张 栋 黄 成 等编著

Linux 服务器搭建实战详解

- FTP、E-mail、Web、DHCP、DNS 、Squid、NFS、LDAP、SSH及MySQL等服务的搭建
- 用Webmin配置服务的方法，Shell命令的应用，Debian系统及APT、YUM软件包管理工具的使用
- 静态IP地址管理、动态IP地址分配及网络地址表管理等网络管理工具的使用
- 用户管理、进程管理、文件系统管理、磁盘管理的操作及GNOME、KDE桌面管理器的应用方法

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

Linux 服务器搭建实战详解

本书系统阐述了各种常用服务的配置方法、管理技巧以及相关应用。全书共21章，分为三大体系：Linux操作系统中各种常用服务器的配置与管理，常用的网络管理和系统管理，以及系统安全。此外，各章节还结合了实例，使读者能够边学边练，增强理解。从而将Linux服务器搭建实战中的难点各个击破，成为网管高手。

ISBN 978-7-121-09898-7

9 787121 098987 >

定价：59.80元

PHEI

天启星

http://www.tqxbook.com

责任编辑：郭鹏飞

责任美编：王　嵩

本书贴有激光防伪标志，凡没有防伪标志者，属盗版图书。

Linux 服务器搭建实战详解

张 栋 黄 成 等编著

電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书首先对 Linux 操作系统进行了相关概述，并讲解了安装 Red Hat Enterprise Linux 5 的相关知识及 Red Hat Enterprise Linux 5 中的 GNOME 和 KDE 两种桌面管理器。然后介绍了在 Linux 操作系统中各种常用服务的配置管理方法及其相关应用，包括 FTP 服务、电子邮件服务、Web 服务、DHCP 服务、DNS 服务、代理服务、Samba 服务、NFS 服务、LDAP 目录服务、SSH 服务及 MySQL 数据库服务等。同时本书还加入了常用的网络管理知识和基础的系统管理知识，包括 Linux 操作系统的网络连通性测试、网络管理、软件基础管理、软件包高级管理、用户和进程管理、Linux 文件系统及磁盘管理，以及 Linux 操作系统安全管理。

本书适合从事网络管理、网络维护工作的工程技术人员作为操作手册使用，也可作相关院校计算机及相关专业学习的教材。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有，侵权必究。

图书在版编目(CIP)数据

Linux 服务器搭建实战详解 / 张栋等编著.—北京：电子工业出版社，2010.1
(Linux 实战详解)
ISBN 978-7-121-09898-7

Ⅰ.L… Ⅱ.张… Ⅲ.Linux 操作系统 Ⅳ.TP316.89

中国版本图书馆 CIP 数据核字（2009）第 210037 号

责任编辑：郭鹏飞
印　　刷：北京市天竺颖华印刷厂
装　　订：三河市鑫金马印装有限公司
出版发行：电子工业出版社
　　北京市海淀区万寿路 173 信箱　邮编　100036
开　　本：787×1092　1/16　印张：36　字数：922 千字
印　　次：2010 年 1 月第 1 次印刷
定　　价：59.80 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：（010）88258888。

前言

PREFACE

Linux 操作系统由于具有稳定、安全等特性，受到了广大用户的青睐，同时越来越多的新用户开始使用该操作系统。为了顺应时代发展及广大用户的需求，本书以 Red Hat 企业版操作系统为例，系统地阐述了各种常用服务的配置方法、管理技巧及相关应用，同时本书还加入常用的网络管理知识和基础的系统管理知识。

本书是一本具有现实意义的参考、学习资料，广大读者通过对本书所涉及到的实例进行实践操作后，便可轻松解决在使用 Linux 操作系统过程中遇到的问题。这也是本书的特色和目的。

全书共 21 章，分为三大体系：Linux 操作系统中各种常用服务器的配置与管理，常用的网络管理和系统管理，以及系统安全。各章节主要内容如下。

第 1 章主要阐述了 Linux 的发展历程、特点及其应用领域，同时讲解了安装 Red Hat Enterprise Linux 5 的相关知识，最后还介绍了 Red Hat Enterprise Linux 5 中的 GNOME 和 KDE 两种桌面管理器的使用及管理，为大家学习 Linux 打下基础。

第 2 章进入服务器的具体配置管理环节，本章以 FTP 工作原理入手，主要讲解了 FTP 服务器的安装及配置方法。

第 3 章首先讲解了电子邮件服务的相关知识，然后着重介绍了 Postfix 邮件服务器以及 POP 和 IMAP 邮件服务的安装及配置。

第 4 章首先对 Web 服务的相关知识进行了介绍，然后重点讲解了如何在 Linux 系统中配置与管理 Apache 服务器，最后还结合实用案例讲解了如何通过 JSP 等来创建动态 Web 网站。

第 5 章以 DHCP 服务的原理入手，主要讲解了 DHCP 服务的安装与配置。

第 6 章首先阐述 DNS 的相关知识，然后讲解了 DNS 服务的安装，最后重点介绍了 DNS 的相关配置。

第 7 章首先概述了代理服务器的相关知识，接着介绍了如何安装 Squid Server，然后讲解了 Squid Server 的基本配置及高级配置，最后还介绍了 Squid 的日志管理。

第 8 章主要介绍了用于在 Windows 操作系统与 Linux 操作系统之间提供资源共享的 Samba 服务的基本概念、安装方法、配置方法及其使用等。

第 9 章主要讲解了如何在 Red Hat Enterprise Linux 5 中搭建及管理 NFS 服务。

第 10 章从 LDAP 服务器的基本概念入手，具体讲解其安装、配置以及应用。

第 11 章首先介绍了 SSH 服务的相关概念，然后讲解了 SSH 服务的安装及配置，最后介绍了在不同平台下如何使用 SSH 客户端。

第 12 章首先阐述了 MySQL 的各种特点及应用领域，然后介绍了 MySQL 的安装，最后重点讲解了 MySQL 的配置及应用。

第 13 章首先介绍了图形化配置工具 Webmin 的主要功能，然后讲解了 Webmin 的安装及

目录

CONTENTS

第 1 章 Linux 系统概述 1

1.1 Linux 的发展 1
1.2 Linux 常见版本 2
1.2.1 Red Hat Linux 2
1.2.2 Debian Linux 2
1.2.3 Ubuntu Linux 3
1.3 Linux 操作系统 3
1.3.1 Linux 操作系统结构 3
1.3.2 Linux 操作系统特点 5
1.3.3 Linux 操作系统的应用领域 6
1.3.4 Red Hat Enterprise Linux 5 7
1.4 Red Hat Enterprise Linux 5 的两个版本 8
1.5 安装和管理 Red Hat Enterprise Linux 8
1.5.1 安装前的准备 8
1.5.2 与其他操作系统并存的问题 10
1.5.3 Red Hat Enterprise Linux 的安装阶段 11
1.6 Linux 操作系统的引导与登录 19
1.6.1 多系统引导管理器——GRUB 及 LILO 19
1.6.2 Linux 的启动过程 20
1.6.3 Linux 的关闭与重新引导 20
1.6.4 Linux 系统的登录和退出 20
1.7 GNOME 和 KDE 桌面管理器 22
1.7.1 GNOME 桌面环境 23
1.7.2 KDE 桌面环境 31
1.7.3 定制自己的桌面 42

第 2 章 FTP 服务器 45

2.1 FTP 服务简介 45
2.1.1 FTP 工作原理 45
2.1.2 匿名用户访问的产生 46
2.1.3 FTP 服务的连接模式 46

2.1.4 几种流行的 FTP 服务器软件 48
2.2 安装 vsFTPd 服务 49
2.2.1 安装 vsFTPd 软件 49
2.2.2 FTP 服务的启动与关闭 50
2.3 vsFTPd 的相关配置 52
2.3.1 配置匿名用户登录环境 53
2.3.2 监听地址与控制端口 53
2.3.3 FTP 模式与数据端口 54
2.3.4 ASCII 模式 55
2.3.5 超时选项 55
2.3.6 负载控制 55
2.3.7 用户设置 56
2.3.8 用户登录控制 57
2.3.9 目录访问控制 57
2.3.10 文件操作控制 58
2.3.11 新增文件权限设置 58
2.3.12 日志设置 58
2.4 安装与配置 PureFTPd 服务器的 59
2.4.1 安装 Apache 服务 59
2.4.2 安装 MySQL 数据库 61
2.4.3 安装 PureFTPd 62
2.4.4 生成 PureFTPd 服务管理脚本 64
2.4.5 配置匿名用户登录环境 64
2.4.6 安装 Zend Optimizer 65
2.4.7 WebPureFTP 的安装与配置 66
2.4.8 生成 MySQL 数据库 67
2.4.9 生成 pure-ftpd 数据库配置文件 68
2.4.10 使用 WebPureFTP 管理虚拟用户 68
2.5 FTP 服务器的访问 69
2.5.1 Linux 环境下访问 FTP 服务器 69
2.5.2 Windows 环境下访问 FTP 服务器 70
第 3 章 电子邮件服务器 71
3.1 电子邮件服务概述 71
3.1.1 电子邮件服务简介 71
3.1.2 邮件专有名词解释 72
3.1.3 POP3 与 SMTP 协议 73
3.2 Postfix 邮件服务器 74

3.2.1 安装 Postfix 邮件服务器 74
3.2.2 Postfix 配置文件 76
3.2.3 Postfix 基本配置 77
3.2.4 虚拟别名域 80
3.2.5 用户别名 80
3.2.6 SMTP 认证 81
3.2.7 启动和停止 Postfix 服务 83
3.2.8 配置与访问电子邮件客户端 84
3.3 POP 和 IMAP 邮件服务 87
3.3.1 dovecot 服务 87
3.3.2 CyrusImapd 服务 88
3.4 Web 支持电子邮件 92
3.4.1 Squirrelmail 的安装与配置 92
3.4.2 登录 Squirrelmail 94
第 4 章 Web 服务器 95
4.1 Web 概述 95
4.1.1 Web 服务简介 95
4.1.2 HTTP 协议简介 96
4.1.3 Web 服务器软件 99
4.2 Apache 概述 99
4.2.1 Apache 的诞生 100
4.2.2 安装 Apache 服务 101
4.3 启动与停止 Apache 服务 103
4.4 Apache 服务的基本配置方法 106
4.4.1 httpd.conf 配置文件 106
4.4.2 Apache 服务主配置文件的构成 107
4.5 Apache 服务器控制存取的方式 112
4.5.1 Options 选项 113
4.5.2 浏览权限的设置 113
4.6 Apache 服务的高级配置 116
4.6.1 虚拟目录简介 116
4.6.2 配置与管理虚拟主机 117
4.6.3 用户认证和授权 119
4.6.4 配置 SSI 123
4.6.5 MIME 类型简介 124
4.7 创建并配置动态网站 124
4.7.1 创建并配置 JSP 动态网站 124

4.7.2 创建并配置 CGI 动态网站 131
第 5 章 DHCP 服务器 133
5.1 DHCP 服务概述 133
5.1.1 DHCP 服务的实现原理 133
5.1.2 IP 地址租约和更新 134
5.1.3 DHCP 服务器分配的 IP 地址类型 135
5.2 安装 DHCP 服务 137
5.2.1 做好 DHCP 服务器安装前的准备 137
5.2.2 安装 DHCP 服务的方法 138
5.2.3 启动及停止 DHCP 服务 141
5.3 配置 DHCP 服务 142
5.3.1 DHCP 配置文件及格式 142
5.3.2 配置 IP 作用域 145
5.3.3 配置客户机的 IP 选项 146
5.3.4 设置租约期限 147
5.3.5 保留特定的 IP 地址 147
5.3.6 多网段的 IP 地址的分配 148
5.3.7 DHCP 服务器配置实例 149
5.4 配置 DHCP 客户机 150
5.4.1 以 Linux 操作系统作为客户端配置 150
5.4.2 以 Windows 操作系统作为客户端配置 152
第 6 章 DNS 服务器 155
6.1 DNS 服务概述 155
6.1.1 DNS 服务简介 155
6.1.2 DNS 提供的服务 156
6.1.3 DNS 查询原理和规划 156
6.1.4 DNS 域名解析 158
6.2 DNS 服务的安装 158
6.2.1 安装 DNS 服务 159
6.2.2 安装 Chroot 软件包 159
6.2.3 DNS 服务的启动与停止 160
6.3 配置主要名称服务器 161
6.3.1 主配置文件 161
6.3.2 设置根区域和主区域 162
6.3.3 设置反向解析区域 165
6.3.4 根服务器信息文件 166

6.3.5 区域文件 166
6.3.6 反向解析区域文件 169
6.3.7 功能实现 170
6.3.8 测试 DNS 服务器 170
6.4 配置辅助 DNS 服务器 174
6.5 配置缓存 Cache-only 服务器 175
6.6 DNS 客户端的配置 176
6.6.1 在 Linux 下配置 DNS 客户端 176
6.6.2 在 Windows 下配置 DNS 客户端 177
第 7 章 代理服务器 179
7.1 代理服务器概述 179
7.1.1 代理服务器的工作流程 179
7.1.2 代理服务器的作用及相关应用 180
7.2 安装 Squid Server 181
7.2.1 Squid 简介 181
7.2.2 Squid Server 的安装 182
7.3 Squid Server 的基本配置 184
7.3.1 设置监听的 IP 和端口 184
7.3.2 设置缓存大小 184
7.3.3 访问控制设置 186
7.3.4 指定 Squid 可以使用内存的理想值 188
7.3.5 其他参数设置 188
7.3.6 设置错误提示为中文 189
7.3.7 初始化 Squid Server 189
7.4 启动与停止代理服务器 192
7.4.1 设置 Squid 服务随系统的启动而自动启动 193
7.4.2 重新载入配置文件 195
7.5 Squid Server 高级配置 195
7.5.1 实现透明代理 195
7.5.2 Squid 安全设置 197
7.5.3 Squid 外部程序支持选项 199
7.6 Squid 日志管理 199
7.6.1 日志选项 199
7.6.2 Squid 代理服务器日志系统的构成 199
7.6.3 日志及流量分析软件——Webalizer 200
7.6.4 Webalizer 软件的应用 204
7.7 不同环境下的代理客户端的配置 205

7.7.1 Linux 系统下 Firefox 代理设置 205
7.7.2 Windows 系统下 IE 代理配置 207
第 8 章 Samba 服务器 209
8.1 Samba 服务概述 209
8.1.1 Samba 服务简介 209
8.1.2 Samba 服务的安装 209
8.1.3 Samba 服务的图形化配置方法 212
8.1.4 启动、停止和重启 Samba 服务 216
8.2 Samba 服务的配置文件 216
8.2.1 Samba 服务的主要配置文件 217
8.2.2 Samba 服务的密码文件 217
8.2.3 Samba 服务的日志文件 218
8.3 smb.conf 文件详解 218
8.3.1 smb.conf 文件结构 219
8.3.2 smb.conf 文件的语法和变量 220
8.3.3 smb.conf 文件详解 220
8.3.4 smb.conf 文件中的段 223
8.3.5 Samba 服务具体参数设置 223
8.4 Samba 服务功能实现 225
8.4.1 指定 Samba 所要加入的工作组 225
8.4.2 注释说明服务器 226
8.4.3 限制可访问服务器 IP 地址的范围 226
8.4.4 打印机设置 227
8.4.5 设置 guest 账号匿名登录 228
8.4.6 设置日志文件名、存放路径及大小 228
8.4.7 服务器安全设置 229
8.4.8 具体的共享设置 232
8.5 共享的资源访问 234
8.5.1 不同客户端主机的资源访问 234
8.5.2 本机 Windows 分区和 Linux 分区的资源互访 235
第 9 章 NFS 服务器 237
9.1 NFS 服务简介 237
9.1.1 使用 NFS 的优势 237
9.1.2 NFS 服务的组件 238
9.2 RPC 简介 238
9.3 安装并配置 NFS 服务 239

9.3.1 安装 NFS 服务 239
9.3.2 配置 NFS 服务 240
9.3.3 NFS 文件的存取权限 242
9.4 启动与停止 NFS 服务 243
9.5 在客户端挂载 NFS 文件系统 244
9.5.1 查看 NFS 服务器信息 244
9.5.2 加载 NFS 服务器共享目录 245
9.5.3 自动挂接 NFS 文件系统 246
第 10 章 LDAP 目录服务 247
10.1 目录服务概述 247
10.1.1 X.500 简介 247
10.1.2 初识 LDAP 248
10.1.3 X.500 与 LDAP 251
10.1.4 流行的目录服务产品 252
10.2 深入认识 LDAP 254
10.2.1 LDAP 的几种基本模型 254
10.2.2 如何规划目录树 256
10.2.3 LDAP 的应用领域 257
10.3 安装 LDAP 257
10.3.1 安装 Berkeley DB 257
10.3.2 安装 OpenLDAP 260
10.4 配置 OpenLDAP 261
10.4.1 OpenLDAP 的基本配置 261
10.4.2 启动 OpenLDAP 服务器 262
10.5 为 OpenLDAP 建立初始化数据 263
10.6 LDAP 图形界面管理工具简介 264
10.6.1 主要的 LDAP 图形界面管理工具 264
10.6.2 安装 phpLDAPadmin 266
10.7 配置 phpLDAPadmin 268
10.7.1 生成 phpLDAPadmin 主配置文件 268
10.7.2 phpLDAPadmin 的基本配置 268
10.8 使用 OpenLDAPadmin 管理目录树 271
10.8.1 进入 phpLDAPadmin 271
10.8.2 创建组织单元 272
10.8.3 创建用户组 274
10.8.4 创建用户账号 276
10.9 实现 LDAP 服务的身份验证 277

10.9.1 实现 Linux 系统用户验证 277
10.9.2 实现 FTP 用户验证 279
10.9.3 实现 Web 用户验证 280
第 11 章 SSH 服务器 281
11.1 SSH 服务概述 281
11.1.1 SSH 服务简介 281
11.1.2 公钥加密体系结构 282
11.2 SSH 服务的安装 283
11.3 配置 SSH 服务器 283
11.4 SSH 服务的启动和停止 286
11.4.1 启动 SSH 服务 286
11.4.2 重启 SSH 服务 287
11.4.3 自动启动 SSH 服务 287
11.4.4 停止 SSH 服务 288
11.5 使用 SSH 客户端 288
11.5.1 SSH 客户端在 Windows 平台下的使用 288
11.5.2 SSH 客户端在 Linux 平台下的使用 289
第 12 章 MySQL 数据库的配置 293
12.1 MySQL 概述 293
12.1.1 MySQL 简介 293
12.1.2 MySQL 的应用领域 294
12.2 安装 MySQL 295
12.2.1 安装前的准备 295
12.2.2 MySQL 的具体安装过程 295
12.3 启动与停止 MySQLD 服务 298
12.4 配置及使用 MySQL 服务 298
12.4.1 修改 MySQL 管理员的口令 299
12.4.2 MySQL 数据库的创建和使用 299
12.4.3 MySQL 数据表的创建和使用 300
12.4.4 索引的创建和删除 304
12.4.5 用户的创建和删除 305
12.4.6 用户权限的设置 306
12.5 图形化配置 MySQL 309
12.5.1 安装 phpMyAdmin 309
12.5.2 配置 phpMyAdmin 虚拟目录 310
12.5.3 使用 phpMyAdmin 311

第 13 章　使用 Webmin 配置各种服务 313
13.1　Webmin 的概述 313
13.1.1　Webmin 简介 313
13.1.2　Webmin 特性 313
13.1.3　Webmin 主要功能 314
13.2　安装和配置 Webmin 317
13.2.1　Perl 语言解释器的安装 318
13.2.2　Net-SSLeay perl 和 OpenSSL 的安装 318
13.2.3　安装与访问 Webmin 319
13.2.4　设置 Webmin 为中文界面 321
13.2.5　停止和启动 Webmin 服务 322
13.3　用 Webmin 配置服务 323
13.3.1　Samba 服务的配置 323
13.3.2　NFS 服务的配置 327
13.3.3　DNS 服务的配置 328
13.3.4　Web 服务的配置 331
13.3.5　网络防火墙与 NAT 服务的配置 336
13.3.6　代理服务配置 342
13.3.7　SSH 服务的配置 349
第 14 章　Linux 命令行 351
14.1　Linux Shell 概述 351
14.1.1　Linux Shell 简介 351
14.1.2　系统默认 Shell——Bash Shell 351
14.1.3　设置使用其他 Shell 352
14.2　启动 Shell 353
14.2.1　终端启动 Shell 353
14.2.2　直接启用 Shell 354
14.2.3　直接启动命令行窗口 354
14.3　Shell 指令详解 355
14.3.1　目录及文件管理指令 355
14.3.2　信息显示指令 360
14.3.3　网络基本配置指令 363
14.3.4　关机和注销指令 366
14.4　配置 Shell 环境 367
14.4.1　Shell 环境变量 367
14.4.2　环境变量设置 369

14.4.3 Shell 环境配置文件 370
14.5 使用 vim 文本编辑器 371
14.5.1 vim 编辑器简介 371
14.5.2 常用 vim 指令 372
14.5.3 vim 的定制 376
第 15 章 Linux 操作系统中的网络连通性测试 377
15.1 网络线路的连通性测试 377
15.1.1 双绞线的线芯 377
15.1.2 测试直通双绞线 378
15.1.3 测试交叉双绞线 378
15.1.4 利用 ping 命令测试网络连通性 379
15.2 路由跟踪 382
15.2.1 traceroute 命令 382
15.2.2 traceroute 显示代码 383
15.2.3 traceroute 命令选项的应用 384
15.3 Linux 图形界面下测试网络 386
15.3.1 进入“Network Tools”窗口 386
15.3.2 网络设备查询 387
15.3.3 网络连通性测试 387
15.3.4 网络信息统计 388
15.3.5 网络路由跟踪 388
15.3.6 网络端口扫描 389
15.3.7 网络查阅 389
15.3.8 查询登录用户的信息 390
15.3.9 域名查询工具 390
第 16 章 Linux 网络管理 391
16.1 静态 IP 地址管理工具 391
16.1.1 IP 地址 391
16.1.2 子网掩码 392
16.1.3 TCP/IP 配置文件 392
16.1.4 在图形界面下配置以太网络 392
16.1.5 在命令模式下配置以太网 397
16.2 动态 IP 地址分配工具 400
16.2.1 获取动态 IP 地址 400
16.2.2 安装 DHCP 服务 401
16.2.3 DHCP 服务的启动与停止 404

16.2.4 DHCP 服务的配置 405
16.3 网络地址表管理工具 407
16.3.1 显示 ARP 缓存 407
16.3.2 管理 ARP 缓存 408
第 17 章 软件基础管理 409
17.1 Kickstart 简介 409
17.1.1 安装 Kickstart 409
17.1.2 Debian/Ubuntu 的自动化安装 419
17.2 Linux 操作系统中的软件包管理 420
17.2.1 RPM 简介 420
17.2.2 RPM 软件包管理系统的应用 423
17.3 Debian 的软件包管理系统 434
17.3.1 DPKG 简介 434
17.3.2 DPKG 软件包管理系统应用 435
第 18 章 软件包高级管理 441
18.1 APT——高级软件包管理 441
18.1.1 APT 概述 441
18.1.2 应用 APT 软件包管理系统 442
18.2 YUM——智能化的 RPM 软件包管理 452
18.2.1 YUM 概述 452
18.2.2 应用 YUM 软件包管理系统 453
18.3 Linux 内核的升级 462
18.3.1 获取内核源代码 463
18.3.2 配置内核 464
18.3.3 编译内核 467
18.3.4 安装内核 469
第 19 章 用户和进程管理 471
19.1 用户管理 471
19.1.1 用户账号文件 471
19.1.2 用户群组文件 474
19.1.3 命令行方式的用户管理 476
19.1.4 图形界面方式的用户管理 486
19.1.5 命令行方式的群组管理 486
19.2 进程管理 490
19.2.1 进程及作业 490

19.2.2 进程控制 490
19.2.3 作业控制 494
第 20 章 管理 Linux 文件系统及磁盘 497
20.1 管理 Linux 文件系统 497
20.1.1 Linux 文件系统结构 497
20.1.2 常见的文件系统 497
20.1.3 文件类型 498
20.1.4 文件压缩与归档 498
20.1.5 文件链接 503
20.1.6 安装文件系统 505
20.2 磁盘管理 507
20.2.1 硬盘简介 507
20.2.2 管理磁盘分区 509
20.2.3 使用磁盘管理命令 529
20.2.4 磁盘配额管理问题 539
第 21 章 Linux 系统安全 549
21.1 系统安全基础设置 549
21.1.1 BIOS 安全 549
21.1.2 账号安全 549
21.2 系统安全高级设置 552
21.2.1 重要文件 552
21.2.2 防止攻击系统 553
21.2.3 开启安全服务 555
21.2.4 审计日志 556
21.2.5 文件和目录许可 558

第 1 章　Linux 系统概述

用户在接触一种新的操作系统时，首先应该了解该系统的分类、特点以及配置方法等基础知识。在进行 Linux 操作系统的学习时，需要先从 Linux 的发展和基本操作等方面着手，在对 Linux 操作系统的特点以及应用有了一定的认识之后，用户就可以进入 Linux 操作系统高级应用的学习。

1.1　Linux 的发展

Linux 最初来源于 UNIX，但是其不仅进行了精简和优化，还整合了大量实用免费的源代码，在 UNIX 基础上有了很大的发展。Linux 操作系统在保证稳定性的同时，也很好地兼顾了普通用户的需求。

在了解 Linux 之前，需要先了解一下 Minix 操作系统、GNU 计划和 POSIX 标准，这是因为 Minix 操作系统、GNU 计划及 POSIX 标准与 Linux 操作系统息息相关。

1.　Minix 操作系统

Minix 的名称来源于 Mini UNIX，是由荷兰阿姆斯特丹自由大学计算机科学系的 Andrew S. Tanenbaum（AST）开发的一个迷你版的 UNIX 操作系统。该系统除了启动部分由汇编语言编写外，其余大部分均由 C 语言编写。

Minix 提供了用 C 语言和汇编语言编写的系统源代码，这使得开发操作系统的程序员有了参考的范本，可以根据实际情况编写属于自己的操作系统。

2.　GNU 计划

GNU 计划和自由软件基金会都是由 Richard M. Stallman 创办的。后者为前者提供了技术、法律和财政支持，而前者的发展又为后者提供资金的保证。为了保证 GNU 软件可以自由地“使用、复制、修改和发布”，所有的 GNU 软件都有一份在禁止其他人添加任何限制的情况下，将所有权利授给任何人的协议条款，即 GNU 通用公共许可证（GNU General Public License，GPL）。

GNU 计划推出之后获得了空前的发展，从编辑系统到调试程序，开发了各种高质量的软件，而这一切也为 Linux 的快速发展奠定了坚实的基础。

3.　POSIX 标准

POSIX（Portable Operating System Interface for Computing Systems）主要解决了 UNIX 版本多的混乱情况，给 UNIX 的开发提供一个通用的标准。POSIX 是由 IEEE（Institute of Electrical and Electronic Engineering）开发的，由 ANSI 和 ISO 标准化的一组标准。

现在，POSIX 已经发展成为一个非常庞大的标准簇。许多操作系统都支持 POSIX 标准，

比如 DEC OpenVMS 和 Microsoft Windows NT，也就是说 POSIX 标准并不局限于 UNIX。POSIX 被国际标准化组织（International Standards Organization，ISO）接受后，命名为 ISO/IEC 9945 标准，而 IEEE 标准则是 IEEE 1003。

POSIX 标准规范了 Linux 的发展，在 POSIX 标准下，很多其他操作系统下的软件都可以很方便地被移植到 Linux 下。

1.2 Linux 常见版本

Linux 从某种意义上来说只是操作系统的内核，并不是一个完整的操作系统。Linux 内核建立了计算机软件与硬件之间通信的平台，提供了文件管理、虚拟内存、设备 I/O 等系统服务。而通常意义上的 Linux 操作系统则是指采用了 Linux 内核的操作系统，比如 Red Hat Linux，Debian Linux，红旗 Linux 等。

1.2.1 Red Hat Linux

Red Hat 最早于 1995 年由 Bob Young 和 Marc Ewing 创建。原来的 Red Hat 版本早已停止技术支持，而目前的 Red Hat 的 Linux 分为两个系列，其中一个系列是由 Red Hat 公司提供收费技术支持和更新的 Red Hat Enterprise Linux，另外一个系列是由社区开发的免费的 Fedora Core，Red Hat Linux 的标志如图 1-1 所示。国内 Linux 用户最熟悉的 Linux 服务器发行版之一，就是 Red Hat Linux AS。

Fedora Core 是由 Red Hat 策划的完全开放的开发项目，标志如图 1-2 所示。Fedora 项目的目标是与 Linux 社区协作，用开放源码软件来创建一个完整的、通用的操作系统。其开发过程是以公开论坛的形式进行的，而原来的 Red Hat 工程组也将继续参与 Fedora Core 的开发，并且会加大邀请和鼓励外界参与的力度。



图 1-1　Red Hat Linux 标志

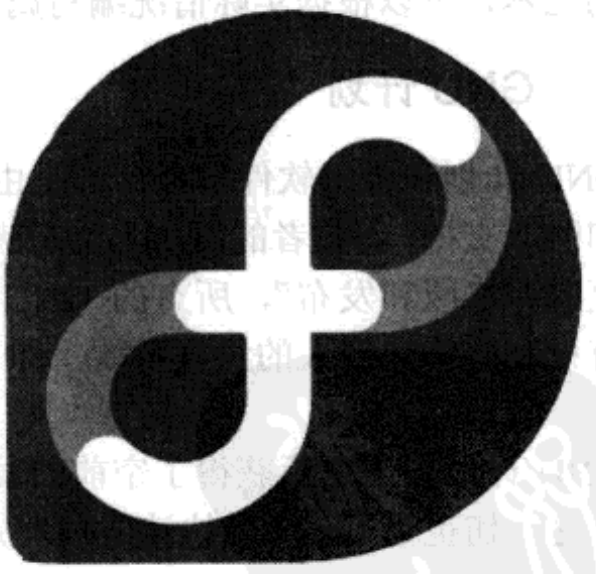


图 1-2　Fedora Core 5 标志

1.2.2 Debian Linux

目前，Debian Linux（也被称为 GNU Linux 或者简称为 Debian）提供了 8 000 多套已经编

译好的软件，按照统一的格式打包，用户可以免费获得并安装。可以说 Debian 是目前最纯粹的开源 Linux 发行版，是社区类 Linux 的领先者，标志如图 1-3 所示。Debian Linux 的软件包采用网络管理方式，系统的安装、升级和维护都比较简便。但不足之处在于，Debian Linux 对网络的依赖比较大，而且在安装时需要使用者具备一定的硬件专业知识，否则安装将遇到麻烦。

1.2.3 Ubuntu Linux

Ubuntu Linux 可以说是目前内容最为丰富的开源 Linux，是 Debian 的衍生物，不但具备了 Debian 的所有优点，而且简化了安装过程，更加方便普通用户使用。Ubuntu Linux 与其他 Linux 版本的不同在于“免费不能用来遮掩，您有权修正它，直到满意为止”，其标志如图 1-4 所示。目前，Ubuntu Linux 包含超过 16 000 种软件，几乎涵盖了所有的桌面应用程序，从文字处理、电子邮件到 Web 服务器及开发设计环境一应俱全。它的不足之处在于保留了 Debian 网络维护的方式。



图 1-3 Debian Linux 标志



图 1-4 Ubuntu Linux 标志

除了以上所说的几种 Linux 版本以外，还有 Mandriva Linux、MEPIS Linux、SUSE Linux、KNOPPIX Linux、Gentoo Linux、Slackware Linux 及 Damn Small Linux 等版本。

1.3 Linux 操作系统

从操作系统的功能来看，Windows 操作系统和 Linux 操作系统实现的功能是一致的，只是所用的形式不同。操作系统是一种用于管理计算机系统资源和控制程序执行的系统软件，可以扩充硬件的功能，为用户提供方便的使用接口和良好的运行环境。

1.3.1 Linux 操作系统结构

Linux 作为一个成熟的操作系统，系统结构都是比较完善的。它不仅继承了 UNIX 的精髓，而且得到了更大的发展。与 UNIX 的内核层和外壳层两部分结构不同，Linux 操作系统由 Linux

内核、Linux Shell 和文件结构 3 个部分组成。

1. Linux 内核

内核负责管理系统的各种资源，比如进程、内存、设备驱动程序、文件和网络系统等，其决定了系统的性能和稳定性。

（1） Linux 内核简介

Linux 的内核源程序遵循 GPL 规定，都是开放的，可以在系统目录/usr/src/linux 下找到。而基于 Linux 的大部分应用软件也遵循 GPL，普通用户同样可以获得相应的源代码。Linux 内核最大的好处就是用户可以通过源程序了解操作系统是如何工作的。通过阅读源代码，程序员可以了解操作系统的工作原理，甚至可以根据自身环境重新编译 Linux 的内核，定制一套符合自己工作习惯的操作系统。

（2） 内核版本号

由于 Linux 的源程序是完全公开的，任何人只要遵循 GPL 就可以对内核加以修改并发布。带来便利的同时也会带来一个问题，那就是哪个版本的 Linux 是原始的版本，哪个版本是其他人开发的呢？为此，Linux 开发人员对 Linux 版本采用了双树系统，即一个是稳定树，另一个是非稳定树。一些新特性、实验性改进首先在开发树中进行，如果在开发树中所做的改进也可以应用于稳定树，那么在开发树中的改进经过测试以后，在稳定树中将进行相同的改进。

开发树与稳定树的区别主要是版本号，一般 Linux 的版本号的形式为 x.y.z。如果是稳定树，那么 y 是偶数，而开发树的 y 值比相应的稳定树大 1（奇数）。

通常新内核会支持更多的硬件，具有更好的进程管理能力，运行速度更快、更稳定。另外，新内核一般还会修复老版本中的许多漏洞。Linux 内核版本发布的官方网站是 http://www.kernel.org。

2. Linux Shell

Shell 是 Linux 操作系统的用户界面，为用户与内核间的交互操作提供接口，接收用户命令并进行编译，然后再送到 Linux 内核执行。

Shell 是一个命令解释器，也可以看作是一种编程语言，用户可以利用 Shell 命令编写程序。与普通编程语言相似，Shell 编程语言也可以使用循环结构和分支控制结构。

Linux 操作系统刚开始发布的时候，大多数的命令都是通过命令行来实现的。随着 Windows 图形界面在用户中的不断普及，Linux 也提供了类似 Microsoft Windows 的图形界面 KDE（King Desktop Environment，桌面环境之王）和 GNOME（GNU Network Object Model Envirment）。这是 Linux 操作系统中最常见的两种窗口管理器，为用户提供了类似 Windows 操作系统的工作环境，通过鼠标就可以完成大多数的系统管理操作。

需要说明的是，Linux 实际上只是一个基于命令行的操作系统，图形界面并不是其组成架构，只是系统的扩展。这与 Windows 操作系统有很大的差别，尤其是 Windows 95 之后的操作系统。此类 Windows 操作系统如果没有图形界面，那么就不能被称为“Windows”。

每个 Linux 系统用户都可以拥有独立的用户界面或者 Shell，也就是说 Linux 的 Shell 及用户界面是可以定制的。Shell 具有多个版本，常见的有以下几个。

- Bourne Shell：是由贝尔实验室开发的。
- BASH：是 GNU 的 Bourne Again Shell，是 GNU 操作系统上默认的 Shell。

- Korn Shell：是对 Bourne SHell 的发展，在内容上基本与 Bourne Shell 兼容。
- C Shell：是 SUN 公司 Shell 的 BSD 版本。

3. Linux 文件结构

文件在磁盘等存储设备上的组织存储方式称为文件结构。任何一种操作系统都是由很多个文件组成的，而这些文件又自成体系，有自己的构成方式。

目录为文件管理提供了一种方便而有效的途径，用户不仅能够从一个目录切换到另一个目录，还可以设置目录和文件的权限以及共享程度。

系统管理员及其他用户可以设置目录和文件的权限，以便允许或拒绝其他人对其进行访问。Linux 目录采用多级树形结构，用户可以浏览整个系统，可以进入任何一个已授权进入的目录，访问那里的文件。

Linux 是一个多用户系统，文件结构的相互关联性使共享数据变得容易，几个用户可以访问同一个文件。

内核、Shell 和文件结构一起形成了操作系统的基本结构。它们使得用户可以运行程序，管理文件以及使用系统。此外，Linux 操作系统还有许多被称为实用工具的程序，辅助用户完成一些特定的任务。

1.3.2 Linux 操作系统特点

Linux 操作系统在推出之后得到了迅猛的发展，这与其良好的性能分不开。Linux 操作系统主要具有以下特性。

1. 开放性

Linux 操作系统的兼容性很高，遵循世界标准规范，特别是开放系统互联（OSI）国际标准，而微软的视窗操作系统源代码基本保密。

2. 多任务

在多核处理器时代，Linux 操作系统的一大优势是：系统调度的每一个进程都可以平等地访问微处理器。而因为 CPU 的处理速度非常快，所以各个应用程序看起来就像是同时运行的。

3. 多用户

Linux 是一个多用户操作系统，系统资源可以被不同用户各自拥有使用，而每个用户对自己的资源（例如文件、设备）都有特定的权限，互不影响。

4. 设备独立性

Linux 可以把所有外部设备统一当作文件，只要安装相关的驱动程序，任何用户都可以像使用文件一样使用这些设备。另外，用户可以免费得到 Linux 的内核源代码，这样，用户便可以自己修改操作系统的内核源代码，以适应新增加的外部设备。

5. 提供了丰富的网络功能

一般的操作系统并没有内置的网络连接功能，也不具备联网的灵活性。而 Linux 操作系统具有完善的与内核结合在一起的网络功能，在通信和网络功能方面优于其他操作系统。

6. 费用低廉

与微软的 Windows 操作系统相比，Linux 系统的使用费用几乎可以忽略。由于 Linux 基于 GPL，系统本身免费，而且不少发行套件包含了丰富的应用软件，基本上都可以满足大多数用户的使用需求。

1.3.3 Linux 操作系统的应用领域

Linux 操作系统常用于基础服务器应用，如 DNS 服务器、DHCP 服务器、Web 服务器、目录服务器、防火墙、文件和打印服务器、Intranet 代理服务器等。另外，Linux 操作系统还用于高性能计算、密集型计算应用等方面，如风险分析、数据分析、数据建模等。

Linux 操作系统从第一个内核诞生到现在，其开放、安全、稳定的特性得到越来越多用户的认可。自由软件的低成本、高安全性，又使得 Linux 得到了各国政府强有力的支持。Linux 的应用领域逐渐扩展，从最早的 Web、FTP、邮件服务，到个人桌面应用、网络安全、电子商务、远程教育、集群运算、网格运算、嵌入式系统等各个领域。

很多对 Linux 操作系统比较陌生的用户，可能会认为其应用领域比较窄。实际上，自 Linux 推出以后，其应用已经遍及服务器、嵌入式操作系统、桌面系统等各个领域，成为了用户工作、生活的良好助手。

1. 服务器系统

目前，Linux 应用最广泛的是服务器操作系统领域。以 Linux 为基础的“LAMP（Linux，Apache，MySQL，Perl/PHP/Python 的组合）”技术，已在开发者群体中广泛流行，也是网站服务供应商最常使用的平台。

2. 桌面系统

新版本的 Linux 操作系统特别在桌面应用方面进行了改进，完全可以作为一种集办公应用、多媒体应用、网络应用等多功能于一体的图形界面操作系统，而且它也是免费的。

3. 嵌入式系统

目前，Linux 操作系统的一项非常重要的应用是作为移动设备的嵌入式系统。例如机顶盒、移动电话及行动装置等，都使用了嵌入式 Linux 操作系统。由于 Linux 内核完全公开，大大降低了基于 Linux 的嵌入式系统的研发成本，而且不受其他非人为因素的干扰。

4. 电子政务

由于众多 IT 知名厂商的支持以及安全性方面的优势，Linux 正在成为一股可以与 Windows 分庭抗衡的力量，并且广泛地应用于政府领域。目前许多国家的电子政务系统都建立在 Linux

平台上，如墨西哥政府已经宣布在全国所有的中、小学配置 Linux 并为学生提供 Internet 服务。中国政府在 Linux 领域一直给予极大的支持，由工业和信息化部、科技部、国家信息化推进工作办公室等相关部门牵头，业界企业、大专院校、科研机构等参与，目标是加强 Linux 软件标准建设的一系列重要举措。这些重要举措是：成立 Linux 软件标准工作组、建立国家 Linux 公共服务平台体系并出台软件政府采购管理办法，如红旗 Linux，已经获得了中国政府的认可。

1.3.4 Red Hat Enterprise Linux 5

Red Hat Enterprise Linux 5 发布于 2007 年 3 月 14 日，是 Red Hat 商业操作系统的第 5 个重要版本，新版本的主要变化是内核升级为 2.6.18，开始支持 Xen 虚拟化技术、集群存储等，如图 1-5 所示。

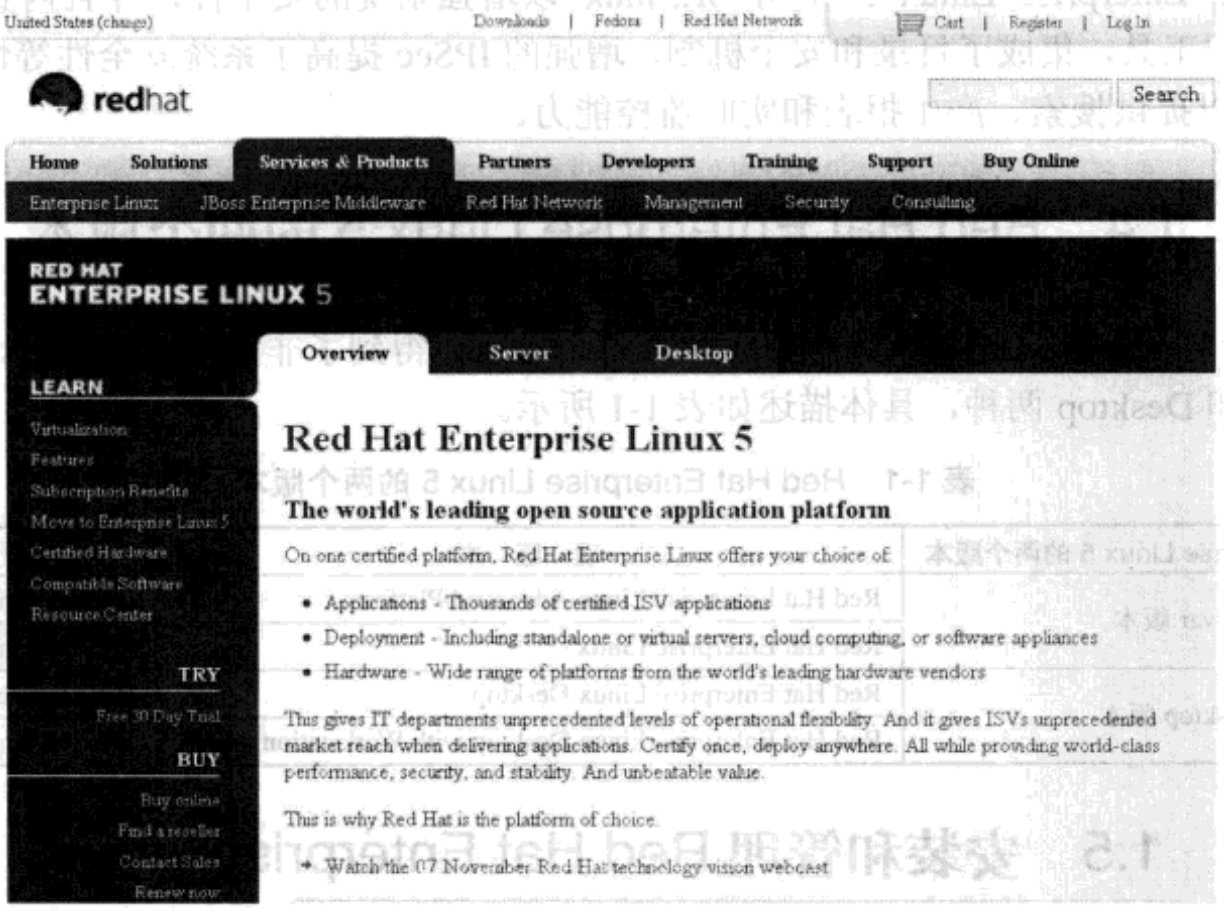


图 1-5 Red Hat Enterprise Linux 5

Red Hat Enterprise Linux 主要有以下特性。

1. 性能强

Red Hat Enterprise Linux 5 基于新的 2.6.18 内核，对于多内核处理器的支持更完善，并支持 Intel Network Accelerator Technology（IOAT），增强了基于 Kexec/Kdump 的 Dump 支持，增强了对于大型 SMP 系统的支持，增强了管道缓存。

2. 虚拟化技术

支持在各种平台上的虚拟化技术，在 Red Hat Enterprise Linux Advanced Platform 上甚至支持了存储与扩展的服务器虚拟化技术，还提供了 virt-manager、libvirt/virsh 管理工具。

3. 网络与存储

Red Hat Enterprise Linux 5 支持 Autofs、FS-Cache 和 iSCSI，让用户可以更方便地管理存储设备，增强了对于 IPv6 的支持。此外，还改进了对于 Microsoft 文件/打印和 Active Directory（目录服务）的支持。

4. 管理

Red Hat Enterprise Linux 5 首先简化了安装过程的系统配置，采用基于 Yum/Pup 的 Red Hat Network 更新以方便修补系统漏洞，此外，它还采用了更好用的 Conga 集群和存储管理。

5. 安全

Red Hat Enterprise Linux 5 采用 SELinux 以增强系统的安全性，并且内置了图形化的 SELinux 管理工具，集成了目录和安全机制，增强的 IPSec 提高了系统安全性等性能，新的审核机制还可以提供搜索、产生报表和实时监控能力。

1.4 Red Hat Enterprise Linux 5 的两个版本

Red Hat Enterprise Linux 5 具有开源和实用的特点，得到了非常广泛的运用。其版本主要包括 Sever 和 Desktop 两种，具体描述如表 1-1 所示。

表 1-1 Red Hat Enterprise Linux 5 的两个版本

Red Hat Enterprise Linux 5 的两个版本	现 版 本	前 版 本
Sever 版本	Red Hat Enterprise Linux Advanced Platform	Red Hat Enterprise Linux AS
	Red Hat Enterprise Linux	Red Hat Enterprise Linux AS
Desktop 版本	Red Hat Enterprise Linux Desktop	Red Hat Desktop
	Red Hat Enterprise Linux Desktop with Workstation option	Red Hat Enterprise Linux WS

1.5 安装和管理 Red Hat Enterprise Linux

Red Hat Enterprise Linux 的安装方法有多种，用户可以根据实际情况选择系统的安装方式，比如从光盘安装或者是用镜像文件从硬盘安装等，其中较为常用的安装方法是从光盘安装。本节将以 Red Hat Enterprise Linux 5 为例，详细介绍 Linux 系统的安装、配置和卸载的方法。

1.5.1 安装前的准备

为了使系统的安装能够顺利进行，在安装 Red Hat Enterprise Linux 5 操作系统之前，需要先确定当前配置所需的发行版本、硬件兼容性，另外为了安全起见，最好先对硬盘资料进行备份。

在新计算机上安装 Red Hat Enterprise Linux 5，只需在驱动器中插入安装光盘，将计算机的 BIOS 设置为从光盘启动，然后重新启动计算机，Red Hat Enterprise Linux 5 安装程序便会启动并自动检测大多数硬件。

1. 硬件要求

为保证操作系统正常运行，在安装任何操作系统之前，都应该了解其最低硬件需求，安装 Red Hat Enterprise Linux 5 系统的基本配置要求如表 1-2 所示。

表 1-2 安装 Red Hat Enterprise Linux5 硬件要求

硬件	CPU	内　存	硬　盘	显　卡	主　板	其他（如键盘、鼠标等）
配置	Pentium 处理器及更高	对于 X86、AMD64/Intel64 和 Itanium2 架构的主机，最少需要 512 MB 的内存，如果主机是 IBM Power 系列，最少需要 1 GB 的内存（推荐 2 GB）	20 GB 或更多磁盘空间	需要 VGA 兼容显卡	任何主板基本上都能与 RedHat Enterprise Linux 5 兼容，一般不会出现问题	兼容声卡和网卡等。光驱是 CD-ROM 或者 DVD 光驱

Red Hat Enterprise Linux 5 支持目前市场上大多数的主流硬件，用户如果想知道自己的硬件设备是否被 Red Hat Enterprise Linux 5 支持，可以访问 Red Hat 硬件认证网页（https://hardware.Red Hat.com/），在其中可以查看哪些硬件通过了 Red Hat Enterprise Linux 5 的认证，如图 1-6 所示。

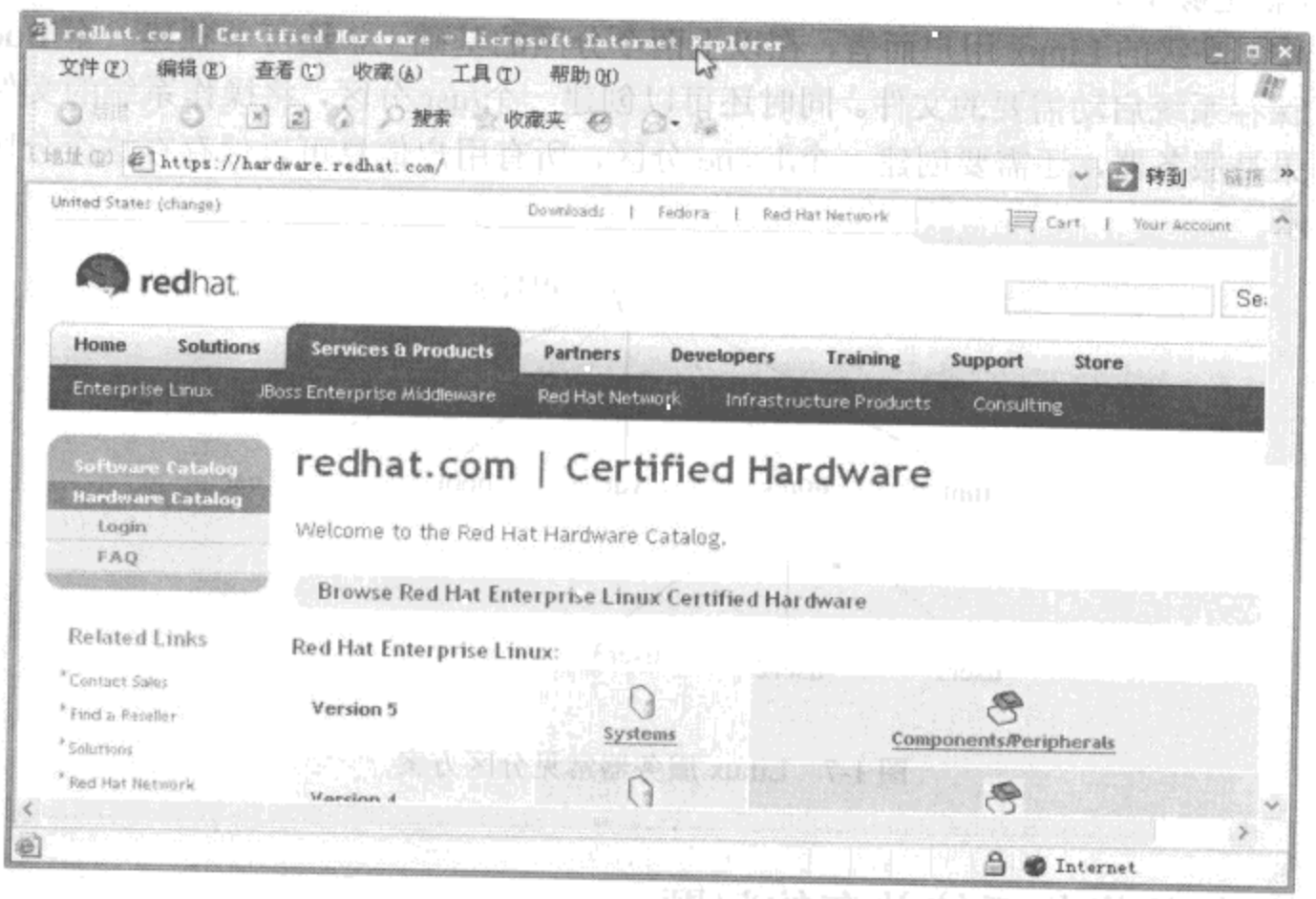


图 1-6 Red Hat 认证的硬件

2. Red Hat Enterprise Linux 5 的 3 种安装方式

Red Hat Enterprise Linux 5 提供了 3 种安装方式，用户可以通过 CD-ROM/DVD 启动安装，也可以通过硬盘安装或者 NFS 服务器安装，甚至可以通过 FTP 或 HTTP 服务器来安装。

（1） CD-ROM/DVD 安装方式

将光驱设置为优先启动，然后将 Red Hat Enterprise Linux 5 安装光盘放入光驱，启动电脑后就可以进入安装向导。

（2） 硬盘安装方式

此类安装方式适合那些从网上下载到光盘镜像，但没有条件刻盘的用户。此类安装也很简单，将下载的 ISO 镜像文件复制到 FAT32 或者 Ext2 分区中，在安装过程中选择硬盘安装，然后选择镜像所在位置即可。

（3） 网络服务器安装方式

此类安装方式对于网络速度要求比较高，目前 Red Hat Enterprise Linux 5 的网络安装支持 NFS，FTP 和 HTTP 3 种方式。这里需要注意的是，通过网络安装 Red Hat Enterprise Linux 5 时，光驱中最好不要放置安装光盘，否则可能会引发一些不知名的错误。

3. Linux 系统分区规划

建议初级 Linux 用户，不要规划太过复杂的分区，一般分两个区就好，一个是用于保存系统和数据的根分区，另外一个是交换分区。交换分区的大小与物理内存同大即可，而根分区则需要根据 Linux 系统安装占用资源的大小和需要保存数据的多少来调整（一般情况下，划分 15~20 GB 就足够了）。

对于中、高级的 Linux 用户而言，在以上两个分区的基础上，还可单独建一个 Boot 分区，主要用于保存系统启动需要的文件。同时还可以创建一个/usr 分区，将操作系统的文件安装在这里。如果是服务器，还需要创建一个/home 分区，所有用户信息可以保存在这个分区下，如图 1-7 所示。

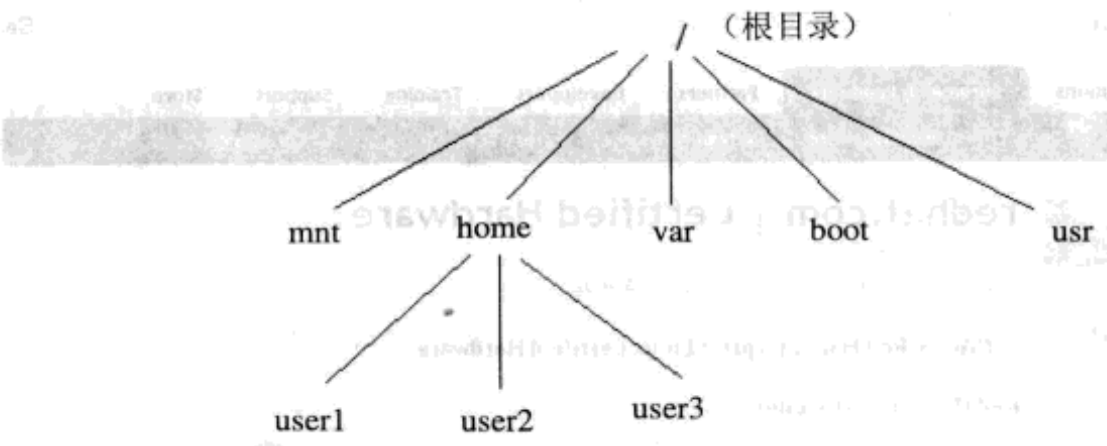


图 1-7 Linux 服务器常见分区方案

1.5.2 与其他操作系统并存的问题

GRUB 多重启动管理器提供了在一台计算机中实现多操作系统并存的服务，其可以引导 FreeBSD、OpenBSD、DOS 和 Windows 等操作系统。计算机启动时，用户可以使用 GRUB 提供的菜单选择需要启动的系统。

硬盘的说明如下。

- 大多数微软系统要安装在主分区中，而一个硬盘最多只有 4 个主分区。为了得到更多

的分区，用户必须要放弃一个主分区，而把它作为一个扩展分区。在 Linux 中，一个 Pata（或者 IDE）硬盘最多可以有 63 个分区，对于 Sata 或 SCSI 的限制是 15 个。

- 扩展分区也消耗一个设备名称，只是定义逻辑分区开始和结束点的边界，不能用于存储。
- Linux 在主分区和逻辑分区中都可以安装和引导。
- 在任何一个硬盘上，Linux 通常把逻辑分区的每一个分区命名为第 5 分区。因此，理论上一个 Pata 硬盘（也就是 hda）中，主分区最多可以使用 hda1，hda2 和 hda3 3 个名称，hda4 作为扩展分区最多支持 59 个逻辑分区（从 hda5 到 hda63）。
- 实际工作中，Linux 一般不显示第 60 个以后的分区，红帽家族所发行的版本甚至不显示第 15 个以后的分区。

知识点 什么是 Swap 分区？

Swap 分区即交换区，系统在物理内存不足时进行 Swap 交换。当系统的物理内存不足时，将物理内存中的一部分空间释放出来，以供当前运行的程序使用，被释放的空间被临时保存到 Swap 空间中。Swap 的调整对 Linux 服务器的性能至关重要，有时甚至可以越过系统性能瓶颈，节省系统升级费用。

1.5.3 Red Hat Enterprise Linux 的安装阶段

操作系统安装最常见的方式是用光盘安装，下面就以光盘安装为例来说明安装 Red Hat Enterprise Linux 5 的具体步骤。

将 Red Hat Enterprise Linux 5 安装光盘放入光驱，然后重新启动计算机。在 BIOS 中设置第一个引导设备为“CD-ROM”，如图 1-8 所示。

完成以上设置后，再次重启计算机，然后通过以下步骤安装 Red Hat Enterprise Linux 5。

第 1 步，重新启动计算机后，显示如图 1-9 所示的 Red Hat 安装界面。



图 1-8 设置从光盘启动安装 Linux

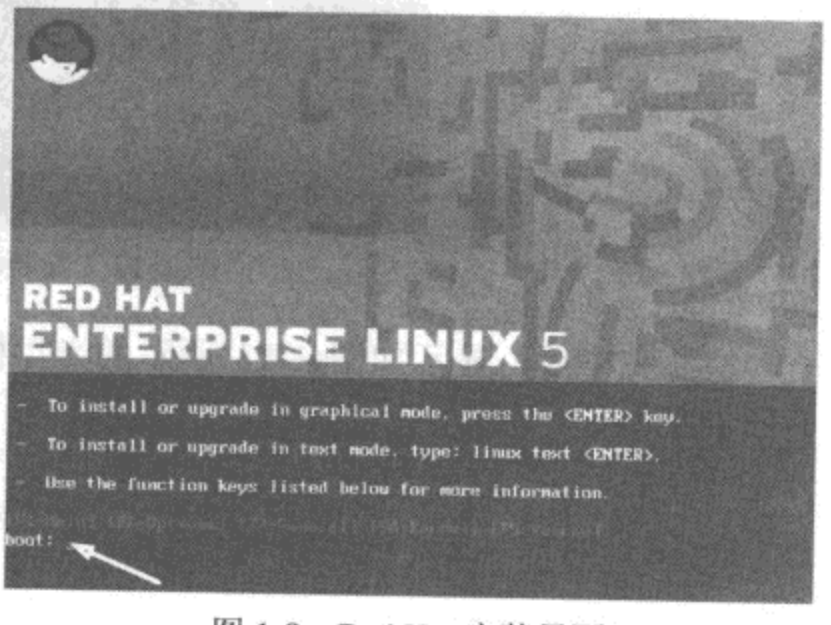


图 1-9 Red Hat 安装界面

此时，用户可以使用快捷键选择进入模式（比如按下 F2 键将会出现更多选项，按下 F3 键显示常规配置，按下 F4 键显示内核参数，按下 F5 键显示急救模式），如果是第一次安装，直接按 Enter 键即可，如图 1-10 所示。

注意 前文提到了硬盘或者网络安装 Red Hat Enterprise Linux 5，此时可以在“Boot:”提示符下输入“linux askmethod”，然后在弹出的窗口中选择从本地硬盘、网络，还是光盘进行安装，如图 1-11 所示。

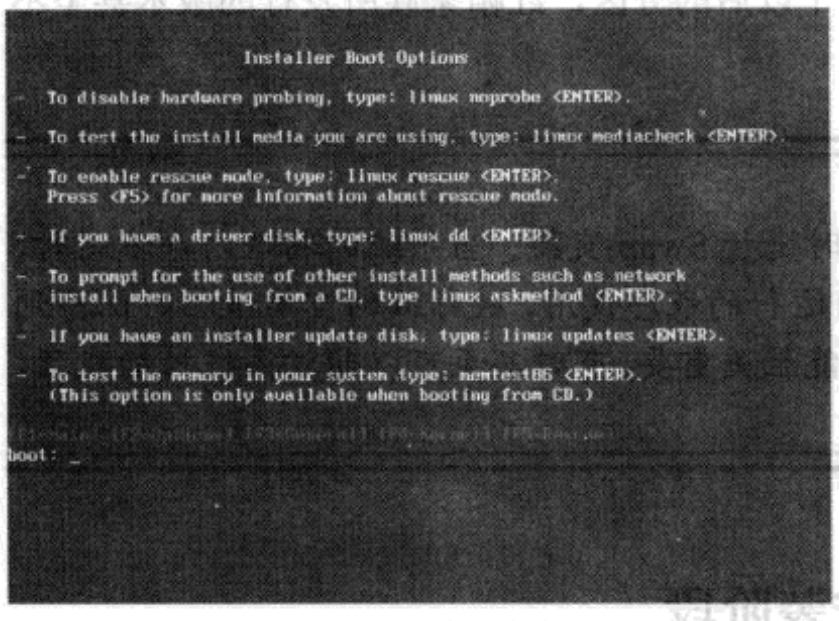


图 1-10 定制安装选项

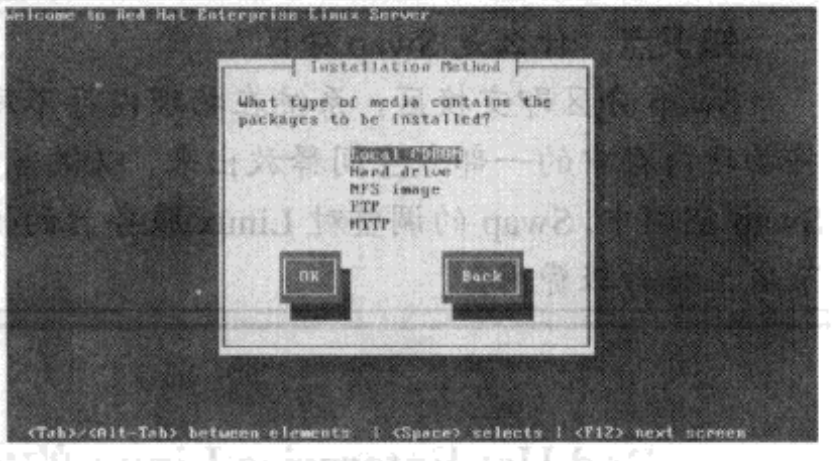


图 1-11 选择安装方式

第 2 步，进入检测计算机硬件系统阶段。

在“Boot”提示符下按 Enter 键，安装程序便会自动检测当前计算机的硬件系统，并会在屏幕上提示相关的信息，比如硬盘、CPU、串行设备等，如图 1-12 所示。

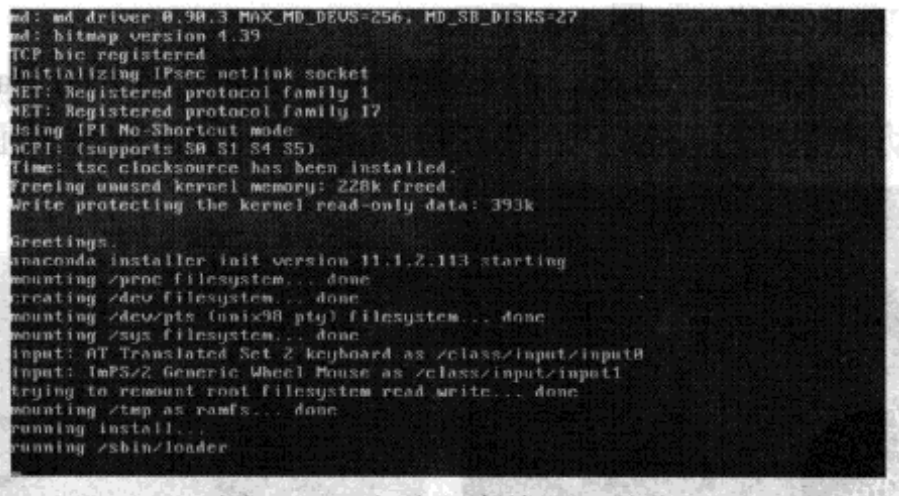


图 1-12 安装程序检测硬件中

检测完毕后，显示如图 1-13 所示的光盘检测窗口。在该窗口中单击“Skip”按钮，进入下一步操作。

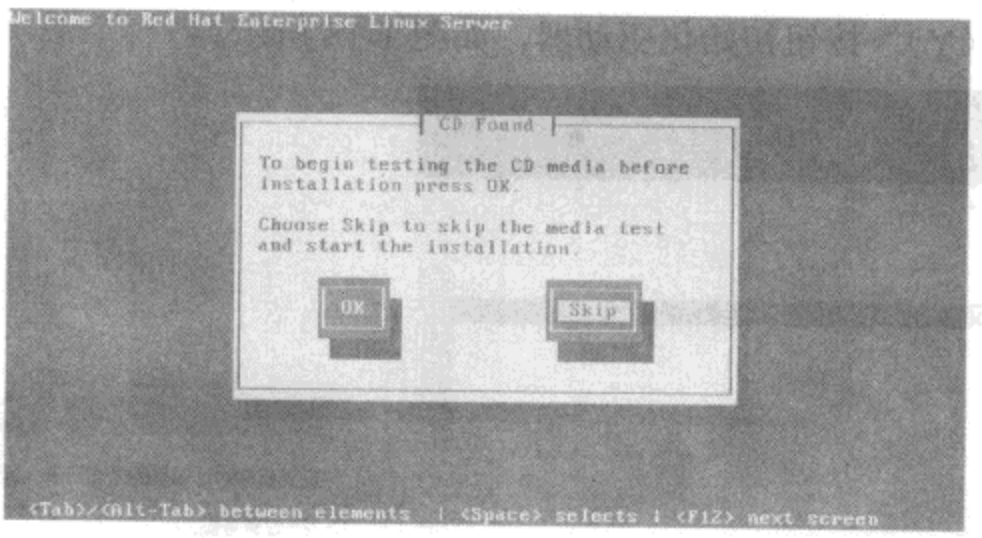


图 1-13　光盘检测窗口

注意　光盘检测窗口是为防止安装光盘出错而专门设计的。这是因为很多用户的 Linux 安装盘都是从网上下载的，这个选项可以检测安装光盘是否有问题。

第 3 步，进入选择安装语言和键盘设置阶段。

当硬件通过检测后，进入 Linux 的图形化安装阶段，显示如图 1-14 所示的欢迎界面。

单击“Next”按钮显示语言设置界面。Red Hat Enterprise Linux 5 本身就内置了数十种语言，用户可以选择自己所需的语言种类，此处选择“简体中文”项，然后单击“Next”按钮，如图 1-15 所示。

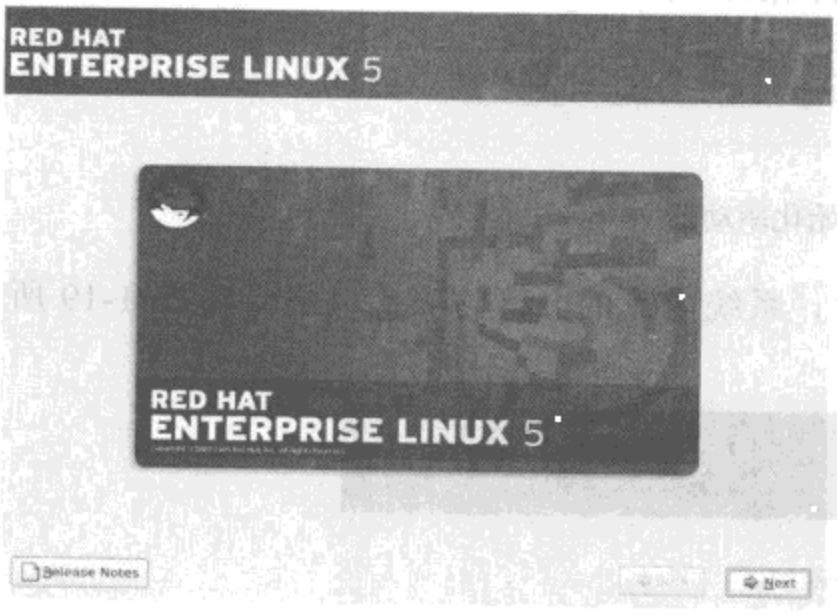


图 1-14　欢迎界面

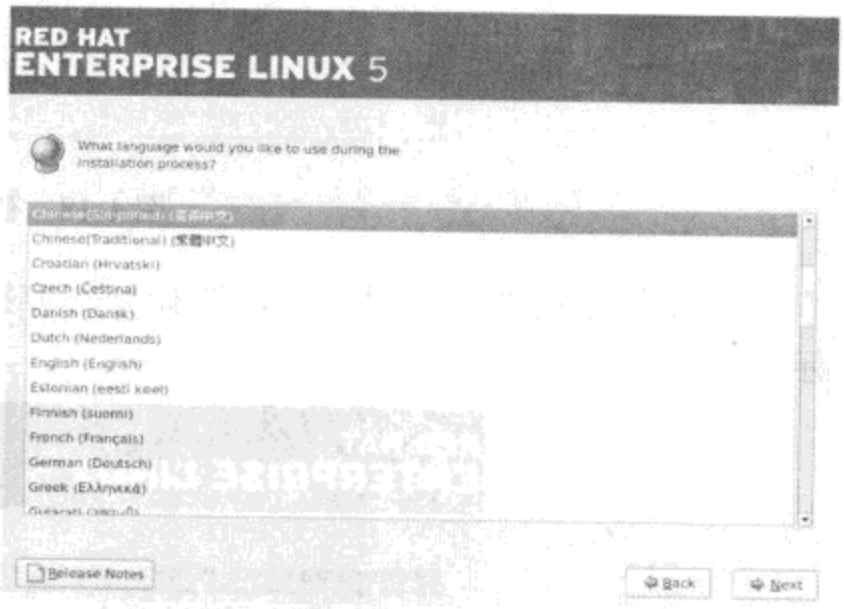


图 1-15　语言设置界面

接着键盘布局选择窗口将以中文方式显示，在该窗口中选择“简体中文”项，如图 1-16 所示。

第 4 步，选择好键盘布局之后，显示如图 1-17 所示的“安装号码”对话框，要求用户输入“安装号码”，该号码与 Windows 的安装序列号类似。在 Red Hat Enterprise Linux 5 中即使跳过输入安装号码，也能够顺利完成系统的安装，不过系统的功能会受到一定的限制，而且无法得到 Red Hat 的技术支持。

第 5 步，进入硬盘分区方案，对硬盘进行分区。

首先，如果是在全新硬盘上安装 Red Hat Enterprise Linux 5，安装程序将提示用户初始化

驱动器，单击“是（Y）”按钮初始化驱动器，如图 1-18 所示。

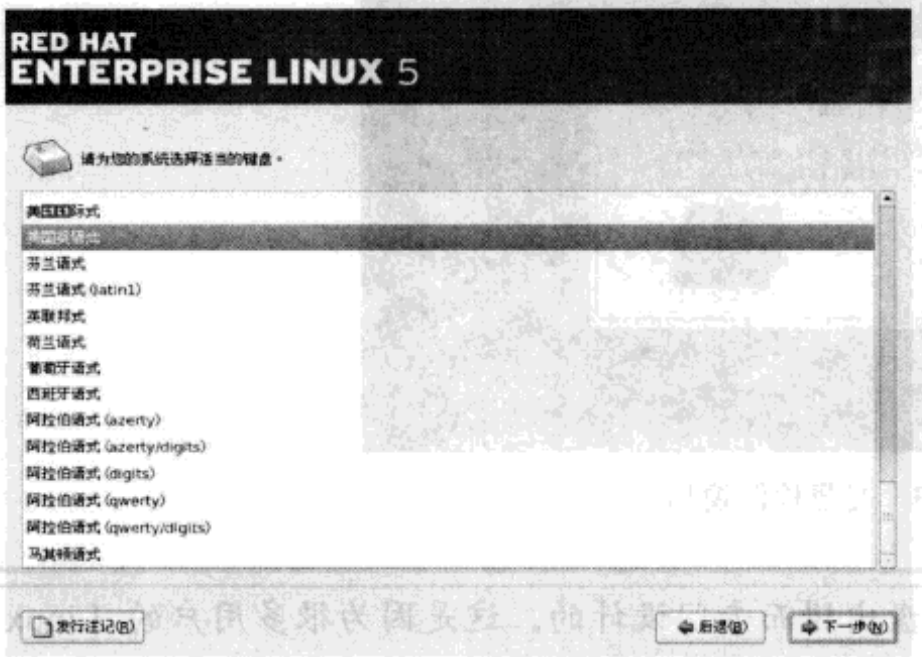


图 1-16　键盘布局选择窗口

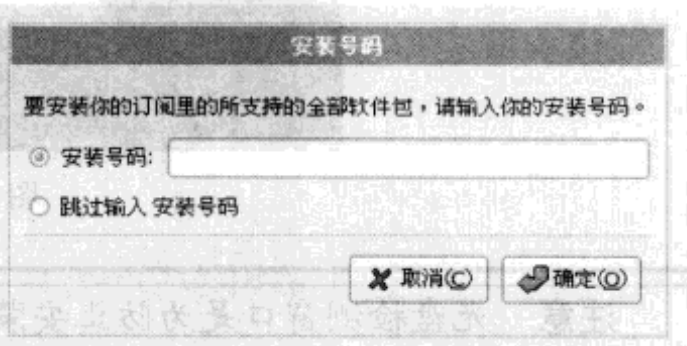


图 1-17　“安装号码”对话框

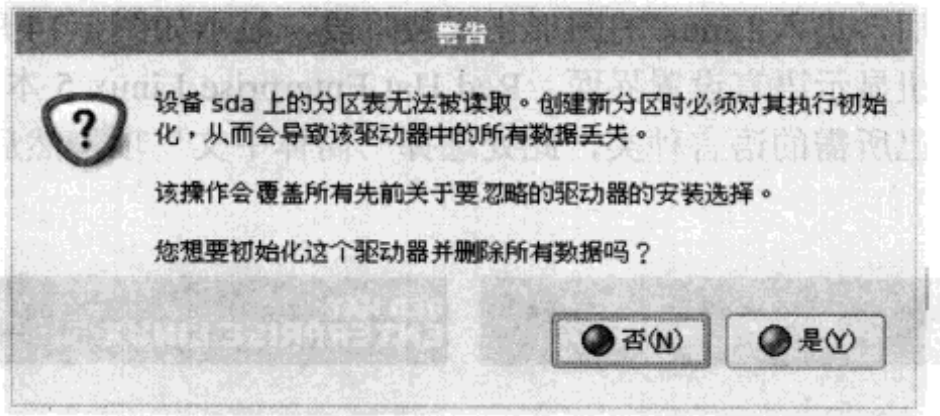


图 1-18　初始化驱动器

初始化驱动器后，用户可以选择分区方案，系统内置的分区方案有 4 种，如图 1-19 所示。

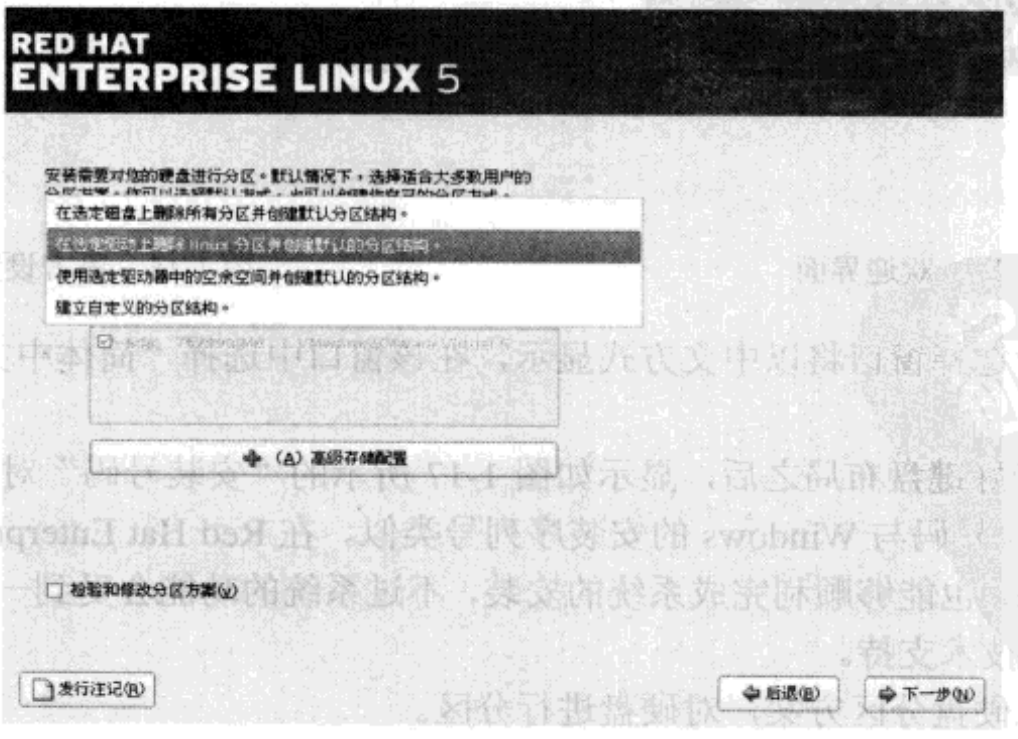


图 1-19　4 种分区方案

各种分区方案的描述如下。

- 在选定的磁盘上删除所有分区并创建默认分区结构

选择该选项后，Linux 将会删除原来硬盘上的所有数据，包括 Windows 的 VFAT、NTFS 分区。适合于整个磁盘上只安装一个 Linux，并且硬盘上不存在其他重要数据的情况。

注意 选择该选项将会导致原来硬盘上的所有分区和数据被删除。建议一般用户不要选择此选项，在数据得到妥善备份之后，再使用此选项。

- 在选定的驱动器上删除 Linux 分区并创建默认的分区结构

选择该选项之后，系统将会删除硬盘上原有的 Linux 分区，但其他操作系统创建的分区比如 Windows 的 VFAT、NTFS 将会得到保留。该选项适用于多操作系统的安装。

- 使用选定的驱动器中的空余空间并创建默认的分区结构

该选项将会保留原来硬盘上的所有数据和分区，前提是硬盘上有足够的空间用来安装新的 Linux 系统。不过现在的硬盘容量都比较大，建议初级用户使用该方法。

- 建立自定义的分区结构

所有的操作都将由用户手工来完成，这一选项适合高端 Linux 用户，对于初次接触 Red Hat Enterprise Linux 5 的用户而言，不建议采用这种方式分区。

注意 建议用户选中分区方案选择窗口下方的“检验和修改分区方案”复选框，这样可以动态地查看将要采用的分区方案的效果。

此时，如果用户选择了 3 种自动分区方案中的一种，并且选中“检验和修改分区方案”复选框，则显示如图 1-20 所示的分区预览窗口。在该窗口中，用户可以单击“新建”按钮，创建一个新分区，单击“编辑”按钮，修改选中的分区，单击“删除”按钮删除不符合要求的分区，单击“重设”按钮，将硬盘恢复到硬盘原始状态。

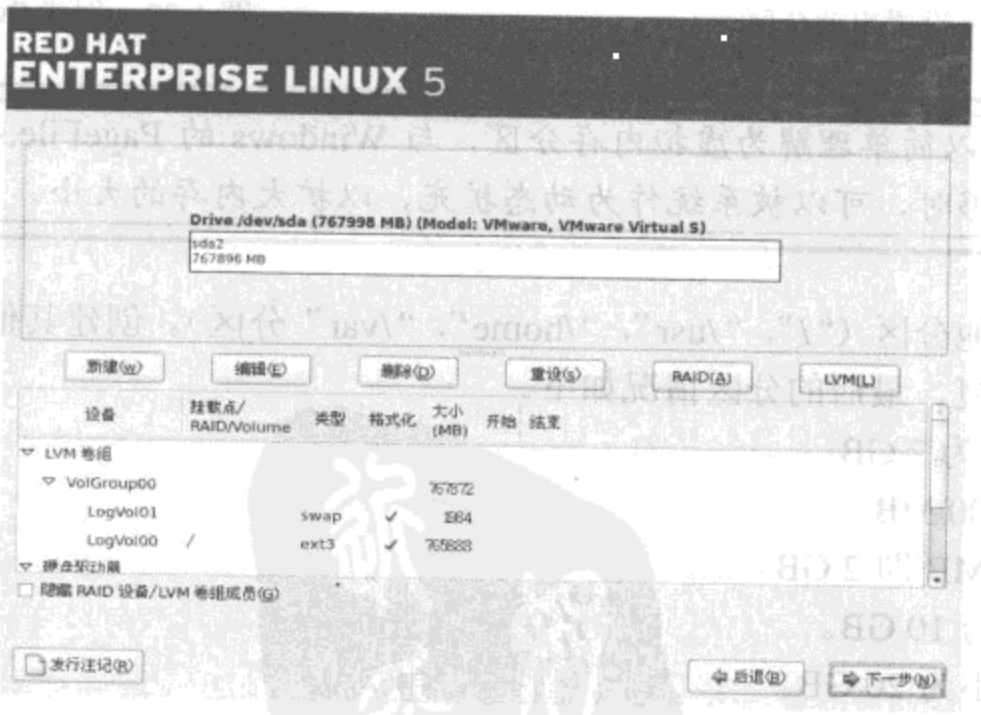


图 1-20 分区预览窗口

然后，创建 boot 分区和交换分区。创建 boot 分区时选中原来的分区，逐个删除，然后开始重建分区。

注意 因为 Red Hat Enterprise Linux 5 默认采用的是 LVM 分区格式，这是特殊的磁盘分区方式，如果使用的是单硬盘，建议用户采用普通的分区格式，并且需要删除重建分区。

新建 boot 分区需要进行如下工作。

单击“新建”按钮，显示如图 1-21 所示的“添加分区”对话框。在“挂载点”下拉列表框中选择“/boot”项，在“文件系统类型”下拉列表框中选择“ext3”项，大小设置为 100 MB（在“大小”框中输入 100，默认单位是 MB），其他的按照默认设置即可。

重新创建交换分区的具体操作如下。

单击“新建”按钮，显示“添加分区”窗口，在“文件系统类型”下拉列表框中选择“swap”项，大小一般设置为物理内存的两倍，如果用户的计算机物理内存大小为 1 GB，那么设置 Swap 分区的大小为 2 GB（2 048 MB）即可，如图 1-22 所示。

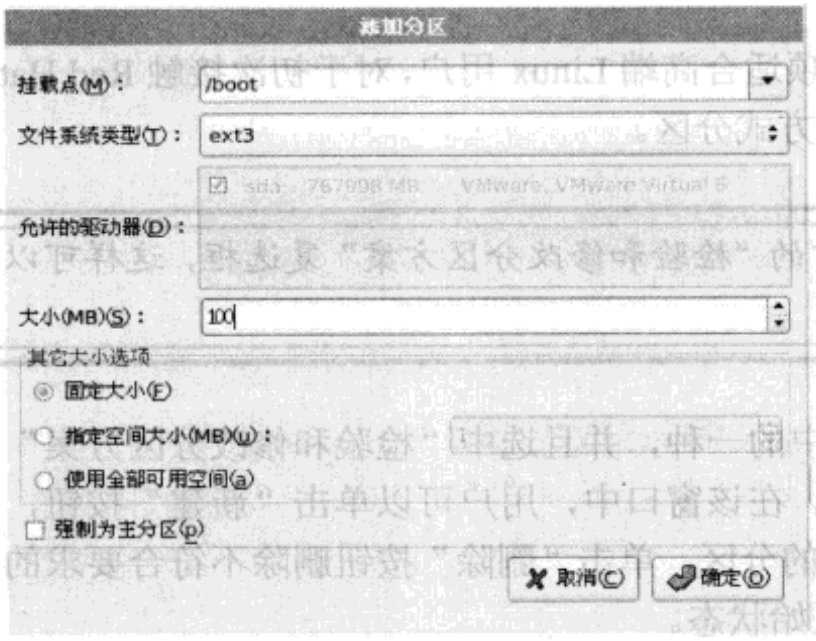


图 1-21 设置启动分区

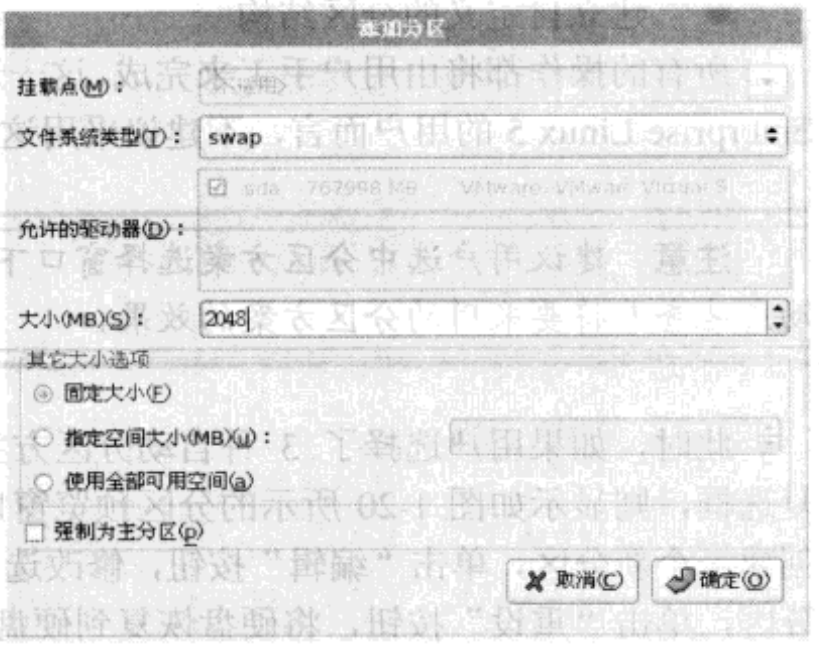


图 1-22 创建 Swap 分区

注意 swap 可以简单理解为虚拟内存分区，与 Windows 的 PageFile.sys 页面类似。当计算机的物理内存不够时，可以被系统作为动态扩充，以扩大内存的大小。

最后，创建其他分区（“/”，“/usr”，“/home”，“/var”分区）。创建其他分区可以重复以上步骤，这里不再重复。最后的分区情况如下。

- swap 大小为 2 GB。
- /boot 给 100 MB。
- /给 2 048 MB 即 2 GB。
- /usr 大小为 10 GB。
- /home 大小为 20 GB。
- /var 大小为 5 GB。

剩下的空间用来创建了一个数据分区和备份分区。

第 6 步，在完成硬盘分区操作之后，Red Hat Enterprise Linux 5 将安装配置引导程序。在 Linux 中有两种引导管理器，即 LILO（使用得比较少）和 GRUB（较多使用）。

这里使用默认安装，选择把 GRUB 安装到 MBR，如图 1-23 所示。这样可以实现 Windows 和 Linux 的双系统启动。

第 7 步，安装 GRUB 后，Red Hat Enterprise Linux 5 将自动检测网络设备，然后将网络设备显示在“网络设备”列表框中，如图 1-24 所示。

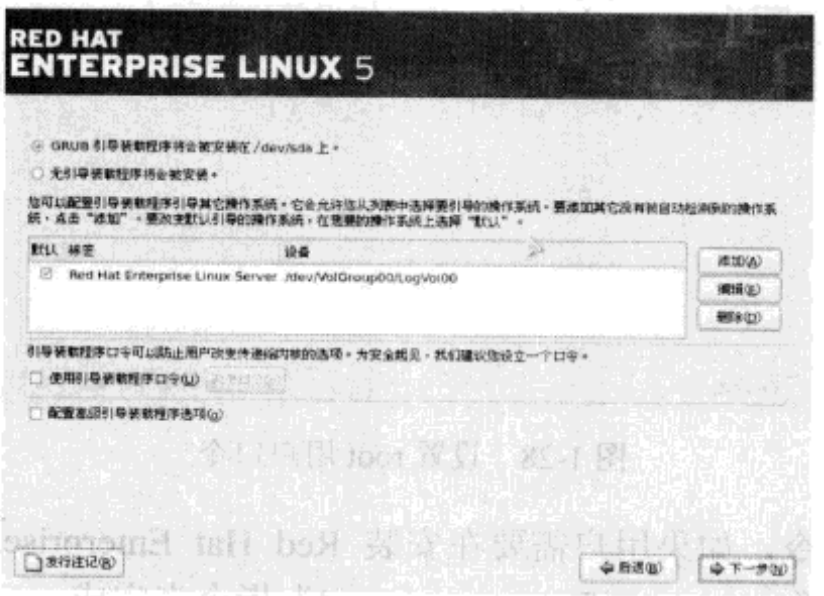


图 1-23　选择 GRUB 安装位置

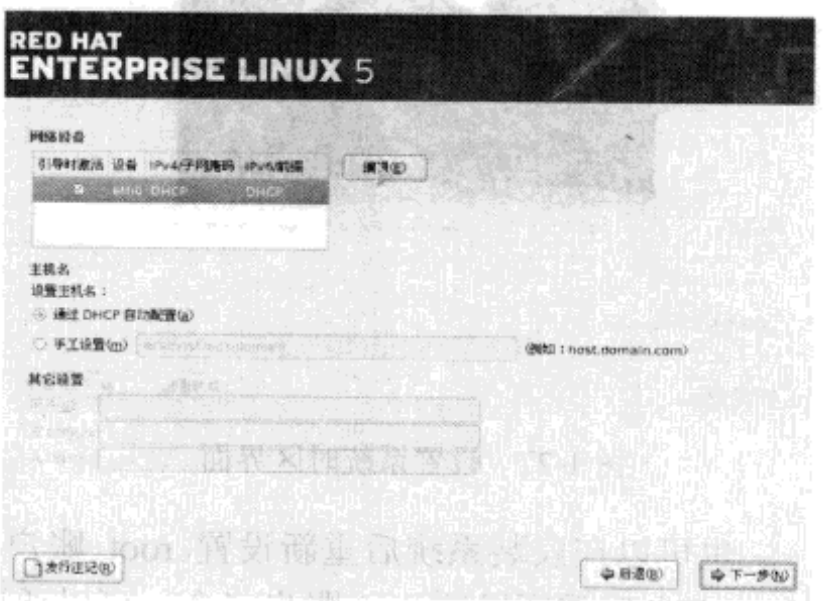


图 1-24　“网络设备”窗口

一般情况下，Red Hat Enterprise Linux 5 使用 DHCP 动态获取 IP 地址，如果用户不需要使用 DHCP 配置网络设备，可以进行以下设置。

首先，选择相应的设备，然后单击右侧的“编辑”按钮，此时会打开该设备的“网络设备”窗口。在该窗口中选中“Manual Configuration”复选框，然后手动配置该设备的 IP 地址和子网掩码，如图 1-25 所示。

单击“确定”按钮，显示如图 1-26 所示的窗口，用户可以在这里设置主机名称、网关、DNS 服务器等内容。

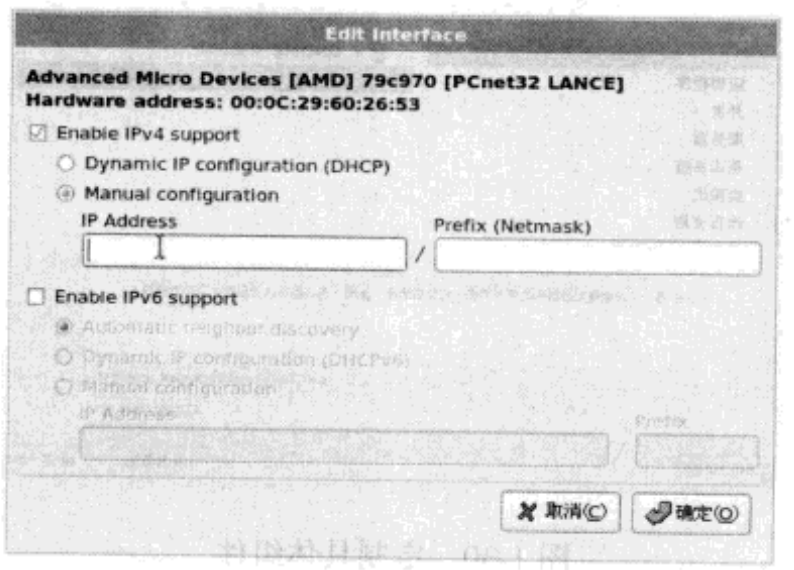


图 1-25　手工设置 IP 地址

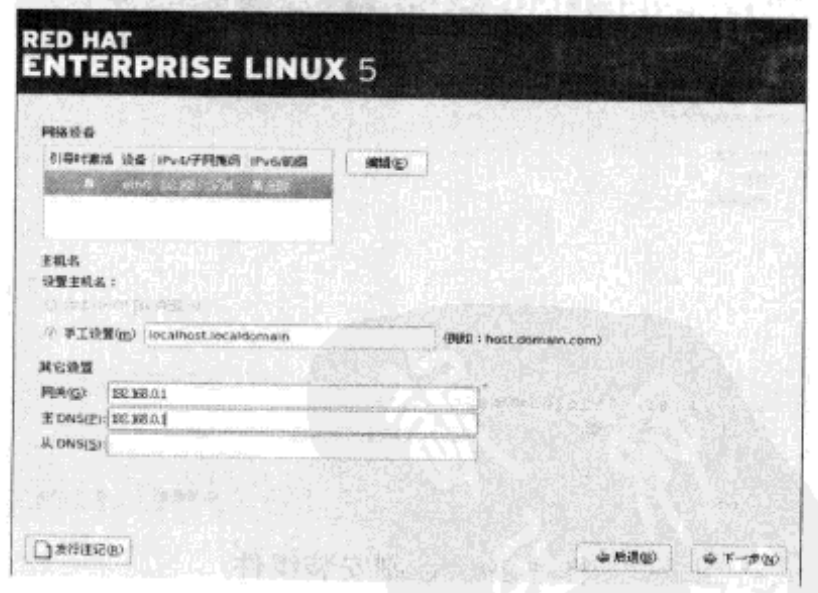


图 1-26　设置网关和 DNS 服务器

第 8 步，进入设置系统时区。

Red Hat Enterprise Linux 5 可以通过两种方式来设置时钟：一种是在下方的“时区选择”下拉列表中选择对应时区；另一种是在设置时区窗口中，通过上方的交互式地图来定位所在时

区，用户单击该区域后，画面上会显示一个红色的“x”，如图 1-27 所示。

第 9 步，设置系统时区完成后，接下来将设置 root（相当于 Windows 中的 Administration 账号）用户口令，如图 1-28 所示。

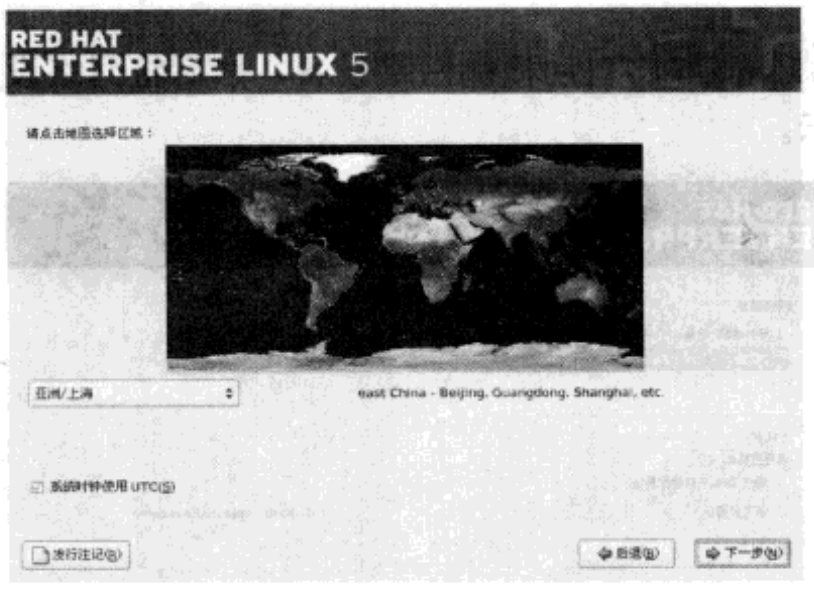


图 1-27　设置系统时区界面

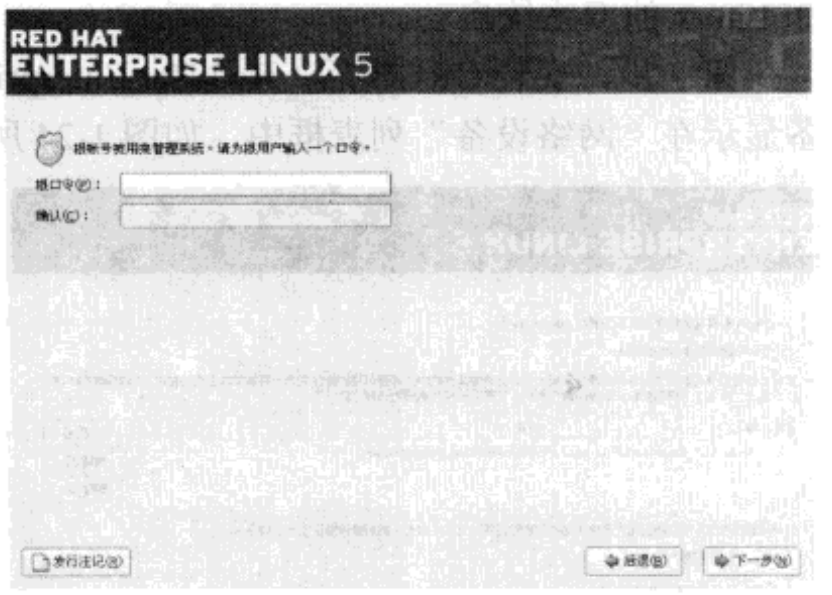


图 1-28　设置 root 用户口令

也可以在安装系统后重新设置 root 账户口令，如果用户需要在安装 Red Hat Enterprise Linux 5 之后重新设置 root 账户口令，可以通过“system-config-rootpassword”指令来完成。

第 10 步，进入定制组件阶段。

Red Hat Enterprise Linux 5 为用户提供了比较丰富的安装组件，这些安装组件随用户安装序列号的不同而不同。一般情况下，Red Hat Enterprise Linux 5 的组件默认显示的只有软件开发、虚拟化和网络服务器 3 部分。如果用户需要进行更详细的定制，则需要进行以下步骤。

首先，在该窗口中选择下方的“现在定制”单选按钮，如图 1-29 所示。

然后，用户便可以自定义多种安装组件（比如 FTP 组建、KDE 桌面等），如图 1-30 所示。

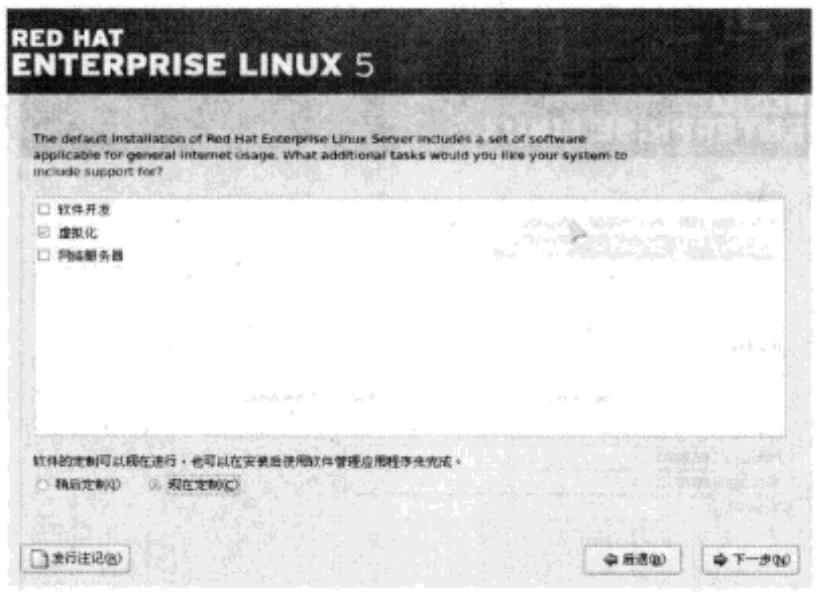


图 1-29　定制安装组件

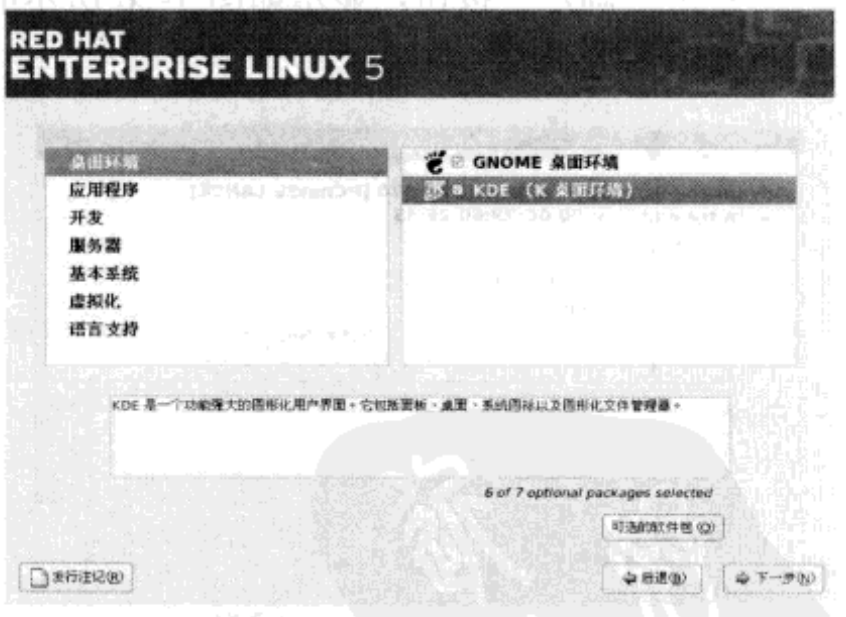


图 1-30　定制具体组件

单击“下一步”按钮，此时系统开始进行相关文件的复制，如图 1-31 所示。Red Hat Enterprise Linux 5 安装进程会详细地显示安装需要的时间、目前正在安装的组件以及组件的简单说明。

安装完成后的，显示如图 1-32 所示的界面，然后将光盘取出，并单击“重新引导”按钮，重新启动安装的系统。

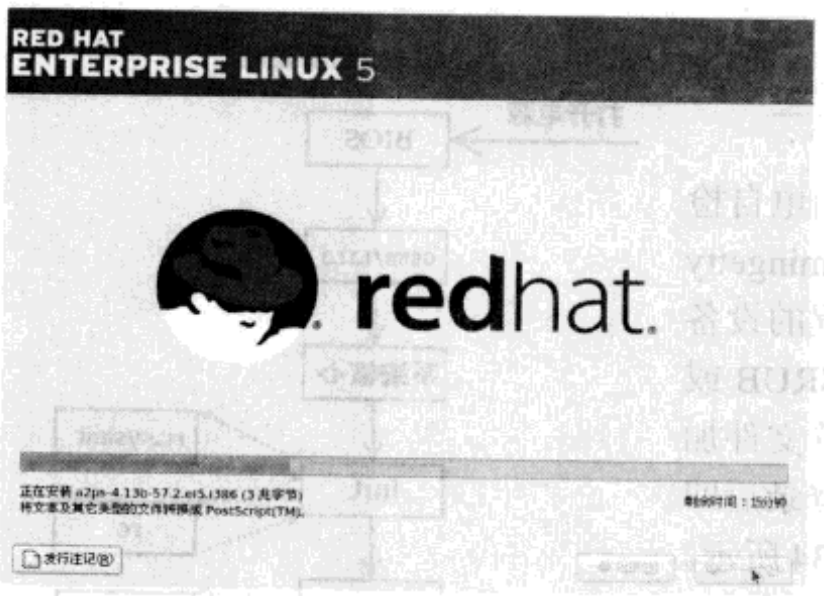


图 1-31　正在安装

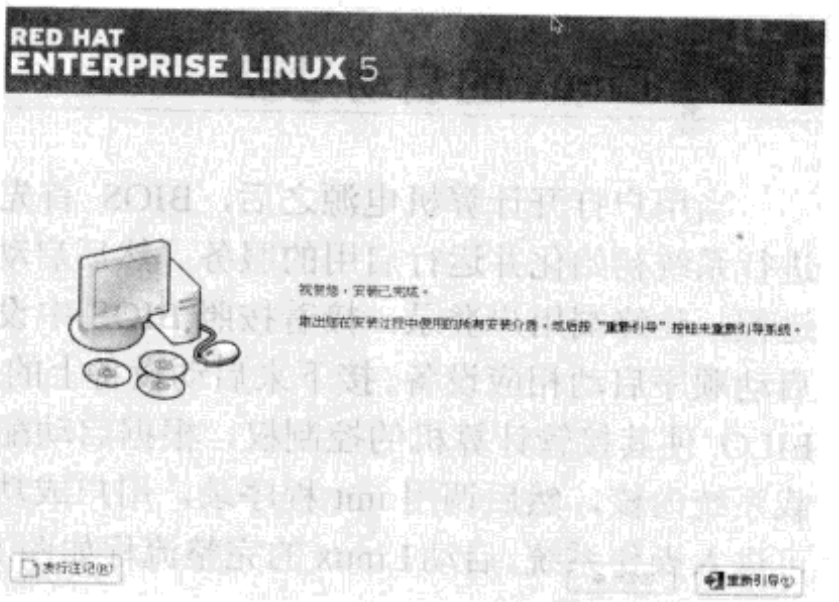


图 1-32　安装完成后的界面

1.6　Linux 操作系统的引导与登录

Linux 操作系统下多系统的引导是通过 GRUB 和 LILO 来实现的。本节将具体介绍如何在 Linux 下通过 GRUB 和 LILO 来引导多系统。

1.6.1　多系统引导管理器——GRUB 及 LILO

GRUB（Grand Unified Bootloader）与 LILO（Linux Loader）都是多系统引导管理器，都可以用来进行多系统引导，其中最常用到的是 GRUB。

GRUB 不仅支持众多操作系统，还具有一个特殊的交互式控制接口，在系统启动时，用户就可以手动确定操作系统所在的位置。因此，即使用户添加了新硬盘或者更改了分区，GRUB 总能找到正确的位置来引导系统，如图 1-33 所示。

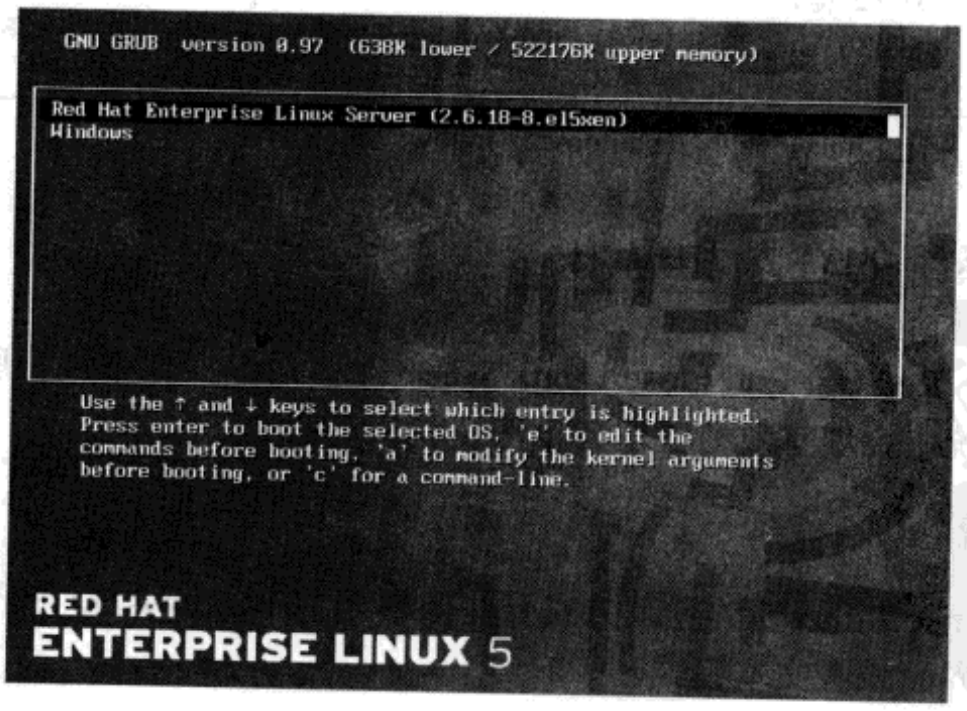


图 1-33　GRUB 引导系统界面

1.6.2 Linux 的启动过程

当用户打开计算机电源之后，BIOS 首先加电自检进行系统初始化并运行启用的服务。然后启动 mingetty 进程，让终端用户登录，接着按照 BIOS 中设置的设备启动顺序启动相应设备。接下来启动设备上的 GRUB 或 LILO 使其接管计算机的控制权，根据启动配置文件加载系统内核，然后调用 init 程序录，用户成功登录后即可进入操作系统。启动 Linux 的完整流程如图 1-34 所示。

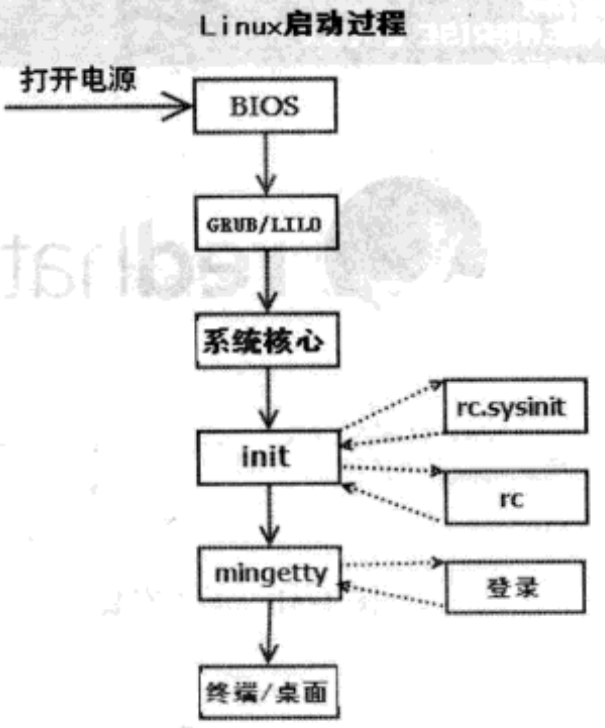


图 1-34 启动 Linux 的完整流程

1.6.3 Linux 的关闭与重新引导

Red Hat Enterprise Linux 系统的关闭可以通过 shutdown 命令来实现，shutdown 命令说明如表 1-3 所示。

表 1-3 shutdown 命令说明表

shutdown 命令	简 要 说 明
-k	不是真正关闭系统，只是警告
-h	关闭后暂停
-r	关闭后重新引导
-c	取消已经运行的关闭操作
-n	不通过 init 直接关闭
-f	快速重新引导
time	关闭的时间（必需）。可以使用下列任一方法指定 time 选项： 格式为 hh:mm 的绝对时间，其中 hh 表示小时（1 位或 2 位数字），mm 表示小时内的分钟（2 位数字）。 格式为 +m 的等待分钟数，其中 m 表示分钟数，now 表示立即关闭，等效于使用 +0 等待 0 分钟

注：一般情况下关闭 Linux，输入“shutdown –h now”命令即可。

1.6.4 Linux 系统的登录和退出

正如前文所说，Red Hat Enterprise Linux 5 是一个多用户操作系统，所以了解用户如何正确地登录和退出系统，合理有效地管理各个用户就很有必要了。

1. 登录系统

Red Hat Enterprise Linux 5 的登录方式有些像早期的 Windows 3.x 的登录方式，用户可以根据需要启动图形界面或者文本模式。

（1） 图形界面登录

对于绝大多数初级用户来说，一般都建议登录 Linux 的图形界面，所以使用默认模式，在用户输入账号和密码后即可登录系统，如图 1-35 所示。

在登录界面的左下角，可以清晰地看到“语言”、“会话”、“重新启动”和“关机”等选项。单击“语言”选项后，用户会发现 Red Hat Enterprise Linux 5 有多种语言可供选择，并可直接切换到相应的语言界面，如图 1-36 所示。

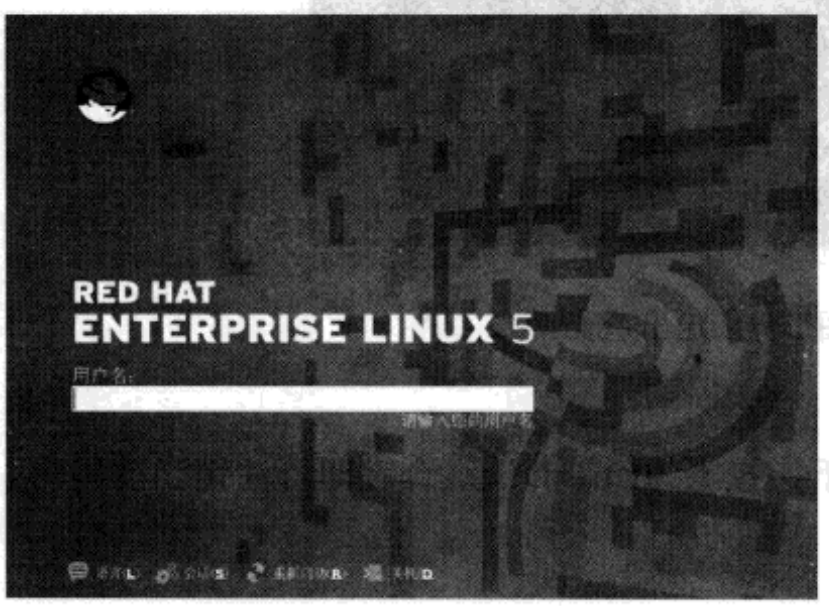


图 1-35　图形界面登录

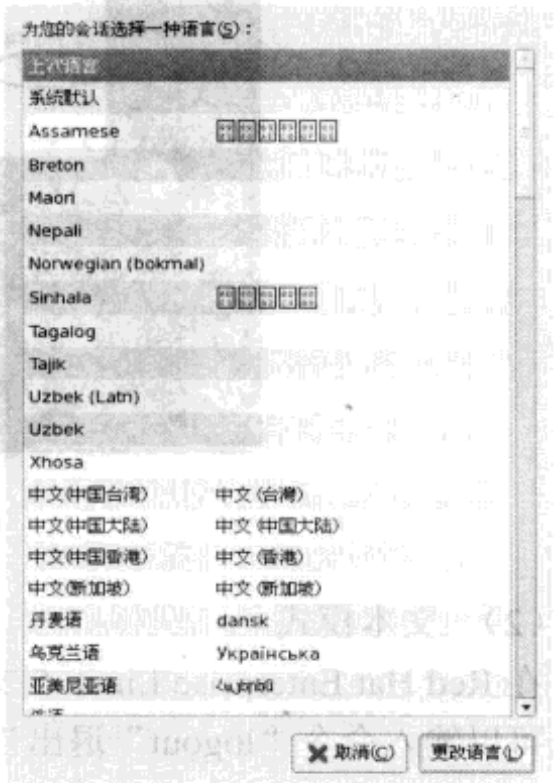


图 1-36　选择并切换语言

单击“会话”选项，弹出如图 1-37 所示的“会话”对话框。在用户安装系统时，如果选择安装 KDE，将会看到有多种界面可以选择，其中包括图形界面和文本界面。

（2）　文本模式登录

如果用户需要登录文本模式，显示 mingetty 的登录界面，可以根据如下登录提示，输入相应的账号和密码登录 Linux 系统，如图 1-38 所示。

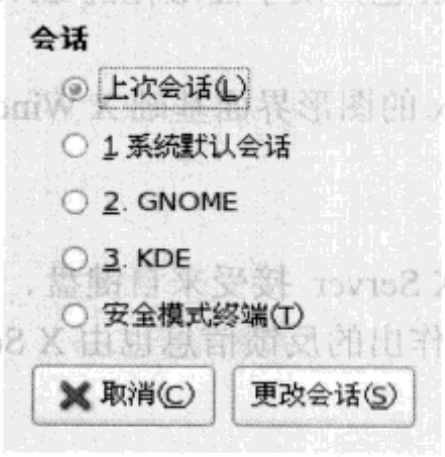


图 1-37　“会话”对话框

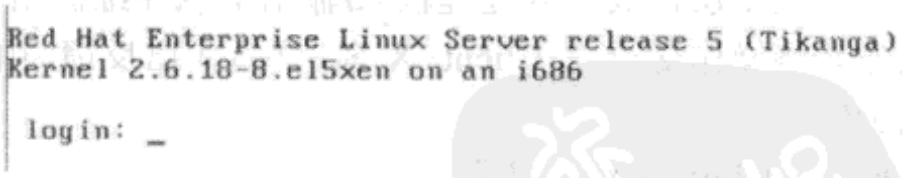


图 1-38　文本模式登录界面

2.　系统退出

图形模式系统和文本模式系统的退出方式分别如下。

（1）　图形模式

在图形桌面环境中，依次单击“系统”→“注销”菜单命令即可退出系统，如图 1-39 所示，这与 Windows 的退出操作相似。

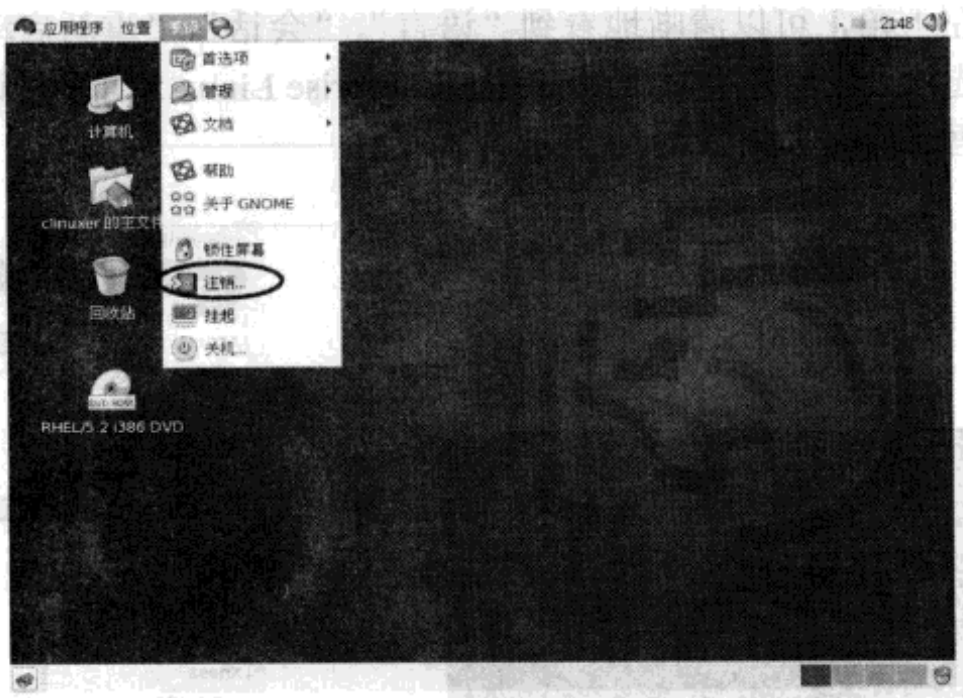


图 1-39　图形界面退出系统

（2）　文本模式

在 Red Hat Enterprise Linux 5 文本模式下，按下“Ctrl+D”组合键即可注销当前用户。用户也可以输入命令“logout”退出当前账户。

1.7　GNOME 和 KDE 桌面管理器

目前，Linux/UNIX 操作系统上较为流行的图形操作环境有 GNOME 和 KDE。两者从出现到现在，已经发展成为功能强大、性能稳定的桌面管理器。Red Hat 和 Ximian 全程支持 GNOME 项目，SuSE 也在 KDE 上做了大量工作。

对于大多数的普通用户而言，都习惯使用 Windows 操作系统。这是因为 Windows 操作系统提供了让用户操作方便的图形界面，而为了方便用户操作，Linux 也开发了图形化的 GNOME 和 KDE 桌面管理器。

在介绍 GNOME 和 KDE 之前，有必要先来介绍 UNIX/Linux 的图形界面基础 X Window。X Window 从逻辑上分为 3 层。

- X Server（X 服务器）

位于最低层，其主要处理输入/输出信息并维护相关资源，X Server 接受来自键盘、鼠标的操作并将操作交给 X Client（X 客户端）作出反馈，由 X Client 作出的反馈信息也由 X Server 来负责输出。

- X Client（X 客户端）

位于最外层，其提供了一个完整的 GUI 界面，负责与用户的直接交互（KDE、Gnome 都是一个 X Client）。

- X Protocol（X 通信协议）

用于 X Server 与 X Client 之间的衔接，充当这两者的沟通管道，如图 1-40 所示。

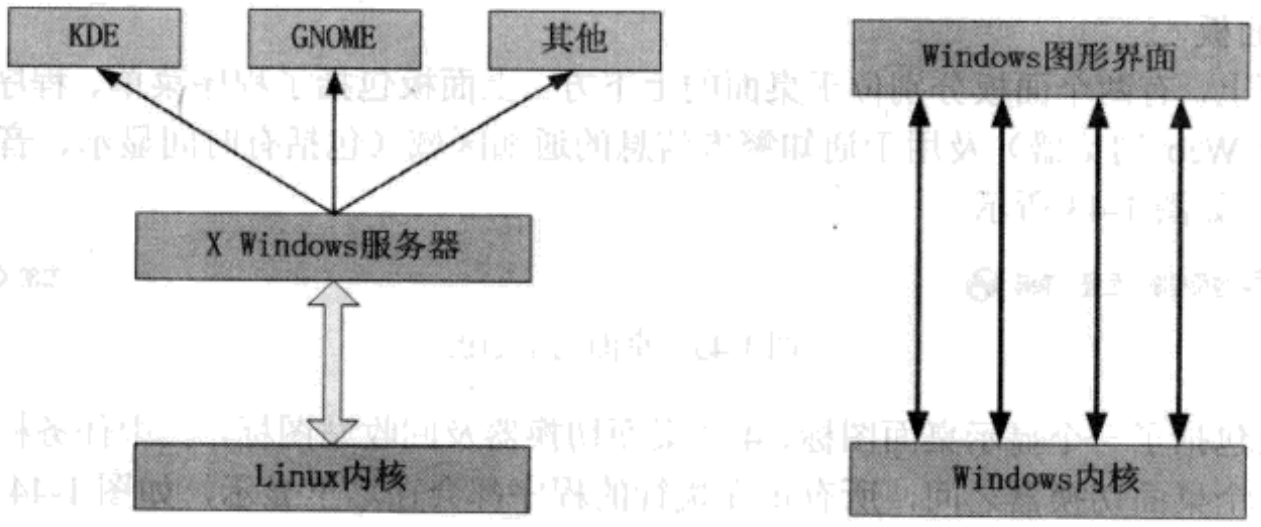


图 1-40　X Window 架构

下面将分别介绍 GNOME 和 KDE 桌面环境。

1.7.1 GNOME 桌面环境

虽然 Linux 的桌面环境与 Windows 的桌面环境有所不同，但图形化界面的操作都比较类似，如图 1-41 所示的就是 Red Hat Enterprise Linux 5 启动后的默认桌面，为典型的 GNOME 桌面环境。

1. 使用 GNOME 桌面

Red Hat Enterprise Linux 5 默认配置下的 GNOME 桌面主要包括 3 个部分：桌面快捷方式、面板图标及应用程序。

（1） 桌面快捷方式

进入桌面，在桌面上有 3 个图标，分别是计算机（相当于 Windows 操作系统下的“我的电脑”）、root 的主文件夹（相当于 Windows 操作系统下的“我的文档”，以 root 用户身份登录将出现此图标，而以其他用户登录则是“用户名的文件夹”图标）及回收站，如图 1-42 所示。

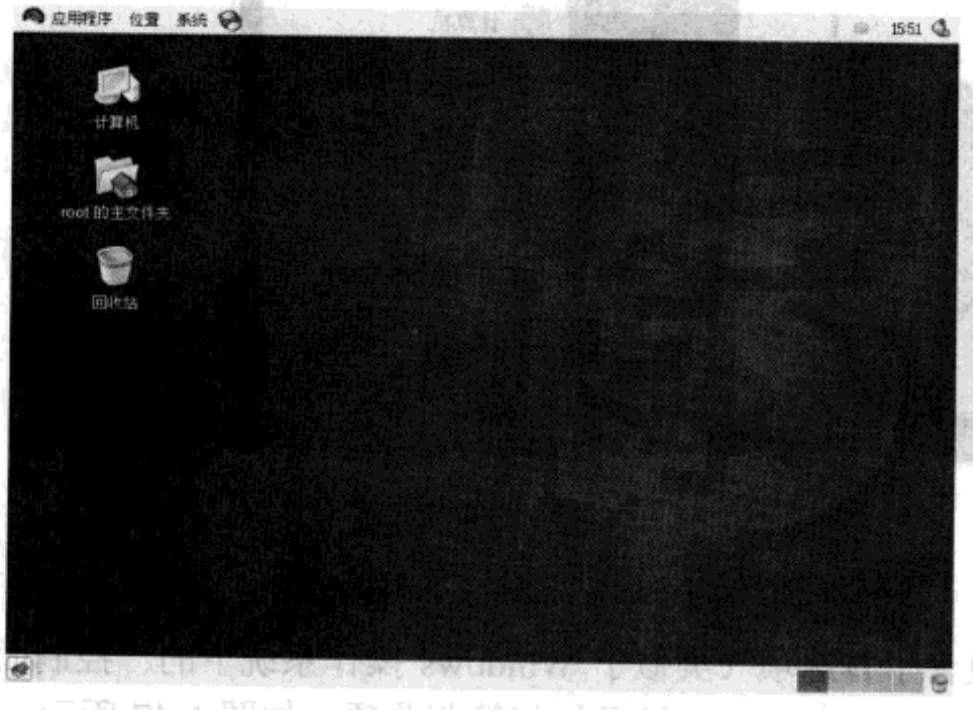


图 1-41　GNOME 桌面环境

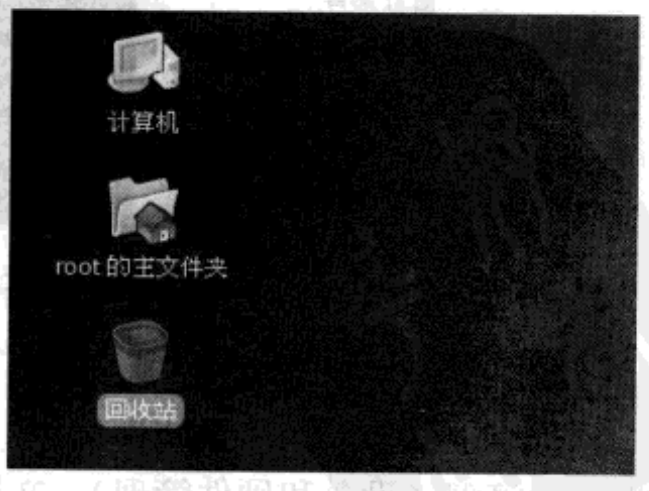


图 1-42　GNOME 桌面快捷方式

（2） 面板

在桌面上，有两个面板分别位于桌面的上下方。上面板包括了程序菜单、程序启动器（默认只有一个 Web 浏览器）及用于通知警告信息的通知区域（包括有时间显示、音量调节和键盘切换等），如图 1-43 所示。



图 1-43 桌面的上面板

下面板包括了一个显示桌面图标、4 个桌面切换器及回收站图标，其中任务栏位于显示桌面图标与 4 个桌面切换器之间，所有正在执行的程序都会在这里显示，如图 1-44 所示。



图 1-44 桌面的下面板

（3） 程序菜单

与 Windows 操作系统的“开始”菜单类似，Red Hat Enterprise Linux 5 操作系统中的很多应用程序也都可以通过程序菜单来启动。

- “应用程序”菜单

默认安装的 Red Hat Enterprise Linux 5 操作系统，其“应用程序”菜单中包括了 Internet、图像、影音、系统工具等几部分，如图 1-45 所示。

- “位置”菜单

位于上面板中的“位置”菜单，为用户提供快速访问主文件夹、桌面、计算机、其他网络服务器，以及最近的文档的方法，如图 1-46 所示。

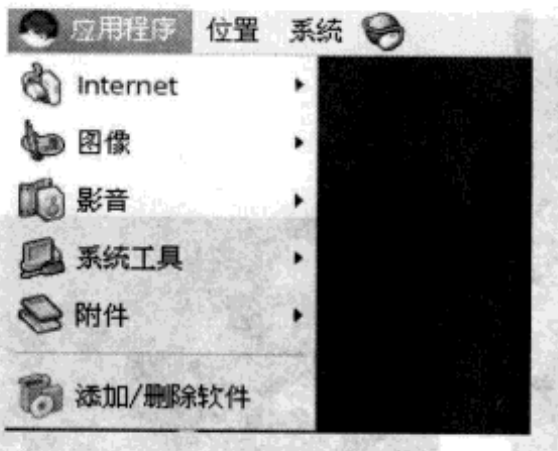


图 1-45 “应用程序”菜单



图 1-46 “位置”菜单

- “系统”菜单

位于上面板中的“系统”菜单，包含了首选项（类似于 Windows 操作系统下的“控制面板”）、管理（设备和硬件管理）、文档（帮助信息）、注销及挂起等菜单项，如图 1-47 所示。

2. Nautilus 文件管理器简介

Nautilus 图形化工具作为 GNOME 的文件管理器。与 Windows 中的资源管理器类似，Nautilus 文件管理器可以帮助用户高效地查看文件夹，管理用户文件（包括复制、剪切、删除、发送到等操作），根据文件类型使用正确的应用程序打开文件，同时，还可以显示网页、访问网络资源（如 FTP，samba，NFS）。

Nautilus 文件管理器具有两种工作模式：一种为 Spatial 模式，另一种为浏览模式。其中 Spatial 模式因其操作方便而被作为 Red Hat Enterprise Linux 5 操作系统中 Nautilus 的默认工作模式。

（1） Spatial 模式

在一般操作系统中，当用户双击文件夹时，会在原本打开的窗口中展示该文件夹中的内容。而在 Linux 操作系统中，使用 GNOME 桌面的 Nautilus 文件管理器的 Spatial 模式来管理文件夹，则是在用户每次双击时都会弹出另一新窗口，该文件夹的内容将会在此窗口中显示。不过很多初用者不习惯 Spatial 模式，其实这种模式在原理上是较为合理的，如图 1-48 所示。

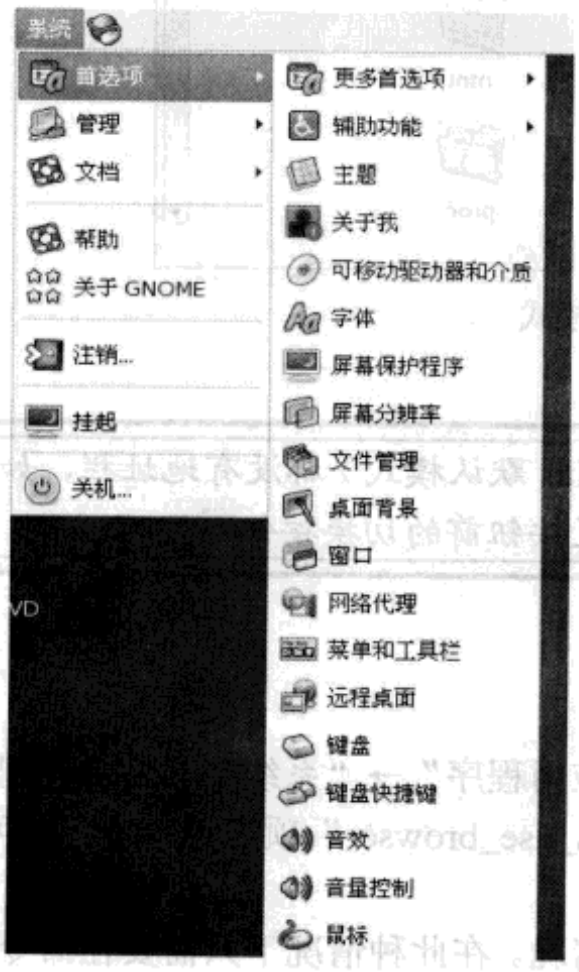


图 1-47 “系统”菜单

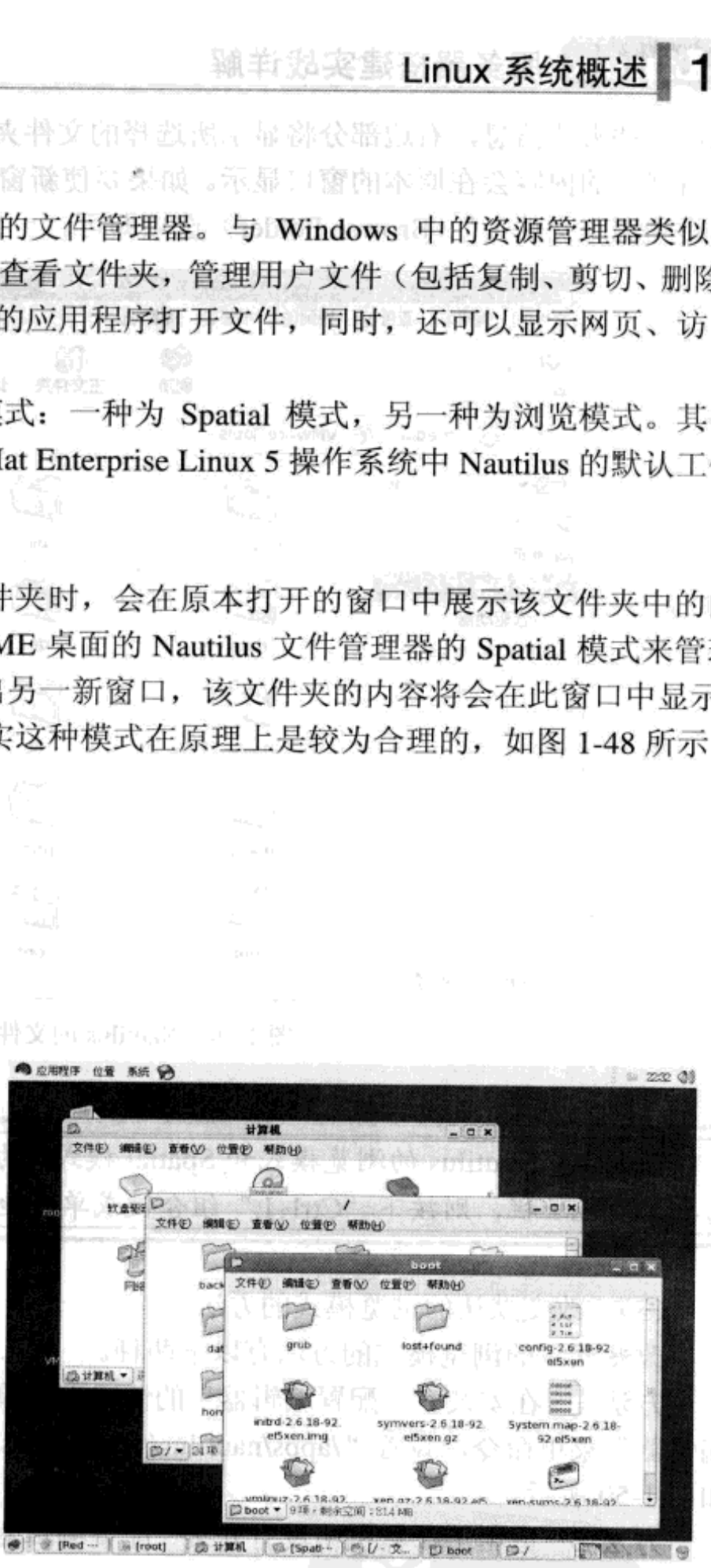


图 1-48 Spatial 工作模式

提示 如果要在 Spatial 模式中使用浏览模式打开文件，只需右击某个文件夹，选择“浏览文件夹”菜单命令即可以浏览模式打开。

（2） 浏览模式

浏览模式是 Nautilus 所支持的另一种模式。在浏览模式下，窗口分为两部分，左边部分将

显示文件夹树信息，右边部分将显示所选择的文件夹内容，如图 1-49 所示。双击文件夹时，资料夹中的内容会在原本的窗口显示。如果要使新窗口以浏览模式工作，只需右击资料夹，并选择“浏览文件夹”（Browse Folder）选项即可。



图 1-49　Nautilus 的文件浏览器模式

知识点　Nautilus 的浏览模式和 Spatial 模式（为 Nautilus 默认模式）都没有地址栏，如果需要显现地址栏，则按下“Ctrl+L”组合键或单击地址位置按钮前的切换按钮即可。

（3）　改变默认的浏览模式的方法。

改变默认的浏览模式的方法有以下两种。

方法 1　在安装有“配置编辑器”的情况下，单击“应用程序”→“系统工具”→“配置编辑器”菜单命令，设置“/apps/nautilus/preferences/always_use_browser”项为“True”即可，如图 1-50 所示。

方法 2　适用于计算机中没有安装“配置编辑器”的情况。在此种情况下只需要在命令行窗口输入以下指令进行设置。

```
[root@localhost ~]#gconftool-2 -t bool -s /apps/nautilus/preferences
/always_use_browser true
```

同样可以达到改变默认配置的目的。

（4）　如何创建文件或文件夹

在 Windows 中新建文件或文件夹是一件非常简单的事情，同样在 Nautilus 文件管理器中创建新文件也十分简单。如果需要在用户的主文件夹中创建一个新文件，可以执行如下操作。

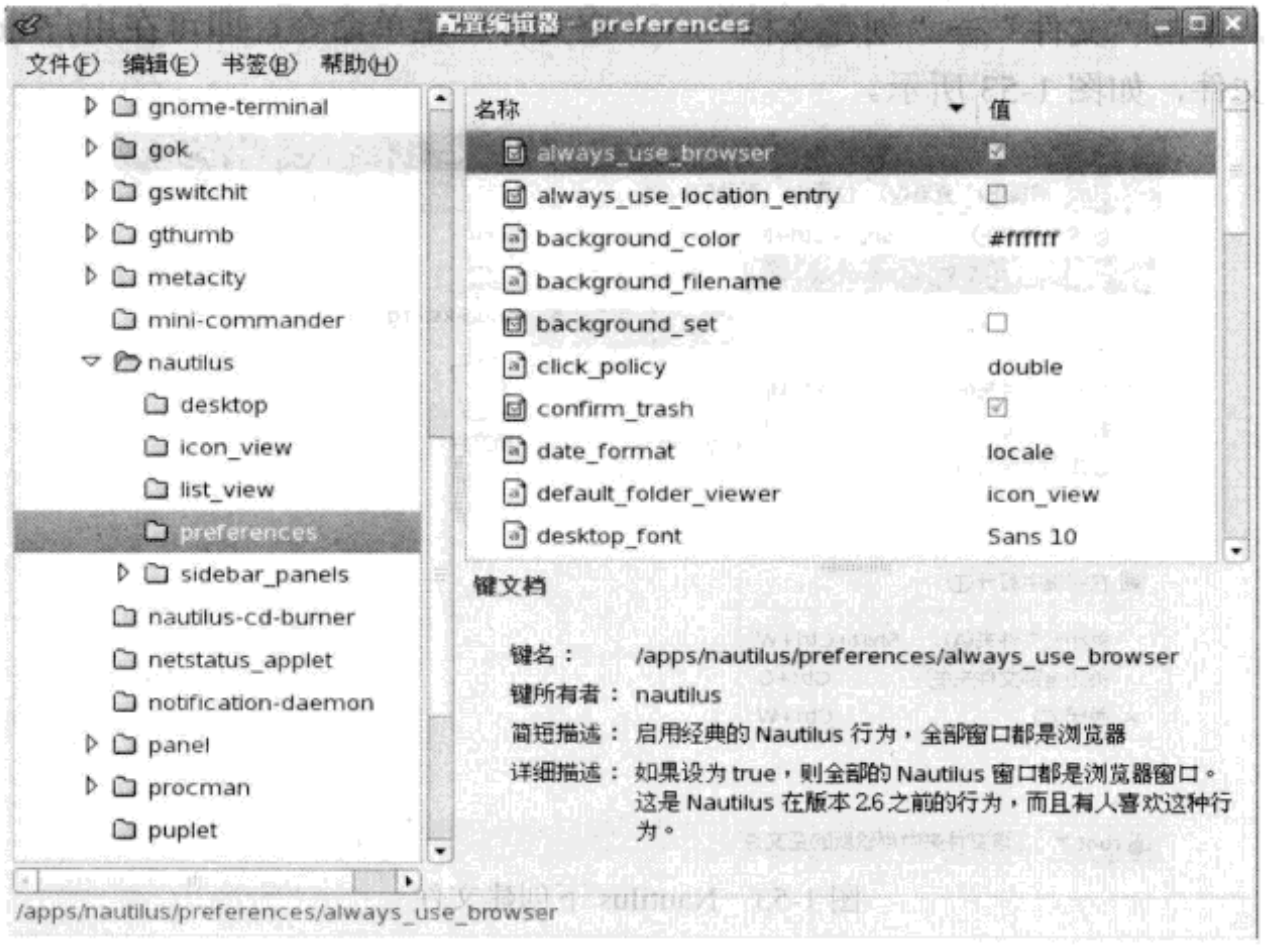


图 1-50　改变 Nautilus 的默认工作模式

第 1 步，打开 root 窗口。

打开 root 窗口的方法有两种。

方法 1　单击“位置”→“主文件夹”菜单命令，打开 Nautilus 文件管理器，如图 1-51 所示。

方法 2　在桌面上右击“root 的主文件夹”图标并选择“打开”菜单命令，如图 1-52 所示。

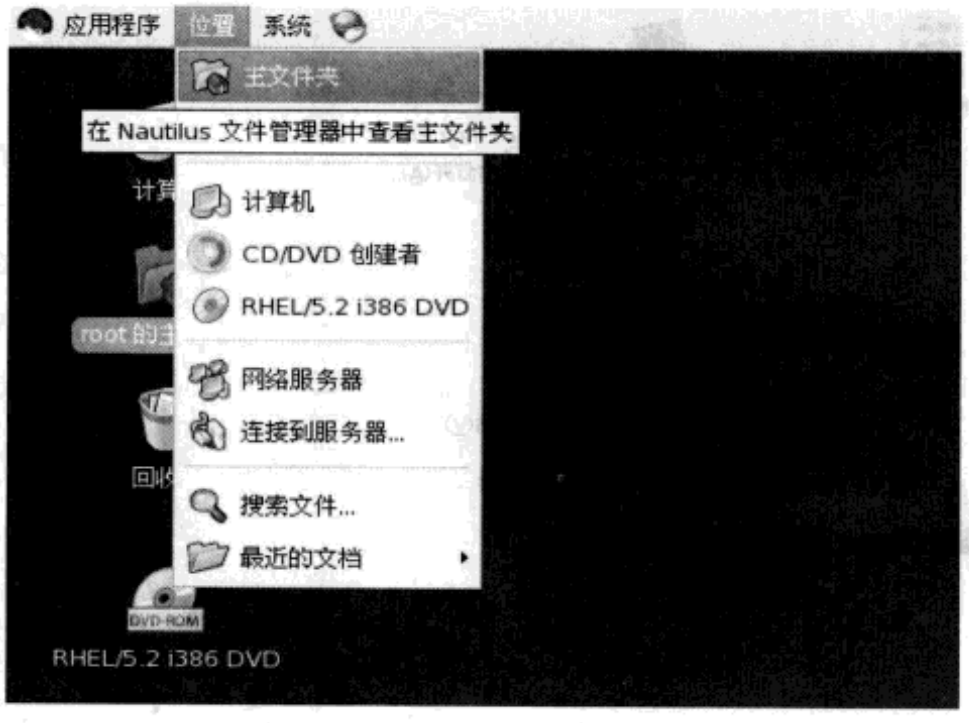


图 1-51　单击“主文件夹”菜单

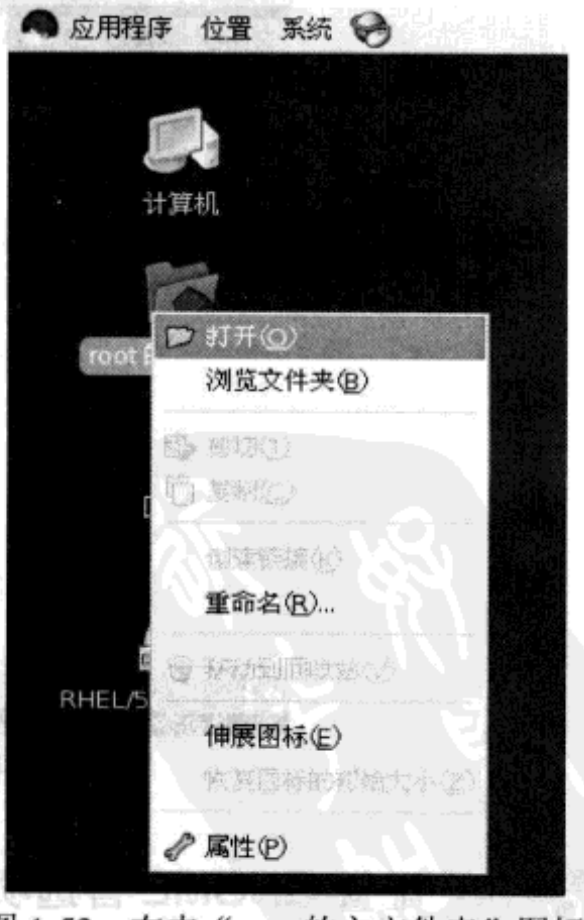


图 1-52　右击“root 的主文件夹”图标

第 2 步，单击“文件”→“创建文档”→“空文件”菜单命令，即可在用户 home 目录中创建一个空文件，如图 1-53 所示。

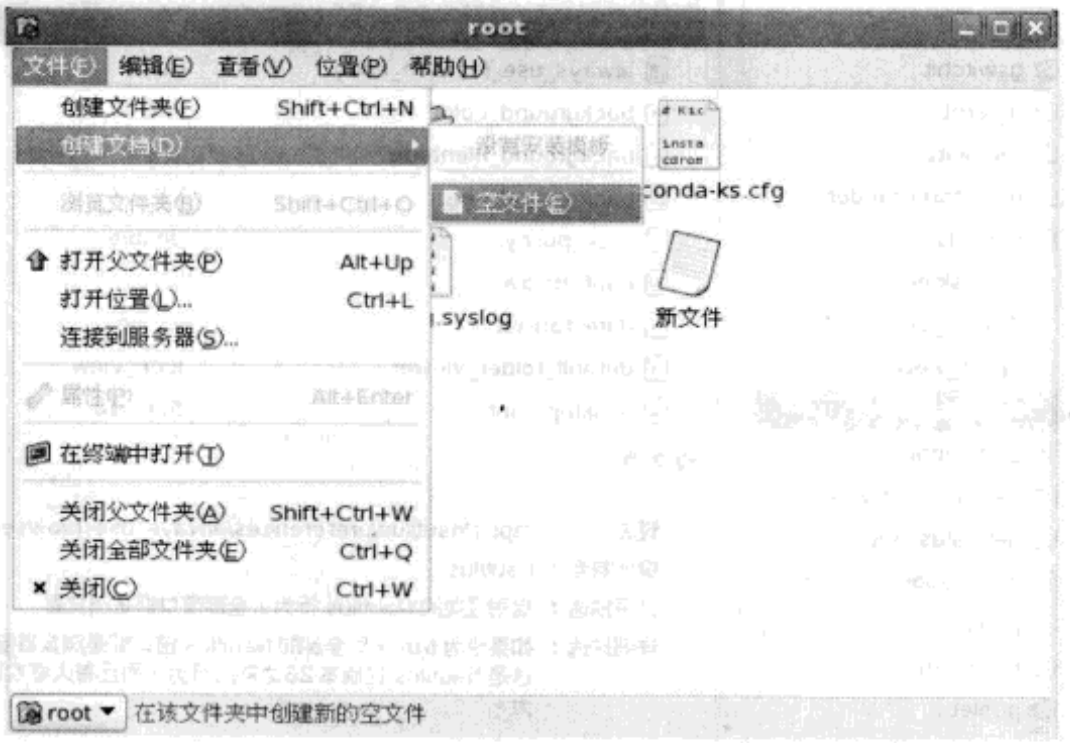


图 1-53　Nautilus 下创建文件

至此，已经完成了在 home 目录中创建新文件的操作，创建目录的操作与此类似，按照上述方法创建即可，只是创建的不是文件而是文件夹。

（5） 如何打开文件

在 Nautilus 文件管理器中，可以通过双击指定文件图标以相应的程序来打开文件，也可以选中该文件并单击鼠标右键，在弹出的快捷菜单中选择“使用其他程序打开”菜单命令，如图 1-54 所示。

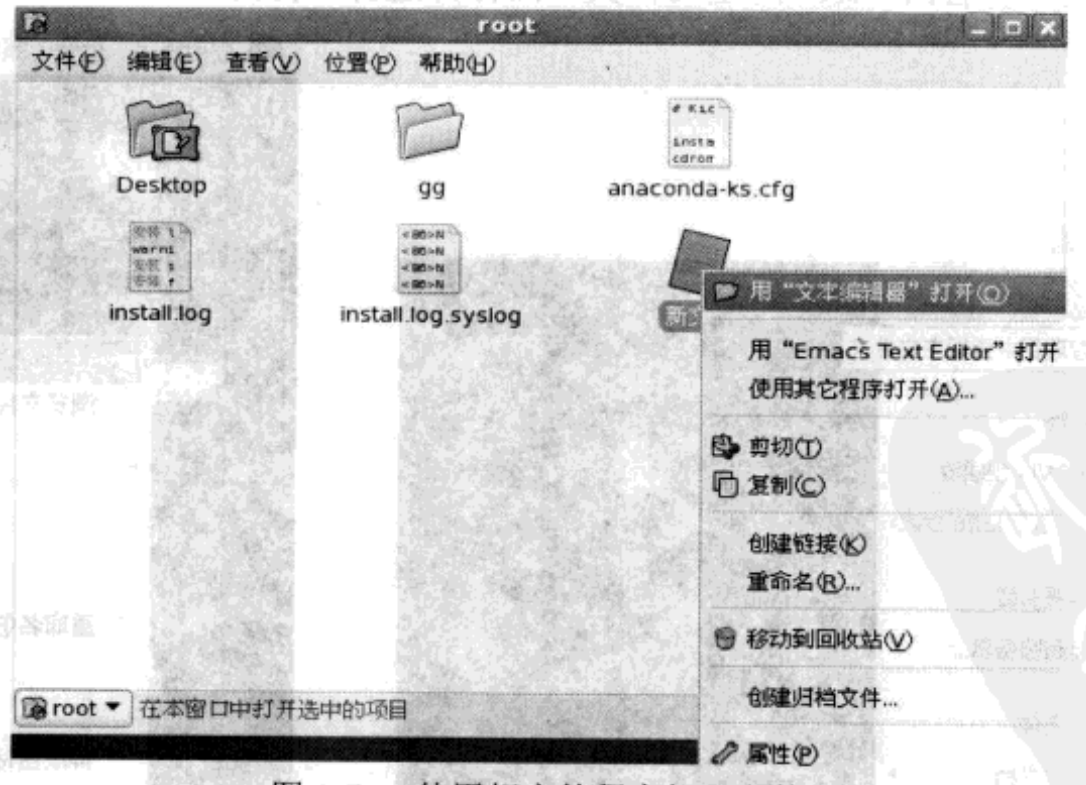


图 1-54　使用相应的程序打开文件

3. 配置 GNOME 首选项

GNOME 提供了强大的定制功能，可以根据自己的需求来定制一个符合自己操作习惯的

GNOME 桌面环境。

（1） 桌面首选项

GNOME 中的“桌面首选项”与 Windows 下的“控制面板”类似。可以单击“系统”→“桌面首选项”菜单命令，访问当中所指定的某个配置项目，如图 1-55 所示。

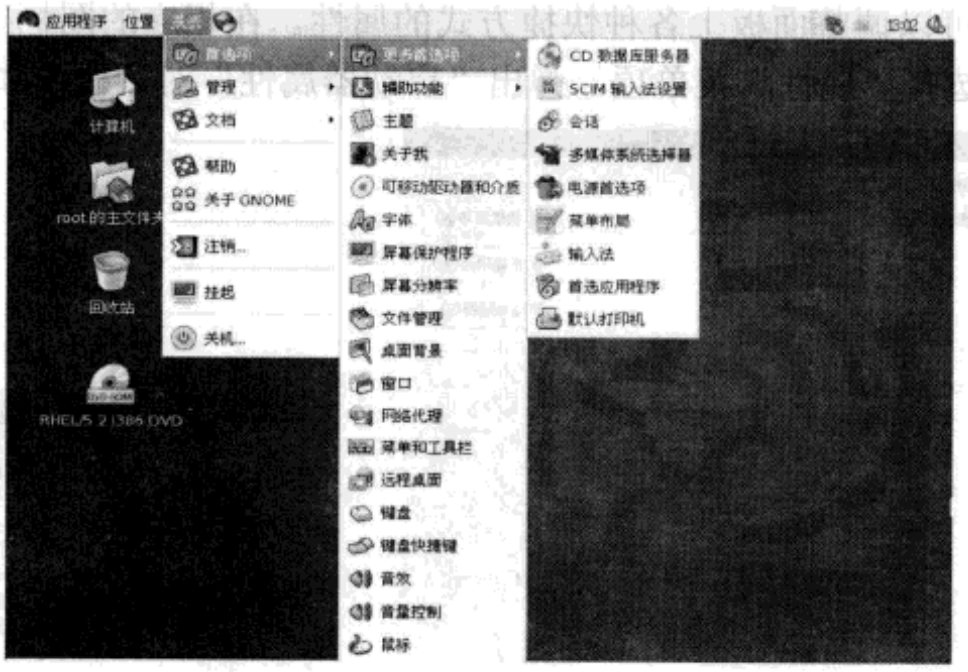


图 1-55 GNOME 中的“桌面首选项”

