接口为以太网接口。
- `HWaddr 52:54:AB:B0:1F:1A`：MAC（网卡的物理地址）为 `52:54:AB:B0:1F:1A`。
- `inet addr:210.77.7.97`：它的 IP 地址为 `210.77.7.97`。
- `Bcast:210.77.7.255`：其广播地址为 `210.77.7.255`。
- `Mask:255.255.254.0`：子网掩码为 `255.255.254.0`。
- `UP`：表示当前接口为活动接口。
- `BROADCAST`：表示支持广播。
- `MULTICAST`：表示支持多播。
- `MTU:1500`：表示一次最大可传输 1500 个字节。
- `RX packets:12092`：表示已经接收了 12092 个数据包。
- `TX packets:835`：表示已发送了 835 个数据包。
- `collisions:149`：表示发生了 149 次冲突。
- `X bytes:1298403 (1.2 Mb)`：表示已经接收了 1.2 Mb 数据。
- `TX bytes:105014 (102.5 Kb)`：表示已经发送了 102.5 Kb 的数据。

- Interrupt:11: 表示使用的中断号为 11。
- Base address:0xc000: 表示基地址为 0xc000。

lo: 表示该接口为环回接口。

注意: eth0 表示系统中的第一块网卡, eth1 表示系统中的第二块网卡, 以此类推。

lo 表示系统中的环回接口。

## 2. 设置网络的 IP 地址

ifconfig 可以用来设置网卡的 IP 地址、子网掩码和广播地址, 其语法如下:

ifconfig eth0 IP 地址 netmask 子网掩码 broadcast 子网掩码

该命令的作用是设置网卡 eth0 的 IP 地址、网络掩码和网络的本地广播地址。同样的方式可以用来配置 eth1, eth2 等等, 通常 netmask 和 broadcast 只需要设置一个。

例: 将 eth0 (第一块网卡) 的 IP 地址设置成 192.168.1.192。命令如下:

```
[root@unknown root]# ifconfig eth0 192.168.1.192
```

使用 ifconfig 查看 eth0 修改后的 IP 地址配置情况:

```
[root@unknown root]# ifconfig eth0
```

结果如下:

```
eth0      Link encap:Ethernet  HWaddr 00:0C:29:B6:CB:D0
          inet addr:192.168.1.192  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1401 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:100398 (98.0 Kb)  TX bytes:168 (168.0 b)
          Interrupt:10 Base address:0x1080
```

结果表明, eth0 的 IP 地址已经更改为 192.168.1.192。

注意: 只有 root 用户才能改变系统的 IP 地址。

如果用户也可以只指定 IP 地址, 子网掩码和广播地址将按 IP 的类别自动确定。

使用 ifconfig 设置的 IP 地址在下次系统重启时不会有效。

如果希望永久改变接口的 IP 地址, 需要使用 Red Hat Linux 提供的 netconfig 命令进行文本方式配置, 或者使用 redhat-config-network 命令进行图形化工具配置。

有时需要为某个设备接口配置多个 IP 地址, 办法是使用设备别名。例如, eth0 设备可以有 eth0, eth0:0, eth0:1 等多个别名, 每个都可以有一个独立的 IP 地址:

```
ifconfig eth0 192.168.1.190 netmask 255.255.255.0
```

```
ifconfig eth0:0 192.168.1.191 netmask 255.255.255.0
```

这样, 192.168.1.190 和 192.168.1.191 都会被绑定在 eth0 设备上, 使用同样的网络设备, 不同的 IP 地址。

### 3. 更改网卡的物理地址

每一块网卡都有一个 MAC 地址（也称为网卡的物理地址、以太网地址），而且是互不相同的，网卡的物理地址的长度为 48bit。MAC 地址是生产厂家定的，每个网卡拥有的惟一地址。

不过用户可以以 root 身份手工改动网卡的 MAC 地址，只要在 /etc/rc.d/init.d/network 文件中加入如下一行：

```
ifconfig eth0 hw ether xx:xx:xx:xx:xx:xx
```

然后重启，再用 ifconfig 命令查看接口 eth0，会发现网卡的 MAC 地址已经变成 xx:xx:xx:xx:xx:xx 了。其中 x 表示一位 16 进制数。

例：将 eth0 的原始 MAC 地址为 00:0C:29:B6:CB:D0，将更改为 00:E0:4C:60:07:DD。

```
ifconfig eth0 hw ether 00:E0:4C:60:07:DD
```

使用 ifconfig 查看 eth0 的物理地址：

```
[root@unknown root]# ifconfig eth0
```

结果如下：

```
eth0      Link encap:Ethernet  HWaddr 00:E0:4C:60:07:DD
          inet addr:192.168.1.190  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2087 errors:0 dropped:0 overruns:0 frame:0
          TX packets:54 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:168043 (164.1 Kb)  TX bytes:7178 (7.0 Kb)
          Interrupt:10 Base address:0x1080
```

结果表明第一块网卡的 MAC 地址已经更改为 00:E0:4C:60:07:DD。

### 4. 停止/启动网络接口

使用如下命令停止网卡：

```
Ifconfig 网络接口 down
```

使用如下命令启动网卡：

```
Ifconfig 网络接口 up
```

例：停止第一块网卡。命令如下：

```
[root@unknown root]# ifconfig eth0 down
```

启动第一块网卡，命令如下：

```
[root@unknown root]# ifconfig eth0 up
```

注意：使用 ifconfig 命令将能对网络接口进行设置和调试，当网络设置出现故障时，ifconfig 是排除故障的有力工具之一。

在 Linux 中提供两个命令 ifup 和 ifdown 与 ifconfig up 和 ifconfig down 的功能分别等价。



### 7.1.4 网络环境配置工具 —— netconfig 命令

Red Hat Linux 9 提供了 netconfig 网络环境配置工具，它具有互动式的问答界面，让用户轻易完成网络环境的设置。Netconfig 是一个基于文本的全屏幕交互网络配置程序。因此它可以在任何模式下运行。

在 shell 提示符下运行 `/usr/sbin/netconfig`，出现如图 7-1 所示的 netconfig 窗口。它询问用户是否进行网络设置，如果选 No 按钮，不进行任何网络设置，并且退出 netconfig；否则选 Yes 按钮，则进行 netconfig 的主界面，如图 7-2 所示。

在图 7-2 所示的界面里，如果需要使用 DHCP（动态主机配置）协议，动态地获得一个 IP 地址，则需要选中 Use dynamic IP configuration 复选框，然后将焦点移至 OK 按钮，并按回车键，保存设置。否则用户需要在 IP Address 文本框中输入系统的 IP 地址，在 Netmask 文本框中输入子网掩码，在 Default gateway(IP)文本框中输入网关的 IP 地址。在 Primary nameserver 文本框中输入域名服务器地址。然后按 Tab 键到 OK 按钮上，再按回车键确认并返回到 shell。

如果用户对 IP 地址、子网掩码、默认网关和域名服务器地址有疑问，请与你的网络管理员联系。



图 7-1 netconfig 的启动界面

注意：使用 netconfig 命令对网络接口进行设置后，不会立即生效，只有当下一次启动系统时候才会生效。

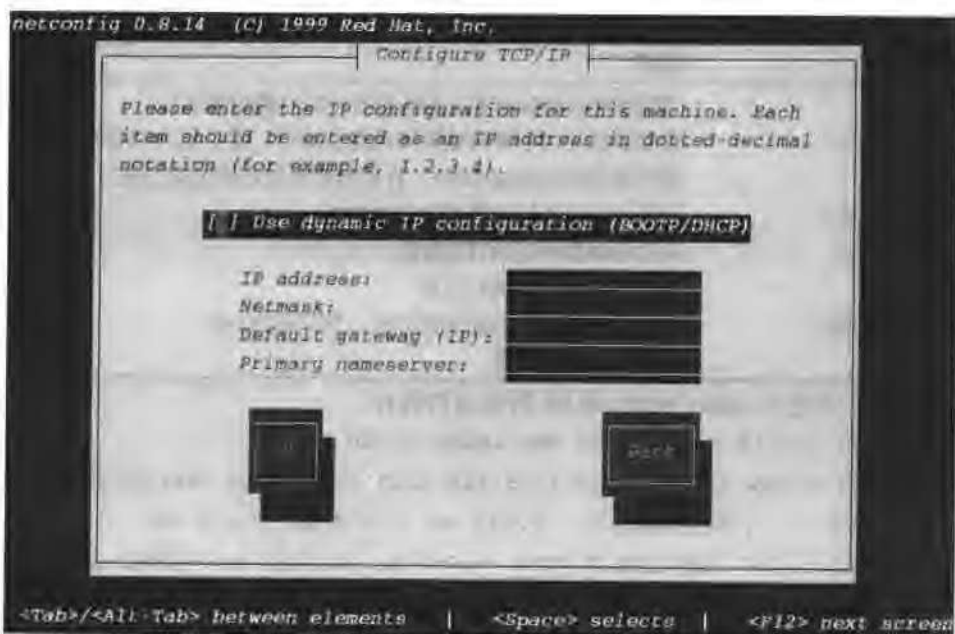


图 7-2 netconfig 的主界面

### 7.1.5 显示数据包到主机间的路径 —— traceroute 命令

Internet 是网络硬件的一个庞大而且复杂的集合体，它们通过网关连接在一起。跟踪一个数据包的路由过程是困难的，traceroute 程序利用了 IP 协议的存活时间字段，并且试图在该数据包到达某远程主机的过程中，从到达的网关提取 ICMP TIME\_EXCEEDED 响应。从而达到跟踪包路由的目的。

语 法：traceroute [-dFlnrvx][<-f<存活数值>][<-g<网关>...][<-i<网络界面>][<-m<存活数值>][<-p<通信端口>][<-s<来源地址>][<-t<服务类型>][<-w<超时秒数>][主机名称或 IP 地址][数据包大小]

功能说明：traceroute 指令让用户追踪网络数据包的路由，它的惟一一个必需的参数是目的主机的域名或者 IP 地址。默认数据包大小是 40Bytes，用户可另行设置。

Traceroute 命令的可选项如表 7-6 所示。

表 7-6 traceroute 命令的常见选型

选项	描述
-d	启用socket级的调试功能
-f<存活数值>	设置第一个检测数据包的存活数值TTL的大小
-F	设置不要分片位
-g<网关>	设置来源路由网关，最多可设置8个
-i<网络接口>	使用指定的网络接口送出数据包
-I	使用ICMP回应取代UDP数据报
-m<存活数值>	设置检测数据包的最大存活数值TTL的大小

续表

选项	描述
-n	直接使用IP地址而非主机名称显示每一跳的地址
-p<通信端口>	设置测试用的UDP传输协议的通信端口, 默认值为33434
-r	忽略普通的routing Table, 直接将数据包送到远端主机上
-s<来源地址>	设置本地主机送出数据包的IP地址
-t<服务类型>	设置检测数据包的TOS数值
-v	详细显示指令的执行过程
-w<超时秒数>	设置等待远端主机回应的的时间, 默认值为5秒
-X	启用IP校验和检查

例: 跟踪从本地到 **www.iscas.ac.cn** 所经过的路由。

```
[root@john root]# traceroute www.iscas.ac.cn
traceroute to www.iscas.ac.cn (159.226.4.215), 30 hops max, 38 byte packets
 1  192.168.1.1 (192.168.1.1)  0.846 ms  0.676 ms  0.676 ms
 2  159.226.5.126 (159.226.5.126)  4.257 ms  21.053 ms  4.952 ms
 3  159.226.5.58 (159.226.5.58)  10.210 ms  11.203 ms  6.284 ms
 4  159.226.4.215 (159.226.4.215)  97.362 ms  3.989 ms  5.268 ms
```

从上面结果可以看出, 首先从本机到达 192.168.1.1 (本机的默认网关), 然后经过 159.226.5.126, 再经过 159.226.5.58, 最后到达 159.226.4.215, 该主机就是 traceroute 应该到达的目的地 **www.iscas.ac.cn**。

### 7.1.6 与 Internet 域名服务器交互查询 —— nslookup 命令

nslookup 命令是一个用于查询 Internet 域名服务器的程序。如果给定的是域名, 它将返回对应的 IP 地址, 如果给定是 IP 地址, 则返回对应的域名。nslookup 有两种工作模式: 交互式和非交互式。在交互式模式中, nslookup 允许用户向域名服务器进行多个主机 IP 地址和域名的互查, 同时也允许打印一个域名的主机列表。在非交互模式中, nslookup 允许用户向域名服务器进行一个主机 IP 地址和域名的查询。

例如: 进行如下的交互式启动 nslookup 程序, 首先查询 **www.tsinghua.edu.cn** 和 **www.xaut.edu.cn** 对应的 IP 地址, 然后查询 61.135.132.125 和 159.226.7.162 对应的域名。

首先启动 nslookup, 命令如下:

```
[globus@unknown globus]$ nslookup
```

```
Note: nslookup is deprecated and may be removed from future releases.
Consider using the 'dig' or 'host' programs instead. Run nslookup with
the '-sil[ent]' option to prevent this message from appearing.
```

```
>
```

它出现了提示符“>”, 用户可以在 nslookup 的提示符后输入域名或者 IP 地址, 域名服务器分别进行名字解析和反向解析。操作如下:

```
> www.tsinghua.edu.cn
```

```
Server:      159.226.5.65
```

```
Address: 159.226.5.65#53
```

```
Non-authoritative answer:
```

```
Name: www.tsinghua.edu.cn
```

```
Address: 166.111.4.100
```

结果显示: 使用的域名服务器为 159.226.5.65, 端口号为 53, 域名 www.tsinghua.edu.cn 对应的 IP 地址为 166.111.4.100。

```
> www.xaut.edu.cn
```

```
Server: 159.226.5.65
```

```
Address: 159.226.5.65#53
```

```
Non-authoritative answer:
```

```
Name: www.xaut.edu.cn
```

```
Address: 202.200.112.6
```

查询结果表明 www.xaut.edu.cn 主机的 IP 地址为 202.200.112.6。

反向查询如下:

```
> 162.105.203.114
```

```
Server: 159.226.5.65
```

```
Address: 159.226.5.65#53
```

```
Non-authoritative answer:
```

```
114.203.105.162.in-addr.arpa name = e.pku.edu.cn.
```

```
114.203.105.162.in-addr.arpa name = e.pku.cn.
```

```
Authoritative answers can be found from:
```

```
105.162.in-addr.arpa nameserver = ns.pku.edu.cn.
```

```
105.162.in-addr.arpa nameserver = dns.edu.cn.
```

```
105.162.in-addr.arpa nameserver = ns2.cuhk.hk.
```

```
105.162.in-addr.arpa nameserver = pkuns.pku.edu.cn.
```

```
105.162.in-addr.arpa nameserver = sun1000e.pku.edu.cn.
```

```
ns2.cuhk.hk internet address = 137.189.6.21
```

可见同一个 IP 地址对应多个域名, 如 e.pku.edu.cn 和 e.pku.cn。

在提示符 “>” 后输入 exit, 按回车键后退出 nslookup 程序。

注意: 在 Red Hat Linux 9 中 nslookup 已经是过时的域名查询客户端程序, 用户可以使用 dig 或者 host 代替 nslookup。

例: 使用 host 确定 www.tsinghua.edu.cn 对应的 IP 地址。命令如下。

```
[globus@unknown globus]$ host www.tsinghua.edu.cn
```

```
www.tsinghua.edu.cn has address 166.111.4.100
```

可见其结果与 nslookup 的查询结果相同。

7.1.7 显示和操作 IP 路由表 —— route 命令

route 命令对内核的 IP 路由表进行操作。它主要对已配置的接口（如通过 ifconfig 程序配）的主机或网络设置静态路由。

route 命令的用法如下：

```
route [-CFvnee]
route [-v] [-A family] add [-net|-host] target [netmask Nm] [gw Gw] [metric N] [mss M] [window W] [irtt I][reject] [mod] [dyn] [reinststate] [[dev] If]
route [-v] [-A family] del [-net|-host] target [gw Gw] [netmask Nm] [metric N] [[dev] If]
route [-V] [--version] [-h] [--help]
```

route 命令的常见选项如表 7-7 所示。

表 7-7 route 命令的常见选型

选项	描述
-v	选用细节操作模式
-A family	用指定的地址族（如inet, inet6）
-n	以数字形式代替解释主机名形式来显示地址，此项对试图检测对域名服务器进行路由发生故障的原因非常有用
-e	用netstat的格式来显示路由表。-ee将产生包括选路表所有参数在内的大量信息
-net	路由目标为网络
-host	路由目标为主机
-F	显示内核的FIB路由表，其格式可以用-c和-ee选项改变
-C	显示内核的路由缓存
del	删除一条路由
add	添加一条路由
target	指定目标网络或主机，可以用点分十进制形式的IP地址或主机/网络名
netmask Nm	为添加的路由指定网络掩码
gw Gw	为发往目标网络/主机的任何分组指定网关。注意：指定的网关首先必须是可达的。也就是说必须为该网关预先指定一条静态路由。如果为本地接口之一指定这个网关地址的话，那么此网关地址将用于决定此接口上的分组将如何进行路由。这是BSD风格所兼容的
metric M	把选路表中的路由值字段（由选路进程使用）设为M
mss M	把基于此路由之上的连接的TCP最大报文段长度设为M字节。这通常只用于优化选路设置
window W	把基于此路由之上的连接的TCP窗口长度设为W字节。这通常只用于AX.25网络和不能处理背对背形式帧的设备
irtt I	把基于此路由之上的TCP连接的初始往返时间设为I毫秒（1-12000）。这通常也只用于AX.25网络。如果省略此选项，则使用RFC1122的默认值300ms

续表

选项	描述
reject	设置一条阻塞路由，以使一条路由查找失败。这用于在使用默认路由前先屏蔽掉一些网络。但这并不能起到防火墙的作用
mod, dyn, reinstate	设置一条动态的或更改过的路由。这些标志通常只由路由进程来设置。这只用于诊断目的
dev If	强制使路由与指定的设备关联，否则内核会自己来试图检测相应的设备（通常检查已存在的路由和加入路由的设备的规格）。在多数正常的网络上无需使用 如果dev If是命令行上最后一个指定的选项，那么可以省略关键字dev，因为它是默认值。否则路由修改对象（metric - netmask - gw - dev）无关紧要

### 1. 查看路由表

使用不带参数，或者仅带-n选项的route，可以查看内核IP路由表。

例：查看系统的IP路由表。命令如下：

```
[root@unknown root]# route
```

Kernel IP routing table

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
192.168.1.0	*	255.255.255.0	U	0	0	0	eth0
169.254.0.0	*	255.255.0.0	U	0	0	0	eth0
127.0.0.0	*	255.0.0.0	U	0	0	0	lo
default	192.168.1.1	0.0.0.0	UG	0	0	0	eth0

路由表输出有以下信息如表7-8所示。

表 7-8 内核IP路由表输出中包含的栏目

栏目	描述
Destination	目标网络或目标主机，Default表示默认路由。
Gateway	网关地址，*表示没有被设置。
Genmask	目标网络的子网掩码。其中255.255.255.255为主机，0.0.0.0为默认路由。
Flags	可能出现的标志有：U路由正常；H主机路由；G（use gateway）使用网关的间接路由；R为动态选路恢复路由；D该路由由选路进程或重定向动态创建；M该路由已由选路进程或重定向修改；！阻塞路由。
Metric	通向目标的距离（通常以跳来计算）。新内核不使用此概念，而选路进程可能会用。
Ref	使用此路由的活动进程个数（Linux内核并不使用）。
Use	查找此路由的次数。
Iface	使用此路由时，发送分组的接口。

上例输出的路由表中常用的路由表项的含义分别为：

- 路由表的第一项表示如果本机与192.168.1.0/255.255.255.0子网中主机通信时不需要经过网关，直接经eth0发送到目的主机。
- 路由表的第三项表示如果本机与环回接口的通信时不需要经过网关，直接经环回

接口 lo 发送数据包。

- 路由表的第四项表示如果本机与其他的主机通信时需要经过默认 (default) 网关，经 eth0 发送到网关 192.168.1.1，路由到最终目的地。

当发送一个数据包时，内核将要发送的数据包的目标地址依次在路由表从前到后匹配，没有找到能够匹配的路由表项，则使用 Destination 为 default 的表项发送数据包。

注意：每当用户更改了接口的 IP 地址时，它的默认路由表项将自动从 IP 路由表中删除。这时用户必须使用 route 命令手工加入默认路由表项。

如果无法匹配其他路由表项，则使用默认路由表项。

## 2. 操作路由表

对路由表常见操作包括添加路由表项和删除路由表项。

例：加入环回接口路由。命令如下：

```
# route add -net 127.0.0.0
```

加入正常的环回接口项，它使用掩码 255.0.0.0（由目标地址决定了它是 A 类网络）并与设备 lo 相关联。

例：添加一条到 192.56.76.0/255.255.255.0 子网，而不需要经过网关的路由。

```
# route add -net 192.56.76.0 netmask 255.255.255.0 dev eth0
```

向 eth0 添加一条指向网络 192.56.76.x 的路由。其中的 C 类子网掩码并不必须，因为 192.\* 是个 C 类的 IP 地址。其中关键字 dev 可省略。

例：加入一条经过网关 192.168.1.1 的默认路由。

```
# route add default gw 192.168.1.1
```

加入一条默认路由。使用此路由的所有分组将通过网关 192.168.1.1 进行传输。

例：删除一条到 192.56.76.0/255.255.255.0 子网的路由表项。

```
# route del -net 192.56.76.0 netmask 255.255.255.0 dev eth0
```

例：删除默认路由表项。

```
# route del -net default
```

## 7.1.8 网络配置综合练习

由于使用 ifconfig 对网络进行设置，使用 route 对内核路由表进行更改后，不能被永久保存，当下一次重新启动后就不再有效，而使用 netconfig 对网络的设置虽然能永久保存，但是不会更改当前的网络设置，只有当系统重新引导后，设置才会生效。

重新引导系统的代价是很大的，不仅重新引导系统花费很多时间，而且必须停止当前的所有任务，特别是服务器，应该一直工作状态，不应该多次重新引导。如何既能保持系统不重新引导，又可以立即更改网络设置，更改的设置又能永久生效呢？

解决的办法是：同时使用 netconfig、route 和 ifconfig 命令。使用了 netconfig 保证下次引导系统时，更改的网络设置 netconfig 和 route 保证不需重新引导系统，而使网络设置的更改有效。

例：系统中网卡的 IP 地址为 192.168.1.190，子网掩码为 255.255.255.0，DNS 服务器的 IP 地址为 159.226.5.65，默认网关的 IP 地址为：192.168.1.1。要求将网卡的 IP 地址更改为

192.168.1.191, 其他设置保持不变。

操作步骤如下:

(1) 使用 netconfig 将永久改变网卡的 IP 地址。

以 root 身份运行 netconfig 命令, 在设置窗口的 IP address 文本框中输入 192.168.1.191, 在 netmask 文本框中输入 255.255.255.0, 在 Default gateway 文本框中输入 192.168.1.1, 在 Primary nameserver 文本框中输入 159.226.5.65。然后按 Tab 至 Ok 按钮, 再按回车键, 保存网络设置, 并退出 netconfig。

(2) 测试当前的网络配置是否发生改变。

使用 ifconfig 查看对 IP 地址更改是否立即生效, 命令如下:

```
[root@john root]# Ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:0C:29:B6:CB:D0
          inet addr:192.168.1.190  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:21178 errors:0 dropped:0 overruns:0 frame:0
          TX packets:11 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:1463905 (1.3 Mb)  TX bytes:575 (575.0 b)
          Interrupt:10 Base address:0x1080
```

可见对 IP 地址的更改没有立即生效。

(3) 使用 ifconfig 立即更改网卡的 IP 地址, 命令如下:

```
[root@john root]# ifconfig eth0 192.168.1.191
```

(4) 检查更改。

检查当前的 IP 地址:

```
[root@john root]# Ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:0C:29:B6:CB:D0
          inet addr:192.168.1.191  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:25774 errors:0 dropped:0 overruns:0 frame:0
          TX packets:67 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:1801176 (1.7 Mb)  TX bytes:8465 (8.2 Kb)
          Interrupt:10 Base address:0x1080
```

可见对 IP 地址的更改已经生效。

检查路由表的变化, 命令如下:

```
[root@john root]# route
```

结果如下:

Kernel IP routing table

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
192.168.1.0	*	255.255.255.0	U	0	0	0	eth0



```
127.0.0.0      *          255.0.0.0      U    0    0    0 lo
```

可见当修改了 IP 地址之后, 与该 IP 地址相关的路由表项都被删除。这时无法通过默认网关访问互联网, 只能访问 192.168.1.0/255.255.255.0 子网的计算机。

为了能够上互联网, 需要手工加入默认的路由表项。命令如下:

```
[root@john root]# route add default gw 192.168.1.1
```

再次使用 route 命令查看路由表:

```
[root@john root]# route
```

结果如下:

```
Kernel IP routing table
```

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
192.168.1.0	*	255.255.255.0	U	0	0	0	eth0
127.0.0.0	*	255.0.0.0	U	0	0	0	lo
default	192.168.1.1	0.0.0.0	UG	0	0	0	eth0

这样就成功进行了所有的网络设置。

(5) 使用 init 6 命令重新引导系统后, 在此使用 ifconfig 和 route 查看网卡的 IP 地址和内核的 IP 路由表, 可以发现 IP 地址已经永久生效。

## 7.2 常用网络工具

计算机网络的主要优点是能够实现资源和信息的共享, 并且用户可以远程访问信息。Linux 提供了一组强有力的网络命令来为用户服务, 这些工具能够帮助用户登录到远程计算机上、传输文件和执行远程命令等。本节介绍下列几个常用网络工具: 传输文件 ftp、登录到远程计算机 telnet、文本模式浏览器 Lynx、优秀的邮件客户程序 mutt、安全 Shell 程序 ssh 和 sftp。

### 7.2.1 文本模式 Web 浏览器——lynx 命令

Red Hat Linux 9.0 默认安装了 Lynx 程序, 它是 Unix 中最常用的文字浏览器, 所以用户可以在大多数 Linux 中找到它的踪迹。与 Netscape Communicator 等图形浏览器不同的是, Lynx 是基于 Xterm 窗口或者 Terminal 窗口的文字浏览器。对于一个痴迷的网络用户来说, 速度是他们追求的目标, Lynx 是纯文字界面, 因此成为许多人津津乐道的互联网利器。

Lynx 软件的最新版本可以在其主页 <http://lynx.browset.org> 中找到, Lynx 支持大多数流行的操作系统, 所以在下载的时候要注意选择相应的版本。

#### 1. 启动方法

在终端中直接键入 lynx 就可以启动 Lynx, 用户也可以在启动 Lynx 时指定需要访问的页面。

例: 键入如下指令就可以访问 [www.google.com](http://www.google.com)。命令如下:

```
lynx http://www.google.com
```

启动 Lynx 以后, 可以键入 h、H 或者?获得系统帮助。

## 2. 打开新的网页

如果在 Lynx 的同一窗口打开一个新的网页,可以按 G 键,此时在屏幕下方会显示 URL to Open:的提示,输入相应网页的 URL,并按回车键就可以打开一个新的网页了。

## 3. 浏览方法

用 Lynx 来浏览网页,可以通过箭头键在所有链接中来回移动,其中上下箭头键可以在前后链接中移动,也可以用 Tab 键移到下一链接,右箭头键就可以打开当前链接,而左箭头键就可以用来返回到上一页面。刚才访问过的网页,会被记录在缓存中,再次访问时会直接从本地硬盘读取。

## 4. 访问历史记录

要想再次访问以前访问过的网页,按 Backspace 或者是 Delete 键可以看到这些链接,可以像浏览其他网页一样选择相应的链接进行浏览,再次按这两个键会切换到刚才浏览的页面。

## 5. Lynx 和 HTML Frames

Lynx 可以辨别出 Netscape 和 Microsoft Explorer 的 Frame (框架网页),但是它不能够用窗口的形式给出框架的定位,而只是提供用卷标链接到相应框架的处理方法,通常这些卷标位于屏幕的左上方,每一个卷标代表着一个框架的内容。为了找到用户需要的内容,就得尝试不同的卷标,甚至到每一个链接查看。在支持 Frame 的浏览器中,会出现在 A 框架中单击链接,相应内容显示在 B 框架中的情形,Lynx 虽然不能用窗口形式给出框架定位,但可以在同一窗口中循链接显示相应内容。

### 7.2.2 远程登录 —— telnet 命令

telnet 用于 Internet 的远程登录。它可以使用户坐在已上网的计算机键盘前通过网络进入的另一台已上网的计算机,即远程登录到远程计算机。这种方式的登录可以发生在同一房间的两台计算机之间,或是在世界范围内已上网的计算机。习惯上把被连通计算机,并且为网络上所有用户提供服务的计算机称之为服务器 (Server),而自己在使用的机器称之为客户机 (Client)。一旦连通后,客户机就成为服务器的一个用户,可以这样想象,就好像用户在远程计算机旁一样,只是使用本地的键盘和显示器代替远程计算机的键盘和显示器,以完成输入和显示输出任务,因此可以享有服务器所提供的一切服务。用户可以执行常规的交互过程(如登录,执行命令或者注销),也可以进入很多特殊的服务器。网上不同的主机提供的各种服务都可以被使用,其前提是用户必须是远程计算机的一个用户,远程计算机必需提供 telnet 服务,而且能够通过 TCP/IP 互通。

使用 Telnet 的最简单的方法是在命令行输入:

```
telnet 远程的计算机名/IP地址
```

下面举一个基本的 telnet 使用实例:

```
[root@john root]# telnet 192.168.1.19
```

```
Trying 192.168.1.19...
```

```
Connected to 192.168.1.19.
Escape character is '^'.
Red Hat Linux release 9 (Shrike)
Kernel 2.4.20-8 on an i686
login: globus
Password:XXXXX
Last login: Thu Jun  3 13:32:28 from 192.168.1.188
[globus@unknown globus]$ ls
hello1.txt hello3.txt newdir3 result.txt vitest
hello2.txt hello4.txt newdir4 testvinew vitestpart
[globus@unknown globus]$ hostname
unknown
[globus@unknown globus]$ rm hello1.txt
[globus@unknown globus]$ ls
hello2.txt hello4.txt newdir4 testvinew vitestpart
hello3.txt newdir3 result.txt vitest
[globus@unknown globus]$ exit
Connection closed by foreign host.
[root@john root]#
```

上述例子是用 Telnet 连接远程计算机 192.168.1.19。本地计算机的主机名为 john，在提示符后输入 telnet 192.168.1.19，并按回车键之后，进入连接过程，由于网络正常，而且主机 192.168.1.19 也启动了 telnet 服务，因此会出现操作系统（Red Hat Linux 9）的提示信息，然后出现登录提示，要求输入用户名，在此输入 globus，按回车键后，提示输入口令，当输入 192.168.1.19 主机上 globus 用户的口令，并按回车键之后，会出现新的 shell 提示符。

细心的读者会发现 shell 提示符由原来的[root@john root]#变成了[globus@unknown globus]\$，这说明当前的计算机的主机名为 john，远程的主机名为 unknown，已经成功地登录到 unknown 主机上了，以后执行的几条命令（包括 ls、hostname、rm）分别列出 unknown 主机上的文件列表、显示 unknown 主机的主机名以及删除了 unknown 主机上 globus 用户主目录下的文件 hello1.txt。在这种情况下，用户无法对本地计算机进行任何操作，除非用户在远程计算机上注销自己。如果用户希望从远程计算机上注销，返回到本地计算机工作环境，可以使用 exit 命令。

当使用了 exit 命令后，将会给出 Connection closed by foreign host. 的提示，表明与远程计算机的连接已关闭，并返回到本地 shell 提示符下，上例中的 shell 提示符变为[root@john root]#，说明已经返回到主机 john 的提示符下，而且当前用户为 root。

实际上，Telnet 是一个可以注册进入远程计算机系统的工具。

**注意：**不仅 Red Hat Linux 9 提供了 telnet 远程登录客户端程序，几乎所有的操作系统都提供了功能类似的程序。在 Red Hat Linux 9 中，可以使用 telnet 远程登录到多种操作系统下，如 Windows，UNIX 和 OS/2。

为了安全考虑，默认情况下超级用户 root 不能直接使用 telnet 命令登录到远

程计算机,如果在远程必须使用 root 帐户,可以首先以普通用户登录,然后使用 su 或者 su -切换到 root 用户身份。

### 7.2.3 远程文件传输 —— ftp 命令

ftp 命令是标准的文件传输协议的用户接口。ftp 是在 TCP/IP 网络上的计算机之间传输文件的简单且有效的方法。它允许用户传输 ASCII 文件和二进制文件。在 ftp 会话过程中,用户可以通过使用 ftp 客户程序连接到另一台计算机上。从此,用户可以在目录中上下移动、列出目录内容、把文件从远程机拷贝到本地机上、把文件从本地机传输到远程系统中。

**注意:** 如果用户没有那个文件的存取权限,就不能从远程系统中获得文件或向远程系统传输文件。为了使用 ftp 传输文件,用户必须知道远程计算机上的合法用户名和口令。这个用户名/口令的组合用来确认 ftp 会话,并用来确定用户对要传输的文件可以进行什么样的访问。

另外,用户还需要知道对其进行 ftp 会话的计算机的名字或 IP 地址。

远程计算机必须运行着 ftp 服务器程序,否则即使上述条件满足,也无法进行 ftp 会话。

ftp 命令的功能是在本地机和远程机之间传送文件。该命令的一般格式如下:

ftp 主机名/IP

其中“主机名/IP”是所要连接的远程计算机的主机名或 IP 地址。

在命令行中,主机名属于可选项,如果指定了主机名,ftp 将试图与远程机的 ftp 服务程序进行连接。如果没有指定主机名,ftp 将给出提示符,等待用户输入命令,此时在 ftp>提示符后面输入 open 命令加主机名或 IP 地址,将试图连接至指定的主机。

不管使用哪一种方法,如果连接成功,需要在远程机上登录。用户如果在远程机上有账号,就可以通过 ftp 使用这一账号,并需要提供口令。

在远程机上,用户账号的读写权限决定了该用户在远程机上能下载什么文件和将上载文件放到哪个目录中。如果用户没有远程机的专用登录账号,许多 ftp 站点设有可以使用的特殊账号。这个账号的登录名为 anonymous (也称为匿名 ftp),当使用这一账号时,要求输入用户的 E-mail 地址作为口令。如果远程系统提供匿名 ftp 服务,用户使用这项服务可以登录到特殊的,供公开使用的目录。

通常 ftp 服务器至少提供两个目录: pub 目录和 incoming 目录。pub 目录包含该站点供公众使用的所有文件,incoming 目录存放上载到该站点的文件。一旦用户使用 ftp 在远程站点上登录成功,将得到 ftp>提示符。现在可以自由使用 ftp 提供的命令,可以用 help 命令取得可供使用的命令清单,也可以在 help 命令后面指定具体的命令名称,获得这条命令的说明。

最常用的命令如表 7-9 所示。

表 7-9 ftp 提供的命令

栏目	描述
ls	列出远程机的当前目录
cd 目录名	在远程机上改变工作目录

续表

栏目	描述
lcd 目录名	在本地机上改变工作目录
ascii	设置文件传输方式为ASCII模式
binary	设置文件传输方式为二进制模式
close	终止当前的ftp会话
hash	每次传输完数据缓冲区中的数据后, 就显示一个#号
get (mget) 文件名	从远程机传送指定文件到本地机
put (mput) 文件名	从本地机传送指定文件到远程机
open 主机名/IP地址	连接远程ftp站点
quit/bye	断开与远程机的连接并退出ftp
?	显示本地帮助信息
!shell命令	在ftp中执行Shell命令

下面通过一个实例说明使用 ftp 命令的过程。

例：从北京大学 ftp 站点（主机名为 ftp.pku.edu.cn）的/pub/packages/wu-ftp/目录下匿名下载文件 wu-ftp-2.6.0.tar.gz，到本地的当前目录。步骤如下：

（1）建立连接。使用 ftp 命令连接到远程主机 ftp.pku.edu.cn，命令如下：

```
[globus@unknown globus]$ ftp ftp.pku.edu.cn
```

连接成功后，给出如下的提示信息：

```
Connected to ftp.pku.edu.cn (202.38.97.197).
```

```
220 vineyard.pku.edu.cn FTP server (Version wu-2.6.1(1) Wed Mar 28 15:17:48
CST 2001) ready.
```

```
Name (ftp.pku.edu.cn:globus): _
```

在提示中包含了远程的主机名和相应的IP地址，以及ftp服务器软件的类型和版本号，以及登录提示。

（2）登录。在提示中ftp.pku.edu.cn:globus表示默认以本地用户globus登录ftp.pku.edu.cn，如果按回车键，则接受这一默认设置。由于要求匿名登录，因此需要在提示后输入匿名登录名anonymous，按后按回车键。操作如下：

```
Name (ftp.pku.edu.cn:globus): anonymous
```

```
331 Guest login ok, send your complete e-mail address as password.
```

```
Password:
```

然后提示输入口令，在此输入本地用户任意一个邮件地址，笔者输入 jzh\_smr@sohu.com，为了安全，在输入口令时不会给出任何提示，然后按回车键，将会出现一些欢迎信息，最后出现ftp命令的提示符：

```
ftp> _
```

（3）将ftp服务器的目录改变到需要下载文件的位置。命令如下：

```
ftp> cd pub/packages/wu-ftp/
```

当服务器接收到cd命令之后，进行改变目录操作，然后给出成功执行应答信息：

```
250 CWD command successful.
```

（4）查看用户在远程ftp服务器的当前目录。命令如下：

```
ftp> pwd
```

结果如下:

```
257 "/pub/packages/wu-ftp" is current directory.
```

(5) 下载文件wu-ftp-2.6.0.tar.gz到本地的当前目录。命令如下:

```
ftp> get wu-ftp-2.6.0.tar.gz
```

当服务器接收到get wu-ftp-2.6.0.tar.gz命令之后, 将执行文件下载。提示信息如下:

```
local: wu-ftp-2.6.0.tar.gz remote: wu-ftp-2.6.0.tar.gz
```

```
227 Entering Passive Mode (202,38,97,197,4,55)
```

```
150 Opening BINARY mode data connection for wu-ftp-2.6.0.tar.gz (339122 bytes)
```

提示说明下载的远程和本地文件的文件名都为wu-ftp-2.6.0.tar.gz, 使用被动(Passive)和二进制(BINARY)模式下载, wu-ftp-2.6.0.tar.gz文件的大小为339122字节。下载所需的时间根据网络速度和文件的大小而定。当文件下载完成后, 服务器将给出如下提示:

```
226 Transfer complete.
```

```
339122 bytes received in 25.1 secs (13 Kbytes/sec)
```

(6) 在ftp提示符后, 执行ls命令查看下载在本地的文件信息。命令如下:

```
ftp> !ls -l wu-ftp-2.6.0.tar.gz
```

结果如下所示, 表示文件下载成功:

```
-rw-rw-r-- 1 globus globus 339122 6月 4 09:46 wu-ftp-2.6.0.tar.gz
```

(7) 关闭连接, 并且提出ftp会话, 返回到shell提示符。命令如下:

```
ftp> exit
```

提示信息如下:

```
221-You have transferred 339122 bytes in 1 files.
```

```
221-Total traffic for this session was 340317 bytes in 1 transfers.
```

```
221-Thank you for using the FTP service on vineyard.pku.edu.cn.
```

```
221 Goodbye.
```

```
[globus@unknown globus]$
```

提示给出了下载的文件数, 传输的字节数以及关闭提示, 最后返回到shell提示符。

注意: 如果使用ftp上传文件, 使用put命令。

#### 7.2.4 简单文本电子邮件客户——mutt命令

多数现代的电子邮件客户允许用户选择用纯文本(plain text)方式, 或者用HTML方式来发送邮件。HTML格式的电子邮件的优越性在于, 它们不仅可以包含图形和到网站的链接信息, 而且可以指定字体, 控制布局, 或添加图片和背景, 这些功能使编写的邮件更加具有视觉魅力。

另一方面, 纯文本邮件只包含纯文本。它们不花哨, 在邮件中没有嵌入图画, 也没有特殊字体。纯文本文件非常简单, 而且减少了携带病毒的可能性。

纯文本这个术语是指ASCII格式的文本数据。纯文本(又称明文, plain text)是最容

易移植的格式，因为各类机器上几乎所有的电子邮件应用程序都支持这种格式。

mutt 是一个用于 Unix 操作系统的小巧玲珑，却又功能强大的基于文本的邮件客户。

`~/muttrc` 是 mut 程序的配置文件。它既为 mut 提供了灵活性和可配置性，又为新用户带来了无限烦恼。mutt 中可用的选项多得惊人，它还允许用户控制几乎所有它用来收发和阅读邮件的功能。正如所有功能强大的软件一样，用户需要花一段时间来理解它的功能和用途。

多数选项是通过 `set` 或 `unset` 命令，使用逻辑或字符串值来启用的，如 `set folder = ~/Mail`。

所有的配置选项随时都能被改变，方法是：键入冒号“:”之后，跟随相关的命令。例如：`:unset help` 关闭屏幕顶端的键盘命令提示功能。要重新启用这个提示功能，键入 `:set help`。

如果用户想不起来要使用的命令，可以随时使用 `tab` 命令自动补全这个功能来为用户提供帮助。

用户不必在每次运行 mutt 时都输入他们所喜欢的配置命令，可以把这些配置命令保存到一个文件中。该文件在每次启动 mutt 程序时被载入，它必须存在于用户的主目录中，必须叫做 `~/muttrc` 或者 `~/mutt/muttrc`。

当用户启动 mutt 后，首先看到的是一个带有电子邮件消息的列表。这个初始菜单叫做索引，如图 7-3 所示。

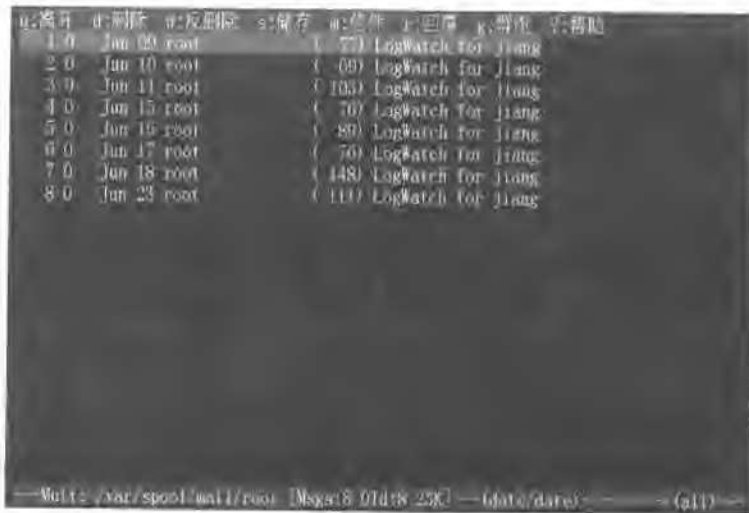


图 7-3 mutt 命令的主屏幕

这些邮件在默认的邮件文件夹中，通常称为邮筒（mail spool），用户可以把它看做收件箱。使用键盘上的 `k` 和 `j` 键来在消息列表中上下移动突出显示的光标。

在索引或分页视图里，使用 `r` 键来回复邮件或 `m` 键来编写新邮件。mutt 会提示 `To:`（在此输入收件人的邮件地址）以及 `Subject:`（在此输入邮件的主题）。然后，一个文本编辑器（被配置文件中 `$EDITOR` 环境变量定义，默认为 `vi` 编辑器）会被启动，允许用户用它来编辑邮件内容。键入邮件内容，保存文件，然后退出编辑器。

邮件编辑完毕后，mutt 会显示编辑菜单。用户在这里可以定制消息头、改变编码、添

加文件附件，或者只按 `y` 键来发送该邮件。

有关 `mutt` 更多的信息，请参考 `muttrc` 和 `mutt` 的手册页，`mutt` 命令的说明书也很有用。

### 7.2.5 安全 shell —— `ssh` 命令

无论是使用 `telnet` 进行远程登录，还是使用 `ftp`，在传输机制上都没有考虑安全机制，其本质上都是不安全的。由于它们在网络上用明文传输数据、用户账号和用户口令，攻击者可以通过网络窃听等网络攻击手段，非常容易截取这些数据、用户账号和用户口令。而且，这些网络服务程序的简单安全验证方式也存在容易遭受“中间人”（`man-in-the-middle`）攻击。所谓的“中间人”攻击，就是“中间人”冒充真正的服务器接收用户传给服务器的数据，然后再冒充用户把数据传给真正的服务器。用户和服务器之间的数据被“中间人”转手之后，就可能导致各种安全问题。

`SSH` 是 `Secure Shell` 的缩写形式。通过使用 `SSH`，用户和服务器之间的数据被加密，这样就阻止了“中间人”攻击。在使用 `SSH` 时，数据在传输之前是被压缩，这样可以加快网络的传输速度。

`SSH` 不仅能完全代替 `telnet`，而且还可以为 `ftp`，`pop`，甚至 `PPP` 协议提供一个安全的通道。

#### 1. `SSH` 的安全性

从客户端来看，`SSH` 提供了两种级别的安全验证：基于口令的安全验证和基于密钥的安全验证。

基于口令的安全验证。只要知道用户账号和口令的人，就可以登录到远程主机，并且所有传输的数据都被加密。但是，这种验证方式不能保证用户正在连接的服务器是他想连接的服务器。可能出现别的服务器冒充真正的服务器，即受到“中间人”攻击。

基于密钥的安全验证。要依靠密钥，也就是说，用户必须为自己创建一对密钥，并把公有密钥放在需要访问的服务器上。如果用户要连接到 `SSH` 服务器上，客户端软件就会向服务器发出请求，请求服务器用用户设定的密钥进行验证。服务器收到请求之后，先在用户的主目录中寻找用户的公钥。如果两个密钥一致，服务器就用共有密钥“质询”（`challenge`），并把它发送到客户端软件。客户端收到质询之后，再用用户的私钥进行解密，然后将解密的结果发送给服务器。

与第一种级别相比，基于密钥的安全验证不需要在网络上传输口令，但是整个登录过程会慢一些。

`SSH` 强化了对身份验证的安全机制，它应用了密码学（`Cryptography`）中的多种加密在机制，如对称和非对称密码算法，单向 `Hash` 函数和随机数生成等。通过使用 `SSH`，可以在本地主机和远程服务器之间建立“加密通道”，这样为 `pop`（邮局协议）、`X`（`X` 协议）应用提供了安全保障。

在 `Red Hat Linux 9` 默认安装了 `OpenSSH` 服务器和客户端软件。

#### 2. 使用用户名和口令建立 `SSH` 连接

使用 `SSH` 客户程序登录远程主机的格式如下：



`ssh -l` 在远程主机上的用户名 远程主机名/IP地址

例：在 `unknown` 主机上，使用 `ssh` 登录到 IP 地址为 `192.168.1.190` 的远程主机。命令如下：

```
[globus@unknown globus]$ ssh -l globus 192.168.1.190
```

命令执行后，如果 `OpenSSH` 工作正常，用户将看到如下提示信息：

```
The authenticity of host '192.168.1.190 (192.168.1.190)' can't be
established.
```

```
RSA key fingerprint is b9:12:06:fa:67:cd:41:c4:31:52:a7:e0:10:26:7c:36.
```

```
Are you sure you want to continue connecting (yes/no)?_
```

`OpwnSSH` 提示用户它不知道这台主机，但是用户不需要担心，因为这是用户第一次登录这台主机，在问号“?”后输入 `yes`，然后按回车键。这将把这台主机的“识别标记”（`fingerprint`）加入到文件“`~/ssh/known_hosts`”中。第二次在访问这台主机时就不需要再出现这些提示信息了。

然后 `SSH` 提示输入账号的口令。提示如下：

```
globus@192.168.1.190's password:
```

输入口令，按回车键之后，就建立了一条 `SSH` 连接，这之后就可以像 `telnet` 一样使用 `SSH` 了。出现如下提示符：

```
[globus@john globus]$
```

### 3. 生成 `SSH` 的密钥

用户可以生成自己的密钥对，并且分发给自己。这样做有两个好处：

- 可以防止中间人攻击。
- 可以只用一个口令就登录到用户想登录的服务器上。

使用下列命令 `ssh-keygen` 命令生成密钥时，如果远程主机使用的是 `SSH 1.x`，则使用 `ssh-keygen -t rsa1` 命令，如果远程主机使用的是 `SSH 2.x`，可以使用 `ssh-keygen -t rsa` 命令，或者使用 `ssh-keygen -t dsa` 命令。因为密钥使用不同的文件存储，所以可以在同一台计算机上生成 `SSH1` 和 `SSH2` 的密钥。

使用 `ssh-keygen` 命令生成密钥的过程如下：

```
[globus@unknown globus]$ ssh-keygen -t rsa
```

命令执行后，出现如下提示：

```
Generating public/private rsa key pair.
```

```
Enter file in which to save the key (/home/globus/.ssh/id_rsa):
```

提示 `rsa` 密钥对保存的位置，按回车键接受默认设置。然后出现输入口令提示：

```
Enter passphrase (empty for no passphrase):
```

输入口令，它不会在屏幕上显示：

```
Enter same passphrase again:
```

在此输入口令：

```
Your identification has been saved in /home/globus/.ssh/id_rsa.
```

```
Your public key has been saved in /home/globus/.ssh/id_rsa.pub.
```

The key fingerprint is:

```
bf:1f:d3:c7:3c:67:9c:f9:02:e5:a1:52:1d:90:c4:bd globus@unknown
```

私钥保存在`~/.ssh/id_rsa`文件中，而公钥保存在`~/.ssh/id_rsa.pub`文件中，此例的username为globus。

使用`ssh-keygen -t dsa`和`ssh-keygen -t rsa`的密钥对成过程几乎一样，只是文件名不同而已。

通过上例，已经得到一对密钥；将公钥分发到你希望登录的所有远程主机上；而私钥保存在本地计算机上，应该保密，防止别人查看和拷贝。

用户的私钥只有本人才能查看，用ls验证如下：

```
[globus@unknown globus]$ ls -l /home/globus/.ssh/id_rsa
```

```
-rw----- 1 globus globus 951 6月 4 14:50 /home/globus/.ssh/id_rsa
```

可见它的权限为“`-rw-----`”。如果你确信你的私钥已被泄露，可以使用`ssh-keygen`生成一对新的密钥。

#### 4. 分发 SSH 的密钥

在每一个需要用SSH连接的远程计算机上，用户需要在远程主机的主目录下创建一个`.ssh`子目录，然后将他生成的公钥文件复制到这个子目录（`~/.ssh`）下，最后执行如下命令：

```
chmod 644 ~/.ssh/id_rsa.pub
```

如果用户希望从不同的计算机登录到远程主机，可以在`id_rsa.pub`中存放多个公共密钥。这需要重新生成一对密钥，然后将生成的公钥文件与原有的公钥文件的内容合并。

### 7.2.6 安全的文件传输 —— sftp 命令

安全的sftp用于进行安全的文件共享。它可以用来从远程主机上上传和下载文件，它在一个安全的通道上进行所有的操作，它与ftp的操作和功能类似，而且可以替代ftp。sftp可能会使用ssh的特征，sftp首先连接，并登录到特定的远程主机，然后进入一个交互式操作过程。

另外，sftp也可以工作在非交互模式。

ftp常用于交互式上传下载文件，命令如下：

```
sftp 用户名@主机名/IP 地址
```

或者

```
sftp 主机名/IP 地址
```

如果使用第一种形式，则同时指定了远程主机名和用户名；如果使用第二种方式，则不指定主机名，将使用本地当前用户名作为远程主机的用户名登录。

例：在本地上传一个名为`wu-ftp-2.6.0.tar.gz`的文件到远程主机192.168.1.190的root用户的主目录。操作过程如下：

(1) 使用sftp命令建立连接，并且登录。命令如下：

```
[globus@unknown globus]$ sftp root@192.168.1.190
```

连接至192.168.1.190后，出现如下提示信息：

```
Connecting to 192.168.1.190...
```

```
root@192.168.1.190's password:
```

在口令提示符中输入用户的口令，如果输入了正确的口令，则进入 **sftp** 提示符：

```
sftp>
```

在 **sftp** 的提示符中，执行 **sftp** 的 **put** 命令，进行文件上传，操作如下：

```
sftp> put wu-ftp-2.6.0.tar.gz
```

然后开始上传过程，提示如下：

```
Uploading wu-ftp-2.6.0.tar.gz to /root/wu-ftp-2.6.0.tar.gz
```

当上传完成时，会再次出现 **sftp** 提示符：

```
sftp>
```

注销，并返回 **shell** 提示符：

```
sftp> exit
```

```
[globus@unknown globus]$
```

整个上传过程结束。

## 思考与练习

- (1) 在 Linux 中，有哪些常用的网络诊断命令？
- (2) 如何设置网卡的 IP 地址、子网掩码以及 DNS？
- (3) 如何查看何修改 Linux 的路由表？
- (4) **ssh** 和 **sftp** 的作用是什么？如何使用它们？

## 第 8 章 文件系统管理

Linux 不仅提供了专用的高效文件系统 ext2 和 ext3, 同时对大多数常见的文件系统提供了良好的支持, 例如 FAT、FAT32、NTFS、SMBFS、NFS 和 ISO9660, 这样使 Linux 既能充分与其他操作系统兼容, 又能充分利用本地和网络上的文件资源。

Linux 提供了一整套用来管理这些文件系统的工具, 这些工具包括 hdparm、fdisk、mkfs、mount、df、du、fsck、tune2fs、sync 和 swapon。Linux 也提供了高级的文件复制命令 dd。

本章将要学习 ext3 文件的结构和特点, 同时熟悉 Linux 提供的文件系统管理工具, 而且也介绍了如何使用 dd 命令进行高级文件复制。

### 8.1 Linux 文件系统简介

#### 8.1.1 Linux 文件系统概述

Linux 操作系统使用文件来表示所有的逻辑实体与非逻辑实体。逻辑实体指文件和目录, 非逻辑实体则泛指硬盘、终端、打印机等硬件设备, 而且每一个文件都有一个文件名。

Linux 文件系统中, 其中根文件系统 (root file system) 位于文件系统的最顶层, 同时也最为重要, 根文件系统在系统启动时挂载在 / 目录, 若无法挂接根目录 (/), 则开机时就无法进入 Linux 系统。

#### 8.1.2 Linux 文件系统的重要目录

根目录下有 /etc、/dev、/boot、/home、/lib、/lost+found、/mnt、/opt、/proc、/root、/bin、/sbin、/tmp、/var 和 /usr 等重要目录, 下面分别将简要介绍。

(1) /etc。本目录下存放着许多系统所需的重要配置和管理文件, 其中的一些为文本文件, 另外还有一些子目录, 子目录对配置文件进行必要的分类, 同一类配置文件都放在同一个子目录中。

(2) /dev。本目录中存放着设备文件 (device file), 用户可以经由内核来存取系统中的硬件设备。当使用设备文件时, 内核会辨别出 I/O 请求 (输入/输出请求), 并传递到相应的设备驱动程序, 以便完成特定的动作。在 /dev 目录下, 每个设备均有一个相对应的设备文件, 另外 /dev 目录下还有一些文件, 是在安装系统时建立的, 它不一定对应到实际的硬件设备, 此外也有一些虚拟的设备, 不对应到任何实际设备 (如空设备 (/dev/null)), 任何写入该文件的请求均会被执行, 但被写入的内容均会消失无踪。

在 /dev 目录下文件分为字符设备文件和块设备文件两大类。块设备文件通常是外部设备, 文件读写都是以整个块为单位进行的, 而字符设备 (如串行口), 文件读写都是以一个字节为单位进行的。

其中 /dev/fd\* 表示软盘介质。/dev/hd\* 指 IDE 硬盘的分区, 如 /dev/hda1 则指硬盘的第一个分区。/dev/sd\* 指 SCSI 磁盘。/dev/console 指系统的控制台, 实际上是连接到 Linux 操作

系统的屏幕。`/dev/tty` 指提供使用者不同终端的操作控制台驱动，并由虚拟控制台功能键切换，可使用 `Ctrl+Alt+F1~F6`。`/dev/ttyS`，`ttyS` 指串行口，`ttyS0` 即为 `COM1`，`ttyS1` 即为 `COM2`。

(3) `/boot`。本目录下放置了与系统启动相关的文件（如 `initrd.img`、`vmlinuz`、`System.map`），它们均为重要的系统文件，用户不要随便删除其中的文件。`initrd.img` 为系统启动时最先装入的文件。`vmlinuz` 即为内核（kernel）的映像（image）文件。`System.map` 包括了内核（kernel）的功能及位置。`top`、`ps` 指令会去读此文件，来显示系统目前的状态。因此，`System.map` 文件必须对应到相应的 kernel，不然会显示出错信息。

(4) `/home`。一般而言，用户的主目录（`$HOME`）就是放在 `/home` 目录下。

(5) `/lib`。系统运行时所需要用到的许多重要的共享函数库（shared libraries）都放在这个目录下，它们包含最重要的 GNU 的 C 函数库，凡是文件名为 `library.so.version` 的共享函数库，通常都放在 `/lib` 目录下。

(6) `/lost+found`。一般在重启或关闭系统时，会自动地执行 `sync` 程序。`sync` 将缓存（cache）中（此时 `dirty` 位为 1）的信息写回磁盘，当再次开机时，系统会去检查每一个 `dirty` 位是否为 1，如果为 1 则会执行 `fsck`。当进行磁盘检查（`fsck`）时，常会询问是否删除 `dirty` 位为 1 的块，如果选择删除，则会把该块（与 `inode` 节点对应的块）集中放在 `lost+found` 目录，因此 `lost+found` 目录存放磁盘分区中已经丢失，但由 `fsck` 找回的已丢失时文件碎片。

(7) `/mnt`。该目录为系统默认的挂接点（mount point）目录，默认有 `/mnt/cdrom` 和 `/mnt/floppy` 挂接点目录，使用自动的挂载程序（如 KDE 桌面上的 `cdrom` 与 `floppy`，GNOME 的 `Drive Mount Applet`），均会自动地把光盘和软盘挂接到这二个目录上。如果要挂接额外的文件系统，一般而言用户都会将 `/mnt` 目录当作挂载点，然后在该目录下建立一个新的子目录，当作挂接目录，将同一张硬盘的一个 Windows 分区挂接（mount）到本机的 Linux 操作系统就可以使用这种方法。

(8) `/proc`。本目录为一个虚拟的文件系统，其功能仍然是文件和进程的统一，它不占用任何磁盘空间，因为该目录下的文件均放置在内存中，每当用户存取 `/proc` 文件系统时，内核会获得的文件存取动作，执行某些程序，然后再动态的产生目录与文件内容。

本目录下主要有两大类文件：kernel 和系统使用的硬件信息，`/proc/ioproports`（IO 端口）、`/proc/dma`（DMA 通道）、`/proc/meminfo`（内存信息）、`/proc/interrupts`（中断表）均属于这类文件。

(9) `/root`。本目录为系统管理者 `root` 的主目录。

(10) `/bin`。`bin` 为 `binary` 的简写，主要存放一些系统必备的执行程序（如 `cat`、`cp`、`chmod`、`df`、`dmesg`、`gzip`、`kill`、`ls`、`mkdir`、`more`、`mount`、`rm`、`su` 和 `tar` 等）。

(11) `/usr/bin`。主要放置一些应用软件工具的必备程序文件（如 `g++`、`gcc`、`dig`、`eject`、`gzip`、`less`、`make`、`man`、`mcop` 和 `wget` 等）。

(12) `/sbin`。主要存放一些必备的系统管理程序（如 `fdisk`、`halt`、`ifconfig`、`ifup`、`ifdown`、`reboot`、`rmmod`、`runlevel` 和 `shutdown` 等）。

(13) `/usr/sbin`。放置一些必备的网络管理程序（如 `dhcpcd`、`httpd`、`imap`、`netconfig`、`nmbd`、`samba`、`sendmail`、`squid`、`swap`、`tcpd` 和 `tcpdump` 等）。

(14) `/tmp`。本目录是供全体用户暂时存放文件的目录，有时某些应用程序执行中产生的临时文件也会暂放在这个目录中，然而系统预设对该目录的权限为可读、写、执行，

但无法删除（1777，drwxrwxrwt）文件，亦即多加上了t权限（save program text on swap device），以避免有人任意删除他人存放在/tmp目录的文件。

（15）/var。Linux操作系统将经常改变的或暂存的数据存放在/var目录下，而后系统新产生的数据会更新这些文件，在这个目录下有一些重要的目录，如/var/log、/var/spool、/var/run等。

- /var/log。log目录记载着Linux操作系统运行过程的日志，包括启动日志、网站访问日志（httpd/access\_log）、网站错误日志（httpd/error\_log）、网站安全记录（httpd/ssl\*）、邮件记录日志（maillog）和登录日志（secure）等。
- /var/spool。/var/spool是一个提供spool服务的目录，该目录下包含lpd、mail、squid三个最重要的子目录，打印文件的打印队列存储在/var/spool/lpd目录。

### 8.1.3 Linux 专用的文件系统

Linux本身提供了两种文件系统，分别是ext2和ext3文件系统类型。

默认情况下，Linux的早期版本使用文件系统ext2，ext2文件系统高效稳定。但是，随着Linux系统在关键业务中的应用，Linux文件系统的弱点也渐渐显露出来了，其中ext2文件系统是非日志文件系统。这在关键行业的应用是一个致命的弱点。

ext3文件系统是直接由ext2文件系统发展而来，目前ext3文件系统已经非常稳定可靠。Red Hat Linux 9默认使用ext3文件系统。ext3完全兼容ext2文件系统。用户可以平滑地过渡到一个日志功能健全的文件系统中来。这实际也是设计ext3日志文件系统的初衷。

ext3日志文件系统具有如下特点：

#### 1. 高可用性

系统使用了ext3文件系统后，即使在非正常关机后，系统也不需要检查文件系统，恢复ext3文件系统的时间只要几十秒钟。

#### 2. 保护数据的完整性

ext3文件系统能够极大地提高文件系统的完整性，避免了意外关机对文件系统的破坏。在保证数据完整性方面，ext3文件系统有两种模式可供选择。其中之一就是“同时保持文件系统及数据的一致性”模式。采用这种方式，用户永远不再会看到由于非正常关机而存储在磁盘上的垃圾文件。