提示 可以通过在终端窗口输入“gnome-control-center”命令来打开“桌面首选项”。

（2） 配置面板

GNOME 有上、下两部分面板，其中上面板中有各种应用程序和其他小工具的快捷方式，用户为了方便自己的操作可以根据自己的需求来添加、删除。

- 添加应用程序和其他快捷方式

想要在面板中显示更多的应用程序和其他快捷方式，则只需在空白处单击鼠标右键，选择“添加到面板”菜单命令，显示项目添加窗口，然后根据自己对应用程序和其他快捷方式的需要添加即可，如图 1-56 所示。

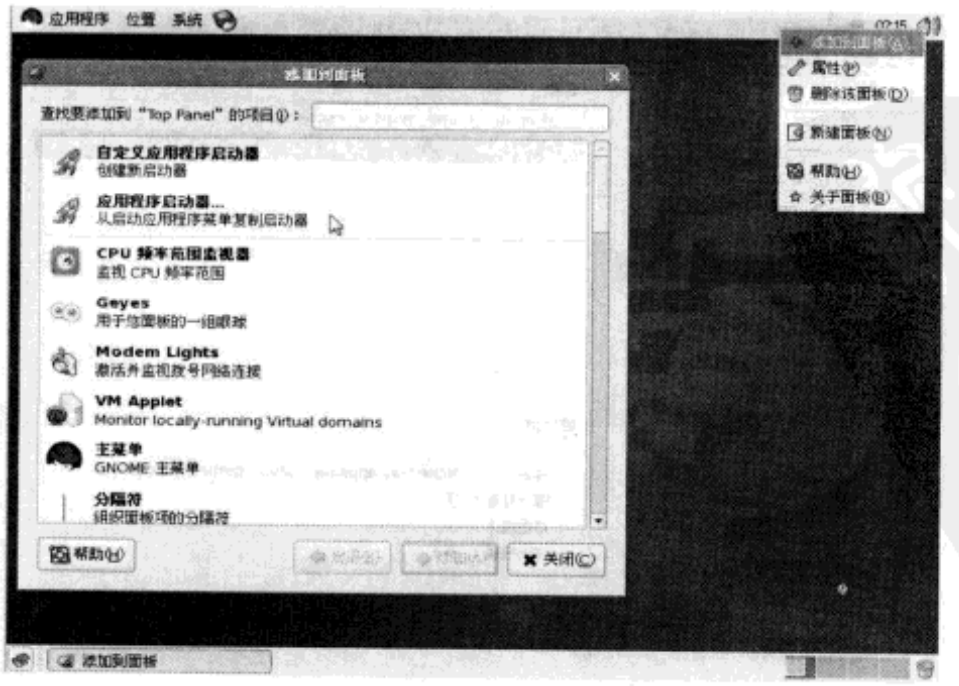


图 1-56 添加快捷方式到面板

● 调整面板上的各种快捷方式

如果对上面板所显示的各种程序菜单的菜单子项不满意，可以调整面板上各种快捷方式的属性。在相应的图标上单击鼠标右键，在弹出的快捷菜单中选择“编辑菜单”项就可以对其进行设置，如图 1-57 所示。

除此之外，还可以调整面板上各种快捷方式的属性。在相应的图标上单击鼠标右键，在弹出的快捷菜单中选择“属性”菜单项，弹出“启动器属性”对话框，如图 1-58 所示。

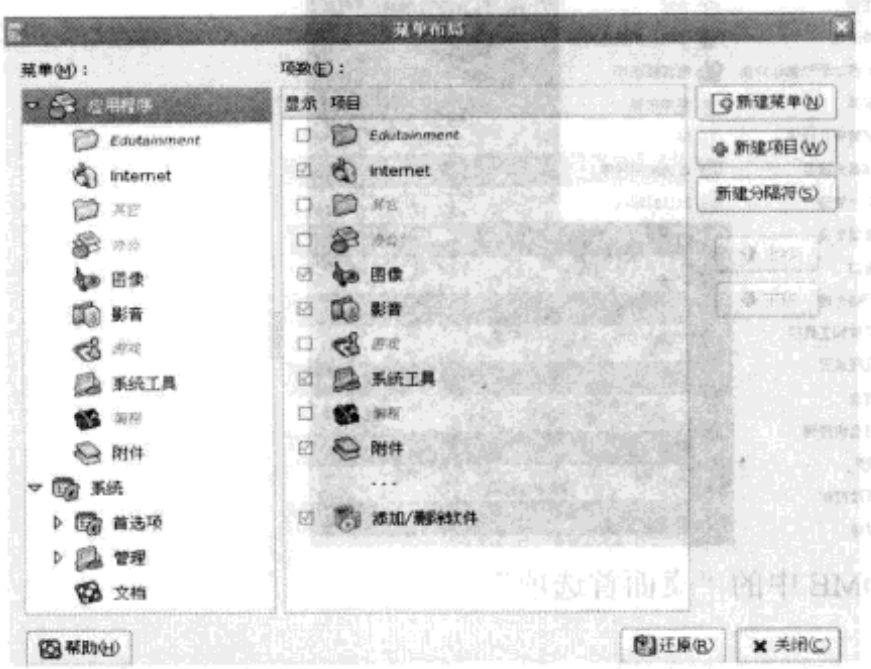


图 1-57 调整面板上的各种快捷方式



图 1-58 “启动器属性”对话框

● 删除面板上的各种快捷方式

除了可以对面板上的各种快捷方式进行添加操作外，还可以对其进行删除操作。要删除面板上多余的快捷方式，只需在相应图标上单击鼠标右键，在弹出的快捷菜单中选择“删除”项即可。

（3） 调整参数

Linux 下的 GConf-Editor 是 GNOME 2.0 桌面系统的一个应用程序参数配置系统。该系统与 Windows 中的注册表类似，可以通过该系统集中存储和管理各种应用程序的配置参数，方便管理员对系统的管理，如图 1-59 所示。

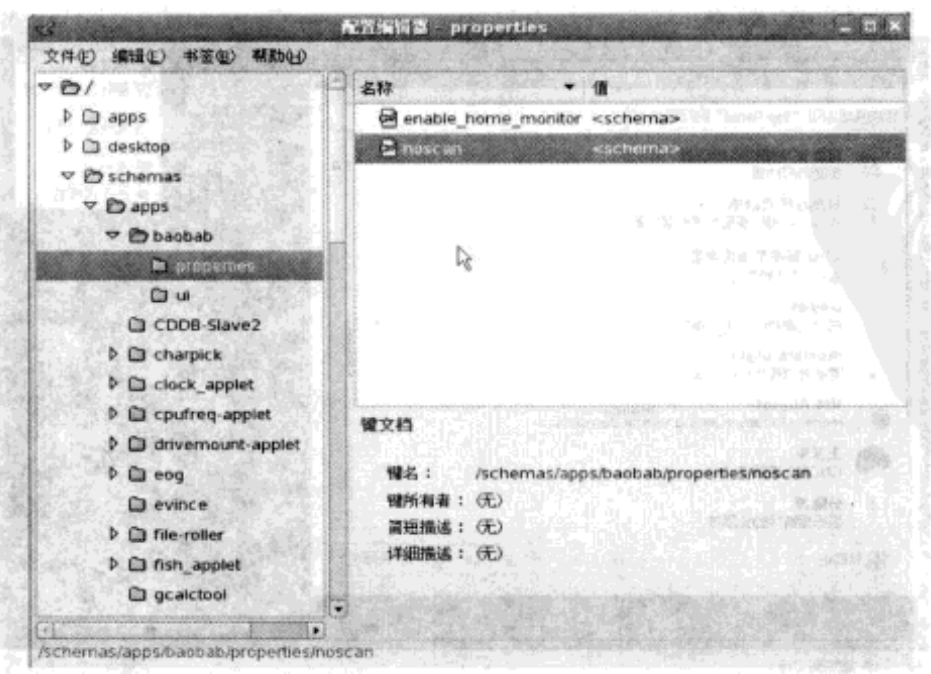


图 1-59 GConf-Editor 工具

提示　在 Red Hat Enterprise Linux 5 中默认没有安装 GConf-Editor 组件，只提供了一个命令行工具 gconftool2。如果要使用图形界面的 GConf-Editor，则需借助 Red Hat Enterprise Linux 5 安装光盘进行手动安装，执行如下命令完成安装。

```
[root@localhost Server]# rpm -ivh gconf-editor-2.16.0-3.el5.i386.rpm
warning: gconf-editor-2.16.0-3.el5.i386.rpm: Header V3 DSA signature: NOKEY,
key ID 37017186
Preparing...                ########################################### [100%]
   1:gconf-editor           ###########################################
[100%]
[root@localhost Server]#
```

4. 退出 GNOME

在 Linux 的 GNOME 中可直接执行“系统”→“注销”命令来注销当前用户。如果要彻底退出 GNOME 环境，则可以通过以下两种方式来完成。

（1） 通过改变运行级别来实现

Linux 内置了 0~6 几个运行级别（Runlevel），每个级别代表不同的功能效果，可以在 /etc/inittab 文件中对默认的运行级别进行设置，其功能级别如表 1-4 所示。

表 1-4 各个运行级别及其功能

运 行 级 别	作　　用
0	关机（不要把 initdefault 配置为 0）
1	单用户模式
2	多用户，没有 NFS
3	完全多用户模式(标准的运行级)
4	没有使用（可由用户定义）
5	X11（xwindow）
6	重新启动（不要把 initdefault 配置为 6）

（2） 通过退出 X Window 来实现

在 GNOME 环境中同时按下 Ctrl+Alt+Backspace 组合键，就可以退出 GNOME 了。但如果系统默认是以图形界面方式启动的，则该操作只达到重新启动 X Window 的效果。

1.7.2 KDE 桌面环境

除 GNOME 外，KDE 桌面环境是另外一个目前较为流行的 Linux 桌面环境。目前主流的 Linux 中，Ubuntu、Debian 等都以 KDE 作为默认的桌面管理器。这些桌面管理器不仅为用户提供了方便的操作方式，还提供了丰富的应用程序。

1. KDE 桌面的安装及使用

在 Red Hat Enterprise Linux 5 中默认采用的是 GNOME 桌面管理器，没有安装 KDE。如果要在 Red Hat Enterprise Linux 5 中使用 KDE 桌面环境，则需要对其进行安装，下面将具体讲解 KDE 桌面管理器的安装。

（1） 安装 KDE 桌面

这里推荐在安装系统时就安装 KDE，不推荐使用命令行安装，用命令行安装将会比较麻烦。因为 KDE 所包含的大量组件与其他库文件之间存在依赖关系，比较容易发生错误。

如果在安装系统安装过程中，忘记安装 KDE 桌面管理器，只需要对 Red Hat Enterprise Linux 5 进行如下的设置就可以完成 KDE 的安装。

第 1 步，在桌面选择“应用程序”→“添加/删除软件”菜单命令，显示如图 1-60 所示的“软件包管理者”窗口。

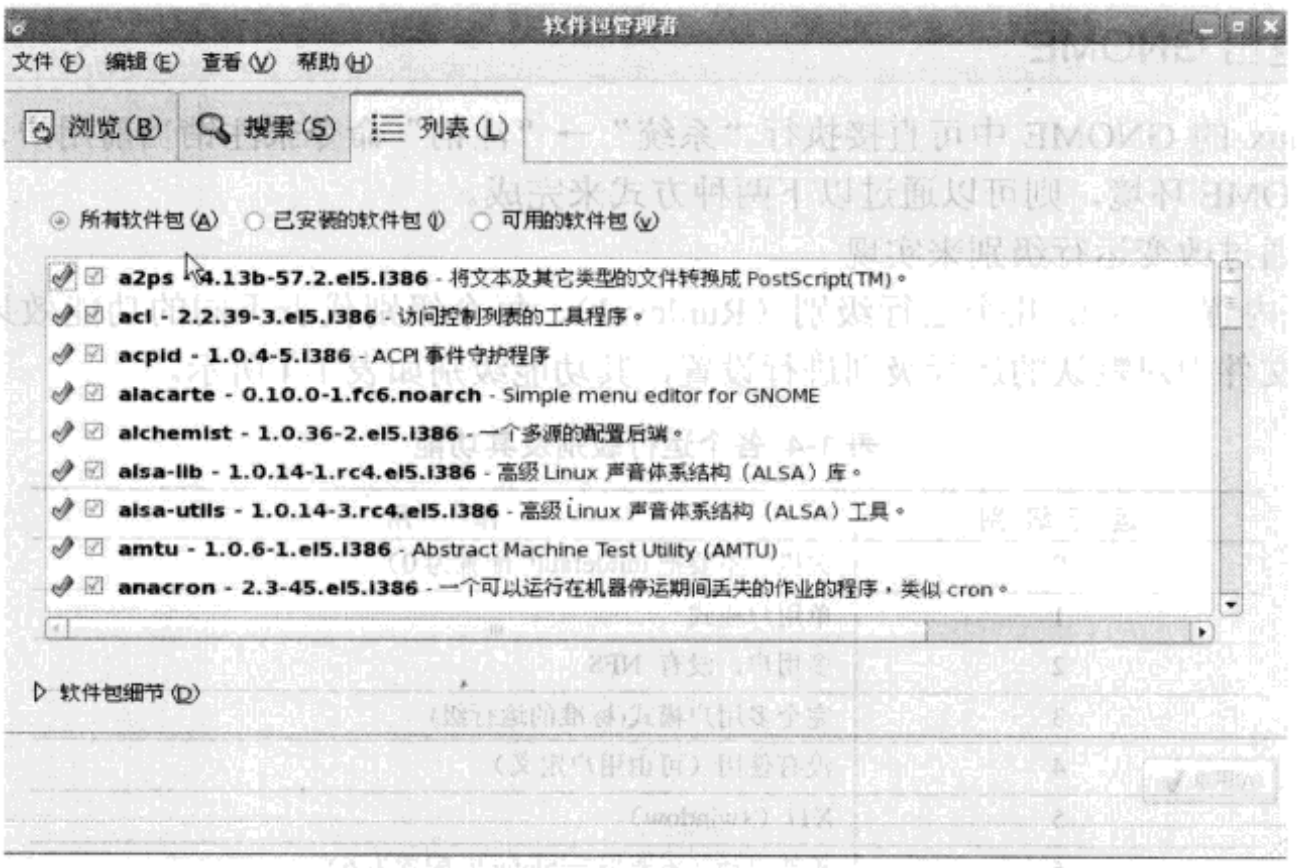


图 1-60 “软件包管理者”窗口

第 2 步，添加光盘安装源。在执行该操作前，需保证 Red Hat Enterprise Linux 5 的安装光盘已经放到了光驱中，并且已将这个光驱加载到了/mnt/cdrom。如果加载到了其他位置，请在“位置”框中根据自己的实际情况进行设置。

第 3 步，加载好光驱后，在“软件包管理者”窗口中单击“编辑”→“软件存储库”命令，显示如图 1-61 所示的“存储库管理工具”对话框。

第 4 步，在“存储库管理工具”对话框中单击“添加”按钮，显示“添加新的存储库”对话框，并按照以下方法进行配置，然后单击“确定”按钮，如图 1-62 所示。

名称：Red Hat Enterprise Linux 5（名称任意）
描述：安装Red Hat Enterprise Linux 5
位置：file:///mnt/cdrom/Server



图 1-61 “存储库管理工具”对话框

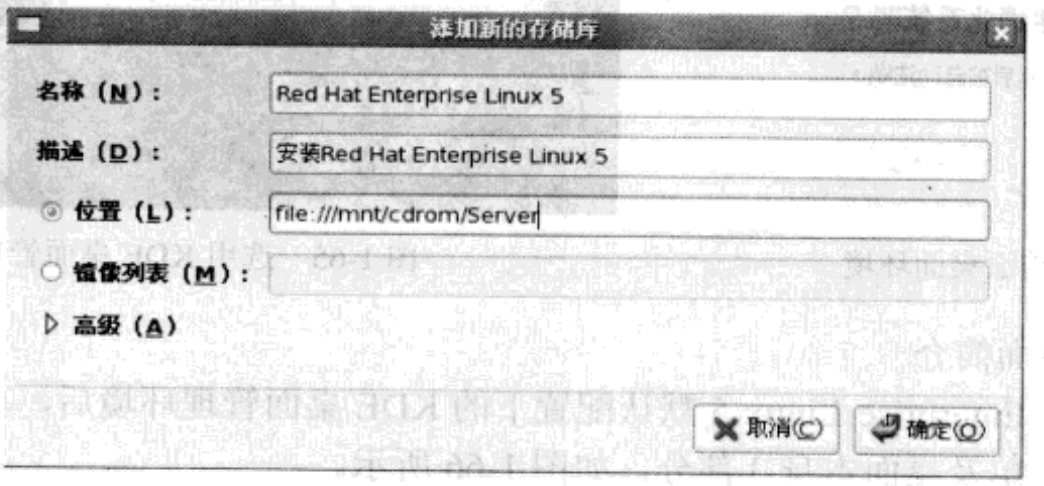


图 1-62 “添加新的存储库”对话框

第 5 步，完成配置后，重新打开“软件包管理者”窗口，切换到“浏览”选项卡，就可以看到如图 1-63 所示的 Red Hat Enterprise Linux 5 光盘上的软件包了。选择“桌面环境”项，然后选中右侧的“KDE（K 桌面环境）”复选框，最后单击“应用”按钮。这样 KDE 就会被安装到 Red Hat Enterprise Linux 5 中了。

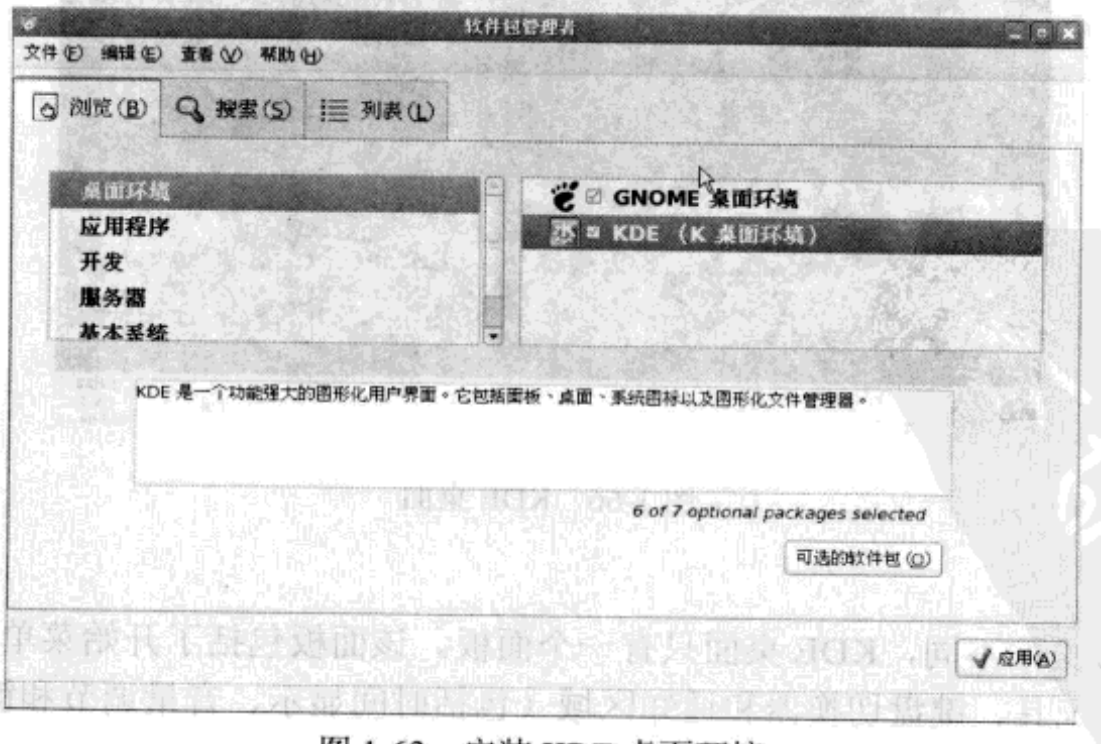


图 1-63 安装 KDE 桌面环境

（2） KDE 桌面的启用

第 1 步，KDE 桌面管理器安装成功后，在原有桌面单击“系统”→“注销”菜单按钮，将显示询问是否注销的对话框，单击“注销”按钮，如图 1-64 所示。执行后返回到登录窗口。

第 2 步，在登录窗口中单击“会话”项，显示“会话”对话框，再选择“KDE”单选按钮，最后单击“更改会话”按钮，如图 1-65 所示。再次登录系统，将会出现 KDE 桌面。

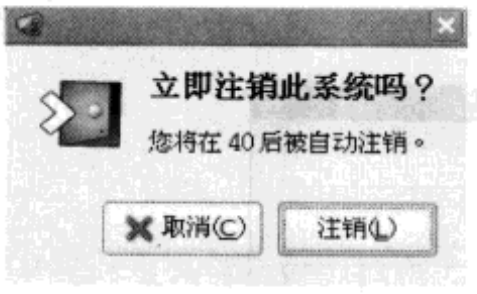


图 1-64　注销原桌面环境

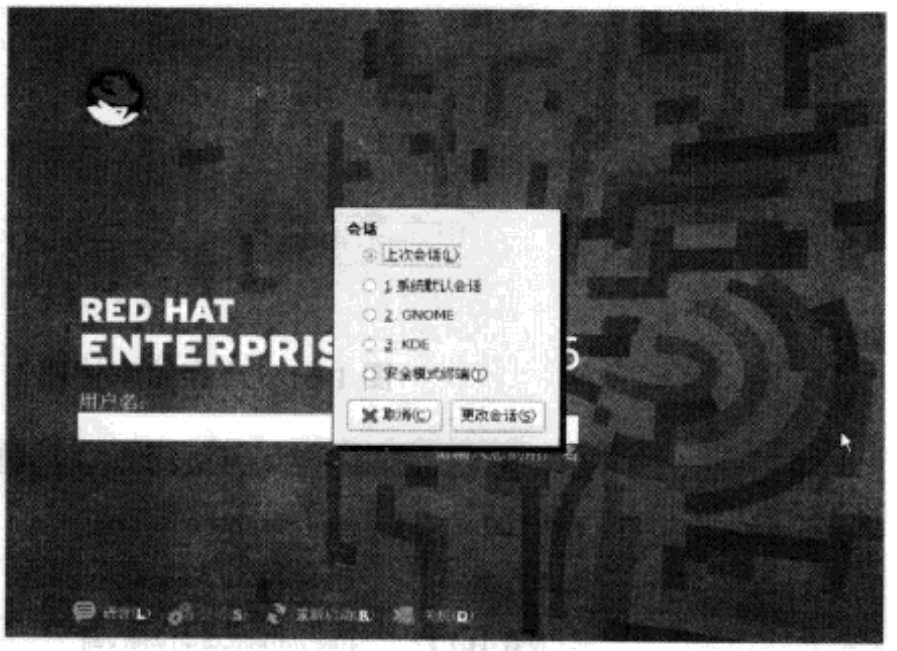


图 1-65　选用 KDE 桌面管理器

（3） KDE 桌面简介

进入 Red Hat Enterprise Linux 5 默认配置下的 KDE 桌面管理环境后，可以看到桌面上有面板图标、开始菜单及桌面本身 3 部分，如图 1-66 所示。

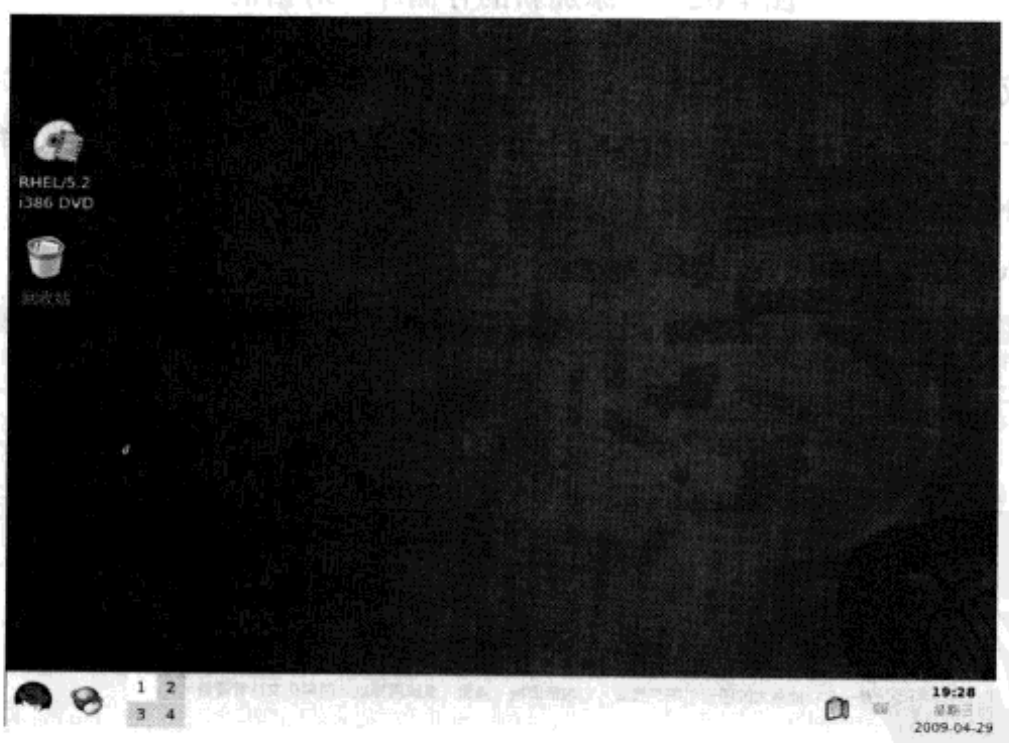


图 1-66　KDE 桌面

- 面板

与 GNOME 桌面不同，KDE 桌面只有一个面板。该面板包括了开始菜单、程序启动器、任务栏、剪贴板工具、键盘切换器和通知区域（包括时间显示、音量调节和键盘切换等），如图 1-67 所示。



图 1-67　KDE 面板

● 开始菜单

在 Red Hat Enterprise Linux 5 的 KDE 桌面上单击左下方的“K Menu”图标即可显示开始菜单，如图 1-68 所示。

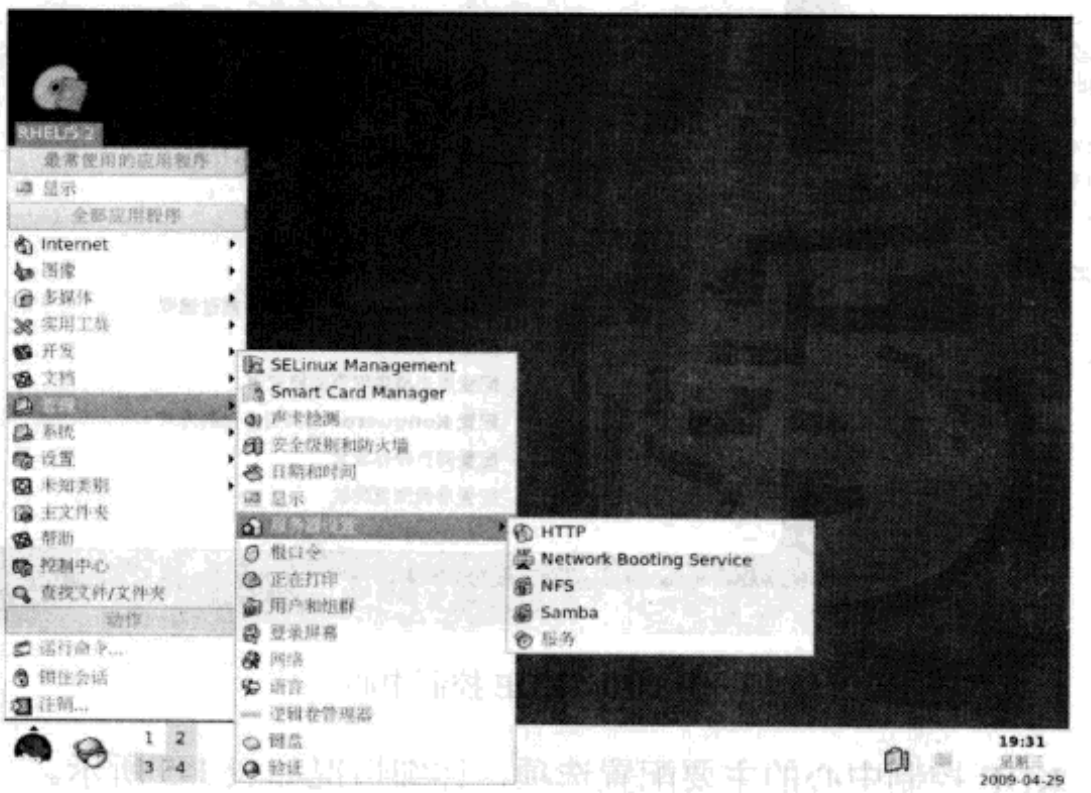


图 1-68　K Menu

由此图可以看出，该桌面管理器的内置应用程序与 GNOME 相比增加了很多。可以观察到，该菜单包括“Administration（系统管理）”、“Documentation（文档）”、“Internet（互联网）”、“Multimedia（多媒体）”、“Settings（设置）”、“System（系统工具）”、“Utilities（工具）”、“Control Center（控制中心）”和“Find File/Folders（搜索）”等多个部分。

● KDE 中的桌面切换器

与 GNOME 类似，KDE 也有桌面切换器，只是表现形式有所差异，功能相同，用户可以在 4 个桌面之间相互切换，如图 1-69 所示。



图 1-69　KDE 中的桌面切换器

2. 配置 KDE 桌面

KDE 为调整桌面和系统的各种参数提供了丰富的选项，下面将具体讲解如何进入 KDE 控制中心以及进入后如何配置。

单击“K 菜单”→“控制中心”菜单命令即可显示如图 1-70 所示的 KDE 控制中心。

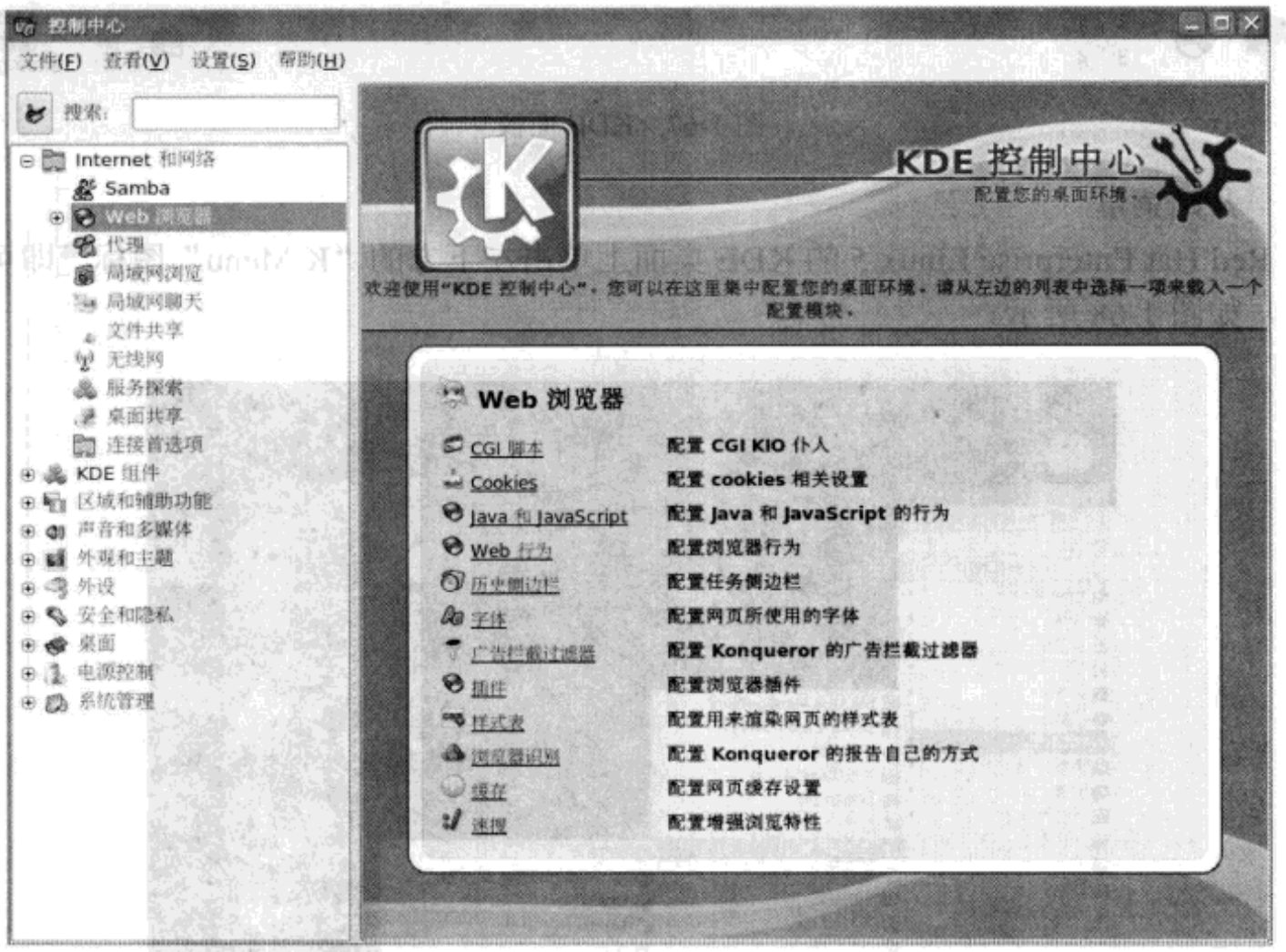


图 1-70　KDE 控制中心

下面将介绍 KDE 控制中心的主要配置选项，详细情况如表 1-5 所示。

表 1-5　KDE 控制中心的主要配置选项

主要配置选项	简 要 说 明
“Internet 和网络”项	在该选项中，可以设置 Konqueror 浏览器、代理服务器、无线网络、以及 Samba 服务器，可以和 Windows 系统进行直接连接，如图 1-71 所示
“KDE 组件”项	在该项中，不仅可以配置 Konqueror 文件管理器，还可以设置文件关联，方便所有文件都可以在 Konqueror 中打开，同时也可以调整 KDE 的性能
“区域和辅助功能”项	该项主要用于设置国家和语言选项，同时可对视觉和听觉有缺陷的用户提供配置辅助功能（如可听和可视的提示，以及鼠标/键盘定制等）
“外观和主题”项	该项提供定制背景图像，或配置字体、主题、图标、面板元素、屏幕保护程序以及窗框装饰等桌面环境视觉效果功能
“外设”项	该项主要用于设置 KDE 图形桌面的分辨率，设置存储介质、键盘、鼠标、打印机、游戏手柄等，另外，还可以设置类似于 Windows 的远程桌面的远程控制
“桌面”项	主要用于定义任务条、面板、虚拟桌面、行为等，如图 1-72 所示
“电源控制”项	主要用于设置笔记本电脑的节电管理
“系统管理”项	可以安装字体，可以设置日期和时间，切换登录管理器等

3. Konqueror 文件管理器

Konqueror 是 KDE 中默认的文件管理器和 Internet 浏览器，主要用于文件管理、多媒体播放、图像浏览和网页浏览等。

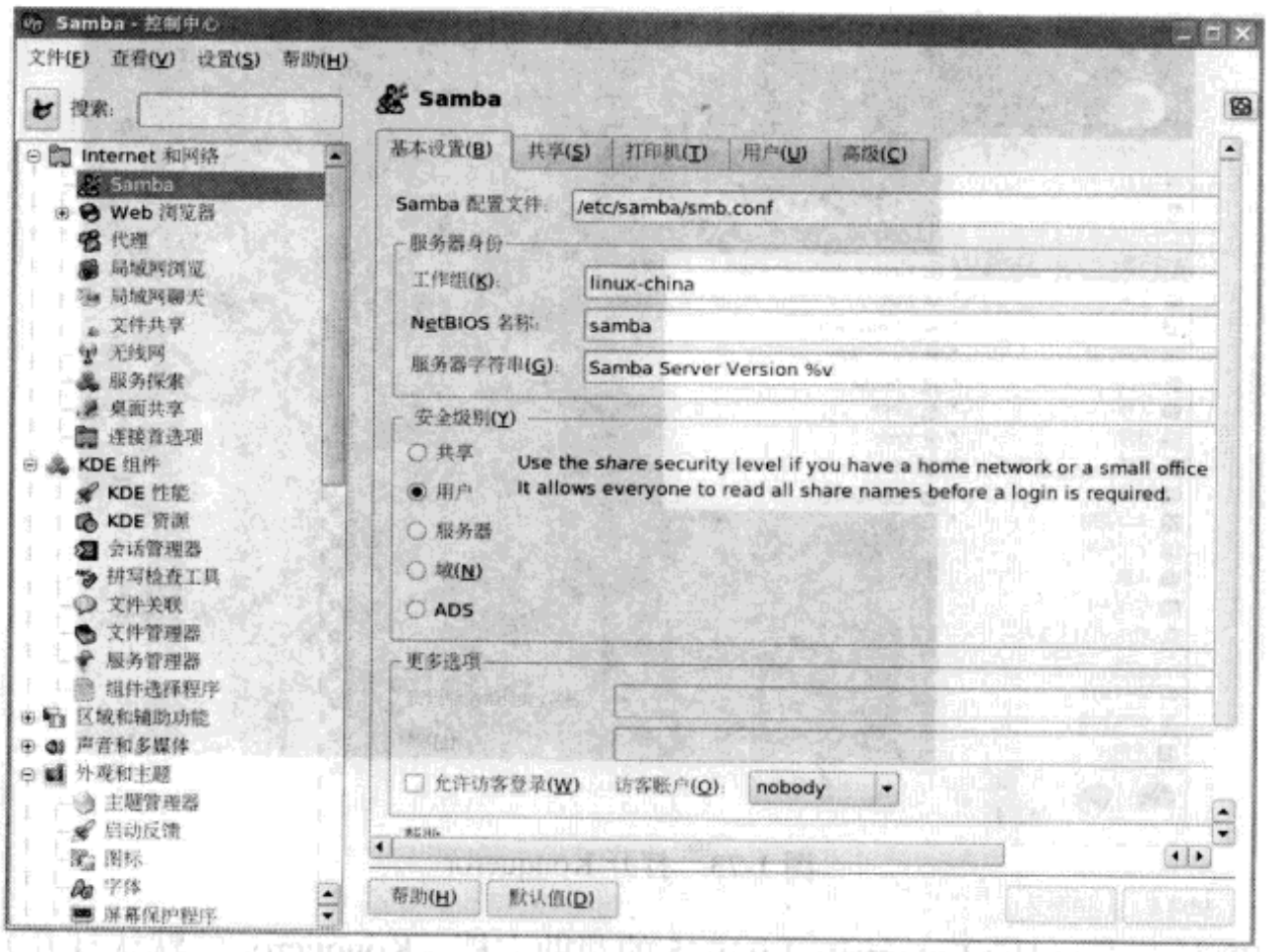


图 1-71 “Internet 和网络”项

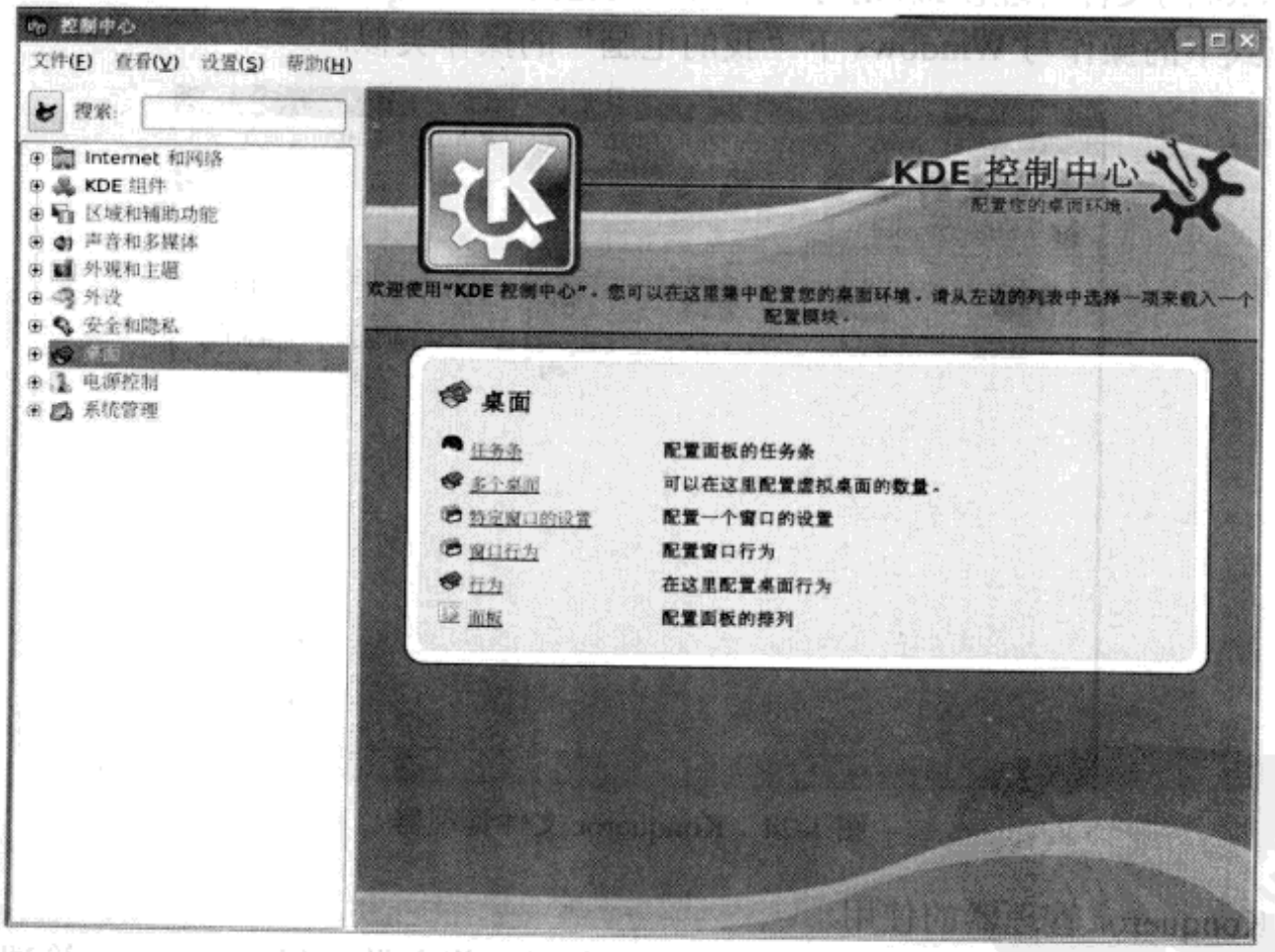


图 1-72 “桌面”项

（1） 打开 Konqueror 管理器

打开 Konqueror 管理器，然后选择“K 菜单”→“Internet”→“Konqueror”菜单命令，如图 1-73 所示。

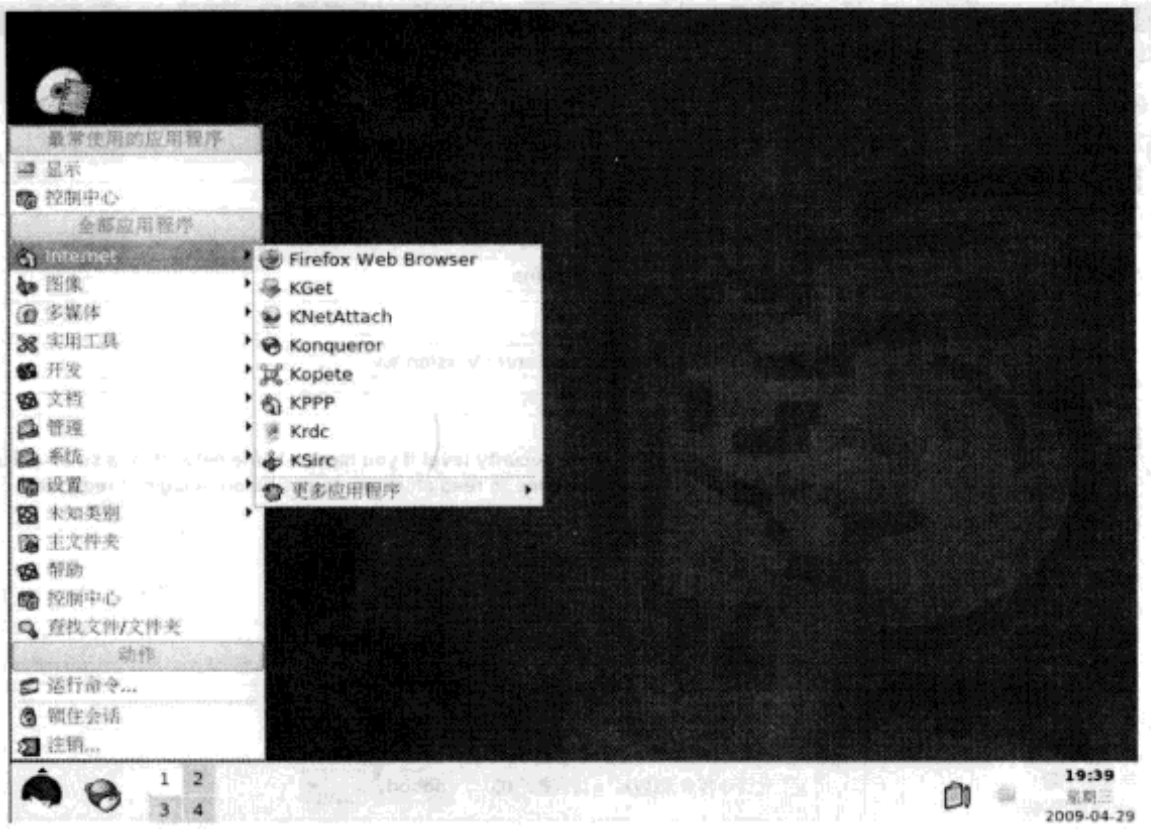


图 1-73　打开 Konqueror

此时将会打开 Red Hat Enterprise Linux 5 的帮助文档（Konqueror 的默认打开），如果需要打开 Konqueror 的文件管理模式，则可以单击工具栏的 Home 按钮，如图 1-74 所示。Konqueror 文件管理模式下的操作与 Windows 下“我的电脑”的操作类似。

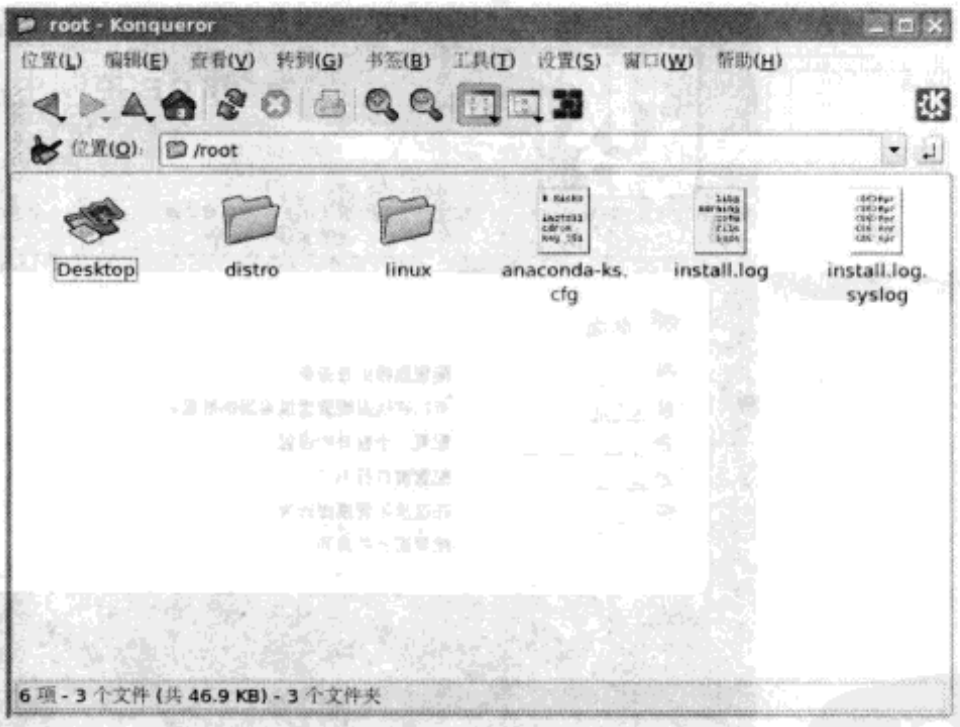


图 1-74　Konqueror 文件管理器

（2）　Konqueror 管理器的使用

在熟悉了 Konqueror 管理器的基本打开操作之后，将会学习到 Konqueror 管理器的使用。

- 使用 Konqueror 管理器打开文件

在 Konqueror 中可以通过直接单击某个程序（如单击某个音乐，可以直接收听；单击 PDF 文件，可以直接查看）来实现打开。

- 使用 Konqueror 管理器实现文件属性的查看和管理

如果要查看某文件或者文件夹的属性，直接选中该文件或者文件夹然后单击鼠标右键，

在弹出的快捷菜单中选择“属性”菜单项，即可显示相应文件的属性窗口，如图 1-75 所示。

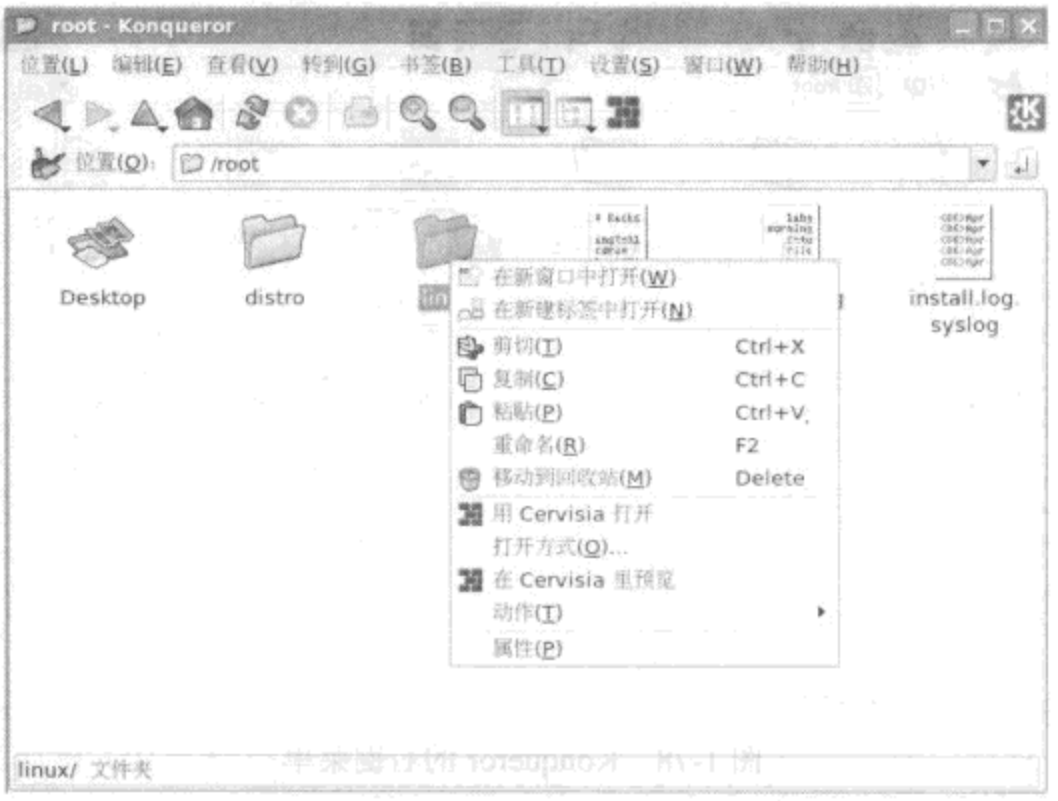


图 1-75 “属性”菜单项

弹出“Linux 的属性-Kongueror”对话框，默认显示“常规（G）”选项卡，在该选项卡下可以查看包括文件类型、位置、大小、修改日期和最后访问日期等基本属性，并可以对其进行重命名，如图 1-76 所示。

在权限（P）选项卡下，可以根据不同的用户身份设置不同的权限（可读、可读写或者可执行），并可以重设文件的拥有者及其群组，如图 1-77 所示。

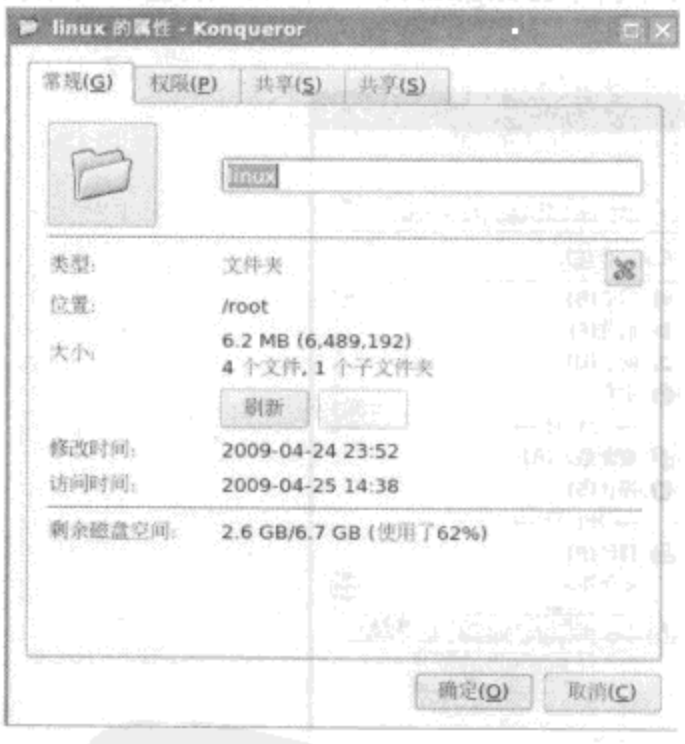


图 1-76 查看文件属性

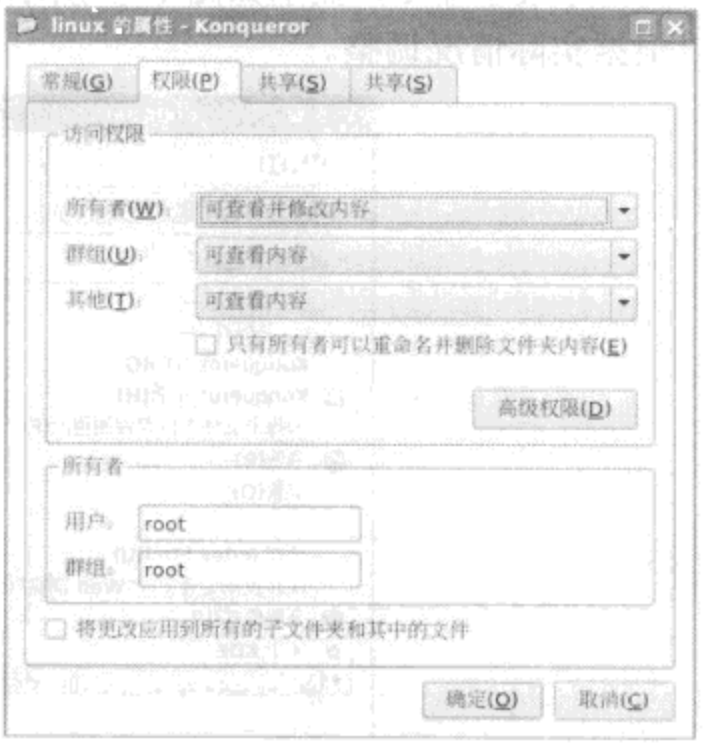


图 1-77 设置文件存取权限

（3） 使用丰富的右键菜单管理文件

Konqueror 配置了丰富的右键菜单。利用右键菜单可以选择“在新窗口中打开”、“在新建标签页中打开”选项，并可以进行复制、剪切和删除等操作，同样也可以选择适合的程序来打开相应文件，如图 1-78 所示。



图 1-78　Konqueror 的右键菜单

4. Konqueror 的配置

Konqueror 是一个良好的文件管理、文件浏览及网页浏览工具，用户可以根据自己的喜好来定义 Konqueror 的工作环境。

（1） Konqueror 工具栏的配置

如果需要把 Konqueror 默认的工具栏调整成适合自己的，则可以直接选择“设置”→“配置工具栏”菜单命令，显示如图 1-79 所示的“配置工具栏-Kongueror”窗口，该窗口中的信息可以根据实际情况调整。



图 1-79　“配置工具栏-Kongueror”窗口

（2） 视图模式的切换

Konqueror 既可以是文件管理器，也可以是网页浏览器。它不仅支持标签式浏览，甚至可以作为一个开发平台，内置了相应的配置文件，只需单击鼠标即可非常方便地切换工作模式了。

注意 选择“设置”→“装入视图配置文件”菜单命令（如图 1-80 所示），显示 KDE 开发、Midnight Commander（类似于 Total Commander 的左右分栏文件管理器）、Web 浏览、文件管理器、标签式浏览和简单浏览 6 种视图配置文件选择。

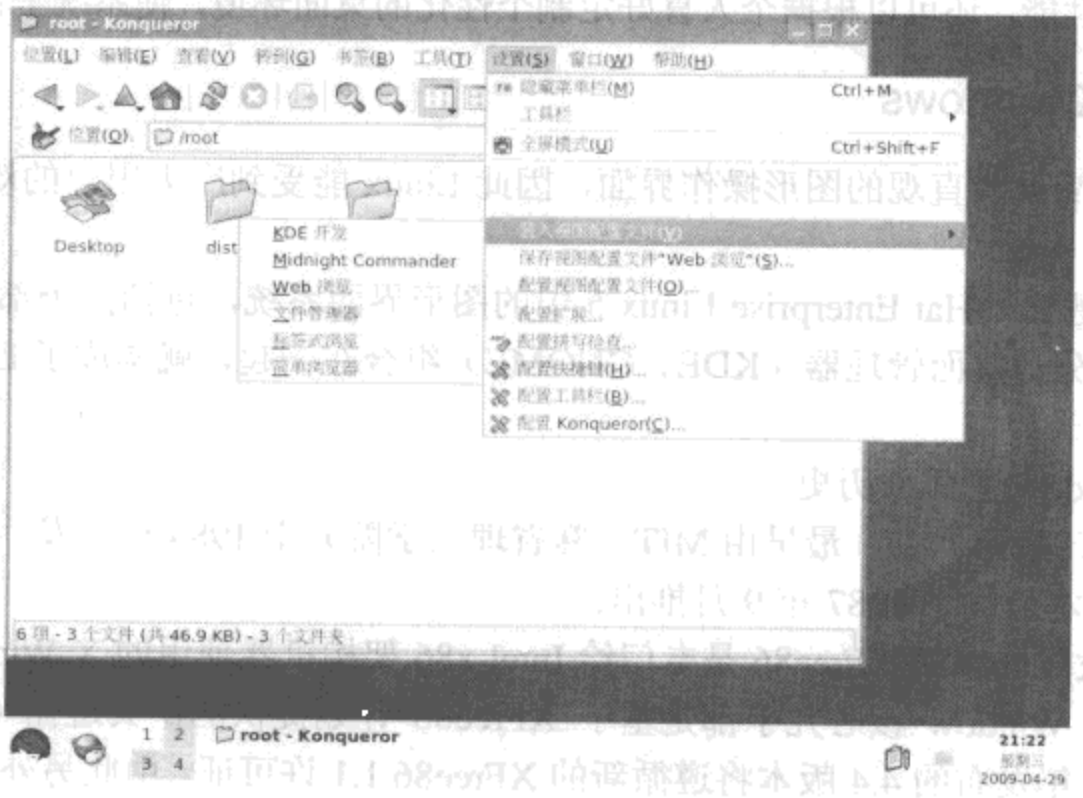


图 1-80 进入多种视图

（3） 其他配置

在 Konqueror 中，专门提供了一个配置工具，该配置工具为 Konqueror 提供了细节调整功能（包括行为、外观、文件关联、Java 和 JavaScript、广告过滤器、字体、cookie 等）。可以通过在 Konqueror 中选择“设置”→“配置 Konqueror”菜单命令，来打开如图 1-81 所示的配置工具。

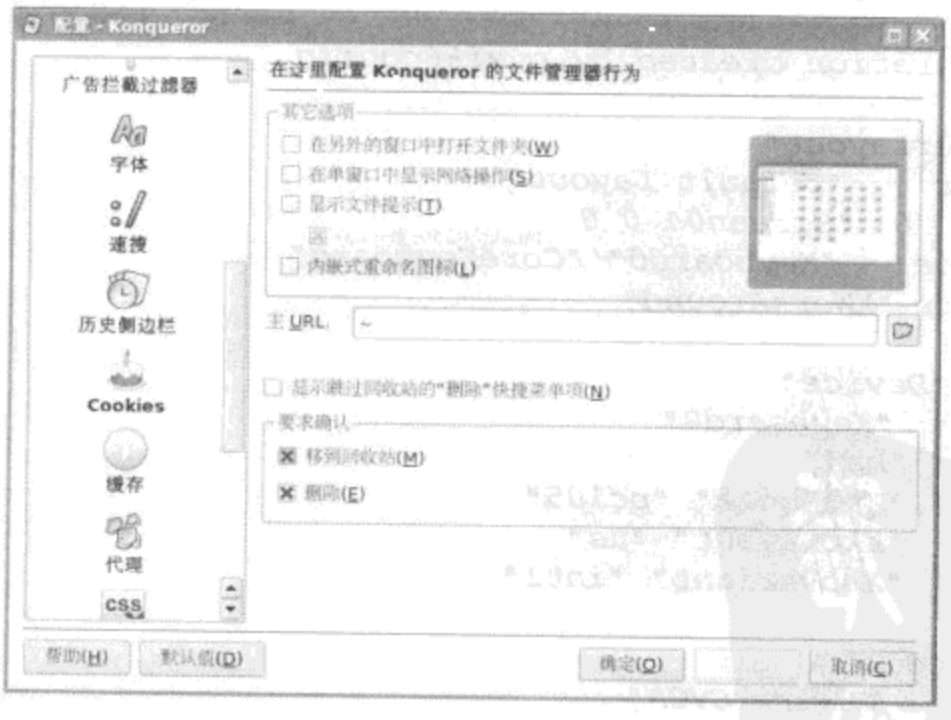


图 1-81 对 Konqueror 进行细调

1.7.3 定制自己的桌面

不同的用户对桌面环境的要求可能会有所不同，系统桌面只有具有较强的可定制性才能满足不同用户的需求。Red Hat Enterprise Linux 5 具有人性化设计的桌面管理，用户不仅可以实时提升系统的性能，还可以根据个人喜好定制个性化的桌面环境，使系统平台更加适合自己。

1. 配置 X windows

用户都喜欢简单、直观的图形操作界面，因此 Linux 能受到广大用户的欢迎与 X Window 是密不可分的。

X Window 是 Red Hat Enterprise Linux 5 中的图形界面系统，包括两个部分：服务器端和客户端。这两部分与桌面管理器（KDE、GNOME）组合在一起，就构成了 Linux 下的图形界面操作环境。

（1） X Window 的成长历史

X Window（又称为 X），最早由 MIT（麻省理工学院）于 1984 年开发。通过不断发展与完善，X11R6 版本最终于 1987 年 9 月推出。

由 X11R6 发展起来的 XFree86 是专门给 Intel x86 架构机器使用的 X Window 系统，早期 Linux 使用的 X Window 核心几乎都是基于 XFree86 计划提供的，其遵循 GPL 许可发布。XFree86 于 2004 年发布的 4.4 版本将遵循新的 XFree86 1.1 许可证，由此另外一个 X 系统架构计划 Xorg 得到了发展。Xorg 遵循 GPL 发布，所以，当今的 Linux 基本都将 Xorg 作为 X 系统的核心。

（2） X 服务器的配置文件

X 服务器的配置文件主要用于早期的 Linux 用户配置图形界面。

XFree86 的配置文件是/etc/X11/XF86Config（区分大小写），Xorg 的配置文件是/etx/X11/xorg.conf（位于同目录下），二者的语法结构、配置方法类似。

Xorg 的配置文件 xorg.conf 内容如下。

```
# Xorg configuration created by pyxf86config

Section "ServerLayout"
    Identifier     "Default Layout"
    Screen      0  "Screen0" 0 0
    InputDevice    "Keyboard0" "CoreKeyboard"
    InputDevice "XWorkAround"
EndSection
Section "InputDevice"
    Identifier  "Keyboard0"
    Driver      "kbd"
    Option      "XkbModel" "pc105"
    Option      "XkbLayout" "us"
    Option      "XkbVariant" "intl"
EndSection
Section "Device"
    Identifier  "VMware SVGA"
    Driver      "vmware"
EndSection
Section "Screen"
```

```
    Identifier "Screen0"
    Device     "VMware SVGA"
    Monitor    "vmware"
    # Don't specify DefaultColorDepth unless you know what you're
    # doing. It will override the driver's preferences which can
    # cause the X server not to run if the host doesn't support the
    # depth.
    Subsection "Display"
        # VGA mode: better left untouched
        Depth     4
        Modes     "640x480"
        ViewPort  0 0
    EndSubsection
    Subsection "Display"
        Depth     8
        Modes     "800x600"
        ViewPort  0 0
    EndSubsection
    Subsection "Display"
        Depth     15
        Modes     "800x600"
        ViewPort  0 0
    EndSubsection
    Subsection "Display"
        Depth     16
        Modes     "800x600"
        ViewPort  0 0
    EndSubsection
    Subsection "Display"
        Depth     24
        Modes     "800x600"
        ViewPort  0 0
    EndSubsection
EndSection
Section "Monitor"
    Identifier    "vmware"
    VendorName    "VMware, Inc"
    HorizSync     1-10000
    VertRefresh   1-10000
EndSection
Section "InputDevice"
    Identifier  "XWorkAround"
    Driver      "void"
EndSection
```

（3） 确认用户的 Linux 系统使用的是何种 X 系统架构

要确认自己的 Linux 系统使用的是 XFree86 还是 Xorg，可以执行以下命令进行确认。

```
[root@localhost ~]#X -version
```

如果显示的版本号是 6.x 或者 7.x，则表明使用的 X 系统架构为 Xorg；如果显示的版本号是 4.x，表明使用的 X 系统架构为 XFree86。

```
[root@localhost Server]# X -version
X Window System Version 7.1.1
Release Date: 12 May 2006
X Protocol Version 11, Revision 0, Release 7.1.1
Build Operating System: Linux 2.6.9-67.0.7.ELsmp i686 Red Hat, Inc.
```

```
Current Operating System: Linux localhost.localdomain 2.6.18-92.el5xen #1 SMP
Tue Apr 29 13:45:57 EDT 2008 i686
Build Date: 02 April 2008
Build ID: xorg-x11-server 1.1.1-48.41.el5
        Before reporting problems, check http://wiki.x.org
        to make sure that you have the latest version.
Module Loader present
[root@localhost Server]#
```

命令执行显示的结果表明 Red Hat Enterprise Linux 5 操作系统采用的 X 系统架构是 Xorg 7.1.1。

（4） 图形界面配置 X

system-config-display 是现在的 Linux 包括 Red Hat Enterprise Linux 5 所提供的图形界面的 X 配置工具。该工具主要用来配置硬件设备和分辨率。其实现在用户不用配置，图形桌面也可以正常地工作。

打开该配置工具有两种方法：一种是使用图形界面，另一种是使用命令。

使用图形界面的方法如下。

如果用户使用的是 GNOME 桌面环境，可以依次单击“应用程序”→“系统设置”→“显示”菜单命令，显示图形界面 X 配置工具；如果是 KDE 桌面，则可以依次单击“K 菜单”→“系统”→“显示”菜单命令，显示如图 1-82 所示的图形界面 X 配置工具。



图 1-82 图形界面 X 配置工具

可以在 GNOME 或者 KDE 的终端窗口，执行指令。

```
[root@localhost ~]# system-config-display
```

2. 选择桌面管理器

在 Red Hat Enterprise Linux 5 中，GNOME 和 KDE 是用户所熟知的桌面环境。不过其还内置了另外两个桌面管理器：twm 和 mwm。其中前者内置于 X11R7.1 中，安装之后可以直接使用；后者则必须手工安装 openmotif 组件。可以执行如下命令打开 twm。

```
[root@localhost ~]#telinit 3
[root@localhost ~]#xinit -e /usr/bin/twm
```

命令执行后，用户就可以根据自己的需求选择桌面管理器。

第 2 章　FTP 服务器

随着网络技术的发展，以 HTTP 协议为基础的 WWW 服务已不能满足文件传输的需要，因此 FTP 服务（File Transfer Protocol，文件传输服务）随之产生。FTP 服务是一种专门用于文件传输的服务，该服务使用的是文件传输协议，使用该服务将极大地方便文件的传输与管理。在 Linux 服务器搭建 FTP 服务平台相对比较复杂，所以本章将详细讲解如何安装并配置 FTP 服务器。

2.1　FTP 服务简介

FTP 的出现已有相当长的时间，其最主要的功能是在服务器端与客户端之间进行文件的传输。FTP 是以 TCP 封包的模式来进行服务器与客户端之间的连接的，当连接建立后，用户便可以在客户端连接 FTP 服务器来进行文件的上传与下载，同时也可以直接管理用户在 FTP 服务器上的文件。

2.1.1　FTP 工作原理

FTP 是基于客户端/服务器模式的，其工作原理如下。

首先，客户端向服务器发出连接请求，同时客户端系统动态打开一个大于 1 024 的端口（比如 1031）等候服务器连接，如图 2-1 所示。

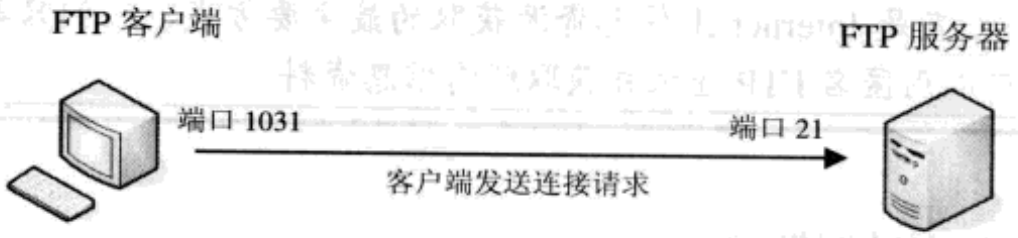


图 2-1　发出连接请求

当 FTP 服务器在端口 21 侦听到该请求后，会在客户端 1031 端口和服务器的 21 端口之间建立起一个 FTP 会话连接，如图 2-2 所示。

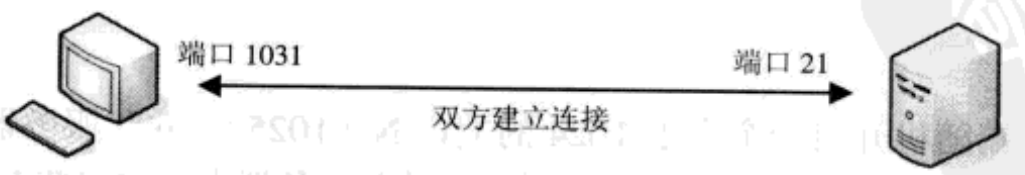


图 2-2　建立 FTP 会话连接

当需要传输数据时，FTP 客户端再动态打开一个大于 1 024 的端口（比如 1032）连接到服务器的 20 端口，并在这两个端口之间进行数据的传输，如图 2-3 所示。

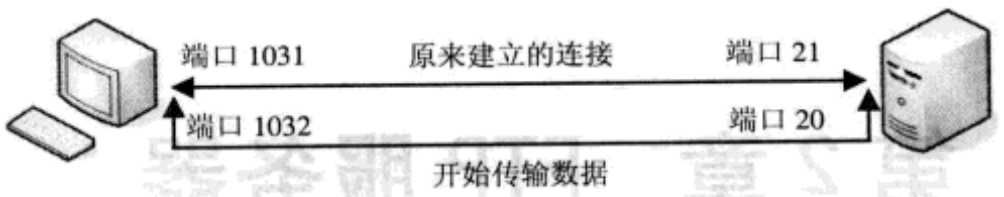


图 2-3　数据传输

数据传输完毕后，FTP 客户端将断开与 FTP 服务器的连接，客户端上动态分配的端口将自动释放掉，如图 2-4 所示。

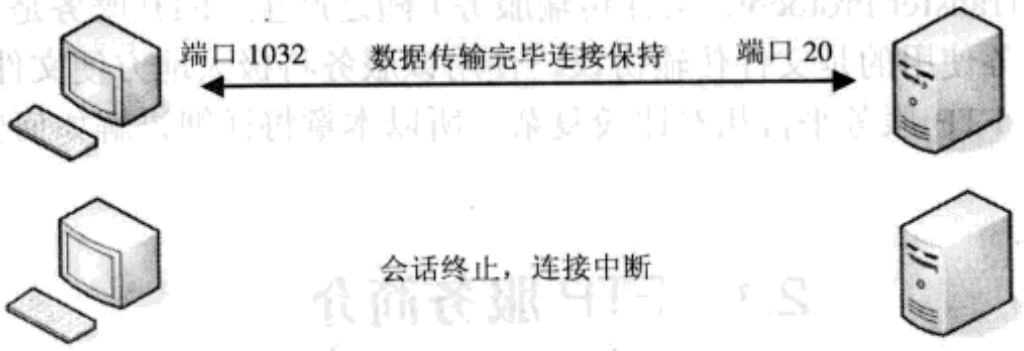


图 2-4　自动释放动态分配的端口

2.1.2 匿名用户访问的产生

要使用 FTP 服务来实现文件的传输，首先需要登录到服务器，然后才能实现该操作。但这对于很多公开提供软件下载的服务器来说非常不方便。为了解决该问题，匿名用户访问随之产生。匿名用户是通过使用一个公用的用户名 Anonymous，密码不限的管理策略（密码一般是用户的邮箱），让任何用户都可以方便地从这些服务器上下载相关资源。

注意　匿名文件传输使用户以匿名身份与远程主机建立起连接，并被允许从远程主机上复制文件。匿名 FTP 一直是 Internet 上信息资源获取的最主要方式，人们只要知道特定信息资源的主机地址，就可以用匿名 FTP 登录并获取所需信息资料。

2.1.3 FTP 服务的连接模式

FTP 不是单一地支持某种工作模式，而是支持主动模式（Active FTP）和被动模式（Passive FTP）两种，下面对两种模式进行简单的介绍。

1. 主动模式

首先，FTP 客户端随机开启一个大于 1024 的端口 N（1025），并和服务器的 21 号端口建立连接，然后开放 N+1 号端口（1026）进行监听，同时向服务器发出 PORT 1026 命令（PORT 命令包括客户端用什么端口接收数据）。服务器端在传送数据的时候，通过自己的 TCP 20 端口发送数据，因此 FTP 必须和客户端建立一个新的连接用于数据传输，如图 2-5 所示。

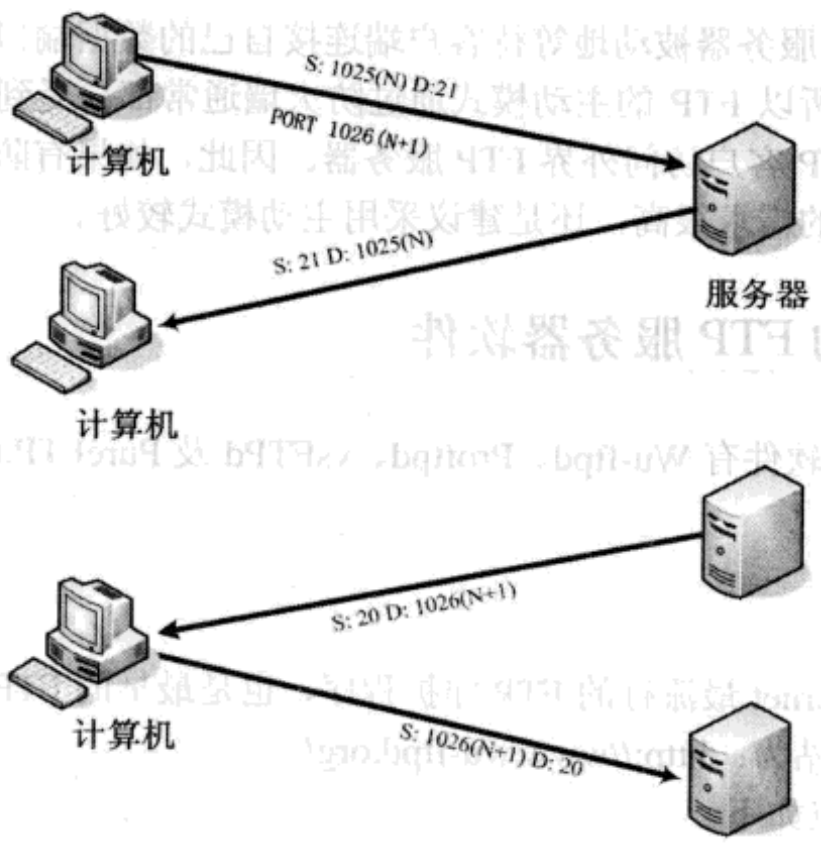


图 2-5　FTP 的主动传输模式

2. 被动模式

在被动模式下建立控制通道类似于在主动模式下建立通道的操作：FTP 客户端随机开启一个大于 1024 的端口 N（1025）向服务器的 21 号端口发起连接，同时会开启 N+1 号端口（1 026），然后向服务器发送 PASV 命令，通知服务器处于被动模式。服务器收到命令后，开放一个大于 1024 的端口 P（1521）进行监听，然后用 PORT P 命令通知客户端，自己的数据端口是 1521。客户端收到命令后，通过 1026 号端口连接服务器的端口 1521，然后在两个端口之间进行数据传输，如图 2-6 所示。

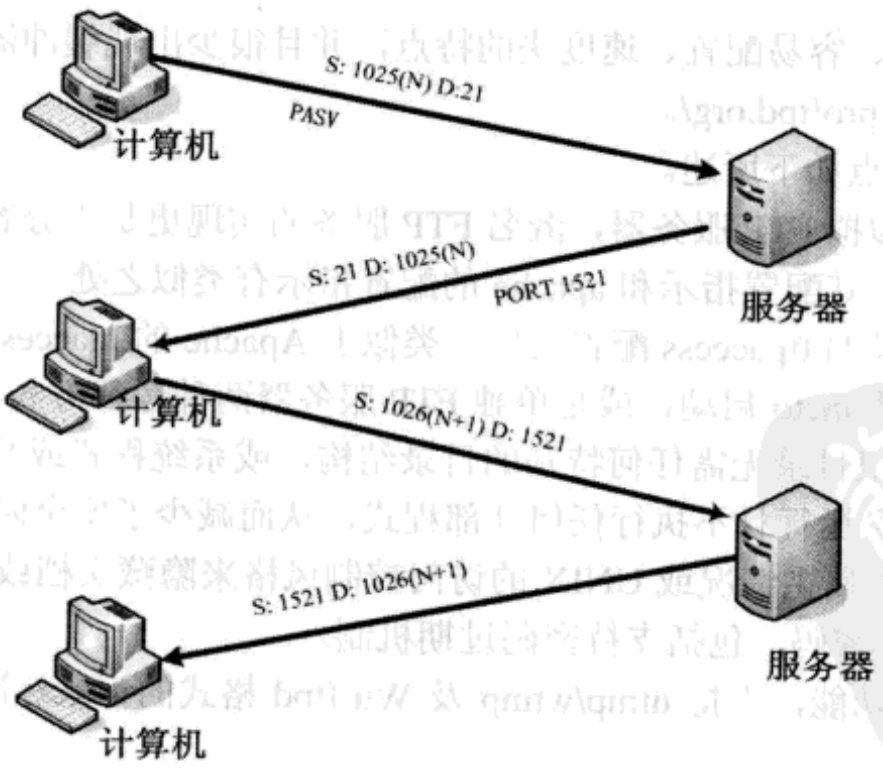


图 2-6　FTP 的被动传输模式

通过对这两种模式的介绍可以看出，主动模式的 FTP 是服务器主动连接客户端的数据端

口，被动模式的 FTP 是服务器被动地等待客户端连接自己的数据端口。多数防火墙都不允许接受外部发起的连接，所以 FTP 的主动模式通过防火墙通常都会受到限制。FTP 的被动模式通常用于防火墙后的 FTP 客户访问外界 FTP 服务器。因此，如果有防火墙，最好是采用被动模式，但是如果对安全的需求很高，还是建议采用主动模式较好。

2.1.4 几种流行的 FTP 服务器软件

常用的 FTP 服务器软件有 Wu-ftpd、Proftpd、vsFTPd 及 PureFTPd 等，下面分别对这些软件逐一介绍。

1. Wu-ftpd

Wu-ftpd 曾经是 Internet 最流行的 FTP 守护程序，也是最早的 FTP 服务器软件之一，拥有强大的功能。其官方网站为：http://www.wu-ftpd.org/。

Wu-ftpd 的主要特点如下。

- 支持虚拟 FTP 主机（Virtual FTP Server）。
- 能够控制不同网域的用户对于 FTP 服务器的存取权限和访问时段。
- 能够记录文档上传和下载的全过程并且可以限制最高访问人数。
- 使用者在下载文档时，能够自动对其进行压缩和解压工作。
- 能够暂时关闭 FTP 服务器，以便系统维护。
- 能够支持匿名 FTP 服务，但需要加载 anonftp 软件包。

虽然 Wu-ftpd 有着极佳的性能同时也是一套很好的软件，然而它有不少的安全漏洞陆续被发现。

2. Proftpd

Proftpd 具有安全、容易配置、速度快的特点，并且很少出现缓冲溢出的错误现象。其官方网站为：http://www.proftpd.org/。

Proftpd 的主要特点如下所述。

- 可设定多个虚拟 FTP 服务器，匿名 FTP 服务的实现更是十分容易。
- 单配置文档，其配置指示和 apache 的配置指示有类似之处。
- 基于单个目录的.ftpaccess 配置文档，类似于 Apache 的.htaccess 文档。
- 能够配置为从 inetd 启动，或是单独 FTP 服务器两种运行方式。
- 匿名 FTP 的根目录无需任何特别的目录结构，或系统程式或其他系统文档。
- 以非 root 身份运行且不执行任何外部程式，从而减少了安全隐患。
- 能够根据文档属主情况或 UNIX 的访问控制风格来隐藏文档或目录。
- 支持 Shadow 密码，包括支持密码过期机制。
- 强大的 Log 功能，支持 utmp/wtmp 及 Wu-ftpd 格式的记录标准，并支持扩展功能的日志记录。

3. vsFTPd

vsFTPd 即 Very SecureFTPd 的缩写形式，是 Red Hat Enterprise Linux 5 内置的 FTP 服务器

软件，支持很多其他 FTP 服务器不支持的功能，具有非常高的安全特性，同时支持带宽限制、IPv6 协议、分配虚拟 IP 地址、创建虚拟用户等功能。其良好的可伸缩性和中等偏上的性能获得了广大用户的广泛欢迎。其官方网站为：http://vsftpd.beasts.org/。

4. PureFTPd

PureFTPd 是内置在 SuSE、Debian 中的 FTP 服务器软件，但 Red Hat Enterprise Linux 5 中没有包含它的软件包，需手动加载，其官方网站为：http://www.pureftpd.org/。

5. Serv-U

Serv-U 是 Windows 系统下最常用的的 FTP 服务器。用户可以将任何一台 PC 机设置成一台 FTP 服务器。用户或其他使用者就能够使用 FTP 协议，通过在同一网络上的任何一台 PC 机与 FTP 服务器连接，进行文件或目录的复制、移动、创建和删除等。其官方站点为 http://www.rhinosoft.com.cn/。

Serv-U 的特点如下。

- 支持实时的多用户连接，支持匿名用户的访问；通过限制同一时间允许的最大的用户访问量来确保 PC 机的正常运转。
- 安全性能出众。在目录和文件层次都可以设置安全防范措施。为不同用户提供不同设置，支持分组管理数量众多的用户，甚至可以基于 IP 对用户授予访问权限。
- 能够设置上传和下载的比率、硬盘空间配额、网络使用带宽等，从而有效分配资源，还可作为系统服务在后台运行。
- 支持文件上传和下载过程中的断点续传；支持拥有多个 IP 地址的多宿主站点。
- 可设置在用户登录或退出时的显示信息，支持具有 UNIX 风格的外部链接。

上面列出的只是 Serv-U 众多功能中的一部分，Serv-U 不仅功能强大，也提供了易于使用的操作界面，是 Windows 下使用最方便的 FTP 服务器软件。

2.2 安装 vsFTPd 服务

在借助任何一种服务实现某种功能之前都需要首先确保该服务是否已经安装，FTP 服务也不例外。在 Red Hat Enterprise Linux 5 下，有 Red Hat Content Acceletatot 和 vsFTPd 两款内置的 FTP 服务器软件。其中，Red Hat Content Acceletatot 提供了 Web 和 FTP 两种服务，并具有高效的特点，但功能较弱。而 vsFTPd 在提供 FTP 服务方面具有高效性外还具有很强的安全性和稳定性，因此这里将主要讲解 vsFTPd 服务的安装。

2.2.1 安装 vsFTPd 软件

由于安装 vsFTPd 服务时，没有要依赖的软件包，因此安装起来极为方便，只要安装一个 RPM 软件包即可。在安装前，首先需要确认该服务在系统中是否已经安装，要实现此操作，可以通过执行以下命令完成。

```
[root@localhost ~]#rpm -q vsftpd
```

执行命令后的结果如下所示，显示了“vsftpd-2.0.5-12.e15”的字样，表明该服务已经安装。

```
[root@localhost ~]# rpm -q vsftpd
vsftpd-2.0.5-12.el5
```

如果检测到系统中还没有安装该服务，则可通过以下方法实现安装。

首先将 Red Hat Enterprise Linux 5 安装光盘放入光驱，然后加载光驱，在光盘的 Server 目录下找到 vsFTPd 服务的 RPM 安装包，最后执行以下命令进行安装。

```
[root@localhost Server]#rpm -ivh vsftpd-2.0.5-12.el5.i386.rpm
```

命令执行结果如下所示，表明 vsFTPd 服务成功安装。

```
[root@localhost Server]# rpm -ivh vsftpd-2.0.5-12.el5.i386.rpm
warning: vsftpd-2.0.5-12.el5.i386.rpm: Header V3 DSA signature: NOKEY, key ID
37017186
Preparing...      ########################################### [100%]
   1: vsftpd      ########################################### [100%]
```

2.2.2 FTP 服务的启动与关闭

在完成 vsFTPd 服务的安装后，接下来学习如何启动、重新启动以及停止 vsFTPd 服务的操作。

1. 启动 vsFTPd 服务

要启动 vsFTPd 服务，只需要执行以下命令。

```
[root@localhost ~]# service vsftpd start
```

执行命令后的结果如下所示，表明启动成功。

```
[root@localhost ~]# service vsftpd start
为 vsftpd 启动 vsftpd:                                    [确定]
```

注意 也可以通过执行以下命令实现服务的启动。

```
[root@localhost ~]# /etc/init.d/vsftpd start
```

执行命令后的结果如下所示。

```
[root@localhost ~]# /etc/init.d/vsftpd start
为 vsftpd 启动 vsftpd:                                    [确定
```

---

2. 重新启动 vsFTPd 服务

可以通过执行以下命令实现 vsFTPd 服务的重启操作。

```
[root@localhost ~]# service vsftpd restart
```

执行命令后的结果如下所示，表明 vsFTPd 服务已成功重启。

```
[root@localhost ~]# service vsftpd restart
关闭 vsftpd:                                               [确定]
为 vsftpd 启动 vsftpd:                                     [确定]
```

注意 重启 vsFTPd 服务的操作同样可以通过执行以下命令实现。

```
[root@localhost ~]#/etc/rc.d/init.dvsftpd restart
```

执行命令后的结果如下所示。

```
[root@localhost ~]#/etc/rc.d/init.dvsftpd restart
关闭 vsftpd:                                               [确定]
为 vsftpd 启动 vsftpd:                                     [确定]
```

同时也可以通过执行以下命令实现，但此种情况只适合于 vsFTPd 是运行状态的情况。

```
[root@localhost ~]#/etc/rc.d/init.dvsftpd condrestart
```

3. 停止 vsFTPd 服务

vsFTPd 服务的停止操作类似于该服务的启动操作，要实现停止该服务可以通过执行以下命令实现。

```
[root@localhost ~]# service vsftpd stop
```

执行命令后的结果如下所示，表明 vsFTPd 服务已经停止。

```
[root@localhost ~]# service vsftpd stop
关闭 vsftpd:                                               [确定]
```

注意 停止 vsFTPd 服务也可以通过执行以下命令实现。

```
[root@localhost ~]#/etc/init.d/vsftpd stop
```

执行命令后的结果如下所示。

```
[root@localhost ~]#/etc/init.d/vsftpd stop
关闭 vsftpd:                                               [确定]
```

4. 自动启动 vsFTPd 服务

启动的 vsFTPd 服务只能保持到此次计算机关闭之前，下次重启计算机后，又需要手动启动 vsFTPd 服务。为了避免此种情况的发生，则可以设置 vsFTPd 服务随系统启动而自动加载，具体方法如下。

首先执行“ntsysv”命令启动自动服务配置程序，然后找到“vsftpd”服务选项，按 Space 键在其前面加上星号“*”，最后按 Tab 键，选择“确定”按钮即可，如图 2-7 所示。

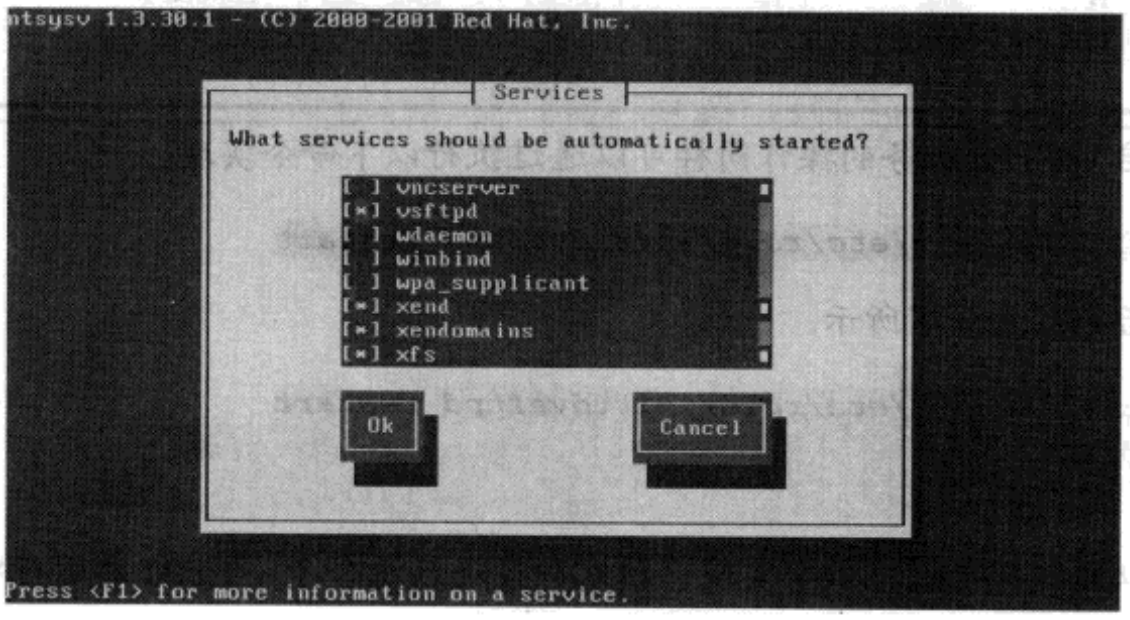


图 2-7 配置 vsFTPd 服务自启动

2.3 vsFTPd 的相关配置

vsFTPd 服务的配置都是通过相关配置文件实现的。该服务的配置文件有多个，包括其主配置文件/etc/vsftpd/vsftpd.conf，以及加强 vsFTPd 服务器用户认证的/etc/pam.d/vsftpd 文件等。如图 2-8 所示。

```
# Example config file /etc/vsftpd/vsftpd.conf
#
# The default compiled in settings are fairly paranoid. This sample file
# loosens things up a bit, to make the ftp daemon more usable.
# Please see vsftpd.conf.5 for all compiled in defaults.
#
# READ THIS: This example file is NOT an exhaustive list of vsftpd options.
# Please read the vsftpd.conf.5 manual page to get a full idea of vsftpd's
# capabilities.
#
# Allow anonymous FTP? (Beware - allowed by default if you comment this out).
anonymous_enable=YES
#
# Uncomment this to allow local users to log in.
local_enable=YES
#
# Uncomment this to enable any form of FTP write command.
write_enable=YES
#
# Default umask for local users is 077. You may wish to change this to 022,
# if your users expect that (022 is used by most other ftpd's)
local_umask=022
#
"/etc/vsftpd/vsftpd.conf" 116L, 4397C
```

图 2-8 vsFTPd 的主配置文件 vsftpd.conf

- /etc/pam.d/vsftpd

vsFTPd 的 The Pluggable Authentication Modules（PAM）配置文件。它主要用来加强 vsFTPd 服务器的用户认证。

- /etc/vsftpd.ftpusers

所有位于此文件内的用户都不能访问 vsFTPd 服务。当然，为了安全起见，这个文件中默认已经包括了 root、bin 和 daemon 等系统账号。

- /etc/vsftpd.user_list

这个文件中包括的用户有可能是被拒绝访问 vsFTPd 服务的，也可能是允许访问的，这主要决定于 vsFTPd 的主配置文件/etc/vsftpd/vsftpd.conf 中的“userlist_deny”参数设置的是“Yes”（默认值）还是“No”。

- /var/ftp

vsFTPd 提供服务的文件集散地，包括一个 pub 子目录。默认配置下，所有的目录都是只读的，不过只有 root 用户有写权限。

2.3.1 配置匿名用户登录环境

在 vsFTPd 的配置文件 vsftpd.conf 中，注释行应以“#”开头，每个选项设置为一行，如图 2-9 所示。在该文件中，默认情况下会存在以下语句。

```
anonymous_enable=YES
```

表明 vsFTPd 服务是允许匿名访问的，且这里的匿名服务器是必须配置的。

```
# Example config file /etc/vsftpd/vsftpd.conf
#
# The default compiled in settings are fairly paranoid. This sample file
# loosens things up a bit, to make the ftp daemon more usable.
# Please see vsftpd.conf.5 for all compiled in defaults.
#
# READ THIS: This example file is NOT an exhaustive list of vsftpd options.
# Please read the vsftpd.conf.5 manual page to get a full idea of vsftpd's
# capabilities.
#
# Allow anonymous FTP? (Beware - allowed by default if you comment this out).
anonymous_enable=YES
#
# Uncomment this to allow local users to log in.
local_enable=YES
#
# Uncomment this to enable any form of FTP write command.
write_enable=YES
#
# Default umask for local users is 077. You may wish to change this to 022,
# if your users expect that (022 is used by most other ftpd's)
local_umask=022
#
"/etc/vsftpd/vsftpd.conf" 116L, 4397C
```

图 2-9 vsftpd.conf 的注释

2.3.2 监听地址与控制端口

如果需要改变 FTP 的用来提供服务的默认端口，可以通过修改 vsFTPd 的主配置文件来实现。

1. 设置监听地址

配置文件中默认只有是否监听的参数，没有关于监听地址的配置参数。若要监听地址则需要添加如下内容。

```
listen_address=ip address
```

此参数在 vsFTPd 使用独立模式（standalone）时有效，定义了在主机的哪个 IP 地址上监听 FTP 请求，即在哪个 IP 地址上提供 FTP 服务。对于只有一个 IP 地址的主机，不需要使用此参数。对于多址主机，不设置此参数，即将监听所有 IP 地址，默认值为无。

2. 控制端口

控制端口是 vsFTPd 服务器监听的端口号。与监听地址一样，控制端口在独立模式下有效，默认值为 21，配置控制端口需要在配置文件中添加关于控制端口的参数，格式如下。

```
listen_port=port_value
```

2.3.3 FTP 模式与数据端口

FTP 有两种工作模式，建立控制连接时都是由客户端和 FTP 服务器的控制端口（默认值为 21）建立控制连接，然后传输操作指令的。区别在于使用数据传输端口（FTP-DATA）的方式不同，主动模式由 FTP 服务器指定数据传输所使用的端口，默认值为 20；被动模式由 FTP 客户端决定数据传输的端口。在 vsftpd.conf 配置文件中，也可以针对主动模式和被动模式对 vsFTPd 服务器进行配置。

1. 主动模式

主动模式的相关配置参数如表 2-1 所示。

表 2-1 主动模式的相关配置参数

参 数	相 关 说 明
port_enable=YES|NO	如果要在数据连接时取消主动模式时设此选项为 NO，默认值为 YES
connetc_from_port_20=YES|NO	控制以主动模式进行数据传输时是否使用 20 端口(FTP-DATA)。YES 使用，NO 不使用，默认值为 NO。RHL 自带的 vsftpd.conf 文件中此参数设为 YES
ftp_data_port=port numbe r	设定 FTP 数据传输端口（FTP-DATA）值，默认值为 20。此参数用于主动模式
port_promiscuous=YES|NO	默认值为 NO。如果为 YES，取消主动安全检查，该检查确保外出的数据只能连接到客户端上，打开此选项时要小心

2. 被动模式

被动模式的相关配置参数如表 2-2 所示。

表 2-2 被动模式的相关配置参数

参　数	相 关 说 明
pasv_enable=YES|NO	YES，表示数据传输时允许使用被动模式；NO，则表示不允许使用。默认值为 YES
pasv_min_port=port number	设定在被动模式下，建立数据传输可以使用 port 范围的下界和上界。0 表示任意，默认值为 0。把端口范围设在数值比较高的一段范围内，比如 50 000~60 000，将有助于安全性的提高
pasv_promiscuous=YES|NO	此选项被激活时，将关闭被动模式的安全检查。该检查确保数据连接和控制连接是来自同一个 IP 地址的。小心打开此选项。此选项唯一合理的用法是，存在于由安全隧道方案构成的组织，默认值为 NO
pasv_address= IP address	此选项为一个数字 IP 地址，作为被动命令的响应。默认值为 none，即地址是从呼入的连接套接字中获取的

2.3.4 ASCII 模式

默认情况下，vsFTPd 是禁止使用 ASCII 传输模式的，以下选项参数控制 vsFTPd 是否使用 ASCII 传输模式。

- ascii_download_enable=YES|NO

控制是否允许使用 ASCII 模式下载文件，YES 允许，NO 不允许，默认为 NO。

- ascii_upload_enable=YES|NO

控制是否允许使用 ASCII 模式上传文件，YES 允许，NO 不允许，默认为 NO。

2.3.5 超时选项

为了节约系统资源，vsFTPd 还提供了超时定义设置，以防止客户端无限制时间地连接在 FTP 服务器上。

- idle_session_timeout

空闲用户会话的超时时间，如果超出这个时间没有数据的传送或是指令的输入，就会强迫断线。单位为 s，默认值为“300”。

- data_connection_timeout

定义数据传输过程中被阻塞的最长时间（以 s 为单位），一旦超出这个时间，客户端的连接将被关闭。单位为 s，默认值为“300”。

以下两个选项是针对客户端的，客户端空闲 1 min 后将自动中断连接，并在中断 1 min 后自动激活连接。

- accept_timeout

接受建立链接的超时设定。单位为 s，默认值为“60”。

- connect_timeout

响应主动模式的数据连接的超时设定，单位为 s，默认值为“60”。

2.3.6 负载控制

FTP 客户端占用过大的宽带将影响服务器的正常运行。为防止客户端占用过大的带宽，保证服务器正常运行，可以通过设置以下参数来实现，如表 2-3 所示。

表 2-3 设置负载控制

参　数	相 关 说 明
max_clients=numerical value	此参数在 vsFTPd 使用独立模式下有效。定义了 FTP 服务器最大的并发连接数，当超过此连接数时，FTP 服务器拒绝客户端连接。默认值为 0，表示不限制最大连接数
max_per_ip=numerical value	此参数在 vsFTPd 使用独立模式下有效。定义了每个 IP 地址最大的并发连接数目，超过这个数目将拒绝连接。此选项的设置将影响到如迅雷、BT 等多线程下载软件。默认值为 0，表示不限制
anon_max_rate=value	设定匿名用户的最大数据传输速度 value，以 b/s 为单位，默认值无
local_max_rate=value	设定本地用户的最大数据传输速度 value，以 b/s 为单位，默认值无。此选项对所有的用户都生效，还可以在用户个人配置文件中使用此选项，以指定特定用户可获得的最大数据传输速率

2.3.7 用户设置

FTP 中有匿名用户、本地用户和虚拟用户 3 种。相应的配置参数如下。

1. 匿名用户

前面对匿名用户的登录作了简单的介绍，下面将对匿名用户的配置参数进行详细的介绍，如表 2-4 所示。

表 2-4 匿名用户配置参数说明

参　数	相 关 说 明
anonymous_enable	控制是否允许匿名用户登录。YES 允许，NO 不允许，默认值为 YES
no_anon_password	控制匿名用户登入时是否需要密码。YES 不需要，NO 需要，默认值为 NO
ftp_username	指定匿名用户与本地账号相对应，该用户的家目录即为匿名用户访问 FTP 服务器时的家目录。默认值为“ftp”
banned_email_file	此参数默认值为 NO。当值为 YES 时，拒绝使用 banned_email_file 参数指定文件中所列出的用 E-mail 地址进行登录的匿名用户。即当匿名用户使用 banned_email_file 文件中所列出的用 E-mail 进行登录时将被拒绝，这对于阻击某些 Dos 攻击有效。当此参数生效时，需追加 banned_email_file 参数，banned_email_file 指定包含被拒绝的 E-mail 地址的文件，默认文件为/etc/vsftpd.banned_emails
anon_root	设定匿名用户的根目录，即匿名用户登入后，被定位到此目录下。主配置文件中默认无此项，默认值为/var/ftp/
anon_world_readable_only	控制是否只允许匿名用户下载可阅读文档。当值为 YES 时，允许匿名用户下载可阅读的文件；为 NO 时，则允许匿名用户浏览整个服务器的文件系统。默认值为 YES
anon_upload_enable	控制是否允许匿名用户上传文件，YES 为允许，NO 为不允许，默认是不设值，即为 NO。除了这个参数外，匿名用户如果要上传文件，还需要两个条件：write_enable 参数为 YES；以及在文件系统上，FTP 匿名用户对某个目录有写权限
anon_mkdir_write_enable	控制是否允许匿名用户创建新目录，YES 为允许，NO 为不允许，默认是不设值，即为 NO。在文件系统上，FTP 匿名用户必须对新目录的上层目录拥有写权限
anon_other_write_enable	控制匿名用户是否拥有除了上传和新建目录之外的其他权限，如删除、更名等。YES 为拥有，NO 为不拥有，默认值为 NO
chown_uploads	是否修改匿名用户所上传文件的所有权。当值为 YES 时，匿名用户所上传的文件所有权将改为另外一个不同的用户所有，用户由 chown_username 参数指定。此选项默认值为 NO
chown_username	指定拥有匿名用户上传文件所有权的用户。此参数与 chown_uploads 联用。不推荐使用 root 用户

2. 本地用户

在使用 FTP 服务的用户中，除了匿名用户外，还有一类在 FTP 服务器所属主机上拥有账号的用户，在 vsFTPd 中称此类用户为本地用户（local users），其具体配置参数如表 2-5 所示。

表 2-5 本地用户配置参数说明

参　数	相 关 说 明
local_enable	是否允许本地用户登录。默认值为“YES”，也就是允许本地用户访问 vsFTPd 服务器。以下选项只有在“local_enable=YES”的前提下才有效
chroot_local_user	当设置为“YES”时，本地用户只能访问到家目录，不能切换到家目录之外
chroot_list_enable	当设置为“YES”时，表示本地用户也有例外，可以切换到家目录之外，例外的用户在“chroot_list_file 指定的文件中（默认文件是“/etc/vsftpd.chroot_list”）
chmod_enable	当设置为“YES”时，以本地用户登录的客户端可以通过“SITE CHMOD”指令来修改文件的权限
local_umask	设置文件创建的掩码（操作方法与 Linux 下文件属性设置相同），默认值为“022”，也就是其他用户具有只读属性
local_root	指定本地用户登录 vsFTPd 服务器时切换到的目录。没有设置默认值

3. 虚拟用户

考虑到安全方面的问题，vsFTPd 除了支持本地用户和匿名用户之外，还支持虚拟用户，虚拟用户的具体配置参数如下。

- guest_enable

当设置为“YES”时，所有非匿名用户都被映射为一个特定的本地用户。该用户通过“guest_username”指令指定。默认值为“No”。

- guest_enable

设置虚拟用户映射到的本地用户，默认值为“ftp”。

2.3.8 用户登录控制

vsFTPd 提供了丰富的登录控制选项，包括登录后客户端可以显示的信息、允许执行的指令以及一些登录中的控制选项等，如表 2-6 所示。

表 2-6 登录控制选项

参　数	相 关 说 明
banner_file	设置客户端登录 vsFTPd 服务器后，服务器显示在客户端的信息被保存在“banner_file”指定的文本文件中
cmds_allowed	设置客户端登录 vsFTPd 服务器后客户端可以执行的指令集合。需要注意的是，如果设置了该指令，则其他没有列在其中的指令都拒绝执行，没有设置默认值
ftpd_banner	设置客户端登录 vsFTPp 服务器后客户端显示的欢迎或者其他相关信息。需要注意的是，如果设置了“banner_file”，本指令将被忽略，没有设置默认值
userlist_enable	此选项被激活后，vsFTPd 将读取 userlist_file 参数所指定的文件中的用户列表。当列表中的用户登录 FTP 服务器时，该用户在提示输入密码之前就被禁止了。即该用户名输入后，vsFTPd 查到该用户名在列表，vsFTPd 就直接禁止该用户，不再进行询问密码等后续步骤。默认值为 NO
userlist_deny	决定禁止还是只允许由 userlist_file 指定文件中的用户登录 FTP 服务器。此选项在 userlist_enable 选项启动后才生效。默认值为 YES，禁止文件中的用户登录，同时也不向这些用户发出输入口令的提示。设置为 NO，只允许文件中的用户登录 FTP 服务器
cp_wrappers	在 vsFTPd 中使用 TCP_Wrappers 远程访问控制机制，默认值为 YES

2.3.9 目录访问控制

vsFTPd 对目录的访问也设置了一些控制选项，如表 2-7 所示。

表 2-7 目录访问控制选项

参　数	相 关 说 明
hide_ids	隐藏文件的所有者和组信息，匿名用户看到的文件所有者和组全部变成 FTP
message_file	用于指定目录切换时显示的信息所在的文件，默认值为“.message”
force_dot_files	设置是否显示以“.”开头的文件，默认不显示
dirmessage_enable	设置当用户切换到一个目录时，是否显示目录切换信息。如果设置为“YES”，则显示“message_file”指定文件中的信息（默认显示.message 文件信息）
dirlist_enable	设置是否允许用户列目录，默认值为“YES”——允许列目录

2.3.10 文件操作控制

vsFTPd 提供了几个用于控制文件的上传和下载的参数，如表 2-8 所示。

表 2-8 vsFTPd 控制上传和下载的参数

参　数	相 关 说 明
download_enable	设置是否允许下载。默认值为“YES”，即允许下载
chown_username	设置匿名用户上传的文件的拥有者，默认值“root”
chown_uploads	当设置为“YES”时，所有匿名用户上传的文件，其拥有者被设置为“chown_username”指令指定的用户。默认值“NO”
write_enable	当设置为“YES”时，FTP 客户端登录后允许使用 DELE（删除文件）、RNFR（重命名）和 STOR（断点续传）指令

2.3.11 新增文件权限设置

vsFTPd 可以记录服务器的工作状态，包括客户端的上传、下载操作等。

- log_ftp_protocol

是否记录所有的 FTP 命令信息。默认值为“NO”。

- dual_log_enable

如果启用则生成两个相似的日志文件，分别为/var/log/xferlog 和/var/log/vsftpd.log。前者是 Wu-ftpd 类型的传输日志，可以用于标准工具分析；后者是 vsFTPd 类型的日志。默认值为“NO”。

- syslog_enable

设置为“YES”时将本来应记录在/var/log/vsftpd.log 的信息，传给 syslogd daemon，由 syslogd 的配置文件决定存储位置。默认值为“NO”。

2.3.12 日志设置

vsFTPd 还可以进行传输日志文件的设置，具体配置选项如下。

- xferlog_std_format

启用后，传输日志文件将以标准 xferlog 的格式书写，如同 Wu-ftpd 一样。此格式的日志文件默认为/var/log/xferlog，也可以通过 xferlog_file 选项来设定。

- xferlog_enable

如果启用此选项将维护一个日志文件，用于详细记录上传和下载。默认情况下，这个日志文件是/var/log/vsftpd.log。但是也可以通过配置文件中的 vsftpd_log_file 选项来指定。默认值为“NO”。

2.4 安装与配置 PureFTPd 服务器的

PureFTPd 是一个高安全性的 FTP 软件。PureFTPd 服务器可以通过某些插件来实现 Web 界面用户管理（如 WebPureFTPd、PureFTPd PHP User Management），一般情况下会使用 MySQL+PureFTPd 的组合形式来共同实现这些功能。因此，还需要安装 Apache 服务以及 MySQL 服务。

2.4.1 安装 Apache 服务

在不确定系统中是否已经安装 Apache 服务时，可以通过执行以下命令进行查询。

```
[root@localhost ~]# rpm -q httpd
```

执行命令后的结果如下所示，表明系统中已经安装有该服务。

```
[root@localhost ~]# rpm -q httpd
httpd-2.2.3-11.el5_1.3
```

如果执行命令后显示“package httpd is not installed”字样，则表明 httpd 服务没有安装，此时应该准备相应的软件包进行安装，httpd 服务的 RPM 安装包可以通过安装光盘获取（在必要情况下，还需准备与其相依赖的相关软件包并进行安装），安装的过程如下。

1. 具体安装步骤

第 1 步，执行以下指令加载光驱。

```
[root@localhost ~]#mkdir /mnt/cdrom
[root@localhost ~]#mount /dev/cdrom /mnt/cdrom
```

第 2 步，在光盘的 Server 目录下找到 httpd 服务的 RPM 安装包文件 httpd-2.2.3-11.e15_1.3.i386.rpm。使用以下命令进行安装。

```
[root@localhost ~]#rpm -ivh
/mnt/cdrom/Server/postgresql-libs-8.1.11-1.el5_1.1.i386.rpm
[root@localhost ~]#rpm -ivh /mnt/cdrom/Server/apr-1.2.7-11.i386.rpm
[root@localhost ~]#rpm -ivh /mnt/cdrom/Server/apr-util-1.2.7-7.el5.i386.rpm
[root@localhost ~]#rpm -ivh
/mnt/cdrom/Server/httpd-2.2.3-11.el5_1.3.i386.rpm
[root@localhost ~]#rpm -ivh
/mnt/cdrom/Server/httpd-manual-2.2.3-11.el5_1.3.i386.rpm
```

注意 在需要的情况下还可以安装帮助文件 httpd-manual-2.2.3-11.e15_1.3.i386.rpm。

2. 测试是否安装成功

在安装完 httpd 服务后，需要测试其是否安装成功。执行以下指令启动 httpd 服务。

```
[root@localhost ~]# service httpd start
```

执行命令后的结果如下所示，表明启动成功。

```
[root@localhost ~]# service httpd start
启动 httpd:                                                [确定]
```

最后在客户端的浏览器中输入 Apache 服务器的 IP 地址进行访问，如果出现如图 2-10 所示的界面表明 Apache 服务器安装成功。

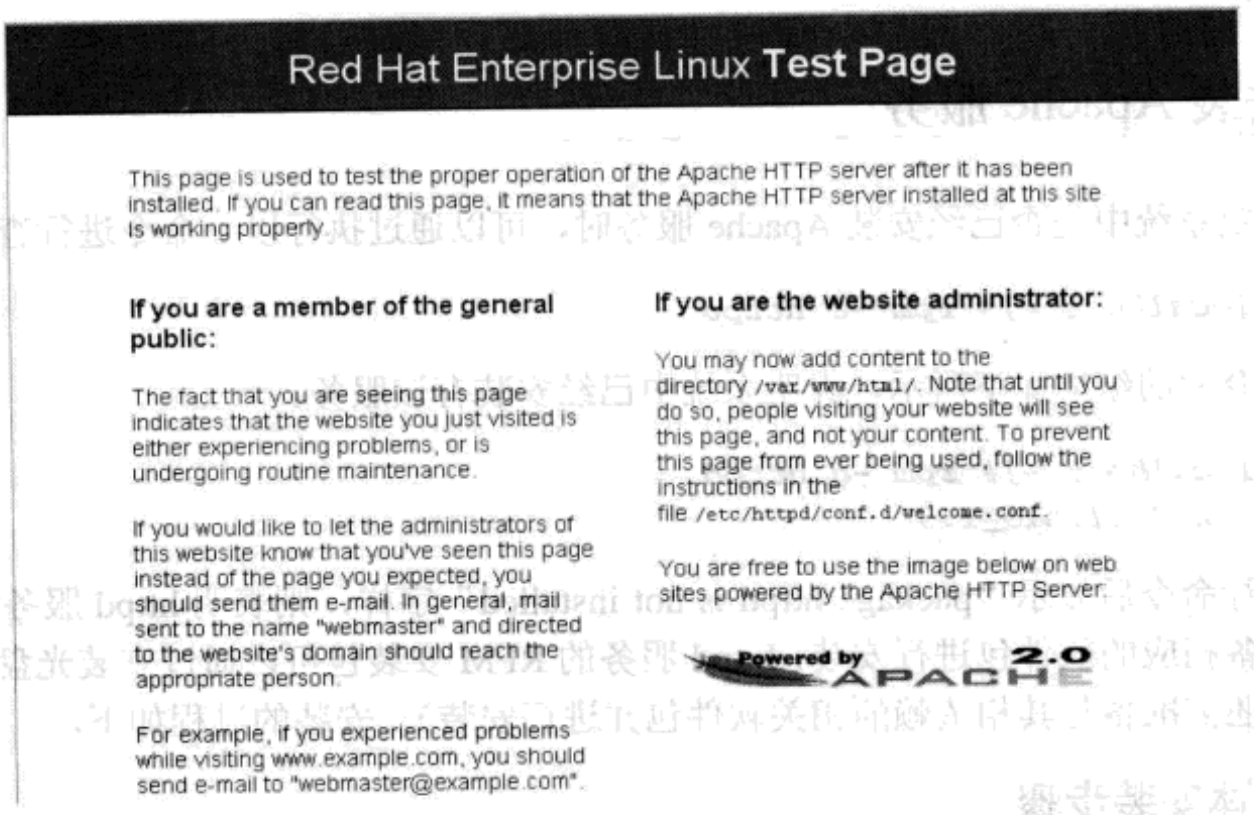


图 2-10　Apache 服务器安装成功

3. 为 Apache 服务放行

Red Hat Enterprise Linux 5 采用了增强的安全模式：SELinux。在默认的配置下，只有 SSH 服务可以通过，而 Apache 服务不能通过。Apache 服务在安装、配置、启动完毕后，还需要为其放行。

第 1 步，在命令行控制台窗口键入“setup”命令，执行后，将会打开 Linux 配置工具选择窗口，如图 2-11 所示。

第 2 步，在 Linux 配置工具选择窗口中选中“Firewall Configuration”选项，单击“Run Tool”按钮，此时将会打开“Firewall Configuration”窗口，如图 2-12 所示。一般情况下，会把“Security Level（安全级别）”设置为“Enabled（启用）”，把 SELinux 设置为“Enforcing（强制启用）”。

第 3 步，单击下侧的“Customize”按钮，将会打开 SELinux 配置窗口，在该窗口中，在需要运行的服务前面打上“*”号标记，如图 2-13 所示。

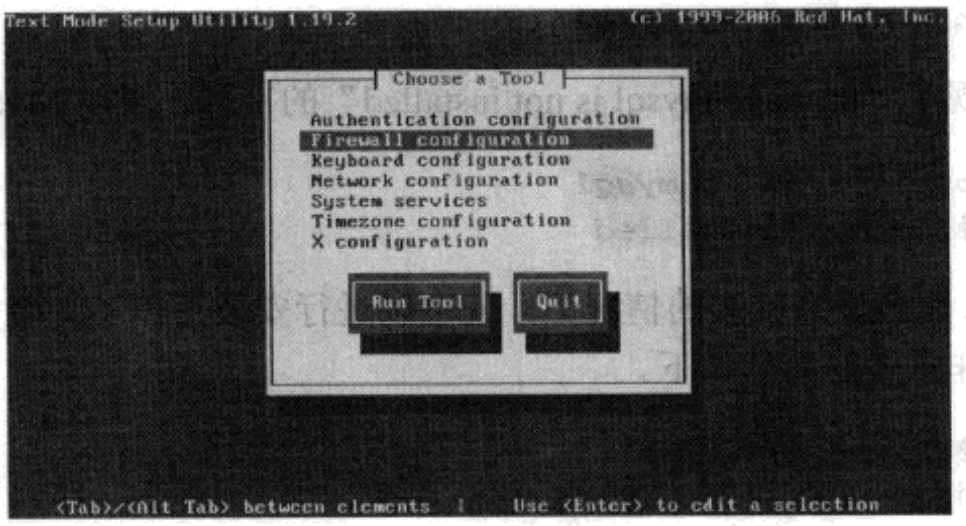


图 2-11　Linux 配置工具选择窗口

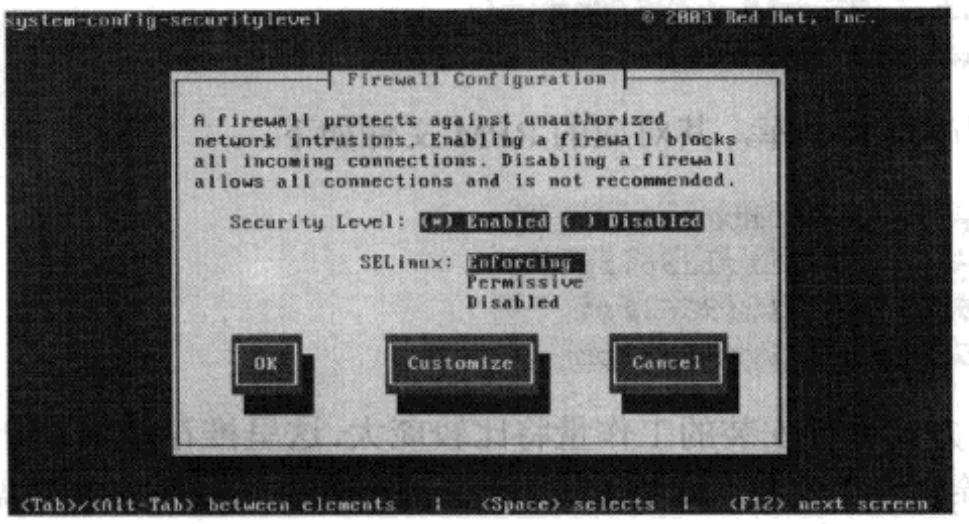


图 2-12　“Firewall Configuration”窗口

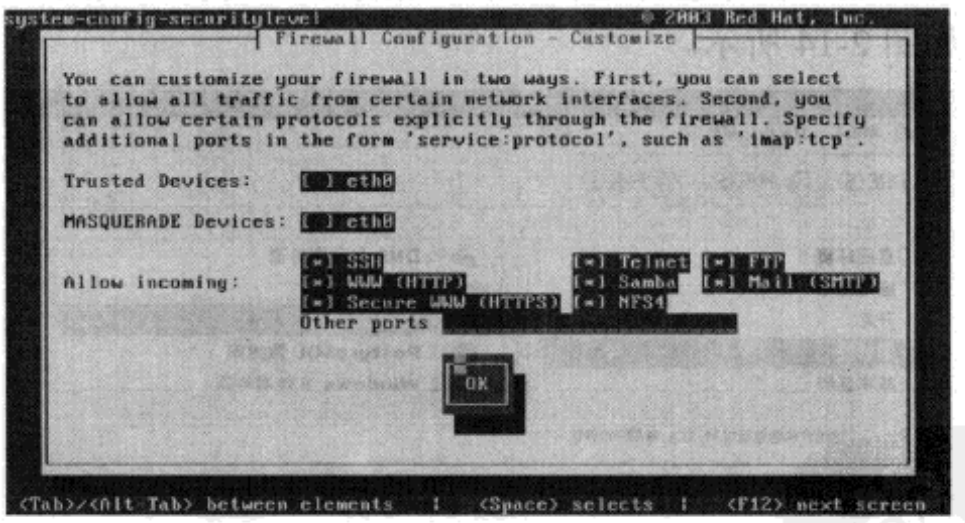


图 2-13　为 httpd 服务放行

注意　在需要运行的服务前面打上“*”号标记的方法：选中该条目后，按下 Space 键。

2.4.2 安装 MySQL 数据库

在安装 MySQL 之前首先需要确定 Red Hat Enterprise Linux 5 中是否已经安装有 MySQL 服务，在不确定的情况下，可以执行以下命令进行确认。

```
[root@localhost ~]#rpm -q mysql
```

执行命令后出现了“package mysql is not installed”的字样，表明 MySQL 服务没有安装。

```
[root@localhost ~]#rpm -q mysql
Package mysql is not installed
```

在确定 MySQL 服务没有安装的情况下，就可以进行安装工作了。而安装 MySQL 服务所需要的软件包文件非常多，具体如下。

```
perl-DBI-1.52-1.fc6.i386.rpm
perl-DBD-MySQL-3.0007-1.fc6.i386.rpm
openssl-devel-0.9.8b-10.el5.i386.rpm
e2fsprogs-devel-1.39-15.el5.i386.rpm
krb5-devel-1.6.1-25.el5.i386.rpm
zlib-devel-1.2.3-3.i386.rpm
```

MySQL 客户程序及共享库，其对应的 RPM 文件如下。

```
mysql-5.0.22-2.1.i386.rpm
mysql-server-5.0.22-2.1.i386.rpm
mysql-bench-5.0.22-2.1.i386.rpm
mysql-devel-5.0.22-2.1.i386.rpm
```

因此通过 RPM 方式进行安装的工作量将比较庞大，这里推荐使用 Red Hat Enterprise Linux 5 自带的“添加/删除软件”安装，即将 Red Hat Enterprise Linux 5 安装光盘添加到“添加/删除软件”安装源中，再进行安装。

MySQL 数据库可以在“软件包管理者”的“服务器”列表中找到，选中其复选框并单击“应用”按钮即可，如图 2-14 所示。

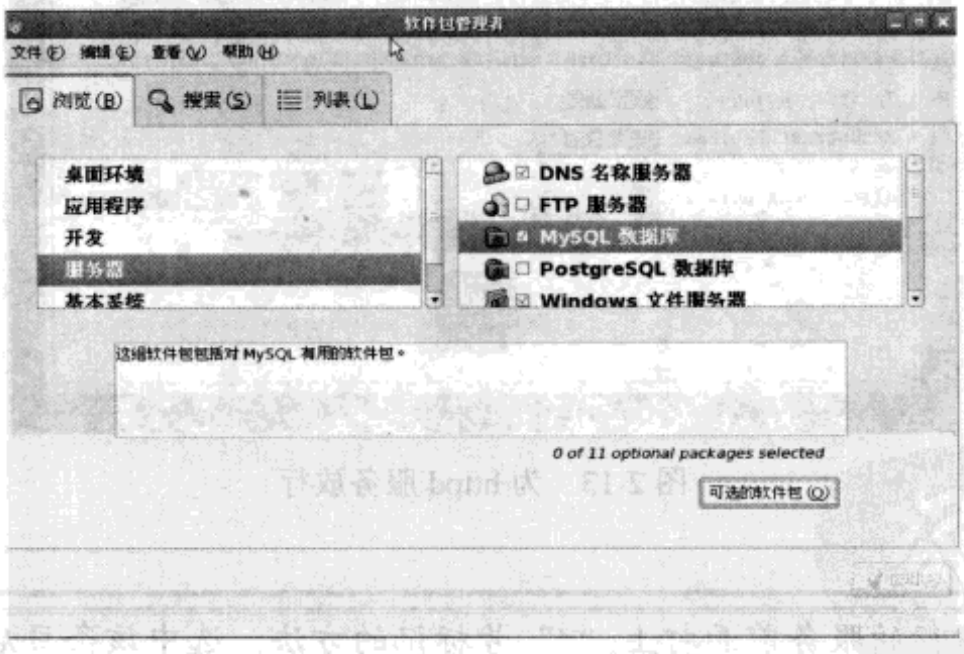


图 2-14　安装 MySQL 数据库

2.4.3 安装 PureFTPd

在 Red Hat Enterprise Linux 5 中，没有内置 PureFTPd 相应的套件，因此需要下载安装。具体步骤如下。

第 1 步，到相应网站下载 PureFTPd 的源代码包（http://download.pureftpd.org/pub/pure-ftpd/releases/），如图 2-15 所示。这里选择 pure-ftpd-1.0.21.tar.bz2。

Index of /pub/pure-ftpd/releases/

../		
obsolete/	20-Feb-2006 17:13	-
distinfo	30-Apr-2009 05:54	710
pure-ftpd-1.0.18.tar.bz2	02-Mar-2004 20:41	458995
pure-ftpd-1.0.18.tar.gz	02-Mar-2004 20:41	536611
pure-ftpd-1.0.19.tar.bz2	20-Jun-2004 17:40	460292
pure-ftpd-1.0.19.tar.gz	20-Jun-2004 17:40	538454
pure-ftpd-1.0.20.tar.bz2	20-Jul-2004 10:46	470915
pure-ftpd-1.0.20.tar.gz	20-Jul-2004 10:46	561412
pure-ftpd-1.0.21.tar.bz2	21-Feb-2006 13:15	476117
pure-ftpd-1.0.21.tar.gz	21-Feb-2006 13:15	594394

图 2-15　下载相应软件包

第 2 步，下载相应包到本地后，执行以下指令将压缩包解开。

```
[root@localhost ~]#tar jxvf pure-ftpd-1.0.21.tar.bz2
```

第 3 步，进入到解压后的目录。

```
[root@localhost ~]#cd pure-ftpd-1.0.21
```

第 4 步，执行指令生成编译配置文件。

```
[root@localhost pure-ftpd-1.0.21]#./configure \
--prefix=/usr/local/pureftpd \          ; #安装到/usr/local/pureftpd 目录下
--with-mysql \                          ; #支持 MySQL，需要 mysql-devel 组件支持
--with-shadow  \
--with-pam  \
--with-paranoidmsg  \
--with-welcomemsg  \
--with-uploadscript  \
--with-cookie  \
--with-virtualchroot  \
--with-virtualhosts  \
--with-virtualroot  \
--with-diraliases  \
--with-quotas  \
--with-puredb \
--with-sysquotas  \
--with-ratios  \
--with-ftpwho  \
--with-throttling  \
--with-language=simplified-chinese ; #登录提示信息为简体中文
```

注意　通过源代码编译安装软件，必须安装有开发工具包和开发库（通过选择“应用程序”→“添加/删除软件”→“开发”菜单命令），否则无法编译软件。若没有安装相关的工具和库文件，也可以到 http://www.rpmfind.net 网站搜索适合 Red Hat Enterprise Linux 5 的 PureFTPd 安装包。

第 5 步，执行以下指令安装 PureFTPd。

```
[root@localhost pure-ftpd-1.0.21]#make check
[root@localhost pure-ftpd-1.0.21]#make install
```

2.4.4 生成 PureFTPd 服务管理脚本

PureFTPd 服务的启动方式与其他服务的启动方式不同，如果要实现像其他服务器那样启动 PureFTPd 服务，则还需要进一步调整和配置，可以通过以下方法进行调整。

第 1 步，执行以下指令生成配置文件。

```
[root@localhost pure-ftpd-1.0.21]#cd configuration-file
[root@localhost configuration-file]#chmod u+x pure-config.pl
[root@localhost configuration-file]#cp pure-config.pl
/usr/local/pureftpd/bin/
[root@localhost configuration-file]#mkdir /usr/local/pureftpd/etc
[root@localhost configuration-file]#cp pure-ftpd.conf /
usr/local/pureftpd/etc
[root@localhost configuration-file]#cd ..
[root@localhost pure-ftpd-1.0.21]#cp pureftpd.* /usr/local/pureftpd/etc
```

第 2 步，执行以下操作生成 PureFTPd 服务。

```
[root@localhost pure-ftpd-1.0.21]#cd contrib
[root@localhost contrib.]#vi redhat.init
```

修改相应的命令行为如下内容。

```
fullpath=/usr/local/pureftpd/sbin/$prog
pureftpwho=/usr/local/pureftpd/sbin/pure-ftpwho
```

第 3 步，执行以下指令安装 PureFTPd 服务。

```
[root@localhost contrib]#cp redhat.init /etc/init.d/pure-ftpd
[root@localhost contrib]#chmod +x /etc/init.d/pure-ftpd
```

至此，支持 PureFTPd 服务的管理脚本 PureFTPd 创建成功。

2.4.5 配置匿名用户登录环境

在 PureFTPd 服务完全成功安装后，为了配置匿名用户登录环境就需要创建一个匿名 FTP 服务器，需要进行以下操作。

第 1 步，编辑/etc/pure-ftpd.conf 文件，保证有以下内容。

```
[root@localhost ~]#vi /etc/pure-ftpd.conf
ChrootEveryone          yes       ; #限制所有用户在其主目录中
AnonymousOnly           yes       ; #是否只允许匿名登录，设置为 no 不允许其他用户登录
NoAnonymous             no        ; #是否允许匿名连接，设置为 no 表示允许
AnonymousCanCreateDirs  yes ; #是否允许匿名用户创建新目录
```

```
MaxLoad  4  /                    ; #如果系统 loaded 超过下面的值匿名用户会被禁止下载
AnonymousRatio              1:10; #匿名用户的上传/下载的比率
AnonymousBandwidth          8    ; #匿名用户的最大带宽（Kb/s）
AllowAnonymousFXP   yes          ; #允许匿名用户进行 FXP 传输
AnonymousCantUpload     no   ; #是否允许匿名用户上传文件(no=允许)
```

第 2 步，在没有 FTP 账号的情况下，执行以下指令添加用户。

```
[root@localhost ~]#useradd ftp -g ftp -d /var/ftp -s /sbin/nologin
```

注意 PureFTPd 服务是以 FTP 账号运行的，所以需要先检查/etc/passwd 下是否有 FTP 账号。

第 3 步，PureFTPd 服务下，匿名用户的默认主目录是/vat/ftp，而该目录不需要手动创建，执行以下命令匿名用户主目录。

```
[root@localhost ~]#mkdir /var/ftp
[root@localhost ~]#chown ftp:ftp /var/ftp  ; 使 ftp 用户对匿名用户主目录有控制权限
```

到此为止，虚拟用户 PureFTPd 服务器创建完成。

注意 若显示“421 无法建立安全匿名 FTP”错误提示，应检查系统账号是否存在，或者匿名用户的主目录是否创建。

2.4.6 安装 Zend Optimizer

Zend Optimizer（简称 ZO）是由 Zend 技术公司开发的免费 PHP 优化软件，用以提高 PHP 应用程序的执行速度，同时可以显著降低服务器的 CPU 负载并能提高服务器的响应速度。用户可以更快地浏览网页，从而提高办事效率，创造更好的业绩。

安装 ZO 的具体步骤如下。

第 1 步，到相应网站下载 Zend Optimizer（http://www.zend.com/en/products/guard/optimizer/），这里使用的是 ZendOptimizer-3.3.3-linux-glibc23-i386.tar.gz 版本。

第 2 步，下载完成后，执行以下指令解压缩 Zend Optimizer 压缩包。

```
[root@localhost Desktop]#tar zxvf ZendOptimizer-3.3.3-linux-glibc23-i386.tar.gz
```

第 3 步，执行以下指令打开 Zend Optimizer 的安装窗口，如图 2-16 所示。然后根据系统提示操作即可。

```
[root@localhost Desktop]#cd ZendOptimizer-3.3.3-linux-glibc23-i386
[root@localhost ZendOptimizer-3.3.3-linux-glibc23-i386]#./install.sh
```

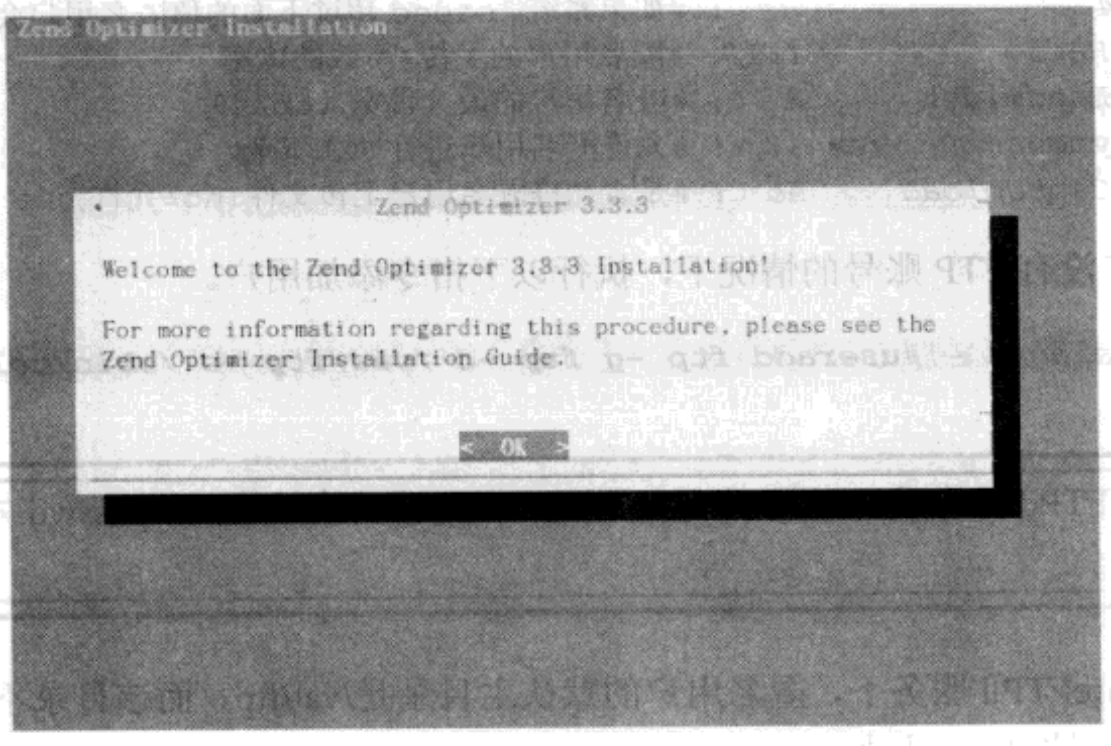


图 2-16　Zend Optimizer 安装中

2.4.7 WebPureFTP 的安装与配置

WebPureFTP 是一款 Web 界面的 PureFTPd 用户管理程序，使用该软件可以非常简便地为 PureFTPd 完成针对用户的限制设置。

1. 安装 WebPureFTP

要安装该软件首先需要到相关网站获取相应的软件包（http://samson_2008.814e.com/default.html），如图 2-17 所示。这里使用的是 0.1 版本即 webpureftp0.1.tar.gz。

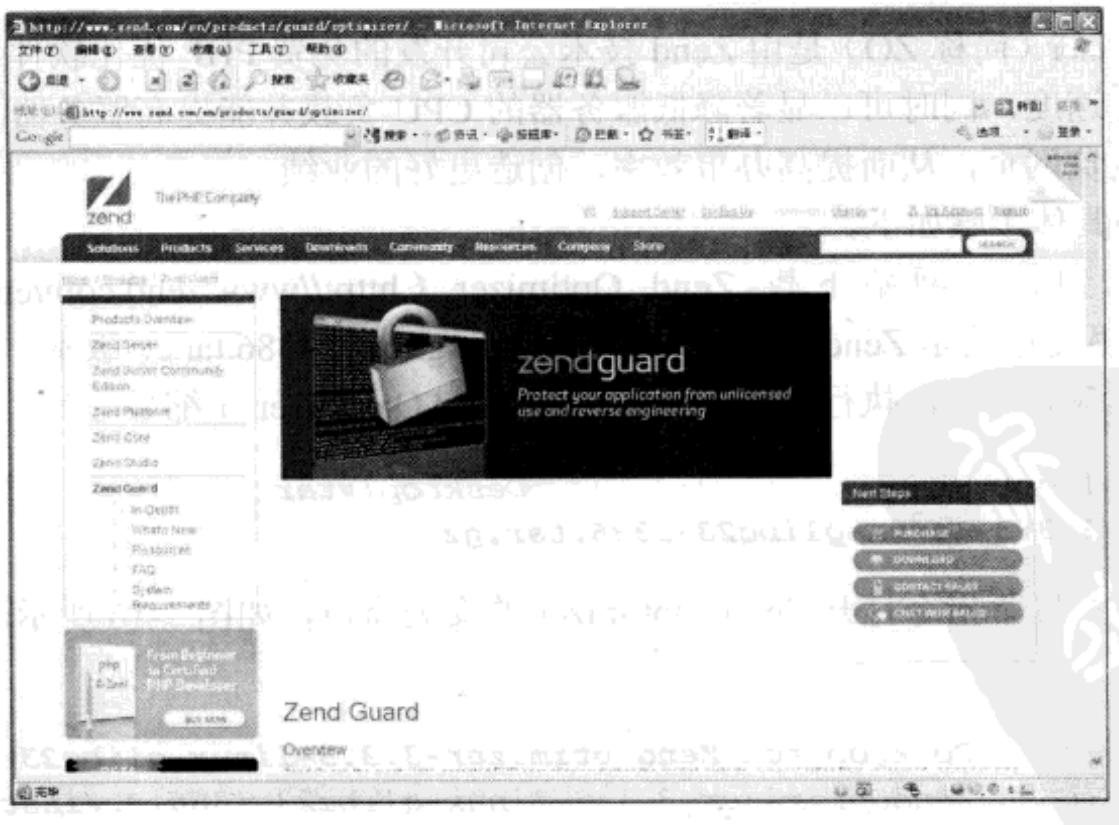


图 2-17　下载 WebPureFTP 安装包

下载到本地桌面后，执行以下指令解压。

```
[root@localhost Desktop]# tar zxvf webpureftp0.1.tar.gz
```

然后使用下列指令将解压出来的 webpureftp 移动到/var/www/html 下并重命名。

```
[root@localhost Desktop]# mv webpureftp0.1 /var/www/html/webpureftp
```

2. 编辑配置文件

用 vi 编辑器打开/var/www/html/webpureftp/config 下的 config.inc.php，然后修改其中的数据库、用户名和密码。

```
[root@localhost ~]# vi /var/www/html/webpureftp/config//config.inc.php
<?php
 $orig_title = "Web PureFTP 管理系统";
 $pagesize=20;
 $topsize[0]=8;
 $topsize[1]=10;
 $topsize[2]=15;
 $topsize[3]=20;
 $onmousecolor="E6F0E6";
 $default_ftp_root="/ftp";
#设置 FTP 的实际目录必须在目录后加上“/”符号，否则 pure-ftpd 会为每个 FTP 用户自动在/
#根目录下建立名为“ftproot 用户名”的主目录。
 $obj_db=new db("localhost","root","mypasswd","pureftp");
#修改用户名和密码
?>
```

2.4.8 生成 MySQL 数据库

在前面编译 PureFTPd 服务的时候已经加入了 MySQL 支持，可以用数据库来保存相关信息（包括保存登录用户、密码、上传带宽限制、下载带宽限制等），但还需要创建数据库、生成配置文件才能实现这些功能，具体步骤如下。

第 1 步，执行以下指令修改 MySQL 密码。

```
[root@localhost ~]#mysqladmin -u root -p password 新密码
Enter password:                          : #输入旧密码，不显示
```

注意 由于 Red Hat Enterprise Linux 5 内置的 MySQL 服务默认密码为空，所以首先要执行以上指令设置一个密码。

第 2 步，执行以下指令为 PureFTPd 服务器创建一个数据库。

```
[root@localhost ~]#mysql –u root –p
Enter Password:
```

```
mysql> Create DataBase pureftpd;
Query OK. 1 row affected (0.00 sec)
```

第 3 步，打开 webpureftp 解压缩目录下 SQL 目录下的 pureftp_0.1.sql 文件。在

```
DROP TABLE IF EXISTS `depart_info;
```

语句前添加如下指令。

```
use pureftp;
```

然后执行如下命令。

```
Mysql>mysql –u root <SQL/pureftp_0.1.sql
```

至此，生成 MySQL 数据库的具体配置已完成。

2.4.9 生成 pure-ftpd 数据库配置文件

PureFTPd 服务默认不是使用 MySQL 来进行用户的配置和管理的，如果需要使用 MySQL 进行用户的配置和管理则需要手动指定，具体步骤如下。

第 1 步，在前文解压的 pure-ftpd 目录下（pure-ftpd-1.0.2），可以看到一个名为 pureftpd-mysql.conf 的文件，然后使用编辑器打开该文件。

```
[root@localhost pure-ftpd-1.0.2]#vi pureftpd-mysql.conf
```

修改 PureFTPd 的数据库名、用户名和密码。

```
MYSQLUser       root
MYSQLPassword   mypasswd
MYSQLDatabase   pureftp
```

第 2 步，将该文件重命名为 pureftpd.conf 并复制到/etc 目录下，覆盖原来的配置文件。

```
[root@localhost pure-ftpd-1.0.2]# mv pureftpd-mysql.conf pure-ftpd.conf
[root@localhost pure-ftpd-1.0.2]# cp pure-ftpd.conf /etc
```

通过以上设置，就可以通过 Web 对 PureFTPd 服务器进行管理了。

2.4.10 使用 WebPureFTP 管理虚拟用户

为了检验以上配置是否成功，则可以通过以下操作进行检测。

首先打开浏览器，然后在地址栏中输入“http://ServerIP/webpureftp”并按下 Enter 键，即可显示如图 2-18 所示的 WebPureFTP 登录对话框。

成功登录后，即可使用 WebPureFTP 方便地管理虚拟用户了。

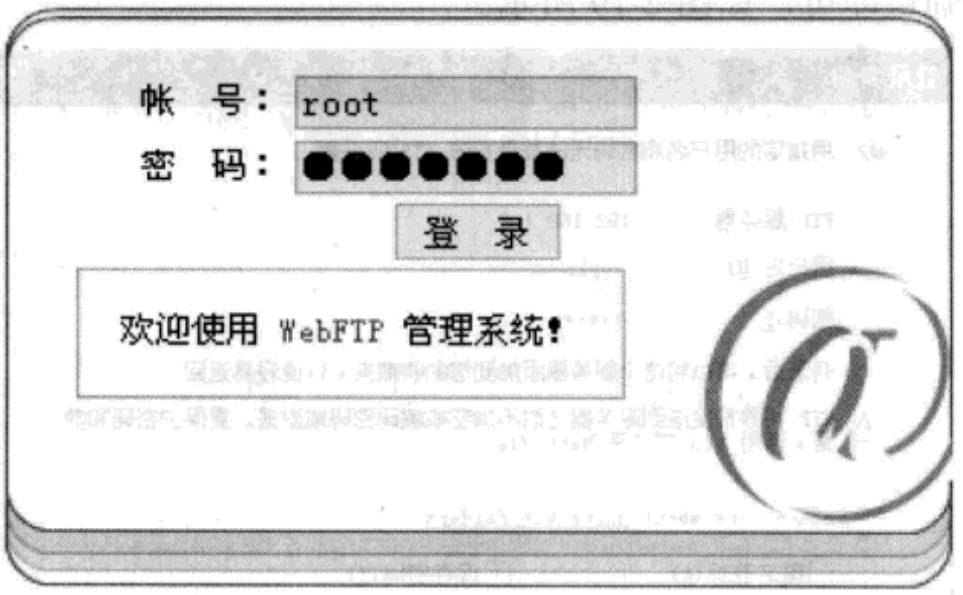


图 2-18 WebPureFTP 登录对话框

2.5 FTP 服务器的访问

FTP 客户端的软件种类非常丰富，且随客户端的操作系统不同（主要是 Windows 操作系统和 Linux 操作系统）而不相同，因此在本节将具体介绍在不同操作系统环境中如何用命令行的方式来访问 FTP 服务器。

2.5.1 Linux 环境下访问 FTP 服务器

在 Linux 环境下，可以通过命令行或浏览器来实现访问 FTP 服务器。

1. 通过命令行方式访问

Red Hat Enterprise Linux 5 下内置了一款功能强大、操作简单的 FTP 客户端软件。在 Red Hat Enterprise Linux 5 的命令行窗口，用户采用刚才创建的 ftpvu2（密码为 ftpvu2pass），输入以下指令登录 vsFTPd 服务器，如下所示。

```
[root@localhost ~]#lftp -u ftpvu1 192.168.1.6
Password:
lftp ftpvu2@192.168.1.6:~>
```

紧接着执行 ls 命令进行查看。

```
lftp ftpvu2~192.168.1.6:~>ls -la
drwxr-xr-x  27    4096 07-18 17:42 .
drwxr-xr-x  27    4096 07-18 17:42 ..
```

2. 使用浏览器访问

除了使用命令行方式访问 FTP 服务器以外，还可以通过使用浏览器和特有的 FTP 客户端工具来访问 FTP 服务器。如果是匿名用户访问，只需直接在地址栏输入“ftp://IP 地址格式的地址”即可；非匿名用户访问则需要用户名和密码，输入“ftp://username:password@IP 地址格

式”，按下 Enter 键确认即可，如图 2-19 所示。

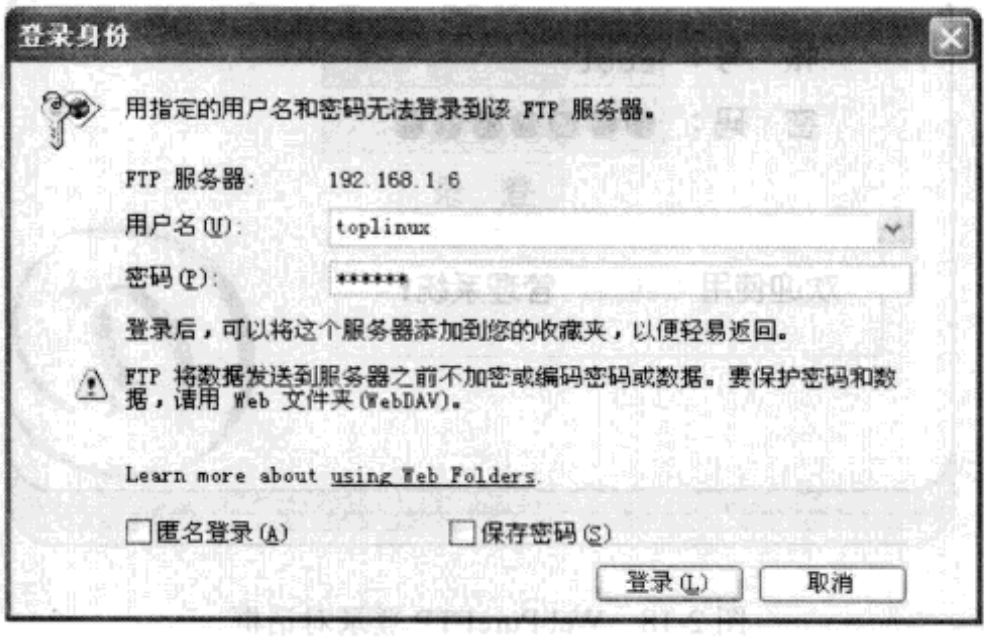


图 2-19　使用浏览器访问

2.5.2 Windows 环境下访问 FTP 服务器

前面已经讲到 Linux 环境下访问 FTP 服务器可以通过命令行方式和浏览器方式两种方法进行访问。在 Windows 环境下访问 vsFTPd 服务器也不例外，同样有 DOS 命令方式和浏览器方式两种方法，这里以 DOS 命令方式访问 FTP 服务为例进行讲解。Windows 内置的 FTP 命令行工具简单易用，具体操作如下。

在 Windows 的命令行窗口，输入以下指令登录 vsFTPd 服务器（IP 地址为 192.168.1.6），当要求输入用户名时输入匿名用户（Anonymous），登录成功。

```
C:\>ftp 192.168.1.6
C:\Documents and Settings \Administrator > ftp 192.168.1.6
Connected to 192.168.1.6.
220 <vsFTPd 2.0.5>
User <192.168.1.6:<none>>: anonymous
331 Please specify the password.
Password:
230 Login successful.
ftp> ls
```

注意　有时候为了直观地上传和下载文件，也使用专用的 FTP 客户端工具。

第 3 章　电子邮件服务器

E-mail 是 Internet 上最常用也是广受好评的服务之一，几乎所有的 Internet 用户都至少有一个 E-mail 账号。E-mail 利用 Internet 发送和接收邮件，与传统的邮政信件服务类似，但是具有更加快捷、经济的特点，E-mail 已经成为当今网络时代最重要的通信工具之一。从目前的情况来看，大多数作为网络服务器的 Linux/UNIX 系统都提供 E-mail 服务。

3.1　电子邮件服务概述

E-mail（Electronic mail，电子邮件）主要用于在 Internet 上进行信息的传递和交流。E-mail 最吸引人的特点之一就是费用低廉甚至免费，使用者几乎不需付出任何代价就可以与其他人进行通信，目前已经成为人们生活、工作中最重要的交流方式之一。

3.1.1　电子邮件服务简介

要在 Linux 操作系统中配置电子邮件服务，最好是先来了解一下其原理和组成部分。一般情况下，一个完整的电子邮件系统包括以下 3 个部分。

1.　邮件用户代理程序

MUA（Mailer User Agent，邮件用户代理程序），主要用于帮助用户发送和收取电子邮件。目前常用的 Outlook、Outlook Express、Foxmail、Thunderbird、Eudora 等都属于 MUA。

2.　邮件传送代理程序

MTA（Mail Transfer Agent，邮件传送代理程序）即通常所说的邮件服务器，用来监控以及传送电子邮件。有许多著名的产品，如 Windows 下的 Exchange、Imail Server 和 Mdaemon 等，Linux/UNIX 下的 Sendmail、Qmail、Postfix 和 Exim 等。

3.　电子邮件协议

由前边两个部分可以知道，电子邮件客户端和服务器端的种类繁多。那么这些软件是怎样进行通信的呢？这就要通过电子邮件协议来完成了。常用的电子邮件协议包括以下几种。

（1）SMTP

SMTP 的全称是 Simple Mail Transfer Protocol，即简单 Mail 传输协议，为用户提供高效、可靠的邮件传输，监听 25 号端口，是一个请求/响应协议，用于接收用户的 Mail 请求，并与远程 Mail 服务器建立起 SMTP 连接。SMTP 有一个重要特点，即能够接力传送邮件，让邮件通过不同网络上的主机接力式地传送。其工作分为两种情况：一是电子邮件从客户机传送到服务器；二是电子邮件从一个服务器传送到另一个服务器。

（2） POP3

POP 的全称是 Post Office Protocol，即邮局协议，使用 TCP 的 110 端口，用于电子邮件的接收，目前常用的是 POP3。

POP3 采用客户端/服务器工作模式：首先，客户端软件（Outlook 或 Foxmail）与 POP3 服务器建立起 TCP 连接，然后， POP3 协议会确认客户机提供的用户名和密码，确认成功后便转入处理状态，这时用户可接收或者删除自己的邮件，之后退出系统转入更新状态，从服务器端删除用户标记为“删除”的邮件。

（3） IMAP4

IMAP 的全称是 Internet Message Access Protocol，是通过 Internet 获取信息的一种协议。同样可以像 POP 那样提供方便的下载邮件服务，实现离线阅读功能。IMAP 还提供了一种摘要浏览功能，让用户可以在阅读完邮件的主题、发件人、大小和到达时间等信息后再决定是否下载。

IMAP 本身是一种邮箱访问协议，可以用来管理在客户端服务器上的邮箱。POP3 是通过将邮件下载到本地以供用户查看的，而 IMAP 的邮件却是保留在服务器上的。IMAP 这一特点是和 Webmail 相似的，但是与 Webmail 相比，又具有更高效、更安全及可以离线阅读等优点。

（4） Web Mail

在这么多电子邮件协议中，最热门的就是 Web Mail 了。目前，像 Yahoo Mail、Hotmail 和 Gmail 等使用的都是 Web Mail 这种邮件管理方式。但实质上，Web Mail 并不是一种协议，而是一种 Web 支持插件。插件专门针对安装在服务器上的邮件程序，实现了直接通过浏览器查收、阅读和发送邮件。

3.1.2 邮件专有名词解释

在配置电子邮件的过程中，经常会涉及到一些专有的概念和名词，只有先弄清楚这些名词的含义，在具体配置的过程中才不容易出错。另外，要想弄明白邮件相关的协议及发送原理，也需要先弄明白这些专有名词的含义。

1. MUA

MUA（Mail User Agent，邮件用户代理人）主要用于将用户的邮件发送到邮件主机上或者将用户的邮件从邮件主机上接收下来。像微软的 Outlook Express、腾讯的 Foxmail、Mozilla 的 Thunderbird（雷鸟）等，都属于 MUA。

2. MTA

MUA 用于将用户的邮件发送到邮件主机上，如果邮件主机能将这封信邮寄出去，那它就是一台 MTA（Mail Transfer Agent，邮传传输代理）。现在的 MTA 一般就是指邮件服务器，其代表有 Sendmail、Postfix、qmail 以及 Windows 下的 Exchange 等。