#### 3. 该文件系统的性能高

尽管使用ext3文件系统时，有时在存储数据时可能要多次写数据，但是从总体上看，ext3比ext2的性能还要好一些。这是因为ext3的日志功能对磁盘的驱动器读写头进行了优化。所以，文件系统的读写性能较之ext2文件系统来说，性能并没有降低。

#### 4. 方便的文件系统转化

由于ext2文件系统转换成ext3文件系统非常容易，只要简单地键入两条命令即可完成整个转换过程，用户不用花时间备份、恢复、格式化分区等。用一个ext3文件系统提供的

小工具 `tune2fs`，它可以将 `ext2` 文件系统轻松转换为 `ext3` 日志文件系统。另外，`ext3` 文件系统可以不经任何更改，而直接加载成为 `ext2` 文件系统。

5. 多种日志模式

`ext3` 有多种日志模式，一种工作模式是对所有的文件数据及 `metadata`（定义文件系统中数据的数据，即数据的数据，也被称为元数据）进行日志记录（`data=journal` 模式）；另一种工作模式则是只对 `metadata` 记录日志，而不对数据进行日志记录，也即所谓 `data=ordered` 或者 `data=writeback` 模式。系统管理人员可以根据系统的实际工作要求，在系统的工作速度与文件数据的一致性之间作出选择。

8.2 Linux 磁盘管理

磁盘管理是 Linux 操作系统重要的系统管理任务之一。磁盘管理包括磁盘分区、制作文件系统、检测分区、检测磁盘的空闲空间、挂载和卸载文件系统和磁盘同步等工具。

8.2.1 显示与设定硬盘的参数 —— `hdparm` 命令

`hdparm` 用于显示和设定硬盘的参数，它可检测、显示与设定 IDE 或 SCSI 硬盘的参数。语法如下：

`hdparm [-CfghilqtTvyYZ][-a <快取分区>][-A <0 或 1>][-c ][-d <0 或 1>][-k <0 或 1>][-K <0 或 1>][-m <分区数>][-n <0 或 1>][-p ][-P <分区数>][-r <0 或 1>][-S <时间>][-u <0 或 1>][-W <0 或 1>][-X <传输模式>][设备]`

`hdparm` 命令的常用参数如表 8-1 所示。

表 8-1 `hdparm` 命令的常用选项

选项	描述
<code>-a</code> <快取分区>	设定读取文件时，预先读取的磁盘块数，若不加上<快取分区>选项，则显示目前的设定
<code>-A</code> <0或1>	启动或关闭读取文件时的快取功能
<code>-c</code>	设定IDE 32位I/O模式
<code>-C</code>	检测IDE硬盘的电源管理模式
<code>-d</code> <0或1>	设定磁盘的DMA模式
<code>-f</code>	将内存缓冲区的数据写入硬盘，并清除缓冲区
<code>-g</code>	显示硬盘的柱面，磁头，扇区等参数
<code>-i</code>	显示硬盘的硬件规格信息，这些信息是在开机时由硬盘本身所提供
<code>-m</code> <扇区数>	设定硬盘多重分区存取的分区分数
<code>-n</code> <0或1>	忽略硬盘写入时所发生的错误
<code>-p</code>	设定硬盘的PIO模式
<code>-v</code>	显示硬盘的相关设定
<code>-X</code> <传输模式>	设定硬盘的传输模式
<code>-y</code>	使IDE硬盘进入省电模式
<code>-Y</code>	使IDE硬盘进入睡眠模式

例：显示硬盘/dev/hda的硬盘参数。命令如下：

```
[root@wang root]# hdparm /dev/hda
```

结果如下：

```
:
/dev/hda:
multcount    = 16 (on)
IO_support   = 0 (default 16-bit)
unmaskirq    = 0 (off)
using_dma    = 1 (on)
keepsettings = 0 (off)
readonly     = 0 (off)
readahead    = 8 (on)
geometry     = 9729/255/63, sectors = 156301488, start = 0
```

结果表明，该硬盘的支持 DMA 模式的预读的扇区数为 8，总扇区数（sector）为 156301488。

## 8.2.2 Linux 下的硬盘分区工具 —— fdisk 命令

在微机系统中主要有两类硬盘：IDE 硬盘和 SCSI 硬盘。IDE 硬盘价格相对便宜，磁盘容量较大（可达数百 GB），但读写速度较慢。SCSI 硬盘价格相对贵，磁盘容量较小（可达数十 GB），而读写速度快，而且需要附加专用的 SCSI 接口卡。一般个人用户的计算机都配置 IDE 硬盘，而服务器为了提高性能，可能会选用 SCSI 硬盘。

无论是 IDE 硬盘，还是 SCSI 硬盘，在使用前都需要进行分区。所谓分区就是将一块硬盘划分成一个或者多个逻辑块，每一块称为一个分区，根据用途，分区可选用不同类型。这些分区信息存放在硬盘的第 0 扇区，该扇区通常称为分区表。

Linux 至少需要一个分区（也称为根文件分区），如果使用交换分区（swap），将会提高系统的性能。对于与 Intel 兼容的硬件，系统在引导时 BIOS 仅能访问磁盘的前 1024 个柱面。由此，对于大硬盘，通常需要创建一个较小的分区（约 100MB），放在 1024 柱面以内，用于 Linux 系统的引导，引导分区被挂接在/boot 目录上，用于存储系统内核和启动时一些辅助文件。为了管理、备份、测试和安全因素，可能会建立多个磁盘分区。

fdisk 是一个菜单驱动的，创建和操纵分区表的程序。它可以操纵 DOS 类型分区表、BSD 或者 SUN 类型的磁盘标签。

硬盘设备通常是如下的设备名之一：

- /dev/hda
- /dev/hdb
- /dev/sda
- /dev/sdb

其中/dev/hd[a-h]表示 IDE 类型磁盘，/dev/sd[a-p]代表 SCSI 类型的磁盘，/dev/ed[a-d]代表 ESDI 磁盘，/dev/xd[ab]代表 XT 磁盘，这些设备名都代表整个磁盘。

一个分区名由一个设备名后加一个分区号组成。例如/dev/hda1 是系统中第一张 IDE 硬



盘的第一个分区。磁盘最多可以有 15 个分区。

对于一个 DOS 分区表可以容纳任意多个分区。在第 0 扇区只能容纳 4 个分区，这些分区也称为主分区（primary partition）。这 4 个分区之一可以是扩展分区（extended partition），在扩展分区中可以任意多个逻辑分区。对于 4 个主分区，无论是否真的存在，它们的分区号分别为 1~4。逻辑分区的分区号从 5 开始编号。

无论是否显示分区表，fdisk 都要对分区表的所有条目进行一致性检查。这个检查验证物理和逻辑的开始和结束位置是否一致，除第一个分区之外，所有其他分区是否从一个柱面的边界开始。

使用 fdisk 的常用格式如下：

fdisk 选项 磁盘对应的设备名

fdisk 常用的选项如表 8-2 所示。

表 8-2 fdisk 命令的常用选项

选项	描述
-l	列出指定磁盘的分区表，然后退出 如果没有指定磁盘，它将显示/proc/partitions文件中包含的所有分区的信息
-u	在显示分区表时，显示扇区数，而不是柱面数
-s partition	在标准输出上以块为单位显示该分区的大小
-v	详细显示指令的执行过程

### 1. 使用 fdisk 命令的 -l 选项显示磁盘的分区表

例：显示/dev/hda 的分区表。执行过程如下：

```
[root@wang root]# fdisk -l /dev/hda
```

结果如下：

```
Disk /dev/hda: 80.0 GB, 80026361856 bytes
255 heads, 63 sectors/track, 9729 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
```

Device	Boot	Start	End	Blocks	Id	System
/dev/hda1	*	1	1020	8193118+	b	Win95 FAT32
/dev/hda2		1021	9729	69955042+	f	Win95 Ext'd (LBA)
/dev/hda5		1021	3570	20482843+	b	Win95 FAT32
/dev/hda6		3571	6120	20482843+	b	Win95 FAT32
/dev/hda7		6121	9729	28989261	b	Win95 FAT32

结果表明，系统中第一块 IDE 硬盘的大小为 80GB，255 磁头、63 扇区和 9729 个柱面。磁盘上共有两个主分区，有三个逻辑分区，所有的分区都为 Windows 95 类型。

### 2. 磁盘分区模式

在 shell 提示符后，以 root 用户身份执行 fdisk /dev/hda 命令，将会进入硬盘分区模式，可以使用如下命令：

- 输入 m。显示所有可用的命令列表。
- 输入 p。显示硬盘分区表。
- 输入 a。设置硬盘的活动分区。
- 输入 n。创建新的硬盘分区。
- 输入 e。硬盘为扩展分区 (extend)。
- 输入 l。显示已知的分区类型。
- 输入 t。改变硬盘分区的分区类型 ID。
- 输入 d。删除硬盘的一个分区。
- 输入 q。分区不存入硬盘, 返回到 shell。
- 输入 w。结束, 并将更改写入硬盘分区表。

**注意:** 无论是创建分区, 删除分区, 还是更改分区属性, 都需要指定分区的编号, 因此建议用户首先使用 p 命令显示分区表, 然后确定要操作的分区编号, 然后再进行下一步操作, 否则可能会因为误操作, 而导致整个分区中数据的丢失。

在分区操作中, 默认单位为柱面, 因此用户需要根据已经建立的分区表确定新建分区的起始柱面和结束柱面号。

### 8.2.3 文件系统创建工具 —— mkfs 命令

当使用 fdisk 进行硬盘分区之后, Linux 还不能直接使用这些新建的分区, 原因是它们的文件系统没有被建立。只有建立文件系统之后, 分区才能被系统使用。

mkfs 是 Linux 操作系统用于建立文件系统的前端工具。mkfs 的格式如下:

```
mkfs [ -V ] [ -t fstype ] [ fs-options ] filesystem [ blocks ]
```

filesystem 要么是一个设备名 (如 /dev/hda1 和 /dev/sdb2), 要么是文件系统的挂接点 (如 /usr 或者 /home)。

在实际中, mkfs 是 Linux 下的一个可用于创建多种文件系统的简单前端程序。在 Red Hat Linux 9 中, 它可以用来创建多种文件系统, 后台的文件系统建立程序包括: mkfs.ext2、mkfs.jfs、mkfs.reiserfs、mkfs.cramfs、mkfs.ext3、mkfs.msdos 和 mkfs.vfat。通常用户可能会用 mkfs.ext2 创建 ext2 文件系统、mkfs.ext3 创建 ext3 文件系统和 mkfs.vfat 创建 Windows 的 FAT32 文件系统。

mkfs 的选项如表 8-3 所示。

表 8-3 mkfs 命令的常用选项

选项	描述
-t fstype	指定需要建立文件系统的类型, Linux 的默认值为 ext2
blocks	用于文件系统的磁盘块数
-c	在创建文件系统之前检查磁盘坏块
-l filename	从文件名 filename 的文件中读取文件坏块列表
-V	详细显示指令的执行过程

例: 在 /dev/hda2 上建一个 VFAT 类型的文件系统, 同时检查是否有坏块存在, 并且显示详细的建立过程。命令如下:

```
mkfs -V -t vfat -c /dev/hda2
```

例：在/dev/hda3 上建一个 ext3 类型的文件系统。命令如下：

```
mkfs -t ext3 /dev/hda3
```

当在指定的分区上创建了文件系统之后，该分区将可以被挂接在文件系统之上，供用户使用，同时也可以对分区进行文件系统检查。

#### 8.2.4 文件系统挂接工具 —— mount 命令

在 Linux 操作系统中，每个分区都是一个文件系统，都有自己的目录层次结构。linux 会将这些分属不同分区的、单独的文件系统按一定的方式，形成一个系统的总的目录层次结构。这里所说的“按一定方式”就是指的挂接。

将一个文件系统的顶层目录挂接到另一个文件系统的子目录上，使它们成为一个整体，称为挂接。把该子目录称为挂载点。

例：假定某一 Linux 系统的根分区和软盘的目录结构分别如图 8-1 和图 8-2 所示。如果将软盘文件系统挂接到根分区/mnt/floppy 目录（也称挂载点），其结果如图 8-3 所示。

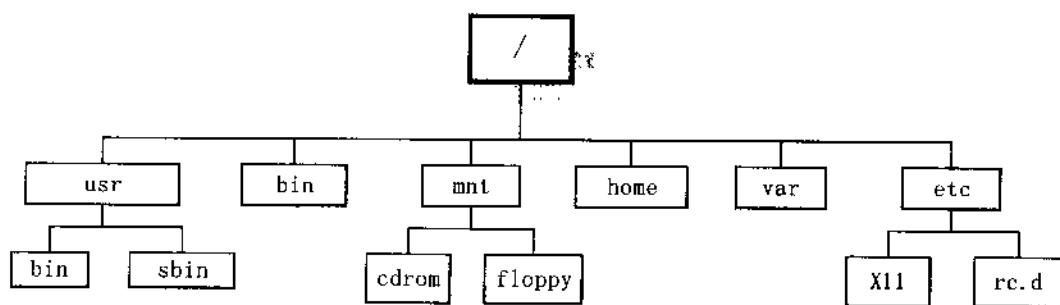


图 8-1 根分区的目录结构示意图

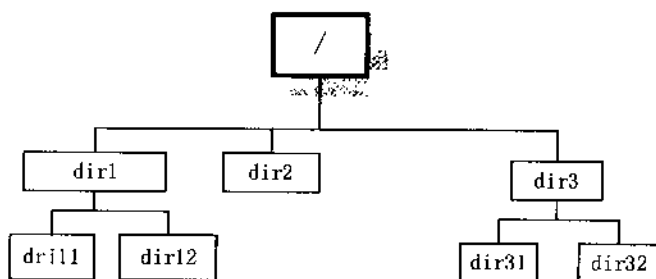


图 8-2 软盘的目录结构

一个分区挂接在一个已存在的目录上，这个目录可以不为空，但挂载后这个目录下以前的内容将不再可用，直到解除挂接。

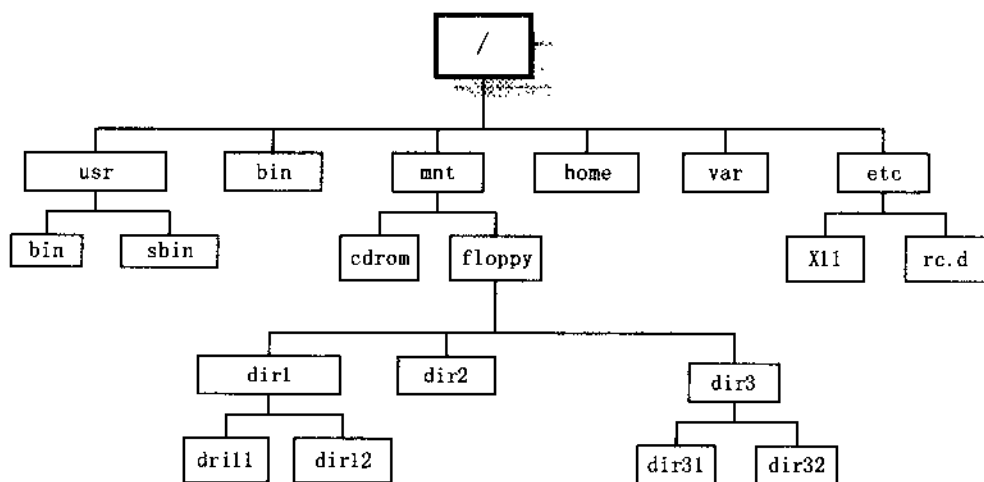


图 8-3 挂接以后的目录结构

对于其他操作系统建立的文件系统，也可以使用相同的方法挂接到 Linux 目录树中。但是需要理解的是：光盘、软盘、其他操作系统使用的文件系统的格式与 linux 使用的文件系统格式是不一样的。光盘是 iso9660；软盘是 fat 或 ext2；Windows NT 是 fat16、NTFS；Windows98 是 fat、vfat；Windows2000 和 Windows XP 是 vfat、NTFS。

**注意：**挂载前要了解 linux 是否支持所要挂载的文件系统格式。

**mount 命令的格式如下：**

```
mount [-hV]
```

```
mount -a [-fFnrsvw] [-t vfstype]
```

```
mount [-fFnrsvw] [-o options [,...]] device | dir
```

```
mount [-fFnrsvw] [-t vfstype] [-o options] device dir
```

mount 的常用选项如表 8-4 所示。

表 8-4 mount 命令的常用选项

选项	描述
-t fstype	指定设备的文件系统类型fstype，常见的文件系统有： ext3 —— Red Hat Linux 9使用的默认文件系统； ext2 —— Linux常用的文件系统； msdos —— MS-DOS的fat，就是fat16； vfat —— Windows98常用的fat32； nfs —— 网络文件系统； iso9660 —— CD-ROM光盘标准文件系统； ntfs —— windows NT/2000的文件系统； auto —— 自动检测文件系统； proc —— 内存文件系统

续表

选项	描述
-o options	指定挂载文件系统时的选项。有些也可用在/etc/fstab中。常见的有： iocharset=XXX —— 文件系统目录和文件的字符集； ro —— 以只读方式挂载； rw —— 以读写方式挂载； nouser —— 使一般用户无法挂载； user —— 可以让一般用户挂载； username —— 远程主机上的登录名； password —— 远程主机上的登录名的口令； sync —— 在同步模式下执行，默认为非同步模式； usbdevfs —— usb磁盘文件系统； loop —— 用来将一个文件当成硬盘分区挂载； noexec —— 不允许在该文件系统中执行任何可执行文件； nosuid —— 不能在该文件系统使用设置uid和gid标志位； nodev —— 不解释该文件系统上的字符和块设备； umask —— 设置文件的umask值，默认为当前进程的umask值； uid, gid —— 设置文件系统中文件中的属主和所属组属性， 默认值为uid=gid=0； mod —— 指定默认文件权限
-V	显示程序版本
-h	显示辅助信息
-a	将/etc/fstab中定义的所有文件系统挂上
-F	通常和-a一起使用，它会为每一个mount的动作产生一个进程负责执行。在系统需要挂上大量NFS文件系统时可以加快挂载的速度
-f	通常用于排错，它会使mount并不执行实际挂上的动作，而是模拟整个挂载的过程。通常会和-v一起使用
-w	等于-o rw
-L	挂上含有特定标签的硬盘分区
-t	指定文件系统的类型，通常不必指定。mount命令会自动选择正确的类型

常见的用法如下：

(1) 使用不带参数的 mount 命令查看系统已挂载的文件系统。

例：显示名为 toshiba 主机已经挂载的文件系统。命令如下：

```
[globus@toshiba globus]$ mount
```

结果如下：

```
/dev/hda3 on / type ext3 (rw)
none on /proc type proc (rw)
usbdevfs on /proc/bus/usb type usbdevfs (rw)
none on /dev/pts type devpts (rw,gid=5,mode=620)
none on /dev/shm type tmpfs (rw)
/dev/hda1 on /mnt/2003 type ntfs (ro,noexec,nosuid,nodev,
```

```
iocharset=gb2312,umask=0222,uid=501)
/dev/hda2 on /mnt/data type vfat (rw,noexec,nosuid,nodev,
iocharset=gb2312,uid=501,mod=777)
```

对于每一行的输出格式如下：

分区名 on 挂载点 type 文件系统类型 (挂载的选项)

上述输出分区的信息如表 8-5 所示。其中有很多分区类型是安装时自动生成的，保持默认值即可，如/dev/hda3 分区被用于根文件系统，分区类型为 ext3。除此之外，还有默认产生的 proc、usbdevfs、devpts 和 tmpfs 类型分区。

上例结果中的最后两个分区分别为/dev/hda1 和/dev/hda2。/dev/hda1 分区类型为 Windows 的 ntfs 分区，它是本机的 Windows 2003 的系统目录，挂载点为/mnt/2003 目录，使用了 ro,noexec,nosuid,nodev,iocharset=gb2312,umask=0222,uid=501 选项，表示该文件系统具有只读属性，文件不具有执行属性，文件系统的字符集为简体中文，umask 为 0222，uid 为 501，表示 uid 为 501 的用户可以完全控制该文件系统。/dev/hda2 与/dev/hda1 类似，只是它的挂载点为/mnt/data，文件类型为 vfat，而且任何用户对所有文件的操作权限为读、写和执行。

表 8-5 上例中已挂载的分区分析

分区名	挂载点	分区类型	挂载选项
/dev/hda3	/	ext3	Rw
无	/proc	Proc	Rw
usbdevfs	/proc/bus/usb	usbdevfs	Rw
无	/dev/pts	devpts	rw,gid=5,mode=620
无	/dev/shm	tmpfs	Rw
/dev/hda1	/mnt/2003	ntfs	ro,noexec,nosuid,nodev,iocharset=gb2312, umask=0222,uid=501
/dev/hda2	/mnt/data	vfat	rw,noexec,nosuid,nodev,iocharset=gb2312, uid=501,mod=777

(2) 挂载/etc/fstab 文件中的所有文件系统，此时必须以 root 用户执行 mount，命令如下：

```
# mount -a
```

(3) 挂载本地的文件系统。

当挂载本地的文件系统时，可以使用如下形式的 mount 命令：

```
# mount -o iocharset=gb2312 分区名 挂载点
```

注意：如果系统中不存在所需的挂载点目录，那么用户需要首先创建这个目录。

如果用户希望挂载 ext2 或者 ext3 类型的文件系统，不需要使用 -o iocharset=gb2312 选项，当挂载简体中文的 Windows 的 vfat 或者 ntfs 文件系统时，需要使用该选项，否则显示中文的文件名和目录名时，会出现乱码。

使用 mount 进行分区挂载时，需要以根用户身份执行 mount 命令。

例：查看系统中硬盘/dev/hdc 中的 Windows 分区，并挂载其中的一个文件系统。过程如下：

首先切换到 root 用户。命令如下：

```
[globus@wang globus]$ su -
```

Password:

```
[root@wang root]#
```

然后显示硬盘/dev/hdc 中的分区信息。命令如下：

```
[root@wang root]# fdisk -l /dev/hdc
```

结果如下：

```
Disk /dev/hdc: 40.0 GB, 40027029504 bytes
255 heads, 63 sectors/track, 4866 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
```

Device	Boot	Start	End	Blocks	Id	System
/dev/hdc1	*	1	765	6144831	7	HPFS/NTFS
/dev/hdc2		766	4866	32941282+	f	Win95 Ext'd (LBA)
/dev/hdc5		1340	2201	6923983+	b	Win95 FAT32
/dev/hdc6		2202	3063	6923983+	b	Win95 FAT32
/dev/hdc7		3064	3925	6923983+	b	Win95 FAT32
/dev/hdc8		3926	4866	7558551	b	Win95 FAT32
/dev/hdc9		766	1275	4096512	83	Linux
/dev/hdc10		1276	1339	514048+	82	Linux swap

在/mnt 目录下建立一个名为 mydir 目录，作为挂接点。

```
[root@wang root]# mkdir /mnt/mydir
```

将/dev/hdc5 挂接到挂接点/mnt/mydir 上。命令如下：

```
[root@wang root]# mount -o iocharset=gb2312 /dev/hdc5 /mnt/mydir
```

验证分区/dev/hdc5 已被挂接，而且已经能支持中文，mount 会自动能够确定文件系统的类型：

```
[root@wang root]# ls /mnt/mydir
```

结果如下：

BCVS最新版	Insight	needtodelete	Program Files	发表论文
BCVS终极版	JavaExample	Oracle	Recycled	周蕙
db1.mdb	matlab61	Oracle9iAS	spss10.0	讲义.pdf
ftpclient.c	msc2002论文集.nrg	pagefile.sys	十日拍拖手册	

注意：Red Hat Linux 9 默认不支持 ntfs 分区的挂接。

对于本地的 NTFS 分区，只有在 Linux 中安装了 ntfs 的驱动模块后才能挂接。

有两种方法安装 ntfs 驱动程序，其一是重新编译和安装 Linux 的内核，设置使之支持 ntfs，其二是在 ntfs 的站点上下载安装包，然后再安装。

可以使用 umount 解挂已挂接的分区。

命令 umount 解挂已挂接的分区，可以通过如下两种方式实现：

- umount 分区名

### ● umount 挂接点

解挂刚挂接的/dev/hdc5 分区, 命令如下:

```
[root@wang root]# umount /dev/hdc5
```

或者

```
[root@wang root]# umount /mnt/mydir
```

#### (4) 挂接映像文件。

将某个映像文件的内容虚拟成文件系统, 然后将其挂在某个目录位置之上。当这个命令执行成功后, 直到用户使用 `umount` 将这个文件系统解挂为止。

用户可以使用 `-o loop` 选项将某个普通文件当成硬盘分区挂上系统。这个功能对于 `ramdisk`、`romdisk` 或 `ISO 9660` 的映像文件的虚拟非常实用和方便。

例: 将 `/tmp/fedora1.iso` 这个光盘的映像文件使用 `loop` 模式挂在 `/mnt/fl` 之下。

```
[root@wang root]# mount -o loop,iocharset=gb2312 /tmp/image.iso /mnt/fl
```

注意: 用户用这种方法, 可以将网络上找到的 `ISO` 文件在不刻录成光盘的情况下, 直接挂接后使用。

#### (5) 挂接 Windows 的共享目录。

在一个局域网环境里, 往往有多台装有 `Windows` 操作系统的计算机, 它们会为其他计算机提供一些共享目录, 或者在 `Linux` 系统中使用 `smb` 共享目录。那么如何在 `Linux` 中使用这些共享的目录呢?

用户可以在 `Linux` 中使用 `smbclient` 程序来完成这一任务, 但是由于这种方式并没有将共享文件变成文件系统的一部分, 因此会带来操作上的不便。

`Linux` 的 `mount` 命令也提供了将共享目录映射成 `Linux` 的一个文件系统的能力。

映射 `Windows` 文件系统成文件系统的 `mount` 命令格式如下:

```
mount -t smbfs -o username=用户名,password=口令 共享名 挂接点
```

例: 在主机 `wang` 上, 将局域网内的主机名为 `jzh` 上的共目录 `temp` 映射成文件系统, 使用 `jzh` 主机上的用户 `administrator`, 其口令为 `1234567`, `wang` 上的挂接点为 `/mnt/smbfs`。过程如下:

如果 `/mnt/smbfs` 不存在, 创建 `/mnt/smbfs` 目录:

```
[root@wang root]# mkdir /mnt/smbfs
```

执行实际的挂接过程:

```
[root@wang root]# mount -t smbfs -o username=administrator,
password=1234567 //jzh/temp /mnt/smbfs
```

注意: 在挂接 `smbfs` 类型的分区时, 不需要使用 `iocharset` 选项来设置字符集。

在 `mount` 命令中, 除了可以使用提供共享的计算机 `NETBIOS` 名 (通常所说的计算机名), 用户也可以使用计算机的 `IP` 地址。

为了安全其间, 可以在 `mount` 命令中不使用 `password` 选项, 在执行 `mount` 命令时, 它会提示用户输入远程用户的口令。

#### (6) 挂接 USB 盘。

为了方便用户, `mount` 也提供了挂接 `U` 盘和移动硬盘的能力。

在 `Linux` 中, `U` 盘和移动硬盘的设备名 `/dev/sda` 和 `/dev/sdb` 等。



```
[root@wang root]# mkdir /mnt/usb
[root@wang root]# mount -o iocharset=gb2312 /dev/sda1 /mnt/usb
```

显示 U 盘中的内容:

```
[root@wang root]# ls /mnt/usb
2003xmb.doc    万能五笔2000智能输入.exe          教师反馈卡.doc
5_4            二00四学年第二学期期末考试(A).doc  新建文件夹
import         二00四学年第二学期期末考试(B).doc  电子现金.doc
vfp6           信息系统分析与设计.DOC            电子现金(old).doc
xiaojie1.doc   小结.doc                          背面.doc
xiaojie.doc    年度考核格式2003.doc              马来西亚之行总结.doc
ZhiNengWuBi    幻灯片
```

查看目前已挂接的文件系统:

```
[root@wang root]# mount
/dev/hdc9 on / type ext3 (rw)
none on /proc type proc (rw)
usbdevfs on /proc/bus/usb type usbdevfs (rw)
none on /dev/pts type devpts (rw,gid=5,mode=620)
none on /dev/shm type tmpfs (rw)
/dev/hda5 on /mnt/d type vfat (rw,iocharset=gb2312)
/dev/hda6 on /mnt/e type vfat (rw,iocharset=gb2312)
/dev/hda7 on /mnt/f type vfat (rw,iocharset=gb2312)
/dev/hda1 on /mnt/c type vfat (rw,iocharset=gb2312)
/dev/sda1 on /mnt/usb type vfat (rw,iocharset=gb2312)
```

(7) 对于本地 Windows 的 VFAT 和 NTFS 分区, 当使用 mount 挂接到 Linux 文件系统时, 默认的文件主为 root, 文件属主为 root, 访问权限为 -rwxr-xr-x, 即 root 具有读、写和执行权限, root 组的用户具有读和执行权限, 其他用户只有读权限。

例: 显示 (6) 中挂接的 U 盘上的文件, 来验证文件的权限信息。命令如下:

```
[root@wang root]# ls -l /mnt/usb/2003xmb.doc
-rwxr-xr-x 1 root root 111104 2月 16 16:11 /mnt/usb/2003xmb.doc
```

对于桌面应用来说, 普通用户在工作时应该使用普通用户登录, 这样普通用户往往无法修改挂接的 VFAT 和 NTFS 分区上的文件。可以在挂接时用 -o uid 或者 -o gid 指定文件主和文件属性来实现。

例: 在主机 wang 上, 对挂接的 USB 盘, 使 globus 用户能完全控制该分区。

首先解挂/dev/sda1, 命令如下:

```
[root@wang root]# umount /mnt/usb
```

挂接分区/dev/sda1 时, 将 uid 改为 globus 用户对应的 uid。过程如下:

查看 globus 对应的 uid:

```
[root@wang root]# id globus
uid=501(globus) gid=501(globus) groups=501(globus)
```

上述结果表明, gloubs 的 uid 为 501。

使用 mount 命令挂接/dev/sda1:

```
[root@wang root]# mount -o iocharset=gb2312,uid=501 /dev/sda1 /mnt/usb
```

例: 显示 U 盘上的文件来验证文件的权限信息。命令如下:

```
[root@wang root]# ls -l /mnt/usb/2003xmb.doc
```

```
-rwxr-xr-x 1 globus root 111104 2月 16 16:11 /mnt/usb/2003xmb.doc
```

注意: 在 Red Hat Linux 9 中, 用户可以使用 id 命令显示指定用户 id, 组 id 和附属组的信息。

### 8.2.5 磁盘空间管理工具 —— df 和 du 命令

系统软件和应用软件都以文件的形式存储在计算机的磁盘中, 而且占用一定的磁盘空间。因此, 应该随时监视磁盘空间的使用情况。Linux 系统提供了一组有关磁盘空间管理的命令。

#### 1. df 命令

df 命令检查文件系统的磁盘空间占用情况。可以利用该命令来获取硬盘已占用了多少空间, 目前还剩下多少空间等信息。

语法: df [选项]

df 命令可显示所有文件系统对 i 节点和磁盘块的使用情况。

该命令的选项如表 8-6 所示。

表 8-6 df 命令的选项

选项	描述
-a	显示所有文件系统的磁盘使用情况, 包括 0 块 (block) 的文件系统 (如/proc)
-k	以 k 字节为单位显示
-i	显示 i 节点信息, 而不是磁盘块
-t	显示各指定类型的文件系统的磁盘空间使用情况
-x	列出不是某一指定类型文件系统的磁盘空间使用情况 (与 t 选项相反)
-T	显示文件系统类型
-h	以人类可读的格式显示文件系统的空间情况

例: 显示当前系统中的所有文件系统的使用情况。命令如下:

```
[root@wang root]# df
```

文件系统	1K-块	已用	可用	已用%	挂载点
/dev/hdc9	4032064	2425416	1401824	64%	/
none	95312	0	95312	0%	/dev/shm
/dev/hda5	20472816	17012560	3460256	84%	/mnt/d
/dev/hda6	20462832	16596408	3866424	82%	/mnt/e
/dev/hda7	28960944	26787456	2173488	93%	/mnt/f
/dev/hda1	8177128	7491276	685852	92%	/mnt/c
/dev/sda1	32128	10734	21394	34%	/mnt/usb

上述结果中，记录了每一个文件系统分区的分区名，以 KB（1KB=1024 字节）为单位显示整个文件系统的容量、已用的容量和可用的容量，磁盘空间使用百分比，以及分区的挂载点目录。

例：以人的阅读习惯显示当前系统中的所有文件系统的使用情况。命令如下：

```
[root@wang root]# df -h
```

文件系统	容量	已用	可用	已用%	挂载点
/dev/hdc9	3.9G	2.4G	1.4G	64%	/
none	94M	0	94M	0%	/dev/shm
/dev/hda5	20G	17G	3.3G	84%	/mnt/d
/dev/hda6	20G	16G	3.7G	82%	/mnt/e
/dev/hda7	28G	26G	2.1G	93%	/mnt/f
/dev/hda1	7.8G	7.2G	670M	92%	/mnt/c
/dev/sda1	32M	11M	21M	34%	/mnt/usb

上述结果表明，磁盘容量，已用的容量和可用的容量都是以人类可读的方式显示，它们以 K、M 和 G 来显示这些容量。

注意：B，K，M 和 G 之间有如下关系：

$$1\text{KB}=2^{10}\text{B}=1024\text{B}$$

$$1\text{MB}=2^{10}\text{KB}=1024\text{KB}$$

$$1\text{GB}=2^{10}\text{MB}=1024\text{MB}$$

## 2. du 命令

du 的英文原义为 disk usage，含义为显示磁盘空间的使用情况。du 统计目录（或文件）所占磁盘空间的大小。

语法如下：

```
du [选项] [Names...]
```

该命令逐级进入指定目录的每一个子目录，并显示该目录占用文件系统数据块（每一个数据块为 1024 字节）的情况。若没有给出 Names，则对当前目录进行统计。

该命令的各个选项含义如表 8-7 所示。

表 8-7 du 命令的选项

选项	描述
-s	对每个Names参数只给出占用的数据块总数
-a	递归地显示指定目录中各文件及子目录中各文件占用的数据块数。若既 不指定-s，也不指定-a，则只显示Names中的每一个目录及其中的各子目 录所占的磁盘块数

续表

选项	描述
-b	以字节为单位列出磁盘空间使用情况（系统默认以K字节为单位）
-k	以1024字节为单位列出磁盘空间使用情况
-c	最后再加上一个总计（系统默认设置）
-l	计算所有的文件大小。对硬链接文件，则计算多次
-x	跳过在不同文件系统上的目录，不予统计
-h	以人类可读的格式显示文件系统的空间情况

例：计算/home所占用的总容量。

```
[root@wang root]# du -s /home
35588    /home
```

结果表明，/home 占用了 35588KB 的磁盘空间。

例：以人类可读的形式显示/tmp 目录所占用的磁盘空间。

```
[root@wang root]# du -s -h /tmp
1.2M    /tmp
```

结果表明/tmp 目录里的文件和目录共占用 1.2M 字节的磁盘空间。

例：显示/tmp 目录所占用的磁盘空间，以字节为单位。

```
[root@wang root]# du -s -b /tmp
1249280 /tmp
```

结果表明/tmp 目录里的文件和目录共占用 1249280 (1249280/1024/1024M=1.2M) 字节的磁盘空间。

例：显示/usr 目录中的每一个子目录占用的磁盘空间。

```
[root@wang root]# du /usr
```

因为显示的结果很长，所以省略。

### 8.2.6 检查与修复 Linux 文件系统 —— fsck 命令

由于异常关机等原因造成的文件系统的逻辑或者物理故障，可以使用 fsck 命令进行检查和修复。

fsck 是 Linux 操作系统用于检查文件系统的前端工具。fsck 的格式如下：

```
fsck [-sACVRP] [-t fstype] [--] [fsck-options] filesystems [...]
```

该命令检查与修复 Linux 文件系统，可以同时检查一个或多个 Linux 文件系统

filesystems 要么是一个设备名（如/dev/hda1 和/dev/sdb2），要么是文件系统的挂接点（例如 /、/usr 或者/home）。

在实际中，fsck 是 Linux 下的一个可用于检查多种文件系统的简单前端程序，在 Red Hat Linux 9 中，它可以用来检查多个文件系统。

fsck 的选项如表 8-8 所示。

表 8-8 fsck 命令的常用选项

选项	描述
-t	给定文件系统的类型, 若在/etc/fstab中已有定义或kernel本身已支持, 则不需加上此参数
-s	依次执行fsck的指令来检查
-A	对/etc/fstab中所有列出来的分区(partition)做检查
-C	显示完整的检查进度
-d	显示出e2fsck的调试(debug)结果
-p	同时有-A条件时, 同时有多个fsck的检查一起执行
-R	同时有-A条件时, 不检查/
-V	详细显示模式
-a	如果检查有错则自动修复
-r	如果检查有错则, 由用户回答是否修复

例: 使用 mount 查看挂接的文件系统。命令如下:

```
[root@wang root]# mount
/dev/hdc9 on / type ext3 (rw)
none on /proc type proc (rw)
usbdevfs on /proc/bus/usb type usbdevfs (rw)
none on /dev/pts type devpts (rw,gid=5,mode=620)
none on /dev/shm type tmpfs (rw)
/dev/hda5 on /mnt/d type vfat (rw,iocharset=gb2312)
/dev/hda6 on /mnt/e type vfat (rw,iocharset=gb2312)
/dev/hda7 on /mnt/f type vfat (rw,iocharset=gb2312)
/dev/hda1 on /mnt/c type vfat (rw,iocharset=gb2312)
/dev/sda1 on /mnt/usb type vfat (rw,iocharset=gb2312,uid=501)
```

根据上述的结果, 检查/dev/hda5 分区的文件系统, 执行命令如下:

```
[root@wang root]# fsck /mnt/d
```

开始检查文件系统, 过程如下:

```
fsck 1.32 (09-Nov-2002)
dosfsck 2.8, 28 Feb 2001, FAT32, LFN
Warning: FAT32 support is still ALPHA.
There are differences between boot sector and its backup.
Differences: (offset:original/backup)
    65:01/00
1) Copy original to backup
2) Copy backup to original
3) No action
?
```

该命令提示引导扇区与其备份扇区的内容不同, 问对引导扇区做备份、使用备份更新

引导扇区，还是忽略这些差异，在此输入 3 忽略，并按回车键继续检查过程。

```
/:7Kr:84H:6Ab:30A:9xH:5kY:5b7:6A0:8rY:5td:5ad:6GS:6DL:30B:559:7RO:7o+:7k
0:798/TOOLS
```

```
Has a large number of bad entries. (3/5)
```

```
Drop directory ? (y/n)
```

提示有许多错误的条目，是否删除/tools 目录，在此选 n，不删除继续检查。

以后将有许多类似的提示，选 y 进行更新，否则保持不变。

fsck 最有用的一个应用是检查 Linux 特有的 ext2 和 ext3 类型的文件系统。经常出现突然停电，或者系统死锁被迫关闭计算机的情况，Linux 系统对文件系统采用了延迟写技术，即对文件的修改不会立即写入磁盘，而放在一个文件缓冲区内，然后对一批缓冲的数据一次写入磁盘，这样减少了磁盘的读写次数，能提高磁盘的效率。当系统关闭时，内核会将所有没有写入磁盘的数据写入磁盘，如果不正常关机，就会导致一些数据无法写入磁盘。每当这样情况发生时，在下次启动时，系统将会提示是否进行文件系统检查，这样检查是非交互的，因此即使对文件系统检查，也可能无法修复磁盘。

在应急模式或者正常模式下，采用交互式文件系统检查方式检查 ext2 或者 ext3 类型文件系统，当出现文件的 i 节点出错时，fsck 会提示是否重建一个，或者是删除该 i 节点，这时用户可以根据情况给出回答，从而完成磁盘修复，最后恢复系统的正常运行。

例：在线检查/文件系统。

```
[root@wang root]# fsck /
```

结果如下：

```
fsck 1.32 (09-Nov-2002)
```

```
e2fsck 1.32 (09-Nov-2002)
```

```
/dev/hdc9 is mounted.
```

```
WARNING!!! Running e2fsck on a mounted filesystem may cause SEVERE
filesystem damage.
```

```
Do you really want to continue (y/n)? yes
```

提示/dev/hdc9 已经挂接，如果对挂接的文件系统运行 fsck，可能导致文件系统的严重损坏。提示是否继续进行，在此选择 y，继续进行检查。结果如下：

```
/: recovering journal
```

```
/: clean, 129873/513024 files, 621761/1024128 blocks
```

结果表明，文件系统没有任何问题 (clean)。

**注意：**由于掉电导致文件系统不一致，通过 fsck 检查通常可以恢复文件系统，原因是缓存的文件往往是日志文件和临时文件，所以对系统的下一次运行没有致命性影响。

### 8.2.7 调整 Linux 文件系统参数 —— tune2fs 命令

Linux 提供了 tune2fs 命令来调整 ext2 文件系统的参数。tune2fs 的选项-j 可以在 ext2 文件系统中加入 ext3 文件系统的日志 (journal) 功能，这时用户必须使用一个支持 ext3 文

件系统的内核，Red Hat Linux 9 默认支持 ext3 文件系统。

例：将/dev/hdc9 文件系统（ext2）转化为 ext3 文件系统。命令如下：

```
[root@wang root]# tune2fs -j /dev/hdc9
```

如果该文件系统本身就是 ext3 文件系统，则会出现如下提示信息：

```
tune2fs 1.32 (09-Nov-2002)
```

```
The filesystem already has a journal.
```

### 8.2.8 同步文件系统 —— sync 命令

为了提高效率，Linux 系统将写入硬盘的数据首先写到文件系统的缓冲区（buffer）中。这个 buffer 是一块内存区域，如果将写入硬盘的数据存于该 buffer 中，而系统又突然断电的话，数据就会流失了，sync 命令会将存于 buffer 中的数据强制写入硬盘中。

例：将文件系统的缓冲区的数据写入磁盘。

```
[root@wang root]# sync
```

### 8.2.9 启用/停用交换分区 —— swapon 命令

使用方式如下：

```
/sbin/swapon -a [-v]
```

```
/sbin/swapon [-v] [-p priority] specialfile
```

```
/sbin/swapon [-s]
```

swapon 的选项如表 8-9 所示。

表 8-9 swapon 命令的常用选项

选项	描述
-h	获得帮助信息
-V	显示版本信息
-s	显示设备的使用情况
-a	自动启用所有swap类型的分区
-p	设定优先权，用户可以在0到32767中间选一个数字，或是在/etc/fstab里面加上pri=[value]（[value]就是0~32767中间一个数字），然后就可以很方便的直接使用swapon -a启动它们，而且有优先权设定

### 8.2.10 /etc/fstab 文件

/etc/fstab 文件是存放文件系统与挂接点对应的文件，超级用户 root 可以使用编辑器来修改/etc/fstab，/etc/fstab 可以更方便的管理需要挂接的文件系统。在 fstab 文件对挂接的分区进行适当的配置，不仅可以在系统启动时自动挂接某些文件系统，而且可以简化 mount 命令的执行。

/etc/fstab 文件由若干行组成，每一行代表一个文件系统及其挂接信息，每一行都是由若干个字段组成，字段之间以空格或者制表符（Tab）分隔。

/etc/fstab 文件的每一行的组成如表 8-10 所示。

表 8-10 fstab 文件包含的字段说明

字段编号	字段名称	描述
1	fs_spec	实际的设备 (device) 名称
2	fs_file	对应的挂接点 (mount point), 它是一个目录名称
3	fs_vfstype	该分区 (partition) 的文件系统类型, 常见的有: ext3、vfat、iso9660、nfs、swap、ntfs、smbfs
4	fs_mntops	在挂接 (mount) 时的选项
5	fs_freq	在使用dump时是否记录, 不需要则输入0
6	fs_passno	决定在开机时执行fsck的先后顺序

下述文件是一个 fstab 的例子。内容如下:

```

LABEL=/      /      ext3 defaults          1 1
none         /dev/pts  devpts gid=5,mode=620  0 0
none         /proc    proc  defaults           0 0
none         /dev/shm tmpfs  defaults            0 0
/dev/hda5    swap     swap  defaults            0 0
/dev/cdrom    /mnt/cdrom udf,iso9660 iocharset=gb2312,kudzu,ro 0 0
/dev/sda1     /mnt/usb   vfat  noauto,iocharset=gb2312,
              users,uid=501,mode=777      0 0
/dev/hda1     /mnt/2003  ntfs  iocharset=gb2312,ro,umask=0222,
              users,uid=501           0 0
/dev/hda2     /mnt/data  vfat  iocharset=gb2312,users,uid=501,
              mod=777                  0 0
//jiang/temp  /mnt/smbfs smbfs noauto,username=administrator,
              users,uid=501,gid=501,mod=777 0 0

```

从这个 fstab 文件中可以看出, 系统中至少有一块 IDE 硬盘, 而且该硬盘包含一个 ntfs 分区 (/dev/hda1) 和一个 FAT32 (/dev/hda2) 分区, 它们的挂接点分别是 /mnt/2003 和 /mnt/data, 这张 IDE 硬盘上另外还包含一个 Linux 根 (/) 分区和一个交换分区 (/dev/hda5)。

对于分区 /dev/hda1, 其类型为 ntfs, 它的挂接选项为:

```
iocharset=gb2312,ro,umask=0222,users,uid=501
```

其中 iocharset=gb2312 表示字符集为简体中文, ro 表示只读, umask=0222 表示创建文件时的默认权限为 555, uid=501 表示 uid 为 501 的用户可以完全控制该分区的所有文件和目录, 即可以读写其中的任何文件。

对于分区 /dev/hda2, 其类型为 vfat (Windows 中的 FAT32), 它的挂接选项为:

```
iocharset=gb2312,users,uid=501,mod=777
```

它的选项与 /dev/hda2 分区的选项类似, 其中 mod=777 表示所有的用户都可以读、写和执行该分区的任何文件。

注意: 在 Linux 中, 对 ntfs 文件进行写操作是危险的, 因此建议用户不要在 Linux 中写 (修改和删除) ntfs 分区中的任何文件。

除了 IDE 磁盘上的分区, 还包括一个光盘 (/dev/cdrom) 驱动器, 类型 iso9660。



`fstab` 中分区 `/dev/sda1` 表示是 U 盘的第一个分区, `//jiang/temp` 表示主机 `jiang` 上的 Windows 操作系统 (或者 `samba`) 共享名 `temp`。

在 `fstab` 的选项中, `noauto` 表示在系统引导时不自动挂载该文件系统, 否则将在系统引导时自动挂载。

用户可以根据需要在 `fstab` 中添加需要挂载的分区, 也可以删除其中的分区信息。

如果在 `fstab` 中包含了某一个分区的信息, 则可以使用如下的两种方式来挂载和解挂该分区:

- `mount/umount` 分区名
- `mount/umount` 分区的挂载点

例: 对于上述 `fstab` 文件, 挂载 `/dev/hda2`。命令如下:

```
[root@wang root]# mount /hda/hda2
```

或者

```
[root@wang root]# mount /mnt/data
```

### 8.3 磁盘复制命令

`dd` 命令把指定的输入文件拷贝到指定的输出文件中, 并且在拷贝过程中可以进行格式转换。它不仅可以实现 DOS 下的 `diskcopy` 命令的作用, 而且可以用来复制分区的引导扇区和复制可引导的光盘 (CD-ROM) 等功能。

对于复制软盘上的数据, 先用 `dd` 命令把软盘上的数据写成硬盘的一个映像文件, 再把这个映像文件写入第二张软盘上, 从而完成 `diskcopy` 的功能。

注意: 复制完成后, 应该用 `rm` 命令删除硬盘上的映像文件。系统默认使用标准输入文件和标准输出文件。

`dd` 命令的选项如表 8-11 所示。

表 8-11 `dd` 命令的常见选项

选项	描述
<code>if=输入文件 (或设备名称)</code>	需要复制的信息来源
<code>of=输出文件 (或设备名称)</code>	需要复制的目标
<code>ibs=bytes</code>	一次读取的字节数为 <code>bytes</code> , 即读入缓冲区的字节数
<code>skip=blocks</code>	跳过读入缓冲区开头的 <code>ibs*blocks</code> 块
<code>obs=Bytes</code>	一次写入字节数为 <code>bytes</code> , 即写入缓冲区的字节数
<code>bs=bytes</code>	同时设置读/写缓冲区的字节数 (等于设置 <code>ibs</code> 和 <code>obs</code> )
<code>cbs=byte</code>	一次转换字节数为 <code>bytes</code>
<code>count=blocks</code>	只拷贝输入的 <code>blocks</code> 块
<code>conv=ASCII</code>	把 EBCDIC 码转换为 ASCII 码
<code>conv=ebcdic</code>	把 ASCII 码转换为 EBCDIC 码
<code>conv=ibm</code>	把 ASCII 码转换为 alternate EBCDIC 码
<code>conv=block</code>	把变动位转换成固定字符
<code>conv=ublock</code>	把固定位转换成变动位

续表

选项	描述
conv=ucase	把字母由小写转换为大写
conv=lcase	把字母由大写转换为小写
conv=notrunc	不截断输出文件
conv=noerror	出错时不停止处理
conv=sync	把每个输入记录的大小都调到ibs选项指定的大小，不够的用 NUL字符填充

例：将根分区（/dev/hda3）的引导扇区复制到/mnt/data目录下，取名为 redhat.lnx。命令如下：

```
[root@wang root]# dd if=/dev/hda3 of=/mnt/data/redhat.lnx bs=512 count=1
```

## 思考与练习

- (1) Linux 支持哪些常见的文件系统？
- (2) Linux 专用的文件系统有哪些？有什么特性？
- (3) 如何挂接和解挂文件系统？对于不同的系统，分别应该注意些什么？
- (4) /etc/fstab 的作用是什么？

## 第9章 用户管理

在 linux 中, 每个普通用户都有一个账户, 包括用户名、密码和主目录等信息。除此之外, 还有一些系统本身创建的特殊用户, 它们具有特殊的意义。其中最重要的是超级用户, 在 linux 中, 它默认用户名是 root。同时 Linux 也提供了用户组, 使每一个用户至少属于一个组, 这样便于进行权限的管理。

Linux 的用户和组管理是 Linux 安全管理的重要组成部分, 本章主要介绍 Linux 提供的用户管理和组管理命令, 如何建立多个超级用户以及为普通用户分配特权的方法。

### 9.1 用户管理的基本操作

在 linux 中, 每个普通用户都有一个账户, 包括用户名、密码和主目录等信息。除此之外, 还有一些系统本身创建的特殊用户, 它们具有特殊的意义。其中最重要的是超级用户, 在 linux 中, 默认的超级用户名是 root。

在 Linux 中管理员可以添加用户、修改用户设置、删除用户、添加组以及删除组。修改用户包括修改用户的组信息、附加组、用户的 shell 类型、用户的口令等信息。

#### 9.1.1 增加用户

Linux 提供了 useradd 来为系统添加新用户信息。useradd 的语法如下:

```
useradd [-c comment] [-d home_dir] [-e expire_date] [-f inactive_time] [-g  
initial_group] [-G group[,...]] [-m [-k skeleton_dir] [-M] [-s shell] [-u uid  
[-o] ] [-n] [-r] login
```

或者

```
useradd -D [-g default_group] [-b default_home] [-f default_inactive] [-e  
default_expire_date] [-s default_shell]
```

在建立新用户账号时, 当不加-D 选项, useradd 命令使用命令行指定的选项, 并结合系统的默认值来创建一个新用户。这个新用户将被加入到系统特定的文件中, 并根据该命令的选项创建用户的主目录, 拷贝初始配置文件。如果不使用-n 选项, useradd 将为每一个用户创建一个与用户同名的组。

useradd 可使用的常见选项如表 9-1 所示。

表9-1 useradd命令的常用选项

选项	意义
-c comment	新账号password文件的说明
-d home_dir	新账号每次登入时所使用的主目录 (home_dir), 默认值为/home/用户名称, 并当成登入时的目录名称
-e expire_date	账号过期日期。日期的指定格式为MM/DD/YY
-f inactive_days	账号过期几日后永久停用。当值为0时账号则立刻被停用; 而当值为-1时, 则关闭此功能, 默认值为-1

续表

选项	意义
-g initial_group	group名称或一个数字作为用户的起始群组(group)。群组名须为现有存在的名称。群组数字也须为现有存在的群组
-G group,[...]	定义此用户为一些附加组的成员。每个群组使用逗号(,)隔开,不可以夹杂空白字符。群组名的限制同-g选项
-m	如果用户目录不存在,则自动创建
-M	不创建用户目录
-n	默认情况下,用户组与用户名称相同,此选项将取消此默认设置
-r	此参数用来建立系统账号。在Red Hat Linux中系统账号的UID小于500。注意useradd此用法所建立的账号不会建立用户主目录。如果要建立用户主目录需使用-m选项
-s shell	用户登入后使用的shell名称。默认为不填写,这样系统会帮你指定默认的登入shell
-u uid	用户的ID值,必须惟一,除非用-o选项。数字不可为负值。默认的最小不得小于99,而且逐次增加。0~99保留给系统账号使用
-D	当-D选项出现时,useradd显示现在的默认值,或者通过命令行方式更改这些默认值

与用户账号信息有关的文件如下:

- /etc/passwd —— 用户账号信息。
- /etc/shadow —— 用户账号信息加密文件
- /etc/group —— 组信息文件
- /etc/default/useradd —— 定义默认设置文件
- /etc/login.defs —— 系统广义设置文件
- /etc/skel —— 默认的初始配置文件目录

例:新建一个用户jiangzhonghua。命令如下:

```
[root@wang root]# useradd jiangzhonghua
```

没有任何提示,表明用户建立成功,这时并没有设置用户的口令,必须使用passwd命令修改用户的密码。

使用id命令查看新建的用户信息,命令如下:

```
[root@wang root]# id jiangzhonghua
uid=502(jiangzhonghua) gid=502(jiangzhonghua)
                                groups=502(jiangzhonghua)
```

修改用户jiangzhonghua的密码:

```
[root@wang root]# passwd jiangzhonghua
```

根据提示两次输入新用户的密码,结果如下:

```
Changing password for user jiangzhonghua.
```

```
New password:
```

```
BAD PASSWORD: it is based on a dictionary word
```

```
Retype new password:
```

```
passwd: all authentication tokens updated successfully.
```

这样完成了口令更改。

**注意：**用 `useradd` 增加一个用户后，应该立刻用 `passwd` 给新用户修改密码，没有设置密码的新账号将不能使用。

每新建一个用户，`/etc/passwd` 文件中都会增加一行信息，其格式为：

用户名:密码:用户ID (UID) :组ID (GID) :全名:用户目录:Shell类型

查看新加入的用户 `jiangzhonghua` 在 `/etc/passwd` 文件中的信息行，命令如下：

```
[root@wang root]# grep jiangzhonghua /etc/passwd
```

```
jiangzhonghua:x:502:502::/home/jiangzhonghua:/bin/bash
```

结果表明用户 `jiangzhonghua` 的 `uid` 和 `gid` 都为 502，主目录为 `/home/jiangzhonghua`，shell 为 `bash`。

用户可以用 `-d` 选项设置新用户的主目录，用 `-g` 选项为用户指定新组名，用 `-G` 选项把新用户设成系统其他一些组的成员。

**例：**新建用户 `newuser`，使其组名为 `mygroup`，主目录为 `/var/myhome`，新用户 `newuser` 也是 `root` 和 `bin` 的成员。过程如下：

首先建立组 `newgroup`。命令如下：

```
[root@wang root]# groupadd mygroup
```

建立用户。命令如下：

```
[root@wang root]# useradd -g mygroup -d /var/myhome -G root,bin newuser
```

**例：**显示 `useradd` 的默认值。命令如下：

```
[root@wang root]# useradd -D
```

```
GROUP=100
```

```
HOME=/home
```

```
INACTIVE=-1
```

```
EXPIRE=
```

```
SHELL=/bin/bash
```

```
SKEL=/etc/skel
```

**注意：**仅仅增加一个新组，可以用命令 `groupadd`。

查看某一个用户的 `uid`、`gid` 和附加组信息，使用 `id` 命令。

修改用户口令的命令为 `passwd`。

查看用户的详细信息的命令为 `grep` 用户名 `/etc/passwd`。

只有超级用户 `root` 才能使用 `useradd` 和 `groupadd` 命令。

除超级用户 `root` 外，用户使用 `passwd` 命令只能更改自己的口令，不带参数的 `passwd` 命令用于修改自己的口令。

### 9.1.2 修改用户账号

`usermod` 命令根据命令选项修改系统账号文件里的相应信息。下列为 `usermod` 可选用的参数。`usermod` 的大多数选项与 `useradd` 命令相同，在此用于修改对应的属性，`usermod` 特有的选项如表 9-2 所示。

表9-2 usermod命令特有的选项

选项	意义
-l login_name	将用户的账号名称更改为login_name。其他属性不变。特别的，用户目录名应该也会跟着更改成与新的用户名同名的目录
-p passwd	指定加密后的口令值，它是使用crypt命令加密后的结果
-L	锁定用户的口令，这将在加密的口令之前加一个感叹号(!)，它有效的禁用了口令，它不能与-p或者-U一起使用
-U	解锁用户的口令，这将在加密的口令之前去掉一个感叹号!，它也不能与-p或者-L一起使用

## (1) 修改密码。

普通用户可以用 passwd 修改自己的密码，只有管理员 root 才能用 passwd username 为其他用户修改密码。

## (2) 修改用户 shell 设置。

使用 chsh 命令可以修改自己的 shell，只有管理员才能用 chsh username 为其他用户修改 shell 设置。

注意：指定的 shell 必须是列入/etc/shells 文件中的 shell，否则该用户将不能登录。用户也可以使用 usermod 命令修改 shell 信息，命令如下：

```
usermod -s [new_shell_path] username
```

其中 new\_shell\_path 和 username 应取相应的值。

例：将用户 jiangzhonghua 的 shell 改为 csh。命令如下：

```
[root@wang root]# usermod -s /bin/csh jiangzhonghua
```

## (3) 修改主目录。

```
usermod -d [new_home_directory] username
```

将用户 jiangzhonghua 的主目录更改为/home/jiangzhonghua，命令如下。

```
[root@wang root]# usermod -d /home/jiangzhonghua jiangzhonghua
```

如果想将现有主目录的主要内容转移到新的目录，应该使用-m 选项，如下所示：

```
usermod -d /new/home -m username
```

## (4) 修改 UID。

```
usermod -u UID username
```

该用户主目录中所拥有的文件和目录都将自动修改 UID 设置。但是，对于主目录外所拥有的文件，只能手工用 chown 命令修改所有权设置。

## (5) 修改默认组设置。

```
usermod -g GID username
```

## (6) 修改账号的有效期。

如果使用了影子口令，则可以使用如下命令来修改一个账号的有效期：

```
usermod -e MM/DD/YY username
```

例如，把用户 jiangzhonghua 的有效期定为 12/31/04，命令如下：

```
usermod -e 12/31/04 username
```

## (7) 禁用用户账号。

如果只是想暂时禁止某个账号，可以使用下列方法：

- 使用无效的 shell。例如使用 `usermod -s newshell username` 将用户的 shell 改为 `/bin/false`（最好把 `/bin/false` 列入 `/etc/shells` 文件里）。
- 使该账号过期。如果使用影子口令，可使用 `usermod -e MM/DD/YY username` 命令使该账号过期。
- 如果想禁止所有账号（root 账号当然除外）的访问，可以创建一个名为 `/etc/nologin` 的文件，说明系统暂时不允许访问。
- 使用 `usermod -L username`，锁定用户。

### 9.1.3 删除或禁止用户账号

使用 `userdel` 命令删除现有用户。

例：下面的命令将删除用户 `jiangzhonghua`。命令如下：

```
[root@wang root]# userdel jiangzhonghua
```

如果想同时删除该用户的主目录以及其中所有内容，要使用 `-r` 开关来递归删除。

注意：无法删除已经进入系统的用户，如果想强行完成，需要先杀死有关的进程，然后再使用 `userdel` 命令。

## 9.2 超级用户授权和创建多个超级用户

9.1 节介绍了用户管理基本操作的内容，下面将继续探讨有关用户管理的高级内容。

### 9.2.1 建立多个超级用户

大多数新系统管理员认为 root 用户是惟一的超级用户，其实 root 只是系统默认的超级用户，root 并非因为它为 root 就成为超级用户。随便打开一个 `/etc/passwd` 文件的例子，你就会发现如下几行：

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
jiangzhonghua:x:0:0::/home/jiangzhonghua:/bin/bash
```

从上面的内容可以发现，root 的 UID 和 GID 都为 0。实际上，超级用户的充要条件就是 UID 和 GID 都等于 0。也就是说，只要用户的 UID 和 GID 都为 0，就与常被称为 root 超级用户一样，如上例中的 `jiangzhonghua` 也是一个超级用户。

因此，可以将任何普通用户变成超级用户。但是，这样做并没有好处，很多时候这都会增加系统的隐患。除非在组织中需要多个系统管理员管理同一个系统，这就需要有多个超级用户账号。这有利于各个管理员明确责任，通过日志知道不同的人分别做过什么事。

还有一种情况，也可能出现多个超级用户账号，那就是黑客入侵后设置一个看起来像普通账号的用户，却修改了 UID 和 GID，使之成为 0。这样根本就不需要知道 root 用户密码，就可以执行超级用户权限了。系统管理员通常不可能时刻注意 passwd 文件的变化，因为这样既费时又费力。可以使用如下的脚本来协助监视：

```
cat /etc/passwd | grep '0:0' | cut -d ":" -f 1 | mail -s "`date` + uid report"
root
```

这是一个很小的脚本程序，使用了一些常规的命令来查看/etc/passwd 文件，把 UID 和 GID 都为 0 的行发送给 root 用户。把这个脚本放在/etc/cron.daily 文件中，让 cron 运行，root 将每天收到一封信，报告当前的超级用户名单。

实际上，由于 PAM（可插入验证模块）的限制，在 telnet 上是不允许超级用户登录的，也就是说，黑客修改了自己的 UID 和 GID 后，想再次登录，并获得超级用户的权限，不修改/etc/securetty 文件是不可能的。一般可以先以普通用户账号登录，然后再使用 su 命令切换到超级用户。

### 9.2.2 为普通用户分配特权

使用 sudo 命令可以允许普通用户执行超级用户才能执行的命令。无论是基于信任的建立需要时间，还是基于是否存在这种必要，管理员都不会把超级用户的所有权限轻易给别人，这是网管工作的原则。所以，当一些用户必须访问某些内容时，管理员可以配置 sudo 以允许单个的普通用户运行特权命令。

sudo 命令允许已经在/etc/sudoers 文件中指定的用户运行超级用户命令。例如，一个已经获得许可的普通用户可以运行：

```
sudo /usr/sbin/useradd newuser1
```

实际上，sudo 的配置完全可以指定某个列入/etc/sudoers 文件的普通用户可以做什么，不可以做什么。

/etc/sudoers 的配置行如下所示。

- 空行或注释行（以#字符打头）：不起动作的行。
- 可选的主机别名行：用来创建主机列表的简称。必须以 Host\_Alias 关键词开头，列表中的主机必须用逗号隔开，如
 

```
Host_Alias linux=wang,jzh
```

 其中 wang 和 jzh 是两个主机名，可以使用 linux（别名）称呼它们。
- 可选的用户别名行：用来创建用户列表的简称。用户别名行必须以 User\_Alias 关键词开头，列表中的用户名必须以逗号隔开。其格式同主机别名行。
- 可选的命令别名行：用来创建命令列表的简称。必须以 Cmnd\_alias 开头，列表中的命令必须用逗号隔开。
- 可选的运行方式别名行：也是用来创建用户列表的简称。不同的是，使用这样的别名可以告诉 sudo 程序以列表中某一用户的身份来运行程序。
- 必要的用户访问说明行。

用户访问的说明语法如下：

```
user host= [run as user ] command list
```



在 `user` 处指定一个真正的用户名或定义过的别名, `host` 也可以是一个真正的主机名或者定义过的主机别名。默认情况下, `sudo` 执行的所有命令都是以 `root` 身份执行。如果你想使用其他身份可以指定。`command list` 可以是以逗号分隔的命令列表, 也可以是一个已经定义过的别名, 如

```
jiangzhonghua wang=/sbin/shutdown
```

这一句说明 `jiangzhonghua` 可以在 `wang` 主机上运行关机命令。

**注意:** 可以在一行定义多个别名, 中间用冒号 (:) 隔开。

可以在命令或命令别名之前加上感叹号 (!), 使该命令或命令别名无效。

有两个关键词: `ALL` 和 `NOPASSWD`。`ALL` 意味着“所有”(所有文件、所有主机或所有命令), `NOPASSWD` 意味着不用密码。

下面是一个 `sudoers` 文件的例子。

```
#sudoers files
#User alias specification
User_Alias ADMIN=jiangzhonghua:POWERUSER=globus,wang
#user privilege specification
ADMIN ALL=ALL
POWERUSER ALL=ALL, !/bin/su
```

第 3 行定义了两个别名 `ADMIN` 和 `POWERUSER`, 第 5 行说明在所有主机上 `ADMIN` 都可以以 `root` 身份执行所有命令。第 6 行给 `POWERUSER` 除了运行 `su` 命令外同等 `ADMIN` 的权限。

## 思考与练习

- (1) 为什么要进行用户管理?
- (2) 如何添加和删除用户组?
- (3) 普通用户和超级用户之间的区别是什么?
- (4) 如何添加、删除用户? 如何修改用户的各种信息?

## 第 10 章 Linux 进程管理

Linux 是一个多用户多任务的操作系统。多用户是指多个用户可以在同一时间使用计算机系统；多任务是指 Linux 可以同时执行几个任务，它可以在还未执行完一个任务时又执行另一项任务。

Linux 内核管理了系统中的所有进程，包括进程的创建、调度以及虚拟内存管理，同时 Linux 为用户提供了一些命令来管理进程，它们包括 ps、who、w、top、kill，它还提供了定期执行作业的两种服务 at 和 cron。

本章首先较为系统地介绍了 Linux 内核的进程管理方式，然后以实例的方式讲解了 Linux 提供的常用的进程控制命令和 at 和 cron 服务，最后介绍了进程查看命令。

### 10.1 内核对进程的管理

进程在操作系统中执行特定的任务。程序是存储在磁盘上包含可执行机器指令和数据的静态实体。进程或者任务是处于活动状态的计算机程序。

本节将讨论 Linux 内核如何在系统中创建、管理以及删除进程。

#### 10.1.1 Linux 的进程

进程是一个随执行过程不断变化的实体。和程序要包含指令和数据一样，进程也包含程序计数器和所有 CPU 寄存器的值，同时它的堆栈中存储着一些临时数据，例如子程序参数、返回地址以及变量等。当前的进程（执行中程序）包含着当前处理器中的活动状态。Linux 是一个多任务操作系统。进程具有独立的权限与职责。如果系统中某个进程崩溃，它不会影响到其余的进程。每个进程运行在其各自的虚拟地址空间中，通过核心控制下可靠的通讯机制，它们之间才能发生联系。

进程在生命期内将使用系统中的资源。它利用系统中的 CPU 来执行指令，用物理内存来放置指令和数据。使用文件系统提供的功能打开并使用文件，同时直接或者间接的使用物理设备。Linux 必须跟踪系统中每个进程以及资源，以便在进程间实现资源的合理分配。如果系统有一个进程独占了大部分物理内存或者 CPU 的使用时间，会影响系统中的其他进程。

CPU 是系统中最宝贵的资源。Linux 是一个多任务操作系统，它最终的目的是：任何时刻系统中的每个 CPU 上都有任务执行，从而提高 CPU 的利用率。如果进程个数多于 CPU 的个数，则有些进程必须等待到 CPU 空闲时才可以运行。当进程需要某个系统资源时它将停止执行并等待到资源可用时才继续运行，操作系统将把 CPU 控制权交给其他可以运行的进程。调度器负责选择适当的进程来运行，Linux 使用一些调度策略以保证合理分配 CPU。

Linux 支持多种类型的可执行文件格式，如 ELF，JAVA 等。由于这些进程必须使用系统共享库，所以对它们的管理要具有透明性。

为了让 Linux 来管理系统中的进程，每个进程用一个 task\_struct 数据结构来表示（任

务与进程在 Linux 中可以混用)。数组 `task` 包含指向系统中所有 `task_struct` 结构的指针。创建新进程时, Linux 将从系统内存中分配一个 `task_struct` 结构并将其加入 `task` 数组。当前运行进程的结构用 `current` 指针来指示。

Linux 还支持实时进程。这些进程必须对外部时间作出快速反应, 系统将区分对待这些进程和其他进程。虽然 `task_struct` 数据结构比较复杂, 其功能组成部分如表 10-1 所示。

表 10-1 `task_struct` 数据结构的组成

组成	描述
State	进程在执行过程中会根据环境来改变state, 状态如下: Running: 进程处于运行或者准备运行状态 Waiting: 进程在等待一个事件或者资源 Stopped: 进程被停止, 通常是通过接收一个信号 Zombie: 这是由于某些原因被终止的进程, 但仍保留在 <code>task_struct</code> 结构中, 它像一个已经死亡的进程
Scheduling Information	调度器根据这些信息判定系统中哪个进程最迫切需要运行
Identifiers	系统中每个进程都有进程标志。进程标志并不是 <code>task</code> 数组的索引, 它仅仅是个数字。每个进程还有一个用户与组标志, 它们用来控制进程对系统中文件和设备的存取权限
Inter-Process Communication	Linux 支持经典的 Unix IPC 机制, 如信号、管道和信号灯以及系统 V 中 IPC 机制, 包括共享内存、信号灯和消息队列
Links	Linux 系统中所有进程都是相互联系的。每个进程对应的 <code>task_struct</code> 结构中包含有指向其父进程和兄弟进程 (具有相同父进程的进程) 以及子进程的指针
Times and Timers	核心需要记录进程的创建时间以及在其生命期中消耗的 CPU 时间。它包括进程在系统和用户模式下消耗的时间量
File system	进程可以自由地打开或关闭文件, <code>task_struct</code> 结构中包含一个指向每个打开文件描述符的指针以及指向两个 VFS inode 的指针。这两个指针, 一个指向进程的根目录, 另一个指向其当前目录
Virtual memory	多数进程都有一些虚拟内存, 以便跟踪虚拟内存与系统物理内存的映射关系
Processor Specific Context	进程被挂起时, 记录进程所有的 CPU 状态信息, 当重新调度时, 所有上下文被重新设定
权限信息	<code>task_struct</code> 包含 4 对进程和组标示符用来设置进程的文件和代码的访问权限: <code>uid, gid, effective uid, effective gid, file system uid, file system gid</code> 以及 <code>saved uid and gid</code>

用户可以使用 `ps` 命令来查看 Linux 系统中运行的进程之间的关系。

另外, 系统中所有进程都用一个双向链表连接起来, 而它们的根是 `init` 进程的 `task_struct` 数据结构。这个链表被 Linux 核心用来寻找系统中所有进程, 它是 `ps` 或者 `kill` 命令的基础。

### 10.1.2 Linux 的进程调度

所有进程一部分时间运行于用户模式, 另一部分时间运行于系统模式。如何支持这些

模式，底层硬件的实现各不相同，但是存在一种安全机制可以使它们在用户模式和系统模式之间来回切换。用户模式的权限比系统模式下的小得多。进程通过系统调用切换到系统模式继续执行。此时核心为进程而执行。

进程常因为执行系统调用而需要等待。由于处于等待状态的进程还可能占用 CPU 时间，所以 Linux 采用了预加载调度策略。在此策略中，每个进程只允许运行很短的时间：200 毫秒。当这个时间用完之后，系统将选择另一个进程来运行，原来的进程必须等待一段时间以继续运行。这段时间称为时间片。

调度器必须选择最迫切需要运行而且可以执行的进程来执行。可运行进程是一个只等待 CPU 资源的进程。Linux 使用基于优先级的简单调度算法来选择下一个运行进程。当选定新进程后，系统必须将当前进程的状态，处理器中的寄存器以及上下文状态保存到 `task_struct` 结构中。同时它将重新设置新进程的状态并将系统控制权交给此进程。为了将 CPU 时间合理地分配给系统中每个可执行进程，调度管理器必须将这些时间信息也保存在 `task_struct` 中。

### 10.1.3 虚拟内存

进程的虚拟内存包括可执行代码和数据。首先加载的是程序映象，如 `ls`。 `ls` 和所有可执行映象一样，是由可执行代码和数据组成的。此映象文件包含所有加载可执行代码所需的信息，同时还将程序数据连接进入进程的虚拟内存空间。然后在执行过程中，进程定位可以使用的虚拟内存，以包含正在读取的文件内容。新分配的虚拟内存必须连接到进程已存在的虚拟内存中才能够使用。最后 Linux 进程调用通用库过程，比如文件处理子程序。如果每个进程都有库过程的拷贝，那么共享就变得没有意义，而 Linux 可以使多个进程同时使用共享库。来自共享库的代码和数据必须连接进入进程的虚拟地址空间以及共享此库的其他进程的虚拟地址空间。

任何时候进程都不同时使用包含在其虚拟内存中的所有代码和数据。虽然它可以加载在特定情况下使用的那些代码，如初始化或者处理特殊事件时，另外它也使用了共享库的部分子程序。但如果将这些没有或很少使用的代码和数据全部加载到物理内存中，将会引起极大的浪费和极大降低的系统效率。Linux 使用请求调页技术来把那些进程需要访问的虚拟内存带入物理内存中。核心将进程页表中这些虚拟地址标记成存在但不在内存中的状态，而无需将所有代码和数据直接调入物理内存。当进程试图访问这些代码和数据时，系统硬件将产生页面错误并将控制转移到 Linux 核心来处理。这样对于处理器地址空间中的每个虚拟内存区域，Linux 都必须知道这些虚拟内存从何处而来以及如何将其载入内存以处理页面错误。

### 10.1.4 进程创建

系统启动时总是处于核心模式，此时只有一个进程，即初始化进程（`init`）。像所有进程一样，初始化进程也有一个由堆栈、寄存器等表示的机器状态。当系统中有其他进程被创建并运行时，这些信息将被存储在初始化进程的 `task_struct` 结构中。在系统初始化的最后，初始化进程启动一个核心线程然后保留在 `idle` 状态。如果没有任何事要做，调度管理器将运行 `idle` 进程。

init 是第一个真正的进程，进程号为 1。它负责完成系统的一些初始化设置任务，以及执行系统初始化程序/sbin/init。init 程序使用/etc/inittab 作为脚本文件来创建系统中的新进程。这些新进程又创建各自的新进程。系统中所有进程都是从 init 核心线程中派生出来。

新进程通过克隆老进程或当前进程来创建。系统调用 fork 或 clone 可以创建新任务，复制发生在核心状态下的核心中。在系统调用的结束处有一个新进程等待调度管理器选择它去运行。系统从物理内存中分配出来一个新的 task\_struct 数据结构，同时还有一个或多个包含被复制进程堆栈（用户与核心）的物理页面，然后创建惟一的标记此新任务的进程标志符。新创建的 task\_struct 将被放入 task 数组中，另外将被复制进程的 task\_struct 中的内容页表拷入新的 task\_struct 中。复制完成后，Linux 允许两个进程共享资源而进行复制。这些资源包括文件、信号处理过程和虚拟内存。

### 10.1.5 时钟和定时器

核心跟踪着进程的创建时间以及在其生命期中消耗的 CPU 时间。每个时钟滴答 (tick) 时，核心将更新当前进程在系统模式与用户模式下所消耗的时间（记录在 jiffies 中）。除此之外，Linux 还支持几种进程相关的时间间隔定时器。

进程可以使用这些定时器，在到时向它发送各种信号，这些定时器如下：

- Real。此定时器按照实时时钟记数，当时钟到期时，向进程发送 SIGALRM 信号。
- Virtual。此定时器仅在进程运行时记数，时钟到期时将发送 SIGVTALRM 信号。
- Profile。当到时时，向进程发送 SIGPROF 信号。

以上时间间隔定时器可以同时或单独运行，Linux 将所有这些信息存储在进程的 task\_struct 数据结构中。通过系统调用可以设置这些时间间隔定时器并启动、终止它们或读取它们的当前值。Virtual 和 Profile 定时器以相同方式处理。每次时钟滴答后当前进程的时间间隔定时器将递减，当到时之后将发送适当的信号。

### 10.1.6 程序执行

Linux 程序通过命令解释器来执行。命令解释器 (shell) 是一个用户进程。

在 Linux 中有多个 shell 程序，除了几个内置命令（如 cd 和 pwd）外，每一个命令都是一个可执行二进制文件。当键入一个命令时，Shell 程序将搜索包含在进程 PATH 环境变量中查找路径中的目录来定位这个可执行映象文件。如果找到，则它被加载且执行。shell 使用上面描述的 fork 机制来复制自身，然后用找到的二进制可执行映象的内容来代替其子进程。一般情况下，shell 将等待此命令的完成或者子进程的退出。

可执行文件可以有多种格式，甚至是一个脚本文件。脚本文件需要恰当的命令解释器来处理它们，如/bin/sh 解释 shell 脚本。可执行目标文件包含可执行代码和数据，这样操作系统可以获得足够的信息将其加载到内存并执行。Linux 最常用的目标文件是 ELF，但是理论上 Linux 可以灵活地处理几乎所有目标文件格式。

通过使用文件系统，Linux 所支持的二进制格式既可以构造到核心又可以作为模块加载。核心保存着一个可以支持的二进制格式的链表。

Linux 上使用最多的格式是 a.out 和 ELF。执行文件并不需要全部读入内存，而使用一种请求加载技术。进程使用的可执行映象的每一部分被调入内存，而没用的部分将从内存

中丢弃。

### 10.1.7 ELF 可执行文件格式

ELF (可执行与可连接格式) 是 Unix 系统实验室设计的一种目标文件格式, 现在已成为 Linux 中使用最多的格式。但与其他目标文件格式相比 (如 ECOFF 和 a.out), ELF 的开销稍大, 但是会更加灵活。ELF 可执行文件中包含可执行代码的正文段 (text) 和数据段 (data)。位于可执行映象中的表描述了程序应如何放入进程的虚拟地址空间中。静态连接映象是通过连接器 ld 得到, 在单个映象中包含所有运行此映象所需代码和数据。此映象同时也定义了映象的内存分布和首先被执行的代码的地址。

当 Linux 将一个 ELF 可执行映象加载到进程的虚拟地址空间时, 它并不真正加载映象。首先它建立其虚拟内存数据结构, 包括进程的 vm\_area\_struct 树和页表。当程序执行时将产生页面错, 引起程序代码和数据从物理内存中取出。程序中没有使用到的部分从来都不会加载到内存中去。如果 ELF 二进制格式加载器发现这个映象无效, 它将把进程的当前映象从虚拟内存移出。

### 10.1.8 ELF 共享库

ELF 动态连接映象并不包含全部运行所需要的代码和数据, 其中一部分仅在运行时才连接到共享库中。ELF 共享库列表还在运行时被动态连接器使用。Linux 使用几个动态连接器, 如 ld.so.1, libc.so.1 和 ld-linux.so.1, 这些都放置在 /lib 目录中。这些库中包含常用代码, 如 C 语言子程序等。如果没有动态连接, 所有程序将不得不将所有库过程拷贝一份并连接进来, 这样将需要更多的磁盘与虚拟内存空间。通过动态连接, 每个被引用库过程的信息都可以包含在 ELF 映象列表中。这些信息用来帮助动态连接器定位库, 并将它连入程序的地址空间。

## 10.2 Linux 进程管理命令

操作系统管理多个用户的请求和多个任务。大多数系统都只有一个 CPU 和一个主存, 但一个系统可能有多个二级存储磁盘和多个输入/输出设备。操作系统管理这些资源并在多个用户间共享资源, 当用户提出一个请求时, 造成好像系统被用户独占的假象。实际上操作系统监控着一个等待执行的任务队列, 这些任务包括用户作业、操作系统任务、邮件和打印作业等。操作系统根据每个任务的优先级为每个任务分配合适的时间片, 每个时间片大约都有零点几秒, 虽然看起来很短, 但实际上已经足够计算机完成成千上万的指令集。每个任务都会被系统运行一段时间, 然后挂起, 系统转而处理其他任务、过一段时间以后再回来处理这个任务, 直到某个任务完成, 从任务队列中去除。

10.1 节讨论了 Linux 内核如何管理系统进程, 本节将从用户的角度讲述如何控制进程。

### 10.2.1 进程及作业

Linux 系统上所有运行的东西都可以称为一个进程。每个用户任务、每个系统管理守护进程, 都可以称之为进程。Linux 用分时管理方法使所有的任务共同分享系统资源。用户

所关心的是如何去控制这些进程，让它们能够很好地为用户服务。

进程的一个比较正式的定义是：在自身的虚拟地址空间运行的一个单独的程序。进程与程序是有区别的，进程不是程序，虽然它由程序产生。程序只是一个静态的指令集合，不占系统的运行资源；而进程是一个随时都可能发生变化的、动态的、使用系统运行资源的程序。一个程序可以启动多个进程。

进程和作业的概念也有区别。一个正在执行的进程称为一个作业，而当使用了管道和重定向命令时，作业可以包含多个进程。

例如，需要中止使用 vi 编辑一个文件而做其他工作时，利用作业控制可以让编辑器暂时挂起，返回 shell 提示符开始做其他的工作。当工作完成后，用户可以重新启动挂起的编辑器，返回到刚才挂起的地方，就像用户从来没有离开编辑器一样。

### 10.2.2 手工启动进程

在 shell 中输入需要运行程序的程序名之后，该程序将会执行，事实上启动了一个进程。在 Linux 系统中每个进程都具有一个进程号，用于进程调度。

启动一个进程有两个主要途径：手工启动和调度启动，后者是事先进行设置，根据用户要求自行启动的。

由用户在 shell 提示符后输入命令，手工启动一个进程。手工启动进程又可以分为前台启动和后台启动。

前台启动是手工启动进程最常用的方式。例如，用户键入一个命令 df -h，就启动了一个进程，而且是一个前台进程，这时候系统已经处于一个多进程状态。

**注意：**当执行 df -h 命令以后，紧接着使用 ps -x 查看，却没有看到 df 进程，原因是 df 进程执行太快，执行 ps 时该进程已经执行结束了。

直接从后台手工启动一个进程不太常用，除非是该进程执行时间很长，且用户也不急于查看执行输出。假设用户要启动一个需要长时间运行的格式化文本文件的进程。为了在整个 shell 在格式化过程中，还能使用该 shell，从后台启动这个进程是明智的选择。

**注意：**当执行 df -h 命令以后，紧接着使用 ps -x 查看，却没有看到 df 进程，原因是 df 进程执行太快，执行 ps 时该进程已经执行结束了。

对于手工后台启动一个进程，需要命令之后加一个“&”字符。

例如：手工后台启动 slocate 程序，更新搜索库的命令如下所示。

```
[root@unknown root]# slocate -u &
```

### 10.2.3 调度启动进程

有时候需要对系统进行一些比较费时而且占用资源的维护工作，这些工作适合在深夜进行，这时候用户就可以事先进行调度安排，指定任务运行的时间或者场合，到时候系统会自动完成这些任务。要使用自动启动进程的功能，就需要掌握以下几个启动命令。

#### 1. 定时运行一批程序 —— at 服务

用户使用 at 命令在指定时刻执行指定的命令序列。该命令至少需要指定一个命令和一个执行时间。at 命令可以只指定时间，也可以时间和日期一起指定。

**注意：**指定时间有个日期决定问题。比如说，用户现在指定了一个执行时间为凌晨 4:00，发出 at 命令的时间是头天晚上的 21:00，那么究竟是在哪一天执行该命令呢？如果用户在 4:00 以前仍然在工作，那么该命令将在这个时候完成；如果用户 4:00 以前就退出了工作状态，那么该命令将在第二天凌晨才得到执行。

下面是 at 命令的语法格式：

```
at [-V] [-q 队列] [-f 文件名] [-mldbv] 时间
```

```
at -c 作业 [作业...]
```

at 允许使用一套相当复杂的时间指定方法，它可以接受在当天的 hh:mm（小时:分钟）式的时间指定。如果该时间已经过去，那么就放在第二天执行。当然也可以使用 midnight（深夜），noon（中午），teatime（饮茶时间，一般是下午 4 点）等比较模糊的词语来指定时间。用户还可以采用 12 小时计时制，即在时间后面加上 AM（上午）或者 PM（下午）来说明是上午还是下午。也可以指定命令执行的具体日期，指定格式为 month day（月 日）或者 mm/dd/yy（月/日/年）或者 dd.mm.yy（日.月.年）。指定的日期必须跟在指定时间的后面。

上面介绍的都是绝对计时法，其实还可以使用相对计时法，这对于安排不久就要执行的命令是很有好处的。指定格式为 now+count time-units，now 就是当前时间，time-units 是时间单位，这里可以是 minutes（分钟）、hours（小时）、days（天）、weeks（星期）。count 是时间的数量，究竟是几天，还是几小时等。

还有一种计时方法就是直接使用 today（今天）、tomorrow（明天）来指定完成命令的时间。下面通过一些例子来说明具体用法。

**例：**指定在今天下午 4:30 执行某个命令。假设现在时间是中午 12:30，2004 年 6 月 7 日，其命令格式如下：

- at 4:30pm
- at 16:30
- at 16:30 today
- at now + 4 hours
- at now + 240 minutes
- at 16:30 7.6.04
- at 16:30 6/7/04
- at 16:30 Jun 7

以上这些命令表达的意义是完全一样的，所以在安排时间的时候完全可以根据个人喜好和具体情况自由选择。一般采用绝对时间的 24 小时计时法可以避免由于用户自己的疏忽造成计时错误，例如上例可以写成：

```
at 16:30 6/7/04
```

对于 at 命令来说，需要定时执行的命令是从标准输入或者使用 -f 选项指定的文件中读取并执行的。如果 at 命令是从一个使用 su 命令切换到用户 shell 中执行的，那么当前用户被认为是执行用户，所有的错误和输出结果都会送给这个用户。但是如果有邮件送出的话，收到邮件的将是原来的用户，也就是登录时 shell 的所有者。

**例：**在 6 月 8 日上午 10 点执行 slocate -u 命令。命令如下：



```
[root@unknown root]# at 10:00 6/8/04
warning: commands will be executed using (in order) a) $SHELL b) login shell
c) /bin/sh
at> slocate -u
at>
[1]+  Stopped                  at 10:00 6/8/04
```

在上面的结果中，输入 `at` 命令之后，会出现提示符 `at>`，提示用户输入命令，在此输入了 `slocate -u`，然后按回车键。还可以输入多条命令，当所有要执行的命令输入结束后，按 `Ctrl+d` 键结束 `at` 命令。

`atq` 命令显示将要执行的作业队列，`atrm` 命令可以删除指定的作业。

例：显示上述提交的作业。执行如下：

```
[root@unknown root]# atq
2          2004-06-08 10:00 a root
```

`atq` 的输出结果中，分别显示了每一个作业（Job）的作业号，执行的日期，执行时间和作业类型。上述结果的作业编号为 2。

例：删除上述列出的作业。命令如下：

```
[root@unknown root]# atrm 2
```

验证是否被删除：

```
[root@unknown root]# atq
```

没有显示输出，表明作业已被删除：

```
[root@unknown root]# at -f work 10:00 6/8/04
```

在任何情况下，超级用户都可以使用这个命令。对于其他用户来说，是否可以使用就取决于 `/etc/at.allow` 和 `/etc/at.deny` 文件。

## 2. 周期性运行一批程序 —— cron 服务

前面介绍 `at` 命令都会在一定时间内完成一定任务，但是它只能执行一次。也就是说，当指定了运行命令后，系统在指定时间完成任务，以后就不再执行了。但是在很多情况下需要周期性重复执行一些命令。

例：不同主机的数据会在每天固定的时间里汇总到一台中心计算机上，这时候就需要使用 `cron` 命令来完成任务。

实际上，`cron` 命令不应该手工启动，`cron` 命令在系统启动时就由一个 `shell` 脚本自动启动，进入后台（所以不需要使用 `&` 符号）。一般的用户没有运行该命令的权限，虽然超级用户可以手工启动 `cron`，不过还是建议将其放到 `shell` 脚本中由系统自行启动。

首先 `cron` 命令会搜索 `/var/spool/cron` 目录，寻找以 `/etc/passwd` 文件中的用户名命名的 `crontab` 文件，被找到的这种文件将装入内存。

例：一个用户名为 `globus` 的用户，对应的 `crontab` 文件应该是 `/var/spool/cron/globus`，即以该用户命名的 `crontab` 文件存放在 `/var/spool/cron` 目录下面。

`cron` 命令还将搜索 `/etc/crontab` 文件，这个文件是用不同的格式写成的。`cron` 启动以后，它将首先检查是否有用户设置了 `crontab` 文件，如果没有就转入睡眠状态，释放系统资源。

所以该后台进程占用资源极少，它每分钟被唤醒一次，查看当前是否有需要运行的命令。命令执行结束后，任何输出都将作为邮件发送给 crontab 的所有者，或者是/etc/crontab 文件中 MAILTO 环境变量中指定的用户。这是 cron 的工作原理，但是 cron 命令的执行不需要用户干涉，用户只需要修改 crontab 中要执行的命令。所以下面介绍 crontab 命令。

crontab 命令用于安装、删除或者显示用于驱动 cron 后台进程的表格。用户把需要执行的命令序列放到 crontab 文件中以获得执行，而且每个用户都可以有自己的 crontab 文件。

crontab 命令的常用方法如下：

- crontab -u //设置某个用户的 cron 服务，root 用户在执行 crontab 时需要此参数。
- crontab -l //列出某个用户 cron 服务的详细内容。
- crontab -r //删除某个用户的 cron 服务。
- crontab -e //编辑某个用户的 cron 服务。

例：root 查看自己的 cron 设置。命令如下：

```
crontab -u root -l
```

例：root 想删除 fred 的 cron 设置。命令如下：

```
crontab -u fred -r
```

在 crontab 文件中如何输入需要执行的命令和时间。该文件中每行都包括 6 个域，其中前 5 个域是指定命令被执行的时间，最后一个域是要被执行的命令。每个域之间使用空格或者制表符分隔。格式如下：

```
minute hour day-of-month month-of-year day-of-week commands
```

对于每一项的说明如表 10-2 所示。

表 10-2 crontab 文件中每一行的格式

项编号	项名称	描述
1	minute	分钟（0~59）
2	hour	小时（0~23）
3	day-of-month	一个月（0~23）的第几天（1~31）
4	month-of-year	一年的第几个月（1~12）
5	day-of-week	一周的星期几（0~6），0代表星期天
6	commands	需要执行的命令

这些项都不能为空，必须指定值。除了数字还有几个特殊的符号就是“\*”、“/”和“-”、“,”，\*代表所有的取值范围内的数字，/代表每的意思，“\*/5”表示每 5 个单位，“-”代表从某个数字到某个数字，“,” 分开几个离散的数字。对于要执行的命令，调用的时候需要写出命令的完整路径。

例：每天早上 6 点，在/tmp/test.txt 文件中加入 another day!文本。在 crontab 文件中对应的行如下：

```
0 6 * * * echo " another day!" >> /tmp/test.txt
```

注意：单独使用 echo，从屏幕上看不到任何输出，因为 cron 把任何输出都发送到 root 用户的信箱了。

例：每两个小时，在/tmp/test.txt 文件中加入 two hours!文本。在 crontab 文件中对应的

行如下:

```
0 */2 * * * echo " two hours!" >> /tmp/test.txt
```

例: 晚上 11 点到早上 8 点之间每两个小时, 在/tmp/test.txt 文件中加入 sleepy! 文本。

在 crontab 文件中对应的行如下:

```
0 23-7/2,8 * * * echo " sleepy!" >> /tmp/test.txt
```

每个月的 20 号的早上 10 点, 并且这一天是星期三到星期六之间的一天, 执行 somecom 程序:

```
0 10 20 * 3-6 somecom
```

每次编辑完某个用户的 cron 设置后, cron 自动在/var/spool/cron 下生成一个与此用户同名的文件。此用户的 cron 信息都记录在这个文件中, 这个文件是不可以直接编辑的, 只可以用 crontab -e 来编辑。用户也可以另外建立一个文件, 使用“cron 文件名”命令导入 cron 设置。

假设有个用户名为 globus, 它需要为自己创建一个 crontab 文件。步骤如下:

(1) 首先可以使用任何文本编辑器建立一个新文件, 并将向该文件加入需要运行的命令和要定期执行的时间, 假设该文件为~/globus.cron。

(2) 然后使用 crontab 命令安装这个文件, 使用 crontab 命令使之成为该用户的 crontab 文件。命令如下:

```
crontab globus. ~/globus.cron
```

这样 crontab 文件就建立好了, 可以转到/var/spool/cron 目录下面查看, 发现多了一个 globus 文件。这个文件就是所需的 crontab 文件。

注意: cron 启动后, 每过一分钟读一次 crontab 文件, 检查是否要执行里面的命令。

因此该文件被修改后不需要重新启动 cron 服务。

### 3. 编辑/etc/crontab 文件配置 cron

cron 服务每分钟不仅要读一次/var/spool/cron 内的所有文件, 还需要读一次/etc/crontab, 因此通过配置这个文件也能得到 cron 的服务。用 crontab 配置是针对某个用户的, 而编辑 /etc/crontab 是针对系统的任务。此文件的文件格式如下:

```
SHELL=/bin/bash
```

```
PATH=/sbin:/bin:/usr/sbin:/usr/bin
```

```
MAILTO=root //如果出现错误, 或者有数据输出, 数据作为邮件发给这个账号
```

```
HOME=/
```

```
# run-parts
```

01 \* \* \* \* root run-parts /etc/cron.hourly //每小时执行一/etc/cron.hourly 里的脚本

02 4 \* \* \* root run-parts /etc/cron.daily //每天执行一次/etc/cron.daily 内的脚本

22 4 \* \* 0 root run-parts /etc/cron.weekly //每星期执行一次/etc/cron.weekly 内的脚本

42 4 1 \* \* root run-parts /etc/cron.monthly //每个月执行一次/etc/cron.monthly 内的脚本

注意：如果去掉 `run-parts` 参数，其后面就是运行的某个脚本名，而不是目录名。

#### 10.2.4 进程的挂起及恢复命令——`bg`、`fg`

作业控制允许进程挂起并可以在需要时恢复进程的运行，被挂起的作业恢复后将从中止处开始继续运行。只要在键盘上按 `Ctrl+Z` 键，即可挂起当前的前台作业。在键盘上按 `Ctrl+Z` 键后，将挂起当前执行的命令 `cat`。使用 `jobs` 命令可以显示 `shell` 的作业清单，包括具体的作业、作业号以及作业当前所处的状态。恢复进程执行时，有两种选择：用 `fg` 命令将挂起的作业放回到前台执行；用 `bg` 命令将挂起的作业放到后台执行。灵活使用上述命令，将给自己带来很大的方便。

### 10.3 进程查看

由于 `Linux` 是个多用户系统，有时需要了解其他用户现在在干什么。`Linux` 是一个多任务系统，经常需要对这些进程进行一些调配和管理。要进行管理，首先就要知道现在的进程情况：有哪些进程、进程的状态如何等。

`Linux` 提供了多种命令来了解进程的状况。

#### 10.3.1 `who` 命令

该命令主要用于查看当前系统中的用户情况。如果用户想和其他用户建立即时通讯，比如使用 `talk` 命令，那么首先要确定的就是该用户确实在线上，不然 `talk` 进程就无法建立起来。又如，系统管理员希望监视每个登录的用户此时此刻的所作所为，也要使用 `who` 命令。`who` 命令应用起来非常简单，可以比较准确地掌握用户的情况，所以使用非常广泛。

例：查看系统中的用户及其状态。命令如下：

```
[root@unknown root]# who
root      :0          Jun  3 09:58
a         pts/1      Jun  5 10:47 (192.168.1.46)
a         pts/2      Jun  5 20:55 (192.168.1.46)
a         pts/4      Jun  6 21:48 (192.168.1.46)
globus    pts/5      Jun  7 15:18 (192.168.1.188)
henry     pts/8      Jun  7 15:35 (192.168.1.168)
```

#### 10.3.2 `w` 命令

该命令也用于显示登录到系统的用户情况，但是与 `who` 不同的是，`w` 命令功能更加强大，它不但可以显示有谁登录到系统，还可以显示出这些用户当前正在进行的工作，并且统计数据相对 `who` 命令来说更加详细和科学，可以认为 `w` 命令就是 `who` 命令的增强。`w` 命令的显示项目按以下顺序排列：

- 当前时间
- 系统启动到现在的时间
- 登录用户的数目

- 系统在最近 1 秒、5 秒和 15 秒的平均负载
- 然后是每个用户的各项数据，项目显示顺序如下：
- 登录账号
- 终端名称
- 远程主机名
- 登录时间
- 空闲时间
- JCPU
- PCPU
- 当前正在运行进程的命令行

其中 JCPU 时间指与该终端 (tty) 连接的所有进程占用的时间。这个时间并不包括过去的后台作业时间，但却包括当前正在运行的后台作业所占用的时间。PCPU 时间则是指当前进程（即在 WHAT 项中显示的进程）所占用的时间。

```
[root@unknown root]# w
16:54:01 up 4 days, 6:56, 6 users, load average: 0.00, 0.01, 0.00
USER TTY FROM LOGIN@ IDLE JCPU PCPU WHAT
root :0 - Thu 9am ? 0.00s 0.21s /usr/bin/gnome
a pts/1 192.168.1.46 Sat10am 19:13m 58.53s 0.17s -bash
a pts/2 192.168.1.46 Sat 8pm 35.00s 1.85s 0.03s /usr/sbin/sshd
a pts/4 192.168.1.46 Sun 9pm 32.00s 1.00s 0.02s /usr/sbin/sshd
globus pts/5 192.168.1.188 3:18pm 0.00s 0.08s 0.01s login
henry pts/8 192.168.1.168 3:35pm 1:17m 0.05s 0.00s login
```

### 10.3.3 ps 命令

ps 命令是最基本又非常强大的进程查看命令。使用该命令可以确定有哪些进程正在运行和运行的状态、进程是否结束、进程有没有僵尸、哪些进程占用了过多的资源等等，大部分进程信息都是可以通过执行该命令得到的。

ps 命令最常用的还是用于监控后台进程的工作情况，因为后台进程是不与屏幕、键盘这些标准输入/输出设备进行通信的，所以如果需要检测其状况，就可使用 ps 命令。

ps 命令的常见选项如表 10-3 所示。

表 10-3 ps 命令的常见选项

选项	描述
-e	显示所有进程
-f	全格式
-h	不显示标题
-l	使用长格式
-w	宽行输出
-a	显示终端上的所有进程，包括其他用户的进程
-r	只显示正在运行的进程
-x	显示没有控制终端的进程

例：显示系统中终端上的所有进行进程，包括其他用户的进程。命令如下：

```
[root@unknown root]# ps -a
  PID TTY          TIME CMD
15394 pts/1    00:00:58 emacs
 1288 pts/5    00:00:00 su
 1293 pts/5    00:00:00 bash
 1489 pts/8    00:00:00 su
 1490 pts/8    00:00:00 bash
 2781 pts/2    00:00:02 emacs
 2812 pts/4    00:00:01 emacs
 3861 pts/5    00:00:00 ps
```

例：显示当前用户的进程列表。命令如下：

```
[root@unknown root]# ps
  PID TTY          TIME CMD
 1288 pts/5    00:00:00 su
 1293 pts/5    00:00:00 bash
 4005 pts/5    00:00:00 ps
```

例：使用长列表显示系统中的所有进程。命令如下（不显示执行结果）：

```
[root@unknown root]# ps -ef
```

### 10.3.4 top 命令

top 命令和 ps 命令的基本作用是相同的，显示系统当前的进程和其他状况，但是 top 是一个动态显示过程，即可以通过用户按键来不断刷新进程的当前状态。如果在前台执行该命令，它将独占前台，直到用户终止该程序为止。其实 top 命令提供了实时的对系统处理器的状态监视。它将显示系统中 CPU 的任务列表。该命令可以按 CPU 使用、内存使用和执行时间对任务进行排序，而且该命令的很多特性都可以通过交互式命令或者在定制文件中进行设定。

top 命令输出的实例如图 10-1 所示。

```
12:00:58 up 4 days, 9:03, 3 users, load averages: 0.00 0.00 0.00
163 processes: 66 sleeping, 2 running, 76 zombie, 1 stopped
CPU states: 0.0% user, 0.0% system, 0.0% nice, 0.0% steal, 100.0% idle
Mem: 1020404k av. 566092k used, 454312k free, 0k shrd, 112740k buff
Mem: 333184k actv, 32772k in d, 1908k in c, 213352k cached
Swap: 2048276k av., 4956k used, 2043320k free
```

PID	USER	PRI	NI	SIZE	RSS	SHR	STI	%CPU	%MEM	TIME	CPU	COMMAND
1	root	15	0	100	80	56	S	0.0	0.0	0.04	0.00	0.00
2	root	15	0	0	0	0	SH	0.0	0.0	0.00	0.00	0.00
3	root	15	0	0	0	0	SH	0.0	0.0	0.00	0.00	0.00
4	root	20	19	0	0	0	SH	0.0	0.0	0.00	0.00	0.00
5	root	25	0	0	0	0	SH	0.0	0.0	0.00	0.00	0.00
6	root	15	0	0	0	0	SH	0.0	0.0	0.00	0.00	0.00
7	root	15	0	0	0	0	SH	0.0	0.0	0.00	0.00	0.00
8	root	15	0	0	0	0	SH	0.0	0.0	0.00	0.00	0.00
10	root	15	0	0	0	0	SH	0.0	0.0	0.00	0.00	0.00
11	root	25	0	0	0	0	SH	0.0	0.0	0.00	0.00	0.00
13	root	25	0	0	0	0	SH	0.0	0.0	0.00	0.00	0.00
15	root	15	0	0	0	0	SH	0.0	0.0	0.00	0.00	0.00
21	root	25	0	0	0	0	SH	0.0	0.0	0.00	0.00	0.00
3671	root	15	0	352	144	268	S	0.0	0.0	0.00	0.00	0.00
3675	root	15	0	164	156	112	S	0.0	0.0	0.00	0.00	0.00
3693	root	15	0	72	0	0	SH	0.0	0.0	0.00	0.00	0.00

图 10-1 top 的运行界面

### 10.3.5 进程调度

当需要中断一个前台进程的时候，通常是使用 Ctrl+c 组合键，而对于后台进程不能用组合键来终止，这时就可以使用 kill 命令。该命令可以终止前台和后台进程。终止后台进程的原因包括：该进程占用 CPU 的时间过多、该进程已经死锁等。

kill 命令是通过向进程发送指定的信号来结束进程的。如果没有指定发送的信号，那么默认值为 TERM 信号。TERM 信号将终止所有不能捕获该信号的进程。至于那些可以捕获该信号的进程可能就需要使用 KILL 信号（它的编号为 9），而该信号不能被捕捉。

kill 命令的语法格式有以下两种方式：

kill [-s 信号 [-p]] [-a] 进程号 ...

kill -l [信号]

其中进程号可以通过 ps 命令的输出得到。-s 选项是给程序发送指定的信号，详细的信号可以用 kill -l 命令查看；-p 选项只显示指定进程的 ID 号，而发送信号。

例：杀死 pid 为的 1490 进程（KILL）。命令如下：

```
$ kill -9 1490
```

例：重新执行（HUP）进程号为 4119 的进程。命令如下：

```
$ kill -HUP 4119
```

例：显示所有的信号及其编号对应关系，如图 10-2 所示。

```
[root@unknown root]# kill -l
1) SIGHUP      2) SIGINT      3) SIGQUIT     4) SIGILL
5) SIGTRAP     6) SIGABRT    7) SIGBUS      8) SIGFPE
9) SIGKILL     10) SIGUSR1   11) SIGSEGV     12) SIGUSR2
13) SIGPIPE    14) SIGALRM   15) SIGTERM     17) SIGCHLD
18) SIGCONT    19) SIGSTOP   20) SIGTSTP     21) SIGTTIN
22) SIGTTOU    23) SIGURG    24) SIGXCPU     25) SIGXFSZ
26) SIGVTALRM  27) SIGPROF   28) SIGWINCH    29) SIGIO
30) SIGPWR     31) SIGSYS    33) SIGRTMIN    34) SIGRTMIN+1
35) SIGRTMIN+2 36) SIGRTMIN+3 37) SIGRTMIN+4 38) SIGRTMIN+5
39) SIGRTMIN+6 40) SIGRTMIN+7 41) SIGRTMIN+8 42) SIGRTMIN+9
43) SIGRTMIN+10 44) SIGRTMIN+11 45) SIGRTMIN+12 46) SIGRTMIN+13
47) SIGRTMIN+14 48) SIGRTMIN+15 49) SIGRTMAX-14 50) SIGRTMAX-13
51) SIGRTMAX-12 52) SIGRTMAX-11 53) SIGRTMAX-10 54) SIGRTMAX-9
55) SIGRTMAX-8 56) SIGRTMAX-7 57) SIGRTMAX-6 58) SIGRTMAX-5
59) SIGRTMAX-4 60) SIGRTMAX-3 61) SIGRTMAX-2 62) SIGRTMAX-1
63) SIGRTMAX
[root@unknown root]#
```

图 10-2 使用 kill 命令显示所有的信号

## 思考与练习

- (1) 什么是进程？
- (2) 如何查看和终止系统中的进程？
- (3) 进程有哪些状态？如何改变进程的状态？

## 第 11 章 内核管理和模块管理

内核是一个操作系统的核心。它负责管理系统的进程、内存、设备驱动程序、文件和网络系统，决定着系统的性能和稳定性。

linux 的一个重要的特点就是其源代码的公开性，所有的内核源程序都可以免费得到，大部分应用软件也都是遵循 GPL 而设计的，而且可以获得相应的源程序代码。全世界任何一个软件工程师都可以将自己认为优秀的代码加入到其中，由此引发的一个明显的好处就是 Linux 修补漏洞的快速性以及利用最新软件技术的利用。

本章将首先介绍 Linux 的内核结构，然后用实例讲述了 Linux 内核的升级过程，最后讲述了 Linux 的内核模块的管理。

### 11.1 Linux 内核和模块简介

Linux 内核是 Linux 操作系统的核心，内核需要具有小巧和高效的特性，本节简要介绍了 Linux 内核的结构。Linux 内核支持可插入模块，本节也将介绍 Linux 中可插入模块的特点。

#### 11.1.1 Linux 内核的发展

1991 年 11 月 Linux 0.10 版本推出，早期的 Linux 只有很少的一些程序，直到 0.95 版，Linux 开始才逐渐接近一个真正成熟的操作系统。

Linux 开发的版本号中，中间的数字为偶数代表稳定版，中间的数字为奇数代表开发版。因此到目前为止，Linux 经历了以下几个稳定版本，它们的版本号和维护者分别如下：1.0.x; 1.2.x; 2.0.x 由 David Weinehall 负责维护；2.2.x 由 Alan Cox 负责维护；2.4.x 由 Marcelo Tosatti 负责维护。直到 2003 年 12 月 18 日推出 Linux 2.6.0，由 Linus Torvalds 亲自负责维护。

Linux 内核采用了单内核和微内核的混合结构，它不仅有效克服了调试内核困难，使内核的效率保持很高的效率，而且可以使用模块来扩充内核的功能。

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

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

为了正确合理地设置内核编译配置选项，从而只编译系统需要的功能代码，一般主要有下面 4 个考虑：

- 自己定制编译的内核将具有更少的代码，运行速度更快。
- 系统将拥有更多的内存，内核部分将不会被交换到虚拟内存。



- 不需要的功能编译进入内核可能会增加系统的漏洞。
- 将某种功能编译为模块会比编译到内核速度要慢一些。

### 11.1.2 Linux 内核的组成

Linux 内核主要由 5 个子系统组成：进程调度、内存管理、虚拟文件系统、网络接口和进程间通信。

(1) 进程调度 (SCHED)。控制进程对 CPU 的访问。当需要选择下一个进程运行时，由调度程序选择最需要运行的进程。可运行进程实际上是仅等待 CPU 资源的进程，如果某个进程在等待其他资源，则该进程是不可运行进程。Linux 使用了比较简单的基于优先级的进程调度算法选择新的进程。

(2) 内存管理 (MM)。允许多个进程安全的共享主内存区域。Linux 的内存管理支持虚拟内存，即在计算机中运行的程序的代码、数据和堆栈的总量可以超过实际内存的大小，操作系统只是把当前使用的程序块保留在内存中，其余的程序块则保留在磁盘中。必要时，操作系统负责在磁盘和内存间交换程序块。内存管理从逻辑上分为硬件无关部分和硬件有关部分。硬件无关部分提供了进程的映射和逻辑内存的对换；硬件相关的部分为内存管理硬件提供了虚拟接口。

(3) 虚拟文件系统 (VFS, Virtual File System)。隐藏了各种硬件的具体细节，为了给所有的设备提供统一的接口，VFS 提供了多达数十种不同的文件系统。虚拟文件系统可以分为逻辑文件系统和设备驱动程序。逻辑文件系统指 Linux 所支持的文件系统，如 ext2、fat 等，设备驱动程序指为每一种硬件控制器所编写的设备驱动程序模块。

(4) 网络接口 (NET)。提供了对各种网络标准的存取和各种网络硬件的支持。网络接口可分为网络协议和网络驱动程序。网络协议部分负责实现每一种可能的网络传输协议。网络设备驱动程序负责与硬件设备通讯，每一种可能的硬件设备都有相应的设备驱动程序。

(5) 进程间通讯 (IPC)。支持进程间各种通信机制。

处于中心位置的进程调度，所有其他的子系统都依赖它，因为每个子系统都需要挂起或恢复进程。一般情况下，当一个进程等待硬件操作完成时，它被挂起；当操作真正完成时，进程被恢复执行。例如，当一个进程通过网络发送一条消息时，网络接口需要挂起发送进程，直到网络硬件成功地完成消息的发送，当消息被成功的发送以后，网络接口给进程返回一个代码，表示操作的成功或失败。其他子系统以相似的方式依赖于进程调度。

### 11.1.3 Linux 内核子系统之间的关系

Linux 内核各个子系统之间的依赖关系如下：

(1) 进程调度与内存管理之间的关系：这两个子系统互相依赖。在多道程序环境下，程序要运行，则必须为之创建进程，而创建进程的第一件事情，就是将程序和数据装入内存。

(2) 进程间通信与内存管理的关系：进程间通信子系统要依赖内存管理支持共享内存通信机制，这种机制允许两个进程除了拥有自己的私有空间，还可以存取共同的内存区域。

(3) 虚拟文件系统与网络接口之间的关系：虚拟文件系统利用网络接口支持网络文件系统 (NFS)，也利用内存管理支持 RAMDISK 设备。

(4) 内存管理与虚拟文件系统之间的关系：内存管理利用虚拟文件系统支持交换，交换进程 (swpd) 定期由调度程序调度，这也是内存管理依赖于进程调度的惟一原因。当一个进程存取的内存映射被换出时，内存管理向文件系统发出请求，同时挂起当前正在运行的进程。

除了这些依赖关系外，内核中的所有子系统还要依赖于一些共同的资源。这些资源包括所有子系统都用到的过程。例如，分配和释放内存空间的过程，打印警告或错误信息的过程，还有系统的调试例程等等。

#### 11.1.4 确定 Linux 内核的版本

Linux 提供了 `uname` 命令，它可以显示 Linux 的系统信息。`uname` 的语法如下：

`uname [选项]`

`uname` 命令的常用选项如表 11-1 所示。

表 11-1 `uname` 的常用选项

选项	描述
<code>-a, --all</code>	显示所有信息，顺序如下：内核名、主机名、内核发行版本号、发行日期、机器硬件名、处理器类型、硬件平台、操作系统名
<code>-s, --kernel-name</code>	显示内核名
<code>-n, --nodename</code>	显示主机名
<code>-r, --kernel-release</code>	显示内核发行版本号
<code>-m, --machine</code>	显示机器硬件名
<code>-p, --processor</code>	显示处理器类型
<code>-i, --hardware-platform</code>	显示硬件平台
<code>-o, --operating-system</code>	显示操作系统名称
<code>--help</code>	显示帮助信息
<code>--version</code>	显示版本信息

例：显示所有的系统信息。命令如下：

```
[root@unknown root]# uname -a
```

```
Linux unknown 2.4.20-8 #1 Thu Mar 13 17:54:28 EST 2003 i686 i686 i386 GNU/Linux
```

显示结果表明：内核名为 Linux、主机名为 unknown、内核发行版本号为 2.4.20-8、发行日期为 Thu Mar 13 17:54:28 EST 2003、机器硬件名为 i686、处理器类型为 i686、硬件平台为 i386、操作系统名为 GNU/Linux。

例：显示内核的版本号。命令如下：

```
[root@unknown root]# uname -r
```

```
2.4.20-8
```

结果表明该系统内核的版本号为 2.4.20-8。

注意：Linux 内核的版本与 Linux 的发行版本号不是一回事。例如 Red Hat Linux 9.0 的发行版本号为 9.0，而它的内核版本号为 2.4.20-8。

## 11.2 升级 Red Hat Linux 定制的内核

为了确保内核的完整性和对它所支持的硬件的兼容性，Red Hat Linux 内核在发行之前一定要通过一系列严格的质量保证测试。

### 11.2.1 Red Hat Linux 的定制内核的特点

Red Hat Linux 内核使用 RPM 格式打包，因而它们易于升级和校验。由 Red Hat 公司发行的 kernel RPM 软件包被安装后，initrd 映像会被创建，这样在用户安装了不同的内核后，就没必要使用 mkinitrd 命令。如果用户安装了 GRUB 或 LILO 引导加载程序，它还会修改引导装载程序的配置文件来包括这个新内核。

在此仅讨论在 x86 系统上升级内核的必要步骤。

Red Hat Linux 发行版随带定制的 2.4 版内核，它包括以下功能：

- 内核源码的目录是 /usr/src/linux-2.4/，而不是 /usr/src/linux/。
- 对 ext3 文件系统的支持。
- 对多重处理器（SMP）的支持。
- 对 USB 的支持。
- 对 IEEE 1394 设备（又称 FireWire™）的初步支持。

### 11.2.2 准备升级 Red Hat Linux 的定制内核

在升级内核之前，用户应该先采取几项预防措施。第一步是确保有一张适用于系统的可运行的引导盘，以防万一出现问题。如果引导装载程序没有被正确配置，那么除非有引导盘，否则就无法引导系统。

要创建引导盘，在 shell 提示下登录为根用户，然后键入以下命令：

```
# /sbin/mkbootdisk `uname -r`
```

在继续前，使用引导盘来重新引导你的机器以校验该软盘的可引导性。

最好避免使用引导盘，但是用户应该把它存放在一个安全的地方，以防万一。要判定系统已安装了哪些内核软件包，在 shell 提示下执行下面的命令：

```
$ rpm -qa | grep kernel
```

依据用户选择的安装类型（对应的版本号和软件包可能不同），该命令的输出会包括某些或全部在下面列出的软件包：

- kernel-2.4.20-2.47.1
- kernel-debug-2.4.20-2.47.1
- kernel-source-2.4.20-2.47.1
- kernel-doc-2.4.20-2.47.1
- kernel-pcmcia-cs-3.1.31-13
- kernel-smp-2.4.20-2.47.1

从输出中，用户可以判定需要下载哪些软件包来执行内核升级。对于单处理器系统而言，只有 kernel 软件包是必需的。

如果计算机不只有一个处理器，则需要包括支持多处理器的 `kernel-smp` 软件包。强烈建议用户仍然安装 `kernel` 软件包，以防多处理器内核不能在用户的系统中正确运行。

如果计算机的内存超过了 4GB，必须安装 `kernel-bigmem` 软件包才能使系统使用多于 4GB 的内存。建议也安装 `kernel` 软件包，以便于调试。`kernel-bigmem` 软件包仅支持 i686 体系结构。

如果用户需要 PCMCIA 支持（如在便携电脑上），`kernel-pcmcia-cs` 软件包就必不可少。

除非用户想重新编译内核，或把系统用于内核开发，否则不需要 `kernel-source` 软件包。

`kernel-doc` 软件包包括内核开发文档，它不是必需的。如果用户的系统被用于内核开发，则推荐安装它。

`kernel-util` 软件包包括能够用来控制内核或系统硬件的工具程序，它也不是必需的。

Red Hat 公司建构的内核为不同的 x86 版本做了优化。选项有：用于 AMD Athlon™ 和 AMD Duron™ 系统的 `athlon`；用于 Intel® Pentium® II、Intel® Pentium® III、和 Intel® Pentium® 4 系统的 i686；用于 Intel® Pentium® 和 AMD K6™ 系统的 i586。如果用户不知道 x86 系统具体的版本，使用 i386 版本的内核，它是为所有基于 x86 的系统建构的。

RPM 软件包的 x86 版本被包括在文件名中。例如：`kernel-2.4.20-2.47.1.athlon.rpm` 是为 AMD Athlon™ 和 AMD Duron™ 系统优化的，`kernel-2.4.20-2.47.1.i686.rpm` 是为 Intel® Pentium® II、Intel® Pentium® III、和 Intel® Pentium® 4 系统优化的。当判定了软件包之后，用户需要升级内核，为 `kernel`、`kernel-smp` 和 `kernel-bigmem` 软件包选择正确的体系。其他软件包使用 i386 版本。

### 11.2.3 下载升级的内核

要判定是否有可用于你的系统的升级内核，方法有好几种。

进入 <http://www.redhat.com/apps/support/errata> 网站，选择你使用的 Red Hat Linux 版本，然后查看它的勘误。内核勘误通常在安全顾问（Security Advisories）部分下。从勘误列表中，单击内核勘误来查看它的详细勘误报告。在勘误报告中，有一个需要的 RPM 列表，以及从 Red Hat FTP 站点下载它们的链接。你还可以从 Red Hat FTP 的镜像站点中下载它们。镜像站点的列表在 <http://www.redhat.com/download/mirror.html> 中。

使用 Red Hat 网络来下载内核 RPM 软件包并安装它们。Red Hat 网络能够下载最新的内核和升级系统上的内核，如果必要，还可以创建初始 RAM 映像并配置引导装载程序来载入新内核。要获取更多信息，请参阅 <http://www.redhat.com/docs/manuals/RHNetwork> 上的 Red Hat 网络 User Reference Guide。

如果用户从 Red Hat Linux 的勘误网页上下载了 RPM 软件包，或只使用了 Red Hat 网络来下载软件包，则可进行升级过程。只不过，不要把内核改成默认引导项，因为 Red Hat 网络会自动把默认内核改成最新版本。

### 11.2.4 执行升级

检索到所有必要的软件包后，用户就可以进行内核升级。在 shell 提示符下以根用户身份，转换到包含内核 RPM 软件包的目录中执行升级，过程如下：

注意：用户最好保留旧内核，以防新内核出问题。

使用 `rpm` 命令的 `-i` 选项来保留旧内核。如果你使用了 `-U` 选项来升级 `kernel` 软件包，它会覆盖当前安装了的内核。该命令为：

```
# rpm -ivh kernel-2.4.20-2.47.1.i386.rpm
```

如果系统是多处理器系统，还需安装 `kernel-smp` 软件包（内核版本和 `x86` 版本会有所不同）。命令如下：

```
# rpm -ivh kernel-smp-2.4.20-2.47.1.i386.rpm
```

如果系统是基于 `i686` 的，并包含超过 `4GB` 的内存，还需安装为 `i686` 体系建构的 `kernel-bigmem` 软件包。命令如下：

```
# rpm -ivh kernel-bigmem-2.4.20-2.47.1.i686.rpm
```

如果用户打算升级 `kernel-source`、`kernel-docs`、或 `kernel-utils` 软件包，用户可能不需要保留老版本。使用下面的命令来升级这些软件包：

```
# rpm -Uvh kernel-source-2.4.20-2.47.1.i386.rpm
```

```
# rpm -Uvh kernel-docs-2.4.20-2.47.1.i386.rpm
```

```
# rpm -Uvh kernel-utils-2.4.20-2.47.1.i386.rpm
```

如果系统需要 `PCMCIA` 支持（例如便携电脑），用户还需要安装 `kernel-pcmcia-cs` 并保留老版本。如果你使用了 `-i` 选项，它可能会返回冲突，因为老内核需要该软件包来引导带有 `PCMCIA` 支持的系统。要绕过这个问题，使用 `--force` 选项，命令如下：

```
# rpm -ivh --force kernel-pcmcia-cs-3.1.24-2.i386.rpm
```

### 11.2.5 校验初始 RAM 磁盘映像

如果系统使用 `ext3` 文件系统或 `SCSI` 控制器，用户就需要初始 RAM 磁盘。初始 RAM 磁盘的目的是：允许模块化的内核在它能够进入模块通常驻留的设备之前具备进入内核需要从该设备引导的模块的能力。

初始 RAM 磁盘通过使用 `mkinitrd` 命令来创建。然而，如果内核及其相关文件是从 Red Hat 发行的 RPM 软件包中安装或升级，这个步骤会被自动执行，因此不必被手工进行。要校验它是否被创建了，使用 `ls -l /boot` 命令来确定 `initrd-2.4.20-2.47.1.img` 文件被创建了，即版本是匹配刚刚安装的内核版本。

## 11.3 从源代码升级 Linux 内核

尽管使用 Red Hat 公司定制的内核非常容易和方便。但是它会受到 Red Hat 公司定制的版本限制，如果用户希望使用最新版的 Linux 内核，而且需要内核的其他附加特性，那么用户就不能使用这些定制的内核。用户只能通过编译内核源码来实现升级。

新用户通过编译 Linux 内核源码升级内核是一件具有挑战性的工作，然而它会更灵活，更有效。通过本节的学习，你会掌握内核的升级的一般方法，而且能够将内核升级到最新版本。

### 11.3.1 内核编译模式

要增加对某部分（如网络）功能的支持，可以把相应部分编译到内核中（build-in），也可以把该部分编译成模块（module），使之动态调用。如果编译到内核中，在内核启动时就可以自动支持相应的功能。其优点是方便、速度快，只要机器一启动，就可以使用其功能。缺点是会使内核变得庞大，不管用户是否需要这部分功能，它都会存在，建议经常使用的部分（如网卡）直接编译到内核。如果编译成模块，就会生成对应的.o 文件，在使用的时候可以动态加载，优点是会使内核过分庞大，缺点是速度可能会降低。

### 11.3.2 新版本内核的获取

要使用新的内核首先当然要取得新内核的源代码，内核的官方网站是 <http://www.kernel.org>，在上面可以得到最新版本的内核。内核的更新是相当快的，用户可以在命令行下使用 `finger @kernel.org` 命令来快速得到当前最新的内核列表。

在此将以最新版 2.6.6 版的内核为准，主机名为 unknown，原来的内核是 Red Hat Linux 所带的默认内核。

所有 Linux 内核的官方版本可以在 <http://www.kernel.org> 找到。2.6.6 版内核的具体下载地址是：<http://www.kernel.org/pub/linux/kernel/v2.6/linux-2.6.6.tar.bz2>。

编译内核需要 root 权限，以下操作都假定当前用户身份是 root 用户。下载了内核源码之后，需要把升级的内核源代码拷贝到 /usr/src/ 下，命令为：

```
#cp linux-2.4.18.tar.gz /usr/src
```

可以使用 `tar jxvf linux-2.6.6.tar.bz2` 命令来提取源码，如果用户下载的是以 tgz 结尾的压缩包就使用 `zxvf` 参数来解压。建议用户不要使用 /usr/src/linux 这个目录来存放源码。因为使用了这个目录，需要手工在 /usr/include 目录下创建一些符号链接，建议为源码单独建立一个目录，笔者使用的目录是 /usr/src/linux-2.6.6。解压缩源代码的步骤如下：

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