从严格意义上来说，MTA 只是具备 SMTP 协议的主机，但是实际上现在的 MTA 基本上都包括了邮件发送、接收、转递等方面的功能。

（1） 接收邮件

MTA 使用 SMTP 协议，接收来自客户端的邮件，工作于端口 25。

（2） 转发邮件

如果邮件用户不在本地，MTA 就需要将这封邮件转发出去，直到到达最终地点。这就是 MTA 的转发（Relay）功能。

（3） 响应用户守信要求

MTA 使用 POP3 或者 IMAP 协议，让客户端通过 MUA 将邮件接收下来。

3. MDA

MDA（Mail Delivery Agent，邮件转发代理）分析 MTA 处理的邮件中的表头或者其他数据，以决定邮件的去向。

3.1.3 POP3 与 SMTP 协议

要实现电子邮件服务，需要涉及到一些重要的 TCP/IP 协议，比如 SMTP，POP3，IMAP 等。本节将介绍其中的 SMTP 和 POP3 协议。

1. SMTP

SMTP（Simple Mail Transfer Protocol，简单文件传输协议）用于从发送方的邮件服务器发送邮件到接收方的邮件服务器，是 Internet 电子邮件应用的核心。Sendmail 程序为 Linux 提供了支持 SMTP 连接所需的服务。

如下是一个使用 SMTP 协议的例子：位于 somedomain.gov 的用户 lu 正在向位于 otherdomain.com 的 dai 发送邮件。

```
$Sendmail -v dai@otherdomain.com<letter
dai@otherdomain.com...Connecting to otherdomain.com via top…
Trying 123.45.67.1…connected
220-otherdomain.com…SMTP ready at Sun, 14 Nov 2007 18:56:22-0500
220 ESMTP spoken here
>>>HELO somedomain.gov
250 otherdomain.com Hello somedomain.gov[123.45.67.2], please to meet you
>>>MAIL From: lu@somedomain.gov
250 <lu@somedomain.gov...Sender ok
>>>RCPT To: dai@otherdomain.com
250 <dai@otherdomain.com> … Recipient ok
>>>DATA
354 Enter mail,  end with "."on a line by itself
>>>.
250 SAA0680 Message accepted for delivery
>>>QUIT
221 otherdomain.com closing connection
dai@otherdomain.com... Sent
$
```

上边第一行直接调用了 Sendmail，-v 命令表示冗余模式显示 SMTP 对话。其余行表示 SMTP

客户机与服务器连接的过程。

2. POP3

POP3（Post Office Protocol 3）是 TCP/IP 协议簇中的一员，规定了如何将个人计算机连接到 Internet 的邮件服务器，以及如何下载电子邮件的电子协议。POP3 协议使用 110 端口，由 RFC 1939 定义，其有 3 个工作阶段：在特许阶段，用户代理以明文形式发送用户名和口令以鉴别用户；在事务处理阶段，用户代理取回报文并可以对邮件进行操作，如增加或取消报文删除标记；在更新阶段，即客户机发出了 quit 命令后，结束该 POP3 会话并删除那些被标记删除的报文。

3.2 Postfix 邮件服务器

大多数的 Linux 发行套件中默认的电子邮件服务都是 Sendmail。而 Postfix 的设计目标是提供一个与 Sendmail 的配置兼容，但更安全的邮件服务器。Postfix 邮件服务器有快速、稳定、配置简单但功能强大等特点，其使用也更加地方便，下面就来配置 Postfix 邮件服务器。

3.2.1 安装 Postfix 邮件服务器

Postfix 和 Sendmail 都是 Red Hat Enterprise Linux 5 中内置的邮件服务，Sendmail 在默认配置下已经启用。这里学习关闭系统中的 Sendmail 服务，安装启用 Postfix 服务。

1. 关闭 Sendmail 服务

Sendmail 和 Postfix 都是邮件服务系统，两者同时启动会造成系统冲突。所以，开始在 Red Hat Enterprise Linux 5 中安装 Postfix 之前，最好先关闭系统默认安装运行的 Sendmail。

第 1 步，查询系统是否已经安装启动了 Sendmail 服务。

在 Red Hat Enterprise Linux 5 下，通过 netstat 指令来查询 Sendmail 服务的情况。

```
[root@localhost ~]# netstat -nutlp |grep :25
[root@localhost ~]# netstat -nutlp |grep :25
tcp        0      0 0.0.0.0:25                  0.0.0.0:*                   LISTEN
5184/sendmail: acce
[root@localhost ~]#
```

命令执行结果显示“sendmail: acce”字样，表示 Sendmail 已经启动。

第 2 步，关闭 Sendmail 服务。

关闭 Sendmail 服务，并将其配置为不会随系统启动而自动运行，直接执行以下指令。

```
[root@localhost ~]# service sendmail stop
 [root@localhost ~]# service sendmail stop
关闭 sm-client:                                            [确定]
关闭 sendmail:                                             [确定]
[root@localhost ~]# chkconfig sendmail off
```

注意 其实，在 Red Hat Enterprise Linux 5 中，还有更加方便的 E-mail 服务器切换方法，即使用图形界面的切换工具——switch。但该工具并不是一个默认安装了的软件，首先需要安装该软件，在安装光盘的 Server 目录下，执行以下指令安装。

```
[root@localhost ~]#rpm -ivh system-switch-mail*.rpm
warning: system-switch-mail-0.5.25-12.noarch.rpm: Header V3 DSA signature:
NOKEY, key ID 37017186
Preparing...                ########################################### [100%]
   1:system-switch-mail     ###########################################
[ 50%]
   2:system-switch-mail-gnom########################################### 
[100%]
[root@localhost Server]#
```

安装完成以后，依次单击“系统”→“管理”→“邮件传输代理（MTA 切换器）”菜单命令即可显示图形化界面的 switch 了，如图 3-1 所示。

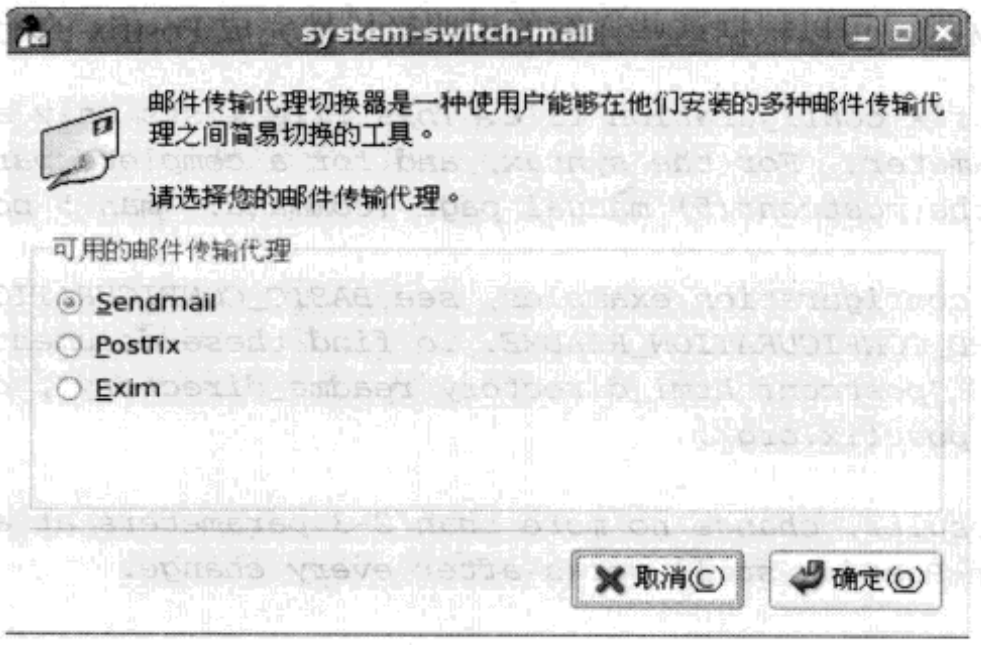


图 3-1 邮件传输代理切换器

2. 安装 Postfix 服务

禁用 Red Hat Enterprise Linux 5 默认安装的 Sendmail 之后，下面进行 Postfix 的安装、启用操作。

第 1 步，查询是否安装了 Postfix 服务。

虽然默认情况下，系统中 Postfix 服务并不会自动安装，但是首先还是应该确认本机是否已经安装这项服务，执行以下指令。

```
[root@localhost ~]#rpm -qa |postfix
```

如果显示有“postfix-2.3.3-2”等字样，则表明 Postfix 服务已经安装。

第 2 步，安装 Postfix 服务。

如果系统尚未安装 Postfix 服务，则可以从 Red Hat Enterprise Linux 5 安装光盘的 Server

目录下找到 Postfix 服务的 RPM 安装包，并执行以下指令安装。

```
[root@localhost ~]#rpm -ivh postfix-2.3.3-2.i386.rpm
warning: postfix-2.3.3-2.i386.rpm: Header V3 DSA signature: NOKEY, key ID
37017186
Preparing...                ########################################### [100%]
   1:postfix                ########################################### [100%]
[root@localhost Server]#
```

3.2.2 Postfix 配置文件

Postfix 安装完成后，还需要进行一些特定的设置，才能实现为特定的域收发邮件的功能。Postfix 的配置文件位于/etc/postfix 目录，主要包括 main.cf、master.cf、access 及 aliases 几个部分。

1. /etc/postfix/main. cf

Postfix 的主配置文件，几乎所有的 Postfix 参数都可以在这里进行设置。文件中以“#”开头的是行为说明文档，可以根据这些详细的说明轻松地完成 Postfix 的配置操作。

```
# Global Postfix configuration file. This file lists only a subset
# of all parameters. For the syntax, and for a complete parameter
# list, see the postconf(5) manual page (command: "man 5 postconf").
#
# For common configuration examples, see BASIC_CONFIGURATION_README
# and STANDARD_CONFIGURATION_README. To find these documents, use
# the command "postconf html_directory readme_directory", or go to
# http://www.postfix.org/.
#
# For best results, change no more than 2-3 parameters at a time,
# and test if Postfix still works after every change.

# SOFT BOUNCE
#
# The soft_bounce parameter provides a limited safety net for
# testing.  When soft_bounce is enabled, mail will remain queued that
# would otherwise bounce. This parameter disables locally-generated
# bounces, and prevents the SMTP server from rejecting mail permanently
# (by changing 5xx replies into 4xx replies). However, soft_bounce
# is no cure for address rewriting mistakes or mail routing mistakes.
……
```

2. /etc/postfix/master. cf

主要用来规定 Postfix 运行时的参数。默认情况下，该文件已经配置完成，不需要进行修改。

3. /etc/postfix/access

Postfix 的存取控制文件，功能与 Sendmail 的配置文件/etc/sendmail/access 相同，主要用来

设置开放传递，拒绝连机的来源或者 IP 地址等信息。/etc/sendmail/access 文件需要先在 main.cf 中启用，并使用 postmap 将其处理为数据库，才能生效。

4. /etc/aliases

定义 Postfix 别名的数据库，用法与 Sendmail 完全一样。

3.2.3 Postfix 基本配置

Postfix 具有可读性很高的配置文件，这也是它的显著优点之一。/etc/postfix/main.cf 文件的内容比较多，但其中大部分内容都是说明性文字，需要定义的参数并不是很多，即使按照默认值也可以正常运行。

注意 在 Red Hat Enterprise Linux 5 系统中，Sendmail 默认只监听本地 127.0.0.1 端口的邮件收发，Postfix 也是如此。要使其成为邮件服务器，必须使其能够监听其他客户端的连接请求才能实现。

1. 语法规则

在 Postfix 的主配置文件 main.cf 中，用来控制 Postfix 运行行为的参数通常都是以类似变量的形式存在的。比如，使用如下语句设置 Postfix 主机名称。

```
myhostname = toplinux.localdomain.tst
```

等号左边是变量名称，等号右边是变量的值。等号右边还可以通过在变量名前面加入一个“$”符号来表示引用该变量。

```
myorigin = $myhostname
```

实际上等同于如下指令。

```
myorigin = toplinux.localdomain.tst
```

注意 Postfix 变量设置的等号两边需要分别加入空格字符。如果某变量有两个以上的设置值，必须用逗号或者空格符隔开。

```
mydestination = $mydomain, $myhostname, localhost.$mydomain
```

2. 打开 Postfix 的网络发送邮件支持

默认情况下，“inet_interfaces”参数的值为“localhost”，也就是说 Postfix 只监听本机的发信需求，只能在本地邮件主机上寄信。如果要打开 Postfix 的网络发送邮件支持，通常的做法是开放所有的网络接口，即设置“inet_interfaces”参数的值为“all”。具体操作为：在“inet_interfaces = localhost”前边的加一个“#”符号，去掉“inet_interfaces = all”前的“#”，

如下所示。

```
inet_interfaces = all                    ; #删除了句首的注释“#”
#inet_interfaces = $myhostname
#inet_interfaces = $myhostname, localhost
#inet_interfaces = localhost    ; #在句首增加了一个注释“#”
```

注意 如果邮件主机上有多个网络接口，但是又不希望全部的网络接口都开放 Postfix 服务，可以使用主机名指定需要开放的接口。

3. 设置运行 Postfix 服务的邮件主机名和域名

“myhostname”参数指定运行 Postfix 服务的邮件主机的主机名称(FQDN 名)。“mydomain”参数指定了该主机的域名称。如果不需要在一台计算机上运行多个主机，这两个参数一般不需要修改。默认情况下，参数“myhostname”的值会被 Postfix 自动删除第一部分，并将其余部分作为“mydomain”参数的值。

```
myhostname = toplinux.localdomain.tst
mydomain = gdvcp. localdomain.tst
```

4. 设置由本机寄出的邮件所使用的域名或主机名称

当前邮件主机寄出的所有邮件的邮件头“Mail from”的地址都是由“myorigin”参数设置的。一封由本地邮件主机寄出的邮件开头部分包含有如“From：‘toplinux’<toplinux@toplinux.localdomain.tst>”这样的内容，表明这封邮件的发送主机是 toplinux.localdomain.tst，原因是 Postfix 默认使用本地主机名作为“$myorigin”。不过按照用户平时的使用习惯，也可以将 myorigin 参数设置为本地邮件主机的域名（即“myorigin = localdomain.tst 或 $mydomain”），邮件开头部分就会包含有如“From：‘lbt’<lbt@localdomain.tst>”的信息。

```
myorigin = $mydomain
```

5. 设置可转发邮件网络

默认配置下，Postfix 只允许转发本地网络的邮件。但是可以使用 mynetworks 或者 mynetworks-style 参数授权其他的网络。

（1） mynetworks

参数值可以是信任的 IP 地址、IP 子网或多个 IP 子网（相互之间用逗号或者空格隔开）。例如，将 mynetworks 参数值设置为 192.168.1.0/26，表示邮件主机只转发子网 192.168.1.0/26 中的客户端发来的邮件，而拒绝其他子网的转发请求。

（2） mynetworks-style

mynetworks-style 也是一个用于控制网络邮件转发的参数，可选择项包含以下 3 种，主要用来设置可转发邮件网络的方式。

- class

Postfix 会根据邮件主机的 IP 地址知道其网络类型（A 类、B 类或者 C 类），并开放相应网段。例如，邮件服务器的 IP 地址为 192.168.1.6，这是一个 C 类网络的 IP 地址，则 Postfix 会自动开放 192.168.1.0/26 整个 C 类网络的转发授权。

- subnet

Postfix 会根据邮件服务器网络接口上所设置的 IP 地址、子网掩码来决定开放的 IP 网段，这也是 Postfix 的默认设置。例如，邮件服务器的 IP 地址为 192.168.1.6，子网掩码为 255.255.255.128，则 Postfix 会开放 192.168.1.0/25 子网。

- host

Postfix 只开放本机的转发权限。

注意 通常，用户不需要去管 mynetworks-style 参数的内容，直接设置 mynetworks 参数即可。如果这两个参数同时进行了设置，也是以 mynetworks 参数的设置为准。

6. 设置允许接收的邮件

发到服务器上的邮件，Postfix 并不是都接收的。mydestination 参数决定了哪些邮件可以接收，只有当发来邮件的收件人地址与该参数值相匹配时，该邮件才会被 Postfix 接收下来。

如果将该参数值设置为 $mydomain 和 $myhostname，来信的收件人地址是 UserName@localdomain.tst（UserName 即用户的邮件账户名）和 xxx@toplinux.localdomain.tst，Postfix 都会接收这些邮件。

```
mydestination = $mydomain, $myhostname
```

7. 设置可转发邮件的网域

由前面的学习我们已经知道，允许转发邮件的来源 IP 地址可以用 mynetworks 参数来定义。在 Postfix 中还可以为域名授权，使用 relay_domains 参数。

如果将该参数值设置为 localdomain.tst，则表示信任所有由 localdomain.tst 域发来的邮件，并自动进行转发。

```
relay_domains = localdomain.tst
```

完成这几个部分的基本设置后，重新启动 Postfix 服务。但是该服务器目前仅支持客户端发信，暂时还不支持收信。

8. DNS 设置

另外，还必须进行 DNS 设置以便 Postfix 能在局域网中更好地转发邮件。

在内部网络的 DNS 服务器上定义一个主区域 localdomain.tst，并在该区域配置文件中定义以下记录（除定义了 SOA、NS 记录外）。

```
Toplinux.localdomain.tst.  IN   A  192.168.1.6
mail.localdomain.tst.  IN CNAME  toplinux.localdomain.tst.
localdomain.tst.       IN MX  10  mail.localdomain.tst.
```

注意 如果要查看 Postfix 的主配置文件是否正确，可以使用 postconf 指令来查看详细的数据。指令执行后显示的数据量是很庞大的，也可以执行如下指令,只显示与内置默认参数不同的那些参数。

```
[root@localhost ~]#postconf -n
```

3.2.4 虚拟别名域

利用虚拟别名域能够实现群组邮递的功能，将发送给虚拟域的邮件实际投到真实域的用户邮箱中。指定一个虚拟邮件地址，当用户给这个邮件地址发送邮件的时候，邮件服务器会自动将邮件转发到真实域中的一组用户的邮箱里。虚拟域可能是不存在的域，真实域可能是本地域（main.cf 文档中的 mydestination 参数指定的域）、远程域甚至是 internet 域。实际上，虚拟域是真实域的一个别名，通过一个虚拟别名表，实现了虚拟域的邮件地址到真实域的邮件地址的重定向。

第 1 步，打开 Postfix 的主配置文件“/etc/postfix/main.cf”，定义以下语句，配置虚拟别名域。

```
virtual_alias_domains = csoftz.cn, localdomain.tst
virtual_alias_maps = hash:/etc/postfix/virtual_
```

第 2 步，编辑配置文件/etc/postfix/virtual_domains，定义如下语句。

```
@csoftz.cn                  @localdomain.tst
sales@localdomain.tst       toplinux
```

第 1 行表示投递到 csoftz.cn 域的邮件实际上会被发送到 localdomain.tst。

第 2 行表示发送到 sales@localdomain.tst 的邮件，实际上会转送到本地用户 toplinux。

第 3 步，在配置文件 main.cf 和 virtual 修改完成以后，要使更改立即生效，应分别执行以下命令。

```
[root@localhost ~]#postmap /etc/postfix/virtual_domains
[root@localhost ~]#postfix reload
```

3.2.5 用户别名

与虚拟别名域类似 Postfix 还支持用户别名。用户别名与虚拟别名不同的是，它是通过别名表（aliases）在系统范围内实现别名邮件地址到真实用户邮件地址的重定向。使用用户别名最重要的功能是实现群组邮递，通过用户别名可以将发送给某个别名邮件地址的邮件转发到多

个真实用户的邮箱中。

第 1 步，打开 Postfix 主配置文件/etc/postfix/main.cf，确认文件中包含以下语句。

```
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
```

第 2 步，编辑配置文件/etc/postfix/virtual_aliases，定义如下语句。

```
team1:      user1, user2, user3, user4
team2:      :include: /etc/mail/team2user
toplinux:       lu
newuser:        lu, test@localdomail.test
```

第 1 行，说明发送给 team1 的邮件，都会自动转发给用户 user1、user2、user3 及 user4。

第 2 行，说明发送给 team2 的邮件，都会自动转发给/etc/mail/ team2user 文件中指定的用户。

第 3 行，说明 toplinux 是 lu 的别名。

第 4 行，说明 newuser 是 lu 和 test@localdomain.tst 的别名。

其中，team2user 文件的内容和格式如下。

```
user21,\
user22,\
user23,\
user24
```

行后面加斜杠只是实现分行，其格式也可以改为和 team1 一样。

```
user21, user22, user23, user24
```

第 3 步，在配置文件 main.cf 和 virtual_aliases 修改完成之后，分别执行以下的命令，使更改立即生效。

```
[root@localhost ~]#postalias /etc/postfix/virtual_aliases
[root@localhost ~]#postfix reload
```

3.2.6 SMTP 认证

根据前面的设置，位于同一网段 192.168.1.0/26 下的机器，可以通过 Postfix 发送邮件，但默认其他用户都被拒绝。要是所有的网段都打开，又容易造成服务器的负荷过重，所以比较理想的做法是给 Postfix 服务加上用户认证，只允许通过了认证的用户发送邮件。

Cyrus SASL（Cyrus Simple Authentication and Security Layer，赛勒斯简单认证和安全层）是为应用程序提供认证的函数库，而应用程序通过函数库所提供的功能定义认证方式，让 SASL 通过与邮件服务器主机的沟通来实现认证的功能。

1. 安装 CyrusSASL 认证包

默认情况下，Red Hat Enterprise Linux 已经安装了 CyrusSASL 认证包，检查系统是否已经安装了 CyrusSASL 认证包或安装了何种版本，可使用如下语句。

```
[root@localhost ~]#rpm -qa | grep sasl
```

安装 CyrusSASL 认证包的方法是：在 Red Hat Enterprise Linux 5 安装光盘的 Server 目录下找到与 CyrusSASL 相关的 RPM 包文件，再使用 rpm -ivh 命令安装。

2. CyrusSASL V2 的密码验证机制

默认情况下，CyrusSASL V2 版使用守护进程 saslauthd 进行密码认证。而可用的密码认证方法有多种，查看当前系统中的 CyrusSASL V2 所支持的密码验证机制，可以使用如下命令。

```
[root@localhost ~]#saslauthd -v
saslauthd 2.1.22
authentication mechanisms: getpwent kerberos5 pam rimap shadow ldap

[root@localhost ~]#
```

由命令执行结果可知，当前可用的密码验证方法有 getpwent、kerberos5、pam、rimap、shadow 及 ldap。其中最简单的验证方法是 shadow，也就是直接用/etc/shadow 文件中的用户账户及密码进行验证。为了操作方便，可以将系统默认的密码验证机制由 pam 改为 shadow，在配置文件/etc/sysconfig/saslauthd 中定义如下语句。

```
MECH=shadow
```

3. 测试 CyrusSASL V2 的认证功能

CyrusSASL V2 默认的密码认证方式是守护进程 saslauthd，查看 saslauthd 进程是否已经运行，可以执行如下命令。

```
[root@localhost ~]#ps aux | grep saslauthd
```

若没有发现 saslauthd 进程，则需启动该进程并设置为开机自启动，使用如下命令。

```
[root@localhost ~]#chkconfig --level 345 saslauthd on
[root@localhost ~]#/etc/init.d/saslauthd start
```

然后，使用如下命令测试 saslauthd 进程的认证功能。

```
/usr/sbin/testsaslauthd -u toplinux -p '123456'
```

如果结果显示有“0：OK Success.”字样，表示已成功启动了 saslauthd 进程的认证功能。

4. 设置 Postfix 启用 SMTP 认证

Postfix 主配置文件/etc/postfix/main.cf 中关于 SMTP 认证的部分如下。

```
smtpd_sasl_auth_enable = yes
broken_sasl_auth_clients=yes
smtpd_client_restrictions = permit_sasl_authenticated
smtpd_sasl_security_options = noanonymous
smtpd_sasl_local_domain = ''
smtpd_recipient_restrictions =
permit_mynetworks,
```

```
permit_sasl_authenticated,
reject_unauth_destination
```

5. 测试 SMTP 认证

通过 TELNET 指令来测试 SMTP 认证。首先，TELNET 连接到 Postfix 服务器的 25 端口，即发信服务器端口，然后，根据以下步骤来确定 SMTP 认证配置是否成功。

第 1 步，计算用户名和密码。

因为前文中配置的 Postfix 的 SMTP 用户身份认证采用的不是明文方式，所以首先要计算出用户名和密码。通过 perl 指令来完成。

```
[root@server named ]#perl -MMIME::Base64 -e 'print encode_base64("toplinux");'
Y2xpbnV4ZXI=          ; #经过编码的用户名 toplinux
[root@      server      named]#perl      -MMIME::Base64      -e      'print
encode_base64("mypassword");'
MTIzNDU2              ; #经过编码的密码 123456
```

第 2 步，测试登录。

通过 TELNET 指令登录到 Postfix 服务器的 25 端口。

```
[root@localhost ~]#telnet server.localdomain.tst 25
220 server.localdomain.tst ESMTP Postfix
EHLO gmail.com
250-server.localdomain.tst
250-PIPELINING
250-SIZE 10240000
250-VRFY
250-ETRN
250-AUTH LOGIN PLAIN
250-AUTH=LOGIN PLAIN
250-ENHANCEDSTATUSCODES
250-8BITMIME
250 DSN
AUTH LOGIN                                    ; #该指令验证用户身份
334 VXNlcm5hbWU6
Y2xpbnV4ZXI=                                  ; #输入用户名
334 UGFzc3dvcmQ6
MTIzNDU2                                      ; #输入密码
235 2.0.0 Authentication successful           ; #身份验证配置是正确的，登录成功
Quit                                          ; #退出
221 2.0.0 Bye
```

TELNET 登录成功，说明之前配置的 Postfix 服务器 SMTP 的用户身份验证是成功的。

3.2.7 启动和停止 Postfix 服务

Postfix 服务安装好之后，该如何完成启动、停止和重启 Postfix 服务的操作呢？控制该服务运行的命令如下。

1. 启动 Postfix 服务

使用如下指令。

```
[root@localhost ~]#/etc/rc.d/init.d/postfix start
启动 postfix:                                              [确定]
```

2. 停止 Postfix 服务

使用如下指令。

```
[root@localhost ~]#/etc/rc.d/init.d/postfix stop
关闭 postfix:                                              [确定]
```

3. 重新启动 Postfix 服务

使用如下指令。

```
[root@localhost ~]#/etc/rc.d/init.d/postfix restart
关闭 postfix:                                              [确定]
启动 postfix:                                              [确定]
```

4. 自动启动 Postfix 服务

如果希望Postfix服务随系统启动而自动加载，可以执行ntsysv命令启动服务配置程序，再找到“postfix”服务，在其前面加上星号“*”，然后选择“确定”按钮即可，如图 3-2 所示。

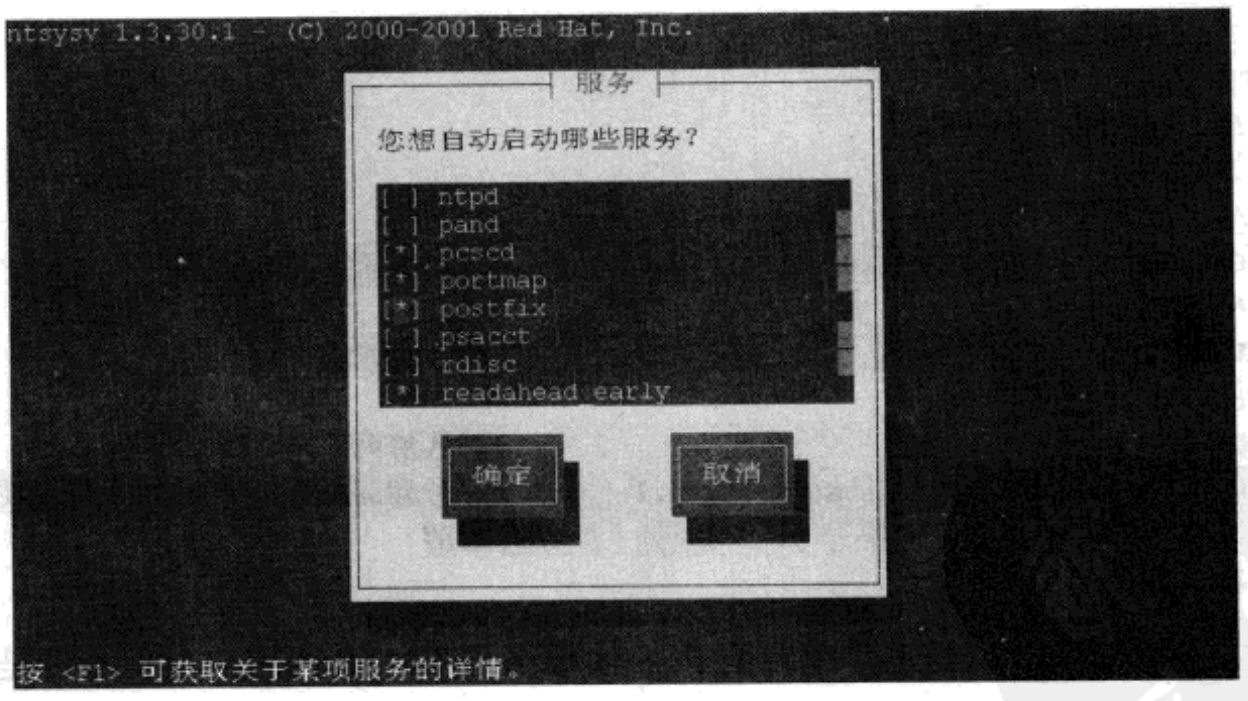


图 3-2 设置 Postfix 服务自动启动

3.2.8 配置与访问电子邮件客户端

在 SMTP、POP3、IMAP 服务都配置完成以后，又该如何使用这个邮箱呢？下边以 Windows

下的 Outlook Express 为例进行介绍。

第 1 步，设置邮件账号。

打开 Outlook Express 主页面，点击“设置邮件账号”超链接，显示新用户创建向导，如图 3-3 所示。

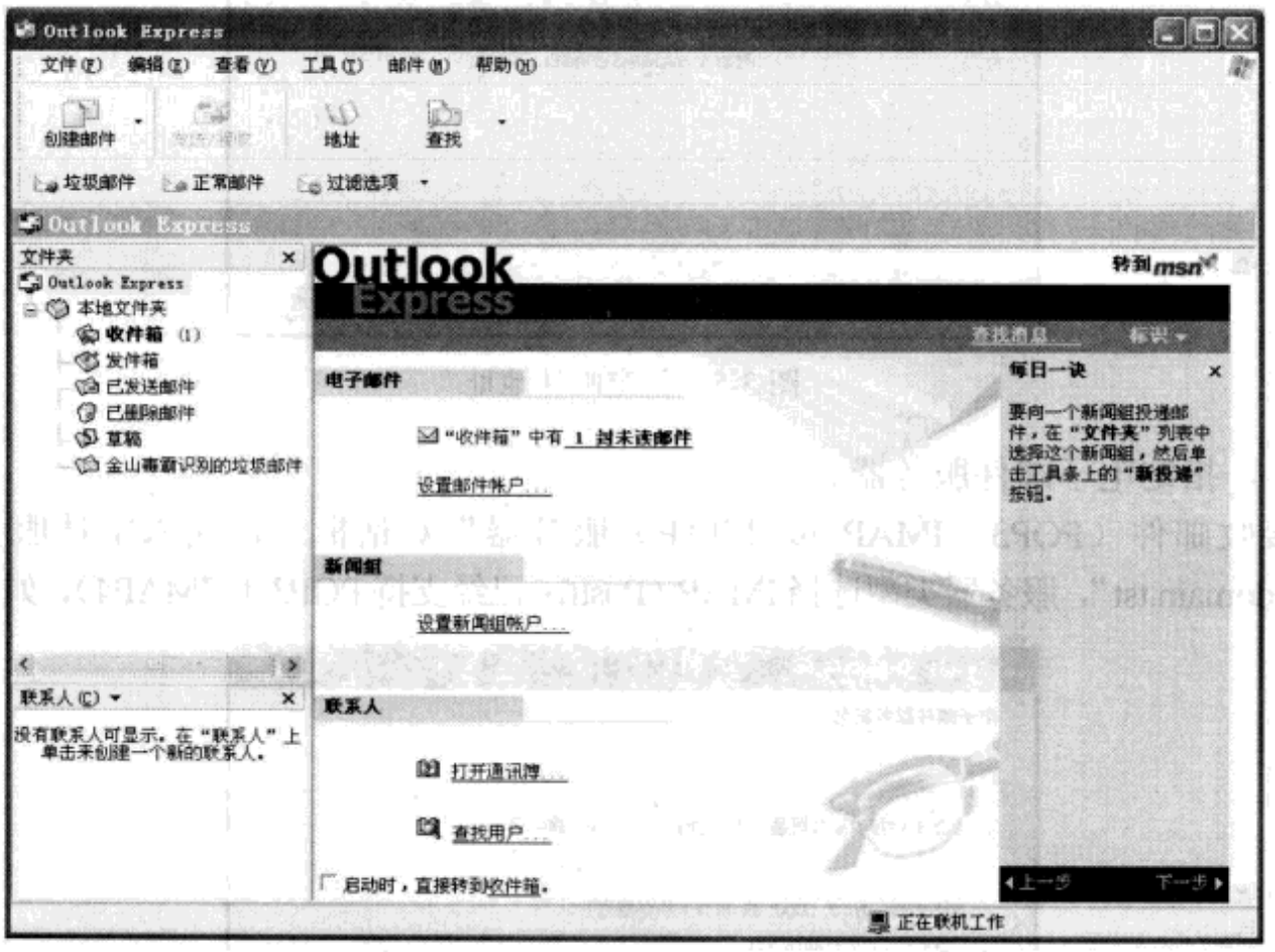


图 3-3　打开新用户创建向导

第 2 步，设置姓名。

在“显示名”文本框中，根据需要键入发件人的名字，如图 3-4 所示。

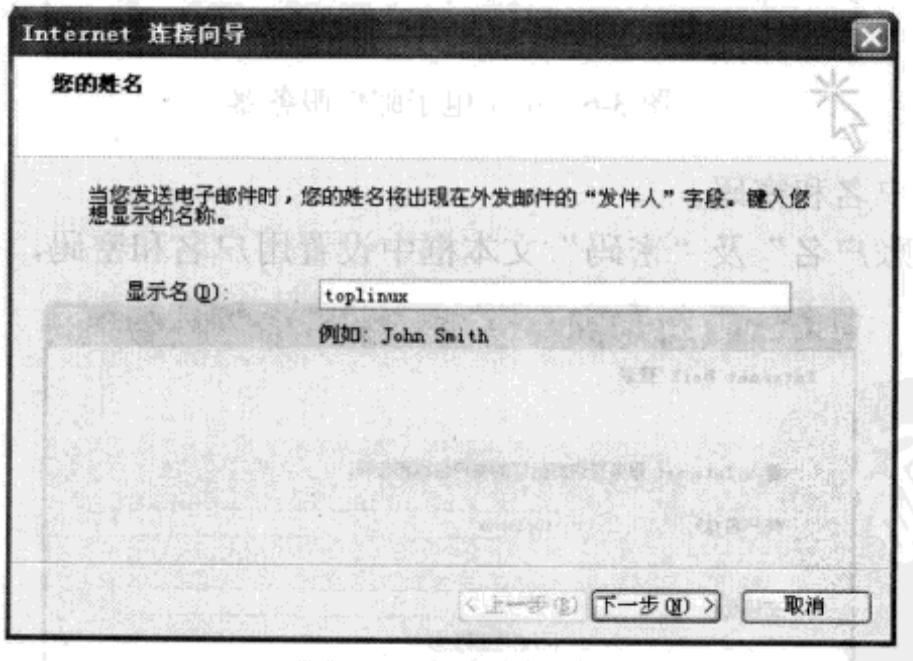


图 3-4　键入显示名

第 3 步，指定邮件地址。

在“电子邮件地址”文本框中，输入前文创建的邮箱 toplinux@localdomain.tst，如图 3-5 所示。

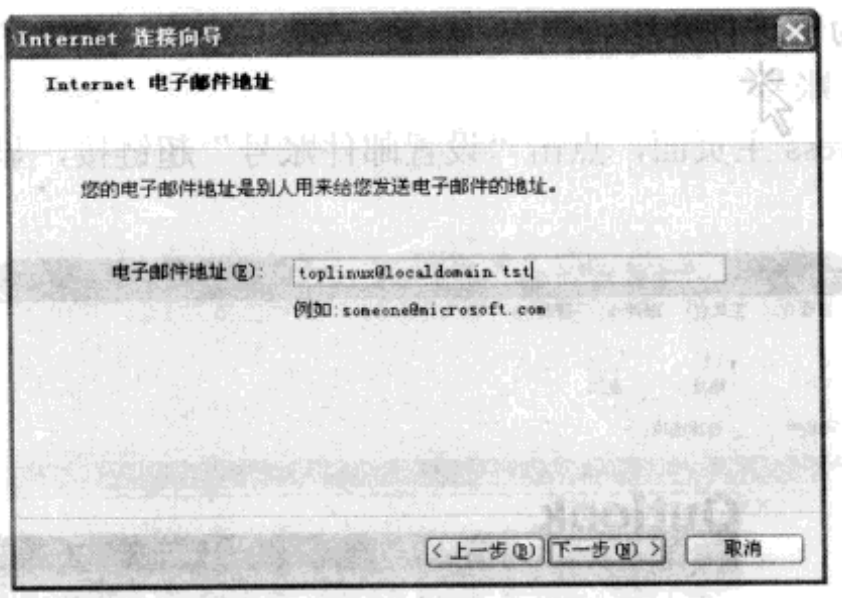


图 3-5　指定邮件地址

第 4 步，指定电子邮件服务器。

在“接收邮件（POP3，IMAP 或 HTTP）服务器”对话框中，输入邮件服务器的名字“server.localdomain.tst”，服务器类型选择 IMAP（Postfix 已经支持 POP3 和 IMAP4），如图 3-6 所示。

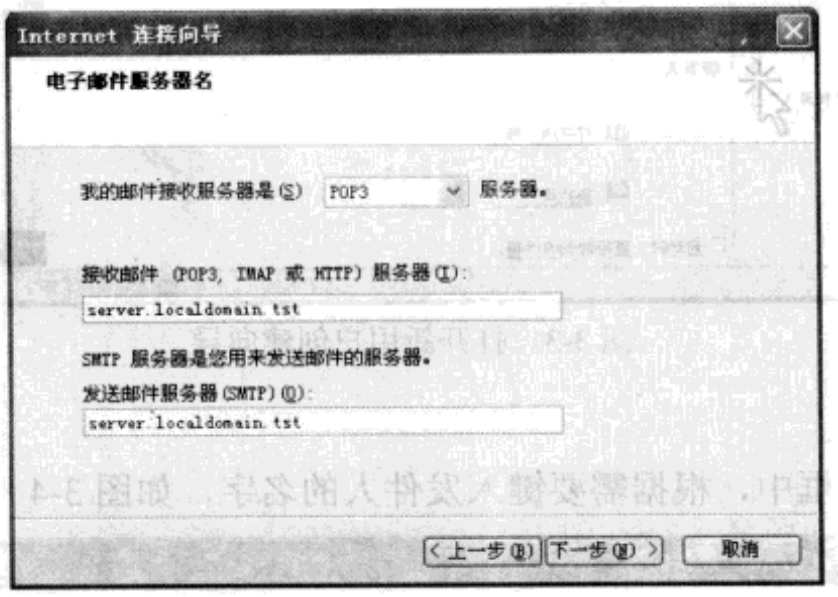


图 3-6　指定电子邮件服务器

第 5 步，设置用户名和密码。

接下来分别在“账户名”及“密码”文本框中设置用户名和密码，如图 3-7 所示。

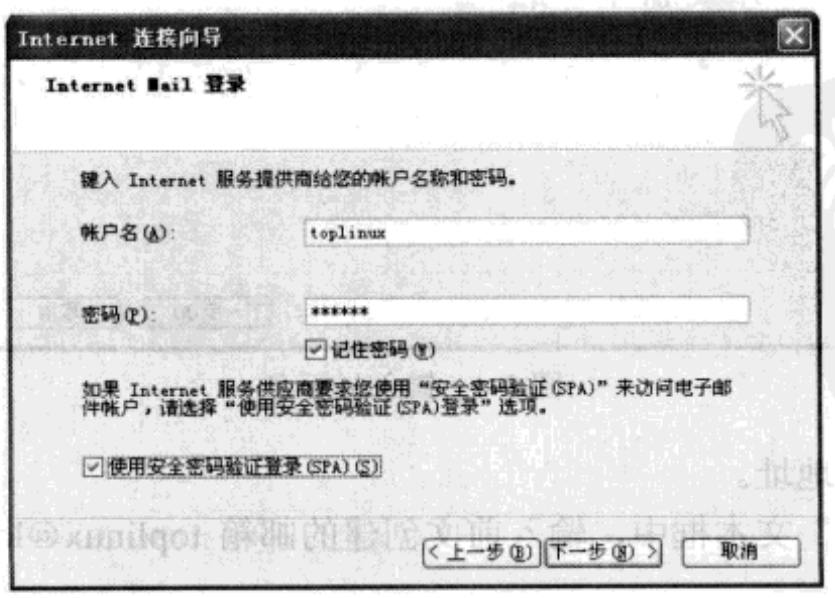


图 3-7　设置登录用户名和密码

至此，电子邮件客户端配置完成，测试能否正常地发送和接收邮件。

3.3 POP 和 IMAP 邮件服务

在完成本章前面部分的操作后，Postfix 服务器已经可以为授权的网段或者网域发送和转发邮件了。但是要让这个邮件服务器能够通过客户端授权邮件，还必须安装相应的 POP3 和 IMAP 功能。

3.3.1 dovecot 服务

Postfix 服务只是一个邮件传输代理，只提供邮件的转发及本地的分发功能即 SMTP 服务。要使一台服务器既可以发送邮件，又可以保存邮件，还必须安装 POP 或 IMAP 服务。通常情况下，将 STMP 服务和 POP 或 IMAP 服务安装在同一台主机上，这台主机就可以称为电子邮件服务器了。

在 Red Hat Enterprise Linux 5 中，dovecot 和 CyrusImapd 两个软件可以同时提供 POP 和 IMAP 服务。按照以下步骤可以通过 dovecot 来实现 Red Hat Enterprise Linux 5 下 Postfix 的 POP3 和 IMAP4 支持。

第 1 步，安装 dovecot。

首先使用如下命令检查 dovecot 服务是否已经安装到系统中。

```
[root@localhost ~]#rpm -qa |grep dovecot
```

如果系统还没有安装 dovecot 服务，在 Red Hat Enterprise Linux 5 安装光盘的 Server 目录下找到 dovecot 服务的 RPM 安装文件 dovecot-1.0.7-2.el5.i386.rpm 及相关程序，然后使用如下命令安装 dovecot 服务和相关程序。

```
[root@localhost ~]#rpm -ivh /mnt/Server/perl-DBI-1.52-1.fc6.i386.rpm
[root@localhost ~]#rpm -ivh /mnt/Server/mysql-5.0.45-7.el5.i386.rpm
[root@localhost ~]#rpm -ivh
/mnt/Server/postgresql-libs-8.1.11-1.el5_1.1.i386.rpm
[root@localhost ~]#rpm -ivh /mnt/Server/dovecot-1.0.7-2.el5.i386.rpm
```

第 2 步，配置 dovecot。

dovecot 服务的配置文件是/etc/dovecot.conf。启用最基本的 dovecot 服务，只需修改该配置文件中的以下内容即可（只需要去掉原配置文件中这些内容的注释，并稍加修改）。

```
protocols =imap imaps pop3 pop3s
protocol imap {
    listen = *:10143
    ssl_listen=*:10943
}
protocol pop3 {
    listen = *:10100
}
```

第 3 步，配置自动启动 dovecot ，执行以下指令。

```
[root@localhost ~]#chkconfig --level 345 dovecot on
[root@localhost ~]#service dovecot start
启动 Dovecot Imap:                                         [确定]
[root@localhost ~]#
```

3.3.2 CyrusImapd 服务

CyrusImapd 服务与 dovecot 服务类似，但是其安装、配置和启动又有所不同。Red Hat Enterprise Linux 默认并没有安装 CyrusImapd 服务，要检查系统是否已经安装了 CyrusImapd 服务，可使用如下命令。

```
[root@localhost ~]#rpm -qa | grep cyrus-imapd
```

由于系统当前还没有安装 CyrusImapd 服务，需要先进行安装。

1. 安装 CyrusImapd 软件包

CyrusImapd 服务依赖的包很多，安装时很容易出错，所以这里所采用的安装方式和本文提到的其他软件的安装方式有所不同，采用的是创建本地安装源的方式安装。

注意　虽然在 Red Hat Enterprise Linux 5 中，已经可以在“软件管理包”中的“查看”→“软件存储库管理”菜单中添加光盘为安装源。但很多用户在日常使用的过程中都碰到过无法查看“软件存储库管理”的问题，在这种情况下，就只能通过编辑配置文件来实现所需功能支持了。

第 1 步，创建本地安装源文件。

新建一个文件，命名为 rhel5.localinstall.repo，保存在/etc/yum.repos/目录下，内容如下

```
[rhel-localinstall]
name= Red Hat Enterprise Linux 5.2 LocalInstall
baseurl=file:///mnt/cdrom/Server/
gpgcheck=1
enabled=1
gpgkey=file:///mnt/cdrom/RPM-GPG-KEY-redhat-release
```

第 2 步，导入 key 文件。

首先将 Red Hat Enterprise Linux 5 的安装盘加载到/mnt/cdrom 下，然后执行以下两条指令，导入 RPM 安装包的验证 key 文件。

```
[root@localhost ~]#rpm --import /mnt/cdrom/RPM-GPG-KEY-redhat-beta
[root@localhost ~]#rpm --import /mnt/cdrom/RPM-GPG-KEY-redhat-release
```

第 3 步，开始安装。

在 Red Hat Enterprise Linux 5 桌面环境下，单击“应用程序”→“添加/删除软件”菜单命令，显示“软件包管理者”窗口，然后单击“编辑”→“软件存储库”菜单命令，显示“存储

库管理工具”对话框，选中“rhel-localinstall”复选框并退出，如图 3-8 所示。

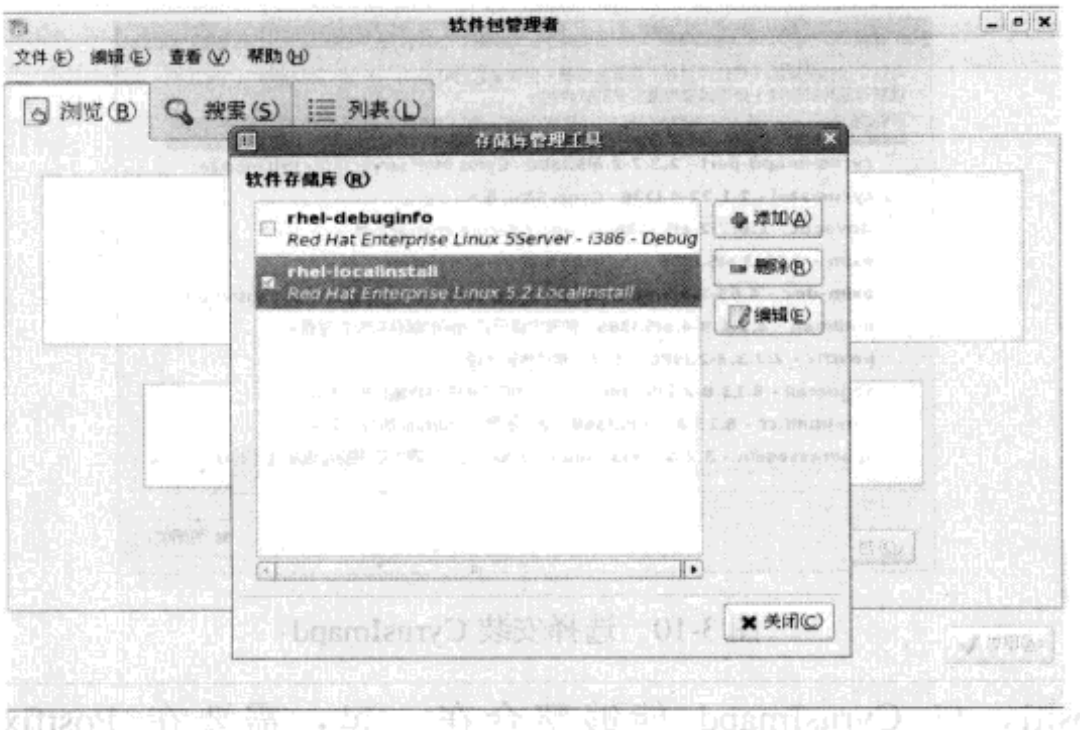


图 3-8　添加软件安装源

在“浏览”选项卡中，单击左侧的“服务器”选项，然后在右侧的下拉列表中找到“邮件服务器”项，如图 3-9 所示。

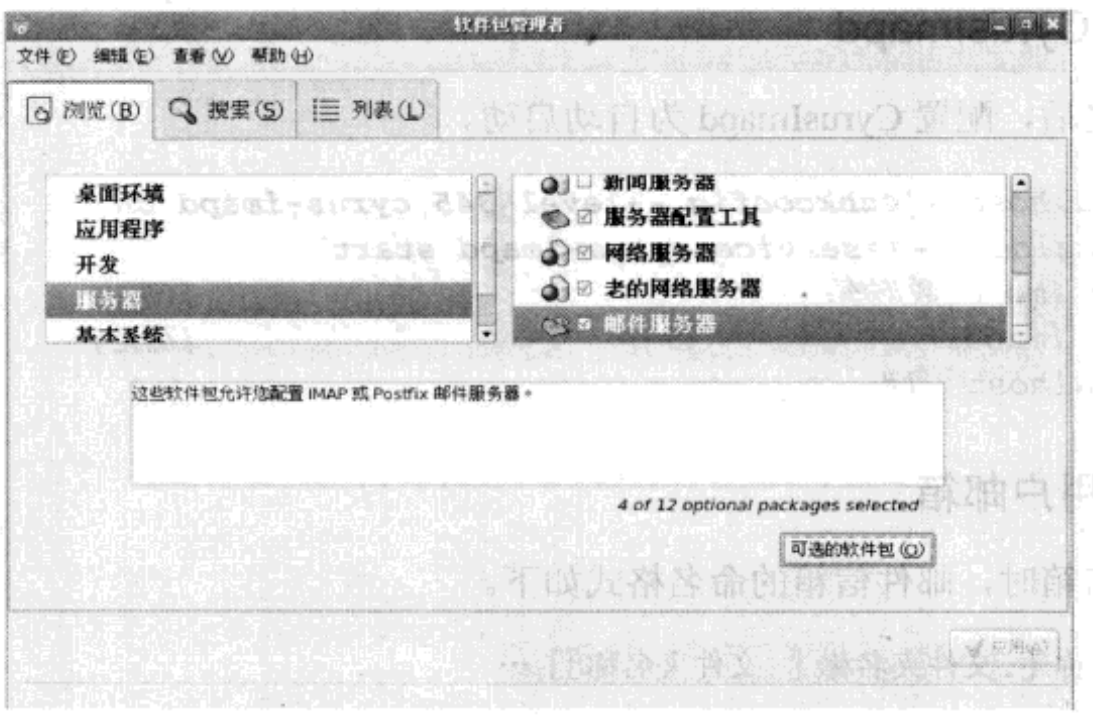


图 3-9　选择安装更多的服务器

选中“邮件服务器”复选框，然后单击下侧的“可选的软件包”按钮，显示“邮件服务器中的软件包”对话框，选中 CyrusImapd 和 CyrusImapd-perl 软件包的复选框，再依次单击“关闭”→“应用”按钮，开始安装 CyrusImapd 服务，如图 3-10 所示。

2. 配置 CyrusImapd

CyrusImapd 服务的配置文件包括 /etc/sysconfig/cyrusimapd 、 /etc/cyrus.conf 和 /etc/imapd.conf。其中/etc/sysconfig/cyrusimapd 是 CyrusImapd 服务的配置文件，用于启动 CyrusImapd 服务；/etc/cyrus.conf 是 CyrusImapd 服务的主要配置文件，包含 CyrusImapd 服务中各个组件（IMAP、POP3、SIEVE 和 NNTP 等）的设置参数；/etc/imapd.conf 是 CyrusImapd

服务中的 IMAP 服务配置文件。

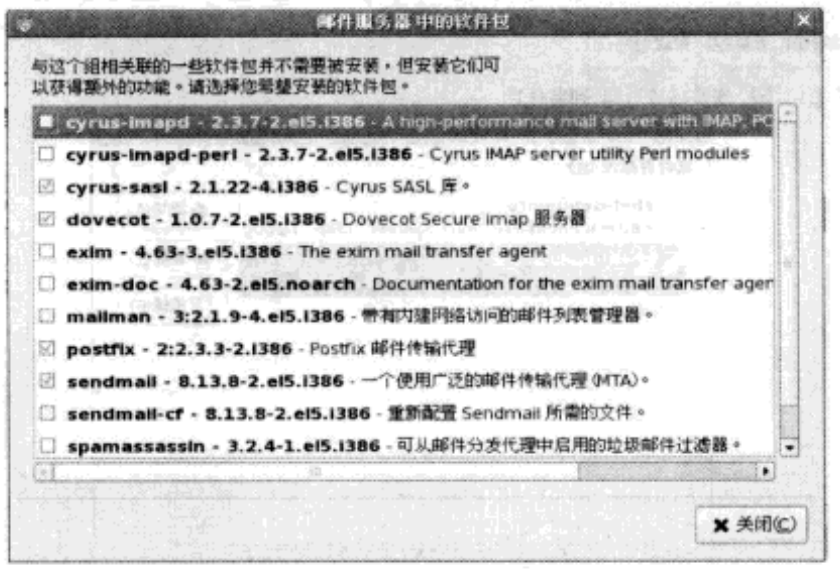


图 3-10　选择安装 CyrusImapd

为了使 Postfix 与 CyrusImapd 能够整合在一起，需要在 Postfix 的主配置文件 /etc/postfix/main.cf 中加入以下内容。

```
mailbox_transport = lmtp:unix:/var/lib/imap/socket/lmtp
```

3. 启动 CyrusImapd

存盘退出之后，配置 CyrusImapd 为自动启动。

```
[root@localhost ~]#chkconfig --level 345 cyrus-imapd on
[root@localhost ~]#service cyrus-imapd start
导入 cyrus-imapd 数据库:                                    [确定]
启动 cyrus-imapd:                                          [确定]
[root@localhost ~]#
```

4. 创建用户邮箱

创建邮件信箱时，邮件信箱的命名格式如下。

```
信箱类型.名称 [.文件夹名称 [.文件夹名称]] …
```

第 1 步，设置 CyrusImap 管理员账户 cyrus 的密码。

```
[root@localhost ~]#passwd cyrus
Changing password for user cyrus.
New UNIX password:                    输入密码
BAD PASSWORD: it is based on a dictionary word
Retype new UNIX password:             确认密码
passwd: all authentication tokens updated successfully.
[root@localhost ~]#
```

第 2 步，为用户创建邮件信箱，使用 cyradm 管理工具。

```
[root@localhost ~]# /usr/bin/cyradm -u cyrus  localhost
```

第 3 步，为用户 toplinux 创建一个邮件信箱，使用以下命令。

```
server.localdomain.tst>creatE-mailbox user.toplinux
```

除了 creatE-mail 命令外，在 cyadm 模式下还有很多其他指令。如表 3-1 所示为常用的 cyradm 管理命令及其缩写形式。

表 3-1 常用的 cyradm 管理命令及其缩写形式

命　令	缩　写	描　述
listmailbox	lm	列出与给定字符串相匹配的所有邮件信箱的名称
creatE-mailbox	cm	创建一个新的邮件信箱
deletE-mailbox	dm	删除一个邮件信箱及其下层的所有文件夹
renamE-mailbox	renm	修改邮件信箱的名称
setaclmailbox	sam	设置邮件信箱用户的访问权限
deleteaclmailbox	dam	删除用户访问邮件信箱的部分或全部权限
listaclmailbox	lam	查看邮件信箱的访问权限列表
setquota	sq	设置邮件信箱配额
listquota	lq	列出邮件信箱配额

第 4 步，在用户邮件信箱下添加其他文件夹。

为用户 toplinux 在其邮件信箱下创建发件箱、草稿箱和垃圾箱等文件夹，使用如下命令。

```
localhost.localdomain> creatE-mailbox user.toplinux.Drafts
localhost.localdomain> creatE-mailbox user.toplinux.Send
localhost.localdomain> creatE-mailbox user.toplinux.Trash
```

第 5 步，设置用户邮件信箱配额。

设置用户 toplinux 的信箱 user.toplinux 的配额为 10 MB，使用如下命令。

```
localhost.localdomain> setquota user.toplinux 10240
```

查看用户邮箱的使用情况，使用如下命令。

```
localhost.localdomain> su -l cyrus -c /usr/lib/cyrus-imapd/quota
```

第 6 步，为电子邮件客户端设置权限。

在创建了用户信息 user.toplinux 之后，即使是管理员 cyrus 也不能直接使用 deletE-mailbox 命令来删除该邮件邮箱。只有先为管理员 cyrus 授予完全的权限（all），才能完成删除操作，使用以下命令。

```
[root@localhost ~]#/usr/bin/cyradm -u cyrus localhost
IMAP Password:
          localhost.localdomain> setacl user.toplinux cyrus all
localhost.localdomain> deletE-mailbox user.toplinux
localhost.localdomain> listacl user.toplinux
Mailbox does not exist
```

CyrusImap 中的用户邮箱权限有多种，具体请参考表 3-2 CyrusImap 中的用户信箱权限。

表 3-2 CyrusImap 中的用户信箱权限

权限缩写	描述
none	无任何权限
read	允许读取信箱内容
post	允许读取和粘贴信息（如发邮件）
append	允许读取、粘贴与插入信息
write	除具有 append 的所有权限外，还具有删除邮件的权限。但不具有变更信箱的权限
all	具有所有权限

3.4 Web 支持电子邮件

目前，大多数流行的邮件服务都支持更加方便的 Web 方式接收和发送邮件功能，比如 Yahoo mail、Gmail 和 Hotmail 等。其实经过适当的配置， Postfix 也可以支持 Web 方式收发电子邮件。

3.4.1 Squirrelmail 的安装与配置

Red Hat Enterprise Linux 5 内置的 Web 支持软件是 Squirrelmail，不仅功能强大而且配置也很方便。

1. 安装 Squirrelmail

虽然 Squirrelmail 是内置的 Red Hat Enterprise Linux 5 Web 支持软件，但却并没有默认安装该软件。确认其是否安装，可以执行以下指令。

```
[root@localhost ~]#rpm -q squirrelmail
```

安装 Squirrelmail 的方法是：运行 Red Hat Enterprise Linux 5 的安装光盘，并在 Server 目录下找到 Squirrelmail 的 RPM 安装包文件，分别是 php-mbstring-5.1.6-20.el5.i386.rpm 和 Squirrelmail-1.4.8-4.0.1.el5.noarch.rpm，使用如下命令进行安装。

```
[root@localhost Server]# rpm -ivh php-mbstring-5.1.6-20.el5.i386.rpm
warning: php-mbstring-5.1.6-20.el5.i386.rpm: Header V3 DSA signature: NOKEY,
key ID 37017186
Preparing...                ########################################### [100%]
   1:php-mbstring           ###########################################
[100%]
[root@localhost Server]# rpm -ivh squirrelmail-1.4.8-4.0.1.el5.noarch.rpm
warning: squirrelmail-1.4.8-4.0.1.el5.noarch.rpm: Header V3 DSA signature:
NOKEY, key ID 37017186
Preparing...                ########################################### [100%]
   1:squirrelmail           ###########################################
[100%]
[root@localhost Server]#
```

注意 如果系统没有安装 Apache 和 PHP，由于软件的依赖性问题会很容易出错。为了避免这样的情况，最好是通过 Red Hat Enterprise Linux 5 的“添加/删除软件”工具来安装。进入“软件包管理者”后，单击“服务器”→“邮件服务器”按钮，弹出“邮件服务器中的软件包”对话框，然后选中 Squirrelmail 软件包的复选框，完成安装操作，如图 3-11 所示。

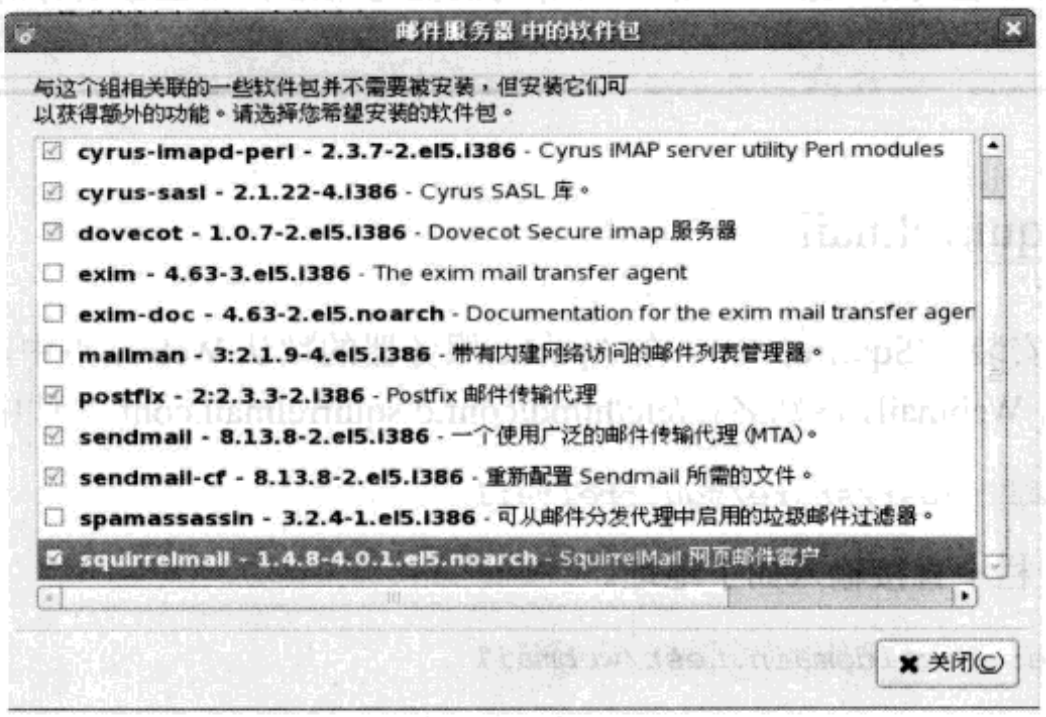


图 3-11 “邮件服务器中的软件包”对话框

2. 配置 Squirrelmail

/etc/squirrelmail/config.php 是 Squirrelmail 的主配置文件，可以通过直接修改该文件的内容来完成 Squirrelmail 的配置工作。但是更方便、更直观的配置方式还是通过 Squirrelmail 的配置工具来完成的。打开 Squirrelmail 的配置工具，执行以下命令。

```
[root@localhost ~]#/usr/share/squirrelmail/config/conf.pl

SquirrelMail Configuration : Read: config.php (1.4.0)
---------------------------------------------------------
Main Menu --
1.  Organization Preferences
2.  Server Settings
3.  Folder Defaults
4.  General Options
5.  Themes
6.  Address Books
7.  Message of the Day (MOTD)
8.  Plugins
9.  Database
10. Languages

D.  Set pre-defined settings for specific IMAP servers
```

```
C   Turn color off
S   Save data
Q   Quit

Command >>
```

Squirrelmail 配置界面包括组织信息、服务器设置、目录设置、一般选项等 10 个部分。

注意 为使新配置的系统生效，本章所有的设置结束后，都应该要保存（S），并退出配置界面（Q）。

3.4.2 登录 Squirrelmail

采用 RPM 包安装的 Squirrelmail，在 Apache 服务器的默认 Web 站点中，已经存在一个安装程序配置的别名 Webmail，该别名在/etc/httpd/conf.d/squirrelmail.conf 文件中定义，定义如下。

```
Alias /webmail /usr/share/squirrelmail
```

在浏览器地址栏中直接输入如下地址。

```
http://server.localdomain.test/webmail
```

即可打开如图 3-12 所示的 Squirrelmail 的登录页面。

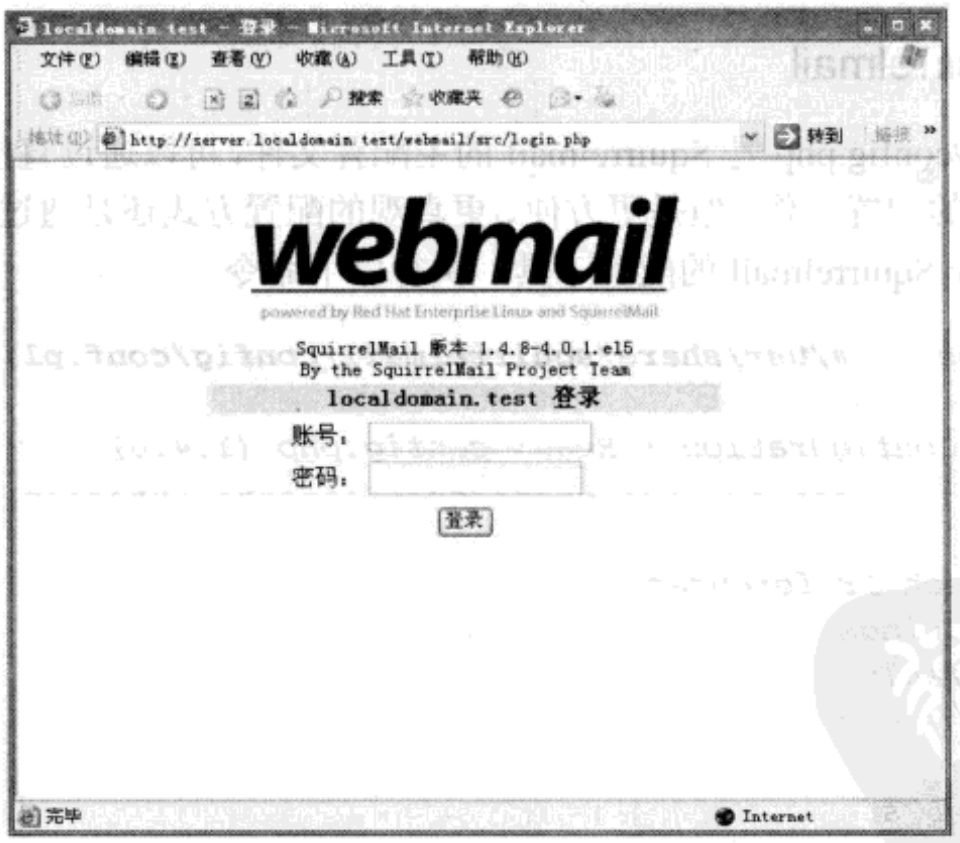


图 3-12 Squirrelmail 的登录界面

第 4 章　Web 服务器

上网经常需要利用 Internet 查询用户所需的信息，而 Web 服务已经成为 Internet 提供的最重要的服务之一。本章首先对 Web 服务的相关知识进行了阐述，然后着重介绍了如何在 Linux 系统中配置与管理 Apache 服务器，并在最后结合实用案例讲解如何利用 JSP 技术来创建动态 Web 网站。

4.1　Web 概述

Web 是一种超文本信息系统，它的一个主要概念是超文本链接，它使得文本不再像一本书一样是固定的、线性的，而是可以从一个位置跳到另外的位置。正是这种多连接性我们才把它称为 Web。

4.1.1　Web 服务简介

Web 服务通常可以分为静态 Web 服务和动态 Web 服务两种。

最初所有的 Web 页面都是静态的。网站上的 Web 页面只是格式化的电子文本，一经生成，无法改变内容，只能更改网页文件，这给网站的维护带来了很大的不便。随着互联网和 Web 的迅速普及，静态的 Web 页面已不能满足人们对于信息的丰富性和多样性的强烈需求。如此便促使了 Web 客户端（即浏览器）和 Web 服务器都开始了从静态方式向动态方式的演进。

1. Web 客户端从静态方式向动态方式的演进

Web 客户端首先引入了对 GIF 格式图像的支持。随着在网页中引用 JavaScript、JavaApplet、VBScript 等脚本语言，HTML 页面的功能得到了极大的丰富，逐渐可以提供更多的客户交互操作方式和更多的客户端应用。而 CSS（Casing Style Sheets，层叠样式表）和 DHTML（Dynamic HTML）技术的引入，丰富了页面内容，实现了 HTML 页面的动态效果。如今人们使用的 Web 客户端更是引入了各种插件（如 QuickTime 插件）、控件（ActiveX 控件），并良好地支持 Flash 播放。人们不仅可以看文字、图片，还可以听声音，看视频等。

2. Web 服务器从静态方式向动态方式的演进

在 Web 客户端不断增强其信息展现能力和客户交互能力的同时，Web 服务器也在由静态向动态逐步发展、不断完善。

最早的 Web 服务器只能响应 Web 客户端发送的 HTTP 请求，并将存储在 Web 服务器上的 HTML 文件返回。如果能根据 Web 客户端的请求在 Web 服务器端动态地创建 HTML 文件，那将是一件多么完美的事情。CGI（Common Gate Interface，通用网关接口）技术的出现使其成为可能。

CGI 是用于连接网页和应用程序的接口，通过 CGI 技术，可以实现在 Web 服务器端进行数据库查询等复杂操作，并对进行了一系列操作后的数据进行封装，动态产生 HTML 文件并返回给 Web 客户端，这为客户端和服务器端间交互动态信息起了传输纽带的作用。

CGI 技术推动了聊天室、论坛、电子商务、信息查询、数据检索等各种 Web 应用的迅速发展。

注意 CGI 程序可以使用 C、C++、Pascal、Perl、Python 等语言来编写。相对于 CGI 程序，Sun 公司相继推出了 Servlet 和 JSP 技术，这些技术的使用方便了程序员使用 Java 语言开发 Web 应用。

随着一系列技术的应用（如 XML、AJAX、Web Service 等），Web 应用的发展空间将越来越宽广。

4.1.2 HTTP 协议简介

HTTP（HyperText Transfer Protocol，超文本传输协议）是分布式的 Web 应用的核心技术协议，如今被广泛应用 。该协议基于请求/响应模式，无状态，在 TCP/IP 协议栈中属于应用层。之所以说 HTTP 协议是无状态协议，是因为此协议一次请求和响应构成一个独立的事务，各个事务间没有状态的联系。

HTTP 协议的服务器软件有：Sun 的 Apache 和 Microsoft 的 IIS，其客户端分别由 Microsoft 的 Internet Explorer 和 Mozilla 的 Firefox 实现。

1. 客户端访问 Web 服务器的几个阶段

默认情况下，当 Web 用户连接到 Web 服务器时，客户机（浏览器）通过与服务器的 80 端口上的监听程序建立连接联系。连接建立后，Web 服务将等待浏览器的请求。例如，当用户使用浏览器浏览资源时，Web 服务器便会接到客户端的请求。该请求包含了 HTTP 协议的 GET 命令（用于向服务器请求下载一个文件）和标题（向服务器提供客户端的信息，客户端可以接受的文件格式）。当服务器接到浏览器发出的请求后，服务器会发出相关信息来响应浏览器，该信息的前部分是有关服务器的信息。

一般情况下，客户端访问 Web 服务器需要经过如下几个阶段。

阶段 1，浏览器使用 HTTP 命令（GET/POST）向服务器发出 Web 请求，如图 4-1 所示。

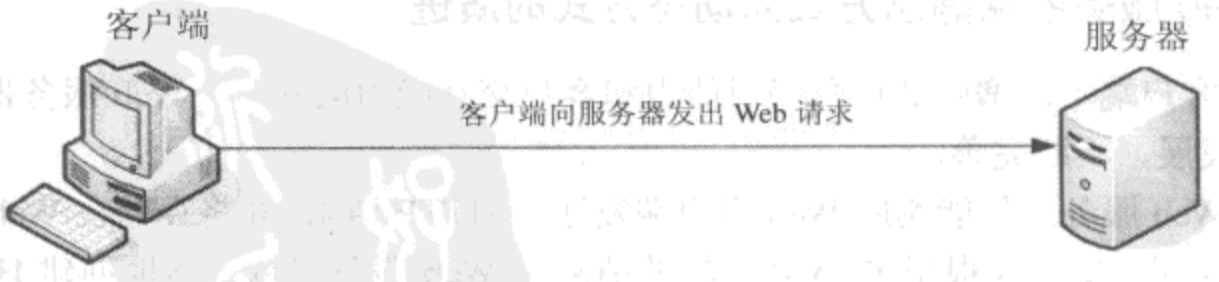


图 4-1 发出 Web 请求

注意 HTTP 最基本的命令是 GET 和 POST。其中，GET 用于获取信息，且对同一 URL（统一资源定位器）的多个请求应该返回同样的结果；而 POST 表示改变服务器上的资源的请求。

阶段 2，服务器接收到页面请求后，服务器响应该请求，并在客户和服务器之间建立连接，如图 4-2 所示。

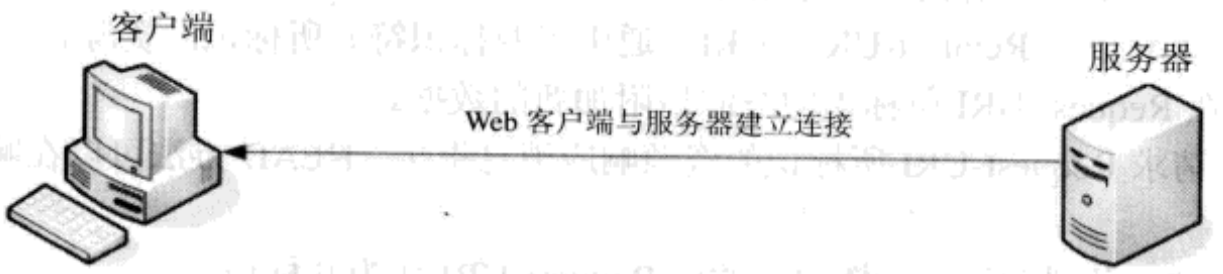


图 4-2 建立连接

阶段 3，Web 服务器查找客户端所请求的资源，有两种结果：其一，如果该资源不存在，则服务器会发送一个相应的错误提示文档给客户端；其二，如果 Web 服务器查找到所需资源，其会将客户端所请求的资源传送给 Web 浏览器，如图 4-3 所示。

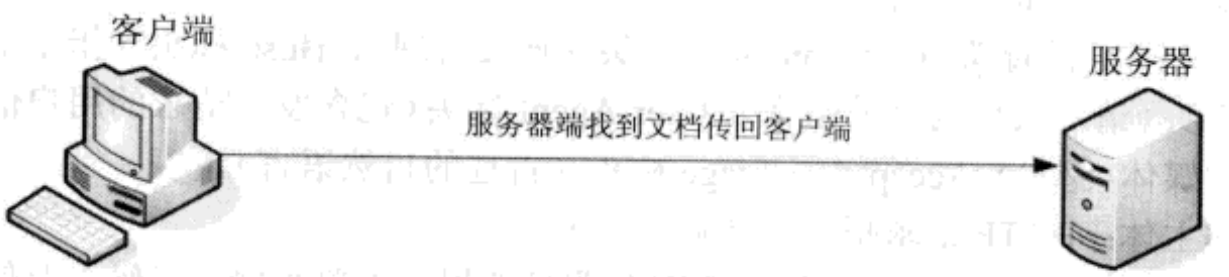


图 4-3 将请求资源传送给 Web 浏览器

阶段 4，Web 浏览器成功接收资源后，资源将被解释并显示在屏幕上，如图 4-4 所示。

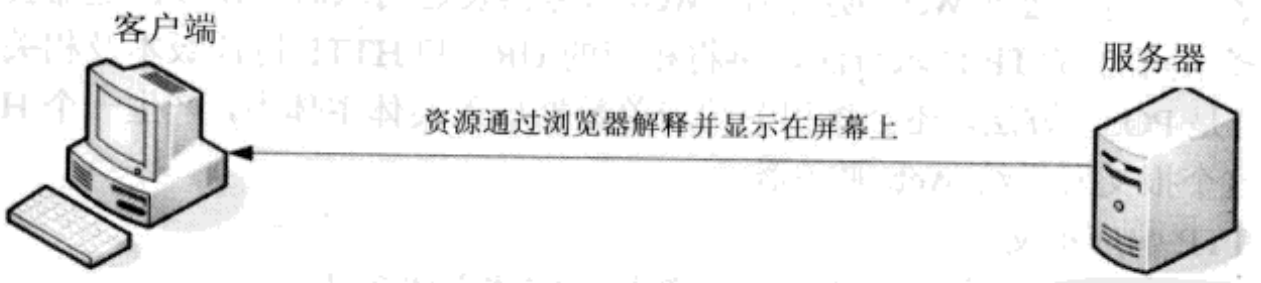


图 4-4 解释并显示资源

阶段 5，客户端浏览资源成功后，便会断开与服务器的连接。

2. HTTP 请求端口

HTTP 请求的默认端口号是 80，此种形式“Host:www.xx.com”后面没有端口号，表明使用的是默认端口，为了实现在同一台服务器上运行多台 Web 服务器，每台服务器监听不同的端口，可以通过配置使某 Web 服务器使用另外一个端口（如 8080），可以通过以下方式访问。

```
http://www.xxx.com:8080/
```

3. HTTP 报文详解

HTTP 有两种报文：请求报文和响应报文，具体介绍如下。

（1）HTTP 请求报文

HTTP 请求报文主要包括请求行、头部行以及实体主体 3 部分。

- 请求行：HTTP 请求报文的第一行

请求行有方法字段、URL 字段和 HTTP 协议版本字段 3 个字段。其中，方法字段严格区分大、小写，当前 HTTP 协议中的方法都为大写，具体的方法字段介绍如下。

GET　请求获取由 Request-URI（URI，通用资源标识符）所标识的资源。

POST　在 Request-URI 所标识的资源后附加新的数据。

HEAD　请求 Request-URI 所标识的资源响应消息报头，HEAD 方法可以在响应时不返回消息体。

PUT　请求服务器存储一个资源，并用 Request-URI 作为其标识。

DELETE　请求服务器删除由 Request-URI 所标识的资源。

TRACE　请求服务器回送收到的请求信息，主要用于测试或诊断。

CONNECT　保留将来使用。

OPTIONS　请求查询服务器的性能，或者查询与资源相关的选项。

- 头部行：位于请求行下面

请求报文中常见的标头有：Connetion 标头（连接管理）、Host 标头（指定请求资源的主机）、Range 标头（请求实体的字节范围）、User-Agent 标头（包含发出请求的用户信息）、Accept 标头（首选的媒体类型）、Accept-Language 标头（首选的自然语言）。

- 实体主体：HTTP 请求报文的最后一部分

实体主体用于 POST 方法中。用户向 Web 服务器提交表单数据的时候需要使用 POST 方法，此时实体主体中包含用户填写在表单中的各个属性字段的值。Web 服务器在收到 POST 方法的 HTTP 请求报文后，可以从实体主体中取出需要的属性字段的值。

简而言之，当用户通过 Web 浏览器向 Web 服务器发送请求时，Web 浏览器会根据用户的具体请求选择不同的 HTTP 请求方法，再将相应的 URL 和 HTTP 协议版本及相关的标头填入头部行中，若是 POST 方法，还会将相关的表单数据填入实体主体中，产生一个 HTTP 请求报文，然后将这个报文发送给 Web 服务器。

（2） HTTP 响应报文

HTTP 响应报文主要包括有状态行、头部行、实体主体 3 部分。

- 状态行：HTTP 响应报文的第一行

状态行包括 3 个字段：协议版本、状态码和相应的状态信息即原因字段，常见的状态码和原因字段的说明如表 4-1 所示。

- 头部行：位于 HTTP 响应报文状态行后

响应报文中常见的标头有：Date 标头（消息生产的时间）、Server 标头（服务器标识）、Accept-Ranges 标头（部分请求）、ETage 标头（不透明验证者）、Location（URL 的备用位置）、Content-Length 标头（实体主体的长度）、Content-Type 标头（实体主体的媒体类型）。

表 4-1 HTTP 状态代码与状态描述

状态代码	状态描述	简要说明
200	ok	客户端请求成功
301	Moved Permanetly	请求对象已经转移，新的 URL 在响应报文 Location 首部行中指定
400	bad request	由于客户端请求有语法错误，不能被服务器所理解
401	unauthorized	请求未经授权。这个状态代码必须和 www-authenticate 报头域一起使用
403	forbidden	服务器收到请求，但是拒绝提供服务。服务器通常会在响应报文中给出不提供服务的原因
404	not found	请求的资源不存在，例如，输入了错误的 URL
505	HTTP Version Not Supported	服务器不支持客户端指定的 HTTP 协议版本

- 实体主体：位于头部行后面

实体主体包含了 Web 客户端请求的对象。Content-Length 标头（实体主体的长度）及 Content-Type 标头（实体主体的媒体类型）用于计算实体主体的位置、数据类型和数据长度。当 Web 服务器接收到 Web 客户端的请求报文后，对 HTTP 请求报文进行解析，并将 Web 客户端请求的对象取出打包，通过 HTTP 响应报文将数据传回给 Web 客户端，如果出现错误则返回包含对应错误的错误码和错误原因的 HTTP 响应报文。

4.1.3 Web 服务器软件

Web 服务器软件的种类非常丰富，但最为流行的 Web 服务器软件是免费的 Apache 和 IIS。用户可以根据不同的需要选择使用何种 Web 服务器软件，这两者具体的优、缺点如下所示。

1. Apache

费用形式：完全免费。

配置复杂程度：相对复杂。

稳定性：虽配置稍微复杂，但设置完毕之后就可以长期工作。

扩展性：不仅可用于 Windows 平台还可以用于 Linux、UNIX 及 FreeBSD 操作系统。对各种插件支持良好，对于 Perl、CGI、PHP 和 Java 等都能够完美支持。

2. IIS

费用形式：是 Windows 服务器操作系统中的内置组件，须购买正版的 Windows。

稳定性：IIS 服务中经常遇到 HTTP 500 内部错误，稳定性相对较弱。

扩展性：IIS 只能在微软公司的 Windows 操作系统下使用。对各种插件支持良好，对于 Perl、CGI、PHP 和 Java 等都能够完美支持。

4.2 Apache 概述

Apache 是 Apache HTTP Server 的简称形式，是由 Apache 软件基金会维护、开发的开放源码的 HTTP 服务器，具有跨平台和较高安全性的特点，有众多优秀程序员对其不断进行完善，因此得到了广泛使用。

4.2.1 Apache 的诞生

Apache 诞生于伊利诺伊大学香槟分校 NCSA（National Center for Supercomputer Applications，国家超级电脑应用中心）。随着 Apache 的不断发展和完善，其功能及速度已明显优越于其他的基于 UNIX 的 HTTP 服务器。

1. “Apache” 名称的由来

Apache 音译为阿帕奇，是北美印第安人的一个部落，叫阿帕奇族，在美国的西南部。相传阿帕奇是一个武士，他英勇善战，且战无不胜，被印第安人奉为勇敢和胜利的代表，因此后人便用他的名字为印第安部落命名。

据相关服务器官方网站的 FAQ（常见问题解答）中介绍：“‘Apache’这个名字是为了纪念名为 Apache（印地语）的美洲印第安人土著的一支，众所周知他们拥有高超的作战策略和无穷的耐性”。

2. Apache 服务成长

Apache 第一个版本发布于 1995 年 5 月，版本号为 0.6.2。随后开发人员为 Apache 设计全新的体系结构(包含模块化的结构以及基于池的内存分配器)，于 1995 年 8 月推出 Apache 0.8.8 版本，此时，Apache 成为 Internet 上排名第一的服务器。1995 年 12 月，Apache 的 1.0 版本正式推出。1996 年 4 月以后，Apache 一直是 Internet 上最流行的 HTTP 服务器。1996 年 7 月，Apache 1.1 发布，包含对 HTTP/1.1 以及 CGI 脚本的支持是随后发布的 Apache 1.2 最重要的扩展。

注意　有兴趣的朋友还可以到 http://www.netcraft.com 去查看 Apache 最新的市场份额占有率，同时还可以查询某个站点使用的服务器，如图 4-5 所示。

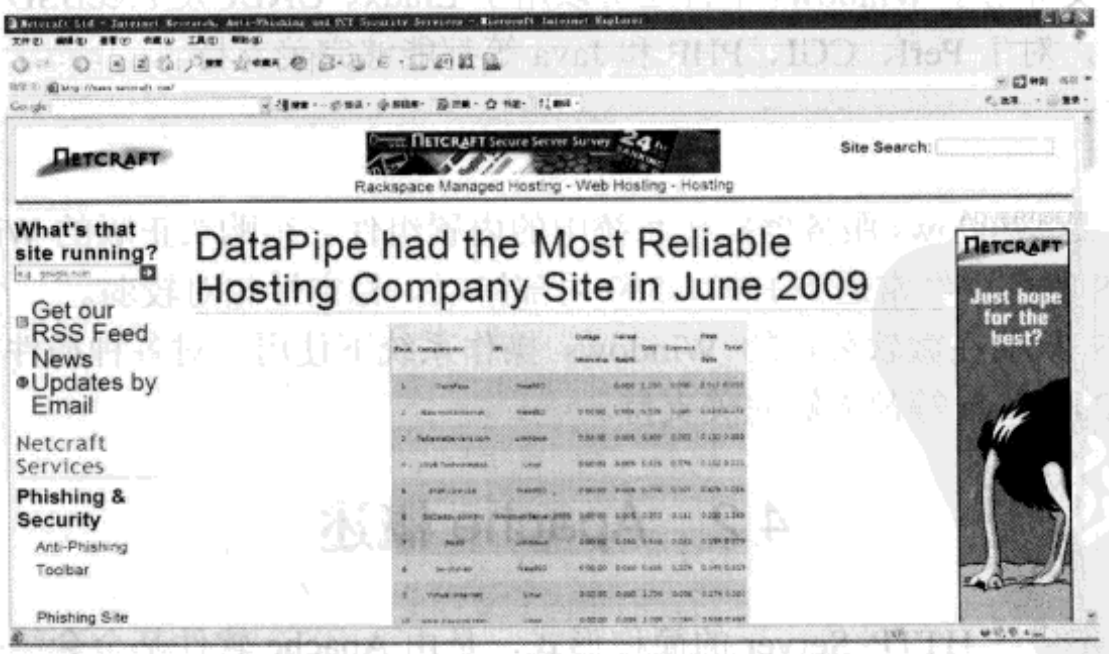


图 4-5　Apache 最新的市场份额占有率

3. Apache 服务器的特点

Apache 功能强大性能可靠，且具有跨平台特性和良好的安全性，Apache 服务器有如下主要特征。

- 支持 HTTP/1.1 协议。Apache 是最先使用 HTTP/1.1 协议的 Web 服务器之一，完全兼容 HTTP/1.1 协议并向后兼容 HTTP/1.0 协议。
- 跨平台，可以运行在几乎所有的计算机平台上。
- 支持通用网关接口（CGI）。Apache 用 mod_cgi 模块来支持 CGI，遵守 CGI/1.1 标准并且提供了扩充的特征。
- 支持 HTTP 认证。Apache 支持基于 Web 的基本认证，还为支持基于消息摘要的认证做好了准备。
- 支持虚拟主机。即通过在一台机器上使用不同的主机名来提供多个 HTTP 服务。Apache 支持包括基于 IP、名字和 Port 3 种类型的虚拟主机服务。
- 支持 CGI 脚本，如 Perl、PHP 等。
- 支持用户 Web 目录。Apache 允许主机上的用户使用特定的目录存放用户自己的主页。
- 集成的代理 Proxy 服务器。Apache 可作为前向代理服务器也可作为后向代理服务器。
- 简单而且强有力地基于文件的配置（httpd.conf）。
- 具有用户会话过程的跟踪能力。
- 支持服务器包含命令（SSI）。
- 支持安全 socket 层(ssl)。
- 支持 FastCGI。
- 支持 Java Servlets。
- 支持多进程。
- 支持动态共享对象。Apache 的模块可在运行时动态加载，模块可以被装入服务器进程空间，从而减少系统的内存开销。
- 可以通过 web 浏览器监视服务器的状态，同时可以自定义日志。

4.2.2 安装 Apache 服务

要通过 Apache 服务实现某些功能，就需要先安装该服务。接下来将具体讲解 Apache 服务具体的安装方法，安装该服务前应该做好相应准备。

1. 安装前的准备

在 Red Hat Enterprise Linux 5 中，Apache 服务的名称为 httpd，因此在安装之前应查询系统中是否已经安装了该服务，可以通过执行以下命令实现此功能。

```
[root@localhost ~]#rpm -q httpd
```

成功执行指令后的结果如下所示，表明该系统已经安装有此服务。

```
[root@localhost ~]# rpm -q httpd
httpd-2.2.3-11.el5_1.3
```

如果执行命令后的显示结果为“package httpd is not installed”，表明 httpd 服务没有安装，此时就需要准备相应的软件安装包来进行安装。

Red Hat Enterprise Linux 5 中的 httpd 服务的 RPM 安装包文件是 httpd-2.2.3-11.e15_1.3.i386.rpm，其帮助文件为 httpd-manual-2.2.3-11.e15_1.3.i386.rpm，这些安装包文件可以通过挂载安装光盘在光盘的 Server 目录下获得，也可以到相关网站下载获取（http://httpd.apache.org/），如图 4-6 所示。在必要情况下，还需准备与其相依赖的软件包。

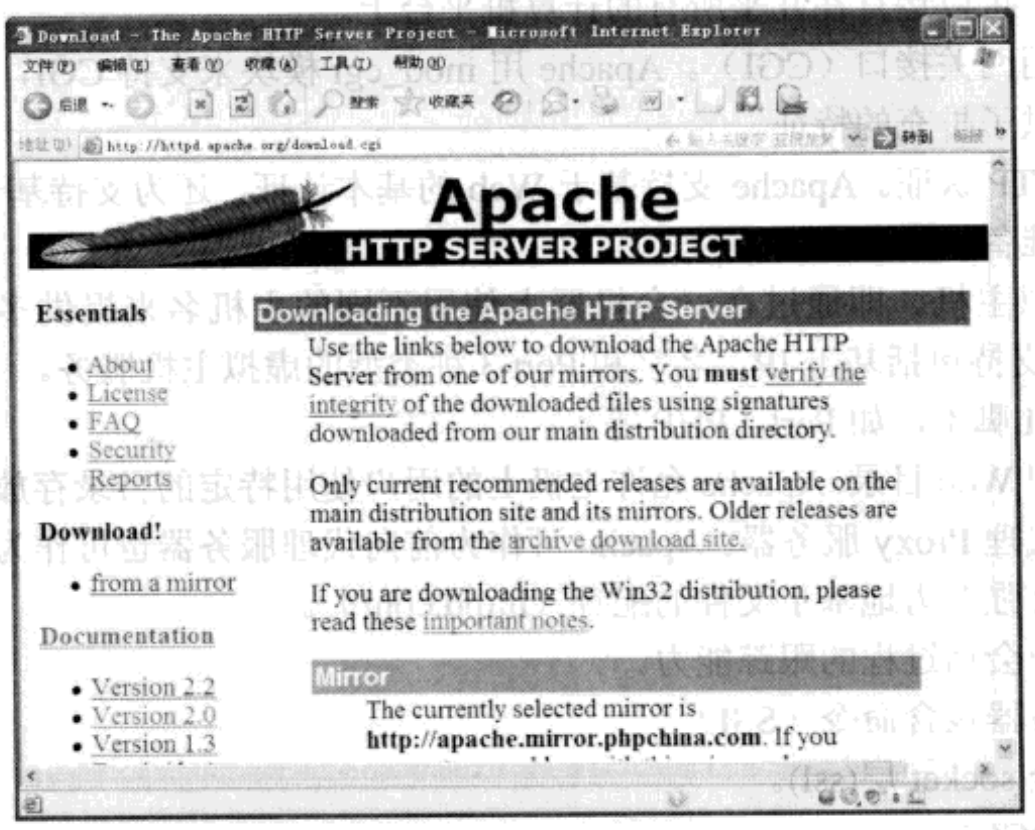


图 4-6　下载 httpd 服务的 RPM 安装包

2. 安装 httpd 服务

一切准备就绪后，将进入 httpd 服务安装阶段，可以使用以下命令进行安装 httpd 服务。

```
[root@localhost ~]#rpm -ivh
/mnt/cdrom/Server/postgresql-libs-8.1.11-1.el5_1.1.i386.rpm
[root@localhost ~]#rpm -ivh /mnt/cdrom/Server/apr-1.2.7-11.i386.rpm
[root@localhost ~]#rpm -ivh /mnt/cdrom/Server/apr-util-1.2.7-7.el5.i386.rpm
[root@localhost ~]#rpm -ivh
/mnt/cdrom/Server/httpd-2.2.3-11.el5_1.3.i386.rpm
[root@localhost ~]#rpm -ivh
/mnt/cdrom/Server/httpd-manual-2.2.3-11.el5_1.3.i386.rpm
```

注意　在有依赖关系的情况下，则需要根据提示先安装有依赖关系的软件包（有时不止一个）。

在安装完 httpd 服务后，需要测试其是否安装成功，具体方法请参见第 2 章 FTP 服务。

3. 为 Apache 服务放行

具体方法请参见第 2 章 FTP 服务。

4. Apache 服务的升级

Apache 为开源软件，因此时常会有新的功能添加以及服务器性能的不断优化。如果安装 Apache 服务器的机器能正常联网，就能方便地进行版本升级与更新，具体方法如下。

第 1 步，到 Apache 的官方网站 http://archive.apache.org/dist/httpd/binaries/rpm/i386/下载新版 Apache 服务器的 RPM 安装包，如图 4-7 所示。

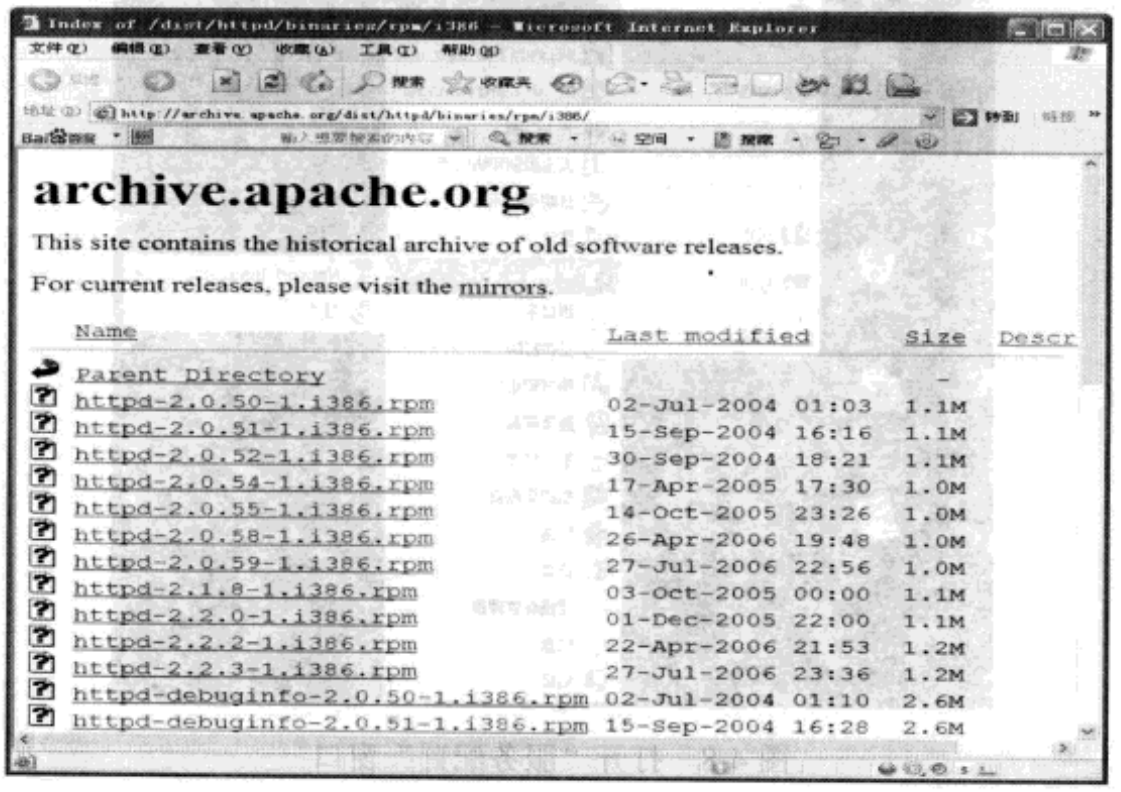


图 4-7 Apache 各版本的 RPM 安装包

第 2 步，Apache 服务器的更新，需执行以下命令。

```
[root@localhost ~]#rpm -force -nodeps -U 最新版本 RPM 安装包
```

第 3 步，使用如下命令查看当前 Apache 服务器的版本。

```
[root@localhost ~]#rpm -q httpd
```

如果安装正确，则会显示相应的 Apache 服务器版本。

4.3 启动与停止 Apache 服务

在安装完成 Apache 服务后，要实现 Apache 服务的启动，可以通过以下两种方式完成。

1. 命令方式

要启动 Apache 服务，需执行以下命令。

```
[root@localhost ~]#service httpd start
```

执行命令后的结果如下所示，表明 httpd 服务已经启动。

```
[root@localhost ~]# service httpd start
启动 httpd:                                                [确定]
```

2. 图形化方式

以图形化方式启动 Apache 服务的具体步骤如下。

第 1 步，在图形界面下，依次单击“系统”菜单→“管理”→“服务器设置”→“服务”菜单命令，如图 4-8 所示，即可打开“服务配置”窗口。



图 4-8 打开“服务配置”窗口

第 2 步，在该窗口中，左侧栏显示的是相应的服务，右侧栏显示的是相应的描述和状态信息，如图 4-9 所示。此时，在左侧栏中找到并选中 httpd 复选框，然后单击其上方的“开始”按钮。

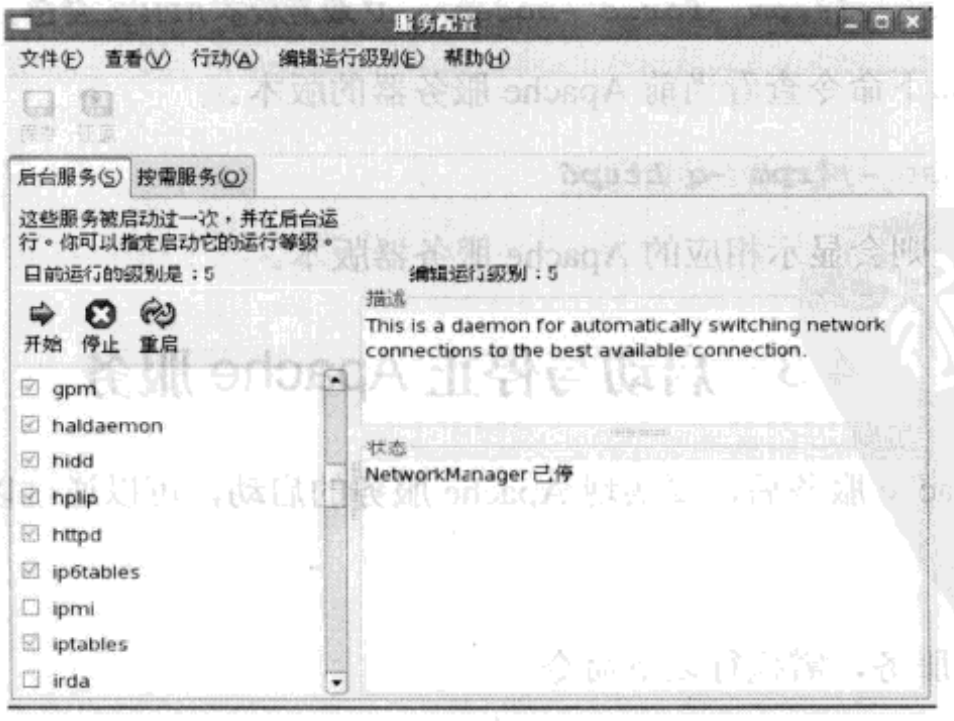


图 4-9 “服务配置”窗口

第 3 步，单击“开始”按钮后，会出现如图 4-10 所示的信息提示对话框，该对话框中的 httpd 状态标识 Apache HTTP 服务器正在运行。

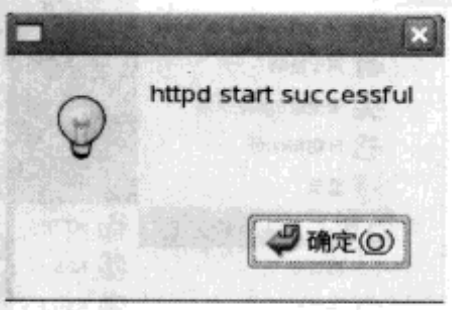


图 4-10　Apache HTTP 服务器启动成功

重启 Apache 服务的方法与启动 Apache 服务类似，同样有命令行和图形界面两种方式，具体操作也差不多，重启 Apache 服务的图形化方式，只是在选择 httpd 服务后，单击“重启”按钮即可。

利用命令行的方式来重启 Apache 服务，只需输入以下命令。

```
[root@localhost ~]#service httpd restart
```

执行命令后的结果如下所示，表明 Apache 服务重启成功。

```
[root@localhost ~]# service httpd restart
停止 httpd:                                                [确定]
启动 httpd:                                                [确定]
```

停止 Apache 服务的方法也有命令行和图形界面两种方式，图形化方式停止 Apache 服务的操作与启动 Apache 服务的操作类似，只是在选择 httpd 服务后，单击“停止”按钮即可。下面介绍一下如何利用命令行方式来停止 Apache 服务，要利用命令行方式停止 Apache 服务，则需执行以下命令。

```
[root@localhost ~]#service httpd stop
```

停止 Apache 服务的图形化方式同启动 Apache 服务类似，执行命令后的结果如下所示，表明 httpd 服务已经停止。

```
[root@localhost ~]# service httpd stop
停止 httpd:                                                [确定]
```

使用上述方法启动 Apache 服务，只能运行到计算机关机之前，重新启动系统后需要重新启动该服务。因此，为了方便操作，可以设置让其随系统的启动而运行，具体步骤如下。

第 1 步，依次单击“系统”→“管理”→“服务器设置”→“服务”菜单命令，如图 4-11 所示，即可显示“服务配置”窗口。

第 2 步，在该窗口的“后台服务”选项组找到 httpd 复选框并选中，如图 4-12 所示。

第 3 步，关闭该对话框，在关闭该对话框时，会弹出询问是否保存更改的对话框，单击“保存”按钮进行保存，如图 4-13 所示。

通过以上设置，Apache 服务就会随系统启动而自动运行了。

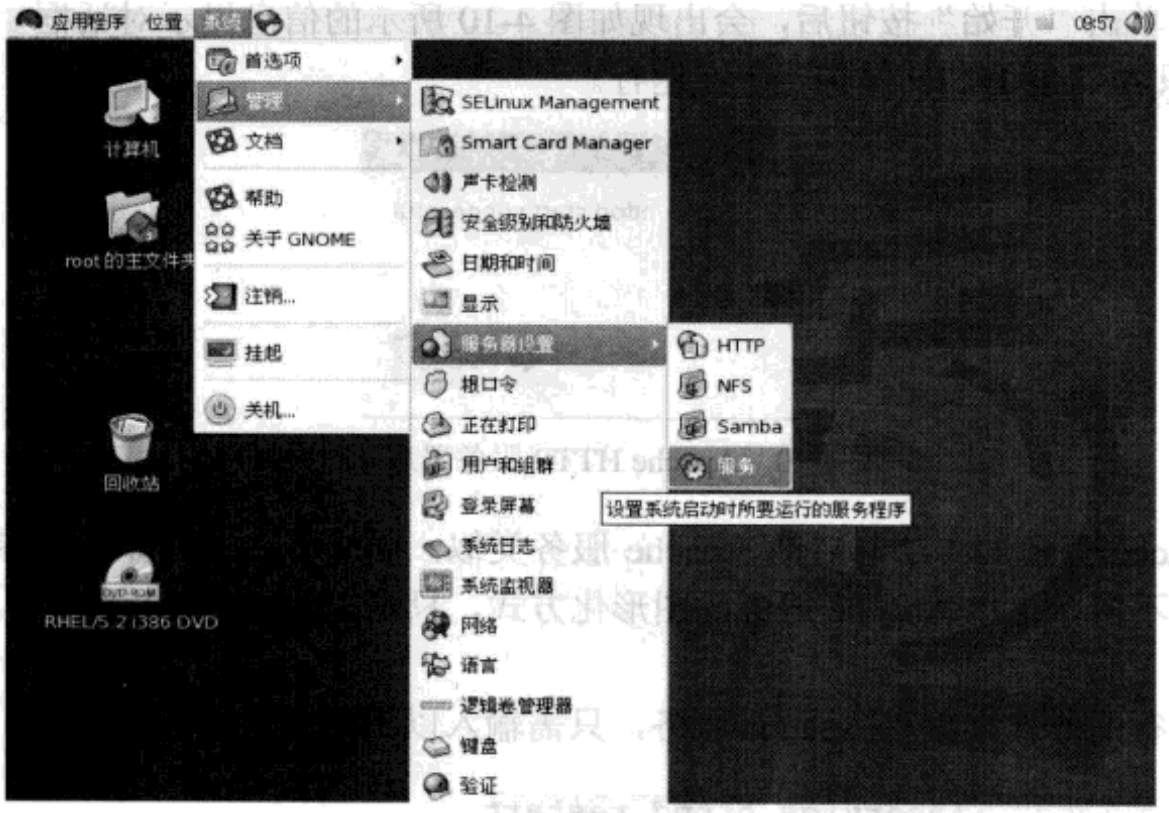


图 4-11　打开“服务配置”窗口

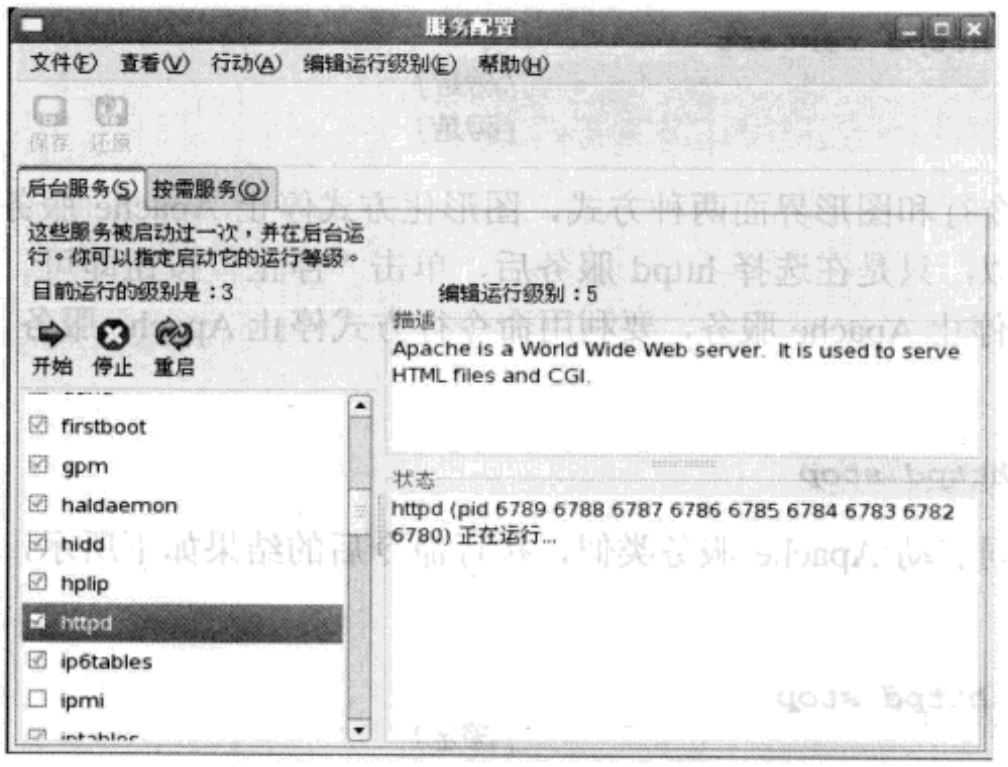


图 4-12　设置 Apache 服务自动运行

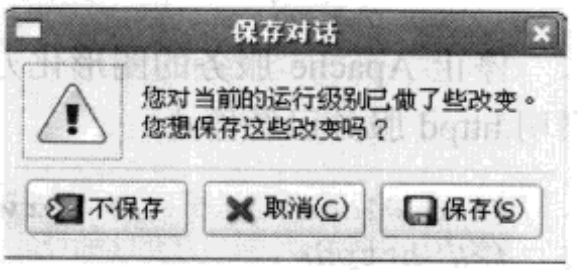


图 4-13　保存设置

4.4　Apache 服务的基本配置方法

Apache 服务具有强大的功能，且性能与扩展性相对于 IIS 而言都比较良好，但 Apache 服务器的配置也稍显复杂。在本节，将具体介绍 Apache 服务器的基本配置方法。

4.4.1　httpd.conf 配置文件

在/etc/httpd 目录下有 conf、conf.d、logs、modules 及 run 几个目录，如下所示。

```
[root@localhost ~]# cd /etc/httpd
[root@localhost httpd]# pwd
/etc/httpd
[root@localhost httpd]# ls -l
总计 28
drwxr-xr-x 2 root root 4096 04-24 22:23 conf
drwxr-xr-x 2 root root 4096 04-24 22:24 conf.d
lrwxrwxrwx 1 root root   19 04-24 22:23 logs -> ../../var/log/httpd
lrwxrwxrwx 1 root root   27 04-24 22:23 modules -> ../../usr/lib/httpd/modules
lrwxrwxrwx 1 root root   13 04-24 22:23 run -> ../../var/run
```

而在 conf 目录下存在 httpd.conf 和 magic 两个文件，其中与 Apache 服务器相关的配置，大多数都是通过配置 httpd.conf 文件来完成的。

注意 文件/etc/hpptd/conf/httpd.conf 是 Apache 的主配置文件。httpd.conf 文件中每行包含一条语句，可以使用反斜杠“\”在行末换行，反斜杠与下一行之间不能出现任何其他字符（包括空白），可以在每一行前用“#”号表示注释。httpd.conf 的配置语句除了选项的参数值以外，所有选项指令均不区分大、小写。

4.4.2 Apache 服务主配置文件的构成

Apache 的主配置文件/etc/hpptd/conf/httpd.conf 共分为 Apache 全局配置（Global Environment）、主服务器配置（Main Server Configuration）、虚拟主机配置（Virtual Hosts）3 部分。

- Apache 全局配置

主要设置全局环境变量。

- 主服务器配置

设置主服务器。

- 虚拟主机配置

设置虚拟主机。

下面将针对这 3 部分进行具体的介绍。

注意 可以通过执行“find / -name httpd.conf”命令搜索配置文件的具体路径。可以通过修改 httpd.conf 配置文件中的各个指令完成对 Apache 服务器的配置。

1. Apache 全局配置（Global Environment）

Global Environment（Apache 全局配置）主要用于设置与 Apache 服务器运行相关的全局环境变量。

（1） Apache 根目录

ServerRoot "/etc/httpd"为 Apache 的根目录。包括配置文件、记录文件及模块文件，都在该

目录下。

（2） 保存 Apache 父进程 ID

PidFile run/httpd.pid 文件保存着 Apache 父进程的 ID。

（3） 超时时间的设定

Timeout 100 表示超时时间。如果客户端超过 100 s 还没有连接上 Server，或者 Server 超过 100 s 还没有传送信息给客户端，则强制断线。

（4） 客户端同时提出多个请求的设置

KeepAlive On 表示允许客户端同时提出多个请求，设为 Off 则表示不允许。

（5） 每次联系允许的最大请求数目

MaxKeepAliveRequests 50 表示每次联系允许的最大请求数目，数字越大，效率越高，0 表示不限制。

（6） 用户端发出请求短线设置

KeepAliveTimeout 18 表示用户端的请求如果超过 18 s 还没有发出，则断线。

（7） MinSpareServers/ MaxSpareServers

MinSpareServers 10 表示最少会有 10 个闲置 httpd 进程来监听用户的请求。如果实际的闲置数目小于 10，则会增加 httpd 进程。

MaxSpareServers 30 表示最大的闲置 httpd 进程数为 30。如果你的网站访问量很大，可将这个数目设置得大一些。

（8） 启动时打开的 httpd 进程数目

StartServers 10 表示启动时打开的 httpd 进程数目为 10。

（9） 限制客户端的同时最大连接数目

MaxClients 200 表示限制客户端的同时最大连接数目为 200。一旦达到此数目，客户端就会收到用户太多或拒绝访问的错误提示。该数目不应该设置得太小。

（10） 限制每个 httpd 进程可以完成的最大任务数

MaxRequestsPerChild 4000 表示限制每个 httpd 进程可以完成的最大任务数目为 4 000。

（11） 设置 Apache 服务的监听端口

```
Listen 12.34.56.78:80
Listen 80
```

设置 Apache 服务的监听端口。一般在不使用 80 端口时设置。

（12） 加载 DSO 模块

LoadModule auth_basic_module modules/mod_auth_basic.so 表示加载 DSO 模块。DSO（Dynamic Shared Object，动态共享对象）很像 Windows 的 DLL（Dynamic Link Library，动态链接库）。

（13） 检测 Apache 的状态信息

ExtendedStatus Off 用于检测 Apache 的状态信息。

（14） 设置 Apache 工作时使用的用户和组

```
User apache
Group apache
```

表示设置 Apache 工作时使用的用户和组。

2. 主服务器配置（Main Server Configuration）

Main Server Configuration（主服务器配置）主要用于配置 Apache 的主服务器。

（1） 配置管理员的电子邮件地址

ServerAdmin 指令可以用于配置 WWW 服务器管理员的电子邮件地址，当服务器发生问题的时候，Apache 会自动向这个邮箱发送相关的邮件给管理员。ServerAdmin 指令的具体形式如下。

```
ServerName  完整的域名:端口号
如，ServerAdmin root@localhost
```

（2） 设置主机的名称

ServerName 指令指定了 Apache 服务器用于识别自身的名字和端口号。ServerName 指令的语法规则如下。

```
ServerName  完整的域名:端口号
```

如 ServerName www.XXX.com:80，其中端口号是可选的，可以不指定。

通常情况下，并不需要设定 ServerName，Apache 服务器将自动通过 DNS 来获得自身的名字。当服务器没有正式的 DNS 名字时，可以通过 ServerName 指令指定服务器所在的 IP 地址。

注意 ServerName 设置不正确将导致 Apache 服务器不能正常启动。

（3） 网页存放目录

DocumentRoot 指令指定了主机中网页文件的根目录，其语法规则如下。

```
Document  目录路径
```

如 Document /var/www/html，表示主机根目录为/var/www/html，Apache 服务器根据 DocumentRoot 指令指定的根目录，将客户端 HTTP 请求中的 URL 地址转换为对应文件在主机上的真实路径。

（4） 根据文件的内容来判断文件的类型

Web 服务器可以通过使用后缀名 MIME 类型映射来对不同的文件类型做不同的处理。除此之外，Apache 服务器还可以通过分析文件的内容来判断文件的具体类型（因为大多数文件内部都有一些特定的标记），这种方式称为 magic 方式（比如 BMP 文件最开始的两个字节就是“BM”，BM 就称为 BMP 文件的 magic）。

mod_mime_magic.c 模块可以使 Apache 由文件内容决定其 MIME 类型。只有载入了 mod_mime_magic.c 模块，才能处理 MIMEMagicFile 文件声明。MIMEMagicFile 指令用于指定 magic 标志文件的位置。MIMEMagicFile 指令的语法为：MIMEMagicFile magic 文件名。

```
<IfModule mod_mime_magic.c>
    MIMEMagicFile conf/magic
</IfModule>
```

（5） 设置预设首页

```
DirectoryIndex index.html index.html.var
```

设置预设首页，默认是 index.html。设置以后，用户通过“http://服务器 IP 地址：端口/”地址访问的其实就是“http://服务器 IP 地址：端口/index.html”。

（6） 设置 Apache 目录访问权限的控制文件

AccessFileName .htaccess

设置 Apache 目录访问权限的控制文件，预设是.htaccess，也可以是其他名字。

（7） 记录警告及其以上的信息

LogLevel warn 主要用于指定警告及其以上的信息，各等级如下。

```
debug：Debug 信息
info：一般信息
notice：通知信息
warn：警告信息
error：错误信息
crit：致命错误
alert：马上需要处理的信息
emerg：系统即将死机
```

（8） 设置 Apache 根目录的访问权限和访问方式

```
<Directory />
Options FollowSymLinks
AllowOverride None
</Directory>
```

（9） 设置 Apache 主服务器网页文件存放目录的访问权限

```
<Directory "/var/www/html">
Options Indexes FollowSymLinks
AllowOverride None
Order allow,deny
Allow from all
</Directory>
```

Apache 主服务器配置的具体情况如表 4-2 所示。

表 4-2 Apache 主服务器配置

设 置 内 容	说 明
<IfModule mod_userdir.c> UserDir disable #UserDir aa_html </IfModule>	表示设置用户是否可以在自己的目录下建立 aa_html 目录来放置网页。如果设置为“UserDir aa_html”，用户就可以通过：“http://服务器 IP 地址：端口/~用户名称”来访问其中的内容
<Files ~ "^\.ht"> Order allow,deny Deny from all </Files>	防止用户看到以“.ht”开头的文件，保护.htaccess、.htpasswd 的内容。主要是为了防止其他人看到预设访问相关内容的用户名和密码
TypesConfig /etc/mime.types	指定存放 MIME 文件类型的文件。可以自行编辑 mime.types 文件

（续表）

设 置 内 容	说　明
DefaultType text/plain	当 Apache 不认识某种文件类型时，自动将该文件当成文本文件处理
HostnameLookups Off	如果设置为 On，则每次都会向 DNS 服务器要求解析该 IP，这样会花费额外的服务器资源，并且降低服务器端的响应速度，所以一般设置为 Off
ErrorLog logs/error_log	指定错误发生时记录文件的位置。对于在<VirtualHost>段特别指定的虚拟主机来说，本处声明会被忽略
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined LogFormat "%h %l %u %t \"%r\" %>s %b" common LogFormat "%{Referer}i -> %U" referer LogFormat "%{User-agent}i" agent	设置记录文件存放信息的格式。自定义 4 种：combined、common、referer 和 agent
CustomLog logs/access_log combined	设置存取文件记录采用 combined 模式
ServerSignature On	设置为 On 时，Server 出错所产生的网页会显示 Apache 的版本号、主机、连接端口等信息；如果设置为 EMail，则会有“mailto:”的超链接
Alias /icons/ "/var/www/icons/" <Directory "/var/www/icons"> Options Indexes MultiViews AllowOverride None Order allow,deny Allow from all </Directory>	定义一个图标虚拟目录，并设置访问权限
ScriptAlias /cgi-bin/ "/var/www/cgi-bin/" <Directory "/var/www/cgi-bin"> AllowOverride None Options None Order allow,deny Allow from all </Directory>	同 Alias，只不过设置的是脚本文件目录
IndexOptions FancyIndexing VersionSort NameWidth=* HTMLTable	采用更好看的带有格式的文件列表方式
AddIconByEncoding (CMP,/icons/compressed.gif) x-compress x-gzip AddIconByType (TXT,/icons/text.gif) text/* …… DefaultIcon /icons/unknown.gif	设置显示文件列表时，各种文件类型对应的图标显示
#AddDescription "GZIP compressed document" .gz #AddDescription "tar archive" .tar #AddDescription "GZIP compressed tar archive" .tgz	在显示文件列表时，各种文件后面显示的注释文件。其格式为：AddDescription "说明文字".文件类型
ReadmeName README.html HeaderName HEADER.html	显示文件清单时，分别在页面的最下端和上端显示的内容 IndexIgnore.?? * *~ *# HEADER* README* RCS CVS *,v *,t 忽略这些类型的文件时，在文件列表清单中不显示出来
DefaultLanguage nl	设置页面的默认语言
AddLanguage ca .ca …… AddLanguage zh-CN .zh-cn AddLanguage zh-TW .zh-tw	设置页面语言

（续表）

设 置 内 容	说 明
LanguagePriority en ca cs da de el eo es et fr he hr it ja ko ltz nl nn no pl pt pt-BR ru sv zh-CN zh-TW	设置页面语言的优先级
AddType application/x-compress .Z AddType application/x-gzip .gz .tgz	增加 MIME 的类型
AddType text/html .shtml AddOutputFilter INCLUDES .shtml	使用动态页面
#ErrorDocument 500 "The server made a boo boo." #ErrorDocument 404 /missing.html #ErrorDocument 404 "/cgi-bin/missing_handler.pl" #ErrorDocument 402 http://www.example.com/subscription_info.html	Apache 支持 3 种格式的错误信息显示方式：纯文本、内部链接和外部链接。其中，内部链接又包括 html 和 script 两种格式
BrowserMatch "Mozilla/2" nokeepalive BrowserMatch "MSIE 4\.0b2;" nokeepalive downgrade-1.0 force-response-1.0	如果浏览器符合这两种类型，则不提供 keepalive 支持
BrowserMatch "RealPlayer 4\.0" force-response-1.0 BrowserMatch "Java/1\.0" force-response-1.0 BrowserMatch "JDK/1\.0" force-response-1.0	如果浏览器是这 3 种类型，则使用“HTTP/1.0”回应

3. 虚拟主机配置（Virtual Hosts）

Apache 提供了多主机支持，对于主服务器之外的主机支持，主要是通过 VirtualHost（虚拟主机）来完成的。

（1） 设置虚拟主机的名字和监听端口

```
#NameVirtualHost *:80
```

（2） 虚拟主机的所有信息相关

```
#<VirtualHost *:80>
#    ServerAdmin webmaster@dummy-host.example.com
#    DocumentRoot /www/docs/dummy-host.example.com
#    ServerName dummy-host.example.com
#    ErrorLog logs/dummy-host.example.com-error_log
#    CustomLog logs/dummy-host.example.com-access_log common
#</VirtualHost>
```

主服务器中采用的很多控制指令都可以在这里使用。

4.5 Apache 服务器控制存取的方式

在 Apache 的配置文件/etc/httpd/conf/httpd.conf 中，有 4 种控制存取的方式：针对目录的<Directory></Directory> 形 式 、 针 对 文 件 的 <Files></Files> 形 式 、 针 对 位 置 的 <Location></Location>形式以及<Limit></Limit>形式。

在存取方式为<Directory></Directory>形式中，存在有 Options 选项。

```
<Directory />
    Options FollowSymLinks
```

```
    AllowOverride None
</Directory>
```

4.5.1 Options 选项

Options 选项用于定义目录的特性。其允许访问控制文件中可以使用 Options 参数定义目录的选项，可选项（预设值为 All）如表 4-3 所示。

表 4-3 Options 选项

选　项	说　明
All	允许所有动作，除了 MultiViews、IncludesNOEXEC 及 SymLinksifOwnerMatch
ExecCGI	允许执行 CGI
FollowSymLinks	允许符号链接到其他目录和文件
Includes	允许 SSI，5.5.5 小节将具体讲解
IncludesNOEXEC	允许 SSI，但 CGI 的#exec 和#include 除外
Indexes	允许显示文件列表
MultiViews	允许不同的显示方式，例如语言和图形
SymLinksifOwnerMatch	允许符号链接到其他目录，但必须是拥有者

注意 Options 可为多个，但所有的 Options 都必须在一行指定，如果通过多行来指定则以位于最后的为准。另外，如果想在现有的项目上进行修改，可以使用“+”和“-”号实现。

4.5.2 浏览权限的设置

可以通过分布式特定目录存取控制和整体存取控制两种方式对 Apache 访问权限进行设置。其中，前者主要在.htaccess 文件中进行，后者主要是通过结合<Directory>、<Location>等指令在其配置文件 httpd.conf 中直接进行。

1. 分布式特定目录存取控制（.htaccess 文件控制存取）

.htaccess 文件是一个访问控制文件，该文件主要用来配置相应目录的访问方法。默认配置情况下相应目录下的该文件不会被读取。

下面看看 AllowOverride 中的配置。

```
AllowOverride none
```

.htaccess 文件完全被忽略，要打开该文件，可以使用如下方法。

首先找到如下配置。

```
<Directory />
    Options FollowSymLinks
    AllowOverride None
</Directory>
```

然后将其中的“AllowOverride None”修改为 “AllowOverride AuthConfig”，修改后的配

置如下。

```
<Directory />
    Options FollowSymLinks
   AllowOverride AuthConfig
</Directory>
```

通过以上设置，即可在需要进行访问控制的目录下创建一个.htaccess 文件（文件前的“.”表明该文件是一个隐藏文件）。文件名也可以在 httpd.conf 中通过“AccessFileName. ToBeProtected”进行重新设置。

注意 如果用户在/home/stella/work/目录中新建了一个.hidden 文件。Apache 将不会直接读取该文件，而是从根目录下依次搜索该文件。此时，Apache 将会读取第一次搜索到的. hidden 文件。例如，在/home/stella/路径中有一个.hidden 文件，而在/home/stella/work/目录中也有一个.hidden 文件，则 Apache 会选择读取/home/stella/.hidden。

位于 httpd.conf 的相应目录中的 AllowOverride，主要是控制.htaccess 文件中允许进行的设置，其 Override 详细参数如表 4-4 所示。

表 4-4 Override 参数及其说明

Overrid 参数	参 数 描 述
All	允许以下所有功能
AuthConfig	允许 AuthDBMGroupFile、AuthDBMUserFile、AuthGroupFile、AuthName、AuthDigestRealmSeed、AuthType、AuthUserFile 及 Require
FileInfo	允许 AddEncoding、AddLanguage、AddType、DefaultType、ErrorDocument 及 LanguagePriority
Limit	能够以主机名称或者 IP 地址进行限制
Options	允许使用 Options
None	以下所有功能不被允许

2. 整体存取控制

除以上设置方式外，整体存取控制是对 Apache 访问权限进行设置的另外一种方式。该方式通过结合<Directory>、<Location>等指令在其配置文件 httpd.conf 中实现，下面将通过实例的方式进行介绍。

实例 1 限制访问所有以 P 开头的目录

该功能可以通过在<DirectoryMatch>指令中配置实现，将“/P”加入其中，指令如下。

```
<DirectoryMatch /P>
    Order deny,allow
    Deny from all
</DirectoryMatch >
```

实例 2 限制文档目录的存取权限

该功能可以通过<Directory>指令实现。Apache 服务器默认配置的主文档目录的存取权限如下。

```
<Directory "/var/www/html">
    Options Indexes FollowSymLinks
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>
```

在以上实例中出现的几个关键字：Order、Allow from 及 Deny from，都对大、小写不敏感。其中，Order 后面的 allow 和 deny 之间仅以“,”分隔，allow 和 deny 的先后顺序需根据用户需要进行安排。下面将对各个关键字做简要介绍。

- Allow from：允许访问的主机

该主机可以是 All（所有主机）、完整或者部分的域名（比如 localdomain.test）、完整的 IP 地址（比如 192.168.1.3）、网络和子网掩码（比如 192.168.0.0/255.255.255.0）或者 CIDR 格式（192.168.0.0/24）。

- Deny from：拒绝访问的主机

该主机可以是 All（所有主机）、完整或者部分的域名（比如 localdomain.test）、完整的 IP 地址（比如 192.168.0.3）、网络和子网掩码（比如 192.168.0.0/255.255.255.0）或者 CIDR 格式（192.168.0.0/24）。

- Order：设置 allow 和 deny 的先后顺序。

下面简要说明一下 allow 和 deny 的先后顺序关系，如果配置文件中出现了以下语句：

```
Order allow,deny
Allow from 192.168.1.4
Deny from all
```

原意只希望 192.168.1.4 这台主机可以访问，但 Order 的顺序是 allow 在先，deny 在后，所以最后还是拒绝了所有主机的访问权限。

下面将讲解如何通过其他主机查看 Apache 服务器的运行状态，要查看 Apache 服务器的运行状态，可以通过<Location>指令来实现，具体步骤如下。

第 1 步，在 Apache 的主配置文件/etc/httpd/conf/httpd.conf 中找到以下内容。

```
<Location /server-status>
    SetHandler server-status
    Order deny,allow
    Deny from all
    Allow from .example.com
</Location>
```

将其前面的注释去掉（“#”），并修改成允许客户机 192.168.1.4 访问。

```
<Location /server-status>
    SetHandler server-status
    Order deny,allow
    Deny from all
    Allow from 192.168.1.4
</Location>
```

第 2 步，为了使服务器可以显示自己的运行状态，以便客户端随时掌控，需将语句

ExtendedStatus On 前的注释去掉。

第 3 步 ，通过设置后，只有 192.168.1.4 这台主机才能通过 http:// 192.168.1.4/server-status 访问 Apache 的服务器状态。

4.6 Apache 服务的高级配置

如果需要 Apache 服务的功能得到深入的扩展，比如，让其支持虚拟目录、虚拟主机及用户身份认证等高级选项，这就需要对 Apache 进行适当的配置了，本节将详细介绍如何进行 Apache 服务的高级配置。

4.6.1 虚拟目录简介

在实际生活中，真实目录相对复杂不便于用户访问，而虚拟目录则是真实目录的映射，具有方便访问、安全性较高以及可以灵活扩展磁盘空间等特点。在 Apache 服务器中可以通过使用 Alias 指令（别名）或者 AliasMatch 指令来为真实的目录指定一个虚拟目录名，Apache 服务器会将对虚拟目录名下文件的访问请求转换为真实的目录路径。

1. 虚拟目录的特点

虚拟目录的特点如下。

- 方便访问

如果用户要访问一个较深的目录，使用正常目录访问将非常繁琐，此时使用别名将会非常方便。

- 安全性高

虚拟目录可以隐藏站点中真正的目录结构，防范黑客的破解。

- 变相扩展磁盘空间

当 Apache 文档所在的磁盘空间不足时，可以通过虚拟目录将某些内容定义到其他磁盘上。

2. 创建虚拟目录

创建虚拟目录可以使用 Alias 指令或 AliasMatch 指令。

（1） 使用 Alias 指令（别名）创建

Alias 的语法规则如下。

```
Alias  虚拟目录  真实目录
```

例如，Alias /music /resources/music 表示对于 www.xx.com/music 的访问就会解析成 /resources/music。

（2） 使用 AliasMatch 指令创建虚拟目录

AliasMatch 指令使用正则表达式指定虚拟目录名，其语法格式如下。

```
AliasMatch  正则表达式  真实的本机目录
```

3. 设置目录权限

创建虚拟目录并将服务器重新启动后，客户端暂时还是不能访问。此时需使用<Directory>容器来配置虚拟目录的目录权限，其语法格式如下。

```
<Directory 目录路径>
    ......
</Directory>
```

<Directory 目录路径>和</Directory>这对语句间封装的是具体的设置目录权限语句，这些语句仅对被设置目录及其子目录起作用。

注意 这里的目录路径必须是真实的主机目录路径。

和普通目录一样，对虚拟目录权限设定的所有指令可以放到<Directory>容器中，也可以单独放到目录访问控制文件中。

例如，为/download 设置目录权限，具体方法如下。

```
<Directory "/export/public/software">
    Options Indexes MultiViews
    AllowOverride None
    Order allow, deny
    Allow from all
</Directory>
```

4.6.2 配置与管理虚拟主机

由于 Apache 支持虚拟主机，因此可以让你在一台计算机上配置几个不同的 Web 站点，如此便可以达到节约计算机硬件资源的目的。

在一台计算机上部署多个网站可以通过两种方法实现。

方法 1 在一台计算机上运行多台 Web 服务器，每台 Web 服务器上部署一个网站，然后为每台 Web 服务器指定其监听的 IP 地址和端口。

注意 对于此种方法，每个站点必须运行在不同的 IP 地址上，这在很大程度上限制了一台计算机上所能部署的 Web 站点的数目。

方法 2 在一台 Web 服务器上部署多个网站，为每个网站指定不同的目录用于存放每个网站的文件，同时需要为不同的网站指定不同的 IP 地址和端口，或者指定不同的主机名称，此种方法最为常用。此时，可以把每个网站看成是一台虚拟主机。

Apache 支持的虚拟主机有两种类型：IP Based（基于 IP 地址的虚拟主机）和 Name Based（基于主机名的虚拟主机）。两者的不同之处在于，前者（基于 IP 的虚拟主机）是一个 IP 对应于一个站点；后者（基于主机名的服务器）允许一个 IP 对应多个站点。接下来将对这两种虚

拟主机进行简单的介绍。

1. 基于 IP 地址的虚拟主机（IP Based）

IP 地址的虚拟主机是一个 IP 对应于一个站点。在配置虚拟主机之前，需要为 Apache 服务器指定其监听的 IP 地址及端口，该设置同样需要在 httpd.conf 配置文件中进行。除了指定监听地址外，还需要使用<VirtualHost>容器来定义虚拟主机。<VirtualHost>容器的语法规则如下。

```
< VirtualHost IP 地址:[端口号] [IP 地址:[端口号]] ...>
    ......
</VirtualHost>
```

其中端口号为可选，但必须为虚拟主机指定一个 IP 地址。

接下来看一个基于 IP 的虚拟主机设置实例。

在一台服务器上分别配置两个 Web 站点，一个的 IP 地址是 192.168.1.2，另外一个 IP 地址是 192.168.1.4，具体操作步骤如下。

第 1 步，使用以下指令绑定 IP 地址 192.168.1.2。

```
[root@localhost ~]#/sbin/ifconfig eth0:0 192.168.1. 2 netmask 255.255.255.0
```

注意 绑定的前提是已经为服务器的网卡设置了 192.168.0.43 这个 IP 地址。

第 2 步，找到 httpd.conf，并添加如下内容。

```
<VirtualHost 192.168.1. 2>                                    ; #绑定第 1 台虚拟主机 IP
    ServerAdmin webmaster@testdomain.tst                      ; #设置管理员邮箱
    DocumentRoot /var/www/testdomain                          ; #设置虚拟主机文档路径
    ServerName www.testdomain.tst                             ; #设置主机名
    ErrorLog /var/log/httpd/www.testdomain.tst/error.log      ; #设置错误记录文档
</VirtualHost>
 <VirtualHost 192.168.1.4>                                    ; #绑定第 2 台虚拟主机 IP
    ServerAdmin webmaster@othertestdomain.tst                 ; #设置管理员邮箱
    DocumentRoot /var/www/othertestdomain                     ; #设置虚拟主机文档路径
    ServerName www.othertestdomain.tst                        ; #设置主机名
    ErrorLog /var/log/httpd/www.othertestdomain.tst/error.log; #设置错误记录文档
</VirtualHost>
```

注意 此种虚拟主机的缺点是会对资源造成很大的浪费，这是因为每台主机都必须有自己的公用 IP 地址。

2. 基于主机名的虚拟主机（Name Based）

基于主机名的虚拟主机的配置是在 Apache 的主配置文件/etc/http/conf/httpd.conf 中实现的。由于基于主机名的虚拟主机使用相同的 IP 地址和端口，因此必须使用 NameVirtualHost 指令指定一个 IP 地址，NameVirtualHost 指令的语法格式如下。

```
NameVirtualHost  IP地址:端口号
```

其中端口号为可选。

假设 IP 地址是 192.168.1.4 的服务器，要为 www.test1.tst 和 www.test2.tst 同时服务，则可以通过在 httpd.conf 中添加如下内容实现。

```
NameVirtualHost     192.168.1.4      ; #指定绑定的 IP 地址。如果想在服务器上的所 IP 地址上进行绑定，可以将 IP 地址该为“*”。
```

第一台虚拟主机配置如下。

```
<VirtualHost *>
ServerName www.test1.tst          ; #设置第一个虚拟主机名
ServerAlias test1.tst *.test1.tst
#通过 ServerName www.test1.tst 配置第一台虚拟主机后，只有在浏览器端输
#入 www.test1.tst 才能访问到服务器，为了输入 test1.tst 也能访问，就
#必须通过 ServerAlias 来定义其他域名
DocumentRoot /var/www/test1       ; #指定虚拟主机文档位置
</VirtualHost>
```

第二台虚拟主机的配置如下。

```
<VirtualHost *>
ServerName www.test2.tst          ; #设置第二台虚拟主机名
DocumentRoot /var/www/test2       ; #指定文档位置
</VirtualHost>
```

4.6.3 用户认证和授权

用户认证即是验证用户通过浏览器提供的用户名、密码是否与服务器上保持的用户信息一致。Apache 中的用户身份认证方法也可以采取“整体存取控制”、“分布式存取控制”两种方式，最常用的是.htaccess 方式，下面将具体讲解如何通过配置.htaccess 文件来实现用户认证。

1. 创建用户名和密码

位于/usr/local/httpd/bin 目录下的 htpasswd 可执行文件是用来创建.htaccess 文件身份认证使用的密码的。其语法格式为如下。

```
[root@localhost ~]#htpasswd [-bcD] [-mdps] 密码文件名字  用户名
```

具体参数如表 4-5 所示。

表 4-5 htpasswd 具体参数

参　数	说　明
-b	用批处理方式创建用户。htpasswd 令不会提示你输入用户密码。不过由于要在命令行输入可见的密码，并不是很安全
-c	新创建（create）一个密码文件
-D	删除一个用户
-m	采用 MD5 编码加密
-d	采用 CRYPT 编码加密，这是预设的方式
-p	采用明文格式的密码。因为安全的原因，目前不推荐使用
-s	采用 SHA 编码加密

实例 创建一个用于.htaccess 密码认证的用户 Neo。执行如下命令。

```
[root@localhost ~]#htpasswd -mb .htpasswd Neo 111111
```

该命令表示，在当前目录创建一个.htpasswd 文件，并添加了一个 Neo 用户，其密码为 111111。

2. 指定相应的认证方式

用户密码文件可以采用多种加密方式，而每种加密方式需要对应一种认证方式，下面将详细介绍这几种认证方式。

Apache 中对于默认的加密方式保存的密码文件，是这样设置加密的：

```
<Directory /home/clinuxer/public>
AuthType Basic
AuthName "Restricted Files"
AuthUserFile /usr/local/apache/passwd/.htpasswd
Require user clinuxer
</Directory>
```

AuthType Basic：指定采用的加密方式是 mod_auth 提供的 Basic。

注意 AuthType 指令用来指定用户认证的类型，Apache 提供基本认证（Basic）和摘要认证（Digest），但 Basic 认证方法并不对来自用户浏览器的密码进行加密，因此，不应该用于高度敏感的数据。

AuthName：AuthName 指令设置了使用认证的领域，具有两个作用：一是该说明会出现在给用户的密码提问对话框中；二是帮助客户端程序确定应该输入哪个密码。

AuthUserFile/AuthGroupFile：AuthUserFile 指令设置了密码文件，也就是刚才用户已经用 htpasswd 建立的文件。

注意 由于该密码文件是纯文本文件，因此用户反应速度会变慢，对此，Apache 还支持把用户信息存入快速数据库，mod_auth_dbm 模块提供了 AuthDBMUserFile 指令，并可以用 dbmmanage 程序建立和操作这些数据库。

AuthGroupFile 则是很多具有相同权限的用户组设置文件，它的语法格式如下。

```
UserGroup: fayero clinuxer
```

不同用户之间用空格分开，每个用户也需要单独通过 htpasswd 指令创建。

Require User / Require Group：前者设置了允许访问的用户，后者则指定了可以访问的群组。

（1） 摘要认证

使用 Apache 服务器提供的 htdigest 程序来生成和管理用户摘要文件。可以通过执行命令 htdigest –c /etc/httpd/digestuserfile “Information” ly 来生成用户摘要文件，并为“Information”

验证域（AuthName 指令的值）添加用户 ly。

如果要配置 Apache 服务器使用摘要认证方式，必须将 AuthType 指令设置为“Digest”。使用摘要认证方式时，如果还需要指明用户访问哪些 URL 时需要认证，就必须设置 AuthDigestDomain 指令。AuthDi gestDomain 的语法规则如下。

```
AuthDigestDomain  URL1  URL2…
```

其中的 URL 可以是绝对 URL（包含完成协议、主机、端口等）或者相对 URL。

注意 用户组文件的创建与基于文本形式的认证中组文件的创建方式相同。

（2） 基本认证

基本认证具有匿名认证、基于文件的认证和基于数据库的认证三种不同的方式。

● 匿名认证

在 Apache 服务器中，所有用户都可以通过 Anonymous 指令设置使用“anonymous”作为用户 ID，通过匿名方式登录进行访问。Anonymous 指令可以指定一个或多个用户名用于匿名访问。

Anonymous 指令的语法规则如下。

```
Anonymous  userName1  userName2 ….
```

该指令的具体用法如下。

如果用户名中有空格符，需要对这个用户名添加引号。

可以使用 Anonymous_NoUser 指令，设置为“Anonymous_NoUser on”，表明不用在提示对话框中输入任何用户名，直接单击对话框的确认按钮即可访问相关的目录。

可以使用 Anonymous_Authoritative 指令，设置为“Anonymous_Authoritative on”，表明匿名认证成为目录唯一的认证方式；如果将 Anonymous_Authoritative 指令设置为 off，表明匿名认证失败后会使用其他认证方式继续进行用户认证。

将 Anonymous_LogEmail 指令设置为“on”，表明 Apache 服务器将用户在密码栏输入的数据记录到访问日志中。

将 Anonymous_MustGiveEmail 指令设置为“on”，表明用户必须在密码栏中输入数据而不能使之空白。

● 基于文件的认证

大多数情况下只使用匿名认证是不安全的，实际上用户使用他的用户名和密码进行登录，此时将会涉及到基于文件的认证。

基于文件的认证方式中最重要的两个指令就是 AuthUserFile 指令和 AuthGroupFile 指令，这两者分别用来指明保存用户文件和用户组文件的存放路径。

这两个指令的语法格式如下。

```
AuthUserFile  userFilePath
AuthGroupFile  groupFilePath
```

- 基于数据库的认证

在用户数量较多的情况下，通常使用数据库进行用户和用户信息的管理。在数据库中使用用户表和组信息表两张表记录用户信息，并都以用户名作为键，用户口令和用户组名分别作为值与用户名关联。

（3） 具体实例

实例 1 如何设置一个虚拟目录“/doc/”，使用户必须输入用户名和密码才能访问，具体操作步骤如下。

第 1 步，创建用户。

执行以下指令来创建一个用户名为 Rose 的用户。

```
[root@localhost ~]# cd /hhwork/public/software
[root@localhost software]#/usr/bin/htpasswd -c
/usr/local/apache/passwd/.htpasswd Rose
```

执行命令后，提示输入该用户的密码并确认，输入相应信息成功后将出现提示“Adding password for user Rose”，表明创建用户成功！

```
[root@localhost ~]# cd /hhwork/public/software
[root@localhost software]#/usr/bin/htpasswd -c
/usr/local/apache/passwd/.htpasswd Rose
New password:
Re-type new password:
Adding password for user Rose
```

另外，在上面已经讲到，还可以在.htpasswd 文件中添加其他用户，具体命令如下。

```
[root@localhost software]#/usr/bin/htpasswd .htpasswd apple
```

第 2 步，在 httpd.con 文件中设置该目录允许采用.htaccess 进行用户身份验证。

首先，使用以下命令打开 httpd.con 文件。

```
[root@localhost ~]#vi /etc/httpd/conf/httpd.conf
```

打开 httpd.con 文件后，加入以下内容。

```
Alias /doc/ "/hhwork/public/software/"
<Directory /hhwork/public/software>
    Options Indexes MultiViews          ; #允许列目录
AllowOverride AuthConfig                ; #启用用户身份验证
Order deny,allow
Allow from all                          ; #许所有用户访问
</Directory>
```

重新设置了 Apache 的主配置文件 httpd.conf 后，需要重启 Apache 服务。这样新的配置才会生效。

第 3 步，在/hhwork/public/software 目录下新建一个具有以下内容的.htaccess 文件。

```
[root@localhost ~]#cd /hhwork/public/software
```

创建.htaccess 文件，执行以下命令。

```
[root@localhost software]#touch .htaccess
```

编辑.htaccess 文件并加入以下内容。

```
[root@localhost ~]#vi .htaccess
AuthName "Test excample"
AuthType Basic
AuthUserFile /usr/local/apache/passwd/.htpasswd
require user Rose
```

注意 在 AuthUserFile 中指定.htpasswd 文件的绝对路径是为了防止.htpasswd 文件不在默认的搜索路径中。

第 4 步，使用用户端浏览器访问 Apache 服务器上访问权限受限的目录，会出现提示输入正确的用户名和密码的认证窗口，只有当输入正确的用户名和密码后，才会看到受限内容。

实例 2 在需要授权或者限制访问的用户数目很多的情况下，对需要授权或者限制访问的用户进行分组授权。

第 1 步，在/usr/local/apache/passwd 目录下创建一个具有如下内容的.htgroup 文件。

```
allowusr: Rose stella
```

第 2 步，打开.htaccess 文件，修改其内容。

```
AuthName "Test Group Zone"
AuthType Basic
AuthUserFile /usr/local/apache/passwd/.htpasswd
AuthGroupFile /usr/local/apache/passwd/.htgroup
Require group allowusr
```

第 3 步，修改成功后，使用同组中的 Rose 和 stella 这两个用户，都可以访问 Apache 服务器上访问权限受限的目录了。

4.6.4 配置 SSI

SSI（Server Side Includes，服务器端包含指令）。在普通的 HTML 页面中嵌入 SSI 命令，可以实现对现有 HTML 文档增加动态内容。Web 服务器需要通过配置才能支持 SSI，下面就以 Apache 服务器为例具体讲解 SSI 的配置方法。

在配置文件 httpd.conf 或.htaccess 中给 Options 指令添加“+Includes”值，格式如下。

```
Options +includes
```

可以通过以下两种方法实现 Apache 服务器指定解析 HTML 文件。

方法 1 给包含 SSI 指令的 HTML 页面指定后缀名。

通常将包含 SSI 指令的 HTML 页面的后缀名指定为.shtml，此时可以通过 AddType 指令添加对应的 MIME 类型，并且使用 AddOutputFilter 指令来建立映射关系，具体设置如下。

```
AddType text/html .shtml
```

```
AddOutputFilter  INCLUDES  .shtml
```

注意 此种方法的缺点是，如果需要在现有的 HTML 文件中添加 SSI 指令，则需要变更 HTML 文件的后缀名。

方法 2 使用 XBitHack 指令。

使用 “XBitHack on” 命令告诉 Apache 服务器解析有执行权的 text/html 文件中的 SSI 指令。此时还需要设置包含 SSI 指令的 HTML 文件为可执行，具体命令如下。

```
chmod  +x  pagename.html
```

4.6.5 MIME 类型简介

MIME 即 Multipurpose Internet Mail Extensions（多用途互联网邮件扩展）的缩写形式。Web 服务器使用 MIME 类型将其发送的多媒体数据的类型告诉浏览器，使得浏览器知道具体数据的类型，然后浏览器对不同类型的数据进行解释，最后将信息呈现给用户。

常见的 MIME 类型有 text/html、text/plain 和 image/gif 等，其中 text/html 表示后缀名为.html 和.htm 的超文本标记语言文件；text/plain 表示后缀名为.txt 的普通文本文件；image/gif 表示后缀名为.gif 的图形文件。

在 Apache 服务器的配置文件中，TypesConfig 指令指定了服务器上存储 MIME 类型与文件后缀名对应关系的文件。

也可以在 Apache 服务器的配置文件 httpd.conf 中使用 AddType 指令添加 MIME 类型，或者覆盖默认文件中的 MIME 类型定义。AddType 指令的语法如下。

```
AddType  Type/SubType  extension
```

例如，为.tgz 包指定 MIME 类型的指令如下。

```
AddType  application/x-tar  .tgz。
```

在 HTTP 协议中，Web 浏览器接收到的是数据流，并不知道文件的后缀名，浏览器则是通过响应报文中的 Content-Type 标头来判断文件的 MIME 类型。所以 Web 服务器在传送 HTTP 响应报文时，必须设置 Content-Type 标头来告诉客户程序数据的 MIME 类型。

4.7 创建并配置动态网站

随着 Web 技术的迅猛发展，如今大多数的网站都向着动态的基于 C/S（客户端/服务器）模式的发展，动态网站不仅可以丰富网站的表现形式，最重要的是还增强了与用户之间的互动性。因此，在本节我们将学习如何使用 JSP 及 CGI 创建并配置动态网站。

4.7.1 创建并配置 JSP 动态网站

JSP（Java Server Pages）是由 Sun Microsystems 公司倡导、许多公司一起参与建立的动态

网页语言，它用 JSP 容器转化为 Servlet，然后再用 Javac 编译器将其编译为二进制文件，具有高效率的特点，如今已被广泛使用。要使用 JSP 创建动态网站，需要在机器上安装 JDK 以及 Tomcat，可以通过整合 Tomcat 与 Apache 来运行 JSP 程序并实现负载均衡。

1. 安装 JDK

对于 JSP 编写的网页，需要 JDK 对 JSP 页面进行编译和执行。执行以下命令即可确认是否已经安装了 JDK。

```
[root@localhost ~]# rpm -qa | grep jdk
```

如果结果显示没有安装 JDK，则需要先从 Sun 的官方网站（www.sun.com）下载相关的安装程序包，如图 4-14 所示。

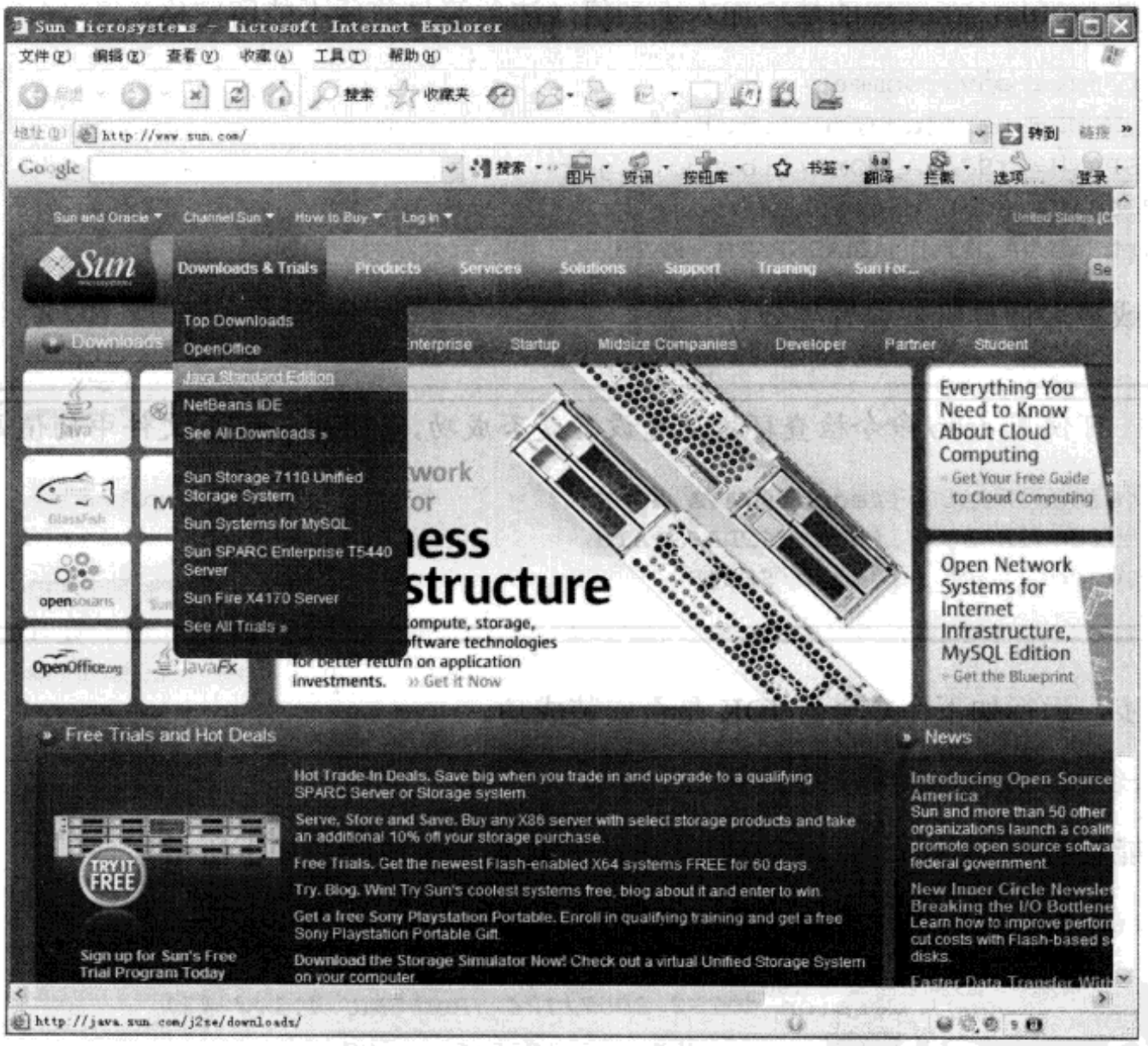


图 4-14 下载 JDK

本书使用的是 JDK 5.0，其具体安装步骤如下。

第 1 步，使用 chmod 命令对下载的 jdk-1_5_0_14-linux-i586-rpm.bin 文件赋予可执行权。

```
[root@localhost ~]# chmod +x jdk-1_5_0_14-linux-i586-rpm.bin
```

第 2 步，执行 jdk-1_5_0_14-linux-i586-rpm.bin 文件，得到 JDK 的 RPM 安装包 jdk-1_5_0_14-linux-i586.rpm。

```
[root@localhost ~]#./jdk-1_5_0_14-linux-i586-rpm.bin
```

第 3 步，需要对 jdk-1_5_0_14-linux-i586.rpm 包赋予执行权，命令如下。

```
[root@localhost ~]# chmod  +x  jdk-1_5_0_14-linux-i586.rpm
```

第 4 步，安装 jdk-1_5_0_14-linux-i586.rpm 包，命令如下。

```
[root@localhost ~]# rpm  -ivh  jdk-1_5_0_14-linux-i586.rpm
```

第 5 步，在正式安装之前会要求接受许可证协议，输入“y”即可完成 JDK 的安装。

JDK 的默认安装路径是/usr/java/jdk1.5.0_14，如果不是该路径，请使用 find / -name jdk 命令来确定 JDK 的安装路径。

第 6 步，安装完成后必须设置正确的环境变量才能使用 JDK。进入/etc 目录，用文本编辑器打开文件 profile，在文件的最后加入下列值（注意等号前后不能留空格）。

```
# set java environment
JAVA_HOME=/usr/java/jdk1.5.0_14
CLASSPATH=.:$JAVA_HOME/lib.tools.jar
PATH=$JAVA_HOME/bin:$PATH
export  JAVA_HOME  CLASSPATH  PATH
```

注销或重启计算机使设置生效。

注意 可使用 echo 命令检查环境变量设置是否成功，与上述 profile 文件中的值是否一致。

```
[root@localhost ~]#echo $JAVA_HOME
[root@localhost ~]#echo $CLASSPATH
[root@localhost ~]#echo $PATH
```

第 7 步，运行如下命令查看 JDK 是否安装成功。

```
[root@localhost ~]#java  -version
```

如果能够看到 JVM 版本及相关信息则说明 JDK 安装成功，如下所示。

```
[root@localhost ~]# java  -version
java version "1.4.2"
gij (GNU libgcj) version 4.1.2 20071124 (Red Hat 4.1.2-42)
Copyright (C) 2006 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

2. Tomcat 服务器

Tomcat 是一款支持 JSP 的开源代码服务器软件，由 Apache-Jakarta 子项目支持开发。其也是 Red Hat Enterprise Linux 5 内置的一款服务器软件，得到 Sun 公司的全力支持。Tomcat 具有小巧、运行效率高的特点，使其得到了广泛运用。如果需要查询自己的机器是否已经安装了

Tomcat 服务，则可以使用如下指令。

```
[root@localhost ~]#rpm -q tomcat5
```

执行命令后的结果如下所示，表明没有安装 Tomcat 服务。

```
[root@localhost ~]# rpm -q tomcat5
package tomcat5 is not installed
```

如果提示没有安装 Tomcat 服务，则可以进行相应的安装。因为 Tomcat 服务关联的程序非常多，安装需要大量的时间和精力，并且很容易出错，所以建议使用 yum 工具进行安装，具体安装步骤如下。

建议大家采用定制 yum 的方式从本地安装 yum。默认情况下 yum 会从 Red Hat 公司的服务器上下载程序并安装，且有订阅号的用户才能使用，下载速度也极其缓慢。

第 1 步，切换目录到/etc/yum.repos.d 下，然后新创建一个具有如下内容的文本文件 rhel5.localinstall.repo。

```
[rhel-localinstall]
name= Red Hat Enterprise Linux 5.2 LocalInstall
baseurl=file:///mnt/cdrom/Server/
gpgcheck=1
enabled=0
gpgkey=file:///mnt/cdrom/RPM-GPG-KEY-Red Hat-release
```

其中，出现了 baseurl 和 enabled 两个参数。这两个参数的具体说明如表 4-6 所示。

表 4-6 baseurl 和 enabled 参数说明

参　数	说　明
baseurl	用于指定 Red Hat Enterprise Linux 5 安装光盘的 Server 目录位置
enabled=0	打开“添加/卸载软件”时不从这里查找软件包

第 2 步，使用如下指令，将 Red Hat Enterprise Linux 5 的安装光盘加载到/mnt/cdrom，执行命令。

```
[root@localhost ~]#mkdir /mnt/cdrom
[root@localhost ~]#mount /dev/cdrom /mnt/cdrom
```

第 3 步，导入 Public key，执行以下指令。

```
[root@localhost ~]#rpm -import /mnt/cdrom/RPM-GPG-KEY-Red Hat-beta
[root@localhost ~]# rpm -import /mnt/cdrom/RPM-GPG-KEY-Red Hat-release
```

第 4 步，在 Red Hat Enterprise Linux 5 的桌面上，依次单击“应用程序”→“添加/删除软件”菜单命令，显示“软件包管理者”窗口。在该窗口一次单击“编辑”→“软件存储库”菜单命令，然后选中前面定制的“rhel-localinstall”，最后单击“关闭”按钮退出，如图 4-15 所示。

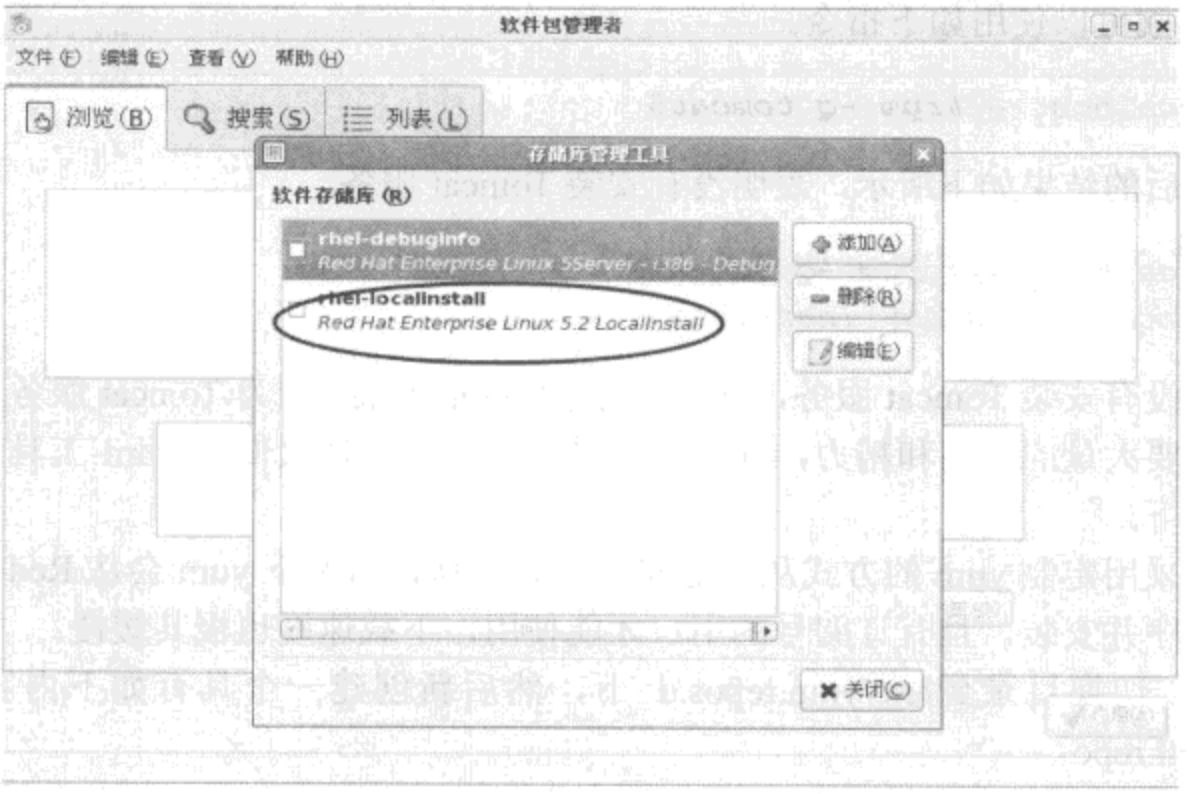


图 4-15　前面定制的软件安装源

第 5 步，通过以上设置后，在“软件包管理者”窗口的“浏览”选项页下，就可以看到大量可以安装、卸载的软件，如图 4-16 所示。

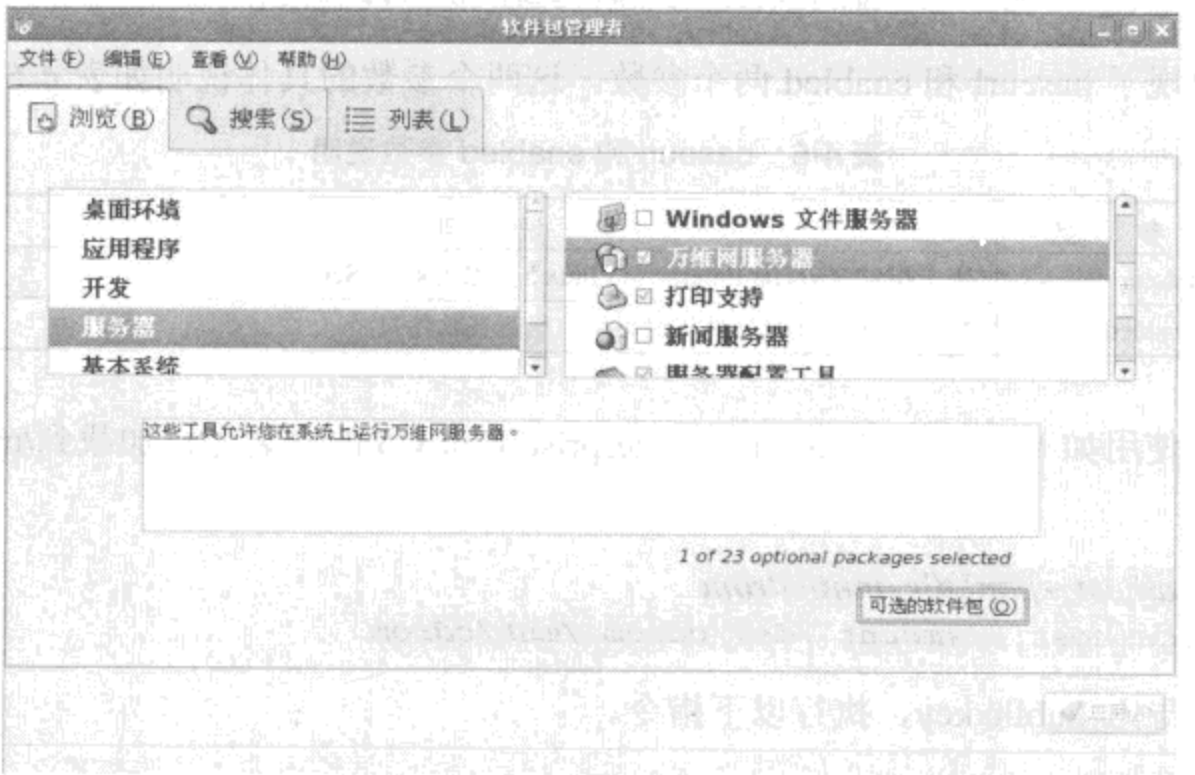


图 4-16　显示光盘上所有的软件

第 6 步，在窗口左侧选择“服务器”选项，在右侧选择“万维网服务器”选项，并单击“可选的软件包”按钮，在弹出的“万维网服务器中的软件包”对话框中，Tomcat 的相关选项将清晰地呈现在用户面前。此时将 Tomcat5-5.5.23-0jpp.7.e15.i386、Tomcat5-admin-webapps 和 Tomcat5-webapps 选中，如图 4-17 所示。最后依次单击“关闭”→“应用”按钮。

第 7 步，在“添加所需的依赖的软件”对话框中，单击“继续”按钮使其自动安装，如图 4-18 所示。

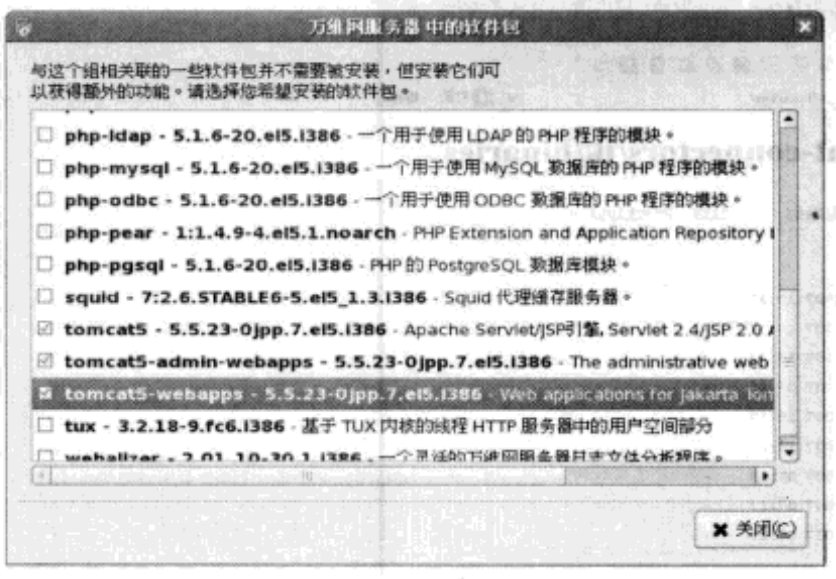


图 4-17　安装 Tomcat

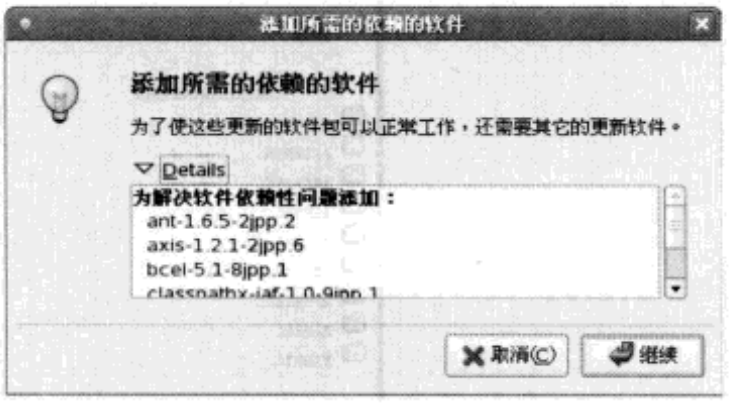


图 4-18　“添加所需的依赖的软件”对话框

（1）　启动和停止 Tomcat 服务

启动 Tomcat 服务可以通过执行以下命令完成。

```
[root@localhost ~]#/etc/rc.d/init.d/tomcat5 start
```

停止 Tomcat 服务可以通过执行以下命令完成。

```
[root@localhost ~]#/etc/rc.d/init.d/tomcat5 stop
```

重启 Tomcat 服务可以通过执行以下命令完成。

```
[root@localhost ~]#/etc/rc.d/init.d/tomcat5 restart
```

注意　使用命令启动 Tomcat 服务后，可以通过打开客户端浏览器访问 Tomcat（访问的具体方法是 http://服务器 IP 地址:8080）来测试 Tomcat 是否安装成功。

Tomcat 服务是不随系统启动而自动加载的。Tomcat 服务启动后，只能维持到本次关机之前。如果要设置 Tomcat 服务随系统启动而自动加载，则可以通过在终端窗口执行“ntsysv”命令来启动服务配置程序。服务配置程序启动后，找到“Tomcat5”服务，在其前面加上星号（*），最后单击“确定”按钮，如图 4-19 所示。

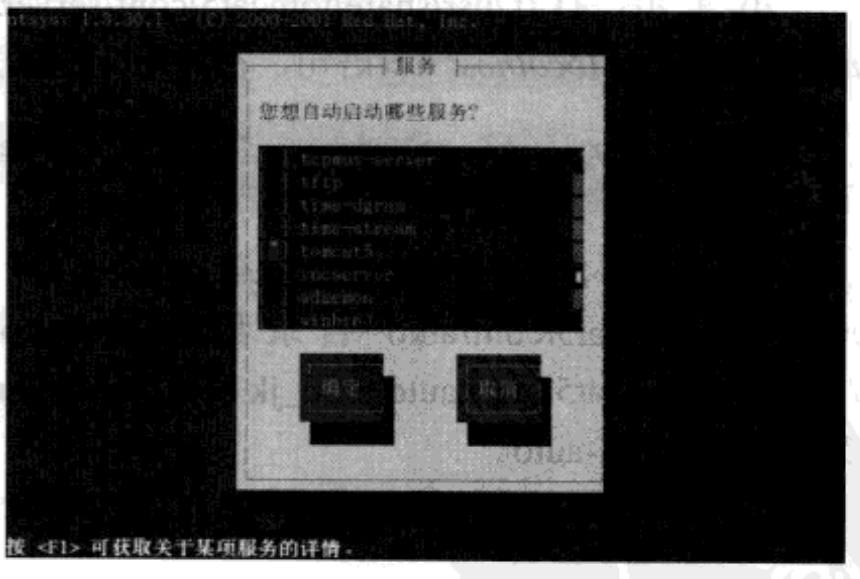


图 4-19　设置 Tomcat5 为自动启动

（2）　Apache 和 Tomcat 的整合

要实现 Apache 和 Tomcat 这二者的整合，需执行以下步骤。

首先，安装并配置 mod_jk。

第 1 步　到 Apache 官方网站下载“The Apache Tomcat Connector”（mod_jk），本书使用的是 1.2.26 版本，如图 4-20 所示。

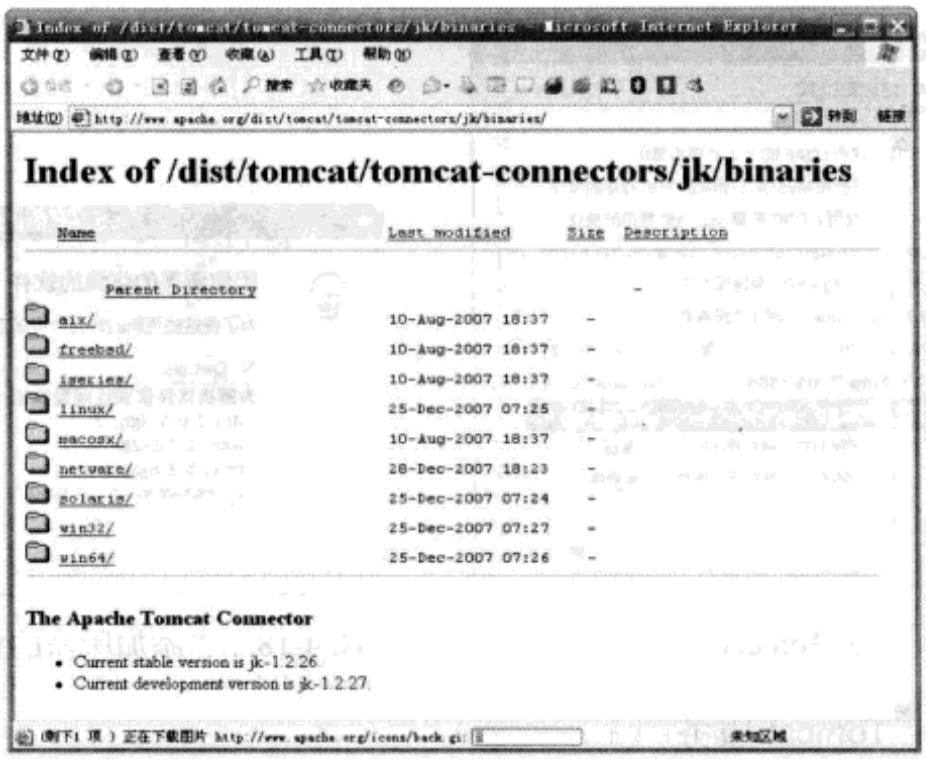


图 4-20　mod_jk 支持各种系统平台

第 2 步，将 mod_jk-1.2.26-httpd-2.2.6.so 复制到/etc/httpd/modules 目录中，并将其重命名为 mod_jk.so。

第 3 步，在/usr/share/tomcat5/conf 目录中新建子目录 jk，同时编辑属性文件 workers.properties，找到相关部分内容并修改为如下内容。

```
workers.tomcat_home=/usr/share/tomcat5
workers.java_home=/usr/lib/jvm/java
ps=/
worker.list= ajp13
worker.ajp13.port=8009
worker.ajp13.host=localhost
worker.ajp13.type=ajp13
worker.ajp13.lbfactor=1
```

第 4 步，打开/usr/share/tomcat5/conf/server.xml 文件，找到*<Engine name="Catalina" defaultHost="localhost">*语句，在该语句后添加如下语句。

```
<Listener  className="org.apache.jk.config.ApacheConfig"  modJk=
"/etc/httpd/modules/mod_jk.so" />
```

第 5 步，以上设置完成后，重新启动 Tomcat 服务。此时将自动生成/usr/share/tomcat5/conf/auto 目录和/usr/share/tomcat5/conf/auto/mod_jk.conf 文件。把文件/usr/share/tomcat5/conf/auto/mod_jk.conf 复制到/usr/share/tomcat5/conf/jk 目录中，并将其重命名为 mod_jk.conf-auto。

第 6 步，将 mod_jk.conf-auto 文件的内容修改为如下内容。

```
<IfModule !mod_jk.c>
   LoadModule jk_module "/etc/httpd/modules/mod_jk.so"
</IfModule>
JkWorkersFile "/usr/share/tomcat5/conf/jk/workers.properties"
JkLogFile "/usr/share/tomcat5/logs/mod_jk.log"
JkLogLevel emerg
```

```
<VirtualHost localhost>
    ServerName localhost
    JkMount /*.jsp ajp13
</VirtualHost>
```

第 7 步，配置 Tomcat 与 Apache。

设置一致的 Apache 和 Tomcat 的主目录是实现 Apache 与 Tomcat 整合的关键所在。编辑 Tomcat 的主配置文件/usr/share/tomcat5/conf/server.xml，找到如下语句。

```
<Host name="localhost" appBase="webapps"
 unpackWARs="true" autoDeploy="true"
 xmlValidation="false" xmlNamespaceAware="false">
```

在后面添加如下行。

```
<Context path="" docBase="/var/www/html" debug="0"/>
```

编辑文件/etc/httpd/conf/httpd.conf，在文件末尾添加如下语句。

```
Include /usr/share/tomcat5/conf/jk/mod_jk.conf-auto
```

第 8 步，重新启动 Tomcat 和 Apache。

通过以上对 Tomcat 和 Apache 的配置后，需要重启 Apache 和 Tomcat5 服务才能使配置生效，执行如下指令。

```
[root@localhost ~]#service httpd restart
[root@localhost ~]#service tomcat5 restart
```

第 9 步，测试是否整合成功。

重新启动 Tomcat 和 Apache 服务后，在 Apache 和 Tomcat5 共同的主目录/var/www/html/下新建一个名为 index.jsp 的文件，内容如下。

```
Hello word!
```

在客户端的浏览器中键入“http://服务器 IP 地址/index.jsp”，并按下 Enter 键，可以测试 Tomcat 与 Apache 的整合是否成功。如果在网页中出现 “Hello word!”的信息，则表示 Apache 和 Tomcat 整合成功。

4.7.2 创建并配置 CGI 动态网站

CGI 是 Common Gateway Interface（公共网关接口）的简写形式，其程序必须运行在网络服务器上，最常见的 CGI 语言是 Perl。

1. Perl 语言解释器的安装

默认配置安装的 Red Hat Enterprise Linux 5 中应该已经将 Perl 语言解释器安装在系统中了。在不确定是否安装了 Perl 语言解释器的情况下，可以通过以下指令确认。

```
[root@localhost ~]#rpm -q perl
```

执行命令后，显示“perl-5.8.8-10.e15_0.2”信息，如下所示，则表示已经安装了 Perl 语言解释器。

```
[root@localhost ~]# rpm -q perl
perl-5.8.8-10.el5_0.2
```

注意　如果系统中没有安装 Perl 语言解释器，就需要将 Red Hat Enterprise Linux 5 第 1 张安装盘放入光驱（或者是 Red Hat Enterprise Linux 5 DVD 安装光盘），然后加载光驱，在光盘的 Server 目录下找到 Perl 解释器的 RPM 安装包文件 perl-5.8.8-10.el5_0.2.i386.rpm，使用以下命令进行安装。

```
[root@localhost ~]#rpm -ivh /mnt/Server/ perl-5.8.8-10.el5_0.2.i386.rpm
```

2. 配置 Apache 支持 CGI

要使 Apache 支持 CGI，则 Apache 服务器需要在放置有 CGI 文件的目录下设置相应的执行权限。默认情况下，即使在 Apache 的配置文件 httpd.conf 中为主服务器定义了一个 ScriptAlias：/cgi-bin/，但 Apache 还是不支持 CGI。此时，需找到配置文件 httpd.conf 其中的“Options none”一行并将其修改为“Options ExecCGI”。

```
<Directory "/var/www/cgi-bin">
    AllowOverride None
    Options ExecCGI
    Order deny,allow
    Allow from all
</Directory>
```

注意　如果需要在非 ScriptAlias 定义的虚拟脚本目录中运行脚本语言（比如 CGI），就需要在 Apache 的配置文件 httpd.conf 中，将如下行

```
#AddHandler cgi-script .cgi
```

前面的注释去掉。

```
AddHandler cgi-script .cgi
```

3. 测试 CGI 运行环境

设置完所有的配置以后，就可以通过以下方法测试一下 CGI 程序是否能够正常运行了。

第 1 步，在 CGI 文件存放的目录（本例使用/var/www/cgi-bin/）下建立一个名为 index.cgi 的文件，内容如下。

```
#!/usr/bin/perl
print "Content-type: text/html\n\n";
print "Hello World!\n";
```

第 2 步，为 index.cgi 文件添加运行权限，执行以下指令。

```
[root@localhost ~]#chmod +x /var/www/cgi-bin/index.cgi
```

第 3 步，打开客户端浏览器，并访问“http://Linux 服务器的 IP 地址/cgi-bin/index.cgi”，如果页面显示为“Hello world!”，则表示运行成功。

第 5 章　DHCP 服务器

IP 地址及与之相关的子网掩码是用来标识主机及与其连接的子网的，如果计算机被转移到其他的子网中，则必须更改其 IP 地址信息。在基于 TCP/IP 的大型网络环境中，如果要为每台计算机配置一个 IP 地址，这将是一项非常艰巨的任务。为了降低重新配置计算机的难度，减少管理员的工作量，使用 DHCP 服务器来配置客户机的 IP 地址将是一个非常不错的选择。

5.1　DHCP 服务概述

DHCP（Dynamic Host Configuration Protocol，动态主机配置协议）是基于客户机/服务器模式的一个简化主机 IP 地址分配管理的 TCP/IP 标准协议。DHCP 服务用于向计算机自动提供 IP 地址、子网掩码和路由信息。目的在于减轻 TCP/IP 网络的规划、管理和维护的负担，解决 IP 地址空间缺乏问题。

DHCP 是一个基于广播的协议，它由 BOOTP（BOOTSTRAP PROTOCOL，自举协议）发展而来，分为服务器端和客户端两部分。DHCP 租约提供了自动在 TCP/IP 网络上安全地分配和租用 IP 地址的机制，实现了统一规划和管理网络中的 IP 地址，提高了 IP 地址的使用率，并方便了管理员的管理。

5.1.1　DHCP 服务的实现原理

DHCP 是一个基于广播的协议，工作时要求客户机和服务器进行交互，由客户机通过广播向服务器发起申请 IP 地址的请求，然后由服务器分配一个 IP 地址以及其他的 TCP/IP 设置信息。DHCP 客户端和服务器端申请 IP 地址、获得 IP 地址的过程一般分为 4 个阶段，具体实现过程如下。

1. 初始化

当 DHCP 客户机的网络首次启动，且没有设置自动获取 IP 地址时，则该计算机将使用源地址为 0.0.0.0 的 TCP/IP 地址向本地子网发送一个“DHCP Discover（DHCP 发现）”广播信息包。该信息包是通过 DHCP/BOOTP 服务器端口进行发送的。

2. 选择

当 DHCP 服务器接收到一个来自客户机的 IP 租用请求时，DHCP 服务器会检查自己是否为该客户机分配了一个 IP 地址。如果是，则以包括有效的 IP 地址、子网掩码、DHCP 服务器的 IP 地址、租用期限等信息的“DHCP Offer（DHCP 提供）”信息包作为响应，发送回 DHCP 客户机。在此过程中，因为没有对客户机进行限制，客户机能收到多个 IP 地址提供的信息。

3. 请求

客户机通常对第一个提议产生响应，并以广播的方式发送“DHCP Request（DHCP 请求）”信息包作为回应。该信息包告诉服务器“是的，我想让你给我提供服务。我接收你给我的租用期限”。一旦信息包以广播方式发送以后，网络中所有的 DHCP 服务器都可以看到该信息包，那些提议没有被客户机承认的 DHCP 服务器就将保留的 IP 地址返回给它的可用地址池。

注意 此时，客户机还可利用 DHCP Request 询问服务器其他的配置选项，如 DNS 服务器或网关地址。

4. 捆绑

服务器收到“DHCP Request”信息包后，如果没有例外发生，其会以一个“DHCP Acknowledge（DHCP 确认）”信息包作为响应，该信息包包括租用期限和客户所请求的所有其他配置信息，该信息包也是以广播的方式发送的。至此，客户机就能使用这个 IP 地址及相关的 TCP/IP 数据，来设置自己的 TCP/IP 堆栈了。

注意 使用 DHCP 中转可以解决客户机和服务器位于不同网络带来的问题。

5.1.2 IP 地址租约和更新

通过以上小节的学习，相信大家对 DHCP 服务器是如何把 IP 地址分配给客户端的过程都有所了解了。DHCP 服务器把 IP 地址分配给客户端后，主机的租赁信息文件将记录这次租约行为，并开始租约计时，这就涉及到 IP 地址租约和更新，接下来将具体讲解 IP 地址租约和更新。

1. IP 地址租约形式

DHCP 服务器是以地址租约的方式为 DHCP 客户端提供服务的，其主要提供限定租期和永久租用两种方式的地址租约。

- 限定租期

当 DHCP 服务器租用 IP 地址给 DHCP 客户端后，DHCP 客户端只是一段时间使用该地址。如果客户端在租约到期时，并没有更新租约，则 DHCP 服务器收回该 IP 地址，并将该 IP 地址提供给其他的 DHCP 客户端使用。如果原 DHCP 客户端又需要 IP 地址，可以向 DHCP 服务器重新租用另一个 IP 地址。

- 永久租用

当 DHCP 服务器租用 IP 地址给 DHCP 客户端后，DHCP 客户端将永远使用其分派的地址。如果有足够的 IP 地址给客户端使用，那么就没有必要限定其租用时期，给客户端自动分派 IP 地址就可以供其永久租用。

2. 进行租约更新

租约更新就是客户端取得 IP 地址后，在租约到期之前客户端将与服务器取得两次联系，准备下一步骤。

- 更新

如果客户端的租用期已达到期满时间的 50%以上，则需要更新该租用期。此时客户端将发送一个“DHCP Request”信息包给获得客户端原始信息的服务器。该信息包用以询问是否能保持 TCP/IP 配置信息并更新其租用期。如果服务器答应该请求，则该服务器通常返回一个 DHCP Acknowledge 信息包给客户端。

- 租约的重新捆绑

当客户端的租用期接近期满时间的 87.5%时，如果客户端请求租用期再次失败，那么客户端为了获得一个有效的 IP 地址，必须尝试与任何一台 DHCP 服务器联系。若其他 DHCP 服务器能够分配一个新的 IP 地址给客户端，则该客户端将再次进入捆绑状态。若客户端当前的 IP 地址租用期满，则客户端必须重新进入初始化状态，重复 IP 地址获取的整个过程。

3. 解除租约

DHCP 服务器与客户端 IP 地址之间的这个有效租约既可以更新也可以解除，租约的更新在上文已经进行了讲解，下面就将讲解解除租约的两种情况。

- 客户端离线

客户端离线时将会造成租约的解除。客户端离线，包括关机（shutdown）、重新开机（reoot）、关闭网络接口（ifdown）。当遇到以上情况时，IP 地址被 DHCP 服务器回收并放入其 IP 地址池中，等待下一个客户端申请。

- 客户端租约到期

客户端租约到期是造成租约解除的另一种情况。如果客户端使用的 IP 地址达到了使用期限的终点，且该客户端没有再次向 DHCP 服务器提出租约更新，那么此时该 IP 地址被 DHCP 服务器回收。

5.1.3 DHCP 服务器分配的 IP 地址类型

客户端向 DHCP 服务器申请 IP 地址时，DHCP 服务器分配给客户端的 IP 地址可以是静态 IP 地址也可以是动态 IP 地址，这就需要根据实际情况而定。

1. 动态 IP 地址

通常情况下，客户端从 DHCP 服务器获得的 IP 地址都不是固定不变的。每次获得的 IP 地址都可能不相同。使用动态 IP 地址是利用客户端不是同时上线的原理，这样让有效资源的利用达到最大化。

2. 静态 IP 地址

在实际的工作环境中，服务器并不是总给予客户端一个动态的 IP 地址。例如，有的单位不仅员工有自己的计算机，而且还有不止一台的服务器。如果这些服务器使用动态 IP 地址，

那么就不利于客户端的访问，并且不方便管理。因此可以设置 DHCP 服务器以记录特定计算机的 MAC 地址，然后为每个 MAC 地址分配一个固定的 IP 地址。至于网卡的 MAC 地址的查询，是根据网卡属于本机还是远程计算机而定的，其所采用的方法也略有不同。

所谓 MAC 地址也叫物理地址或硬件地址，具有唯一性，其存在于硬件的内部。在实际工作中需要通过 MAC 地址来区分不同的节点。

（1） 如何查询本机网卡 MAC 地址

查询本机网卡的 MAC 地址比较简单，直接使用 ifconfig 指令即可。

```
[root@localhost ~]#ifconfig
```

执行命令后的结果如下所示。

```
[root@localhost /]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0C:29:7C:06:77
          inet addr:192.168.1.106  Bcast:255.255.255.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe7c:677/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:816 errors:0 dropped:0 overruns:0 frame:0
          TX packets:65 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:97800 (95.5 KiB)  TX bytes:15351 (14.9 KiB)
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:2465 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2465 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:3613264 (3.4 MiB)  TX bytes:3613264 (3.4 MiB)
```

如上所示，第一行“HWaddr 00:0C:29:3D:1D:77”，中的“00:0C:29:3D:1D:77”即为网卡的 MAC 地址。

（2） 如何查看远程计算机的 MAC 地址

一般情况下，用户会用 Ping 命令来获取对方的 MAC 地址信息。

```
[root@localhost ~]#ping -c 1 192.168.1.6
```

ping IP 地址为 192.168.1.6 的计算机，执行命令后的结果如下所示。

```
[root@localhost /]# ping -c 1 192.168.1.6
PING 192.168.1.6 (192.168.1.6) 56(84) bytes of data.
64 bytes from 192.168.1.6: icmp_seq=1 ttl=64 time=7.94 ms
--- 192.168.1.6 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 7.949/7.949/7.949/0.000 ms
```

可以看到这样操作 MAC 地址不会被显示出来，此时就需要借助其他的工具来完成，执行如下命令。

```
[root@localhost ~]#arp -n
```

查看 ping 命令中缓存在本地计算机中的IP地址为 192.168.1.6 的计算机的 MAC 地址信息，执行命令后的结果如下所示。

```
[root@localhost ~]#arp -n
Address         HWtype  HWaddress           Flags Mask      Iface
192.168.1.6     ether   00:1F:D0:B1:FF:3A     C             eth0
```

以上所示的“00:1F:D0:B1:FF:3A ”即 IP 地址 192.168.1.6 的计算机的 MAC 地址。

5.2 安装 DHCP 服务

DHCP 服务器套件作为 Red Hat Enterprise Linux 5 安装光盘上的附带产品，在默认安装情况下不会自动安装，需要用户手工安装。在 Linux 作为 DHCP 服务器的情况下，只需要安装 DHCP RPM 包即可。可以利用 rpm 来安装，也可利用 yum 在线安装。

5.2.1 做好 DHCP 服务器安装前的准备

在安装 DHCP 服务器之前，应该做好以下准备。

1. 确认 DHCP 服务是否已经安装

在不明确系统中是否已经安装了 DHCP 服务的情况下，可执行以下命令进行确认。

```
[root@Localhost ~]# rpm -qa | grep dhcp
```

若执行命令后的结果如下所示，表明 DHCP 服务器没有安装，只安装了一个支持 IPv6 的客户端软件。

```
[root@localhost ~]# rpm -qa | grep dhcp
dhcpv6-client-1.0.10-4.el5
```

如果检测到没有安装任何 DHCP 服务器，则需要进行相应的安装。

2. 配置 dhcrelay 守护进程

如果计划把 DHCP 服务器用于远程网络，那么必须把 dhcrelay 守护进程配置在局域网之间的路由器/网关计算机上。

3. 确认服务器的网卡支持广播功能

要确认服务器的网卡支持广播功能需执行以下命令。

```
[root@localhost ~]#ifconfig eth0
```

执行命令后的结果如下所示，可以看到在第 5 行所示的信息中显示有 MULTICAST 字样，说明该网卡支持广播功能。

```
[root@localhost ~]# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:0C:29:7C:06:77
```

```
          inet addr:192.168.1.106  Bcast:255.255.255.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe7c:677/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:961 errors:0 dropped:0 overruns:0 frame:0
          TX packets:67 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:110478 (107.8 KiB)  TX bytes:15491 (15.1 KiB)
```

做好以上准备工作后，即可进行 DHCP 服务的安装。

5.2.2 安装 DHCP 服务的方法

一切准备就绪后，就可以进行 DHCP 服务的安装了，安装 DHCP 服务的方法有命令行手工安装和图形化安装两种方法。

1. 命令行手工安装

由于 DHCP 服务所需要的文件较少（只需要一个 dhcp-3.0.5-13.el5.i386.rpm 包即可），不存在依赖关系，因此使用命令行方式安装 DHCP 服务是一个不错的选择，具体安装步骤如下。

第 1 步，挂载 Red Hat Enterprise Linux 5 光盘。

将 Red Hat Enterprise Linux 5 安装光盘放入光驱，执行以下命令。

```
[root@localhost ~]#mkdir /mnt/cdrom          ; #创建一个用户挂接光盘的目录
[root@localhost ~]#mount /dev/cdrom /mnt/cdrom    ; #挂载光盘
```

第 2 步，找到与 DHCP 服务相关的 RPM 软件包。

将终端的当前工作目录切换到 Server 目录，在此目录下有很多 RPM 包文件，这里需要的是 DHCP 服务所需要的 RPM 包，可以执行以下命令进行查找。

```
[root@localhost Server]# ls dhcp*
```

执行该命令后的结果如下所示，显示 DHCP 服务器的包和版本号。

```
[root@localhost Server]# ls dhcp*
dhcp-3.0.5-13.el5.i386.rpm       dhcpv6-1.0.10-4.el5.i386.rpm
dhcp-devel-3.0.5-13.el5.i386.rpm  dhcpv6-client-1.0.10-4.el5.i386.rpm
```

第 3 步，安装 DHCP 服务所需要的 RPM 包，执行以下命令。

```
[root@localhost Server]# rpm -ivh dhcp*
```

执行命令后的结果如下所示，表明安装成功。

```
[root@localhost Server]# rpm -ivh dhcp-3.0.5-13.el5.i386.rpm
warning: dhcp-3.0.5-13.el5.i386.rpm: Header V3 DSA signature: NOKEY, key ID
37017186
Preparing...        ########################################### [100%]
   1:dhcp           ########################################### [100%]
[root@localhost Server]# rpm -ivh dhcp-devel-3.0.5-13.el5.i386.rpm
warning: dhcp-devel-3.0.5-13.el5.i386.rpm: Header V3 DSA signature: NOKEY, key
ID 37017186
```

```
Preparing...        ########################################### [100%]
   1:dhcp-devel     ########################################### [100%]
```

第 4 步，确认 Linux 中是否已经成功安装 DHCP 服务，执行如下命令。

```
[root@localhost Server]# rpm -qa | grep dhcp
```

执行命令后的结果如下所示，表明 DHCP 服务安装成功。

```
[root@localhost Server]# rpm -qa | grep dhcp
dhcpv6-client-1.0.10-4.el5
dhcp-devel-3.0.5-13.el5
dhcp-3.0.5-13.el5
```

注意　/etc/dhcpd.conf 文件是 DHCP 服务器的主要配置文件，可能在某些 Linux 版本中不存在，因此需要创建该文件。具体格式可以参考该文件的模板文件，位置在："/usr/share/doc/dhcp-<version>/dhcpd.conf.sample"。

2. 图形化安装

图形化安装 DHCP 服务的具体步骤如下。

第 1 步，将 Red Hat Enterprise Linux 5 安装光盘放入光驱，然后挂接。

注意　挂接的方法有 mount 指令手工挂接和编辑/etc/fstab 自动挂接两种方式。

第 2 步，在 GNOME 桌面，依次单击"应用程序"→"添加/删除软件"菜单命令，即可显示"软件包管理者"窗口，如图 5-1 所示。

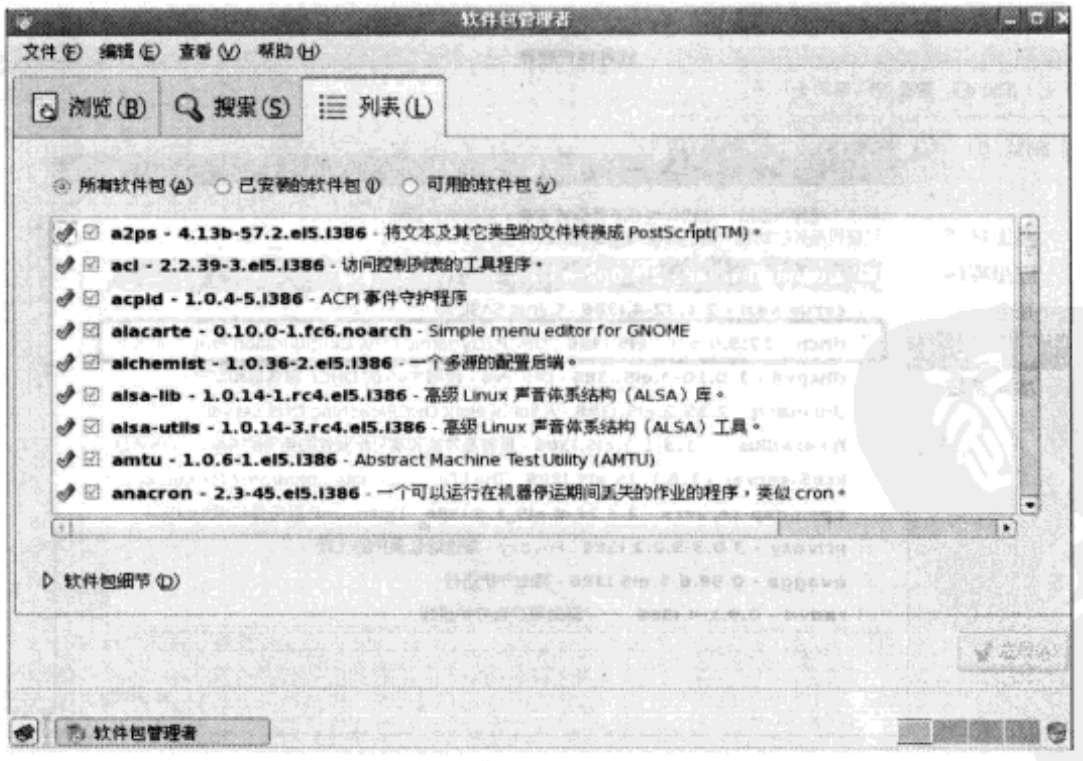


图 5-1　"软件包管理者"窗口

第 3 步，依次单击“编辑”→“软件存储库”菜单命令，显示如图 5-2 所示的“存储库管理工具”对话框。在该对话框中添加 Red Hat Enterprise Linux 5 安装光盘为新的安装源。

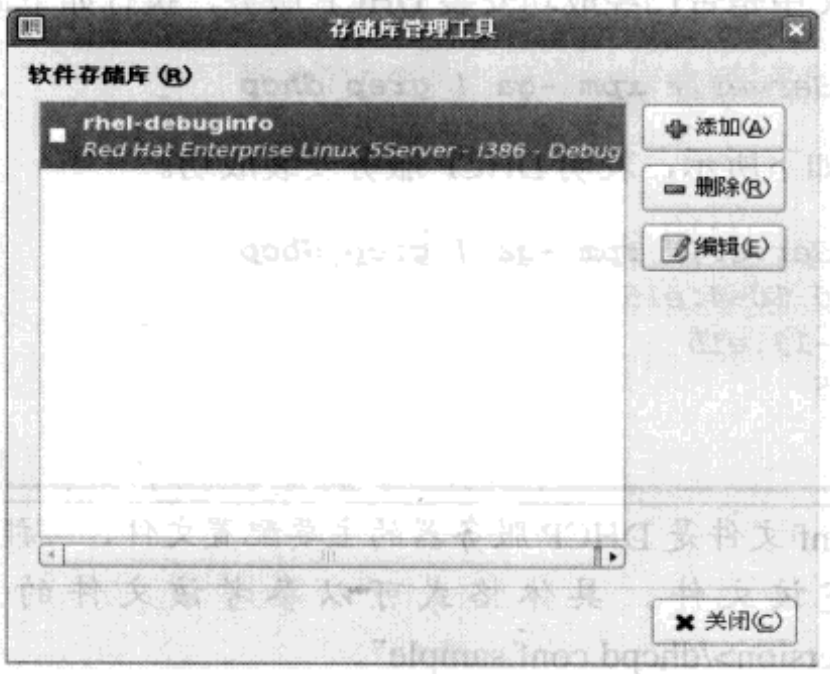


图 5-2 “存储库管理工具”对话框

注意 添加 Red Hat Enterprise Linux 5 安装光盘为新的安装源的具体方法为：单击“添加”按钮，将弹出“添加新的存储库”窗口。在该窗口中按照时间情况填写相关内容。

第 4 步，添加完新的“软件存储库”后，返回到“软件包管理者”窗口。在“浏览”选项组下，显示与安装 Red Hat Enterprise Linux 5 过程中一样的组件选择窗口，然后单击窗口左侧的“服务器”选项，再选中右侧的“网络服务器”复选框，最后单击下方的“可选的软件包”按钮，显示“网络服务器中的软件包”对话框，选中“dhcp-12:3.0.5-13.el5.i386”复选框，如图 5-3 所示。

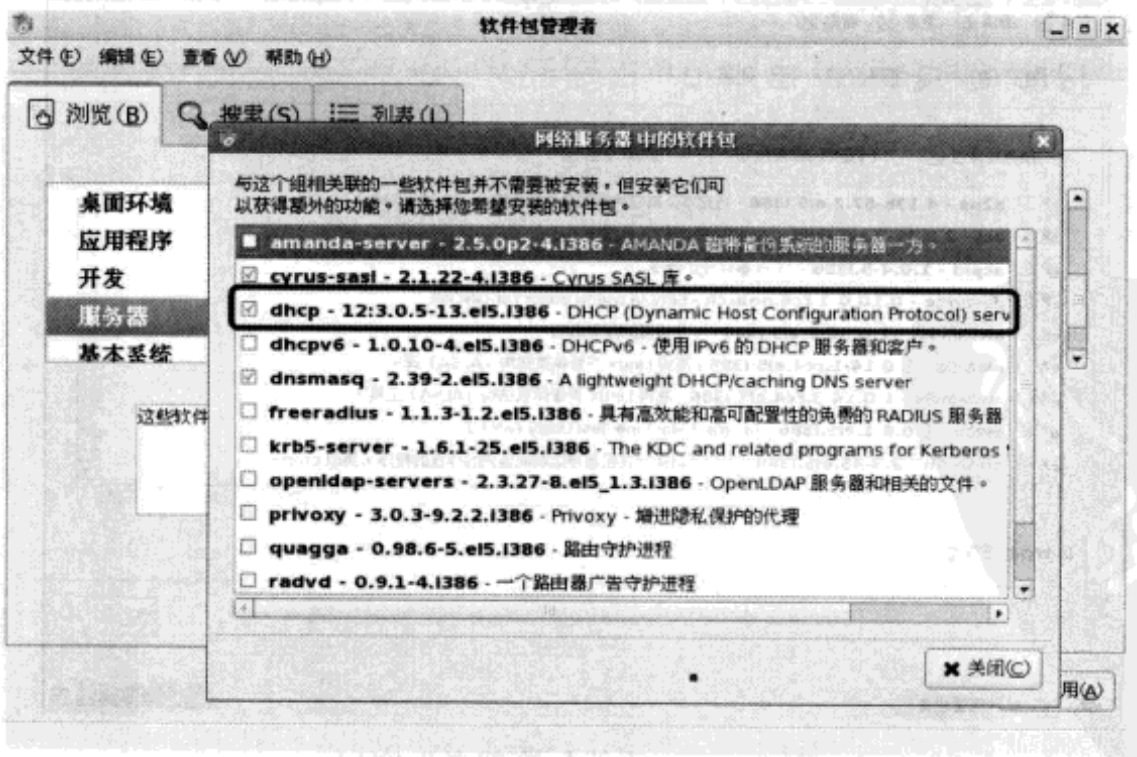


图 5-3 “网络服务器中的软件包”对话框

第 5 步，依次单击“关闭”→“应用”→“继续”菜单命令完成操作。至此，DHCP 服务器已经安装到系统中了。

5.2.3 启动及停止 DHCP 服务

DHCP 服务遵循一般的服务启动、停止规范，下面将具体讲解 DHCP 服务的启动、重启以及停止操作。

1. 启动 DHCP 服务

启动 DHCP 服务需输入以下命令。

```
[root@localhost ~]#service dhcpd start
```

执行命令后的结果如下所示，表明 DHCP 服务启动成功。

```
[root@localhost ~]#service dhcpd start
启动 dhcpd:                                    [确定]
```

注意　启动 DHCP 服务也可以通过执行以下命令实现。

```
[root@localhost ~]# /etc/init.d/dhcpd start
```

2. 重启 DHCP 服务

DHCP 服务的重启操作类似于其启动操作，要停止并重新启动该服务，可以通过执行以下重启命令实现。

```
[root@localhost Server]#service dhcpd restart
```

执行命令后的结果如下所示，表明该服务已重新启动。

```
[root@localhost Server]#service dhcpd restart
关闭 dhcpd:                                    [确定]
启动 dhcpd:                                    [确定]
```

注意　重启 DHCP 服务也可以通过执行以下命令实现。

```
[root@localhost ~]# /etc/init.d/dhcpd restart
```

3. 停止 DHCP 服务

DHCP 服务的停止方式也类似于其启动操作，要停止该服务，只需输入以下停止服务的命令即可。

```
[root@localhost Server]#service dhcpd stop
```

执行命令后的结果如下所示，表明该服务已停止。

```
[root@localhost Server]#service dhcpd restart
关闭 dhcpd:                                                [确定]
```

注意 停止 DHCP 服务也可以通过执行以下命令实现。

```
[root@localhost ~]# /etc/init.d/dhcpd stop
```

4. DHCP 服务的自动运行

一般情况下，启动 DHCP 服务后，只能运行到此次计算机关闭之前，再次启动系统后又要重新启动该服务。为了避免此种情况的发生，可以通过 ntsysv、chkconfig 或者服务控制中心来实现 DHCP 服务随系统启动而自动运行，具体做法如下。

在“终端”窗口输入 ntsysv 命令，显示 Red Hat Enterprise Linux 5 下的“服务”配置小程序，找到 dhcpd 项，并在其前面加上“*”，如图 5-4 所示。至此完成了 DHCP 服务随系统启动而自动运行的配置。

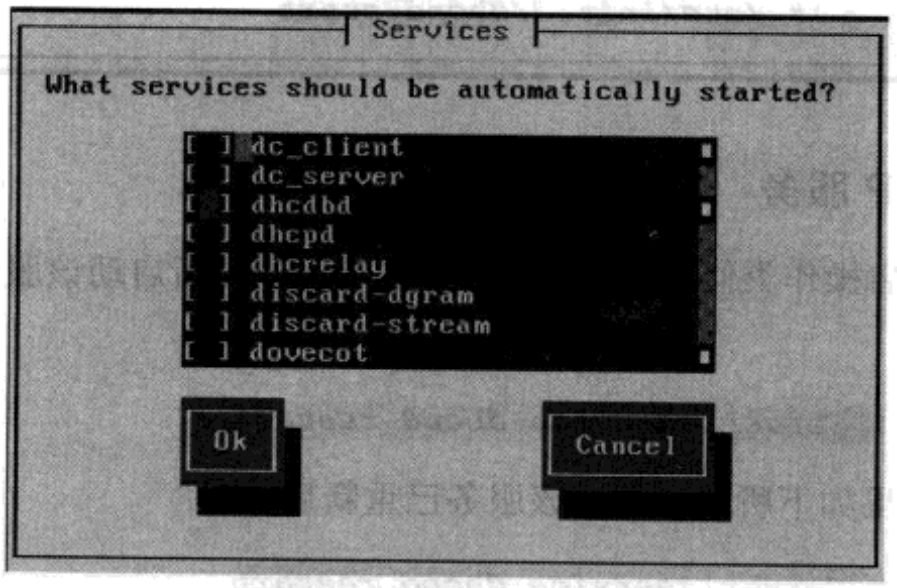


图 5-4 设置 DHCP 服务为自动运行

5.3 配置 DHCP 服务

DHCP 服务器是根据客户的申请而给客户端提供 IP 地址和其他相关信息的服务。其性质和 DHCP 服务的工作原理是一致的。在启用该服务之前，首先必须为其指定可用的 IP 地址池，并进行其他的相关配置。

5.3.1 DHCP 配置文件及格式

想熟悉掌握配置 DHCP 服务器的方法，首先必须熟悉该服务的配置文件，需要了解 DHCP

配置文件及配置文件的格式、参数、声明和选项等信息。

1. DHCP 配置文件参数

要查看 DHCP 配置文件（即 dhcpd.conf）的参数，可以通过以下步骤实现。

第 1 步，进入 etc 目录，执行如下命令。

```
[root@localhost ~]# cd /etc
```

第 2 步，使用 vi 打开该配置文件，执行如下命令。

```
[root@localhost etc]# vi dhcpd.conf
```

执行命令成功后，配置的文件内容显示如下。

```
ddns-update-style interim;
ignore client-updates;
subnet 192.168.0.0 netmask 255.255.255.0 {
# --- default gateway
    option routers              192.168.0.1;
    option subnet-mask          255.255.255.0;
    option nis-domain           "domain.org";
    option domain-name          "domain.org";
    option domain-name-servers  192.168.1.1;
    option time-offset          -18000; # Eastern Standard Time
#   option ntp-servers          192.168.1.1;
#   option netbios-name-servers 192.168.1.1;
# --- Selects point-to-point node (default is hybrid). Don't change this unless
# -- you understand Netbios very well
#   option netbios-node-type 2;
    range dynamic-bootp 192.168.0.128 192.168.0.254;
    default-lease-time 21600;
    max-lease-time 43200;
    # we want the nameserver to appear at a fixed address
    host ns {
        next-server marvin.redhat.com;
        hardware ethernet 12:34:56:78:AB:CD;
        fixed-address 207.175.42.254;
    }
```

（1） 全局设置

全局设置包括设置租约期限、路由器的 IP 地址、DNS 服务器的 IP 地址、动态 DNS 的更新类型。如果在静态 IP 地址和动态 IP 地址设置中没有明确指定这些参数，那么就以全局设置为标准，几种常见的全局参数如表 5-1 所示。

（2） 静态 IP 地址

利用 host 参数来指定客户端的 MAC 地址，以及分配给客户端的 IP 地址信息。

（3） 动态 IP 地址

根据网段和子网掩码以及 range 参数进行配置的一种 IP 地址设置方式。

表 5-1　全局参数及其说明

全局参数	全局参数说明
default-least-time（默认租约时间）	指定默认租赁时间的长度，默认是 21 600 s，即 6 h
max-lease-time（最大租约时间）	当客户端超过这个时间而没有发出更新租期请求时，DHCP 服务器会回收该 IP 并放回到 IP 地址池中待其他客户端申请
domain-name-server（DNS 服务器）	指定 DNS 服务器，客户端会自动以它作为自己的 DNS 服务器
option domain-name（域名）	指定本地网络的域名。如果在/etc/resolv.conf 中设置了“search xxxx”选项，则搜索主机名称时，DNS 系统会自动将这个域名加进去
option subnet-mask（子网掩码）	设置子网掩码
option routers（网关）	指定客户端的网关 IP 地址
ddns-update-style(动态更新 DNS 资料）	ddns-update-style 有：ddns-update-style interim，ddns-update-style ad-hoc 和 ddns-update-style none 3 种方式。其中最常用的是 ddns-update-style interim
option broadcast-address（广播地址）	默认情况下，系统会自动根据 A、B 和 C 类网段进行计算

2. etc/dhcpd. conf 的配置格式

dhcpd.conf 文件对关键字严格区分大、小写，每行除了括号以外都必须以分号（“;”）结束；以“#”号开始的行表示该行为注释行，格式如下。

```
全局选项/参数          ; #这些选项/参数全局有效并位于声明之上
声明{
        选项/参数      ; #这些选项/参数局部有效
}
```

注意　安装好 DHCP 服务后并不会自动新建 dhcpd.conf 这个配置文件，用户必须手动新建一个配置文件，具体方法参见前文。

dhcpd.conf 配置文件包括一套声明集和一套参数集。

（1）　声明集

用来描述与系统连接的网络环境，以及存在于该环境中的各个事务的地址范围。声明之间可以互相嵌套。

以下是一个网段的声明集。

```
subnet 192.168.1.0 netmask 255.255.255.0 {
option routers 192.168.1.254;
option subnet-mask 255.255.255.0;
option domain-name "www.domain.com";
option domain-name-servers 192.168.1.1;
option time-offset -18000;
range 192.168.1.10 192.168.1.100;
}
```

（2）　参数集/选项集

用来描述服务器如何操作以及对配置机器的响应。根据参数所在位置不同，参数设置可

以是全局或者局部的声明。

参数集的参数总是由设置项和设置值两部分组成，并以“;”结束。选项集的选项与参数集的参数不同，其总是以 option 关键字引导，后面跟具体的选项和选项值，也必须以“;”结束。

3. etc/dhcpd. conf 中的声明

DHCP 配置文件中 declarations（声明）用来描述网络布局，提供客户的 IP 地址等，主要内容如表 5-2 所示。

表 5-2 DHCP 配置文件中的声明

声　明	功 能 说 明
shared-network	用来告知一些子网络是否分享相同网络
subnet	描述一个 IP 地址是否属于该子网
range 起始 IP 终止 IP	提供动态分配 IP 的范围
host 主机名称	参考特别的主机
group	为一组参数提供声明
allow unknown-clients; deny unknown-client	是否动态分配 IP 给未知的使用者
allow bootp; deny bootp	是否响应激活查询
allow booting; deny booting	是否响应使用者查询
filename	开始启动文件的名称，应用于无盘工作站
next-server	设置服务器从引导文件中加装主机名，应用于无盘工作站

4. etc/dhcpd. conf 中的选项

DHCP 配置文件中的 option（选项）用来配置 DHCP 可选参数，全部用 option 关键字作为开头，主要内容如表 5-3 所示。

表 5-3 DHCP 配置文件中的选项

选　项	说　明
subnet-mask	为客户端设定子网掩码
domain-name	为客户端指明 DNS 名字
domain-name-servers	为客户端指明 DNS 服务器 IP 地址
host-name	为客户端指定主机名称
routers	为客户端设定默认网关
broadcast-address	为客户端设定广播地址
ntp-server	为客户端设定网络时间服务器 IP 地址
time－offset	为客户端设定和格林威治时间的偏移时间，单位是 s

5.3.2 配置 IP 作用域

当 DHCP 客户端向 DHCP 服务器请求 IP 地址时，DHCP 服务器为了响应客户端的请求，就需要从某个 IP 作用域中选择一个尚未分配的 IP 地址，并将其分配给该 DHCP 客户端。因此，

就需要在 dhcpd.conf 文件中用 subnet 语句为 DHCP 服务设置 IP 作用域。

```
subnet 子网 ID  netmask  子网掩码 {
        range  起始 IP 地址 结束 IP 地址;
# 指定可分配给客户端的 IP 地址范围
        IP 参数;
# 定义客户端的 IP 参数，如子网掩码、默认网关等
    }
```

例如，如果只想对子网为 192.168.3.1，子网掩码为 255.255.255.0 的网段中的 192.168.3.10~192.168.3.100 和 192.168.3.120~192.168.3.220 之间的 IP 地址进行分配，这时可做如下设置。

```
Subnet 192.168.3.0 netmask 255.255.255.0 {
range  dynamic-bootp192.168.3.10 192.168.3.100;
range  dynamic-bootp 192.168.3.120 192.168.3.220;
}
```

5.3.3 配置客户机的 IP 选项

DHCP 服务器既可以指定 DHCP 客户端 IP 地址，又可以设置 DNS 服务器地址、子网掩码等工作环境。要为 DHCP 客户端设置 IP 选项，可以使用以下方法。

为 DHCP 客户端设置 IP 选项的指令格式如下。

```
option 选项代码 设置内容;
option routers  IP 地址;                    #为客户设置默认网关和路由器 IP 地址
option subnet-mask  子网掩码;               #设置了子网掩码
option nis-domain  “名称”;                  #设置 NIS 服务器的域名
option domain-name  “名称”;                 #设置了 DHCP 客户端的 DNS 域名
option domain-name-servers  IP 地址;        #设置了 NDS 服务器的 IP 地址(可多个)
option broadcast-address IP 地址;           #设置 DHCP 客户端在该 IP 子网中的广播地址
default-lease-time  time;                   #设置了默认的租约时间长度，单位为 s
max-lease-time time;                        #设置了最大的租约期限，单位为 s
host computer1
{
    hardware ethernet MAC 地址;             #绑定的网卡地址
    fixed-address IP 地址;                  #绑定的 IP 地址
}
```

如果需要设置相关客户机的 IP 地址，按照以上格式进行设置即可。

```
option routers                  192.168.3.5;            #设置默认网关的 ip 地址是 192.168.3.5
option subnet-mask              255.255.254.0;          #设置子网掩码 255.255.254.0
option nis-domain               "JavaEye.org";          #设置 NIS 服务器的域名为 JavaEye.org
option domain-name   " JavaEye.org";                    #设置 DHCP 客户端的 DNS 域名为
# JavaEye.org
option domain-name-servers 192.168.3.2,202.115.3.11;    #设置 DNS 服务器的 IP
#地址为 192.168.3.2 和 202.115.3.11
option broadcast-address 192.168.3.255;                 #设置 DHCP 客户端在该 IP 子网中的广播
#地址为 192.168.3.255
default-lease-time              2400;                   #设置默认的租约时间长度为 3 600 s
max-lease-time        7200;                             #设置最大的租约期限为 7 200 s
```

```
host computer1
{
    HWaddr ethernet 00:0C:29:7C:06:77;          #绑定的网卡地址为#00:0C:29:7C:06:77
    fixed-address 192.168.1.32;                 #绑定的 IP 地址为 192.168.1.32
}
```

5.3.4 设置租约期限

DHCP 服务器为 DHCP 客户端提供的 IP 地址是临时的，这就涉及到租约期限的设置。所谓租约期限是在 DHCP 服务器上指定的一段时间，在此时间段内 DHCP 客户端可以临时使用从 DHCP 服务器租借到的 IP 地址。可以在 dhcpd.conf 文件中使用如下命令来设置 IP 地址的租约时间的长度（以 s 为单位）。

1. 默认的租约期限

```
default-lease-time time;                    #配置默认租期期限
default-lease-time 3600;                    #配置默认租期期限 1 h
```

2. 最大租约期限

```
max-lease-time time;                        #配置最大租约期限
max-lease-time 43200;                       #配置最大租约期限 12 h
```

5.3.5 保留特定的 IP 地址

保留特定的 IP 地址给指定的客户端使用，即当该客户端每次向 DHCP 服务器索取 IP 地址或更新租约时，DHCP 服务器都会给该客户端分配相同的 IP 地址。这样做不仅可以防止用户随意更改 IP 地址，还方便了用户对相关信息进行设置的工作（无需对 IP 地址、网关地址、DNS 服务器等信息进行设置）。

要保留特定的 IP 地址给指定的 DHCP 客户端使用，可先用 arp 命令将该客户端网卡的 MAC 地址查出，然后在/etc/dhcpd.conf 文件中加入如下格式的 host 语句。

```
host 主机名 {
next-server marvin.redhat.com;
#设置用于定义服务器从引导文件装入的主机名，一般不用，仅用于设置无盘工作站
        hardware ethernet  网卡的 MAC 地址
#指定 DHCP 客户端网卡的 MAC 地址
        fixed-address  IP 地址
#指定为该 DHCP 客户端分配的 IP 地址/
        IP 参数
#指定默认网关等其他 IP 参数。
}
例如：
host computer1{
    hardware Ethernet 00.20.c2.94.b3.10;
```

```
    fixed-address 192.168.6.7;
}
```

本段代码表示将网卡的 MAC 地址 00.20.c2.94.b3.10 和 192.168.6.7 的 IP 地址绑定到一起。此时将 IP 地址 192.168.6.7 分配给网卡地址为 00.20.c2.94.b3.10 的客户机。

5.3.6 多网段的 IP 地址的分配

在一个规模比较大的物理网络中可能存在多个 IP 子网，而每个子网中的主机的 IP 地址都需要使用 DHCP 服务器来提供、分配，该工作可以使用以下两种方法来实现。

方法 1 为每一个网段设置一个 DHCP 服务器

通过该方法，DHCP 服务器将分别为各个子网分配 IP 地址，不过此方法会浪费较多资源。

方法 2 在某一个子网中安装 DHCP 服务器

还有一种方法则比较常用，就是在一个子网中安装 DHCP 服务器并让其为多个子网分配 IP 地址，要实现此功能则必须使用 DHCP 的中继代理功能。DHCP 的中继代理（dhcrelay）允许将无 DHCP 服务器的子网内的 DHCP 客户请求转发给其他子网内的一台或多台 DHCP 服务器。

具体的实现方法是在每个子网中设置一台计算机作为 DHCP 中继代理，网络中的主机将 IP 地址的请求发给中继代理，由中继代理与 DHCP 服务器联系将获得的 IP 地址再发给请求的主机，如图 5-5 所示。

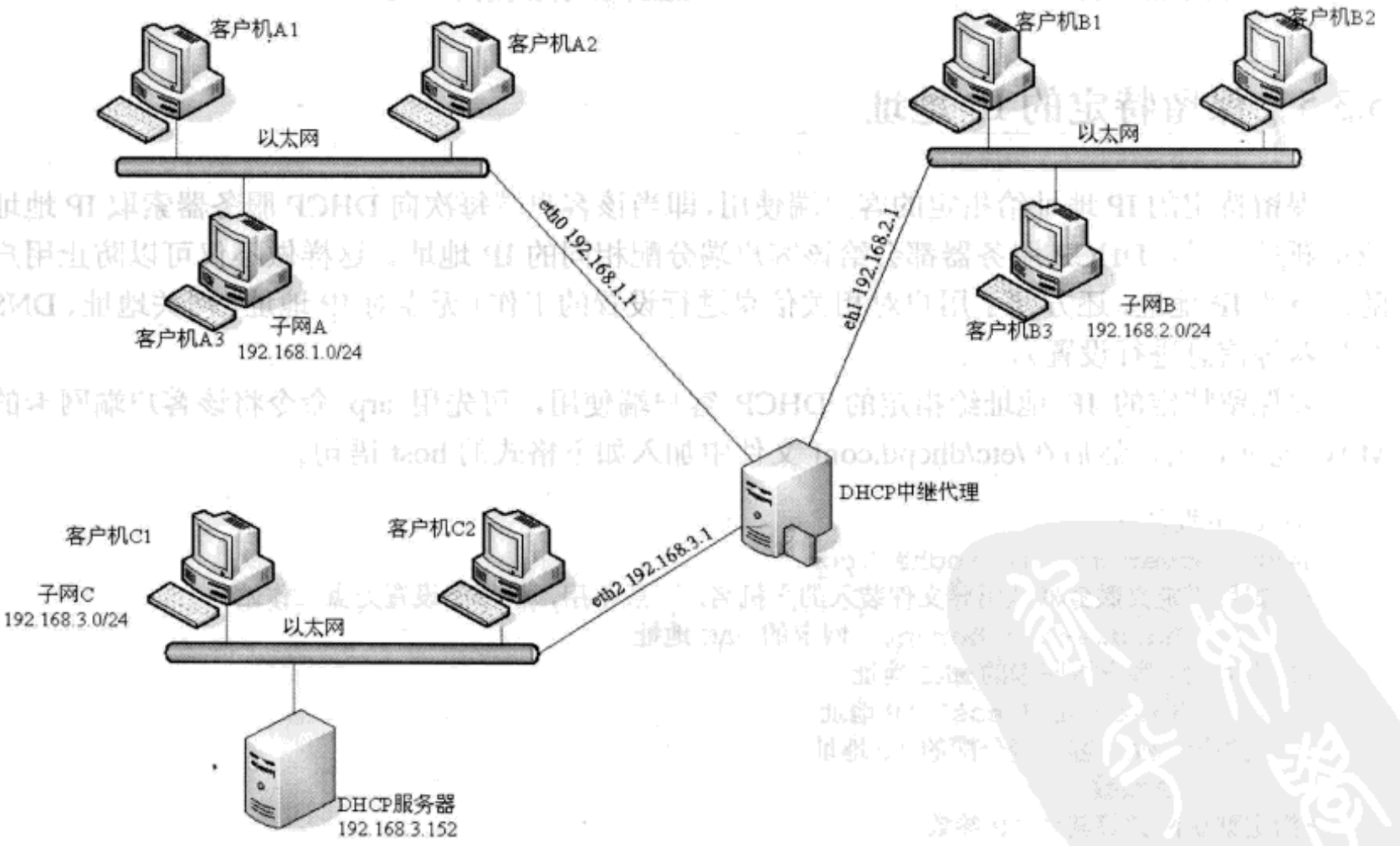


图 5-5 多网段的 DHCP 服务器架设

1. 设置 DHCP 中继代理

DHCP 的中继代理允许将无 DHCP 服务的子网内的 DHCP 和 BOOTP 请求转发给其他子网内的一台或多台 DHCP 服务器。当某台 DHCP 客户请求信息时，DHCP 中继代理把该请求转发给 DHCP 服务器，当某个 DHCP 服务器返回一个响应时，该响应将被广播或单播发送给最初请求的网络。同时，中继代理本身要有静态 IP 地址。

启动 DHCP 中继代理的命令如下。

```
[root@localhost ~]#service dhcrelay start
```

注意 除非使用 INTERFACES 指令在/etc/sysconfig/dhcprelay 文件中制定了接口，DHCP 中继代理监听所有接口上的 DHCP 请求。

2. 指定 DHCP 中继代理

一般情况下，DHCP 中继代理监听所有接口上的 DHCP 请求。假如某台 DHCP 服务器位于网络接口为 eh0 的子网内，那么就可以用 dhcrelay 向 eth1 和 eth2 连接的子网内提供 DHCP 服务，具体命令如下。

```
[root@localhost ~]#dhcrelay -i eth1 -i eth2 192.168.2.6
```

以上命令表明这里 IP 地址为 192.168.2.6，所指定的主机为 eth0 连接的子网内的 DHCP 服务器。参数 “-i”，表示让 DHCP 中继代理向指定的子网提供 DHCP 服务。如果没有任何参数，表示向所有子网提供服务。

5.3.7 DHCP 服务器配置实例

假设某公司有一台采用 Red Hat Enterprise Linux 5 作为操作系统的 DHCP 服务器，主要为整个局域网提供共享上网服务以及 DHCP 分配 IP 地址功能，其他的网络要求如下。

- 内部网段为 192.168.0.0/20，路由器 IP 地址为 192.168.0.1。
- 可以分配动态 IP 地址的范围为 192.168.0.50~192.168.0.200。
- DHCP 分配的 IP 地址默认租约为半天，最长为两天。
- 有台 Web 服务器，其 MAC 地址为“00-50-56-C0-00-01”，机器名为 Host1，需要分配给 Web 服务器的 IP 地址为 192.168.0.5。

配置文件的指令如下。

```
[root@localhost ~]#vi /etc/dhcpd.conf
#=============1. Global Settings(全局配置)================
default-lease-time 43200;
# 设置默认租约时间：半天
max-lease-time     172800;
# 设置最大租约时间： 2 天
option domain-name-servers 192.168.1.1,61.128.128.68;
```

```
#  设置 DNS 服务器，其最多可指定三个，相互之间以“,”分割
option domain-name "localdomain.tst";
#  所属网域名称
option routers 192.168.0.1 ;
# 网关 IP 地址(Gateway)
option broadcast-address 192.168.0.255 ;
# 广播地址
ddns-update-style interim;
#========2.Dynamic IP Address Settings（设置动态 IP 地址）============
   subnet 192.168.0.0 netmask 255.255.255.0 {
# 设置网段和子网掩码
     range dynamic-bootp 192.168.0.50 192.168.0.200 ;
# 定义动态 IP 地址范围，此范围可以有多个，一个范围写一行
#========3.Static IP Address Settings（设置静态 IP 地址）============
host 1Tiger {
        hardware ethernet 00:50:56:C0:00:01;
        fixed-address 192.168.0.5;
        option host-name "Host1";
        }
}
```

5.4 配置 DHCP 客户机

DHCP 服务是一种标准的基于 TCP/IP 网络的应用。当 DHCP 服务器配置完成后，客户机就可以使用 DHCP 功能，通过设置网络属性中的 TCP/IP 通信协议属性来完成对 IP 地址获得方式的设定以及对 DNS 服务器地址获得的设定。因此避免了为每台客户机设置 IP 地址、网关地址、子网掩码和 DNS 服务器地址等属性的麻烦。DHCP 服务的客户端可以是 Linux、UNIX、Windows 以及 MAC OSX。接下来以 Linux 和 Windows 两种客户端为例来介绍如何进行 DHCP 的设置。

5.4.1 以 Linux 操作系统作为客户端配置

在 Linux 操作系统中，为 DHCP 客户端提供了图形化配置与命令行配置两种方法，图形化配置方法相对直观，而熟悉命令行的朋友则会认为命令行配置方法简单。下面将讲解这两种方法的具体配置过程。

1. 图形化配置

DHCP 客户端在安装 Red Hat Enterprise Linux 时已经自动安装，配置过程也比较简单。用户既可以在安装过程中进行设置，也可以在系统安装好之后在网络中重新进行配置，接下来将介绍在系统安装好之后进行配置的具体方法。

第 1 步，在桌面上依次单击“系统”→“管理”→“网络”菜单命令，即可显示如图 5-6 所示的“网络配置”窗口。

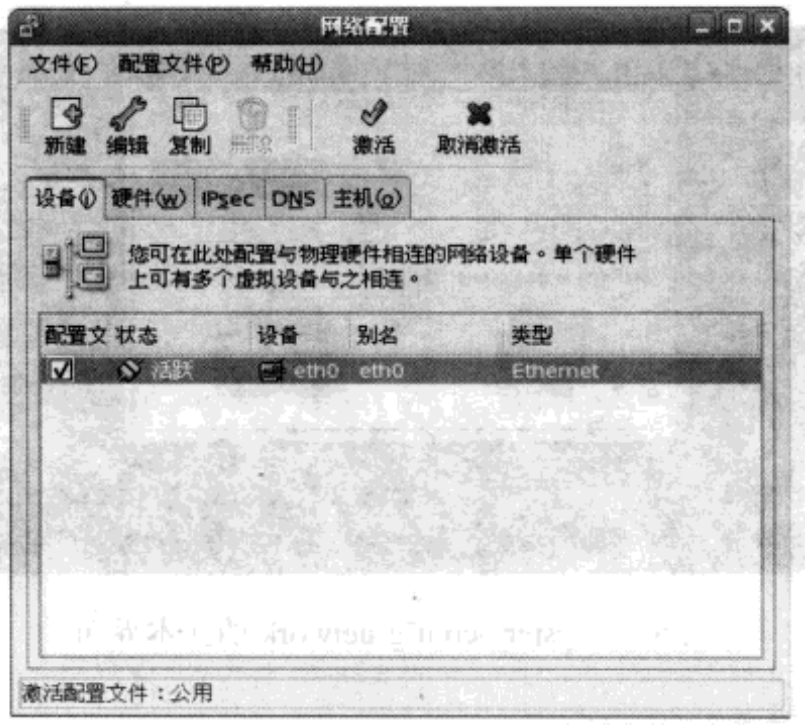


图 5-6 “网络配置”窗口

第 2 步，在“网络配置”窗口中选中需要设置 IP 地址的网卡，然后单击“编辑”按钮，弹出如图 5-7 所示的“以太网设备”对话框。

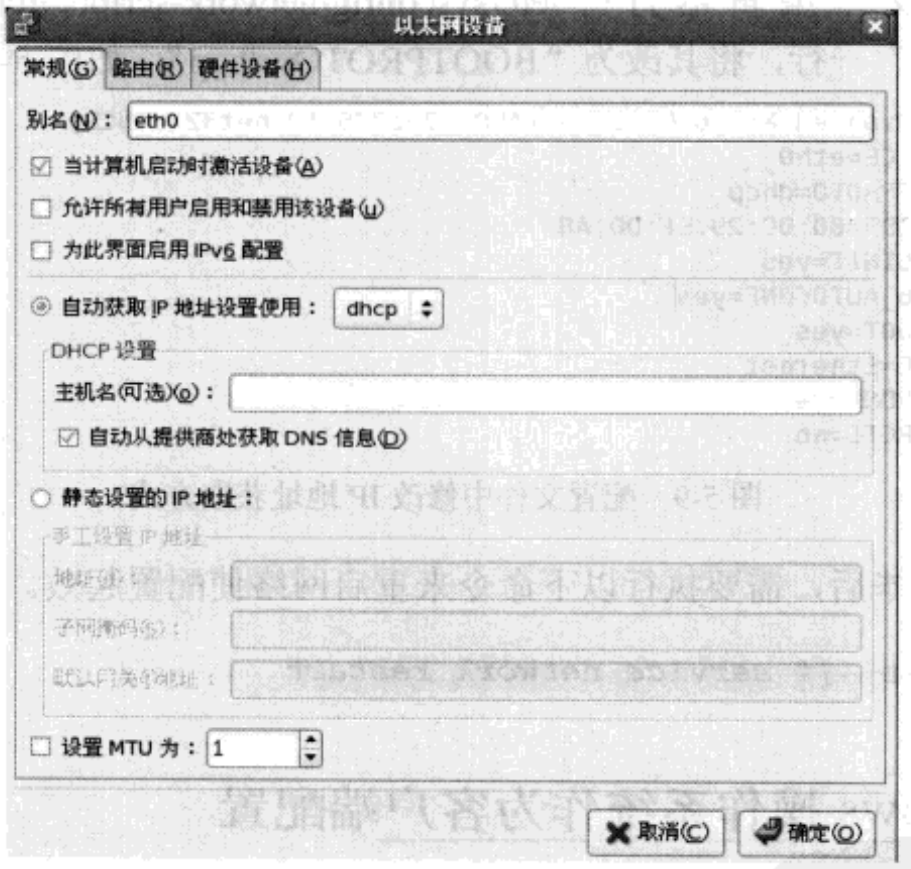


图 5-7 “以太网设备”对话框

第 3 步，在此对话框中，设置“自动获取 IP 地址设置使用”为“dhcp”，然后选中“自动从提供商处获取 DNS 信息”复选框。设置完成后单击“确定”按钮，至此 DHCP 的设置完成了。最后在终端使用 ifconfig 命令查看客户端的 DHCP 设置是否正常。

另外，在 Red Hat Enterprise Linux 5 中的某些图形界面配置工具也可以在命令行中配置使用。因此，在本例中，使用命令：system-config-network-tui（tui 就是 Text User Interface，文本用户界面）即可实现，如图 5-8 所示。

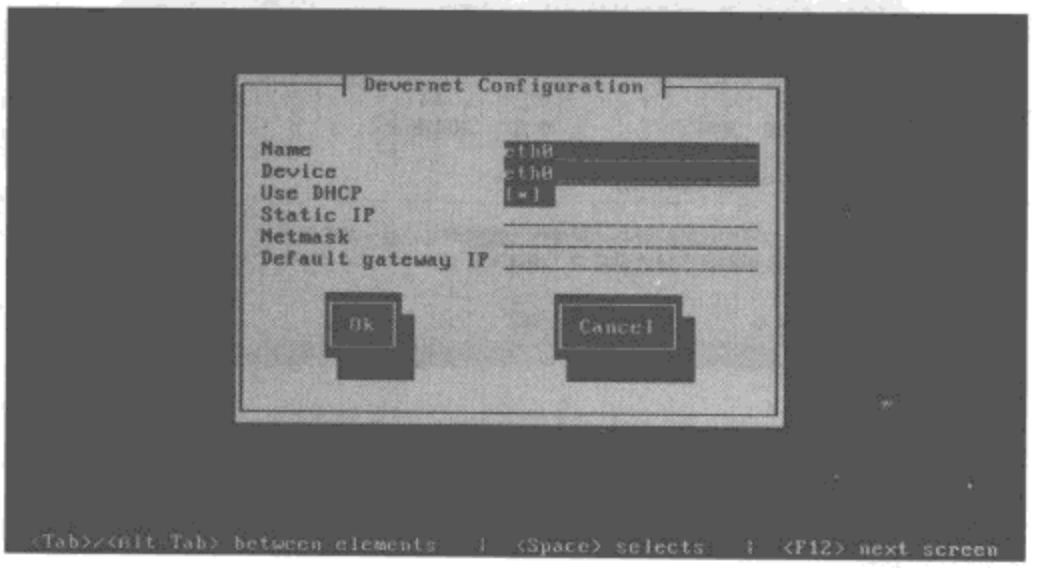


图 5-8　system-config-network 的文本界面

2. 使用命令行方法设置

要在 Linux 操作系统中使用命令行的方法来配置 DHCP 客户端，只需修改 ifcfg-eth0 文件即可，具体实现方法如下所述。

第 1 步，用文本编辑器打开 /etc/sysconfig/network-scripts/ifcfg-eth0 文件，找到“BOOTPROTO=none”一行，将其改为“BOOTPROTO=dhcp”，如图 5-9 所示。

```
# Advanced Micro Devices [AMD] 79c970 [PCnet32 LANCE]
DEVICE=eth0
BOOTPROTO=dhcp
HWADDR=00:0C:29:EF:DD:A8
IPV6INIT=yes
IPV6_AUTOCONF=yes
ONBOOT=yes
TYPE=Ethernet
PEERDNS=yes
USERCTL=no
```

图 5-9　配置文件中修改 IP 地址获取方式

第 2 步，设置完毕后，需要执行以下命令来重启网络使配置起效。

```
[root@localhost ~]# service network restart
```

5.4.2 以 Windows 操作系统作为客户端配置

在 Windows 操作系统中，DHCP 的客户机配置比较简单，下面就以 Windows XP 操作系统为例，具体讲解在 Windows 操作系统中如何配置 DHCP 客户端。

第 1 步，在桌面依次单击“开始”→“设置”→“控制面板”菜单命令，即可显示“控制面板”窗口，然后双击“网络连接”图标，如图 5-10 所示。

执行以上操作后，将显示“网络连接”窗口，如图 5-11 所示。

第 2 步，使用鼠标右键单击“本地连接”图标，选择“属性”菜单项，显示如图 5-12 所示的“本地连接 属性”对话框。

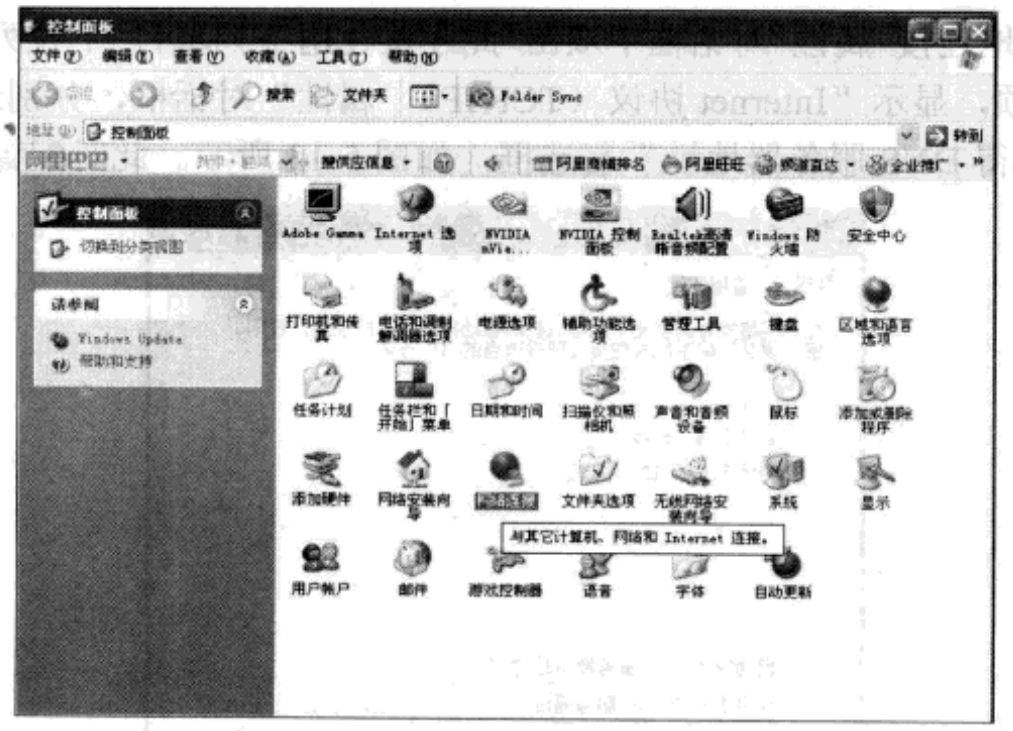


图 5-10 “控制面板”窗口

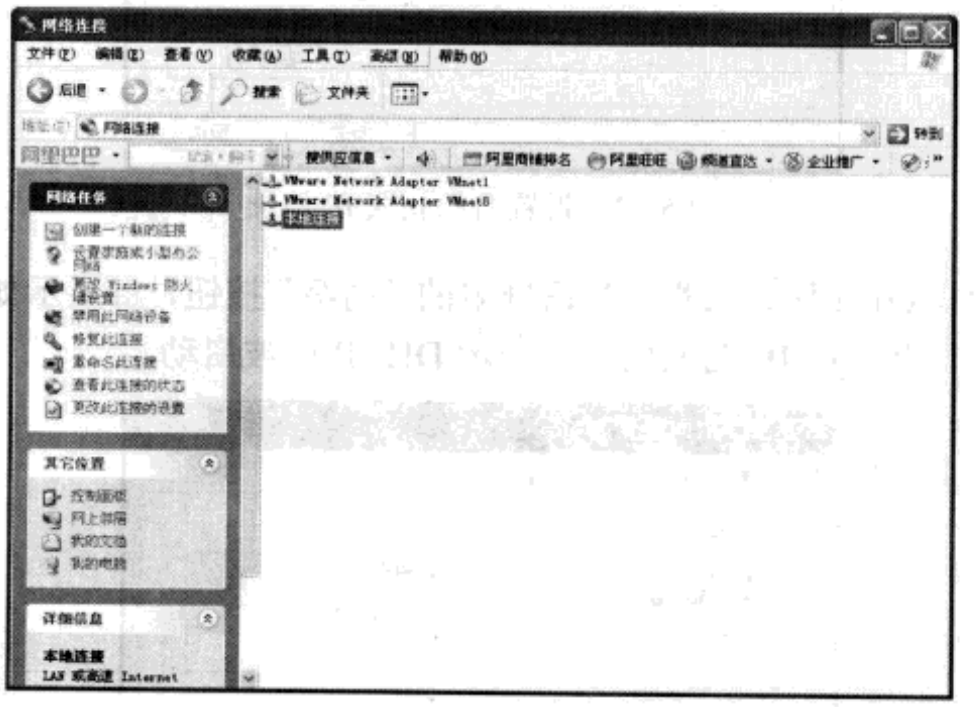


图 5-11 “网络连接”窗口



图 5-12 “本地连接 属性”对话框

第3步，在“本地连接 属性”对话框中双击“此连接使用下列项目”下拉列表框中的“Internet 协议（TCP/IP）”项，显示“Internet 协议（TCP/IP） 属性”对话框，分别选中“自动获得 IP 地址”和“自动获得 DNS 服务器地址”复选框，如图 5-13 所示。此时配置已完成。

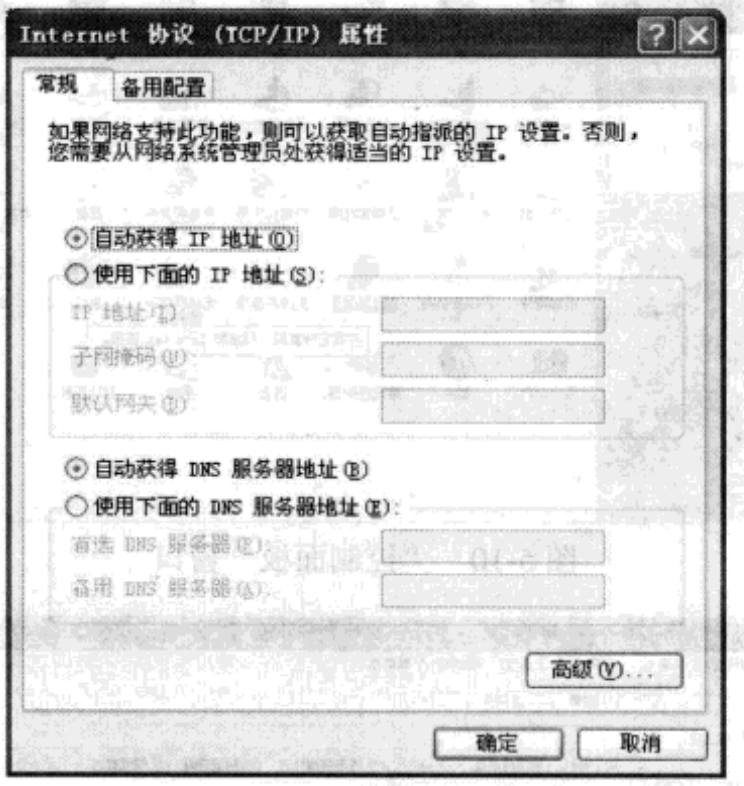


图 5-13　设置自动获取 IP 地址

第 4 步，单击“Internet 协议属性”对话框中的“高级”按钮，将显示如图 5-14 所示的“高级 TCP/IP 设置”对话框，在 IP 地址栏中将显示 DHCP 已被启动。

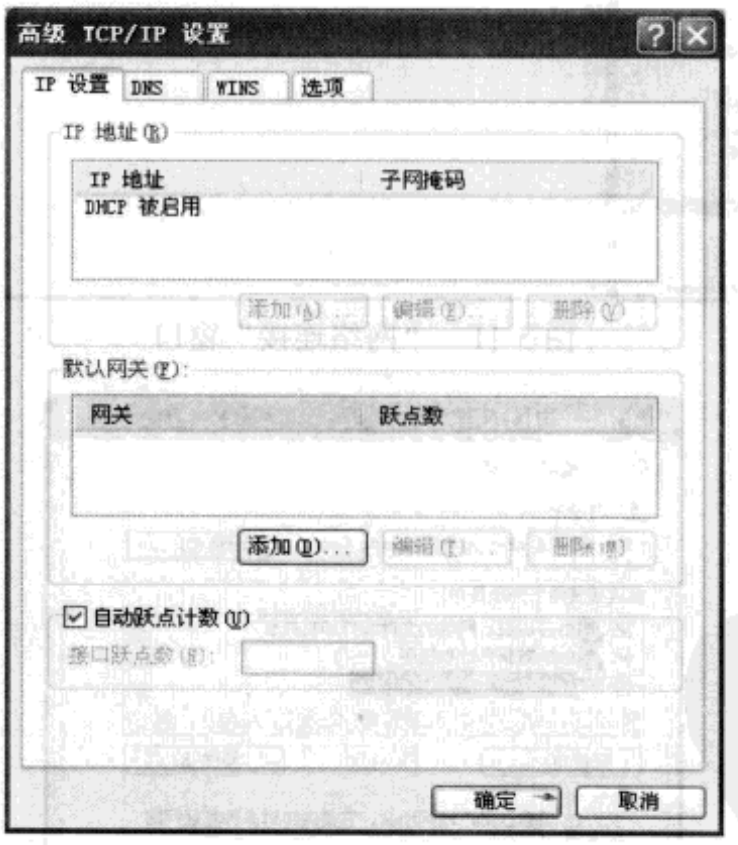


图 5-14　确认 DHCP 已被启动

另外，在 Windows 2000/XP/2003/Vista 下，DHCP 客户端有还具有以下功能：可以利用命令“ifconfig /all”查看详细的网络配置信息；利用命令“ifconfig /release”自动将 IP 地址释放；利用命令 “ifconfig/renew”更新 IP 地址租约。

第 6 章　DNS 服务器

现在，用户打开一个网站页面只需输入该网站的网址，而很多用户可能并不知道，在 Internet 发展的早期，只有通过 IP 地址才能打开需要浏览的网页。如今，大多数情况下，登录各个网站都不需要用到 IP 地址，而是通过网站的中、英文网址，甚至只是部分关键字就可以访问特定的网页。实现这一便捷功能的网络服务就是 DNS 域名解析服务，它实现了 IP 地址到域名的对应，使用户能够更加方便、快捷地访问 Internet 资源。

6.1　DNS 服务概述

DNS（Domain Name Server，域名服务器）是为了便于访问 Internet 而采用的一种分布式的域名→IP 地址的映射查询和管理方法。用户在不知道主机 IP 地址而只知道主机域名的情况下，也可以轻松访问服务器，这就是 DNS 的功劳。

6.1.1　DNS 服务简介

Internet 基于 TCP/IP 协议。而目前最常用的 IP 协议是 IPv4，由 32 位数字组成。为便于记忆，将其转换成了 4 组 10 进制的数字，即用户平时看到的 IP 地址。在访问 Internet 时，通过这串数字实现定位和访问服务器。

1.　/etc/hosts

在访问某个主机的时候，由于一连串数字的格式不便于记忆，所以要先去查询对方的 IP 地址。在 Internet 发展的早期，利用特定的文件（如/etc/hosts）将主机名和 IP 地址一一对应，让访问者可以通过主机名获得对方主机的 IP 地址。局域网中全部主机及其对应 IP 地址的清单分别保存在每一台主机上，这也是 DNS 的前身。早期的主机利用/etc/hosts 文件实现主机和 IP 地址的一一对应。

```
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1               localhost.localdomain localhost
::1             localhost6.localdomain6 localhost6
```

2.　Bind 现身

使用/etc/hosts 文件对应主机与 IP 地址的时候，主机中的清单无法自动更新，且只有向 InterNIC 注册，才能获得主机名与 IP 地址的对应。若 IP 数量太多，文件的对应关系多到不可想象，将不利于主机同步化。

值得庆幸的是，伯克利大学发展出另一个阶层式管理主机名称对应 IP 的系统，填补了这

一缺憾。Berkeley Internet Name Domain 简称 Bind，是目前全世界使用最广泛的域名解析系统。通过分布式管理，可以很轻松地进行维护工作。

6.1.2 DNS 提供的服务

DNS 最主要的功能是实现主机名到 IP 地址的转换，除此之外，其还提供其他一些重要的服务。

1. 主机别名——hody aliasing

具有复杂主机名的主机可以有别名，数量可以是一个甚至是多个。例如，主机名为 hhwork1.sichuan.enterprise.com 的主机有两个别名：enterprise.com 和 www.enterprise.com。这种情况下，主机名 hhwork1.sichuan.enterprise.com 称为正规主机名（canonical hostname），另外两个主机名则是别名主机名（alias hostname）。

别名主机名比正规主机名往往更容易记忆。调用 DNS 可以获取给定别名主机名的正规主机名和 IP 地址。

2. 邮件服务器别名——mall server aliasing

一般来说，电子邮件地址最好是便于记忆的。例如，toplinux 有一个 163 账号，他的电子邮件地址有可能是简单的 toplinuxt@163.com。电子邮件应用可以调用 DNS 获取所给定别名主机名的正规主机名和 IP 地址。

3. 负载分担——load distribution

DNS 还越来越多地用于执行在多个复制成的服务器之间的负载分担。像 163.com 那样的繁忙站点往往把 Web 服务器复制成多个，每个服务器运行在不同的端系统上，具有不同的 IP 地址。

客户端发出针对映射到一组 IP 地址的某个主机名的 DNS 查询后，服务器响应以整组 IP 地址，不过每次响应的地址顺序是轮转的。访问 Web 站点时，DNS 轮转可以把 Web 站点的访问负载分担在所有复制成的服务器上。电子邮件应用也可以使用 DNS 轮转，这样多个邮件服务器就可以有相同的别名。近来，有些公司已经以更为复杂的方式使用 DNS 提供 Web 内容分发服务。

6.1.3 DNS 查询原理和规划

DNS 系统包含 DNS 服务器和 DNS 客户端两部分。提供 DNS 服务的主机叫做 DNS 服务器，又称为域名服务器，而提出“域名查询”请求的主机叫做 DNS 客户端。DNS 客户端同样也拥有简单的 DNS 查询功能，通过本地的 3 个配置文件：/etc/host.conf、resolv.conf 和 hosts 实现。

1. DNS 查询原理

客户端发送的每条查询信息都包括指定的查询类型、DNS 域名的类别及合格的 DNS 域名

（FQDN）。

当需要解析域名时，先在本机进行域名解析，如果查询不到，就将这个请求发送给自己的域名解析服务器，而并不是直接到 DNS 域名服务器进行解析。

（1） 本地域名解析

不同系统的本地域名解析机制有所不同。Windows 系统会先查询“WINDOWS\system32\drivers\etc”中的 Hosts 文件，如果包含相关条目，则直接使用其 IP 地址。而 Linux 系统则需要按照主机上/etc/host.conf 文件的定义，来决定域名查询的顺序。/etc/host.conf 文件中默认的设置是：先通过/etc/hosts 文件进行域名解析，如果解析不到，再到 DNS 服务器进行域名解析。其声明语句格式如下。

```
order hosts , bind
```

（2） DNS 服务器查询历程

本地不能解析的域名，会被客户端发送到 DNS 服务器进行解析。比如，对于域名 www.cnnic.net.cn，客户端会先将其发送到 DNS 服务器，DNS 服务器查询若未在本地缓存找到相关记录，就会将该域名发送到根服务器，根服务器向“cn 域”查询“www.cnnic.net.cn”，“cn 域”在其本地也没有找到相关信息，就会向“.net.cn 域”查询“www.cnnic.net.cn”的 IP 地址，然后“.net.cn 域”再询问“cnnic.net.cn”，最后得出“www.cnnic.net.cn”的 IP 地址，如图 6-1 所示。

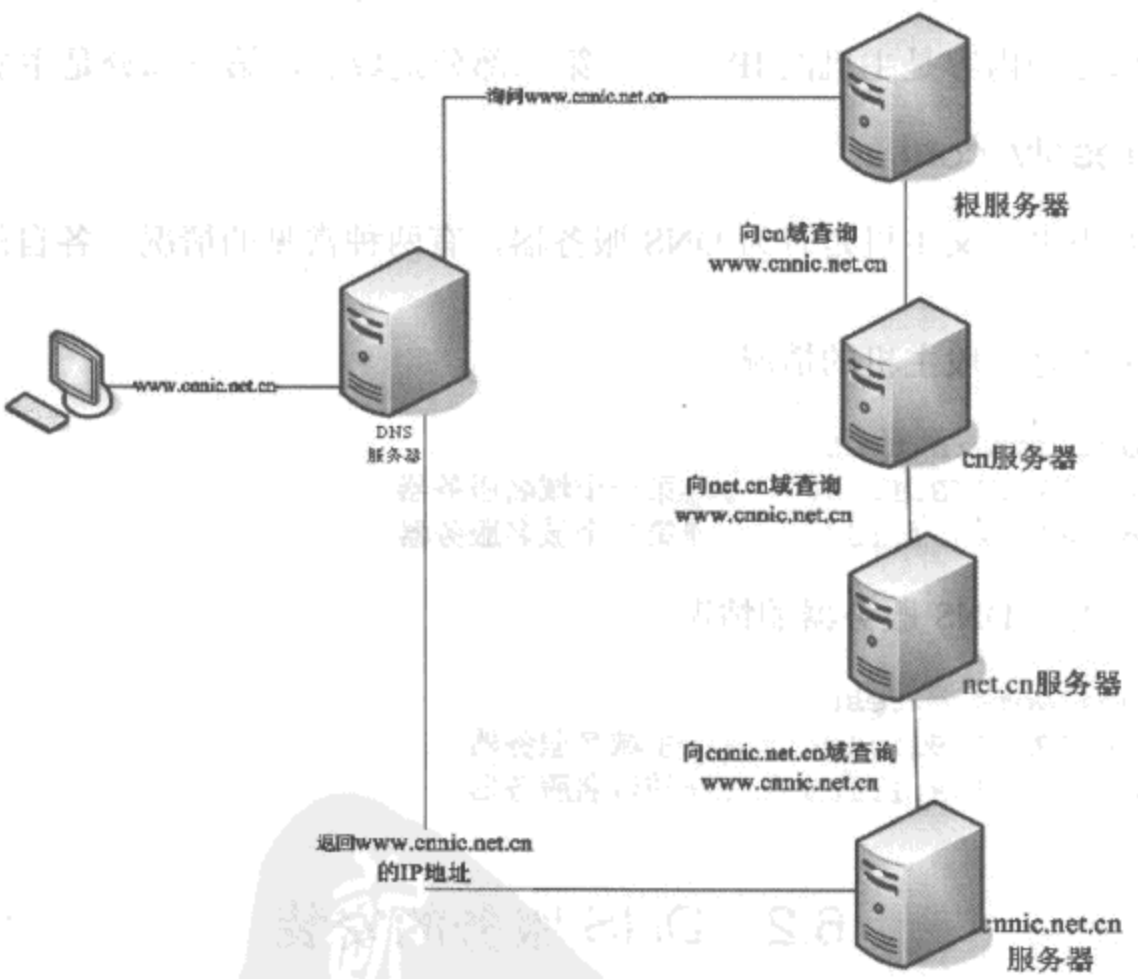


图 6-1 查询 www.cnnic.net.cn 的 IP 地址

2. DNS 规划

在使用 DNS 之前，必须要先规划 DNS 域名称空间。名称空间规划的内容包括如何使用

DNS 命名和使用 DNS 要达到什么样目的等。选择并注册一个顶级或二级域名作为父 DNS 域名，该域名应当是可用于维护 Internet 上的公司的，如 localdomain.test。将父域名称与公司名称或其特定位置组合起来，形成其子域名或者主机名，如 certer.localdomain.test。

6.1.4 DNS 域名解析

在 Linux 操作系统中，和 DNS 域名解析的实现密切相关的有 3 个文件是：/etc/host.conf、resolv.conf 及 hosts。

1. /etc/host. conf

该文件可以定义域名解析的搜索顺序，其常见的搜索顺序有如下 3 种。

- order hosts,dns
- order hosts,NIS,dns
- order hosts,dns,wins

2. /etc/hosts

该文件的配置指令格式如下。

```
192. 168. 1. 104 toplinux.localdomain.test   toplinux
```

其中第一部分的内容是主机的 IP 地址，第二部分是域名，第三部分是主机的别名。

3. /etc/resolv. conf

该文件主要用来定义主机是否为 DNS 服务器，有两种常见的情况，各自的文件配置指令格式如下。

- 当本主机是一般主机的情况

```
search localdomain.test
nameserver 192.168.1.104     ; #第一个域名服务器
nameserver 61.128.1.105      ; #第二个域名服务器
```

- 当本主机是 DNS 服务器的情况

```
domain localdomain.test
nameserver 192.168.1.104     ; #主域名服务器
nameserver 192.168.1.105     ; #辅域名服务器
```

6.2 DNS 服务的安装

在 Red Hat Enterprise Linux 5 系统按照默认设置进行安装的情况下，是不会安装 DNS 服务的；在安装过程中必须选择安装该服务，DNS 才会随系统安装。

在系统组件选择窗口中，选择左侧的“服务器”标签，并选中右侧的“DNS 名称服务器”复选框，最后单击“下一步”按钮，如图 6-2 所示。

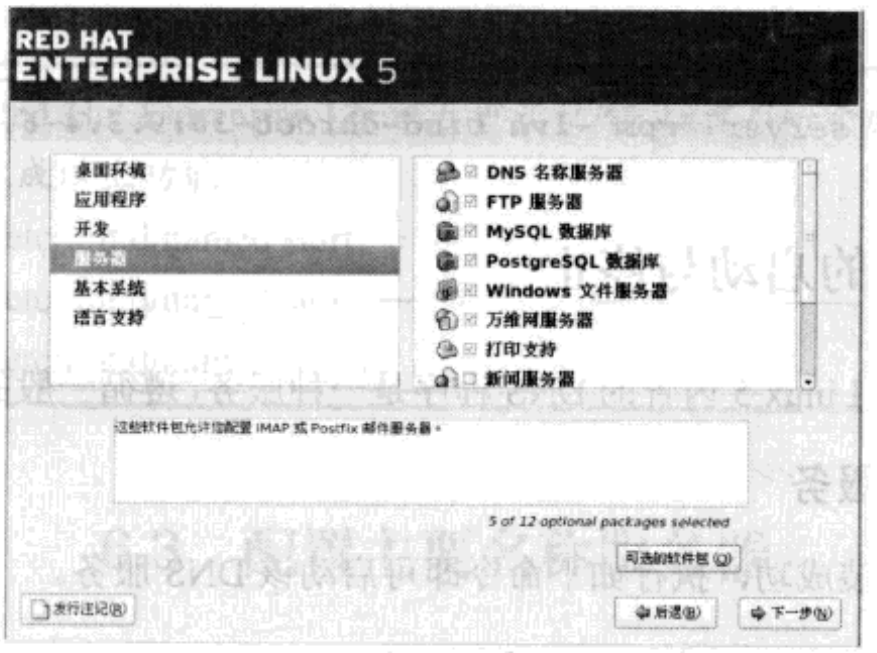


图 6-2　手动安装 DNS 名称服务器

6.2.1 安装 DNS 服务

默认安装的 Red Hat Enterprise Linux 5 系统中，是不包含 DNS 服务的，也就是说用户需要手动进行安装。

第 1 步，查看系统是否已经安装 DNS 服务，指令执行如下。

```
[root@localhost etc]#rpm -qa |grep bind
ypbind-1.19-8.el5
bind-libs-9.3.4-6.P1.el5
bind-9.3.4-6.P1.el5
bind-utils-9.3.4-6.P1.el5
bind-chroot-9.3.4-6.P1.el5
[root@localhost ~]#
```

由指令执行的结果可以确定，系统已经安装 DNS 服务。但是如果指令执行没有返回相应软件包，则需要进行下一步来安装 DNS 服务。

第 2 步，如果系统尚未安装这项服务，则可在 Red Hat Enterprise Linux 5 安装盘的 Server 目录下，找到 DNS 服务的 RPM 安装包（包括多个文件）。使用如下命令进行安装。

```
[root@localhost server]#rpm -ivh bind-libs-9.3.4-6.P1.el5.i386.rpm
[root@localhost server]#rpm -ivh bind-9.3.4-6.P1.el5.i386.rpm
[root@localhost server]#rpm -ivh bind-utils-9.3.4-6.P1.el5.i386.rpm
```

6.2.2 安装 Chroot 软件包

Chroot（Change Root）用来改变程序执行时的根目录位置。早期的很多系统程序默认情况下所执行的根目录都是“/”。黑客或者其他的非法入侵者很容易通过绝对路径/etc/passwd 窃取系统机密。有了 Chroot 后，比如 bind 的根目录就被改变到了/var/named/chroot，这样黑客即使突破了 Bind 账号，也只能访问到/var/named/chroot 文件，把对系统攻击的危害降低到了最小。

为了让 DNS 运行更加安全，就需要安装 Chroot。在 Red Hat Enterprise Linux 5 安装光盘的 Server 目录下，找到 Chroot 的 RPM 安装包 bind-chroot-30:9.3.4-6.P1.el5.i386.rpm。使用以

下命令进行安装。

```
[root@localhost server]#rpm -ivh bind-chroot-30:9.3.4-6.P1.el5.i386.rpm
```

6.2.3 DNS 服务的启动与停止

Red Hat Enterprise Linux 5 内置的 DNS 程序是一种服务，遵循一般的服务启动、停止规范。

1. 启动 DNS 服务

如果 DNS 已经安装成功，执行如下命令即可启动该 DNS 服务。

```
[root@localhost ~]#service named start
```

2. 停止 DNS 服务

类似于启动操作，要停止 DNS 服务，可以采用类似的命令。

```
[root@localhost ~]#service named stop
```

3. 重启 DNS 服务

类似于 DNS 服务的启动操作，要重启该服务，也可以采用类似的命令。

```
[root@localhost ~]#service named restart
```

4. 让 DNS 服务自动运行

需要注意的是，使用如上方法启动的 DNS 服务，在下一次系统重新启动后就失效了，需要再次启动。能不能让该服务随系统启动而自动运行呢？答案是肯定的，而且操作也不复杂。

在 Red Hat Enterprise Linux 5 终端窗口，输入 ntsysv 指令，打开“服务”配置小程序，找到 named 服务，并按 Space 键在其前面加个“*”，DNS 服务就会随系统启动而自动运行了，如图 6-3 所示。

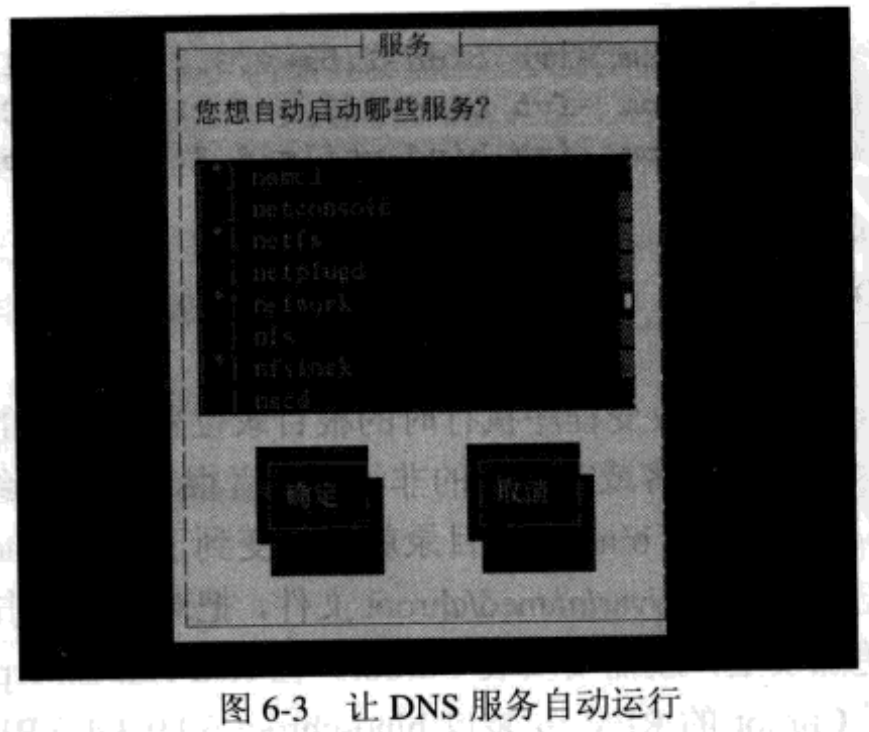


图 6-3 让 DNS 服务自动运行

注意　其实，在 Red Hat Enterprise Linux 5 中启动/停止/重启一个服务有很多种不同的方法，比如按如下语句完成以上功能。

启动 DNS 服务：/etc/init.d/named start

停止 DNS 服务：/etc/init.d/named stop

重启 DNS 服务：/etc/init.d/named restart

6.3　配置主要名称服务器

DNS 服务安装好之后，要使其能够正常运行，还必须先编写 DNS 的配置文件、本机区域反向解析文件、主机区域正、反向解析文件以及 DNS cache 文件。

一般的 DNS 配置文件及正反向解析区域声明文件必须存在以下 4 个文件。

```
/etc/named.conf                    ; #区域配置主文件
/var/named/named.localhost.arpa    ; #本机反向解析区域声明文件
/var/named/named.domain            ; #正向解析区域声明文件
/var/named/named.domain.arpa       ; #反向解析区域声明文件
```

6.3.1　主配置文件

在 Red Hat Enterprise Linux 5 中，Bind 的主配置文件是 named.conf，存放在/etc 目录下。该文件主要用来声明域名服务器的 cache 文件以及正、反向解析区域文件的名称及存放位置。只有在/etc/named.conf 文件中进行了相关定义，才能在域名服务器上设置存取限制，或者进行其他特殊设置。其中，cache 文件主要用来存放 DNS 服务器查询域名解析后的结果（域名与 IP 地址的对应关系）；正向解析区域（Zone）文件主要用来完成域名到 IP 地址的对应转换工作，默认情况是存放在/var/named 目录下的；反向解析区域（Reverse Zone）文件的作用是完成从 IP 地址到对应域名的转换，默认情况下也是放在/var/named 目录下的。

```
# named.conf - configuration for bind
#
# Generated automatically by bindconf, alchemist et al.
options {
    directory "/var/named/";
};
# 指定 named 从 /var/named 目录下读取 DNS 数据文件，这个目录用户可自行指定
# 并创建，指定后所有的 DNS 数据文件都存放在此目录下
zone "." {
    type hint;
    file "named.ca";
};
# 指定 named 从 named.ca 文件中获取 Internet 的根服务器地址
zone "0.0.127.in-addr.arpa" {
    type master;
```

```
    file "0.0.127.in-addr.arpa.zone";
};
# 指定 named 为 127.0.0 网段地址转换主服务器，named.local 文件中包含 127.0.0.*
# 形式的地址到域名的转换数据（127.0.0 网段地址是局域网接口的内部 loopback 地址）
zone "localhost" {
    type master;
    file "localhost.zone";
};
# 指定将包含 localhost 的 DNS 文件数据存放在/var/named/localhost.zone 中
zone "test.localdomain.test"{
    type master;
    file "named.localdomain";
}
# 指定正向解析区域和文件名
zone "1.168.192.in-addr.arpa"{
    type master;
    file "named.localdomain.arpa";
}
# 指定反向解析区域和文件名
```

6.3.2 设置根区域和主区域

在 DNS 服务器当中，为了实现 DNS 服务器的解析功能和数据信息的稳定存储，需要设置根区域和主区域。根区域和主区域的具体设置方法如下。

1. 设置根区域

在主配置文件/etc/named.conf 文件中需要定义根区域。DNS 服务器在处理递归查询时，对于本地区域文件不能进行查询的解析，会将其转到根 DNS 服务器查询。

```
zone "." {
    type hint;
file "named.ca";
};
```

其中“zone”定义 DNS 区域，“hint”设置区域类型，“file”设置根服务器列表文件名。

2. 设置主区域

DNS 服务器中，某个区域（如 localdomain.test）的数据信息所存储的区域，即是主区域。在/etc/named.conf 中有区域正向域名解析文件的定义。

```
Zone "localdomain.test" {
    type  master;
    file  "named.localdomain ";
allow-transfer {
    192.168.0.128;
    192.168.1.128;
};
};
```

区域正向域名解析文件中的声明方式其实并不难懂。下面，就以在 named.localdomain 文件中，对 localdomain.test 这个域中的主机进行域名解析为例，简单介绍如何编写正向域名解析文件。

假设要规划的 DNS 基本信息为：DNS 主机的完整域名是 server.localdomain.test，IP 地址是 192.168.0.128，系统管理员的 E-mail 是 toplinux@server.localdomain.test。该域中需要注册的主机如下。

- server.localdomain.test（IP 地址为：192.168.1.1），别名为 hhwserver.localdomain.test，正式名称为 mail.localdomain.test、www.localdomain.test，提供 DNS、Mail、WWW 和 Samba 服务。
- ftp.localdomain.test（IP 地址为：192.168.1.2），提供 FTP 和 Proxy 服务。
- asp.localdomain.test（IP 地址为：192.168.1.3），Windows Server 2003 主机，提供 ASP 服务。
- mail.localdomain.test（IP 地址为：192.168.1.4），提供 Mail 和 News 服务。

named.localdomain 文件按定义可划分为 5 个部分，各部分含义如下。

第一部分

```
@       IN  SOA     server.localdomain.test.
toplinux.server.localdomain.test.
①       ②  ③           ④       ⑤
(
2009012001  ;Serial     ⑥
10800       ;refresh    ⑦
3600        ;retry      ⑧
604800      ;expire     ⑨
86400       ;default_ttl    ⑩
)
```

这一部分主要用来声明 DNS 主机 FQDN（Fully Qualified Domain Name，完成规范域名）、正向解析区域的序号、管理员账号以及各类网络参数等域名服务器基本信息。

① @符号是区域名称的简写，正向解析区域文件中的@符号，都可以替换为在“named.conf”中声明的相关正向解析区域名称，比如：localdomain.test。

② 此字段定义 class，代表指定的网络类型，通常为“IN”（Internet）。

③ 此字段定义 DNS 的资源记录类型。SOA 是 Starts of Authority 的缩写，主要用来声明所负责区域的数据。

④ 声明 DNS 主机的完整域名。

⑤ 设置 DNS 服务器系统管理员的 E-mail 地址，当 DNS 数据更新时，相关数据会自动发送到指定的 E-mail 邮箱中。

⑥ Serial 是正向解析区域的序列号。一般来说，建议采用日期来作序列号，比如 2009012001 表示 2009 年 1 月 20 日第一版（每一次修改正向解析区域声明文件的数据后，都必须增加 1，比如 2009012002）。值得注意的是，为了使辅 DNS 服务器能够自动更新 DNS cache 数据库，当更改过主 DNS 服务器数据后，必须将 Serial 数据加大，并且保证正向解析区域和反向解析区域的序号一致。

⑦ 每隔一段时间，辅 DNS 服务器与主 DNS 服务器都需要做序列号对比工作，而这一间

隔时间是由 Refresh 设定的。当辅 DNS 服务器发现其序列号比主 DNS 服务器的序列号小，则会自动更新 DNS cache 数据库。可以使用 s（秒）、h（小时）、d（天）、w（星期）等作为单位，但默认的单位是 s（秒）。

⑧ 如果在 Refresh 设定的时间内，未能成功地进行序列号对比，在一定时间内还可以重试，这个时间由 Retry 设定，默认的单位是 s（秒）。

⑨ 当辅 DNS 服务器出现更新不正常的情况时，由辅 DNS 服务器提供的域名解析数据的有效期是由 Expire 设定的，如果辅 DNS 服务器没有与主 DNS 服务器取得联系，时间超过这一设定值，接到客户端的域名解析请求时，就会返回“数据过期，不能使用”的提示。默认时间是 604 800 s（1 w）。

⑩ 主机的 default_ttl（Time to Live）决定了其域名信息在本地 cache 中保存的时间。比如，一台主机的 default_ttl 为 86 400 s（24 h），当某一客户端查询到其 IP 地址，DNS 服务器就会将该主机的域名信息保存在本地 cache 中一天，在其他客户端查询时，自动返回 cache 中的数据。

第二部分

```
@        IN  NS      server.localdomain.test.          ; ①
server   IN  A       192.168.1.1                ; ②
localhost    IN  A       127.0.0.1                     ; ③
loopback     IN  CNAME   localhost                     ; ④
```

这一部分主要用来定义 DNS 域名解析服务器以及本级的正向解析记录。

① 定义 localdomain.test 域的域名解析服务器，该域所有的域名解析都交给 server.localdomain.test 来完成。

② 定义 localdomain.test 域名解析服务器的 IP 地址。

注意 这两行是 DNS 服务器里最重要的声明，若不指明域名服务器信息，将无法提供域名解析服务。

③ 定义本机 IP 地址为 127.0.0.1。

④ 定义 localhost 的别名为 loopback，可以加快 loopback 域名解析的速度。

第三部分

```
mail         IN  MX      10  mail.localdomain.test.
mail1    IN  MX      11  server.localdomain.test.
```

MX 声明了邮件服务器的优先级别，一般来说序号越小的邮件服务器优先级别越高。当有人通过 E-mail 服务器发送邮件时，会按 MX 记录所指定的邮件服务器序号来寻找邮件主机。

第四部分

```
ftp      IN  A       192.168.1.2
asp      IN  A       192.168.1.3
mail         IN  A       192.168.1.4
```

本部分是声明区，所有主机与 IP 地址的对应关系都应该在这里得以声明。在这里，主机名部分可以省略域名，其完整的声明方式如下。

ftp.localdomain.testIN A 192.168.1.2

第五部分

```
mail1   IN  CNAME   server
news    IN  CNAME   mail
proxy   IN  CNAME   ftp
www IN  CNAME   server
samba   IN  CNAME   server
```

本部分用于为主机起不同的别名（Canonical Name）。通过 www.localdomain.test 和 samba.localdomain.test 访问的似乎是不同的主机，但实际上位于同一台服务器。

6.3.3 设置反向解析区域

在明白了区域正向域名解析的原理和配置文件的声明方式之后，就很容易理解区域反向域名解析配置文件了。首先，在/etc/named.conf 中按照如下格式，为区域反向域名解析进行定义。

```
zone "1.168.192.in-addr.arpa"{
    type master;
    file "named.localdomain.arpa";
}
```

named.localdomain.arpa 反向域名解析文件的格式如下。

第一部分

```
@       IN  SOA     server.localdomain.test.
toplinux.server.localdomain.test.
 (2009012001            ;Serial
10800                   ;refresh
3600                    ;retry
604800                  ;expire
86400                   ;default_ttl
)
```

这一部分的编写格式与正向域名解析文件一模一样，事实上，绝大多数情况下，只需要将正向域名解析文件中的内容复制、粘贴过来即可。

第二部分

```
@       IN  NS      server.localdomain.test.
1       IN  PTR     server.localdomain.test.
2       IN  PTR     ftp.localdomain.test.
3       IN  PTR     asp.localdomain.test.
4       IN  PTR     mail.localdomain.test.
```

这一部分主要用来设置 IP 地址和域名的对应关系，是属于反向域名解析服务的精华部分。在正向域名解析配置文件中，用“IN A”来实现域名到 IP 地址的对应，而在反向域名解析的配置文件中，使用“IN PTR”来实现 IP 地址到域名的对应。

6.3.4 根服务器信息文件

当 Bind 接到客户端主机的查询请求时，如果在本地找不到相应的数据，就会通过根服务器进行逐级查询。一般来说，/var/named/named.ca 指定了根服务器，包括 Internet 的根服务器名称和地址。

named.ca 文件经常会随着根服务器的变化而发生变化，因此建议下载最新的版本，将其文件改名为 named.ca，并复制到/var/named/chroot/var/named 目录下。

6.3.5 区域文件

DNS 服务器中存放了区域内所有的数据（比如主机名、对应 IP 地址、刷新间隔和过期时间等），这些数据的载体文件就称为区域文件（区域文件使用“;”符号注释）。

区域文件一般存放在 /var/named/ 目录下。同一区域文件可以存放在多台 DNS 服务器中，一台 DNS 服务器内也可以存放多个区域文件。下面是一个/var/named/chroot/var/named/example.com.zone 文件的完整例子。

```
$TTL 38400
example.com.    IN SOA dns.example.com. admin.example.com. (
2009012003
10800
3600
604800
36000 )
example.com.                IN NS dns.example.com.
dns                         IN A 192.168.1.107
www.example.com.            IN A 192.168.1.8
mail1.example.com.          IN A 192.168.1.108
mail2.example.com.          IN A 192.168.1.109
mail3.example.com.          IN A 192.168.1.112
rhel4                       IN CNAME dns
bbs                         IN CNAME www
samba                       IN CNAME www
example.com.                IN MX 10 mail1.example.com.
example.com.                IN MX 11 mail2.example.com.
example.com.                IN MX 12 mail3.example.com.
```

1. 设置允许客户端缓存来自查询的数据的默认时间

$TTL 选项通常在文件的第 1 行，定义了允许客户端缓存来自查询的数据的默认时间，单位是 s（秒）。在数据不是经常变动的情况下，若其值设置得太小（如几个小时），会引起不必要的 DNS 查询流量，通常情况下设置为几天。

在默认情况下，区域文件的时间数字都以 s（秒）为单位，为了方便理解，也可以用 h（小时）、d（天）和 w（星期）来作单位，36 000 s 和 10 h 的表示方式是一样的。

2. 设置起始授权机构 SOA 资源记录

```
example.com. IN SOA dns.example.com. admin.example.com. (
2009012003
10800
3600
604800
36000 )
```

通常，SOA 资源记录紧跟在$TTL 选项之后或放在区域文件的第 1 行。SOA 即起始授权机构（Start of Authority），是由担任主要名称服务器的 DNS 服务器创建的。在主要名称服务器的区域文件中，是必须要设定的资源记录。SOA 资源记录用来定义域名数据的基本信息和其他属性，比如更新和过期间隔。

（1） 设置所管辖的域名

例中，“example.com.”定义了当前 SOA 所管辖的域名（注意域名后有一个句点号“.”），若不使用域名，也可以使用符号“@”代替。为了便于管理，最好是使用域名。

（2） 设置 Internet 类

例中，“IN”代表类型属于 Internet 类，是固定不可改变的格式。

（3） 设置授权主机名

例中，“dns.example.com.”定义了负责该区域名称解析授权的主机名，通过此定义，DNS 服务器就知道是谁控制了这个区域。在区域文件中，必须有一条授权主机名称的 A 资源记录。例中“dns”主机号（末尾没有句点号）资源记录使用了相对名称，Bind 会在其后面自动添加“.example.com”。区域文件中所有语句都适用于此规则。

任何末尾没加句点号“.”的主机名称都会被视为本区域内的相对域名，如“www.example.com”（末尾没有句点号）会被当作“www.example.com.example.com”解析。为表示完整的主机名，必须在完整的主机地址末尾加上一个句点号，此规则对区域文件中的所有语句都适用。

（4） 设置负责该区域的管理员的 E-mail 地址

例中，负责该区域的管理员的 E-mail 地址定义在“admin.example.com.”中。在 E-mail 地址中应使用句点号“.”代替“@”，这是因为在 DNS 中使用符号“@”代表本区域的名称，避免混淆。

（5） 设置 SOA 资源记录的各种选项

SOA 资源记录的各种选项值，保存在小括号“()”里。其设置的主要目的是为了与辅助名称服务器同步 DNS 数据。注意“(”号必须和 SOA 写在同一行。

- 设置序列号

例中，序列号的值定义为“2009012003”，其格式通常为“年月日+修改次数”（从 0 开始，每次修改主区域文件后即加 1，但不能超过 10 位数字）。当辅助名称服务器与主要名称服务器之间需要进行区域复制操作（即同步辅助名称服务器的 DNS 数据）的时候，就会比较序列号；只有发现当前数值比最后一次更新时的数值大时，才会进行区域复制操作。序列号的作用是标

识该区域是否有数据更新，所以每次修改主区域文件后都应增加序列号的值。

- 设置更新间隔

例中，更新间隔的值定义为“10 800”。其定义了辅助名称服务器与主要名称服务器之间隔多久进行一次区域复制操作。

- 设置重试间隔

例中，重试间隔的值定义为“3 600”。其定义的内容是：当辅助名称服务器与主要名称服务器在更新间隔到期后，仍然无法取得联系时，重试区域复制的间隔。该间隔的值通常应小于更新间隔。

- 设置过期时间

例中，过期时间的值定义为“604 800”。在一定时间内，辅助名称服务器与主要名称服务器之间一直不能取得联系，则放弃重试并丢掉这个区域的数据（因为这些数据有可能失效或错误），这个时间就是过期时间。

- 设置最小默认 TTL

例中，最小默认 TTL 的值定义为“36 000”。其定义的时间是允许辅助名称服务器查询数据缓存的默认时间。当文件开头没有“$TTL”选项时，则以此值为准。

3. 设置名称服务器 NS 资源记录

```
example.com.    IN NS dns.example.com.
```

名称服务器 NS（Name Server）资源记录定义了该域名由哪个 DNS 服务器负责解析，区域权威名称服务器也是由 NS 资源记录定义的。权威名称服务器被其他服务器或客户端当作权威的来源，NS 服务器负责维护和管理所管辖区域中的数据，为 DNS 客户端提供数据查询，并且肯定应答区域内所含名称的查询。

4. 设置主机地址 A 资源记录

```
dns                IN A 192.168.1.107
www.example.com.   IN A 192.168.1.8
mail1.example.com.   IN A 192.168.1.108
mail2.example.com.   IN A 192.168.1.109
mail3.example.com.   IN A 192.168.1.112
```

最常用的记录是主机地址 A（Address）资源记录。该记录定义 DNS 域名对应的 IP 地址信息。上面的例子中，包含两种 A 资源记录定义的方式，即使用相对名称和使用完全规范域名 FQDN（Fully Qualified Domain Name）两种方式。这两种方式的书写形式不同，但在使用上没有任何的区别。Bind 自动在相对名称 dns 的后面加上后缀“.example.com”，相当于完全规范域名的“dns.example.com”。

5. 设置别名 CNAME 资源记录

```
rhel4           IN CNAME dns
bbs             IN CNAME www
samba           IN CNAME www
```

别名 CNAME（Canonical Name）资源记录即规范名称资源记录。为使某些任务更容易执行，CNAME 资源记录允许将多个名称映射到同一台计算机上。

6. 设置邮件交换器 MX 资源记录

```
example.com.        IN MX 10 mail1.example.com.
example.com.        IN MX 11 mail2.example.com.
example.com.        IN MX 12 mail3.example.com.
```

邮件交换器 MX（Mail eXchanger）资源记录指向邮件服务器，当电子邮件系统发邮件时，根据收信人邮件地址后缀来定位邮件服务器。

用户可以设置多个 MX 资源记录，指明多个邮件服务器，优先级别由 MX 后的数字决定，数字越大，则邮件服务器的优先级别越低。优先级高的邮件服务器是邮件传送的主要对象，当邮件传送给优先级高的邮件服务器失败时，可以把它传送给优先级低的邮件服务器。

由于 MX 资源记录只登记了邮件服务器的域名，而在邮件实际传输时，是通过邮件服务器的 IP 地址进行通信的，因此邮件服务器还必须保证在区域文件中有一个 A 资源记录，以指明邮件服务器的 IP 地址，否则会导致传输邮件失败。

6.3.6 反向解析区域文件

反向解析区域文件主要是用于建立 IP 地址映射到 DNS 域名的指针 PTR 资源记录，其文件结构和格式与区域文件类似。反向解析文件的设置如下。

1. SOA 和 DNS 资源记录设置

反向解析区域文件必须包括 SOA 和 DNS 资源记录，用固定格式的反向解析区域 in-addr.arpa 作为域名，结构和格式与区域文件类似。

2. 指针 PTR 资源记录设置

只有在反向解析区域文件中，才能出现指针 PTR 资源记录。正好与 A 资源记录相反，PTR 资源记录将 IP 地址解析成 DNS 域名。与区域文件的其他资源记录类似，也可以使用相对名称和完全规范域名 FQDN，如对于资源记录“177”，Bind 会自动在其后加上“.16.168.192.in-addr.arpa”，相当于完全规范域名的 177.16.168.192.in-addr.arpa。

```
177.16.168.192.in-addr.arpa.   IN  PTR   dns.example.com.
9                              IN  PTR   www.example.com.
178                            IN  PTR   mail1.example.com.
179                            IN  PTR   mail2.example.com.
180                            IN  PTR   mail3.example.com.
```

注意 对于指针 PTR 资源记录，资源记录的后面只有主机号（末尾没有句点号），Bind 就会自动在其后加上“.16.168.192.in-addr.arpa”。

6.3.7 功能实现

DNS 服务具体实现哪些方面的功能呢？下面就具体介绍一下 DNS 的功能实现。

1. 实现负载均衡功能

DNS 服务器中，为同一个域名配置多个 IP 地址（即为一个主机名设置多条 A 资源记录），记录在 DNS 主机文件当中。在应答 DNS 查询时，会按照记录的 IP 地址顺序返回不同的解析结果，把客户端的访问引导到不同的计算机上，使不同的客户端访问不同的服务器，从而达到负载均衡。

例如，在网络中有 3 台内容相同的 FTP 服务器，共同承担客户对网站的访问，IP 地址分别对应 192.168.1.4、192.168.1.11 和 192.168.1.12。要实现 3 台 FTP 服务器的网络负载均衡，只要在 DNS 服务器的区域文件中加入以下 3 条 A 资源记录即可。

```
ftp  IN A   192.168.1.4
ftp  IN A   192.168.1.11
ftp  IN A   192.168.1.12
```

2. 实现直接解析域名

在默认情况下，DNS 服务器不能直接将域名解析成 IP 地址，只能解析完全规范的域名 FQDN。为了便于用户访问，在 DNS 服务器的区域文件中加入了特殊的 A 资源记录，以支持直接解析域名功能的实现。

```
localdomain.test.  IN   A   192.168.1.105
```

或

```
IN   A   192.168.1.105
```

3. 实现泛域名的解析

当一个域名下的所有主机和子域名都被解析到同一个 IP 地址上，这种情况称为泛域名。

为了实现泛域名解析功能，可以在 DNS 服务器的区域文件末尾加入一条特殊的 A 资源记录（通配符“*”表示任何字符）。

```
*.localdomain.test.     IN   A   192.168.1.13
```

或

```
*                       IN   A   192.168.1.13
```

6.3.8 测试 DNS 服务器

首先创建一台 DNS 服务器，其具体配置如下。

DNS 主机的完整域名是 server.localdomain.test，IP 地址为 192.168.1.1，系统管理员的 E-mail 是 toplinux@server.localdomain.test。该域中需要注册的主机分别如下。

- server.localdomain.test（IP 地址为：192.168.1.1），别名为 hhwserver.localdomain.test，正式名称为 mail.localdomain.test、www.localdomain.test，提供 DNS、Mail、WWW 和 Samba 服务。
- ftp.localdomain.test（IP 地址为：192.168.1.2），提供 FTP 和 Proxy 服务。
- asp.localdomain.test（IP 地址为：192.168.1.3），Windows Server 2003 主机，提供 ASP 服务。
- rhel5.localdomain.test（IP 地址为：192.168.1.4），提供 Mail 和 News 服务。

创建 DNS 服务器的步骤如下。

第 1 步，下载、安装 named.ca。

从网上下载域名解析根服务器的最新版本，然后将该文件改名为 named.ca，然后复制到 /var/named/chroot 下。

第 2 步，编辑主配置文件，添加根服务器信息。

输入以下指令，打开 DNS 服务器的主配置文件/var/named/chroot/etc/named.conf。

```
[root@localhost ~]# vi /var/named/chroot/etc/named.conf
```

添加如下信息。

```
options {
        directory "/var/named";
};

zone "." IN {
        type hint;
        file "named.ca";
};
include "/etc/rndc.key";
```

第 3 步，添加 localdomain.test 域信息。

在 named.conf 中添加如下内容，以添加 localdoamian.com 域的相关信息。

```
zone "localdomain.test" {
    type master;
    file "named.localdomain ";
allow-transfer {
    192.168.1.3;
    192.168.1.4;
    };
};
```

第 4 步，创建 named.localdomain 域控制文件并配置相关信息。

按以下指令新建一个域控制文件 named.localdomain。

```
[root@localhost ~]#vi /var/named/chroot/var/named/named.localdomain
```

然后添加以下信息。

```
@       IN  SOA     server.localdomain.test.
toplinux.server.localdomain.test.
(2009012001 ;Serial
10800       ;refresh
3600        ;retry
604800      ;expire
86400       ;default_ttl
)
@       IN  NS      server.localdomain.test.
server  IN  A       192.168.1.1
localhost   IN  A       127.0.0.1
loopback    IN  CNAME       localhost
mail        IN  MX      10  rhel51.localdomain.test.
mail1   IN  MX      11  server.localdomain.test.
ftp     IN  A       192.168.1.2
asp     IN  A       192.168.1.3
mail        IN  A       192.168.1.4
mail1   IN  CNAME   server
news    IN  CNAME   mail
proxy   IN  CNAME   ftp
www IN  CNAME   server
samba   IN  CNAME   server
```

第 5 步，在 DNS 主配置文件中添加反向域名解析信息。

首先，在 DNS 服务器主配置文件中执行以下指令。

```
[root@localhost ~]#vi /var/named/chroot/etc/named.conf
```

然后添加反向域名解析相关信息。

```
zone "1.168.192.in-addr.arpa"{
    type master;
    file "named.localdomain.arpa";
};
```

第 6 步，创建反向域名解析文件并进行配置。

首先执行以下指令，创建相应的反向域名解析文件。

```
[root@localhost ~]#vi /var/named/chroot/var/named/named.localdomain.arpa
```

然后添加如下信息。

```
@       IN  SOA     server.localdomain.test.
toplinux.server.localdomain.test
 (
2009012001  ;Serial
10800       ;refresh
3600        ;retry
604800      ;expire
86400       ;default_ttl
)
@       IN  NS      server.localdomain.test.
1       IN  PTR     server.localdomain.test.
```

```
2          IN  PTR     ftp.localdomain.test.
3          IN  PTR     asp.localdomain.test.
4          IN  PTR     mail.localdomain.test.
```

第 7 步，实现负载均衡功能。

本来，FTP 服务器的 IP 地址是 192.168.1.2，但其性能不能满足客户端大流量的访问。需要新添加两台服务器 192.168.1.106 和 192.168.1.107，实现 DNS 服务器的负载均衡功能，从而提供更加可靠的 FTP 功能。在 DNS 服务器的主配置文件中，添加如下信息。

```
ftp        IN  A   192.168.1.2
ftp        IN  A   192.168.1.106
ftp        IN  A   192.168.1.107
```

第 8 步，实现域名的直接解析。

在默认情况下，DNS 服务器不能直接将域名解析成 IP 地址，只能解析完全规范域名 FQDN。为了实现直接解析域名的功能以便于访问，可以在 DNS 服务器的区域文件中加入一条特殊的 A 资源记录。

```
localdomain.test.     IN      A       192.168.1.13
```

或

```
.                     IN      A       192.168.1.13
```

第 9 步，实现泛域名的解析。

为了实现泛域名解析功能，在 DNS 服务器的区域文件末尾添加如下特殊的 A 资源记录（通配符“*”表示任何字符）。

```
*.example.com.    IN      A       192.168.1.13
```

或

```
*                 IN      A       192.168.1.13
```

第 10 步，主要名称服务器的测试。

在 Red Hat Enterprise Linux 5 中，可以用指令检查、测试域名解析正确与否，以下是 MX 记录、主机名和别名的测试指令。

- 检查 MX 记录

执行以下指令检查安装域的 MX 记录。

```
[root~localhost ~]#host -t mx localdomain.test
```

则会显示如下内容。

```
localdomain.test mail is handled by 10 rhel51.localdomain.test.
localdomain.test mail is handled by 11 server.localdomain.test.
```

- 检查主机名

执行以下指令检查主机名。

```
[root~localhost ~]#host server.localdomain.test
```

则会显示如下内容。

```
serverlocaldomain.test  has address 192.168.1.1
```

- 检查别名

执行以下指令检查主机别名。

```
[root~localhost ~]#host www.localdomain.test
```

则会显示如下内容。

```
www.serverlocaldomain.test  is an alias for server.localdomain.com
www.serverlocaldomain.test  has address 192.168.1.1
```

确定主要名称服务器各部分的配置无误，然后启动 DNS 服务，下面就来进行实际的测试工作。

1. 用 ping 命令测试

启动 DNS 之后，用 ping 命令不定期地测试 DNS 服务器的 IP 地址是否可以 ping 通，如果可以 ping 通，则表明 DNS 配置正确，可以正常工作。

```
[root@localhost ~]# ping 192.168.1.2
PING 192.168.1.2 (192.168.1.2) 56(84) bytes of data.
64 bytes from 192.168.1.2: icmp_seq=1 ttl=128 time=2.20 ms
64 bytes from 192.168.1.2: icmp_seq=2 ttl=128 time=0.408 ms
64 bytes from 192.168.1.2: icmp_seq=3 ttl=128 time=0.270 ms
64 bytes from 192.168.1.2: icmp_seq=4 ttl=128 time=0.287 ms
64 bytes from 192.168.1.2: icmp_seq=5 ttl=128 time=0.289 ms
64 bytes from 192.168.1.2: icmp_seq=6 ttl=128 time=0.309 ms
64 bytes from 192.168.1.2: icmp_seq=7 ttl=128 time=0.286 ms

--- 192.168.1.2 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6001ms
rtt min/avg/max/mdev = 0.270/0.579/2.208/0.666 ms
[root@localhost ~]#
```

2. 在浏览器中测试

利用浏览器进行域名测试，需要先配置一个简单的 apache 服务。在这里配置一个系统默认的 apache 服务，修改其主目录的指向，视其指向目录/var/www/usage。配置 apache 服务器的主机 IP 为 192.168.1.6。完成以上配置之后，就可以用浏览器直接输入 DNS 域名进行测试了。分别在浏览器中输入 DNS 服务器的 IP、DNS 域名的别名，看能否成功打开网页，网页成功打开则说明配置无误。

6.4 配置辅助 DNS 服务器

辅助名称服务器也有客户机提供域名解析功能，但数据不是直接输入，而是一份副本，

从其他服务器（主要名称服务器或其他的辅助名称服务器）中复制过来。所以辅助名称服务器中的数据是无法修改的。区域中设置多台辅助名称服务器的作用如下。

- 提供容错能力。

在主要名称服务器发生故障时，提供服务。

- 减轻主要名称服务器的负担。

在 DNS 客户端较多的情况下，辅助名称服务器为客户端提供查询服务，有效地减轻主要名称服务器的负担。

- 加快查询的速度。

例如，不同地区的分公司 DNS 客户端可以从当地辅助名称服务器进行查询，而不需要通过总公司的 DNS 服务器查询。

配置域名解析辅助服务器，使其满足以下条件。

- 建立 localdomain.test 从区域，设置主要名称服务器的 IP 地址为 192.168.1.1。
- 建立反向域名解析从区域 1.168.192.in-addr.arpa，设置主要名称服务器的地址为 192.168.1.1。

第 1 步，打开 DNS 主配置文件，指令语句如下。

```
[root@localhost ~]#vi /var/named/chroot/etc/named.conf
```

第 2 步，建立 localdomain.test 区域和反向域名解析从区域 1.168.192.in-addr.arpa，在配置文件中添加如下语句。

```
options {
        directory "/var/named";
  };
include "/etc/rndc.key";
zone "." {
        type hint;
        file "named.ca";
};
zone "localdomain.test" {
        type slave;
        file "slaves/named.localdomain ";
        masters {192.168.1.1;};
        };
zone "1.168.192.in-addr.arpa" {
        type slave;
        file "slaves/named.localdonaim.arpa";
        masters {192.168.1.1;};
        };
```

6.5 配置缓存 Cache-only 服务器

配置一台 Cache-only 域名解析服务器，以实现将客户机的所有 DNS 查询转发到 IP 地址为 61.128.128.68 的 DNS 服务器上。

第 1 步，打开 DNS 主配置文件，输入如下指令语句。

```
[root@localhost ~]# vi /var/named/chroot/etc/named.conf
```

第 2 步，将 DNS 服务器配置成 Cache-only 类型，在配置文件中添加如下语句。

```
options {
        directory "/var/named";
        forward only;
        forwarders {
                61.128.128.68;
        };
};
include "/etc/rndc.key";
```

6.6 DNS 客户端的配置

完成 DNS 服务器的配置工作后，并不能够马上运用其服务，接下来应当进行 DNS 客户端的配置工作。

6.6.1 在 Linux 下配置 DNS 客户端

在 Linux 中为客户端指定 DNS 服务器，一般有图形界面和字符模式两种方式。下面分别讲述这两种配置方式。

1. 在字符模式下的配置方式

在字符模式下，直接编辑/etc/resolv.conf 即可，通过 nameserver 选项来指定 DNS 服务器的 IP 地址。用户可以指定多个 DNS 服务器，每个服务器占据一行的位置。