```
[root@unknown src]# tar jxvf linux-2.6.6.tar.bz2
```

使用 tar 命令后，它将自动生成了 /usr/src/linux-2.6.6 目录，并将所有的源代码解压缩至该目录中。

### 11.3.3 升级 module-init-tools 软件包

要正常编译和运行新的内核，需要升级一些软件包，这些信息在源码目录下的 Documentation/Changes 文件中，请根据实际情况选择升级。

**注意：** 2.4.x 版本的 modutils 工具包已经不再适合新的 2.6.x 版本内核，必须将其升级到 module-init-tools 工具包，下载 module-init-tools-3.0.tar.gz 源码包的位置为

```
http://www.kernel.org/pub/linux/kernel/people/rusty/modules/module-init-tools-3.0.tar.gz.
```

按照下列步骤安装 module-init-tools 工具包：

（1）解压缩 module-init-tools 工具包源代码。

```
# tar -zxvf module-init-tools-3.0.tar.gz
(2) 在 module-init-tools-3.0 目录下, 进行编译和安装。
# ./configure --prefix=/
# make moveold
# make all install
# ./generate-modprobe.conf /etc/modprobe.conf
```

命令 `make moveold` 的作用是把系统原来的 `modutils` 工具程序改名为 `*.old` (例如将 `lsmod` 改为 `lsmod.old`)。

**注意:** `make moveold` 是非常重要的一步, 千万不要省略。这将使原有的 `linux-2.4.x` 系统可以继续使用, 因为在 `2.4.x` 系统下, 新的 `module-init-tools` 工具包实际上还依赖原来 `*.old` 程序来加载内核模块。如果忘记了这一步, 可以先下载并安装原来的 `modutils` 程序包, 然后按照上述的步骤重新安装 `module-init-tools-3.0` 来解决。

新的 `module-init-tools` 工具包不再使用原来的 `/etc/modules.conf` 配置文件, 而是使用新的配置文件 `/etc/modprobe.conf`。因此必须用命令 `./generate-modprobe.conf /etc/modprobe.conf` 来生成新的配置文件 `/etc/modprobe.conf`。

如果用户使用 `devfs` 系统, 还需要将 `modprobe.devfs` 文件复制到 `/etc` 目录下。

当升级了相关的软件包之后, 准备工作就完成了, 下面将进入到编译阶段。

#### 11.3.4 配置、编译和安装 `linux-2.6.x` 内核

`linux-2.6.x` 内核的编译和创建与 `linux-2.4.x` 有很大的差别, 实际上它会更加简单与方便。

(1) 转到 `linux-2.6.6` 的源代码目录:

```
[root@unknown linux-2.6.6]# cd /usr/src/linux-2.6.6
```

(2) 删除原来编译产生的垃圾, 命令如下:

```
[root@unknown linux-2.6.6]# make mrproper
```

**注意:** 如果是新内核没有编译过, 就不需要用 `make mrproper` 命令。

(3) `make menuconfig`, 进入内核选项设置, 命令如下:

```
[root@unknown linux-2.6.6]# make menuconfig
```

在界面的上部, 是配置工作的使用说明, 使用箭头键在各选项间移动, 按回车键进入下一层菜单, 每个选项上的高亮字母是键盘快捷方式, 使用它可以快速的到达你想设置的菜单项上。在括号中:

- 按 `y` 键将这个项目编译进内核。
- 按 `m` 键编译为模块。
- 按 `n` 键取消选择。
- 按 `?` 键将打印这个选项的帮助信息。
- 按 `Esc` 键将返回到上层菜单。

笔者主要使用 `menuconfig` 来设置各个选项, 选项的次序以它为准, 其他工具中各选项也基本与之对应。切记把声卡的模块编译进内核, 否则将无法发声。

在编译内核的过程中，最烦杂的事情就是这步配置工作了，很多新手都不清楚到底该如何选取这些选项。实际上在配置时，大部分选项可以使用其默认值，只有小部分需要根据用户不同的需要选择。选择的的原则是：

- 将与内核其他部分关系较远且不经常使用的部分功能代码编译成为可加载模块，有利于减小内核的长度，减小内核消耗的内存，简化该功能相应的环境改变时对内核的影响。
- 不需要的功能就不要选。
- 与内核关心紧密而且经常使用的部分功能代码直接编译到内核中。

**注意：**新内核中有两个 GUI 配置工具，过去的 `make xconfig` 现在是一个使用 QT 库作为界面方案的配置工具了，所以用户要使用它必需先安装 QT 库。一般的发行版光盘中都包含了这个软件包。

另一个是 `make gconfig`。这是一个使用 GTK 库作为界面方案的工具，同样用户要使用它，就要先安装 GTK 库。这两个工具与原来的 GUI 工具在界面上有很大的变化，更加符合一般图形界面程序的使用习惯。

Red Hat Linux 9.0 使用的是 ext3 文件系统，所以要把 ext3 文件系统编译进内核。如果系统是 SCSI 硬盘，并且根文件在 SCSI 硬盘上，配置内核时在设备驱动项（第八项）中的 SCSI device support 子项中，要把 SCSI 磁盘支持编译进内核。总之，升级后的系统需要什么，就要编译相应的内容。

在 2.6.6 中还添加了几个以调试为目的的 `make` 选项，它们是 `allyesconfig`、`allnoconfig` 和 `allmodconfig`，分别表示将所有选项加载为 y、将所有选项加载为 n、将所有选项加载为 m。它们是为内核开发人员与测试人员准备的，与普通用户关系不大。

有关 `make menuconfig` 选项的详细信息，请见 11.3.6 小节的内容。

（4）当内核配置完成后，需要编译内核本身，命令如下：

```
[root@unknown linux-2.6.6]# make
```

（5）编译选择的模块。

```
[root@unknown linux-2.6.6]# make modules
```

（6）将编译后的模块转移到系统标准位置。

```
[root@unknown linux-2.6.6] # make modules_install
```

（7）安装内核。

```
[root@unknown linux-2.6.6] # make install
```

该命令是从 2.5.69 内核（按照内核发布时间算）开始才有的选项，它可以帮用户完成很多工作：

把压缩内核映像拷贝到 `/boot` 目录下，并创建相应的 `System.map` 符号链接；

修改 `bootloader` 的配置文件；

调用 `mkinitrd` 程序创建内核的 `initrd` 映像。

**注意：**一定要先编译并安装模块，再运行 `make install`；否则可能会出现错误，导致内核升级失败。

修改 `/boot/grub/grub.conf` 配置文件。新内核并不识别 `root=LABEL=` 这个启动参数，如果现在马上重新启动新内核的话，将发生 `kernel panic` 错误。因此，必须把这个 `LABEL=`



替换成根分区的分区名。

如何确定根目录“/”在哪个分区，命令为：

```
[root@unknown linux-2.6.6]# df
```

Filesystem	1K-blocks	Used	Available	Use%	Mounted on
/dev/hda1	20161172	8009776	11127256	42%	/
none	515200	0	515200	0%	/dev/shm

结果表明根分区为/dev/hda1，需要将启动参数改为“root=/dev/hda1”。

**make install** 命令也会修改引导装载配置文件，对于 GRUB 而言，将在/etc/grub.conf 配置文件增加如下类似的配置行：

```
title Red Hat Linux (2.6.6)
root(hd0, 0)
kernel /boot/vmlinuz-2.6.6 ro root=LABEL=/
initrd /boot/initrd-2.6.6.img
```

### 11.3.5 引导测试

最后，使用 **init 6** 命令重新引导系统，在引导菜中就会多出一个 2.6 内核的选项，使用该选项就可以用 Linux 2.6 的内核引导系统。

### 11.3.6 make menuconfig 选项说明

**make menuconfig** 出现的主界面中，共有 14 个配置项。

#### 1. Code maturity level options (代码成熟度选项)

按回车键 (Enter) 进入后的选项如下：

- **[\*] Prompt for development and/or incomplete code/drivers**  
默认情况下是选择的，这将会在设置界面中显示还在开发或者还没有完成的代码与驱动程序。用户应该选择它，因为有许多设备可能必需选择这个选项才能进行配置，实际上它是安全的。
- **[\*] Select only drivers expected to compile cleanly**  
选择这个选项，将不会看到一些已知的、存在问题的驱动程序选项，默认的情况下也是选择的。如果用户有设备没有找到驱动选项，可以将这一项去掉，或许就可以找到相关驱动了，不过它可能是有漏洞 (BUG) 的。
- **[\*] Select only drivers that don't need compile-time external firmware**  
如果用户不需要在内核对一些外部设备的固件作映射支持，就选择它，这也是默认选项。

#### 2. 第二项 General setup

其中的选项如下：

- **[\*] Support for paging of anonymous memory**  
这个选项将使你的内核支持虚拟内存，也就是让用户的计算机好像拥有比实际内

存更多的内存空间，用来执行很大的程序。默认是选择的。

- **[\*] System V IPC**  
为进程提供通信机制，这将使系统中各进程间有交换信息与保持同步的能力。有些程序只有在选 Y 的情况下才能运行，所以不用考虑，这里必须选中。
- **[ ] BSD Process Accounting**  
这里选择 Y，将会让内核为用户层的进程建立一个帐目（进程通过一个特殊的系统调用来通知内核），当程序退出时内核会将该进程的相关信息记录到帐目文件中，主要包括进程的创建时间、创建者、内存占用等信息。如果需要，可以选中。
- **[\*] Sysctl support**  
这将提供一个接口，让用户可以动态地更改一些核心参数与变量，而不需要重新启动系统。打开这个选项将会至少增加 8KB 的内核大小。如果用户的内核仅用于制作安装与恢复系统的系统盘，那么可以不选，以减少对内存的占用。
- **[ ] Auditing support**  
启用审计基础设施，可以用于其他的内核子系统，例如 SELinux。
- **[ ] Kernel log buffer size (16 => 64KB, 17 => 128KB)**  
设置内核日志缓冲区大小，输入十进制数 n，它表示  $2^n$ B。例如 n=16 时，结果为 64KB。默认值为 14，即 16KB。
- **[\*] Support for hot-pluggable devices**  
是否支持热插拔设备，一般需要选中，默认选中。
- **[ ] Kernel .config support**  
这将会把内核的配置信息与相关的文档说明编译进内核中，以后可以使用一些工具来提取它用来重新构建内核，一般不用选它。
- **[ ] Configure standard kernel features (for small systems)**  
这个选项允许禁用或者弱化一些基本的内核选项和设置，以减少内核体积。这是为适合特定环境下的非标准内核选项，它适合嵌入式内核开发，一般不要选用。

### 3. Loadable module support (可加载模块选项)

其选项如下：

- **[\*] Enable loadable module support**  
这个选项可以让内核支持模块。模块是一小段代码，编译后可在系统内核运行时动态的加入内核，从而为内核增加一些特性或是对某种硬件提供支持。一些不常用到的驱动或特性可以编译为模块以减少内核的体积。在运行时可以使用 modprobe 命令来加载它到内核中去（在不需要时还可以移除它）。  
**注意：**是否将一些特性编译为模块的原则：不常使用的不需要的驱动可以编译为模块，那些在系统启动时就要用到的驱动（如文件系统，系统总线的支持）不要编为模块，否则无法启动系统。
- **[ ] Module unloading**  
这个选项可以让用户卸载不再使用的模块，否则将不能卸载任何模块（有些模块一旦加载就不能卸载，不管是否选择了这个选项）。建议选用。

- ☐ **Module versioning support (EXPERIMENTAL)**  
这个选项可以让用户使用其他版本内核中编译的模块，不过并不一定可靠。
- ☒ **Automatic kernel module loading**  
一般情况下，如果内核在某些任务中要使用一些被编译为模块的驱动或特性时，用户要先使用 `modprobe` 命令来加载它，内核才能使用。不过，如果选择了这个选项，在内核需要一些模块时它可以自动调用 `modprobe` 命令来加载需要的模块，这是个很好的特性，默认为选中。

#### 4. Processor type and features (处理器类型和特征)

选项如下：

- **Subarchitecture Type (PC-compatible)**  
这是一个比较新的特性，主要的目的是使 Linux 可以支持多种 PC 标准，一般用户使用的 PC 机遵循所谓 IBM 兼容结构 (pc/at)。这个选项可以让你选择一些其他架构。一般选择 PC-compatible 就可以了。
- **Processor family (Pentium-4/Celeron (P4-based)/Xeon)**  
根据你的 CPU 类型，选择相应的处理器家族即可。
- ☐ **Generic x86 support**  
这个选项提供了对 x86 系列 CPU 最大的兼容性，用来支持一些很少见的 x86 体系的 CPU，它可能会降低系统性能。如果你的 CPU 能够在上面的列表中找到就不要选用这个选项。
- ☐ **HPET Timer Support**  
这也是一个新的特性，HPET 是 intel 制定的新的用于代替传统的 8254 (PIT) 中断定时器与 RTC 的定时器，全称叫作高精度事件定时器。如果用户有一台较新的机器就选它，一般来说，它是一个安全的选项，即使用户的硬件不支持 HPET，也不会造成问题，因为它会自动用 8254 替换。
- ☒ **Symmetric multi-processing support**  
如果系统使用了多处理器系统就选用，否则不要选中。
- ☒ **Preemptible Kernel**  
这是一个新特性，它就是可抢先式内核。也就是说被一些优先级很高的程序可以先与一些低优先级的程序执行，即使这些程序是在核心态下执行（这实际上仍然不是真正的抢先式内核）。从而减少内核潜伏期，提高系统的响应。当然在一些特殊的点，内核是不可抢先的，比如内核中的调度程序自身在执行时就是不可被抢先的。这个特性可以提高桌面系统、实时系统的性能，建议选用。
- ☐ **Local APIC support on uniprocessors**  
本地的 APIC（高级可编程中断控制器，Advanced Programmable Interrupt Controller）支持，它是一个与 CPU 集成的中断控制器。如果你的系统是单 CPU 的，即使你的 CPU 不支持 APIC 也不会造成问题，建议选用它。
- ☒ **Machine Check Exception**  
如果系统出现一些问题（如 CPU 过热），内核将会在屏幕上显示相关信息来提醒

用户。这个功能是需要硬件支持的。你可以查看 `/proc/cpuinfo` 是否有 `mce` 标志，如果有，则选中。如果选了它之后出现问题，可以在启动时加 `nomce` 参数来关闭它。

- **[\*] Check for non-fatal errors on AMD Athlon/Duron / Intel Pentium 4**  
打开这个选项将会检查用户机器上可能存在的问题，如果有一个非致命错误出现，将会自动的修复并且记录，这可以帮助用户查出程序出现问题的原因，是一个不错的选项，当然它只能用在 AMD Athlon/Duron / Intel Pentium 4 这类 CPU 上。
- **<M> Toshiba Laptop support**  
这个选项是针对 Toshiba 笔记本的，可以用来访问 Toshiba 的系统管理模式，也就是可以直接设置 BIOS。不过要注意它只在 Toshiba 自己的 BIOS 中起作用。假如一台 Toshiba 笔记本的 BIOS 是 Phoenix 的，那这个选项仍然无用。
- **<M> Dell laptop support**  
这个同上面的选项基本相同，在此不再重复。
- **<M> /dev/cpu/microcode - Intel IA32 CPU microcode support**  
这个选项将使用户可以更新 Intel IA32 系列处理器的微代码，当然还必须在文件系统选项中选择 `devfs` 才能正常使用。如果用户把它编译为模块，那么还需要在 `modprobe.conf` 中加一行：`alias char-major-10-184 microcode`。  
内核本身并不带有微代码的二进制文件，用户可以到网址上得到新信息。
- **<M> /dev/cpu/\*/msr - Model-specific register support**  
桌面用户一般用不到这个选项，它主要用在 Intel 的嵌入式 CPU 中，这个寄存器的作用也因不同的 CPU 类型而异。
- **<M> /dev/cpu/\*/cpuid - CPU information support**  
这会在 `/dev/cpu` 中建立一系列的设备文件，用于访问指定的 CPU。
- **Firmware Drivers**  
这可以打开实模式下 BIOS 中的增强磁盘设备服务，以决定从哪个磁盘上启动。这一般是安全的，不过大多数 BIOS 提供商都没有实现这个特性。
- **High Memory Support (4GB)**  
如果有大容量的内存（超过 4G，或者超过 64GB），要选用它，以便内核可以使用这部分内存。这个选项一般为选 `off`。否则可选 4GB 或者 64GB。
- **[ ] Allocate 3rd-level pagetables from highmem (NEW)**  
虚拟内存系统（VM）为每一物理内存页分配一个页表项，对于大内存系统而言，将会对前面的底端内存造成极大的浪费，设置这个选项将把用户的页表项放在高端内存部分。
- **[ ] Math emulation**  
如果 CPU 上没有数学协处理器，打开这个选项可以让内核模拟一个，以提供浮点计算能力。如果 CPU 不比 486SX 旧，就根本不需要这个选项。
- **[\*] MTRR (Memory Type Range Register) support**  
在 Intel 家族的处理器（如 Ppro、PII 或更高）中，有一个内存类型范围寄存器，可用来控制处理器访问的内存范围。打开它一般可以提高显卡的显示性能。

- Use register arguments (EXPERIMENTAL)

这个选项将使内核编译时使用 `-mregparm=3` 选项，它使用了不同的参数传递方法。只有 3.0 以及更高的 gcc 版本才支持这个选项，否则会生成不正确的输出。

## 5. Power management options (ACPI, APM) (高级电源管理)

选项如下：

- ☒ Power Management support

如果用户想希望 Linux 支持高级电源管理（也称为软关机，系统休眠等）需要选择它。

- ☐ Software Suspend (EXPERIMENTAL)

选择这个选项，用户可以挂起他的计算机（有点像 XP 中的休眠）。打开这项功能后，用户可以使用 `swsusp` 或者 `shutdown -z` 来挂起计算机。这样系统会将当前正在进行的工作，作一个镜像保存到交换分区中，在用户下一次启动时使用启动参数 `"resume=/dev/对应的交换分区"`，内核就会将上一次的工作内核从镜像文件中恢复到内存，这可以大大提高系统的启动速度。当你不想恢复上次的工作时间向内核传递参数 `noresume`。不过系统启动后交换分区将不可用，可以使用 `mkswap` 命令来重新格式化这个交换分区。这个功能不需要高级电源管理的支持。

- ☐ Suspend-to-Disk Support

这个选项与上面的功能基本相同，不过更灵活一些，可以通过下面的子选项来指定一个专用的交换分区来保存内存镜像。

() Default resume partition: 与上面的选项配合，来指定保存镜像的分区。

- ACPI (Advanced Configuration and Power Interface) Support: 从这里进入 ACPI 电源管理的配置界面，要注意 ACPI 与 APM 不能同时使用。

- APM (Advanced Power Management) BIOS Support: 高级能源管理支持，它提供了若干个选项，供用户选择。

- CPU Frequency scaling

频率调节允许用户动态的改变 CPU 的速度，它是节省笔记本电脑电池电量的一个好方法。

## 6. Bus options (PCI, PCMCIA, EISA, MCA, ISA) (总线选项)

选项如下：

- ☒ PCI support

如果使用了 PCI 的卡，必须选中。

- ☐ Legacy /proc/pci interface

选择这个选项，将启用 `procfs` 文件 `/proc/pci`，它提供了 PCI 设备的一个描述。这个特征已经过时，因此建议不要选用。

- PCI device name database

默认情况下，内核有一个包含了所有已知 PCI 设备名数据库，通过该数据库在 `/proc/pci`、`/proc/ioproports` 和类似的文件中生成用户能够理解的信息。这个数据库使

内核的大小增大了约 80KB。如果不使用这个选项，那么系统将提供 PCI 设备的 ID 号，而不是设备名。

- **[\*] ISA support**

首先看一下你的主板上是否有 ISA 插槽，如果没有就不需要选择该选项。ISA (Industry Standard Architecture) 是一种总线系统，其他的总线类型包括 PCI、EISA、MCA (微通道总线, MicroChannel) 和 VESA。ISA 是旧总线标准，它已经被 PCI 总线所取代。因此通常不需要该选项。

- **[ ] MCA support (微通道总线支持)**

MCA (微通道总线, MicroChannel Architecture) 存在于某些 IBM PS/2 和笔记本电脑上。该总线类似于 PCI 和 ISA，通常不需要 MCA 总线支持。

- **<> NatSemi SCx200 support**

该选项提供了对 National Semiconductor SCx200 处理器的基本支持，它仅是 GPIO 引脚的驱动程序。除非你知道确实需要该项，否则不要选用。

- **PCMCIA/CardBus support**

该选项对 PCMCIA 卡提供支持，只是笔记本电脑 PCMCIA 的设置项，它有 5 个子选项。如果是笔记本电脑，那么都需要选中它的子选项的第一项 (PCMCIA/CardBus support)，然后根据实际情况，选用后面的 4 个子选项。

- **PCI Hotplug Support**

如果系统主板上 PCI 热插拔控制器 (PCI Hotplug controller)，那么应该选用该选项。这使用户能在系统引导和运行状态下装入和卸载 PCI 卡，为了与 PCI 热插拔控制器进行交互，系统必须挂载 pcihpfs 文件系统。

## 7. Executable file formats (可执行文件格式)

选项如下：

- **[\*] Kernel support for ELF binaries**

ELF (可执行和可连接格式, Executable and Linkable Format) 是一种可以支持不同平台和不同操作系统的库和可执行文件格式。除了也使用 Linux 传统的 a.out 格式以外，目前 Linux 主要使用 ELF 格式，原因是：ELF 是可移植的。许多新的可执行文件只使用 ELF 格式发布。这是当前 Linux 普遍支持的可执行文件格式，必须编译到内核中。

- **<M> Kernel support for a.out and ECOFF binaries**

a.out (Assembler.OUTPUT) 是一种早期 UNIX 使用的库和可执行文件格式。在 ELF 替代 a.out 之前，Linux 曾经使用了 a.out 的 QMAGIC 和 ZMAGIC 两种格式。这是旧版的 Linux 的可执行文件和函数库的格式，如果你确认肯定用不到这种格式的可执行文件，那么就可以不把它编译到内核当中。

- **<M> Kernel support for MISC binaries**

如果选用该选项，系统将能够使用 Java、Python、.NET、Emacs-Lisp 和 DOS 仿真程序。默认以模块形式加载，建议选中该选项。

## 8. Device Drivers (设备驱动程序)

这里面的包括了各种外部设备的驱动程序, 用户应该根据系统的实际需要进行选择, 在此就不一一介绍了。

## 9. File systems (文件系统)

Linux 提供了对几乎所有的文件系统都提供了支持。采用在默认选项的基础上进行修改方法, 默认选项包含了常用的文件系统选项, 能支持大多数应用, 如果用户有不常用的文件系统支持, 可以另外选择不常见的文件系统。

如果系统中存在 Windows 的 NTFS 文件系统, 那么需要在 DOS/FAT/NT Filesystems 选项的子选项下再选中<M> NTFS file system support (仅支持文件读操作) 和[\*] NTFS write support (用于支持 NTFS 文件系统的写操作)。

## 10. Profiling support (评测支持)

如果使用 y 选中该选项, 将会启用扩展的评测支持机制, 供诸如 OProfile 的多种评测器 (profiler) 使用。评测是表示不同性能特性和特征的数据的形式化总结或分析, 通常以图形和表的形式出现。Oprofile 是用于 Linux 的若干种评测和性能监控工具中的一种。它可以工作在不同的体系结构上, 包括 IA32, IA64 和 AMD Athlon 系列。

## 11. Kernel hacking (内核调试)

配置了这个选项, 即使在系统崩溃时, 用户也可以进行一定的工作。普通用户用不着这个功能。

## 12. Security options (安全选项)

当选用了这个选项之后, 用户可以选择配置内核中的多种安全模型, 否则 Linux 将使用默认的安全模型。如果没有特殊需求, 则不选用这个选项。

## 13. Cryptographic options (密码选项)

这个选项提供了最基本的密码 API。

## 14. Library routines (库例程)

有如下选项:

- Library routines

如果在内核树里不需要这个 CRC32 (32 位循环冗余检查, Cyclic Redundancy-Check) 函数, 而在内核树以外的模块需要这些函数, 而且如果让这些模块使用 CRC32 库函数, 可以需要使用 M 将其编译为模块。

- CRC32c (Castagnoli, et al) Cyclic Redundancy-Check

如果在内核树里不需要这个选项 CRC32c 函数, 而在内核树以外的模块需要这些函数, 而让这些模块使用 CRC32c 库函数, 可以需要使用 M 将其编译为模块。

上面是 Linux 2.6.6 内核的所有编译选项, 用户需要根据具体情况选用不同的选项, 否则会出现:

- 安装的选项太多。模块太多，导致占用大量的磁盘空间。内核太大，影响系统整体的性能。
- 安装的选项太少。导致一些功能无法实现。

## 11.4 Linux 模块管理

### 11.4.1 模块的概念

模块 (modules) 指的是内核模块 (kernel modules)，通常一个模块提供一个功能，如 isofs、minix、nfs、lp 等。模块化有两种解决方法：设计者可以把各项功能分离到单独的叫作线程的处理中去，或者是将内核以包含/排除一些功能的方式重新编译。如果把功能分离到线程中去，那么内核就叫做“微内核” (micro-kernel)，这种解决方法增加了线程间协调工作的通信开销。就像名字暗示的那样，这种解决方案的优点在于内核的很小。

Linux 的解决方案是包含内核模块，这些模块是可以按需要随时装载和卸载的。这样做可以使得内核的大小和通信量都达到最小，模块与内核独立。这样做有 3 种优点：

(1) 将来修改 kernel 时，不必全部重新编译，可节省不少时间。

(2) 若需要安装新的模块，不必重新编译内核，只要插入 (通过 insmod 指令) 对应的模块即可。

(3) 减少内核对系统资源的占用，内核可以集中精力做最基本的事情，把一些扩展功能都交给模块实现。

模块也可以用来尝试新的内核代码而不需要每次都创建和重新激活内核。但是，这样做带来的问题是：使用内核模块通常会轻微地增加性能和内存开支。一个可加载模块肯定会产生更多的代码，这种代码和额外的数据结构会占用更多的内存。另外，间接访问内核资源也让模块的效率降低。

模块化的思想已经被广泛接受，主要的原因在于它可以扩展系统的功能，用户可以灵活配置系统。Apache 也采取了这种功能扩展方式，本书主要讨论是内核的模块安装与卸载。

### 11.4.2 加载模块

加载内核模块的方法有两种。第一种使用 insmod 命令手工把它插入到内核。另一个更智能的方法是在需要的时候加载这个模块，这叫做按需加载 (demand loading)。当内核发现需要一个模块的时候，例如当用户安装一个不在内核的文件系统的时候，内核会请求内核守护进程 (kerneld) 试图加载合适的模块。内核守护进程 kerneld 非常智能，能够主动把用户需要的模块自动插入内核，将没用到的模块从内核中清除。Kerneld 由两个独立的部分构成：一部分工作于 Linux 的内核，负责向 daemon 发送请求；另一部分工作于系统的用户数据区，负责调入由内核请求指定的模块。如果没有 kerneld，就只能通过用 insmod 或 modprobe 命令手工加载模块。

### 11.4.3 与模块操作有关的命令

与模块有关的命令有



- `lsmod`: 列出已经被内核调入的模块。
- `insmod`: 将某个 `module` 插入到内核中。
- `rmmod`: 将某个 `module` 从内核中卸载。
- `depmod`: 生成依赖文件, 告诉 `insmod` 将来要从哪儿调入模块。这个依赖文件在 `/lib/modules/[当前内核版本]/modules.dep` 中。
- `kerneld`: 负责自动将模块调入内核和把模块从内核中卸载。

#### 11.4.4 手工加载模块

如果要以手工的方式加载模块, 建议使用 `modprobe`, 因为它可以解决模块之间的依赖性问题。以声卡的部分来说, 以 `sound blaster` 为例, 它由以下模块组成:

- `sb 33652 0 (autoclean)`
- `uart401 6160 0 (autoclean) [sb]`
- `sound 56492 0 (autoclean) [sb uart401]`
- `soundcore 2372 5 (autoclean) [sb sound]`

这些模块都要加载上来, 整个声卡才能工作, 而且它们之间是有依赖性关系的。最核心的 `soundcore` 必须首先装入, 最后装入 `sb`。但一般人是不知其先后顺序的。因此, `modprobe` 就是用来解决这个问题的。

通常用户只要执行:

```
# modprobe sb
```

它就会自动找出 `sb` 用到的所有的模块, 并将它们都加载进来。

那么内核是怎么知道这些模块间的依赖性关系的呢? 原来, 在系统启动脚本里有一条 `depmod -a` 命令, 会给系统中的所有可用的模块创建一个依赖关系的列表。而 `modprobe module-name` 会使用这个列表, 在装入指定的模块之前, 先装入那些应该事先装入的模块。如果在这个从属列表中找不到 `module-name`, 它会给出相应的出错信息。

但若使用 `insmod`, 它不会自动完成其他模块的调入。比如说, 我们要加入 `PPP` 模块, 用这个命令:

```
# insmod ppp
```

如果操作成功, 系统出现操作提示符。如果没有成功, 可能出现下列信息:

```
/lib/modules/2.2.10/net/ppp.o: unresolved symbol slhc_init_Rsmp_1ca65fca
/lib/modules/2.2.10/net/ppp.o: unresolved symbol slhc_compress_Rsmp_cfd3a418
/lib/modules/2.2.10/net/ppp.o: unresolved symbol slhc_free_Rsmp_b99033d9
/lib/modules/2.2.10/net/ppp.o: unresolved symbol slhc_toss_Rsmp_a152cec0
/lib/modules/2.2.10/net/ppp.o: unresolved symbol slhc_remember_Rsmp_07972313
/lib/modules/2.2.10/net/ppp.o:unresolved symbol slhc_uncompress_Rsmp_3bb36b01
```

这说明, `PPP` 模块没有加载成功, 错误提示中的 `unresolved symbol` 说明, `PPP` 模块所需要的一些模块还没有载入。错误提示第一行的内容是: `slhc_init_Rsmp_1ca65fca`, 这其中可能需要一些经验来做判断, 它是以 `slhc` 开头的, 就使用如下命令:

```
# insmod slhc
```

一切正常, 然后我们再加加载 `PPP` 模块

```
# insmod ppp
```

这次没有出现错误提示，说明 PPP 模块被成功加载。

#### 11.4.5 从内存中卸载模块

要卸载一个模块，首先用 `lsmod` 看看该模块是否确实已经被加载，然后再卸载。除此之外，在碰到有依赖关系的模块时，从内核中卸载模块的过程与载入的过程恰好相反，它遵循“先进后出”的准则，即在一系列有依赖关系的模块中，必须先卸载最后加载进来的模块，最后卸载最先加载进来的模块。例如，如果要用 `rmmod` 移除正在使用中的模块。如要卸载 `slhc`，PPP 模块仍在在使用它，会出现错误提示：Device or resource busy。所以在将 PPP 模块从内存中卸载后，才可能将 `slhc` 模块从内存中卸载。

```
# rmmod ppp
```

```
# rmmod slhc
```

总之，在卸载模块时，对于可能出现的模块间依赖性问题的，Linux 会给用户提示足够的信息，仔细查看这些信息，就能够采取相应的操作，最终能这解决这些问题。

### 思考与练习

- (1) Linux 内核的功能是什么？
- (2) 升级 Linux 内核有哪两种方法？这两种方法的优点和缺点分别是什么？
- (3) 简述从源代码升级 Linux 内核的步骤。
- (4) 管理 Linux 内核模块的命令有哪些？它们的功能分别是什么？

## 第 12 章 压缩和备份

数据具有很高的价值，数据丢失或者损坏后再重新生成，需要花费宝贵的时间和令人绝望的努力，有时甚至不可能再生。由于数据是一种投资，用户必须保护它，并采取措施避免数据丢失。否则要花费惨重的代价，众多的服务器被摧毁，对于在备份管理上有漏洞的公司，只好自吞苦果。相反，有很好的备份措施，则可以将灾难损失降低到最低程度。

由于经常备份数据有导致备份数据量很大，这样不仅增加了存储的代价，同时在网络传输时也会占用大量的网络带宽。为了降低备份的费用，可以使用 Linux 自身提供的压缩工具或者第三方的压缩工具，减少备份数据的容量。

本章将以实例的形式讲述了 Linux 常用的数据备份命令和压缩命令，而且讲解了如何将备份和压缩命令结合起来使用。

### 12.1 备份的重要性

丢失数据常见的原因有：硬件失败、软件曲线、人为因素或自然灾害。虽然现代硬件已经相当可靠，但仍可能出现自然损坏。起决定性作用的数据存储硬件是硬盘，它依赖微小的磁道和扇区在充满电子干扰信号的世界里保存脆弱的数据。加之现代软件依然不可靠，不够强壮；人更不可靠，他们常常不小心，有时会疲倦，因此很容易犯错误，甚至为有些人（入侵者）的某种目的恶意地破坏数据。不可抗拒的自然作用本身没有邪恶的目的，但也可能造成数据破坏。绝对的数据安全几乎是不可能的，因此及早备份数据是保护数据安全的根本。

备份是保护数据的重要方法。如果有数据的多个拷贝，只要这些数据不会全部毁坏，人们就可以从数据的备份中恢复丢失的数据，以保护数据不丢失。

备份所需的最重要的策略是选择备份介质。一般用软盘和磁带。软盘尽管很便宜、比较可靠、速度不快、容易得到，但不便存储大量的数据。磁带也很便宜，比较可靠，存储速比较快，很容易得到，是一种比较理想的备份方式。在网络快速发展的今天，人们通常将备份的数据打包并压缩后，将拷贝传递到多个地理分散的地方，实现数据备份。

为了节省存储空间，常常将备份文件进行压缩。下面分别介绍用户经常使用备份与压缩的命令。

### 12.2 归档（备份）命令

本节介绍了 Linux 提供的多种备份工具，而且都给出了使用的实例，以便用户根据需要选择使用。

#### 12.2.1 tar 命令 (tape archive)

tar 可以为文件和目录创建档案。利用 tar，用户可以为某一特定文件创建档案（备份文

件),也可以改变档案中文件,或者向档案中加入新的文件。`tar` 最初被用来在磁带上创建档案,现在用户可以在任何设备(如软盘、硬盘和光盘等)上创建档案。利用 `tar` 命令,可以把大量的文件和目录一起打包成一个文件,这对于备份文件或将几个文件组合成为一个文件以便于网络传输是非常有用的。Linux 上的 `tar` 是 GNU 版本。

`tar` 命令语法格式如下:

`tar [主选项+辅选项] 文件或者目录列表`

使用 `tar` 命令时,主选项必须而且一次只能选一个,它告诉 `tar` 要做什么;辅选项是辅助使用的,应根据需要须选择一个或者多个。主选项和辅选项分别如表 12-1 和表 12-2 所示。

表 12-1 `tar` 命令的主选项

选项	描述
<code>-c</code>	创建新的档案文件。用于备份一个或者多个目录或文件
<code>-r</code>	把要存档的文件追加到档案文件的末尾。当用户已经有备份文件,又发现还有目录或文件需要放在已有的备份文件中时,使用该选项
<code>-t</code>	列出档案文件中已经备份的文件列表
<code>-u</code>	更新文件,用新文件替代档案中原始文件,如果备份文件中不存在该文件,则把它追加到备份文件的末尾
<code>-x</code>	从档案文件中还原文件

表 12-2 `tar` 命令的辅选项

选项	描述
<code>-b</code>	该选项是为磁带机设定的。其后跟一数字,用来说明区块的大小。系统预设值为 20 块(20*512 字节)
<code>-f 文件名</code>	归档到普通文件
<code>-k</code>	保存已经存在的文件。例如在还原的过程中,遇到相同的文件,不会进行覆盖
<code>-m</code>	在还原文件时,把所有文件的修改时间设定为现在时间
<code>-M</code>	创建多卷的档案文件,以便在几个磁盘中存放
<code>-v</code>	报告 <code>tar</code> 详细处理过程
<code>-w</code>	每一步都要求确认
<code>-z</code>	用 <code>gzip</code> 来压缩/解压缩文件,加上该选项后可以将档案文件进行压缩,但还原时一定要使用该选项进行解压缩
<code>-j</code>	通过 <code>bzip2</code> 压缩程序,即在归档时使用 <code>bzip2</code> 压缩,在提取文件时使用 <code>bzip2</code> 解压缩
<code>-Z</code>	通过 <code>compress</code> 程序过滤档案,即在归档时使用 <code>compress</code> 压缩,在提取文件时使用 <code>compress</code> 解压缩

注意:在 Linux 系统中,普通用户使用 `tar` 命令时,其最常用的一个用法是使用 `-f` 选项,将多个文件和目录子树打包成一个文件。它本身并不压缩文件,但是它可以使用选项与其他压缩程序联合使用,实现即打包,又压缩的功能。

例:将当前目录下 `newdir3` 目录和 `kernel` 主目录中的所有文件以及 `vitest` 文件归档到文

件 **keydata.tar**，并显示详细的归档过程。命令如下：

```
[globus@unknown globus]$ tar cvf keydata.tar newdir3 kernel vitest
```

显示过程略。选项为 **cvf**，**c** 表示新建备份，**v** 表示显示备份过程，**fkeydata.tar** 表示将备份保存到文件 **keydata.tar** 中，后面的 **newdir3**、**kernel** 和 **vitest** 分别表示要备份目录子树和文件名。结果在当前目录将要创建 **keydata.tar** 备份文件。

显示已创建的归档文件：

```
[globus@unknown globus]$ ls -l keydata.tar
-rw-rw-r-- 1 globus globus 236738560 6月 9 15:41 keydata.tar
```

12.2.2 dump 命令

备份工具 **dump** 可将整个文件系统备份至指定的设备，或备份成一个大文件。

**dump** 命令的语法格式为：

**dump [-cnu][0123456789][-b <区块大小>][-B <区块数目>][-d <密度>][-f <设备名称>][-h <层级>][-s <磁带长度>][-T <日期>][目录或文件系统] 或 **dump [-wW]****

例：以管理员身份备份/boot 文件系统，保存到 **boot.dump** 文件。命令如下：

```
[root@unknown root]# dump -f boot.dump /boot
```

在 **dump** 执行过程中，会显示 **dump** 过程。当备份完成时，显示 **DUMP: DUMP IS DONE** 的信息，并在当前目录产生了 **boot.dump** 备份文件。

12.2.3 ar 命令

用户可以使用 **ar** (**archive**) 命令集合许多文件，成为单一的档案文件。在档案文件中，所有成员文件皆保存有原来的属性与权限。**ar** 命令可以建立、修改档案文件或者从档案文件中抽取文件。

**ar** 语法格式如下：

**ar [-dmpqrx] [cfosSuvV] [a<成员文件>] [b<成员文件>] [i<成员文件>] [档案文件] [成员文件]**

**ar** 命令包括指令参数和选项参数，分别如表 12-3 和表 12-4 所示。

表 12-3 ar 命令的指令参数

选项	描述
-d	删除档案文件中的成员文件
-m	变更成员文件在档案文件中的次序
-p	显示档案文件中的成员文件内容
-q	将文件附加在档案文件末端
-r	将文件插入档案文件中
-t	显示档案文件中所包含的文件
-x	从档案文件中取出成员文件

表 12-4 ar 命令的选项参数

选项	描述
a<成员文件>	将文件插入档案文件中指定的成员文件之后
b<成员文件>	将文件插入档案文件中指定的成员文件之前
c	建立档案文件
f	为避免过长的文件名不兼容于其他系统的ar指令指令, 利用此参数, 截掉要放入档案文件中过长的成员文件名
i<成员文件>	将文件插入档案文件中指定的成员文件之前
o	保留档案文件中文件的日期
s	若档案文件中包含了对象模式, 利用此参数可建立档案文件的符号表
S	不产生符号表
u	只将日期较新文件插入档案文件中
v	显示执行时详细的信息
V	显示版本信息

例: 将目标文件 bar.o 加入到档案文件 foo.a 中。命令如下:

```
[globus@unknown globust]# ar r foo.a bar.o
```

#### 12.2.4 cpio(copy in/out) 命令

cpio 命令是一个能产生多卷备份的用户级命令, 它支持 cpio 或 tar 两种文件格式, 它具备一些 tar 命令没有的功能。只有很少的 Linux 软件包是以 cpio 格式发行的, 因此 cpio 的使用并不普遍。

cpio 的语法格式为:

```
cpio [-0aABckLovV] [-C <输入/输出大小>] [-F <备份档>]
    [-H <备份格式>] [-O <备份档>] [--block-size=<块大小>]
    [--force-local] [--help] [--quiet] [--version]
```

或

```
cpio [-bBcdfikmnrsStuvV] [-C <输入/输出大小>] [-E <范本文件>] [-F <备份档>] [-H <
备份格式>] [-I <备份档>] [-M <回传信息>] [-R <拥有者><:/./><所属群组>] [--block-size=<
区 块 大 小 >] [--force-local] [--help] [--no-absolute-filenames] 面 通 知
[--no-preserve-owner] [--only-verify-crc] [--quiet] [--sparse] [--version] [范本样
式 ...] 或 cpio [-0adkiLmpuvV] [-R <拥 有 者 ><:/./>< 所 属 群 组 >] [--help]
[--no-preserve-owner] [--quiet] [--sparse] [--version] [目的目]
```

cpio 命令用于将目录结构拷贝到一个普通文件中, 从而实现备份功能。如果需要恢复目录结构, 可再用 cpio 命令将产生的 cpio 文件还原成目录结构。这两个过程通过使用不同的选项完成。

用-i 选项时, cpio 从标准输入设备读文件和目录名表, 并将其内容按文件格式拷贝到标准输出设备。使用-o 选项时, cpio 从标准输入设备读取先已建好的档案, 重建目录结构。用-d 选项, 使 cpio 根据需要自行建立目录。用-m 选项, cpio 不更换文件的时间信息。

cpio 命令常用以下命令制作一个完整的目录档案:

```
find fromdir -print | cpio -o > archive
```

根据档案文件重建一个目录结构命令为：

```
cpio -id < archive
```

例：将当前目录的所有内容，包括子目录备份到 mycpio.cpio 文件中。命令如下：

```
[globus@unknown globus]$ find -print | cpio . -o > mycpio.cpio
```

该命令执行时，它将显示处理的每一个文件信息，最后给出所备份的数据块数。该例在处理过程中显示的最后一部分信息如下：

```
cpio: kernel/module-init-tools-3.0/inmod.static: truncating inode number
cpio: keydata.tar: truncating inode number
cpio: zipfile.zip: truncating inode number
cpio: testvinew.old: truncating inode number
cpio: mycpio.cpio: truncating inode number
cpio: File ./mycpio.cpio grew, 924889088 new bytes not copied
3612849 blocks
```

产生的文件 mycpio.cpio 的信息如下：

```
[globus@unknown globus]$ ls mycpio.cpio -l
-rw-rw-r-- 1 globus globus 1849778688 6月 9 19:09 mycpio.cpio
```

例：将当前目录转到 my.cpio 目录，并提取（展开）mycpio.cpio 中的文件。

改变当前目录：

```
[globus@unknown globus]$ cd my.cpio
```

展开归档文件：

```
[globus@unknown my.cpio]$ cpio -id < ../mycpio.cpio
```

结果将在该目录下创建与备份之前相同的目录子树。

## 12.3 压缩命令

减少文件大小有两个明显的好处：

- (1) 减少存储空间。
- (2) 通过网络传输文件时，可以减少传输的时间。

Linux 提供了 compress、gzip、ungzip、zip、unzip、bzip2 和 bunzip2 等多种压缩工具。compress、gzip、zip 和 bzip2 命令分别用于压缩和解压缩，ungzip、unzip 和 bunzip2 只能用于解压缩相应的压缩文件。

### 12.3.1 compress 命令

compress 是一个非常古老的 Unix 文件压缩指令，现在已很少使用。

格式：compress 选项 文件列表

使用 compress 压缩文件时，将在原文件名之后加上扩展名.Z。如果不指定文件，则压缩标准输入，并将其结果返回标准输出。

compress 命令的常见选项如表 12-5 所示。

表 12-5 compress 命令的常用选项

选项	描述
-r	如果文件列表中包含目录，则递归压缩该目录及其子目录中的所有文件
-c	将压缩数据显示到标准输出，compress默认将结果输出到文件
-v	显示每个文件的压缩百分比
-d	对compress压缩的文件（.Z）解压缩

例：压缩/home/globus/vitest 文件。命令如下：

```
$ compress /home/globus/vitest
```

结果：压缩后生成 vitest.Z 文件。使用 compress -d vitest 命令。uncompress vitest 命令等价。

注意：compress -d 与 uncompress 等价；gzip -d 与 gunzip 等价，bzip2 -d 与 bunzip2 等价。

12.3.2 gzip 命令

gzip 是 Linux 系统最常用的一个文件压缩和解压命令，它方便实用。

gzip 命令的语法如下：

gzip [选项] 压缩（解压缩）的文件名

gzip 命令用 Lempel-ziv 编码压缩文件。gzip 默认将原文件压缩为.gz 文件，并删除原文件。

注意：

gzip 比 compress 的压缩比高。

其常用选项的含义如表 12-6 所示。

表 12-6 gzip 命令的常用选项

选项	描述
-c	将输出写到标准输出上，并保留原有文件
-d	对压缩文件进行解压缩
-l	对每个压缩文件，显示下列字段：压缩文件的大小，未压缩文件的大小，压缩比，未压缩文件的名字
-r	递归式地查找指定目录，并压缩或者解压缩其中的所有文件
-t	检查压缩文件是否完整
-v	对每一个压缩和解压缩的文件，显示文件名和压缩比
-num	用指定的数字num设置压缩比。-1或-fast表示最低压缩比，-9或-best表示最高压缩比。系统默认值为6

例：使用 gzip 压缩/home/globus/vitest 文件。命令如下：

```
$ gzip -v /home/globus/vitest
```

```
vitest: 62.2% -- replaced with vitest.gz
```

结果产生了 vitest.gz 压缩文件，原文件/home/globus/vitest 被删除。

例：在/home/globus/目录下，解压缩 vitest.gz 文件，命令如下：

```
$ gunzip -d vitest.gz
```



vitest 被解压缩，压缩文件 vitest.gz 被删除。

12.3.3 gunzip 命令

解压缩用 gzip 命令 (.gz)、使用 compress 命令 (.Z)，以及 zip 命令或者 Windows 系统的 winzip 命令 (.z) 压缩过的文件。

格式：gunzip 选项 文件列表

gunzip 默认解压缩指定文件。如果加入 -l 选项，将只列出压缩文件的内容，而不解压缩。选项 -r 递归解压缩，深入目录结构中，递归地解压缩指定目录中的文件及其子目录内的文件。

例：使用 gzip 命令压缩 wu-ftp-2.6.0.tar 文件。命令如下：

```
[globus@unknown globus]$ gzip -l wu-ftp-2.6.0.tar
              compressed      uncompressed  ratio uncompressed_name
              339140           1710080    80.2% wu-ftp-2.6.0.tar
```

例：解压缩 wu-ftp-2.6.0.tar.gz。命令如下：

```
$ gunzip wu-ftp-2.6.0.tar.gz
```

结果产生 wu-ftp-2.6.0.tar 文件，压缩文件 wu-ftp-2.6.0.tar.gz 被删除。

12.3.4 zip 命令

zip 与 Windows 的 winzip 压缩程序产生相同压缩文件 (.zip)。使用 zip 创建一个压缩文件通常为：

zip 压缩文件名 被压缩的文件列表

例：利用 zip 命令压缩当前目录中的 result.txt 文件和 vitestpart 文件，结果保存到 myzip.zip 文件。命令如下：

```
[globus@unknown globus]$ zip myzip.zip result.txt vitestpart
adding: result.txt (deflated 37%)
adding: vitestpart (deflated 72%)
```

命令执行结果是产生了 myzip.zip 压缩文件。

12.3.5 unzip 命令

在 Windows 系统下，使用 winzip 压缩软件压缩的文件可以用 unzip 命令解压缩，该命令用于解扩展名为.zip 的压缩文件。

语法：unzip [选项] 压缩文件名.zip

其常用选项如表 12-7 所示。

表 12-7 unzip 命令的常用选项

选项	描述
-x 文件列表	解压缩文件，但不包括“文件列表”中指定的文件
-v	查看压缩文件目录，但不解压
-t	测试文件有无损坏，但不解压

续表

选项	描述
-d 目录	把压缩文件解到指定目录下
-z	只显示压缩文件的注解
-n	不覆盖已经存在的文件
-o	覆盖已存在的文件且不要求用户确认
-j	不重建文档的目录结构, 把所有文件解压到同一目录下

例: 解压缩刚才生成的 myzip.zip 文件。

```
[globus@unknown globus]$ unzip myzip
Archive: myzip.zip
replace result.txt? [y]es, [n]o, [A]ll, [N]one, [r]ename: A
  inflating: result.txt
  inflating: vitestpart .
```

解压缩时, 由于原始文件已经存在, 提示用户输入下列字母进行选择: y 替代; n 不替代; A 替代所有; N 所有都不替代; r 重命名文件。在此输入 A, 表示替代所有文件。

### 12.3.6 bzip2 命令

bzip2 压缩程序采用新的压缩算法, 压缩效果比传统的 LZ77/LZ78 压缩算法都要好。若没有加上任何参数, bzip2 压缩完文件后会产生扩展名为.bz2 的压缩文件, 并删除原始的文件。

bzip2 的语法格式如下:

```
bzip2 [-cdfhkLstVz] [--repetitive-best] [--repetitive-fast] [-压缩等级] [要压缩的文件]
```

bzip2 的选项如表 12-8 所示。

表 12-8 bzip2 命令的常用选项

选项	描述
-c或--stdout	将压缩与解压缩的结果送到标准输出
-d或--decompress	执行解压缩
-f或--force	bzip2在压缩或解压缩时, 若输出文件与现有文件同名, 预设不会覆盖现有文件。若要覆盖, 需要使用此参数
-h或--help	显示帮助
-k或--keep	bzip2在压缩或解压缩后, 会删除原始的文件。若要保留原始文件, 需要使用此参数
-s或--small	降低程序执行时内存的使用量
-t或--test	测试.bz2压缩文件的完整性
-v或--verbose	压缩或解压缩文件时, 显示详细的信息
-z或--compress	强制执行压缩
-V或--version	显示版本信息
--repetitive-best	若文件中有重复出现的内容时, 可利用此参数提高压缩效果
--repetitive-fast	若文件中有重复出现的内容时, 可利用此参数加快执行速度
-压缩速度	压缩时的速度, 其值从-1(--fast)到-9(--best), 依次加快

例：使用压缩程序 bzip2 压缩当前目录下的归档文件 keydata.tar。命令如下：

```
[globus@unknown globus]$ bzip2 keydata.tar
```

命令执行结果是：产生 keydata.tar.bz2 文件，原有文件 keydata.tar 被删除。

例：对文件 keydata.tar.bz2 执行解压缩。命令如下：

```
[globus@unknown globus]$ bzip2 -d keydata.tar.bz2
```

### 12.3.7 bunzip2 命令

bunzip2 可解压缩.bz2 格式的压缩文件。bunzip2 实际上是 bzip2 的符号连接，执行 bunzip2 与 bzip2 -d 的效果相同。在此不再重复。

## 12.4 综合实例

用户经常在网上看到类似 wu-ftpd-2.6.0.tar.gz, linux-2.6.6.tar.bz2, oracle10g.cpio.gz 等的文件，这些文件名中同时包含了两个不同 tar、gz、bz2 或者 cpio 扩展名。这些文件名表示首先被归档（如使用 tar 或者 cpio），然后在被压缩（例如 gzip 或者 bzip2）。

### 12.4.1 各种与压缩和归档相关的文件名及其功能

对于归档后再压缩的文件都可以使用两条命令来完成，一些只需要一条命令即可。常见文件格式及其相应的压缩和加压缩操作，如表 12-9 所示。

表 12-9 备份文件取名惯例及其操作

文件名惯例	举例	生成命令	释放命令
xxx.tar	wu-ftpd-2.6.0.tar	tar cvf xxx.tar 原文件列表	tar xvf xxx.tar
xxx.gz	mytest.gz	gzip xxx	gzip -d xxx.gz
xxx.bz2	patch-2.6.6.bz2	bzip2 xxx	bzip2 -d xxx.bz2
xxx.tar.gz	wu-ftpd-2.6.0.tar.gz	tar cvzf xxx.tar.gz 原文件列表	tar xvzf xxx.tar.gz
xxx.tar.bz2	linux-2.6.6.tar.bz2	tar cvjf xxx.tar.bz2 原文件列表	tar xvjf xxx.tar.bz2
xxx.tgz	testtargzip.tgz	tar cvzf xxx.tar.tgz 原文件列表	tar xvzf xxx.tgz
xxx.cpio	oracle10g.cpio	find 目录 -print   cpio -o > xxx.cpio	cpio -id < xxx.cpio
xxx.cpio.bz2	oracle10g.cpio.bz2	find 目录 -print   cpio -o > xxx.cpio; bzip2 xxx.cpio	bzip2 xxx.cpio.bz2; cpio -id < xxx.cpio

### 12.4.2 常见文件格式操作实例

（1）对于 xxx.tar.bz2 形式文件。

例：从压缩文件 linux-2.6.6.tar.bz2 中展开 Linux2.6.6 版本的所有内核源代码文件，有两种方式。

- 一条命令完成：  
tar xvjf linux-2.6.6.tar.bz2
- 使用两条命令完成：  
bzip2 linux-2.6.6.tar.bz2

```
tar xvf linux-2.6.6.tar
```

(2) 对于 xxx.tar.gz 形式的文件。

例：从压缩文件 wu-ftp-2.6.0.tar.gz 中展开 ftp 服务器的所有源代码文件，也有两种方式。

- 一条命令完成：

```
tar xvzf wu-ftp-2.6.0.tar.gz
```

- 使用两条命令完成：

```
gzip -d wu-ftp-2.6.0.tar.gz
```

```
tar xvf wu-ftp-2.6.0.tar
```

(3) 将当前目录下的 kernel 子目录和 newdir3 子目录使用 tar 递归归档，使用 gzip 压缩归档的文件，并将压缩的归档文件保存到 my.tar.gz 文件中有两种方式。

- 一条命令完成：

```
tar cvzf my.tar.gz kernel newdir3
```

- 使用两条命令完成：

```
tar cvf my.tar kernel newdir3
```

```
gzip my.tar
```

(4) 将当前目录下的 kernel 子目录和 newdir3 子目录使用 tar 递归归档，使用 bzip2 压缩归档的文件，并将压缩的归档文件保存到 my.tar.bz2 文件中。有两种方式。

- 一条命令完成：

```
tar cvjf my.tar.bz2 kernel newdir3
```

- 使用两条命令完成：

```
tar cvzf my.tar kernel newdir3
```

```
bzip2 my.tar
```

(5) 解压缩 patch-2.6.6.bz2，得到文件 patch-2.6.6，命令如下：

```
bzip2 -d patch-2.6.6.bz2
```

注意：gzip 和 bzip2 压缩后的目标文件 (.gz 或者 .bz2) 中都只包含一个文件，因此如果用户希望使用它们将多个文件压缩在一个文件中，必须首先使用 tar 或者 cpio 将它们归档成一个文件后，再使用 gzip 和 bzip2。

## 思考与练习

- (1) 在 Linux 中，常见的归档命令有哪些？
- (2) 在 Linux 中有哪些常见的压缩命令？
- (3) 评价各种压缩命令的效率。
- (4) Linux 中常见的压缩和归档使用方式有哪些？它们如何使用？

## 第 13 章 应用软件安装方法

Linux 操作系统装好了,以后根据需要再另外安装一些应用软件是难免的。但是在 Linux 中,安装应用软件和 Windows 下的方法有很大的不同。对于 Linux 初学者来说,安装软件是一件很让人头疼的事情。由于很多 Linux 应用软件都是以源代码的形式发布的,这就要求用户根据自己系统的实际情况和自身的需要来配置、编译源程序后,才能使用这些软件。

本章首先介绍编译源代码的安装方法,然后介绍 RPM 包的命令行安装方法,最后介绍使用图形工具安装 RPM 包的方法。当然还有其他的安装方法,在此不再介绍。

### 13.1 源代码安装

大多数 Linux 初学者往往不知道该如何进行配置和编译。下面笔者就简单介绍一下如何安装 Linux 应用软件。在安装应用软件之前,还要确保软件安装的编译环境,比如编译器 GCC、必要的库文件等。不过,用户不需要担心这些问题,因为大多数 Linux 发行版本都已经默认安装了 GCC 编译器。

目前流行的软件包有两种比较常见的形式,一种是以 RPM 为代表的软件包,另一种是 file.tar.gz (或者 file.tar.bz2) 形式的压缩的源代码。本文将介绍 file.tar.gz 的安装、RPM 软件包在文本环境和图形环境下的安装方法。

应用软件的源代码一般是 file.tar.gz、file.tgz 或者 file.tar.bz2 格式,也就是说是以 tar 来打包,然后用 gzip 或者 bzip2 来压缩。

以应用软件 parted-1.6.9.tar.gz 为例,parted 是一个用于创建、删除、改变分区大小、检查和复制分区和文件系统的工具,它与 Windows 系统下的 PQMagic 的功能相同,功能非常强大的自由软件。用户可以从 <http://www.gnu.org/software/parted/parted.html#downloading> 网站获得现在最新的版本。

安装方法如下:

先进入存放安装文件 parted-1.6.9.tar.gz 的目录,然后使用 gzip 解,以及 tar 解包。命令如下:

```
[root@jiang tools]# tar xvzf parted-1.6.9.tar.gz
```

进入解压完成之后新建立的目录,一般的应用软件中都有一个用来配置软件的可执行文件 configure (执行时,它会给出许多处理的细节信息),它的参数比较多,而且用法也非常灵活。当然,安装不同的软件参数也不相同,具体有些什么参数,可以用命令 ./configure help 取得详细的帮助。

```
[root@jiang tools]# cd parted-1.6.9
```

```
[root@jiang parted-1.6.9]# ./configure
```

注意: part 软件需要 e2fsprogs-1.32-6.i386.rpm 包和 uuid 包。

在运行 ./configure [options] 配置软件后,它会根据你当前系统、编译、安装相关的信息,生成一个安装配置文件 Makefile。Makefile 文件通常是用来编译和安装软件的,运行 make

命令时系统会自动根据 Makefile 文件中的设置对软件进行编译和安装。

```
[root@jiang parted-1.6.9]# make
```

如果编译能够顺利通过的话，运行 `#make install` 命令来安装。

```
[root@jiang parted-1.6.9]# make install
```

然后可以直接使用 parted 软件，命令如下：

```
[root@jiang parted-1.6.9]# parted
```

## 13.2 RPM 包在文本环境的安装

RPM 是 Redhat Package Manager 的缩写，是由 RedHat 公司开发的软件包安装和管理程序，同 Windows 平台上的 installer 比较类似。使用 RPM，用户可以自行安装和管理 Linux 上的应用程序和系统工具。

RPM 可以让用户直接以 binary 方式安装软件包，并且可替用户查询是否已经安装了有关的库文件；在用 RPM 删除程序时，它又会聪明地询问用户是否要删除有关的程序。

如果使用 RPM 来升级软件，RPM 会保留原先的配置文件，这样用户就不用重新配置新的软件了。RPM 保留一个数据库，这个数据库中包含了所有的软件包的资料，通过这个数据库，用户可以进行软件包的查询。RPM 虽然是为 Linux 而设计的，但是它已经移植到 SunOS、Solaris、AIX、Irix 等其他 UNIX 系统上了。RPM 遵循 GPL 版权协议，用户可以在符合 GPL 协议的前提下自由使用和传播 RPM。

### 13.2.1 使用 RPM

RPM 共有 4 种基本的模式：安装、查询、验证、删除。

- 安装模式：`rpm -i [安装选项] <软件包>`
- 查询模式：`rpm -q [查询选项]`
- 验证模式：`rpm -V 或 -verify [验证选项]`
- 删除模式：`rpm -e <软件包>`

使用 RPM 的一般选项（用于所有模式）如表 13-1 所示。

表 13-1 rpm 用于所有模式的常见选项

选项	描述
-v	显示信息
-h	用“#”显示完成的进度
--quiet	安静工作，只有当出现错误时才给出提示信息
--help	显示帮助

### 13.2.2 安装和升级选项

使用 RPM 来安装的一般格式为：`rpm -i [安装选项] <软件包>`

使用 RPM 来升级的一般格式为：`rpm -U [安装选项] <软件包>`

例：要安装一个新的软件包，它的名字为 `e2fsprogs-1.32-6.i386.rpm`。使用下面的命令

就可以把它安装到系统上:

```
[root@jiang tools]# rpm -ivh e2fsprogs-1.32-6.i386.rpm
warning: e2fsprogs-1.32-6.i386.rpm: V3 DSA signature: NOKEY, key ID db42a60e
Preparing... ##### [100%]
1:e2fsprogs ##### [100%]
```

升级同安装是一样的,只不过升级会把原来低版本的文件删除掉。RPM 可以使用 FTP 安装,这时要指定 FTP 的完整路径。使用 FTP 方式安装时, RPM 会先下载所需的软件包,然后再安装。

当安装的软件版本已经安装在系统上,或者是系统上现有的版本比要安装的版本高,系统就会发出警告。这时,用户如果坚持安装的话,可以用--force 选项。

使用 RPM 安装前, RPM 会检查该软件包的依赖关系,即正确运行该软件包所需的其他软件包是否已经安装。如果没有安装的话,系统会给出警告。这时,用户如果要强行安装,可以使用选项--nodeps。但是作者并不推荐这种做法,因为这样安装的软件十有八九是不能运行的。

不实际安装,只是检查并显示可能存在的冲突,使用选项--test。

### 13.2.3 查询选项

查询 RPM 包的命令格式如下:

```
rpm -q [查询选项] [软件包名字]
```

使用“软件包名字”选项,查询系统上是否已经安装了这个软件包。

查询常用选项如表 13-2 所示。

表 13-2 rpm 用于查询模式的常见选项

选项	描述
-a	查询所有安装的软件包信息
-i	显示软件包的信息,包括名称、版本、描述信息
-R	显示这个软件包所依赖的软件包
--provides	显示这个软件包所提供的功能
-l	列出这个软件包内所包含的文件
-s	显示这个软件包的状态,状态是指normal、not installed、replaced

例:查询已安装的 e2fsprogs-1.32-6.i386.rpm 包的信息。命令如下:

```
[root@jiang tools]# rpm -qi e2fsprogs-1.32-6
```

将会显示 e2fsprogs-1.32-6 包的详细信息。

### 13.2.4 验证选项

验证的一般格式为: rpm -V|--verify [验证选项]

验证会把软件包所安装的软件与原先数据库中存储的软件信息做一个比较。用户可以用 RPM 来验证已经安装的软件包, RPM 可以验证文件的大小, MD5 校验码, 文件权限, 类型, 属主等信息。

验证输出的格式是一个包括 8 个字符的字符串，这 8 个字符中每个字符都代表同数据库中信息所比较的结果。如结果是“.”，就代表没有问题。下面是相应字符分别为：S（MD5 校验错）、D（设备文件错）、U（用户名错）、G（组名错）、S（文件大小错）、L（符号链接错）、T（文件修改时间错）、M（属性错，包括读写权限和文件类型）。

### 13.2.5 删除选项

删除包的一般格式为：

```
rpm -e <软件名>
```

注意：这里只能是软件名，而不是软件包的文件名。

如果要删除的软件包是别的软件所必需的，在删除时用户会得到警告。这时，如果确认要强行删除，可以使用--nodeps 选项。使用--test 选项不真正删除，只是测试。

### 13.2.6 RPM 使用举例

#### 1. 软件安装

如果用户手头上有一个 RPM 软件包，可以使用命令：

```
# rpm -ivh e2fsprogs-1.32-6.i386.rpm
```

如果用户安装的软件包已经安装，再安装的话会出现下列错误信息：

```
root@jiang tools]# rpm -ivh e2fsprogs-1.32-6.i386.rpm
warning: e2fsprogs-1.32-6.i386.rpm: V3 DSA signature: NOKEY, key ID db42a60e
Preparing... ##### [100%]
package e2fsprogs-1.32-6 is already installed
```

在安装软件时，可能出现如下错误：

#### ● 软件包之间有冲突

假如用户要安装的软件包里的某些文件，先前已经有其他的软件包安装过，即各种软件包之间有冲突，则会出现相应的错误信息。

#### ● 缺少相关的文件

有时安装的 RPM 软件包需要其他的部件才能安装成功。这时要强行安装，可以使用选项--nodeps，示例如下：

```
[root@jiang tools]# rpm -ivh e2fsprogs-1.32-6.i386.rpm --nodeps
```

#### 2. 删除软件

要删除软件非常简单，只要执行下面的命令就行：

```
[root@jiang tools]# rpm -e e2fsprogs-1.32-6
```

注意：使用的是软件的名称 e2fsprogs-1.32-6，而不是软件包的名称 e2fsprogs-1.32-6.i386.rpm。

如果要删除的软件是其他软件所需要的，用户会得到相应的错误信息。若用户一定要删除，可以使用下面的命令：

```
[root@jiang tools]# rpm -e e2fsprogs-1.32-6 --nodeps
```



### 3. 升级软件

如果用户以前安装过 `e2fsprogs-1.32-6.i386.rpm`, 现在又得到了 `e2fsprogs-1.35.i386.rpm`, 则使用下面的命令可以升级这个软件包:

```
[root@jiang tools]# rpm -Uvh e2fsprogs-1.35.i386.rpm
```

**注意:** `-U` 参数实际上是告诉 RPM 先删除原先的版本, 再安装新的版本。所以不管以前是否安装过这个软件包, 用户都可以使用 `rpm -Uvh` 命令来安装软件。

### 4. 查询软件包

如果用户得到一个软件包, 却不知道它里面有哪些内容, 可以使用命令:

```
[root@jiang tools]# rpm -qpl e2fsprogs-1.32-6.i386.rpm
```

如果用户想知道它会在系统上安装哪些内容, 可以使用命令:

```
[root@jiang tools]# rpm -qpl e2fsprogs-1.32-6.i386.rpm
```

如果用户在 Internet 上遇到一个软件包, 想要知道它里边的内容时, 可以使用命令:

```
[root@jiang tools]# rpm -qpl ftp://ftp.redhat.com/redhat/RPMS/  
e2fsprogs-1.32-6.i386.rpm
```

### 5. 校验

如果用户不小心删除了几个文件, 却又不知道到底删除了哪些文件, 可以使用下面的命令对系统进行校验:

```
[root@jiang tools]# rpm -Va
```

**注意:** `xxx.i386.rpm` 与 `xxx.src.rpm` 是不同的。

在 Internet 上, 用户经常会看到这样的目录: `RPMS/` 和 `SRPMS/`。目录 `RPMS/` 下面存放的就是一般的 RPM 软件包, 这些软件包是由软件的原代码编译生成的可执行文件, 再包装成 RPM 软件包的。`SRPMS/` 目录下存放的都是以 `.src.rpm` 结尾的文件, 这些文件是由软件的源代码包装而成的, 用户要安装这类 RPM 软件包, 必须使用命令:

```
# rpm --recompile e2fsprogs-1.35.src.rpm
```

这个命令会把源代码解包并编译、安装它, 如果用户使用命令:

```
# rpm --rebuild e2fsprogs-1.35.src.rpm
```

那么, 在安装完成后, 还会把编译生成的可执行文件重新包装成 `i386.rpm` 的 RPM 软件包。

## 13.3 RPM 包在 GUI 环境中的安装

相对命令方式来说, 在图形环境下, 安装、升级、卸载和查询 RPM 软件包是一件非常简单轻松的事情,

### 13.3.1 使用文件管理器 Nautilus

图形化桌面包括了一个叫做 Nautilus 的文件管理器。它给用户提供了系统和个人文件的图形化显示。然而, Nautilus 不仅仅是文件的可视列表, 它还允许你从一个综合界面来配

置桌面、配置你的 Red Hat Linux 系统、浏览影集和访问网络资源等等。总之，Nautilus 已成为你整个桌面的“外壳”（shell）。

Nautilus 不仅提供了高效的工作环境，它还为你提供了另一种漫游文件系统的方法。用户可以在与“主菜单”相连的各类子菜单中搜索，或者使用 shell 提示来漫游文件系统。

要作为文件管理器来启动 Nautilus，双击主目录图标：

Nautilus 出现后，可以在主目录中或文件系统的其他部分漫游。要回到主目录，单击“主目录”按钮。

浏览器窗口中包含文件夹和文件。可以使用鼠标来把它们拖放或复制到新位置。可以选择“文件”|“新建窗口”命令，打开一个新的 Nautilus。打开了一个新的 Nautilus 之后，用户就可以把文件拖放到不同的目录中。按照默认设置，把文件从一个目录中拖放到另一个目录中会移动文件。要把文件复制到另一个目录中，在拖放时按 Ctrl 键。

按照默认设置，主目录中的图像文件会被显示为缩略图标（thumbnail）。对于文本文件来说，这意味着你会在图标中看到实际文本的一部分。对于图像文件来说，会看到一个该图像的按比例缩小的版本（或 thumbnail）。要关闭这项功能，选择“编辑”|“首选项”，从左侧的菜单中选择“预览”；从“显示缩略图标”的下拉菜单中选择“从不”。禁用这项（以及其他）预览功能会加快 Nautilus 的运行速度。

### 13.3.2 使用 Nautilus 安装 RPM 包

用户可以双击 Nautilus 窗口中的 RPM 包文件，弹出一个 RPM 安装对话框，然后就会自动安装该软件包。

## 思考与练习

- （1）在 Linux 中，有哪些常见的软件安装方法？
- （2）RPM 有哪几项大的功能？
- （3）从源代码安装软件的步骤是什么？

## 第 14 章 系统全局设置

对 Linux 系统进行全局设置是控制 Linux 系统并使之为用户服务的一个重要手段。

本章首先介绍了 Linux 系统中特有的 `/proc`（内存）文件系统的组成，通过获得系统多种重要信息的方法。然后介绍了设置 Linux 系统主机名的方法，接着介绍 Linux 系统引导时的重要的配置文件 `inittab` 和 `/etc/rc.d`（子目录树）的组成和作用，之后介绍了如何管理系统服务的方法，包括使用 `ntsysv` 和 `chkconfig` 设置服务的运行级别，以及为特定服务创建启动脚本的方法。

### 14.1 获得 Linux 系统信息

在 Linux 里，用户可以使用 `proc` 虚拟文件系统来访问各种系统信息，包括当前正在运行的进程和电池状态。由于 `proc` 文件系统在内存中创建，并在系统状态更改时动态地更新，所以 `proc` 文件系统是虚拟的。要获取系统信息的访问权，用户应该使用标准文件 API（如 `fopen` 和 `fread`），在 `proc` 文件系统里读取适当的文件。

```
[globus@wang globus]$ cat /proc/swaps
```

Filename	Type	Size	Used	Priority
/dev/hdc10	partition	514040	0	-1

输出结果是易读的，而且很容易在代码中分析。但不是每种 `proc` 文件的格式都一样，因为数据的指令不同，不过它们都是大同小异的。

如果仔细观察输出，就会发现分析这些字段异常简单。

利用 `proc` 文件系统可以获得如下信息：

（1）进程信息：对于系统中的任何一个进程来说，在 `/proc` 的子目录里都由一个与进程 ID 同名的子目录。在这个目录下，用户将可以找到以下条目：`cmdline`、`cwd`、`environ`、`exe`、`fd`、`maps`、`mem`、`root`、`stat`、`statm` 和 `status`。某些信息仅对超级用户账号可见，例如进程的根目录。使用其中的 `cmdline` 文件可以获取进程的命令行信息。

（2）高级电源管理（APM）信息：如果需要了解电池状态，用户也可以从 `/proc/apm` 文件中获得。它包含了一个系统在使用 AC 还是电池、电池的充电程度以及电池的剩余时间的标记。不过，剩余时间的标记不太准确。

（3）CPU 信息：利用 `cpuinfo` 文件，用户可以获得 CPU 当前的准确信息。例如：

```
[globus@wang proc]$ cat /proc/cpuinfo | more
```

结果反映了 CPU 的编号、供应商 ID、CPU 家族、型号、CPU 频率和 `cache` 大小等信息，该例表明处理器编号为 0，供应商为 Intel，CPU 家族编号为 6，型号为赛扬 CPU（Celeron(TM)），CPU 频率为 1000MHz，`cache` 为 256KB。结果如下：

```
processor       : 0
vendor_id      : GenuineIntel
cpu family     : 6
```

```

model          : 11
model name     : Intel(R) Celeron(TM) CPU          1000MHz
stepping      : 1
cpu MHz       : 1004.311
cache size    : 256 KB
fdiv_bug      : no
hlt_bug       : no
f00f_bug      : no
coma_bug      : no
fpu           : yes
fpu_exception : yes
cpuid level   : 2
wp            : yes
flags         : fpu vme de pse tsc msr pae mce cx8 sep mtrr pge mca cmov pat p
se36 mmx fxsr sse
bogomips      : 2005.40

```

(4) 负载信息: `loadavg` 文件包含了负载信息。负载是系统被使用的程度。它由 3 个从 0~1 变化的数字来行量。其中一个是最高利用率, 一个是平均利用率, 一个是最低利用率。平均值是最有用的。

例: 显示当前到俄负载信息。命令如下:

```

[globus@wang proc]$ cat /proc/loadavg
0.00 0.00 0.00 1/39 2561

```

结果表明系统的负载最轻。

(5) 系统内存信息: `meminfo` 文件包含了系统内存的详细信息。它显示了物理内存的数量, 可用交换空间的数量, 空闲内存的数量等。例如:

```

[globus@wang proc]$ cat /proc/meminfo

```

结果如下:

```

total:      used:      free:  shared: buffers:  cached:
Mem: 195203072 42819584 152383488      0 6311936 21106688
Swap: 526376960      0 526376960
MemTotal:      190628 kB
MemFree:      148812 kB
MemShared:      0 kB
Buffers:      6164 kB
Cached:      20612 kB
SwapCached:      0 kB
Active:      32568 kB
ActiveAnon:      6264 kB
ActiveCache:      26304 kB

```

```

Inact_dirty:      136 kB
Inact_laundry:    0 kB
Inact_clean:     336 kB
Inact_target:    6608 kB
HighTotal:        0 kB
HighFree:         0 kB
LowTotal:         190628 kB
LowFree:          148812 kB
SwapTotal:        514040 kB
SwapFree:         514040 kB

```

(6) 设备和系统状态: `/proc/pci` 中包括了 `pci` 设备的系统信息, `/proc/devices` 文件包括系统中所有的字符和块设备信息, `/proc/filesystems` 包含所有的文件系统, `/proc/interrupts` 包含系统中断信息, `/proc/iomem` 包含了设备的端口地址, `partitions` 包含了磁盘分区信息, `dma` 包含了 DMA 通道的信息, `uptime` 引导时的时间, `version` 系统的版本信息。swaps 包含了 swap 分区及其使用状态。bus 目录中包含了系统总线的信息, 常用的有 PCI、USB 或者 SCSI 总线。ide 包含了 IDE 设备的信息 (硬盘或者光盘信息)。SCSI 目录包含了 scsi 设备信息。net 目录包含了与网络有关的所有信息, 包括网络协议各层的信息 (如 arp、igmp、snmp、tcp 和 udp 等)、路由表 (route) 和网络状态 (netstat) 等信息。

例: 显示系统中的中断使用情况。命令如下:

```
[globus@wang proc]$ cat interrupts
```

结果如下:

```

          CPU0
0:      392287      XT-PIC  timer
1:         4      XT-PIC  keyboard
2:         0      XT-PIC  cascade
8:         1      XT-PIC  rtc
11:     33570      XT-PIC  usb-uhci, usb-uhci, eth0
12:        42      XT-PIC  PS/2 Mouse
14:       150      XT-PIC  ide0
15:       3807      XT-PIC  ide1
NMI:         0
ERR:         0

```

## 14.2 设置主机名

### 14.2.1 显示主机名

Linux 系统提供了 `hostname` 来查询系统的主机名信息。实例如下:

```
[globus@wang net]$ hostname
```

Wang

结果表明主机名为 wang，这与 shell 提示中显示的结果一样。

### 14.2.2 临时更改主机名

为了某种目的，通常需要更改系统的主机名，hostname 可以临时更改系统的主机名，但是在下次重新引导系统后，更改的主机名将不再生效。

将主机名 wang 更改成 wangwang，以超级用户身份执行如下命令：

```
[root@wang root]# hostname wangwang
```

显示修改后的结果：

```
[root@wang root]$ hostname
```

```
wangwang
```

### 14.2.3 永久更改主机名

为了永久的更改主机名，可以使用如下更改配置文件的方法实现。具体如下：

(1) 编辑/etc/sysconfig/network 文件。如果将主机名永久改变成 wangwang，那么/etc/sysconfig/network 文件内容应改为：

```
NETWORKING=yes
```

```
HOSTNAME=wangwang
```

(2) 重新引导系统，登录后成功后，出现如下提示符：

```
[globus@wangwang globus]$
```

使用 hostname 命令，显示系统的主机名：

```
[globus@wangwang globus]$ hostname
```

```
wangwang
```

表明主机名更改成功。

## 14.3 管理 hosts 文件

允许用户从/etc/hosts 文件中添加、编辑、或删除主机。该文件包含 IP 地址和它们相对应的主机名。

当系统试图把主机名解析为 IP 地址或判定 IP 地址的主机名时，它在使用名称服务器前，首先它首先在/etc/hosts 文件（是 Red Hat Linux 的默认配置）查找。如果 IP 地址被列在/etc/hosts 文件中，名称服务器就不会被使用。如果用户的网络包括没有列在 DNS 内的 IP 地址，推荐把它们添加到/etc/hosts 文件中。

Red Hat Linux 默认的 hosts 文件的内容与下属文件类似：

```
[globus@wangwang globus]$ cat /etc/hosts
```

```
# Do not remove the following line, or various programs
```

```
# that require network functionality will fail.
```

```
127.0.0.1          wang localhost.localdomain localhost
```

hosts 文件的格式如下：

IP 地址      全限定的主机名    主机别名

例：局域网的域名为 `some.com.cn`，有一台主机的主机名为 `somehostname`，IP 地址为 `192.168.1.240`，如果用户希望使用全限定的主机名 `somehostname.some.com.cn`，别名为 `somehostname`，则应在 `hosts` 文件中加入下述命令。

```
192.168.1.240    somehostname.some.com.cn    somehostname
```

那么在以后需要访问 `192.168.1.240` 时，可以使用其 IP 地址，全限定的主机名或者别名，效果是一样的，如下述 3 条命令的功能是相同的：

- `ping 192.168.1.240`
- `ping somehostname.somedomain.com.cn`
- `ping somehostname`

注意：任何一台主机希望使用全限定的主机名 `somehostname.somedomain.com.cn` 的主机，都需要在它的 `hosts` 文件文件中加入一行 `somehostname.somedomain.com.cn` 对应的纪录。

典型的 `hosts` 文件如下：

```
[globus@wang globus]$ cat /etc/hosts
127.0.0.1            localhost.localdomain localhost
192.168.1.190       wang.somedomain.com.cn wang
192.168.1.240       somehostname.somedomain.com.cn    somehostname
```

它将本地主机也加入到 `hosts` 中了，这样在使用主机名 `wang` 时，不再代表 `127.0.0.1`，而代表 `192.168.1.240`。

## 14.4 配置文件 `inittab`

`init` 进程是系统所有进程的起点，内核在完成核内引导以后，即在本线程（进程）空间内加载 `init` 程序，它的进程号是 1。`init` 程序需要读取 `/etc/inittab` 配置文件。`inittab` 是以行为单位的描述性（非执行性）文本，每一个指令行都具有以下格式：

```
id:runlevel:action:process arguments
```

其中 `id` 为入口标识符，标示符长度可以是 2~4 个字符，对于 `getty` 或其他 `login` 程序项，要求 `id` 与 `tty` 的编号相同，否则 `getty` 程序将不能正常工作。`runlevel` 为运行级别，它定义了该行应用哪个运行级别。`action` 为动作代号，`process` 定义了需要运行什么进程，用户可以在其后定义任意个参数。

可以使用如下命令显示 `/etc/inittab` 文件的内容：

```
$ cat /etc/inittab
```

`runlevel` 是 `init` 所处于的运行级别的标识，一般使用 0~6 以及 S 或 s。0、1、6 运行级别被系统保留，0 作为 shutdown 动作，1 将重新引导到单用户模式，6 为系统重启；S 和 s 意义相同，表示单用户模式，且无需 `inittab` 文件，因此也不在 `inittab` 中出现，实际上，进入单用户模式时，`init` 直接在控制台（`/dev/console`）上运行 `/sbin/sulogin`。

在一般的系统实现中，都使用了 2、3、4、5 几个级别。在 Redhat 系统中，2 表示无 NFS 支持的多用户模式，3 表示完全多用户模式（也是最常用的级别），4 保留给用户自定

义, 5 表示启动 X Windows 方式和 XDM 图形登录方式。7~9 级别也是可以使用的, 传统的 Unix 系统没有定义这几个级别。runlevel 可以是并列的多个值, 以匹配多个运行级别, 对大多数 action 来说, 仅当 runlevel 与当前运行级别匹配成功才会执行。

例: 如果想让 init 在单用户和多用户模式下都运行一个进程命令如下:

```
id:13:action:process arguments
```

initdefault 是一个特殊的 action 值, 用于标识默认的启动级别; 当 init 由核心激活以后, 它将读取 inittab 中的 initdefault 项, 取得其中的 runlevel, 并作为当前的运行级别。如果没有 inittab 文件, 或者其中没有 initdefault 项, init 将在控制台上请求输入 runlevel。

sysinit、boot、bootwait 等 action 将在系统启动时无条件运行, 而忽略其中的 runlevel, 其余的 action (不含 initdefault) 都与某个 runlevel 相关。各个 action 的定义可以通过 man inittab 命令查看手册页的说明。

在 Redhat 系统中, 一般情况下 inittab 都会有如下几项:

```
id:3:initdefault:
```

表示当前默认运行级别为 3, 即完全多任务模式, 即命令行方式。如果希望以图形方式启动, 需要将其级别改为 5, 即 id:5:initdefault:。

```
si::sysinit:/etc/rc.d/rc.sysinit
```

表示启动时自动执行/etc/rc.d/rc.sysinit 脚本。

```
13:3:wait:/etc/rc.d/rc 3
```

表示当运行级别为 3 时, 以 3 为参数运行/etc/rc.d/rc 脚本, init 将等待其返回。

```
0:12345:respawn:/sbin/mingetty tty0
```

表示在 1~5 的各个级别上以 tty0 为参数执行/sbin/mingetty 程序, 打开 tty0 终端用于用户登录, 如果进程退出则再次运行 mingetty 程序。

```
x:5:respawn:/usr/bin/X11/xdm -nodaemon
```

表示在 5 级别上运行 xdm 程序, 提供 xdm 图形方式登录界面, 并在退出时重新执行。

## 14.5 rc 启动脚本

上一节已经提到 init 进程将启动运行 rc 脚本, 本节将介绍 rc 脚本具体的工作过程。

在 Red Hat Linux 系统中, rc 启动脚本都位于/etc/rc.d 目录下, rc.sysinit 文件中最常见的动作就是激活交换分区, 检查磁盘, 加载硬件模块, 这些动作无论哪个运行级别都是需要优先执行的, rc.local 文件可以执行仅当 rc.sysinit 执行完以后 init 才会执行其他的 boot 或 bootwait 动作。所有的 init 脚本执行完成之后, 将要执行 rc.local 文件中的脚本, 用户可以将自己的初始化代码放在该文件中, 以便进行一些必要的初始化工作。

/etc/rc.d 目录下除了包含 rc.sysinit、rc.local、rc.serial 和 rc 四个文件以外, 还有 7 个子目录, 它们分别是: init.d、rc0.d、rc1.d、rc2.d、rc3.d、rc4.d 和 rc5.d。init 目录中存放了所有服务的 shell 脚本文件, 每一个服务对应一个文件, 这些脚本能接受 start、stop、restart、status 等参数。rc0.d、rc1.d、rc2.d、rc3.d、rc4.d 和 rc5.d 都是指向/etc/rc.d/init.d/目录下各个 Shell 脚本的符号连接, 分别对应 runlevel 的 0~5 的 5 个级别, 当系统属于 x 级别时, 它将执行 rcx.d 子目录中的所有脚本, 其中 x 为 1~5 中的任意值。



如果没有其他的 boot, bootwait 动作, 在运行级别 3 下, /etc/rc.d/rc 将会得到执行, 命令行参数为 3, 即执行 /etc/rc.d/rc3.d/ 目录下的所有文件。rc3.d 下的文件都是指向 /etc/rc.d/init.d/ 目录下各个 Shell 脚本的符号连接, 而这些脚本一般能接受 start、stop、restart、status 等参数。rc 脚本以 start 参数启动所有以 S 开头的脚本, 在此之前, 如果相应的脚本也存在 K 打头的链接, 而且已经处于运行态了 (以 /var/lock/subsys/ 下的文件作为标志), 则将首先启动 K 开头的脚本, 以 stop 作为参数停止这些已经启动了的服务, 然后再重新运行。显然, 这样做的直接目的就是当 init 改变运行级别时, 所有相关的服务都将重启, 即使是同一个级别。

## 14.6 管理系统服务

由于 Red Hat Linux 启动时使用了大量的文件、符号链接和目录。可以使用一些工具简化对初始化文件的管理。

### 14.6.1 使用 ntsysv 管理服务

使用 ntsysv 命令管理系统引导时启动的服务。它提供了简单的菜单界面开启或者关闭任何级别上的服务。如果不带任何参数运行该命令, 则屏幕上显示如图 14-1 所示的窗口。

可以滚动的窗口显示了当前运行级别下, 系统引导时是否要启动的所有服务。可以按空格键选中某服务 (前面打星号\*), 使系统引导时启动该服务, 如果想关闭服务, 只要再次按空格键, 将前面的星号去掉即可。修改完成后单击“确定”按钮, 保存所作的修改, 或者单击“取消”按钮, 取消所作的修改。如果修改其他级别的配置, 可以使用 --level 选项提供希望修改的级别。例如:

```
[root@jiang tools]# ntsysv --level 5
```

将可以修改 runlevel 5 的配置了。



图 14-1 ntsys 程序的主界面

### 14.6.2 使用 chkconfig 管理服务

chkconfig 程序可以使用户设置开启或者终止某一级别服务的符号链接。该程序可以管理/etc/rc.d/init.d 中所有的脚本的运行级别配置。它的作用将在下面详细说明。

#### 1. 列出可管理的服务列表

要查看服务将在哪个运行级别上开启或者停止，可执行下列命令：

```
chkconfig --list
```

该命令将列出所有可被 chkconfig 管理的服务，并按运行级别的顺序显示开启或者终止那些服务。也可以查看某一个特定服务的各运行级别的配置情况，只要在命令中指定服务名称，即/etc/rc.d/init.d 中的文件名。

例：查看 sendmail 服务的配置情况。命令如下：

```
[root@jiang tools]# chkconfig --list sendmail
```

结果如下：

```
sendmail          0:关闭  1:关闭  2:启用  3:启用  4:启用  5:启用  6:关闭
```

结果表明：sendmail 在级别 2、3、4、5 上运行，而在级别 0 和 6 上关闭。

#### 2. 加入新服务

如果在/etc/rc.d/init.d 目录中添加了新的脚本，并想在特定的级别上才提供该服务，可以执行如下 chkconfig 命令：

```
chkconfig [--level 运行级别] 服务名 on
```

例：在当前运行级别上添加名为 httpd 的服务。命令如下：

```
# chkconfig httpd on
```

例：在当行级别 3 上添加名为 httpd 的服务。命令如下：

```
# chkconfig --level 3 httpd on
```

#### 3. 删除已有服务

可以使用如下命令删除已有的服务。命令格式为：

```
chkconfig [--level 运行级别] 服务名 off
```

例：在当前运行级别上删除名为 httpd 的服务。命令如下：

```
# chkconfig httpd off
```

例：在当行级别 3 上删除名为 httpd 的服务。命令如下：

```
# chkconfig --level 3 httpd off
```

### 14.6.3 手工启动和停止系统服务

使用 ntsysv 和 chkconfig 对系统设置后，只有在下次系统引导时才起作用。重新引导系统是一件令人烦恼的事情，用户可以使用如下的方法手工和停止启动系统中服务。

### 1. 使用 service 命令

service 的语法格式如下:

`service 服务名 start | stop | restart | status`

其中 `start` 表示启动“服务名”对应的服务, `stop` 表示停止“服务名”对应的服务, `restart` 表示重新启动“服务名”对应的服务, `status` 表示查看服务的状态(运行或者停止)。

例: 首先查看 `sshd` 服务的状态, 然后停止该服务, 再启动该服务, 查看运行状态, 最后重新运行该服务。执行过程如下:

```
[root@jiang root]# service sshd status
sshd (pid 1390 1065) 正在运行...
[root@jiang root]# service sshd stop
停止 sshd: [ 确定 ]
[root@jiang root]# service sshd status
sshd (pid 1390) 正在运行...
[root@jiang root]# service sshd start
启动 sshd[ 确定 ]
[root@jiang root]# service sshd restart
停止 sshd: [ 确定 ]
启动 sshd[ 确定 ]
```

### 2. 直接执行服务脚本

执行的语法格式如下:

`/etc/rc.d/init.d/服务名 start | stop | restart | status`

例: 首先查看 `sshd` 服务的状态, 然后停止该服务, 再启动该服务, 查看运行状态, 最后重新运行该服务。命令序列如下:

```
[root@jiang root]# /etc/rc.d/init.d/sshd status
[root@jiang root]# /etc/rc.d/init.d/sshd stop
[root@jiang root]# /etc/rc.d/init.d/sshd status
[root@jiang root]# /etc/rc.d/init.d/sshd start
[root@jiang root]# /etc/rc.d/init.d/sshd restart
```

注意: 服务名和服务脚本文件名都是相同的。

## 14.6.4 创建特定级别的新服务

如果用户使用已经打包的 RPM 形式的服务软件包, 通常会提供 `/etc/rc.d/init.d` 中需要的服务运行脚本。但有时用户需要给没有服务运行脚本的重要软件添加启动脚本。在这种情况下, 需要为服务创建一个服务脚本。

首先查看 `/etc/rc.d/init.d` 目录中已有的 `httpd` 服务的脚本, 该脚本用于开启和终止 Apache Web 服务器。命令如下:

```
[root@jiang tools]# cat /etc/rc.d/init.d/httpd
```

结果为:

```
#!/bin/bash
#
# Startup script for the Apache Web Server
#
. /etc/rc.d/init.d/functions
if [ -f /etc/sysconfig/httpd ]; then
    . /etc/sysconfig/httpd
fi
INITLOG_ARGS=""
apachectl=/usr/sbin/apachectl
httpd=${HTTPD-/usr/sbin/httpd}
prog=httpd
RETVAL=0
# check for 1.3 configuration
check13 () {
    CONFFILE=/etc/httpd/conf/httpd.conf
    GONE="(ServerType|BindAddress|Port|AddModule|ClearModuleList|"
    GONE="${GONE}AgentLog|RefererLog|RefererIgnore|FancyIndexing|"
    GONE="${GONE}AccessConfig|ResourceConfig)"
    if grep -Eiq "^([[:space:]])*($GONE)" $CONFFILE; then
        echo
        echo 1>&2 " Apache 1.3 configuration directives found"
        echo 1>&2 " please read /usr/share/doc/httpd-2.0.40/migration.html"
        failure "Apache 1.3 config directives test"
        echo
        exit 1
    fi
}
start() {
    echo -n $"Starting $prog: "
    check13 || exit 1
    daemon $httpd $OPTIONS
    RETVAL=$?
    echo
    [ $RETVAL = 0 ] && touch /var/lock/subsys/httpd
    return $RETVAL
}
stop() {
```

```
    echo -n $"Stopping $prog: "
    killproc $httpd
    RETVAL=$?
    echo
    [ $RETVAL = 0 ] && rm -f /var/lock/subsys/httpd /var/run/httpd.pid
}
reload() {
    echo -n $"Reloading $prog: "
    check13 || exit 1
    killproc $httpd -HUP
    RETVAL=$?
    echo
}
# See how we were called.
case "$1" in
    start)
        start
        ;;
    stop)
        stop
        ;;
    status)
        status $httpd
        RETVAL=$?
        ;;
    restart)
        stop
        start
        ;;
    condrestart)
        if [ -f /var/run/httpd.pid ] ; then
            stop
            start
        fi
        ;;
    reload)
        reload
        ;;
    graceful|help|configtest|fullstatus)
```

```

$apachectl $@
RETVAL=$?
;;
*)
    echo                                "$Usage:                                $prog
{start|stop|restart|condrestart|reload|status|fullstatus|graceful|help|confi
gtest}"
    exit 1
esac
exit $RETVAL

```

这一个脚本实际上是一个简单的 **case** 语句，可以利用这个脚本来创建新的服务脚本。假定新服务为 **someservice**，可执行文件位于 **/usr/local/someservice**。

下面是创建 **/etc/rc.d/init.d/someservice** 脚本的步骤：

(1) 在 **/etc/rc.d/init.d/** 目录中拷贝一个现有的脚本（例如 **/etc/rc.d/init.d/httpd**），新文件取名为 **/usr/local/someservice**。

(2) 使用编辑器（如 **vi**）将文件中所有的 **httpd** 替换为 **someservice**。

(3) 如果没有将可执行文件放入 **/sbin/**、**/usr/sbin** 或者 **/usr/bin** 中，那么就必须修改 **daemon someservice** 这一行，加入可执行文件的路径，此处应将 **daemon someservice** 改为 **daemon /usr/local/someservice/someservice**。

(4) 如果 **/usr/local/someservice/someservice** 没有创建 **PID** 文件（**/var/someservice.pid**），那么就必须将 **rm -f /var/lock/subsys/httpd /var/run/ someservice** 去掉。

(5) 执行 **chmod 750 /etc/rc.d/init.d/someservice** 命令修改文件许可，这样 **root** 账号就能读写并执行该脚本了。

这样就创建了 **someservice** 服务脚本。用户可以使用 **ntsysv** 或者 **chkconfig** 程序将该服务加入到特定的运行级别。

## 思考与练习

- (1) 在 Linux 中，如何永久更改主机名？
- (2) Linux 中 **hosts** 文件的作用是什么？
- (3) 文件 **inittab** 的作用是什么？
- (4) 如何管理 Linux 系统下的服务？

## 第 15 章 X Windows 系统的配置和使用

XFree86 是 Linux 及其他类似操作系统的 X Windows 窗口系统，内容包括 X 服务器、字体、一个 X 下的终端程序 (xterm)、一个简单的窗口管理器 (twm)、很多有用的小工具、显示输出驱动和包括键盘鼠标在内的许多输入模块。

本章首先介绍了安装和配置 XFree86 的方法，然后介绍了使用 XFree86 的方法。

### 15.1 安装和配置 XFree86

#### 15.1.1 安装 XFree86

Red Hat Linux 9 的安装光盘包含了 XFree86 所需的所有软件包。在安装 Red Hat Linux 时，用户可以安装并配置 XFree86。如果用户还没有安装 XFree86，则可以使用安装 CD-ROM 中的 RPM 包安装 XFree86。

用户转到 Red Hat Linux 安装 CD-ROM 的 RPMS 目录，然后运行如下命令，进行安装：

```
# rpm -ivh XFree86*.rpm
```

#### 15.1.2 配置 XFree86

在配置 XFree86 之前，需要首先了解如下的信息：

- 显存大小
- 显卡的厂商名
- 显卡的型号
- 显示器的最大分辨率
- 显示器刷新频率、制造商和型号

如果要确定系统中的显卡和声卡设备等硬件设备，使用 `lspci`：

```
[root@wangwang proc]# lspci
```

```
00:00.0 Host bridge: Intel Corp. 82815 815 Chipset Host Bridge and Memory
Contro
ller Hub (rev 02)
00:01.0 PCI bridge: Intel Corp. 82815 815 Chipset AGP Bridge (rev 02)
00:1e.0 PCI bridge: Intel Corp. 82801BA/CA/DB PCI Bridge (rev 11)
00:1f.0 ISA bridge: Intel Corp. 82801BA ISA Bridge (LPC) (rev 11)
00:1f.1 IDE interface: Intel Corp. 82801BA IDE U100 (rev 11)
00:1f.2 USB Controller: Intel Corp. 82801BA/BAM USB (Hub #1) (rev 11)
00:1f.3 SMBus: Intel Corp. 82801BA/BAM SMBus (rev 11)
00:1f.4 USB Controller: Intel Corp. 82801BA/BAM USB (Hub #2) (rev 11)
00:1f.5 Multimedia audio controller: Intel Corp. 82801BA/BAM AC'97 Audio (rev
```

11

)

01:00.0 VGA compatible controller: nVidia Corporation NV5 M64 [RIVA TNT2 Model 64

/Model 64 Pro] (rev 15)

02:03.0 Ethernet controller: Realtek Semiconductor Co., Ltd. RTL-8029(AS)

可见系统中的显卡为 nVidia 公司 TNT2 M64 型号的显卡, 通常 Linux 系统不能精确检测出显存的数量, 用户通常可以查看系统引导时的显存显示得到。

在确定了系统信息之后, 用户可以通过以下两种方法配置 XFree86:

- 在命令行上使用图形化配置工具 `redhat-config-xfree86`。
- 手工建立 XF86Config 文件。这是 XFree86 下建立可用 X Window 环境的主要配置文件。该文件位于 `/etc/X11` 目录中, 而且通常由一个符号链接指向 `/etc/X11/XF86Config`, 这个符号链接名为 `/usr/X11R6/lib/X11/XF86Config`。

### 15.1.3 使用 `redhat-config-xfree86` 配置 XFree86

利用 `redhat-config-xfree86` 工具, 用户能够很容易地建立 XF86Config 配置文件。

运行如下程序:

```
# /usr/bin/redhat-config-xfree86
```

它将会弹出如图 15-1 所示的“显示设置”对话框。在图 15-1 所示的“显示”选项卡中, 中间显示了显卡的型号, 下面的设置项包括“分辨率”下拉列表框和“色彩深度”列表框。在“分辨率”下拉列表框中, 显示了当前使用的分辨率, 图 15-1 中显示的 800×600, 在“色彩深度”下拉列表框中, 可以选择的选项包括“256 种颜色”(8 位颜色)、“上千颜色”(16 位颜色)和“上百万颜色”(24 位颜色)。图 15-1 中的当前设置为“上百万颜色”。用户在图 15-1 中单击“高级”标签, 可以切换到“高级”选项卡。



图 15-1 显示器设置界面



在图 15-2 的显示器高级设置中，包括“显示器”和“视频卡”（或者显卡）两大选项区域。在“显示器”选项区域中，显示了当前的显示器类型、水平分辨率、垂直分辨率和以及设置显示器的物理分辨率（DPI）。用户可以通过单击“配置”按钮，在弹出的“显示器设置”对话框中设置新的水平分辨率和垂直分辨率。通过单击“设置 DPI”按钮设置显示器的物理分辨率。

在“视频卡”选项区域中，显示了显卡的类型，显存的大小以及与该显卡对应的驱动程序。用户单击该节的“配置”按钮，在弹出的“视频卡”对话框中设置显示卡的类型、显示驱动程序，以及显存的大小。

**注意：**Linux 系统往往不能正确地检测出一些显示器的品牌、型号，水平和垂直刷新频率，如果用户确实知道显示器的特性，可以在此设置它的属性。例如对于较新的显示器，通常可以将水平和垂直刷新频率分别设置为 30.0-100.0 和 50.0-100.0。

由于 Linux 不能精确地确定显卡的显存大小，为了得到更好地显示效果，用户需要在此将显存更改成其真实值。

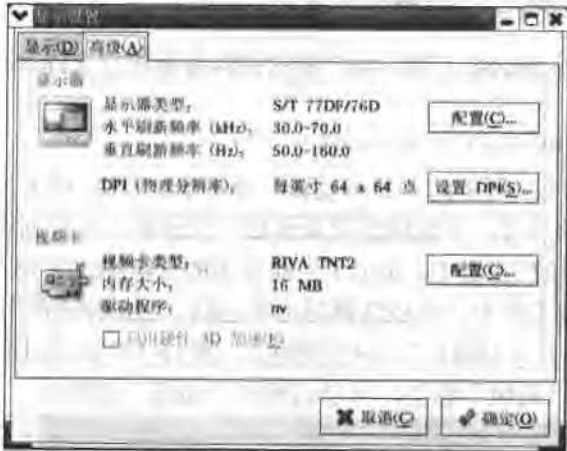


图 15-2 显示器的高级设置

除了使用 `redhat-config-xfree86` 配置显卡和显示器之外，以后还也可以使用如表 15-1 所示的配置工具对 XFree86 使用的其他设备进行配置。

表 15-1 XFree86 使用的其他设备的常见配置工具

配置工具	描述
<code>/usr/bin/redhat-config-keyboard</code>	图形方式键盘配置工具
<code>/usr/bin/redhat-config-language</code>	图形方式语言配置工具
<code>/usr/bin/redhat-config-mouse</code>	图形方式鼠标配置工具
<code>/usr/sbin/mouseconfig</code>	文本方式鼠标配置工具

除此之外，用户还可以使用 `redhat-config-soundcard` 配置声卡。

#### 15.1.4 手工修改 XFree86 配置文件

X Window 系统提供了许多配置工具，但最终都是修改了配置文件 XF86Config，该文件存放在/etc/X11 目录下。用户可以首先直接查看它，然后对它做一些小的调整，将会是一种更好的配置 XFree86 的方法。笔者在此将对这个配置文件进行简单的分析。它包括许多小节，每个小节分别配置其中的一部分。

注意：XF86Config 文件的每一小节都是以 Section “节类型名”表示小节的开始，以 EndSection 表示该小节的结束。

为了提高配置效率，用户可以首先使用配置工具进行配置，如果还有问题，再手工修改配置文件 XF86Config。

##### 1. 布局 (Layout) 配置小节

该节用于设定布局包含的输出设备（显示器）、输入设备（鼠标和键盘）和屏幕。实例如下：

```
Section "ServerLayout"
    Identifier      "Default Layout"
    Screen         0  "Screen0" 0 0
    InputDevice     "Mouse0" "CorePointer"
    InputDevice     "Keyboard0" "CoreKeyboard"
    InputDevice     "DevInputMice" "AlwaysCore"
EndSection
```

##### 2. 文件 (File) 配置小节

设置 X Window 系统所用的字体路径。实例如下：

```
Section "Files"
    RgbPath         "/usr/X11R6/lib/X11/rgb"
    FontPath        "unix/:7100"
EndSection
```

主要设置了 RgbPath（设置 RGB 色彩数据库的路径），以及 FontPath（用来设置字体的路径），Red Hat linux 采用字体服务器管理所有字体。用户通常不需要修改这两个选项。

##### 3. 模块节 (Module) 配置小节

该节主要包括了 XFree86 使用的模块。实例如下：

```
Section "Module"
    Load "dbe"
    Load "extmod"
    Load "fbdevhw"
    Load "glx"
    Load "record"
```

```

Load "freetype"
Load "type1"
Load "dri"
EndSection

```

#### 4. 输入设备 (InputDevice) 配置小节

**InputDevice** 主要用于配置键盘和鼠标。

对于输入设备键盘，可以设置各式键盘，在此可以指定键盘的传输协议、语系、信号、字符对照表，下面是一个配置实例：

```

Section "InputDevice"
    Identifier "Keyboard0"
    Driver      "keyboard"
    Option      "XkbRules" "xfree86"
    Option      "XkbModel" "pc105"
    Option      "XkbLayout" "us"
EndSection

```

**Identifier** 用于确定键盘的名称，**Driver** 指定键盘的驱动程序，**Option** 表示该设备的一些选项。这些选项分别用于设置键盘的规则（如 xfree86）、键盘型号（如 105 键的键盘），以及键盘的布局（美式键盘）。

对于输入设备鼠标，可以设置诸如鼠标、触摸屏等标准输入设备。下面是一个配置实例：

```

Section "InputDevice"
    Identifier "Mouse0"
    Driver      "mouse"
    Option      "Protocol" "IMPS/2"
    Option      "Device" "/dev/psaux"
    Option      "ZAxisMapping" "4 5"
    Option      "Emulate3Buttons" "no"
EndSection

```

它的设置项与设置键盘类似。其中重要的部分包括 **Driver**、**Protocol** 选项、**Device** 选项、**Emulate3Buttons** 选项。这需要根据具体的鼠标而定。

**Protocol** 选项用于设置鼠标使用的传输协议，其中常用的值为：

- **BusMouse**：总线型鼠标，早期的串口鼠标。
- **GlidePoint**：使用 ALPS 串口版本的 GlidePoint 触摸屏鼠标（用于笔记本）。
- **GlidePointPS/2**：使用 ALPS PS2 版本的 GlidePoint 触摸屏及其兼容产品。
- **IMPS/2**：使用微软 PS/2 版本的 IntelliMouse 及其兼容产品。
- **PS/2**：现在最广泛使用的 PS/2 接口的鼠标。

**Device** 选项用来设置连接鼠标的外围设备代码，通常都是 `/dev/psaux` 或者 `/dev/input/mice`。

Emulate3Button 选项将双键鼠标模拟成为三键鼠标。

### 5. 显示器 (Monitor) 配置小节

设置显示器的水平、垂直扫描频率, 同时定义各个显示模式与扫描频率之间的对应关系。X Server 会根据显示器扫描频率的设置, 选择最佳的显示模式。下面是一个配置实例:

```
Section "Monitor"
    Identifier      "Monitor0"
    VendorName      "Monitor Vendor"
    ModelName       "S/T 77DF/76D"
    DisplaySize     320      240
    HorizSync       30.0 - 70.0
    VertRefresh     50.0 - 160.0
    Option          "dpms"
EndSection
```

其中 Identifier、VendorName 和 ModelName 三个配置项用来标识不同的显示器。

HorizSync 设置水平扫描频率, 它的默认单位是 KHz, 可以使用形如 30~70 的方式来表示一个范围。具体的值应该根据显示器的使用说明上的数据来写。

VertRefresh 设置垂直扫描频率, 它的默认单位是 Hz, 表示方式与 HorizSync 一致, 建议根据显示器的使用说明上的数据填写。

### 6. 显卡配置小节

设置显卡的型号、驱动程序、供应商名称、显存。下面是一个配置实例:

```
Section "Device"
    Identifier      "Videocard0"
    Driver          "nv"
    VendorName      "Videocard vendor"
    BoardName       "RIVA TNT2"
    VideoRam        16384
EndSection
```

常用的设置项如下:

- Identifier。设定名称, 以便 Screen section 中可以辨认。
- VendorName。显示卡制造商的名称。
- BoardName。显示卡型号。
- VideoRam。显存的大小。

### 7. 屏幕 (Screen) 配置小节

设置将采用的 X Server、显卡及显示器之外, 还有显示模式、色彩深度、分辨率和虚拟桌面的设置。下面是一个配置实例:

```
Section "Screen"
```

```
Identifier "Screen0"
Device      "Videocard0"
Monitor     "Monitor0"
DefaultDepth 24
SubSection "Display"
    Depth    24
    Modes     "800x600" "640x480"
EndSubSection
EndSection
```

常用的设置项如下：

- **Identifier**。设定屏幕的名称。
- **Device**。使用的显卡的设备名。
- **Monitor**。使用的显示器设备名。
- **DefaultDepth**。默认的显示颜色深度，如 24 代表 24 位色。
- **SubSection** 的 **Depth** 表示颜色深度。
- **SubSection** 的 **Modes** 表示可选的显示分辨率，如本配置中表示 24 位色可以使用 800×600 和 640×480 两种分辨率。

## 8. DRI 配置小节三

该节设置 DRI (Direct Rendering Infrastructure)。实例如下：

```
Section "DRI"
    Group      0
    Mode       0666
EndSection
```

## 15.2 使用图形化桌面

Red Hat Linux 包括了一个功能强大的图形化桌面环境。其中最为常用的桌面环境包括 GNOME 和 KDE，它们都是基于 XFree86 (X 服务器)。这两种桌面都非常流行，用户可以根据自己的爱好选择其中一种桌面环境，例如笔者更喜欢 KDE。通过这些桌面环境，用户可以从中简易地进入你的应用程序、文件和系统资源，以便充分利用 Red Hat Linux 系统。

### 15.2.1 使用桌面

用户的初始图形化桌面看起来如图 15-3 所示。

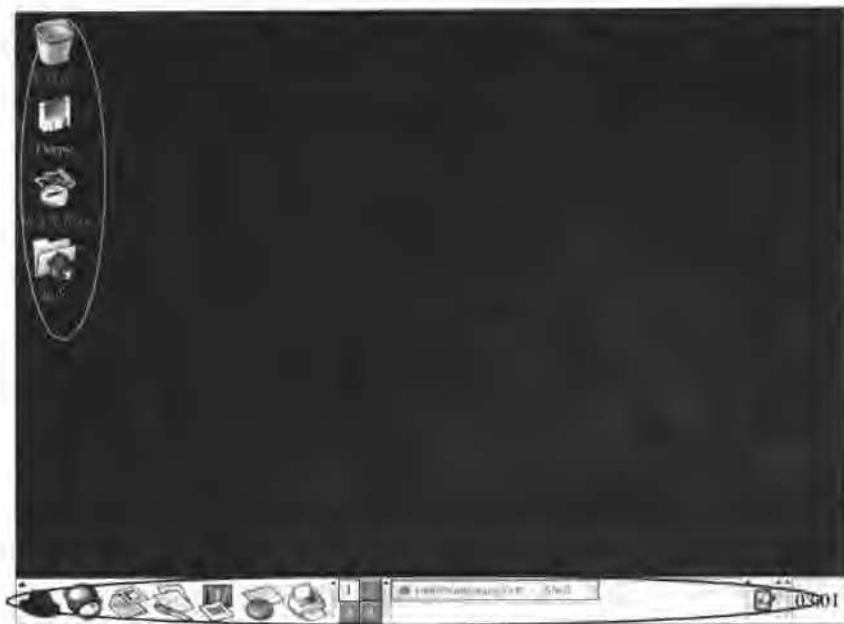


图 15-3 图形化桌面

图形化桌面环境使用户能够进入你的计算机上的应用程序和进行系统设置。它提供了 3 种主要工具来使用系统上的应用程序：面板图标、桌面图标、以及菜单系统。

横贯桌面底部的长条叫做面板 (panel)。面板上包含应用程序启动器、用于通知警告图标的通知区域、以及小程序 (applet, 小型应用程序)。它们允许用户控制音量、切换工作区、并显示系统状态。

桌面上其他地方的图标是文件夹、应用程序启动器、光盘、软盘之类的可移设备 (在它们被挂载后出现) 的快捷方式。要打开一个文件夹或启动一个应用程序, 双击相应的图标即可。

可以通过单击“主菜单”按钮来进入菜单系统, 或者单击桌面上的“从这里开始”图标, 然后选择“应用程序”图标来进入菜单系统。

桌面的工作方式和用户使用其他操作系统的桌面类似。用户可以:

- 把文件或程序的图标拖放到容易存取的地方。
- 可以在桌面、面板和文件管理器中为文件和程序添加新图标。
- 可以改变工具和应用程序的外观。
- 可以使用配置工具来改变系统的设置。

### 15.2.2 使用面板

桌面面板是屏幕底部的长条, 如图 15-4 所示。它包含了便于用户使用系统的图标和小型程序。面板上还包含“主菜单”, 其中包含到所有应用程序的菜单项。嵌入在面板中的小程序允许你运行指定任务或者监控系统的状态, 通知区域中放置一些通知警告图标 (如 Red Hat 网络图标)。因此, 用户可以通过这样的图标快速得到重要消息警告。



图 15-4 XWindows 桌面面板

用户可以单击面板上的“主菜单”按钮将主菜单展开成一棵菜单树。该集合允许你进入系统内的应用程序。

用户可以从这里启动 Red Hat Linux 中大多数应用程序。

**注意：**Red Hat Linux 菜单中的子菜单包含大量应用程序。从“主菜单”中，用户还可以注销，从命令行运行应用程序、寻找文件，或锁住屏幕（这会运行用口令保护的屏幕保护程序）。

### 15.2.3 使用小程序

小程序是运行在面板上的小型应用程序。小程序通常会让你监控你的系统上的各类情况。某些小程序执行有用的任务，而另一些则纯粹为娱乐而设计的。

Red Hat Linux 面板上默认运行几个小程序。它们分别是：

- 工作区切换器。图形化桌面提供了使用多个工作区的能力，如图 15-5 所示。因此用户不必把所有运行着的应用程序都堆积在一个可视桌面区域。工作区切换器把每个工作区（或桌面）都显示为一个小方块，然后上面显示运行着的应用程序。使用鼠标单击任何一个小方块来切换到那个桌面上去。用户还可以使用键盘快捷方式 Ctrl+Alt+向上方向键、Ctrl+Alt+向下方向键、Ctrl+Alt+向右方向键或 Ctrl+Alt+向左方向键切换桌面。



图 15-5 工作区切换器

- 任务条。工作区切换器旁边的小程序是任务条，如图 15-6 所示。任务条显示任意虚拟桌面上运行的应用程序名称的小程序。这对应用程序最小化时很有用，因为用户可以单击它来恢复应用程序画面。



图 15-6 任务条

### 15.2.4 使用通知区域

#### 1. Red Hat 网络更新通知工具

Red Hat 网络更新通知工具是通知区域的一部分，如图 15-7 所示。它为用户提供了一

种简捷的系统更新方式，确保系统时刻使用 Red Hat Linux 的最新更新。如果单击该图标，将会出现一个可用更新列表。单击该按钮可以启动 Red Hat 更新代理，进行系统更新。



图 15-7 Red Hat 网络更新通知工具

## 2. 验证图标

通知区域有时会显示一个钥匙图标，如图 15-8 所示。这是一个安全通知，当需要取得系统的根权限验证时，它将就会出现该图标。



图 15-8 验证图标

## 3. 打印机通知警告图标

打印机通知警告图标允许用户管理打印作业。单击这个图标来查看正在运行的打印作业，可以右击作业，并选择“取消”按钮来取消作业。

**注意：**如果看不到任何通知警告图标，表明通知区域可能已被删除。要重新加入面板，右击面板，并选择“添加到面板”|“工具”|“通知区域”命令。

### 15.2.5 在面板上添加图标和小程序

用户可以定制面板。如果要在面板上添加小程序，右击面板上的未用区域，选择“添加到面板”，然后从“附件”菜单中选择小程序。选定了小程序后，它将会出现在面板上。

要在面板上添加一个启动器，右击面板上的未用区域，选择“添加到面板”|“启动器...”。这会启动一个对话框。用户可在该对话框中输入应用程序的名称、位置和启动它的命令（如 `/usr/usr/local/arcoread5/arcoread`），甚至为这个应用程序选择一个图标。然后单击“确定”按钮，这个新启动器图标就会出现在面板上。

**注意：**另一种在面板上添加启动器的快捷方式是：右击面板上的未用区域，选择“添加到面板”|“从菜单启动”。然后，选择一个出现在菜单中的应用程序。这会按照该程序在“主菜单”中的属性来添加启动器图标。

### 15.2.6 配置桌面面板

可以自动或手工隐藏面板、把它放置在桌面上的任一边上、改变它的大小和颜色或者改变它的行为方式。要改变默认的面板设置，右击面板上的未用区域，选择“属性”，然后设置面板的大小、在桌面上的位置，以及是否想在不用面板时自动隐藏它（“自动隐藏”）。如果用户选择了自动隐藏面板，那么当鼠标从面板上移开时，它就不会出现在桌面上。



## 思考与练习

- (1) X Windows 的作用是什么？
- (2) 如何手工更改 XFree86 的配置文件？
- (3) Linux 图形化桌面有哪些元素组成？这些元素的功能是什么？

## 第 16 章 安装和使用常用应用软件

对于桌面用户来说,通常需要办公、播放多媒体(包括各种音乐、视频文件)、进行即时通信和上网等多种活动。Red Hat Linux 发行版本提供了提供了较多的应用软件,而且用户可以从互联网上得到许多功能很强的软件。这对进行桌面应用也是非常方便的。

本章先后介绍了安装和使用 Acrobat Reader 的 PDF 格式阅读器、多媒体播放器 MPlayer, Linux 下的即时消息工具 gaim、Flash 动画播放器、新中文字体和 NTFS 驱动的方法。

### 16.1 安装和使用 PDF 阅读器

PDF 文件格式是 Adobe 公司开发的电子读物文件格式。PDF 的优点在于这种格式的电子读物美观、便于浏览、安全性高。但是这种格式不支持 CSS、Flash、Java、JavaScript 等基于 HTML 的技术,所以它只适合于浏览静态的电子图书和系统帮助文档。如果用户希望看到具有动态效果电子图书,那么 PDF 格式还不是最好的选择。

#### 16.1.1 Linux 提供的 PDF 阅读器

PDF 格式的电子读物需要支持 PDF 文件格式的阅读器来阅读。这要求在读者的计算机中安装这种阅读器。Red Hat Linux 9 发行版已经提供了两个 PDF 文件查看器,其中之一是 KGhostView (“PS / PDF 查看器”),另一个是 xpdf (PDF Viewer)。这两种阅读器都是免费的,KGhostView 默认支持中文,但是它并没有提供复制文件中的本文或者图像的能力。xpdf 提供了即不提供复制文本的能力,默认也不支持中文字符集。因此,这两种 PDF 阅读都不太适合于阅读简体中文 PDF 文件。

用户可以通过“主菜单”|“图形”|PDF Viewer 命令和“主菜单”|“图形”|“更多图形程序”|“PS / PDF 查看器”命令分别启动 xpdf 和 KGhostView。

Adobe 公司提供了免费的 PDF 文件阅读器 Adobe Acrobat Reader,它的功能比 Red Hat Linux 9 提供的这两种阅读器的功能都要强大,默认支持英语,但是通过安装提供的简体中文字体补丁,可以较好地支持简体中文,而且能方便地进行文本和图形方式的复制功能。用户可以从 Adobe 公司下载 Linux 版本的 Acrobat Reader,以及简体中文字体补丁。

#### 16.1.2 安装 Acrobat Reader

在 Linux 中可以使用的 Acrobat Reader 的最高版本为 5.0.8,它需要 25MB 的磁盘空间。使用下列步骤下载和安装 Acrobat Reader 5.0.8:

(1) 从 <http://www.adobe.com/products/acrobat> 网页下载 Acrobat Reader 5.0.8 安装程序和简体中文字体补丁。它们分别为:

- `acrobat-linux-508.tar.gz`
- `chsfont.tar.gz`

(2) 以 root 用户身份执行安装程序。

解压缩安装程序 `acrobat-linux-508.tar.gz`，命令如下：

```
tar xvzf acrobat-linux-508.tar.gz
```

然后，执行它的安装程序：

```
./INSTALL
```

安装程序首先显示许可证信息。用户可以多次按空格键，滚动到许可证的结束处，安装程序提示：如果接受许可证协议，就输入 `accept`，否则输入 `decline`，然后按回车键确认。信息如下：

```
To accept the terms and conditions of this agreement enter "accept".
```

```
To decline the terms and conditions of this agreement enter "decline".
```

```
Please type "accept" to accept the terms and conditions license agreement;
Type "decline" to exit.
```

然后提示用户指定 Acrobat 的安装目录。默认目录为 `/usr/local/Acrobat5`，如果接受默认目录，按回车键继续，否则输入用户自己设定的安装目录，然后再按回车键确认。

```
Enter installation directory for Acrobat 5.0.8 [/usr/local/Acrobat5]
```

本例安装使用默认安装命令，然后进行真正的安装过程，直到安装结束，安装结束后返回 shell 提示符。

(3) 安装简体中文字体。

解压缩中文字体文件，目录如下：

```
tar xvzf chsfont.tar.gz
```

转到 CHSKIT 目录，并执行安装简体中文补丁，命令如下：

```
cd CHSKIT/
```

```
./INSTLANG
```

该安装包可以在 Acrobat Reader 5.0 及其更高版本上安装。在出现 `Continue installation?` [y] 提问后面，输入 `y` 继续安装。与安装 Acrobat Reader 类似，需要用户同意许可证。输入 `accpet`，并按回车键继续。

接下来，它需要用户指定已安装的 Acrobat Reader 的位置。默认值为 `/usr/local/Acrobat5`，该默认值就是本例中 Acrobat Reader 的安装位置。按回车键后，真正进入安装过程，当安装完成后，返回 shell 提示符。

这样中文简体字体补丁就安装完成了。

### 16.1.3 使用 Acrobat Reader

为了启动 Acrobat Reader，在 shell 提示符中输入如下命令：