```
nameserver 61.128.128.68
nameserver 192.168.1.1
```

2. 在图形模式下的配置方式

在 Red Hat Enterprise Linux 5 的图形桌面下，可以通过内置的“网络”配置工具来设置 DNS 服务器。

第 1 步，依次单击“系统”→“管理”→“网络”菜单命令，即可显示“网络配置”窗口，如图 6-4 所示。

第 2 步，在“网络配置”窗口中，首先切换至 DNS 选项卡。在默认情况下，Red Hat Enterprise Linux 5 可以配置多达三个的 DNS 服务器，这里根据用户实际情况进行设置，如图 6-5 所示。



图 6-4　打开网络配置的图形界面窗口

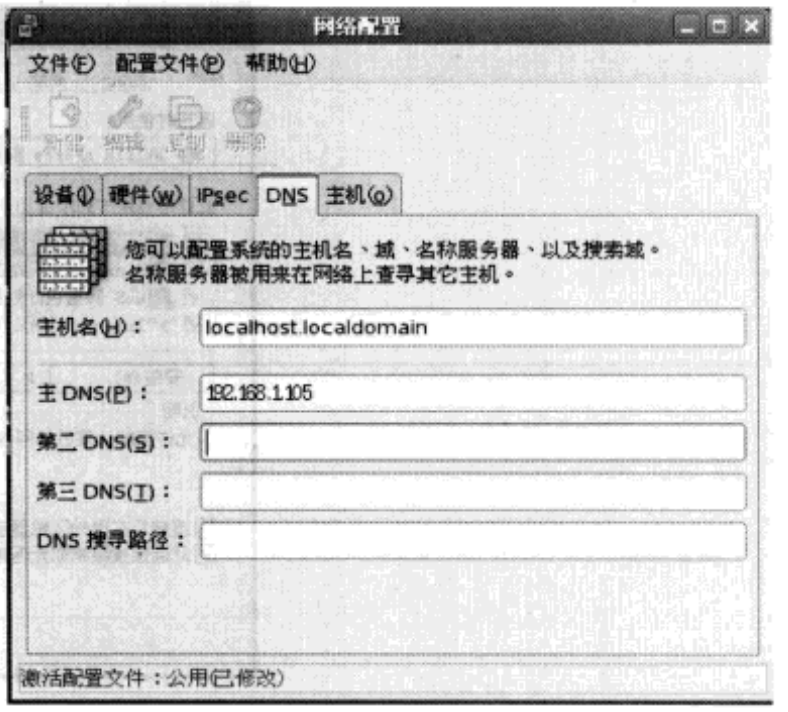


图 6-5　配置 DNS 服务器

6.6.2　在 Windows 下配置 DNS 客户端

Windows 系统，因其本身是图形化操作界面，所以客户端的操作和 DNS 服务器设置都很简单方便。

第 1 步，依次单击“开始”→“设置”→“控制面板”→“网络连接”菜单命令，或者使用鼠标右击桌面上的“网上邻居”图标并选择“属性”菜单，将弹出“网络连接”窗口，如图 6-6 所示。

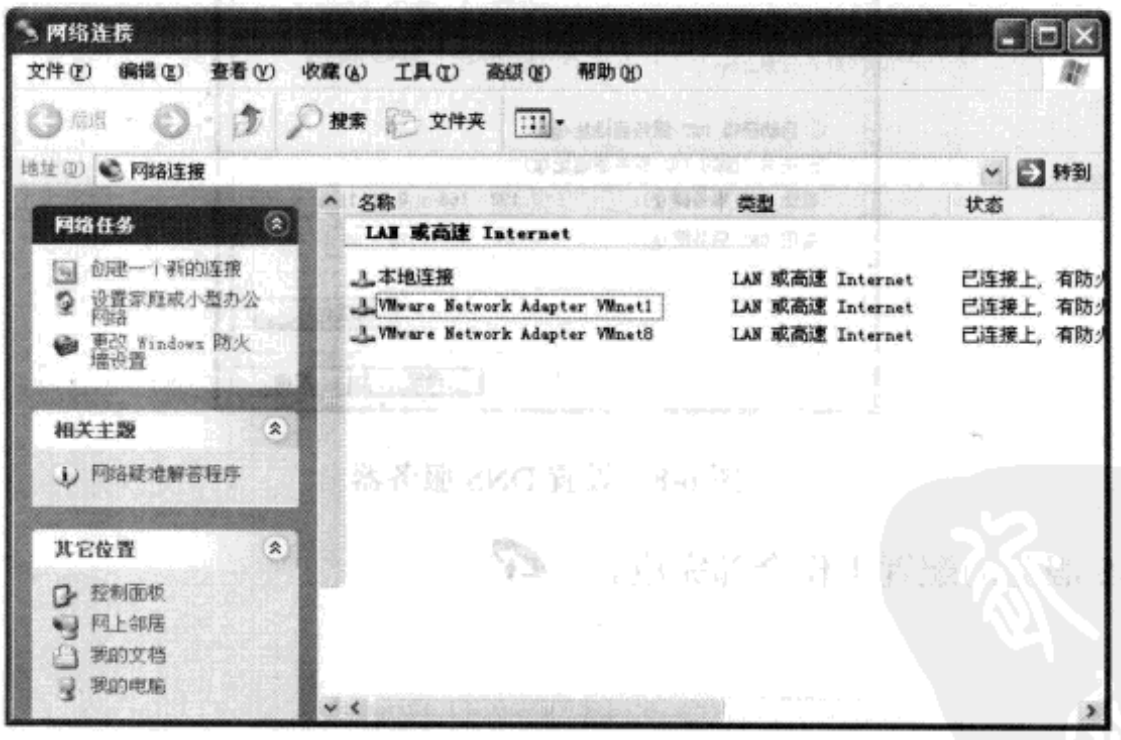


图 6-6　“网络连接”窗口

第 2 步，使用鼠标右击“本地连接”图标，选择“属性”菜单，弹出“本地连接 属性”对话框，如图 6-7 所示。

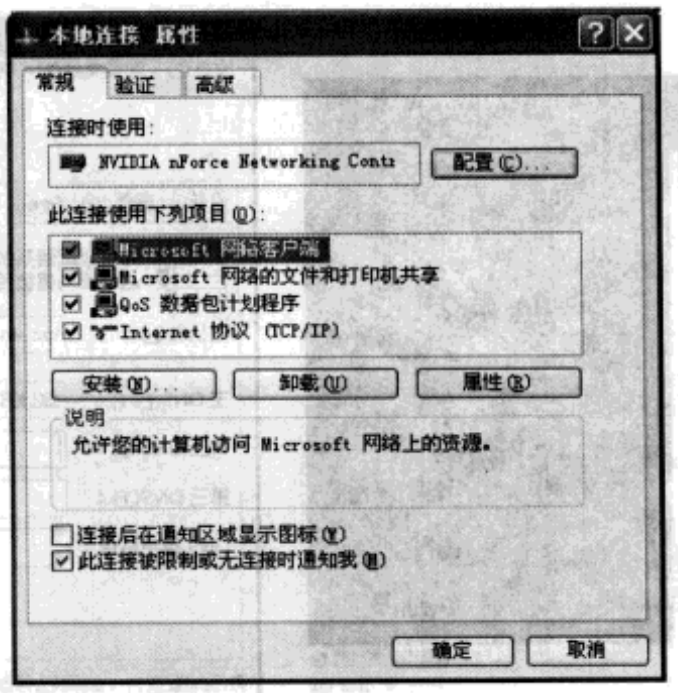


图 6-7 “本地连接 属性”对话框

第 3 步，双击“此连接使用下列项目”列表框中的“Internet 协议（TCP/IP）”复选框，打开“Internet 协议（TCP/IP）属性”对话框，选择“使用下面的 DNS 服务器地址”单选按钮，然后设置 DNS 服务器即可，如图 6-8 所示。

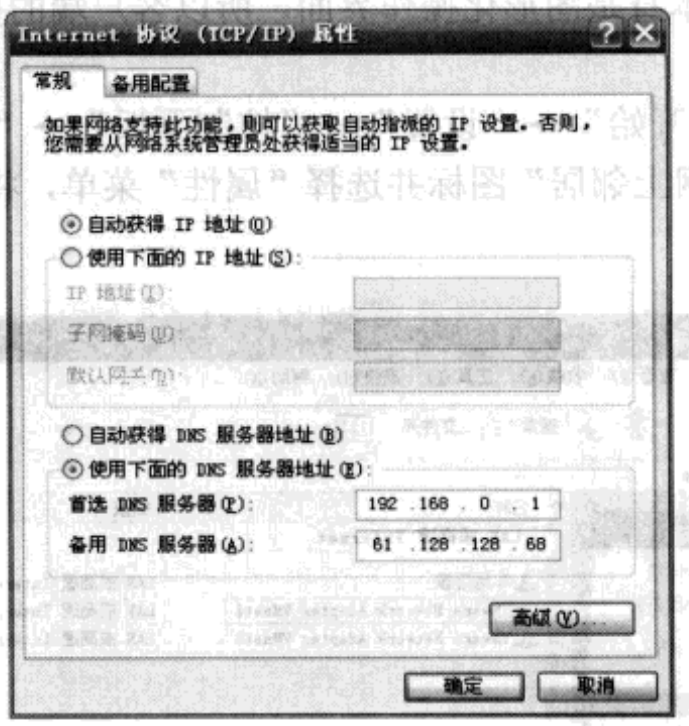


图 6-8 设置 DNS 服务器

至此，DNS 服务的配置工作全部完成。

第 7 章　代理服务器

Internet 方便了信息的传递，让人们足不出户便知天下事，并在方便了人们的工作的同时还丰富了人们的生活，因此得到了广泛应用。在 Internet 终端用户不断增多的情况下，出现了一系列问题，包括 IP 资源短缺、内部网络不安全等。为了应对这些问题，代理服务器应运而生，本章将主要讲解配置代理服务器的相关问题。

7.1　代理服务器概述

代理服务器（Proxy Server）的主要功能就是代理网络用户去取得网络信息，网络信息则是通过代理服务器进行中转的。代理处于服务器与客户机之间，对于服务器而言，代理相当于客户机，其提出请求，服务器响应；而对于客户机而言，代理相当于服务器，其接受客户机的请求，并接受目的主机的相关信息，储存于代理服务器的硬盘中，然后代理服务器再将信息发送给要求信息的客户。

7.1.1　代理服务器的工作流程

代理服务器是介于浏览器和 Web 服务器之间的一台服务器，是以 HTTP 协议为基础的、建立在 TCP/IP 协议应用层上的一种服务软件。在没有代理服务器的情况下，用户使用网络浏览器直接去连接其他 Internet 站点取得网络信息时，需要发出请求信号以便获得响应，然后对方再把相关信息传送回来。代理服务器出现之后，浏览器不是直接到 Web 服务器取回网页而是向代理服务器发出请求，请求信号会先送到代理服务器，由代理服务器取回浏览器所需要的信息并传送给用户的浏览器。

其工作过程可以简要的概括为 4 步。

第 1 步，客户端向服务器发送的请求到达代理服务器。

第 2 步，代理服务器把请求转发给客户端真正需要联系的服务器。

第 3 步，服务器向代理服务器返回响应。

第 4 步，代理服务器把响应返回给客户端。

下面将以一个内网客户端通过代理服务器访问 Internet 上的远程 Web 服务器的过程为例来讲解代理服务器的工作原理，如图 7-1 所示。

首先，启动代理服务器，初始化代理服务器缓存。此时代理服务器主进程会一直监听某个绑定的端口。

然后，客户端 A 向代理服务器发出访问 Web 服务器资源的请求。

其次，代理服务器主进程监听到客户端 A 所发出的请求后，会创建一个子进程应对客户端 A 发出的请求。主进程则继续执行监听工作。

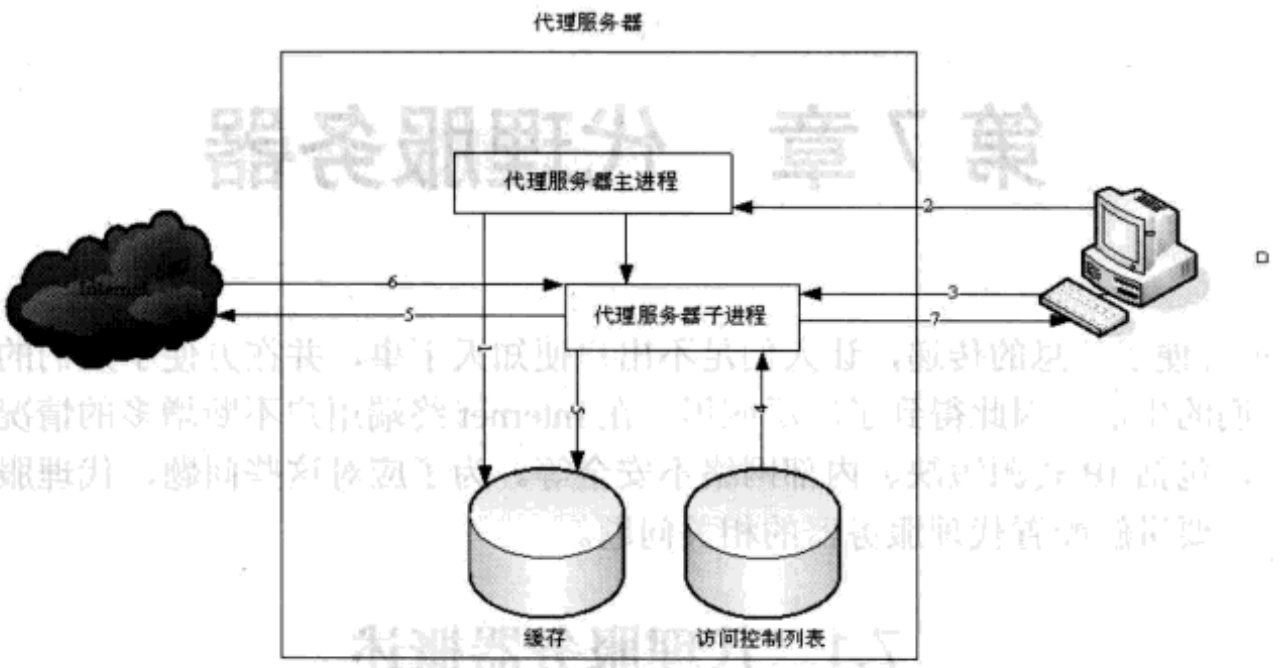


图 7-1　代理服务器工作原理

第四，代理服务器子进程与客户端建立连接，读取并解析客户请求。在代理服务器中，预设有访问规则列表，用于检验当前接收到的请求是否求满足规则约束，如果满足，则可以在代理服务器缓存中查找是否存在需要的信息。

第五，子进程开始查询缓存，根据查询结果作出相应的处理。

主要会出现两种情况，一是在缓存中命中客户端 A 请求的信息，并且该信息没有过期，则将该信息直接传送给客户端。依据具体的缓存更新算法更新代理服务器缓存。二是缓存中命中客户端 A 请求的信息，但是该信息已经过期，则该子进程代理客户端 A 向目标 Web 服务器发出请求。

第六，目标 Web 服务器响应代理服务器子进程发来的请求，同时代理服务器子进程依据具体的缓存更新算法更新代理服务器缓存。

第七，代理服务器子进程将目标 Web 服务器回应的信息转送给客户端 A。

第八，代理服务器子进程处理完客户端 A 的代理服务后，进程自动终止，所有相关资源（各种连接）全部释放。

7.1.2 代理服务器的作用及相关应用

代理服务器是 Internet 链路级网关所提供的一种重要的安全功能，其主要工作在开放系统互连（OSI）模型的对话层。其主要的功能及应用有如下几个方面。

1. 共享网络

共享上网是代理服务器最常见的用途之一。在如今的工作环境中，大量的用户都在通过各种代理服务器（如 Squid、SyGate、WinGate、ISA、CCProxy 以及 NT 系统自带的网络共享等代理服务器）访问外部站点的信息。这些代理服务器也都能提供企业级的文件缓存、复制和地址过滤等服务。充分利用局域网出口的有限带宽，加快内网用户的访问速度，解决仅仅有一条线路、一个公有 IP 的公有 IP 资源严重不足的问题，满足局域网众多用户同时共享上网的需求。

2. 突破访问限制

通过代理服务器，教育网等网络用户可以访问国外网站。

3. 访问代理

在现在的网络环境中，网络常常会出现拥挤或故障。因此，可以通过代理服务器绕道访问目的站点。

4. 提高浏览速度和效率

代理服务器中通常会备份有相当数量的缓存文件，如果用户当前所访问的数据在代理服务器的缓存文件中，则可直接读取，而无需再连接到远程 Web 服务器上。这样可以达到加快访问网站速度，节约通信带宽的目的。

5. 掩藏身份

代理服务器使内部用户访问 Internet 时受到保护，内部网的用户要对外发布信息就需要使用代理服务器的反向代理功能。这样就不会影响内部网络的安全性能，起到了掩藏身份的目的。

黑客的活动很多都是通过代理服务器，运用扫描、刺探等方法对局域网内机器进行渗透的。黑客攻击的时候都是中转了多次才攻击目标机器。隐藏了身份，就确保了自己的安全。

6. 充当防火墙

因为所有使用代理服务器的用户都必须通过代理服务器访问远程站点，因此在代理服务器上就可以设置相应的限制，以便过滤或屏蔽掉某些信息。

7. 防止攻击

通过代理服务器完成内部主机的访问使内部主机地址等信息不会发送到外部，隐藏了自己的真实地址信息，还可隐藏自己的 IP，保护了内部主机。

7.2 安装 Squid Server

在安装 Squid 之前，有必要认识该软件。Squid 是 Linux 和 UNIX 平台下著名的、最为流行的代理服务器软件之一，具有权限管理灵活、性能好和效率高等特点，也能支持 FTP、SSL 等多种协议的代理。同时也可以使用 ICP、HTCP、CARP 及 Cache Digests 等协议和方式及其他运行 Squid 的服务器进行协同操作。

7.2.1 Squid 简介

Squid 是由众多在 Internet 上的开发人员共同研发完成的，其前身是美国政府 ARPA 出资大力发展的缓存服务器 Harvest Research Project 研究计划。Squid 的官方网站是 http://www.squid-cache.org/，如图 7-2 所示。

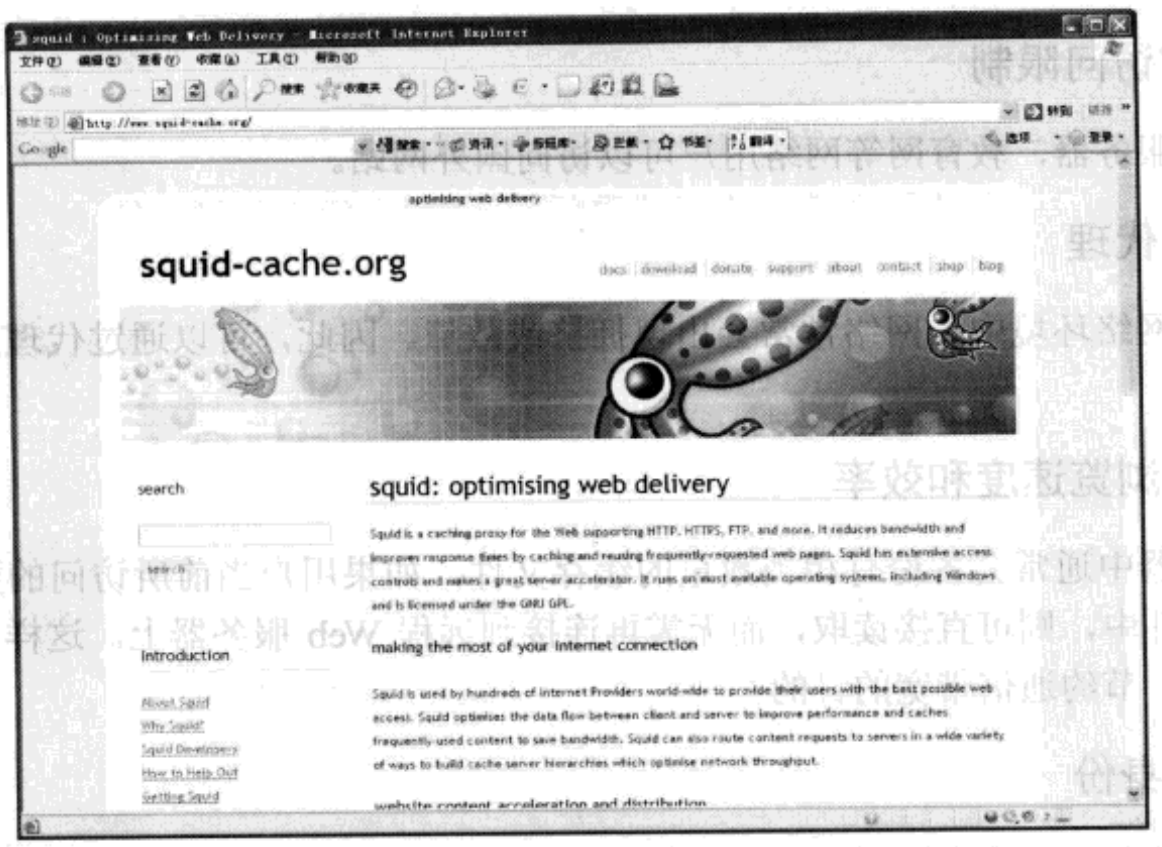


图 7-2　Squid 官方网站

1. Squid 优越性

Squid 在内存里保存频繁访问的对象，并且在硬盘内维护了一个健壮的访问对象数据库，并接受客户软件的连接，然后把连接转发到相应的网络主机上去。Squid 在向用户返回网络主机信息的同时，也在本地的磁盘上做一份备份。因此，如果有同样的内容再次请求，那么 Squid 就不用再转发请求而是直接使用磁盘中的备份资源。如此一来，用户就会觉得速度非常快。另外，Squid 系统使用访问控制列表（ACL）和访问权限列表（ARL）进行权限管理和内容过滤，使用访问控制清单和访问权限清单来禁止内部网的主机访问有威胁的网站。

2. Squid 对硬件和操作系统的要求

Squid 可以运行在所有流行的 Linux 系统和 UNIX 系统上，也可以运行在 Windows 系统上，但对 Windows 的支持还在不断持续改善。

Squid 对硬件要求不高。其中最为重要的资源就是内存和磁盘空间，内存短缺会严重影响性能，而磁盘空间受限意味着缓存目标受限同时命中率下降。另外，磁盘转速和 CPU 速度也对系统性能有很大的影响，但并不是影响性能的关键因素。

7.2.2 Squid Server 的安装

在对 Squid 有了基本认识后，下面将学习如何安装 Squid Server，在安装 Squid 服务之前有必要先确认系统中是否已经安装 Squid 服务，可以通过执行以下命令实现查询。

```
[root@localhost ~]#rpm -q squid
```

执行结果如下所示，则表明 Squid 服务已经安装，且版本为 squid-2.6.STABLE6-5.el5_1.3.i386.rpm。

```
[root@localhost ~]# rpm -q squid
squid-2.6.STABLE6-5.el5_1.3
```

如果系统中没有安装 Squid 服务，则需要进行相应的安装，具体方法如下。

首先，到相关网站获取 Squid 最近的 RPM 包文件，这里以 squid-3.0.STABLE7-1.el5.i386.rpm 安装为准。

获取相应安装包文件后，使用以下命令进行安装 Squid 服务。

```
[root@localhost ~]#rpm -ivh squid-3.0.STABLE7-1.el5.i386.rpm
```

执行命令后的结果如下所示，表明 Squid 服务成功安装。

```
[root@localhost ~]#rpm -ivh squid-3.0.STABLE7-1.el5.i386.rpm
Preparing...      ########################################### [100%]
   1: squid       ########################################### [100%]
```

成功完成了 Squid Server 服务器的安装工作后，在目录“/usr/local/squid/etc”中会自动产生一个 “squid.conf”配置文件。每行以一个配置参数开始，后面跟着数字值或者关键字。在读取配置文件时，Squid 会忽略空行和注释掉的行，另外 Squid 还对配置文件中的参数的大、小写敏感。

除了使用 RPM 包安装方式进行安装外，还可以使用源代码的安装方式进行安装，使用源代码安装的具体方法如下。

找到 Squid 的官方网站“http://www.squid-cache.org/Versions/v3/3.0/”，下载 Squid 3.0（squid-3.0.STABLE10.tar.gz），如图 7-3 所示。

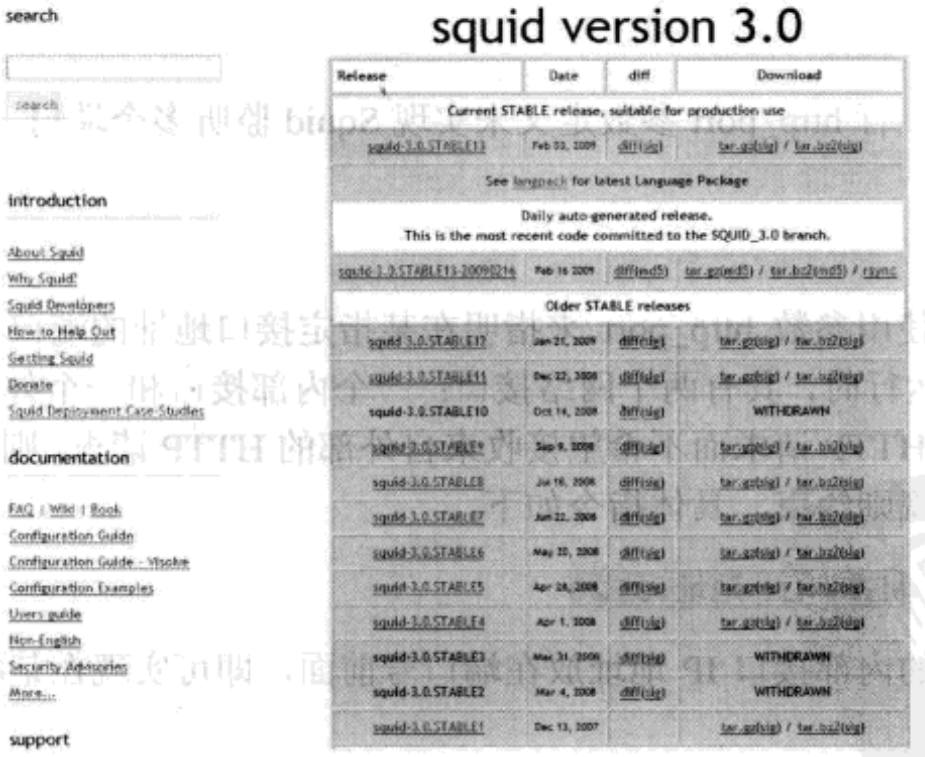


图 7-3　下载 Squid 3.0

解压该文件并进入“squid-3.0.STABLE10”目录。进入目录后，使用“./configure”命令进行配置测试，然后使用 GNU make 工具编译源代码，编译代码成功后，即可使用 make 命令进行安装。最后还可以使用“groupadd”命令添加 Squid 用户组以及使用命令“useradd”添加用户。

注意 在进行配置测试时，命令“./configure”会自动查询你的系统配置情况以及你的系统中使用的头文件。不加参数的“./configure”会把 Squid Server 安装在目录“/usr/local/squid”中；如果你想使用其他目录，用如下命令“./configure --prefix=/ app/other/directory”，则会把 Squid Server 安装在目录“/app/other/directory”中。

7.3 Squid Server 的基本配置

绝大多数服务的配置工作都是通过配置文件实现的，此服务也不例外，Squid Server 的基本配置都是通过“squid.conf”配置文件实现的。该配置文件是在成功安装 Squid Server 后在目录“/usr/local/squid/etc”中自动生成的，此配置文件包括了各个配置选项的详细说明，可以通过修改该文件来满足用户的不同需要，接下来将对几个最常用的配置选项进行详细的介绍。

7.3.1 设置监听的 IP 和端口

“squid.conf”配置文件中的 http_port 参数定义了 Squid Server 在哪个 IP 地址的哪个端口监听来自客户机的 HTTP 请求。此配置文件中的该参数默认的配置是在本机的 3128 端口进行监听。如果不指定 IP 地址，就不能监听来自内部网络客户的请求，因此要将 Squid Server 作为 Web 服务器的加速器使用。可以通过以下设置完成。

```
http_port 80
```

也可以通过附加一行 http_port 参数定义来实现 Squid 监听多个端口。

```
http_port 1324
http_port 8080
```

除此之外，可以使用参数 http_port 来指明在某指定接口地址的端口上进行监听。例如，当 Squid 作为防火墙运行时，其有两个网络接口：一个内部接口和一个外部接口。有时可能只需要接收来自内部的 HTTP 请求而不希望接收来自外部的 HTTP 请求，则可以使 Squid 仅仅监听内部接口来达到此规则约束，具体指令如下。

```
http_port 内部主机接口 IP 地址:3128
```

通过把当前主机的内部接口 IP 地址放在端口号前面，即可实现在某指定接口地址的端口上进行监听。

7.3.2 设置缓存大小

cache-mem 选项用于设置代理服务器使用多少物理内存作为缓存，Squid 会将最常用的一些缓存放到该块内存中。参数 cache_dir 是 squid.conf 配置文件里最重要的指令之一，告诉 Squid 以何种方式存储 cache 文件到磁盘的什么位置。

1. cache_dir 参数

（1） 参数 cache_dir 的定义格式

参数 cache_dir 的定义格式如下。

```
cache_dir scheme directory size L1 L2 [options]
```

（2） 具体子参数

scheme：Squid 支持许多不同的存储机制，默认的是 ufs。根据操作系统的不同，可以选择不同的存储机制。在./configure 目录下，必须使用--enable-storeio=LIST 选项来编译其他存储机制的附加代码。

directory：该参数是文件系统目录，Squid 将 cache 对象文件存放在此目录下。推荐在每个物理磁盘中仅仅设置一个 cache 目录。

size：该参数指定了 cache 目录的大小。是 Squid 能使用的 cache_dir 目录的空间上限。

L1 L2：对 ufs、aufs 和 diskd 机制，Squid 在 cache 目录下创建二级目录树。L1 和 L2 参数指定了第一级和第二级目录的数量。默认的是 16 和 256。

options：Squid 有两个依赖于不同存储机制的 cache_dir 选项：read-only 标签和 max-size 值，具体说明如表 7-1 所示。

表 7-1 依赖于不同存储机制的 cache_dir 选项

选　项	说　明
read-only 标签选项	指示 Squid 继续从 cache_dir 读取文件，但不往里面写新目标。在 squid.conf 文件里表示为：cache_dir ufs /cache0 7000 16 256 read-only。该项适用于把 cache 文件从一个磁盘迁移到另一个磁盘
max-size 值选项	指定存储在 cache 目录里的最大目标大小。例如：cache_dir ufs /cache0 7000 16 256 max-size=1048576 max-siz 值是以 KB 为单位的。一般情况下，不必增加该选项

存储在磁盘上的对象的置换是通过参数 cache_swap_low 和 cache_swap_high 控制的。其值是最大 cache 体积的百分比，而此最大 cache 体积来自所有 cache_dir 大小的总和。

```
cache_swap_low 80
cache_swap_high 90
```

可以通过请求 cache 管理器的 storedir 页面来查看当前磁盘的使用状况。在稳定状态下，磁盘使用总是相对接近 cache_swap_low 值。如果磁盘使用总量低于 cache_swap_low，Squid 将不会删除 cache 目标；如果 cache 体积增加，Squid 会逐渐删除目标。

2. 计算使用内存情况

Squid 是这样来计算使用多少内存的：Squid 本身的进程大概需要 10~20 MB，如果用户设置的 cache 目录的大小是 500 MB，那么其放在内存里的 hash 索引大概需要 20 MB 左右，然后再加上这里设置的 cache_mem 的值。在机器条件许可的情况用户总希望 cache_mem 越大越好。

注意 官方文档建议实际内存大小应该是 Squid 所需要总内存的两倍以上。

7.3.3 访问控制设置

Squid 的配置文件“squid.conf”在默认情况下是拒绝所有客户端的请求的，为了使所有终端客户通过 Squid 代理服务器访问 Internet 资源，则可以在所有终端使用该代理服务器之前在“squid.conf”文件里加入附加的访问控制规则。

附加访问控制规则最简单的实现方法是：定义一个针对终端客户 IP 地址的访问控制列表（Access Control List，ACL）和一系列访问规则，告诉 Squid 服务器允许来自哪些 IP 地址的 HTTP 请求。访问控制列表 ACL 是用来过滤进出代理服务器数据的，是 Squid 进行网络控制的有力工具。构成访问控制列表 ACL 的基本元素是 acl 参数定义，用来指定包括 IP 地址、端口号、主机名和 URL 匹配等变量。每个 acl 参数有相应的名称，在编写访问控制规则时需要引用。

1. acl 参数定义的语法格式

acl 参数定义的语法格式如下。

```
acl 列表名称 列表类型 [-i] 列表值1 列表值2 ...
```

2. 具体参数

列表名称：用于区分 Squid 的各个访问控制列表 ACL，任何两个访问控制列表 ACL 不能定义相同的列表名称。列表名称可以任意定义，不过建议大家尽量使用有意义的、便于理解的列表名称。

列表类型：是可以被 Squid 识别的类型。Squid 支持的控制类别很多，在这里按照重要性的降序来列举，具体参数如下。

src：IP 地址在访问控制元素里是最普遍使用的。src 类型指向终端客户的源 IP 地址。

dst：dst 类型指向目标服务器 IP 地址。

myip：myip 类型指向 Squid 的 IP 地址，与客户连接。

dstdomain：检查请求 url 里的主机名。可以使用基于域名的访问控制去阻塞对某些站点的访问，去控制 Squid 如何转发请求，以及让某些响应不可缓存。

srcdomain：客户机所在域。

port：port ACL 允许定义单独的端口或端口范围。使用 port ACL 来限制对某些目标服务器端口号的访问。

myport：Squid 也有 myport ACL。port ACL 指向目标服务器的端口号，myport 指向 Squid 自己的端口号，用以接受客户请求。如果在 http_port 指令里指定不止一个端口号，那么 Squid 就可以在不同的端口上监听。

method：method ACL 指 HTTP 请求方法（包括 GET、POST、PUT 等方法）。

proto：该类型指明 URI 访问（或传输）协议。Squid 支持的协议有：HTTP、HTTPS（same as HTTP/TLS）、FTP、Gopher、URN、Whois 和 Cache Object。

time：time ACL 允许控制基于时间（每天中的具体时间，和每周中的每天）的访问。日期以单个字母来表示，如表 7-2 所示。

表 7-2 time 参数的日期代码表

日 期 代 码	日　　期
S	Sunday（星期天）
M	Monday（星期一）
T	Tuesday（星期二）
W	Wednesday（星期三）
H	Thursday（星期四）
F	Friday（星期五）
S	Saturday（星期六）
D	All weekdays(M-F)（星期一到星期五）

ident：ident ACL 匹配被 ident 协议返回的用户名。

poxy_auth：通过外部程序进行用户认证。

Squid 当前支持三种技术协议以接受用户验证：HTTP 基本协议、数字认证协议和 NTLM。

src_as：该类型检查终端客户的源 IP 地址所属的具体 AS 号。

dst_as：dst_as ACL 经常与 cache_peer_access 参数一起使用。在该方法中，Squid 使用与 IP 路由一致的方式转发 cache 丢失。考虑某 ISP，比其他 ISP 更频繁地更换路由，每个 ISP 处理各自的 cache 代理，这些代理能转发请求到其他代理。理论上，ISP A 将 ISP B 网络里主机的 cache 丢失转发到 ISP B 的 cache 代理。

maxconn：maxconn ACL 指来自终端客户 IP 地址同时连接的最大连接数。

arp：arp ACL 用于检测 cache 客户端的 MAC 地址（以太网卡的物理地址）。地址解析协议（ARP）是主机查找对应于 IP 地址的 MAC 地址的方法。

srcdom_regex：srcdom_regex ACL 允许使用正则表达式匹配终端客户的域名。这与 srcdomain ACL 相似，都具有相同的限制，即对某些客户地址不能反向解析到域名。

url_regex：url_regex ACL 用于匹配请求 URL 的任何部分，包括传输协议和目标服务器主机名。例如，如下 ACL 匹配从 FTP 服务器的 exe 文件请求。

```
acl FTPEXE url_regex -i ^ftp://.*\. exe$
```

urlpath_regex：urlpath_regex 与 url_regex 非常相似，不过传输协议和主机名不包含在匹配条件里。这让某些类型的检测非常容易。

browser：browser ACL 对 user-agent 头执行正则表达式匹配。

req_mime_type：req_mime_type ACL 指客户 HTTP 请求里的 Content-Type 头部。req_mime_type ACL 值是正则表达式。该类型头部通常仅仅出现在请求消息主体里。可以使用该类型 ACL 来检测某些文件上传，和某些类型的 HTTP 隧道请求。POST 和 PUT 请求可能包含该头部，但 GET 从不包含该头部。

rep_mime_type：rep_mime_type ACL 指目标服务器的 HTTP 响应里的 Content-Type 头部，且仅在使用 http_reply_access 规则时才有用。该 ACL 基于服务器响应，而所有的其他访问控制形式是基于客户端请求的。

ident_regex：ident_regex 允许使用正则表达式，代替严格的字符串匹配，这些匹配是针对 ident 协议返回的用户名进行的。

proxy_auth_regex：该 ACL 允许对代理认证用户名使用正则表达式。

[-i]：表示忽略列表值的大、小写。

列表值：针对不同的类型，列表值的内容是不同的。例如，对于类型 src 或 dst，列表值的内容是某台主机的 IP 地址或子网地址；对于类型 time，列表值的内容是时间；对于类型 srcdomain 和 dstdomain，列表值的内容是主机域名。

7.3.4 指定 Squid 可以使用内存的理想值

在/etc/squid/squid.conf 配置文件中，可利用 cache-mem 选项设置代理服务器使用多少物理内存作为缓存。用户可以根据自己的需要设置高速缓存的大小，若该代理服务器只用于代理网内的其他用户上网，则可以将高速缓存设置大一些，否则将会影响主机的整体性能。

7.3.5 其他参数设置

Squid Serve 除了有以上几个较为重要的配置参数以外，还具有以下功能。

- 设定 cache_effective_user 参数

设定使用缓存的有效用户。如果没有设置 cache_effective_user，而是以 root 来启动 Squid，Squid 便使用 nobody 作为默认值。无论选择什么用户 ID，都要确认对以下目录有读访问权：/usr/local/squid/etc/usr/local/squid/libexec 和/usr/local/squid/share。同时该用户 ID 也必须有对日志文件和缓存目录的写访问权。

- 设定 cache_effective_group 参数

设定使用缓存的有效用户组。Squid 也有一个 cache_effective_group 指令，但可以不必设置。默认情况下 Squid 使用 cache_effective_user 的默认组（从/etc/passwd 文件读取）。

- 设定 DNS 服务器的地址

为了能够使 Squid 解析域名，必须告诉 Squid 有效的 DNS 服务器。

dns_nameservers 52.135.55.121

- 设置日志文件路径

默认的日志目录是 Squid 安装位置下的 logs 目录。

必须确认日志文件所存放的磁盘位置空间足够。在 Squid 写日志时如果接收到错误，其会退出或重启，此时管理员应重视，检查是否有异常行为出现，特别是当前的系统是否被滥用或者被攻击。

注意 Squid 有三个主要的日志文件：cache.log、access.log 和 store.log 。具体介绍见 7.6.2 Squid 代理服务器日志系统的构成。

某些操作系统对单个文件强制执行 2 GB 的大小限制，即使有充足的磁盘空间，超过该限制会导致写错误，这样 Squid 就会退出。为了保证日志文件大小合理，应该创建任务来有规律地重命名和打包日志。

- 设置运行 Squid 主机的名称

vsible_hostname 参数定义了运行 Squid 的主机的名称。当访问发生错误时，该参数的值会显示在错误提示网页中，建议输入主机的 IP 地址。

visible_hostname 192.168.1.4

- 设置管理员的联系信息

设置 cache_mgr 参数是作为对终端用户的帮助。它是一个 E-mail 地址，假如问题发生，用户能以邮件形式发送信息。cache_mgr 地址默认出现在 Squid 的错误消息里。

- 允许或拒绝某个访问控制列表的 HTTP 请求

Squid 会针对客户的 HTTP 请求检查 http_access 规则，根据定义的访问控制列表，使用 http_access 参数决定哪个访问控制列表被禁止或者被许可。

注意　http_access 参数的基本格式如下。

http_access [allow | deny] 访问控制列表名称

[allow | deny]：定义允许或禁止指定的访问控制列表。
访问控制列表名称：需要 http_access 控制的 ACL 名称。
例如，定义允许名称为 all 访问控制列表的 HTTP 请求。

```
http_access allow all
```

7.3.6 设置错误提示为中文

默认情况下，Squid 代理服务器向客户端返回的错误信息都是英文，要想将错误信息提示设置成中文，则可以按如下操作来实现。

找到存放有 Squid 的各种语言版本的错误提示信息的文件（一般是在/usr/share/squid/errors 中），按目录存放在不同的子目录中。其中，/etc/squid/errors 连接文件决定了 Squid 的错误提示信息，该文件默认指向/usr/share/squid/errors/English 目录，因此只需要修改该连接文件指向/usr/share/squid/errors/Simplify_Chinese 目录，即可以中文的形式显示错误提示信息，使用如下命令。

```
[root@localhost ~]# mv /etc/squid/errors /etc/squid/errors.bak
[root@localhost ~]# ln -s /usr/share/squid/errors/Simplify_Chinese
/etc/squid/errors
[root@localhost ~]# /etc/init.d/squid reload
```

7.3.7 初始化 Squid Server

成功安装并配置好 Squid 服务器后，若要使 Squid 正常工作，则需要对 Squid 进行一些初始化操作包括初始化硬盘缓冲区目录，还需要对 Squid 进行测试，这些工作具体如下。

1. 初始化硬盘缓冲区（cache 目录）

要使 Squid 在硬盘中缓冲终端客户访问目标服务器的内容，则可以在初次运行 Squid 之前，或者修改了 cache_dir 设置后初始化硬盘缓冲区目录。初始化硬盘缓冲区目录的命令如下。

```
[root@localhost var]# /usr/local/squid/sbin/squid -z
```

执行该命令后，即可进行硬盘 cache 目录的初始化，该命令表示在每个 cache_dir 下面创建了所需的子目录，同时可以使用“ls”命令查看目录“/usr/local/squid/var/cache”的内容，将会看到 Squid 已经根据配置文件的定义建立了目录结构，如图 7-4 所示。

```
Making directories in /usr/local/squid/var/cache/00
Making directories in /usr/local/squid/var/cache/01
Making directories in /usr/local/squid/var/cache/02
Making directories in /usr/local/squid/var/cache/03
Making directories in /usr/local/squid/var/cache/04
Making directories in /usr/local/squid/var/cache/05
Making directories in /usr/local/squid/var/cache/06
Making directories in /usr/local/squid/var/cache/07
Making directories in /usr/local/squid/var/cache/08
Making directories in /usr/local/squid/var/cache/09
Making directories in /usr/local/squid/var/cache/0A
Making directories in /usr/local/squid/var/cache/0B
Making directories in /usr/local/squid/var/cache/0C
Making directories in /usr/local/squid/var/cache/0D
Making directories in /usr/local/squid/var/cache/0E
Making directories in /usr/local/squid/var/cache/0F
```

图 7-4　硬盘 cache 的目录结构

注意　在该阶段属主和许可权是通常遇到的问题。Squid 在特定的用户 ID 下运行，这在 squid.conf 文件里的参数 cache_effective_user 里指定。用户 ID 必须对每个参数 cache_dir 目录有读和写权限。否则，将报如下错误。

```
Creating Swap Directories
FATAL: Failed to make swap directory /usr/local/squid/var/cache/00:
(13) Permission denied
```

因此，应该确认“/usr/local/squid/var/cache”目录的所有资料必须都可被 squid.conf 给定的用户 ID 写访问。

使用-X 选项可以观察 cache 目录初始化过程，执行以下命令。

```
[root@localhost var]# /usr/local/squid/sbin/squid -zX
```

注意　cache 目录初始化工作可能花费一些时间，这取决于 cache 目录的大小和数量，以及磁盘驱动器的速度。

2. 在终端窗口中测试 Squid

成功初始化了 cache 目录后，就可以在终端窗口里运行 Squid，将日志记录重定向到标准错误，以方便查找错误和问题。

在终端窗口运行 Squid 的命令如下。

```
[root@localhost ~]# /usr/local/squid/sbin/squid -N -d1
```

各选项具体说明如下。

-N 选项表示保持 Squid 在前台运行。

-d1 选项表示在标准错误里显示 1 级别的调试信息。

通过以上设置，用户就能轻易地找出任何错误或问题，并能确认 Squid 是否成功启动。在显示信息的开始几行里，可以方便检查出警告信息。而最普通的错误是文件/目录许可问题，和配置文件语法错误。

3. 使用 Squidclient 程序

Squidclient 程序是随 Squid 一起发布的，如果 Squid 正常工作，执行以下命令后，Squid 的主页 html 文件将会在终端窗口里滚动，如图 7-5 所示。

```
[root@localhost ~]# /usr/local/squid/bin/squidclient http://www.squid-cache.org
```

```
[root@localhost ~]# /usr/local/squid/bin/squidclient http://www.squid-cache.org
HTTP/1.0 403 Forbidden
Server: squid/3.0.STABLE10
Mime-Version: 1.0
Date: Mon, 03 Nov 2008 15:31:32 GMT
Content-Type: text/html
Content-Length: 1395
Expires: Mon, 03 Nov 2008 15:31:32 GMT
X-Squid-Error: ERR_ACCESS_DENIED 0
X-Cache: MISS from localhost.localdomain
Via: 1.0 localhost.localdomain (squid/3.0.STABLE10)
Proxy-Connection: close

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html><head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<title>ERROR: The requested URL could not be retrieved</title>
<style type="text/css"><!--BODY{background-color:#ffffff;font-family:verdana,sans-serif}PRE{f
ont-family:sans-serif}--></style>
</head><body>
<h1>ERROR</h1>
<h2>The requested URL could not be retrieved</h2>
<hr>
<p>The following error was encountered while trying to retrieve the URL: <a href="http://www.
squid-cache.org">http://www.squid-cache.org</a></p>
```

图 7-5 Squidclient 访问站点 www.squid-cache.org

等完全确认 Squid 工作正常后，可以使用 Ctrl+C 组合键来中断 Squid 进程。在以后应用过程中以后台运行的方式启动 Squid 服务。

7.4 启动与停止代理服务器

代理服务也遵循一般的服务启动、停止规范，要启动代理服务，可以通过执行以下命令实现。

```
[root@localhost ~]# service squid start
```

执行命令后的结果如下所示，表明代理服务器启动成功。

```
[root@localhost ~]# service squid start
启动 squid: .                                              [确定]
```

注意 还可以通过执行以下命令实现启动代理服务的功能。

```
[root@localhost ~]# /etc/init.d/squid start
```

而重启 Squid 代理服务的操作类似于启动操作，要重启 Squid 代理服务则可以通过执行以下命令完成。

```
[root@localhost ~]# service squid restart
```

执行命令后的结果如下所示。

```
[root@localhost ~]# service squid restart
停止 squid:                                                [确定]
启动 squid: .                                              [确定]
```

注意 还可以通过以下方法来实现代理服务器的重启功能。

```
[root@localhost ~]# /etc/init.d/squid restart
```

而停止代理服务器的操作也与启动操作类似，可以通过执行以下命令完成。

```
[root@localhost ~]# service squid stop
```

执行命令后的结果如下所示，表明停止代理服务成功。

```
[root@localhost ~]# service squid stop
停止 squid:                                                [确定]
```

停止代理服务器的方法有多种，具体如下。

方法 1 通过执行以下命令实现。

```
[root@localhost ~]# /etc/init.d/squid stop
```

方法 2 使用 squid -k shutdown 命令。

```
[root@localhost ~]# /usr/local/squid/sbin/squid -k shutdown
```

方法 3 使用 kill 指令强行停止代理服务器。

首先使用指令 ps 找到 Squid 的进程 ID，然后手工 kill Squid 进程，执行以下命令。

```
[root@localhost ~]# ps ax |grep squid
[root@localhost ~]# kill -TERM 进程 ID
```

方法 4 使用 squid -k interrupt 命令。

```
[root@localhost ~]# /usr/local/squid/sbin/squid -k interrupt
```

7.4.1 设置 Squid 服务随系统的启动而自动启动

由于 Squid 服务不会随系统的启动而自动启动，启动了的 Squid 服务只能保持到本次计算机关机之前，下次重启计算机时，还需要重新启动 Squid 服务，为了避免反复操作，可以设置 Squid 服务随系统的启动而自动启动。

注意 对于不同的操作系统，其启动脚本如何工作也不相同。

下面将描述一些通用的设置方法。

1. 通过修改“/etc/rc. local”脚本实现

此种方法为最容易的机制之一，“/etc/rc.local”脚本在每次系统启动时以 root 运行，在该脚本中增加如下一行命令。

```
/usr/local/squid/sbin/squid -s
```

同时根据安装位置可能不同，可能会使用其他命令行选项（但在此处建议不要使用-N 选项）。

2. 图形化方式

使用图形化方式实现此功能的具体步骤如下。

第 1 步，在系统桌面依次单击“系统”→“管理”→“服务器设置”→“服务”菜单选项，如图 7-6 所示。

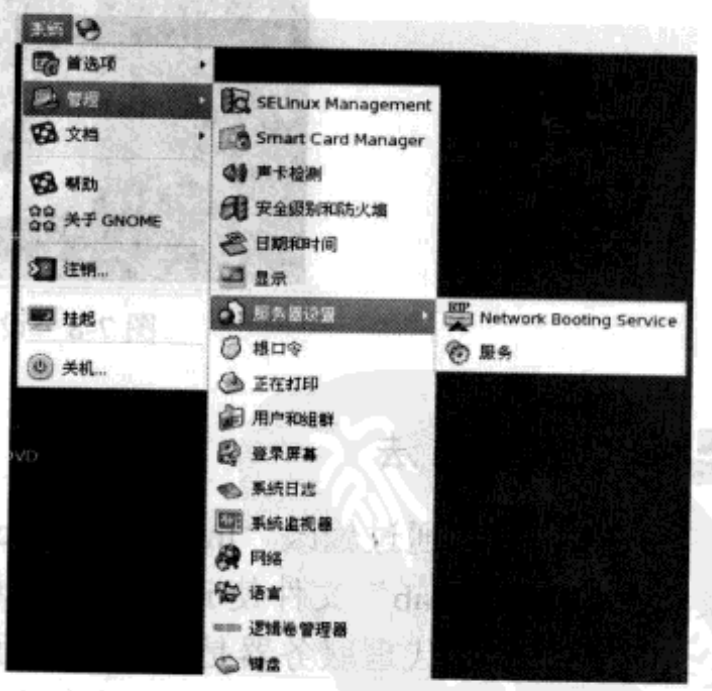


图 7-6 显示“服务配置”窗口

第 2 步，此时会显示“服务配置”窗口，在该窗口中，选中“squid”复选框，然后保存，如图 7-7 所示。

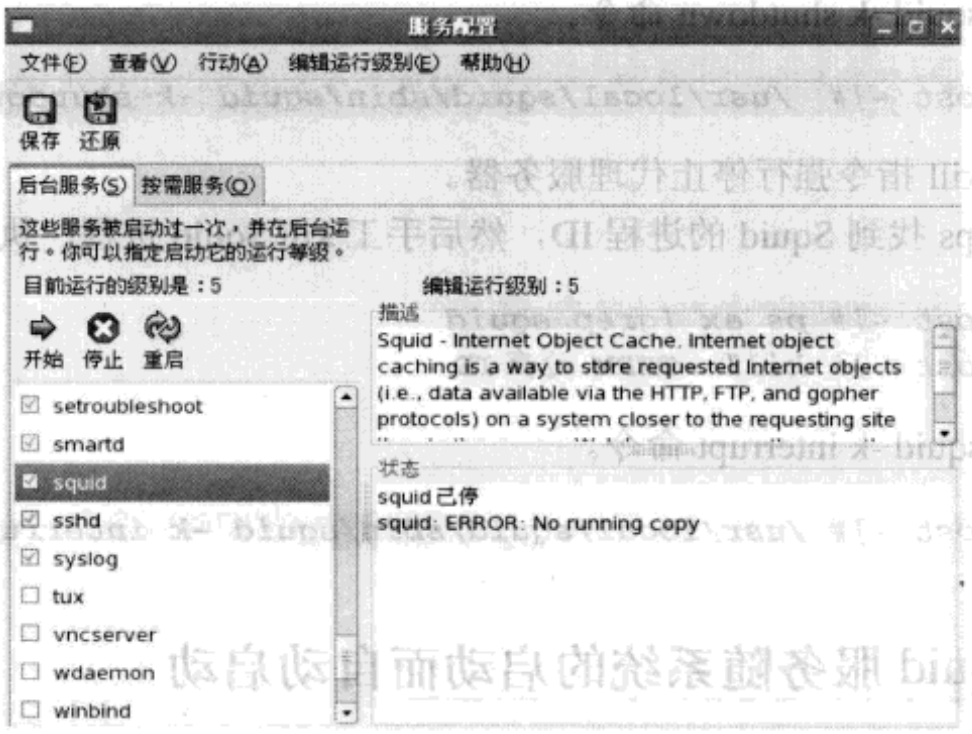


图 7-7　设置 Squid 自动启动

3. 在控制终端启动服务器配置程序

在控制终端输入“ntsysv”命令启动服务器配置程序，找到“squid”服务，在其前面加上星号（“*”），然后单击“确定”按钮，如图 7-8 所示。

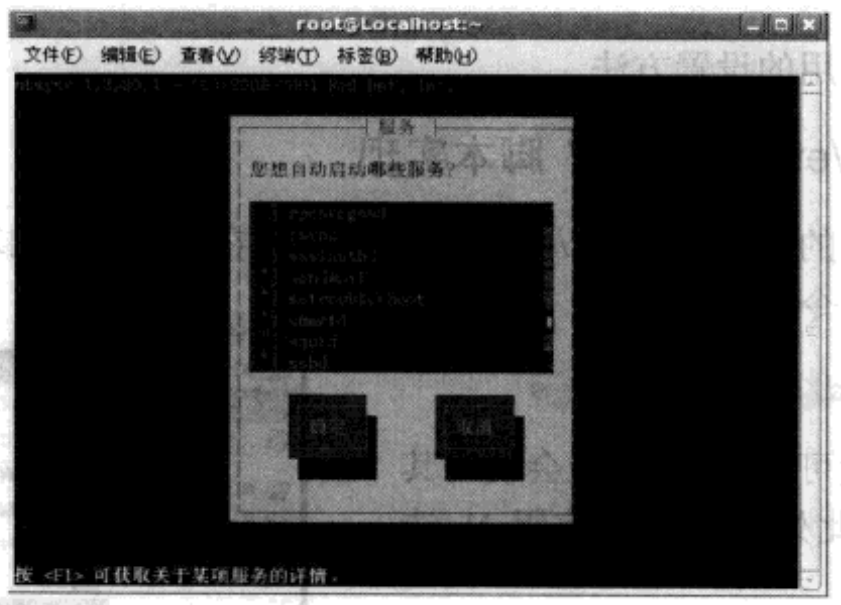


图 7-8　设置代理服务为自动启动

4. 其他方法

另外，可以通过修改“init.d”和“rc.d”脚本文件使用独立的 shell 脚本来启动服务，以及修改“/etc/inittab”文件使 init 进程启动基于运行等级的服务等来实现设置。

实例　测试代理服务器是否处于运行状态

检测代理服务器是否处于正常运行状态的方法有两种：一种是以命令方式（squid -k check）；另一种是进入 Squid 所在目录查看访问日志文件 Access.log 中是否保存有浏览过的网页的方式。

方法 1 命令方式（squid -k check）

使用以下命令来检查 Squid 是否处于正常运行状态。

```
[root@localhost ~]# /usr/local/squid/sbin/squid -k check
```

此方法是 Squid 默认的检查代理服务器运行的方法。

方法 2 查看访问日志文件 Access.log

执行以下命令检查日志文件 Access.log。

```
[root@localhost ~]# gedit /usr/local/squid/var/logs/access.log
```

7.4.2 重新载入配置文件

在对 Squid 服务的配置文件进行修改后，要想使新的配置生效，可以通过重启服务的方式实现。但是如果要重启在大型应用的网络中正处于服务状态的 Squid 服务器是不现实的。面临此类问题时，可以采用重新载入配置文件的方法，来实现使新的配置生效。

重新载入配置文件的的方法有使用 reload 命令和使用 squid -k reconfigure 命令两种，具体做法如下。

- 使用 reload 命令

要重新载入配置文件，可以使用以下命令。

```
[root@localh2ost ~]# /etc/rc.d/init.d/squid reload
```

- 使用 squid -k reconfigure 命令

要重新载入配置文件，可以使用以下命令。

```
[root@localhost ~]# /usr/local/squid/sbin/squid -k reconfigure
```

执行该命令时，HUP 信号被发送到运行中的 Squid 服务进程。然后 Squid 服务进程读取并解析 squid.conf 文件。

7.5 Squid Server 高级配置

前一节已经向大家讲解了 Squid Server 的一些基本配置，在本节将继续学习 Squid Server 的高级配置，Squid Server 的高级配置包括透明代理配置和 Squid 安全配置，接下来将为大家介绍如何进行透明代理配置和 Squid 的安全配置。

7.5.1 实现透明代理

透明代理是把 NAT 技术和代理技术二者有机结合起来的一种应用，透明代理可以让客户端感觉不到代理的存在，故称之为“透明”。要实现透明代理的应用，只需将 Linux 服务器的 IP 地址设置为默认网关即可。

要使用透明代理，终端客户必须在 TCP/IP 中正确设置 DNS 服务器地址。当终端客户访问 Internet 资源，请求 Internet 资源的数据包途经 Linux 服务器转发时，Linux 服务器上的防火

墙 iptables 服务会应用 NAT 技术将终端客户的 HTTP 请求重定向到 Squid 代理服务器，由 Squid 代理服务器代替终端客户访问外部的 Internet 信息资源，再将获取到的信息资源回传给终端客户。

在 Linux 平台下用户使用 iptables+squid 来实现透明代理和网络地址转换，设置透明代理的具体步骤如下。

第 1 步，配置 Squid。

要实现透明代理首先要配置 squid.conf 文件，下面将具体讲解如何配置此文件。编辑“/etc/squid/squid.conf”文件，修改为如下内容。

```
#squid 监听 http 客户端端口
httpd_port 192.168.1.4:8080
#缓存设置
cache_mem 128MB
cache_swap_low 90
cache_swap_high 95
cache_dir ufs /cache/squid 1000 16 256
#缓存日志
cache_access_log /cache/squid/access.log
cache_log /cache/squid/cache.log
cache_store_log /cache/squid/store.log
#透明代理
httpd_accel_host virtual
httpd_accel_port 80
httpd_accel_with_proxy on
httpd_accel_uses_host_header on
#子网掩码
client_netmask 255.255.255.255
#代理权限
http_access allow all
```

在表 7-3 中简要描述了实现透明代理的几个相关参数。

表 7-3　实现透明代理的相关参数

参 数 设 置	相 关 描 述
设置 httpd_accel_host 为 virtual	指定当前采用虚拟主机模式
设置 httpd_accel_port 为 80	指定需要加速的请求端口
设置 httpd_accel_with_proxy 为 on	表示 Squid 服务器既是 Web 请求的加速器，又是缓存代理服务器
设置 httpd_accel_uses_host_header 为 on	在透明代理模式下，代理服务器的缓存功能才能正确工作，这时 Squid 会把存储的对象加上主机名作为索引

注意　参数 httpd_accel_host 和 httpd_accel_port 本来是用来定义 Squid 加速模式的。采用以上设置后，Squid 默认情况是取消了缓存和 ICP 功能的。如果需要这些功能，则必须设置 httpd_accel_with_proxy 参数。

第 2 步，编写防火墙规则。

通过使用 iptables 实现 NAT，本例使用 iptables 实现窗口重定向，这里 Squid 使用的端口号是 8080，执行以下命令。

```
[root@localhost ~]# iptables -t nat -A PREROUTING -i eth0 -p tcp -m tcp --dport 80 -j REDIRECT --to-ports 8080
[root@localhost ~]# iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

执行以上命令表示将所有进入 eth0 网络接口 80 端口的 Web 服务的请求直接转发到 8080 端口，交由 Squid 代理处理。

7.5.2 Squid 安全设置

身份认证对于实现网络通信安全至关重要，是保护网络系统资源的第一道防线。在安全的网络通信中，通信各方需要搭配满足需求的用户身份认证机制，保证合法终端客户以合法的权限访问网络系统资源，Squid 服务同样如此。默认情况下，Squid 服务本身不包括任何身份认证程序，但可以通过绑定外部用户认证程序的方式实现 Squid 服务本身的用户身份认证，下面是一些 Squid 服务能够绑定的用户认证程序。

- LDAP 认证

使用 Lightweight 目录访问协议（Lightweight Directory Access Protocol）。

- NCSA 认证

使用 NCSA 风格的用户名和密码。

- SMB 认证

使用 SMB 协议的服务，如 SAMBA 或 Windows NT。

- MSNT 认证

使用 Windows NT 的域验证。

- PAM 认证

使用 Linux 的可装载验证模块。

- Getpwam 认证

使用 Linux 密码档。

另外，Squid 不支持在透明代理模式下启用用户身份认证功能。

Squid 代理服务绑定 NCSA 用户认证的实现过程可以大概描述为：首先配置 squid.conf 文件，然后再利用 htpasswd 程序建立账号文件，最后进行用户认证的测试。

其具体实现过程如下所述。

1. 配置 squid. conf 文件

Squid 代理服务可以绑定的用户认证程序有多种，但最为常见的是 NCSA。

在 Squid 主配置文件“/usr/local/squid/etc/squid.conf”中实现 NCSA 认证方式绑定配置的重要部分如下。

```
auth_param basic program /usr/bin/ncsa_auth /usr/local/squid/etc/password
#该选项指出了认证方式（basic）、需要的程序（ncsa_auth）和对应的密码文件（password）。
auth_param basic children 5
#指定认证程序的进程数。
auth_param basic realm My Proxy Caching Domain
#浏览器显示输入用户/密码对话框时的领域内容。
auth_param basic credentialsttl 2 hours
```

```
#基本的认证有效时间。
acl normal proxy_auth REQUIRED
http_access allow normal
#普通用户需要通过认证才能访问 Internet。
```

其中，包括了认证方式、认证程序的进程数、认证有效时间、认证领域内容和允许经过认证的用户访问的设置，具体说明如表 7-4 所示。

表 7-4　具体设置项

具体设置项	简 要 描 述
设置认证方式	参数 auth_param basic program 定义了认证方式为 basic（从 Squid2.5 版以后，NCSA 认证包含在 basic 中）、认证程序路径和认证程序需要读取的账号文件。 auth_param basic program /usr/bin/ncsa_auth /usr/local/squid/etc/password 该设置说明绑定 NCSA 认证，认证程序在“/usr/bin/ncsa_auth”路径下，用户账号文件在“/usr/local/squid/etc/password”路径下
设置认证程序的进程数	参数 auth_param basic children 定义了认证程序的进程数。 auth_param basic children 5 该设置说明 NCSA 认证程序许可的进程数是 5
设置认证有效时间	参数 auth_param basic credentialsttl 定义了用户经过认证后的有效时间，用户在有效时间后如果还需要继续使用 Squid 代理服务，则必须重新输入用户名和密码。 auth_param basic credentialsttl 2 hours 该设置说明 NCSA 认证的有效时间是两个小时
设置认证领域内容	参数 auth_param basic realm 定义了 Web 浏览器显示用户/密码对话框时的领域内容。 auth_param basic realm My Proxy Caching Domain 该设置说明当用户首次使用 Web 浏览器访问外网资源，系统提示输入用户名/密码时，对话框中的标题内容
设置允许经过认证的用户访问	定义了一条名为 auth_user 的 ACL，ACL 类型为 proxy_auth，通过外部程序进行用户认证方式，列表值是 REQUIRED，然后使用 http_access 选项允许该访问控制列表。 acl normal proxy_auth REQUIRED http_access allow normal

2. 利用 htpasswd 程序建立账号文件

可以利用 Apache 的 htpasswd 程序生成供用户认证使用的账号文件，执行如下命令。

```
[root@localhost ~]# htpasswd -c /usr/local/squid/etc/passwd test
```

该账号文件位于“/usr/local/squid/etc/passwd”路径下，并且文件每行包含一个用户账号信息（用户名和经过加密后的密码）。

3. 测试用户认证

在 Squid 的安全配置工作完成后，需要测试一下用户认证的配置是否合适，具体实现步骤如下。

第 1 步，重新设置后，需要重启 Squid 服务。

第 2 步，在终端客户的 Web 浏览器中配置好代理服务器的 IP 地址和端口号。

第 3 步，在配置好代理服务器的 IP 地址和端口号后，访问任意外部站点，此时 Web 浏览器会弹出提示用户输入用户名和口令的登录对话框。

第 4 步，在成功输入用户名和口令后，即可连上 Internet；如果用户名和口令不正确，则显示“网络缓存服务器拒绝访问”的错误提示信息。

7.5.3 Squid 外部程序支持选项

Squid 支持 3 种方式用于从用户端采集验证信用项：基本、摘要（Digest）和 NTLM。这些方式指定 Squid 如何从客户端接收用户名和密码。

从安全角度考虑，摘要和 NTML 验证相对安全，基本验证相对不安全。对每种方式，Squid 提供一些验证模块，或辅助进程，用于实际处理认证的过程。

7.6 Squid 日志管理

Squid 具备强大的日志管理功能，该功能对于网络管理员尤为重要。管理员通过查看系统记载的流量日志，可以清楚地看到用户对网络服务器的使用情况，从中发现网络中可能存在的安全问题，加强了对网络服务器的维护与管理。通过查看相关日志文件，用户可以实时、准确地了解代理服务器的访问记录、cache 的存储状况、cache 的访问记录等许多有用的信息，从而高效地对代理服务器进行管理。

7.6.1 日志选项

主配置文件 squid.conf 中有 3 个与日志有关的选项，分别为 cache-access-log、cache-log 以及 cache-store-log，具体说明如下。

- cache-access-log 选项

此选项定义了访问记录日志文件的路径，该日志文件记录了用户访问外部网络的详细信息，通过访问日志文件可以查看每台客户机的上网记录。

- cache-log 选项

此选项定义了记录缓存的相关信息的日志文件的路径。

- cache-store-log 选项

此选项定义了记录网页在缓存中调用情况的日志文件的路径。

7.6.2 Squid 代理服务器日志系统的构成

Squid 代理服务器的日志系统位于/var/log/squid 目录中，有 access.log、cache.log、store.log 3 个主要的日志文件，其位置和名称由主配置文件中的相关选项设定。

- access.log（访问记录日志文件）

包含了对 Squid 发起的每个终端客户请求，每个请求以一行记录，每行约 150 B。用 cache_access_log 指令来改变该日志文件的路径。

```
cache_access_log  /squid/logs/access.log
```

如果不希望 Squid 记录终端客户请求日志，可以设定日志文件的路径为/dev/null。

- cache.log（缓存记录日志文件）

包含了状态性的和调试性的消息。刚开始运行 Squid 时，应密切关注该文件。假如 Squid

拒绝运行，理由也许会出现在 cache.log 文件的结尾处。在正常条件下，该文件不会变得很大。也请注意，假如以-s 选项来运行 Squid，重要的 cache.log 消息也可被送到 syslog 进程。可以通过使用 cache_log 参数，改变该日志文件的路径。

```
cache_log /squi d/logs/cache.log
```

- store.log（网页缓存日志文件）

包含了进入和离开缓存的每个目标的记录。平均记录大小典型的是 175~200 B。然而，Squid 不在 store.log 里对 cache 单独创建接口，所以比 access.log 的记录少得多。请使用 cache_store_log 参数来改变位置。

```
cache_store_log /squid/logs/store.log
```

另外，如果要完全禁止 store.log 日志，则可以通过指定路径为 none 来实现。

```
cache_store_log none
```

注意 访问日志是 Squid 最重要的日志文件，其访问路径为“/var/log/squid/access.log”。该日志文件记录了客户使用代理服务器的许多有用信息，包括使用时间、请求的内容、客户端信息等。该文件的格式如下所示。

```
time    eclapsed    remotehost    code/status    bytes    method    URL    rfc931
peerstatus/peerhost type
```

如上可以看到，该日志文件的记录共包含 10 个字段，每个字段的含义如表 7-5 所示。

表 7-5 access.log 日志文件字段

字　段	说　明
time	记录客户访问代理服务器的时间，从 1970 年 1 月 1 日到访问时所经历的秒数，精确到 ms
eclapsed	记录处理缓存所花费的时间，以 ms 计数
remotehost	记录访问客户端的 IP 地址或者是域名
code/status	结果信息编码/状态信息编码
bytes	缓存字节数
method	HTTP 请求方法，GET 或者 POST
URL	访问的目的地址的 URL，如 www.sina.com.cn
rfc931	默认的，暂未使用
peerstatus/peerhost	缓存级别/目的 IP
type	缓存对象类型，如 text/html

7.6.3 日志及流量分析软件——Webalizer

Webalizer 是 Linux 下的日志及流量分析软件之一，是一个高效、免费的 Web 服务器日志分析程序。有兴趣的用户可以到其官方站点（http://www.webalizer.org/）进行更进一步的了解，如图 7-9 所示。

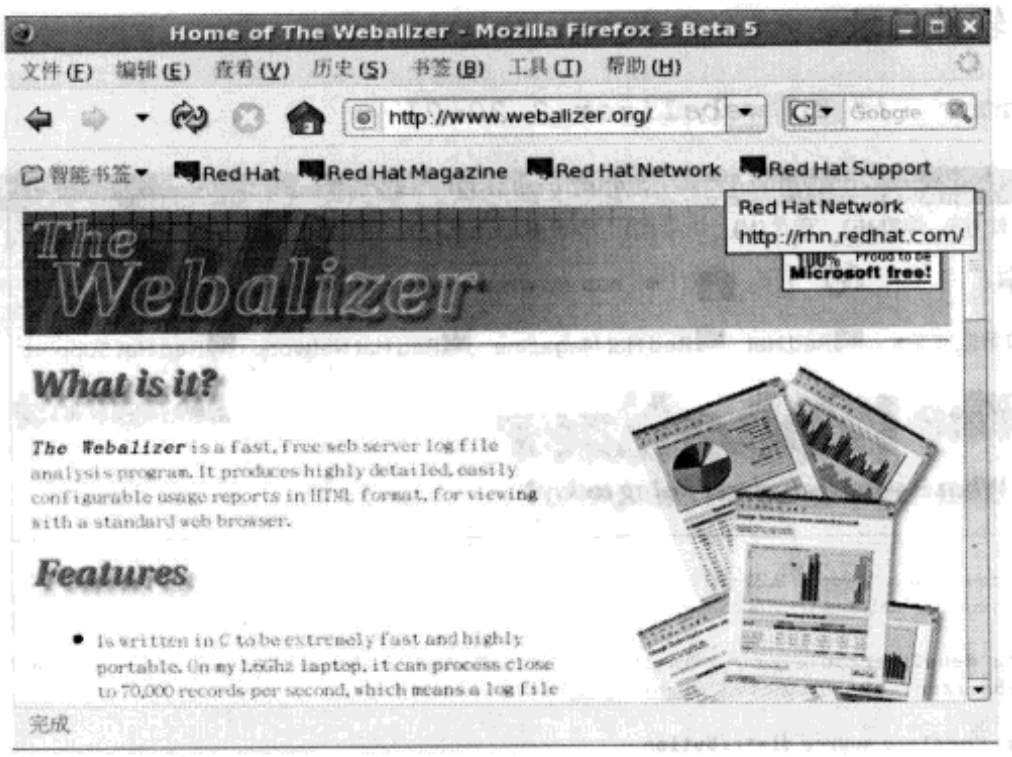


图 7-9　Webalizer 官方站点

Internet 上的很多站点都使用 Webalizer 进行 Web 服务器日志分析，Webalizer 分析结果以 HTML 文件格式保存，便于通过 Web 服务器进行浏览。Webalizer 具有以下一些特性。

- 用 C 语言编写的程序，该系统具有很高的运行效率。
- 支持命令行配置以及配置文件。
- 支持多种语言，也可以自己进行本地化工作。
- 支持多种平台，比如 UNIX、Linux、NT、OS/2 和 MacOS。
- Webalizer 不仅可以支持标准的一般日志文件格式（Common Logfile Format），还支持几种 NCSA 的组合日志格式（Combined Logfile Format）的变种，从而可以统计客户情况以及客户操作系统类型。并且现在 Webalizer 已经可以支持 wu-ftpd/proftpd xferlog 日志格式以及 Squid 代理服务器日志文件格式了。

1. 安装 Webalizer

在基本了解了 Webalizer 后，接下来将学习 Webalizer 的安装。安装 Webalizer 的方法主要有 RPM 包安装和源代码安装两种，下面将具体讲解如何使用这两种方法安装 Webalizer。

（1） 源代码安装方式

使用源代码的方式安装软件比使用 RPM 包方式安装软件稍微复杂一点，使用该方法安装 Webalizer 的具体步骤如下。

第 1 步，访问 http://www.webalizer.org/download.html，下载 Webalizer 安装文件。进入该页面后，单击“The Complete source distribution”栏目中的“Tar/Gzip archive”超链接，然后下载“webalizer-2.20-01-src.tgz”文件（本书使用的是 2.20-01 版本），如图 7-10 所示。

第 2 步，下载完毕后，进入 webalizer2.20-01 的下载目录，在此目录中存在一个“webalizer-2.20-01-src.tgz”文件，然后执行以下指令解压该安装包文件。

```
[root@localhost ~]# tar zxvf webalizer-2.20-01-src.tgz
```

第 3 步，成功解压缩后，看到在当前目录下有一个 webalizer-2.20-01 文件目录，执行以下

指令进入解压好的软件目录。

```
[root@localhost ~]# cd webalizer-2.20-01
```

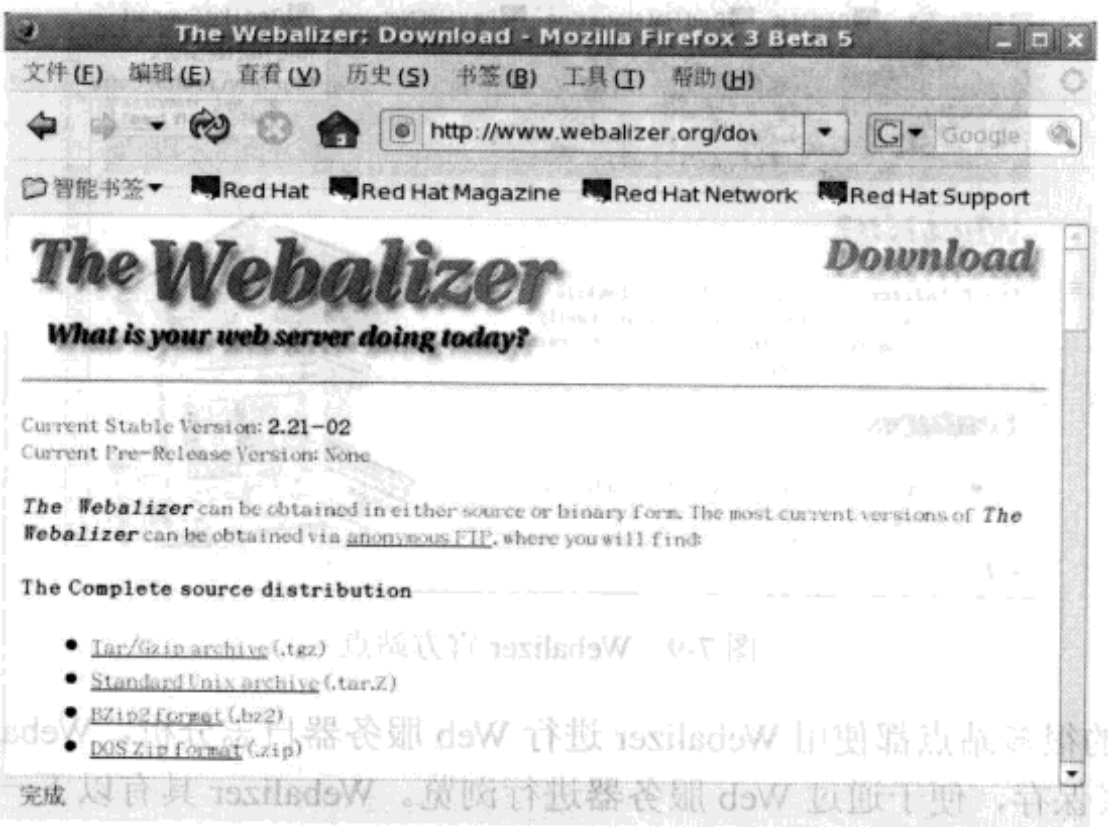


图 7-10　下载 Webalizer

第 4 步，进入软件目录后，执行以下命令进行配置测试。

```
[root@localhost webalizer-2.20-01]#./configure
```

第 5 步，测试成功后，使用 GNU make 工具编译源代码。

```
[root@localhost webalizer-2.20-01]# make --with-language=chinese
```

第 6 步，编译成功后，会产生一个 Webalizer 可执行文件，将该文件复制到/usr/sbin/目录下，执行以下命令。

```
[root@localhost webalizer-2.20-01]# cp Webalizer /usr/sbin/
```

复制成功后，Webalizer 的安装工作就已结束。

（2） RPM 包安装方式

用 RPM 包方式安装 Webalizer 的具体步骤如下。

第 1 步，在不确定系统是否已经安装了 Webalizer 的情况下，可以使用以下命令检查。

```
[root@localhost ~]#rpm -qa|grep webalizer
```

执行命令后的结果如下所示，表明安装有 Webalizer。

```
[root@localhost ~]# rpm -qa|grep webalizer
webalizer-2.01_10-30.1
```

第 2 步，如果发现当前系统还未安装 Webalizer，则需首先在 Red Hat Enterprise Linux 5 的安装光盘中找到并安装 webalizer-2.01_10-11.i386.rpm RPM 包文件。然后将 Red Hat Enterprise Linux 5 的安装盘放入光驱，加载光驱后在光盘的 Server 目录下找到相应的 RPM 安装包文件，

最后使用 rpm -ivh 命令安装。

```
[root@localhost ~]# rpm -ivh webalizer-2.01_10-11.i386.rpm
```

以上操作即是用 RPM 包方式安装 Webalizer 的具体方法。不管以何种方式进行安装，安装完成后，下面将进入配置工作阶段。

2. 配置 Webalizer

配置 Webalizer 的方式主要有通过命令行配置 Webalizer 和通过配置文件进行配置两种方式。使用命令行配置的方式不是很直观，因此通常采取的都是通过配置文件进行配置的方式，通过此方式进行配置也较为常见。接下来将通过一个实例来讲解如何通过配置文件的方式来配置 Webalizer。

注意 Webalizer 的配置文件的路径是“/etc/webalizer.conf”，其有一个对应的模板文件/etc/webalizer.conf.sample。一般情况下，该配置文件的默认参数配置都能满足一定的应用需要，可以直接使用。

第 1 步，使用 vi 编辑 Webalizer 配置文件，执行以下命令。

```
[root@localhost ~]# vi /etc/webalizer.conf
```

第 2 步，设置访问日志的存放路径。在该配置文件中找到如下语句，将系统默认值

```
LogFile /var/log/httpd/access_log
```

修改为如下内容。

```
LogFile /var/log/squid/access.log
```

第 3 步，设置访问日志的格式类型。在该配置文件中找到如下语句，将格式类型参数

```
LogType clf
```

修改为如下内容。

```
LogType squid
```

第 4 步，设置报表输出目录。在该配置文件中找到如下语句，将默认值

```
OutputDir /var/www/html/usages
```

修改为如下内容。

```
Squid 的统计报表，保存目录。
```

第 5 步，使用 crontab 命令让 Webalizer 每天生成 Squid 的当日流量统计分析。

```
[root@localhost ~]# crontab -e
```

第 6 步，在该配置文件中，添加如下配置行，使得 Webalizer 每晚 10 点生成当天的 Squid

流量统计分析结果。

```
22 0 * * * /usr/bin/webalizer -c /etc/webalizer.conf
```

7.6.4 Webalizer 软件的应用

通过 Webalizer 软件，可以将 Squid 日志信息综合起来进行统计并得出可视化的显示结果。网络管理者通过该可视化的结果，可以得到 Squid 服务器的使用情况。

以下是 Webalizer 软件与 Squid 服务器相结合使用的实例。

1. 启动 Web 服务器以及 Webalizer

通过执行以下命令来启动 Apache 服务。

```
[root@localhost ~]# service httpd start
```

一般情况下，配置好的 Webalizer 软件只需通过执行以下命令来启动。

```
[root@localhost ~]# /usr/sbin/webalizer -c /etc/webalizer.conf
```

注意 日志统计的详细情况需通过 Web 页面呈现，因此需要启动 Apache 服务。

启动工作完成以后，即可通过浏览器访问 Apache 服务器的地址“http://服务器 IP/usage”来查看 Squid 代理服务器的流量分析结果。

2. 认证和授权配置

在使用 Webalizer 软件时，为了系统的安全，通常并不需要每个用户都有权限来查看代理服务器的流量情况。有些非法用户为了掩盖自己非法使用代理服务器事实的目的，对流量情况统计的结果进行修改、删除等操作。因此，需要使用访问控制策略来对上述分析图表的查看权限进行控制。

要实现查看权限的控制，可以在 Apache 的主配置文件/etc/htpd/conf/httpd.conf 中进行认证和授权的配置，具体配置工作如下。

```
< Directory "/var/www/html/usage">
AuthType Basic                          ; #使用基本认证方式
AuthName "admin"                        ; #指定认证域名称
AuthUserFile /var/www/passwd/admin      ; #指定认证口令文件存放的位置
require valid-user                      ; #授权给认证口令文件中的所有用户
< /Directory>
```

认证和授权配置成功后，需要创建认证口令文件以及添加用户，执行以下命令。

```
[root@localhost ~]# mkdir /var/www/passwd
[root@localhost ~]# cd /var/www/passwd
[root@localhost passwd]# htpasswd -C admin jackiee
New password:
```

```
Re-type new password:
Adding passwor for user jackiee
```

然后执行以下命令将口令文件的属主改为 Apache，并重新启动 Apache。

```
[root@localhost ~]# chown apache.apache admin
[root@localhost ~]# service httpd restart
```

通过以上配置后，访问“http://服务器 IP/usage”时，只有具有认证和授权的用户才能查看 Squid 代理服务器的流量分析报表信息。

7.7 不同环境下的代理客户端的配置

代理客户端可以是 Linux 操作系统也可以是 Windows 操作系统，由于操作系统的不同，其配置也因此不同。在本节将分别讲解在不同操作系统中的代理客户端的具体配置。

7.7.1 Linux 系统下 Firefox 代理设置

Firefox 是 Linux 环境下常使用的 Web 浏览器，在 Red Hat Enterprise Linux 5 操作系统环境中进行代理客户端配置工作的具体步骤如下。

第 1 步，打开 Firefox 浏览器，在浏览器窗口中，选择“编辑”菜单的“首选项”子菜单，此时会显示如图 7-11 所示的“Firfox 首选项”窗口。

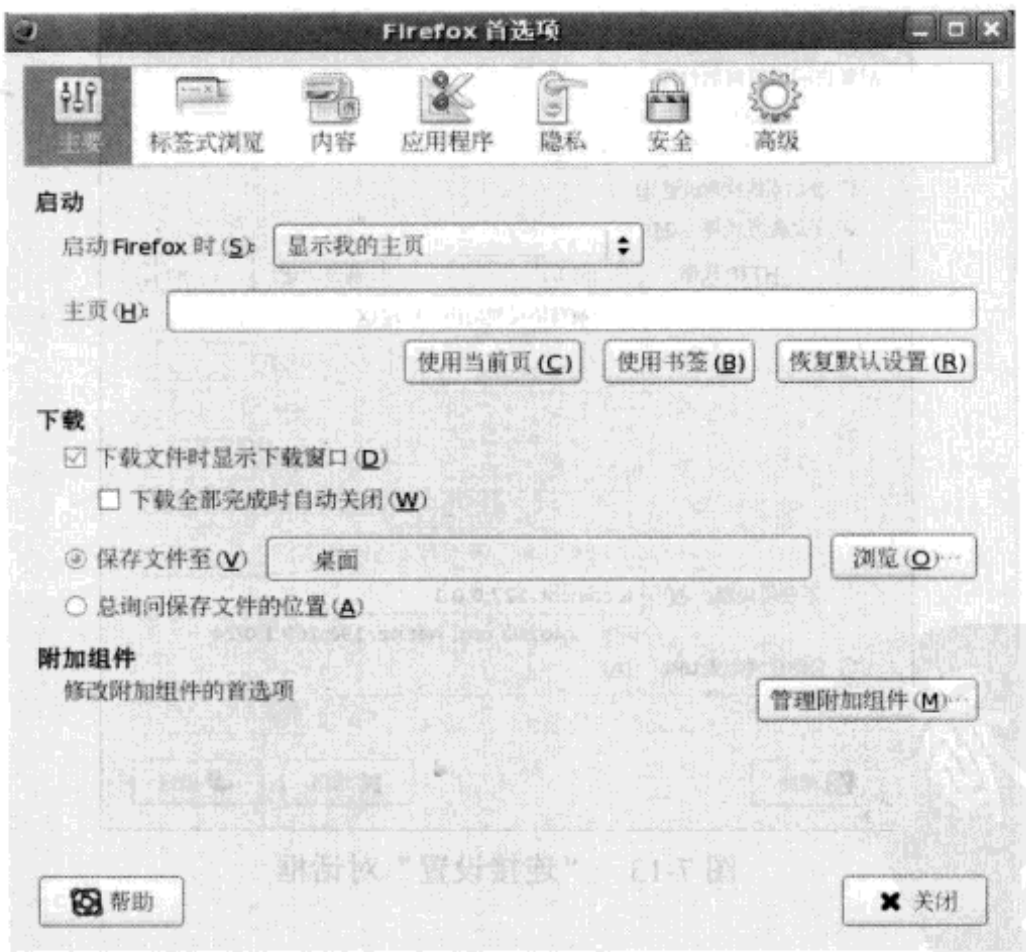


图 7-11 “Firefox 首选项”窗口

第 2 步，单击该窗口中的“高级”菜单，并切换到“网络”选项卡，如图 7-12 所示。

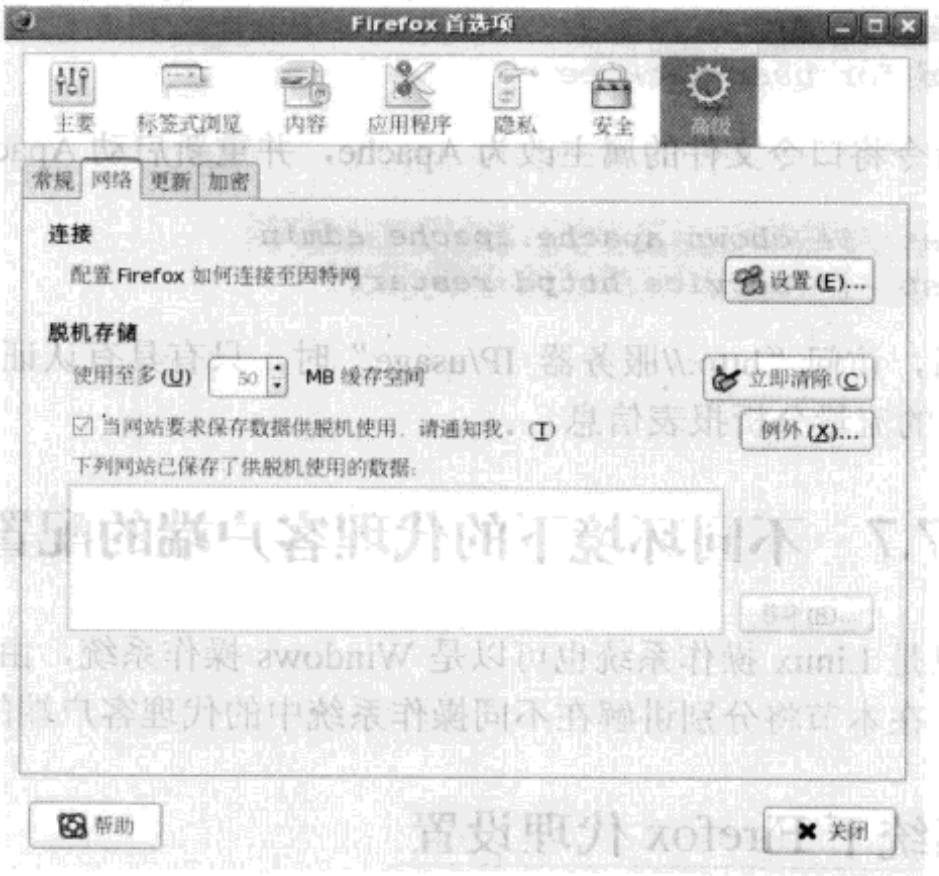


图 7-12　“网络”选项卡

第 3 步，单击“设置”按钮，显示“连接设置”对话框，在该对话框中首先选择“手动配置代理”单选按钮，然后在“HTTP 代理”文本框中输入正确的代理服务器的 IP 地址和端口号，最后单击“确定”按钮完成代理客户端的配置工作，如图 7-13 所示。

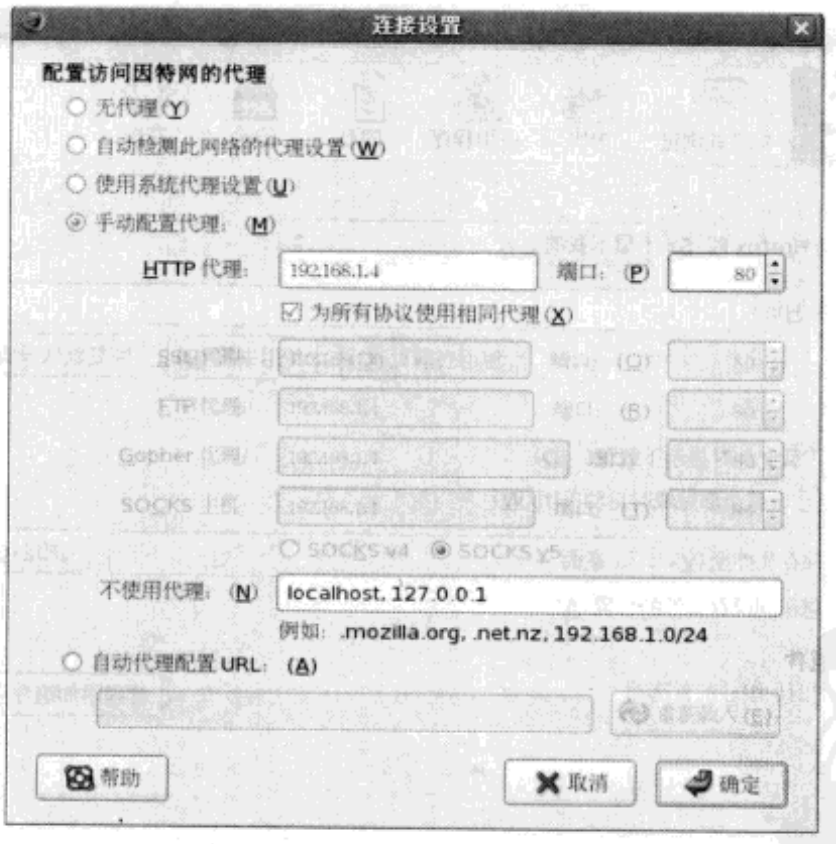


图 7-13　“连接设置”对话框

注意　如果还想通过代理服务器使用 SSL、FTP、Gopher 和 SOCKS 协议，则可以选中“为所有协议使用相同代理”复选框。

7.7.2 Windows 系统下 IE 代理配置

Windows 系统下默认的 Web 浏览器为 IE，在该操作系统中配置代理客户端的具体步骤如下。

第 1 步，在桌面 Internet Explorer 图标上单击鼠标右键，在右键菜单中选择“属性”选项，此时会显示如图 7-14 所示的“Internet 属性”对话框。

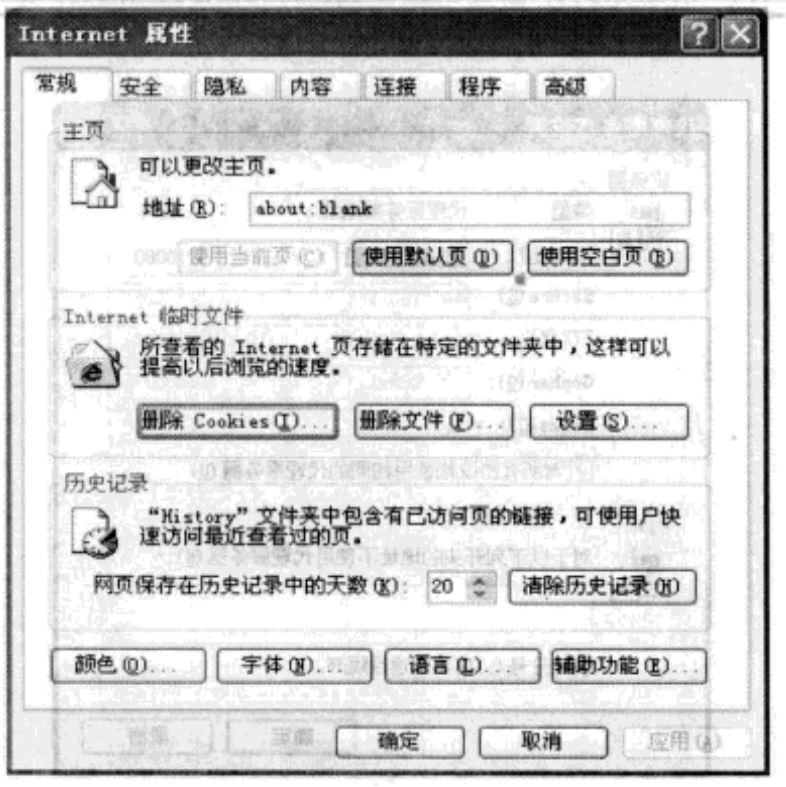


图 7-14 “属性”对话框

第 2 步，切换到“连接”选项卡，然后单击“局域网设置”按钮，显示“局域网（LAN）设置”对话框。在此对话框选中“为 LAN 使用代理服务器”复选框，并在“地址”文本框中输入正确的代理服务器的 IP 地址，在“端口”文本框中输入正确的代理服务器的端口号，单击“确定”按钮，如图 7-15 所示。

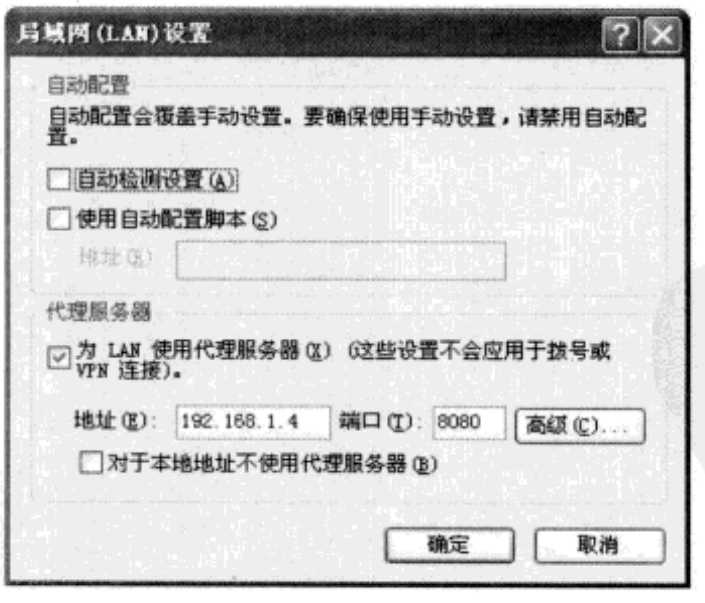


图 7-15 “局域网（LAN）设置”对话框

成功完成以上设置后，就可使内网的计算机通过代理服务器上网了。

注意　如果还想通过代理服务器使用 SSL、FTP、Gopher 和 SOCKS 等协议，可以单击“高级”按钮进行设置，具体设置工作如下。

首先单击“高级”按钮，即可显示“代理服务器设置”对话框，在该对话框中取消选择“对所有协议均使用相同的代理服务器”复选框，然后根据各个协议代理服务器的实际配置情况，正确填写相应的代理服务器地址和访问端口，如图 7-16 所示。最后单击“确定”按钮完成代理客户端的配置工作。

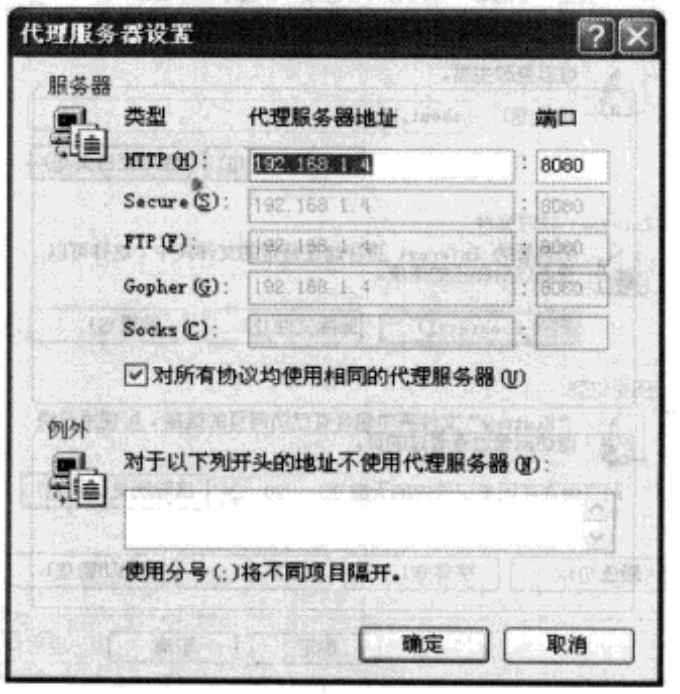


图 7-16　IE 的“代理服务器设置”对话框

第 8 章　Samba 服务器

在同一网络中，为了更加合理地利用网络资源，经常需要共享文件和打印机。比如在 Windows 网络环境下利用“网上邻居”来实现资源共享。而在网络中同时存在 Linux 主机和 Windows 主机时，则需要借助 Samba 服务来共享资源。该服务主要用于在 Windows 操作系统和 Linux 操作系统之间提供资源共享服务。

8.1　Samba 服务概述

Samba 是为了在 UNIX（Linux）上实现与 DOS（Windows）的文件共享而产生的，是由澳大利亚国立大学的 Andrew Tridgell 在 1991 年开发的一个开放源代码的项目。

8.1.1　Samba 服务简介

Samba 用于在 UNIX 上实现 SMB（Session Message Block，服务器信息块协议），是一个工具套件。在 Windows 系列中，SMB 协议通常用于实现文件和打印机共享。

注意　在 UNIX/Linux 计算机上，通过 NFS 就可以实现文件的共享。而 Windows 计算机的“网上邻居”实现文件系统的共享所使用的文件系统是 CIFS（Common Internet File System）。

SMB 协议运行在 NetBIOS 的顶端，虽然在 NetBEUI、IPX/SPX 和 TCP/IP 的顶端，都可以运行 NetBIOS，如图 8-1 所示。但对于 Samba 来说，目前只在 TCP/IP 上实现了 SMB。

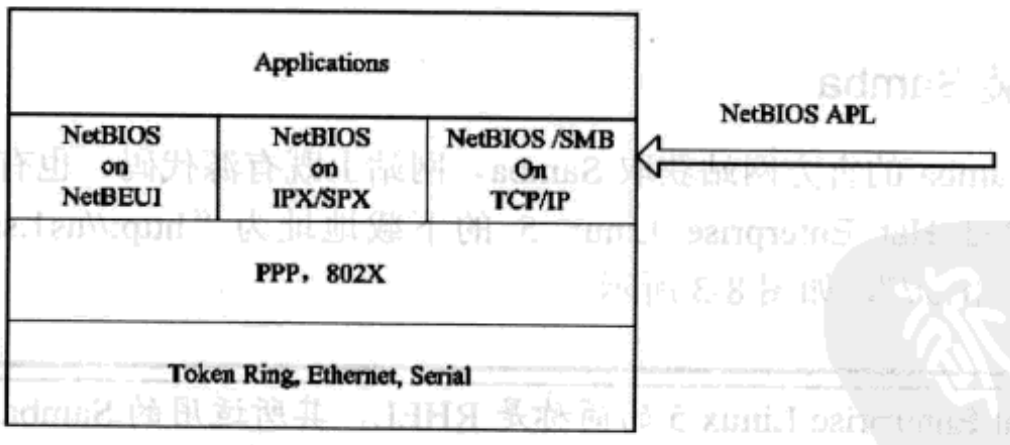


图 8-1　NetBIOS 上的 SMB 协议

8.1.2　Samba 服务的安装

Samba 服务器功能的实现主要靠两种服务：处理文件和打印机共享请求的 SMB 服务，和

处理 NetBIOS 名称服务请求和网络浏览功能的 NMB 服务。Samba 服务器的安装配置可以按如下步骤来实现。

1. 从光盘安装 Samba

Samba 服务在 Red Hat Enterprise Linux 5 中包含有内置安装程序，在安装系统时就可以进行安装。如果在安装系统的时候没有选择安装 Samba 服务，还可以通过单击“应用程序”→“添加/删除软件”菜单命令，显示“软件包管理者”窗口，从中找到 Samba 软件包（Windows 文件服务器）进行安装，如图 8-2 所示。

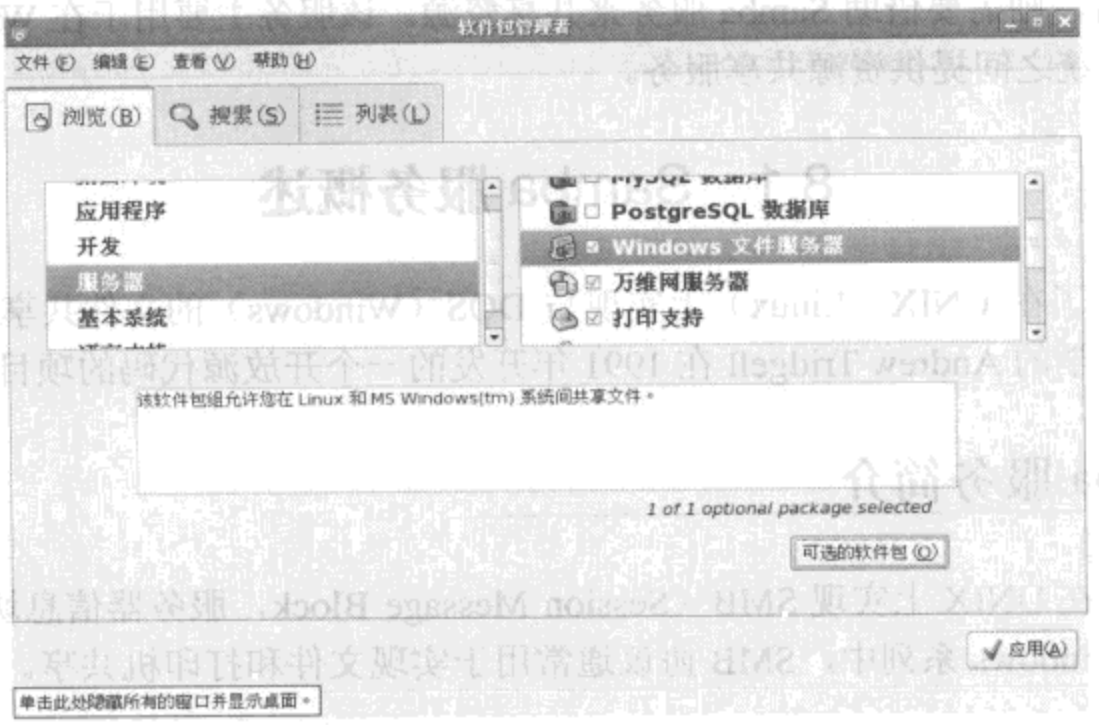


图 8-2 “软件包管理者”窗口

注意 Red Hat Enterprise Linux 5 安装之后，默认设置下，是不能通过“添加/删除软件”来直接从光盘上安装 Samba 组件的。要实现此功能，需要经过一定的设置。具体设置在本书第 3 章 KDE 桌面环境部分有详细介绍，这里就不再赘述。

2. 下载安装 Samba

第 1 步，从 Samba 的官方网站获取 Samba。网站上既有源代码，也有不同 Linux 发行套件版本供下载，Red Hat Enterprise Linux 5 的下载地址为“http://us1.samba.org/samba/ftp/Binary_Packages/RedHat/”，如图 8-3 所示。

注意 Red Hat Enterprise Linux 5 的简称是 RHEL，其所适用的 Samba 二进制安装包，一般就存在于含有字符 RHEL 的目录中。

第 2 步，在开始安装 Samba 之前，最好先查询系统中是否已经安装有 Samba 老版本，再决定是全新安装还是升级到最新的版本，执行如下指令。

```
[root@localhost ~]#rpm -qa |grep samba
samba-common-3.0.28-0.el5.8
```

```
samba-client-3.0.28-0.el5.8
samba-3.0.28-0.el5.8
system-config-samba-1.2.39-1.el5
[root@localhost ~]#
```

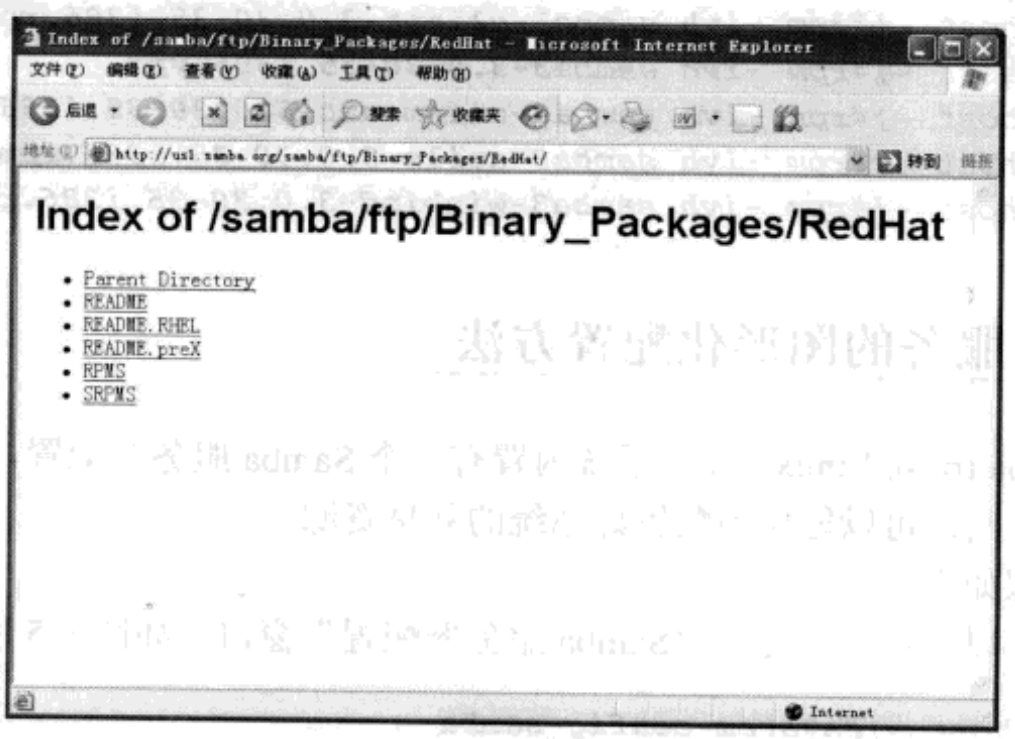


图 8-3　下载 Samba 的安装文件

第 3 步，卸载旧版 Samba，执行以下指令。

```
[root@localhost ~]#rpm -e samba-client-3.0.23c-2
[root@localhost ~]#rpm -e samba-comon-3.0.23c-2
```

注意　通过命令行方式卸载软件，不能够很好地解决文件的依赖性问题。最好是通过单击“应用程序”→“添加/删除软件”菜单命令，显示“软件包管理者”窗口，然后卸载已经安装的 Samba，如图 8-4 所示。取消选择“Samba client”，再单击“应用”按钮即可自动卸载 Samba 服务。

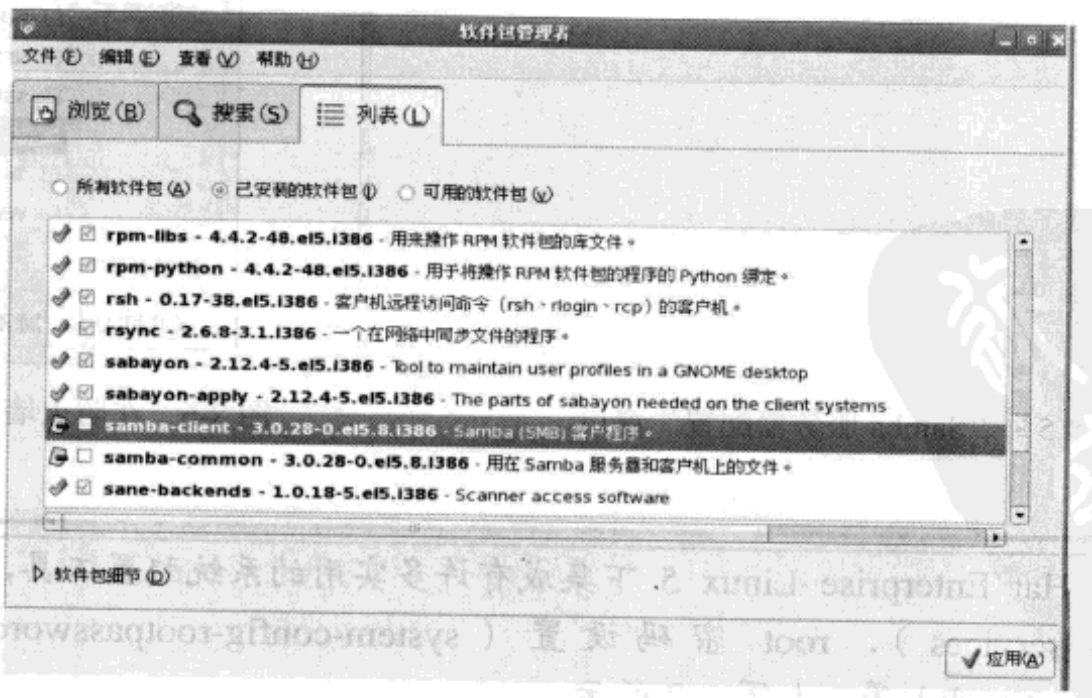


图 8-4　卸载 Samba

第 4 步，执行以下指令安装 Samba 服务。

```
[root@localhost ~]#rpm -ivh libsmbclient-3.0.30-35.i386.rpm
[root@localhost ~]#rpm -ivh libsmbclient-devel-3.0.30-35.i386.rpm
[root@localhost ~]#rpm -ivh samba3-client-3.0.30-35.i386.rpm
[root@localhost ~]#rpm -ivh samba3-3.0.30-35.i386.rpm
[root@localhost ~]#rpm -ivh samba3-cifsmount-3.0.30-35.i386.rpm
[root@localhost ~]#rpm -ivh samba3-utils-3.0.30-35.i386.rpm
[root@localhost ~]#rpm -ivh samba3-winbind-3.0.30-35.i386.rpm
```

8.1.3 Samba 服务的图形化配置方法

在 Red Hat Enterprise Linux 5 下，系统内置有一个 Samba 服务器配置，这是一个十分好用的图形界面配置工具，可以轻松地配置好系统的共享资源。

具体使用步骤如下。

第 1 步，运行以下指令，打开“Samba 服务器配置”窗口，如图 8-5 所示。

```
[root@localhost ~]#system-config-samba
```

注意 要使用 Samba 进行网络文件和打印机共享，必须先在 Red Hat Enterprise Linux 5 的防火墙中进行设置。依次单击“系统”→“管理”→“安全级别和防火墙”菜单命令，在“信任的服务”列表框下选中“Samba”复选框，如图 8-6 所示。

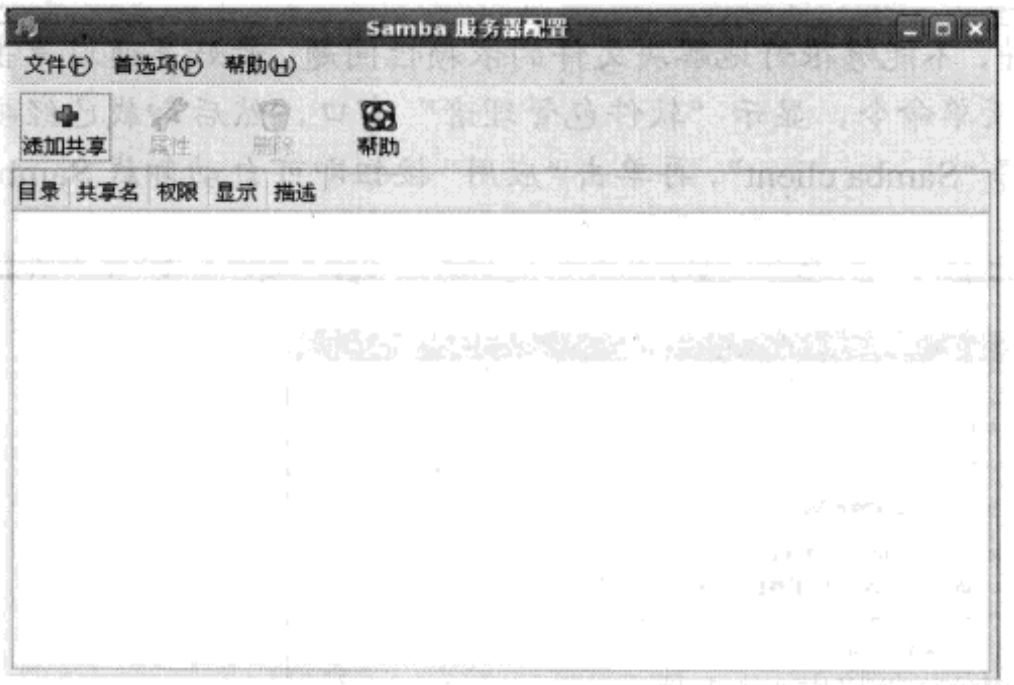


图 8-5 “Samba 服务器配置”窗口



图 8-6 在防火墙上开放 Samba 端口

注意 Red Hat Enterprise Linux 5 下集成有许多实用的系统配置工具，如系统服务配置（system-config-services）、root 密码设置（system-config-rootpassword）、网络配置（system-config-network）等，如图 8-7 所示。

第 2 步，为便于他人识别这台计算机，需要先为这台计算机设置合适的描述和工作组。单击“首选项”→“服务器设置”菜单命令，设置工作组为“hhwork”（根据实际情况而定），保留“描述”的默认设置，如图 8-8 所示。

```
[root@localhost ~]# system-con
system-config-authentication      system-config-packages
system-config-date                system-config-printer
system-config-display             system-config-rootpassword
system-config-kdump               system-config-samba
system-config-keyboard            system-config-securitylevel
system-config-language            system-config-securitylevel-tui
system-config-lvm                 system-config-selinux
system-config-netboot             system-config-services
system-config-network             system-config-soundcard
system-config-network-cmd         system-config-time
system-config-network-gui         system-config-users
system-config-network-tui         system-control-network
```

图 8-7　Red Hat Enterprise Linux 5 下的系统配置工具列表



图 8-8　设置计算机相关信息

第 3 步，切换至“安全性”选项卡，Samba 默认采用的验证模式是“用户”级别，即“security=user”的模式，表明由 Samba 服务器完全控制用户身份的认证，如图 8-9 所示。

下面简单介绍一下 3 种用户验证模式。

- 用户身份验证模式

Samba 支持包括 ADS、域、服务器、共享和用户级别等用户认证模式，如图 8-10 所示。

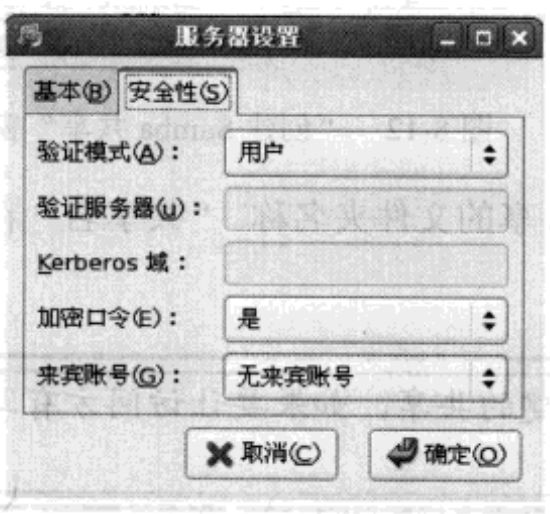


图 8-9　Samba 默认的身份验证方法是“用户”等级

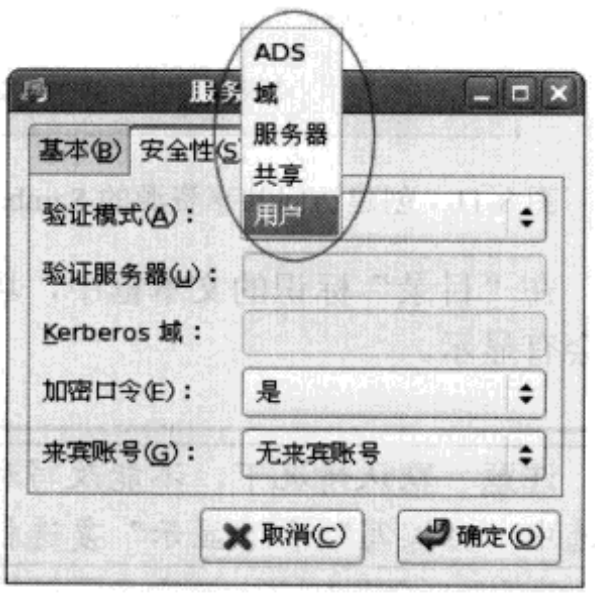


图 8-10　Samba 支持的用户身份验证模式

ADS（Active Directory System）：活动目录级别，只在 Windows 2000 以上版本包含该级别。

域（Domain Level）：Samba 成为域的一部分，并且使用主域控制器（PDC）来进行用户身份验证。一旦通过了身份验证，这个用户将会获得一个特殊的标志，允许他访问有访问权的各种共享。

服务器（Server Level）：Samba 使用一个单独的 SMB 服务器来对用户身份进行验证。

用户（User Level）：Samba 服务器对用户身份进行验证，只有通过验证的用户才可以访问相应的共享。

共享（Share Level）：工作组中的每个共享都有一个或者多个与之相联系的密码，任何一个用户只要知道该密码就可以访问该共享（Windows 95/98/ME 中采用这种级别）。

- 口令加密

Samba 采用的是加密口令传输，而对于某些比较老的系统，如 Windows 95、NT 默认并不支持加密口令传输，要访问 Samba 服务器就必须修改注册表。

- 来宾账号

默认配置下，Samba 支持匿名用户访问（就是不必输入用户名和密码就可以访问）的选项并没有被启用。如果没有安全方便的顾虑，想要创建一个所有人都可以访问的共享，可以考虑开启来宾账号。

第 4 步，在设置 Samba 共享之前，还必须为共享的用户创建 Samba 账号。在“Samba 服务器配置”主窗口，单击“首选项”→“Samba 用户”菜单命令，创建一个名为“adm”的 Samba 账号，如图 8-11 所示。

第 5 步，单击“确定”按钮返回“Samba 服务器设置”主窗口，然后单击左上角的“添加共享”按钮，显示如图 8-12 所示的“创建 Samba 共享”窗口。

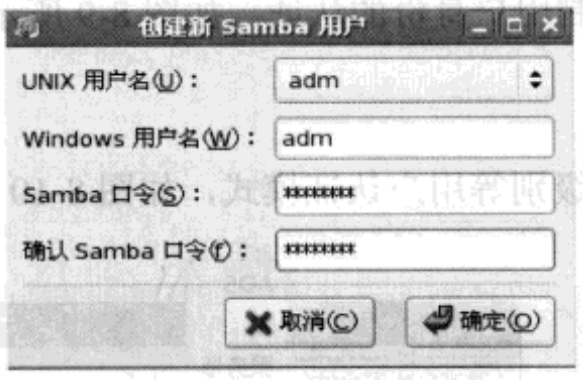


图 8-11　创建访问共享资源的 Samba 账号

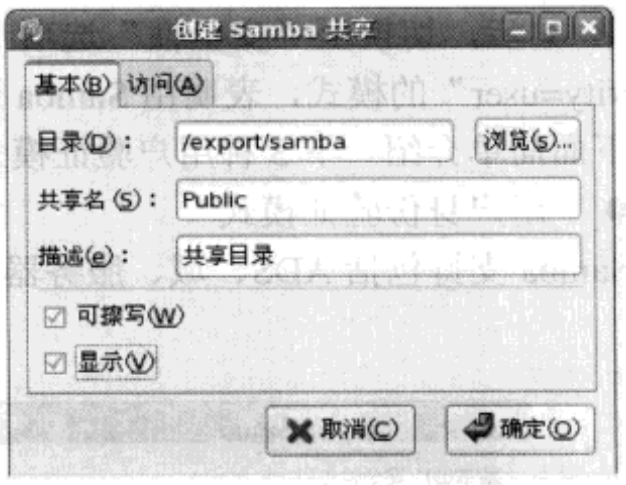


图 8-12　“创建 Samba 共享”窗口

在“目录”标识的文本框中，填入本地设置为共享的文件夹名称。“共享名”在网上邻居中会有显示。

注意　默认情况下，不能改写和显示 Samba 创建的共享，如果要让访问方有写权限，应当选中“可擦写”和“显示”复选框。

第 6 步，为了使访问的安全性更加得完善，在设置了“共享目录”之后，最好还要针对用户，在“编辑 Samba 共享”窗口的“访问”选项卡下进行相应的访问设置，如图 8-13 所示。

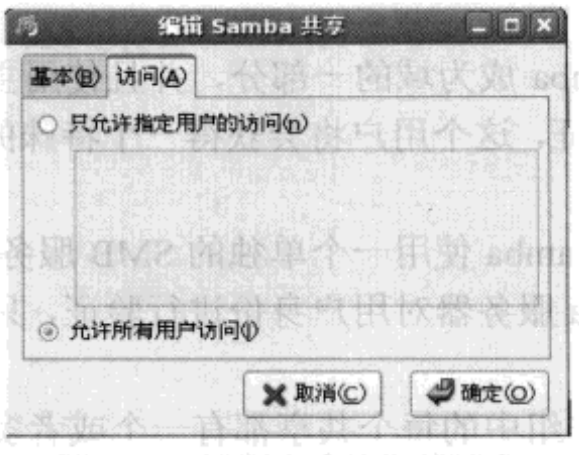


图 8-13　设置允许访问的用户

注意 虽然已经开启了共享，但在 Windows 下单击目录或者 Samba 计算机时，可能会显示提示信息“您可能没有权限使用网络资源。请与这台服务器的管理员联系以查明您是否有访问权限。找不到网络路径。”，如图 8-14 所示。

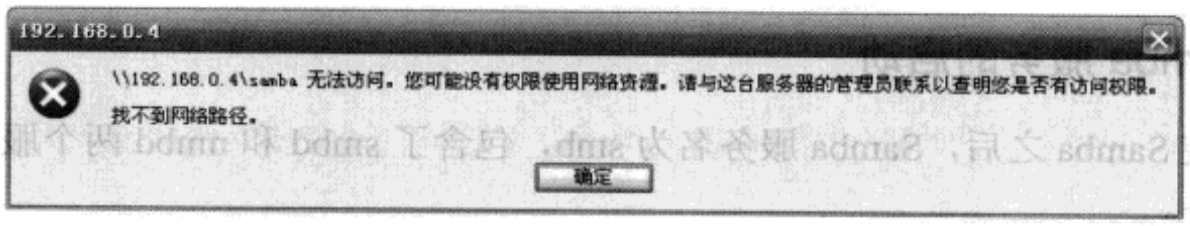


图 8-14 没有权限使用网络资源

在 Samba 服务器上，显示有 SELinux 的拒绝访问提示，如图 8-15 所示。

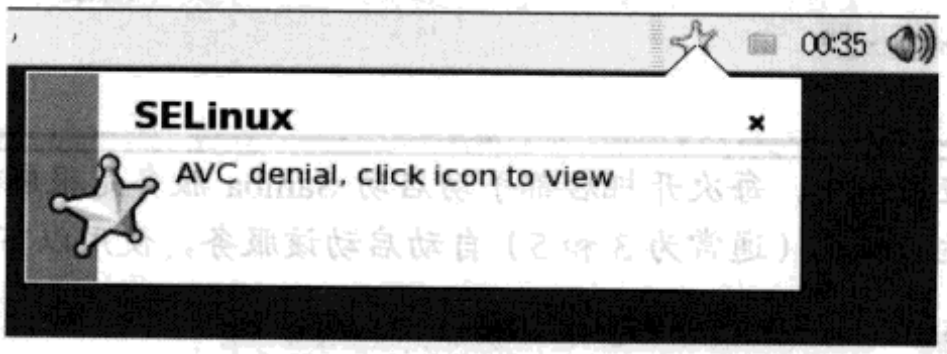


图 8-15 SELinux 拒绝访问提示

要让客户端访问 Linux 服务器上的共享资源，还需要特殊的设置。单击 SELinux 的提示星号，在随后显示的窗口中包含有详细的说明和开放目录的方法，如图 8-16 所示。

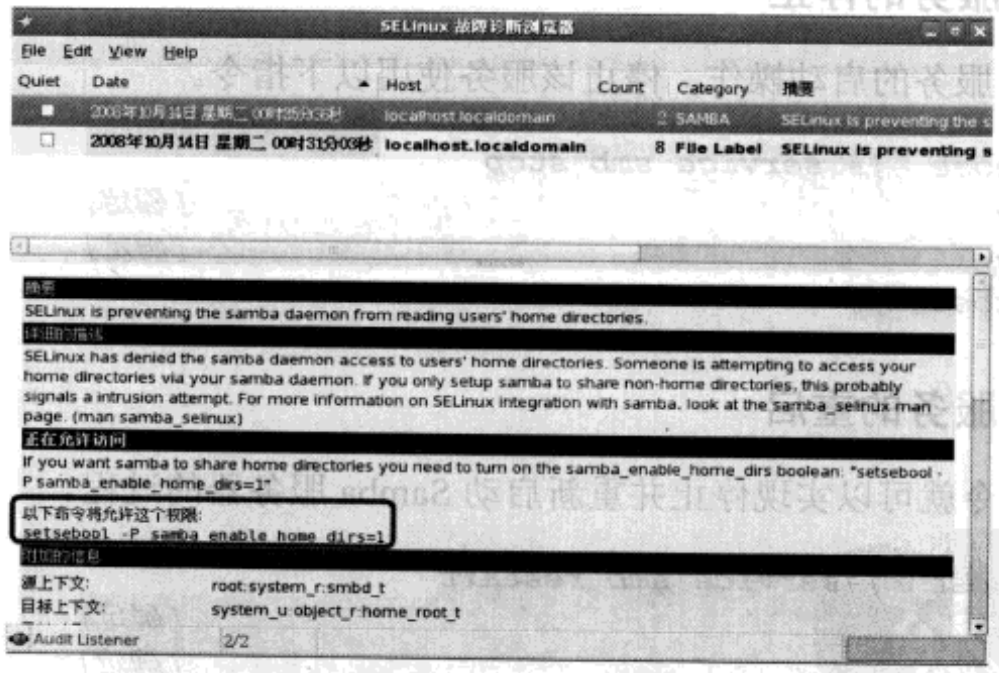


图 8-16 设置开放家目录访问权限的办法

开放家目录访问权限的指令如下。

```
[root@localhost ~]#setsebool -P samba_enable_home_dirs=1
```

开放“/etc/samba”目录访问权限的指令如下。

```
[root@localhost ~]#chcon -R -t samba_share_t  /etc/samba
```

8.1.4 启动、停止和重启 Samba 服务

在 Red Hat Enterprise Linux 5 下，Samba 程序是被安装为服务方式的，所以遵循一般的服务启动、停止规范。

1. Samba 服务的启动

成功安装 Samba 之后，Samba 服务名为 smb，包含了 smbd 和 nmbd 两个服务。执行以下指令启动服务。

```
[root@localhost ~]#service smb start
启动 SMB 服务:                                          [确定]
启动 NMB 服务:                                          [确定]
[root@localhost ~]#
```

注意　在实际应用当中，每次开机后都手动启动 Samba 服务是很麻烦且不现实的，可以设置系统在指定的运行级别（通常为 3 和 5）自动启动该服务，使用以下指令。

```
[root@localhos3t ~]#chkconfig --level 35smbd on
[root@localhost ~]#chkconfig --level 35 nmbd on
```

2. Samba 服务的停止

类似于 Samba 服务的启动操作，停止该服务使用以下指令。

```
[root@localhost ~]# service smb stop
关闭 SMB 服务:                                          [确定]
关闭 NMB 服务:                                          [确定]
[root@localhost ~]#
```

3. Samba 服务的重启

通过简单的指令就可以实现停止并重新启动 Samba 服务器的过程。

```
[root@localhost ~]#service smb restart
关闭 SMB 服务:                                          [确定]
关闭 NMB 服务:                                          [确定]
启动 SMB 服务:                                          [确定]
启动 NMB 服务:                                          [确定]
[root@localhost ~]#
```

8.2 Samba 服务的配置文件

Samba 服务的图形化配置方法简单，用户很容易掌握，但是其可定制性不够丰富。Samba

服务有许多功能只有通过其配置文件才能实现，也就是说要想完全掌控 Samba 的文件共享服务，就应当学习其配置语句。

8.2.1 Samba 服务的主要配置文件

smb.conf 是 Samba 服务的核心控制文件，主要用来设置如何为其他计算机提供文件和打印机共享服务。Samba 服务配置文件 smb.conf 的内容大致如下。

```
#-----------------------Global
Settings-------------------------------------

[global]
    workgroup = linux-china
    server string = Samba Server Version %v
    netbios name = Samba Server

# ----------------------- Standalone Server Options ----------------------

;   security = user
    passdb backend = tdbsam

# --------------------------- Printing Options -----------------------------

    cups options = raw
    username map = /etc/samba/smbusers

#                                                 ----------------------------Share
Definitions-------------------------------

[homes]
    comment = Home Directories
    browseable = no
    writeable = yes

[printers]
    comment = All Printers
    path = /var/spool/samba
    browseable = no
    printable = yes

[Public]
    comment = toplinux 的共享资源
    path = /export/samba
    writeable = yes
    guest ok = yes
```

8.2.2 Samba 服务的密码文件

在 Red Hat Enterprise Linux 5 下，随着加密方法的不同，Samba 服务的密码文件也各不相同。比如密码文件名是 passdb.tdb（位于/etc/samba 目录下），默认采用的加密方式是 tdbsam。

```
[root@localhost ~]# cd /etc/samba
[root@localhost samba]# ls
lmhosts  passdb.tdb  secrets.tdb  smb.conf  smbusers
[root@localhost samba]# ls -la
总计 76
drwxr-xr-x   2 root root  4096 04-29 19:09 .
drwxr-xr-x 107 root root 12288 07-06 16:01 ..
-rw-r--r--   1 root root    20 2008-04-03 lmhosts
-rw-------   1 root root  4096 04-29 19:09 passdb.tdb
-rw-------   1 root root  8192 04-29 19:09 secrets.tdb
-rw-r--r--   1 root root  9733 2008-04-03 smb.conf
-rw-r--r--   1 root root    97 2008-04-03 smbusers
[root@localhost samba]#
```

8.2.3 Samba 服务的日志文件

要检测编写好的 Samba 服务控制文件 smb.conf 是否有问题，最简单的办法就是使用 Red Hat Enterprise Linux 5 内置的 smbclient 指令来完成。

```
[root@localhost ~]#smbclient -U% -L localhost
Domain=[MYGROUP] OS=[Unix] Server=[Samba 3.0.28-0.el5.8]
        Sharename       Type      Comment
        ---------       ----      -------
        IPC$                    IPC             IPC Service (Samba Server Version
3.0.28-0.el5.8)
        VMware_Virtual_Printer Printer   VMware_Virtual_Printer
Domain=[MYGROUP] OS=[Unix] Server=[Samba 3.0.28-0.el5.8]
        Server               Comment
        ---------            -------
        LOCALHOST            Samba Server Version 3.0.28-0.el5.8
        Workgroup            Master
        ---------            -------
        MYGROUP
[root@localhost ~]#
```

注意 如果 smb.conf 文件配置有问题的话，可以通过查看文件/var/log/samba/%m.log 中的记录信息并且检查系统的输出来确定问题的所在。其中“%m”指客户端的 NetBIOS 名称。当主机 toplinux 登录 Samba 主机时，信息会被记录在/var/log/samba/toplinux.log 中。如果来源 IP 没有 Netbios 名称，就会产生一些错误信息并记录在 log.smbd 中，可以根据这个文件确定连接 Samba 主机时发生了什么问题。

8.3 smb.conf 文件详解

Samba 主要是通过定制其配置文件 smb.conf 来获得各种功能的。由于 smb.conf 是一个非

常复杂的协议，在配置的过程中如果出错就比较麻烦了，所以最好是在配置该文件之前创建一个备份，以便在出错时进行还原。

8.3.1 smb.conf 文件结构

对 Windows 很熟悉的用户，对 smb.conf 文件应该很容易理解，其文件结构与 Windows 下的某些配置文件很相似。

1. 整体结构

smb.conf 文件的整体结构如下。

```
[global]
......
[homes]
......
[printers]
......
[Public]
......
```

方括号中的名字对应配置文件的段，其中[homes]和[Public]定义了两个共享，[printers]定义了服务器上连接打印机的选项。

（1） [global]

[global]出现在每一个 Samba 的配置文件中，定义了服务器本身的配置参数及其他共享资源的默认配置参数，相当重要。

> 注意 如果其他部分包含有和[global]相同的选项并且赋值，此时规定以其他部分设置为准。

（2） [homes]

该部分指定 Windows 共享的主目录，如果在 Windows 工作站登录的用户名和密码与 Linux 的用户名及密码相同，那么通过打开“网上邻居”，再双击“共享目录”的图标就可以成功访问共享目录了。从 Windows 访问 Linux 主目录时，用户名将作为主目录共享名。

（3） [printers]

该部分用于指定如何共享 Linux 网络打印机，从 Windows 系统访问 Linux 网络打印机时，共享的是 printcap 中指定的 Linux 打印机名。

2. 注释

为了方便阅读，smb.conf 中包含了很多注释信息，以一个“#”开始的行为注释行。

3. 连续行

Samba 配置文件中，每一行最多可支持 255 个字符。为显示方便，可以分行输入过长的行，行之间以“\”分割。

```
comment =    这是 toplinux 的共享资源，欢迎大家访问 \
             也欢迎大家踊跃上传
```

4. 配置文件检查

在 Samba 工作期间，每隔 60 s 会检查一次配置文件，如果文件发生了变化，就使改变立即生效。如果希望配置马上生效，还可以向 smbd 和 nmbd 进程发送 SIGHUP 信号来实现。

已知 smbd 的进程号是 9 353，要让 smbd 马上去读取配置文件，执行以下指令就可以了。

```
[root@localhost ~]# kill SIGHUP 9353
```

8.3.2 smb.conf 文件的语法和变量

除了段的信息之外，Samba 配置文件中还包括一些参数的赋值，即“options = value”。这些参数绝大多数都是浅显易懂的。Samba 还支持不少变量，用来动态描述服务器和客户端的连接信息。变量以%开始，后面跟一个单独的大写或者小写字母。

```
[pub]
path = /home/%u
```

“% u”代表当前的 Linux 用户名，更多的变量如表 8-1 所示。

表 8-1 Samba 变量表

变量分类	变量名称	说明
客户端变量	%a	客户端体系，如：Windows 95、WfWg、Windows NT、Samba
	%I	客户端 IP 地址
	%m	客户端 NetBios 名
	%M	客户端 DNS 名
用户变量	%g	用户%u 主要组
	%H	用户%u home 目录
	%u	UNIX 当前用户名
共享变量	%P	当前共享的根目录
	%S	当前的共享名
服务器变量	%d	当前服务器进程 ID
	%h	Samba 服务器的 DNS 名字
	%L	Samba 服务器的 Netbios 名字
	%N	Home 目录服务器，来自于 automount 的映射
	%v	Samba 版本
其他变量	%R	经过协商的 Samba 协议层
	%T	当前日期和时间

8.3.3 smb.conf 文件详解

对于初学者来说，Linux 下的配置文件并不是很容易理解的。但是 Samba 服务的配置文件分成了几个大的部分，只要分别清楚了每一部分所表达的含义，就会发现使用配置文件来配置 Samba 服务并不困难。

1. [global]

```
# ----------------------- Network Related Options -------------------------
    workgroup = hhwork
    server string = Samba Server Version %v
```

设置 Samba 计算机所属的工作组，和相关描述信息。

```
netbios name = Samba Server
```

设置 Samba 服务器在网上邻居中显示的名称，默认是“Samba Server Version 3.0.28-0.e15.8”。为便于 Windows 客户端识别，需进行明确的指定。

```
;   interfaces = lo eth0 192.168.12.2/24 192.168.13.2/24
;   hosts allow = 127. 192.168.12. 192.168.13.
```

hosts allow 参数用于指定允许访问 Samba 服务器的客户机的 IP 地址范围，以提供安全限制。默认情况下，这行配置前面有注释信息，即所有的客户都可以访问 Samba 服务器。

```
# logs split per machine
;   log file = /var/log/samba/%m.log
# max 50KB per log file, then rotate
;   max log size = 50
```

指定 Samba 服务器记录文件的文件名和具体位置（默认的是/var/log/samba/%m.log，其中%m 表示客户端计算机名），并且设置记录文件的最大体积（单位是 KB）。

```
;   security = user
    passdb backend = tdbsam
```

设置 Samba 服务器的用户验证模式，本部分设置的 Samba 密码验证采用的是 tdbsam 机制。验证模式包括 ADS、域、服务器、用户和共享 5 种级别。

```
# ----------------------- Domain Members Options ------------------------
#   password server = My_PDC_Name [My_BDC_Name] [My_Next_BDC_Name]
# or to auto-locate the domain controller/s
#   password server = *
;   security = domain
;   passdb backend = tdbsam
;   realm = MY_REALM
;   password server = <NT-Server-Name>
```

设置密码验证服务器的名字，用于服务器、域和 ADS 验证模式下。

```
# ----------------------- Domain Controller Options ----------------------
;   security = user
;   passdb backend = tdbsam

;   domain master = yes
;   domain logons = yes

# the login script name depends on the machine name
```

```
;   logon script = %m.bat                ; #每一台计算机运行一个指定的登录批处理文件
# the login script name depends on the unix user used
;   logon script = %u.bat                ; #每一个用户运行一个指定的登录批处理文件
;   logon path = \\%L\Profiles\%u        ; #指定登录路径
# disables profiles support by specifing an empty path
;   logon path =

;   add user script = /usr/sbin/useradd "%u" -n -g users
;   add group script = /usr/sbin/groupadd "%g"
;   add machine script = /usr/sbin/useradd -n -c "Workstation (%u)" -M -d /nohome
-s /bin/false "%u"
;   delete user script = /usr/sbin/userdel "%u"
;   delete user from group script = /usr/sbin/userdel "%u" "%g"
;   delete group script = /usr/sbin/groupdel "%g"
```

设置的主域控制器、密码验证模式，以及登录前和登录后需要进行的操作，在域模式下适用。

```
#   ----------------------          Browser         Control         Options
-----------------------------
;   local master = no
;   os level = 33
;   preferred master = yes
```

浏览器控制部分，用于设置 OS 级别和是否参加浏览器竞选。

```
#-------------------------------          Name                Resolution
-------------------------------
;   wins support = yes
;   wins server = w.x.y.z
;   wins proxy = yes
;   dns proxy = yes
```

设置 Samba 服务器是否支持 WINS 和 DNS 代理。

```
# --------------------------- Printing Options -----------------------------
;   load printers = yes
    cups options = raw
    username map = /etc/samba/smbusers
;   security = user
;   encrypt passwords = yes
;   guest ok = no
;   guest account = nobody
;   encrypt passwords = yes          ; #使用密码加密
;   guest ok = no                    ; #不允许匿名用户登录
;   guest account = nobody           ; #映射匿名用户为 Linux 下的 nobody 账号

;   printcap name = /etc/printcap  ; #指定打印机名称
#obtain list of printers automatically on SystemV
;   printcap name = lpstat
;   printing = cups                  ; #指定打印机类型
```

设置打印机控制部分。

2. [homes]

```
    comment = Home Directories
    browseable = no
    writeable = yes
;   valid users = %S
;   valid users = MYDOMAIN\%S
```

为用户设置一个随用户名变化而变化的动态目录，映射到相应的 Linux 用户家目录中。

3. [printers]

```
    comment = All Printers
    path = /var/spool/samba
    browseable = no
;   guest ok = no                    ; #不允许匿名访问
;   writeable = no
    printable = yes                  ; #允许打印
```

本部分用于设置共享的打印机。

4. 其他共享

由于前文中设置有一个 Public 共享，所以在 smb.conf 中会单独列出一个[Public]部分来。

```
[Public]
    comment = toplinux 的共享资源
    path = /export/samba
    writeable = yes                  ; #允许写入
;   browseable = yes
    guest ok = yes                   ; #允许匿名访问
```

8.3.4 smb.conf 文件中的段

一般 smb.conf 文件由 3 个标准段和若干个用户自定义共享段所组成。以下将分别介绍各个段。

- Global：用于定义全局参数和默认值。
- Homes：用于定义用户的 Home 目录共享。
- Printers：用于定义打印机共享。
- Userdefined_ShareName：用户自定义共享（可有多个）。

8.3.5 Samba 服务具体参数设置

利用 Samba 服务进行文件共享，在很多情况下都需要加上相关参数实现。只有明白 Samba 服务中各种参数的含义，才能更好地掌握文件共享的配置管理。

1. 全局参数

- Workgroup=域名/工作组名

用于设定 Samba 服务器所属工作组/域的名称。

- Server string=字符串

用于设置 Samba 服务器的描述。

- hosts allow = IP 地址或 IP 地址段

用于设置允许连接到服务器的 IP 地址。

- guest account = 客户账号

用于设置允许作为客户登录的账号。

- security=user/share/domain/server

用于设置 Samba 的共享安全级别。

注意 “share //”表示共享级访问，服务器不对客户机进行身份验证；“user //”表示用户级访问；“server //”表示服务器级访问，被访问的 Samba 服务器请求另一台 Samba 服务器对客户机进行身份验证；“domain //”表示域安全级别访问，由域控制器对客户机进行身份验证。

- password server=Ip 地址/主机名

用于指定密码服务器的位置，当 security 值为 server/domain 时必须设置该参数。

- encrypt passwords=yes/no

用于指定是否使用加密口令。

注意 对于 Windows 客户机来说，应该设置该选项为“yes”。

- username map=文件路径

用于指定 SMB 用户名与 Linux 用户名映射文件的路径。

```
username map=/etc/samba/smbusers
```

- interfaces=接口名/ip 地址

用于设置 Samba 服务器的 IP 地址，在不想使用默认 IP 地址或者想同时使用多个 IP 地址的时候，可以设置该选项。

- smb passwd file=文件路径

用于指定 Samba 用户密码的文件位置。

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

2. 打印机设置参数

- printcap name=路径

用于设定打印机的配置文件的位置。

```
printcap name=/etc/printcap
```

- load printers =yes/no

用于设定是否自动装载打印机的装载文件，当要作为打印服务器时必须设置为“yes”。

- printing = /cups/bsd/lprng/sysv

用于设置打印相关指令时所采取的模式。

```
printing=lprng
```

- printer name=名称

用于设置打印机的名称。

3. 日志设置参数

- log file=路径

用于定义日志文件的位置。

```
log file=/var/log/samba/%m.log
```

- max log size=数字

用于定义日志文件的最大千字节。

4. wins 设置参数

- wins support=yes/no

设置是否支持 wins 解析。

```
wins support=yes
```

- wins server=服务器

用于指定 wins 服务器的位置。

```
wins server=192.168.0.2
```

- wins proxy=yes/no

用于设置 Samba 服务器是否支持 wins 代理功能。

```
wins proxy=yes
```

8.4 Samba 服务功能实现

在对 Samba 服务的配置文件有一个基本的认识之后，下面就来实现其不同的功能。如果说用户还是觉得 smb.conf 文件过于复杂，也可以根据本节中各种功能实现的方法来更进一步地理解 Samba 服务的配置。

8.4.1 指定 Samba 所要加入的工作组

配置 Samba 服务器，让其与 Windows 进行通信，首先要将其加入一个工作组，对应的设

置在 smb.conf 文件中。

```
# workgroup = NT-Domain-Name or Workgroup-Name
workgroup = mygroup
```

该语句设定了 Samba 服务的工作组，默认设置是“workgroup = mygroup”，该设置可以是 Windows 网络的工作组或者域。

加入工作组的语法格式如下。

```
workgroup = <工作组名>;
```

如果要 Samba 的工作组设置为和 Windows 同一个工作组，例如 Samba 工作组的名称为“toplinux”，这时可使用如下语句。

```
workgroup = toplinux
```

网上邻居中显示有共享的此计算机用户名，并进行通信。

8.4.2 注释说明服务器

samba.conf 关于服务器的描述如下。

```
# server string is the equivalent of the NT Description field
server string = Samba Server
```

该语句的语法格式如下。

```
server string = <说明>
```

默认设置为 sarver string = Samba Server。

支持的变量包括：t%（访问时间）、I%（客户端 IP）、m%（客户端主机名）、M%（客户端域名）和 S%（客户端用户名）。

例如，“server string = this is a Samba Server”设定了 Windows 网上邻居的“Samba Server”注释为“this is a Samba Server”。

8.4.3 限制可访问服务器 IP 地址的范围

如果希望只允许指定 IP 地址段的用户访问服务器，就需要更进一步地限定访问。可以使用“host allow”命令来实现，相关代码语句如下。

```
# This option is important for security. It allows you to restrict
# connections to machines which are on your local network. The
# following example restricts access to two C class networks and
# the "loopback" interface. For more examples of the syntax see
# the smb.conf man page
; hosts allow = 192.168.1. 192.168.2. 127.
```

该语句的语法格式如下。

```
hosts allow = <IP 地址>; ...
```

默认设置为“hosts allow = 192.168.1. 192.168.2. 127.”。

限制允许连接到 Samba 服务器的机器，参数之间以空格隔开。表示方法可以为完整的 IP 地址，如“192.168.0.1”；或者网段，如“192.168.0.x”。

注意 如果要激活该语句，则需删除句头的“;”。

例如，“hosts allow = 192.168.1. 192.168.0.1”表示允许网段“192.168.1”和网址为“192.168.0.1”的机器连接 Samba 服务器。

8.4.4 打印机设置

在 Samba 服务当中，打印机的设置也是相当重要的，在 smb.conf 文件中实现该功能的办法如下。

1. 定义打印机配置文件

默认情况下，在可浏览共享对象的列表中就包含有打印机，在 smb.conf 文件中，打印机的配置文件相关语句如下。

```
# if you want to automatically load your printer list rather
# than setting them up individually then you'll need this
printcap name = /etc/printcap
load printers = yes
```

“printcap name”的语法格式如下：

```
printcap name = <打印机配置文件>;
```

默认设置为“printcap name = /etc/printcap”，可以进行修改。

2. 设置是否自动加载打印机

设置打印机是否自动加载，也可以在 smb.conf 文件中进行，相关语句格式如下。

```
# if you want to automatically load your printer list rather
# than setting them up individually then you'll need this
printcap name = /etc/printcap
load printers = yes
```

“load printers”的语法格式如下：

```
load printers = <yes/no>;
```

默认设置为“load printers = yes”，若不想自动加载打印机，将设置改为“no”，就不会自动加载了。

3. 设置打印机类型

若使用的是标准打印机，则其类型就不需要进行设置了，但在使用并非标准打印机的情

况下，还需要设置打印机的类型。

```
# It should not be necessary to spell out the print system type unless
# yours is non-standard. Currently supported print systems include:
# bsd, sysv, plp, lprng, aix, hpux, qnx
; printing = cups
```

“printing”的语法格式如下：

```
printing = <打印机类型>;
```

默认设置为 “printing = cups”，若想激活该语句，删除语句前的“;”即可。

语句“# bsd，sysv，plp，lprng，aix，hpux，qnx”列出的是此系统支持的打印机类型。

8.4.5 设置 guest 账号匿名登录

Samba 允许创建一个标准的来宾账号(即访问时不用输入用户名和密码的账户)。如让一些用户有权限访问特定内容，但不可能为每个用户配置密码，这时可以通过 smb.conf 文件进行如下设置。

```
# Uncomment this if you want a guest account, you must add this to /etc/passwd
# otherwise the user "nobody" is used
;guest account = pcguest
```

“guest account”的语法格式如下：

```
guert account = <账户名称>;
```

默认设置为 guert account = pcguest。

当不激活此命令时默认用户为“nobody”，激活该语句则去掉此语句前的分号“;”。激活该命令，还需要确保 pcguest 是本地 Linux 系统上的一个真实用户。

例如，guert account = toplinux 设定访问 samba server 的来宾账户以 toplinux 登录，则此登录账户享有 toplinux 用户的所有权限。

8.4.6 设置日志文件名、存放路径及大小

Samba 服务器的每一台计算机，都需要配置独立的日志文件。例如，一台名为 hhwork 的 Windows 计算机，通过设置日志文件，在指定目录下就可以找到日志文件 hhwork.log 并查找到调试信息。通过 smb.conf 文件可以进行如下设置。

```
# this tells Samba to use a separate log file for each machine
# that connects
log file = /var/log/samba/%m.log
# all log information in one file
# log file = /var/log/samba/smbd.log
# Put a capping on the size of the log files (in Kb).
max log size = 50
```

“log file”的语法格式如下：

```
log file = <日志文件>
```

默认设置为“log file = /var/log/samba/%m.log”。

设定 Samba 服务日志文件的文件名和储存位置（“%m”代表客户端主机名）。

“max log size”的语法格式如下。

```
max log size = <? KB>;
```

指定日志文件的最大容量，单位为 KB。预设 max log size = 50。

8.4.7 服务器安全设置

Samba 服务的功能配置好以后，最好不要忙于使用该服务。在正式实现资源共享之前，一定要先考虑安全方面的问题。下边就来进行 Samba 服务的安全配置工作。

1. Samba 服务器的安全等级

Samba 支持的用户身份验证等级有 5 种，包括：共享安全级别（share）、用户安全级别（user）、服务器安全级别（server）、域安全级别（domain）和活动目录安全级别（ads）。相关的设置位置在/etc/samba/smb.conf 中，如图 8-17 所示。

```
security = [share |user | server |domain |ads]
```

```
# workgroup = NT-Domain-Name or Workgroup-Name, eg: MIDEARTH
        workgroup = MYGROUP

# server string is the equivalent of the NT Description field
        server string = Samba Server

# Security mode. Defines in which mode Samba will operate. Possible
# values are share, user, server, domain and ads. Most people will want
# user level security. See the Samba-HOWTO-Collection for details.
        [redacted]

# This option is important for security. It allows you to restrict
# connections to machines which are on your local network. The
# following example restricts access to two C class networks and
# the "loopback" interface. For more examples of the syntax see
# the smb.conf man page
;   hosts allow = 192.168.1. 192.168.2. 127.
```

图 8-17 Samba 支持 5 种安全等级

（1） 共享安全级别 security=share

共享级的权限，这是 Windows 95 文件和打印服务器的默认设置。客户端在连接请求期间发送一个口令，不需要任何相关的用户信息。但是因为安全性不高，且目前 Windows 95 操作系统已基本淘汰，所以不推荐使用。

（2） 用户安全级别 security=user

目前 Samba 默认的安全等级，Samba 在接收到用户的访问请求时，会检查密码（必须是用户名和密码在/etc/samba/smbpasswd 中有定义的情况）。这里 Samba 所使用的密码，是微软的编码方式。

为了保证安全，还必须进行如下设置。

```
encrypt passwords = yes
```

这种方式有一个好处是：密码不会在网络上传送，所以不会存在明文密码在网上被嗅探的安全问题。

（3） 服务器安全级别 security=server

该模式下，Samba 把密码验证的工作交给指定的服务器，只有在无法通过认证的情况下才会切换至 security=user 模式。

（4） 域安全级别 security=domain

该模式下，类似于 Windows 服务器安装时加入域的动作，Samba 会模拟一台加入 Windows 域的服务器，但必须有 Windows 域的管理员账号。

注意 在 security=server 和 security=domain 安全级别模式下，都需要指定“password server = ServerName”选项。

（5） 活动目录安全级别 security=ads

从 Samba 3.0 开始，就可以很完美地支持 Windows 2000/2003 服务器 ADS。采用该模式的客户端必须是 Windows 2000 以上的版本。

2. 设置服务器的安全级别

Windows 网络基本的安全模式有好几个。一般来说，到底选择什么样的安全模式，是由共享目录和共享网络类型决定的。通过 smb.conf 文件可以进行如下设置。

```
# Security mode. Most people will want user level security. See
# security_level.txt for details.
security = SHARE
```

“security”的语法格式如下：

```
security = <等级>
```

默认设置“security = share”。设定访问 Samba 服务器的安全级别共有 5 种，如表 8-2 所示。

表 8-2 安全级别描述

安全级别	具体描述
share	不需要提供用户名和密码
user	需要提供用户名和密码，身份验证由 Samba 服务器负责
server	需要提供用户名和密码，身份验证可指定其他机器（Windows NT/2000/XP）或另一台 Samba 服务器执行
domain	需要提供用户名和密码，身份验证指定 windows NT/2000/XP 域服务器执行
ads	用于 Windows Active Directory 域中的一个 Member Server 系统

3. 设置密码验证级别

Samba 中可以使用的密码设置有好几个，并且可以指定为 Windows 用户名与密码配置的集中式服务器。

smb.conf 文件中有如下设置。

```
# Use password server option only with security = server
; password server = <NT-Server-Name>
# Password Level allows matching of _n_ characters of the password for
# all combinations of upper and lower case.
; password level = 8
; username level = 8
```

（1） password server 语法格式

```
password server = <IP 地址/主机名>
```

默认设置为“password server = <NT-Server-Name>”。

激活该语句的方法是删除语句前的“;”。

该语句用于在已经设置了“security = server”或“security = domain”的情况下，要指定用于网络的那些密码服务器。例如，知道 PDC 和 BDC 的名称是 NTSer1 和 NTSer2，则可以插入下面的这条命令。

```
password server = NTSer1 NTSer1
```

如果不知道 PDC 和 BDC 的名称，下面这条命令便把 Samba 服务器设置在对域控制器的一个搜索结果上。

```
password server = *。
```

（2） password level 和 username level 语法格式

```
password level = <位数>,
username level = <位数>,
```

设定用户名和密码的位数，默认设置为 8 个字符。

```
; password level = 8
; username level = 8
```

如果想激活该语句，可以删除语句前的“;”。

4. 对密码进行加密设置

通常情况下 Samba 会被设置为从标准 Samba 密码文件中发送加密密码。该密码文件包含用“smbpasswd –a”命令添加的 Windows 用户名和密码。因为还是存在有 Windows 计算机不能处理加密密码的情况，所以需要设置是否允许 Samba 密码文件发送加密密码。smb.conf 文件中可进行如下设置。

```
# You may wish to use password encryption. Please read
# ENCRYPTION.txt, Win95.txt and WinNT.txt in the Samba documentation.
```

```
# Do not enable this option unless you have read those documents
;encrypt passwords = yes
;smb passwd file = /etc/samba/smbpasswd
```

“encrypt passwords”语法格式如下。

```
encrypt passwords = <yes/no>
```

默认设置 encrypt passwords = yes。

注意 激活该语句需要删除语句前的“;”，如果未激活，则默认通过网络明文形式发送密码。

5. 建立账号映射关系

如何把 Linux 与 Windows 中具有不同用户名的用户相连起来呢？通过在 /etc/samba/smbusers 数据库文件中存储对应的名称就能实现。要使用该数据库，需要先激活 smb.conf 文件相应的语句。

```
# Unix users can map to different SMB User names
; username map = /etc/samba/smbusers,
```

“username map”语法格式如下。

```
username map = <存储对应的名称数据库文件>
```

默认设置 username map = /etc/samba/smbusers。

8.4.8 具体的共享设置

在 smb.conf 文件中配置共享，还可以设置共享的各种参数和操作系统级别等。具体的配置方法如下。

1. 描述共享

在 smb.conf 文件中，共享的描述可以通过以下参数进行设置。

```
comment = <共享描述>
```

例如：“comment = myshare”

2. 提供共享服务的路径

在 smb.conf 中，对共享的路径可以通过以下参数进行设置。

```
path = <共享服务的路径>
```

例如：path = /temp。

3. 共享路径及资源配置

提供资源分享时，欲分享的资源需要先加上符号“[ ]”，通常还会加上指令和参数来表示资源的设定和存取权限等，具体参数如表 8-3 所示。

表 8-3　分享资源的参数说明

参　数	具 体 描 述
comment	注释说明
path	分享资源的完整路径名称，除了路径要正确外，目录的权限也要设对
browseable	是 yes/否 no 在浏览资源中显示共享目录，若为否则必须指定共享路径才能存取
read only	是 yes/否 no 以只读方式共享，当与 writable 发生冲突时以 writable 为准
writable	是 yes/否 no 不以只读方式共享，当与 read only 发生冲突时，忽略 read only
read list	设定此名单内的成员为只读（用户名/@组名）
write list	若设定为只读时，则只有此设定的名单内的成员才可做写入动作(用户名/@组名)
deny hosts	设定只有此网段/IP 的用户不能访问共享资源。allow hosts=本网段指定 IP，deny hosts=本网段指定 IP
printable	是 yes/否 no 允许打印
public	是 yes/否 no 公开共享，若为否则进行身份验证（只有当 security = share 时此项才起作用）
vaild users	设定只有此名单内的用户才能访问共享资源
invalid users	设定此名单内的用户不能访问共享资源
allow hosts	设定只有此网段/IP 的用户才能访问共享资源

打开配置文件 smb.conf 组织本地共享资源并进行路径配置，smb.conf 文件内容如下所示。

```
[root@localhost ~]# vi /etc/samba/smb.conf
[Resource]
    comment = Local Resource
    path = /var/samba/resource
    public = yes
writeable = yes
// -p parent(no error if existing, make parent directories as needed)
# mkdir -p /var/samba/resource
#cd /var/samba/resource
//-s, --symbolic
# ln -s /usr/share/doc doc
; #为所有用户配置只读共享和读写共享
 [tmp]
    comment = Temporary file space
    path = /tem
    read only = no
    public = yes

[cdrom]
    comment = CDROM
    path = /mnt/cdrom
    read only = yes
public = yes
; #为指定用户或组配置共享
[root@localhost ~]# useradd samba
[root@localhost ~]# passwd samba
[root@localhost ~]# mkdir /var/samba/samba
; #更改目录属主
[root@localhost ~]# chown samba.samba /var/samba/samba
[root@localhost ~]# vi smb.conf

[sambadir]
```

```
    comment = Samba's service
    path = /var/samba/samba
    valid users = samba
    public = no
writable = yes

; #保存修改
```

Linux 环境下访问 Samba 共享如下。

```
[root@localhost ~]# cat /etc/samba/lmhosts
; smbclient -L IP -U 用户名
[root@localhost ~]# smbmount //win/tools /mnt/smb/win
```

4. 操作系统级别设置

通过 smb.conf 文件设置操作系统的级别如下。

```
# OS Level determines the precedence of this server in master browser
# elections. The default value should be reasonable
;   os level = 33
```

“os level”的语法格式如下。

```
os level = <数字>;
```

默认设置“os level = 33”。通过这条语句可以设定 Samba 服务的操作系统级别，“os level”的有效值在 0~255 之间，Windows 95/98 的 os level 为 1。

若 Samba 服务器作 LMB 或 DMB，则“os level”至少要大于 Windows NT 的 33。

8.5 共享的资源访问

在通过 Samba 服务配置好各种共享资源之后，下面就将介绍共享资源的访问方式了。而 Windows 系统与 Linux 系统访问共享资源的方式又有所不同。本节将分别介绍异机资源共享和同主机的不同操作系统之间的资源共享。

8.5.1 不同客户端主机的资源访问

Samba 服务器配置无误并且设置好共享资源之后，就可以通过客户端访问共享资源了。而根据客户端操作系统的不同，Samba 访问共享资源的方式也有一定的差异。

1. 从 Windows 客户端访问共享资源

Windows 用户使用正确的账户名和密码，通过“网上邻居”，可以直接访问 Samba 服务器共享资源，如图 8-18 所示。

2. 从 Linux 客户端访问共享资源

Linux 客户端在图形桌面环境下，访问共享资源与 Windows 没有太大的区别。不同的是，Linux 提供强大的命令行工具，某些时候使用会更方便。

第 1 步，在 Red Hat Enterprise Linux 5 下，可以从命令行访问服务器段提供的共享资源。使用 smbclient 工具，连接到服务器上的 music 共享文件，执行如下指令。

```
[root@localhost ~]#smbclient //192.168.0.4/music -U Administrator
Password:
Domain=[toplinux] OS=[Windows 5.1 ] Server=[Windows 2000 LAN Manager]
smb:\>
```



图 8-18 Samba 共享文件夹

第 2 步，在“smb:\>”提示符下，命令行的使用就如同 FTP 工具，可以输入各种指令以实现对应功能，如 ls——列表，mget——多文件下载，mput——多文件上传。

> 注意 若需查看帮助信息，输入“help”指令即可。

8.5.2 本机 Windows 分区和 Linux 分区的资源互访

Samba 服务解决了异机之间资源共享和互访的问题。但是如何实现同一台计算机上不同的操作系统之间的资源共享呢？比如，在一台计算机上同时安装有 Windows 系统和 Linux 系统，这两个系统之间需要实现资源的传送和互访，该如何进行呢？本节将详细介绍实现的方法。

1. 在 Linux 系统下访问 Windows 分区

Red Hat Linux 提供了对几乎所有文件系统的支持，所以在 Red Hat Linux 下可以直接读写 Windows 系统的分区。在安装 Red Hat Linux 的时候，需要指出计算机中存在的 Windows 分区

的位置，并指定载入到 Linux 文件系统中相应的位置。

如果在安装系统时未指定 Windows 分区的位置，那么在 Linux 下察看 Windows 分区的步骤如下。

第 1 步，在 Linux 系统下输入如下命令。

```
[root@localhost ~]#fdisk -l
```

显示所有的分区信息，包括 Windows 的分区信息。

第 2 步，在 Linux 的/mnt 目录下建立 1 个子目录用于存放 Windows 操作系统下的文件。

```
[root@localhost ~]#mkdir /mnt/winpart
```

其中“winpart”是新建的子目录。

第 3 步，执行如下命令。

```
[root@localhost ~]#mount -t auto /dev/hda1 /mnt/winpart
```

至此，完成了 Windows 系统和 Linux 系统的资源互访。在 Linux 操作系统下打开“winpart”目录，即可显示 Windows 分区下的内容。取消在 winpart 目录下查看 Windows 分区文件的功能，使用如下命令。

```
[root@localhost ~]#umount /mnt/winpart
```

2. 在 Windows 系统下访问 Linux 分区文件

与 Red Hat Linux 操作系统相比，Windows 操作系统在处理各种文件系统方面要稍微差一点。从 Windows 操作系统查看 Linux 分区的内容，需要借助于其他软件工具。常用的软件有 explore2fs 和 fsdext2。下面将介绍 fsdext2，该软件提供了在 Windows 操作系统下访问 Linux 分区的功能。

下载安装 fsdext2 软件之后，使用 fsdext2 提供的“mount”命令，将 ext2 文件系统载入到 Windows 系统下。mount 命令的参数是 Linux 分区的设备名（命名规则与 Linux 下一样）。然后 fsdext2 会自动寻找一个未分配的盘符代表 Linux 分区。这样，访问 Linux 一个分区就像是访问 Windows 操作系统下的一个分区一样简单。

第 9 章　NFS 服务器

众所周知，在 Windows 操作系统中可以通过设置“网上邻居”的属性来限制或者允许用户连接到共享资源，而在 Linux 操作系统中的 NFS（Network File System，网络文件系统）同样具有此类功能，它也可以为用户提供类似“网上邻居”的共享服务。接下来本章将具体讲解搭建 NFS 服务器的相关知识。

9.1　NFS 服务简介

NFS 是一个流行的基于 TCP/IP 网络的文件共享协议，最早由 Sun 公司所开发，开发的目的在于，让不同计算机的不同操作系统之间可以彼此共享文件。NFS 开始是 UNIX 操作系统间共享文件和操作系统的一种方法，后来被 Linux 操作系统完美继承，并成为文件服务的一种标准（RFC1904、RFC1813 和 RFC3010）。

9.1.1　使用 NFS 的优势

NFS 最大的功能就是可以透过网络，让不同的机器可以相互分享资源。使用 NFS 可以让磁盘空间得到较高的使用率，让本地工作站使用更少的磁盘空间，这是因为可以把常用数据存放在一台计算机，然后通过网络来访问。

使用 NFS 可以减少整个网络上的可移动介质设备量（例如 CD-ROM、DVD-ROM 等存储设备可以在网络上被其他机器使用）。

另外，使用 NFS 后，用户不必在每个网络上的机器里头都设置一个 home 目录，home 目录可以存放在 NFS 服务器上，可供其他机器访问使用。具体说明如下。

例如，当 Linux 计算机每次启动时就自动挂在 Server1 上的/export/nfs 分区，该分区为共享分区，该计算机上的共享分区被共享到每个用户的家目录中，如图 9-1 所示。具体指令如下。

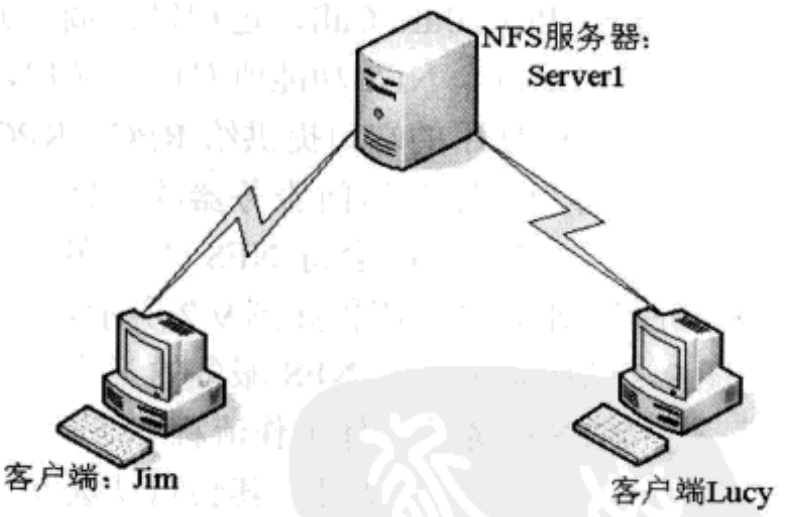


图 9-1　客户端加载服务器上的共享目录到本地

```
[Jim@localhost ~]$mount Server1:/export/nfs /home/Jim/nfs
[Lucy@localhost ~]$mount Server1:/export/nfs /home/Lucy/nfs1
```

因此，该计算机上的 Jim 和 Lucy 这两个用户都可以把/home/用户名/nfs 下当作本地硬盘，从而不用考虑网络访问问题。

9.1.2 NFS 服务的组件

NFS 是一个复杂的组件，Linux 下的 NFS 服务主要由 rpc.nfsd 进程、rpc.mounted 进程两个必需进程和三个可选进程共五部分组成，具体说明如下。

rpc.nfsd 进程：该进程为必需进程。该守护进程主要用于对客户端登录主机的权限进行验证，同时负责处理 NFS 请求。

rpc.mounted 进程：该进程为必需进程。该守护进程主要用于对 NFS 的文件系统的管理。当客户端成功登录主机后，rpc.mounted 进程会根据/etc/exports 文件来检查客户端是否有访问 NFS 服务器上的资源的权限。

rpc.quotad 进程：该进程为可选进程。该守护进程提供了 NFS 和配额管理程序之间的接口。

rpc.locked 进程：该进程为可选进程。该守护进程可以帮助 rpc.stated 守护进程使用本进程来处理崩溃系统的锁定恢复问题。锁定文件在于可以避免多个用户同时操作一个文件时发生的问题。

rpc.statd 进程：该进程为可选进程。该守护进程负责处理客户端与服务器之间的文件锁定问题以确保文件的一致性（与 rpc.locked 有关）。

9.2 RPC 简介

NFS 不同于其他网络服务，绝大部分的网络服务都有固定的端口，而 NFS 却没有。这是因为 NFS 是一个复杂的组件，其支持文件传输、身份验证等相当多的功能，每种功能都会占用一个端口。为了避免 NFS 服务占用过多的固定端口，因此 NFS 会采用动态端口的方式来工作（即每个功能提供服务时都会随机取用一个小于 1024 的端口来提供服务）。

但 NFS 提供的端口众多，因此客户端将不确定访问哪个端口才能获得 NFS 提供的服务，RPC（Remote Procedure Call，远程进程调用）服务的出现便解决了此问题。RPC 服务最主要的功能就是记录每个 NFS 功能所对应的端口，NFS 启动的时候，会自动向 RPC 服务器注册，把自己各个功能使用的端口提供给 RPC。RPC 工作于固定端口 111，当客户端向 NFS 服务发出请求时，客户端则会访问服务器的 111 端口（RPC），RPC 此时会将 NFS 工作端口返回给客户端，具体流程如图 9-2 所示。

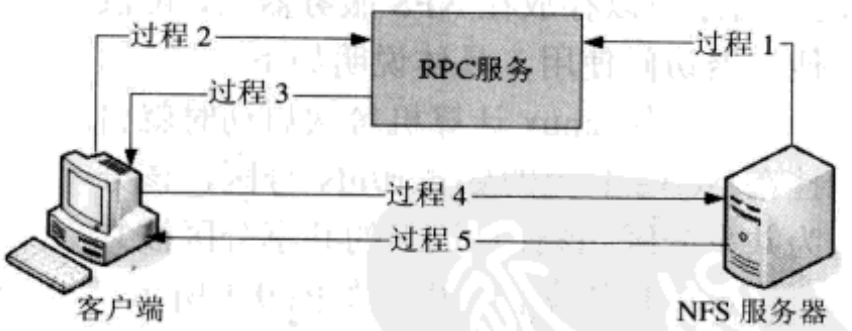


图 9-2 NFS 和 RPC 一起为客户端提供服务

图 9-2 清晰地显示了 NFS 服务的工作流程，常规 NFS 服务的具体工作流程如下。

过程 1，当 NFS 启动时，其会自动选择工作端口 1011（小于 1024），同时注册于 RPC（工作于 111 端口），RPC 做下记录。

过程 2，当客户端需要 NFS 提供服务时，其会向工作于 111 端口的 RPC 查询 NFS 提供某个具体功能的具体端口。

过程 3，此时 RPC 对客户端提出的请求作出响应，告诉其此次 NFS 工作于 1011 端口。

过程 4，客户端得到响应后，将直接访问 NFS 服务器的 1011 端口，请求服务。

过程 5，经过权限认证后的 NFS 服务将允许客户端对其进行数据访问。

除此之外，由于 NFS 需要向 RPC 服务器注册，因此为了使 NFS 服务正常工作，RPC 服务一定要优先于 NFS 服务启用。重新启动 RPC 服务后，NFS 服务也需要重启。

9.3 安装并配置 NFS 服务

使用 NFS 服务的前提是确认系统中已经安装 NFS 服务组件，本节将具体讲解 NFS 服务的安装与配置。

9.3.1 安装 NFS 服务

在不确定是否安装了 NFS 服务的情况下，可以先检查电脑中是否已经安装了 NFS 支持套件，在 Red Hat Enterprise Linux 5 中，要启用 NFS 服务，则必须具有 portmap 和 nfs-utils 两个套件，具体说明如下。

- portmap 套件

该套件借助 RPC 服务的帮助，负责端口映射工作以保证 NFS 服务的正常运行。

- nfs-utils 套件

该套件为 NFS 服务的主要套件，提供 rpc.nfsd 和 rpc.mounted 这两个守护进程与其他相关文档、执行文件的套件。

在明确所需套件后，即可使用以下命令来查询是否安装了这两个套件。

```
[root@localhost ~]#rpm -qa |grep nfs-utils
[root@localhost ~]#rpm -qa |grep portmap
```

执行命令后的结果如下所示，显示了相应组件的版本信息，表明组件已经正确安装。

```
[root@localhost ~]# rpm -qa |grep nfs-utils
nfs-utils-lib-1.0.8-7.2.z2
nfs-utils-1.0.9-33.el5
[root@localhost ~]# rpm -qa |grep portmap
portmap-4.0-65.2.2.1
```

除此之外，为了避免在以后使用过程中出现错误，建议查询一下 NFS 的各个程序是否在正常运行，执行以下命令。

```
[root@localhost ~]#rpcinfo -p
```

执行命令后的结果如下所示，没有显示 nfs 和 mounted 选项，则说明 nfs 没有运行。

```
[root@localhost ~]# rpcinfo -p
   程序 版本 协议   端口
    100000    2   tcp    111  portmapper
    100000    2   udp    111  portmapper
    100024    1   udp    832  status
    100024    1   tcp    835  status
```

如果检测到系统没有安装任何 NFS 组件，此时需要通过 Red Hat Enterprise Linux 5 安装光盘或者 NFS 官方网站（http://sourceforge.net/projects/nfs）获得最近的 NFS 组件 RPM 安装包，

如图 9-3 所示。

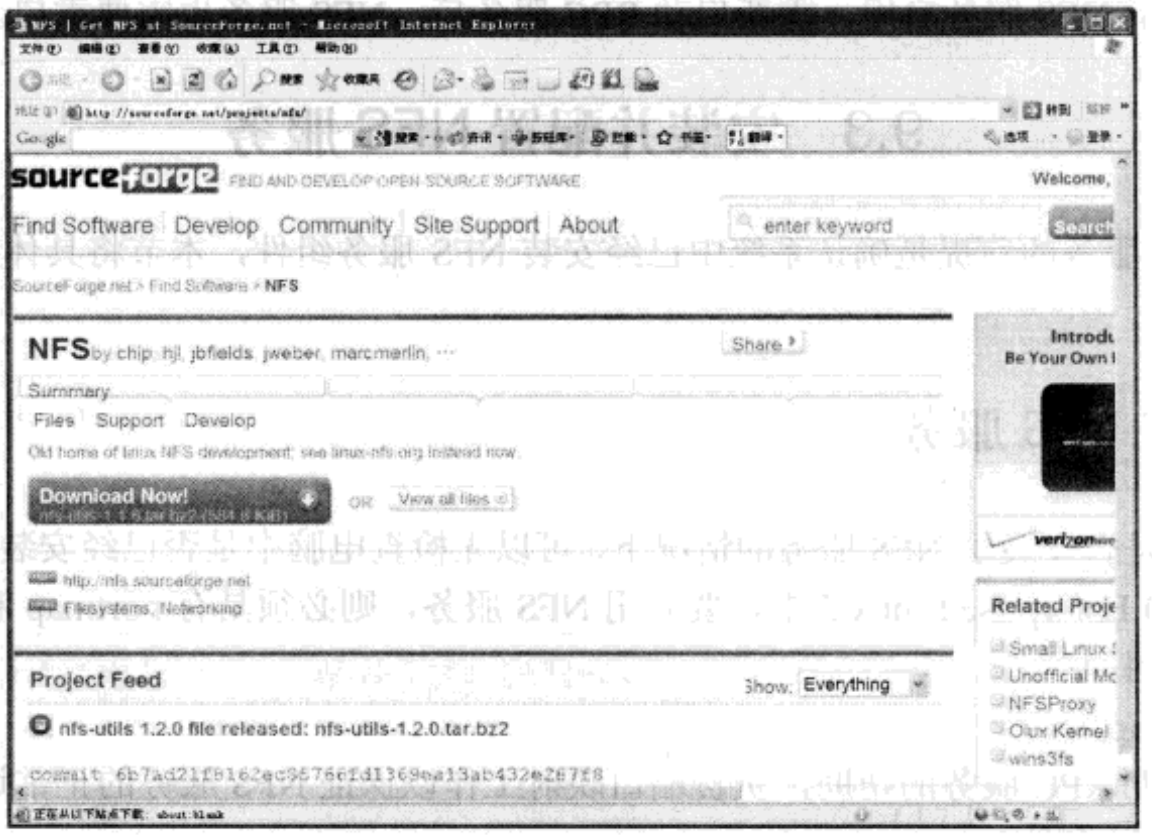


图 9-3　下载 NFS 组件

获得安装包之后可以通过执行如下指令进行安装。

```
[root@localhost ~]# rpm -ivh nfs-utils-1.0.9-40.el5.i386.rpm
```

9.3.2 配置 NFS 服务

创建配置文件（/etc/exports，/etc/hosts.allow 和/etc/host.deny）并配置文件中的各选项是配置 NFS 服务器的主要工作，接下来将对各个文件进行具体的介绍。

1. 创建/etc/exports 文件的文件格式

该文件定义了服务器上的某几个部分与网络中的其他计算机共享，以及共享的规则等，在 Red Hat Enterprise Linux 5 中没有主动提供/etc/exports 文件，此时就需要用户自己手工创建。该配置文件由很多基本条目构成，每个基本条目采用如下格式。

```
<目录> <主机 1(选项 1,选项 2)> <主机 2(选项 1,选项 2)> [主机 n…]
```

<目录>：描述需要共享的目录的全路径，当然该目录下的所有与当前指定目录同属一个文件系统的子目录及文件全部会被共享。

<主机>：允许访问共享目录的客户端主机。指定客户端主机的 IP 地址或客户端主机的 DNS 名称。

<选项>：对于许可的客户端主机访问服务端主机共享目录时的限制参数。

- rw　read-write，可读、写权限。
- ro　read-only，只读权限。
- sync　数据同步写入到内存与硬盘当中。

- async　数据会先暂存于内存当中，而非直接写入硬盘。
- no_root_squash　登录 NFS 主机使用共享目录的用户，如果是 root 的话，那么对于这个共享的目录来说，他就具有 root 的权限。这个设置“极不安全”，不建议使用。
- root_squash　在登录 NFS 主机使用共享之目录的用户如果是 root 时，那么这个用户的权限将被压缩成为匿名用户，通常他的 UID 与 GID 都会变成 nobody（nfsnobody）的系统账号的身份。
- all_squash　不论登录 NFS 的用户身份是什么，他的身份都会被压缩成匿名用户，通常也就是 nobody（nfsnobody）。
- anonuid　anon 意指 anonymous（匿名者），前面关于*_squash 提到的匿名用户的 UID 设定值，通常为 nobody（nfsnobody），但是可以自行设定这个 UID 的值。当然，这个 UID 必须存在于/etc/passwd 当中。
- Anongid　同 anonuid，但是变成 Group ID。

创建该文件的具体指令如下。

```
[root@localhost ~]#vi /etc/exports
/tmp          192.168.0.8/24(ro)      localhost(rw)        * (ro,sync)
# [共享目录] [第一台主机(权限)]    [可用主机名]      [其他主机（可用统配符）]
```

该方法表明/tmp 被分别共享给 3 个不同的主机或域。其中，“#”号表示批注。主机后面用小括号“()”来设置权限参数，当权限参数有多个时，需用逗号“,”分开，且主机名须与小括号连在一起。

注意　如果仅仅改变了/etc/exports 文件的内容，则无需重启 NFS 服务，只需执行如下指令让/etc/exports 生效即可。

```
[root@localhost ~]# exportfs -arv
```

其中，exportfs 命令的基本格式如下。

```
[root@localhost ~]# exportfs [-aruv]
```

-a 表示输出在配置文件/etc/exports 中设置的所有目录；-r 表示重新读取/etc/exports 文件中的设置，并使设置立即生效，但不需要重启 NFS 服务；-u 表示停止某一共享目录；-v 表示回显所设置的共享目录。

---

2. 设置主机名的方式

主机名称的设定主要有以下两种方式。

- 使用主机名称，该主机名称要在/etc/hosts 内或使用 DNS 找到。在使用主机名称的情况下，可以支持通配符（如*或?）。
- 使用完整的 IP 地址或网段。

3. /etc/hosts. allow 和/etc/host. deny

hosts.deny 和 hosts.allow 文件存在于/etc 目录下，通过配置这两个文件，可以指定哪些机

器可以使用 NFS 服务，哪些不可以使用 NFS 服务，实现的方式如下所述。

当服务请求到达服务器时，TCP wrapper 就按照下列顺序查询这两个文件，直到遇到一个匹配项为止。

当在/etc/hosts.allow 里面有一项与请求服务的主机地址项匹配，那么就允许该主机获取该服务。否则，如果在/etc/hosts.deny 里面有一项与请求服务的主机地址项匹配，就禁止该主机使用该项服务。

如果相应的配置文件不存在，那么访问控制软件就会认定当前的访问控制配置是一个空文件，这说明用户可以通过删除或者移走配置文件实现清除所有访问控制配置功能。

配置上述两个配置文件的基本格式如下。

程序名列表：主机名/IP 地址列表

主机名/IP 地址列表指定允许或者禁止使用该服务的一个或者多个主机的标识，主机名之间用逗号或空格分隔。程序名和主机地址都可以使用通配符。

注意　在 Linux 操作系统中常采用如下通配方式指定进程或者主机名列表。

- ALL 表示匹配所有可能性。
- EXPECT 表示除去后面所定义的主机。
- LOCAL 表示匹配所有主机名中不包含“.”的主机。
- 以“.”起始的域名串。
- 以“.”结尾的 IP 串。
- 格式“a.a.a.a/n.n.n.n”表示“网络/掩码”，如果请求服务的主机的 IP 地址与掩码的位与结果等于 a.a.a.a 那么该主机与该项目匹配。

另外，在所有的配置完成后，推荐执行 tcpdchk 命令。tcpdchk 命令是检查 TCP_WAPPERS 配置的程序，用于检查 TCP_WAPPERS 的配置，并报告它发现的问题或潜在的问题。

```
[root@localhost ~]# tcpdchk
```

9.3.3 NFS 文件的存取权限

当客户端访问 NFS 服务时，NFS 服务器需要通过一些标准来识别用户，这是因为 NFS 服务本身并不具备用户身份验证功能，这些标准主要有以下 3 种。

- root 账号

基于安全因素的考虑，如果客户端用 root 账号去访问 NFS 服务器资源，服务器会主动将客户端改成匿名用户。因此，root 账号只能访问服务器上的匿名资源。

- NFS 服务器上有客户端的账号

客户端是根据用户和组（UID、GID）来访问 NFS 服务器资源的，如果 NFS 服务器上存在对应的用户名和组，用户就访问与客户端同名的资源。

- NFS 服务器上没有客户端账号

客户端只能访问匿名资源。

9.4 启动与停止 NFS 服务

Red Hat Enterprise Linux 5 中内置的 NFS 程序遵循一般的服务启动、停止规范，与其他服务的启动、停止操作相同。但由于该服务依赖于 portmap 服务，因此在启动 NFS 服务之前需要先启动 portmap 服务。

要启动 NFS 服务，则需先启动 portmap 服务，执行如下命令。

```
[root@localhost ~]#service portmap start
```

启动 portmap 服务后，可以通过执行以下命令来启动 NFS 服务。

```
[root@localhost ~]#service nfs start
```

执行命令后的结果如下所示，表明启动成功。

```
[root@localhost ~]# service portmap start
启动 portmap:                                              [确定]
[root@localhost ~]# service nfs start
启动 NFS 服务:                                             [确定]
关掉 NFS 配额:                                             [确定]
启动 NFS 守护进程:                                         [确定]
启动 NFS mountd:                                           [确定]
```

停止 NFS 服务的操作类似于 NFS 服务的启动操作，但在停止或重启 NFS 服务的时候，portmap 服务可以不停止。要停止该服务，可以通过执行如下指令实现。

```
[root@localhost ~]#service nfs stop
```

执行命令后的结果如下所示，表明停止 NFS 服务成功。

```
[root@localhost ~]# service nfs stop
关闭 NFS mountd:                                           [确定]
关闭 NFS 守护进程:                                         [确定]
关闭 NFS 服务:                                             [确定]
```

重启 NFS 服务的操作同样类似于 NFS 服务的启动操作，要重启该服务，可以通过执行以下命令实现。

```
[root@localhost ~]#service nfs restart
```

执行命令后的结果如下所示。

```
[root@localhost ~]# service nfs restart
关闭 NFS mountd:                                           [确定]
关闭 NFS 守护进程:                                         [确定]
关闭 NFS 服务:                                             [确定]
启动 NFS 服务:                                             [确定]
关掉 NFS 配额:                                             [确定]
启动 NFS 守护进程:                                         [确定]
启动 NFS mountd:                                           [确定]
```

通过此种方式启动的 NFS 服务只能运行到此次计算机关闭之前，下一次系统重新启动后同样需要重新启动 NFS 服务，显得比较麻烦。为了简便该操作，可以设置 NFS 服务随系统的启动而自动运行，此种功能的具体实现方法如下。

首先在桌面上单击鼠标右键，选择“打开终端”命令，然后在打开的终端窗口中输入“ntsysv”命令，此时将会打开 Red Hat Enterprise Linux 5 中的“服务”配置小程序，找到“nfs”服务，并在其前面加个“*”，最后单击“确定”按钮，如图 9-4 所示。至此，完成了 NFS 服务随系统启动而自动运行的设置。

另外，还可以通过执行命令来检查 PortMap 和 NFS 服务的运行状态，具体指令如下。

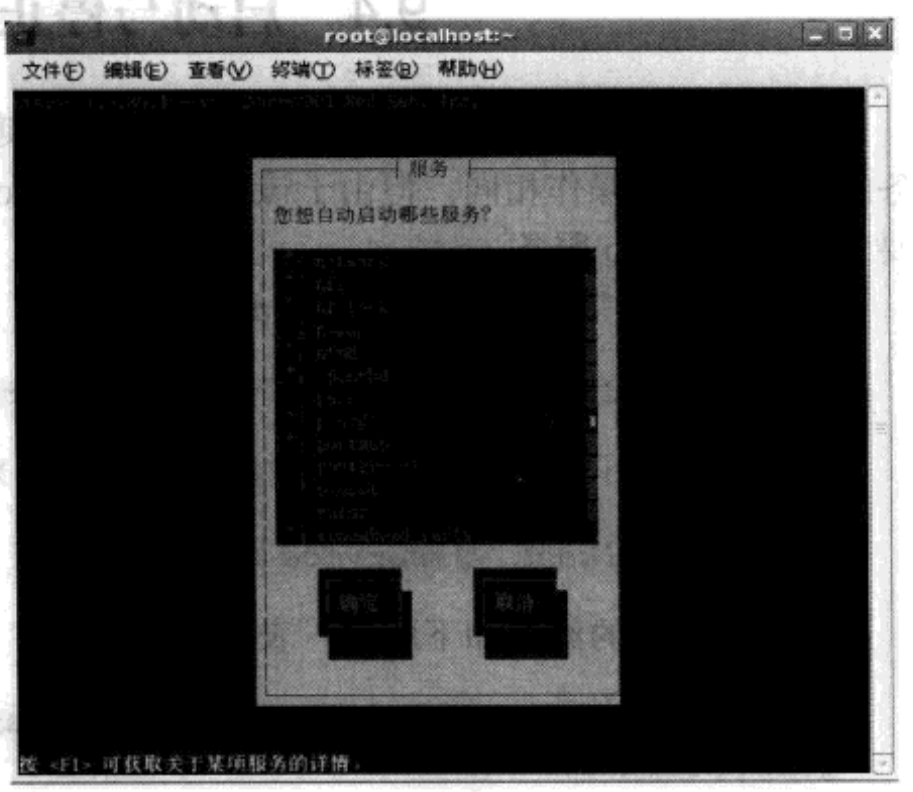


图 9-4　设置 NFS 服务自动运行

```
[root@localhost ~]# service portmap status
[root@localhost ~]# service nfs status
```

执行命令后的结果如下所示。

```
[root@localhost ~]# service portmap status
portmap (pid 4435) 正在运行...
[root@localhost ~]# service nfs status
rpc.mountd (pid 16681) 正在运行...
nfsd (pid 16678 16677 16676 16675 16674 16673 16672 16671) 正在运行...
```

9.5　在客户端挂载 NFS 文件系统

通过以上小节的学习，相信大家都已经掌握了 NFS 服务的安装与配置方法，接下来本节将具体介绍如何在 Linux 操作系统中查看、连接、卸载、使用 NFS 服务器上的共享资源。

9.5.1　查看 NFS 服务器信息

在 Red Hat Enterprise Linux 5 中如果需要查看 NFS 服务器上的共享资源，则可以通过 showmount 指令实现。

showmount 指令的语法格式如下。

```
[root@localhost ~]#showmount [-adehv] [ServerName]
```

具体参数及说明如表 9-1 所示。

表 9-1 showmount 指令参数及描述

参　数	说　明
-a	查看服务器上的输出目录和所有连接客户端信息。显示格式为“host:dir”
-d	只显示被客户端使用的输出目录信息
-e	显示服务器上所有的输出目录（共享资源）

实例 当需要查看 IP 地址为 192.168.0.16 的服务器上的 NFS 共享资源信息时，需要执行以下指令。

```
[root@localhost ~]#showmount -e 192.168.0.16
```

执行命令后的结果如下所示。

```
[root@localhost ~]#showmount -e 192.168.0.16
Expot list for 192.168.0.16:
/exports/nfs    *
/exports/tmp    *
/exports/tmp/rhel  192.168.0.0/24
/exports/user_linux 192.168.0.12
/exports/public (everyone)
```

注意 默认情况下，SELinux 的存在会造成客户端访问服务器有问题，为了避免此种情况的发生，需要执行以下指令在服务器端进行简单的配置。

```
setsebool -P portmap_disable_trans=1
setsebool -P nfs_export_all_pro=1
```

9.5.2 加载 NFS 服务器共享目录

在 Red Hat Enterprise Linux 5 中加载 NFS 服务器上的共享目录，可以通过使用 mount 命令实现。

具体命令如下。

```
[root@localhost ~]#mount -t NFS
```

实例 如果要加载 IP 为 192.168.0.16 的服务器上的/exports/rhel 目录，则需要执行以下步骤。

第 1 步，创建一个本地目录来加载 NFS 服务器上的输出目录，使用以下命令。

```
[root@localhost ~]#mkdir /mnt/nfs
```

第 2 步，使用相应的 mount 指令加载服务器目录。

```
[root@localhost ~]#mount -t nfs 192.168.0.16:/exports/rhel /mnt/nfs
```

此时加载服务器目录已完成。

而如果要卸载刚才加载的 NFS 共享目录，执行以下指令即可。

```
[root@localhost ~]#umount /mnt/nfs
```

9.5.3 自动挂接 NFS 文件系统

在 Red Hat Enterprise Linux 5 中，自动加载文件系统是在/etc/fstab 中定义的，NFS 文件系统支持自动挂载。

1. 启动时自动挂接 NFS

启动时自动挂接 NFS 的具体步骤如下。

第 1 步，用文本编辑器打开/etc/fstab 文件，执行如下命令。

```
[root@localhost ~]#vi /etc/fstab
```

在其中添加如下行。

```
192.168.0.16:/exports/rhel /mnt/nfs    nfs defaults    0 0
```

第 2 步，执行以下指令重新加载 fstab 文件中定义的文件系统来使设置生效。

```
[root@localhost ~]#mount -a
```

2. 使用时自动挂接 NFS

如果客户端需要使用 NFS 服务器所提供的共享资源，则自然会想到在 Linux 启动时自动挂接和使用 mount 指令手动挂接两种方法。其实，Red Hat Enterprise Linux 5 内置的 autofs 还提供了第 3 种方法：让用户使用共享资源时自动挂接。

autofs 能自动挂接的文件系统类型很多，包括 smbfs、c 等。它是这样实现让用户使用共享资源时自动挂接的：autof 预先定义一些挂接资源，当客户端访问这些资源时才自动挂接，在客户端不使用相应资源超过默认设置时间（5 分钟）以后，就会自动卸载。

注意 Red Hat Enterprise Linux 5 中 autofs 的主配置文件是/etc/auto.master。

在 Red Hat Enterprise Linux 5 中配置自动加载的语法格式如下。

```
MountPiont map-name
```

其中，MountPoint 表示指定 autofs 的挂接点；Map-name 表示设置挂接点文件，里面包含了所有自动挂接到 MountPoint 下的资源列表。

第 10 章　LDAP 目录服务

在当今的信息世界，网络为人们提供了丰富的资源。随着网络资源的日益丰富，迫切需要一种能有效管理各种资源信息并利于检索查询的服务技术，目录服务技术随之产生。目录服务技术就是一种能适应网络信息飞速发展的管理资源信息的技术，因此，本章将从 LDAP 服务器的基本概念入手，具体讲解其安装、配置以及应用。

10.1　目录服务概述

目录服务（Directory Services）可以有效地解决众多网络服务的用户账号问题。目录服务是一种基于客户/服务器模型的信息查询服务，是一个特定的数据库。

目录服务规定了统一的身份信息数据库、身份认证机制和接口，实现了资源和信息的统一管理，保证了数据的一致性和完整性。目录服务具有比关系型数据库更擅长查询的特点，并具有良好的检索功能。目录数据库中的数据读取和查询效率非常高，比关系型数据库差不多快一个数量级。目录服务适用于基于目录和层次结构的信息管理，可以有效地管理企业网络中的用户和各种资源，在降低了成本的同时网络的安全性也得到了保障。很多的新型技术通过目录服务技术实现了信息存储、管理和查询。但是目录服务的数据写入效率较低，适用于用户信息的数据不需要经常改动的系统（如电子邮件系统）。

目录服务是以树状的层次结构来描述数据信息的，此种模型适应了众多行业应用的业务组织结构（如企业单位的人员和资源的组织方式等）。目前目录服务有 X.500 和 LDAP 两个国际标准协议，接下来将具体介绍这两个国际标准协议。

10.1.1　X.500 简介

X.500 是由 ITU-T（International Telecommunication Union-Telecommunication Sector，国际电信联盟-电信标准局）和 ISO（International Organization for Standardization，国际标准化组织）定义的目录标准，由一个协议簇组成。

1.　X.500 协议簇

X.500 协议簇包括了从 X.501 到 X.525 的目录服务。其中，X.501 模型强调目录服务的基本模型和概念；X.509 认证框架是在 X.500 中处理目录客户和服务器的认证；X.511 抽象服务定义 X.500 被要求提供的功能性服务；X.518 分布式操作过程表明如何跨越多台服务器处理目录服务；X.519 协议规范即是 X.500 协议，包括目录访问协议 DAP（Directory Access Protocol）、目录系统协议 DSP（Directory System Protocol）、目录操作绑定协议 DOP（Directory Operator Protocol）和目录信息 Shadowing 协议 DISP（Directory Information Shadowing Protocol）；X.520 选定的属性类型要求是 X.500 自己使用的属性类型；X.521 选定的对象类型为 X.500 自己使用

的对象类；X.525 复制是如何在目录服务器之间复制目录内容。

2. X.500 标准包含的内容

X.500 标准中主要定义有多种内容，具体如表 10-1 所示。

表 10-1 X.500 标准中的主要定义

定义内容	简要说明
信息模型	确定目录中信息的格式与字符集以及如何在项中表示目录信息
命名空间	全球统一的名字空间。确定对信息进行的组织和引用，如何组织和命名项——目录信息树 DIT 和层次命名模型
功能模型	确定可以在信息上执行的操作
认证框架	保证目录中信息的安全，访问控制模型实现目录中信息的授权保护
分布操作模型	确定数据如何进行分布及如何对分布数据执行操作，如何将全局目录树划分为管理域进行管理
目录管理模型	用于服务端通信的目录访问 DAP 协议，将用户请求进行连接所需要的目录系统协议 DSP，以及选定的信息进行复制所需要的目录信息映像协议 DISP，用于自动在服务器之间协商连接配置的目录操作绑定协议 DOP

3. X.500 的优缺点

无可厚非，X.500 具有显著的特性，但同时也有不足的地方，下面就 X.500 的优、缺点进行说明。

（1） X.500 优点

X.500 目录服务被公认为是实现一个目录服务的最好途径，其具有以下一些优点。

- 强大的搜索功能

X.500 具有强大的搜索功能，支持用户建立的任意的复杂查询。

- 结构化信息结构

X.500 目录中定义了信息结构，允许进行本地扩展。

- 分散维护

运行 X.500 的每个站点只负责其本地目录部分，所以可以立即进行更新和维护操作。

- 单一全局命名空间

类似于 DNS，X.500 为用户提供单一命名空间。与 DNS 相比，X.500 的命名空间更加灵活并且更容易扩展。

（2） X.500 的缺点

虽然 X.500 具有如此多的优点，但实现起来投资过大，且效率不高。

由于目录访问协议 DAP 这种应用层协议是严格遵照复杂的 ISO 七层协议模型制定的，对相关层协议环境要求过多，主要运行在 UNIX 机器上，在许多小系统上无法使用，也不适应 TCP/IP 协议体系。因此，使得这种协议越来越不适应需要，此时更适应需求的 LDAP 应运而生，接下来将对 LDAP 进行具体的讲解。

10.1.2 初识 LDAP

LDAP（Lightweight Directory Access Protocol，轻量级目录访问协议）是基于 TCP/IP 协议的目录访问协议，是 Internet 上目录服务的通用访问协议。LDAP 的出现简化了 X.500 目录的

复杂度，降低了开发成本，是 X.500 标准中的目录访问协议 DAP 的一个子集，同时也作为 IETF（Internet Engineering Task Force）的一个正式标准。

1. LDAP 协议的产生

1993 年 LDAP 协议通过批准，产生了 LDAP V1 版本，1997 年发布了 LDAP V3 版本。LDAP V3 是 LDAP 协议发展的一个重要标志，其使 LDAP 协议作为 X.500 的简化版的同时还提供了 LDAP 协议许多自有的特性，这使得 LDAP 协议的功能变得更为强大，更为完善。

2. LDAP V3 协议

LDAP V3 协议可以说是一个协议簇。

- RFC 2251

LDAP V3 核心协议，定义了 LDAP V3 协议的基本模型和基本操作。

- RFC 2252

对 LDAP V3 中的语法、匹配规则、属性类型和对象类等基本数据模式（Schema）以及标准的系统数据模式进行了定义。

- RFC 2253

定义了 LDAP V3 中的分辨名 DN（Distinguished Name）表达方式。

- RFC 2254

对 LDAP V3 中的过滤器的表达方式进行了定义。

- RFC 2255

LDAP 统一资源地址的格式。

- RFC 2256

在 LDAP V3 中使用 X.500 的 Schema 列表。

- RFC 2829

为 LDAP V3 中的认证方式进行了定义。

- RFC 2830

为如何通过扩展使用 TLS 服务进行了定义。

- RFC 1823

为 C 的 LDAP 客户端 API 开发接口进行了定义。

- RFC2798

定义了一个名为 InetOrgPerson 的 LDAP 对象类以及一组该对象类可用的属性 。

- RFC 2847

定义了 LDAP 数据导入、导出的文件接口 LDIF。

3. 协议内容

这些协议主要定义了信息模型、命名空间、功能模型、安全框架、分布式操作模型和 LDAP 扩展框架（这与 X.500 标准类似），具体说明如表 10-2 所示。

表 10-2 协议中的主要定义

定义内容	简要说明
信息模型	确定 LDAP 目录中信息的格式和字符集，如何表示目录信息（定义对象类、属性、匹配规则和语法等模式）
命名空间	确定对信息进行的组织方式——目录信息树 DIT，以 DN 和 RDN 为基础的命名方式，以及 LDAP 信息的 Internet 表示方式
功能模型	确定可以在信息上执行的操作的通信协议以及在客户端进行这些操作的 API 接口
安全框架	保证目录中信息的安全，匿名、用户名/密码、SASL 等多种认证方式，以及与 TLS 结合的通信保护框架
分布式操作模型	基于 Referral 方式的分布式操作框架
LDAP 扩展框架	基于控制和扩展操作的 LDAP 扩展框架

4. LDAP 的存储方式

LDAP 的信息是以树型结构存储的，具体信息存储在条目的数据结构中。一个目录信息树由若干条目组成，一个条目是一个对象，每个条目具有唯一的标识名 DN，并由多个属性组成，每个属性可以对应一个或多个值。

LDAP 目录服务器是通过目录数据库来存储网络信息以提供目录服务的。为了方便用户迅速查找和定位信息，目录信息树及其相关概念构成了 LDAP 协议的信息模型，一个目录树如图 10-1 所示。

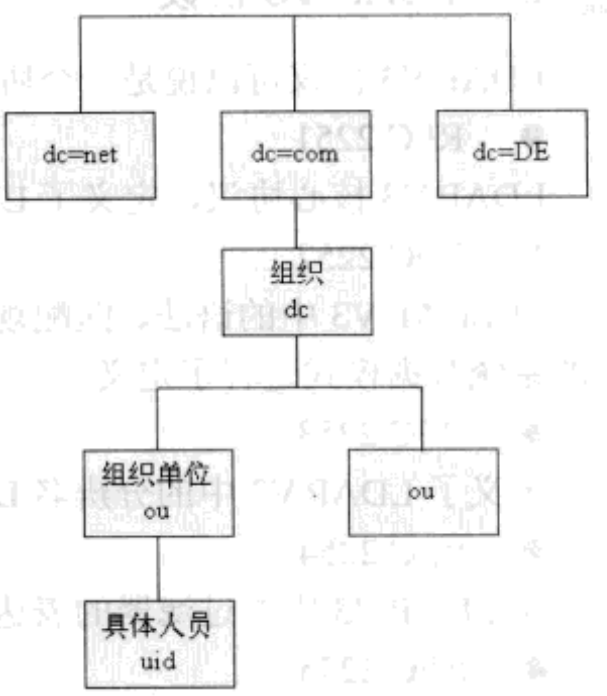


图 10-1 信息目录树

注意 图中出现的“dc”、“ou”、“uid”等关键字，代表了不同的含义。其中，关键字 dc（domain component）表示域名的部分，其格式是将完整的域名分为几部分；关键字 uid（user id）表示用户 ID；关键字 ou（organization unit）表示组织单位，类似于 Linux 文件系统中的子目录，是一个容器对象。另外，关键字 cn（common name）表示公共名称；关键字 sn（surname）表示姓；关键字 dn（distinguished name）表示唯一辨别名，类似于 Linux 文件系统中的绝对路径，每个对象都有一个唯一的名称；关键字 rdn（relative dn）表示相对辨别名，类似于相对路径，与目录树结构无关；关键字 c（country）表示国家；关键字 o（organization）表示组织名。

5. LDAP 的特点

LDAP 之所以得到了广泛的支持，并正被越来越多的系统作为标准的功能所集成，是因为其具有以下优点。

- LDAP 是一个标准的协议，并具有跨平台的特点。
- LDAP 是一个安全的协议，LDAPv3 支持 SASL、SSL 和 TLS，使用认证来确保事务的安全性，另外，LDAP 提供了不同层次的访问控制，以限制不同用户的访问权限。
- 安装简单，易于优化和维护。
- 支持二进制图片等异类数据存储。

6. LDAP 协议的发展方向

LDAP 标准正向访问控制模型、复制协议（DUP）以及扩展操作方向发展，这些扩展操作包括查询的排序和分页、动态目录、语言标签、LDAP 服务发现等。

10.1.3 X.500 与 LDAP

从目录服务技术的发展来看，X.500 标准是 LDAP 标准的原始版本，除了一些扩展操作外，X.500 支持全部 LDAP 操作。下面对这两个标准进行比较。

1. LDAP 与 X.500 的相同之处

X.500 与 LDAP 都是用来从分层目录中提取信息的。LDAP 是 X.500 的一个简化版本，最初 LDAP 只是 X.500 的功能子集，随着 LDAP 的不断发展，便增加了许多自己独立的新特性。如今的 LDAP 既可以为 X.500 目录服务提供一个轻型前端，也可以实现一个独立的目录服务。

这两个目录服务技术标准有以下一些相似点。

- 在信息模型上

都使用了项、对象类、属性等概念和模式来描述信息；在命名空间方面，都使用了目录信息树结构和层次命名模型。

- 在功能模型上

都使用了相似的操作命令来管理目录信息；在认证框架方面，都可以实现用户名和密码的认证，或者基于安全加密方式的认证机制。

- 在平台上

都实现了一个通用的平台结构，提供了一个操作系统和应用程序需要的信息服务类型，可以被许多平台和应用程序接收和实现。

- 在分布性方面

目录信息都可以分布在多个目录服务器中，这些服务器可以由各组织管理，既保证了目录信息总体结构的一致性，又满足了分级管理的需要。

- 在灵活性上

目录的规模可大可小，大到全球目录树，小到只有一台目录服务器。

2. LDAP 的优越之处

LDAP 的独特之处主要有以下几点。

- AP（Access Protocol）既是一个 X.500 的访问协议，又是一个灵活的可以独立实现的目录系统。
- LDAP 不需要经过 OSI 会话层和表示层，直接运行在 TCP/IP 或其他可靠的传输协议层上。这样就避免了在 OSI 会话和表示层的开销，使连接的建立和包的处理更简单，速度更快捷。而 X.500 基于 OSI（Open System Interconnection，开放式系统互连）协议，其建立在应用层上，需要在 OSI 会话层和表示层上进行许多的建立连接和包处理的任务。

- LDAP 是通过引用机制来实现分布式访问的，通过客户端 API 实现分布式操作并平衡了负载。X.500 DSA 通过服务器之间的链操作来实现分布式的访问，这样查询的压力集中便都集中在服务器端。
- LDAP 协议不仅继承了 X.500 最好的特性，还省去了 X.500 中深奥难懂的和较少使用的服务控制和安全特性。
- LDAP 具有投资少、易配置和易管理的特点。

10.1.4 流行的目录服务产品

目前大部分目录服务器都支持 LDAP，LDAP 服务器软件有很多，下面是几个目前流行的目录服务产品。

1. NDS（Novell Directory Services）

NDS 是 Netware4 目录服务中的对象，是随着 Netware4 一起发布的较早的面向企业网络的目录服务产品。该产品有效地将网络系统的各种资源组织到一起，也将各种应用软件集成到同一个资源管理平台上，同时也支持各种规模的网络环境。该软件的可靠性较高，也支持多个目录服务器之间的复制。随着发展，NDS 从系统软件中分离出来，用以支持不同的系统平台，便出现了产品 Novell eDirectory，如图 10-2 所示。



图 10-2 Novell 的 eDirectory

Novell eDirectory 是一种支持 LDAP，基于目录的身份管理系统。Novell eDirectory 对用户身份、访问特权和其他网络资源实行集中管理。由于 eDirectory 支持动态组功能，能够持久搜索和实时连续备份，因此大大降低了管理成本。eDirectory 卓越的安全功能包括通过安全套接层（SSL）加密的口令、“Novell 国际密码基础结构”、RSA 非对称密钥加密、安全性鉴定服务、

智能卡和 X.509 v3 数字证书。管理员可以为指定用户针对某些特定资源授予特定的访问权限。

Novell 的目录服务（NDS）是根据 X.500 的目录服务标准设计的。NDS 采用树形结构建立一个分级的以对象及其属性为基础的目录结构，其把目录树细分为组织单元（OU），该组织单元是一个逻辑单元，该组织单元是树中的一些资源对象的集合。

2. OpenLDAP

OpenLDAP 是一款非常出色的开放源码的 LDAP 服务器软件，该软件是在 Internet 环境下进行集体开发的项目产品，如图 10-3 所示。由于 OpenLDAP 是开放源码的软件产品，在 UNIX 和 Linux 平台上都得到了很好的支持，并迅速地发展。同时，OpenLDAP 也逐步在向其他系统平台（如 Windows 平台）移植，以满足用户的不同需求。

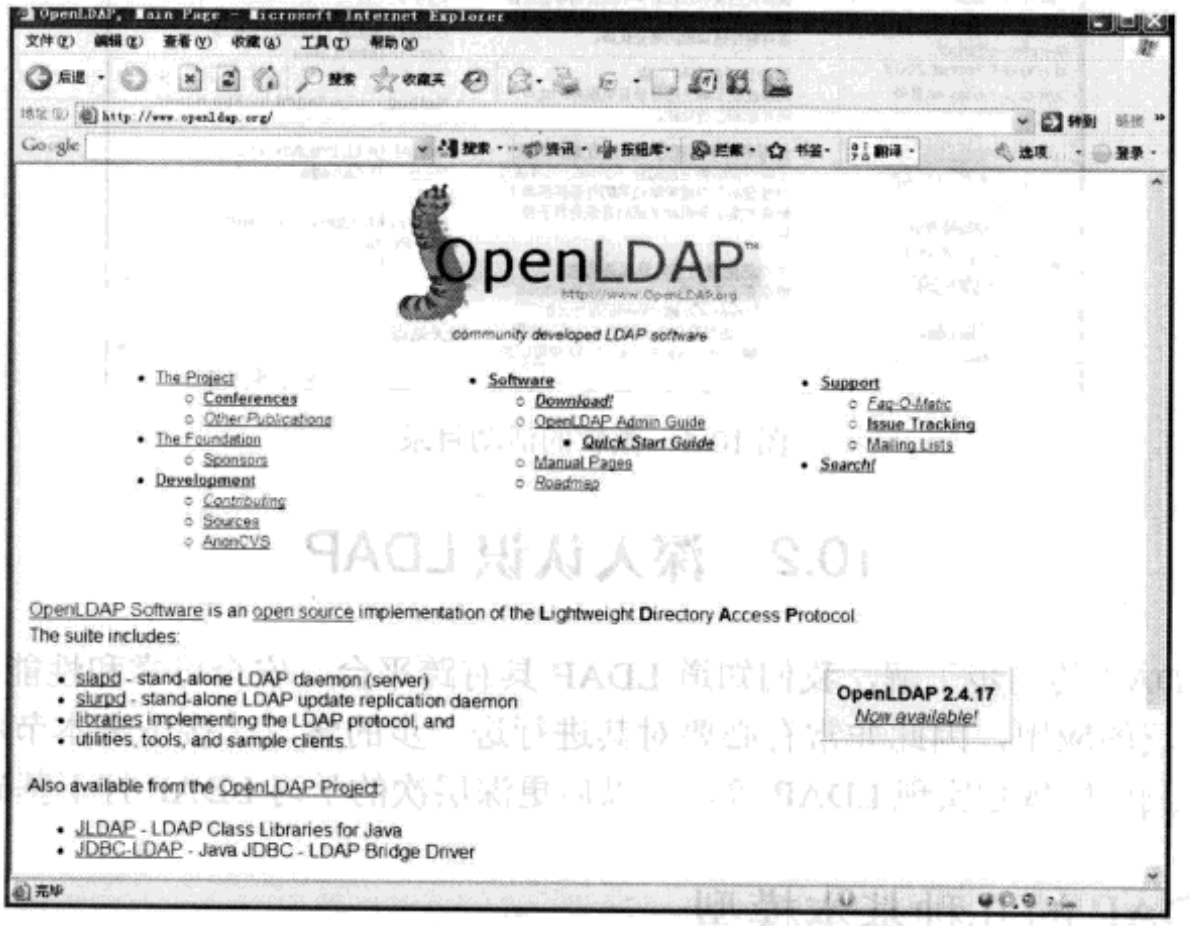


图 10-3 OpenLDAP

3. Microsoft Active Directory（活动目录）

活动目录是 Windows 平台的核心组件，其为用户管理网络环境各个组成要素的标识和关系提供了一种有效的方法，如图 10-4 所示。

Active Directory 是面向 Windows Standard Server、Windows Enterprise Server 以及 Windows Datacenter Server 的目录服务。其使用了一种结构化的数据存储方式，存储有关网络对象的信息，可以让管理员和用户轻松地查找和使用这些信息。

注意 Active Directory 不能运行在 Windows Web Server 上，但是可以对运行 Windows Web Server 的计算机进行管理。



图 10-4　微软的活动目录

10.2　深入认识 LDAP

通过对 LDAP 的初步了解，我们知道 LDAP 具有跨平台、安全可靠和性能良好等特点，从而得到了广泛的应用，因此非常有必要对其进行进一步的学习。接下来本节将讲解 LDAP 的工作机制以及如何规划实现 LDAP 等，为以后更深层次的学习 LDAP 打下基础。

10.2.1　LDAP 的几种基本模型

LDAP 的体系结构由信息模型、命名模型、功能模型以及安全模型 4 种基本模型组成。其中，信息模型描述 LDAP 的信息表示方式；命名模型描述 LDAP 的数据如何组织；功能模型描述 LDAP 的数据操作访问方式；安全模型描述 LDAP 的安全机制。接下来将分别对这 4 种基本模型进行阐述。

1.　信息模型

LDAP 信息模型定义了能够在目录中存储的数据类型和基本的信息单位。LDAP 中的信息是以树状方式进行组织的，在树状信息中的基本数据单元是条目（即关于对象的信息集合），而每个条目由属性构成，属性中存储属性值，每个属性类型又有对应的语法和匹配规则。通常，条目中的信息说明真实世界的对象。

2.　命名模型

在 LDAP 中每个条目均有自己的 DN 和 RDN（Relative Distinguished Name，相对标识名）。

其中，DN 是该条目在整个树中的唯一名称标识，而 RDN 是条目在父节点下的唯一名称标识。

3. 功能模型

LDAP 功能模型说明了能够使用 LDAP 协议对目录执行的操作，在 LDAP 中共有 4 类 10 种操作。

- 更新类操作

包括添加条目、删除条目、修改条目、修改条目名。

- 查询类操作

包括搜索、比较。

- 认证类操作

包括绑定、解绑定。

- 其他操作

包括放弃和扩展操作。

注意　除了扩展操作，另外 9 种是 LDAP 的标准操作。扩展操作是 LDAP 中为了增加新的功能所提供的一种标准的扩展框架。不同的 LDAP 厂商均定义了自己的扩展操作。

---

4. 安全模型

LDAP 中的安全模型主要通过身份认证、安全通道和访问控制来实现，分别如下。

（1）身份认证

身份认证在 LDAP 中提供的认证机制主要有以下 3 种。

- 匿名认证

该方法适用于没有数据安全问题且不涉及访问权限的完全公开方式。

- 基本认证

均是通过用分辨名（DN）和密码进行身份识别，密码识别又分为简单密码和摘要密码认证。

- SASL（Simple Authentication and Secure Layer）认证

即 LDAP 提供的在 SSL 和 TLS 安全通道基础上进行的身份认证，包括数字证书的认证。

（2） 通信安全

通信安全在 LDAP 中提供了基于 SSL/TLS 的通信安全保障。SSL/TLS 是基于 PKI 信息安全技术的，是目前 Internet 上广泛采用的安全服务。TLS 服务可以被 LDAP 通过 StartTLS 方式启动，该服务既可以提供通信中的数据保密性、完整性保护也可以实现对客户端身份和服务器端身份的双向验证。

（3） LDAP 访问控制的标准

LDAP 目前并无访问控制的标准，但从一些事实上的 LDAP 产品的访问控制情况可以看出，LDAP 访问控制非常得灵活和丰富。在 LDAP 中访问控制是通过访问控制策略语句来实现的；关系型数据库系统和应用系统是通过访问控制列表来实现的；其实现方式与关系型数据库系统和应用系统的实现方式有所不同。

10.2.2 如何规划目录树

规划一个灵活的且易于扩展的目录树是实现 LDAP 的关键，做好 LDAP 的规划工作可以减少后期维护目录树的工作量。下面通过一个案例来讲解如何规划目录树。

假设，存在一个名为 Tree 的企业，其 DNS 域名为 Tree.com，组织结构如图 10-5 所示。

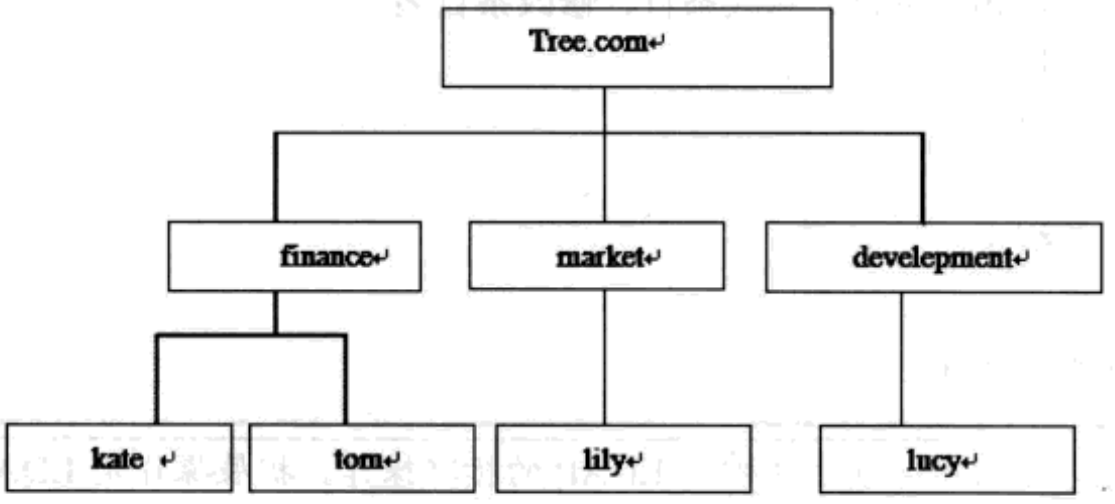


图 10-5　Tree 公司组织结构图

要为该企业实现 LDAP 可以通过以下方法完成。

首先，为整个目录树建立一个基准 DN，也叫根（即 root）。根是目录的最顶层，以后建立的所有对象都由这个根发展而来，根可以用 3 种格式表示：如果是使用 X.500 标准格式，则表示为 o=Tree，c=CN；如果是直接使用公司的 DNS 域名，则表示为 o=Tree.com。如果是使用公司 DNS 域名的不同部分则可以表示为 dc= Tree，dc=com。当然为了便于以后目录树的扩展，建议使用最后一种格式。

然后建立组织单元。组织单元 OU（Organizational Unit）则是公司中的部门，如“ou= market”表示公司中的市场部门。OU 是目录树的分支节点，其下面可以包含其他的分支节点或者叶子节点。

最后建立用户。目录树的最底层为用户，即叶子节点，可以根据用户所在的部门将其放置在不同的组织单元 OU 中，如“uid= kate”表明 kate 是财务部的一名员工。

因此，通过以上描述，可以将 Tree 企业的组织结构图转换为如图 10-6 所示的 LDAP 目录信息树。

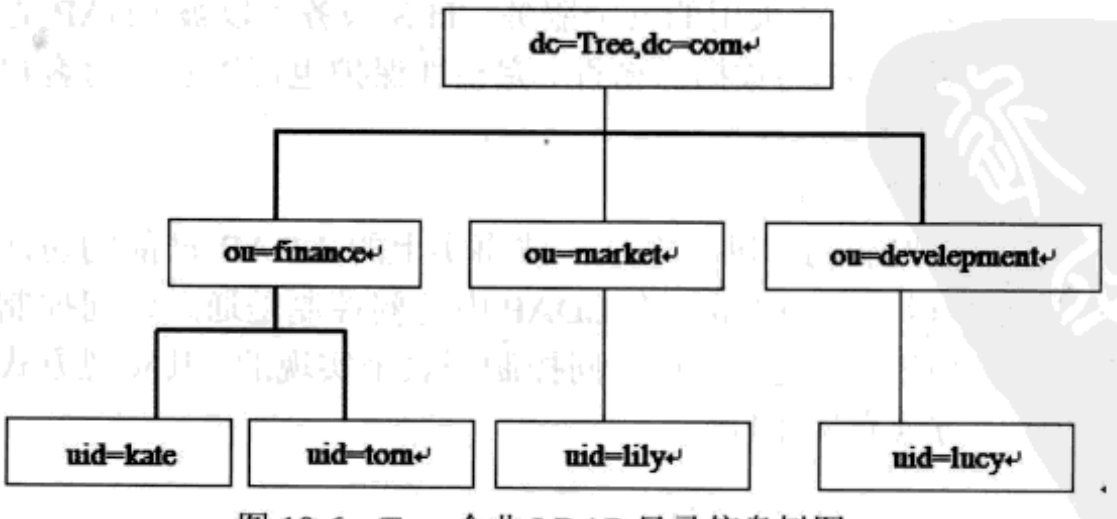


图 10-6　Tree 企业 LDAP 目录信息树图

10.2.3 LDAP 的应用领域

LDAP 被广泛地用于基础性、关键性信息的管理，主要涉及以下应用领域。

- 信息安全类

数字证书管理、授权管理、单点登录。

- 网络资源管理类

MAIL 系统、DNS 系统、网络用户管理、电话号码簿。

- 科学计算类

DCE（Distributed Computing Envirionment，分布式计算环境），UDDI（Universal Description，Discovery and Integration，统一描述，发现和集成协议）。

- 电子政务资源管理类

内网组织信息服务，电子政务目录体系，人口基础库，法人基础库。

LDAP 之所以能广泛地用于管理用户信息、网络资源信息等，是由于 LDAP 具有高效率的查询，树状的信息管理模式，分布式的部署框架以及细致灵活的访问控制的特点。

10.3 安装 LDAP

通过前面几节的学习，相信大家对 LDAP 已经有了系统的认识，在对 LDAP 有了理论认识后，接下来将通过 OpenLDAP 开源软件来具体讲解如何搭建 LDAP 环境。

10.3.1 安装 Berkeley DB

搭建 LDAP 环境的首要任务是安装 OpenLDAP 所支持的后台数据库，OpenLDAP 可以支持多种后台数据库，Berkeley DB 是其中之一。BDB 是 OpenLDAP 后台数据库的默认配置，因此在安装 OpenLDAP 之前应先安装 BDB。

1. Berkeley DB 简介

Berkeley DB（BDB）是由美国 Sleepycat Software 公司开发的一套开放源代码的嵌入式数据库系统，其为数据的存取和管理提供了一组简洁的函数调用 API 接口。BDB 具有轻便灵活、良好的可伸缩性、高性能的事务处理机制等优点，因此许多大公司（如 HP、Sun 和 Cisco）都采用该数据库。

- 轻便灵活

可以运行于几乎所有的 UNIX 和 Linux 系统及其变种系统、Windows 操作系统以及多种嵌入式实时操作系统之下。

- 可伸缩

Database library 本身不大，却能够管理规模高达 256 TB 的数据库，并支持高并发度，成千上万个用户可同时使用同一个数据库。

2. 安装前的准备

在安装该软件之前，需要做好以下准备。

第 1 步，获取软件包。

使用浏览器访问 BDB 官方网站 http://www.oracle.com/technology/global/cn/products/berkeley-db/db/index.html，单击页面中的“下载”超链接（本书使用的 BDB 的版本是 4.6.21 版本），如图 10-7 所示。



图 10-7 单击页面中的“下载”超链接

第 2 步，单击 “下载”超链接后，会显示“Berkeley DB 版本历史”页面，在该页面中单击“Berkeley DB 4.6.21.tar.gz”超链接下载 Berkeley DB 4.6.21 的源代码安装文件，如图 10-8 所示。

获得安装包后，将进入 BDB 的安装。

3. 安装 BDB

安装 BDB 的具体步骤如下。

第 1 步，编译、安装 Berkeley DB。

首先进入 Berkeley DB 4.6.21 的下载目录，解压 db-4.6.21.tar.gz 文件，执行如下命令。

```
[root@localhost ~]# tar zxvf db-4.6.21.tar.gz
```

解压缩后，进入 db-4.6.21 文件目录的 build_unix 目录。

```
[root@localhost ~]# cd db-4.6.21/build_unix
```

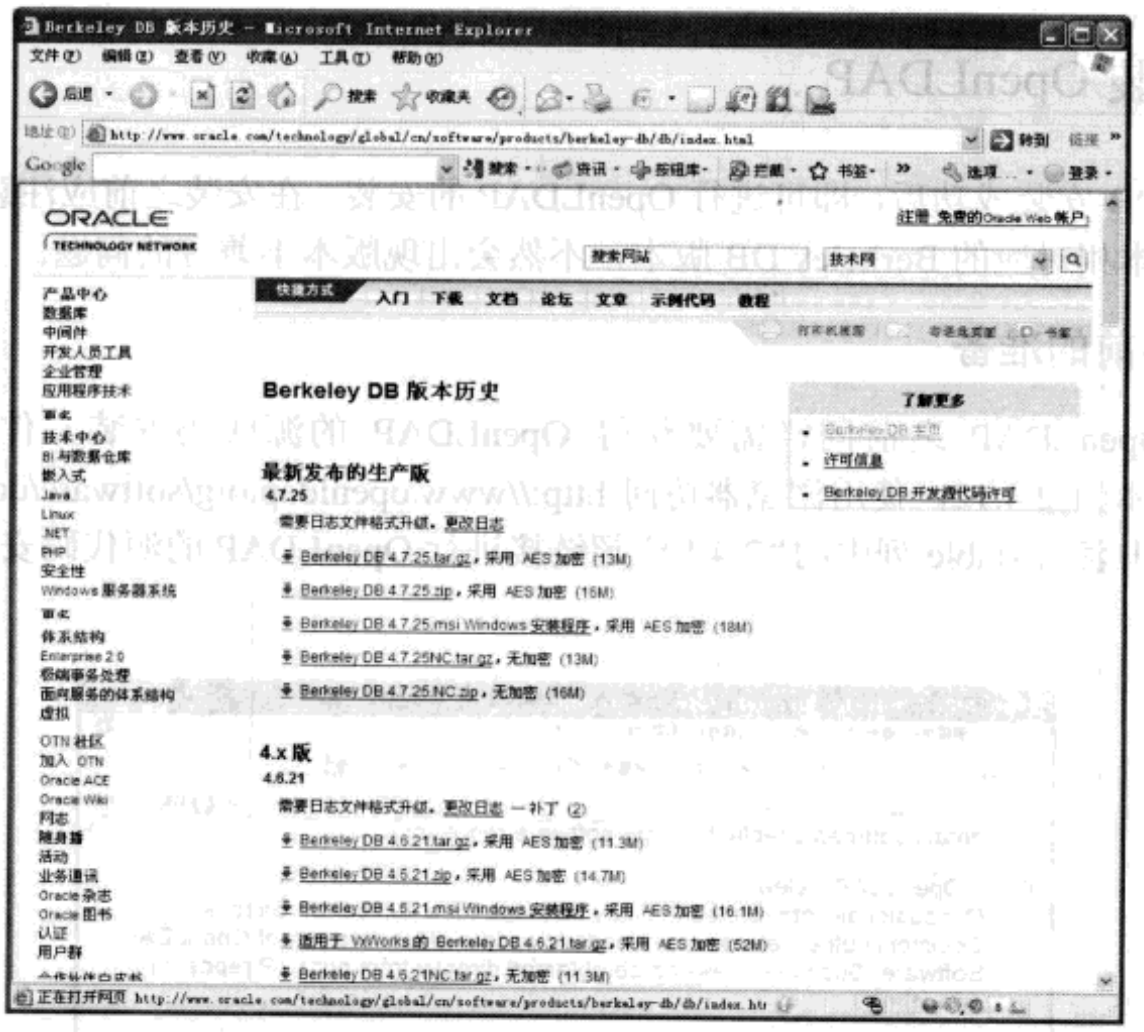


图 10-8 “Berkeley DB 版本历史”页面

然后执行以下命令进行配置测试。

```
[root@localhost db-4.6.21]#../dist/configure
```

测试完毕后，执行命令编译源代码。

```
[root@localhost db-4.6.21]# make
```

最后执行安装。

```
[root@localhost db-4.6.21]# make install
```

第 2 步，配置系统动态链接库的路径。

默认情况下 BDB 被安装在/usr/local/BerkeleyDB.4.6/目录中，为了能让 OpenLDAP 使用 BDB 的库文件，还需将 BDB 库文件所在的目录添加到系统动态链接库的路径中，具体实现方法如下。

编辑系统动态链接库的配置文件/etc/ld.so.conf，在文件的末尾添加语句/usr/local/BerkeleyDB.4.6/lib。

然后使用 ldconfig 命令刷新系统动态链接库缓存。

```
[root@localhost ~]# /sbin/ldconfig
```

在系统动态链接库的配置文件 ld.so.conf 中存放着可被 Linux 共享的动态链接库所在目录的名字（系统目录/lib,/usr/lib 除外）。

10.3.2 安装 OpenLDAP

Berkeley DB 安装成功后，即可进行 OpenLDAP 的安装。在安装之前应注意，需要选择与 OpenLDAP 版本相对应的 Berkeley DB 版本，不然会出现版本不兼容的问题。

1. 安装前的准备

在安装 OpenLDAP 之前同样需要获得 OpenLDAP 的源代码安装文件。这里使用的 OpenLDAP 版本为 2.4.12，使用浏览器访问 http://www.openldap.org/software/download/页面，在该页面中单击表格 stable 列中的“2.4.12”超链接进行 OpenLDAP 的源代码安装文件的下载，如图 10-9 所示。

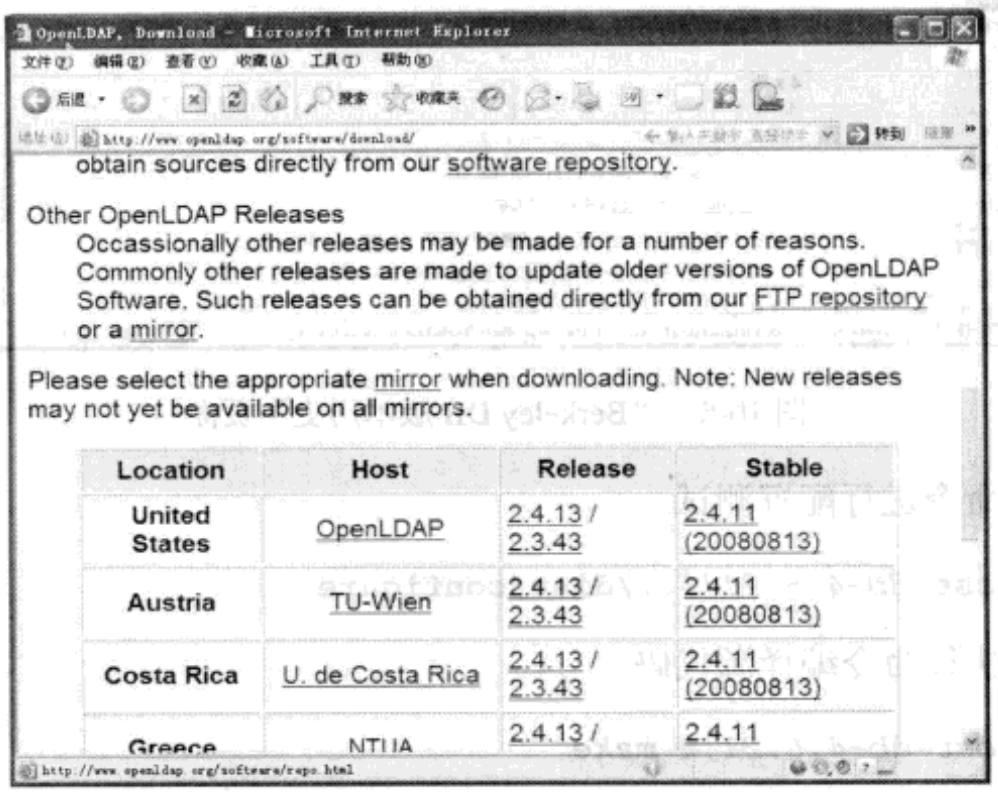


图 10-9 下载 OpenLDAP 安装文件

2. 安装 OpenLDAP

下载完安装文件后，进入 OpenLDAP 的安装，安装的具体步骤如下。

第 1 步，进入 OpenLDAP 2.4.12 的下载目录，解压 openldap-2.4.12.tgz 文件。

```
[root@localhost ~]# tar zxvf openldap-2.4.12.tgz
```

第 2 步，进入解压后的 openldap-2.4.12 文件目录。

```
[root@localhost ~]# cd openldap-2.4.12
```

第 3 步，配置环境执行如下命令。

```
[root@localhost openldap-2.4.12]#env CPPFLAGS="-I /usr/local/BerkeleyDB.4.6/include" LDFLAGS="-L /usr/local/BerkeleyDB.4.6/lib" ./configure --prefix=/usr/local/ zopenldap --enable-ldbm
```

注意 需要设置资料库的 include 和 lib 路径，否则在配置到资料库相关内容时会提示 Berkeley DB 版本不兼容，并中断配置。如果没有--enable-ldbm 选项，在测试目录时会提示 ldbm 找不到。

第 4 步，使用 GNU make 工具编译源代码。

```
[root@localhost openldap-2.4.12]# make depend
[root@localhost openldap-2.4.12]# make
[root@localhost openldap-2.4.12]# make test
```

第 5 步，执行以下指令进行安装。

```
[root@localhost openldap-2.4.12]# make install
```

10.4 配置 OpenLDAP

OpenLDAP 安装完成后，将进入 OpenLDAP 的配置阶段，本节将对 OpenLDAP 的基本配置以及如何启动 OpenLDAP 服务器进行讲解。

10.4.1 OpenLDAP 的基本配置

通过 OpenLDAP 的主配置文件/usr/local/openldap/etc/openldap/slapd.conf，可以配置以下选项。在每次对配置文件进行修改后，需要重启 OpenLDAP 服务，这样新的配置才能生效。

1. 设置 LDAP 使用的 Schema

Schema（模式）定义了 LDAP 中的对象类型、属性、语法和匹配规则等，类似于关系数据库中的表结构。Schema 是变化的，因此可以自定义，通过自定义 Schema 来实现扩展 LDAP 中存储信息的模式。OpenLDAP 安装目录的 etc/openldap/schema/子目录下包含了许多常用的 Schema 定义文件，一般情况下，使用这些文件已经可以满足日常应用的需要了。

可以使用如下方法来设置 LDAP 使用的 Schema。

编辑 slapd.conf 文件在其中找到如下语句。

```
include  /usr/local/openldap/etc/openldap/schema/core.schema
```

在该语句的后面添加以下语句。

```
include  /usr/local/openldap/etc/openldap/schema/corba.schema
include  /usr/local/openldap/etc/openldap/schema/cosine.schema
include  /usr/local/openldap/etc/openldap/schema/dyngroup.schema
include  /usr/local/openldap/etc/openldap/schema/inetorgperson.schema
include  /usr/local/openldap/etc/openldap/schema/java.schema
include  /usr/local/openldap/etc/openldap/schema/misc.schema
include  /usr/local/openldap/etc/openldap/schema/nis.schema
include  /usr/local/openldap/etc/openldap/schema/openldap.schema
```

2. 为目录树设置后缀

编辑 slapd.conf 文件，找到如下语句。

```
suffix   "dc=my-domain, dc=com"
```

将其改为如下内容。

```
suffix   "dc=Tree, dc=com"
```

3. 为 LDAP 管理员设置 DN

编辑 slapd.conf 文件，找到如下语句。

```
rootdn  "cn=Manager, dc=my-domain, dc=com"
```

将其改为如下内容。

```
rootdn  "cn=Manager, dc=Tree, dc=com"
```

4. 为 LDAP 管理员设置口令

编辑 slapd.conf 文件，找到如下语句。

```
rootpw  secret
```

将其改为如下内容。

```
rootpw  {SSHA}NXV9F128qCHMmA6P sjhVX0uejTKE6OYr
```

注意 rootpw 选项后面的大括号中描述的内容是密码的加密方式，常见的加密方式有 MD5、SSHA 和 CRYPT。本例口令使用的是 SSHA 算法加密，密文“*NXV9Fl28qCHMmA6P sjhVX0uejTKE6OYr*”是明文“helloldap”经过 SSHA 加密运算后得到的结果。

可以看到，rootpw 选项后面的大括号中的内容是加密后产生的结果。那么是如何实现口令信息加密的呢？对口令信息加密可以通过 OpenLDAP 提供的 slappasswd 命令来实现，具体方法如下。

执行命令“/usr/local/openldap/sbin/slappasswd”，会提示输入两次口令，输入两次口令后，slappasswd 命令会生成加密后的密文，如下所示。将生成后的密文复制到配置文件 slapd.conf 中即可。

```
[root@localhost ~]# /usr/local/openldap/sbin/slappasswd
New password:
Re-enter new password:
{SSHA}o10x4cDm4kZnKjThulvqwj/lb7T/flK
```

10.4.2 启动 OpenLDAP 服务器

OpenLDAP 配置完成后，下面将讲解 OpenLDAP 服务器的启动方法。要启动 OpenLDAP 服务器，只需执行以下命令。

```
[root@localhost ~]# /usr/local/openldap/libexec/slapd
```

然后执行以下命令确保 slapd 进程启动成功。

```
[root@localhost ~]# pstree|grep "slapd"
```

执行结果如下所示，显示了“l-slapd”，表明 lapd 进程已经成功启动。

```
[root@localhost ~]# /usr/local/openldap/libexec/slapd
[root@localhost ~]# pstree|grep "slapd"
        |-slapd----{slapd}|
```

另外，如果 Linux 服务器开启了防火墙功能，就需要关闭防火墙功能或设置允许 TCP 协议的 389 端口通过，这是因为 LDAP 服务使用 TCP 协议的 389 端口。使用以下命令就可以用命令方式开放 TCP 协议的 389 端口。

```
[root@localhost ~]# iptables -I INPUT -p tcp --dport 389 -j ACCEPT
```

10.5 为 OpenLDAP 建立初始化数据

为 OpenLDAP 建立初始化数据是为了以后通过使用 LDAP 客户端软件来管理目录树。建立初始化数据的方法并不复杂，只需将初始化数据保存在文件中，然后使用 ldapadd 命令导入 LDAP 数据库中即可。

通过 LDIF（LDAP Data Interchange Format，LDAP 数据交换格式）可以将 LDAP 条目保存为一种简单的文本格式。该格式可以用于在符合 LDAP 标准的目录上完成如添加、修改、删除和导入、导出数据等批量操作。另外，在使用的后台数据库不同或者是在不同的操作系统上运行的情况下，LDIF 还可以在来自两个不同厂家的 LDAP 服务器间交换数据，接下来将学习如何建立初始化数据。

首先建立 LDIF 文件。使用 vi 为文本编辑工具建立一个名为“excample.ldif”的 LDIF 文件，内容如下。

```
dn:dc=Tree, dc=com
objectClass:dcObject
objectClass:organization
dc: Tree
o: Tree, Inc.
dn:cn=Manager, dc= Tree, dc=com
objectClass:organizationalRole
cn:Manager
```

在创建文件成功后将 LDIF 中的数据导入到 LDAP 数据库中，需执行如下命令。

```
[root@localhost ~]#/usr/local/openldap/bin/ldapadd -x -W -D "cn=Manager,
dc=Tree, dc=com" -f hope.ldif
```

执行命令成功后，将出现“Enter LDAP Password”的提示，在正确输入密码后，即可出现“adding new entry ‘dc=Tree.dc=com’”的提示，该提示表明 LDAP 数据导入成功。

10.6 LDAP 图形界面管理工具简介

LDAP 图形界面管理工具有很多种，比如 phpLDAPadmin、GQLDAP client 以及 LDAP Browser/Editor 等。通过该图形界面管理工具用户可以轻松、方便地对 LDAP 目录树进行管理，避免了使用命令方式管理目录树的麻烦。因此，本节将重点介绍如何安装与配置 LDAP 图形界面管理工具。

10.6.1 主要的 LDAP 图形界面管理工具

LDAP 图形界面管理工具主要有以下 3 种。

1. phpLDAPadmin

phpLDAPadmin 是一款使用 PHP 编写的开放源代码，基于 Web 的用于管理 LDAP 服务器的 LDAP 管理工具。该工具为用户提供了友好的用户界面，并具有强大的添加、删除、修改和查询等功能，同时可以浏览支持 LDAP 第 3 版服务器发布的 Schema 信息。可以通过该工具管理本地或远程的 LDAP 服务器，它的下载地址为：http:// php!dapadmin.sourceforge.net/，如图 10-10 所示。

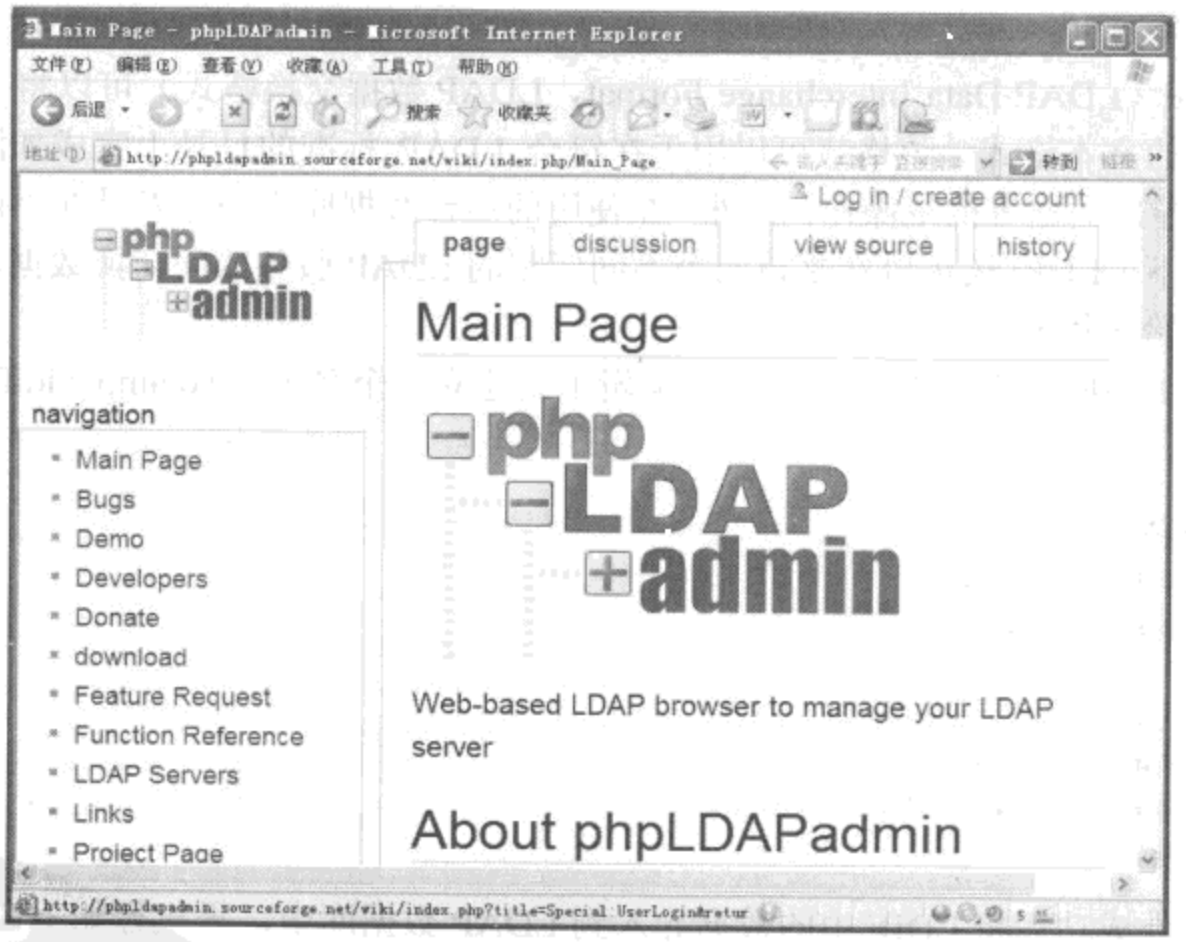


图 10-10 phpLDAPadmin

除此之外，phpLDAPadmin 管理器还具有以下优点。

- 可以删除一个实体或整棵子树及其节点。
- 可以移动或复制子节点或整棵子树。
- 可以以树状结构浏览 LDAP 目录树并观察选定节点的属性。

- 导入导出 LDIF 格式文件等编辑功能。

2. 基于 KDE 环境的 GQ LDAP client

GQ LDAP client 是基于 KDE 环境的 LDAP 图形界面管理工具，如图 10-11 所示，其官方站点为：http://biot.com/gq。

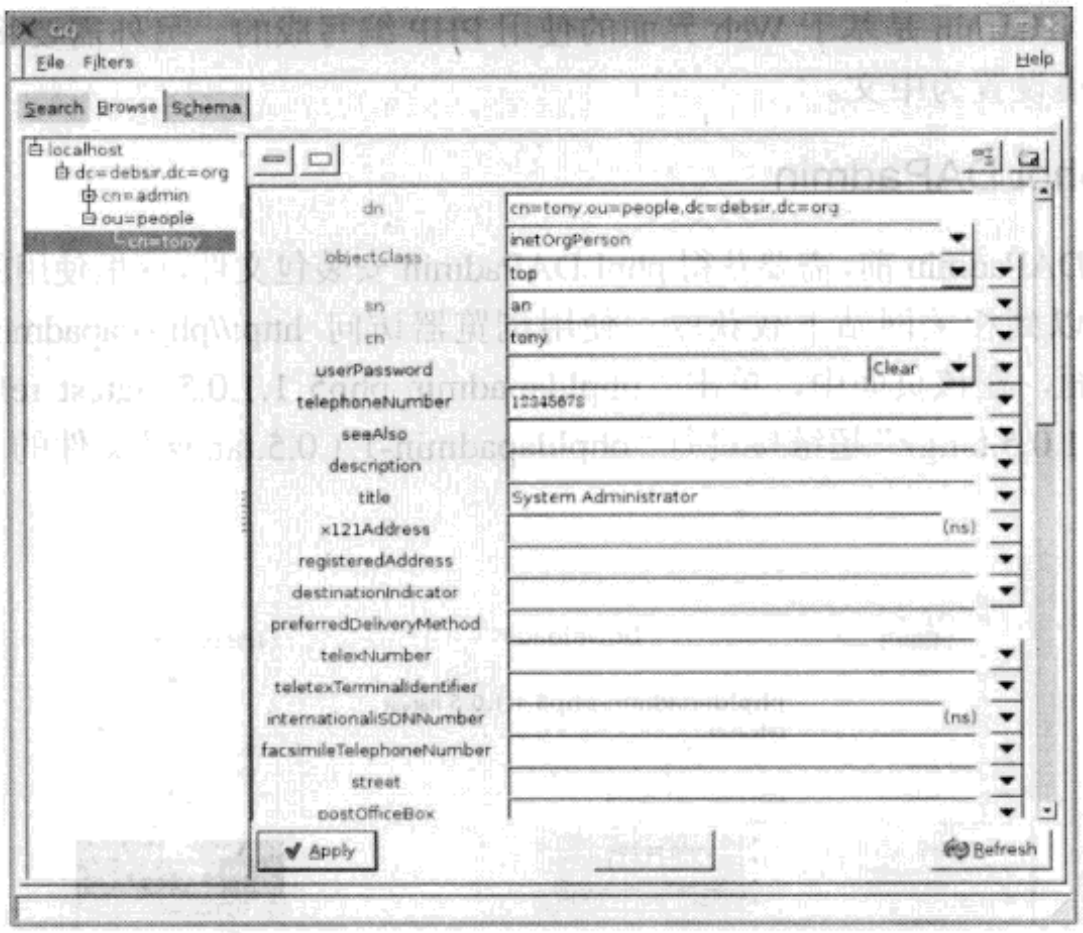


图 10-11 GQ LDAP client

3. LDAP Browser/Editor

LDAP Browser/Editor 是用 Java 编写的 LDAP 图形界面管理工具，如图 10-12 所示。其官方站点为：http://www-unix.mcs.anl.gov/~gawor/ldap。

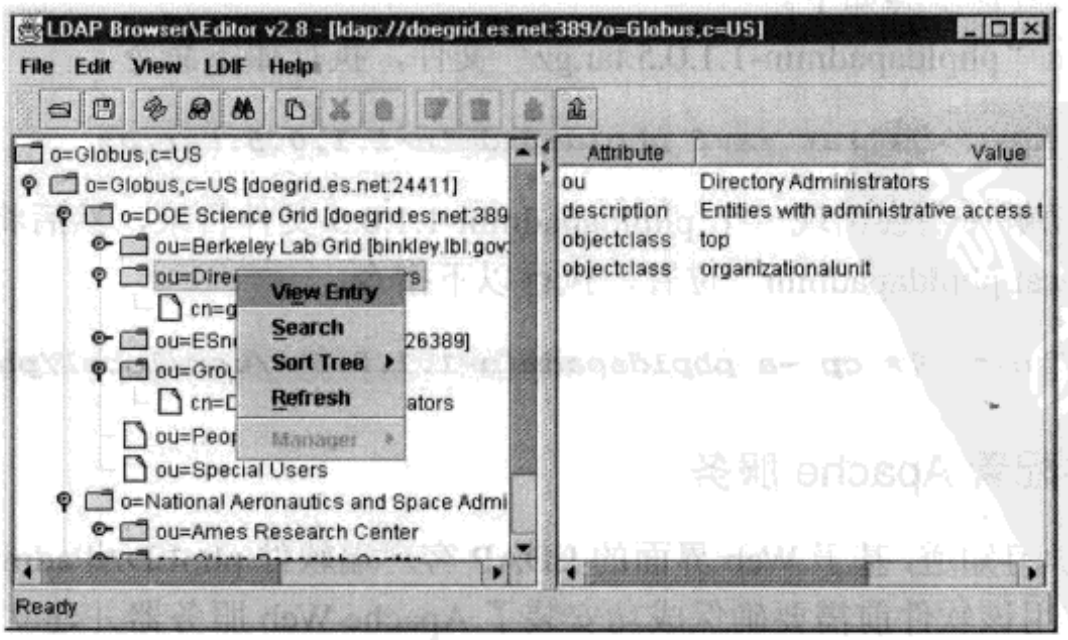


图 10-12 LDAP Browser/Editor

10.6.2 安装 phpLDAPadmin

phpLDAPadmin 是一种开源的 LDAP 图形界面管理工具，在使用 LDAP 客户端软件 phpLDAPadmin 软件前需要确保成功安装了 Apache Web 服务器并建立好了 PHP 的运行环境。这是因为 phpLDAPadmin 是基于 Web 界面的使用 PHP 编写成的。另外需要将 Apache Web 服务器的默认字符集设置为中文。

1. 安装 phpLDAPadmin

在安装 phpLDAPadmin 前，需要获得 phpLDAPadmin 安装包文件，这里使用的是 1.1.0.5 版本，该安装包文件可以到相关网站下载获取。使用浏览器访问 http://phpldapadmin.sourceforge.net/download.php 页面，在该页面中，单击“phpldapadmin-php5 1.1.0.5 (latest release)”栏目中的“phpldapadmin-1.1.0.5.tar.gz”超链接进行“phpldapadmin-1.1.0.5.tar.gz”文件的下载，如图 10-13 所示。



图 10-13 下载 phpLDAPadmin

成功下载“phpldapadmin-1.1.0.5.tar.gz”文件后，将进入 phpLDAPadmin 的安装，安装 phpLDAPadmin 的具体步骤如下。

第 1 步，解压“phpldapadmin-1.1.0.5.tar.gz”文件，执行如下命令。

```
[root@localhost ~]# tar zxvf phpldapadmin-1.1.0.5.tar.gz
```

第 2 步，成功解压后会出现一个 phpldapadmin-1.1.0.5 文件目录，然后将解压后的安装文件部署到“/usr/local/phpldapadmin”位置，执行以下指令。

```
[root@localhost ~]# cp -a phpldapadmin-1.1.0.5  /usr/local/phpldapadmin
```

2. 安装并配置 Apache 服务

通过以上的学习知道，基于 Web 界面的 LDAP 客户端软件 phpLDAPadmin 是使用 PHP 编写成的，因此在使用该软件前需要确保成功安装了 Apache Web 服务器并建立好了 PHP 的运行环境。另外需要将 Apache Web 服务器的默认字符集设置为中文，以便用户操作。

Apache 服务的具体安装方法请参见本书第 4 章 Web 服务的配置与管理。

在确认 Apache 服务器已经安装后，需要执行“/etc/init.d/httpd　start”命令确认 Web 服务已经启动。在客户端通过在浏览器中输入 Linux 服务器的地址访问进行测试，如果显示 Apache 的测试页面，则表示服务器已经运行正常，如图 10-14 所示。

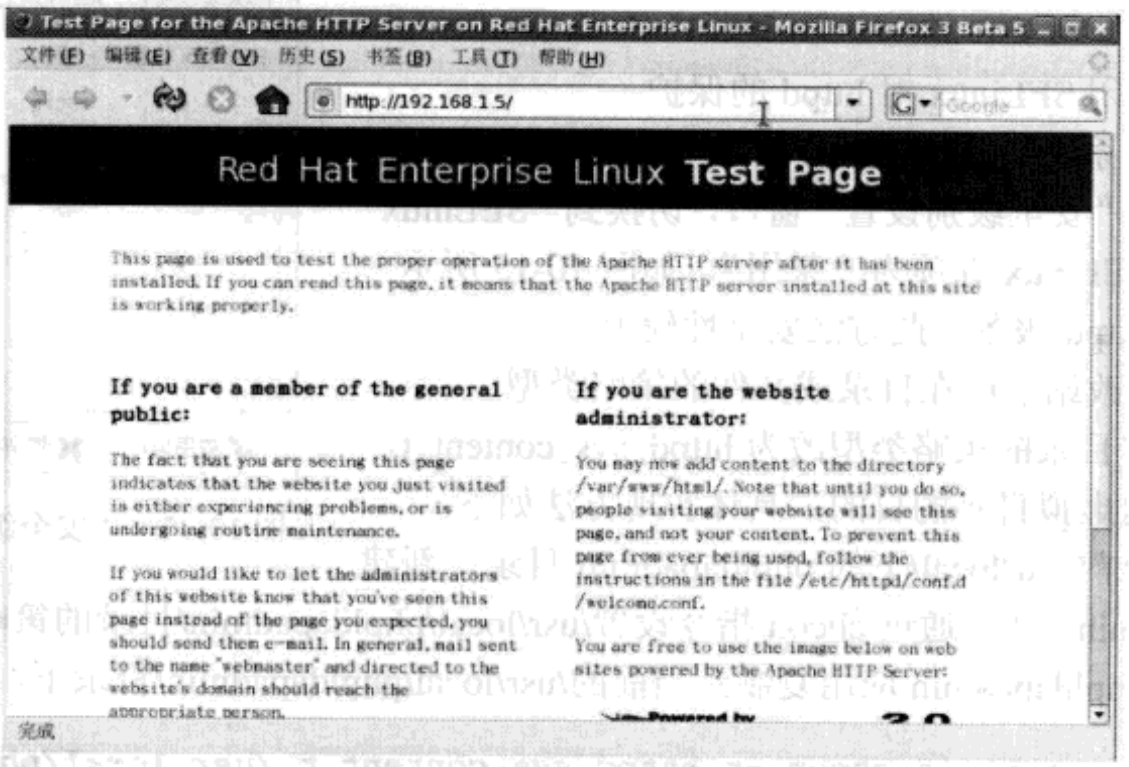


图 10-14　Web 服务器运行成功

确认 Apache 服务已经成功安装并能启动后，还需要对其进行以下配置。

（1） 加入用户认证功能

为了进一步加强认证安全，需要加入 Apache 的用户认证功能。使用如下命令建立口令文件。

```
[root@localhost ~]# htpasswd -c /etc/httpd/php_ldap_admin_pwd admin
```

命令执行成功后，输入用户 admin 的口令，即可生成口令文件“/etc/httpd/php_ldap_admin_pwd”，如下所示。

```
[root@localhost ~]# htpasswd -c /etc/httpd/php_ldap_admin_pwd admin
New password:
Re-type new password:
Adding password for user admin
```

另外，必须创建虚拟目录才可以在主目录以外的目录中进行发布。在 Web 浏览器中可以使用虚拟目录的别名来访问虚拟目录，接下来将会讲解如何建立虚拟目录。

（2） 建立虚拟目录

在 Apache 的主配置文件 httpd.conf 中加入以下语句建立虚拟目录。

```
Alias /phpldapadmin/ "/usr/local/phpldapadmin/"
<Directory "/usr/local/phpldapadmin">
Options Indexes MultiViews
AllowOverride None
Order allow,deny
Allow from all
AuthType Basic
AuthName "Please Login to phpldapadmin"
AuthUserFile /etc/httpd/php_ldap_admin_pwd
Require user admin
```

```
</Directory>
```

（3） 如何访问虚拟目录

创建好虚拟目录后，可以通过以下两种方法实现对虚拟目录的访问。

方法 1 关闭 SELinux 对 httpd 的保护

依次单击“系统”→“管理”→“安全级别和防火墙”菜单命令，弹出“安全级别设置”窗口，切换到“SELinux”选项卡，设置 SELinux 选项为“禁用”，如图 10-15 所示。

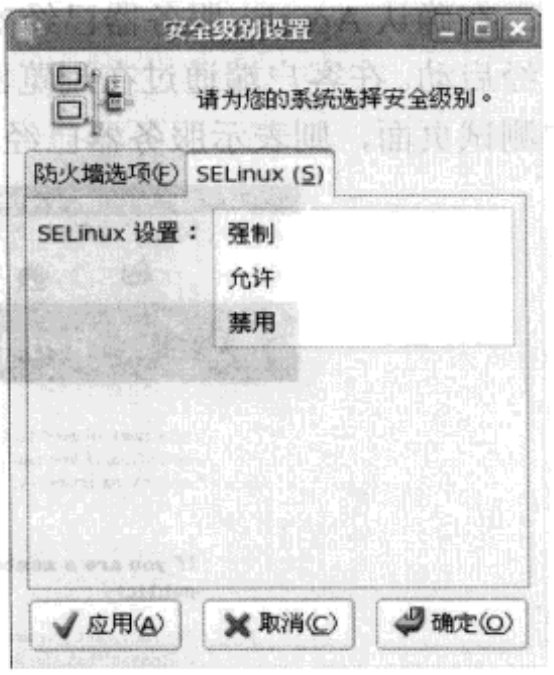


图 10-15 “安全级别设置”窗口

然后重启 httpd 服务，此方法安全性较差。

方法 2 修改站点所在目录或文件的策略类型

把站点所在目录的策略类型改为 httpd_sys_content_t，也可以达到访问虚拟目录的目的，具体实现方法如下。

首先删除以前/usr/local/下的 phpldapadmin 目录，新建目录 phpldapadmin，然后通过 chcon 指令设置/usr/local/phpldapadmin/目录的策略类型，最后重新将解压好的 phpldapadmin 应用复制到当前的/usr/local/phpldapadmin/目录下，执行如下命令。

```
[root@localhost ~]# chcon -t httpd_sys_content_t /usr/local/phpldapadmin
```

此时/usr/local/phpldapadmin/目录下的所有子目录将继承它的策略类型。

10.7 配置 phpLDAPadmin

成功完成 phpLDAPadmin 的安装后，需要通过适当的配置才有利于以后的使用，因此本节将学习有关 phpLDAPadmin 配置方面的知识。

10.7.1 生成 phpLDAPadmin 主配置文件

phpLDAPadmin 主配置文件的模板文件是/usr/local/phpldapadmin/config/config.php.example 文件，用户可以通过复制主配置文件模板的方式来生成一个名为 config.php 的主配置文件，执行如下命令。

```
[root@localhost ~]# cp /usr/local/phpldapadmin/config/config.php.example
/usr/local/phpldapadmin/config/config.php
```

生成该主配置文件后，即可通过修改此配置文件，使系统 phpLDAPadmin 能够良好地支持中文，以后的配置工作都是基于该主配置文件的。

10.7.2 phpLDAPadmin 的基本配置

通过编辑 phpLDAPadmin 的主配置文件/usr/local/phpldapadmin/config/config.php 可以对 phpLDAPadmin 进行支持中文的配置，对口令加密信息的设置以及对 LDAP 服务器名称、地址和端口等一系列选项的设置，接下来将详细介绍这些选项的相关配置。

1. 对 phpLDAPadmin 进行支持中文的配置

安装好 phpLDAPadmin 后的界面是英文的，对不同国度的用户，需要根据自己的需求来配置合适的语种。因此，对于中国用户而言，需要对其进行支持中文的设置。通过上一小节的实验，已经生成了 phpLDAPadmin 应用系统的主配置文件 config.php，然后需要对其中的配置选项进行修改，以便更好地支持中文。具体操作步骤如下。

第 1 步，编辑/usr/local/phpldapadmin/config/config.php 文件，并找到如下语句。

```
// $config->custom->appearance['language']='auto';
```

将语句修改为如下内容。

```
$config->custom->appearance['language']='zh_CN';
```

注意 需要将“//”注释符号去掉，在以后步骤中遇到类似情况也需要将其去掉。

第 2 步，转换 phpLDAPadmin 语言文件的编码，执行以下指令。

```
[root@localhost ~]# iconv -f gbk -t utf8
/usr/local/phpldapadmin/locale/zh_CN/LC_MESSAGES/messages.po >
/usr/local/phpldapadmin/locale/zh_CN/LC_MESSAGES/messages.new.po
[root@localhost ~]#msgfmt -o
/usr/local/phpldapadmin/locale/zh_CN/LC_MESSAGES/messages.po >
/usr/local/phpldapadmin/locale/zh_CN/LC_MESSAGES/messages.new.po
```

第 3 步，编辑/usr/local/phpldapadmin/htdocs/header.php 文件，并查找如下语句。

```
$language = isset($config) ? $language =
$config->GetValue('appearance','language') : 'auto';
```

将该语句修改为如下内容。

```
$language = isset($config) ? $language =
$config->GetValue('appearance','language') : 'zh_CN';
```

通过以上设置后，与 phpLDAPadmin 有关的语言选项便配置完成。

2. 设置口令加密信息

为了安全地加、解密一些敏感信息，必须配置一个口令加密字符串。在配置文件中查询语句。

```
$config->custom->session['blowfish']='';
```

将该语句修改为如下内容。

```
$config->custom->session['blowfish']='hildap';
```

该设置表明把口令加密字符串设置为“hildap”。

3. 为 LDAP 服务器设置名称、地址和端口等信息

在配置文件中查询以下语句。

```
$ldapservers->SetValue($i,'server',' name','My LDAP Server');
$ldapservers->SetValue($i,'server','host','127.0.0.1');
$ldapservers->SetValue($i,'server','port','389');
```

并将其依次修改为如下内容。

```
$ldapservers->SetValue($i,'server',' name','My LDAP Server');
$ldapservers->SetValue($i,'server','host','192.168.5.10');
$ldapservers->SetValue($i,'server','port','389');
```

4. 为 LDAP 服务器设置管理员的 DN

在配置文件中查询以下语句。

```
$ldapservers->SetValue($i,'login','dn','cn=Manager,dc=tree,dc=com');
```

需要根据实际的 LDAP 服务器管理员的 DN 进行修改。

5. 为 LDAP 服务器设置管理员的口令

在配置文件中查询以下语句。

```
$ldapservers->SetValue($i,'login','pass','');
```

表明其口令为空。

注意　如果用户认证方式设置的是 config，则必须在此填写正确的 LDAP 服务器管理员的口令，否则会造成 phpLDAPadmin 不能连接 LDAP 服务器的错误。

6. 设置用户认证方式

在配置文件中查询以下语句。

```
$ldapservers->SetValue($i,'Server','auth_type','cookie');
```

确保认证方式为 cookie 方式。

phpLDAPadmin 提供了 cookie、session 和 config 3 种认证方式，区别如下。

- config

config 是 phpLDAPadmin 默认的认证方式，不需要用户登录，但需要将 LDAP 管理员的 DN 和口令写在 phpLDAPadmin 的主配置文件的 login_dn 和 login_pass 选项中。

- cookie

phpLDAPadmin 提供了一个登录的界面，让用户输入正确的 LDAP 管理员的 DN 与口令会保存在客户端的 Web 浏览器的 cookie 里面。

- session

类似于 cookie，不同之处在于 LDAP 管理员的 DN 与口令保存在服务器端，而不是客户端。

7. 为目录树设置基准 DN

在配置文件中查询以下语句。

```
$ldapservers->SetValue($i,'server','base',array('dc=tree,dc=com'));
$queries[$q]['base']='dc=tree,dc=com';
```

需要根据实际的目录树基准进行修改。

10.8 使用 OpenLDAPadmin 管理目录树

安装并配置好 phpLDAPadmin 的各选项后，就可以使用该软件进行目录树的管理了，下面将通过实例学习如何使用该软件进行目录树的管理。

10.8.1 进入 phpLDAPadmin

要对目录树进行管理首先需要登录 phpLDAPadmin，登录的具体步骤如下。

第 1 步，首先重启 Apache 服务，然后使用浏览器访问 http://Linux 服务器的 IP 或者域名/phpldapadmin/，提示输入用户名和口令，如图 10-16 所示。

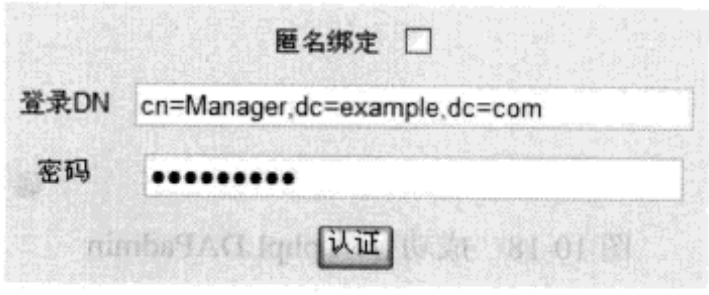


图 10-16 提示输入用户名和口令

第 2 步，正确输入用户名和口令后即可进入 phpLDAPadmin 的管理主页面，如图 10-17 所示。

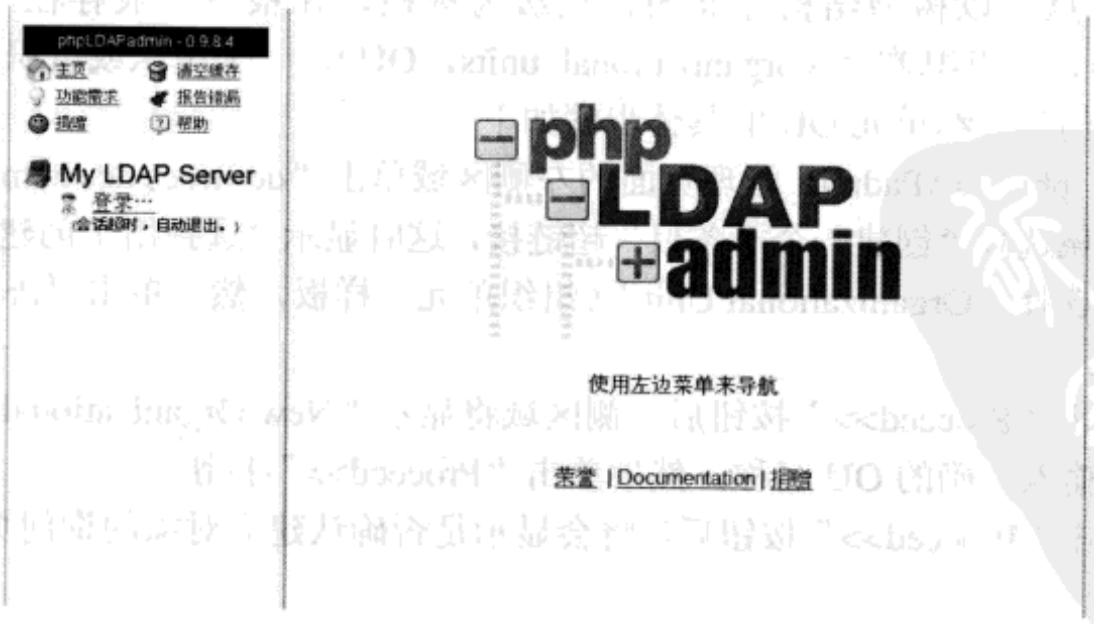


图 10-17 进入 phpLDAPadmin 的管理主页面

第 3 步，登录 LDAP 服务器。

在页面左侧区域中单击“登录…”超链接，然后在右侧区域中输入正确的 LDAP 管理员的 DN 和口令进行登录。成功登录后，在页面的左侧区域中就会显示 LDAP 目录树的信息，并提供各种操作目录树的超链接，如图 10-18 所示。

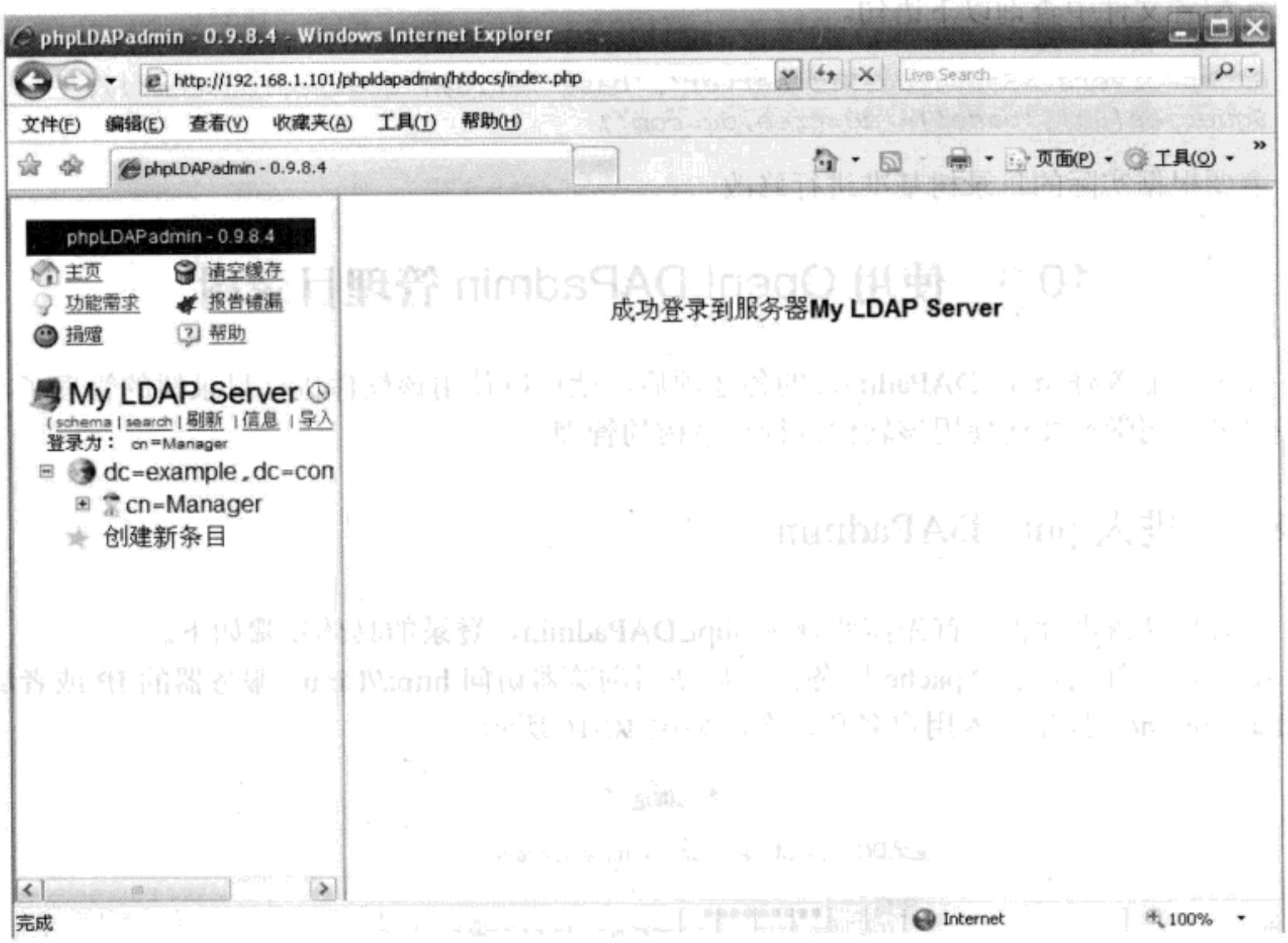


图 10-18　成功登录 phpLDAPadmin

10.8.2 创建组织单元

LDAP 的信息是以树型结构存储的，顶级为树根，在根下一般存在一个或多个组织（organization，O）或组织单元（organizational units，OU），一个组织或组织单元可能包含基本属性等信息。创建组织单元 OU 的具体步骤如下。

第 1 步，在 phpLDAPadmin 管理页面的左侧区域单击“dc=tree,dc=com”超链接，然后在页面的右侧区域选择“创建一个子条目”超链接，这时显示“选择用于创建过程的样板”页面，在该页面中选择“Organizational Unit”（组织单元）样板，然后单击“Proceed>>”按钮，如图 10-19 所示。

第 2 步，单击“Proceed>>”按钮后，侧区域将显示“New Organizational Unit”页面，如图 10-20 所示。输入正确的 OU 名称，然后单击“Proceed>>”按钮。

第 3 步，单击“Proceed>>”按钮后，将会显示是否确认建立对象的询问页面，如图 10-21 所示。



图 10-19 创建一个子条目



图 10-20 创建 OU



图 10-21 确认创建 OU

第 4 步，进入询问页面后，单击“创建对象”按钮。此时新的 OU 属性已建立，最后 pLDAPadmin 会显示新建 OU 的属性，如图 10-22 所示。

通过以上配置，用户创建组织单元的工作已完成。



图 10-22　显示 OU 的属性

10.8.3 创建用户组

由于 Linux 系统存在用户组的概念，因此当 LDAP 用户登录 Linux 系统时，会报“id: cannot find name for group ID”的错误，该错误说明没有建立用户组，需要在 LDAP 中建立用户组。

创建用户组的具体操作步骤如下。

第 1 步，在 phpLDAPadmin 管理页面中单击“dc=tree，dc=com”超链接，此时会在页面的右侧区域中显示功能超链接，单击“创建一个子条目”超链接。单击该超链接后，会在右侧区域显示“选择用于创建过程的样板”页面，如图 10-23 所示。然后选择“Posix Group”（用户组）样板，最后单击“Proceed>>”按钮。



图 10-23　选择 Posix Group 样板

第 2 步，单击“Proceed>>”按钮后，将会显示“New Posix Group”页面，如图 10-24 所

示。该页面会提示输入用户组名称、用户组 ID 以及选择 OU。

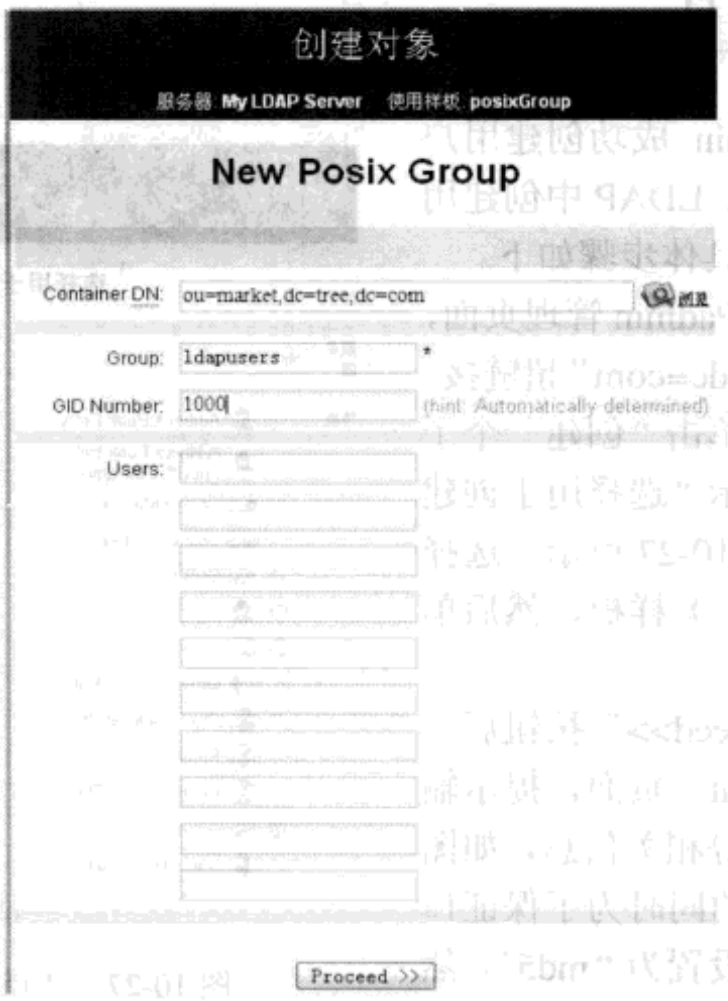


图 10-24 “New Posix Group”页面

第 3 步，在正确执行以上操作后，单击“Proceed>>”按钮，此时 phpLDAPadmin 将显示一个是否确认建立对象的询问页面，如图 10-25 所示。

第 4 步，单击“创建对象”按钮，最后 phpLDAPadmin 会显示新建的用户组的属性，如图 10-26 所示。

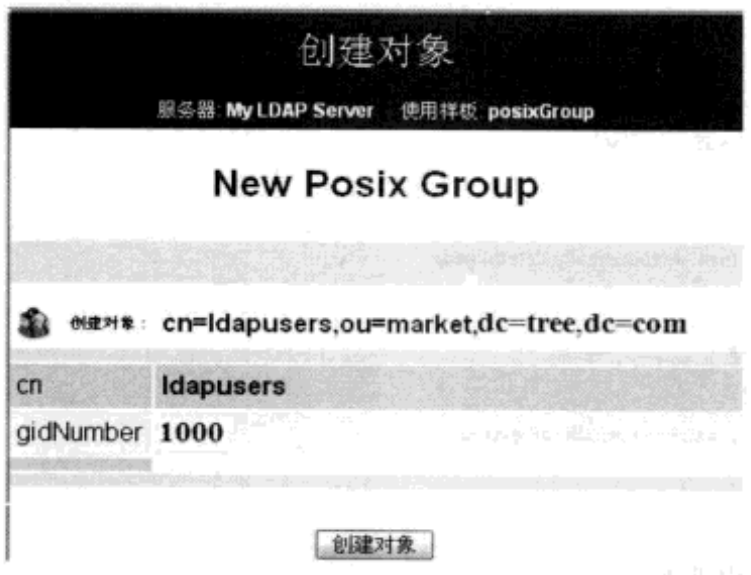


图 10-25 确认创建用户组

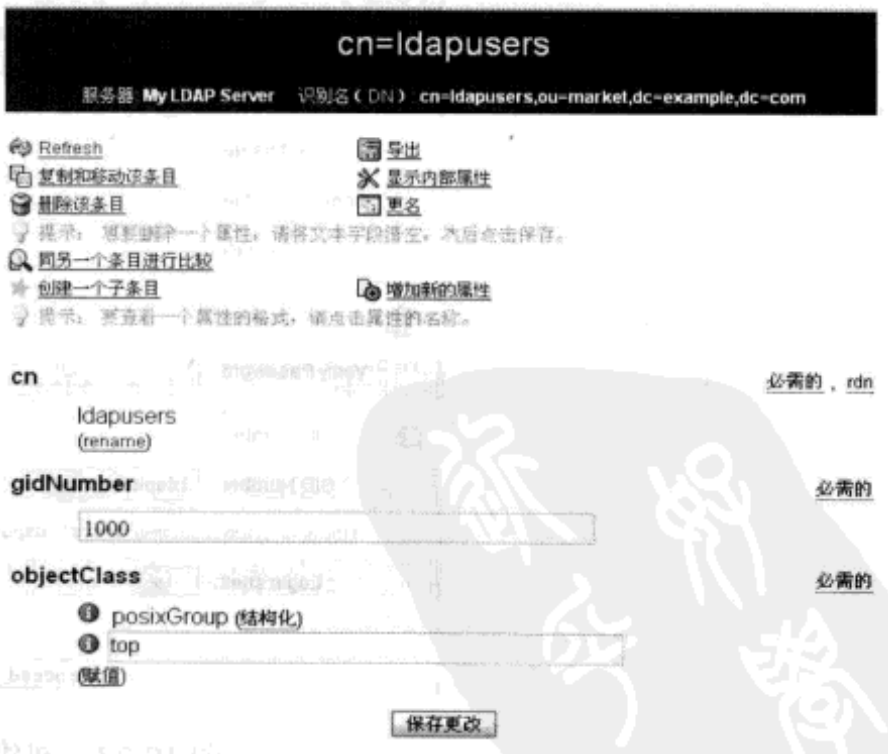


图 10-26 显示新建的用户组的属性

通过以上配置后，用户组的创建工作已完成。

10.8.4 创建用户账号

在使用 phpLDAPadmin 成功创建用户组之后，下面将介绍如何在 LDAP 中创建用户账号。创建用户账号的具体步骤如下。

第 1 步，登录 phpLDAPadmin 管理页面，单击此页面中的“dc=tree，dc=com”超链接，此时会显示功能超链接，单击“创建一个子条目”超链接。接着将显示“选择用于创建过程的样板”页面，如图 10-27 所示。选择“User Account”（用户账号）样板，然后单击“Proceed>>”按钮。



图 10-27 选择用于创建过程的样板

第 2 步，单击“Proceed>>”按钮后，将显示“New User Account”页面，提示输入用户账号，选择用户组等相关信息，如图 10-28 所示。在填选信息的同时为了保证口令安全，需要将加密方法设置为“md5”，然后单击“Proceed>>”按钮。



图 10-28 创建用户账号

第 3 步，单击“Proceed>>”按钮后，phpLDAPadmin 将显示一个是否确认建立对象的询问页面。在该询问页面中单击“创建对象”按钮，最后 phpLDAPadmin 会显示新建的用户账

号的属性信息，如图 10-29 所示。

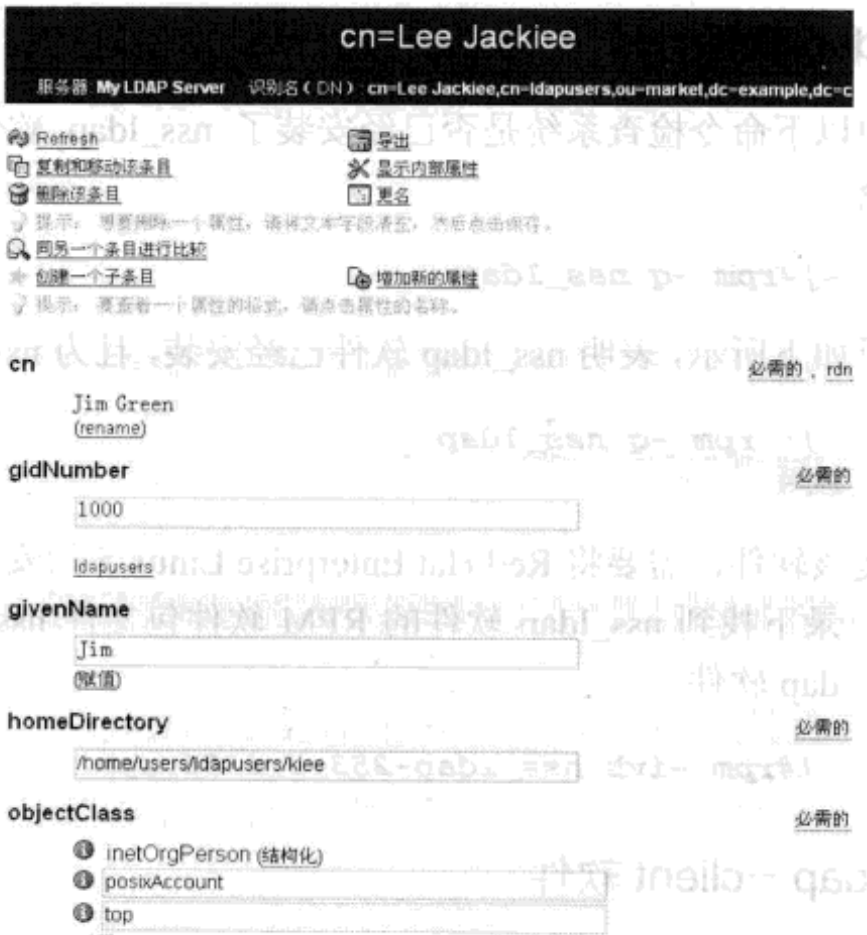


图 10-29　新建的用户账号的属性信息

通过以上配置后，用户账号的创建已完成。

10.9　实现 LDAP 服务的身份验证

目前 LDAP 协议认证被 Linux 操作系统及广大的服务类软件所支持，各服务软件在使用 LDAP 协议认证作为用户验证时可以参考相关资料，这里就以 Linux 系统用户验证、FTP 用户验证和 Web 用户验证这 3 类验证为代表，具体讲解如何实现 LDAP 协议认证作为用户验证。

10.9.1　实现 Linux 系统用户验证

Linux 操作系统支持用户通过 LDAP 协议认证、登录操作系统，接下来以 Red Hat Enterprise Linux 5 操作系统为例进行讲解。

首先需要安装能使 Linux 和 LDAP 服务器能进行良好通信的软件，包括 nss_ldap 和 openldap-clients 软件。

- nss_ldap

该软件主要用于与 LDAP 服务器通信，是可以查询和处理主机用户、用户组等信息的程序。

- openldap-clients

该软件是 OpenLDAP 的客户端程序，为用户提供了一些操作 OpenLDAP 的实用程序。

安装这两个软件的具体步骤如下。

1. 安装 nss_ldap 软件

在安装前，需使用以下命令检查系统是否已经安装了 nss_ldap 软件或查看已经安装了何种版本，执行如下命令。

```
[root@localhost ~]#rpm -q nss_ldap
```

执行命令后的结果如下所示，表明 nss_ldap 软件已经安装，且为 nss_ldap-253-12.el5 版本。

```
[root@localhost ~]# rpm -q nss_ldap
nss_ldap-253-12.el5
```

如果系统没有安装该软件，需要将 Red Hat Enterprise Linux 5 的安装盘放入光驱，加载光驱后在光盘的 Server 目录下找到 nss_ldap 软件的 RPM 软件包文件 nss_ldap-253-3.i386.rpm，使用如下命令安装 nss_ldap 软件。

```
[root@localhost ~]#rpm -ivh nss_ldap-253-3.i386.rpm
```

2. 安装 openldap－client 软件

在安装前，需使用如下命令查看系统是否已经安装了 openldap－client 软件或查看已经安装了何种版本，执行如下命令。

```
[root@localhost ~]# rpm -q openldap－client
```

执行命令后的结果如下所示，表明 openldap-client 软件未安装，需要进行相应的安装。

```
[root@localhost ~]# rpm -q openldap－client
package openldap－client is not installed
```

通过检测，openldap-client 软件未安装，因此可以通过以下方法进行安装。

将 Red Hat Enterprise Linux 5 的安装盘放入光驱，加载光驱后在光盘的 Server 目录下找到 openldap-client 软件的 RPM 软件包文件 openldap-clients-2.3.27-5.i386.rpm，使用如下命令安装 openldap-client 软件。

```
[root@localhost ~]#rpm -ivh openldap-clients-2.3.27-5.i386.rpm
```

命令成功执行后，显示如下所示的结果。

```
[root@localhost Server]# rpm -ivh vsftpd-2.0.5-12.el5.i386.rpm
Preparing...                ########################################### [100%]
   1: openldap－client      ########################################### [100%]
```

如果系统开启了 SELinux 功能，则需要在设置用户认证前关闭 SELinux，设置并测试完后再启用 SELinux，如此一来可以避免错误情况的发生。

关闭 SELinux 的方法是：编辑 SELinux 的配置文件/etc/selinux/config，找到语句“SELINUX=ecforcing”并将其修改为：“SELINUX=disable”，然后重启系统，让新的配置生效。

在安装完相应的软件后，即可设置使用 LDAP 进行用户认证，具体步骤如下。

第 1 步，在控制台执行“authconfig-tui”指令启动用户验证配置程序，成功执行该指令后，将显示“验证配置”对话框，在该对话框中选中“用户信息”栏目中的“使用 LDAP”选项和“验证”栏目中的“使用 LDAP 验证”选项，如图 10-30 所示。

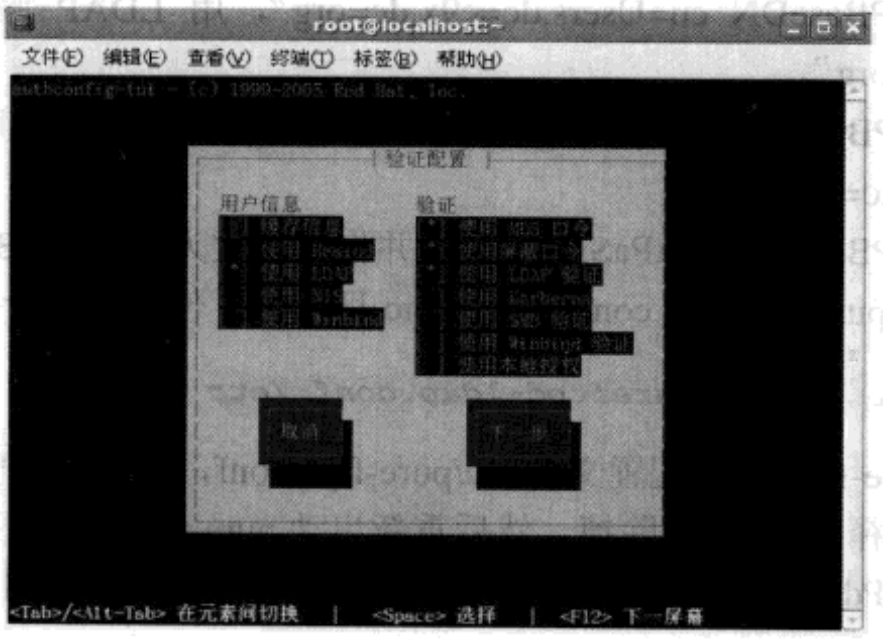


图 10-30 “验证配置”对话框

第 2 步，选中此两项后，单击“下一步”按钮，此时将弹出“LDAP 设置”对话框，在该对话框中输入 LDAP 服务器的地址和基点 DN，最后单击“确定”按钮，如图 10-31 所示。

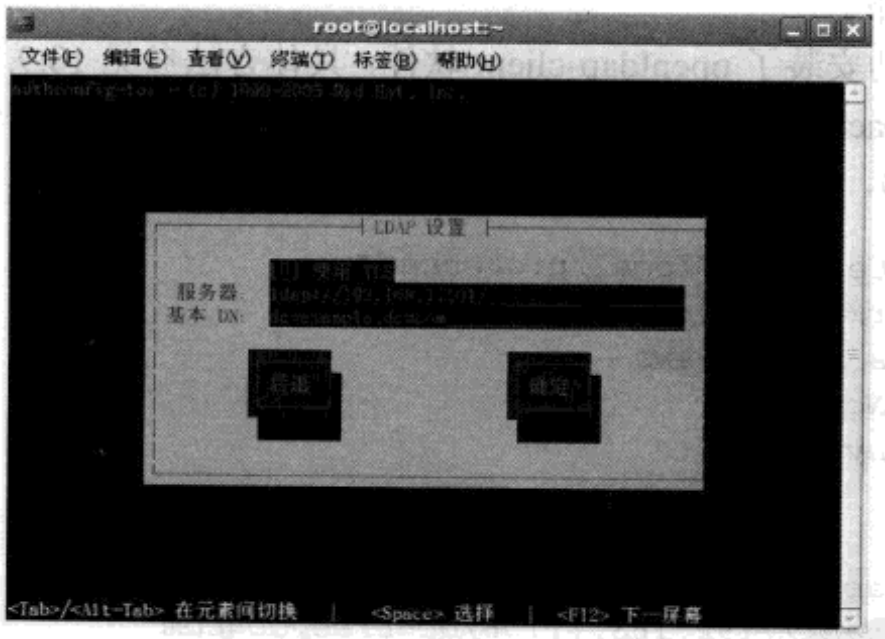


图 10-31 “LDAP 设置”对话框

通过以上配置，用户就可以使用 LDAP 协议进行用户身份验证了。

10.9.2 实现 FTP 用户验证

使用 FTP 用户认证同样需要 openldap-client 软件的支持，同时还需要安装、配置 PureFTPd 服务，具体实现步骤如下。

第 1 步，确保成功安装，具体方法参见 10.9.1 Linux 系统用户验证。

第 2 步，安装 PureFTPd 服务，并在安装编译 PureFTPd 时在./configure 脚本命令行中加入“-with-ldap”参数。

第 3 步，进入 Pure-FTPD 源文件解压后的目录，编辑配置文件 pureftpd-ldap.conf，然后对其进行以下修改。

找到语句“LDAPServer ldap.c9x.org”，用 LDAP 服务器的真实地址替换“ldap.c9x.org。

找到语句“LDAPBaseDN cn=Users,dc=c9x,dc=org”，用 LDAP 服务器的基点 DN 替换“cn=Users,dc=c9x,dc=org”。

找到语句“LDAPBindDN cn=Manager,dc=c9x,dc=org”，用 LDAP 服务器的 BindDN 替换“cn=Manager,dc=c9x,dc=org。

找到语句“LDAPBindPW r00tPaSsw0rD”，并将其修改为“LDAPBindPW helloldap”。

第 4 步，将文件 pureftpd-ldap.conf 复制到/etc 目录中，使用以下命令。

```
[root@localhost ~]# cp pureftpd-ldap.conf /etc
```

第 5 步，编辑 pure-ftpd 的主配置文件/etc/pure-ftpd.conf，找到语句“# LDAPConfigFile/etc/pure-ftpd.conf”，将注释符“#”删除掉，然后重新启动 pure-ftpd 服务。

重新启动 PureFTPd 服务后，就可以使用在 LDAP 服务器建立的用户登录 FTP 服务了。

10.9.3 实现 Web 用户验证

采用 Apache 作为 Web 服务器时，Apache 服务支持 LDAP。Web 用户验证的实现方法比较简单，具体步骤如下。

第 1 步，确保成功安装了 openldap-client 软件，具体方法参见 10.9.1 Linux 系统用户验证。

第 2 步，编辑 Apache 服务的主配置文件 httpd.conf，并添加如下语句（这里是对 protected 目录进行用户的验证），并重新启动 Apache 服务。

```
Alias /myprivate "/usr/local/protected"
<Directory /usr/local/protected >
Options Indexes MultiViews
AllowOverride None
Order allow,deny
Allow from all
AuthType Basic
AuthName "Please Login,thanks"
AuthLDAPURL "ldap://192.168.1.106/dc=tree,dc=com"
Require valid-user
</Directory>
```

第 3 步，重新启动 Apache 服务后，可以通过在/usr/local/protected 目录下建立一个测试页面 test.jsp，来测试配置的 Web 用户验证是否成功，此时就可以使用在 LDAP 服务器建立的用户来登录、访问 protected 这个虚拟目录。

测试页 *test.php* 代码如下。

```
<%out.println"Hello world!"%>
```

第 4 步，在成功建立页面后，即可使用浏览器访问 http://192.168.1.106/protected 页面，如果配置的 Web 用户验证成功，则可看到在页面中显示“Hello world”的字样。

第 11 章　SSH 服务器

随着网络应用的不断普及，通过网络传输的信息也越来越多，在保证信息能够正确到达目的地的同时，信息安全尤为重要。在网络上传输的数据如果没有经过加密，又或者是加密的强度不够，就将带来很大的安全隐患。目前，为保证网络传输数据的安全，一般都采用 SSH 加密技术。信息经过 SSH 加密并且压缩，不仅使数据的安全得到了保障，还使得信息传输的速度得到了提高。

11.1　SSH 服务概述

SSH（Secure Shell，安全外壳协议）实现了将网络上各主机之间的数据安全地连接在一起，使得网络上安全传输任意移动数据成为了可能。目前，一般使用免费的 OpenSSH 代替 SSH 通信协议。

11.1.1　SSH 服务简介

在利用 TELNET、rlogin 和 FTP 进行网络连接时，网络传输中的数据是没有经过加密的，安全性不高。而 SSH 是一个在应用程序中提供安全通信的协议，对传输的数据进行加密，保证数据在传输时不被恶意破坏、泄露和篡改。

目前通常使用 SSH 代替 TELNET 进行远程管理，SSH 协议有两个版本，SSH 1 和 SSH 2，二者采用不同的协议实现，互不兼容。目前广泛使用的是在安全、功能和性能上更有优势的 SSH2。

OpenSSH 是免费的 SSH 协议替代软件，需要 OpenSSL-0.97 及以上版本的支持，提供许多安全的实用程序，简要描述如表 11-1 所示。

表 11-1　OpenSSH 的实用程序

程 序 名	作 用 描 述
sshd	运行 OpenSSH 的服务器软件，监听来自客户端的连接请求，收到请求后，执行身份验证来为客户端提供服务
ssh	用于登录其他主机或者在其他主机上运行程序，属于客户端程序，有时也被称为“slogin”
scp	用于在主机间运行安全性复制工作的程序，替代 rcp 程序
ssh-keygen	用于产生 Public Key 验证密钥，即 RSA 或 DSA 的主机及用户验证密钥
ssh-agent	保存验证用的 RSA 密钥
ssh-add	产生新的密钥
sftp-server	SFTP 服务器软件
sftp	安全性更高的 FTP 程序
ssh-keyscan	集中管理 SSH 公钥软件
ssh-keysign	SSH 的说明程序，在主机验证时运行

OpenSSH 支持 SSH1 和 SSH2 协议，当前 OpenSSH 和 OpenSSL 最新版本的下载地址如下。

ftp://ftp.openbsd.org/pub/OpenBSD/OpenSSH/openssh-5.1.tar.gz

ftp://ftp.openssl.org/source/openssl-0.9.8h.tar.gz

OpenSSH 源代码软件包提供了 SSH 客户端和服务端程序（sshd）。

11.1.2 公钥加密体系结构

现代密码学的一项重要发明和进展是公开密钥密码体制。密码学的主题包括两个方面：一方面是保护信息传递的机密性；另一方面是身份验证、信息记录和数据完整性的保障。

公共密钥体制的理论结构基础是非对称性算法，其原理是：加密密钥和解密密钥分离。加密密钥谁都可以使用，但解密密钥只有解密人自己知道。不需专门使用安全通道传递密钥，简化了密钥管理。公共密钥体制有时也称为公开密钥算法或简称公钥算法。

常见的公钥密码方案有两种，如表 11-2 所示。

表 11-2 常见的公钥密码方案

密钥方案	出现时间	简　述
RSA	1978 年	所有公钥密码体系中最著名、应用最广泛的一种
DSA	1991 年	主要用于数字签名方面

这里需要了解一下。什么是证书？将公钥和公钥主人的名字联系起来，请一个大家都信得过的有信誉的公正、权威机构进行确认，并加上该机构的签名，就形成了证书。这里的权威机构就是电子签证机关（即 CA），网上的公众用户对其签字进行验证。任何人都可以通过 CA 验证其所签发的证书。

用户想得到属于自己的证书，应先向 CA 提出申请。在验证身份后，CA 为其分配一个公钥，并且将该公钥与申请者身份信息绑在一起，为之签字，形成证书发给用户（申请者）。

鉴别一个证书的真伪，可以使用 CA 的公钥对那个证书上的签字进行验证，验证通过，则说明该证书有效。

除了签发证书外，CA 还有另一个重要的作用是证书和密钥的管理。

知识点 CA 本身拥有一个证书（内含公钥），加上私钥，所以具有签字能力。CA 签字实际上是经过 CA 私钥加密的信息，签字验证的过程实际上还伴随使用 CA 公钥解密的过程。

用户 A 发送数据给用户 B，主要经过以下步骤进行加密，如图 11-1 所示。

第 1 步，用户 B 通过网络等方式向外界公开他的公钥 PUB_B。

第 2 步，用户 A 得到用户 B 的公钥 PUB_B，然后使用 PUB_B 对信息加密并发送给用户 B。

第 3 步，用户 B 在收到密文后，使用其私钥 PRI_B 解密。

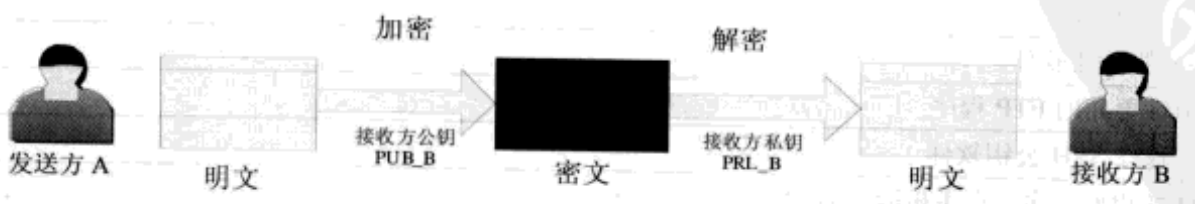


图 11-1 公共密钥算法加、解密过程

11.2 SSH 服务的安装

在 Red Hat Enterprise Linux5 系统按默认设置安装的情况下，OpenSSH 服务、客户程序以及 OpenSSL 都会自动安装。可通过命令查看系统是否已经安装 OpenSSH 服务。

第 1 步，在命令行窗口，输入以下命令查看系统中 OpenSSH 服务的安装情况。

```
[root@localhost ~]#rpm -qa|grep ssh
```

若系统已经安装了 OpenSSH 服务，将输出以下 4 项。

```
openssh-4.3p2-16.e15                    ;    #OpenSSH 的核心软件包
openssh-askpass-4.3p2-16.e15
openssh-client-4.3p2-16.e15             ;    #OpenSSH 的客户端程序
openssh-server-4.3p2-16.e15             ;    #OpenSSH 的服务端程序
```

第 2 步，如果命令执行没有返回以上 4 项，则说明当前系统还未安装 OpenSSH 服务，需要在 Red Hat Enterprise Linux 5 的安装光盘中找到以下 RPM 文件并安装。

```
openssh-4.3p2-16.e15.i386.rpm
openssh-askpass-4.3p2-16.e15.i386.rpm
openssh-clients-4.3p2-16.e15.i386.rpm
openssh-server-4.3p2-16.e15.i386.rpm
```

第 3 步，加载 Red Hat Enterprise Linux 5 的安装盘，然后在光盘的 Server 目录下找到相应的 RPM 安装包文件，使用 rpm -ivh 命令安装。

```
[root@localhost Server]#rpm -ivh openssh-4.3p2-16.e15.i386.rpm
[root@localhost Server]#rpm -ivh openssh-askpass-4.3p2-16.e15.i386.rpm
[root@localhost Server]#rpm -ivh openssh-clients-4.3p2-16.e15.i386.rpm
[root@localhost Server]#rpm -ivh openssh-server-4.3p2-16.e15.i386.rpm
```

注意 提供远程访问的服务器，必须安装 OpenSSH 服务器软件；用于远程登录服务器的客户机，必须安装 OpenSSH 的客户端软件包。

11.3 配置 SSH 服务器

OpenSSH 服务器软件包包含服务器进程程序（/usr/sbin/sshd）、配置文件、启动脚本和帮助文档几个部分。查看软件包所安装的文件及安装的位置，可以使用以下命令。

```
[root@localhost ~]#rpm -ql openssh-server
/etc/pam.d/sshd
/etc/rc.d/init.d/sshd
/etc/ssh
/etc/ssh/sshd_config
/usr/libexec/openssh/sftp-server
/usr/sbin/sshd
```

```
/usr/share/man/man5/sshd_config.5.gz
/usr/share/man/man8/sftp-server.8.gz
/usr/share/man/man8/sshd.8.gz
/var/empty/sshd
/var/empty/sshd/etc
[root@localhost ~]#
```

其中 sshd 进程启动脚本是文件/etc/rc.d/init.d/sshd，SSH 服务的配置文件是/etc/ssh/sshd_config，sftp 服务进程是文件/usr/libexec/openssh/sftp-server，sshd 守护进程程序是/usr/bin/sshd。

可以通过修改配置文件/etc/ssh/sshd_config，设置 SSH 服务的运行参数。以下是对/etc/ssh/sshd_config 文件中各选项的说明。

Port：默认值是 22，指定 sshd 监听的连接端口号。

Protocol：默认值是“2，1”（多版本编号间用逗号分隔），指定 sshd 支持的 SSH 通信协议版本的顺序。表示默认使用 SSH2 协议，不成功才使用 SSH1 协议。一般来说，为了安全起见，可以设置为只使用 SSH 2 协议。

ListenAddress：默认值是“0.0.0.0”（IPv4 地址）和“::”（IPv6 地址），指定 sshd 监听的地址，可以是 IPv4 或 IPv6 的地址。

HostKey：指定 SSH 私有密钥保存的文件，SSHv1 的默认值是“/etc/ssh/ssh_host_key”，而 SSHv2 则保存“/etc/ssh/ssh_host_rsa_key”和“/etc/ssh/ssh_host_dsa_key”两个文件。

KeyRegenerationInterval：默认值为 3600，在 SSHv1 中，为了避免入侵者有时间将窃取的密钥进行解密，服务器使用的密钥在一定时间间隔后会自动再生。此处如果设为 0，表示不用再生密钥。

ServerKeyBits：最小值为 512，默认为 768，定义 SSHv1 的服务器密钥长度。

SyslogFacility：默认值是“AUTH”，可用的值有：AUTH、DAEMON、USER、LOCAL0、LOCAL1、LOCAL2、LOCAL3、LOCAL4、LOCAL5、LOCAL6 和 LOCAL7，用于设置 sshd 记录信息中的设备码（Facility Code）。

LogLevel：默认的记录层级为“INFO”，可用的选项有：QUIET、FATAL、ERROR、INFO、VERBOSE、DEBUG、DEBUG1、DEBUG2 和 DEBUG3 等，用于设置记录文件内容的详细程度。

LoginGraceTime：默认值是 600 s，在这个设置时间内，如果用户不能够成功地登录，那么服务器将中断这次连接，设为 0 则表示没有时间限制。

PerimitRootLogin：是否允许用 root 账号登录，默认值是“yes”。如表 11-3 所示为可用的选项及说明。

表 11-3　PerimitRootLogin 选项说明

选　项	说　明
yes	允许以 root 账号登录
without-password	停止使用 root 账号的密码验证
fored-commands-only	允许用公钥法验证 root 账号登录
no	不允许以 root 账号登录

StrictModes：默认值是“yes”，用于在接受用户登录前，sshd 是否需要检查文件的模式和

主目录的拥有权。

RSAAuthentication：只适用于 SSHv1，默认值是“yes”，用于指定是否允许使用 RSA 验证。

PubkeyAuthentication：只适用于 SSHv2，默认值是“yes”，用于指定是否允许公钥验证。

AuthorizedKeysFile：默认值是“.ssh/authoried_keys”，用于指定一个文件来保存公钥（可验证用户身份）。

RhostsAuthentication：用于是否使用 rhosts 或/etc/hosts.equiv 文件来进行验证。这种方法的不足之处在于安全性低，建议使用“RhostsRSAAuthentication”代替，默认值是“no”，但只适用于 SSHv1。

IgnoreRhosts：默认值是“yes”，用于替换“RhostsAuthentication”、“RhostsRSAAuthentication”和“HostbasedAuthentication”选项中的“.rhosts”和“.shosts”，文件为“/etc/hosts.equiv”和“/etc/ssh/shosts.equiv”。

RhostsRSAAuthentication：适用于 SSHv1，默认值是“no”，用于指定是否允许同时使用 RSA 和 rhosts 或/etc/hosts.equiv 文件来进行验证。

HostbasedAuthentication：和“RhostsRSAAuthentication”相似，只适用于 SSHv2，默认并不应用该功能。决定是否允许同时使用 rhosts 或/etc/hosts.equiv 验证，以及公钥客户端主机验证。

IgnoreUserKnownHosts：默认值是“no”，该选项用于在应用“RhostsRSAAuthentication”或“HostbasedAuthentication”功能时，指定 sshd 是否忽略“$HOME/.ssh/known_hosts”中的主机。

PasswordAuthentication：默认值是“yes”，用于指定是否允许密码验证。

PermitEmptyPasswords：默认值是“no”，在允许密码验证的情况下，设置是否允许用户使用空格密码进行登录。

ChallengeResponseAuthentication：默认值是“yes”，用于指定是否应用 Challenge Response 的验证方法。

KerberosAuthentication：默认值是“no”，用于指定是否应用 Kerberos 验证。

KerberosOrLocalPassword：默认值是“yes”，使用该选项，密码不能通过 Kerberos 验证的话，其正确性将由本地的机制（Mechanism）来决定，例如/etc/password。

KerberosTicketCleanup：默认值是“yes”，用于在用户注销后，指定是否自动删除用户的 Tick Cache 文件。

AFSTokenPassing：默认值是“no”，用于指定 AFS Token 是否可以传送到该服务器。

KerberosTgtPassing：默认值是“no”，用于指定是否允许将 Kerberos TGT 传送至该服务器。

PAMAuthenticationViaKbdInt：默认值是“yes”，用于指定是否应用 PAM Challenge Response 验证法。

X11Forwording：用于指定是否允许 X11 传输。

X11DisplayOffset：默认值是 10，用于指定 sshd X11 传送时第一个可供使用的显示编号。

X11UserLocalhost：默认值是“yes”，用于指定 sshd 是否将 X11 传送服务器的 IP 地址转换为 Loopback 地址或通用地址（Wildcard Address）。

PrintMotd：默认值是“yes”，在用户登录后，用于指定 sshd 是否显示/etc/motd 文件的内容。

PrintLastLog：默认值是“yes”，用于指定 sshd 是否显示用户上次登录的时间及日期。

KeepAlive：默认值是“no”，用于指定服务器是否传送 TCP keepalive 信息给客户端，应用该选项后，任何闲置的未断开连接的客户端，都将收到警示信息。

UserLogin：默认值是“no”，用于是否允许在会话式的登录阶段使用“login”命令。

UserPrivilegeSeparation：默认值是“no”，用于指定是否由 sshd 来建立多个子程序，来处理不同的权限要求。

Compression：默认值是“yes”，用于指定是否应用文件压缩功能。

MaxStartups：默认值是 10，用于指定在同一时间段内允许未获授权连接的用户人数上限。

Banner：该选项只适用于 SSHv2，默认并不应用该功能，用于指定在验证用户身份前，传送给远程用户的网页内容。

VerifyReverseMapping：默认并不应用该功能，用于指定是否允许 sshd 尝试验证远程的主机名，并检查解析出的远程主机 IP 地址是否相同。

Subsystem：只适用于 SSHv2，默认值是“/usr/libexec/openssh/sftp-server”，定义了一个外部子系统。

11.4 SSH 服务的启动和停止

启动和停止 SSH 服务，可以通过 OpenSSH 服务器的启动脚本文件“/etc/rc.d/init.d/sshd”来实现。另外，也可以使用 service 命令来启动、停止 SSH 服务。

11.4.1 启动 SSH 服务

启动 SSH 服务，可以有两种不同的方式：通过 OpenSSH 服务器的启动脚本文件“/etc/rc.d/init.d/sshd”来启动和使用 service 命令来启动。

1. 通过 OpenSSH 服务器的启动脚本文件启动 SSH 服务

通过 OpenSSH 服务器的启动脚本文件“/etc/rc.d/init.d/sshd”启动 SSH 服务，命令执行如下。

```
[root@localhost ~]# /etc/rc.d/init.d/sshd start
启动 sshd:                                                 [确定]
[root@localhost ~]#
```

命令执行后，SSH 服务开始运行。

2. 使用 Linux 下的 service 命令启动 SSH 服务

使用 service 命令启动 SSH 服务，命令执行如下。

```
[root@localhost ~]#service sshd start
启动 sshd:                                                 [确定]
[root@localhost ~]#
```

命令执行后，SSH 服务停止运行。

11.4.2 重启 SSH 服务

重启 SSH 服务，可以有两种不同的方式：通过 OpenSSH 服务器的启动脚本文件“/etc/rc.d/init.d/sshd”重启和使用 service 命令重启。

1. 通过 OpenSSH 服务器的启动脚本文件重启 SSH 服务

通过 OpenSSH 服务器的启动脚本文件“/etc/rc.d/init.d/sshd”重启 SSH 服务，命令执行如下。

```
[root@localhost ~]# /etc/rc.d/init.d/sshd restart
停止 sshd:                                                 [确定]
启动 sshd:                                                 [确定]
[root@localhost ~]#
```

命令执行后，SSH 服务自动停止并再次启动。

2. 运用 Linux 下的 service 命令重启 SSH 服务

运用 service 命令重启 SSH 服务，命令执行如下。

```
[root@localhost ~]# service sshd restart
停止 sshd:                                                 [确定]
启动 sshd:                                                 [确定]
[root@localhost ~]#
```

命令执行后，SSH 服务自动停止并再次启动。

11.4.3 自动启动 SSH 服务

为了让系统每次启动时自动运行 SSH 服务，可以设置 SSH 为自启动。执行“ntsysv”命令启动服务配置程序，在弹出的对话框中找到“sshd”服务，并在其前面加上星号“*”，选择“确定”按钮，如图 11-2 所示。

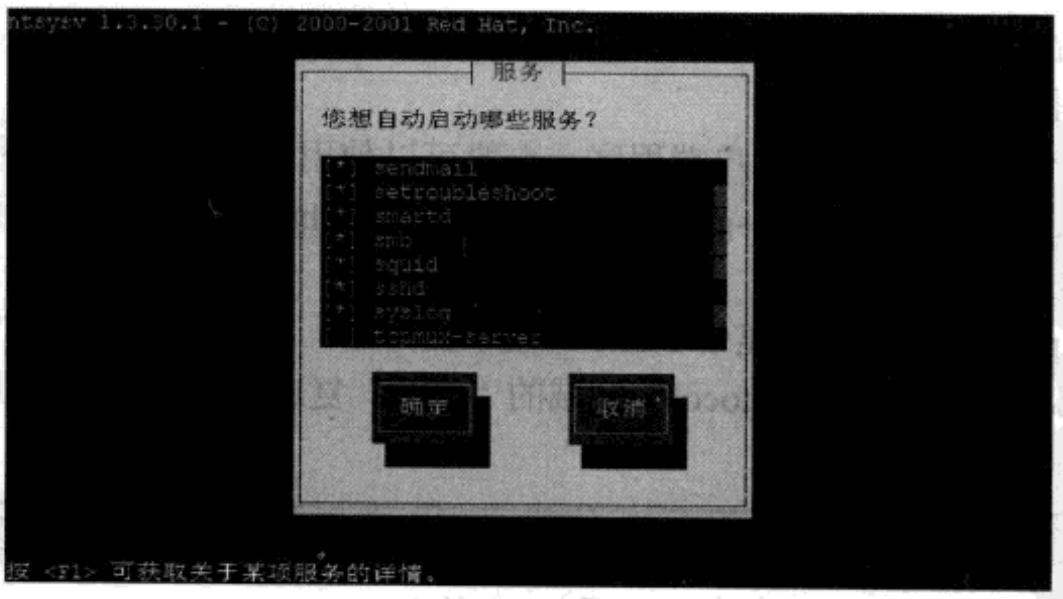


图 11-2 自动启动 SSH 服务

11.4.4 停止 SSH 服务

停止 SSH 服务，可以有两种不同的方式：通过 OpenSSH 服务器的启动脚本文件“/etc/rc.d/init.d/sshd”停止和使用 service 命令停止。

1. 通过 OpenSSH 服务器的启动脚本文件停止 SSH 服务

通过 OpenSSH 服务器的启动脚本文件“/etc/rc.d/init.d/sshd”停止 SSH 服务，命令执行如下。

```
[root@localhost ~]# /etc/rc.d/init.d/sshd stop
停止 sshd:                                                 [确定]
[root@localhost ~]#
```

命令执行后，SSH 服务停止运行。

2. 运用 Linux 下的 service 命令停止 SSH 服务

运用 service 命令停止 SSH 服务，命令执行如下。

```
[root@localhost ~]#service sshd stop
停止 sshd:                                                 [确定]
[root@localhost ~]#
```

命令执行后，SSH 服务停止运行。

11.5 使用 SSH 客户端

在设置并成功启动了 OpenSSH 服务器之后，客户端就可以通过不同的工具与服务器建立起连接。一般来说，SSH 客户端在不同的操作环境中的使用有一定的差异。

11.5.1 SSH 客户端在 Windows 平台下的使用

SSH 客户端在 Windows 平台下的使用步骤如下。

第 1 步，获取 PuTTY 程序。

Windows 下有许多 SSH 的客户端程序，一般可以使用免费的 PuTTY 程序。下载地址：http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html，页面如图 11-3 所示。

第 2 步，连接 SSH 服务器。

启动下载的 putty.exe 文件，显示 PuTTY 程序主界面，在“Host Name”文本框中输入服务器的 IP 地址或域名，选择“Protocol”区域的“SSH”复选按钮，最后单击“Open”按钮开始连接，如图 11-4 所示。