```
/usr/local/Acrobat5/bin/acroread
```

Acrobat Reader 启动后，单击 `File | open` 菜单项，在弹出的对话框中输入 PDF 的文件，然后打开 PDF 文件，打开的简体中文 pdf 文件实例如图 16-1 所示。

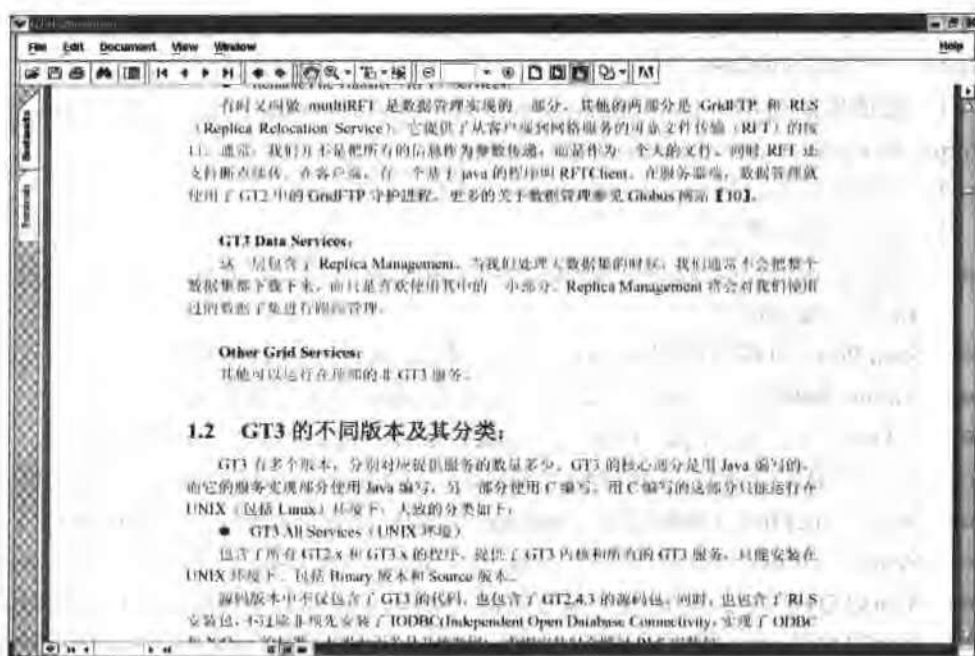


图 16-1 Acrobat Reader 的界面

## 16.2 安装和使用多媒体播放器 MPlayer

Red Hat Linux 本身提供了 MMS 媒体播放器, MMS 本身的功能非常强大, 但是免费提供的解码器却比较少, 因此它不能满足大多数桌面用户的需要。

MPlayer 是一种完全开放源码的多功能 Linux 媒体播放器, 它是 linux 下播放速度最快, 支持媒体格式最多的播放器, 它几乎能播放所有的 Windows 媒体文件, 它本身可以运行在文本模式下, 也可以运行在图形 (GUI) 模式下, 它能使用众多的本地的 XAnim, RealPlayer 和 Win32 DLL 的编解码器, 能播放 MPEG, VOB, AVI, OGG, VIVO, ASF/WMV, QT/MOV, FLI, RM, NuppelVideo, yuv4mpeg, FILM, RoQ 文件。用户还能观看 VideoCD, SVCD, DVD, 3ivx, RealMedia, 和 DivX 格式的电影。MPlayer 的另一个大的特色是广泛的输出设备支持。它可以在 X11, Xv, DGA, OpenGL, SVGAlib, fbdev, AAlib, DirectFB 下工作, 而且也支持 GGI 和 SDL, 由此可以使用它们支持的各种驱动模式和一些低级的硬件相关的驱动模式 (比如 Matrox, 3Dfx 和 Radeon, Mach64, Permedia3), 能在全屏下观赏电影。MPlayer 还支持通过硬件 MPEG 解码卡显示, 诸如 DVB 和 DXR3 与 Hollywood+。可以使用 European/ISO 8859-1,2 (匈牙利语、英语、捷克语等等), 西里尔语, 韩语的字体清晰放大, 并且反锯齿的字幕 (支持 10 种格式) 和 OSD (on screen display)。

### 16.2.1 下载 MPlayer

用户可以从 Mplayer 网站 (<http://www.mplayerhq.hu>) 下载 MPlayer 的安装程序和解码

器。用户可以下载和安装源代码版本，也可以下载和安装 RPM 安装包版本。

本节将采用 RPM 安装包安装 MPlayer V1.0 pre 2。这些 RPM 包包括播放器、GUI、字体、skin、一些驱动和编解码器。

(1) 播放器的通用包包括 `mplayer-common-1.0pre2-1.i386.rpm`，`mplayer-gui-1.0pre2-1.i386.rpm` 和 `mplayer-skin-default-1.0-2.noarch.rpm`。

(2) 字体文件包包括 `mplayer-font-iso1-1.1-1.noarch.rpm`。

(3) 常用的解码器包括

- 关键包 `mplayer-codecs-extralite-1.0-1.i386.rpm`（它包括 DMO、Indeo、QT、Real 和 3ivX 编解码器）
- Real Player 9 编解码器 `mplayer-codecs-linux-real-9.0-1.i386.rpm`
- Xanim 编解码器（`mplayer-codecs-linux-xanim-1.1-1.i386.rpm`）
- XAnim 3ivX 编解码器（`mplayer-codecs-linux-xanim-3ivx-1.1-1.i386.rpm`）
- Win32 DMO (WMV9/WMA9) 编解码器（`mplayer-codecs-win32-1.1-1.i386.rpm`）
- Win32 Intel Indeo 编解码器（`mplayer-codecs-win32-indeo-1.1-1.i386.rpm`）
- Win32 MJPEG2000 编解码器（`mplayer-codecs-win32-mjpeg2k-1.0-1.i386.rpm`）
- Win32 QT6 (QuickTime6) 编解码器（`mplayer-codecs-win32-qt-6.0-1.i386.rpm`）
- Win32 QTExtras 编解码器包（`mplayer-codecs-win32-qt-extras-1.1-1.i386.rpm`）

(4) VIDIX 支持包。

它包括两个包：`mplayer-vidix-nvidia-1.0pre2-1.i386.rpm` 和 `mplayer-vidix-1.0pre2-1.i386.rpm`。

### 16.2.2 安装 MPlayer

(1) 安装通用包，命令如下：

```
rpm -Uvh mplayer-common-1.0pre2-1.i386.rpm mplayer-gui-1.0pre2-1.i386.rpm
mplayer-skin-default-1.0-2.noarch.rpm
```

(2) 安装 ios-8859-1 字体包，命令如下：

```
rpm -Uvh mplayer-font-iso1-1.1-1.noarch.rpm
```

(3) 安装必要的编解码器，具体如下：

- 安装关键包：

```
rpm -Uvh mplayer-codecs-extralite-1.0-1.i386.rpm
```

- 安装 Linux RealPlayer 9 编解码器

```
rpm -Uvh mplayer-codecs-linux-real-9.0-1.i386.rpm
```

- 安装 Linux XAnim 编解码器

```
rpm -Uvh mplayer-codecs-linux-xanim-1.1-1.i386.rpm
```

- 安装 Linux XAnim 3ivX 编解码器

```
rpm -Uvh mplayer-codecs-linux-xanim-3ivx-1.1-1.i386.rpm
```

- 安装 Win32 DMO (WMV9/WMA9)编解码器

```
rpm -Uvh mplayer-codecs-win32-1.1-1.i386.rpm
```

- 安装 Win32 Intel Indeo 编解码器

```
rpm -Uvh mplayer-codecs-win32-indeo-1.1-1.i386.rpm
```

- 安装 Win32 MJPEG2000 编解码器

```
rpm -Uvh mplayer-codecs-win32-mjpeg2k-1.0-1.i386.rpm
```

- 安装 Win32 QT6 编解码器

```
rpm -Uvh mplayer-codecs-win32-qt-6.0-1.i386.rpm
```

- 安装 Win32 QTExtras 编解码器

```
rpm -Uvh mplayer-codecs-win32-qt-extras-1.1-1.i386.rpm
```

#### (4) 安装 VIDIX 支持。

```
rpm -Uvh mplayer-vidix-nvidia-1.0pre2-1.i386.rpm mplayer-vidix-1.0pre2-1.i386.rpm
```

### 16.2.3 使用 MPlayer

有两种使用 MPlayer 的方法，分别是文本模式和图形模式。

#### (1) 文本模式启动，命令如下：

```
mplayer /路径/要播放的媒体文件名
```

#### (2) 图形方式启动 MPlayer，命令如下：

```
gmplayer
```

在标准播放模式，一般按 F 键可以切换到全屏模式，再按 F 键复原，按 Q 键可以退出；播放时如果发现声音和画面不同步，可以按+号（shift+=）或-号调整；播放 divx 的 avi 文件时，如果想支持中文字幕，则在电影文件目录下必须要有和电影同名的 idx 和 sub 这两种格式的字幕文件存在；支持网络播放流式文件，譬如输入 mplayer http://IPAddress/45.asf 可以在线播放 45.asf。

MPlayer 的播放控制界面（默认皮肤）如图 16-2 所示。



图 16-2 MPlayer 的界面

常用的控制如下所示。

- : “文件打开”按钮；
- : “关闭播放器”按钮；
- : “播放”按钮；
- : “停止播放”按钮；
- : 创建播放列表按钮；
- : 音量调节条。

“播放列表生成”对话框如图 16-3 所示。它包含了目录树（Directory Tree）、选中的

目录中的文件（Files）和选中的文件（Selected Files）。用户可以在 Files 列表中的文件通过 Add（添加）按钮添加到 Selected Files 列表中。然后单击 OK（确定）按钮确定。



图 16-3 “播放列表生成”对话框

## 16.3 安装和使用 Gaim

Linux 默认提供了 Gaim 和 Licq 两个即时通信工具。Licq 专用于 ICQ 的即时通信，而 Gaim 是一个通用的即时通信工具，它通过插件的方式提供特定的即时通信协议。Gaim 默认提供了 ICQ、Yahoo 等多种插件，可以方便地进行即时消息服务。然而 Red Hat Linux 9 提供的 Gaim 0.59.8 并不能正常支持 OICQ 和 MSN。现在已经可以得到 OICQ 的 Gaim 插件。

本节将介绍下载、安装和使用带 OICQ 插件的 Gaim。

### 16.3.1 下载 Gaim

从 <http://download.linuxsir.org/gaim/gaim-0.76cvs-qq-0.2-final-src.tbz> 下载带 OICQ 插件的 Gaim 源代码。该软件需要事先安装 GTK2.0。

注意：用户可以执行 `redhat-install-packages` 命令安装“GNome 开发”组件，以实现安装 GTK2.0 的目的。它需要 Red Hat Linux 9 的第二张安装光盘。

### 16.3.2 安装 Gaim

以下是 Gaim 0.76 的安装步骤：

（1）解压缩 Gaim 源代码。命令如下：

```
tar xvjf gaim-0.76cvs-qq-0.2-final-src.tbz
```

（2）转至源代码目录。命令如下：

```
cd gaim-0.76cvs-qq-0.2-final-src
```

（3）生产合适的 Makefile。命令如下：

```
./autogen.sh
```

```
./configure
```

(4) 建立适当的符号链接。命令如下:

```
sudo ln -s /usr/lib/libspell-modules.so.1 /usr/lib/libspell-modules.so
sudo ln -s /usr/lib/libspell.so.4 /usr/lib/libspell.so
sudo ln -s /usr/share/pixmaps /usr/local/share/pixmaps
```

(5) 编译源代码。

```
make
```

(6) 安装目标文件。命令如下:

```
make install
```

### 16.3.3 使用 Gaim

当 Gaim 安装成功后, 只需要在命令行输入 `gaim` 即可启动 Gaim。它的登录界面如图 16-4 所示。

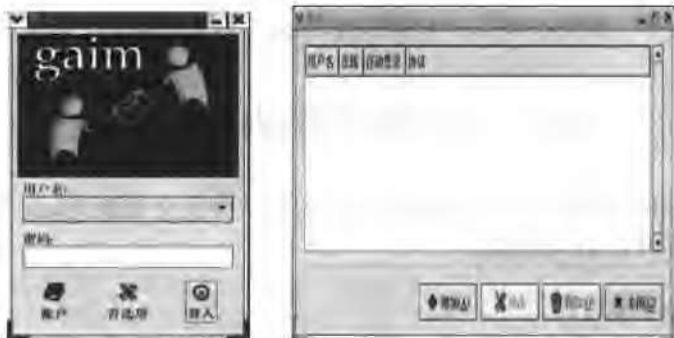


图 16-4 Gaim 的登录界面

在“账户”对话框中单击 Add (添加) 按钮, 弹出如图 16-5 所示的“添加账户”对话框。在该对话框中选择“协议”, 然后输入“用户名”和“密码”, 再输入“别名”。如果希望 Gaim 在下次启动时记住密码, 选中“记住密码”复选框。选中“自动登录”复选框可以使下次 Gaim 启动时自动登录。设置完成后, 单击 Save (保存) 按钮, 保存设置。



图 16-5 在 Gaim 中添加账户



用户登录后的好友列表和聊天窗口分别如图 16-6 所示。

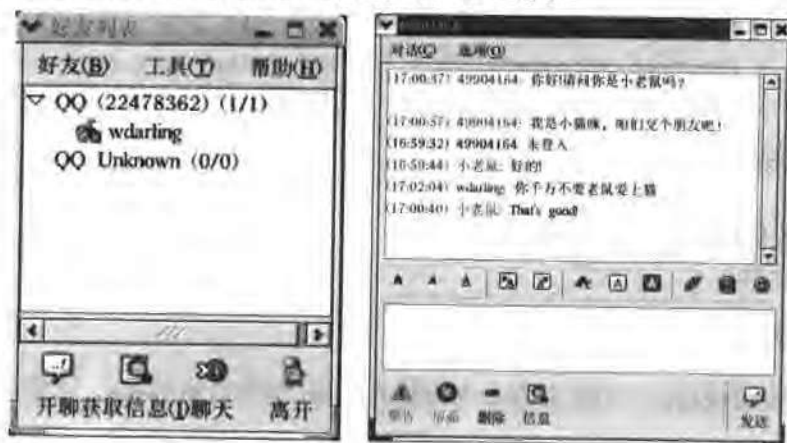


图 16-6 Gaim 中 OICQ 聊天的好友列表和聊天窗口

## 16.4 安装和使用 Flash 播放器

Linux 的 Mozilla 浏览器支持 Flash 动画，但默认并没有安装 Flash 插件。下面介绍如何下载、安装和使用 Flash 播放器。

### 16.4.1 下载 Flash 播放器

到 <http://www.macromedia.com/shockwave/download> 下载 Linux 下的 Flash 播放器，文件名为 `install_flash_player_7_linux.tar.gz`。

### 16.4.2 安装 Flash 播放器

(1) 解压缩。

```
$ tar xvzf install_flash_player_6_linux.tar.gz
```

(2) 转至 `install_flash_player_6_linux` 目录。

```
$ cd install_flash_player_6_linux
```

(3) 安装 Flash 播放器。

```
# ./flashplayer-installer
```

连续按 3 次回车键后选择安装 flash 插件 (mozilla、netscape 或者 opera) 的目录，Red Hat Linux 9 中为 mozilla 浏览器安装插件的目录是 `/usr/lib/mozilla-1.2.1`。

单击“帮助”|“关于 Mozilla”命令，弹出的 Web 页面显示了如图 16-7 所示的内容，它表示 Flash 播放器已经被成功安装。

Shockwave Flash

File name: libflashplayer.so  
Shockwave Flash 6.0 r79

MIME Type	Description	Suffixes	Enabled
application/x-shockwave-flash	Shockwave Flash	swf	Yes
application/futuresplash	FutureSplash Player	spl	Yes

图 16-7 验证已经安装 Flash 播放器

### 16.4.3 使用 Flash 播放器

使用了 Flash 播放器之后, 网页中嵌入的 Flash 动画将会播放出来。例如在 Mozilla 浏览器的地址栏中输入网址 <http://www.sohu.com>, 按回车键, 得到如图 16-8 所示的页面。可以看到, 里面的 Flash 动画被播放出来了。



图 16-8 Flash 动画播放效果

## 16.5 安装和使用新的中文字体

Red Hat Linux 9 默认提供了一些简体中文字体, 但是这些字体与 Windows 系统的字体不完全兼容, 因此出现使用 OpenOffice 的字处理程序 Writer。打开微软 Word 文档时, 汉字不能正确地显示, 用户可以有两种方法解决这个问题:

(1) 使用 Ctrl+A 键全部选中文字, 然后在字体列表中选择 AR PL SungtiL GB 或者 AR PL KaitiL GB, 使 Word 文档以 Linux 中的中文字体显示。这种方式改变了原文件的格式。

(2) 加入 Windows 支持的字体, 这样在 Linux 就可以编辑和查看与 Windows 完全相同的 Word 文档。

添加的新字体可以被 GNOME、KDE、Mozilla 浏览器以及多种其他的程序使用。

如果想给 Red Hat Linux 9 系统添加新字体，根据使用新字体的字体子系统而定，Red Hat Linux 9 中支持两种字体子系统，它们分别是：核心 X 字体子系统和 fontconfig 字体子系统。下面以添加 simsun（Windows 的宋体）为例，说明如何在 Red Hat Linux 中为核心 X 字体子系统添加字体（TrueType 字体或者非 TrueType 字体）。

### 16.5.1 准备要安装的字体

互联网上提供了一些免费的集体中文字体，在此笔者以 Windows 系统中的 simsun.ttc 为例（其中 simsun.ttc 位于 Windows 2003 系统的字体目录中，C:\WINDOWS\Fonts 确良 \simsun.ttc，对于 Windows 的其他版本，其文件名可能是 simsun.ttf）。

（1）创建 /usr/share/fonts/local/ 目录。

```
# mkdir /usr/share/fonts/local/
```

（2）将 simsun.ttc 文件复制到 /usr/share/fonts/local/ 目录。

在此假定 Windows 2003 中的系统盘在 Linux 中对应的分区设备名为 /dev/hda1，而且 /mnt/2003 目录不存在。复制字体文件的命令如下：

```
# mkdir /mnt/2003
```

```
# mount /dev/hda1 /mnt/2003
```

```
# cp /mnt/2003/ WINDOWS/Fonts/simsun.ttc /usr/share/fonts/local/
```

### 16.5.2 生成字体信息

如果要在 X 字体服务器中添加字体，则必须有字体的信息文件。用户使用以下命令来生成新字体信息文件：

```
# ttmkfdi -d /usr/share/fonts/local/ -o /usr/share/fonts/local/fonts.scale
```

然后把 fonts.scale 拷贝一份，命名为 fonts.dir。命令如下：

```
# cp /usr/share/fonts/local/fonts.scale /usr/share/fonts/local/fonts.dir
```

### 16.5.3 将字体目录加入字体服务器路径中

用户创建的目录 /usr/share/fonts/local/ 必须添加到 X 字体服务器的路径中，命令如下：

```
# chkfontpath --add /usr/share/fonts/local/
```

### 16.5.4 重新启动 xfs 字体服务器

为了使加入的文件生效，可以使用如下命令重新启动字体服务器：

```
# /sbin/service xfs reload
```

### 16.5.5 使用新添加的字体

在 OpenOffice 的 Writer 中，可以在“字体”下拉列表中选择 Simsun 字体，如图 16-9 所示。

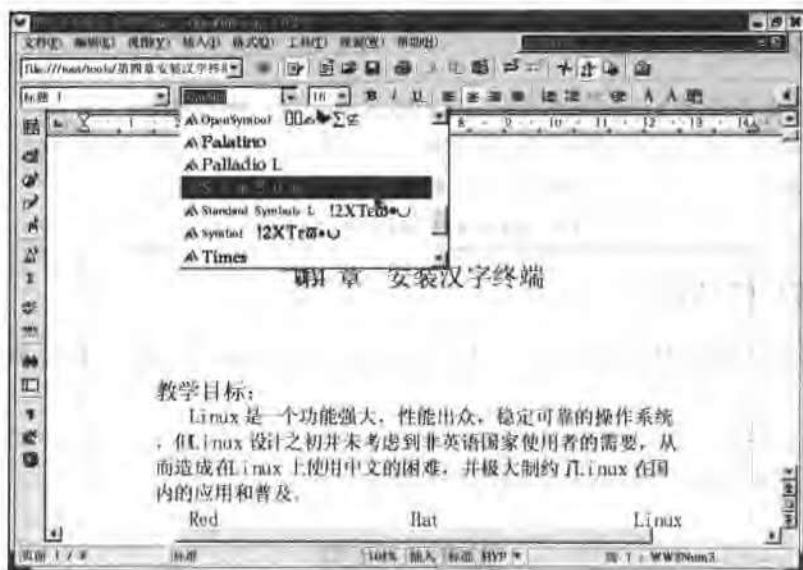


图 16-9 使用安装的宋体 (Simsun) 字体

## 16.6 支持 NTFS 文件系统

Red Hat Linux 9 的二进制内核版本中没有提供 NTFS 文件系统的驱动程序。用户可以通过重新编译 Linux 内核源代码的方式，加入 NTFS 文件系统支持。更简单的方式是直接在网上下载 NTFS 驱动程序的 RPM 包，进行安装。

### 16.6.1 下载 NTFS 的 RPM 包

用户可以从 <http://linux-ntfs.sourceforge.net> (Linux ntfs Project) 网站上下载 Red Hat Linux 9 系统下的 NTFS 文件系统驱动安装包，该网站提供了各个 Linux 操作系统和硬件平台的 RPM 包。用户应该根据自己的需要下载一个 RPM 包。

检查 Linux 的内核版本，命令如下：

```
[root@jiang install_flash_player_6_linux]# uname -a
```

```
Linux jiang 2.4.20-8 #1 Thu Mar 13 17:18:24 EST 2003 i686 athlon i386 GNU/Linux
```

结果表明内核的版本为 2.4.20-8。

检查 CPU 构架，命令如下：

```
[root@jiang install_flash_player_6_linux]# cat /proc/cpuinfo | grep 'model name'
```

```
model name      : AMD Athlon(tm) XP
```

结果表明 CPU 为 AMD Athlon。

根据网页的表可知，应该下载文件 kernel-ntfs-2.4.20-8.athlon.rpm。

### 16.6.2 安装 NTFS 的 RPM 包

使用 rpm 命令安装 kernel-ntfs-2.4.20-8.athlon.rpm 包，命令如下：

```
[root@jiang tools]# rpm -ivh kernel-ntfs-2.4.20-8.athlon.rpm
Preparing... ##### [100%]
 1:kernel-ntfs ##### [100%]
```

### 16.6.3 挂载 NTFS 的分区

使用如下命令即可挂载 ntfs 分区，假设 ntfs 分区的设备名为/dev/hda1，挂载点目录为/mnt/ntfs：

```
mount -t ntfs /dev/hda1 /mnt/ntfs
```

在这里，/dev/hda1 分区的分区类型为 NTFS。

## 思考与练习

- (1) 如何安装和使用 PDF 阅读器？
- (2) 如何在 Linux 下播放各种多媒体？
- (3) 如何在 Linux 下聊天？
- (4) 如何在 Linux 下添加字体？

## 第 17 章 OpenOffice 套件

Red Hat Linux 9 包括多个管理文档的工具。无论用户要筹备商务或学业的演示文稿，还是要编写正式公函，或者从电子邮件附件中打开文档，Red Hat Linux 9 都为用户提供了合适的文档。

Red Hat Linux 9 包括了一个功能强大的商业套件 OpenOffice.org（生产率套件）。它把几个互补的应用程序综合成一个软件包集合。使用 OpenOffice.org 要比学习复杂的文档格式化标签和编码要快得多，也容易得多。它允许用户完全控制文档的内容和布局，还可以让用户一边编辑一边查看效果。这种实时、可视形式的文档格式化叫做所见即所得（what you see is what you get, WYSIWYG）编辑。

OpenOffice.org 与微软公司的 Office 极其类似，而且 OpenOffice 兼容微软的 Office。这给 Windows 客户带来福音，因为它使得在 Linux 编辑的文档直接可以在 Windows 系统中使用，反之亦然。

本章首先介绍了 OpenOffice 办公套件的功能，然后分别介绍了 OpenOffice 套件的字处理程序 Writer、电子表格处理程序 Calc、演示文稿创建程序 Impress、图形处理程序 Draw 的使用方法。

### 17.1 OpenOffice.org 办公套件的功能

OpenOffice.org（生产率套件，Productivity suite，以后都简称为 OpenOffice）是一个应用程序集合。它的目的是节省时间，并为用户在工作、学校和家庭使用提供协助。通常，生产率套件是图形化的，包括的应用程序有：文字处理器、电子表格、以及文稿演示工具。这些组成生产率套件的应用程序是集成（integrated）在一起的，这意味着用户在编写的文档中可以嵌入由电子表格程序创建的图表、图形化演示程序创建的幻灯图片。把软件综合成一个生产率套件可以帮助用户提高演示文稿、演讲、或者印刷品制作的生产效率。

OpenOffice 套件包含多个用创建和编辑文档的应用程序、电子表格、商业演示文稿、艺术作品。它包括快速创建基本专业文档和演示文稿的模板、表格和向导。通常用户都使用过扩展名为.doc 或.xls 的文件，这些文件都是与 Microsoft 办公套件相关联的。OpenOffice 套件能够读取、编辑、和创建好几种格式的文件，包括与 Microsoft Office 套件相关联的文件。表 17-1 显示了用户可以在 OpenOffice 套件中使用的多种不同文件类型，以及可以用它来完成的任务。

从表 17-1 可知，OpenOffice 套件具有许多文件兼容性功能，并能够广泛用于学术、商业、或者家庭使用办公场合。

表 17-1 OpenOffice 支持的文档类型

文件格式 (扩展名)	OpenOffice 应用程序	用途
.sxw、.sdw、.doc、.rtf、 .txt、.htm/.html	Writer	正式公函、商业表格、学术论文、简历、新闻简报、报告
.sxc、.dbf、.xls、.sdc、 .slk、.csv、.htm/.html	Calc	电子表格、图表、表格、人事通讯录、地址簿、收据和帐 单、预算、简单数据库
.sxi、.ppt、.sxd、.sdd	Impress	商业和学术演示文稿、万维网演示、演讲、幻灯片放映
.sxd、.sda	Draw	文件可以被导出到多种图像格式，包括.jpg、.bmp、.gif、 和.png 图示、线条绘图、剪贴图片、机构图表

## 17.2 使用 OpenOffice 的字处理程序 Writer

使用 OpenOffice.org 进行文字处理和编辑文档与使用使用微软公司的 Office 套件极为类似。文字处理器不仅具有文本编辑器的功能，而且附加了许多其他功能，这些功能可以用于格式化、设计、打印文档，而不必记忆复杂的格式化标签或编码。OpenOffice 的 Writer 是一个功能强大的文字处理器，无论是将文档打印出来，还是把文件交给他人查看，它都具有 WYSIWYG（所见即所得）格式。图 17-1 显示了运行中的 OpenOffice 的 Writer：

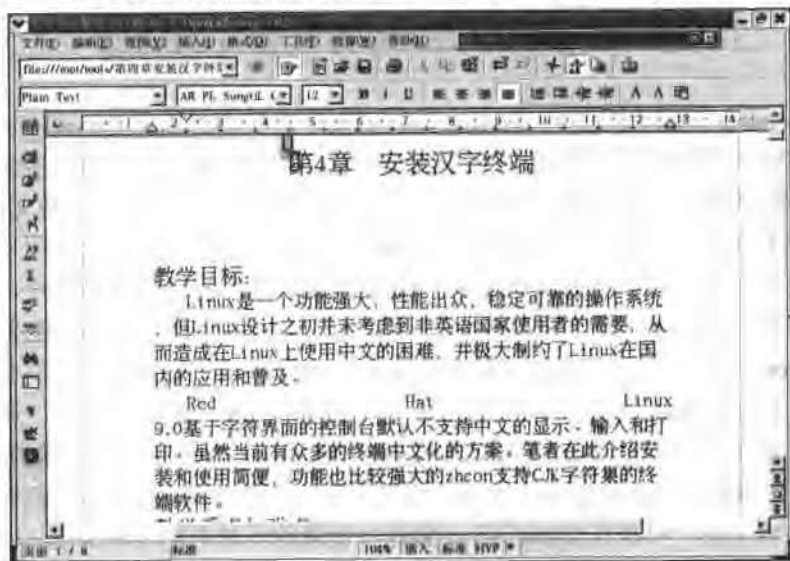


图 17-1 OpenOffice 的字处理程序 Writer

要从桌面面板上启动字处理程序 Writer，选择“主菜单”|“办公”|OpenOffice.org Writer，如果从 shell 提示符启动，则输入 oowriter，然后按回车键。

主界面是文档编辑区域（窗口中间部分）。在窗口顶部的工具栏里提供了各类功能工具，用户利用它们控制字体、纸张大小、文字对齐（向左、向右、或居中对齐文字）等。其中还包括一个文本框，用户可以在里面键入文档的确切位置，以便将它载入编辑区域。工具

栏上还有打开、保存和打印文档的按钮，以及创建新文档的按钮。

窗口左侧工具栏上的按钮可以用来检查拼写、自动突出显示拼错的单词、单词或词组搜索，以及其他便利的编辑功能。可以使用鼠标在某一工具栏按钮上停留几秒钟时间，将会弹出一个提示（Tip），显示该按钮功能的简短说明。用户也可以单击“帮助”菜单，选择“说明文”来显示详细的提示。

用户可以使用默认设置在文档编辑区域立即开始输入文本。要保存文本，单击“存盘文档”按钮。它会打开一个弹出的文件浏览器。用户可以从浏览器窗口底部的“文件类型”下拉菜单中选择文件类型。如果只使用 OpenOffice 应用程序来操作文件，默认的文件类型是恰当的；如果需要把文件发送给微软 Office 用户，或者需要把文件保存为.doc 格式，以电子邮件附件发送，那么用户可以把文件保存为 Microsoft Word 97/2000/XP 类型，其他人就可以使用微软公司的 Word 字处理软件来打开和编辑该文件。

**注意：**默认情况下，用户在 Linux 下打开微软公司的 Word 字处理软件编辑的.doc 文件时，汉字将不能正常显示，这是需要按 Ctrl+A 键全选文件内容，然后在工具栏的字体下拉列表中选择 AR PL SungtiL GB 或者 AR PL KaitiL GB 字体，分别以 Linux 的宋体和楷体显示。

另外，以后也可以在 Linux 中添加微软 Windows 的中文字体来解决这个问题。

OpenOffice Writer 不但可用于常规的文本编辑，也可以在文档中添加对象（如图像、图示、图表以及表格），使文档图文并茂，更富有表达力。要添加图像，单击“插入”|“图形”|“从文件...”菜单项，然后从弹出的文件浏览器中选择图像。该图像会出现在光标所在的位置，可以被扩大或缩小来适应用户的需要。图 17-2 显示了图像被添加到文档中的情形。



图 17-2 在文档中添加图像

创建了文档后，用户可以把它保存为需要的任何格式。请参考表 17-1 中的可用文件格式。

**注意：**用户还可以把文档导出为 HTML 或 PDF 格式，这两种格式可以被几乎所有 Web



浏览器(如 Mozilla 和 IE)或 PDF 查看程序(如 xpdf 或 Adobe Acrobat Reader)读取。

### 17.3 使用 OpenOffice 的电子表格处理程序 Calc

从大型企业到家庭办公室, 各行各业的专业人员都使用电子表格来保存记录、创建商业图表以及处理数据。OpenOffice Calc 是一个电子表格应用程序, 它允许在组织成行列的单元格(cell)中输入和处理数据。单元格是单个数据的容器, 如数量、标签、或数学公式。用户可以执行一组单元格的计算, 或根据包含在一组单元格中的数值来创建图表。用户甚至可以把电子表格的数据插入到 Writer 编辑的文档中。

用户可以从桌面面板上单击“主菜单”|“办公”|OpenOffice.org Calc, 启动 OpenOffice Calc。也可以在 shell 提示下, 键入 oocalc 命令, 启动 OpenOffice.org Calc。图 17-3 显示了 OpenOffice Calc 的运行界面。

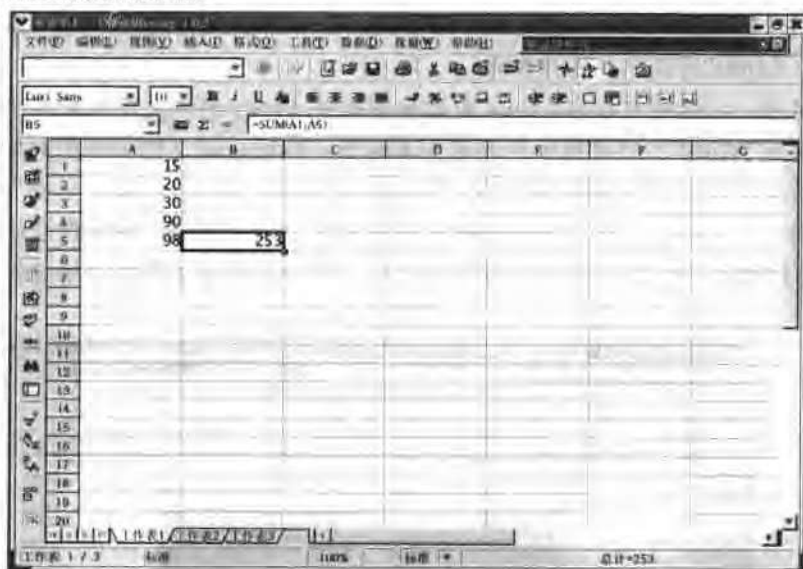


图 17-3 OpenOffice 的电子表格

OpenOffice Calc 允许输入并处理个人或商业数据。用户可以在 A (第一列) 中输入学生成绩, 双击 OpenOffice Calc 中的单元格可以直接在该单元格内输入数据, 也可以使用“输入行”(工具栏上的文本框)为某单元格输入数据。在第二列(B 列)上运行算术公式来求总成绩。OpenOffice Calc 中有几个预设的函数和计算功能, 例如=SUM()用于加法或乘法求和、=quotient()用于除法, 以及=subtotal()用于准备收据。有关在 OpenOffice Calc 中创建函数来计算数值数据的详细信息, 请单击“帮助”|“内容”菜单项, 在相应的帮助文档中找到。

如果需要创建课堂或商业演示所需的图表或图例, OpenOffice 中包括几个图表模板供用户选择。首先突出显示编制图表的区域, 然后单击“插入”|“图表...”。在“自动格式图表”对话框中, 可以对选定的数据范围会进行进一步定制。单击“继续”按钮, 显示使用

这些数据能够创建的不同图表。选择用户需要的式样，然后单击“完成”按钮。该图会在电子表格窗口内显示。用户可以把这些图移动到屏幕上任何位置打印，也可以把该图保存为对象，这样可以把它嵌入到 OpenOffice Writer 文档或 OpenOffice Impress 演示文稿中。

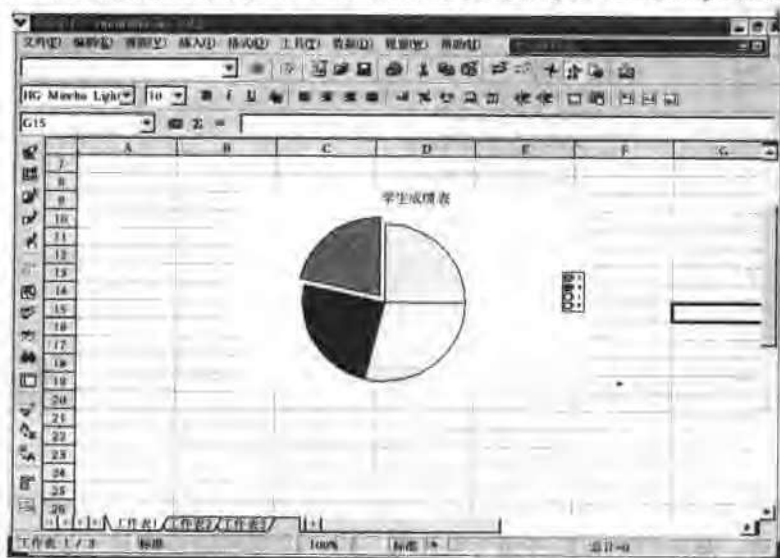


图 17-4 使用 OpenOffice 电子表格创建图表

用户可以把创建的电子表格保存为多种文件格式，包括原始的.sxc 格式和与 Microsoft Office 兼容.xls 格式。此外，用户也可以把绘制的图例另存为种图像文件格式，以便把它们集成到文档文件、网页和演示文稿中。

关于使用 OpenOffice Calc 详细信息，请单击“帮助”|“内容”命令，参阅弹出的帮助页。

## 17.4 使用 OpenOffice 的演示文稿创建程序 Impress

OpenOffice 的 Impress 是一个能够帮助用户制作有说服力的演示文稿的图形化工具。

用户可以单击“主菜单”|“办公”|OpenOffice.org Impress，从图形化桌面上启动 OpenOffice Impress。或者从 shell 提示符下键入 ooinpress，启动 OpenOffice Impress。

OpenOffice Impress 中有一个自动文件辅助 (AutoPilot) 的渐进演示向导。它允许用户从样式模板集合中创建演示。利用它可以创建带有逐项列表、轮廓或者图像的幻灯片，用户甚至可以把 OpenOffice Calc 中的图表导入幻灯片中。图 17-5 显示了运行中的 OpenOffice Impress。

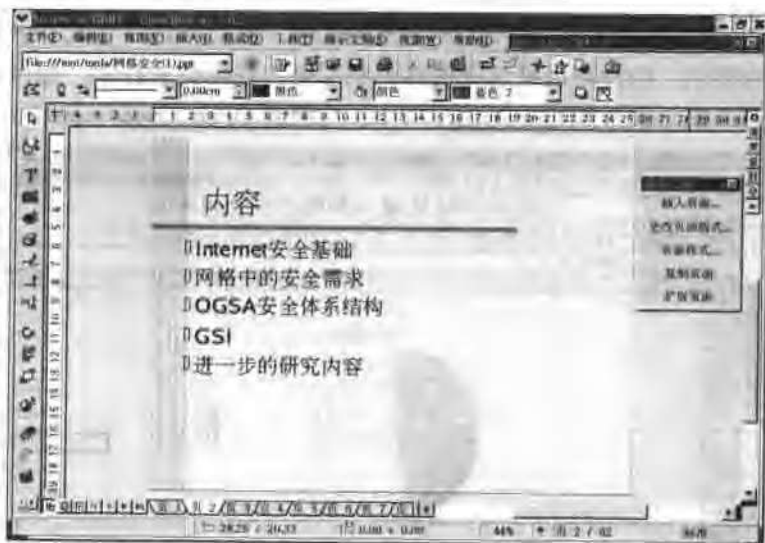


图 17-5 OpenOffice 的 Impress

当用户第一次启动 OpenOffice Impress 时，会看到自动文件助理设置屏幕。可以在次选择幻灯片风格、演示幻灯片的介质（普通纸、幻灯用透明纸或者显示器），以及在计算机上演示幻灯片时所要应用的视觉效果，如图 17-6 所示。

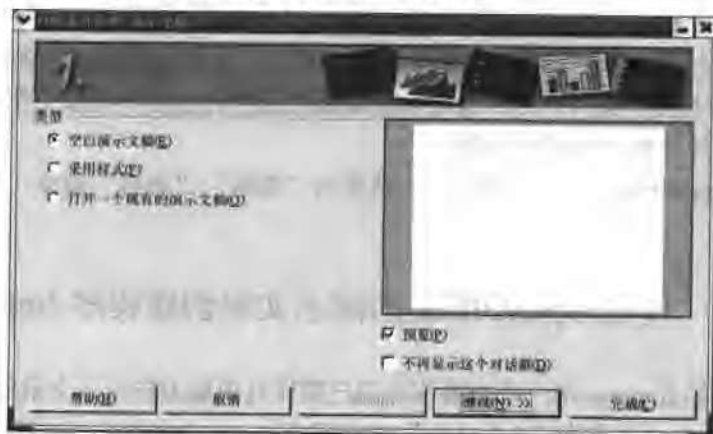


图 17-6 OpenOffice Impress 自动文件助理演示文稿

使用自动文件助理工具选定了首选项后，便可以选择要创建的幻灯片类型。用户可以从列表选择一个预格式化的幻灯片，或选择空白的幻灯片，然后自行定制布局。如果需要在演示文稿添加幻灯片，单击浮动工具栏上的“插入页面...”，会出现一个弹出的对话框，允许用户选择新幻灯片的布局。用户可以在演示文稿中添加任意张所需的幻灯片。

用户还可以通过选择“演示文稿”|“放映演示文稿”命令来随时预览演示文稿。文稿演示会使用全屏模式，可以等待演示完毕后自动退出，或者按 Esc 键来强制终止播放。

用户可以将演示文稿另存为多种文件格式。用户可以保存为默认的 OpenOffice Impress 格式，也可以保存为微软 PowerPoint 格式或者 StarImpress 格式 (.sdd)。用户可以单击“文

件”|“打印”菜单，把演示文稿打印到普通或透明纸张上。

## 17.5 使用 OpenOffice 的图形处理程序 Draw

如果想为文档和演示文稿创建图形，用户可以使用 OpenOffice 的程序 Draw。它将鼠标当作画笔，用户能够进行描画，并把图形保存为多种格式，而后进行打印、添加到网页中或者使用电子邮件发送。

用户可以单击“主菜单”|“办公”|OpenOffice.org Draw，从桌面面板上启动 OpenOffice Draw。也可以在 shell 提示符下输入 oodraw，启动 OpenOffice Draw 程序。图 17-7 显示了 OpenOffice Draw 的运行界面。

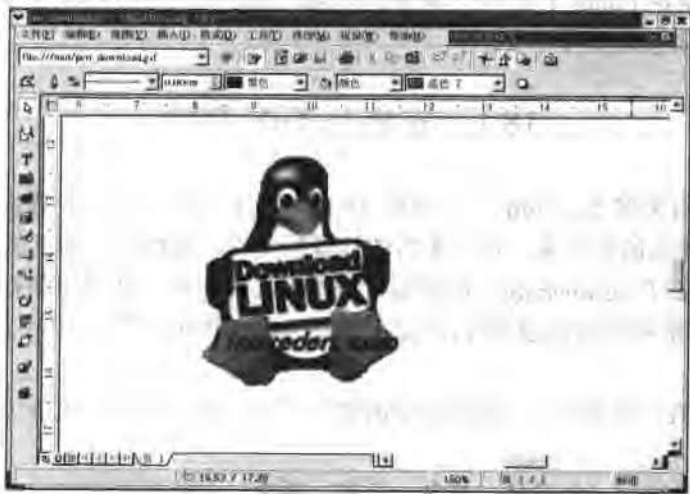


图 17-7 OpenOffice 的图形处理程序 Draw

如果用户熟悉 Linux 下的 GIMP 和 Windows 下的 Photo Shop 之类的图形程序程序，那么会发现 OpenOffice Draw 具有许多相同的基本功能。它有创建直线、曲线这样的线条，圆圈、方块之类的基本形状，圆锥、立方体之类的三维物体等的工具栏。用户可以创建图像，使用主工具栏上的“平面样式/平面充填”下拉列表中的颜色来填充。OpenOffice.org Draw 允许用户打开和导入图像，并使用所提供的工具对图像进行修改。

当用户完成了对图像或图示的修改之后，可以把文件保存为多种文件格式，或者导出为.jpg 或.png 等通用格式。请参阅表 17-1 获得兼容图像文件格式的完整列表。

关于使用 OpenOffice Draw 的详细信息，请单击“文件”|“帮助”|“内容”命令，查阅弹出的文档。

## 思考与练习

- (1) 如何安装办公套件 OpenOffice?
- (2) OpenOffice 有哪些组件? 每一个组件的功能是什么?
- (3) 在 OpenOffice 下能与微软的 Office 套件兼容吗?

## 第 18 章 排版软件 LaTeX 的配置和使用

目前世界上存在多种排版系统,各有特点和适用范围。例如方正电子排版系统已经是国内报社排版的首选软件,而普通用户,则使用所见即所得的排版系统(如微软公司的 Word 和 OpenOffice.org 的 Writer)。

本章介绍的 TeX 系统是一种使用方便、价格低廉的排版软件。当排版论文、报告和书籍时,其输出质量并不逊色于价格高昂的大型排版系统,特别是在数学公式排版方面,它是排版质量最好的系统。

本章首先介绍在 Linux 下 LaTeX 系统的安装,然后简要介绍 LaTeX 的使用,最后介绍安装和配置 CJK 宏包以支持中文。

### 18.1 安装 LaTeX 套件

TeX 系统是由美国 Stanford 大学教授 Donald E.Knuth 研制的计算机排版软件系统。Knuth 是国际上著名的数学家,也是著名的计算机专家,他编著了著名的计算机系列专著 *The Art of Computer Programming*。在编写这些专著的过程中,他对出版社给他的校样不满意,认为校样的字形和版面都很难看,于是他花费大量的时间和精力开发了举世著名的 TeX 系统。

LaTeX 是在 TeX 的基础上,增加方便的宏包系统,使用户使用更加方便。LaTeX 系统的特色包括:

- 自动生成编号(如章号、节号、段落、图表、文献和页码的自动编号)。
- 自动生成目录。
- 自动生成索引附录。

LaTeX 深受国内外科学工作者的喜爱,因为它能排版复杂的图表和数学公式。许多国际专业学会的期刊杂志都要求使用 TeX 格式投稿。

Red Hat Linux 9 的安装盘中已经自带了 LaTeX 的 Linux 版本 **teTeX**。用户可以在 shell 提示符后输入 **redhat-config-packages** 命令(或者单击“主菜单”|“系统设置”|“添加/删除程序”菜单项)启动软件包管理器。在弹出的“软件包管理”对话框中,在“桌面”类选中“授权和出版”复选框(如果还未选中),如图 18-1 所示。然后单击“更新”按钮,进行更新。

该套件由如下几个部分组成。

- **tetex**: LaTeX 排版软件。
- **xdvi**: VDI 文件的 X 查看器,查看由 LaTeX 源文件(.tex)生产的文件。
- **xpdf**: 可以查看由 dvi 文件生成的 PDF 文件。

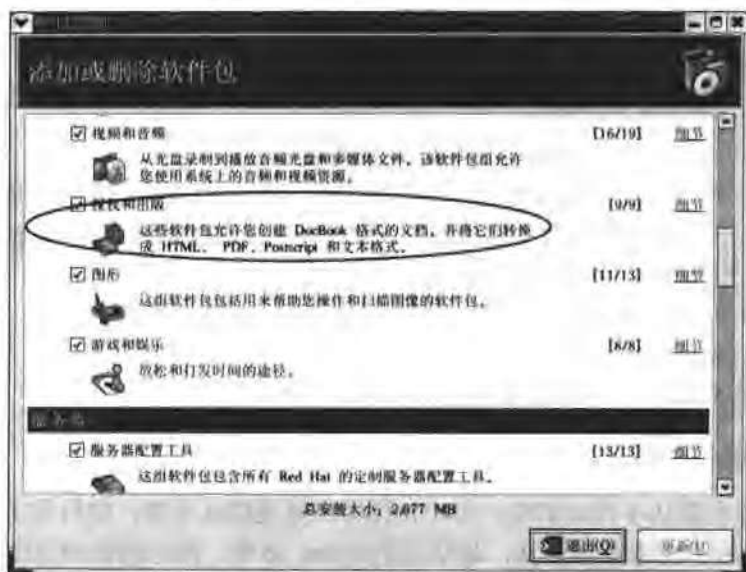


图 18-1 安装 LaTeX 套件

## 18.2 简单使用 LaTeX

### 18.2.1 LaTeX 文档的编写

#### 1. 文章架构

用 LaTeX 编排文章只有几个法则，但已够用来编排复杂的文章。LaTeX 也尽可能简化了编译的过程。

实际上 LaTeX 是一种文件标记语言，它把文档的正文和文字的输出形式分开处理。例如 TeX 规定：每一节标题的格式，为预留 36 点的垂直距离，然后输出两倍粗体的标题和页数，隔两个字符的垂直距离，再继续正文的输入。相对的，LaTeX 使用 `\section{}` 命令，它会完成上述的所有的工作。如果需要改变它的格式，可以修改 `\section{}` 来代替更改文件内容。这将会使得重新格式化超过数十页的文件，会节省大量的时间。

所有 LaTeX 文件包含 3 个部份：导言、正文和结尾。这些都是 TeX 专家常用的语法。在导言部份，至少需要指定文档的类别和文件正文的开始标志。示例如下：

```
\documentclass{article}
```

```
\begin{document}
```

文件的结尾通常很简单，如果无特殊情况，它只有如下一行：

```
\end{document}
```

其中 `\begin{document}` 和 `\end{document}` 必须成对使用，在 LaTeX 中，被称作做环境，所有输入文档的具体内容都须要在环境中，而且许多命令只有在它们的环境中才有作用，然而在 LaTeX 中，document 环境是惟一必需的环境。

文件格式也叫做参数，正文内定字高度为 10 点，若要改变预定格式，可以更改参数，如

```
\documentclass[12pt]{article}
```

字高度改为 12 点。

下列是一些常用的文档格式。

- **article**: 文章类，一般用于学术论文的编排。
- **report**: 报告类，它预先设定一个主题名称，每一节由新的一页开始。
- **letter**: 书信类，它包含一些特别的 **addresses**、**salutations** 和 **closings** 定义。
- **book**: 书籍类，用于书籍的排版。

用户可以使用 `\usepackage{}` 命令，来加入宏包，具体格式如下：

```
\usepackage{宏包名}
```

注意：`\usepackage{}` 必须在 `\begin{document}` 之前。

## 2. 字符与正文的编写

一般 LaTeX 会默认字符的字型，同时允许用户更改默认字型，如粗体字、斜体字或者画底线等。有几种方法来强调正文，最常用的是 `\em` 命令，在它的范围里面所有的正文，都用强调字型排版，如

```
This word will be {\em emphasized.}
```

如果用户已经用强调体字型格式化了某个段落，可以用 `\noem` 取消用强调字型的命令，使用，一个反斜线和一个向前的斜线，如

```
This example {\em will\/} print correctly.
```

```
This example will {\em not} print correctly.
```

LaTeX 也提供了许多特殊字符和数学符号，常用的符号举例如表 18-1 所示。

表 18-1 常见的符号

输出字符	输入序列
#	\#
\$	\\$
%	\%
{	\{
}	\}
~	\~{}
^	\^{}

常见的数学符号举例如表 18-2 所示。

表 18-2 常见的数学符号举例

符号	表示
A	\alpha
γ	\gamma
Λ	\Lambda
Δ	\Delta

### 3. 边缘和行距

TeX 或 LaTeX 文件里的边缘不可以直接变更，它往往跟正文有关系，放置变更命令的位置也是很重要的。

对于控制 LaTeX 文件宽度的命令，可使用 `\evensidemargin` 和 `\oddsidemargin`，它们分别会改变偶数页的左侧和奇数页的右侧。例如：

```
\evensidemargin=1in
\oddsidemargin=1in
```

使得正常离边缘一英寸的偶数页的左侧和奇数页的右侧，再额外增加 1~2 英寸，而且这些命令会影响到全部的文件，不会被其他局部的命令影响，所以诗文体和列表很适合使用这些命令。

设定行距也有些复杂。`baselineskip` 的量是设定文章的行距，通常都会给一个正的数，举例如下：

```
\baselineskip=24pt
或使用较好的命令：
\setlength{\baselineskip}{24pt}
```

二者之间的差异是：`setlength` 会在其他影响文章编排命令之前执行。

使用 `baselineskip` 的问题是，它也会影响到区段的标题，注脚和相似的命令。如果使用 `baselineskip`，必须随时注意它对正在格式的文档的影响。

### 4. 文章格式

LaTeX 提供标准化的格式提供给文件输入，诸如目录、引用段落、脚注及其他文件的格式供人使用。下列包含一些常用的标准格式。

文章格式和报告格式是相似的，主要差别是报告格式需要预先设定一个主题名称，每一节由新的一页开始。

例：为了要生成主题名称、摘要和每一行，预先在前面输入。

```
\title{Using LaTeX}
\author{Zhonghua Jiang}
\abstract{The basic usage is discussed in this paper, and many examples are
gived.}
```

然后，必须使用如下命令：

```
\maketitle
```

会使得用报告格式的文章在开始的地方或第一页的顶端产生主题名称和摘要。

其他段落的命令定义如下：

```
\section
\subsection
\subsubsection
```

这些命令将会在文件中产生标准的并已编号的小节，若不让每小节自动编号，则使用如下命令：

```
\section*
```



```
\subsection*
```

```
\subsubsection*
```

LaTeX 提供许多表现输出文件的环境供用户使用。用户能用引用环境，包括要引用的部分。命令如下：

```
\begin{quotation}
```

```
Start of paragraph to be quoted...
```

```
... end of paragraph.
```

```
\end{quotation}
```

比较短的引述，可以使用引述环境。

至于要格式化诗文，则用诗文体环境。命令如下：

```
\begin{verse}
```

```
Because I could not stop for death\\
```

```
He kindly stopped for me
```

```
\end{verse}
```

可以使用两个反斜线 (\\)，达到跳行的目的，否则 LaTeX 会以当前的诗体格式，继续编排直到这一行结束。

至于列表也有几种方式，用 list 环境的列举格式。命令如下：

```
\begin{list}
```

```
\item
```

```
This is the first item of the list.
```

```
\item
```

```
This is the second item of the list...
```

```
\item
```

```
... and so on.
```

```
\end{list}
```

用 enumerate 环境标示号码顺序：

```
\begin{enumerate}
```

```
\item
```

```
Item No. 1.
```

```
\item
```

```
Item No. 2.
```

```
\item
```

```
\dots
```

```
\end{enumerate}
```

用 description 环境描述主题：

```
\begin{description}
```

```
\item{Oven} Dirty, needs new burner.
```

```
\item{Refrigerator} Dirty. Sorry.
```

```
\item{Sink and drainboard} Stained, drippy, cold water faucet.
```

```
\end{description}
```

## 5. 信件格式

信件格式特别为商业信件制定一些特殊的用法。

该信件格式有一点需要注意，即收信人的住址。`address` 的命令必须在文件导言部分给出，而定义寄件人名字的 `signature` 命令则在结束之后出现。

一个使用 LaTeX 信件格式的简单例子如下，文件名为 `letter.tex`：

```
\documentclass[12pt]{letter}
\signature{Zhonghua Jiang}
\address{4 South St.\\Zhongguancun, Haidian District. Beijing 100080\\Tel:
86-10-82683089}
\begin{document}
\begin{letter}{ABC Def Co.\\100 cat St.\\BeiJing, 82683089}
\opening{Dear Sir:}
```

LaTeX is a high-quality typesetting system, with features designed for the production of technical and scientific documentation. LaTeX is the de facto standard for the communication and publication of scientific documents.

```
(Remainder of the body of the letter.)
```

```
\closing{Sincerely,}
```

```
\end{letter}
```

```
\end{document}
```

注意住址中两个反斜线的地方，就是跳行的地方。

## 6. 举例

一个最简单的 LaTeX 文档示例为 `hello.tex` 文件，其内容如下：

```
\documentclass{article}
\begin{document}
Hello. This is a simple LaTeX document.
\end{document}
```

### 18.2.2 编译 LaTeX 文档

(1) 使用 LaTeX 命令编译 LaTeX 源文件。

命令格式如下：

```
latex LaTeX源文件
```

`latex` 命令的功能是将扩展名为 `.tex` 的源文件编译成同名的扩展名为 `.dvi` 的目标文件。`dvi` 文件是跨平台的文件，可以被任何 DVI 浏览器阅读，在 Red Hat Linux 9 中提供了两个 DVI 查看器：DVI 查看器 (`xdvi` 命令) 和 DVI viewer (`kveiwshtml dvi` 命令)。

例：编译 18.2.1 小节的 `letter.tex` 文件。命令如下：

```
[globus@jiang globus]$ latex letter.tex
```

编译过程如下:

```
This is TeX, Version 3.14159 (Web2C 7.3.1)
(letter.tex
LaTeX2e <2001/06/01>
Babel <v3.7h> and hyphenation patterns for american, french, german,
ngerman,i
talian, nohyphenation, loaded.
(/usr/share/texmf/tex/LaTeX/base/letter.cls
Document Class: letter 1999/04/29 v1.2z Standard LaTeX document class
(/usr/share/texmf/tex/LaTeX/base/size12.clo)
No file letter.aux.
[1] (letter.aux) )
Output written on letter.dvi (1 page, 872 bytes).
Transcript written on letter.log.
```

结果表明, 生成了文件为 letter.dvi 的目标文件。

例: 编译 hello.txt 文件。命令如下:

```
[globus@jiang globus]$ latex hello.tex
```

结果生成 hello.dvi 的目标文件。

(2) 查看 DVI 文件。

用户分别单击“主程序”|“图形”|“DVI 查看器”命令和“主程序”|“图形”|“更多图形程序”|DVI viewer 命令, 打开 DVI 查看器和 DVI viewer。当启动程序后, 弹出打开文件的对话框, 然后选择要查看的 dvi 文件, 便可以查看 DVI 文件的内容。

例: 使用 xdvi 查看刚生成的 letter.dvi 文件。命令如下:

```
[globus@jiang globus]$ xdvi letter.dvi
```

显示结果如图 18-2 所示。



图 18-2 Linux 下的 xdvi 查看器

例：使用 DVI 浏览器查看刚生成的 hello.dvi 文件，如图 18-3 所示。



图 18-3 用 DVI 查看器查看 hello.dvi 文件

(3) 将 DVI 文件转化成 PDF 格式的文件。

在 Red Hat Linux 9 中，用户可以使用 `dvipdf` 命令将 DVI 格式的文件转化成 PDF 文件。`dvipdf` 命令的语法格式如下：

```
$ dvipdf DVI文件名
```

`dvipdf` 命令将 DVI 格式的文件转化成同名的 PDF 文件。

例：将 `letter.dvi` 文件转化成了 `letter.pdf`。命令如下：

```
[globus@jiang globus]$ dvipdf letter.dvi
```

使用 `ls` 验证转换结果，命令如下：

```
[globus@jiang globus]$ ls letter*
```

```
letter.aux letter.dvi letter.log letter.pdf letter.tex
```

结果表明 `letter.pdf` 文件已经生成。

使用 Acrobat Reader 5.0.8 显示生成的 `letter.pdf` 文件，命令如下：

```
[globus@jiang globus]$ /usr/local/Acrobat5/bin/acroread letter.pdf
```

显示的结果如图 18-4 所示。

(4) 为了打印 TEX 文件，也可以使用 `dvips` 命令将 DVI 格式转化成 PS 格式，如

```
[globus@jiang globus]$ dvips letter.dvi
```

结果将 `letter.dvi` 转化成了 `letter.ps` 文件，可以使用 PS/PDF 查看器查看和打印 PS 格式的文档。



图 18-4 用 Acrobat Reader 阅读刚生成的 PDF 文件

## 18.3 LaTeX 中文化

LaTeX 默认不支持双字节字符，因此不能支持中文字符集。CJK 是一个 LaTeX2e 的宏包 (macro package)，它能让用户在 LaTeX 文件中使用 CJK (中文/日文/韩文，Chinese/Japanese/Korean) 的文字编码。

为了使 LaTeX (Linux 中为 TeX) 支持中文，需要首先安装 CJK 宏包，然后安装和转化简体中文字体。

### 18.3.1 下载和安装 CJK4.5.2 宏包

(1) 下载 CJK4.521 宏包。

从 <http://cjk.ffii.org> 下载 CJK4.5.2 源文件 `cjk-4.5.2.tar.gz` 和格式化的帮助文件 `cjk-4.5.2-doc.tar.gz`。

对源文件进行解压缩，命令如下：

```
tar xvzf cjk-4.5.1.tar.gz
```

(2) 安装 CJK 宏包。

将 CJK 宏包的文件拷贝到 LaTeX 的 `texmf` 目录，命令如下：

```
cd cjk-4.5.2
```

```
cp -rf texinput /usr/share/texmf/tex/LaTeX/CJK
```

至此，CJK 宏包已经安装结束。

### 18.3.2 安装和配置中文字体

(1) 下载并解压 `gbkfonts-linux-0.3.tar.bz2`。

在 google 中搜索 `gbkfonts-linux`，可以找到文件 `gbkfonts-linux-0.3.tar.bz2`，下载到本地之后，执行如下命令：

```
# tar xvjf gbkfonts-linux-0.3.tar.bz2
```

解压缩文件。

(2) 生成相应的字体转换文件。

本文从 Windows 2003 的字体文件夹中，拷贝 Windows 的宋体字体文件 `simsun.ttc`，如果操作系统不是 Windows 2003，字体文件可能是 `simsun.ttf`。命令如下：

```
# mkdir /usr/share/texmf/fonts/truetype
```

```
# cp /mnt/2003/WINDOWS/Fonts/simsun.ttc /usr/share/texmf/fonts/truetype
```

使用 `gbkfonts` 命令生成 CJK 需要的字体映射文件，命令如下：

```
# cd gbkfonts-linux-0.3
```

```
# ./gbkfonts /usr/share/texmf/fonts/truetype/simsun.ttc song
```

目录中的 `song` 表示宋体。

注意：用户可以使用 `gbkfonts` 命令为多个字体生成字体转换文件。例如 `simfang.ttf`（仿宋字体文件）、`simkai.ttf`（楷体字体文件）和 `simhei.ttf`（黑体字体文件），它们可以分别使用 `fs`、`kai` 和 `hei` 表示。

将生成的文件复制到目标目录，命令如下：

```
# cp -rf dvips tex dvipdfm fonts /usr/share/texmf/
```

(3) 修改 `teTeX` 相应的配置文件。

```
# ./appendconf /usr/share/texmf
```

再编辑文件 `/usr/share/texmf/dvips/config/config.ps`，在它的最后加一行，加入：

```
p +cjk.map
```

(4) 更新系统配置。

在 Shell 中执行如下命令：

```
# mktexlsr
```

这样系统配置就已经生效，并且所有的配置就完成了。

### 18.3.3 在 LaTeX 中使用中文

为了使用 CJK 宏包，必须在编辑的 `.tex` 文件中的导言部分，使用 `usepackage` 命令加载 CJK 宏包。

要使用中文字体，必须将中文字符放在 `\begin{CJK*}{GBK}{song}` 和 `\end{CJK*}` 之间。

以下是一个使用 CJK 宏包的中文文件例子。

编辑一个 `test.tex` 文件，内容如下：

```
\documentclass{article}
\usepackage{CJK}
\begin{document}
```

```
\begin{CJK*}{GBK}{song}
```

这是一个测试简体中文字体的文件。

```
\end{CJK*}
```

```
\end{document}
```

将 test.tex 文件编译成目标文件 test.dvi，再将 test.dvi 转化为 test.pdf。命令如下：

```
[globus@jiang globus]$ latex test.tex
```

```
[globus@jiang globus]$ dvi2pdf test.dvi
```

注意：由于 xdvi 本身的问题，如果 tex 文件中有中文字符，那么 xdvi 将会非常慢，

因此建议将 DVI 文件转化成 PDF 文件后，再用 PDF 阅读器查看。

使用 Acrobat Reader 查看 test.pdf，结果如图 18-5 所示。

```
[globus@jiang globus]$ /usr/local/Acrobat5/bin/acroread test.pdf
```

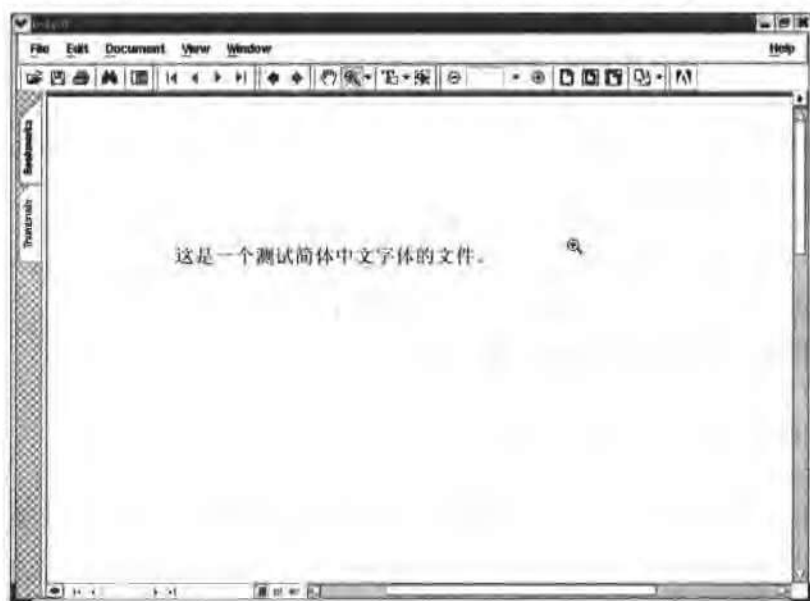


图 18-5 使用 LaTeX 生成中文的 PDF 文件

注意：如果用户为 LaTeX 添加了多种中文字体，那么用户可以在 LaTeX 编辑的源文件（.tex）中使用“\CJKfamily{字体标识}”改变文字使用的中文字体。

## 思考与练习

- (1) LaTeX 的作用是什么？
- (2) LaTeX 文件的格式的特点是什么？
- (3) 如何使用 LaTeX 处理中文？

## 第 19 章 架设 Web 服务器

Apache Web 服务器是世界上使用最多的 Web 服务器之一，它历史悠久，历经了多年的考验和不断完善，使它能在大多数硬件和软件平台下使用。良好的跨平台能力、稳定的性能、丰富的功能、支持多种扩充模块的灵活性是 Apache Web 服务器的特色。

本章将介绍了 Red Hat Linux 9 自带的 2.0 版本的 Apache Web 服务器，讲述了安装、使用、配置 Apache Web 服务器的知识。

### 19.1 安装和使用 Apache 服务器

在 Red Hat Linux 9 中，Apache Web 服务器的版本为 2.0.4，其配置文件的配置项与早期版本有些差异。Apache 版本 2.0.4 的 RPM 包的包名为 httpd。

#### 19.1.1 安装 Apache 服务器

Red Hat Linux 9 的 Apache 软件包的文件名为 httpd-2.0.40-21.i386.rpm，该文件在它的第一张安装光盘上。

##### 1. 检查 httpd 服务器是否被安装

检查 httpd 软件是否已被安装，命令如下：