注意　在第一次连接到服务器的时候，PuTTY 程序显示警告信息和服务器信息，这是因为服务器公钥还没有缓存在注册表中，如图 11-5 所示。

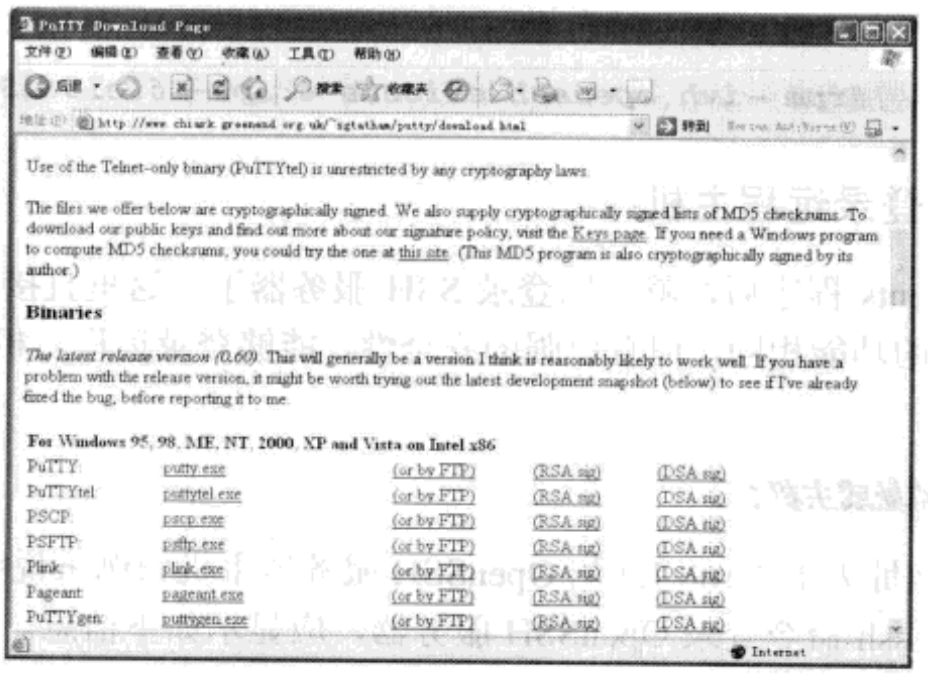


图 11-3　PuTTY 下载地址

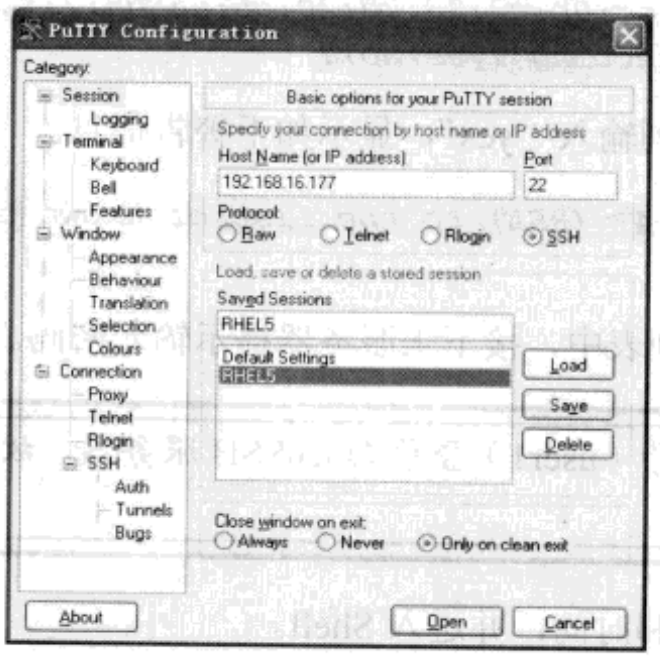


图 11-4　PuTTY 应用

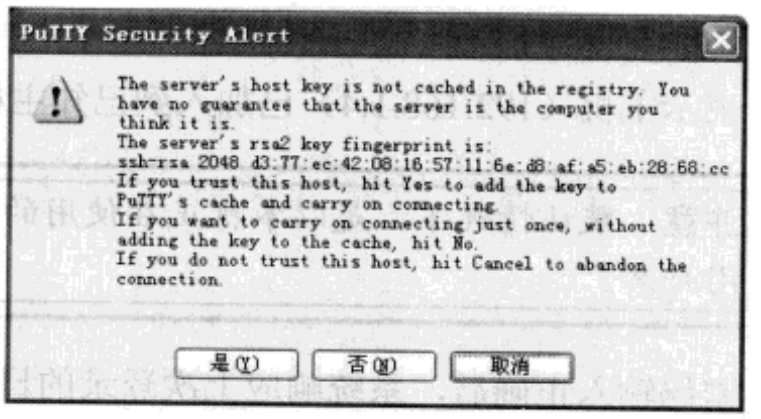


图 11-5　PuTTY 第一次连接到服务器

第 3 步，登录远程系统。

PuTTY 成功地连接到 SSH 服务器后，显示登录信息并提示用户输入用户名和口令，正确输入用户名和口令登录远程系统。

11.5.2 SSH 客户端在 Linux 平台下的使用

OpenSSH 的客户程序是 openssh-clients，用来连接 SSH 服务器。Red Hat Enterprise Linux5 已经默认安装了 SSH 客户程序，需要检查系统是否安装了 SSH 客户程序或查看安装版本，可以使用以下命令。

```
[root@localhost ~]#rpm -q openssh-clients
```

如果检查到系统未安装 SSH 客户程序，可在 Red Hat Enterprise Linux 5 的安装光盘 Server 目录下，找到 SSH 客户程序的 RPM 安装文件 openssh-clients-4.3p2-16.el5.i386.rpm，然后使用

以下命令安装。

```
[root@localhost ~]#rpm -ivh openssh-clients-4.3p2-16.el5.i386.rpm
```

1. 以 ssh 命令登录远程主机

安装好 openssh-clients 程序后，就可以登录 SSH 服务器了，这里直接使用 ssh 命令。ssh 与 rlogin、rsh 和 telnet 的功能相同，但有更强的安全性，能够登录远程主机并运行程序。以下是 ssh 的命令格式。

```
ssh  服务器的 IP 地址或主机名
```

下面将以登录 IP 地址为 192.168.1.4 的 OpenSSH 服务器主机为例来说明用户登录的信息。用户如果是第一次使用 ssh 命令登录 OpenSSH 服务器，应显示如下信息。

```
[root@localhost ~]# ssh 192.168.1.4
The authenticity of host '192.168.1.4 (192.168.1.4)' can't be established.
RSA key fingerprint is 2e:bd:5e:e6:2d:ec:0b:c3:38:7d:20:c5:aa:d8:73:1d.
Are you sure you want to continue connecting (yes/no)?
```

表示目前尚未建立主机的验证数据，要继续则输入“yes”，显示如下警告信息。

```
Warning: Permanently added '192.168.1.4' (RSA) to the list of known hosts.
root@192.168.1.4's password:
```

表示主机（192.168.1.4）已加入到已知主机列表中，接下来服务器提示输入密码。

注意 默认情况下，是以本机正在使用的账号（user1）登录 OpenSSH 服务器，故不要求输入用户名。

密码输入正确后，系统响应上次登录的日期和时间，并进入 Shell。

```
root@192.168.1.105's password:                    : #输入密码
Last login: Thu Jul  9 11:42:25 2009
[root@localhost ~]#
```

（1） 使用和现在不同的账号名登录

使用如下语法格式。

```
ssh -l 账号名 服务器的 IP 地址或主机名
```

或者

```
ssh 账号名@服务器的 IP 地址或主机名
```

实例 使用 testuser 账户名登录 192.168.1.105 服务器。

```
[root@localhost ~]# ssh -l testuser 192.168.1.105
testuser@192.168.1.105's password:
Last login: Fri May  8 11:26:09 2009 from 192.168.1.104
[testuser@localhost ~]$
```

（2） 允许用户在未登录远程主机的情况下，在远程主机上执行程序或命令
语法格式如下。

ssh 服务器的 IP 地址或主机名 命令

实例 不登录远程主机 192.168.1.105，在远程主机 192.168.1.105 中，运行“ls -l”命令。

```
[root@localhost ~]# ssh testuser@192.168.1.105 ls -l
Address 192.168.1.105 maps to localhost.localdomain, but this does not map back
to the address - POSSIBLE BREAK-IN ATTEMPT!
testuser@192.168.1.105's password:
总计 8
drwxr-xr-x 2 testuser testuser 4096 05-07 18:50 Desktop
[root@localhost ~]#
```

2. 以 sftp 命令与远程主机传输文件

如果安装了 OpenSSH 2.5.0p1 以上版本，就包含有比 ftp 安全性更高的 sftp 命令。sftp 命令使用时的语法如下。

sftp 账号名@服务器的 IP 地址或主机名

实例 使用 testuser 账号登录主机 192.168.1.105 的 OpenSSH 服务器。

```
[root@localhost ~]# sftp testuser@192.168.1.105
Connecting to 192.168.1.105...
Address 192.168.1.105 maps to localhost.localdomain, but this does not map back
to the address - POSSIBLE BREAK-IN ATTEMPT!
testuser@192.168.1.105's password:
sftp>
```

命令行出现“sftp>”提示符，说明成功登录了远程 OpenSSH 服务器。
（1） 使用 sftp 命令将本地文件上传到远程主机
语法格式如下。

sftp>put 本地文件名 远程文件名

实例 假设本地计算机当前目录下，包含一个名为“testroot.ls”的文件，使用 sftp 命令将其传送到远程主机 192.168.1.101 中，然后以“test_local.ls”的名称保存。

```
sftp> put testroot.ls test_local.ls
Uploading testroot.ls to /home/testuser/test_local.ls
testroot.ls                                   100%  127     0.1KB/s   00:00
sftp>
```

（2） 使用 sftp 命令将远程主机上的文件下载到本地主机
语法格式如下。

sftp>get 远程文件名 本地文件名

实例 假设远程主机 192.168.1.101 的当前目录下，包含一个名为“test_local.ls”的文件，使用 sftp 命令下载该文件到本地主机当前目录，并以“remote_1”的名称保存。

```
sftp> get test_local.ls remote_1
Fetching /home/testuser/test_local.ls to remote_1
/home/testuser/test_local.ls                  100%  127     0.1KB/s   00:00
sftp>
```

3. 以 scp 命令与远程主机传输文件

scp 命令是用来在主机间传送文件的，与 rcp 的功能一样，但是比 rcp 有更高的安全性。

（1） 使用 scp 命令将本地文件复制到远程主机

语法格式如下。

```
scp 本地文件 账号名@服务器的 IP 地址或主机名 /远程文件名
```

实例 假设本地计算机当前目录下，包含一个名为“remote_1”的文件，使用 scp 命令和将其传送到远程 192.168.1.105 主机中，并保存到“/home/sheriff”目录下。

```
[root@localhost ~]# scp remote_1 testuser@192.168.1.105: /home/testuser
Address 192.168.1.105 maps to localhost.localdomain, but this does not map back
to the address - POSSIBLE BREAK-IN ATTEMPT!
testuser@192.168.1.105's password:
scp: .: not a regular file
[root@localhost ~]# cd /home/testuser
[root@localhost testuser]# ls
remote_1
[root@localhost testuser]#
```

（2） 将远程主机上的文件复制到本地计算机

语法格式如下。

```
scp 账号名@服务器的 IP 地址或主机名 /远程文件名 本地文件名
```

实例 假设远程主机 192.168.1.105 的“/home/testuser”目录下包含一个各为“remote_1”的文件，使用 scp 命令和 testuser 账号，复制该文件到本地计算机的当前目录，并以“remote_1”的名称保存。

```
[root@localhost ~]# scp testuser@192.168.1.105:/home/testuser/remote_1
remote_1Address 192.168.1.105 maps to localhost.localdomain, but this does not map
back to the address - POSSIBLE BREAK-IN ATTEMPT!
testuser@192.168.1.105's password:
scp: /home/testuser/remote_1: No such file or directory
[root@localhost ~]# ls
 logsina.log  testfile~  anaconda-ks.cfg       remote_1    testroot.ls
Desktop                server rpm
[root@localhost ~]# cd
```

第 12 章　MySQL 数据库的配置

在 MySQL 出现之前，市面上大型的数据库不仅使用价格昂贵，而且运行效率较低。而 MySQL 则能提供一种快速、稳定而且易于使用的数据库解决方案，在其推出之后，受到了广大用户的欢迎，特别是受到一些中、小企业的青睐。这些企业在架设动态网站时通常采用 LAMP 的形式（即 Linux、Apache、MySQL 和 PHP 组合），采用此种方案来开发和架设 Web 站点，在稳定及安全的前提下不仅提高了效率，而且降低了企业的运营成本。

12.1　MySQL 概述

MySQL 是一个开放源码的小型关系型数据库管理系统，开发者为瑞典 MySQL AB 公司，最初是 Michael Widenius 为瑞典的 TcX 公司创建的名为 UNIREG 的数据库系统，随后由 Widenius 开发出了 MySQL 最初版本。

MySQL 是一个高性能、多线程、多用户的，建立在客户/服务器结构上的关系型数据库管理系统（RDBMS），目前被广泛地应用于 Internet 上的中、小型网站中。

12.1.1　MySQL 简介

MySQL 本身就很精简，并且具有开发源码、快速以及成本低的优点，广泛应用于许多中、小型网站中。选择 MySQL 作为数据库是降低成本的一个很好途径。有兴趣的朋友可以到 MySQL 官方网站（http://www.mysql.com）详细了解有关 MySQL 和 MySQL DB 的最新的消息，如图 12-1 所示。

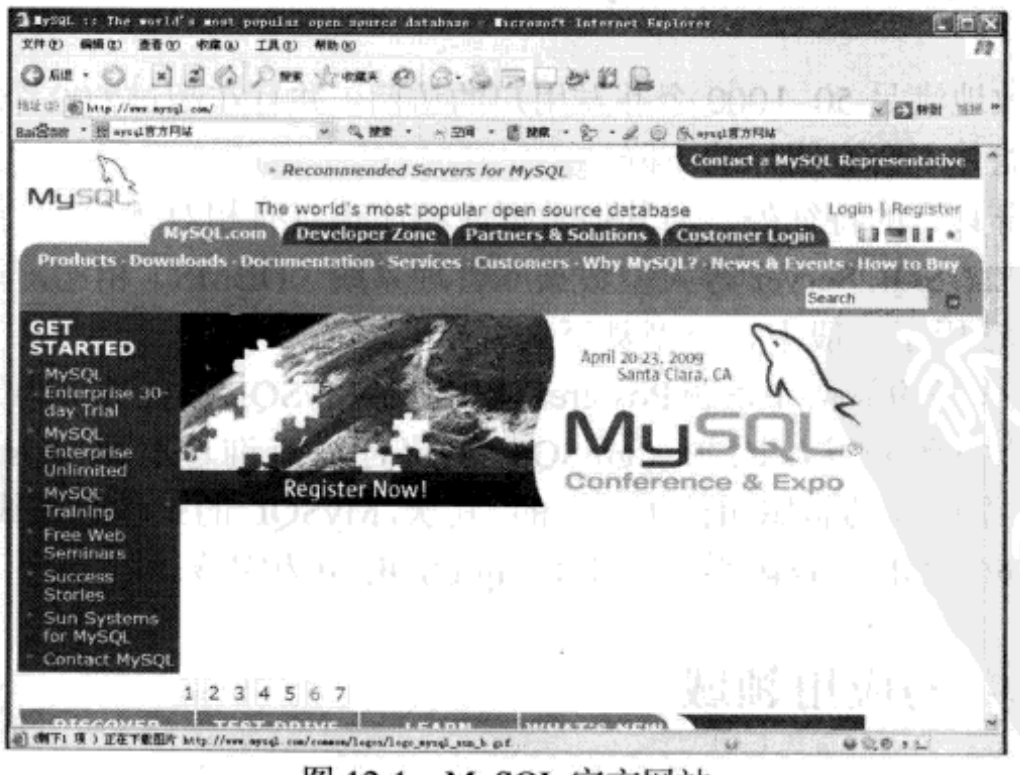


图 12-1　MySQL 官方网站

MySQL 具有以下特征。

- 开放源代码

MySQL 是开源产品，可以自由下载使用。

- 支持多操作系统

MySQL 可运行在不同的操作系统下。简单地说，MySQL 可以支持 Windows95/98/NT/2000 以及 UNIX、Linux 和 SUN OS 等多种操作系统平台。这意味着在一个操作系统中实现的应用可以很方便地移植到其他的操作系统下。

- 多线程

MySQL 的核心程序采用完全的多线程编程。线程是轻量级的进程，使用线程可以灵活地为用户提供服务，而不占用过多的系统资源。用多线程和 C 语言实现的 MySQL 能充分利用 CPU，并且可以采用多 CPU 体系结构。

- 具有较高安全性

MySQL 拥有十分灵活和安全的权限和密码系统，允许基于主机的验证。当客户连接到服务器时，所有的密码传输均采用加密形式，保证了密码安全。

- 高效稳定的性能

MySQL 拥有一个非常快速而且稳定的基于线程的内存分配系统，具有良好的稳定性。事实上，MySQL 的稳定性足以应付一个超大规模的数据库，因此 Yahoo、Google、Cisco、HP 和 NASA 都采用 MySQL 作为自己的数据库引擎。

- MySQL 支持大型的数据库处理

可以方便地支持上千万条的记录。作为一个开放源代码的数据库，MySQL 可以针对不同的应用进行相应的修改。

- 强大的查询功能

MySQL 支持查询使用的 SELECT 和 WHERE 语句及全部运算符和函数，并且可以在同一查询中混用来自不同数据库的表，从而使得查询变得快捷和方便。

- 多种应用程序支持

有 C、C++、JAVA、Perl、PHP 和 Python 等多种编程语言和 API 的支持。

- 多用户支持

MySQL 可有效地满足 50~1000 个并发用户的访问，并且在超过 600 个用户的限度的情况下，MySQL 的性能并不会有明显的下降。

MySQL 除本身具有的特征外，与其他数据库相比也具有相对优势。

与 Oracle、DB2、SQL Server 等大型数据库管理系统（DBMS）相比，其具有免费、小巧、查询快速的特点。适用于一般中、小型甚至大型企业。

与同属开放源代码的数据库系统 PostgreSQL 相比，MySQL 的优势更为突出。首先，MySQL 是由成熟的商业公司所开发，比 PostgreSQL 更流行，查询速度比 PostgreSQL 快许多，PostgreSQL 是由一个庞大的志愿团体所开发的。其次，MySQL 的工作级是线程，而 PostgreSQL 的工作级是进程。MySQL 在权限系统上比 PostgreSQL 更为完善。

12.1.2 MySQL 的应用领域

由于 MySQL 本身就是针对网络的数据库，因此多被用于 Internet 上。目前 MySQL 主要

应用在 Internet 网站、搜索引擎、电子商务、电子政务和网上教育等各个方面。

12.2 安装 MySQL

MySQL 既然如此优秀，有什么理由拒绝使用呢？接下来将学习在 Linux 操作系统中安装 MySQL 的具体方法和步骤。

12.2.1 安装前的准备

在安装 MySQL 之前需要进行一些准备工作。首先需要明确 Red Hat Enterprise Linux 5 中是否已经安装有 MySQL 服务，在不明确的情况下，可以执行以下命令进行确认。

```
[root@localhost ~]#rpm -q mysql
```

执行命令后的结果如下所示，表明 MySQL 服务没有安装。

```
[root@localhost ~]# rpm -q mysql
Package mysql is not installed
```

如果检测到 MySQL 服务没有安装，则需要准备相关的软件包文件，这些软件包在 Red Hat Enterprise Linux 5 的安装光盘中可以获得，安装 MySQL 服务所需要的 RPM 包文件如下。

```
perl-DBI-1.52-1.fc6.i386.rpm
perl-DBD-MySQL-3.0007-1.fc6.i386.rpm
openssl-devel-0.9.8b-10.el5.i386.rpm
e2fsprogs-devel-1.39-15.el5.i386.rpm
krb5-devel-1.6.1-25.el5.i386.rpm
zlib-devel-1.2.3-3.i386.rpm
```

MySQL 客户程序及共享库，其对应的 PRM 包文件如下。

```
mysql-5.0.22-2.1.i386.rpm
mysql-server-5.0.22-2.1.i386.rpm
mysql-bench-5.0.22-2.1.i386.rpm
mysql-devel-5.0.22-2.1.i386.rpm
```

12.2.2 MySQL 的具体安装过程

安装 MySQL 的方法有多种，下面就以 RPM 包安装和源码安装两种方式为例进行讲解。

1. RPM 包安装方式

要安装上述 RPM 包文件，可将 Red Hat Enterprise Linux 5 的安装盘放入光驱，加载光驱后在光盘的 Server 目录下找到相应的 RPM 安装包文件，然后执行 rpm -ivh 命令进行安装。

```
[root@localhost ~]#rpm -ivh perl-DBI-1.52-1.fc6.i386.rpm
[root@localhost ~]#rpm -ivh mysql-5.0.22-2.1.i386.rpm
[root@localhost ~]#rpm -ivh perl-DBD-MySQL-3.0007-1.fc6.i386.rpm
```

```
[root@localhost ~]#rpm -ivh krb5-devel-1.6.1-25.el5.i386.rpm
[root@localhost ~]#rpm -ivh zlib-devel-1.2.3-3.i386.rpm
[root@localhost ~]#rpm -ivh openssl-devel-0.9.8b-10.el5.i386.rpm
[root@localhost ~]#rpm -ivh e2fsprogs-devel-1.39-15.el5.i386.rpm
[root@localhost ~]#rpm -ivh mysql-server-5.0.22-2.1.i386.rpm
[root@localhost ~]#rpm -ivh mysql-bench-5.0.22-2.1.i386.rpm
[root@localhost ~]#rpm -ivh mysql-devel-5.0.22-2.1.i386.rpm
```

2. 源码安装方式

要使用源码的方式对 MySQL 数据库进行安装，则需要先从相关网站下载获得相应的 MySQL 安装包文件（mysql-5.0.15.tar.gz），然后进行相关的安装工作，安装的具体步骤如下。

第 1 步，为“mysqld”服务增添一个登录组和用户名，执行如下命令。

```
[root@localhost ~]#groupadd mysql
[root@localhost ~]#useradd -g mysql mysql
```

第 2 步，解压 MySQL 安装包，执行如下命令。

```
[root@localhost ~]#gunzip < mysql-5.0.15.tar.gz | tar -xvf -
[root@localhost ~]#cd mysql-5.0.15
```

第 3 步，进行相关的配置和编译。

给 configure 分配可执行文件，执行如下命令。

```
[root@localhost ~]#chmod +x configure
```

改变字符集为 GBK [默认字符集为 ISO-8859-1(Latin1)]，执行如下命令。

```
[root@localhost ~]#./configure --prefix=/usr/local/mysql --with-charset=gbk
[root@localhost ~]#make
```

第 4 步，执行如下命令进行安装。

```
[root@localhost ~]#make install
```

另外，如果想安装选项文件，使用当前存在的‘support-files’文件夹作为模板，执行如下指令。

```
[root@localhost ~]#cp support-files/my-medium.cnf /etc/my.cnf
```

同时，如果需要让 MySQL 每次开机时自动启动，需要执行以下指令。

```
[root@localhost ~]#cp -r support-files/mysql.server /etc/init.d/mysql
[root@localhost ~]#cd /etc/rc.d/init.d
[root@localhost ~]#chmod +x mysql
[root@localhost ~]#/sbin/chkconfig --del mysql
[root@localhost ~]#/sbin/chkconfig --add mysql
```

第 5 步，执行以下命令进入安装目录。

```
[root@localhost ~]#cd /usr/local/mysql
```

注意　如果在之前没有安装过 MySQL，必须创建一个 MySQL 准许表，具体命令如下。

```
[root@localhost ~]#bin/mysql_install_db --user=mysql
```

第 6 步，将程序的所有权限授给“root”，并且把数据目录的所有权授给可以运行“mysqld”的用户，假设 MySQL 的安装目录为“/usr/local/mysql”，执行以下命令。

首先把文件拥有权授给“root”，执行如下命令。

```
[root@localhost ~]#chown -R root
```

然后把数据目录拥护权授给“mysql”用户，执行如下命令。

```
[root@localhost ~]#chown -R mysql var
```

最后把组的权限授给“mysql”组，执行如下命令。

```
[root@localhost ~]#chgrp -R mysql
```

第 7 步，所有的配置完成后，执行以下命令测试并运行 MySQL。

```
[root@localhost ~]#/usr/local/mysql/bin/mysqld_safe --user=mysql &
[root@localhost ~]#service mysql start
```

第 8 步，测试一切正常后，为了安全起见，最好更改管理员的密码。可以运行 mysqladmin，执行如下命令。

```
[root@localhost ~]#cd bin
[root@localhost bin]#./mysqladmin -u root password **********
```

注意　此命令不一定在你的路径中，因此最好是转到此命令的目录中直接执行。

第 9 步，允许其他用户访问本机，执行如下命令。

```
[root@localhost ~]#./mysql -u root -p mysql
```

这里输入刚才更改的管理员密码。

```
mysql> update user set host='' where host='localhost' and user='root';
mysql> exit
Bye
```

第 10 步，修改 MySQL 数据库端口号，执行如下命令。

```
[root@localhost ~]#vi /etc/my.cnf
```

修改里面的“**** ”为需要的端口号。

第 11 步，重启应用，执行如下命令。

```
[root@localhost ~]#service mysql restart
```

以上安装方法即为 MySQL 的源码安装方式，这两种安装方式都稍显复杂，建议在“软件包管理者”窗口中进行安装。

12.3 启动与停止 MySQLD 服务

Red Hat Enterprise Linux 5 中内置的 MySQL 程序是以服务的方式运行的，遵循一般的服务启动、停止规范，接下来我们将学习如何启动与停止 MySQLD 服务。

要启动 MySQLD 服务，只需执行以下命令。

```
[root@localhost ~]#service mysqld start
```

如果 MySQL 服务是第一次启动，此时将会创建正常运行所需要的各种数据库和表格。

停止 MySQLD 服务的操作与启动 MySQLD 服务的操作类似，只需执行以下命令。

```
[root@localhost ~]#service mysqld stop
```

同样，重启 MySQL 服务的操作也类似于其启动操作，具体指令如下。

```
[root@localhost ~]#service mysqld restart
```

通过此种方式启动的 MySQLD 服务只能运行到计算机关机之前，下一次系统重新启动后又需要重新启动 MySQLD 服务。为了方便使用该服务，可以配置其随系统启动而自动运行，具体方法如下。

在“终端”窗口输入“ntsysv”命令来启动自动服务配置小程序。找到“mysqld”服务，并按下 Space 键在其前面加上“*”星号，然后按 Tab 键，单击“OK”按钮即可，如图 12-2 所示。

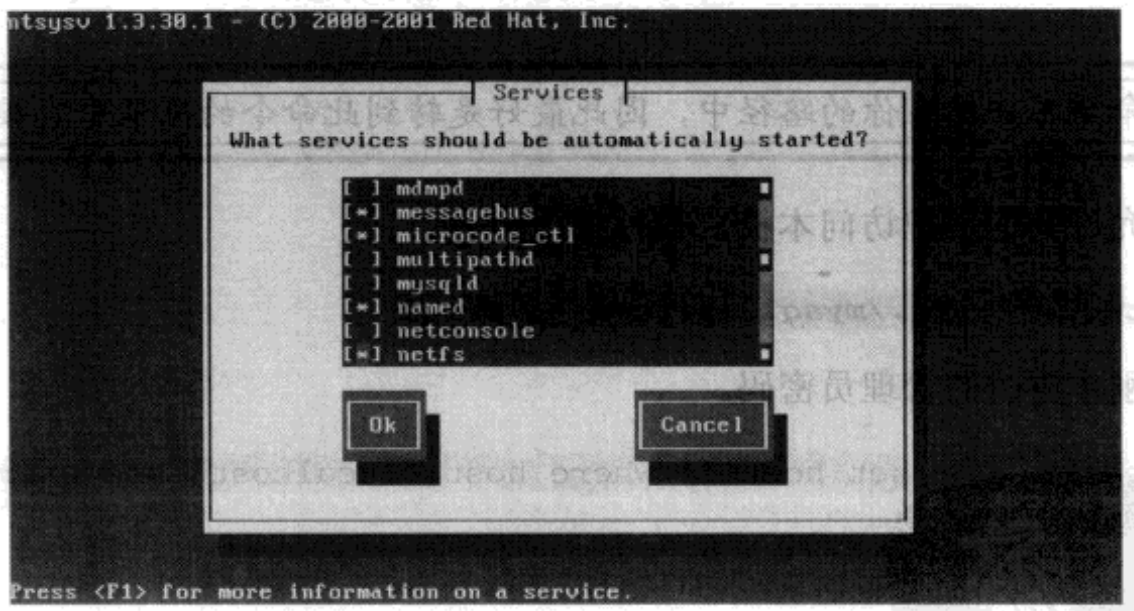


图 12-2 设置 MySQL 服务为自动运行

12.4 配置及使用 MySQL 服务

MySQL 数据库管理系统的相关配置及操作非常多，且都是通过命令行实现的。因此，如果想要熟练操作此数据库系统，学习好必要的命令是必不可少的，接下来将对这些命令进行讲

解。

12.4.1 修改 MySQL 管理员的口令

MySQL 安装成功后，MySQL 管理员默认为 root 用户，但此 root 用户不同于 Linux 系统的 root 用户，此管理员的初始密码为空。当客户端连接 MySQL 时，只需键入命令“mysql”即可，安全性不够高。为了增强安全性，需要修改 MySQL 管理员的密码，修改 MySQL 管理员密码的方法有多种，具体方法如下。

方法 1 直接创建 root 密码

直接创建 root 密码的指令如下。

```
[root@localhost ~]#mysqladmin -u root password new_password
```

注意 如果 root 账号已经设置了密码，则需要执行如下指令进行密码修改。

```
[root@localhost ~]#mysqladmin -u root password new_password -p
```

方法 2 使用 UPDATE 命令进行修改

首先登录 MySQL，然后打开 MySQL 数据库，执行如下命令。

```
[root@localhost ~]#mysql -u root -p
mysql>use mysql;
```

然后修改 root 账号密码，执行如下命令。

```
mysql>UPDATE user SET password=PASSWORD ('new_password')
    where user='root';
```

注意 因为 MySQL 现在使用的是加密的密码，所以一定要使用 PASSWORD() 函数来生成密码，而不是直接使用 password= ‘new_password’ 来设置。

最后需要执行以下指令使修改生效。

```
mysql>FLUSH PRIVILEGES;
```

方法 3 使用 mysqladmin 命令进行修改

登录 MySQL，执行如下命令。

```
[root@localhost ~]#mysql -u root -p
```

然后执行以下命令修改 root 账号密码。

```
mysql> SET PASSWORD = PASSWORD('new_password');
```

12.4.2 MySQL 数据库的创建和使用

任何一种数据库，其应用的基础都是要学会如何在数据库系统中创建、使用和删除数据

库，MySQL 也不例外。MySQL 可以使用 SQL 的数据定义语言（DDL）来创建、删除数据库。接下来将以实例的形式向大家讲解如何创建、使用和删除数据库。

1. 创建数据库

在 MySQL 中创建数据库的 SQL 语句格式如下。

```
create database 数据库名;
```

实例 创建一个名为 BBSTestDB 的数据库。

```
mysql>CREATE DATABASE BBSTestDB;
```

2. 选择数据库

选择一个数据库成为所有事务的当前数据库的 SQL 语句格式如下。

```
use 数据库名称;
```

实例 选择一个名为 BBSTestDB 的数据库。

```
mysql>USE BBSTestDB;
```

3. 删除数据库

在 MySQL 中删除数据库的 SQL 语句格式如下。

```
drop database 数据库名称;
```

实例 删除一个名为 BBSTestDB 的数据库。

```
mysql>drop database BBSTestDB;
```

12.4.3 MySQL 数据表的创建和使用

如果说创建数据库是基础，那么创建数据表则是关键。创建完数据库后，可以在该数据库中创建一张或多张数据表。这是因为，一个数据库由一张或多张用来存储数据的数据表组成，每张数据库表则由行和列组成，每一行为一条记录，每个记录包含多个列（字段）。

在 MySQL 中，对数据表的操作包括创建表、修改表以及删除表等，下面将学习这些操作的具体实现方法。

1. 创建表

在 MySQL 中创建表格的格式如下。

```
CREATE TABLE 表名称（
字段 1 数据类型 属性 默认值 其他，
字段 2 数据类型 属性 默认值 其他，
...
）
```

其中，数据类型是指规定某个字段所允许输入的数据的类型，几种常用的 MySQL 支持的数据类型如表 12-1 所示。

表 12-1　常见的几种 MySQL 支持的数据类型

数 据 类 型	具 体 描 述
INT	允许值在 0~232-1（无符号）或 231~231-1（有符号）之间的标准整数
DOUBLE	双精度浮点数
CHAR	最大长度不超过 255 个字符的定长字符串
VARCHAR	最大长度不超过 255 个字符的变长字符串
TEXT	最大长度为 65 535 个字符的字符串
BLOB	可变数据的二进制类型
DATE	YYYYMMDD 格式表示的日期值

字段约束指的是进一步对某个字段所允许输入的数据进行约束，几种常用的字段约束如表 12-2 所示。

表 12-2　几种常用的字段约束

字 段 约 束	具 体 描 述
PRIMARY KEY	为表指定主键
FOREIGN KEY…REFERENCES	为表指定外键
INDEX	创建索引
UNIQUE	为某个字段建立索引，该字段的值必须唯一
FULLTEXT	为某个字段建立全文索引

表级约束是指指定表的主键、外键、索引和唯一约束，具体信息如表 12-3 所示。

表 12-3　表级约束

表 约 束	具 体 描 述
PRIMARY KEY	为表指定主键
FOREIGN KEY…REFERENCES	为表指定外键
INDEX	创建索引
UNIQUE	为某个字段建立索引，该字段的值必须唯一
FULLTEXT	为某个字段建立全文索引

实例　可以通过以下步骤实现在刚才创建的 BBSTestDB 数据库中创建数据表。

第 1 步，登录 MySQL，执行如下命令。

```
[root@localhost ~]#mysql -u root -p
```

第 2 步，使用刚才新创建的数据库 BBSTestDB。

```
mysql>use BBSTestDB;
```

此时进入创建表过程，假设要创建一个有关用户信息的表，包括用户 ID、姓名、出生年月、电话号码和序号等字段。执行以下命令。

```
mysql>Create Table userInfo (
id int(4) DEFAULT '0' NOT NULL,
```

```
name char(10) DEFAULT '' NOT NULL,
birth_day char(8) DEFAULT '' NOT NULL,
telephone char(60) DEFAULT '' NOT NULL,
KEY id(id)
);
```

2. 复制表

在 MySQL4.1 中，复制表结构的 SQL 语句如下。

```
mysql>create table 新表名称 like 源表名称;
```

实例 复制表 userInfo 为另一个表并以 userInfo1 命名，可使用以下命令。

```
mysql>create table userInfo like userInfo1;
```

3. 在表中插入记录

表格创建好之后，可以根据需要使用 Insert 指令来添加数据，语法格式如下。

```
INSERT INTO 数据表名称 VALUES(字段 1 的值，字段 2 的值……)
```

实例 向数据表 userInfo 插入一条用户信息的记录。

```
mysql>INSERT INTO userInfo VALUES(1, 'ss', '86/10/22' ,'02365106900')
```

这里需要说明的是，在添加表记录时，可以只添加特定字段的值。

```
mysql>INSERT INTO userInfo (id,name, telphone)
VALUES(
        1,
        'jackel',
'13569785425'
);
```

注意 为了以后操作直观，建议多插入几条数据。

4. 修改记录

如果用户需要修改指定的数据，则可以通过 Update 指令来完成，但 Update 指令的操作对象分单表与多表的情况，具体语法格式如下。

单表语法如下。

```
UPDATE [LOW_PRIORITY] [IGNORE] 数据表名称
    SET 字段 1=表达式 1 [, 字段 2=表达式 2 ...]
    [WHERE 条件表达式]
    [ORDER BY ...]
    [LIMIT row_count]
```

多表语法如下。

```
UPDATE [LOW_PRIORITY] [IGNORE] 参照数据表集
    SET 字段1=表达式1 [, 字段2=表达式2 ...]
    [WHERE 条件表达式]
```

实例 执行包括多个表的 Update 操作。

```
mysql> UPDATE 表1,表2 SET 表1.列名1=表2.列名1
WHERE 表1.列名2=表2.列名2;
```

注意 Update 语法可以用新值更新原有表行中的各列；SET 子句指示要修改哪些列和要赋予的值；WHERE 子句指定应更新哪些行，如果没有 WHERE 子句，则更新所有的行；如果指定了 ORDER BY 子句，则按照被指定的顺序对行进行更新；LIMIT 子句用于给定一个限值，限制可以被更新的行的数目。

5. 删除表中记录

在已经创建好的表格中，用户可以通过 Delete 指令来删除一些指定的数据，Delete 指令的操作对象也分单表、多表的情况，删除表记录的具体语法格式如下。

单表语法如下。

```
DELETE [LOW_PRIORITY] [QUICK] [IGNORE] FROM 数据表名称
    [WHERE 条件表达式]
    [ORDER BY ...]
    [LIMIT row_count]
```

多表语法如下。

```
DELETE [LOW_PRIORITY] [QUICK] [IGNORE]
    数据表名称1[.*] [,数据表名称2[.*] ...]
    FROM 数据表参照集
    [WHERE 条件表达式]
```

或

```
DELETE [LOW_PRIORITY] [QUICK] [IGNORE]
    FROM 数据表名称1[.*] [,数据表名称2[.*] ...]
    USING 参照数据表集
    [WHERE 条件表达式]
```

实例 以下子句用于查找与 WHERE 子句对应的行，并删除该行。

```
mysql>DELETE FROM userinfo
WHERE user = ' kristy '
```

注意 另一种删除表中所有记录的方法是使用如下的 truncate 命令。

```
mysql> truncate table 表名;
```

6. 修改表

在 MySQL 中修改已经创建的表，语法格式如下。

```
ALTER TABLE 表名称
ADD Column 字段 1 数据类型 属性 默认值 其他
ADD INDEX 索引名 索引类型 （索引字段，…）
ADD CONSTRAINT
ALTER Column 字段 1 {SET DEFAULT literal | DROP DEFAULT}
CHANGE Column 旧字段 1 新字段 1 数据类型 属性 默认值 其他
MODIFY Column 字段 1 数据类型 属性 默认值 其他
DROP Column 字段名
DROP PRIMARY KEY
DROP INDEX 索引名
DROP FOREIGN KEY 外键名
```

注意 ALTER TABLE 用于更改原有表的结构：可以增加或删减列、创建或取消索引、更改原有列的类型、重新命名列或表，还可以更改表的评注和表的类型。

实例 可以使用如下方法在刚才创建的 TestDB 数据库中修改表格 userInfo，取消 userInfo 表格中的 telephone 列，具体实现步骤如下。

第 1 步，登录 MySQL，执行如下命令。

```
[root@localhost ~]#mysql -u root -p
```

第 2 步，打开数据库，执行如下命令。

打开刚才新创建的数据库 testDB

```
mysql>use TestDB;
```

第 3 步，取消电话号码列，执行如下命令。

```
mysql>Alter Table userInfo
Drop Column Telephone;
```

7. 删除表

MySQL 中删除一个或多个表的 SQL 语句格式如下。

```
drop table 表名称 1[, 表名称 2, …];
```

实例 删除名为 userInfo 的表。

```
mysql>drop Table userInfo;
```

12.4.4 索引的创建和删除

大家知道，在查询的数据量较少的情况下，利用数据库来实现查询数据是一件很方便的

事情，但如果查询的数据量庞大，则会影响查询效率。另外，MySQL 还可以对单列或者多列的组合进行索引。为了节约查询数据的时间可以为数据库创建索引，接下来将学习索引的创建与删除。

1. 创建索引

创建索引的方法有创建表的同时创建索引与向已存在的表添加索引两种方法。

（1） 创建表的同时创建索引

使用 Primary Key 子句为创建的表格指定一个主键 sno，MySQL 就会自动为该表的 sno 字段创建索引。另外，在创建表时，还可以用 INDEX 子句或 UNIQUE 子句来创建索引。

注意 如果存在子句 index name_index（name）并将其修改为 unique（name），则表示创建的是 unique 索引，该索引要求索引字段中的值必须是唯一的，即表中各条记录中该字段的值不能相同。

（2） 向已存在的表添加索引

已存在的表格要添加索引，可以用 CREATE INDEX 语句完成，语法格式如下。

```
mysql>CREATE [UNIQUE] INDEX 索引名 ON 表名称 (字段名1 [(长度)], …);
```

实例 为表 userInfo 的 name 字段创建名为 name_index 的索引，可以使用以下命令。

```
mysql>create index sna on student(sname);
```

2. 删除索引

如果不需要索引，则可使用 DROP INDEX 语句删除，语句的格式如下。

```
mysql>DROP INDEX 索引名 ON 表名称
```

实例 删除表 userInfo 的 name 字段的 name_index 索引，可以使用以下命令。

```
mysql>drop index name_index on userInfo;
```

12.4.5 用户的创建和删除

在 MySQL 数据库中，对于操作安全方面的控制主要是通过用户来进行的，因此用户的添加与删除显得很重要，接下来同样以实例的方式向大家讲解如何创建用户。

1. 创建新用户

在 MySQL 中，添加用户的方法有多种，主要有以下两种方式可以进行用户的添加。

方法 1 直接修改表格与权限相关的部分。

修改表格与权限相关的部分，具体步骤如下。

第 1 步，登录 MySQL，并使用 MySQL 数据库，执行以下指令。

```
[root@localhost ~]#mysql -u root -p
mysql>use mysql;
```

第 2 步，新建用户，执行如下命令。

```
mysql>INSERT INTO user VALUES('localhost','lily',PASSWORD('000000'),'
 Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y');
```

方法 2 使用 GRANT 指令来增加用户。

例如，要新添加一位账户名为 Crazy 的用户，密码为 123456，并使其拥有所有的权限，可以通过以下步骤实现。

第 1 步，登录 MySQL，并使用 MySQL 数据库，执行如下命令。

```
[root@localhost ~]#mysql -u root -p
mysql>use mysql;
```

第 2 步，使用 GRANT 语句来添加新用户 Crazy，执行如下命令。

```
mysql>GRANT ALL PRIVILEGES
    on service_db.*
    to Crazy IDENTIFIED BY
    '123456' ;
```

另外，如果只给 Crazy 用户 SELECT、INSERT 和 UPDATE 权限，则可以通过以下指令实现。

```
mysql>GRANT SELECT, INSERT, UPDATE
    on service_db.*
    to CrazyIDENTIFIED BY
    '123456';
```

2. 更改用户密码

在 MySQL 中除了可以使用 update 语句修改用户密码外，还有一种更简单的更改用户密码的方法，即使用 set 语句。

实例 要将用户 kristy 的密码改为“123456”，可执行以下命令。

```
mysql>set password for kristy@'%'=password('123456');
```

3. 删除用户

MySQL 中可以使用 delete 指令来实现对用户的删除操作。

实例 要删除 kristy 用户，则可以执行以下指令。

```
mysql> delete from userInfo where user='kristy';
```

12.4.6 用户权限的设置

在 MySQL 中存在授权表，而该授权表主要是用来控制用户连接数据库服务器和访问数据

库的权限的，MySQL 授权表中权限字段有以下两种类型。

- ENUM

在表 user、db 和 host 中，所有权限字段都被声明为 ENUM（“N”，“Y”），表 12-4 列出了表 user、db 和 host 中可设置权限的字段。

注意　ENUM（“N”，“Y”），即每一个权限字段值都可以被设置为“N”或“Y”，默认值为“N”，表示不具备该权限。

表 12-4　表 user、db 和 host 中可设置权限的字段

字　段	权　限　名	拥有该权限的用户可执行的操作
select_priv	SELECT	数据查询
insert_priv	INSERT	数据插入
update_priv	UPDATE	数据修改
delete_priv	DELETE	数据删除
create_priv	CREATE	创建数据库和表
drop_priv	DROP	删除数据库和表
reload_priv	RELOAD	重载/刷新 MySQL 服务器
Shutdown_priv	SHUTDOWN	关闭正在运行的 MySQL 服务器
process_priv	PROCESS	在 MySQL 服务器上跟踪一项活动
file_priv	FILE	在服务器上读写文件
grant_priv	GRANT	授予其他用户拥有该用户拥有的权限
references_priv	REFERENCES	创建、编辑和删除外键引用
index_priv	INDEX	创建、编辑和删除表索引
alter_priv	ALTER	修改表
show_db_priv	SHOW DATABASES	显示可使用的数据库
super_priv	SUPER	执行管理命令
create_tmp_table_priv	CREATE TEMPORARY TABLES	创建临时表
lock_tables_priv	LOCK TABLES	创建和删除表锁定
execute_priv	EXECUTE	执行被保存的程序
repl_slave_priv	REPLICATION SLAVE	在复制内容中读主二进制日志
repl_client_priv	REPLICATION CLIENT	在复制内容中申请主、副信息

注意　字段 select_priv、insert_priv、update_priv、delete_priv、create_priv、drop_priv、index_priv、alter_priv、crete_tmp_table_priv、lock_tables_priv 为授权表 user、db 和 host 共有的字段。

- SET

在表 tables_priv 和 columns_priv 中，权限字段被声明为 SET 类型。表 12-5 列出了表 tables_priv 和 columns_priv 中权限字段及可设置的权限。

表 12-5　表 tables_priv 和 columns_priv 中权限字段及可设置的权限

表　名	字　段　名	权限集合中的成员
tables_priv	table_priv	SELECT、INSERT、UPDATE、DELETE、CREATE、DROP、GRANT、REFERENCES、INDEX、ALTER
	column_priv	SELECT、INSERT、UPDATE、REFERENCES
columns_priv	column_priv	SELECT、INSERT、UPDATE、REFERENCES

MySQL 提供了两种修改授权表中的访问权限的方法：一种是使用 insert、update 和 delete 等 dml 语句手工修改表中的信息；另一种是用 grant 和 revoke 语句。为了避免出错，建议使用后者。

1. 使用 grant 语句授权

使用 grant 语句授权的格式如下。

```
GRANT 权限列表[(字段列表)] [,类型 [(字段列表)] ...]
ON 数据库名称.表名称
TO 用户名@域名或 IP 地址
[IDENTIFIED BY '密码值'][WITH GRANT OPTION]
[WITH GRANT OPTION]
```

实例 授予 cici 用户可以从任意主机连接到数据库服务器，并具有完全访问据库 BBSTestDB 的权限，可执行以下命令。

```
mysql>grant all on BBSTestDB.* to cici@'%' identified by '123456';
```

注意 使用 grant 语句时，可以使用通配符。grant 授权语句中的“权限列表”使用关键字 ALL，表示授予全部的权限，此外，还可以使用关键字 USAGE，表示不授予任何权限。使用 grant 授予权限时，如果指定的用户名不存在，则 MySQL 会自动创建这个新用户。

2. 使用 revoke 语句撤权

revoke 语句的基本格式如下。

```
REVOKE 权限类型 [(字段列表)] [, 权限类型 [(字段列表)] ...]
ON {数据库名称.表名称}
FROM 用户名@域名或 IP 地址
```

实例 管理员撤销用户 cici 对数据库 BBSTestDB 所拥有的创建数据库及表的权限，并撤销该用户可以把自己所拥有的权限授予其他用户的权限，可执行以下命令。

```
mysql>revoke create,drop on BBSTestDB.* from cici;
mysql>revoke grant option on BBSTestDB.* from cici;
```

注意 revoke 语句中的“用户名@域名或 IP 地址”部分必须匹配原来 grant 语句中的“用户名@域名或 IP 地址”部分，而“权限类型”部分可以是所授权的一部分权限。而且，revoke 只能撤销权限，不能删除用户账户。在授权表 user 中仍保留该用户的记录。用户仍可以连接到数据库服务器，如果要完全删除用户，则使用前面提到的 delete 语句从 user 表中删除该用户记录。

12.5 图形化配置 MySQL

一般情况下，对 MySQL 的操作都是通过命令行的方式来进行的，其配置文件也是采用命令行和基于文本模式的，因此对于习惯图形界面的用户而言，对 MySQL 的操作及管理都比较困难。为了方便用户对 MySQL 数据库的操作及管理，一些图形化管理工具也随着产生，其中 phpMyAdmin 应用得最为广泛。其官方站点为：http://www.phpmyadmin.net，如图 12-3 所示，通过该站点可以获得 phpMyAdmin 软件包。

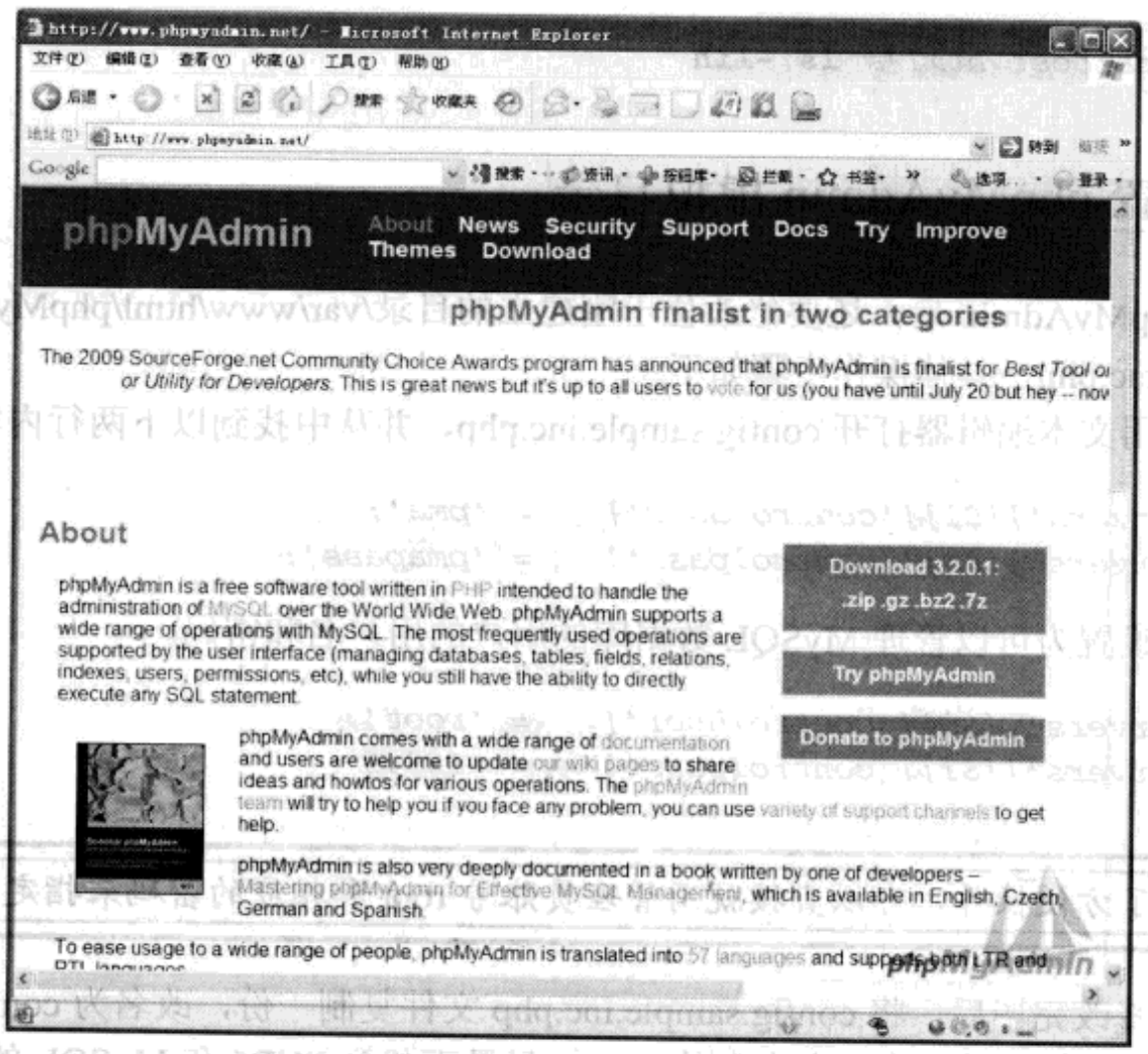


图 12-3 phpMyAdmin 的官方网站

12.5.1 安装 phpMyAdmin

phpMyAdmin 是一种使用 PHP 开发的基于浏览器/服务器（B/S）模式的开放源代码的工具，获得相应的安装软件包后即可进行安装，安装 phpMyAdmin 的方法比较简单，具体步骤如下。

第 1 步，获得相应安装文件后（这里使用的是 phpMyAdmin-2.11.6-all-languages.tar.gz），需要解压缩文件，执行如下命令。

```
[root@localhost ~]#tar zxvf  phpMyAdmin-2.11.6-all-languages.tar.gz   -C
/var/www/html
```

注意　执行命令后，将在/var/www/html 目录下生成一个子目录 phpMyAdmin-2.11.6-all-languages。

第 2 步，重命名该目录为 phpMyAdmin，执行如下命令。

```
[root@localhost ~]# cd /var/www/html
[root@localhost html]# mv phpMyAdmin-2.11.6-all-languages phpMyAdmin
```

最后执行以下命令查看结果。

```
[root@localhost html]# ls -lih
```

12.5.2 配置 phpMyAdmin 虚拟目录

安装好 phpMyAdmin 后，还要修改位于刚建立的目录/var/www/html/phpMyAdmin 中的配置文件 config.inc.php，具体操作步骤如下。

第 1 步，用文本编辑器打开 config.sample.inc.php，并从中找到以下两行内容。

```
$cfg['Servers'][$i]['controluser']    = 'pma';
$cfg['Servers'][$i]['controlpass']    = 'pmapass';
```

然后将其设置为可以管理 MySQL 数据库的用户的用户名和密码。

```
$cfg['Servers'][$i]['controluser']    = 'root';
$cfg['Servers'][$i]['controlpass']    = '12345
```

注意　为了方便操作，可以直接就用管理员账号 root 和相应的密码来指定。

第 2 步，修改完毕后，将 config.sample.inc.php 文件复制一份，改名为 config.inc.php。

此时，还需挂载安装光盘，在光盘的 Server 目录下找到 PHP5 和 MySQL 的扩展库两个软件包并进行安装，具体指令如下。

```
[root@localhost ~]#rpm -ivh php-pdo-5.1.6-5.e15.i386.rpm
[root@localhost ~]#rpm -ivh php-mysql-5.1.6-5.e15.i386.rpm
```

第 3 步，用文本编辑器打开/etc/php.ini 文件，找到以下一行。

```
; extension=msqlso
```

将其改为如下内容。

```
extension=mysql.so
```

第 4 步，完成了上述设置后，要测试 phpMyAdmin 能否正常工作，可在客户端浏览器的地址栏中输入“http://服务器 IP 地址/phpMyAdmin”。如果配置无误，则可以看到 phpMyAdmin 管理界面了，如图 12-4 所示。

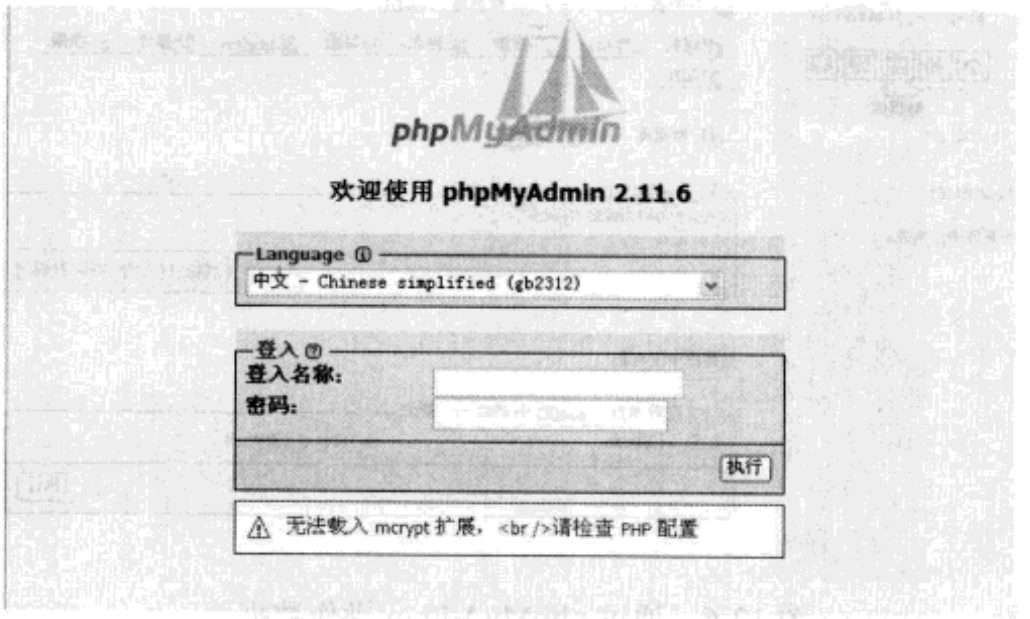


图 12-4　phpMyAdmin 管理界面

12.5.3 使用 phpMyAdmin

phpMyAdmin 成功安装并配置好后，即可用它来进行数据库的访问。使用 phpMyAdmin 工具可以方便地创建、管理数据库和表格，使用该工具的具体步骤如下。

第 1 步，进入 phpMyAdmin 的登录页面，在该页面输入正确的 MySQL 数据库的账号和密码后，即可进入 phpMyAdmin 主页面。

第 2 步，在登录后的 phpMyAdmin 主页面的“创建一个新的数据库”文本框中输入相应数据库的名字，然后单击“创建”按钮即可完成数据库的创建，如图 12-5 所示。

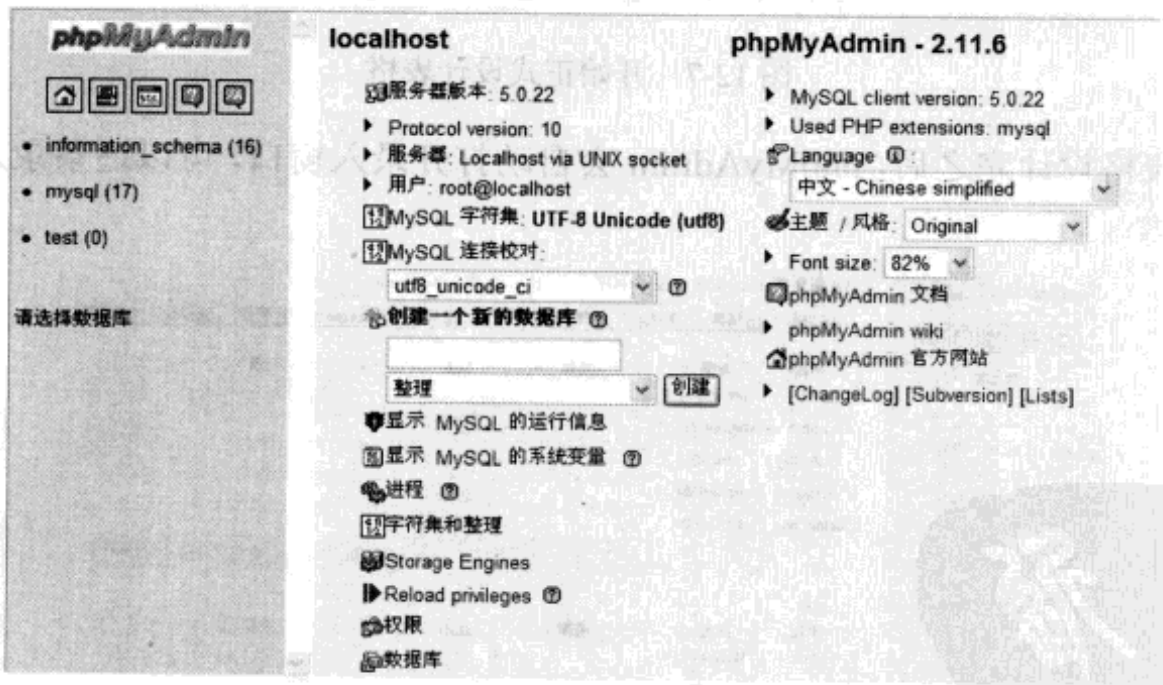


图 12-5　创建数据库

第 3 步，创建数据库 TestDB 后，phpMyAdmin 会自动展开，在下方的“在数据库 TestDB 中创建一个新表”文本框中输入表格的名字，比如 userInfo，后面定义表格的字段数，比如 5，单击“执行”按钮开始创建表格，如图 12-6 所示。

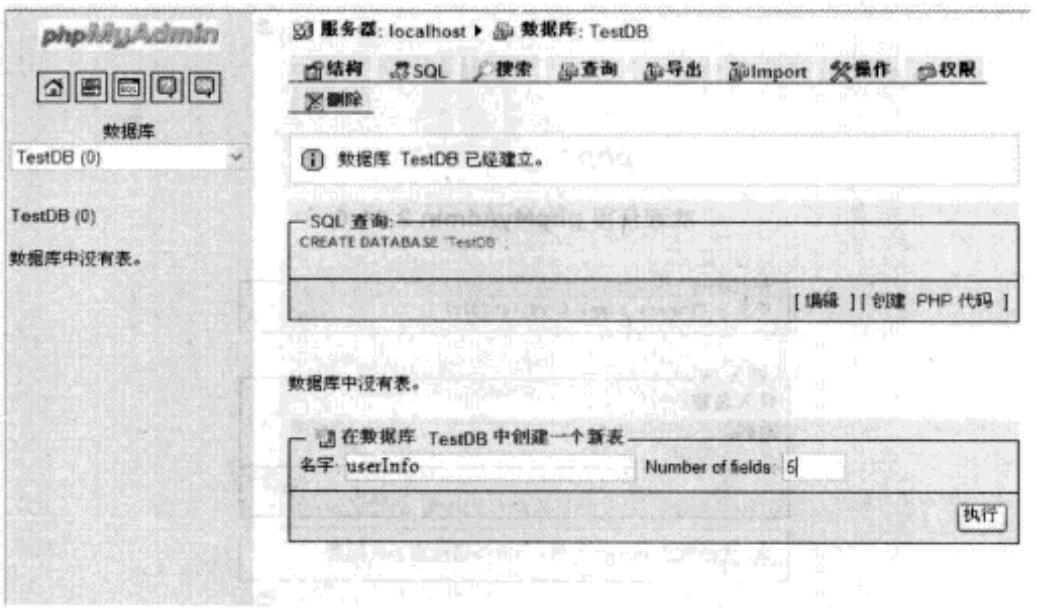


图 12-6　通过 phpMyAdmin 操作数据库

第 4 步，单击“执行”按钮后，将打开一个页面，字段数为 5，如图 12-7 所示。可以分别设置这 5 个字段的名字、字段类型、长度等。



图 12-7　开始正式设计表格

第 5 步，表格设计完之后，phpMyAdmin 会自动打开录入窗口，可以逐条录入相应的数据，如图 12-8 所示。



图 12-8　为表格录入数据

至此已经利用 phpMyAdmin 完成了建立数据库和数据表的工作。当然 phpMyAdmin 的功能远不只这些，还可以通过 phpMyAdmin 来创建、删除、修改记录，创建、修改用户等。

第 13 章　使用 Webmin 配置各种服务

命令行模式是 Linux 操作系统的特色，它可以简单而方便地实现各种服务配置与管理工作，所以本书前文所涉及的各种服务大多是通过命令行方式进行安装和配置的。但是在文本界面下，各种安装包和文件路径难以识别，也不能够很好地解决软件包的依赖问题，容易导致安装出错。在这里专门为初级用户介绍了一款更为简单、直观、易操作的图形化网络服务配置工具——Webmin。

13.1　Webmin 的概述

对于不熟悉 Linux 命令行的用户来说，文本模式的配置方式非常不便，而在 Red Hat Enterprise Linux 5 下，有一些服务，如 Apache、FTP、E-mail 等都不能在桌面系统中进行配置。这些服务，其实是可以采用 Webmin 来配置的。

13.1.1　Webmin 简介

Webmin 是一个基于浏览器的 Linux 系统管理工具，其具有强大的远程操控功能（如图 13-1 所示）。通过浏览器访问 Webmin，可以为用户提供用户账号、Apache、DNS、文件共享等服务的图形化配置，这样可以避免因手工编辑配置文件造成的错误。

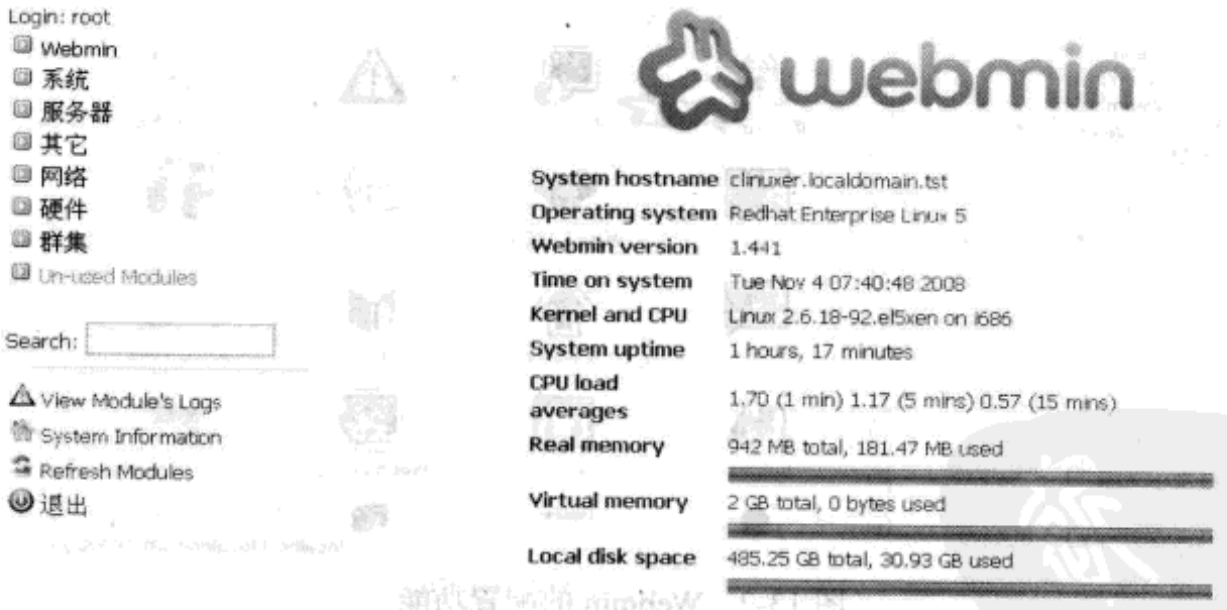


图 13-1　Webmin 界面

13.1.2　Webmin 特性

Webmin 适合目前网络环境的特性很多，具体描述如表 13-1 所示。

表 13-1　Webmin 特性

特　性	描　述
支持多种操作系统	Webmin 不仅支持 Linux，还支持其他 UNIX 平台，如 AIX、HPUX、Solaris 和 Mac OS X 等
管理功能强大	具有本地和远程管理的能力，性能非常高
采用插件式结构	扩展性非常强，通过作者提供的开发向导，用户可以很方便地开发出适合自己的模块
支持多国语言	为用户的操作提供了极大的方便

注意　Webmin 采用访问控制和 SSL 支持来保障系统的安全。

13.1.3 Webmin 主要功能

从 Webmin 的界面中可以看到，其主要管理功能体现在系统（System）、服务器（Server）、其他（Others）、网络（Networking）、硬件（hardware）和群集（Cluster）等部分。各部分功能的简单介绍如下。

1. Webmin

Webmin 功能主要用于 Webmin 的备份、语言设置、用户管理、服务器的查找和配置等几个方面。其中，Webmin 的配置是该部分最常用的功能，用来具体设置允许访问 Webmin 服务器的 IP 地址、服务的端口定义、用户界面定制、显示语言等，如图 13-2 所示。



图 13-2　Webmin 的配置功能

2. 系统

系统（System）功能主要体现在系统的日常管理方面。例如，管理用户与群组、设置开机启动的服务和程序、设置磁盘限额、修改用户密码、设置磁盘和文件系统、d 添加和升级软件包、管理进程和记录文件的备份等，如图 13-3 所示。



图 13-3 系统（System）的功能

3. 服务器

服务器（Servers）主要用于各类服务器的设置，包括 Apache、sendmail、Postfix、Samba、DHCP、代理服务器、MySQL 和 DNS 等，如图 13-4 所示。

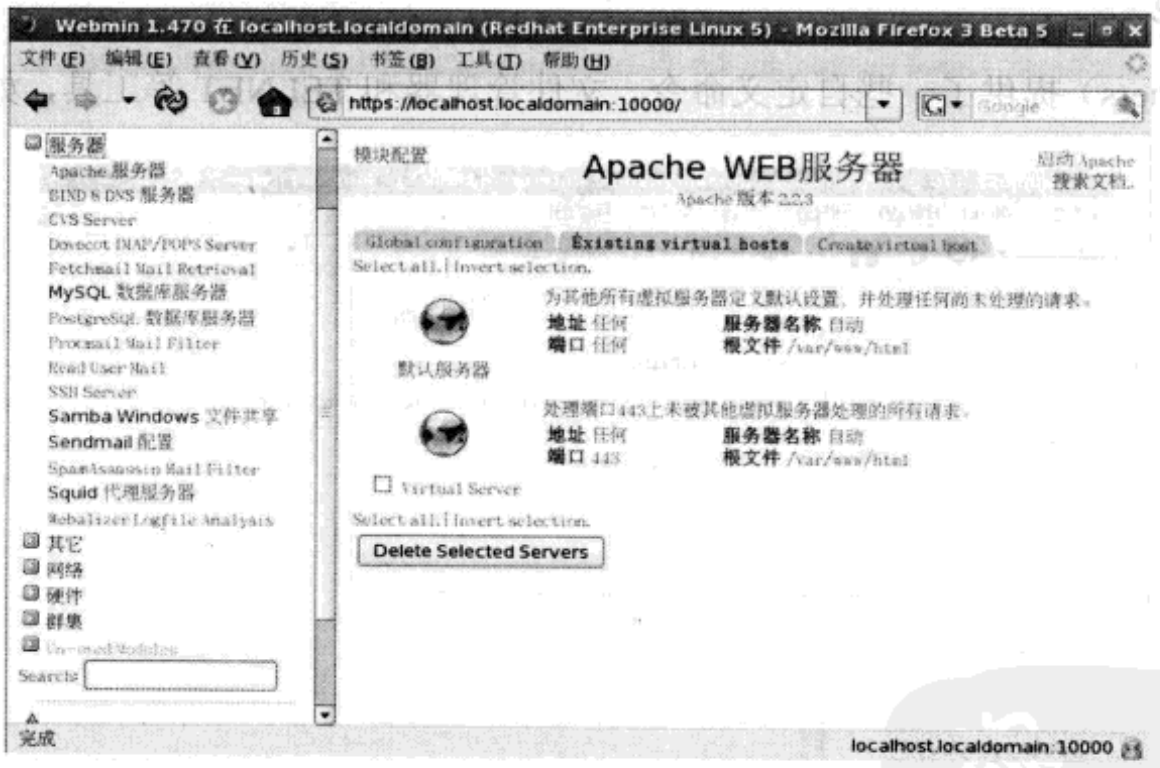


图 13-4 服务器的功能

4. 硬件

硬件（Hardware）的功能主要体现在 Linux 中的各种外围设备的设置方面。例如，创建新的开机内核、创建和配置 RAID、配置网络、磁盘分区设置、设置打印机等，如图 13-5 所示。

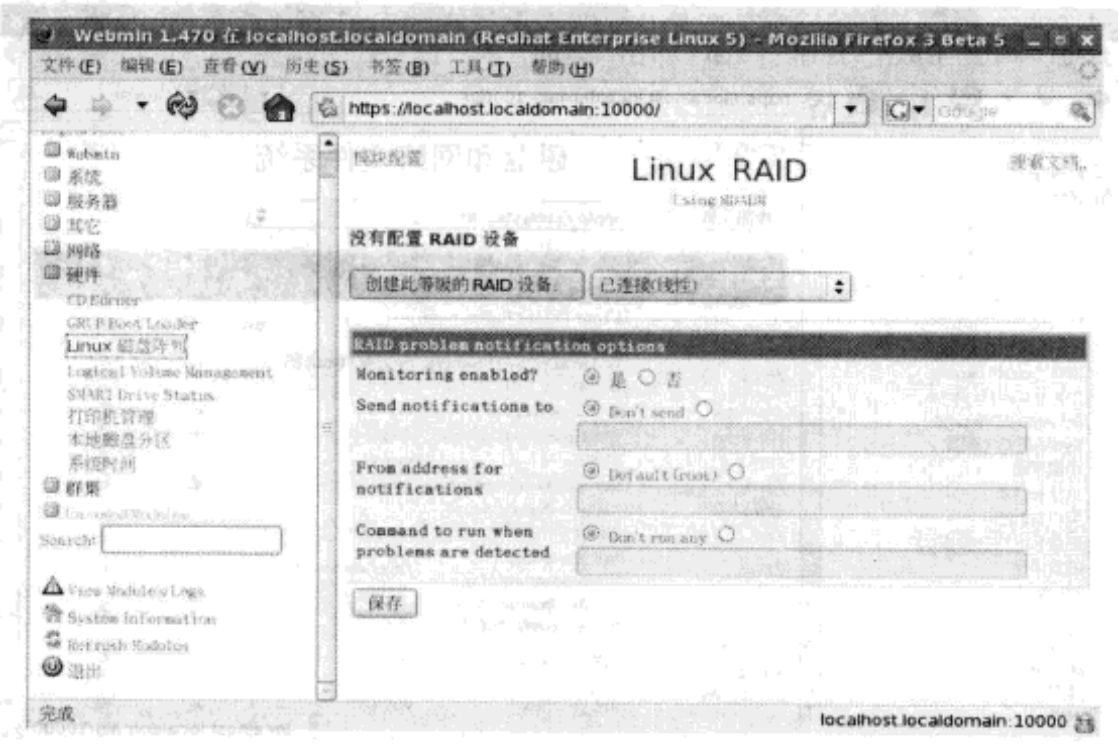


图 13-5　硬件（hardware）的功能

注意　在使用硬件（Hardware）的功能对磁盘进行分区时，磁盘上必须有剩余空间才能添加新的逻辑分区。

5. 其他

其他（Others）提供了一些自定义命令、文件管理器和 TELNET 等工具，如图 13-6 所示。

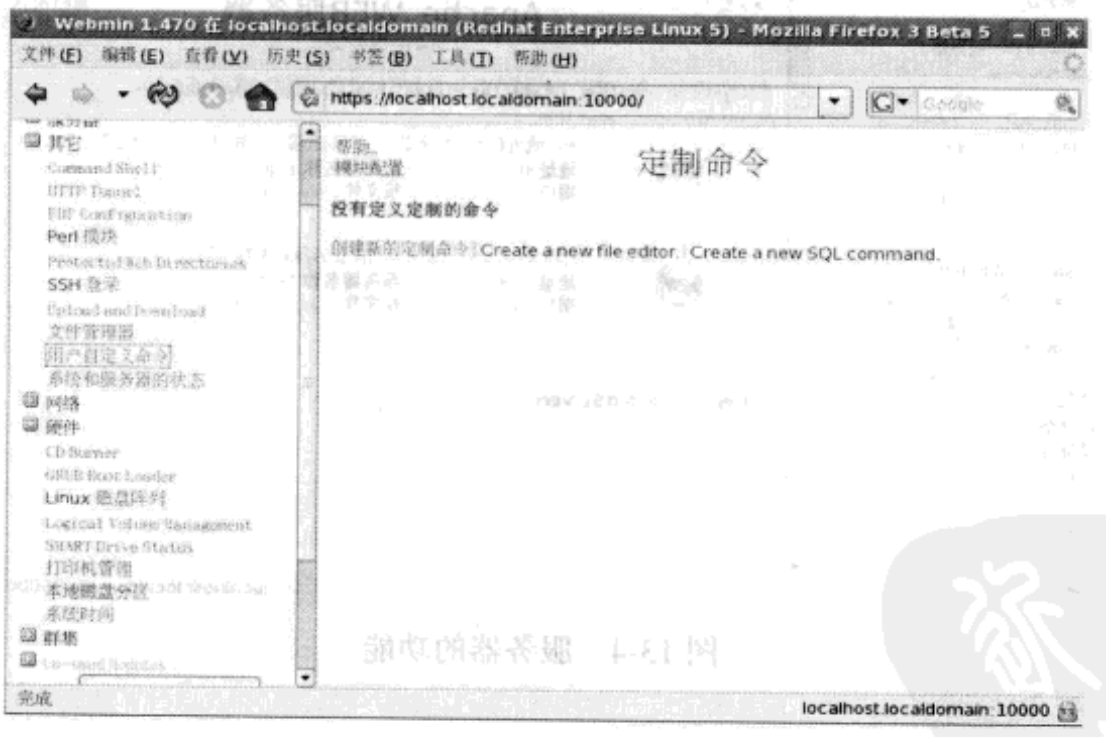


图 13-6　其他（others）功能

6. 群集

群集（Cluster）是用于配置 Red Hat Enterprise Linux 5 集群的选项。例如，在进行系统故障检测时同时检测用户、组和软件包，如图 13-7 所示。

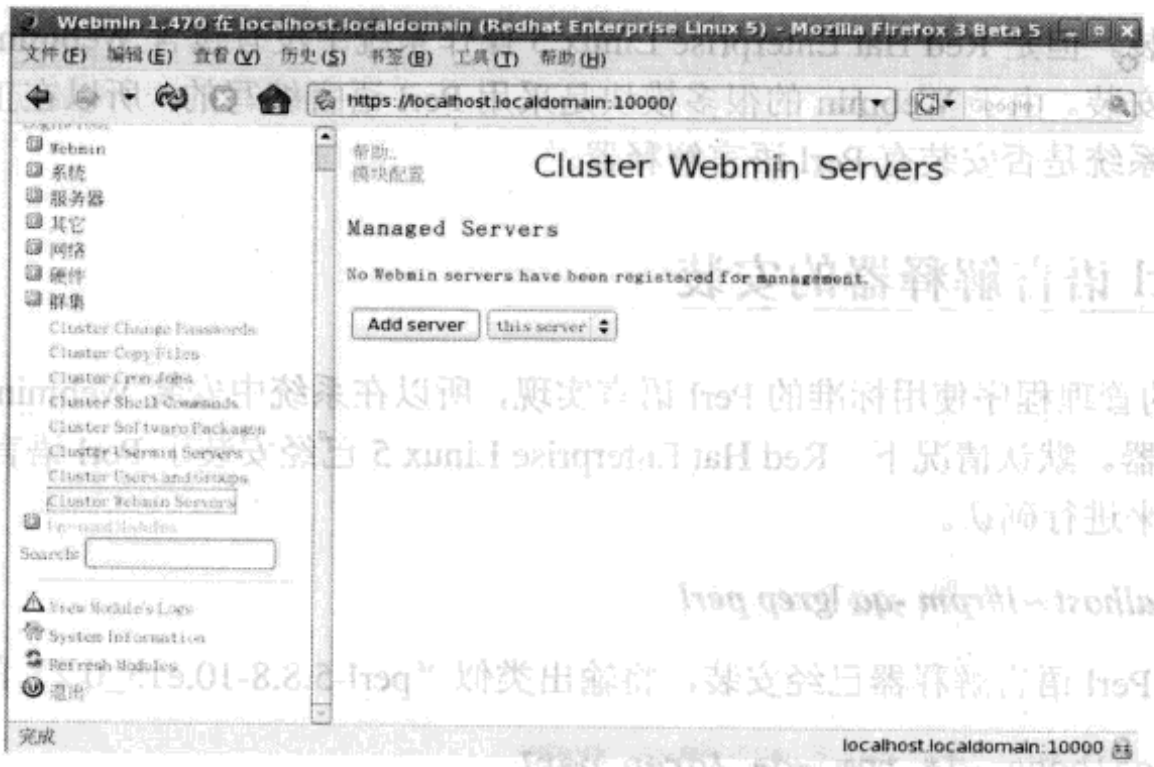


图 13-7　群集（Cluster）的功能

注意　集群在这里指的是一组需要同步配置的系统。

7. 网络

网络（Network）功能主要是用于网络、ADSL 拨号、防火墙、NFS 输出、inetd（Internet 超级服务器）服务和带宽检测等信息的配置，如图 13-8 所示。

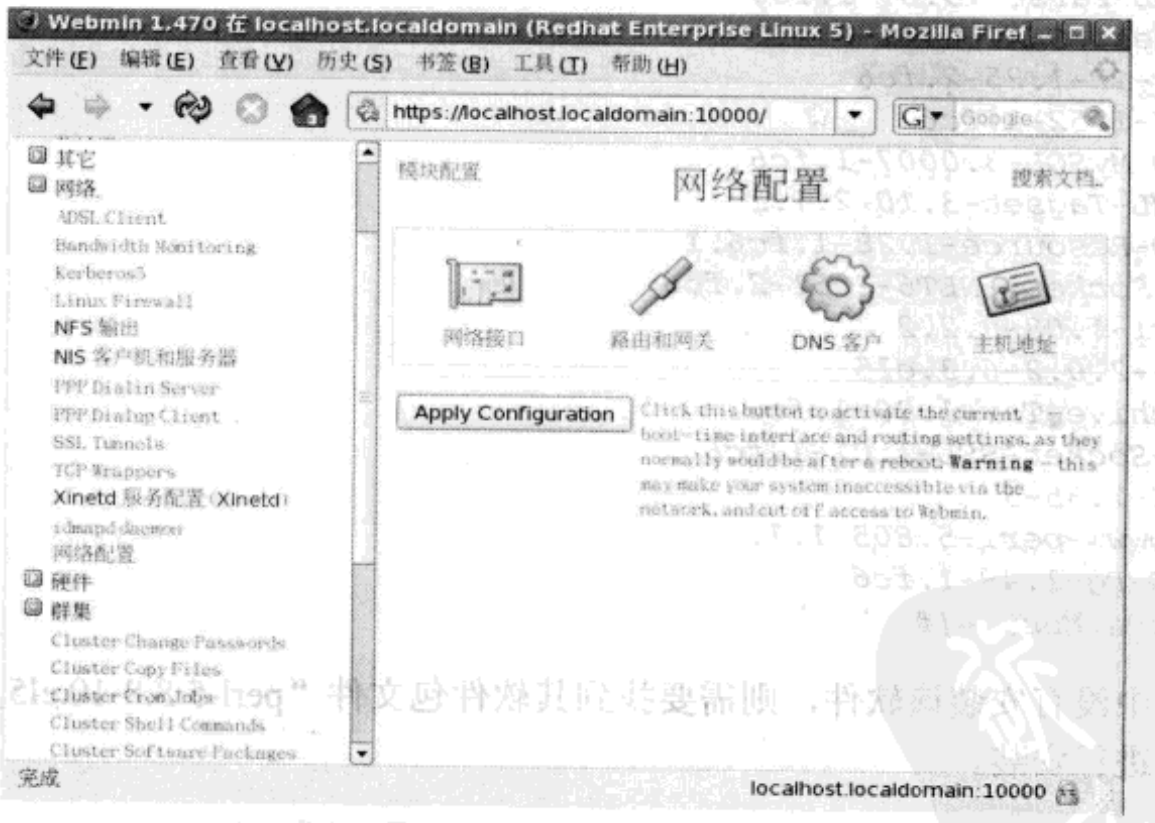


图 13-8　网络（Network）的功能

13.2　安装和配置 Webmin

常用的网络服务器，在 Red Hat Enterprise Linux 操作系统中一般都有内置的安装文件，可

以直接进行安装。但是 Red Hat Enterprise Linux 5 操作系统中没有包含 Webmin 的相关套件，用户需要自行安装。由于 Webmin 的很多模块是采用 Perl 语言编写的，所以在正式安装前，用户应检查当前系统是否安装有 Perl 语言解释器。

13.2.1 Perl 语言解释器的安装

Webmin 的管理程序使用标准的 Perl 语言实现，所以在系统中安装 Webmin 的前提是已经安装 Perl 解释器。默认情况下，Red Hat Enterprise Linux 5 已经安装了 Perl 语言解释器，可以执行以下命令来进行确认。

```
[root@localhost ~]#rpm -qa |grep perl
```

若系统中 Perl 语言解释器已经安装，将输出类似“perl-5.8.8-10.e15_0.2”的结果。

```
[root@localhost ~]# rpm -qa |grep perl
perl-Compress-Zlib-1.42-1.fc6
perl-Digest-HMAC-1.01-15
perl-Socket6-0.19-3.fc6
perl-SGMLSpm-1.03ii-16.2.1
perl-IO-Zlib-1.04-4.2.1
cyrus-imapd-perl-2.3.7-2.el5
perl-Net-DNS-0.59-3.el5
perl-Net-SSLeay-1.30-4.fc6
perl-5.8.8-10.el5_0.2
perl-String-CRC32-1.4-2.fc6
perl-HTML-Parser-3.55-1.fc6
perl-Digest-SHA1-2.11-1.2.1
perl-Net-IP-1.25-2.fc6
perl-DBI-1.52-1.fc6
perl-DBD-MySQL-3.0007-1.fc6
perl-HTML-Tagset-3.10-2.1.1
perl-BSD-Resource-1.28-1.fc6.1
perl-IO-Socket-INET6-2.51-2.fc6
newt-perl-1.08-9.2.2
mod_perl-2.0.2-6.3.el5
perl-Archive-Tar-1.30-1.fc6
perl-IO-Socket-SSL-1.01-1.fc6
perl-URI-1.35-3
perl-libwww-perl-5.805-1.1.1
perl-DBD-Pg-1.49-1.fc6
[root@localhost ~]#
```

如果系统中没有安装该软件，则需要找到其软件包文件“perl-5.8.8-10.el5_0.2.i386.rpm”，执行如下命令进行安装。

```
[root@localhost ~]#rpm -ivh perl-5.8.8-10.el5_0.2.i386.rpm
```

13.2.2 Net-SSLeay perl 和 OpenSSL 的安装

默认情况下，Webmin 的用户身份验证是明文形式的，安全性不是很高。对此，可以采用

OpenSSL 和 Net-SSLeay 模块来实现用户身份信息的验证和加密传输。

1. OpenSSL

默认情况下，Red Hat Enterprise Linux 5 已经内置了 OpenSSL 模块，可以使用以下命令查询安装情况。

```
[root@localhost ~]#rpm -qa |grep openssl
openssl-devel-0.9.8b-10.el5
openssl-0.9.8b-10.el5
[root@localhost ~]#
```

查询命令执行以后，输出结果包含“openssl-0.9.8b-10.e15”，表示 OpenSSL 已经安装。

若没有信息提示，可以从 Red Hat Enterprise Linux 5 的安装光盘中，找到相关软件包，通过以下命令进行安装。

```
[root@localhost ~]#rpm -ivh openssl-0.9.8b-10.el5.i386.rpm
```

2. Net-SSLeay perl

对于 Net-SSLeay 的 perl 模块，则需要用户通过以下命令手动安装。

```
[root@localhost Server]# rpm -ivh perl-Net-SSLeay-1.30-4.fc6.i386.rpm
warning: perl-Net-SSLeay-1.30-4.fc6.i386.rpm: Header V3 DSA signature: NOKEY,
key ID 37017186
Preparing...                ########################################### [100%]
   1:perl-Net-SSLeay        ###########################################
[100%]
[root@localhost Server]#
```

13.2.3 安装与访问 Webmin

Webmin 是目前功能最强大的基于 Web 的系统管理工具之一，但是在 Red Hat Enterprise Linux 5 操作系统中没有内置软件包，如果要正常使用，用户必须从网上下载该软件包进行安装。

1. 安装 Webmin

第 1 步，通过 Webmin 的官方网站 www.webmin.com，下载相应的 RPM 格式安装包文件到系统桌面，如图 13-9 所示。



图 13-9 下载 Webmin 的 RPM 安装包文件

第 2 步，显示 Red Hat Enterprise Linux 5 的控制台界面，执行以下命令安装 Webmin。

```
[root@localhost ~]# cd Desktop
[root@localhost Desktop]# rpm -ivh webmin-1.470-1.noarch.rpm
warning: webmin-1.470-1.noarch.rpm: Header V3 DSA signature: NOKEY, key ID
11f63c51
Preparing...                ########################################### [100%]
```

```
Operating system is Redhat Enterprise Linux
   1:webmin              ########################################## [100%]
Webmin      install      complete.      You      can      now      login      to
https://localhost.localdomain:10000/
as root with your root password.
[root@localhost Desktop]#
```

第 3 步，依次单击“系统”→“管理”→“安全级别和防火墙”菜单命令，如图 13-10 所示。

第 4 步，弹出“安全级别设置”窗口，单击“其他端口”按钮，添加一个端口“10000”，最后单击“确定”按钮使设置生效，如图 13-11 所示。

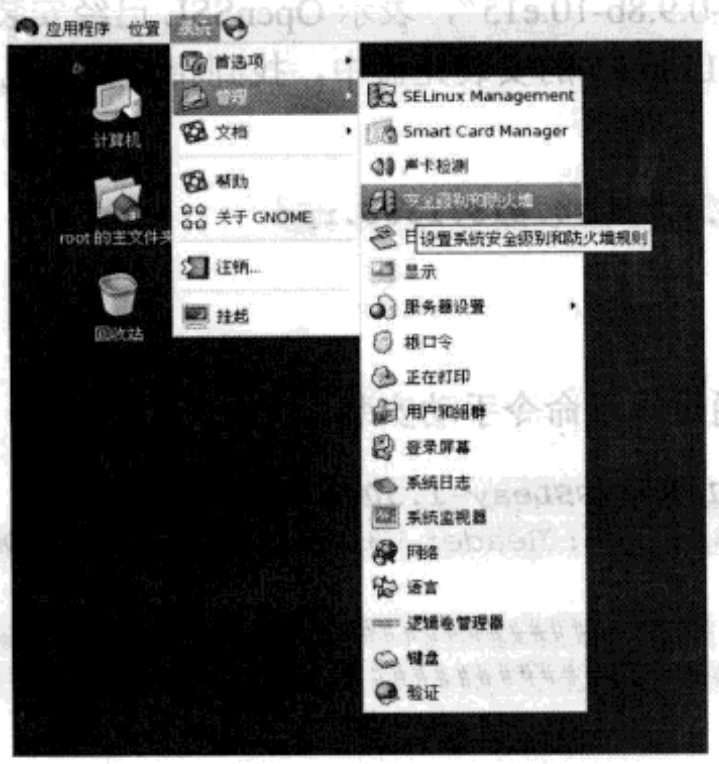


图 13-10　选择“安全级别和防火墙”菜单命令

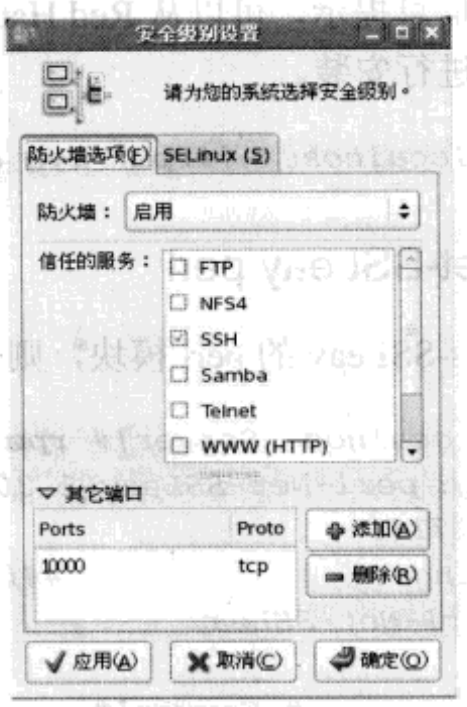


图 13-11　“安全级别设置”窗口

注意　Webmin 使用了自带的 Mini Server 进行数据传输，需要使用自定义的 10000 端口，而不是 Apache 默认的 80 端口。

在运行 Webmin 的后，若显示“Webmin install complete. You can now login to http://toplinux.localdomain.tst:10000/ as root with your root password..”的提示，则表示输入的 root 的账号和密码通过服务器的 10000 端口管理系统。

2. 本地访问 Webmin

本地访问 Webmin，在 Firefox 地址栏输入“https://localhost.localdomain:10000/”，即可显示如图 13-12 所示的 Webmin 登录界面。

注意　首次登录 Webmin，最好使用默认的 root 账号和密码，登录之后可以在进行进一步设置。

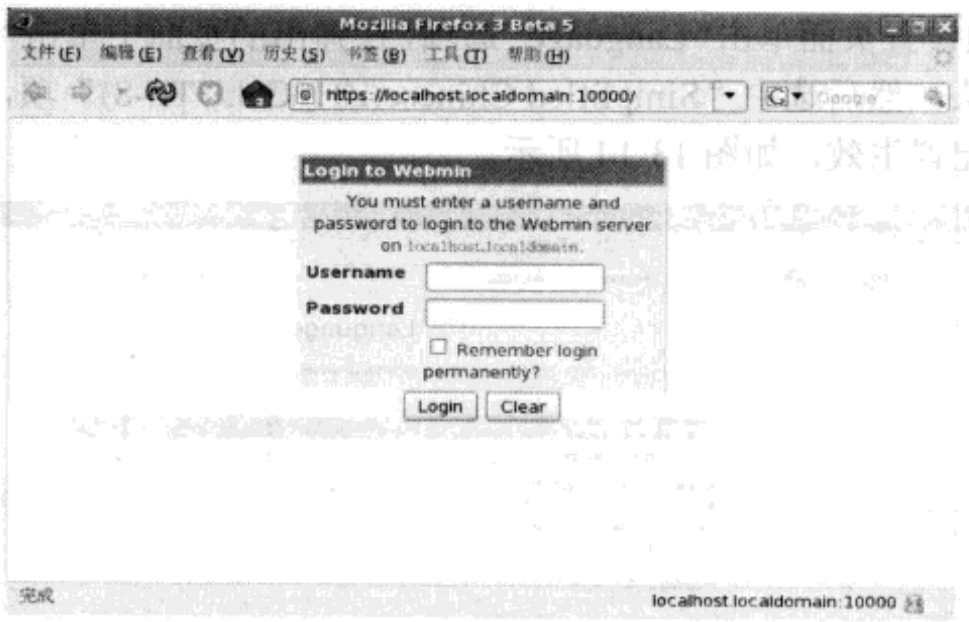


图 13-12　Webmin 登录界面

3. 远程访问 Webmin

在 IE、Firefox、Opera 或者 Safari 的地址栏中输入“https://主机 IP 地址:10000/”，即可远程访问 Webmin。除了使用 IP 地址外，用户还可以通过域名远程访问 Webmin。

注意　对于一些安全要求较高的系统，可以使用 SSL 进行 Webmin 的数据传输，输入的地址为 https://toplinux.localdomain.tst:10000/（http 后面多了个“s”，表示加密传输）。但要注意，若开启了 SSL 传输，是不能通过一般方式访问 Webmin 的。

13.2.4 设置 Webmin 为中文界面

可能许多国内的用户不习惯 Webmin 默认的英文语言界面，可以在使用之前将 Webmin 界面设置为中文。具体设置方法如下。

第 1 步，在 Webmin 的主页面，依次单击页面左侧的“Webmin”→“Webmin Configuration”菜单命令，显示如图 13-13 所示的 Webmin 选项设置页面。

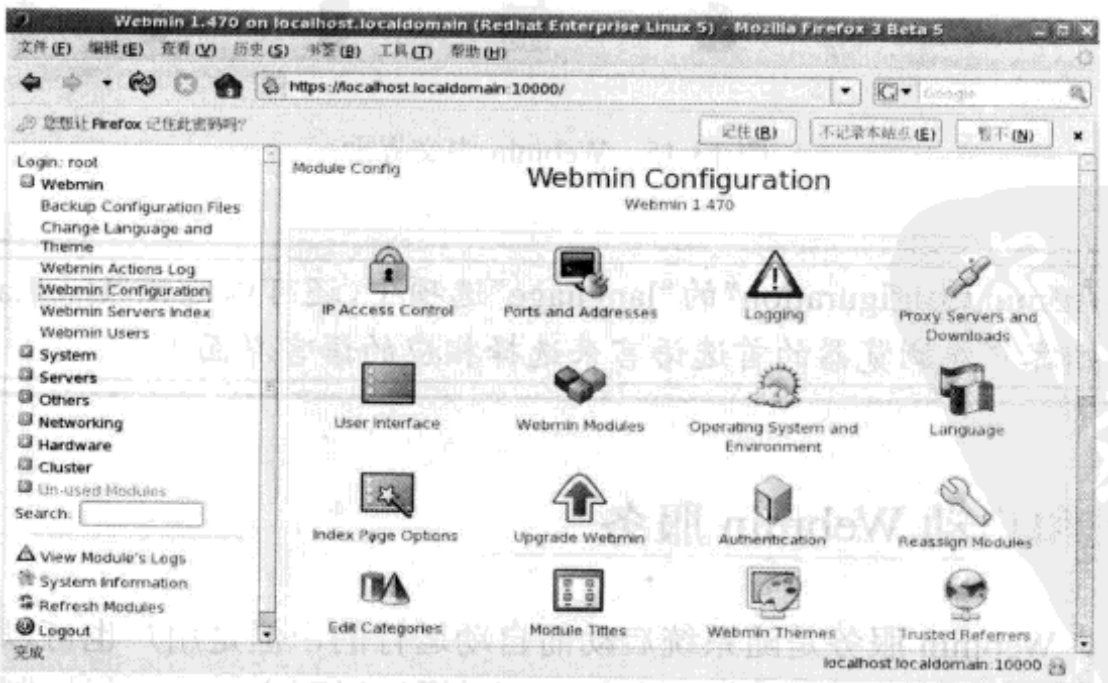


图 13-13　Webmin 选项设置页面

第 2 步，在选项设置页面单击“Language”选项，在弹出的页面中单击“Display in language”选项右侧的下拉箭头，然后选中“Simplified Chinese(ZH_CN.UTF-8)”项，最后单击“Change Language”按钮使配置生效，如图 13-14 所示。

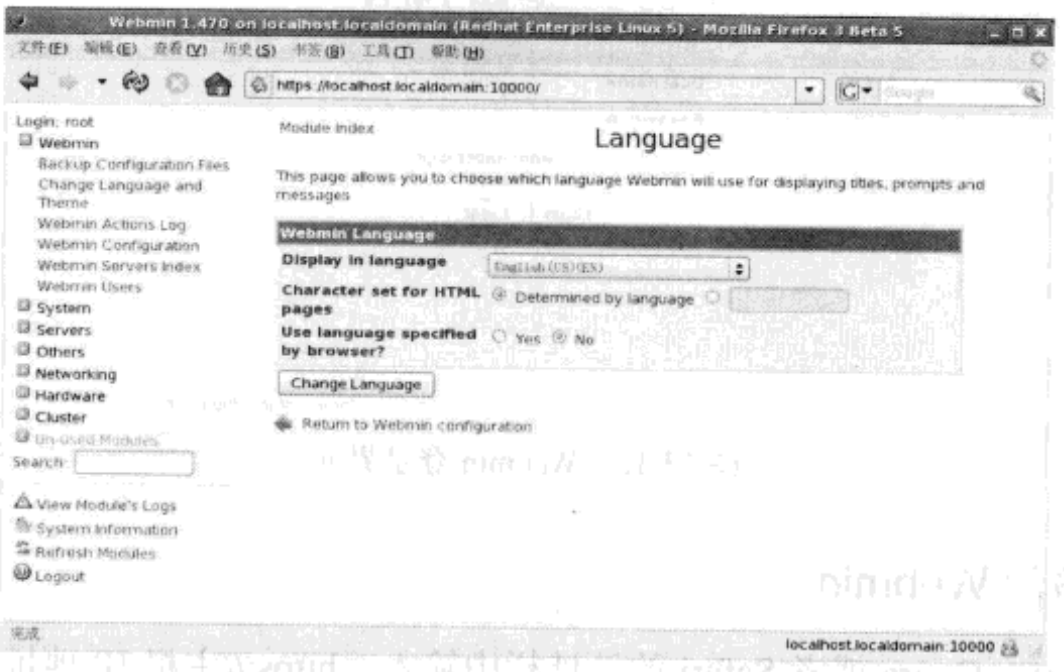


图 13-14　设置“Language”选项

Webmin 中文界面如图 13-15 所示。

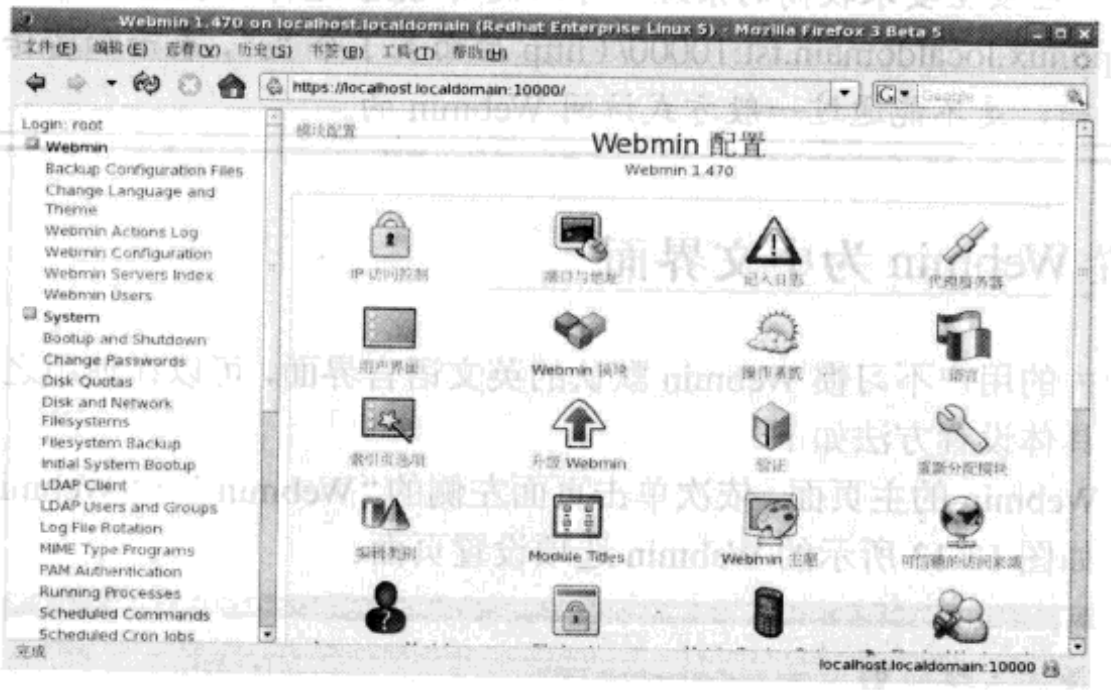


图 13-15　Webmin 中文界面

注意　在“Webmin Configuration”的“language”选项中，还可以选择“Use Language specified by browser”（根据客户端浏览器的首选语言来选择相应的语言界面）。

13.2.5 停止和启动 Webmin 服务

默认情况下，Webmin 服务是随系统启动而自动运行的，但是用户也可以设置 Webmin 服务在需要时才启动。Webmin 服务的停止和启动可以通过以下方式来进行控制。

1. 启动 Webmin 服务

启动 Webmin 服务的命令如下。

```
[root@localhost ~]#service webmin start
```

2. 停止 Webmin 服务

停止 Webmin 服务的命令如下。

```
[root@localhost ~]#service webmin stop
```

3. 重新启动 Webmin 服务

执行以下命令可以重新启动 Webmin 服务。

```
[root@localhost ~]#service webmin restart
```

13.3 用 Webmin 配置服务

Red Hat Enterprise Linux 5 虽然提供了很多图形界面的配置工具，但与 Webmin 相比显得较复杂，而且缺少很多必要的功能，使用起来也没有 Webmin 方便。通过前面的学习，想必读者对 Webmin 已经有了一个全新的认识了。下面就通过几个常用服务的配置实例来体验一下 Webmin 的强大功能。

13.3.1 Samba 服务的配置

使用 Webmin 配置 Samba 服务的具体操作步骤如下。

第 1 步，在远程客户端打开浏览器，并在地址栏输入“https://localhost.localdomain:10000/”访问 Webmin，然后输入 root 账号和密码登录。

第 2 步，在 Webmin 的主页面，依次单击“服务器”→“Samba Windows 文件共享”菜单命令，显示如图 13-16 所示的“Samba 共享管理器”主页面。

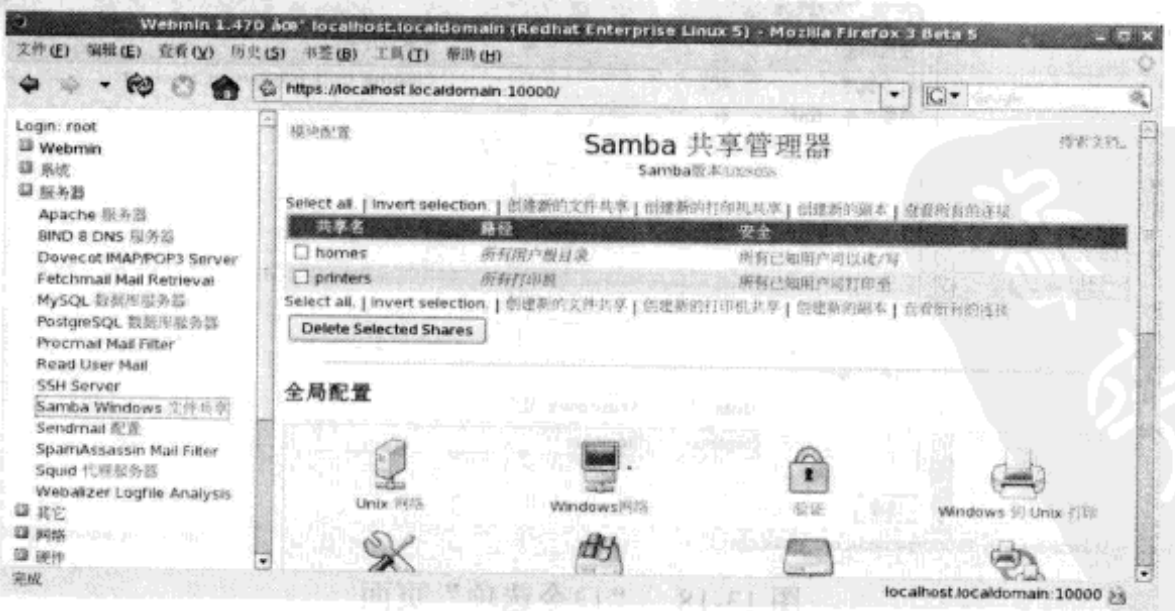


图 13-16 “Samba 共享管理器”主页面

注意 图中所示的 Samba 主配置默认的共享是 homes 和 printers。

第 3 步，在 Samba 配置主页面中的全局配置选项中单击“Windows 网络”图标，并在弹出的“Windows 网络选项”页面中设置“工作组”、“服务器名称”，然后选择“安全”下拉列表框的“用户级别”项，最后单击“保存”按钮使设置生效，如图 13-17 所示。



图 13-17 Windows 网络配置选项

注意 Windows 系统的计算机是通过计算机名和群组来确定位置的。

第 4 步，对工作组等配置完成后，返回到 Samba 配置的主页面，并在全局配置选项中单击“验证”图标，显示“口令选项”页面。在该页面中，将“使用加密口令”设置为“是”，然后在下方的用户名映射区域中设置对应的用户名，如图 13-18 所示。



图 13-18 “口令选项”页面

注意 在 Samba 配置的主页面单击“文件共享默认值”图标，在显示的页面中选择“口令和访问控制”命令后，可以设置允许访问的计算机。允许访问的计算机默认设置为所有的主机，一般采用默认情况即可。

第 5 步，回到 Samba 配置的主页面，并单击页面上方的“创建新的文件共享”链接，将显示“创建文件共享”页面。然后设置共享名、共享的目录和简单注释，如图 13-19 所示。

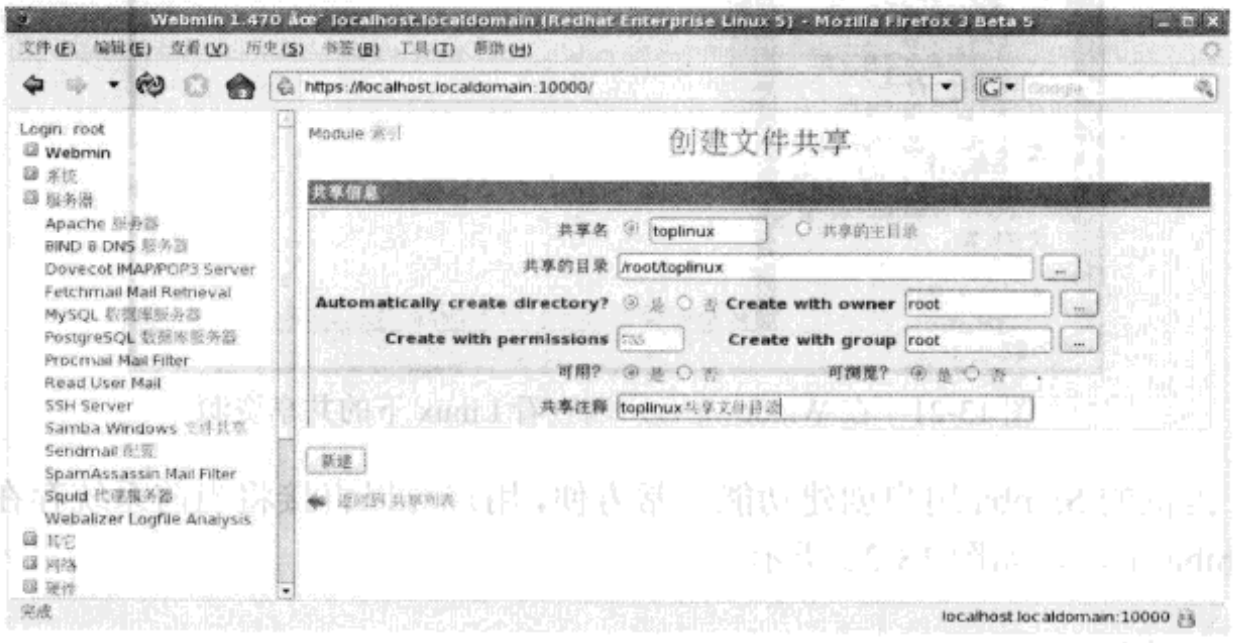


图 13-19 “创建文件共享”页面

注意 要从 Windows 操作系统访问 Linux 操作系统上的共享资源，需要通过用户名和密码进行身份验证。

第 6 步，要使设置生效，需回到 Samba 配置的主页面，并单击页面下方的“重新启动 Samba 服务器”按钮，如图 13-20 所示。

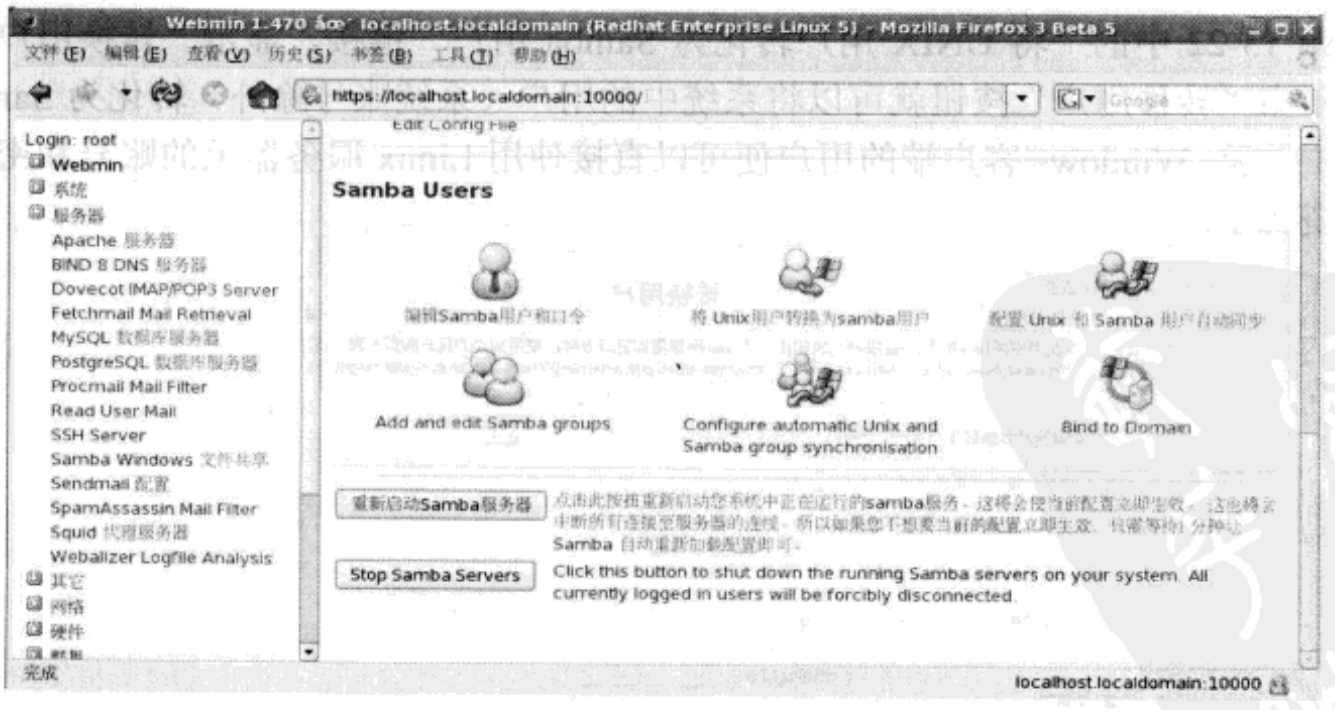


图 13-20 重新启动 Samba 服务器使设置生效

第 7 步，在 Windows 客户端，输入相应的 Samba 账号和密码，从网络邻居访问 Samba 服务器，查看 Linux 下的共享资源，如图 13-21 所示。

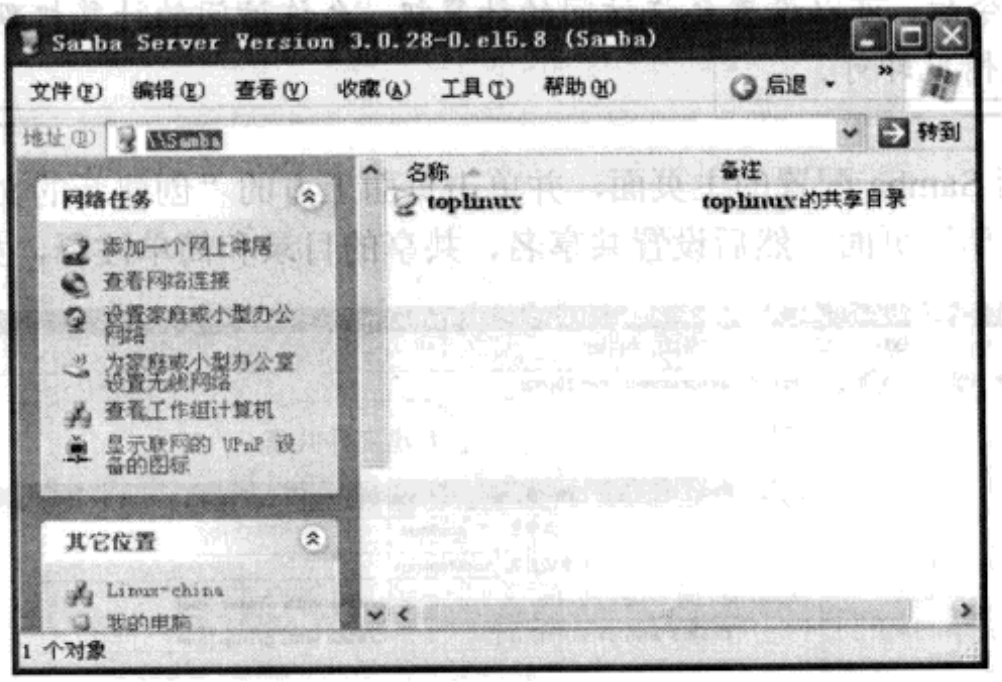


图 13-21　在 Windows 客户端查看 Linux 下的共享资源

Webmin 提供的 Samba 用户创建功能非常方便，用户可以直接将当前系统存在的 UNIX 用户转化为 Samba 用户，如图 13-22 所示。

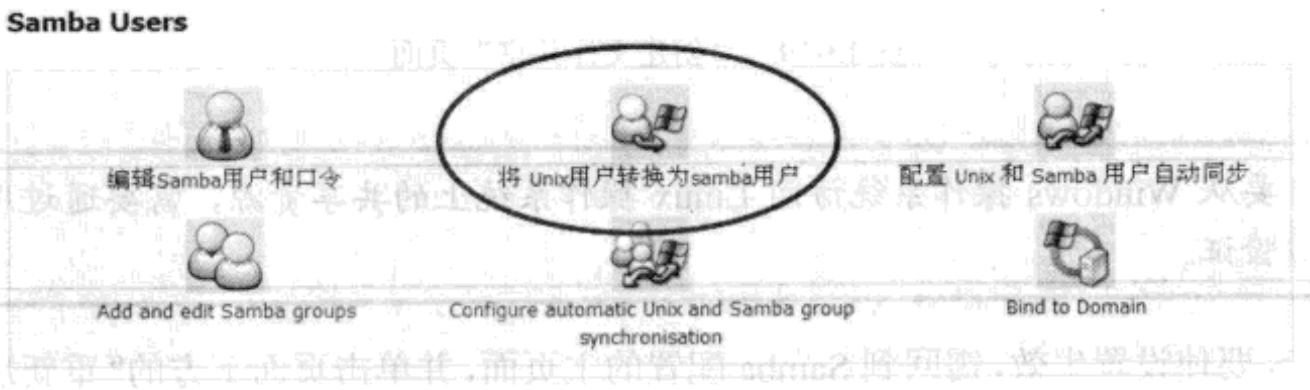


图 13-22　转化 UNIX 用户为 Samba 用户

单击图 13-22 中的“将 UNIX 用户转化为 Samba 用户”链接，显示“转换用户”页面，然后直接单击“转移用户”按钮就可以将系统中的用户（系统账号除外）转化为 Samba 用户，如图 13-23 所示。Windows 客户端的用户便可以直接使用 Linux 服务器上的账号和密码访问授权的共享资源了。

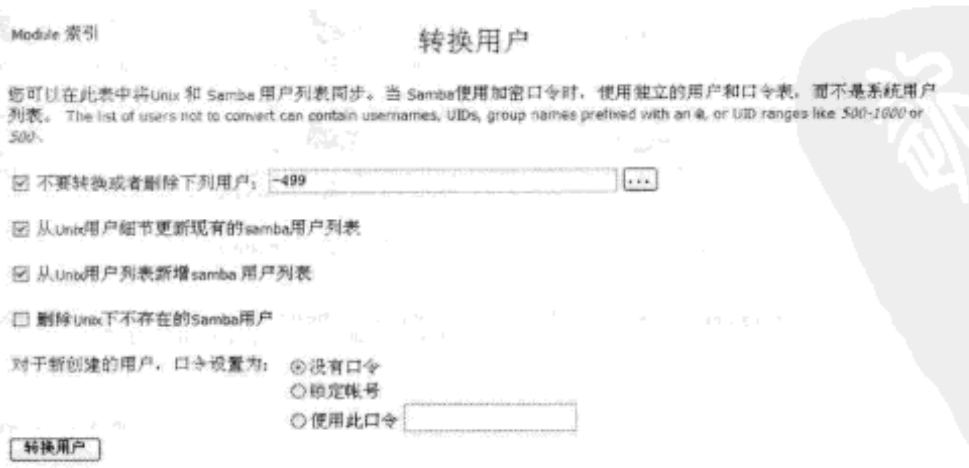


图 13-23　“转换用户”页面

13.3.2 NFS 服务的配置

使用 Webmin 配置 NFS 服务，具体操作如下。

第 1 步，在 Webmin 的主页面，依次单击页面左侧的“网络”→“NFS 输出”菜单命令，显示“NFS 输出”页面，如图 13-24 所示。

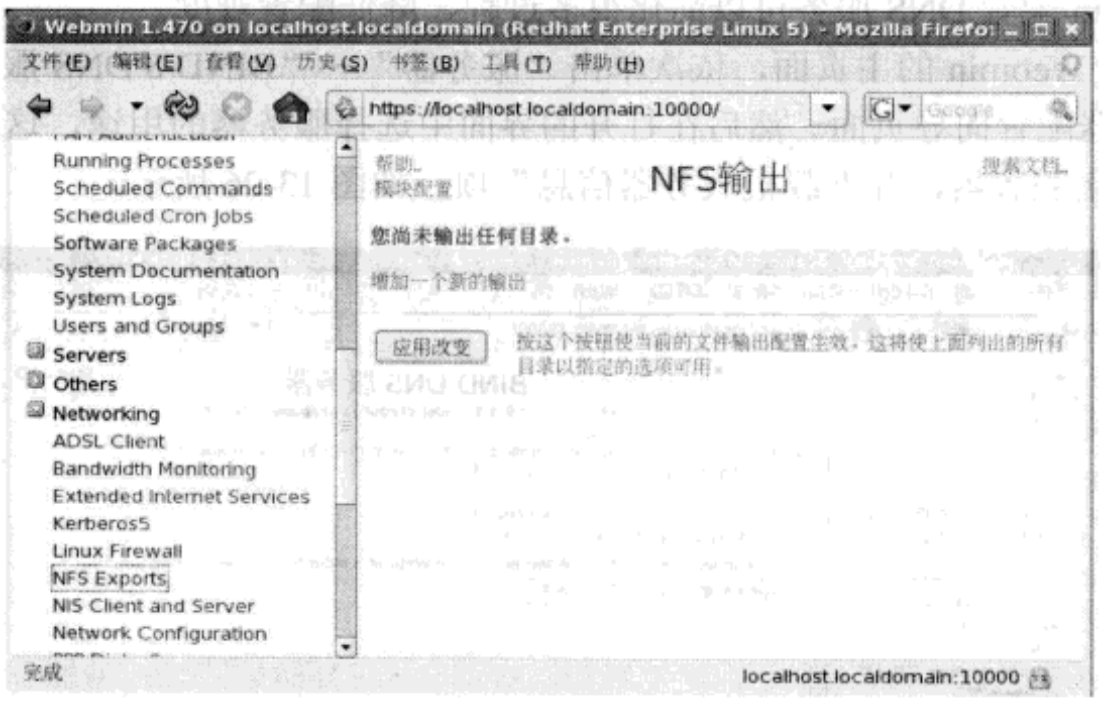


图 13-24 “NFS 输出”页面

第 2 步，在“NFS 输出”配置页面，单击“增加一个新的输出”链接即可显示“创建输出”页面，如图 13-25 所示。然后单击 “要输出的目录”后的“浏览”按钮找到需要输出的目录，单击“OK”按钮确定。



图 13-25 在这里创建 NFS 输出目录并设置权限

然后再对下方的安全权限进行简单的配置。

第 3 步，在 Linux 客户端中执行以下命令进行客户端测试。若没有错误提示，则表示 NFS 服务器的输出挂接成功。

```
[toplinux@localhost /etc]#mkdir /mnt/nfs    ; #创建一个目录，用于挂接 NFS 输出
[toplinux@localhost /etc]#mount -t nfs toplinux.localdomain.tst:/ export/
toplinux /mnt/nfs
                                            ; #挂接 NFS 服务器的输出目录
```

13.3.3 DNS 服务的配置

使用 Webmin 配置 DNS 服务可以简化很多过程，降低配置难度。

第 1 步，在 Webmin 的主页面，依次单击“服务器”→“BIND 8 DNS 服务器”超链接，显示 DNS 服务器配置向导页面。然后在打开的界面中选择服务器的用途，这里选择的是“设置为 Internet 命名服务器，并下载根服务器信息”项，如图 13-26 所示。

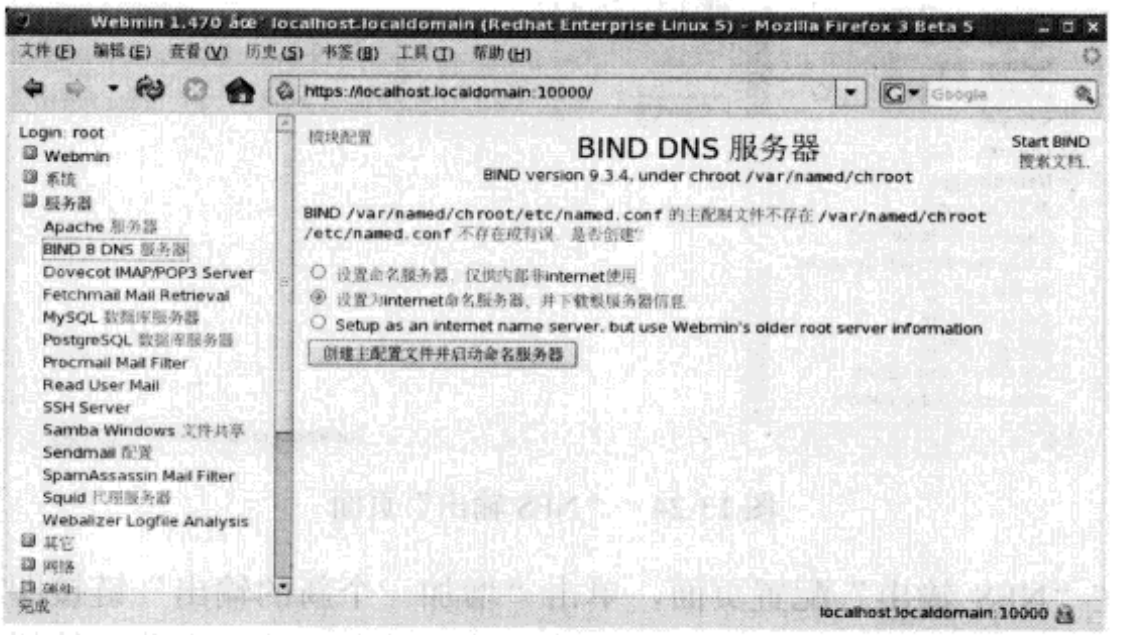


图 13-26　配置 DNS 服务器的用途

第 2 步，单击“创建主配置文件并启动命名服务器”按钮，使 DNS 服务器运行，弹出 DNS 配置主页面，如图 13-27 所示。



图 13-27　DNS 配置主界面

第 3 步，在 DNS 配置主页面中，单击“现有 DNS 区域”下方的“创建新的主区域” 超链接。然后进行相关设置，如图 13-28 所示。

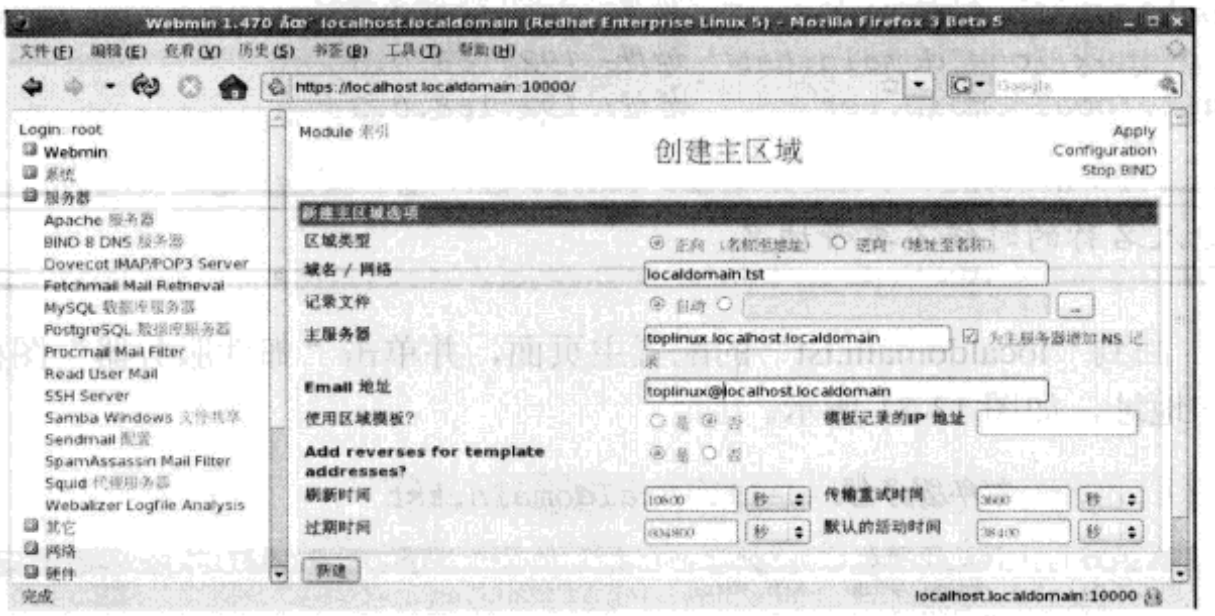


图 13-28　设置“新建主区域选项”

第 4 步，单击“新建”按钮，将弹出“localdomain.tst”的配置页面，如图 13-29 所示。



图 13-29　“localdomain.tst”的配置页面

第 5 步，在“localdomain.tst”的配置页面中，单击“地址”图标，显示“地址个记录”页面，在该页面中设置相关的地址，如图 13-30 所示。

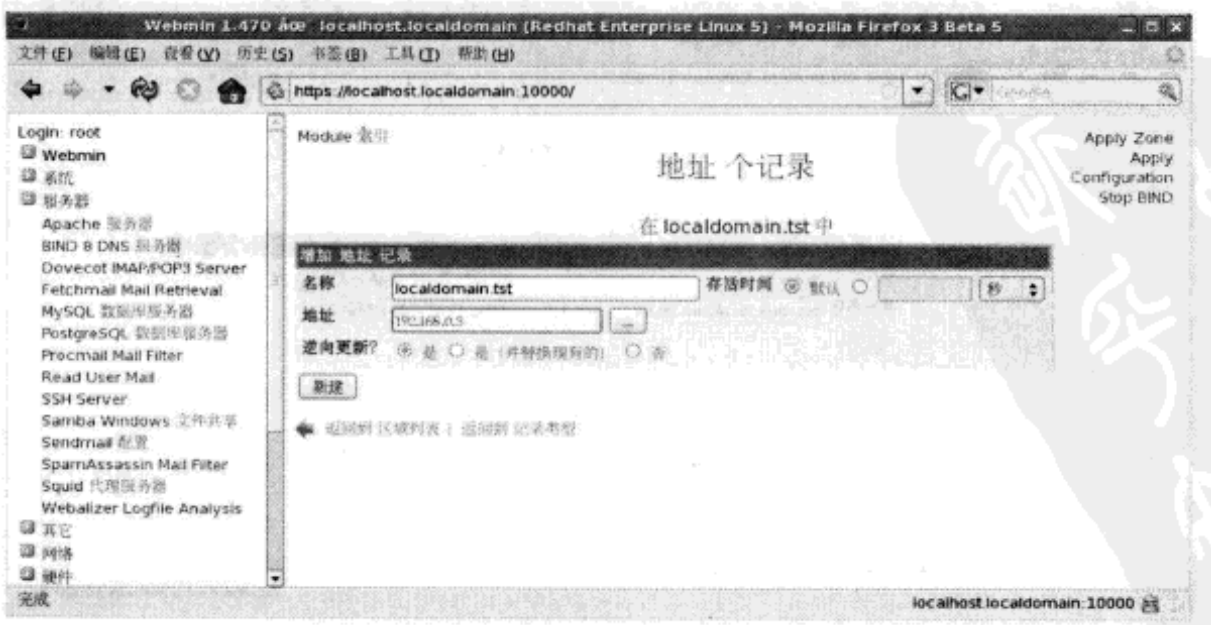


图 13-30　设置相关地址

```
名称：localdomain.tst                地址：192.168.0.3
名称：toplinux.localdomain.tst       地址：192.168.0.3
名称：mail.localdomain.tst           地址：192.168.0.3
```

注意　输入名称的时候不要带域名。

第 6 步，回到“localdomain.tst”的配置主页面，并单击“邮件服务器”图标，然后设置相关的名称和地址，如图 13-31 所示。

```
名称：mail          邮件服务器：mail.localdomain.tst
```

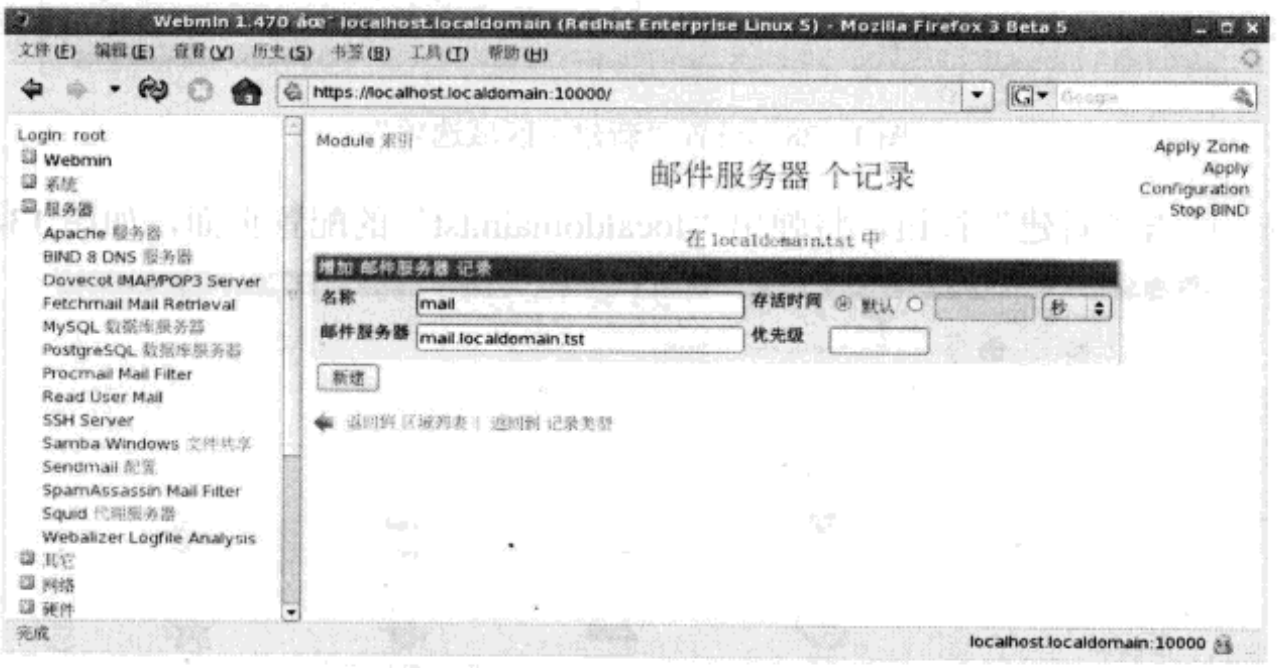


图 13-31　设置邮件服务器的名称和地址

第 7 步，回到“localdomain.tst”的配置主页面，并单击“名称别名”图标，然后进行相关的名称设置，如图 13-32 所示。

```
名称：www
真实名称：toplinux.localdomain.tst
```

这样设置以后，访问 www.localdomain.tst 也相当于访问 toplinux.localdomain.tst 服务器。

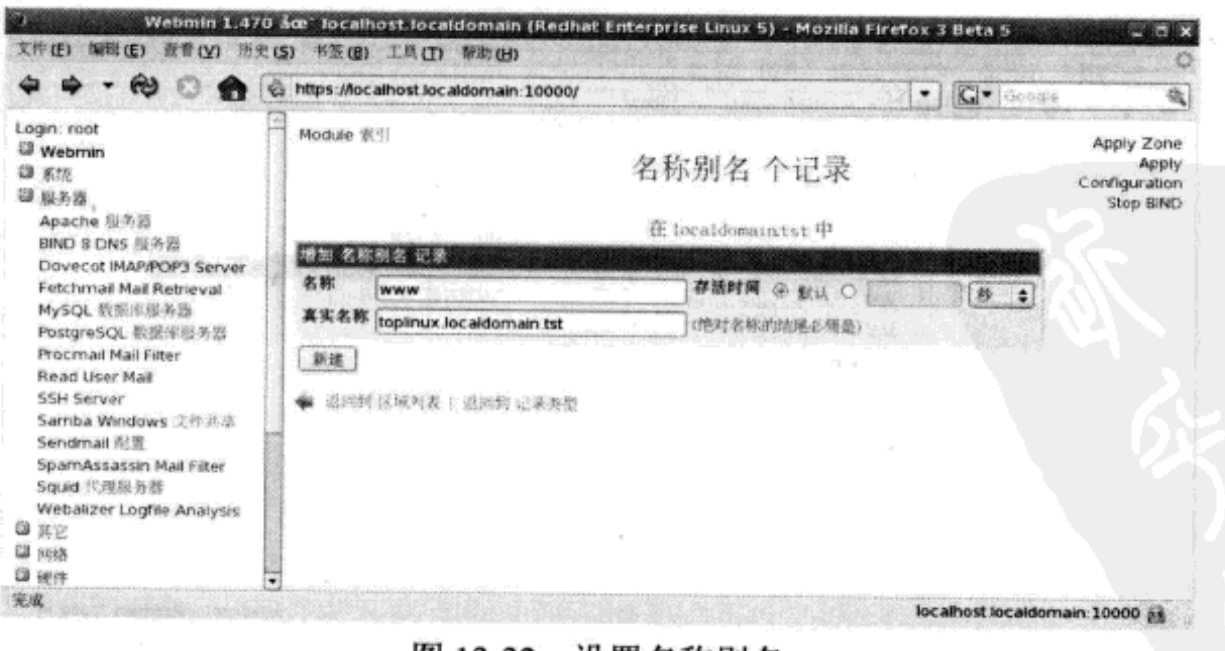


图 13-32　设置名称别名

第 8 步，回到 DNS 服务器配置的主页面，再次单击“创建新的主区域”图标，并进行以下设置，如图 13-33 所示。

```
区域类型：      逆向
域名：    192.168.0
主服务器：      toplinux.localdomain.tst
Email 地址：    root@localdomain.tst
```

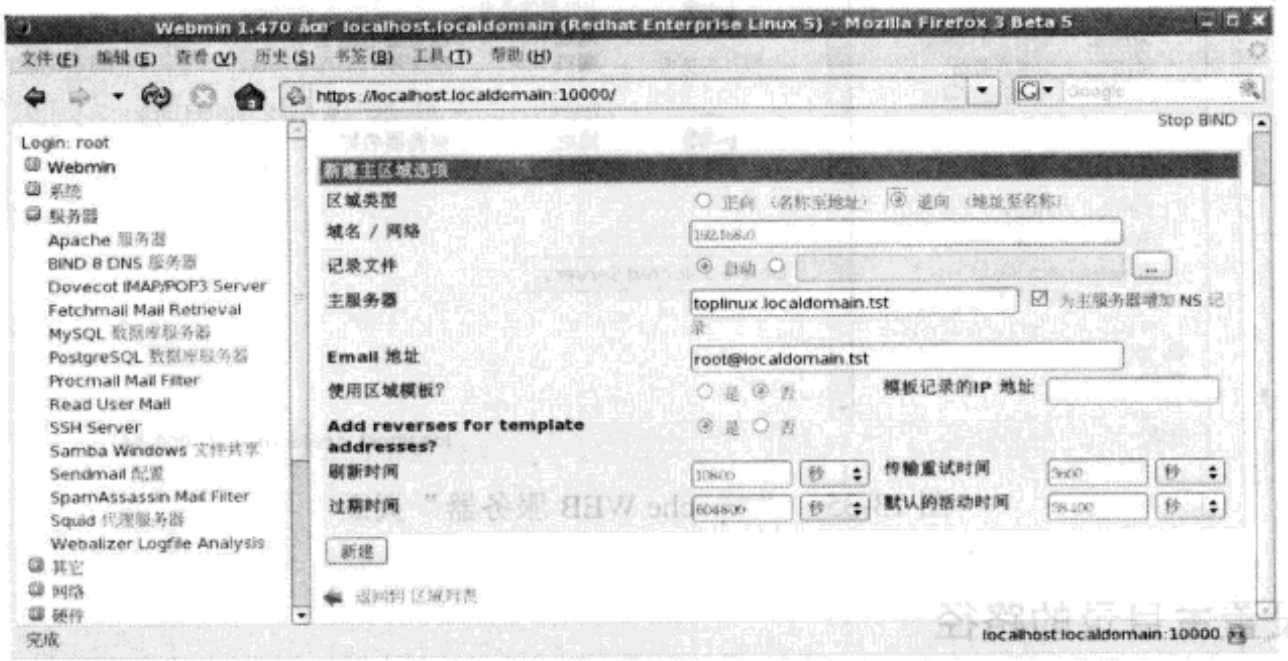


图 13-33　创建逆向主区域

第 9 步，在逆向地址配置页面中（地址为“192.168.0”），单击“逆向地址”超链接，将显示“逆向地址个记录”页面，进行逆向地址设置，如图 13-34 所示。

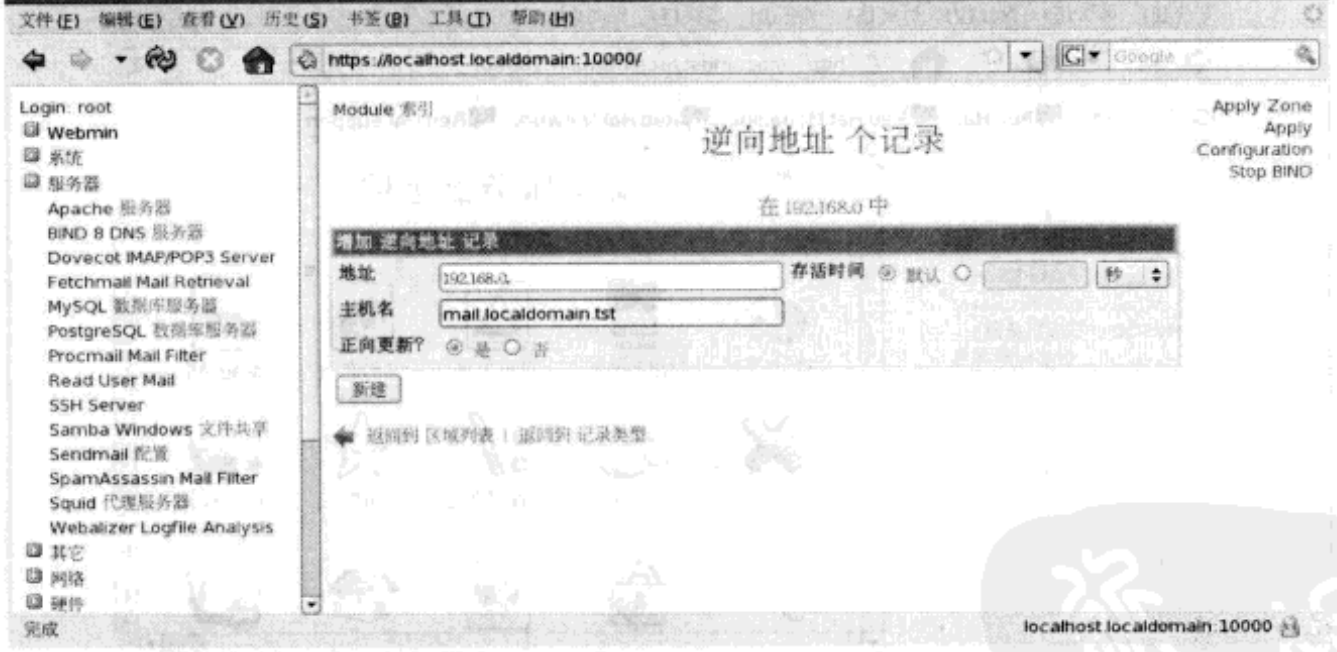


图 13-34　设置逆向 IP 地址

13.3.4 Web 服务的配置

打开 Webmin 管理页面后，依次单击目录区中的“服务器”→“Apache 服务器”超链接，弹出“Apache WEB 服务器”主页面，如图 13-35 所示，在这里进行 Web 服务器的配置工作。

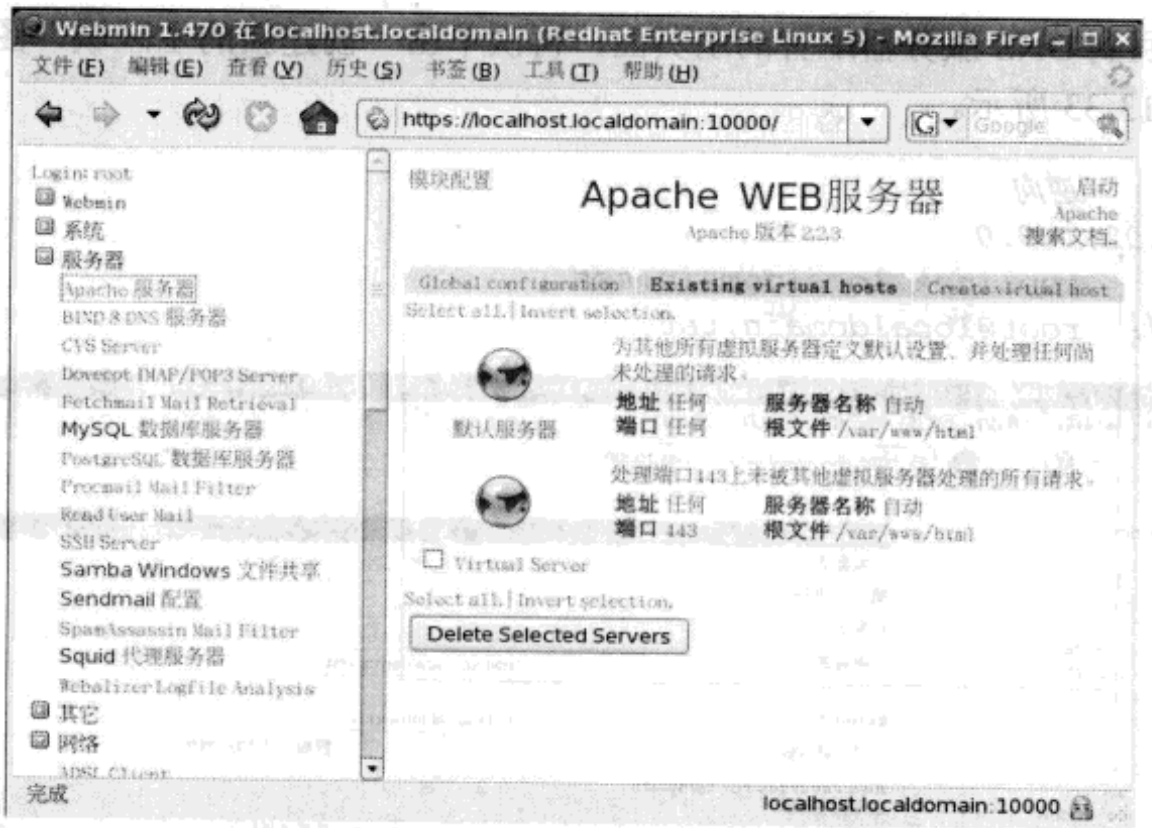


图 13-35 “Apache WEB 服务器”页面

1. 设置主目录的路径

具体的操作步骤如下。

第 1 步，切换至如图所示的“Virtual Server”区域，将显示“虚拟服务器选项”页面，如图 13-36 所示。



图 13-36 设置默认服务器选项

第 2 步，单击“文档选项”按钮，显示“文档选项”设置页面，如图 13-37 所示。

第 3 步，在“文档根目录”文本框中输入要设置的新目录路径“/home/www”，并选择“默认”单选按钮。

第 4 步，单击“保存”按钮，保存设置。

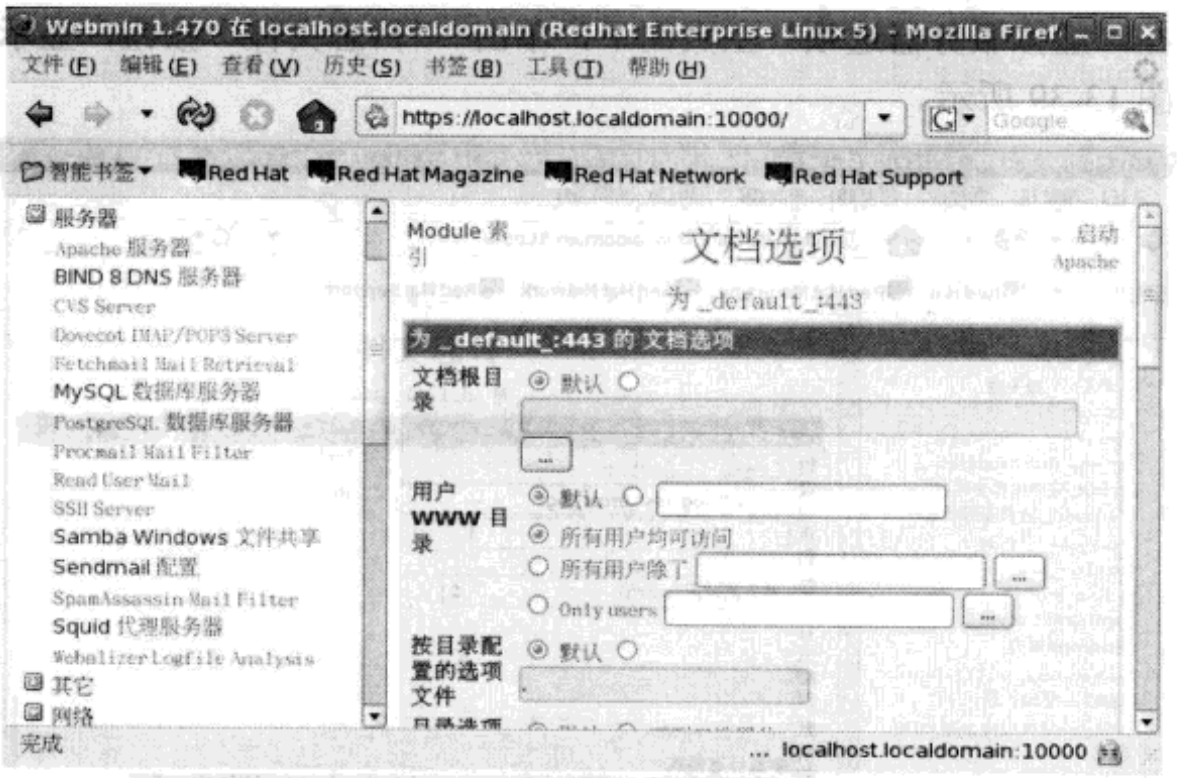


图 13-37 “文档选项”页面

2. 设置默认文档

具体的操作步骤如下。

第 1 步，在“虚拟服务器选项”主界面下，单击 “目录索引”按钮，显示如图 13-38 所示的“目录索引”设置页面。

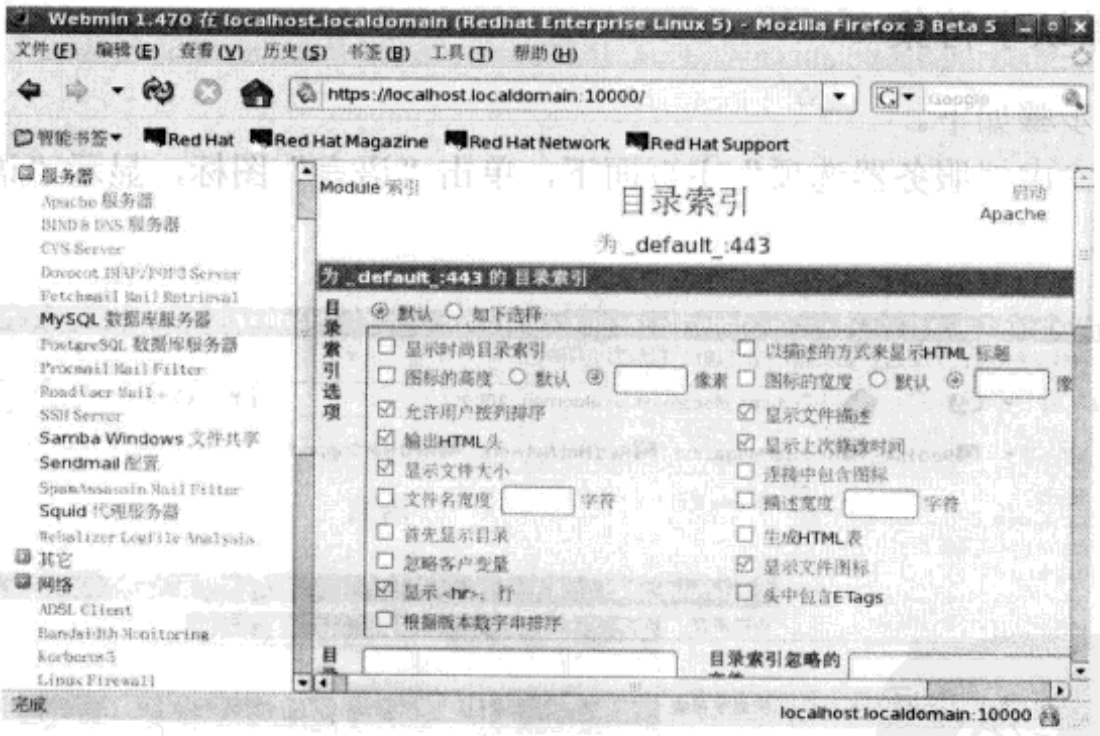


图 13-38 “目录索引”页面

第 2 步，在“目录索引文件”后的文本框中输入要设置的默认文档，如“index.html”，每个默认文档之间用换行符隔开。

第 3 步，最后单击“保存”按钮，保存设置。

3. 设置日志文件

具体操作步骤如下。

第 1 步，在“虚拟服务器选项”主页面下，单击 “日志文件”按钮，显示“日志文件”设置页面，如图 13-39 所示。

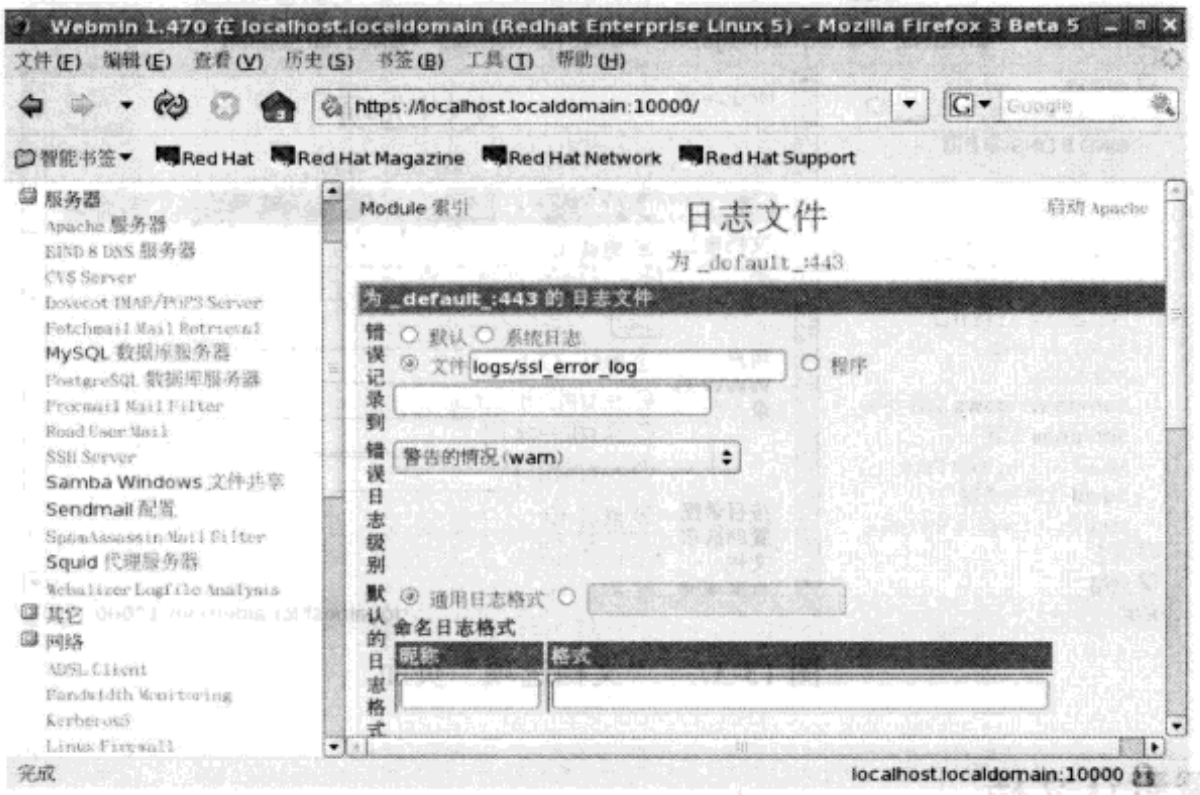


图 13-39 “日志文件”页面

第 2 步，在“默认的日志格式”区域内根据实际需要设置日志格式等信息。

第 3 步，单击“保存”按钮，保存设置。

4. 设置默认字符集

具体的操作步骤如下。

第 1 步，在“虚拟服务器选项”主页面下，单击“语言”图标，显示“语言”设置页面，如图 13-40 所示。

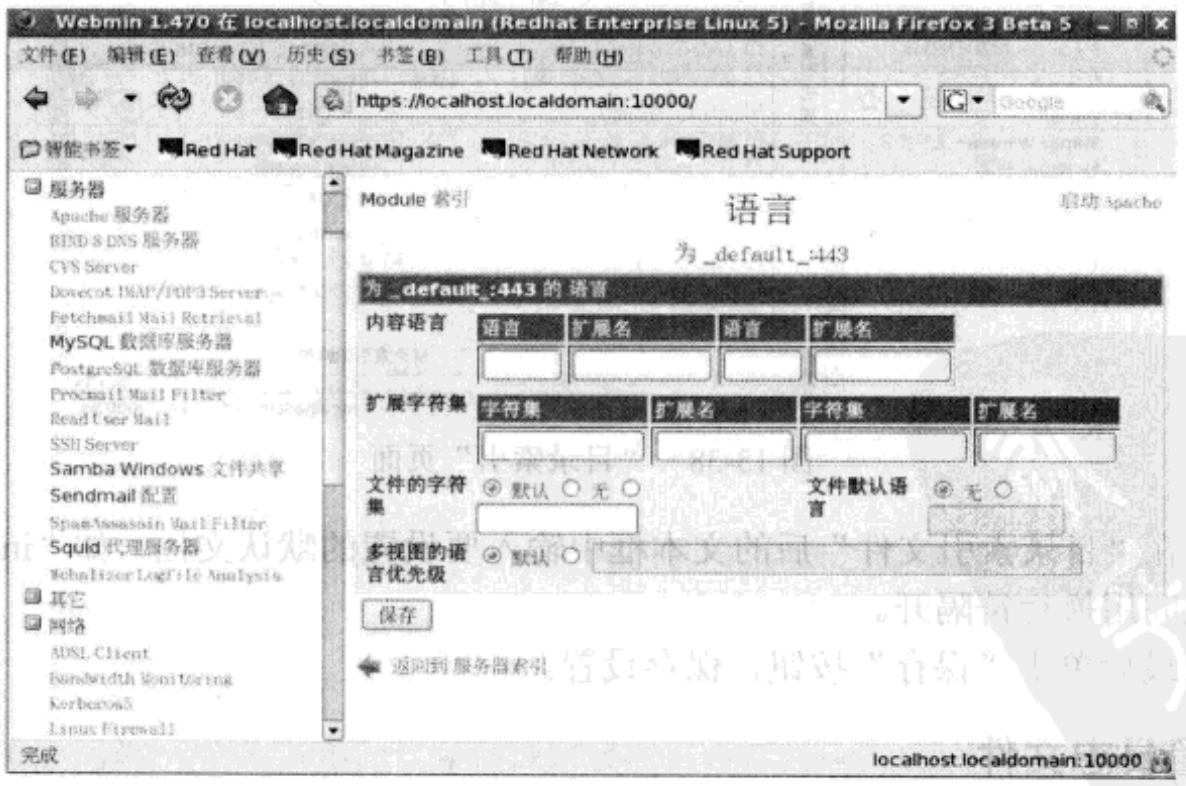


图 13-40 “语言”页面

第 2 步，在“文件的字符集”文本框中输入默认的字符集，如设置默认简体中文则输入“GB2312”。

第 3 步，单击“保存”按钮，保存设置。

5. 创建虚拟目录

具体的操作步骤如下。

第 1 步，在“虚拟服务器选项”主页面下，单击“别名和重定向”图标，显示“别名和重定向”设置页面，如图 13-41 所示。

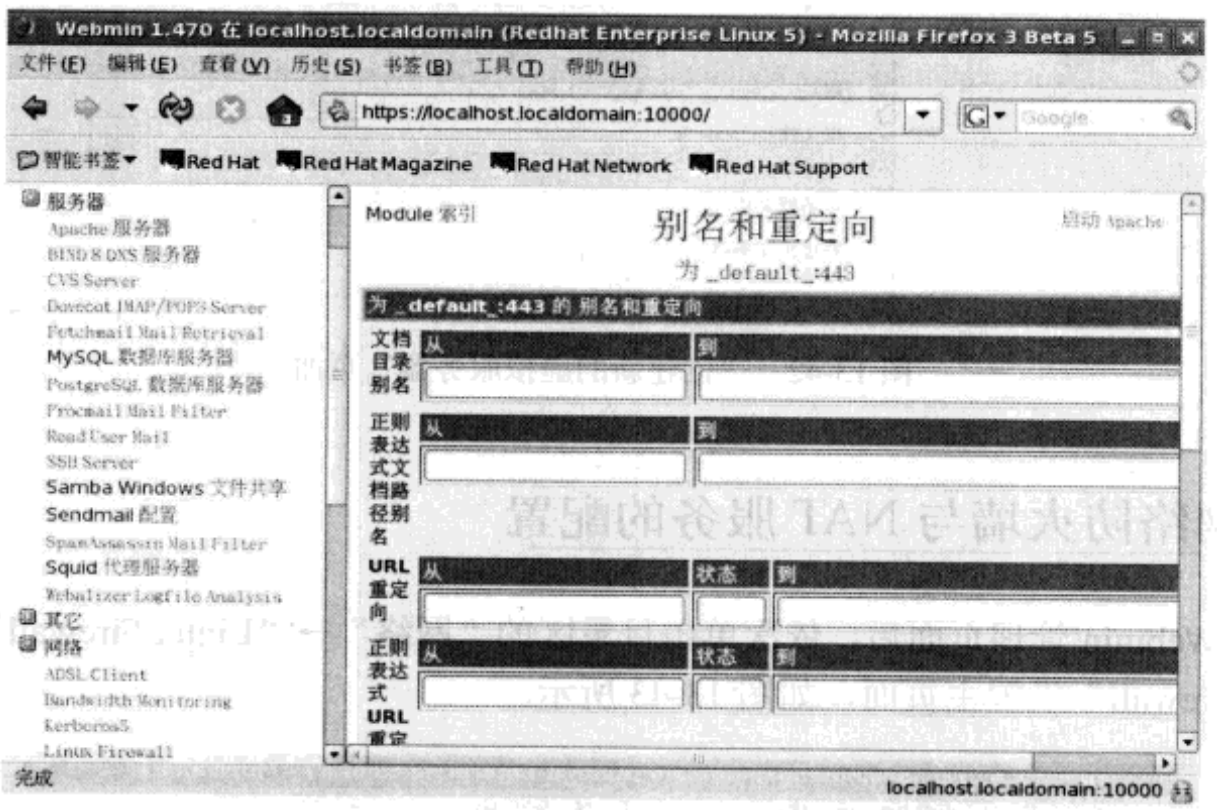


图 13-41 “别名和重定向”页面

第 2 步，在“文档目录别名”区域内的表中输入需要设置的虚拟目录，在“从”一列输入虚拟目录路径，在“到”一列输入对应的物理路径。

第 3 步，单击“保存”按钮，保存设置。

6. 虚拟主机的配置

配置虚拟主机的具体步骤如下。

第 1 步，在“Apache WEB 服务器”管理主页面下，选择“Create virtual host”选项卡，显示“创建新的虚拟服务器”页面，如图 13-42 所示。