```
[root@jiang gbkfonts-linux-0.3]# rpm -qa | grep httpd
```

如果已被安装，其显示结果如下：

```
httpd-manual-2.0.40-21
```

```
redhat-config-httpd-1.0.1-18
```

```
httpd-2.0.40-21
```

其中 httpd-2.0.40-21 是 Apache 服务器软件包，对于 Apache 服务器来说是必需的。httpd-manual-2.0.40-21 是 Apache 服务器的手册，它是可选的，但建议安装。redhat-config-httpd-1.0.1-18 是 Red Hat Linux 提供的用于设置 Apache 服务器的工具，它被称为“Apache 配置”工具，这个工具是 Red Hat Linux 独有的，建议安装，它能极大地简化 Apache 服务器的配置。

##### 2. 安装 Apache 服务器及其相关软件

如果没有安装这些 RPM 包，则以根用户身份，执行如下命令安装：

###### ● 安装 Apache 服务器 RPM 包

```
# rpm -ivh httpd-2.0.40-21.i386.rpm
```

###### ● 安装 Apache 服务器手册

```
# rpm -ivh httpd-manual-2.0.40-21.i386.rpm
```

###### ● 安装 Apache 配置工具

```
# redhat-config-httpd-1.0.1-18.i386.rpm
```



### 19.1.2 启动 Apache 服务器

(1) 根用户可以在 shell 提示符后使用如下命令启动 Apache 服务器:

```
# service httpd start
```

(2) 使机器开机后自动启动 Apache 服务器

如果希望系统引导时, 自动启动 Apache 服务器, 需要以根用户身份执行如下命令:

```
#chkconfig --level 345 httpd on
```

### 19.1.3 测试 Apache 服务器

在浏览器中输入服务器的地址, 可以访问 Apache 服务器。例如, 访问 192.168.1.190 上的 Apache 服务器, 需要在浏览器上输入 `http://192.168.1.190`, 判断服务器是否启动, 设置是否生效, 如图 19-1 所示。



图 19-1 Apache 服务器的测试页

## 19.2 配置 Apache 服务器的过程

用户有两种方法配置 Apache 服务器。一种方法是直接修改 Apache 服务器的配置文件 `/etc/httpd/conf/httpd.conf`, 另一种方法是使用 Red Hat Linux 提供的基于 X 窗口系统的 Apache 配置工具, 间接修改配置文件 `/etc/httpd/conf/httpd.conf`。

通常来说, 使用 Apache 配置工具可以简化配置工作, 但是它的灵活性却不如直接修改配置文件 `/etc/httpd/conf/httpd.conf`。用户可以通过图形化界面来配置虚拟主机、记录属性和最大数量连接等服务器属性。

HTTP 配置工具只能配置包括在 Red Hat Linux 中预装的模块。如果用户安装了额外的

模块, 则不能使用这个工具来安装和配置。为了使用 Apache 的配置工具, 需要 X 窗口系统和根权限。通过单击“主菜单”|“系统设置”|“服务器设置”|“HTTP 服务器”, 或在 shell 提示符后输入 `redhat-config-httpd` 命令来启动 Apache 服务器配置工具。

为了节省篇幅, 笔者简要介绍如何使用 Apache 配置工具配置 Apache 服务器。Apache 配置工具的界面如图 19-2 所示。



图 19-2 Apache 配置工具界面

**注意:** 如果用户使用 Apache 配置工具, 就不要手工编辑 `/etc/httpd/conf/httpd.conf` 文件。HTTP 配置工具会自动生成这个文件。如果想添加额外的模块或 Apache 配置工具不支持的配置选项, 则不能使用该配置工具。

使用 Apache 配置工具配置 Apache 服务器的一般步骤如下:

- (1) 在“主”选项卡下进行基本设置。
- (2) 单击“虚拟主机”选项卡配置默认选项。
- (3) 在“虚拟主机”选项卡下配置虚拟主机。
- (4) 在“服务器”选项卡下进行服务器设置。
- (5) 在“调整性能微调”选项卡下配置连接设置。
- (6) 把所有页面文件拷贝到文档根 (DocumentRoot) 和 `cgi-bin` 目录。
- (7) 退出程序并保存设置, 最终更新了 `/etc/httpd/conf/httpd.conf`。
- (8) 为了使设置生效, 重新启动 Apache 服务器。

## 19.3 对 Apache 进行基本设置

使用 Apache 配置工具的窗口中的“主”选项卡进行基本的服务器设置, 如图 19-2 所示。主要设置如下:

- 服务器名称。在“服务器名称”文本框中输入有权使用的完整域名。该选项和 `httpd.conf` 中的 `ServerName` 指令相对应。`ServerName` 指令设置万维网服务器的主机名, 它用来创建 URL 的重导向。服务器名称也可以是 IP 地址。服务器实际的 DNS 名称是 `jiang.mydomain.com`, 则可以将服务器名称设为 `www.mydomain.com`。
- Web 服务器管理员的邮件地址。在“网主电子邮件地址”文本框中输入 Web 服务

器管理员的邮件地址，它与 `httpd.conf` 中的 `ServerAdmin` 指令相对应。它主要用于事件通知。

- 服务器的监听的端口号。“可用地址”文本框用来定义服务器的端口号，它与 `httpd.conf` 中的 `Listen` 指令对应。Apache 服务器的默认端口号为 80。单击“添加”按钮可以定义接受请求的其他端口。

注意：如果服务器的端口号被设置成小于 1024，那么只有根用户才能启动它。否则 `httpd` 服务可以被普通用户启动。

## 19.4 对 Apache 进行默认项设置

当设置了“服务器名称”、“网主电子邮件地址”以及“可用地址”之后，单击 Apache 管理工具的“虚拟主机”选项卡，然后再单击上面的“编辑默认设置”按钮。将会弹出如图 19-3 所示的对话框。用户可以在该对话框中配置 Apache 服务器默认项。

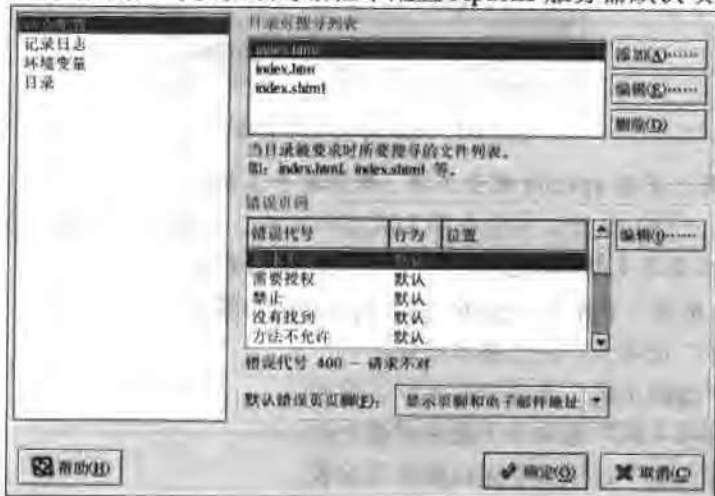


图 19-3 “站点配置”对话框

### 19.4.1 站点配置

“目录页搜索列表”中列出的文件设置了 Apache 配置文件中的 `DirectoryIndex` 指令。`DirectoryIndex` 是用户通过在目录名后加一个斜线 (/) 来请求目录索引时由服务器提供的默认网页。

注意：当某用户请求网页 `http://www.mydomain.com/webapp1` 时，如果 `DirectoryIndex` 网页存在，服务器将用 `DirectoryIndex` 网页作为请求的响应，否则服务器将生成目录列表返回用户。服务器会试图查找 `DirectoryIndex` 指令中列出的文件，并返回找到的第一个文件。如果没找到任何文件，并且 `Options Indexes` 为该目录设置，服务器就会生成并返回一个 HTML 格式的列表，列出该目录中的子目录和文件。

使用“错误代号”节来配置 Apache 服务器在出现错误时，将用户的请求重定向到的位

置（本地或外部 URL）。该选项修改 `ErrorDocument` 指令。当客户连接 Apache 服务器出错后，服务器默认显示“错误代号”列中的简单错误信息。如果要覆盖默认设置，选择该错误代号，然后单击“编辑”按钮。选择“默认”来显示默认的简短错误讯息，选择 URL 可以将用户的请求重定向到一个外部 URL，并可以在“位置”字段中输入包括 `http://` 在内的 URL。选择“文件”可以将用户重定向到内部 URL，并在 Web 服务器的文档根下输入文件的路径。位置必须以斜线（/）开头，并相对于文档根的位置。

在“默认错误页页脚”下拉列表框中，用户有如下选择：

- 显示页脚和电子邮件地址。指示在所有错误页中显示默认页脚以及 Apache 服务器管理员的邮件地址（参见 `ServerAdmin` 指令）。
- 显示页脚。在错误页的底部只显示默认的页脚。
- 无页脚。在错误页的底部不显示页脚。

#### 19.4.2 记录日志设置

服务器默认把传输日志记录在 `/var/log/httpd/access_log` 文件中，把错误日志记录在 `/var/log/httpd/error_log` 文件，如图 19-4 所示。

传输日志包含所有对 Apache 服务器企图连接的列表，它记录试图连接的用户的 IP 地址、试图连接的日期和时间以及试图请求服务器上的文件。用户可以在“记录到文件”文本框中输入日志文件的路径和文件名。如果路径和文件名不以斜线（/）开头，则路径相对于配置的服务器根目录。该选项设置配置文件的 `TransferLog` 指令。



图 19-4 记录日志

用户还可以配置定制的日志格式。选择“使用定制记录设施”单选按钮，然后在“定制日志字符串”文本框中输入定制的日志字符串，它设置了 `LogFormat` 指令。

错误日志包含服务器发生的错误列表。可以在“记录到文件”文本框中输入该日志文件的路径和文件名。该选项修改 `ErrorLog` 指令。

“日志级别”下拉列表用来设置错误日志信息的详细程度。从最简略到最详细有如下级别：`emerg`、`alert`、`crit`、`error`、`warn`、`notice`、`info` 和 `debug`。该选项修改 `LogLevel` 指令。

“反向 DNS 查寻”下拉列表用来设定 HostnameLookups 指令。它包括“无逆向查寻”、“反向查寻”和“双重逆向查寻”3 种选择。一般来说，应该把该选项设为“无逆向查寻”，因为 DNS 请求会给服务器增加负载，这样会导致服务器性能的下降。特别当服务非常繁忙时，试图执行逆向查寻或双重逆向查寻对性能的影响非常明显。

### 19.4.3 环境变量

为了 CGI 脚本或 SSI（服务器端嵌入）页，有时需要修改环境变量。Apache 服务器可以使用 mod\_env 模块来设置被传递给 CGI 脚本和 SSI 页面的环境变量，用户可以使用如图 19-5 所示的“环境变量”页进行配置。



图 19-5 设置环境变量

用户使用“为 CGI 脚本设置”组设置要传递给 CGI 脚本和 SSI 页面的环境变量。例如将环境变量 USERNAME 设为 jiang，则单击“为 CGI 脚本设置”组的“添加”按钮。然后在“环境变量”文本框中输入 USERNAME，并在“设置的值”中输入 jiang，最后单击“确定”按钮，完成环境变量 USERNAME 的设置。

可以使用“为 CGI 脚本设置”来设置配置文件的 SetEnv 指令。使用“传递给 CGI 脚本”在服务器首次启动 CGI 脚本时传递环境变量值。shell 下的 env 可以查看环境变量列表。用户单击“传递给 CGI 脚本”右边的“添加”按钮，在弹出的对话框中输入环境变量的名称。然后单击“确定”按钮添加到列表中。“传递给 CGI 脚本部分”用来修改 PassEnv 指令。

“为 CGI 脚本取消设置”可以删除某个环境变量，它与 UnsetEnv 指令对应。

### 19.4.4 目录

使用“目录”页来为指定目录配置选项。它与<Directory>指令对应。用户可以对目录配置如下选项：

- ExecCGI。若选定，则允许执行 CGI 脚本。
- FollowSymLinks。允许追随符号链接。

- Includes。允许服务器端嵌入。
- IncludesNOEXEC。允许服务器端嵌入，但禁用#exec 和#include 命令。
- Indexes。如果目录中不存在 DirectoryIndex（如 index.html），则生成目录列表。
- Multiview。该选项被默认禁用。
- SymLinksIfOwnerMatch。只有在所有者，才能追随该符号链接。

## 19.5 虚拟主机设置

用户可以利用 Apache 配置工具来配置虚拟主机。利用虚拟主机，允许用户为不同的 IP 地址、主机名或同一机器上的不同端口运行不同的服务器。

为某个虚拟主机设置的指令只应用于该虚拟主机。如果某指令使用“编辑默认设置”按钮，对于整个服务器全局设置，而虚拟主机设置中却没有被定义，那么默认设置就会被使用。

### 19.5.1 添加和编辑虚拟主机

虚拟主机配置界面如图 19-6 所示。



图 19-6 虚拟主机配置界面

为了添加虚拟主机，用户可以单击“虚拟主机”选项卡，然后单击“添加”按钮。如果用户为了编辑虚拟主机，可以从列表中选择一個虚拟主机，然后单击“编辑”按钮。

“常规选项”设置只应用于正在配置的虚拟主机。“虚拟主机名称”、“文档根目录”和“网主电子邮件地址”与“主”选项卡类似。“主机信息”中可以选择“默认虚拟主机”、“基于 IP 的虚拟主机”和“基于名称的虚拟主机”之一。

“默认虚拟主机”。用户只能配置一个默认虚拟主机。当请求的 IP 地址没有在另一个虚拟主机中确切列出时，默认的虚拟主机就会被使用。如果默认虚拟主机没有被定义，则主服务器设置就会被使用。

“基于 IP 的虚拟主机”选项用来配置<VirtualHost>指令，可以指定“IP 地址”，要指定端口，使用 IP Address:Port 格式。



“基于名称的虚拟主机”。如果用户选择“基于名称的虚拟主机”，一个根据服务器的主机名称来配置 `NameVirtualHost` 指令的窗口就会出现。在“IP 地址”字段内指定 IP 地址。要指定多于一个 IP 地址，用空格把它们分开。要指定端口，使用 `IP Address:Port` 格式。使用“:”来为该 IP 地址配置所有端口。在“服务器主机名”字段中指定虚拟主机的主机名。在“别名”部分，单击“添加”按钮来添加主机名的别名。添加别名会在 `NameVirtualHost` 指令内添加 `ServerAlias` 指令。

### 19.5.2 SSL 支持

SSL 支持的界面如图 19-7 所示。



图 19-7 Apache 的 SSL 支持

如果 Apache 服务器没有配置 SSL（安全套节字层）支持，那么 Apache 服务器与客户之间的通信是以明文传输的，而没有被加密。这对于不包括私人或保密信息的网站来说是可行的。例如，发布开源软件和文档的网站就不必使用安全通讯。然而对于获取信用卡信息的电子商务网站就应该使用 Apache SSL 来加密通信通道。

当启用 Apache SSL 时，会启用 `mod_ssl` 安全模块。要通过 Apache 配置工具来启用它，用户必须在“主”选项卡的“可用地址”中允许通过端口 443 的访问。然后在“虚拟主机”选项卡中选择虚拟主机名，并单击“编辑”按钮。从左侧的列表中选择 SSL，再在右边选择“启用 SSL 支持”复选框，如图 19-7 所示。“SSL 配置”部分被预配置了虚构的数码证书。数码证书为安全 Web 服务器提供验证，并向客户浏览器表明安全服务器的身份。为了安全起见，用户必须在国际著名 CA（认证机构）另行购买自己的数字证书。不要在正式的网站上使用 Red Hat Linux 中提供的虚构证书。关于购买 CA 认可的数字证书的信息，请查阅相关的资料。

注意：用户不能在 SSL 中使用基于名称的虚拟主机，因为 SSL 握手（浏览器接受 Web 服务器的证书时）发生在识别正确的基于名称的虚拟主机的 HTTP 请求之前。如果想使用基于名称的虚拟主机，它们只能在不安全的 Web 服务器中使用。

## 19.6 服务器设置

在 Apache 配置工具的“服务器”选项卡，允许用户配置基本的服务器设置，如图 19-8 所示。默认设置在多数情况下都是适用的，而不需要修改。



图 19-8 Apache 配置工具的服务器配置页

(1) 锁文件。它的值与 LockFile 指令相对应。在服务器使用 USE\_FCNTL\_SERIALIZED\_ACCEPT 或 USE\_FLOCK\_SERIALIZED\_ACCEPT 编译时，该指令把路径设为锁文件所用的路径。它必须贮存在本地磁盘上。

(2) PID 文件。它的值与 PidFile 指令相对应。该指令设置服务器记录进程 ID (PID) 的文件。该文件应该只能够被根用户读取。

(3) 核心转储目录。它的值与 CoreDumpDirectory 指令相对应。Apache HTTP 服务器在转储核心前会试图转换到该目录中，默认值是 ServerRoot。然而，如果运行服务器的用户所使用的身份没有该目录的写权限，核心转储就无法被写入。如果用户想把核心转储写入磁盘以用于调试目的，请把这个值改为能够被服务器的运行身份写入的目录。

(4) 用户。它的值与 User 指令相对应。它设置服务器回答请求所用的 userid。用户的设置决定服务器的访问权限。该用户所无法访问的文件，网站的来宾也不能够访问。其默认值为 apache。该用户应该仅拥有一定特权，因此它能够存取外部用户可以看见的文件。该用户还是所有被服务器产生 CGI 进程的所有者。

在正常操作中，httpd 父进程首先以根用户身份运行，但是它会立即把控制权交给 apache 用户。服务器必须以根用户启动，原因是它需要关联到 1024 以下的端口。1024 以下的端口是为系统使用而保留的，因此只有根用户才有使用权。一旦服务器把自己连接到它的端口，它就会在接受任何连接请求前，把进程交给 apache 用户。

“群组”的值与 Group 指令相对应。Group 指令和 User 指令类似。它设置服务器回答请求所用的组群。默认组群也是 apache。



## 19.7 调整性能

单击“调整性能”标签来配置想服务器使用的最大子进程数，以及客户连接方面的 Apache HTTP 服务器选项，如图 19-9 所示。在多数情况下，这些选项的默认设置都是很好的。改变这些设置会影响 Apache 服务器的整体性能。



图 19-9 Apache 的调整性能

“最多连接数量”设置服务器能够同时处理的客户请求的最多数量。服务器为每个连接创建一个 `httpd` 子进程。进程数量达到最大限度后，直到某子进程结束，Apache 服务器才能够收新客户连接。如果不重新编译 Apache，该选项设置的值最好不能超过 256。该选项与 `MaxClients` 指令相对应。

“连接超时”定义服务器在通信时等候传输和回应的秒数。“连接超时”定义了 Apache 服务器在接收 GET 请求时的超时时间、在接收 POST 或 PUT 请求的 TCP 包的超时时间、以及在回应 TCP 包的 ACK 之间的超时秒数。默认设为 300 秒，它与 `Timeout` 指令相对应。

“每次连接最多请求数量”为每个持续连接所允许的最多请求次数。默认值为 100，这应该在多数情况下都是适用的，该选项与 `MaxRequestsPerChild` 指令相对应。

如果选中“允许每次连接可有无限限制请求”选项，则 `MaxKeepAliveRequests` 指令的值就会是 0，这会允许无限制的请求次数。

如果选择了“允许持久性连接”选项，`KeepAlive` 指令就会被设为 `true`，并且 `KeepAliveTimeout` 指令的值会被设为“下次连接的超时时间”中选定的值。该指令设置的超时秒数是服务器在回应了一项请求之后，关闭连接之前，等待下一个请求时会等候的秒数。一旦接收到请求，服务器就会改用“连接超时”中的值。

## 19.8 保存设置

如果不想保存所做的服务器配置，单击 Apache 配置工具窗口右下角的“取消”按钮。然后在确认提示中，单击“是”按钮确认选择，这时设置不会被保存。

如果想保存所做的 Apache 服务器配置，单击 Apache 配置工具窗口右下角的“确定”按钮。然后出现一个对话框，如果单击“是”按钮，设置就会被保存在 `/etc/httpd/conf/httpd.conf` 文件中，原有的配置会被覆盖。

如果第一次使用 Apache 配置工具，用户会看到一个警告，提示配置文件已经被手工修改的对话框。如果 Apache 配置工具检测到 `httpd.conf` 配置文件已被手工修改，它会把手工修改的文件保存为 `/etc/httpd/conf/httpd.conf.bak`。

保存设置之后，必须使用 `service httpd restart` 命令来重新启动 `httpd` 守护进程。只有根用户才能执行该命令。

## 思考与练习

- (1) 如何安装、启动和测试 Apache 服务器？
- (2) 配置 Apache 服务器的步骤是什么？
- (3) 如何对 Apache 的默认项进行设置？
- (4) 如何设置 Apache 的虚拟主机？

## 第 20 章 架设域名服务器

DNS (域名服务) 是 Internet 的一个重要的基础设施, 由于域名数据库是一个分布在互联网上的分布式数据库, 因此需要域名服务器提供局部域名数据库。一个企业往往都需要一台甚至是多台域名服务器。

BIND 是 Unix/Linux 下的域名服务器, Red Hat Linux 9 自带了 BIND 软件包, Red Hat 提供了 BIND 配置工具, 这将使网络管理员很方便地架设基于 Linux 的域名服务器。

本章首先介绍了 DNS 的原理, 然后简要介绍了 BIND 软件, 最后较为详细地介绍了如何用 Red Hat Linux 9 提供的 BIND 配置工具来配置 BIND 服务器。

### 20.1 DNS 和 BIND

#### 20.1.1 DNS 原理简介

DNS (域名服务, Domain Name System) 可以实现主机域名和 IP 地址之间的相互转换, 因此在 Linux 环境下构建一个安全可靠、高效的 DNS 系统是非常有用的。

在 DNS 中, 各个域分别由不同的组织管理。每个组织都可将它所在的域再分成一定数量的子域, 并将这些子域委托给其他组织管理。域名被用做 DNS 数据库中的索引, 子域中任何域名都被认为是域的一部分。判断一个域是否为另一域的子域, 简单方法是比较它们的域名, 一般子域名以其父域名结尾。

设计域名系统的一个主要目的是让管理分散化。管理域的组织将该域划分成子域, 每一个子域可以由其他组织管理, 这意味着那些下级组织负责维护自己子域的所有数据。它们可以改变数据, 甚至可以将自己管理的子域再划分成更多的子域并重新分配, 而父域中仅包含指向这些子域的指针。

实现域名查询和提供域名信息的程序被称为域名服务器。通常域名服务器拥有部分域名空间 (称之为区, 或者 Zone) 的完整信息, 一个域名服务器可以拥有对多个区的授权。域既包含了该层主机, 又包括了子域的所有信息; 而区仅包含了一个域中除了分配出去让其他组织管理的子域以外的域名数据信息。如果一个域没有分配任何子域, 则该区包含该域的所有域名数据信息。

DNS 定义了 PM (主域, Primary Master) 和 SM (次域, Secondary Master) 两类域名服务器。PM 从配置文件中读取区域数据, 这些数据对整个区域来说都是具有权威性的。SM 域名服务器是从其他具有该区授权的 PM 域名服务器上获得区数据, 并定期查询 PM 域名服务器, 以保证区数据为最新版本。一般情况下, 最好设立一台 PM 域名服务器和若干台 SM 域名服务器。这样可以分担负载, 以确保区中所有主机都有比较靠近的域名服务器方便访问。

对域名系统的访问是按照客户/服务器模式进行的, 采用缓存 (Caching) 技术在保证数据可靠真实性的同时, 又保证了较高的效率。应用程序一般是利用解析器来实现域名解析

的。解析器是一组库函数，任何需要解析域名的应用程序都会调用这组函数来完成域名解析工作。解析器是应用程序实现域名查询的接口。

域名服务器在接收到客户的查询请求时，一般是按照递归的方式来进行的。下面用一个例子来说明什么叫递归方式查询。一个域名服务器 MyDM 接收到查询 `www.linuxaid.com.cn` 的查询请求。它首先询问本地存储的根域名服务器列表中的任何一个根域名服务器，负责 `cn` 域的名字服务器有哪些。前面曾经说过，上级域名服务器有指向下级子域名服务器的指针。所以根名字服务器就会返回负责 `cn` 域数据信息的名字服务器的列表。MyDM 会缓冲这些数据继续查询其中的一个，询问负责 `linuxaid` 子域的域名服务器有哪些。得到信息后再递归查询，直到找到 `www.linuxaid.com.cn` 的 IP 信息。查询中得到的所有域名服务器信息都会被缓存起来，以加速以后的查询。

域名服务器有生存期（TTL）的概念，一旦生存期到了，名字服务器必须丢弃缓存数据并从权威的名字服务器中重新获取新的数据。这样可以确保域数据在整个网络上的一致性。

从 TCP/IP 协议栈的角度来看，DNS 属于应用层协议，运行在传输层之上，但是它并不使用 TCP 提供的服务，而是使用 UDP 服务。

### 20.1.2 BIND 简介

BIND（Berkeley Internet Name Domain）是 DNS 协议的一个实现，而且提供了一个开放的可充发布的参考实现，它包含如下组件：

- 一个域名服务器（`named`）。
- 一个域名系统解析器库。
- 验证 DNS 服务器的一些工具。

BIND 域名服务器是互联网用得最广泛的名字服务器软件，它为企业提供了一个鲁棒和稳定的名字服务构架。BIND 是 UNIX/Linux 构架名字服务器的软件。

## 20.2 安装和配置 BIND

### 20.2.1 安装 BIND

如果在安装系统时已经安装了 BIND，而且安装了图形化配置工具，那么可以使用 `rpm` 查询已安装的 RPM 包，应该有如下结果：

```
[root@jiang root]# rpm -qa | grep BIND
redhat-config-BIND-1.9.0-13
BIND-utils-9.2.1-16
BIND-9.2.1-16
```

结果表明系统已经安装了 BIND 服务器软件、BIND 实用工具和图形化的配置工具。如果用户的机器没有安装这些包，使用如下命令安装这些包：

```
# rpm ivh BIND-9.2.1-16.i386.rpm
# rpm ivh BIND-utils-9.2.1-16.i386.rpm
```

```
# rpm ivh redhat-config-BIND-1.9.0-13.i386.rpm
```

### 20.2.2 启动和停止 BIND

BIND 服务器在 Linux 中的服务名被称作 `named`。

如果需要启动 BIND 服务器，则使用如下命令：

```
# /sbin/service named start
```

如果需要停止 BIND 服务器，则使用如下命令：

```
# /sbin/service named stop
```

如果需要 BIND 服务器重新加载配置文件，则使用如下命令：

```
# /sbin/service named reload
```

让 BIND 服务器开机后自动启动，命令如下：

```
# chkconfig --level 345 named on
```

### 20.2.3 配置 BIND

下面将要讲述如何使用 BIND 配置工具 (`redhat-config-BIND`) 来配置基本的 BIND 服务器区块。BIND 配置工具在每次保存设置时，都会创建 `/etc/named.conf` 配置文件和 `/var/named` 目录中的区块配置文件。

**注意：**不要编辑 `/etc/named.conf` 配置文件。BIND 配置工具在每次保存设置时会重新生成该文件。如果用户想设置 BIND 配置工具无法配置的设置，请把这些设置添加到 `/etc/named.custom` 文件中。

BIND 配置工具需要 X 窗口系统和根用户权限。用户可有如下两种方法启动 BIND 配置工具：

- 单击面板上的“主菜单”|“系统设置”|“服务器设置”|“域名服务”菜单项
- 在 shell 提示符下，输入 `redhat-config-BIND` 命令，然后按回车键。

BIND 配置工具的主界面如图 20-1 所示。



图 20-1 BIND 配置工具

BIND 配置工具把默认的区块目录配置到 `/var/named` 目录。所有指定的区块文件都是相对于该目录的。BIND 配置工具还对输入值进行基本的语法检查。例如，IP 地址不能包括

除数字和点 (.) 字符之外的字符。

BIND 配置工具允许添加一个正向主区, 一个逆向主区和一个从区。添加了这些区块后, 就可以从主窗口中编辑或删除它们, 如图 20-1 所示。

添加、编辑、删除某区之后, 用户必须单击工具栏上的“保存”按钮, 或者选择“文件”|“保存”命令来保存设置, 它将会把相应的信息写入/etc/named.conf 配置文件和 /var/named 目录中的每个区块文件。保存设置以后, BIND 配置工具还会使 named 服务重新加载配置文件。选择“文件”|“退出”命令会在退出 BIND 配置工具之前保存所做改变。

### 1. 添加正向主区 (域名转化成 IP 地址)

要添加正向主区 (又称首要主区), 单击工具栏上“新建”按钮, 将会弹出如图 20-2 所示的“选择一个区块类型”对话框, 选择“正向主区块”单选按钮, 然后在“域名”文本框内输入主区的域名。



图 20-2 “选择一个区块类型”对话框

当单击图 20-2 中的“确定”按钮后, 弹出一个如图 20-3 所示的“名称到 IP 的翻译”对话框, 该对话框中包含以下选项:

- “名称”。它是在前一个窗口中输入的域名。
- “文件名”。DNS 数据库文件的文件名, 相对于/var/named 而言。它被预设为由域名补了.zone 的域名。
- “联系”。主区的主要电子邮件联系地址。
- “主名称服务器 (SOA)”。它是授权的开始 (SOA) 记录, 它指定最适合该域信息的名字服务器。
- “序列号码”。DNS 数据库文件的序列号码。在每次文件发生改变时, 这个号码都应该递增, 因此该区块的次名称服务器就能够检索到最新的数据。BIND 配置工具在每次配置发生改变的时候都会递增该号码。它还可以被手工递增, 方法是单击“序列号码”旁边的“设立”按钮。
- “时间设置”。贮存在 DNS 数据库文件中的“刷新”、“重试”、“过期”和“至少”生存周期 (TTL) 值, 它们都以秒为单位。
- “记录”。添加、编辑和删除关于“主机”、“别名”、和“名称服务器”之类的资源记录。



图 20-3 “名称到 IP 的翻译”对话框

必须指定“主名称服务器 (SOA)”，你必须单击“记录”选项区域的“添加”按钮来至少指定一个名称服务器记录。

当正向主区配置完成后,单击“确定”按钮来返回到如图 20-1 所示的主窗口。然后单击工具栏上的“保存”按钮,将设置写入/etc/named.conf 配置文件,并在/var/named 目录中建立所有单独的区块文件,最后使 named 守护进程重新加载配置文件。

该配置在/etc/named.conf 文件中创建了和以下相似的项目:

```
zone "mydomain.com.cn" {
    type master;
    file "mydomain.com.cn.zone";
};
```

它还创建了带有以下信息的/var/named/mydomain.com.cn.zone 文件:

STTL 86400

```
@      IN      SOA      jiang.mydomain.com.cn.  root.localhost (
                                2 ; serial
                                28800 ; refresh
                                7200 ; retry
                                604800 ; expire
                                86400 ; ttl
                                )
```

IN NS 1192.168.1.190.

```
jiang IN A 192.168.1.190
```

```
maindns IN CNAME jiang
```

## 2. 添加逆向主区（将域名转换为 IP 地址）

要添加逆向主区，单击图 20-1 所示的窗口工具栏上的“新建”按钮，并选择创建“逆向主区块”。输入想配置的 IP 地址范围的前 3 个八位组。譬如，如果想配置的 IP 地址范围是 192.168.1.0/255.255.255.0，在“IP 地址（前 3 个八位组）”文本框中输入 192.168.1，如图 20-4 所示。

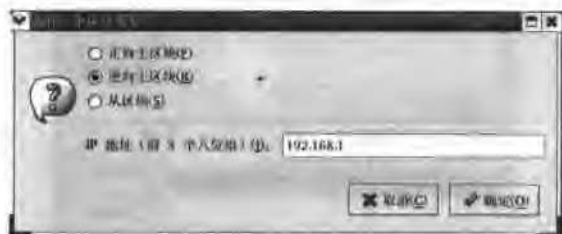


图 20-4 输入 IP 地址范围

在图 20-4 中单击“确定”按钮之后，将弹出如图 20-5 所示的对话框，其中包括下列选项：

- “IP 地址”。它是在 20-4 中输入 IP 地址的前 3 个八位组。
- “逆向 IP 地址”。不可编辑，它是根据输入的 IP 地址填充的。
- “联系”。主区管理员的电子邮件联系地址。
- “文件名”。/var/named 目录中 DNS 数据库文件的名称，默认值即可。
- “主名称服务器 (SOA)”。授权开始 (SOA) 记录，它指定最适合该域信息的名称服务器。
- “序列号码”。DNS 数据库文件的序列号码。在每次文件发生改变时，这个号码都应该递增，因此该区块的次名称服务器就能够检索到最新的数据。BIND 配置工具在每次配置发生改变的时候都会递增该号码。它还可以被手工递增，方法是单击“序列号码”值旁边的“设立”按钮。
- “时间设置”。贮存在 DNS 数据库文件中的“刷新”、“重试”、“过期”和“至少”生存周期 (TTL) 值，它们都以秒为单位。
- “名称服务器”。为逆向主区添加、编辑或删除名称服务器，至少需要一个名称服务器。
- “逆向地址表”。在逆向主区和它们的主机名内的 IP 地址列表。例如，对于逆向主区 192.168.1，可以在“逆向地址表”中添加 192.168.1.19，以及主机名 unkown.mydomian.com.cn。主机名一定要以点 (.) 结束，表明它是主机的全名。





图 20-5 添加反向主区

- “主名称服务器 (SOA)” 必须被指定，必须单击“反向地址表”选项区域的“添加”按钮来至少指定一台名字服务器记录。

配置了反向主区后，单击“确定”来返回到如图 20-1 所示的主窗口。从下拉菜单中，单击“保存”按钮来写入/etc/named.conf 配置文件以及/var/named 目录中所有单独的区块文件，并使守护进程重新加载配置文件。

该配置在/etc/named.conf 文件中创建了以下项目：

```
zone "1.168.192.in-addr.arpa" {
    type master;
    file "1.168.192.in-addr.arpa.zone";
};
```

它还创建了带有以下信息的/var/named/1.168.192.in-addr.arpa.zone 文件：

```
$TTL 86400
```

```
@      IN      SOA      jiang.mydomain.com.cn.  root.localhost (
                                2 ; serial
                                28800 ; refresh
                                7200 ; retry
                                604800 ; expire
                                86400 ; ttk
                                )
```

```
@      IN      NS      jiang.mydomain.com.cn.
```

```

19      IN      PTR      jiang.mydomain.com.cn.
190     IN      PTR      jiang.mydomain.com.cn.

```

### 3. 添加从区块

要添加从区块（又称次要主区），单击图 20-1 对应的窗口工具栏上的“新建”按钮，并选择“从区块”。在“域名”文本框中输入从区块的域名。

一个如图 20-6 所示的对话框就会出现，其中包括下列选项：

- “名称”。在前一个窗口中输入的域名。
- “主区块列表”。从区块中检索数据的名称服务器。该值必须是有效的 IP 地址，只能在文本区域内输入数字和点（.）。
- “文件名”。/var/named 目录中 DNS 数据库文件的名称。



图 20-6 添加从区块

配置了从区块后，单击“确定”按钮来返回到如图 20-1 所示的主窗口。从工具栏中，单击“保存”按钮来写入/etc/named.conf 配置文件，以及/var/named 目录中所有单独的区块文件，并使守护进程重新载入配置文件。

该配置在/etc/named.conf 文件中创建了与以下相似的项目：

```

zone "second.mydomain.com.cn" {
    type slave;
    file "second.mydomain.com.cn.zone";
    masters {
        192.168.1.190;
    };
};

```

配置文件/var/named/slave.example.com.zone 在 named 服务从主区块服务器中下载区块数据时被创建。

## 20.3 测试 DNS 服务器

### 20.3.1 DNS 客户机端的改动

对于内部网上的 Unix 主机，应该在 `/etc/resolv.conf` 中增加一行 `nameserver` 说明，使得系统能够搜索后备 DNS 服务器。本文例子的 `/etc/resolv.conf` 的内容为：

```
search mydomain.com.cn
nameserver 192.168.1.190
nameserver 192.168.1.191
```

其中 `nameserver 192.168.1.190` 为主名字服务器，而对于 `mydomain.com.cn` 域，使用后备名字服务器 `mydomain.com.cn`。

然后重新启动 `xinetd` 服务，使修改的 `/etc/resolv.conf` 生效。

注意：在 Red Hat Linux 中可以使用 `netconfig` 和 `redhat-config-network` 命令设置 DNS 服务器的 IP 地址。

对于 Windows 95/NT/XP/2003，则是在 DNS 服务器的列表中将 `192.168.1.190` 作为主 DNS 服务器添加进去即可，然后再添加第二 DNS 服务器 `192.168.1.190`，顺序决定了访问各 DNS 服务器的优先级。

### 20.3.2 使用 nslookup 测试 DNS 服务器

无论是 Windows 系统，还是 Unix 和 Linux 系统，都可以使用 `nslookup` 查询名字服务器，而 Red Hat Linux 还提供了 `dig`、`host` 和 `nslookup` 等多种名字查询工具。

本节将以 `nslookup` 命令为例，说明查询在 20.2 节中设置的 DNS 服务器的过程。

```
[root@jiang linux-2.6.6]# nslookup
```

```
Note: nslookup is deprecated and may be removed from future releases.
```

```
Consider using the 'dig' or 'host' programs instead. Run nslookup with
the '-sil[ent]' option to prevent this message from appearing.
```

```
> jiang.mydomain.com.cn
```

```
Server:          192.168.1.190
```

```
Address:         192.168.1.190#53
```

```
Name:   jiang.mydomain.com.cn
```

```
Address: 192.168.1.190
```

```
> maindns.mydomain.com.cn
```

```
Server:          192.168.1.190
```

```
Address:         192.168.1.190#53
```

```
maindns.mydomain.com.cn canonical name = jiang.mydomain.com.cn.
```

```
Name:   jiang.mydomain.com.cn
```

```
Address: 192.168.1.190
> www.xaut.edu.cn
Server:      192.168.1.190
Address:     192.168.1.190#53
```

Non-authoritative answer:

```
Name:  www.xaut.edu.cn
Address: 202.200.112.6
```

结果表明,使用的名字服务器的 IP 地址为 192.168.1.190,测试了 3 个查询,分别是 `jiang.mydomain.com.cn` (主机记录)、`maindns.mydomain.com.cn` (别名记录) 和 `www.xaut.edu.cn` (查询其他名字服务器),都能得到正确的结果。

## 思考与练习

- (1) DNS 的作用是什么? DNS 系统是如何组织的?
- (2) 如何配置 BIND?
- (3) 主区块、从区块的作用是什么?
- (4) 如何测试 BIND 服务器?

## 第 21 章 多种平台之间的文件共享

Samba 使用 SMB 协议，通过网络连接来共享文件和打印机。支持 SMB 协议的操作系统包括 Microsoft Windows（通过它的网络邻居）、OS/2 和 Linux。它最大的优点是能方便地实现 Windows 和 Linux 之间的文件和打印机共享。

网络文件系统（NFS）是一种网络上的机器间共享文件的方法，文件就如同位于客户的本地硬盘驱动器上一样。Red Hat Linux 9 既可以是 NFS 服务器也可以是 NFS 客户，这意味着它可以把文件系统导出给其他系统，也可以挂接从其他机器上导入的文件系统。它主要用于 Linux 和 Unix 之间共享文件。

本章将介绍了如何在 Red Hat Linux 9 中配置和使用 Samba 和 NFS 服务器，并且介绍了客户如何连接 Samba 服务器和 NFS 服务器。

### 21.1 与 Windows 系统共享文件：Samba

如果局域网中既有 Windows 机器又有 Linux 机器，Samba 就会发挥作用。Samba 允许文件和打印机被该局域网中的所有系统共享。如果用户只打算在 Red Hat Linux 机器间共享文件，也可以使用 NFS（网络文件系统）。

#### 21.1.1 配置 Samba 服务器

通过配置文件/etc/samba/smb.conf，允许用户通过 Samba 共享 Linux 的目录，同时它也能将 Linux 打印机作为 Samba 共享打印机，供 Linux 用户和 Windows 用户共享。例如可以在 Linux 系统上连接打印机，然后利用 Samba，Windows 机器能通过网络使用 Linux 打印机，进行文档打印。

可以手工修改配置文件/etc/samba/smb.conf，也可以通过图形化配置工具来间接配置 Samba。本节将介绍如何使用图形化工具配置 Samba 服务器。

要使用图形化界面来配置 Samba，需要 Samba 服务器配置工具。Samba 服务器配置工具是用来管理 Samba 共享、用户以及基本服务器设置的图形化工具。它修改/etc/samba/目录中的配置文件。

##### 1. 启动 Samba 服务器配置工具

必须处于 X 窗口系统，并且安装了 redhat-config-samba 的 RPM 包。启动方式：

- 从桌面启动。单击面板上的“主菜单”|“系统设置”|“服务器设置”|“Samba 服务器”。
- 从 shell 提示符启动。输入 redhat-config-samba，并按回车键。

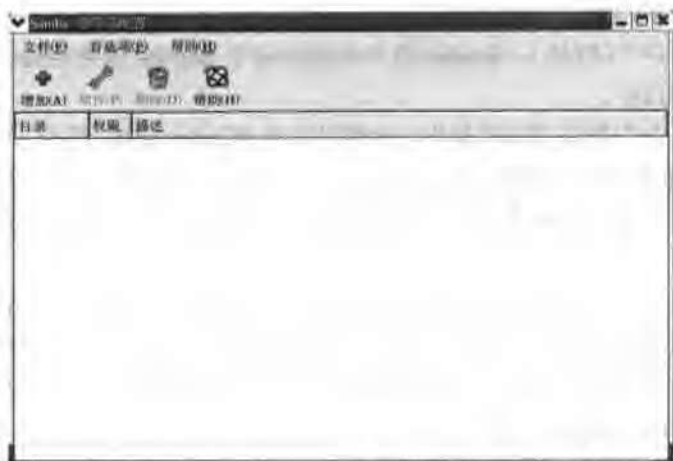


图 21-1 Samba 服务器配置工具

## 2. 配置服务器设置

配置 Samba 服务器，首先需要配置服务器的基本设置和几个安全选项。单击 Samba 配置工具的“首选项”|“服务器设置”菜单项。弹出的对话框的“基本”选项卡如图 21-2 所示。

在“基本”选项卡上，指定该 Samba 服务器的工作组（与 Windows 的工作组含义相同），以及描述信息。它们与 smb.conf 文件中的 workgroup 和 server string 选项相对应。

“服务器设置”对话框的“安全性”选项卡如图 21-3 所示。

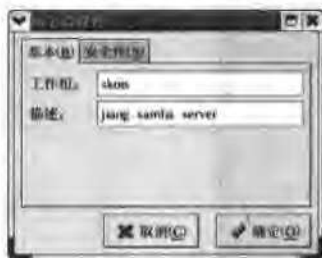


图 21-2 配置服务器设置的基本设置



图 21-3 服务器安全配置

“安全性”选项卡包含以下选项：

- “验证模式”。它和 security 选项相对应。“验证模式”下拉列表有如下选项：
  - “域”。Samba 服务器将依赖于 Windows NT 主域或备份域控制器来校验用户。需要在“验证服务器”文本框指定主域或备份域控制器的 NetBIOS 名。这时必须将“加密口令”选项设置为“是”。
  - “服务器”。Samba 服务器让另一个 Samba 服务器来验证用户名和口令。若无法校验，服务器会使用“用户”验证模式。这时需在“验证服务器”中输入另一个 Samba 服务器的 NetBIOS 名称。
  - “共享”。Samba 用户不必为每个 Samba 服务器都输入用户名/口令。只有在

连接指定的共享时才提示输入用户名和口令。

- “用户”(默认)。Samba 用户必须为每个 Samba 服务器提供一个有效的用户名和口令。
- “加密口令”(默认值为“是”)。如果用户从 Windows 98、打补丁的 Windows NT 4.0 或其他更新版本 Microsoft Windows 中连接, 该选项必须被启用。口令在服务器和客户间使用加密格式而非以纯文本格式传输。它与 encrypted passwords 选项相对应。
- “来宾账号”。当用户或来宾用户要登录 Samba 服务器时, 它们必须被映射到服务器上的有效用户。选择系统上的已有的用户名之一作为 Samba 来宾账号。当用户使用来宾账号登录 Samba 服务器时, 他们拥有与这个用户相同的特权。该选项与 guest account 选项相对应。

注意: 在图 21-3 中来“来宾账号”设置为 guest, Linux 系统默认没有这个用户, 需要使用 useradd guest 添加一个新的账号。

单击了“确定”按钮后, 所做改变会被写入配置文件, 守护进程会被重新启动, 因此改变会立即生效。

### 3. 管理 Samba 用户

Samba 服务器配置工具要求在添加 Samba 用户之前, 对应的 Linux 系统上必须存在一个活跃的已有的用户账号。Samba 用户和这个已有的 Linux 用户账号相关联。

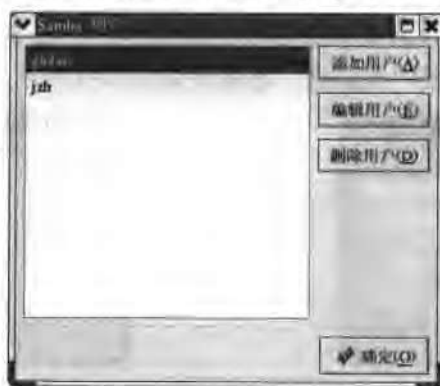


图 21-4 管理 Samba 用户

选择“首选项”|“Samba 用户”菜单, 然后单击“添加用户”按钮, 添加 Samba 用户。在“创建新 Samba 用户”窗口中的本地系统上的已有的用户列表中选择“Unix 用户名”。

如果用户在 Windows 机器上有一个不同的用户名, 并将从 Windows 系统登录该 Samba 服务器, 需要在“Windows 用户名”字段中指定 Windows 用户名。“服务器设置”首选项的“安全性”选项卡上的“验证模式”必须被设置为“用户”, 才能是这个选项生效。在图 21-4 中的两个 Samba 用户 globus 和 jzh 使用了 Linux 和 Windows 相同的用户名。

还需要为 Samba 用户配置一个“Samba 口令”, 并再输入一次确认口令。即使用户选

择了为 Samba 使用加密口令,也建议用户为所有用户设置的 Samba 口令不同于 Linux 系统相应的口令。单击“编辑用户”按钮,编辑某个已有的用户。单击“删除用户”按钮,删除已有的 Samba 用户。

单击了“确定”按钮后,用户设置会立即生效。

#### 4. 添加共享

单击工具栏上的“添加”按钮(单击“文件”|“添加共享”菜单)添加共享。添加 Samba 共享对话框的界面(“创建 Samba 共享”)如图 21-5 所示。



图 21-5 “创建 Samba 共享”对话框

“创建 Samba 共享”对话框中的“基本”选项卡可以配置以下选项:

- “目录”。通过 Samba 共享的目录,这个目录必须存在。
- “描述”。对共享的简短描述。
- “基本权限”。用户具有读取或者读写共享目录中的文件。

在“访问”选项卡上,选择是否要“只允许指定的用户来访问”共享,还是“允许所有用户”(Samba 用户)访问共享。如果选择了“只允许指定用户的访问”选项,则从可用的 Samba 用户列表中选择这些用户。

单击了“确定”按钮后,被添加的共享会立即生效。

#### 5. 启动和停止 Samba 服务

用户也可以直接修改/etc/samba/smb.conf 配置文件。如果修改了这个配置文件,只有重启 Samba 守护进程(可以使用 service smb restart 命令)后,修改才会生效。

如果希望 smb 服务在开机后自动启动,可以使用 ntsysv、chkconfig 或者服务配置工具来设置它。例如使用如下命令设置 Samba 服务在引导时自动启动:

```
# chkconfig --level 345 smb on
```

使用以下命令来查看 Samba 守护进程的状态:

```
# /sbin/service smb status
```

使用以下命令来启动 Samba 守护进程:

```
# /sbin/service smb start
```

使用以下命令来停止 Samba 守护进程:

```
# /sbin/service smb stop
```



### 21.1.2 连接 Samba 共享

要从微软 Windows 系统上连接 Linux 下的 Samba 共享, 使用网络邻居或图形化文件管理器。例如上述实例中 Samba 服务器的主机名为 jiang, 一个共享名为 cdrom, 建立了两个 Samba 用户, 他们分别是 globus 和 jzh。

在 Windows 系统的 IE 浏览器和文件管理器中, 用户可以与访问 Windows 中的共享名相同的方式访问 Linux 系统中的 Samba 共享, 图 21-6 显示了在 Windows 2003 操作系统中访问 jiang 主机 (操作系统为 Linux) 上的 cdrom 共享。



图 21-6 从 Windows 系统访问 Linux 下的 Samba 共享

要从 Linux 系统中连接 Samba 共享, 可以从 shell 提示下, 输入以下命令:

```
smbclient //主机名/共享名 -U 用户名
```

其中“主机名”为需要连接的 Samba 服务器的主机名或 IP 地址, “共享名”为需要连接的共享目录名, “用户名”为服务器设置的 Samba 用户名。然后会提示用户输入口令, 输入口令后按回车键继续, 如果不需要口令, 则直接按回车键。访问 cdrom 共享的实例如下:

```
[globus@jiang globus]$ smbclient //jiang/cdrom -U globus
added interface ip=192.168.1.190 bcast=192.168.1.255 nmask=255.255.255.0
Password:
Domain=[SKOIS] OS=[Unix] Server=[Samba 2.2.7a]
smb: \>
```

连接成功后, 将会出现 smb:> 提示符。在 Smba 提示后, 键入 help 能获得一个命令列表。将“共享名”替换成你的用户名, 它会浏览你的主目录里的内容。如果省略 -U 选项, smbclient 将使用当前用户。smbclient 提供的命令与 ftp 命令类似。

在 smb:> 提示符后输入 exit, 可以退出 smbclient 程序。

用户也可以在 Linux 中使用 nautilus 或者 Konqueror 查看网络上可用的 Samba 共享。在

命令提示符中输入 `nautilus`，然后按回车键，将会启动 Nautilus。用户可以在 `nautilus` 的“位置”栏里输入 `smb:` 来查看工作组。

如图 21-7 所示，在网络上每个可用 SMB 工作组旁边都会出现一个图标。

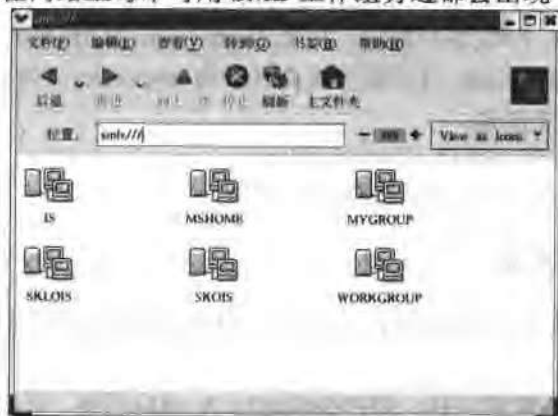


图 21-7 在 `nautilus` 中查看 SMB 工作组

双击工作组中的图标，来查看对应工作组内的计算机列表。如图 21-8 所示，工作组内每台计算机都有一个图标。双击某个图标来查看该机器上的 Samba 共享。如果需要，`nautilus` 会提示用户输入用户名和口令。

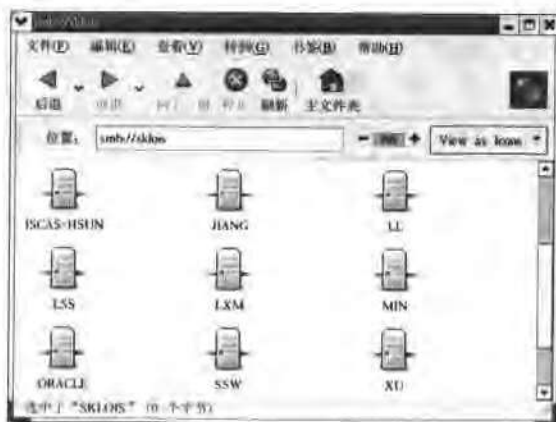


图 21-8 在 `nautilus` 中显示 SMB 工作组中包括的计算机列表

用户也可以在“位置”栏内使用以下格式来提供用户名和口令，来打开共享：  
`smb://用户名:口令@服务器主机名/共享名/`

## 21.2 与 Unix/Linux 共享文件：NFS

NFS 对于在同一网络上的多个用户间共享目录很有用途。例如，一个工程项目组的用户可以通过使用 NFS 文件系统（通常称为 NFS 共享）中的一个挂载为 `/projectdir` 的共享目录来存取该项目的文件。要存取共享的文件，用户进入各自机器上的 `/myproject` 目

录。NFS 共享非常方便，使用这种方法既不用输入口令，也不用记忆特殊命令，该目录就仿佛位于用户的本地计算机上一样。

在本节提供的例子中，假定 NFS 服务器主机名为 `jiang`，其 IP 地址为 `192.168.1.190`，导出的目录为 `/projectdir`。NFS 客户机的主机名为 `unkown`，其 IP 地址为 `192.168.1.19`，挂接 NFS 服务器 `jiang` 上的 `/projectdir` 目录的挂接点目录为 `/myproject`。如果这两个目录不存在，则分别使用如下两条命令创建：

```
[root@jiang root]# mkdir /projectdir/
[root@unknown root]# mkdir /myproject
```

### 21.2.1 导出 NFS 文件系统

从 NFS 服务器中共享文件也称为导出目录。NFS 服务器配置工具可以用来把系统配置成 NFS 服务器。

要使用 NFS 服务器配置工具，用户必须运行 X 窗口系统，具备根特权，并且安装了 `redhat-config-nfs` 的 RPM 包。

有以下两种方法启动 NFS 服务器配置工具：

- 单击面板上的“主菜单”|“系统设置”|“服务器设置”|“NFS 服务器”。
- 在 shell 提示符后输入 `redhat-config-nfs` 命令，然后按回车键。

启动后的“NFS 服务器配置”工具的界面如图 21-9 所示。

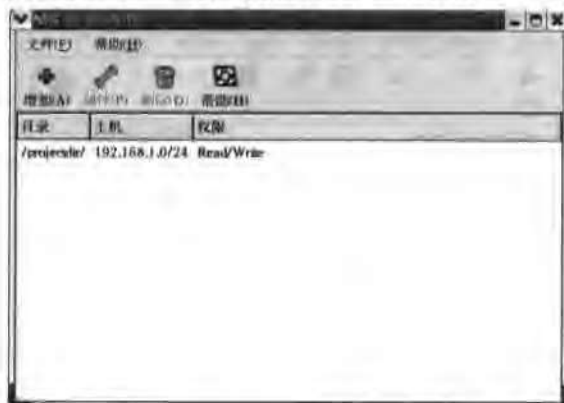


图 21-9 NFS 服务器配置工具

要添加 NFS 共享，单击“NFS 服务器配置”工具的工具栏上的“添加”按钮。如图 21-9 所示的“添加 NFS 共享”对话框会出现。

“添加 NFS 共享”对话框的“基本”选项卡包括以下信息：

- “目录”文本框。用来指定要共享的目录，如图 21-10 上的 `/projectdir`。
- “主机”文本框。指定要将目录共享给哪些主机。
- “基本权限”。设置 NFS 服务器的客户在访问该共享目录时，应该有只读权限还是具有读写权限。

指定要共享目录的主机时，可以使用以下格式：

- 单台机器。一个全限定域名、主机名或 IP 地址。

- 使用通配符指定的多台机器。使用“\*”或“?”字符来进行字符串匹配。通配符不能被用在 IP 地址中，因为这样会使反向 DNS 查询利用 IP 地址中的通配符。在全限定域名，点(.)不包括在通配符的匹配范围内。
- IP 子网。使用 n1.n2.n3.n4/z，这里的 n1.n2.n3.n4 是网络地址，z 是子网掩码的位数，如 192.168.1.0/24 表示 192.168.1.1-192.168.1.254 范围内的主机构成的子网。另一种可以接受的格式是 n1.n2.n3.n4/netmask，这里的 n1.n2.n3.n4 是网络，netmask 是子网掩码，如 192.168.1.0/255.255.255.0。

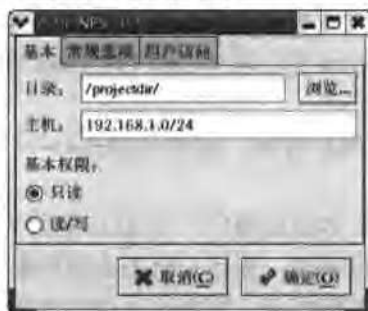


图 21-10 添加共享目录的“基本”选项卡

“添加 NFS 共享”的“常规选项”选项卡的“基本”选项卡如图 21-11 所示，包括以下选项：

- “允许来自高于 1024 的端口的连接”。在小于 1024 的端口上的服务必须以根用户身份启动。如果选中这个选项，那么普通用户将可以启动 NFS 服务。该选项与配置文件的 insecure 相对应。
- “允许不安全的文件锁定”。该选项和 insecure\_locks 相对应，它表明不需要锁定请求。

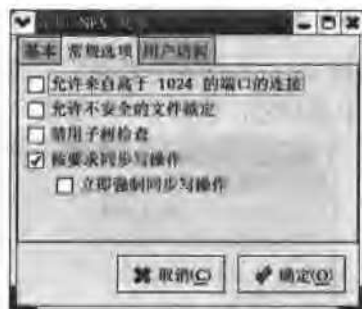


图 21-11 添加 NFS 共享目录的“常规选项”选项卡

- “禁用子树检查”。该选项和 no\_subtree\_check 相对应。如果某文件系统的子目录被导出，但是整个文件系统却没有被导出，服务器会检查所请求的文件是否在导出的子目录中。这种检查被称作子树检查(subtree checking)。如果选择该选项，将禁用子树检查。如果整个文件系统被导出，选择禁用“子树检查”可以提高传输率。

- “按要求同步写操作”。默认被启用，该选项作用是：只有当磁盘写操作完成之后，才给客户发送回复信息。该选项和 `sync` 相对应，如果不选择该选项，则 `async` 选项会被使用。
- “立即强制同步写操作”。不推迟写入磁盘的操作，该选项和 `no_wdelay` 相对应。

默认情况下，NFS 服务的远程客户以其他用户身份访问 NFS 导出的目录。举例来说，如果 NFS 服务器导出了目录 `/projectdir`，它的目录属性如下：

```
[root@jiang root]# ls -ld /projectdir/
drwxr-xr-x  2 root  root    4096  6月 17 14:26 /projectdir/
```

那么对于主机名为 `unknown` 的主机中的任何非 `root` 和 `root` 组的用户对该 `/projectdir/` 只有读 (`-x`) 权限。

“添加 NFS 共享”的“用户访问”选项卡如图 21-12 所示，它可以改变默认的权限对应关系，包括以下选项：

- “把远程根用户当作本地根用户”。注意这里的“远程根用户”表示客户机上的根用户。如果这个选项被选，根用户就不会被映射为匿名用户，客户上的根用户就会对导出的目录拥有根特权。选择这个选项会大大降低系统的安全性。除非绝对必要，请不要选择它。该选项和 `no_root_squash` 相对应。
- “把所有客户用户当作匿名用户”。如果该选项被选，所有用户和组群 ID 都会被映射为匿名用户。该选项和 `all_squash` 相对应。根权限压缩 (Root squashing) 把用户 ID 为 0 和组群 ID 为 0 映射为匿名的用户和组群 ID，因此客户上的根用户就不会在 NFS 服务器上具备根特权。
- “为匿名用户指定本地用户 ID”。如果选中“把所有客户用户当作匿名用户”，这个选项为匿名用户指定一个用户 ID。该选项和 `corresponds to anonuid` 相对应。
- “为匿名用户指定本地组群 ID”。如果选中“把所有客户用户当作匿名用户”，这个选项会让用户为匿名用户指定一个组群 ID。该选项和 `corresponds to anongid` 相对应。

要编辑 NFS 共享，从如图 21-9 的列表中选择它，然后单击“属性”按钮。要删除某个存在的 NFS 共享，从列表中选择它，然后单击“删除”按钮。



图 21-12 添加 NFS 共享目录的“用户访问”选项卡

单击“确定”按钮来从列表中添加、编辑、或删除某个 NFS 共享后，改变就会立即生效，服务器守护进程被重新启动，原有的配置文件被保存为 `/etc/exports.bak`。新的配置文件

被写入`/etc/exports`。

NFS 服务器配置工具直接读写`/etc/exports` 配置文件。因此，这个文件在使用该工具后可以被手工修改，手工修改了该文件后也可以使用这个工具再进行进一步配置。

### 21.2.2 启动和停止 NFS 服务器

在导出 NFS 文件系统的服务器上，nfs 服务必须处于运行状态。

使用以下命令来查看 NFS 守护进程的状态：

```
# /sbin/service nfs status
```

使用以下命令来启动 NFS 守护进程：

```
# /sbin/service nfs start
```

使用以下命令停止 NFS 守护进程：

```
# /sbin/service nfs stop
```

要在引导时启动 nfs 服务，使用以下命令：

```
# /sbin/chkconfig --level 345 nfs on
```

### 21.2.3 在客户端挂接 NFS 文件系统

1. 使用 mount 命令在命令行挂接 NFS 文件系统。

使用 mount 命令来挂接另一台机器上的 NFS 文件系统（导出的 NFS 目录）。

例：将主机 jiang（192.168.1.190）上导出的/projectdir 目录挂接到主机 unknown（192.168.1.19）的/myproject 目录。命令如下：

```
# mount 192.168.1.190:/projectdir /myproject
```

在该命令中，192.168.1.190 是 NFS 文件服务器的主机名；/projectdir 是 192.168.1.190 要导出的文件系统，/myproject 是该文件系统在本机上的挂接位置。mount 命令运行之后，客户用户可以执行 ls /myproject 命令来显示 192.168.1.190 上的/projectdir 目录中的文件列表。

2. 使用/etc/fstab 来挂接 NFS 文件系统

要挂接其他机器上的 NFS 共享的另一种方法是：在客户机的/etc/fstab 文件中添加一行。这一行中必须声明 NFS 服务器的主机名、要导出的目录以及要挂接 NFS 共享的本地目录。注意只有根用户才有权修改/etc/fstab 文件。

/etc/fstab 中 NFS 行的实例如下：

```
192.168.1.190:/projectdir /myproject nfs wsize=8192,timeo=14,intr
```

挂接点/myproject 在客户机上必须存在。在客户系统的/etc/fstab 文件中把这一行添加完后，在 shell 提示符下输入命令 mount /myproject。

3. 使用 autofs 来挂接 NFS 文件系统

挂接 NFS 共享的第三种方法是使用 autofs。autofs 使用 automount 守护进程来管理系统的挂接点，它只在文件系统被访问时才动态地挂接。

autofs 守护进程查询主映射配置文件/etc/auto.master 来决定要定义哪些挂接点。然后，它使用适用于各个挂接点的参数来启动 automount 进程。主映射配置中的每一行都定义一个挂接点，一个分开的映射文件定义在该挂接点下要挂接的文件系统。例如，/etc/auto.misc 文件可以定义/myproject 目录中的挂接点，这种关系在/etc/auto.master 文件中会被定义。

auto.master 文件中的每一行包含 3 个字段。第一个字段是挂接点，第二个字段是映射文件的位置，第三个字段可选。第三个字段可以包括超时数值之类的信息。

例：在主机名为 unknown 的机器上的/project/myproject 挂接点上挂接远程机器 192.168.1.190 上的/projectdir 目录。在 auto.master 文件中添加如下信息。

```
/project /etc/auto.misc --timeout=60
```

在/etc/auto.misc 文件中添加如下信息：

```
myproject1 -rw,soft,intr,rsize=8192,wsiz=8192 192.168.1.190:/projectdir
```

/etc/auto.misc 中的第一个字段是 myproject1 子目录的名称。该目录被 automount 动态地创建。它不应该在客户机上实际存在。第二个字段包括挂接选项，如 rw 代表读写访问权。第三个字段是要导出的 NFS 目录的位置，包括主机名和目录。

注意：在本地文件系统中/project 目录必须存在，而且它不应该有子目录。

autofs 是一种服务。要启动这项服务，在 shell 提示符下输入以下命令：

```
# /sbin/service autofs restart
```

要查看活跃的挂接点，在 shell 提示符下键入以下命令：

```
# /sbin/service autofs status
```

如果在 autofs 运行时修改了/etc/auto.master 配置文件，则必须在 shell 提示符下输入以下命令来通知 automount 守护进程重新载入配置文件：

```
# /sbin/service autofs reload
```

## 思考与练习

- (1) 什么是 Samba 协议？在 Linux 下 Samba 服务器的功能是什么？
- (2) 如何配置文件共享和打印机共享？
- (3) NFS 服务器的作用是什么？
- (4) 服务器如何导出 NFS 文件系统？客户端如何挂接 NFS 文件系统？

## 第 22 章 架设邮件服务器

在 Internet 出现 WWW 服务之前,电子邮件系统曾经是使用最为广泛的一种应用服务。电子邮件为人们的沟通带来了方便。现在电子邮件仍然是个人用户和企业用户不可缺少的联系方式,它为人们的沟通提供了许多方便。

Sendmail 是 Linux/UNIX 下的邮件服务器软件,它的功能非常强大,同时也是配置最为复杂的软件,Red Hat Linux 可以默认安装 Sendmail 软件。

本章首先介绍了电子邮件的工作原理、流行的电子邮件协议和电子邮件系统的组成,然后介绍了如何安装和配置 Sendmail 软件,最后简要介绍了使用 M4 配置 Sendmail 的步骤和配置过程。

### 22.1 电子邮件简介

#### 22.1.1 电子邮件系统的工作原理

电子邮件系统的运作方式与其他网络应用有着根本上的不同。在其他的绝大多数的网络应用中,网络协议直接负责将数据发送到目的地。在电子邮件系统中,发送者并不等待发送工作完成,而是仅仅将要发送的内容发送出去。

文件传输协议(FTP)就像打电话一样,实时地接通对话双方,如果一方暂时没有应答,则通话就会失败。电子邮件系统则不同,发送方将要发送的内容通过自己的电子邮局将信件发给接收方的电子邮局。如果接收方的电子邮局暂时繁忙,那么发送方的电子邮局就会暂存信件,直到可以发送。当接收方未上网时,接收方的电子邮局就暂存信件,直到接收方去取。可以这么说,电子邮件系统就像是在 Internet 上实现了传统邮局的功能,而且是更加快捷方便地实现。

现在作为一个普通的用户想要连接到 Internet,那么就必须找一家 Internet 服务商(ISP)提供连接服务。在中国,最大的 ISP 就是中国电信。ISP 在提供连接的同时,还会提供一个电子邮局,分配一个电子信箱给用户使用。在 Internet 上还有许多免费的电子邮局提供电子邮件服务。

值得一提的是,电子邮件有一个空间大小的限制,这个空间就是电子邮局给每个用户所提供的暂存信件的空间。

#### 22.1.2 电子邮件的标准

电子邮件的协议标准是 TCP/IP 协议族的一部分。它规定了电子邮件的格式和在邮局间交换电子邮件的协议。

每个电子邮件都分为两部分:邮件头和邮件内容。TCP/IP 对电子邮件的邮件头的格式作了确切的规定,而将邮件内容的格式让用户自定义。在邮件头中最重要的两个组成部分就是发送者和接收者的电子邮件地址。电子邮件地址的格式如下:



用户名@电子邮局域名

例: henry@sohu.com

电子邮件的传输协议（也就是在邮局间交换电子邮件的协议）主要有 SMTP（简单邮件传输协议）、POP（电子邮局协议）以及现在新兴的 IMAP（互联网邮件应用协议）。

### 22.1.3 电子邮件系统的组成

整个电子邮件应用系统由两大部分构成：

- 电子邮局系统；
- 电子邮件发送、接收系统。

电子邮件发送、接收系统则像遍及千家万户的邮箱，发送者和接收者通过它从电脑中发送和接收邮件。这个部分是一个运行在电脑中的客户端程序，最常用的有 Microsoft 的 Outlook Express, Foxmail 和 Linux 中的 mutt 等。用户可以根据自己的喜爱来选择不同的程序。从根本上说，它们实现的功能是一样的。

电子邮局行使着像传统邮局的功能，它在发送者和接收者之间起着一个桥梁作用。它是运行在电子邮局服务器上的一个服务器端程序。最常用的有 Microsoft 的 IIS 和 sendmail 等。在这里我们介绍的就是功能强大的、免费的，基于类 UNIX 操作系统的电子邮件服务端程序 sendmail 的最基本的配置与使用方法。

## 22.2 安装和启动 Sendmail

### 22.2.1 安装 Sendmail 软件

如果是完全安装 Red Hat Linux 9，那么系统已经安装了 Sendmail 8.12.8-4 服务器。如果不能确定是否已经安装 sendmail，可以在终端命令窗口输入如下命令：

```
[root@wangwang root]# rpm -q sendmail  
sendmail-8.12.8-4
```

如果结果显示为 sendmail-8.12.8-4，则说明系统已经安装 sendmail 服务器。如果安装 Red Hat Linux 9 时没有选择 Sendmail 服务器，则可以在图形环境下单击面板的“主菜单”|“系统设置”|“添加删除应用程序”菜单项，在打开的“软件包管理”对话框中确保选中“邮件服务器”选项，然后单击“更新”按钮，按照屏幕提示插入第一张安装光盘即可开始安装。

另外，用户也可以直接插入第一张安装光盘，定位到/RedHat/RPMS 目录（它包含 sendmail-8.12.8-4.i386.rpm 包），然后在终端命令窗口运行以下命令即可开始安装进程：

```
[root@wangwang root] # rpm -ivh sendmail-8.12.8-4.i386.rpm
```

用类似的方法安装 sendmail-cf.8.12.8-4.i386.rpm、sendmail-doc. 8.12.8-4.i386.rpm，它们位于第三张安装光盘的/RedHat/RPMS 目录下。

邮件服务器是一个企业所必须的服务，在这里介绍 Red Hat Linux 9 自带的 Sendmail 来架设邮件服务器。

## 22.2.2 启动/重新启动/停止 Sendmail 服务

安装 Sendmail 服务器以后, 最简单的启动方式是在终端命令窗口运行如下命令:

```
[root@wangwang root]# /sbin/service sendmail start
```

应该会出现以下结果, 表明邮件服务器已经启动成功:

```
启动 sendmail: [ 确定 ]
```

```
启动 sm-client: [ 确定 ]
```

除以上方式, 还可以使用带参数的 Sendmail 命令控制邮件服务器的运行, 例如:

```
[root@wangwang root]# sendmail -bd -q15m
```

Sendmail 的命令参数的含义如下所示。

- **-b:** 指定 Sendmail 在后台运行, 并且监听端口 25 的请求。
- **-d:** 指定 Sendmail 以 Daemon 方式运行 (守护进程)。
- **-q:** 当 Sendmail 无法将邮件成功地发送到目的地时, 它会将邮件保存在队列里。

该参数指定邮件在队列里保存的时间。例子中的 15m 表示保留 15 分钟。

在终端命令窗口运行以下命令来重新启动 Sendmail 服务:

```
[root@wangwang root]# /sbin/service sendmail restart
```

在终端命令窗口运行以下命令来关闭 Sendmail 服务:

```
[root@wangwang root]# /sbin/service sendmail stop
```

也可以在终端命令窗口运行以下命令来检测 Sendmail 服务器的运行状态:

```
[root@wangwang root]# /sbin/service sendmail status
```

如果 Sendmail 已启动, 那么系统应该显示:

```
sendmail (pid 2782) 正在运行...
```

使系统引导时自动启动 Sendmail 服务, 在终端命令窗口运行命令如下:

```
[root@wangwang root]# chkconfig --level 345 sendmail on
```

## 22.3 配置 Sendmail

Sendmail 的配置十分复杂。它的配置文件是 `sendmail.cf`, 位于 `/etc/mail` 目录下。由于 `sendmail.cf` 的语法深奥难懂, 很少有人会直接去修改该文件来对 Sendmail 服务器进行配置。一般方式是通过 m4 宏处理程序来生成所需的 `sendmail.cf` 文件。创建的过程中还需要一个模板文件, 系统默认在 `/etc/mail` 目录下有一个 `sendmail.mc` 模板文件。

### 22.3.1 配置 Sendmail 的一般步骤

用户可以根据简单、直观的 `sendmail.mc` 模板来生成 `sendmail.cf` 文件, 而无需直接编辑 `sendmail.cf` 文件。可以直接通过修改 `sendmail.mc` 模板来达到定制 `sendmail.cf` 文件的目的。这里介绍创建 `sendmail.cf` 文件的步骤:

(1) 备份原有 `sendmail.cf` 文件。

在终端命令窗口执行以下命令:

```
# cp /etc/mail/sendmail.cf /etc/mail/sendmail.cf.original
```

(2) 生成 `sendmail.cf` 文件。

根据 `sendmail.mc` 模板文件产生 `sendmail.cf` 配置文件,并导出到 `/etc/mail/` 目录下,在 shell 提示符后执行如下命令:

```
# m4 /etc/mail/sendmail.mc > /etc/mail/sendmail.cf
```

(3) 重启 `sendmail` 服务。

```
[root@wangwang root]# /sbin/service sendmail restart
```

### 22.3.2 根据情况修改 `sendmail.mc` 文件

显然,用 `m4` 宏编译工具创建 `sendmail.cf` 文件比较方便、不容易出错,而且可以避免某些带有安全漏洞或者过时的宏所造成的破坏。一个 `sendmail.mc` 模板的大致内容如下:

```
divert(-1)dnl
.....
include('/usr/share/sendmail-cf/m4/cf.m4')dnl
VERSIONID('setup for Red Hat Linux')dnl
OSTYPE(linux')dnl
.....
dnl #
dnl define('SMART_HOST', 'smtp.your.provider')
dnl #
define('confDEF_USER_ID', '8:12')dnl
define('confTRUSTED_USER', 'smmsp')dnl
dnl define(' confAUTO_REBUILD' )dnl
.....
```

这里介绍 `sendmail.mc` 模板的语法组成。

- `dnl`: 用来注释各项,同时 `dnl` 命令还用来标识一个命令的结束。
- `divert (-1)`: 位于 `mc` 模板文件的顶部,目的是让 `m4` 程序输出时更加精简一些。
- `OSTYPE ('OperationSystemType')`: 定义使用的操作系统类型,显然这里应该用 `linux` 代替 `OperationSystemType`,注意要用一个反引号和一个正引号把对应的操作系统类型括起来。
- `define`: 定义一些全局设置,对于 `Linux` 系统,设置了 `OSTYPE` 之后,可以定义下面的一些全局参数,如果不定义,就使用默认值。这里举两个简单例子:`define ('ALIAS_FILE', '/etc/aliases')`,表示定义别名文件(`ALIAS_FILE`)的保存路径,默认是 `/etc/aliases`; `define ('STATUS_FILE', '/etc/mail/statistics')`,表示 `sendmail` 的状态信息文件。

### 22.3.3 为新用户开设电子邮件账号

在 `Linux` 里为新用户开设电子邮件账户比较简单,只需在 `Linux` 系统里新增一个用户即可。单击“主菜单”|“系统设置”|“用户和组群”命令,打开“Red Hat 用户管理器”对话框,单击“添加用户”按钮,在出现的“创建新用户”对话框里指定用户名、登录口

令即可。

假设添加了一个用户 `henryr` (密码为 `good`)，这样该用户就有了一个邮件地址 `henry@DomainName.com` (此处的 `DomainName.com` 用自己域名代替)。

以上过程也可以在终端命令窗口运行以下命令来实现：

```
[root@wangwang root]# useradd henry -p good
```

#### 22.3.4 为电子邮件账户设置别名

有一些用户想使用多个电子邮件地址，可以使用别名 (alias) 来避免开多个邮件账号。

例：用户 `henry` 想拥有以下 3 个电子邮件地址

`henry@DomainName.com`、`hen@DomainName.com` 和 `henrygood@DomainName.com` 可以通过以下步骤来实现这样的别名设置。

新增一个账号 `henry`；然后用 `vi` 文本编辑器打开 `/etc/aliases`，在里面加上如下两行：

```
hen: henry
```

```
henrygood: henry
```

保存该文件 `/etc/aliases`，然后退出。

仅这样还不能让 `Sendmail` 接受新增的别名，必须在终端命令窗口运行 `newaliases` 命令，以要求 `Sendmail` 重新读取 `/etc/aliases` 文件。如果一切无误，应该可以看到类似以下的回应消息：

```
[root@wangwang root]# newaliases
```

```
/etc/aliases: 65 aliases, longest 10 bytes, 647 bytes total
```

这样，发给 `henry` 的邮件可以使用 3 个邮件地址，而 `henry` 只需要使用一个电子邮件账号 `henry@DomainName.com` 就可以接收所有寄给以上 3 个地址的电子邮件。

#### 22.3.5 指定邮箱容量限制

当一个邮件服务器为许多人提供邮件服务时，无限量的电子邮件将很容易塞满服务器的硬盘，造成系统拒绝服务。如果不想为用户提供无限空间的邮件暂存空间，可以使用“邮件限额”来给用户一个有限的暂存空间。

其实，它是利用磁盘配额功能来实现的。电子邮件的暂存空间是在 `/var/spool/mail` 目录下，只要通过磁盘配额设定每一个用户在这个目录下能使用的最大空间即可。

#### 22.3.6 支持 POP 和 IMAP 功能

到现在为止，用户已经可以用 `Outlook Express` 发送邮件，或者登录服务器使用 `mail`、`mutt` 命令收取和管理邮件。但是还不能用 `Outlook Express` 等客户端从服务器下载邮件，这是因为 `Sendmail` 并不具备 `POP3` (`IMAP`) 的功能，所以我们必须自己安装。

##### 1. POP 和 IMAP 服务器安装

安装 `Red Hat Linux 9` 时，可以选择安装 `POP` 和 `IMAP` 服务器，可以在终端命令窗口运行以下命令进行验证：

```
[root@wangwang root]# rpm -q imap
imap-2001a-18
```

如果没有安装，可以将第二张安装光盘放进光驱，然后在终端命令窗口运行以下命令即可开始安装：

```
[root@wangwang root]# cd /mnt/cdrom/RedHat/RPMS
[root@wangwang root]# rpm -ivh imap-2001a-18.i386.rpm
```

由于 Red Hat Linux 9 已经将 POP 和 IMAP 打包成一个单独的套件，安装好 `imap-2001a-18.i386.rpm`，就会同时安装这两个服务器

## 2. 启动 POP 和 IMAP 服务

要成功启动 POP 和 IMAP 服务器，首先要确定这些服务存在于 `/etc/services` 文件，使用如下两条命令分别对 IMAP 和 POP3 进行修改，并在系统引导时启动：

```
[root@wangwang root]# chkconfig --level 345 imap on
[root@wangwang root]# chkconfig --level 345 ipop3 on
```

修改好 `/etc/services` 文件，接下来就要对相应服务配置文件进行定制：

启动 POP3 服务：必须修改 `/etc/xinetd.d/ipop3` 文件，将其中的 `disable=yes` 改为 `disable=no`，保存该文件。最后必须重新启动 `xinetd` 程序来读取新的配置文件，使得设定内容生效：

```
[root@wangwang root]# service xinetd restart
```

停止 xinetd: [ 确定 ]

启动 xinetd: [ 确定 ]

启动 IMAP 服务：必须修改 `/etc/xinetd.d/imap` 文件，将其中的 `disable = yes` 改为 `disable =no`，保存该文件。最后必须重新启动 `xinetd` 程序来读取新的配置文件，使得设定内容生效：

```
[root@wangwang root]# service xinetd restart
```

停止 xinetd: [ 确定 ]

启动 xinetd: [ 确定 ]

设置好以后，就可以在 Outlook Express 里面设置好邮件账号，以后就可以直接利用 Sendmail 服务器进行收发邮件了。由于邮件服务器就在本地局域网，所以速度很快。

## 思考与练习

- (1) 一个完整的电子邮件系统由哪几部分组成？
- (2) Sendmail 支持哪些邮件功能？
- (3) 常用的邮件客户端软件有哪些？
- (4) 完整的邮件系统包括哪些协议？它们的作用是什么？

## 第 23 章 架设 FTP 服务器

FTP（文件传输）服务提供了互联网上高效的文件传输服务。尽管 SSH 提供了安全的文件传输服务，但是由于数据是加密传输的，因此效率较低，而且需要严格的认证。相对而言，FTP 服务可以用于提供公开的文件传输服务，例如开放源码的软件，公司内部使用的公共文件。

Red Hat Linux 9 安装盘中自带的 vsftp 是一种容易使用、控制灵活的 FTP 服务器软件，能够胜任企业和大型组织的文件传输服务。

本章将从使用的角度介绍配置和使用 vsftp 服务器，同时也给出了一些配置实例，以供用户使用时参考。

### 23.1 安装 vsftp 服务器软件

#### 23.1.1 vsftp 简介

FTP（file transfer protocol，文件传输协议）是一种传送文件最常用的方式。vsftp 是一种安全与效能兼备的 FTP 服务器软件。

在 Red Hat Linux 9 中，FTP 服务器软件与原来的 Red Hat Linux 版本相比，发生了如下两大变化：

- FTP 服务器软件只剩下 vsftp，原有的 wu-ftp 被抛弃。
- vsftp 从 xinetd 服务中独立出来了，并将配置文件从 `/etc/vsftpd.conf` 转移到了 `/etc/vsftpd/vsftpd.conf`。

这表明 vsftp 已有独立运行的能力，不再需要 xinetd 服务来做更进一步的管理，并且类似 sendmail、httpd、ssh 和 samba 等服务，将配置文件的放入 `/etc` 下的独立子目录中。

FTP 文件传输式有两种传输模式。一种为 PORT 方式，它是早期的 FTP 软件提供的惟一方式。另一种是 PASV 方式，它是现在流行的一种方式。

##### （1）PORT 方式。

这是一般形式的 FTP，它首先会建立控制连接，默认的连接端口号为 21，并透过此连接发送 FTP 控制命令和返回响应。当需要传输服务器上的目录列表和传输文件时，这时由 FTP 服务器端建立一个新的数据连接，默认端口号为 20，并通过端口进行数据传输。因此进行 FTP 会话时，通常需要建立控制和数据两个连接。

##### （2）PASV（被动）方式。

它与 PORT 方式类似，首先会建立控制连接，默认的连接端口号为 21，并透过此连接发送 FTP 控制命令和返回响应。与 PORT 方式不同的是，由客户端做出数据传输的请求，包括数据连接的端口号

其差异主要表现在：PORT 方式数据连接的端口号是由 FTP 服务器指定的，而 PASV 方式数据连接端口由 FTP 客户决定。客户通常使用 PASV 方式，这样在防火墙环境，可以

由客户决定数据连接的端口号，这样会更安全。

### 23.1.2 安装 vsftpd

Red Hat Linux 9 自带的 vsftpd 的版本为 1.1.3-8。如果在安装系统时已经安装了 vsftpd，那么使用 rpm 查询已安装的 RPM 包，应该有如下结果：

```
[root@jiang mnt]# rpm -q vsftpd
vsftpd-1.1.3-8
```

结果表明系统已经安装了 vsftpd 服务器软件。如果用户的机器没有安装这个包，使用如下命令安装这些包（这个包在 Red Hat Linux 9 第三张安装盘的/RedHat/RPMS 目录下）：

```
[root@jiang mnt]# rpm -ivh vsftpd-1.1.3-8.i386.rpm rpm ivh
```

### 23.1.3 启动和停止 vsftpd

vsftpd 服务器在 Linux 中的服务名被称作 vsftpd。

如果需要启动 vsftpd 服务器，则使用如下命令：

```
[root@jiang mnt]# service vsftpd start
```

如果需要停止 vsftpd 服务器，则使用如下命令：

```
[root@jiang mnt]# service vsftpd stop
```

如果需要 vsftpd 服务器重新加载配置文件，则使用如下命令：

```
/sbin/service vsftpd reload
```

让 vsftpd 服务器开机后自动启动，命令如下：

```
chkconfig --level 345 vsftpd on
```

## 23.2 使用和设置 vsftpd

本节将以实例的形式讲述 vsftpd 服务器的使用和配置。首先讲述 vsftpd 的默认设置和最基本的配置，然后讲述如何设置基本的访问权限，接着介绍匿名服务器的下载和上传配置，最后讲述更高级的访问控制。

### 23.2.1 vsftpd 服务器的默认设置

在 Red Hat Linux 9 中，vsftpd 服务器的配置文件为/etc/vsftpd/vsftpd.conf。为了改变 vsftpd 服务器的行为，必须手工修改该文件。

在 Red Hat Linux 9 中，vsftpd 服务器的主要的默认设置如下：

- 允许匿名用户和本地用户登录。
- 匿名用户的登录名为 ftp 或 anonymous，口令为一个 E-mail 地址。
- 匿名用户不能离开匿名服务器目录/var/ftp，且只能下载文件，而不能上传。
- 本地用户的登录名为本地用户名，口令为此本地用户的口令。
- 本地用户可以离开其主目录，切换至有权访问的其他目录，并可以在权限允许的情况下进行上传/下载。
- 列在文件/etc/vsftpd.ftpusers 中的本地用户禁止 ftp 登录。

- 匿名用户可以在 pub (FTP 服务器的/var/ftp/pub 子目录) 子目录下下载文件。

通过如下步骤来分析和验证其默认配置。

- (1) 启动 vsftpd 服务器, 命令如下:

```
[root@jiang mnt]# service vsftpd start
```

- (2) 将当前目录的文件 parted-1.6.9.tar.gz 复制到 ftp 匿名用户的公共目录, 命令如下:

```
[root@jiang tools]# cp parted-1.6.9.tar.gz /var/ftp/pub
```

- (3) 在 IP 地址为 192.168.1.19 的主机上, 使用 ftp 命令匿名登录该服务器。

```
[globus@unknown globus]$ ftp 192.168.1.190
```

```
Connected to 192.168.1.190 (192.168.1.190). #服务器的IP地址
```

```
220 (vsFTPd 1.1.3) #服务器的类型及版本
```

```
Name (192.168.1.190:globus): anonymous #使用匿名登录
```

```
331 Please specify the password. #提示输入一个电子邮件地址
```

```
Password:
```

```
230 Login successful. Have fun. #登录成功
```

```
Remote system type is UNIX.
```

```
Using binary mode to transfer files. #二进制文件传输方式
```

```
ftp> cd pub #输入远程改变目录命令
```

```
250 Directory successfully changed.
```

```
ftp> ls #显示服务器当前目录文件列表
```

```
227 Entering Passive Mode (192,168,1,190,74,28) #使用被动模式连接
```

```
150 Here comes the directory listing. #结果表明有文件parted-1.6.9.tar.gz
```

```
-rw-r--r-- 1 0 0 1380370 Jun 18 08:25 parted-1.6.9.tar.gz
```

```
226 Directory send OK.
```

从服务器上下载文件 parted-1.6.9.tar.gz, 过程如下:

```
ftp> get parted-1.6.9.tar.gz #下载文件parted-1.6.9.tar.gz
```

```
local: parted-1.6.9.tar.gz remote: parted-1.6.9.tar.gz
```

```
227 Entering Passive Mode (192,168,1,190,140,206)
```

```
150 Opening BINARY mode data connection for parted-1.6.9.tar.gz (1380370 bytes).
```

```
226 File send OK.
```

```
1380370 bytes received in 0.267 secs (5e+03 Kbytes/sec)
```

如果这时上传文件 (如 hello3.txt), 会出现如下错误:

```
ftp> put hello3.txt #执行文件上传命令
```

```
local: hello3.txt remote: hello3.txt
```

```
421 Timeout.
```

```
Passive mode refused. #表明匿名用户不能上传文件
```

(4) 在 IP 地址为 192.168.1.19 的主机上, 使用 ftp 命令以 globus 用户身份登录该服务器。这时用户名应该是 globus, 口令是 globus 用户在服务器 (IP 地址为 192.168.1.190) 上的本地用户 globus 的口令 (即 vsftp 使用了 Linux 系统中的用户和密码)。



当登录成功后，默认的目录使用 globus 用户的主目录，可以使用 ls 命令验证如下：

```
ftp> ls                                #显示服务器的当前目录文件列表
227 Entering Passive Mode (192,168,1,190,58,241)
150 Here comes the directory listing.
drwx----- 3 500  500  4096 Jun 17 09:13 Desktop
lrwxrwxrwx  1 500  500    12 Jun 15 05:49 globus -> /home/globus
-rw-rw-r--  1 500  500    8 Jun 16 07:40 hello.aux
-rw-rw-r--  1 500  500  268 Jun 16 07:40 hello.dvi
-rw-rw-r--  1 500  500 1905 Jun 16 07:40 hello.log
-rw-rw-r--  1 500  500   97 Jun 16 07:39 hello.tex
.....
226 Directory send OK.
```

上面结果表明，该命令已经是 globus 用户的主目录。用户可以在它有权限的目录中上传、下载和删除文件。

如果将 globus 用户加入到/etc/vsftpd.ftpusers 文件，那么 globus 将不能再登录 ftp 服务器。登录时，将会出现如下错误提示：

```
530 Login incorrect.
Login failed.
```

### 23.2.2 vsftp 服务器的基本配置

如下重要的基本配置都是修改/etc/vsftpd/vsftpd.conf 配置文件，vsftpd.conf 文件中的配置形式非常单一，每一行为一项设置。若是空行或者以#开头的行，将会被忽略。

**注意：**在每一次配置修改之后，如果让该配置生效，则需要 shell 提示符后，执行 service vsftpd reload 命令重新加载这些配置。

这些配置项分别如下：

(1) listen\_port。更改 ftp 服务器控制连接的端口号。

为了安全起见，将 ftp 服务器的端口号改为 8888。应该在/etc/vsftpd/vsftpd.conf 文件的最后添加一行，内容如下：

```
listen_port=8888
```

(2) chroot\_list\_enable。使指定用户（非匿名用户）不能切换到主目录以外的目录。用户的默认目录为其主目录，如果让用户不能改变到主目录以外的目录，步骤如下：

修改/etc/vsftpd/vsftpd.conf，将以下 3 行

```
#chroot_list_enable=YES
```

```
# (default follows)
```

```
#chroot_list_file=/etc/vsftpd.chroot_list
```

改为

```
chroot_list_enable=YES
```

```
# (default follows)
```

```
chroot_list_file=/etc/vsftpd/chroot_list
```

另外，新建一个文件 `/etc/vsftpd/chroot_list`，并添加如下一行：

```
globus
```

这样 `globus` 用户就不能切换到主目录以外的目录。

(3) `dirmessage_enable`。Red Hat Linux 9 默认设置允许欢迎信息，当用户登录时，能够看到欢迎话语，如果取消欢迎信息，则修改配置如下：

```
dirmessage_enable=NO
```

否则，（即取值 YES）可以在每一个目录下新增名为 `.message` 的文件，该文件包含欢迎信息。

(4) `setproctitle_enable`。对于每一个连接，都以独立的进程来运行。默认情况下，系统中只有一个 `vsftpd` 进程在运行，可以在 `/etc/vsftpd/vsftpd.conf` 添加如下一行，可以使每一个连接，都以独立的进程来运行：

```
setproctitle_enable=YES
```

(5) `ascii_download_enable`。控制是否可用 ASCII 模式下载。默认值为 NO。

(6) `ascii_upload_enable`。设置是否可用 ASCII 模式上传。默认值为 NO。

(7) `banner_file`。此选项当用户登录成功后，显示的欢迎信息或说明所在的文件内容，通常为默认值无。

(9) `ftpd_banner`。该选项定义欢迎信息的字符串，相比较而言，`banner_file` 选项设置的是文件的形式，而 `ftpd_banner` 是字符串的格式，默认值为无。

(10) `xferlog_enable=YES/NO`，取值 YES，`vsftpd` 会将上传与下载的信息完整的纪录 `xferlog_file` 所定义的日志文件中。默认值为 YES。

(11) `xferlog_file`，该选项设置日志文件文件名，默认值为 `/var/log/vsftpd.log`。

(12) `xferlog_std_format`，取值 YES 表示设置日志记录的格式 `xferlog` 的标准格式，默认值为 YES。

(13) `connect_from_port_20=YES/NO`。是否使用端口号 20，建立数据连接，默认值为 YES。

(14) `pasv_enable`。若设为 NO，则不允许使用 PASV 的模式建立数据连接。默认值为 YES。

### 23.2.3 vsftp 服务器的匿名用户设置

(1) `anonymous_enable`。拒绝 `anonymous` 用户登录，若是允许匿名登录，修改如下：

```
anonymous_enable=NO
```

`anonymous_enable` 默认值为 YES，表示允许匿名访问。

(2) `anon_max_rate`。限制匿名用户传输文件的速度。

例如，规定匿名用户的最高速度限制为 100KB/s。需要在 `/etc/vsftpd/vsftpd.conf` 文件中添加如下行：

```
anon_max_rate=100000
```

`anon_max_rate` 的单位是字节/秒。

(3) `anon_upload_enable=YES/NO`。是否允许匿名用户上传文件，默认值为 NO。

(4) `anon_mkdir_write_enable=YES`。是否允许匿名用户有创建目录的权利，默认值为

NO。

(5) `deny_email_enable=YES`。是否允许禁止匿名用户使用某些邮件地址。

(6) `banned_email_file=/etc/vsftpd.banned_emails`。输入禁止的邮件地址的路径和文件名。

(7) `anon_umask`。匿名用户创建文件时的 `umask` 值，默认值为 077。

(8) `no_anon_password`。若是启动这项功能，则使用匿名登录时，不会询问密码。默认值为 NO。

(9) `anon_other_write_enable`。如果设为 YES，匿名用户会拥有多于上传与建立目录之外的权限（如删除或更名）。默认值为 NO。

(10) `anon_world_readable_only`。如果设为 YES，匿名用户可以下载可阅读的文件。默认值为 YES。

(11) `ftp_username`。定义匿名登入的用户名称。默认值为 `ftp`。

#### 23.2.4 vsftpd 服务器的高级配置

(1) `local_max_rate`。限制本地传输文件的速度。例如，规定本地用户的最高速度限制为 300KB/s。需要在 `/etc/vsftpd/vsftpd.conf` 文件中添加如下行：

```
local_max_rate=300000
```

`local_max_rate` 的单位为字节/秒。

(2) `user_config_dir`。针对不同的用户限制不同的下载速度。

要设置 `globus` 用户所能使用的最高速度为 500KB/s，`jzh` 用户所能使用的最高速度为 200KB/s。首先应该对 `/etc/vsftpd/vsftpd.conf` 配置文件作添加如下行：

```
user_config_dir=/etc/vsftpd/userconf
```

然后新建一个目录 `/etc/vsftpd/userconf`，命令如下：

```
mkdir /etc/vsftpd/userconf
```

接着，在 `/etc/vsftpd/userconf` 目录下新建 `globus` 文件，并在文件中添加如下行：

```
local_max_rate=500000
```

最后，在 `/etc/vsftpd/userconf` 目录下新建 `jzh` 文件，并在文件中添加如下行：

```
local_max_rate=200000
```

(3) `userlist_enable=YES/NO`。若启用此功能（YES），则会读取 `/etc/vsftpd.user_list` 文件中的用户名称。此项功能可以在输入密码前就出现失败信息，而不需要检查密码。默认值为 NO。

(4) `userlist_deny=YES/NO`。该选项只有在 `userlist_enable` 启用时才会被验证。如果将这个选项设为 YES，则在 `/etc/vsftpd.user_list` 中的用户将无法登录，若取值为 NO，则只有在 `/etc/vsftpd.user_list` 中的用户才能登入。而且此项功能可以在输入密码前就出现错误讯息，而不需要检查密码。

(5) `tcp_wrappers`。vsftpd 与 TCP\_wrapper 结合。

若是读者希望直接在 `/etc/hosts.allow` 之中定义允许或是拒绝来源地址，可执行以下步骤。这是简易的防火墙设定。

首先在 `/etc/vsftpd/vsftpd.conf` 中将 `tcp_wrappers` 设定为 YES，结果如下：

```
tcp_wrappers=YES
```

这是 Red Hat Linux 9 的默认设置，不需修改。然后修改/etc/hosts.allow 文件。

例如，允许 192.168.1.19 和 192.168.1.188 两个 IP 地址访问 ftp 服务器，则修改/etc/hosts.allow 文件如下：

```
vsftpd : 192.168.1.19 192.168.1.188 : allow
ALL : ALL : DENY
```

(6) ls\_recurse\_enable=YES/NO。YES 表示启用此功能，允许用户使用 ls -R 指令。默认为 NO。

(7) write\_enable=YES/NO。这个选项可以控制是否启用文件写命令（它们包括 STOR、DELE、RNFR、RNT0、MKD、RMD、APPE 和 SITE）。默认为关闭。

(8) accept\_timeout。超时设置，接受建立连接的超时设定，单位为秒。默认值为 60。

(9) local\_enable=YES/NO。是否允许本地用户登录，默认值为 YES。

(10) local\_umask=022。默认的 umask 码，可以设置来改变建立文件后的默认文件权限。

(11) idle\_session\_timeout=600。设置会话的 idle 超时时间。

(12) data\_connection\_timeout=120。设置数据连接的超时时间。

(13) nopriv\_user=ftpsecure。运行 vsftpd 需要的非特权系统用户默认为 nobody。

(14) async\_abor\_enable=YES/NO。是否允许运行特殊的 ftp 命令 async ABOR。

(15) max\_clients=Number。如果以 standalone 模式起动，那么只有 Number 个用户可以连接，其他的用户将得到错误信息，默认是 0，表示不限制。

(16) message\_file。设置访问一个目录时获得的目录信息文件的文件名，默认值是.message。

(17) max\_per\_ip。若是 vsftpd 使用 standalone 的模式，可使用这个参数定义每个 IP 地址，可以同时建立连接的数目。超过这个数目将会拒绝连接，0 表示不限。默认值为 0。

(18) chown\_upload=YES/NO。是否改变上传文件的属主。

(19) chown\_upload=username。设置上传的文件的属主。

### 23.2.5 vsftpd 服务器配置实例

例：允许匿名 FTP，匿名用户允许上传，可以 chroot 用户目录，上传文件属主改为 root，每一个 IP 的最大连接数为 4，只允许 IP 地址为 192.168.1.1 至 192.168.1.254 之间计算机能登录使用该 FTP 服务器。

则该 vsftp 服务器的配置文件如下：

```
anonymous_enable=YES
local_enable=YES
write_enable=YES
local_umask=022
anon_upload_enable=YES
anon_mkdir_write_enable=YES
dirmessage_enable=YES
```

```
xferlog_enable=YES
connect_from_port_20=YES
chown_uploads=YES
chown_username=root
xferlog_file=/var/log/vsftpd.log
xferlog_std_format=YES
pam_service_name=vsftpd
chroot_local_user=YES
max_per_ip=4
tcp_wrappers=YES
```

修改/etc/hosts.allow 文件，修改后内容如下：

```
vsftpd : 192.168.1. : allow
ALL : ALL : DENY
```

## 思考与练习

- (1) FTP 服务器的功能是什么？
- (2) 如何安装、启动和停止 vsftp 服务器？
- (3) 在 vsftp 服务器中，如何设置用户的权限？
- (4) 在 vsftp 服务器中，有哪些性能调节参数？

## 第 24 章 构建防火墙

Linux 具有许多内置的能力，使开发人员可以根据自己的需要定制其工具、行为和外观，而无需昂贵的第三方工具。如果 Linux 系统连接到因特网或 LAN、服务器或连接 LAN 和因特网的代理服务器，所要用到的一种内置能力就是针对网络上 Linux 系统的防火墙配置。2.4.x 版本的 Linux 内核中已经集成了 netfilter/iptables IP 包过滤系统，用户可以通过该包过滤系统提供强大的防火墙能力。

最新的 Linux 内核 2.4.x 具有 netfilter/iptables 系统这种内置的 IP 数据包过滤工具，它使配置防火墙和数据包过滤变得便宜且方便。netfilter/iptables 系统使其用户可以完全控制防火墙配置和数据包过滤。它允许为防火墙建立可定制化的规则来控制数据包过滤。它还允许配置有状态的防火墙。

本文讨论了防火墙原理和 Linux 2.4 版内核的防火墙机制 netfilter，并用实例介绍了 netfilter 支持的 IP 包过滤、NAT 和数据包处理（packet mangling）等功能，以及用它们构造安全系统的方法。

### 24.1 安装 iptables

#### 24.1.1 防火墙简介

防火墙是一种网络安全的防范技术和访问控制机制，通过一定的安全策略，防止非法入侵，只有被授权的通信才可以通过防火墙，因此它能有效地保护企业内部网络的安全。

防火墙系统的实现技术一般分为两种：包过滤（Packet Filtering）和代理服务（Proxy service）技术。包过滤技术在网络层对通过防火墙的数据包进行合法性分析和检查。代理服务也称为代理网关，是一个应用层的网关作为代理服务器。

建立适当的安全策略是保障网络安全的第一步，有两种可选策略：

（1）没有被列为允许访问的服务都是被禁止的。

（2）没有被列为禁止访问的服务都是被允许的。前一种安全策略是更严格的访问控制策略。

Linux 内核（Kernel）的一致性使得 Linux 产品之间的防火墙具有通用性。其防火墙的性能和专用的防火墙相当。在 Linux 2.0 和 Linux 2.2 内核中防火墙分别称为 ipforward 和 ipchains，它们的模块化程度低、规则复杂和无法提供传递数据包到用户空间。2.4 版内核中的防火墙被称为 netfilter（也称 iptables），它提供了一个抽象的、通用化的框架，使它比以前任何一版 Linux 内核的防火墙子系统都要完善强大。

#### 24.1.2 安装和启动防火墙

在 Red Hat Linux 9 中的标准安装中包含了 iptables 用户空间工具。但在默认情况下，这个工具是关闭的。为了使该工具运行，需要执行以下步骤：

首先设置系统启动时自动启动 iptables。在 shell 提示符后，执行如下命令：

```
# chkconfig --level 345 iptables on
```

然后，手工启动 iptables 服务。在 shell 提示符后，执行如下命令：

```
# service iptables start
```

## 24.2 Netfilter 防火墙系统框架

### 24.2.1 Netfilter 防火墙的特点

netfilter (iptables) IP 包过滤系统是一种功能强大的工具，可用于添加、编辑和除去规则，这些规则是在进行数据包过滤决定时，防火墙所遵循和组成的规则。这些规则存储在专用的包过滤表中，而这些表集成在 Linux 内核中。在包过滤表中，规则被分组放在所谓的链 (chain) 中。稍后将会详细讨论这些规则以及如何建立这些规则，并将它们分组在链中。

虽然 netfilter/iptables IP 包过滤系统被称为单个实体，但实际上它由两个组件 netfilter 和 iptables 组成。它们的功能分别如下：

- netfilter 组件是内核的一部分，由一些包过滤表组成，这些表包含内核用来控制包过滤处理的规则集。
- iptables 组件是一种工具，它是用户空间的一部分，它用来维护包过滤表中的规则，包括包过滤规则的插入、修改和删除。

通过使用 iptables，用户可以构建自己的定制规则，这些规则存储在内核空间的包过滤表中。这些规则告诉内核对来自某些源、前往某些目的地或具有某些协议类型的包做些什么。如果某个包与规则匹配，那么使用目标 ACCEPT 允许该包通过。还可以使用目标 DROP 或 REJECT 来阻塞并杀死包。对于可对数据包执行的其他操作，还有许多其他目标。

根据规则所处理的包的类型，可以将规则分组在链中。处理入站包的规则被添加到 INPUT 链中。处理出站包的规则被添加到 OUTPUT 链中。处理正在转发的包的规则被添加到 FORWARD 链中。这 3 个链是基本包过滤表中内置的默认主链。另外，还有其他许多可用的链类型（如 PREROUTING 和 POSTROUTING），以及提供用户定义的链。每个链都可以有一个策略，它定义“默认目标”，也就是要执行的默认操作，当包与链中的任何规则都不匹配时，执行此操作。

建立规则，并将链放在适当的位置之后，就可以开始进行真正的包过滤工作了。这时内核空间从用户空间接管工作。当包到达防火墙时，内核先检查包的头信息，尤其是包的目的地。通常将这个过程称为路由。

如果包源自外界并前往系统，而且防火墙是打开的，那么内核将它传递到内核空间包过滤表的 INPUT 链。如果包源自系统内部或系统所连接的内部网上的其他源，并且此包要前往另一个外部系统，那么包被传递到 OUTPUT 链。类似地，源自外部系统并前往外部系统的包被传递到 FORWARD 链。

接下来，将包的头信息与它所传递到的链中的每条规则进行比较，看它是否与某条规则完全匹配。如果包与某条规则匹配，那么内核就对该包执行由该规则的目标指定的操作。

但是，如果包与这条规则不匹配，那么它将与链中的下一条规则进行比较。最后，如果包与链中的任何规则都不匹配，那么内核将参考该链的策略来决定如何处理该包。理想的策略应该告诉内核 DROP 该包。

### 24.2.2 Netfilter 防火墙在 IPv4 中实现的原理和结构

Netfilter 为每种网络协议（IPv4、IPv6 等）定义一套钩子函数（IPv4 定义了 5 个钩子函数），这些钩子函数在数据包流过协议栈的几个关键点被调用。内核的任何模块可以对每种协议的一个或多个钩子进行注册，实现挂接。当某个包传递给 netfilter 时，调用模块的注册时使用的回调函数，这样模块能检查、修改或丢弃该数据包及指示 netfilter 将该数据包传入用户空间的队列。被排队的数据包被传递给用户空间，并异步地进行处理。用户进程能检查、修改、甚至重新将该数据包通过离开内核的同一个钩子函数中注入到内核中。所有的包过滤和 NAT 等都基于该框架，它极大的简化了内核网络子系统的设计。

#### 1. IPv4 包在 Netfilter 中的流动过程

在 IPv4 中，数据包按照图 24-1 所示的过程通过 Netfilter 防火墙，IPv4 共有 5 个钩子函数，分别为：

- NF\_IP\_PRE\_ROUTING;
- NF\_IP\_LOCAL\_IN;
- NF\_IP\_FORWARD;
- NF\_IP\_POST\_ROUTING;
- NF\_IP\_LOCAL\_OUT.

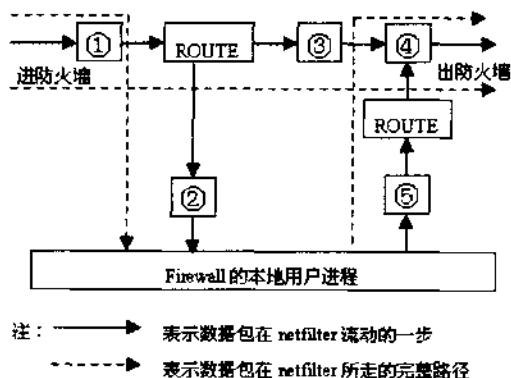


图 24-1 Linux 2.4 内核 Netfilter 防火墙的 Ipv4 实现框架

从网络接口传入的 IP 数据包从左边进入防火墙，进行校验以后，经过第一个钩子函数 ① 进行处理；然后就进行路由选择，以决定该数据包是需要转发，还是发送给本机；若应发给本机，则它经过钩子函数 ② 处理后传递给上层协议；若应被转发，则它被 ③ 处理；转发的数据包经最后一个钩子函数 ④ 处理后，再传输到网络上。由本地产生的数据包经钩子函数 ⑤ 处理可以后，进行路由选择处理，然后经 ④ 处理后发送到网络上。任一合法的数据包仅经过三条路径（图 24-1 中用虚线表示）之一。



## 2. 使用 iptables 进行数据包选择

基于 Netfilter 框架（被称为数据包过滤系统）在 Linux2.4 内核中被应用。在图 24-1 中的①~⑤，在 Netfilter 中分别对应 PREROUTING 链、INPUT 链、FORWARD 链、POSTROUTING 链及 OUTPUT 链。

## 3. iptables 规则表

在 iptables 中内置了过滤（filter 表）、网络地址转换（NAT 表）和数据包处理（mangle 表）3 个规则表。任何包仅被这 3 个规则表中的一个击中，并按规则表中的规则对数据包进行处理。这 3 种规则表处理功能如下：

- 包过滤。filter 表格（默认表格），不会对数据包进行修改，只进行过滤。它是通过钩子函数 NF\_IP\_LOCAL\_IN, NF\_IP\_FORWARD 及 NF\_IP\_LOCAL\_OUT 接入 netfilter 框架的。对于任何一个数据包只有一个地方对其进行过滤。
- 地址转换表(NAT)。NAT 表格监听 3 个 Netfilter 钩子函数：NF\_IP\_PRE\_ROUTING、NF\_IP\_POST\_ROUTING 及 NF\_IP\_LOCAL\_OUT。NF\_IP\_PRE\_ROUTING 实现对需转发数据包的源地址进行转换，NF\_IP\_POST\_ROUTING 对需转发的数据包的目的地址进行转换。对于本地数据包的目的地址的转换则由 NF\_IP\_LOCAL\_OUT 来实现。只有新连接的第一个数据包遍历 NAT，随后进行同样的转换处理。NAT 可被用在源 NAT 和目的 NAT 处，伪装（其是源 NAT 的特例）及透明代理（其是目的 NAT 的特例）。
- 数据包处理（Packet mangling）。mangle 表格在 NF\_IP\_PRE\_ROUTING 和 NF\_IP\_LOCAL\_OUT 钩子中进行注册。
- 连接跟踪。连接跟踪是 NAT 的基础，但是已经作为一个单独的模块被实现。该功能扩展了包过滤功能，用它实现“基于状态”的防火墙。

## 24.3 理解 Netfilter 防火墙技术

### 24.3.1 建立规则和链

当 netfilter/iptables 系统运行之后，接下来需要建立一些规则和链来过滤数据包。

通过向防火墙提供有关对来自某个源、到某个目的地或具有特定协议类型的数据包要做些什么的指令，规则控制数据包的过滤。通过使用 netfilter/iptables 系统提供的特殊命令 iptables，建立这些规则，并将其添加到内核空间的特定数据包过滤表内的链中。关于添加、删除和编辑规则的一般语法如下：

```
iptables [-t table] command [match] [target]
```

其中参数和选项如下：

（1）表（table）。[-t table]选项允许使用标准表之外的任何表。表是包含仅处理特定类型数据包的规则和链的数据包过滤表。有 3 种可用的表选项：filter、nat 和 mangle。该选项不是必需的，如果未指定，则 filter 用作默认表。这些表的作用如下：

- filter 表用于一般的数据包过滤，它包含 INPUT、OUTPUT 和 FORWARD 链。
- nat 表用于要转发的数据包，它包含 PREROUTING、OUTPUT 和 POSTROUTING 链。
- 如果数据包及其头内进行了任何更改，则使用 mangle 表。该表包含一些规则来标记用于高级路由的数据包，该表包含 PREROUTING 和 OUTPUT 链。

注意：PREROUTING 链由指定数据包一到达防火墙就改变它们的规则所组成，而 POSTROUTING 链由指定正当数据包准备离开防火墙时改变它们的规则所组成。

(2) 命令 (command)。具有强制性的 command 部分是 iptables 命令最重要的部分。它告诉 iptables 命令要做什么。例如，插入规则、将规则添加到链的末尾或删除规则。以下是最常用的一些命令：

- -A 或—append。该命令将一条规则添加到链的末尾，如  
`$ iptables -A INPUT -s 205.168.0.1 -j ACCEPT`  
 该示例将一条规则附加到 INPUT 链的末尾，确定来自源地址 205.168.0.1 的数据包可以 ACCEPT。
- -D 或—delete。通过用 -D 指定要匹配的规则或者指定规则在链中的位置编号。该命令从链中删除该规则。下面的例子说明了这两种方法：  
`$ iptables -D INPUT --dport 80 -j DROP`  
`$ iptables -D OUTPUT 3`  
 第一条命令从 INPUT 链删除规则，它指定 DROP 前往端口 80 的数据包。第二条命令只是从 OUTPUT 链删除编号为 3 的规则。
- -P 或—policy。该命令设置链的默认策略。所有与链中任何规则都不匹配的数据包，都将被强制使用此链的默认策略，如  
`$ iptables -P INPUT DROP`  
 该命令将 INPUT 链的默认目标指定为 DROP。这意味着，将丢弃所有与 INPUT 链中任何规则都不匹配的数据包。
- -N 或—new-chain。用命令中所指定的名称创建一个新链，如  
`$ iptables -N newchain`
- -F 或—flush。如果指定链名，该命令删除链中的所有规则，如果未指定链名，该命令删除所有链中的所有规则。此参数用于快速清除。例：  
`$ iptables -F FORWARD`  
`$ iptables -F`
- -L 或—list。列出指定链中的所有规则。例：  
`$ iptables -L newchain`

(3) match (匹配)。iptables 命令的可选 match 部分指定数据包与规则匹配所应具有的特征（如源和目的地地址、协议等）。匹配分为两大类：通用匹配和特定于协议的匹配。这里将讲述可用于采用任何协议的数据包的通用匹配。常用的通用匹配实例如下：

- -p 或—protocol。该通用协议匹配用于检查某些特定协议。协议示例有 TCP、UDP、ICMP、用逗号分隔的任何这 3 种协议的组合列表以及 ALL（用于所有协议）。ALL 是默认匹配。可以使用 ! 符号，它表示不与该项匹配。例：

```
$ iptables -A INPUT -p TCP, UDP
```

```
$ iptables -A INPUT -p !ICMP
```

在上述示例中，这两条命令都执行同一任务，它们指定所有 TCP 和 UDP 数据包都将与该规则匹配。通过指定 !ICMP 打算允许所有其他协议（在这种情况下是 TCP 和 UDP），而将 ICMP 排除在外。

- **-s 或—source**。该源匹配用于根据数据包的源 IP 地址来与它们匹配。该匹配还允许对某一范围内的 IP 地址进行匹配，可以使用 ! 符号，表示不与该项匹配。默认源匹配与所有 IP 地址匹配。例：

```
$ iptables -A OUTPUT -s 192.168.1.1
```

```
$ iptables -A OUTPUT -s 192.168.0.0/24
```

```
$ iptables -A OUTPUT -s ! 203.16.1.89
```

第二条命令指定该规则与所有来自 192.168.0.0 到 192.168.0.254 的 IP 地址范围的数据包匹配。第三条命令指定该规则将与除来自源地址 203.16.1.89 外的任何数据包匹配。

- **-d 或--destination**：该目的地匹配用于根据数据包的目的地 IP 地址来与它们匹配。该匹配还允许对某一范围内 IP 地址进行匹配，可以使用 ! 符号，表示不与该项匹配。例：

```
$ iptables -A INPUT -d 192.168.1.1
```

```
$ iptables -A INPUT -d 192.168.0.0/24
```

```
$ iptables -A OUTPUT -d ! 203.16.1.89
```

(4) **target (目标)**。目标是由规则指定的操作，对与那些规则匹配的数据包执行这些操作。除了允许用户定义的目标之外，还有许多可用的预定义目标选项。下面是常用的一些目标：

- **ACCEPT**。当数据包与具有 ACCEPT 目标的规则完全匹配时，会被接受（允许它前往目的地），并且它将停止遍历链（虽然该数据包可能遍历另一个表中的其他链，并且有可能在那里被丢弃）。该目标被指定为 -j ACCEPT。
- **DROP**。当数据包与具有 DROP 目标的规则完全匹配时，会阻塞该数据包，并且不对它做进一步处理。该目标被指定为 -j DROP。
- **REJECT**。该目标的工作方式与 DROP 目标相同，但它比 DROP 好。和 DROP 不同的是，REJECT 不会在服务器和客户机上留下死套接字。另外，REJECT 将错误消息发回给数据包的发送方。该目标被指定为 -j REJECT。例：

```
$ iptables -A FORWARD -p TCP --dport 22 -j REJECT
```

- **RETURN**。在规则中设置的 RETURN 目标让与该规则匹配的数据包停止遍历包含该规则的链。如果链是如 INPUT 之类的主链，则使用该链的默认策略处理数据包。它被指定为 -jump RETURN。例：

```
$ iptables -A FORWARD -d 203.16.1.89 -jump RETURN
```

还有许多用于建立高级规则的其他目标，如 LOG、REDIRECT、MARK、MIRROR 和 MASQUERADE 等。

保存规则。读者已经学习了如何建立基本的规则、链以及如何从数据包过滤表中添加

或删除它们。但是，应该注意，用上述方法所建立的规则会被保存到内核中，当重新引导系统时，会丢失这些规则。因此，如果成功地测试了规则集，并将这些规则添加到数据包过滤表，同时希望在重新引导之后再次使用这些规则，那么必须将该规则保存在文件中。

可以使用 `iptables-save` 命令来做到这一点：

```
$ iptables-save > iptables-rules
```

现在，数据包过滤表中的所有规则都被保存在文件 `iptables-rules` 中。无论何时再次引导系统，都可以使用 `iptables-restore` 命令将规则集从该脚本文件恢复到数据包过滤表，命令如下：

```
$ iptables-restore iptables-rules
```

如果需要在每次引导系统时自动恢复这些规则，则可以将上面的这条命令放到任何一个初始化 shell 脚本中。

### 24.3.2 netfilter/iptables 系统的优点

netfilter/iptables 具有如下两大优点：

(1) netfilter/iptables 的最大优点是它可以配置有状态的防火墙。

这是 `ipfwadm` 和 `ipchains` 等以前的工具都无法提供的一种重要功能。有状态的防火墙能够指定并记住为发送或接收数据包所建立的连接的状态。防火墙可以从数据包的连接跟踪状态获得该信息。在决定新的数据包过滤时，防火墙所使用的这些状态信息可以增加其效率和速度。这里有 4 种有效状态，名称分别为 `ESTABLISHED`、`INVALID`、`NEW` 和 `RELATED`。

状态 `ESTABLISHED` 指出该数据包属于已建立的连接，该连接一直用于发送和接收数据包并且完全有效。`INVALID` 状态指出该数据包与任何已知的流或连接都不相关联，它可能包含错误的的数据或头。状态 `NEW` 意味着该数据包已经或将启动新的连接。`RELATED` 表示该数据包正在启动新连接，以及它与已建立的连接相关联。

(2) netfilter/iptables 的另一个重要优点是：它使用户可以完全控制防火墙配置和数据包过滤。用户可以定制自己的规则来满足特定需求，从而只允许用户需要的 IP 包进入系统。

(3) netfilter/iptables 是免费的，这可以节省大笔开支，它可以代替昂贵的防火墙解决方案。

### 24.3.3 防火墙实例

本文以 Redhat Linux 9 自带的模块化内核，光纤接入互联网，两块网卡的防火墙为例，介绍用 `iptables` 构造防火墙的方法，如图 24-2 所示。内网为 192.168.0.0/24，防火墙的外网接口的 IP 为 202.200.121.1。该防火墙设置内网的主机可以访问任何外网的任何主机的任何服务，但是外网主机却不能访问内网中的任何服务，从而杜绝了外网攻击者向内网的攻击。

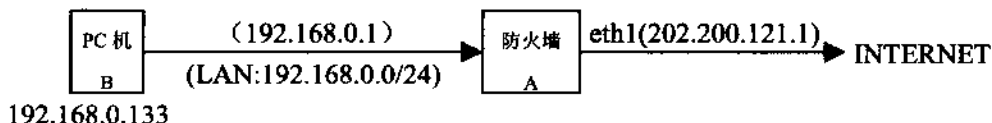


图 24-2 防火墙实例

## (1) 安全策略。

例：默认的转发规则为丢弃数据包。命令如下：

```
iptables -P FORWARD DROP
```

## (2) 包过滤 filter。

例：允许 192.168.0.0/24 转发本网段到任何地方去的包。命令如下：

```
iptables -A FORWARD -s 192.168.0.0/24 -j ACCEPT
```

## (3) NAT (网络地址转换)。

Netfilter 支持任何一种 NAT，它一般可分为源 NAT 和目的 NAT。源 NAT 在数据包经过 NF\_IP\_POST\_ROUTING 时修改数据包的源地址，伪装是一个特殊的 SNAT。

例如，对 192.168.0.0/24 子网进行伪装。

```
iptables -t nat -A POSTROUTING -o eth1 -s 192.168.0.0/24 -j MASQUERADE
```

最常见的透明代理的使用是结合开放源代理软件 Squid，实现局域网内用户在不用设置任何浏览器代理的条件下透明地使用代理上网。如果我们的 Squid 的 HTTP 端口运行在 3128，则使用下面的规则：

```
iptables -t nat -A PREROUTING -i eth0 -p tcp -s 192.168.0.0/24 --dport 80 -j DNAT --to 192.168.0.1:3128
```

本例需要的完整的防火墙规则可以从如下 iptables 命令生成，可以将该 iptables 命令集以 root 用户身份加入到/etc/rc.d/rc.local 文件中，生成规则集的 iptables 命令内容如下：

```
modprobe ip_tables
echo > /proc/sys/net/ipv4/ip_forward
iptables -F INPUT
iptables -F FORWARD
iptables -F POSTROUTING -t nat
iptables -P FORWARD DROP
iptables -A FORWARD -s 192.168.0.0/24 -j ACCEPT
iptables -A FORWARD -p tcp -i eth1 -d 192.168.0.0/24 --syn --sport 20 -j ACCEPT
iptables -A FORWARD -p tcp -i eth1 -m state --state ESTABLISHED,RELATED -j ACCEPT
```

```
iptables -t nat -A POSTROUTING -o eth1 -s 192.168.0.0/24 -j MASQUERADE
```

当保存文件并执行 init 6 命令重新引导系统后，将会使防火墙的规则生效。

可以使用如下命令查看内核的防火墙规则列表：

```
[root@jiang root]# iptables -L
```

结果如下：

```
Chain INPUT (policy ACCEPT)
target    prot opt source                destination

Chain FORWARD (policy DROP)
target    prot opt source                destination
ACCEPT    all  --  192.168.0.0/24        anywhere
```

```
ACCEPT tcp--anywhere 192.168.0.0/24 tcp spt:ftp-data flags:SYN,RST,ACK/SYN
ACCEPT    tcp  --  anywhere      anywhere      state RELATED,ESTABLISHED
```

```
Chain OUTPUT (policy ACCEPT)
```

```
target    prot opt source                destination
```

这样，内网的主机可以访问外网的任何主机的任何网络服务，而外网的主机则不能访问内网的任何主机的任何服务，从而可以保护内网的安全。

## 思考与练习

- (1) Netfilter 的体系结构如何？
- (2) Netfilter 有哪些链？它们的作用是什么？
- (3) 如何实现 IP 伪装？
- (4) 如何实现 NAT？