第 2 步，在“处理到地址的连接”区域内，选择“指定地址…”单选按钮，并在文本框中输入虚拟主机的 IP 地址。

第 3 步，在“端口”区域内，选中“任何”单选按钮，并在文本框中输入端口号。在“根文件”文本框中输入虚拟主机的主目录。

第 4 步，在“服务器名称”区域内，选中“自动”单选按钮，并输入服务器的域名。

第 5 步，单击“Create Now”按钮，创建新的虚拟主机。

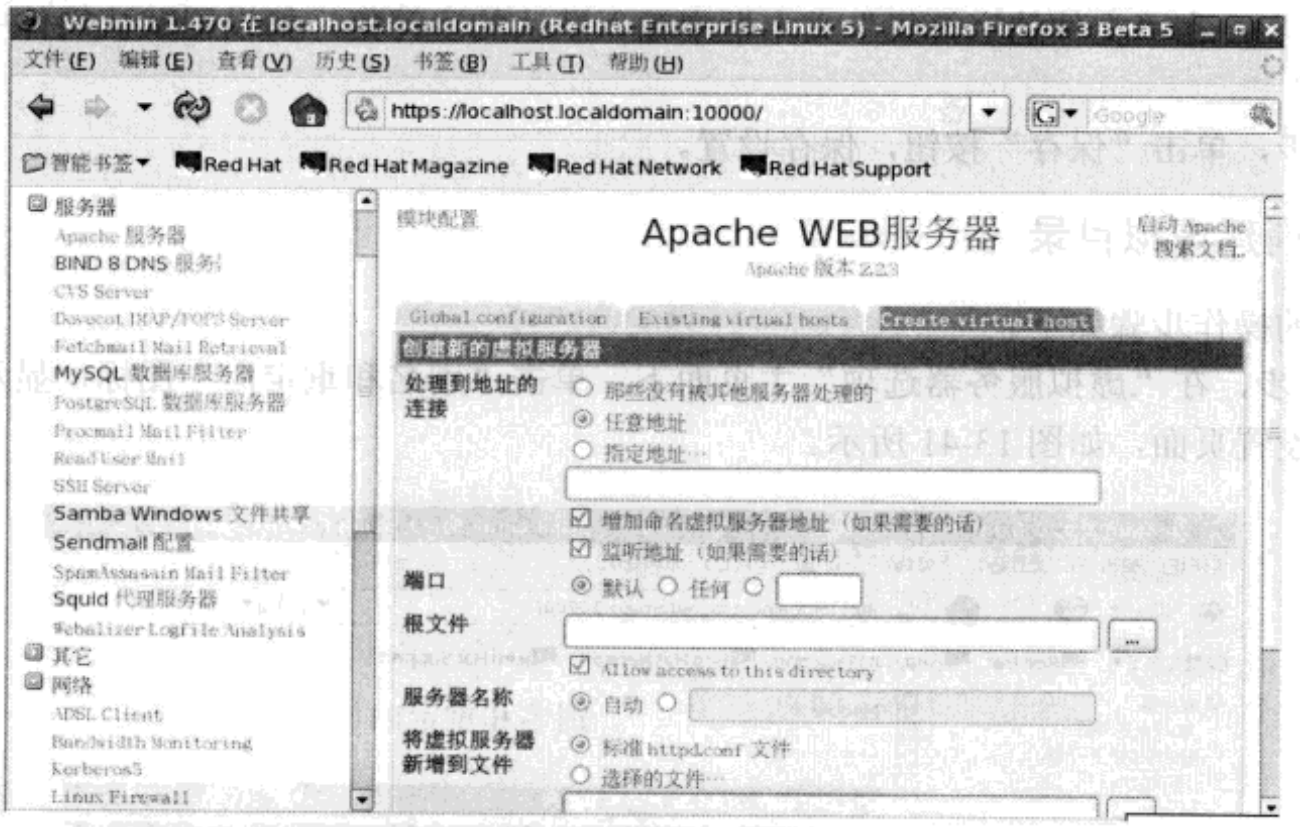


图 13-42 “创建新的虚拟服务器”页面

13.3.5 网络防火墙与 NAT 服务的配置

在打开 Webmin 管理页面后，依次单击目录区的“网络”→“Linux Firewall”超链接，显示“Linux Firewall”设置主页面，如图 13-43 所示。

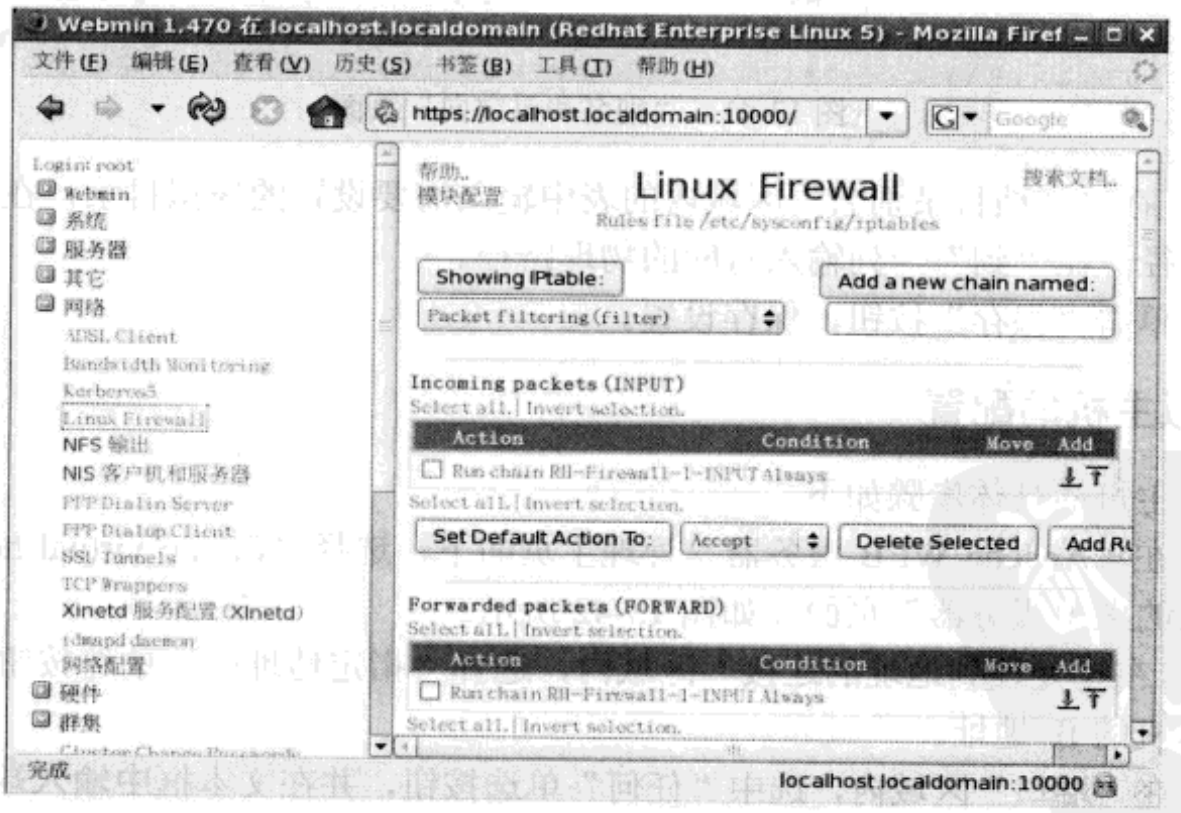


图 13-43 “Linux Firewall”页面

1. 禁止客户机访问不健康网站

禁止客户机访问不健康网站的具体步骤如下。

第 1 步，在“Linux Firewall”主页面的“Forwarded packets（FORWARD）”区域内，单击“Add Rule”按钮，如图 13-44 所示。

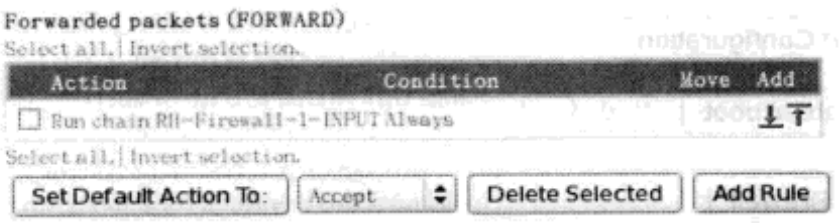


图 13-44　打开 FORWARD 链规则设置界面

第 2 步，显示“Add Rule”配置页面后，在“Rule comment”文本框中输入规则名称。在“Action to take”区域内，选择“Drop”单选按钮，设置动作选项为 Drop，如图 13-45 所示。

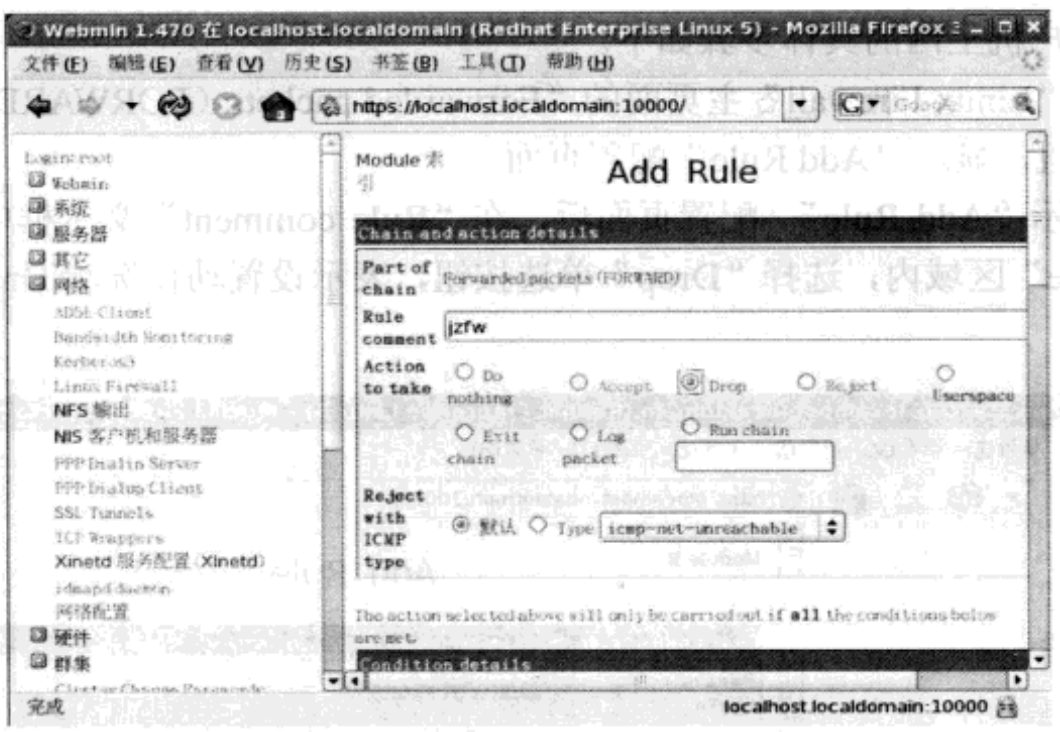


图 13-45　设置规则名称和动作选项

第 3 步，选择“Destination address or network”区域的下拉列表，然后设置值为“Equals”，并在其后的文本框中输入目的地址（禁止访问的网址），如图 13-46 所示。

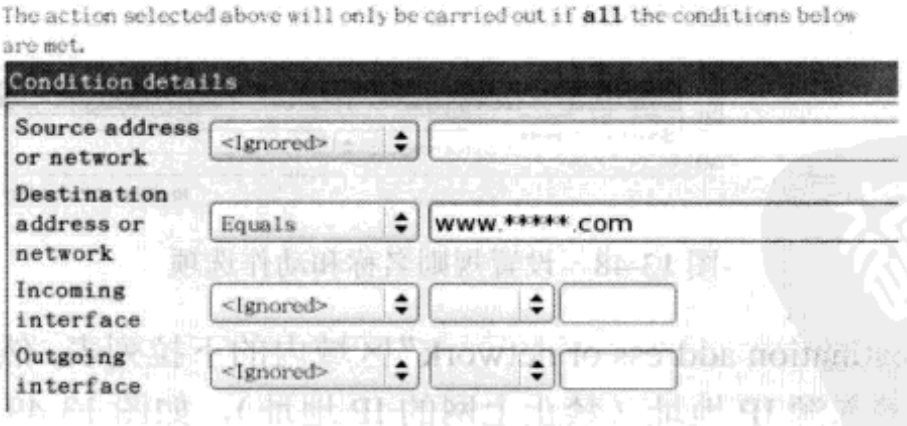


图 13-46　设置规则的目的地址

第 4 步，单击该界面最下方的“新建”按钮，返回“Linux Firewall”页面，然后单击下方的“Apply Configuration”按钮应用规则，如图 13-47 所示。

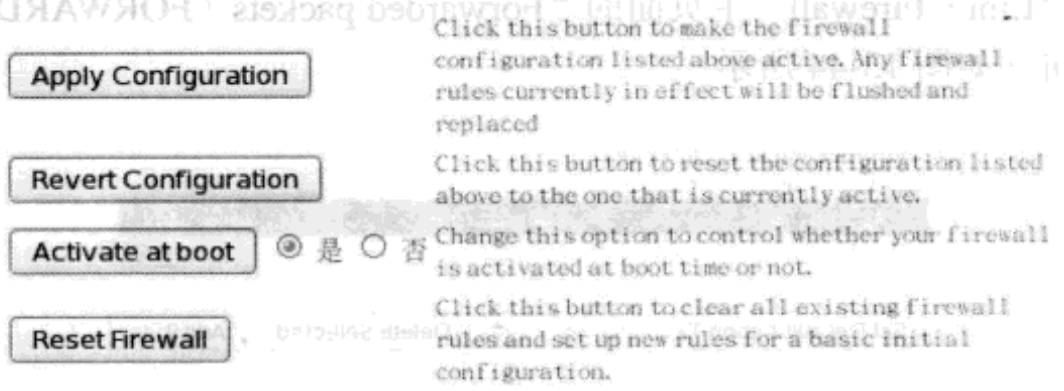


图 13-47　单击“Apply Configuration”按钮应用规则

2. 禁止某些客户机上网

禁止指定客户机上网的具体步骤如下。

第 1 步，在“Linux Firewall”主页面的“Forwarded packets（FORWARD）”区域内，单击“Add Rule”按钮，显示“Add Rule”配置页面。

第 2 步，显示“Add Rule”配置页面后，在“Rule comment”文本框中输入规则名称。在“Action to take”区域内，选择“Drop”单选按钮，表示设置动作选项 Drop，如图 13-48 所示。

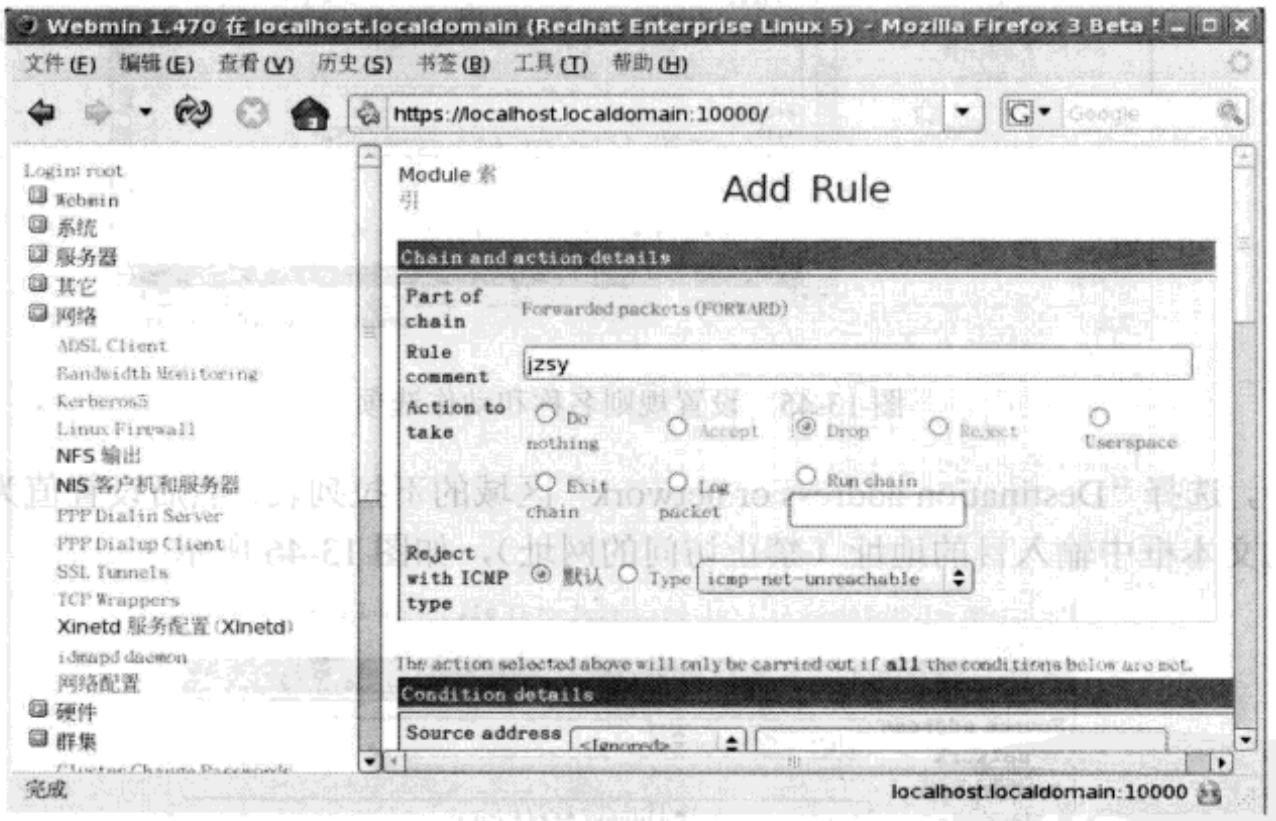


图 13-48　设置规则名称和动作选项

第 3 步，选择“Destination address or network”区域内的下拉列表，然后选择值为“Equals”，并在其后的文本框中输入源 IP 地址（禁止上网的 IP 地址），如图 13-49 所示。

第 4 步，单击最下方的“新建”按钮，返回“Linux Firewall”页面，然后单击下方的“Apply Configuration”按钮应用规则。

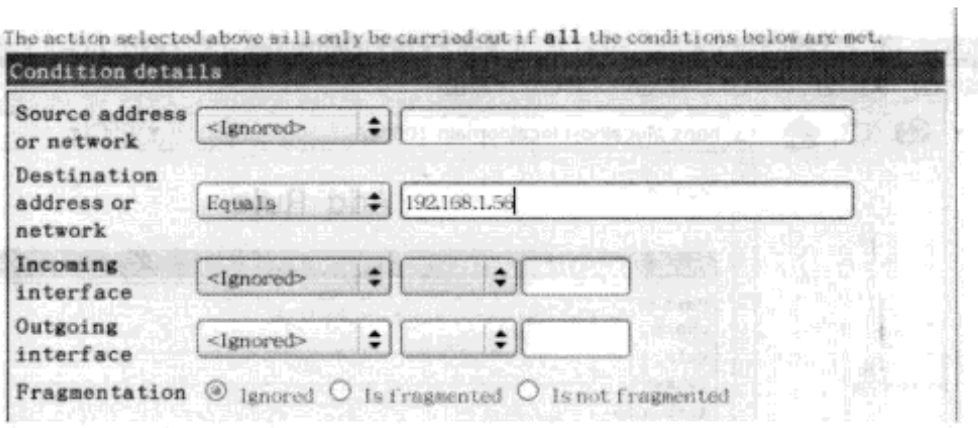


图 13-49　设置规则的源地址

3. 强制访问指定的站点

强制访问指定的站点，具体操作步骤如下。

第 1 步，首先打开“Linux Firewall”主页面，选择“Showing IPtable”的值为“Network address translation”，然后单击“Showing IPtable”按钮，如图 13-50 所示。

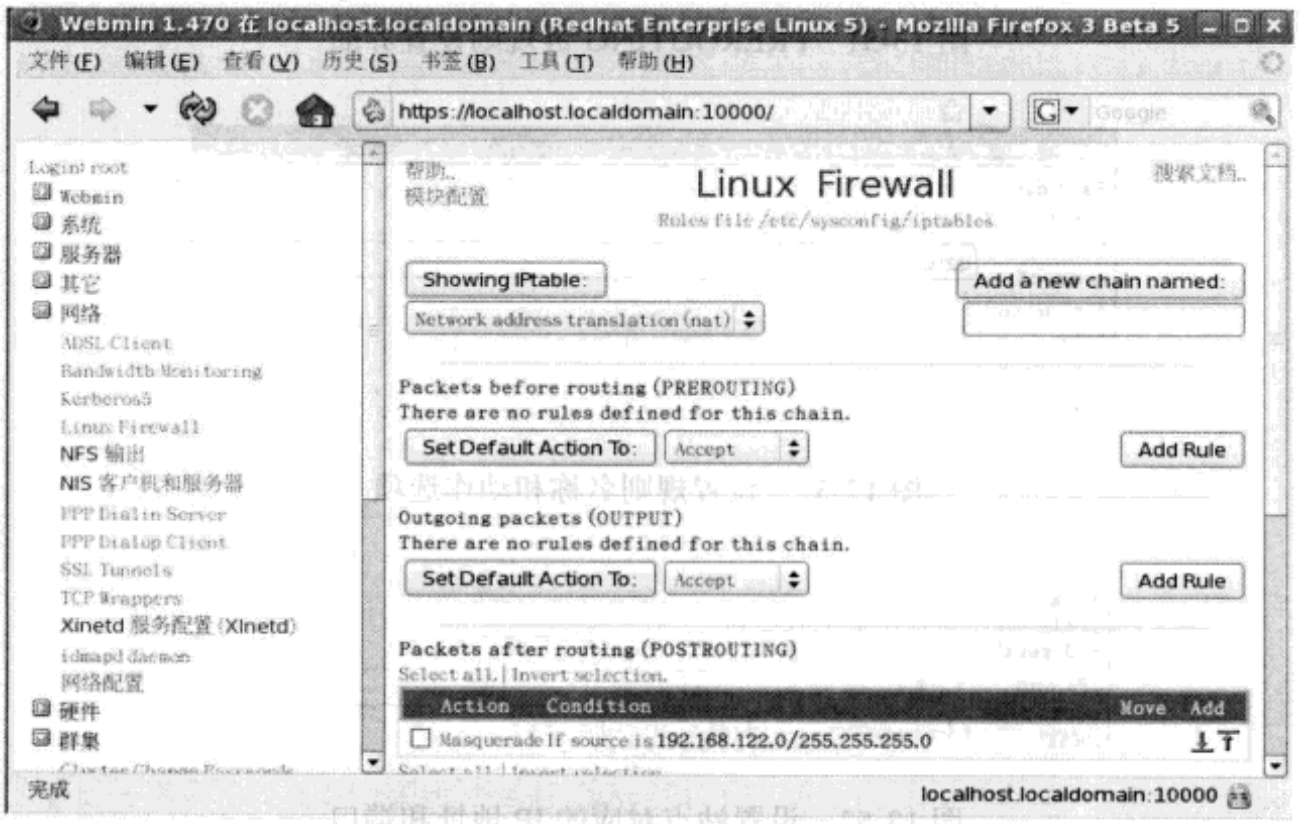


图 13-50　“Linux Firewall”主界面

第 2 步，在 “Packets before routing（PREROUTING）”区域内，单击“Add Rule”按钮，显示“Add Rule”页面，如图 13-51 所示。

第 3 步，在“Rule comment”文本框中输入规则名称，在“Action to take”区域内，选择“Destination NAT”单选按钮，设置动作选项为 DNAT，如图 13-52 所示。

第 4 步，在“IPs and ports for DNAT”区域内，选择“IP range”单选按钮，然后在文本框中输入对应的 IP 地址和端口，如图 13-53 所示。

第 5 步，在“Incoming interface”区域内选择值为“Equals”和“eth0”，如图 13-54 所示。

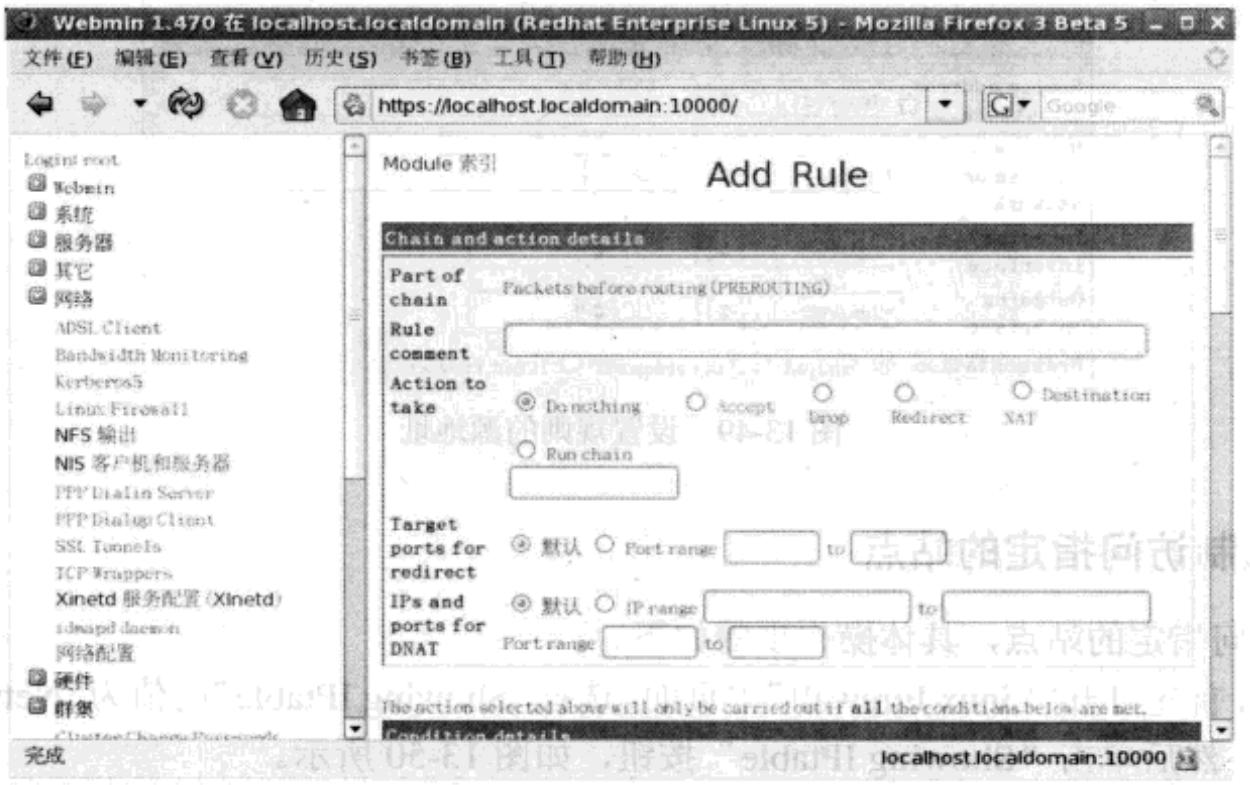


图 13-51　PREROUTING 链规则设置界面

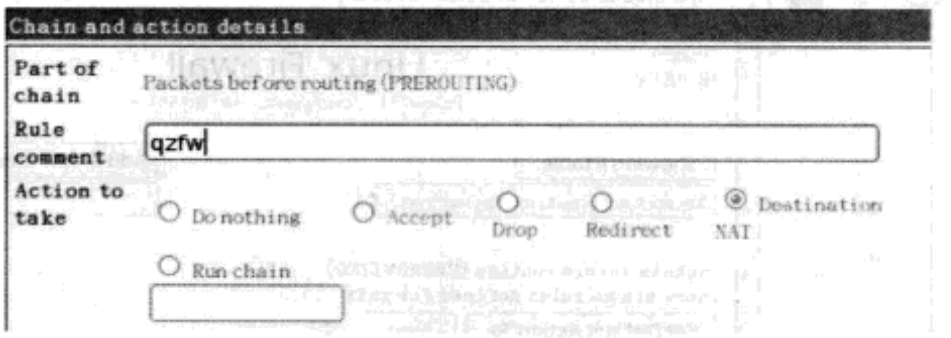


图 13-52　设置规则名称和动作选项

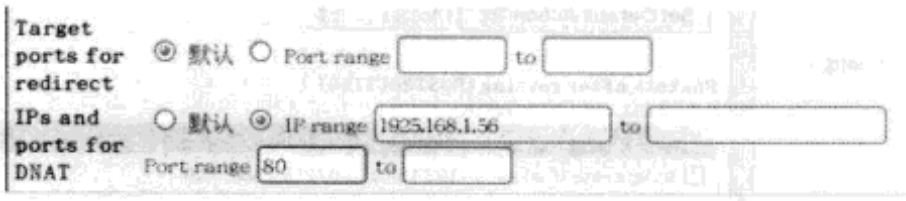


图 13-53　设置站点对应的 IP 地址和端口

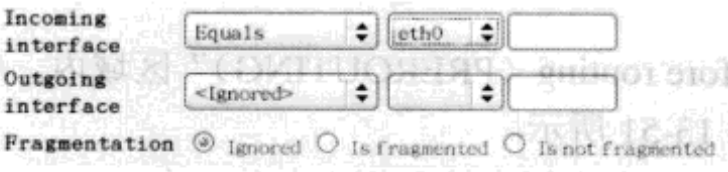


图 13-54　设置规则数据的显示接口

第 6 步，在“Network protocol”区域内设置值为“Equals”，并选择“TCP”协议。然后在“Destination TCP or UDP port”区域内设置该选项的值为“Equals”，并在“Port（s）”后的文本框中输入匹配端口号“80”，如图 13-55 所示。

第 7 步，单击最下方的“新建”按钮，返回“Linux Firewall”页面，然后单击下方的“Apply Configuration”按钮，应用规则。

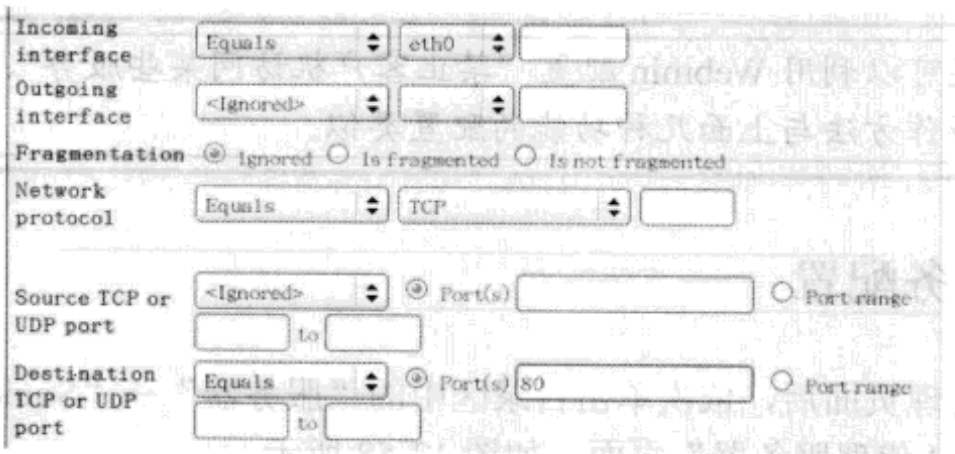


图 13-55　设置规则的目标 TCP 端口号

4. 配置内部网络服务器

配置内部网络服务器的具体操作步骤如下。

第 1 步，在“Linux Firewall”主页面的“Packets before routing（PREROUTING）”区域内，单击“Add Rule”按钮，显示“Add Rule”页面。

第 2 步，在“Rule comment”文本框中输入规则名称，在“Action to take”区域内，选择“Destination NAT”单选按钮，设置动作选项为 DNAT。

第 3 步，在“IPs and ports for DNAT”区域内，选择“IP range”单选按钮，然后在文本框中输入服务器对应的 IP 地址和端口。

第 4 步，在“Incoming interface”区域内，从第 1 个下拉列表中选择其值为“Equals”，从第 2 个下拉列表中选择“others”，在其后的文本框中输入“ppp0”。

第 5 步，在“Outgoing interface”区域内选择其值为“Equals”和“eth0”。在“Network protocol”区域内设置该选项的值为“Equals”，并选择“TCP”协议，如图 13-56 所示。

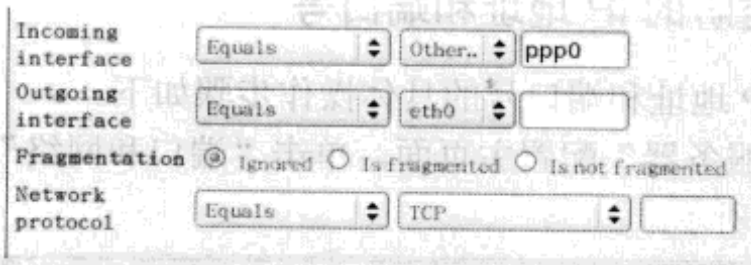


图 13-56　设置规则的数据进出接口和协议

第 6 步，在“Destination TCP or UDP port”区域内设置值为“Equals”，并在“Port（s）”后的文本框中输入匹配端口号“80”，如图 13-57 所示。

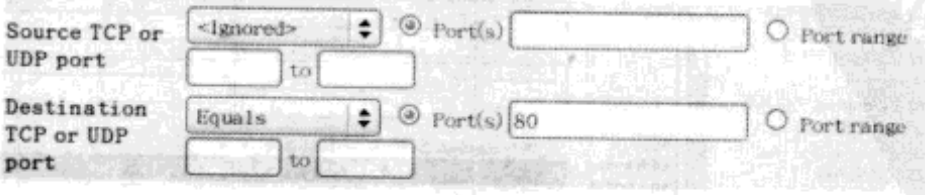


图 13-57　设置规则的目的 TCP 端口号

第 7 步，单击最下方的“新建”按钮，返回“Linux Firewall”主页面，单击下方的“Apply Configuration”按钮，应用规则。

注意　另外，还可以利用 Webmin 配置“禁止客户机访问某些服务”、“禁止使用 ICMP 协议”等功能，其操作方法与上面几种功能的配置类似。

13.3.6 代理服务配置

打开 Webmin 管理页面后，依次单击目录区中的“服务器”→“Squid 代理服务器”超链接，即可显示“Squid 代理服务器”页面，如图 13-58 所示。



图 13-58　“Squid 代理服务器”页面

1. 设置代理服务器监听的 IP 地址和端口号

设置代理服务器监听的 IP 地址和端口号的具体操作步骤如下。

第 1 步，在“Squid 代理服务器”配置主页面，单击“端口和网络”图标，进入“端口和联网”配置页面，如图 13-59 所示。

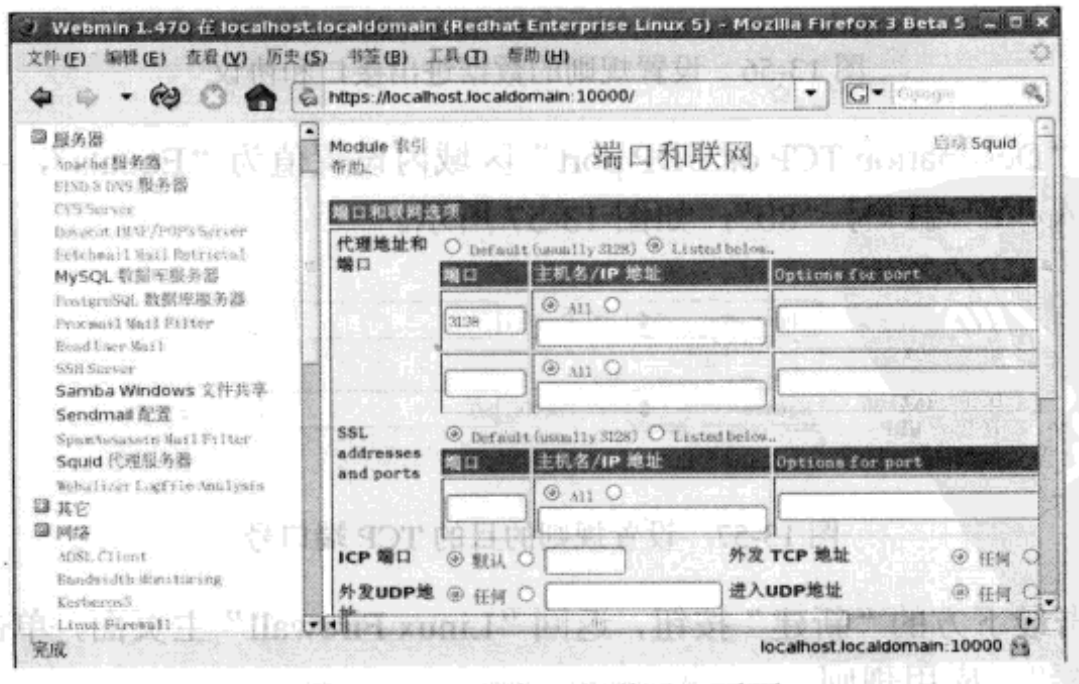


图 13-59　“端口和联网”页面

第 2 步，在“代理地址和端口”区域内，选择“Listed below…”单选按钮。在“端口”列的第 1 行中输入监听的端口号。

第 3 步，在“主机名/IP 地址”列的第 1 行中选择“All”单选按钮，并在文本框中输入监听的 IP 地址。

第 4 步，单击“保存”按钮，保存设置。

2. 设置内存缓冲大小

设置内存缓冲大小的具体操作步骤如下。

第 1 步，在“Squid 代理服务器”配置主页面，单击“内存使用”图标，显示“内存使用”配置页面，如图 13-60 所示。

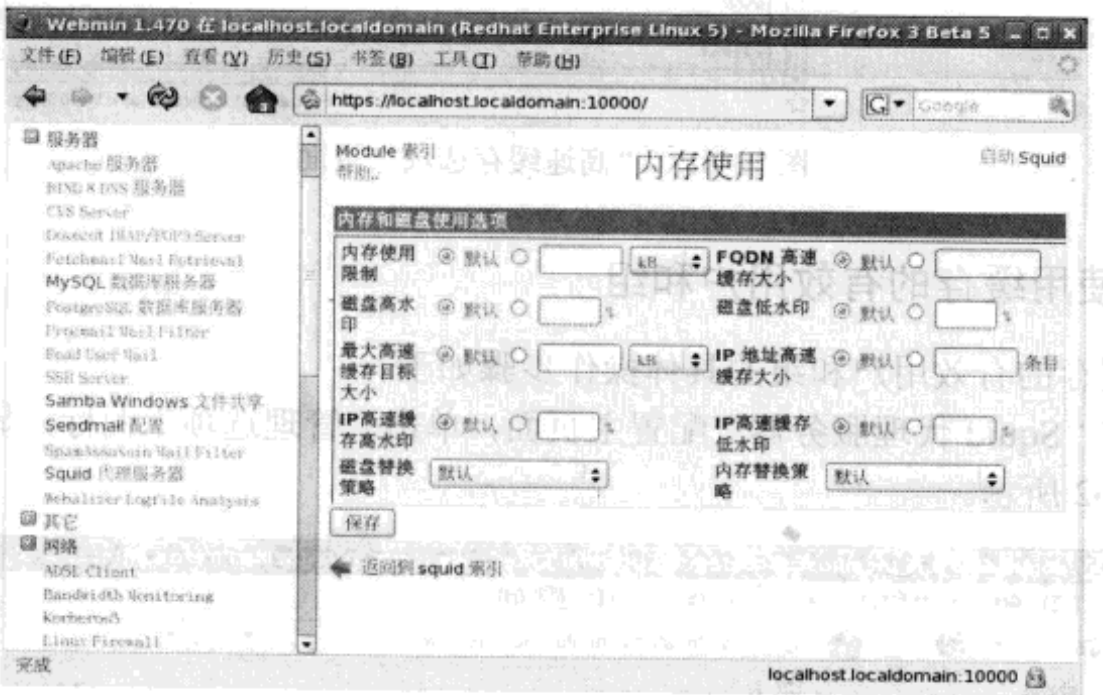


图 13-60　设置内存缓冲的大小

第 2 步，在“内存使用限制”区域内，选择“默认”单选按钮，在后面的下拉列表中选择单位“MB”，然后在文本框中输入想要设定的内存大小。

第 3 步，单击“保存”按钮，保存设置。

3. 设置硬盘缓冲的大小

设置硬盘缓冲大小的具体操作步骤如下。

第 1 步，在“Squid 代理服务器”配置主页面，单击“高速缓存选项”图标，显示“高速缓存选项”配置页面，如图 13-61 所示。

第 2 步，在“高速缓存目录”区域内，选择“列出的”单选按钮。

第 3 步，在“目录”列的第 1 行中配置缓冲的目录；在“类型”列的下拉列表选择缓冲的存储类型；在“大小（MB）”列的文本框中输入缓冲大小的值。然后设置“一级目录”和“二级目录”的值。

第 4 步，单击“保存”按钮，保存设置。



图 13-61 “高速缓存选项”页面

4. 设置使用缓存的有效用户和组

设置使用缓存的有效用户和组的具体操作步骤如下。

第 1 步，在“Squid 代理服务器”配置主页面，单击“管理选项”图标，显示“管理选项”页面，如图 13-62 所示。

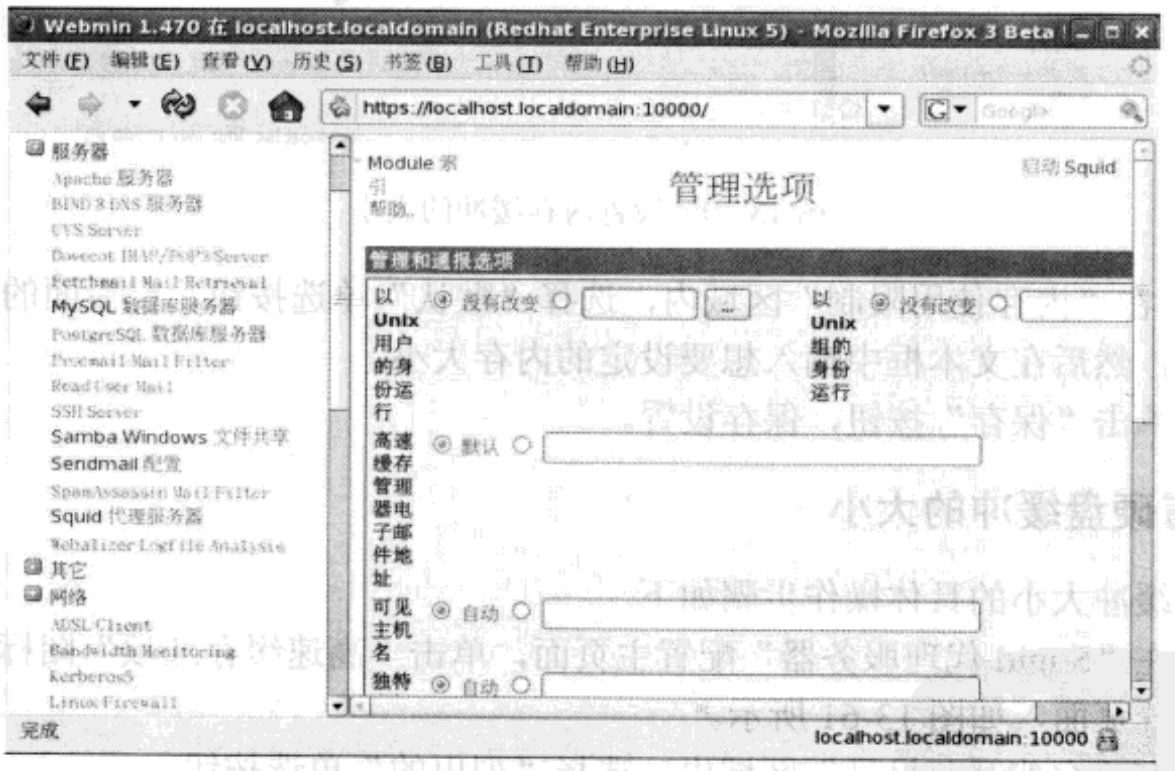


图 13-62 “管理选项”页面

第 2 步，在“以 UNIX 用户的身份运行”区域内，选择“没有改变”单选按钮，并选择一个缓存的有效用户。

第 3 步，在“以 UNIX 组的身份运行”区域内，选择“没有改变”单选按钮，并选择一个缓存的有效组。

第 4 步，单击“保存”按钮，保存设置。

5. 定义 DNS 服务器地址

定义 DNS 服务器地址的具体操作步骤如下。

第 1 步，在“Squid 代理服务器”配置主页面，单击“帮助程序”图标，显示“帮助程序”页面，如图 13-63 所示。



图 13-63 “帮助程序”页面

第 2 步，在“DNS 服务器地址”区域内，选择“来自 resolv.conf”单选按钮，然后在文本框中输入 DNS 服务器的 IP 地址。

第 3 步，单击“保存”按钮，保存设置。

6. 设置日志文件

设置日志文件的具体操作步骤如下。

第 1 步，在“Squid 代理服务器”配置主页面，单击“登录”图标，显示“登录”页面，如图 13-64 所示。

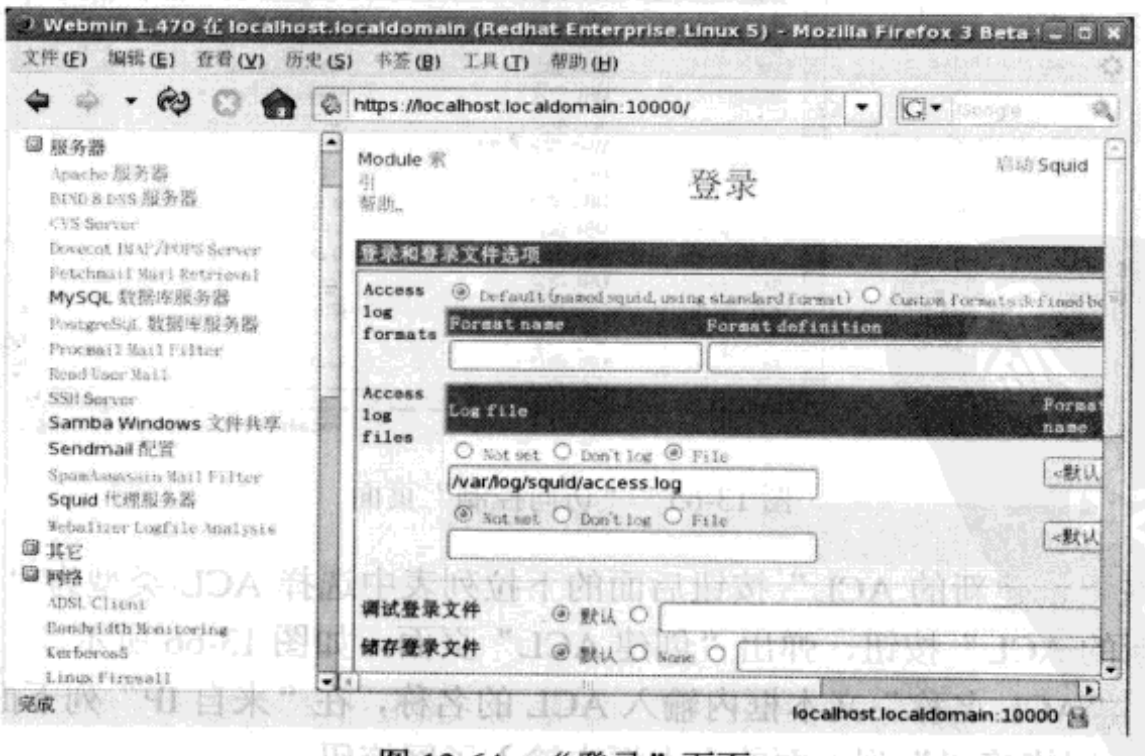


图 13-64 “登录”页面

第 2 步，在“Access logfiles”区域内，选择“File”单选按钮，在其后的文本框中输入访问日志文件名和存放路径。

第 3 步，在“调试登录文件”区域内，选择“默认”单选按钮，在其后的文本框中输入缓存日志文件名和存放路径。

第 4 步，在“储存登录文件”区域内，选择“默认”单选按钮，在其后的文本框中输入网页缓存日志文件名和存放路径。

第 5 步，单击“保存”按钮，保存设置。

7. 设置运行 Squid 主机的名称和管理员的 E-mail 地址

设置运行 Squid 主机的名称和管理员的 E-mail 地址的具体操作步骤如下。

第 1 步，在“Squid 代理服务器”配置主页面，单击“管理选项”图标，显示“管理选项”页面。

第 2 步，在“高速缓存管理器电子邮件地址”区域内，选择“默认”单选按钮，并输入管理员的 E-mail 地址。

第 3 步，在“可见主机名”区域内，选择“自动”单选按钮，并输入主机的 IP 地址。

第 4 步，单击“保存”按钮，保存设置。

8. 设置默认访问控制列表

设置默认访问控制列表的具体操作步骤如下。

第 1 步，在“Squid 代理服务器”配置主页面，单击“访问控制”图标，显示“访问控制”页面，如图 13-65 所示。

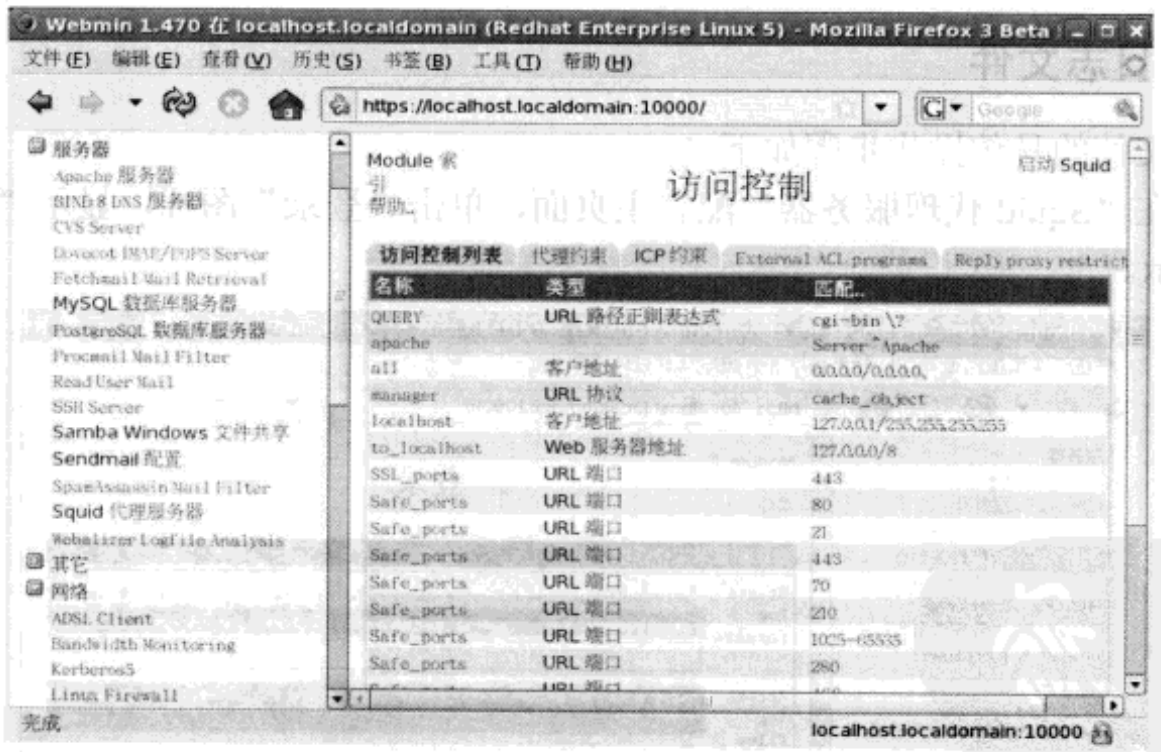


图 13-65 “访问控制”页面

第 2 步，在“创建新的 ACL”按钮后面的下拉列表中选择 ACL 类型为“客户地址”，然后单击“创建新的 ACL”按钮，弹出“创建 ACL”页面，如图 13-66 所示。

第 3 步，在“ACL 名称”文本框内输入 ACL 的名称，在“来自 IP”列下面的文本框中输入 IP 地址，在“网络掩码”列下面的文本框中输入子网掩码。

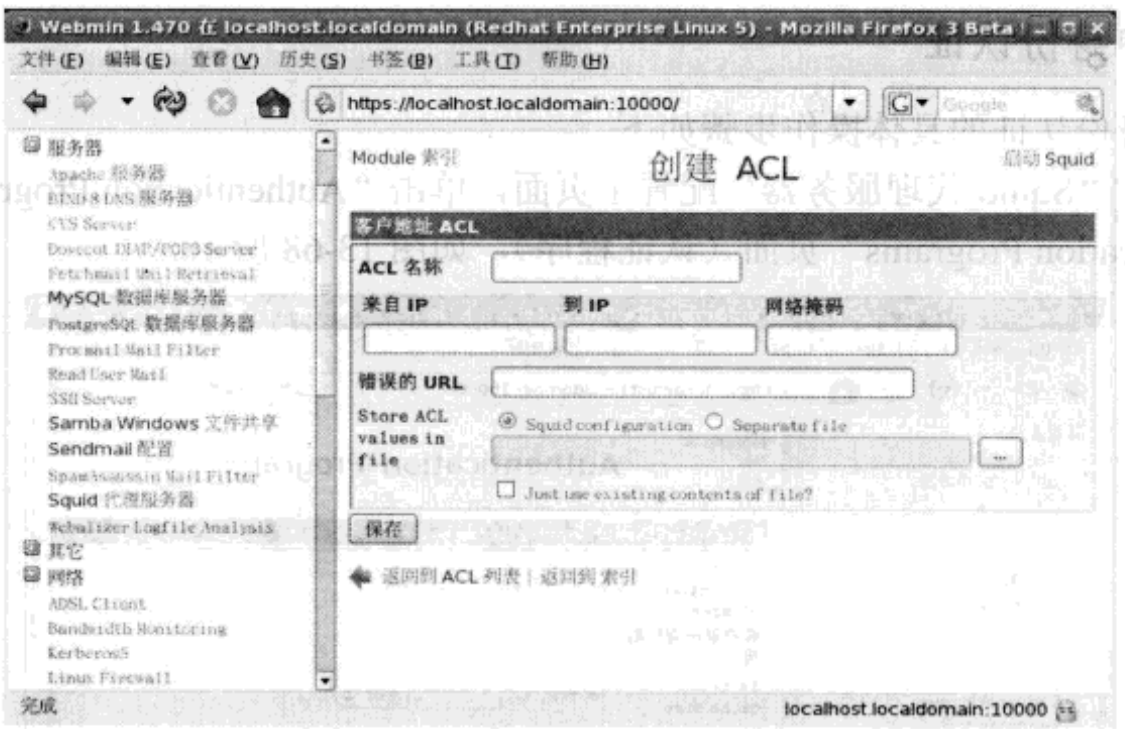


图 13-66 “创建 ACL”页面

第 4 步，单击“保存”按钮，保存设置。

第 5 步，在“Reply proxy restrictions”区域内，单击“添加代理约束规则”超链接，显示“Creat Proxy Reply Restriction”页面，如图 13-67 所示。

第 6 步，在“动作”区域内选择“拒绝”单选按钮，然后在“匹配 ACL”区域中选择“all”项。

第 7 步，单击“保存”按钮保存设置。

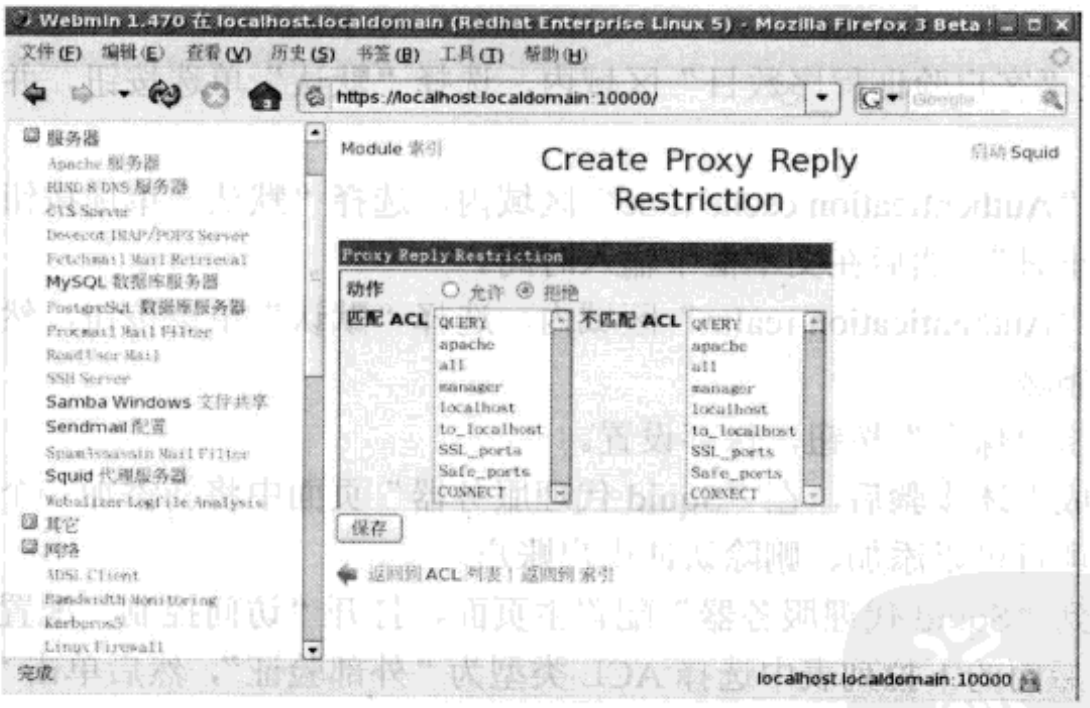


图 13-67 创建默认约束规则

9. 初始化 Squid

在“Squid 代理服务器”配置主页面，单击最上方“初始化高速缓存”按钮，Webmin 将自动帮助用户完成初始化工作。

10. 用户身份认证

配置用户身份认证的具体操作步骤如下。

第 1 步，在“Squid 代理服务器”配置主页面，单击“Authentication Programs”图标，将显示“Authentication Programs”页面（认证程序），如图 13-68 所示。

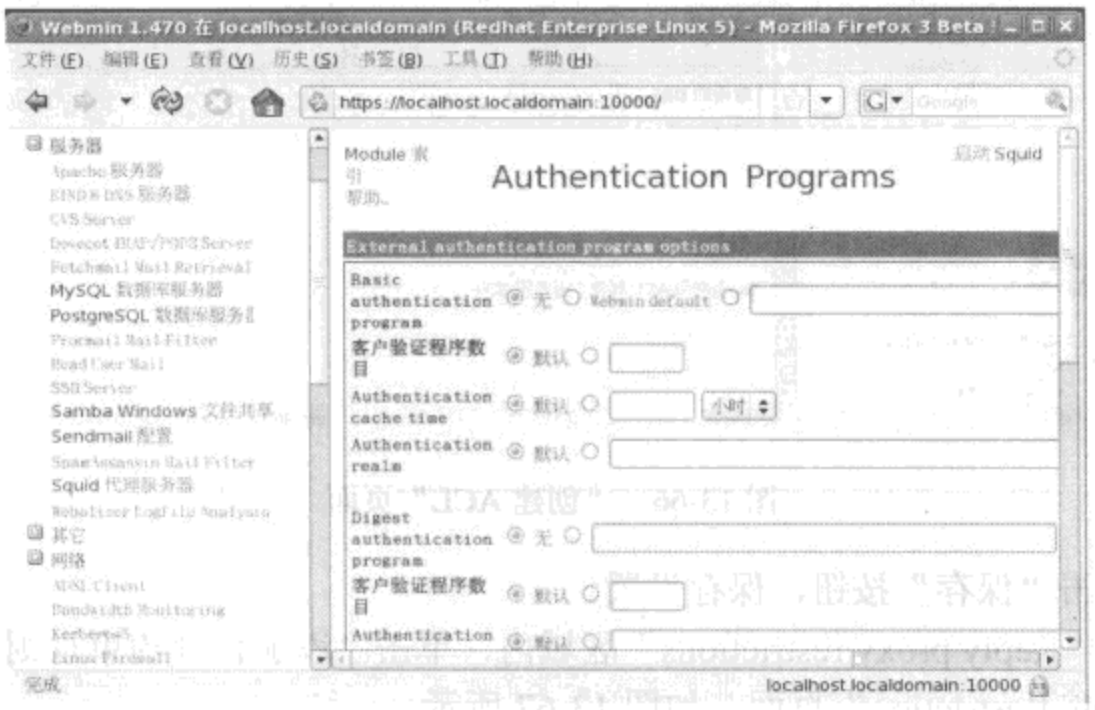


图 13-68 “Authentication Programs”图标

第 2 步，在“Basic authentication program”区域内，选择“Webmin default”单选按钮，并输入认证程序的路径以及认证程序需要读取的账户文件“/usr/lib/squid/ncsa.auth /etc/squid/passwd”。

第 3 步，在“客户验证程序数目”区域内，选择“默认”单选按钮，并在该区域的文本框中输入数字。

第 4 步，在“Authentication cache time”区域内，选择“默认”单选按钮，在单位下拉列表中选择单位“小时”，然后在文本框中输入时间。

第 5 步，在“Authentication realm”区域内，选择“默认”单选按钮，然后在该区域的文本框中输入认证领域。

第 6 步，单击“保存”按钮，保存设置。

第 7 步，完成上述步骤后，在“Squid 代理服务器”页面中将新添加一个“代理验证”管理项目，利用该项目可以添加、删除认证用户账户。

第 8 步，回到“Squid 代理服务器”配置主页面，打开“访问控制”配置页面。在“创建新的 ACL”按钮后面的下拉列表中选择 ACL 类型为“外部验证”，然后单击“创建新的 ACL”按钮，显示“创建 ACL”页面，如图 13-69 所示。



图 13-69 创建外部验证 ACL

第 9 步，在“创建 ACL”界面下的“ACL 名称”区域内，输入外部验证的名称，单击“保

存”按钮，保存设置。

第 10 步，在“访问控制”配置页面下的“代理约束”区域内，单击“添加代理约束规则”超链接，显示“创建代理约束”页面，在“动作”区域内，选择“允许”单选按钮，并选择“匹配 ACL”后面的列表框中刚才设置的外部验证的名称项。最后单击“保存”按钮，保存设置。

13.3.7 SSH 服务的配置

打开 Webmin 管理界面后，单击目录区中的“服务器”图标，然后单击“SSH Server”图标，将显示“SSH 服务器”页面，如图 13-70 所示。



图 13-70 “SSH 服务器”页面

1. SSH 服务的网络设置

在 SSH 服务配置主页面中单击“网络”图标，显示“网络”页面，如图 13-71 所示。



图 13-71 SSH 服务器的网络设置

为了安全，可以设置只使用 SSH2 协议，然后单击“保存”按钮保存设置。

2. SSH 的认证方式

设置 SSH 的认证方式的操作步骤如下。

第 1 步，在 SSH 服务配置主界面中单击“访问控制”图标，将显示“访问控制”页面，如图 13-72 所示。

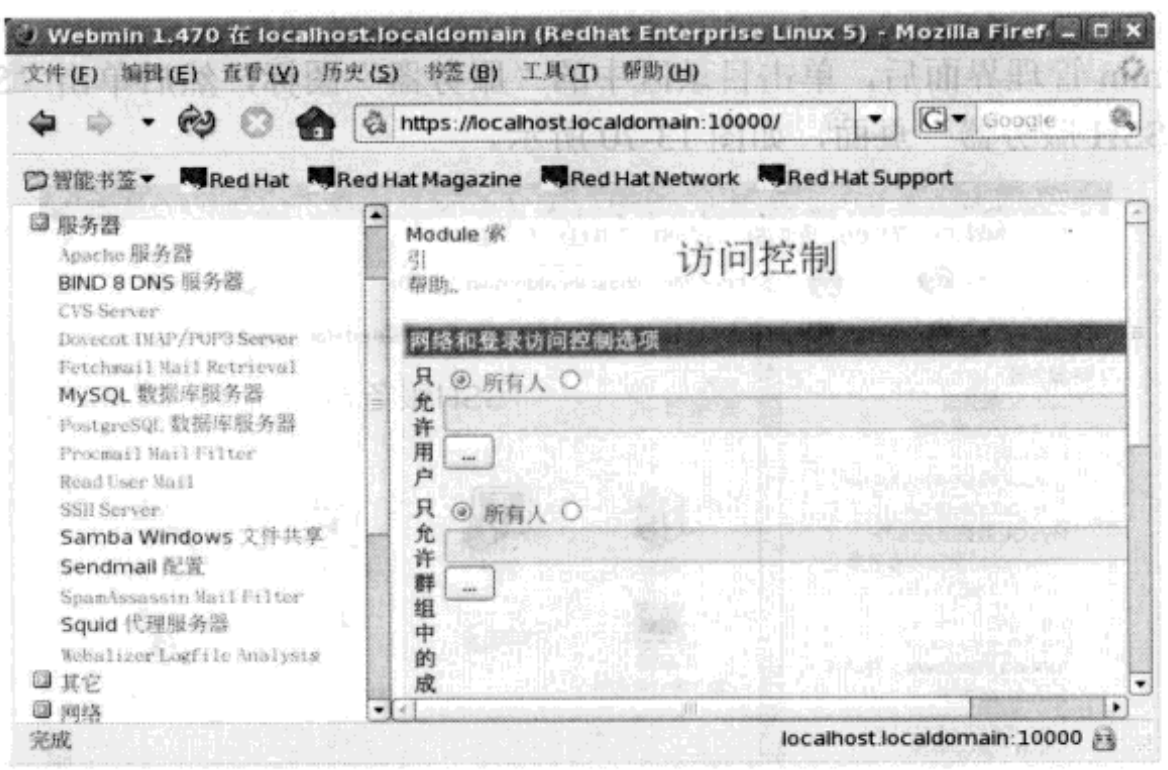


图 13-72　SSH 服务器的访问控制

第 2 步，设置只有特定用户可以通过 SSH 服务器登录，在“只允许用户”区域内，选择“所有人”单选按钮，并在该区域的文本框中输入允许的用户账户。

第 3 步，单击“保存”按钮，保存设置。

第 14 章　Linux 命令行

操作系统的核心功能是管理和控制计算机硬件、软件资源，尽量合理有效地组织多个用户共享多种资源。在 Linux 系统下，Shell 是介于用户和操作系统核心程序（Kernel）之间的接口，定义了各种变量和参数。用户在 Shell 中输入相关命令，然后调用相应的应用程序。

14.1　Linux Shell 概述

Shell 是系统的用户界面，为用户与内核提供交互操作的接口。各种操作系统都有自己的 Shell，比如 DOS/Windows 95/98 的 Shell 是 Command.com，Windows NT/2000/XP/2003/Vista 的 Shell 是 cmd.exe，Red Hat Enterprise Linux 5 的默认 Shell 是 Bash。

14.1.1　Linux Shell 简介

目前，使用最为广泛的 Shell 是 Bourne Shell（A&T Shell）、C Shell（Berkeley Shell）和 Korn Shell（Bourne Shell 的超集）及 Bash Shell 4 种，具体描述如表 14-1 所示。

表 14-1　使用较为广泛的 Shell

名　称	编 写 人	发 布 时 间	相 关 描 述
Bourne Shell	Stephen Bourne	1979 年	默认提示符为$，内置有 rc start、stop 和 shutdown 等系统管理命令，常用于系统管理，有简洁、快速的特点，是标准的 UNIX/Linux Shell
C Shell	Bill Joy	1979 年	加入了早期 Bourne Shell 所不具备的特性，如命令行记录、别名、命令行自动完成和作业控制等，常用交互模式
Korn Shell	David Korn	1986 年	是 Bourne Shell 的超集，增加了一些特性，比 C Shell 更为出色，支持现在 Shell 所支持的绝大多数功能，比如历史记录、别名、函数、正规表达式、和作业控制等
Bash Shell	Brian Fox	1987 年	发布后由 Case Western Reserve 大学 Chet Ramey 进一步发展。包括了早期 Bourne Shell 和 Korn Shell 的所有功能，并且加入了 C Shell 的某些功能

注意　Bourne Shell 和 Korn Shell 的兼容性非常好，在 Bourne Shell 下开发的程序，甚至不用进行修改就可以在 Korn Shell 下运行。

14.1.2　系统默认 Shell——Bash Shell

默认情况下，系统使用的 Shell 类型为 Bash，前边已经提到，Bash（Bourne Again Shell）是标准的 UNIX/Linux Shell。其包含了早期 Bourne Shell 和 Korn Shell 的全部功能，并且加入了 C Shell 的某些功能。

如果不清楚系统使用的到底是哪种 Shell，可以使用如下指令进行查询。

```
[root@localhost ~]#echo $SHELL      : #查看 SHELL 变量的值
/bin/bash                           : #说明使用的是 Bash Shell
[root@localhost ~]#
```

14.1.3 设置使用其他 Shell

虽然 Red Hat Enterprise Linux 5 默认采用的是 Bash Shell，但却并不一定适合每一个用户。例如，某些程序编写人员喜欢语法结构更像编程语言的 Tcsh Shell。这就涉及到 Shell 种类的选择，Shell 的选择一般分两种情况：暂时使用其他种类的 Shell，在系统重新启动后仍然使用默认的 Bash Shell；固定使用其他种类的 Shell，系统重新启动后不再回到默认的 Bash Shell。

1. 暂时使用其他 Shell

Red Hat Enterprise Linux 5 具体内置了哪些 Shell 类型，可以通过/etc/Shells 文件进行查看，具体指令如下。

```
[root@localhost ~]# cat /etc/shells
/bin/sh
/bin/bash
/sbin/nologin
/bin/tcsh
/bin/csh
/bin/ksh
/bin/zsh
[root@localhost ~]#
```

Red Hat Enterprise Linux 5 内置有多种 Shell，为用户提供了极大的方便。若需要暂时切换到其他 Shell，用户只需要在命令行下输入相应 Shell 的名称执行即可。

注意　除了使用“cat /etc/Shells”指令以外，还可以使用指令 chsh，以列表的形式显示 Linux 系统包含的 Shell 种类，具体语法如下。

```
[root@localhost ~]# chsh -list
```

---

2. 固定使用其他 Shell

如果系统默认的 Shell 不适合用户的日常应用，可以使用 chsh 指令永久更改系统的 Shell。例如，将用户 toplinux 的默认 Shell 类型改为 Tcsh Shell 的指令如下。

```
[root@localhost ~]# chsh -s /bin/tcsh toplinux
Changing shell for toplinux.
Shell changed.
```

注意　只有在管理员账户（root）下，才能使用 chsh 命令修改系统 Shell。

14.2　启动 Shell

Shell 就是 Linux 中的命令行，用户输入相应的命令，系统执行完毕后显示相应的结果，如图 14-1 所示。

```
[root@localhost ~]# ls
anaconda-ks.cfg  Desktop  hhwork  install.log  install.log.syslog
[root@localhost ~]# ls /etc
a2ps.cfg                init.d          quotatab
a2ps-site.cfg           initlog.conf    racoon
acpi                    inittab         rc
adjtime                 inputrc         rc0.d
alchemist               iproute2        rc1.d
aliases                 isdn            rc2.d
aliases.db              issue           rc3.d
alsa                    issue.net       rc4.d
alternatives            java            rc5.d
```

图 14-1　Red Hat Enterprise Linux5 中的命令行

14.2.1　终端启动 Shell

Red Hat Enterprise Linux 5 默认的操作方式是图形界面的 GNOME 或者 KDE，所以其启动 Shell 的方法类似于 Windows 操作系统下打开“命令指示符”窗口的方法。在 Red Hat Enterprise Linux 5 桌面环境下，依次单击“应用程序”→“附件”→“终端”（如图 14-2 所示）菜单命令，弹出命令行窗口，如图 14-3 所示。



图 14-2　打开终端

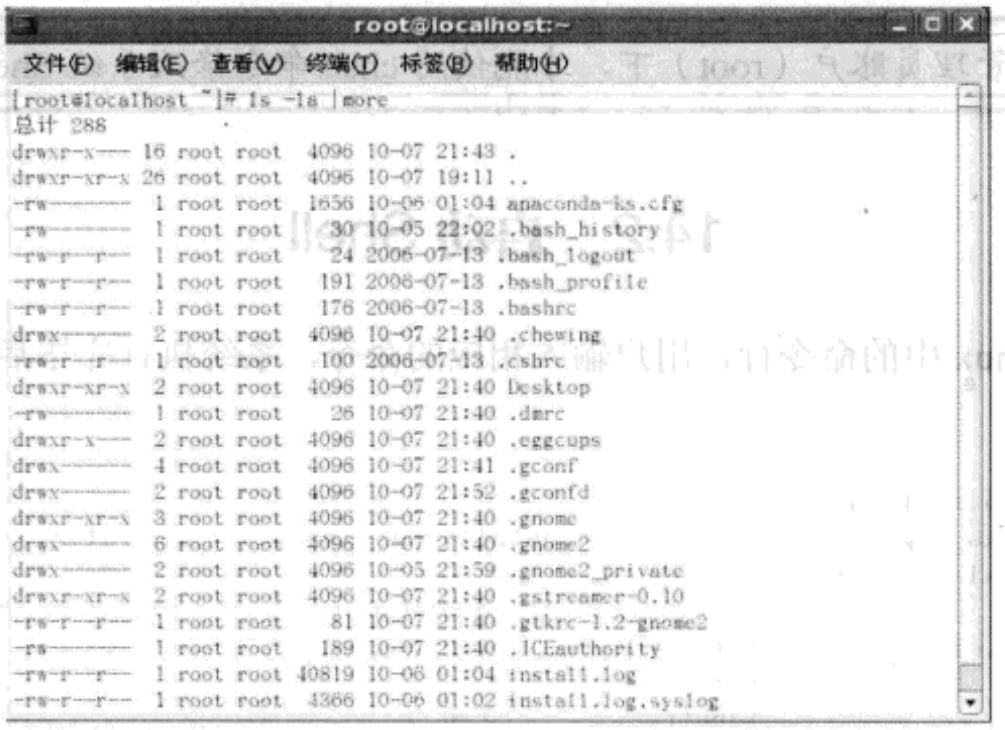


图 14-3　命令行窗口

14.2.2 直接启用 Shell

在系统启动的最后使用 Ctrl+Alt+F1~F6 组合键可以进入 Shell 提示符环境，重新登录后打开其中任意一个虚拟终端。启动命令行窗口的结果如图 14-4 所示。

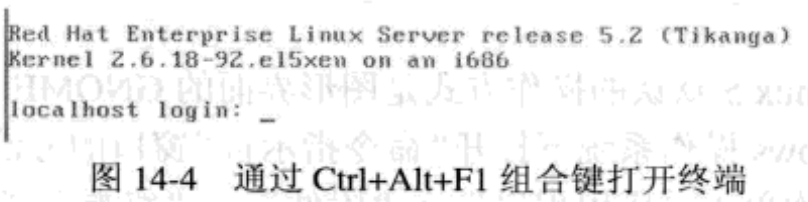


图 14-4　通过 Ctrl+Alt+F1 组合键打开终端

注意　在纯命令行模式下，使用 Alt+F1 ~ F6 组合键可以切换终端。

系统以不同的提示符来表示用户的类型。符号“#”表示系统管理员，符号“$”则表示普通用户。

```
[toplinux@localhost ~]$                ; #普通用户
[ltoplinux@localhost ~]$su root        ; #切换到 root 账号
Password:                              ; #输入账号口令
[root@localhost toplinux]#             ;
```

注意　从 root 切换到普通用户，不需要输入用户口令，而从普通用户切换到 root，则需要输入 root 的口令，并且输入结果不会显示在屏幕上。同时，当用户需要返回图形桌面环境时，只需使用 Ctrl+Alt+F7 的组合键，就可以回到刚才切换过来的桌面环境。

14.2.3 直接启动命令行窗口

如果用户在平时操作中只需要用到 Linux 的命令行界面，而不需要用到图形界面，那么最

好是开机时直接打开纯命令行窗口。为了实现开机时直接启动命令行窗口，可以修改/etc/inittab文件，具体操作步骤如下。

第 1 步，使用文本编辑器打开/etc/inittab 文件。

```
[root@localhost ~]# vi /etc/inittab
```

命令执行后，显示 etc/inittab 文件的内容如下。

```
# inittab       This file describes how the INIT process should set up
#               the system in a certain run-level.
#
# Author:       Miquel van Smoorenburg, <miquels@drinkel.nl.mugnet.org>
#               Modified for RHS Linux by Marc Ewing and Donnie Barnes
#

# Default runlevel. The runlevels used by RHS are:
#   0 - halt (Do NOT set initdefault to this)
#   1 - Single user mode
#   2 - Multiuser, without NFS (The same as 3, if you do not have networking)
#   3 - Full multiuser mode
#   4 - unused
#   5 - X11
#   6 - reboot (Do NOT set initdefault to this)
#
id:5:initdefault:

# System initialization.
si::sysinit:/etc/rc.d/rc.sysinit

10:0:wait:/etc/rc.d/rc 0
```

第 2 步，在/etc/inittab 文件中找到“id:5:initdefault:”，并将其修改为“id:3:initdefault:”，然后重新启动计算机，系统开机将直接进入命令行模式，而不会进入图形界面。

注意 若需要在开机时直接启动图形界面而非命令行模式，只需要在命令行模式下，执行“startx”指令打开图形模式，将“id:3:initdefault:”改回“id:5:initdefault:”即可。

14.3 Shell 指令详解

目前，虽然大多数 Linux 系统的管理工作都可以通过图形界面来完成，但是图形界面毕竟不是很完善，许多功能只有通过命令行才能实现。如果熟悉指令，命令行会让 Linux 系统的管理工作更加快捷、灵活。

14.3.1 目录及文件管理指令

在命令行模式下，用户首先接触到的就是目录和文件管理的指令。以下是常用的目录和文件管理指令及其相关参数介绍。

1. 文件复制——cp

cp 是 Linux 下用来复制文件的指令，其参数很多，但常用的只有-f 和-I，具体描述如表 14-2 所示。

表 14-2 cp 常用参数

参　数	具 体 描 述
-f	强制复制，即使是覆盖文件也不会提示
-I	交互复制，在复制每一个文件之前，都要提醒用户确认

注意 默认设置下，运行该命令时不会显示任何信息，只有在出现错误时才会给出相应提示。

实例 1 复制 index.htm 为 index.html，具体指令如下。

```
[root@localhost ~]#cp index.htm index.html
```

实例 2 将当前目录下的所有.dll 文件复制到/home/toplinux 下，具体指令如下。

```
[root@localhost ~]#cp *.dll/home/toplinux
```

2. 文件移动——mv

mv 指令用于将文件从一个位置移动到另外一个位置，其常用参数如表 14-3 所示。

表 14-3 mv 常用参数

参　数	具 体 描 述
-f	强制移动
-I	交互式移动

注意 在不同分区之间移动文件的时候，系统会先复制文件再进行移动，要多花费一定的时间。

实例 将/home/toplinux 下的所有文件移动到/tmp 目录下。

```
[root@localhost ~]#mv /home/toplinux/* /tmp
```

注意 如果是在同一目录下移动文件，与重命名的效果是一样的。

```
[root@localhost ~]#mv /home/toplinux/a.c   /home/toplinux/b.c
```

3. 显示文件和目录列表——ls

Linux 下 ls 的功能与 Windows 下的 dir 的功能类似，不同点在于 ls 的参数更多，如表 14-4 所示。

表 14-4 ls 参数

参　数	具 体 描 述
-a	显示所有文件和目录
-l	显示详细信息，包括权限、拥有者、群组、大小、创建日期以及是否链接文件等
-r	递归显示指定目录及其子目录中的所有内容
-h	更清晰地显示文件大小，如 1 KB、256 MB、4 GB
-i	显示索引信息（inode）

实例 显示当前目录下的文件（包括索引信息）的详细信息，指令如下。

```
[root@localhost ~]#ls -lih
总计 84K
557066 -rw------- 1 root root 1.6K 04-28 19:14 anaconda-ks.cfg
557095 drwxr-xr-x 2 root root 4.0K 05-07 00:17 Desktop
557246 drwxr-xr-x 2 root root 4.0K 06-10 14:04 hhworks
558025 -rw-r--r-- 1 root root  13K 05-08 17:36 logsina.log
557259 -rw-r--r-- 1 root root  127 07-09 23:13 remote_1
557150 drwxr-xr-x 3 root root 4.0K 05-07 00:15 server rpm
558062 drwxr-xr-x 3 root root 4.0K 06-10 14:20 test
557136 -rw------- 1 root root   22 05-03 18:34 testfile~
557239 -rw-r--r-- 1 root root  127 05-07 10:32 testroot.ls
557267 drwxr-xr-x 3 root root 4.0K 05-08 17:33 www.22bz.com
557234 -rw------- 1 root root    0 05-08 20:08 新文件~
```

注意 若要查看 ls 指令的详细帮助信息，执行“man ls”指令即可，也可以执行“man ls | col -b | lpr”打印帮助信息。

4. 删除文件——rm

rm 指令的常用参数说明如表 14-5 所示。

表 14-5 rm 指令常用参数

参 数	具 体 描 述
-i	交互式删除，次删除前需要用户确认
-f	强制删除，本指令应当慎用，Linux 下文件被删除后很难恢复
-v	冗余模式，显示更多提示信息
-r	递归模式，不仅删除指定内容，还删除其子目录下的内容

实例 删除 FileName，指令如下。

```
[root@localhost ~]#rm FileName
```

5. 创建目录——mkdir

Linux 下的 mkdir 指令用于创建目录，语法格式如下。

```
[root@localhost ~]#mkdir [-p] DirName
```

实例 1 在当前目录下，创建一个 hhwork 目录，指令如下。

```
[root@localhost ~]#mkdir hhwork
```

实例 2 在 hhwork 目录下创建目录 hhwork1，若不存在目录 hhwork，则先创建，具体指令如下。

```
[root@localhost ~]#mkdir -p hhwork/hhwork1
```

注意 需要注意的是，在实例 2 中如果不添加-p 参数，hhwork 不存在时将提示错误信息且创建失败。

6. 删除目录——rmdir 和 rm

Linux 下删除目录的指令是 rmdir 和 rm，常用参数如表 14-6 所示。

实例 1 删除/hhwork 目录的指令如下。

```
[root@localhost ~]#rmdir /hhwork
[root@localhost ~]#rm -rf /hhwork
```

表 14-6 rmdir 和 rm 常用参数

参　数	具 体 描 述
-v	冗余模式，显示更多的提示信息
-p	不但删除指定目录，还删除其上级目录

注意 rm 指令加上参数还具备删除附录的功能，删除目录的时候会删除该目录下的所有内容。而 rmdir 只有在被删除目录为空时才执行删除操作。

实例 2 删除/hhwork/hhwork1 及其上级目录/hhwork，具体指令如下。

```
[root@localhost ~]#rmdir -p /hhwork/hhwork1
```

7. 查看当前目录——pwd

pwd 用于查看系统当前目录的详细路径。需要注意的是，Windows 下显示当前目录的指令是 cd，而 Linux 下执行 cd 指令将会直接切换到用户的家目录。

pwd 的命令格式如下。

```
[root@localhost ~]#pwd
```

实例 在目录/system.d 下，显示其详细路径的指令如下。

```
[root@localhost system.d]#pwd
/etc/dbus-1/system.d
[root@localhost system.d]#
```

命令执行结果表示/system.d 位于目录/etc/dbus-1 下。

8. 切换目录——cd

与在 Windows 下一样，在 Linux 下切换目录的指令也是 cd，不过 Linux 下的 cd 指令的功能更为强大，具体描述如表 14-7 所示。

表 14-7 cd 指令相关介绍

命　令	具 体 描 述
cd	切换到用户家目录
cd -	返回到上一个目录
cd ~	同样是返回到家目录
cd /	切换到根目录
cd /root	切换到 root 用户的家目录
cd ..	切换到上级目录
cd /home/toplinux	切换到 toplinux 用户的家目录

需要注意的是，使用 cd 指令让用户在不同的目录间切换，前提是该用户必须拥有足够的权限进入目的目录。

9. 查找文件或目录

假如用户知道某一目录或者文件存在，但是又不知道其具体位置，这个时候就要用到相应的查找、搜索工具了。Linux 命令行具有查找、搜索功能的指令包括 find、locate、which、where 和 whatist 等。

（1） find

find 指令的格式如下。

```
[root@localhost ~]#find StartDir Options
```

常用参数如表 14-8 所示。

表 14-8 find 常用参数

参 数	具 体 描 述
-atime n	查找至少在 n*24 h 以内没有访问过的文件
-ctime n	查找至少在 n*24 h 以内没有修改过的文件
-amin n	查找 n min 之前访问过的文件
-ctime n	查找 n min 之前修改过的文件
-print	列表显示找到的文件
-name fname	查找文件名称为 fname 的文件，可以使用正则表达式
-size n[bck]	查找文件长度为 n 块/字符/千字节（每块等于 512 B）的文件

注意 find 指令如果未指定路径，则会从当前目录搜遍所有子目录。

实例 1 搜索/hhwork 下至少 15 d 没有被访问过的文件，具体指令如下。

```
[root@localhost ~]#find /hhwork -atime 15 -print
```

实例 2 搜索/home/toplinux 下所有后缀为.dll 的文件并且删除，具体指令如下。

```
[root@localhost ~]#find /home/hhwork -name "*.dll" -exec rm
```

从实例 2 中可以看到 find 指令可以和“exec”指令结合使用。需要注意的是，文件名会被传递给使用字符串“{}”的命令，并且命令的结尾以“\;”结束。例如，要查找/usr/src/redhat 目录下的所有文件，找到包含有“tcp”的文件并打印出来。

```
[root@localhost ~]#find /usr/src/redhat -name "*.*" -exec grep 'tcp'{}\;
-print
```

（2） locate

locate 与 find 指令的不同之处在于 locate 指令是通过在一个数据库中搜索文件或者目录，速度比 find 指令要快许多，格式如下。

```
[root@localhost ~]#locate Something
```

注意 如果没有更新数据库，使用 locate 指令时可能会提示搜索结果错误。

（3） which、where 和 whatis

which、where 和 whatis 的具体描述如表 14-9 所示。

表 14-9 which、where 及 whatis 命令的具体描述

名　称	语 法 格 式	描　述
which	which command	主要用于搜索二进制文件、可执行文件或者 Shell 命令行的位置
whereis	whereis command	主要用于查询二进制文件的位置、源代码的位置和 man 帮助文件位置。例如，“whereis find”会返回如下信息： /usr/bin/find /usr/share.man.manlp/find.lp.gz ……
whatis	whatis command	主要用于从 man 帮助页面返回软件信息

- which

which 的语法格式为如下。

```
[root@localhost ~]#which command
```

主要用来搜索二进制文件、可执行文件或者 Shell 命令行的位置。

- whereis

whereis 的语法格式为如下。

```
[root@localhost ~]#whereis command
```

主要用来查询二进制文件的位置、源代码的位置和 man 帮助文件位置。比如“whereis find”会返回如下信息。

```
/usr/bin/find /usr/share.man.manlp/find.lp.gz
/usr/shre/man/man1/find.1.gz
```

- whatis

whatis 的语法格式为如下。

```
[root@localhost ~]#whatis command
```

主要是用于从 man 帮助页面返回软件信息。

14.3.2 信息显示指令

功能强大的信息显示指令是 Linux 的一大特色，非常适合命令行模式下文本文件的管理，例如文件的头尾查看、分页查看、二进制文件查看等。下面要介绍的就是 Linux 的几个重要的信息显示指令。

1. 查看文件头——head

head 指令的语法格式如下。

```
[root@localhost ~]#head <FileName>
```

注意 默认情况下，head 指令只显示文件的前 10 行。若想显示更多内容，例如显示文件的前 30 行，可以使用如下格式。

```
[root@localhost ~]#head -30 <FileName>
```

2. 查看文件尾——tail

tail 指令最重要的用处就在于显示记录系统信息的文件。由于最新的系统记录总是位于文件的最后，所以 tail 指令经常与参数-f 结合使用，该参数可以让 tail 指令实时更新显示文件的最后的 10 行。

实例 显示最新系统信息的指令如下。

```
[root@localhost ~]#tail -f /var/log/messages
Jul 10 21:57:28 localhost dhclient: DHCPACK from 192.168.1.1
Jul 10 21:57:28 localhost dhclient: bound to 192.168.1.104 -- renewal in 2988
seconds.
Jul 10 22:47:16 localhost dhclient: DHCPREQUEST on eth0 to 192.168.1.1 port 67
Jul 10 22:47:16 localhost dhclient: DHCPACK from 192.168.1.1
Jul 10 22:47:16 localhost dhclient: bound to 192.168.1.104 -- renewal in 3248
seconds.
Jul 10 23:33:02 localhost scim-bridge: The lockfile is destroied
Jul 10 23:33:02 localhost scim-bridge: Cleanup, done. Exitting...
Jul 10 23:41:24 localhost dhclient: DHCPREQUEST on eth0 to 192.168.1.1 port
67
Jul 10 23:41:24 localhost dhclient: DHCPACK from 192.168.1.1
Jul 10 23:41:24 localhost dhclient: bound to 192.168.1.104 -- renewal in 3344
seconds.
```

3. 分页查看——more

more 指令是最为简易的文本查看指令，其语法格式如下。

```
[root@localhost ~]#more <FileName>
```

在文件显示的时候，按下 Space 键可以向前翻页，按下 Enter 键可以向前翻行，要退出则按下 Q 键。

```
[root@localhost ~]# more anaconda-ks.cfg
# Kickstart file automatically generated by anaconda.
lang zh_CN.UTF-8
keyboard us-acentos
xconfig --startxonboot
network --device eth0 --bootproto dhcp
rootpw --iscrypted $1$95dYZk.0$1VtLiHiiYQ1IedRWAhhXO/
firewall --enabled --port=22:tcp
authconfig --enableshadow --enablemd5
selinux --enforcing
timezone --utc Asia/Chongqing
```

```
bootloader --location=mbr --driveorder=sda --append="rhgb quiet"
......
--More--(48%)
```

4. 分页查看——less

less 指令比 more 指令多出的功能在于它可以直接向后翻页和翻行，具体如下。

“Space”键或“Page Down”键：向前翻页。

“B”键或“Page Up”键：向后翻页。

“↑”键：向前翻行。

“↓”键：向后翻行。

“Q”键：退出。

less 指令的语法格式如下。

```
[root@localhost ~]#less <FileName>
```

实例 查看文件 anaconda-ks.cfg 的内容，执行如下指令。

```
[root@localhost ~]# less anaconda-ks.cfg
```

文件内容如下所示。

```
# Kickstart file automatically generated by anaconda.
lang zh_CN.UTF-8
keyboard us-acentos
xconfig --startxonboot
network --device eth0 --bootproto dhcp
rootpw --iscrypted $1$95dYZk.0$1VtLiHiiYQ1IedRWAhhXO/
firewall --enabled --port=22:tcp
authconfig --enableshadow --enablemd5
selinux --enforcing
timezone --utc Asia/Chongqing
bootloader --location=mbr --driveorder=sda --append="rhgb quiet"
......
anaconda-ks.cfg
```

注意 使用 less 指令还可以查询文本文件中包含的字符串。按下“/”键，然后输入需要查询的字符串执行即可。

5. 字符输入——cat

cat 指令是一个集查看和输入字符两种功能为一体的指令，单独执行 cat 指令，系统会接受用户通过键盘输入的字符，输入完毕后按下 Enter 键，即可在屏幕上实时显示。

注意 使用组合键 Ctrl+D 可以退出字符输入状态。

实例 使用 cat 指令查看系统信息文件的指令如下。

```
[root@localhost ~]#cat /var/log/messages
```

注意 当文件内容过多时，cat 指令显示结果会快速翻屏完成，这时可以使用 less 指令配合，例如，cat 指令与 less 指令配合查看系统信息文件的指令如下。

```
[root@localhost ~]#cat /var/log/messages | less
```

6. 从文件中搜索字符串——grep

grep 指令的语法格式如下。

```
[root@localhost ~]#grep String <FileName>
```

实例 从/var/log/messages 文件中查找错误信息的指令如下。

```
[root@localhost ~]#grep "error" /var/log/messages
```

注意 在 Linux 下，可以一次执行多条指令，但不同指令间需要用 ";" 隔开。

7. 二进制文件查看——od

若需要查看系统中的二进制文件，就需要用到 od 指令了，其语法格式如下。

```
od [-abcdfhilovx] [-A <字码基数>][-j <字符数目>][-N <字符数目>][-s <字符串字符数>] [-t <输出格式>][-w <每列字符数>][--help][--version][文件...]
```

od 指令的常用参数如表 14-10 所示。

表 14-10 od 指令常用参数

参　数	描　述
-A<字码基数>	选择要以何种基数计算字码
-j<字符数目>或--skip-bytes=<字符数目>	略过设置的字符数目
-N<字符数目>或--read-bytes=<字符数目>	到设置的字符数目为止
-s<字符串字符数>或--strings=<字符串字符数>	只显示符合指定的字符数目的字符串
-t<输出格式>或--format=<输出格式>	设置输出格式
-v 或--output-duplicates	输出时不省略重复的数据
-w<每列字符数>或--width=<每列字符数>	设置每列的最大字符数
--help	在线帮助
--version	显示版本信息

14.3.3 网络基本配置指令

网络配置是服务器的日常管理工作之一，也是管理员必须掌握的基本技能之一。Red Hat Enterprise Linux 5 中基本网络配置指令包括配置各种网络参数的 ifconfig 指令、配置/修改路由表的 route 指令、启动和关闭某个网络适配器的 ifup 和 ifdown 指令。

1. ifconfig

ifconfig 是 Linux 下最主要的网络配置指令，主要用于配置 IP 地址、网段、子网掩码等，还可以用来启动或关闭网络适配器。其语法格式如下。

```
[root@localhost ~]#ifconfig interface [options]
```

ifconfig 指令的常用参数如表 14-11 所示。

表 14-11 ifconfig 指令常用参数

参 数	具 体 描 述
network	配置网段
broadcast	广播网段
netmask	子网掩码
up\down	启动或关闭网络接口

实例 查看 eth0 网卡的网络配置信息，指令执行如下。

```
[root@localhost~] # ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:0C:29:33:75:72
          inet addr:192.168.1.104  Bcast:255.255.255.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1026 errors:0 dropped:0 overruns:0 frame:0
          TX packets:51 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:277221 (270.7 KiB)  TX bytes:9201 (8.9 KiB)
 [root@localhost ~]#
```

执行结果中，各项信息的详细描述如表 14-12 所示。

表 14-12 信息描述

信 息	信 息 说 明
eth0	网卡代号，一般指的是第一个 Ethernet 网卡
HWaddr	网卡的 MAC 地址，也就是硬件地址
Inet addr	网卡的 IPv4 地址
Inet6 addr	网卡的 IPv6 地址
Bcast	广播地址
Mask	子网掩码
MTU	Maximum Transmission Unit 的简称，指相应接口一次所能传输的最大封包 (以 B 为单位)
RX	网络由启动到目前为止的接收情形
TX	网络由启动到目前为止的传送情形
collisions	网络讯号碰撞的情况说明
txqueuelen	传输缓冲区长度大小

注意 相应的接口一次所能传输的封包 MTU 值并不是越大越好，也不是越小越好。设置错误时，可能会让网络无法连接上某些网站。

2. route

route 指令主要用于显示或修改路由表，语法格式如下。

```
[root@localhost ~]#route [-nNvee] [-FC] [<AF>]          ; #查看内核路由表
[root@localhost ~]#route [-v] [-FC] {add|del|flush}     ; #修改路由表
```

route 指令的常用参数如表 14-13 所示。

表 14-13 route 指令常用参数

参　数	描　述
-n	列出的信息以 IP 地址的形式显示
-ee	列出较长列的信息
add	增加路由信息
del	删除一个路由信息
-net	增加一个“网段”的路由，例如 C 类网段
-host	增加到某个 IP 主机的路由
netmask	子网掩码
gw	路由网关，gw 是以 IP 地址的形式来设置的
dev	路由网关，dev 是以接口（internet）形式来设置的

实例 查看内核路由表，执行以下指令。

```
[root@localhost ~]#route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.1.0     0.0.0.0         255.255.255.0   U     0      0        0 eth0
192.168.122.0   0.0.0.0         255.255.255.0   U     0      0        0 virbr0
0.0.0.0         192.168.1.1     0.0.0.0         UG    0      0        0 eth0
[root@localhost ~]#
```

3. ifup、ifdown

Linux 下使用 ifup 和 ifdown 指令来启用或者暂停某个网络接口，其命令格式如下。

启动网络接口。

```
[root@localhost ~]#ifup[interface]
```

例如，启动网卡 eth0。

```
[root@localhost ~]#ifup eth0
```

禁用网络接口。

```
[root@localhost ~]#ifdown [interface]
```

例如，禁用网卡 eth0。

```
[root@localhost ~]#ifdown eth0
```

4. 综合实例

网卡 IP 地址设置为 192.168.1.5、网关设置为 192.168.1.1 并启动网卡。

具体操作步骤如下。

第 1 步，执行如下命令设置网卡的 IP 地址。

```
[root@localhost ~]#ifconfig eth0 192.168.1.5 netmask 255.255.255.0 broadcast
192.168.1.255
```

第 2 步，设置网关，具体指令如下。

```
[root@local root]# route add -net 192.168.1.0 netmask 255.255.255.0 dev eth0
[root@local root]#route add default gw 192.168.1.1
```

第 3 步，在/etc/resolv.conf 文件中设置相应的 DNS 主机名和 IP 地址使系统能够访问 Internet。

```
; generated by /sbin/dhclient-script
nameserver 192.168.0.1
search localdomain
```

第 4 步，执行指令启动网卡，使设置生效。

```
[root@localhost ~]#ifup eth0
```

14.3.4 关机和注销指令

Linux 下有很多与用户注销、关机相关的指令，而且这些指令各具特色，为用户的操作提供了极大的方便。

1. 注销——logout

logout 指令不需要任何参数，格式如下，结果如图 14-5 所示。

```
[root@localhost ~]#logout
```

```
Red Hat Enterprise Linux Server release 5 (Tikanga)
Kernel 2.6.18-8.el5xen on an i686

localhost login: _
```

图 14-5 注销之后就返回到了登录界面

注意 logout 指令运行完毕后，系统将返回到登录界面。

2. 关机——shutdown

shutdown 指令是 Linux 下最重要的关机指令，拥有众多的参数。shutdown 指令与对应的参数结合使用可以实现立即关机、某个时间段后关机、关闭电源、重新启动、某个时刻重新启动等功能。shutdown 指令的格式如下。

```
[root@localhost ~]# shutdown [-t sec] [-arkhncfFHP] time [warning message]
```

shutdown 指令的常用参数如表 14-14 所示。

表 14-14 shutdown 指令常用参数

参　数	描　述
-t sec	过 sec 秒再关机
-a	只有用户名位于/etc/shuwdown.allow 时才允许关机
-k	模拟关机（只发送警告信息）
-h	关闭电源
-f	重启时跳过 fsck 检测
-F	重启时强制进行 fsck 检测
time	何时关机
-c	终止上次进行的重启或者关机操作
Warning-message	在用户端显示警告信息

实例 1 立即关机并关闭电源的指令如下。

```
[root@localhost ~]#shutdown -h now
```

实例 2 10 min 后关机并关闭电源的指令如下。

```
[root@localhost ~]#shutdown -h +10
```

实例 3 在 11：20 重新启动计算机，指令如下。

```
[root@localhost ~]#shutdown -r 11:20
```

3. 重启、关机——reboot、halt

如果只是普通的关机和重新启动，可以使用 reboot 和 halt 指令，这两个指令不需要任何参数，直接执行即可。

注意 halt 指令实际上就是调用“shutdown -h”完成关机操作。

4. 重启、关机——init

init 指令可以完成很多操作，其指令格式如下。

```
init 0          关闭计算机
```

init 指令的其他功能如表 14-15 所示。

表 14-15 init 指令的其他功能

名 称	功 能 描 述
1	单用户模式
2	多用户模式，但不支持 NFS
3	全功能多用户模式
5	图形界面模式
6	重启

14.4 配置 Shell 环境

Linux 作为一个多用户操作系统，每一个登录用户都有一个专用的运行环境。这个运行环境由一组环境变量进行定义，用户可以根据需要对环境变量进行修改或定制。/etc/profile 作为系统的初始启动文件，也就是 Shell 环境变量配置文件。

14.4.1 Shell 环境变量

在 Linux 下，可以通过变量名来访问相应的环境变量，通过 export 来设置环境变量。

1. 显示环境变量

引用环境变量通过符号“$”加变量名来实现，比如查看用户家目录，就要用到“$HOME”环境变量，可以使用 echo 指令来显示该变量值。

例如，显示用户家目录，指令执行如下。

```
[root@localhost ~]#echo $HOME
/root
```

执行指令得出当前用户 root 的家目录为/root。

注意　环境变量需要区分大、小写，例如上面用来代表用户家目录的$HOME，字母全部要大写。

2. 显示所有环境变量

显示所有环境变量使用 env 指令，命令执行如下所示。

```
[root@localhost ~]#env
SSH_AGENT_PID=6255
HOSTNAME=localhost.localdomain
DESKTOP_STARTUP_ID=
TERM=xterm
SHELL=/bin/bash
HISTSIZE=1000
GTK_RC_FILES=/etc/gtk/gtkrc:/root/.gtkrc-1.2-gnome2
WINDOWID=54526033
USER=root
......
```

3. 显示所有本地定义的环境变量

使用 set 指令可以显示所有本地定义的 Shell 变量，格式如下。

```
[root@localhost ~]#set
```

对于 Red Hat Enterprise Linux 5 来说，默认的环境变量可以参考表 14-16。

表 14-16　默认的 Shell 环境变量

变 量 名	描　　述
SHELL	包括 Shell 命令的完整路径名，通常是/bin/bash
BASH_VERSION	表示当前 Bash 命令版本号的数字
EUID	当前用户的有效 ID 号，是在 Shell 启动时根据/etc/passwd 文件中的用户条目分配的
FCEDIT	如果设置了该变量，将使用 fc 命令编辑 history 命令所使用的文本编辑器。如果未设置该变量，则使用 vi 命令
HISTFILE	历史命令文件的位置，通常为$HOME/.bash_history
HISTFILESIZE	可储存的历史条目数，达到该数后，会抛弃最早的命令，默认值为 1 000
HISTCMD	返回历史列表中当前命令的编号
HOME	用户家目录
HOSTTYPE	该值描述运行 Linux 系统的计算机架构。对于兼容 intel 的 PC 机，该值是 i386、i486、i586、i686 或 i386-linux。对于 AMD 64 位计算机，该值是 x86_64
OLDPWD	更改到当前工作目录之前的工作目录
OSTYPE	标识当前操作系统的名字，对于 Fedora Core Linux，OSTYPE 值是 linux 或 linux-gnu，这取决于所用的 Shell 类型（Bash 可以运行在其他操作系统上）
PPID	启动当前 Shell 的命令进程 ID
PROMPT_COMMAND	设置为在每次显示 Shell 提示符前运行的命令名。例如设置 PROMPT_COMMAND=date，将在提示符出现前列出当前日期/时间
PS1	设置 Shell 提示符，可以将很多条目加入到提示符中（日期、时间、用户名、主机名等）
PWD	当前目录，每次改变目录时，该值都会变化
RANDOM	产生一个 0～99 999 之间的数值
SHLVL	与当前 Shell 会话有关的 Shell 级数。登录 Shell 时，SHLVL 是 1。每次启动一个新的 Bash 命令时，该数依次递增
TMOUT	可以设置为一个数值，表示 Shell 不再接收输入的空闲时间，以 s 为单位。达到这个秒数后，Shell 会退出。这是一个安全功能，可使闲置的 Shell 不会被未授权的人访问（必须在登录 Shell 中设置该值，这样 Shell 才会实际注销用户）
UID	分配给用户名的用户 ID 号，用户 ID 号存储在/etc/password 文件中

14.4.2 环境变量设置

对环境变量的设置主要体现在环境变量的赋值、取消、设置只读属性 3 方面，下面具体介绍这 3 种设置。

1. 为环境变量赋值

为环境变量赋值的指令格式如下。

```
[root@localhost ~]# variable=value
```

在上面的格式中，variable 表示变量名称，value 则表示为变量赋予的值。

注意 变量赋值以后，最好马上使用 export 指令使环境变量生效。

```
[root@localhost ~]# TMOUT =300
[root@localhost ~]#export TMOUT
```

---

2. 取消环境变量

使用 unset 指令可以取消环境变量。例如，取消环境变量 PS1 的指令如下。

```
[root@localhost ~]#unset $ PS1
```

3. 设置只读环境变量

使用 readonly 指令可以将某个环境变量设置成为只读属性，这样可以避免环境变量被其他用户修改和清除。例如，将环境变量 PS1 设置成为只读属性的指令如下。

```
[root@localhost ~]#export PS1="This is a PS1 env…"
[root@localhost ~]#readonly TEST
```

注意 环境变量位于/etc/profile 文件，如果需要增加新的环境变量可以添加以下语句。

```
export path=$path:/path1:/path2:/pahtN
```

---

4. 其他设置环境变量的方法

除了上面介绍的 3 种基本设置环境变量的操作外，这里还向读者推荐 3 种简单且行之有效的操作方法。

方法 1 修改/etc/profile 文件

第 1 步，使用 vim（文本编辑器）打开/etc/profile 文件。

第 2 步，在文件的末尾加入环境变量的定义，然后重新启动系统使修改生效。

```
JAVA_HOME=/usr/share/jdk
PATH=$JAVA_HOME/bin:$PATH
CLASSPATH=.:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar
export JAVA_HOME
export PATH
export CLASSPATH
```

注意 环境变量下的多个路径之间应用“:”号隔开。

方法 2 在当前 Shell 下设置变量

这种方法只对当前 Shell 有效，如果用户注销或重新启动系统则设置失效，但操作非常简单，非常适合用户的临时需求。具体指令如下。

只需要在 Shell 下执行以下 3 条指令即可。

```
[root@localhost ~]#export JAVA_HOME=/usr/share/jdk
[root@localhost ~]#export PATH=$JAVA_HOME/bin:$PATH
[root@localhost ~]#export CLASSPATH=.:$JAVA_HOME/lib/dt.jar:$JAVA_
HOME/lib/tools.jar
```

方法 3 修改~/.bashrc 文件

这种方法只对当前用户有效，但重新启动系统后设置仍然有效。操作也非常简单，与方法 1 类似，首先使用 vim 打开~/.bashrc 文件，然后在文件的末尾添加如下所示的代码，最后重新登录系统使设置生效即可。

```
[root@localhost ~]#set JAVA_HOME=/usr/share/jdk
[root@localhost ~]#export JAVA_HOME
[root@localhost ~]#set PATH=$JAVA_HOME/bin:$PATH
[root@localhost ~]#export PATH
[root@localhost ~]#set CLASSPATH=.:$JAVA_HOME/lib/dt.jar:$JAVA_
HOME/lib/tools.jar
[root@localhost ~]#export CLASSPATH
```

14.4.3 Shell 环境配置文件

Shell 环境配置文件，可以通过运行以下指令来查看。

```
[root@localhost ~]#ls -lih .bash*
557139 -rw------- 1 root root 12K 07-11 00:56 .bash_history
557060 -rw-r--r-- 1 root root  24 2006-07-13 .bash_logout
557061 -rw-r--r-- 1 root root 191 2006-07-13 .bash_profile
557062 -rw-r--r-- 1 root root 176 2006-07-13 .bashrc
```

注意 编辑 Shell 环境配置文件不需要管理员（root）权限，用户可以根据自己的习惯和喜好进行编辑，当系统设置与用户的设置发生冲突时，将采用用户的设置。

Shell 环境配置文件的具体描述如表 14-17 所示。

表 14-17 Shell 环境配置文件具体描述

文 件 名	具 体 描 述
.bash_history	记录用户以前输入的命令
.bash_logout	表示用户退出 Shell 时要执行的命令，默认的指令为清屏
.bash_profile	表示用户登录 Shell 时，要执行的命令。该文件为某些内容设置环境变量，如邮箱的位置或历史文件的大小
.bashrc	表示 iyonghu 每次打开新的 Shell 时，要执行的命令

注意 一般情况下，定义好的环境变量（如 PATH、HOME）放在.bash_profile 文件中，aliases（别名）和函数之类的变量放在.bashrc 文件中，为了使用方便也可以将所有配置都放在.bashrc 文件中。

14.5 使用 vim 文本编辑器

文本编辑器，顾名思义就是用来编辑文本文件的一种程序软件。像用户比较熟悉的 Windows 下的记事本和写字板，都是非常不错的文本编辑工具。在 Linux 下，最常用的编辑器就是 vi，而在 Red Hat Enterprise Linux 5 中包含了其增强版 vim。

14.5.1 vim 编辑器简介

vi 是 Linux 和 UNIX 上最基本的文本编辑器，其增强版是 vim，几乎存在于每一个版本的 Linux 当中。

1. vim 编辑器的基本模式

vim 编辑器包括指令行（command mode）、插入（Insert mode）和底行（last line mode）3 种模式，如表 14-18 所示。

表 14-18 vim 编辑器的 3 种模式

指 令 模 式	功 能 描 述
指令行模式	用于对文件进行编辑操作或者切换到其他模式。打开 vim 时，首先进入的是该模式，此时输入的任何字符都被当作指令
插入模式	用于从当前光标所在位置插入正文，是用户使用最多的工作模式，按 Esc 键可以从插入模式切换到指令行模式
底行模式	用于向 vi 输入特定的扩充指令，如寻找字符串、列出行号等。在指令模式中键入“：”可以切换至指令行模式。在指令行模式中，用户输入的单行指令会显示在屏幕的最底行，按下 Enter 键后确认执行

注意 在使用时一般把 vi 简化成两个模式，就是将底行模式也算入指令行模式。

2. vim 的基本操作

由于 vim 编辑器有指令行、插入、和底行 3 种模式，所以操作要稍微复杂一点。

（1） 进入 vim

进入 vim 的方法很简单，只需要在系统提示符下输入 vi 及文件名称即可进入 vim 的全屏幕编辑界面，进入 vim 后，系统会载入所要编辑的文件或是打开一个新文件，如图 14-6 所示。

```
[root@localhost ~]#vim <文件名称>
```

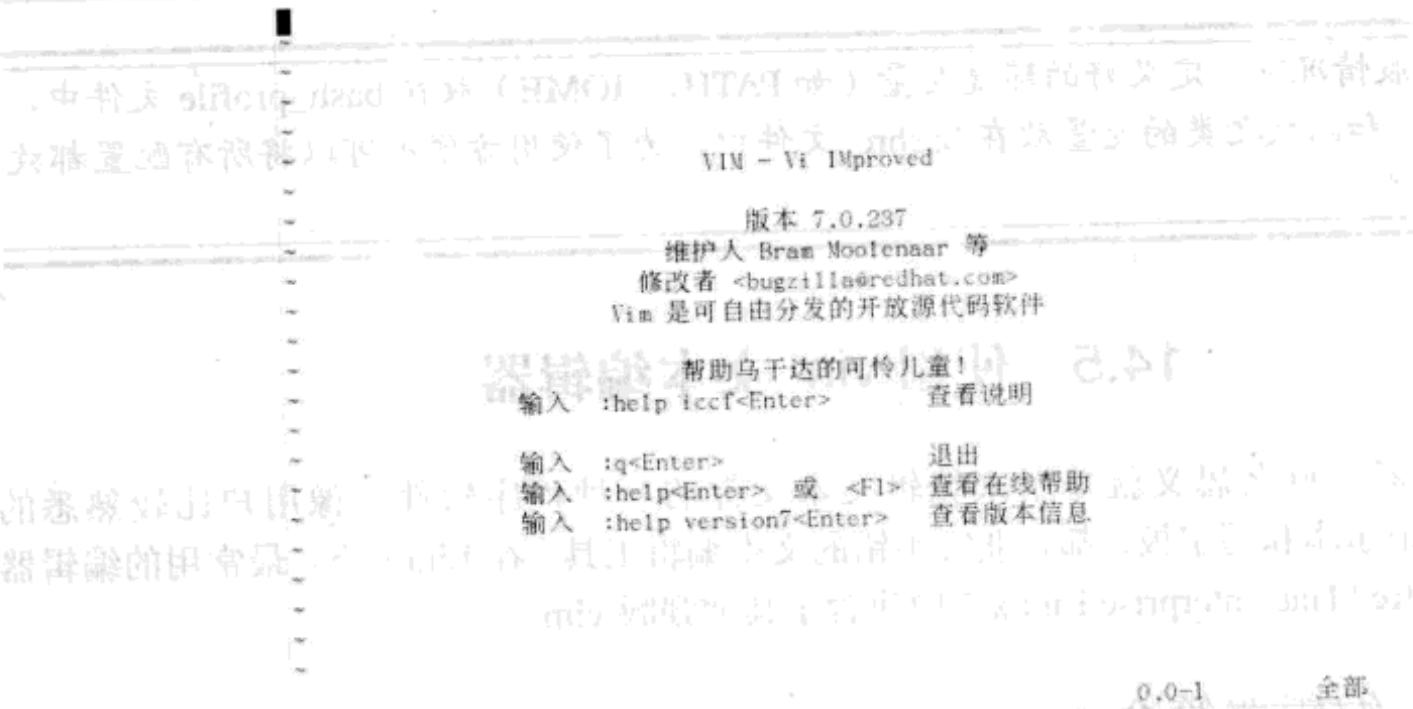


图 14-6 进入 vim 的全屏幕编辑界面

注意 vim 默认的模式是指令行模式，此时输入的任何字符都将作为命令执行。例如，在 vim 中 “H” 命令表示移动光标到荧屏的第一行，如果在指令行模式下按下 H 键，屏幕上显示的不是字符 “H”，而是将光标移动到荧屏的第一行。

（2） 切换至插入模式编辑文件

若用户想要编辑文件，需要切换到插入模式（insert mode）才能够输入字符。这时在指令行模式下按 i 键即可进入插入模式。

14.5.2 常用 vim 指令

vim 的指令很多，但是用户在实际操作中结合起来记忆还是很容易熟悉和掌握的。由于 vim 的应用主要体现在编辑方面，这里就向读者介绍一些编辑过程中常用的命令。

1. 插入模式中的基本操作

插入模式中最为常见的命令有 3 个，分别是新增（append）、插入（insert）、开始（open），如表 14-19 所示。

表 14-19　插入模式中的命令

命 令 名 称	功 能 描 述
a	从当前光标所在位置之后插入正文
A	从当前光标所在列最后面的地方插入正文
i	从当前光标所在位置之前插入正文
I	从当前光标所在列的第一个非空白字符之前插入正文
o	在当前光标所在列下方新增一列并进入插入模式
O	在当前光标所在列上方新增一列并进入插入模式

除了上面的基本指令，用户还可以使用如下指令将其他文件插入到当前光标所在位置。

```
:r file
```

注意　使用指令“:r [Shell 命令]”可以将 Shell 命令执行后的结果插入光标所在的位置。

2. 移动光标

光标的移动是编辑文档过程中最基本和最重要的操作，只有熟练地掌握光标的操作，才能使文档的编辑得心应手。为此，vim 除了基本的光标移动指令（h、j、k、l）和方向键外，还额外提供了许多移动光标的指令，如表 14-20 所示。

表 14-20　光标移动指令

指　令	功 能 描 述
H	移动到荧屏的第一行
M	移动到荧屏的中间行
L	移动到荧屏的最后一行
G	移动到文档的最后一行
B	移动到下一个单词的第一个字母
W	移动到上一个单词的第一个字母
E	移动到下一个单词的最后一个字母
0	移动到当前行的第一个字符，其功能等同于“Home”键
$	移动到当前行的最后一个字符，其功能等同于“End”键
^	移动到光标所在行的第一个非空白字符
n-	减号移动到上一列的第一个非空白字符，前面加上数字可以指定往上移动 n 行
n+	加号移动到下一列的第一个非空白字符，前面加上数字可以指定往下移动 n 行
nG	移动到第 n 行
fx	往右移动到 x 字符上
Fx	往左移动到 x 字符上
tx	往右移动到 x 字符前
Tx	往左移动到 x 字符前
/string	往右移动到有 string 的地方
?string	往左移动到有 string 的地方
n	配合上述/和?指令使用，重复一次

（续表）

指　令	功 能 描 述
N	配合上述/和?指令使用，反方向重复一次
n(	左括号移动到句的最前面，前面加上数字可以指定往前移动 n 个句子（句子是以!.?三种符号来确定的）
n)	右括号移动到下句的最前面，前面加上数字可以指定往后移动 n 个句子
n{	左括弧移动到段落的最前面，前面加上数字可以指定往前移动 n 个段落
n}	右括弧移动到下个段落的最前面，前面加上数字可以指定往后移动 n 个段落
Ctrl-u	向上滚屏半页
Ctrl-b	向上滚屏一页（等同 PageUp）
Ctrl-d	向下滚屏半页
Ctrl-f	向下滚屏一页（等同 PageDown）

3. 删除与修改

若用户在编辑文档的过程中出现错误，可以切换到命令模式，将光标移动到需要修改处，然后使用相关命令删除或修改。具体的删除与修改命令如表 14-21 所示。

表 14-21　删除、修改命令

命　令	功 能 描 述
x	删除光标所在字符
dd	删除光标所在的行
r	修改光标所在字符，r 后接着要插入修正的字符
R	进入覆盖状态，直到按“Esc”回到指令模式下为止
s	删除光标所在字符，并进入插入模式
S	删除光标所在的行，并进入插入模式

注意　按下 Esc 键可从插入模式切换到指令模式。

4. 高级编辑指令

前面介绍的都是一些基本的操作，接下来介绍一些高级编辑指令，通过确定范围可以简化操作。例如，使用“y（”指令可以复制光标所在位置到该句子的第一个字母，其中“y”是复制指令，“（”用于指定范围。常用指令如表 14-22 所示。

表 14-22　高级编辑常用指令

指　令	功 能 描 述
d	删除(delete)
y	复制(yank)
p	放置(put)
c	修改(change)

常用范围描述如表 14-23 所示。

表 14-23 常用范围

范围名称	描述
E	光标所在位置到该单词的最后一个字母
w	光标所在位置到下个单词的第一个字母
b	光标所在位置到上个单词的第一个字母
$	光标所在位置到该列的最后一个字母
0	光标所在位置到该列的第一个字母
)	光标所在位置到下个句子的第一个字母
(	光标所在位置到该句子的第一个字母
}	光标所在位置到该段落的最后一个字母
{	光标所在位置到该段落的第一个字母

注意 执行指令进行删除与复制操作时，系统会将指定范围的内容放到缓冲区，以方便用户使用指令“p”将其粘贴到其他位置，这是 vim 用来处理区段复制与搬移的方法。

vim 另外提供了更方便的编辑指令，例如，删除整列的指令“dd”、修改整行的指令“cc”、复制整行文字的指令“yy”、删除光标到该行结束为止所有的文字的指令 D。

5. 查找

与查找相关的指令如表 14-24 所示。

表 14-24 与查找相关的指令

指令名称	功能描述
set nu	显示行号
/或?	查找，在“/”或“？”后输入要查找的内容
n	与“/”一起使用，如果查找的内容不是想要找的关键字，按 n 或向后（与/合用）或向前（与?合用）继续查找，直到找到为止

6. 存盘退出

与存盘退出相关的指令如表 14-25 所示。

表 14-25 与存盘退出相关的指令

命令格式	功能描述
:w filename:	将文章以指定的文件名 filename 保存
:wq filename	储存正在编辑的文件为 filename，并退出 vi
:n,mw filename	将第 n 行到第 m 行的文字存放到指定的 filename
q	存盘并退出 vi
:q!	不存盘强制退出 vi
ZZ	存盘退出（指令模式下使用）

7. 综合实例

编辑 C 语言程序“Hello reader”。

第 1 步，进入 Linux Shell，执行以下指令打开 vim。

```
[root@localhost ~]#vim HelloWorld.c
```

第 2 步，输入“i”进入插入模式，输入以下内容。

```
#include <studio.h>

int main()
{
    printf("Hello World\n");
    returen 0;
}
```

第 3 步，按下 Esc 键回到指令模式，然后执行以下指令存盘退出。

```
:wq
```

14.5.3 vim 的定制

vim 是一个具有强大可定制性的文本编辑器，提供了背景颜色设置和多窗口编辑等功能，用户在使用过程中可以根据需要进行定制。

1. 更换背景颜色

vim 提供了多种背景颜色供用户选择，通过修改颜色配置文件即可更换。vim 的颜色配置文件位于安装文件 colors 下，执行如下指令即可进行背景颜色的切换。

```
:colorscheme
```

执行上述命令后，需要按下 Space 键，然后通过 Tab 键选择喜好的颜色。例如，使用了第一颜色“Blue”后的结果如图 14-7 所示。

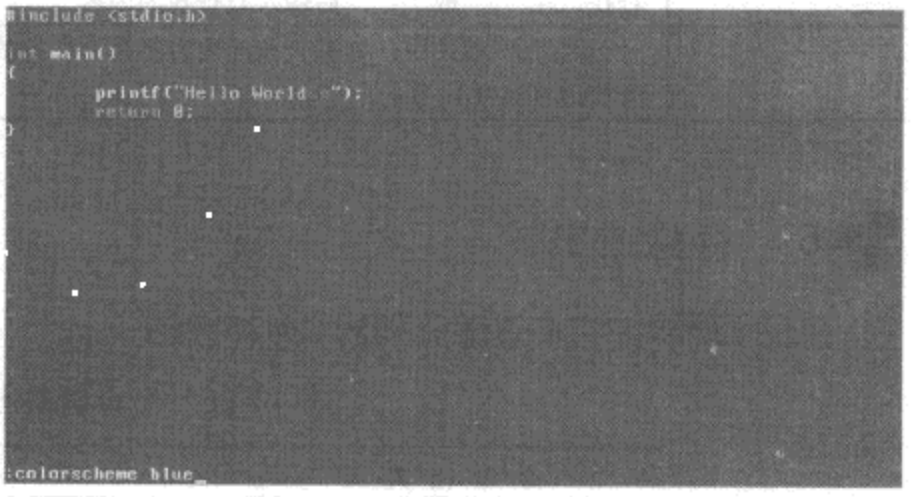


图 14-7 设置 vim 背景颜色

2. 多窗口编辑

执行如下指令可以将当前窗口分为两个窗口并打开 helloworld.cpp 编辑文件。

```
:split helloworld.cpp
```

3. 打开自动缩进功能

默认情况下，vim 的很多功能没有启用，需要用户在家目录下创建一个“.vimrc”文件，然后对这个文件进行相应的配置才能启用。例如，打开自动缩进功能的指令如下。

```
:set autoindent
```

第15章　Linux操作系统中的网络连通性测试

在组建网络之后，网络管理员首先要确保网络设备以及网络节点能够正常工作，同时要确认线路能正常通信。此时则需要一些工具来辅助管理员完成相应的任务。在Linux操作系统中，可以使用工具软件进行网络的配置及测试工作。本章将结合实例介绍如何利用系统的工具来测试网络的连通性。

15.1　网络线路的连通性测试

在一组网络环境中，把网络设置并部署好后，并不代表相互间就能通信成功，可能会存在一些问题，为了避免问题的产生，此时需要进行网络线路的连通性测试。由于计算机所连接的网络设备不同，所选用的双绞线不同，连通性的测试结果也存在差异。一般而言，计算机网卡都是RJ—45接口，使用8芯的双绞线与交换机或HUB连接。

15.1.1　双绞线的线芯

双绞线按照T568A或T568B的标准制作完成后，只要双绞线没有质量问题，都能正常通信。

目前正规的五类或超五类双绞线有四对线芯，所谓五类双绞线是指支持10 M/100 M的带宽，超五类双绞线则是可以支持10 M/100 M/1 000 M的带宽。这四队线芯的具体信息如下。

- 第一对

由序列号1（白橙）和列号2（橙）组成。

- 第二对

由序列号3（白绿）和列号6（绿）组成。

- 第三对

由序列号5（白蓝）和列号4（蓝）组成。

- 第四对

由序列号7（白棕）和列号8（棕）组成。

五类双绞线与超五类双绞线在材质上有所区别，其信号定义上也有一些区别，由于五类双绞线与超五类双绞线都是8芯，因此可以通用，只是五类双绞线的材质要差一些，信号定义如表15-1所示。

表 15-1　五类和超五类双绞线信号定义

RJ－45 接口 ID	T568A 准	T568B 标准	五类双绞线信号义	超五类双绞线信号义
1	白绿	白橙	TX+	BI_DA+
2	绿	橙	TX-	BI_DA-
3	白橙	白绿	RX+	BI_DB+
4	蓝	蓝	未使用	BI_DC+
5	白蓝	白蓝	未使用	BI_DC-
6	橙	绿	RX-	BI_DB-
7	白棕	白棕	未使用	BI_DD+
8	棕	棕	未使用	BI_DD-

15.1.2 测试直通双绞线

直通双绞线是指两端接口都采用 T568A 标准，或都采用 T568B 标准制作的双绞线，其主要用于不同设备的连接，如计算机网卡接口与交换机接口、计算机网卡接口与 HUB 的接口、计算机与光收发器（具有调控功能的光收发器的接口）、交换机的 Uplink 与另一交换机的 MDI 接口、HUB 的 Uplink 与另一 HUB 的 MDI 接口、交换机的 Uplink 与 HUB 的 MDI 接口、交换机的 MDI 与 HUB 的 Uplink 接口。

要测试直通双绞线，可以借助网线测试仪进行测试，该测试仪上有 8 个指示灯，如果测试仪上 8 个指示灯都依次为绿色闪过，证明网线制作成功；如果网线测试仪上任何一个指示灯显示为红灯或黄灯，说明存在断路或者接触不良的问题，此时最好先将双绞线两端的 RJ－45 接头用网线钳再压一次，再次进行测试；如果多次利用网线钳压制后，仍然测试有故障，可以核对双绞线两端的 RJ－45 接头中线芯的颜色是否都按照 T568A 或 T568B 的标准制作，如果双绞线两端线芯的排列顺序不相同，将双绞线的任意一端剪掉，重新制作即可。

另外，如果以上方法都无效，则可能是双绞线存在质量问题，或是网线测试仪已损坏。

15.1.3 测试交叉双绞线

交叉双绞线的一端采用 T568A 标准制作 RJ－45 接头，另一端采用 T568B 标准制作 RJ－45 接头，其用于连接相同设备的相同接口，如计算机网卡与计算机网卡之间的连接，交换机的 MDI 与另一台交换机的 MDI 接口等。

要对交叉双绞线进行测试，则可以利用网线测试仪实现，如图 15-1 所示。测试交叉双绞线的具体方法为：绿色指示灯闪动次序为 3、6、1、4、5、2、7、8，如果测试过程中指示灯闪动的次序与此不相同，就需要剪掉不正确的一端，重新按照相应的标准制作 RJ－45 接头；如果出现其他异常情况，则可以按照直通线的测试方法进行诊断。

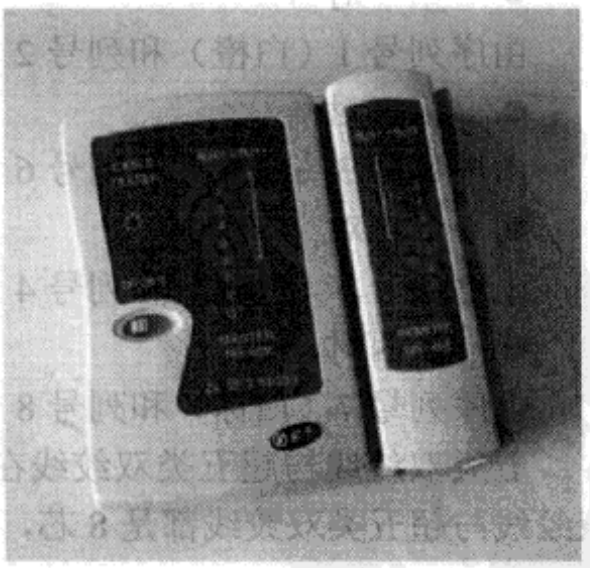


图 15-1　网线测试仪

15.1.4 利用 ping 命令测试网络连通性

ping 命令（Packet internet groper 的缩写形式）是网络测试中使用最频繁的命令之一，主要用于确定网络的连通性问题。该命令对确定网络是否正确连接，以及网络连接的状况（包丢失率）十分有用，同时还可以提供基本的网络性能统计数据，该数据可用于诊断与通信质量相关的网络问题。

1. ping 命令格式

ping 命令格式如下。

```
ping [选项] <主机名或 IP 地址>
```

该命令各选项及其说明如表 15-2 所示。

表 15-2 ping 命令各选项说明

命 令 选 项	相 关 说 明
-d	使用 Socket 的 SO_DEBUG 功能
-c<发送包次数>	设置发送测试包的次数
-f	极限检测，利用报文进行泛洪
-i<间隔秒数>	指定发送数据包的间隔时间
-I<网络界面>	使用指定的网络界面送出数据包
-l<前置载入>	设置在送出要求信息之前，先行发出的数据包
-n	在返回信息中不显示目标主机的名称，而只是显示 IP 地址
-p<范本样式>	设置填满数据包的范本样式
-q	不显示“ping”命令的执行过程，只显示结果统计信息
-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	详细显示命令的执行过程

2. 测试网络设备系统可用性

ping 命令可用于测试网络设备的一般可用性，测试本机与网络中一个 ADSL Modem 是否可用，可以执行以下命令。

```
[root@localhost ~]# ping -c 4 192.168.1.9
```

执行后的结果如下所示。

```
[root@localhost ~]# ping -c 2 192.168.1.6
PING 192.168.1.6 (192.168.1.6) 56(84) bytes of data.
64 bytes from 192.168.1.6: icmp_seq=1 ttl=64 time=12.7 ms
64 bytes from 192.168.1.6: icmp_seq=2 ttl=64 time=0.372 ms
--- 192.168.1.6 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 999ms
rtt min/avg/max/mdev = 0.372/6.560/12.749/6.189 ms
```

由以上返回结果可以看出，本机利用“ping”命令向 ADSL Modem 发出了两条测试信息，总计返回了两条报文，说明测试过程中没有报文丢失。

注意 若本机与 ADSL Modem 不能正常通信，则返回如下报文信息。

```
PING 192.168.1.6 (192.168.1.6) 56(84) bytes of data.
--- 192.168.1.6 ping statistics ---
2 packets transmitted, 0 received, 100% packet loss, time 3000ms
```

使用“ping”命令测试与网络中其他设备之间的连通性时，若不加任何参数，而是直接 ping 网络中的主机，则本机将会一直不断地向目标设备发送测试数据包，并返回报文信息，直到按下 Ctrl+C 组合键结束发送数据包，并统计报文信息才结束。

另外，在使用“ping”命令时，若本机与要测试的目标机不在同一网络中，而这两个网络间的路由设备又没有正常工作，那么利用“ping”命令测试连通性的结果报文将全部丢失，表示网络连接不通。

3. 测试网络性能

使用 ping 命令测量发送一个消息到远程目标的时间的本质，就是测试给定时长的两个设备之间路径的有关性能，但不是设备间连通性或设备的详细分析。在不同的时间测试相同的网络通信性能，返回结果都有可能不同，这是由于影响网络通信质量的因素有很多（包括天气、网络设备所处的环境湿度、噪音等一系列因素）。

该命令提供确定系统响应时间的方法，但其更多的是做一些用以确定观察到的性能问题的方法，确认其性能是否与系统反应较慢或网络组件的延迟相关。默认情况下，ping 命令每秒向远程主机发出一个 ICMP 请求，在收到返回的报文信息后，再计算发出 ICMP 请求与收到报文信息的时间间隔，执行以下命令。

```
[root@localhost ~]# ping 192.168.1.3
```

执行后结果如下所示。

```
[root@localhost ~]# ping 192.168.1.104
PING 192.168.1.104 (192.168.1.104) 56(84) bytes of data.
64 bytes from 192.168.1.104: icmp_seq=1 ttl=64 time=1.27 ms
64 bytes from 192.168.1.104: icmp_seq=2 ttl=64 time=1.10 ms
64 bytes from 192.168.1.104: icmp_seq=3 ttl=64 time=1.10 ms
64 bytes from 192.168.1.104: icmp_seq=4 ttl=64 time=1.14 ms
64 bytes from 192.168.1.104: icmp_seq=5 ttl=64 time=1.15 ms
64 bytes from 192.168.1.104: icmp_seq=6 ttl=64 time=0.731 ms
--- 192.168.1.104 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5001ms
rtt min/avg/max/mdev = 0.731/1.085/1.278/0.171 ms
```

从返回的报文信息中可以看出，每条报文信息中都包括报文尺寸、目标主机名或 IP 地址、报文序号、返回时间值和统计信息摘要，由以上返回报文结果可以看出，向 IP 地址 192.168.1.104 发送数据包请求测试，每个数据包的大小为 64 B，用户可以根据自己的需要更

改数据包的大小，方法是使用 ping 命令时添加“-s”选项。例如，向主机 IP 为 192.168.1.104 的主机发送数据包大小为 50 B 的信息，执行如下命令。

```
[root@localhost ~]#ping -s 50 192.168.1.104
```

执行命令后的结果如下所示。

```
[root@localhost ~]# ping -s 50 192.168.1.104
PING 192.168.1.104 (192.168.1.104) 50(78) bytes of data.
58 bytes from 192.168.1.104: icmp_seq=1 ttl=64 time=1.15 ms
58 bytes from 192.168.1.104: icmp_seq=2 ttl=64 time=2.57 ms
58 bytes from 192.168.1.104: icmp_seq=3 ttl=64 time=1.14 ms
58 bytes from 192.168.1.104: icmp_seq=4 ttl=64 time=1.21 ms
58 bytes from 192.168.1.104: icmp_seq=5 ttl=64 time=1.08 ms
58 bytes from 192.168.1.104: icmp_seq=6 ttl=64 time=1.09 ms
--- 192.168.1.104 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5004ms
rtt min/avg/max/mdev = 1.089/1.378/2.578/0.538 ms
```

注意 发送数据包的体积越大，则需要的网络性能越好，因此发 50 B 的数据包要比发送 64 B 的数据包所要求的网络通信质量更高。

由以上的返回结果可以看出，“ping”命令要求发送的数据包大小为 50 B，而实际发送的数据包大小为 58 B，也就是说在指定的数据包大小上加了 8 B。尽管数据包的大小尺寸有所增加，但是不会对网络有太大的影响。

另外，ping 命令还使用序号来跟踪请求和应答，每个请求的序号给出下一个数值并且每个请求用相应的应答来匹配，该序号化是在请求没有收到相应的应答的情况下，用来确定报文是否丢失的。

实例 测试与美国 HP 网站的连通性，执行以下命令。

```
[root@localhost ~]# ping -c 10 www.hp.com
```

执行后结果如下所示。

```
[root@localhost ~]# ping -c 10 www.hp.com
PING www.hpgtm.nsatc.net (15.192.45.26) 56(84) bytes of data.
--- www.hpgtm.nsatc.net ping statistics ---
10 packets transmitted, 0 received, 100% packet loss, time 9013ms
```

由以上返回的报文结果可以看出，本机总共发出了 10 个测试数据包，返回了 0 个报文，表明报文全部丢失。

注意 报文丢失率还与系统在后继的 ICMP 请求之间需要的额外时间间隔有关，时间间隔越长，则报文丢失率就越低；否则，相反。

设置时间的具体方法为：在“ping”命令后添加“-i”选项，并指定时间间隔值。

实例 测试与中国 DELL 网站的通信情况，设置时间延迟 5 s，可以执行如下命令。

```
[root@localhost ~]# ping -c 10 -i 5 www.dell.com.cn
```

执行后结果如下所示。

```
[root@localhost ~]# ping -c 5 -i 5 www.dell.com.cn
PING a946.cn.w.tl88.net (118.123.240.16) 56(84) bytes of data.
64 bytes from 118.123.240.16: icmp_seq=1 ttl=56 time=1206 ms
64 bytes from 118.123.240.16: icmp_seq=2 ttl=56 time=61.2 ms
64 bytes from 118.123.240.16: icmp_seq=3 ttl=56 time=15.5 ms
64 bytes from 118.123.240.16: icmp_seq=4 ttl=56 time=16.9 ms
64 bytes from 118.123.240.16: icmp_seq=5 ttl=56 time=16.3 ms
--- a946.cn.w.tl88.net ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 22241ms
rtt min/avg/max/mdev = 15.581/263.313/1206.382/471.855 ms
```

由以上报文信息可以看出，中国 DELL 网站的通信状态良好，没有任何报文丢失，这与延迟报文时间有关，还和网络繁忙状态也有关。

4. ping 命令的其他选项

前面大家已经接触到 ping 命令的相关选项，可知 ping 命令的选项不少，但常用的也只有“-c”、“-s”、“-i”、“-q”和“-n”几个选项。

如果添加“-f”选项，则可向目标计算机发送大量的 ICMP 请求，速度就是目标主机 CPU 的处理速度，但是添加此参数后会导致网络阻塞，或是使目标主机瘫痪。

实例 向网络中的主机 192.168.1.104 发送 10 条 ICMP 数据请求，则只显示报文统计结果，执行以下命令。

```
[root@localhost ~]# ping -c 10 -q 192.168.1.104
```

执行后结果如下所示。

```
[root@localhost ~]# ping -c 10 -q 192.168.1.104
PING 192.168.1.104 (192.168.1.104) 56(84) bytes of data.
--- 192.168.1.104 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9003ms
rtt min/avg/max/mdev = 0.662/1.970/11.445/3.165 ms
```

15.2 路由跟踪

在 Internet 中，为了方便不同网段中的各主机相互通信，需要使用路由器来实现此功能，路由器可以将 IP 报文信息从一个接口转发到另一个接口。利用 traceroute 命令可以进行路由的跟踪，该命令探测并记录到达指定网络目标的路径。

15.2.1 traceroute 命令

traceroute 命令通过以下过程实现跟踪路由的功能，向外发送生存期（TTL）为 1 的请求

（或探测）并将 TTL 加 1，到达目标主机或超出 TTL 最大值时结束。默认情况下，TTL 的最大值为 30，也可根据实际情况改变该设置。

1. traceroute 命令格式

traceroute 命令格式如下。

```
traceroute [选项] <主机名或 IP 地址>
```

该命令的各选项及说明如表 15-3 所示。

表 15-3 traceroute 命令的选项

命 令 选 项	相 关 说 明
-d	使用 socket 层级的排错功能
-f<存活数值>	设置第一个检测数据包存活数值 TTL 的大小
-F	设置勿离断位
-g<网关>	设置来源路由网关，最多可设置 8 个
-i<网络界面>	使用指定的网络界面送出数据包
-I	使用 ICMP 回应取代 UDP 资料信息
-m<存活数值>	设置检测数据包的最大存活数值 TTL 的大小
-n	直接使用 IP 地址而非主机名称
-p<通信端口>	设置 UDP 传输协议的通信端口
-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>	设置等待远程 tkgg 主机回报的时间
-x	开启或关闭数据包的正确性检验

2. 具体实例

实例 显示本机到 192.168.1.6 主机的路由列表。可以通过执行以下命令实现。

```
[root@localhost ~]# traceroute 192.168.1.6
```

traceroute 命令对目标由什么构成没有任何限制，只要该设备支持 IP 地址。

注意 默认情况下，“traceroute”命令总共发送了 3 个探测（各自有不同的 TTL 值）到每一跳。

15.2.2 traceroute 显示代码

使用 traceroute 命令在跟踪远程主机的路由的过程中，由于沿途中各路由设备的环境变化、繁忙等情况，可能造成网络波动和一些无法预知的 traceroute 输出，这些输出将在屏幕上以相应的代码进行显示，具体说明如表 15-4 所示。

表 15-4 traceroute 显示的代码

显示代码	具体说明
*	探测报文没有获得响应
!	接收到的报文中的 TTL 值为 1
!H	目标主机不可到达
!N	目标网络不可到达
!P	目标协议不可到达
!S	源路由选项错误，一般情况下该代码不会出现，若出现该代码，则表示网络中某一路由器有程序错误或硬件故障
!F	探测报文需要分段，一般情况下该代码不会出现，若出现该代码，则表示网络中某一路由器有程序错误或硬件故障
!X	由于通信被管理性禁止，该路径被阻塞或关闭
!N>	一个 ICMP 错误代码，N 是错误号

实例 使用 traceroute 命令跟踪“www.google.com”主机的路由地址，具体指令如下。

```
[root@localhost ~]# traceroute www.google.com
traceroute to www.usa.com (206.207.87.12), 30 hops max, 38 byte packets
 1  help5 (192.168.0.1)  2.073 ms  2.193 ms  8.422 ms
 2  1.39.67.125.broad.dz.sc.dynamic.163data.com.cn (125.67.39.1)  25.904 ms
21.420 ms  24.930 ms
.....
15  gi2-8.dc1.boi.spro.net (198.60.194.26)  349.814 ms  340.946 ms  339.913
```

从以上返回结果可以看出，出现从本地到达网络 206.207.87.12 超时，并以“*”指明该问题错误，可能是由于远程路由繁忙所导致。

15.2.3 traceroute 命令选项的应用

在 traceroute 命令的后面跟上相应的选项，就可指定跟踪方式或者显示结果。

例如，由于 TTL 的默认最大值是 30，当目标主机所处的网络与本机所处的网络间路由数较多（多于 30 个）时，则无法跟踪到目标，此时可利用 traceroute 命令的“-m”参数进行修改，跟踪 IP 地址为 128.197.2.200 的主机，并设置其最大路由量为 35，操作指令如下。

```
[root@localhost ~]#traceroute -m 35 128.197.2.200
traceroute to 128.197.2.200 (128.197.2.200), 35 hops max, 38 byte packets
 1  help5 (192.168.0.1)  58.712 ms  2.486 ms  3.039 ms
 2  1.38.67.125.broad.dz.sc.dynamic.163data.com.cn (125.67.38.1)  8.921 ms
12.405 ms  15.580 ms
 3  61.188.246.49 (61.188.246.49)  24.282 ms  21.901 ms  9.612 ms
 4  222.213.1.93 (222.213.1.93)  10.311 ms  14.585 ms  11.344 ms
 5  202.97.70.105 (202.97.70.105)  34.990 ms  10.744 ms  14.167 ms
 6  202.97.70.50 (202.97.70.50)  31.417 ms  17.039 ms  15.630 ms
.....
```

在默认情况下，traceroute 命令基于 UDP 协议发送探测报文，但在网络中并非所有的路由器都支持 UDP 协议，此时可利用 traceroute 命令的“-I”参数，表示使用 ICMP 协议来替代 UDP 协议。

```
[root@localhost ~]# traceroute -m 35 -I 128.197.2.200
```

```
traceroute to 128.197.2.200 (128.197.2.200), 35 hops max, 38 byte packets
 1  help5 (192.168.0.1)  58.003 ms  1.814 ms  5.073 ms
 2  1.38.67.125.broad.dz.sc.dynamic.163data.com.cn (125.67.38.1)  15.942 ms
14.624 ms  23.995 ms
 3  61.188.246.49 (61.188.246.49)  17.672 ms  18.352 ms  14.571 ms
 4  222.213.1.93 (222.213.1.93)  12.075 ms  13.324 ms  10.050 ms
 5  202.97.70.105 (202.97.70.105)  30.344 ms  11.002 ms  13.698 ms
 6  202.97.70.50 (202.97.70.50)  31.737 ms  9.950 ms  16.890 ms
 7  202.97.43.153 (202.97.43.153)  56.029 ms  33.470 ms  35.080 ms
 8  202.97.33.118 (202.97.33.118)  69.581 ms  50.986 ms  48.959 ms
 .....
```

若网络中所有的路由设备都支持 UDP 协议，默认情况下 traceroute 命令将使用 UDP 协议的 33434 端口进行跟踪，当该端口被其他应用程序占用了，则可使用 traceroute 命令的“-p”参数指定端口号，此时系统将会以新的端口号替代默认端口号。

```
[root@localhost ~]# traceroute -m 35 -p 65000 128.197.2.200
traceroute to 128.197.2.200 (128.197.2.200), 35 hops max, 38 byte packets
 1  help5 (192.168.0.1)  1.138 ms  0.934 ms  0.998 ms
 2  1.131.215.222.broad.dz.sc.dynamic.163data.com.cn (222.215.131.1)  41.987
ms  8.872 ms  23.120 ms
 3  61.188.246.49 (61.188.246.49)  18.312 ms  11.548 ms  11.127 ms
 4  222.213.1.93 (222.213.1.93)  17.880 ms  12.073 ms  11.786 ms
 5  202.97.70.105 (202.97.70.105)  32.934 ms  9.128 ms  16.923 ms
 6  202.97.70.50 (202.97.70.50)  33.828 ms  16.257 ms  15.953 ms
 7  202.97.43.153 (202.97.43.153)  63.941 ms  32.359 ms  33.912 ms
 8  202.97.33.158 (202.97.33.158)  107.944 ms  42.156 ms  115.410 ms
 9  202.97.51.222 (202.97.51.222)  158.935 ms  132.942 ms  135.896 ms
10   p11-0.core01.lax04.atlas.cogentco.com  (154.54.13.105)   237.503  ms
184.341 ms  222.914 ms
11  p0-0.core01.lax01.atlas.cogentco.com (154.54.6.241)  269.376 ms  242.941
ms
 .....
```

traceroute 命令有“-v”和“-n”两个选项来设置输出结果：“-v”选项用来为每一跳显示响应报文的大小和响应报文的目标；“-n”选项则用来显示地址而不是使用符号化的名字显示。这样就使 traceroute 命令不用解析每个路由的主机名，从而减轻网络服务的负荷。

例如，跟踪本机到“www.google.cn”的路由，显示每一跳的报文大小及响应报文的目标，同时不显示路由的主机名称，其操作指令如下。

```
[root@localhost ~]# traceroute -v -n www.google.cn
traceroute:  Warning:  www.google.cn  has  multiple  addresses;  using
203.208.33.100
traceroute to cn.l.google.com (203.208.33.100), 30 hops max, 38 byte packets
 1  192.168.0.1 36 bytes to 192.168.0.10  17.073 ms  5.017 ms  4.737 ms
 2  222.215.131.1 36 bytes to 192.168.0.10  18.838 ms  32.129 ms  22.102 ms
 3  61.188.246.49 36 bytes to 192.168.0.10  11.431 ms  17.027 ms  30.551 ms
 4  222.213.1.93 36 bytes to 192.168.0.10  17.813 ms  22.247 ms  26.607 ms
 5  202.97.70.105 36 bytes to 192.168.0.10  31.458 ms  25.109 ms  16.201 ms
 6  202.97.42.37 36 bytes to 192.168.0.10  36.777 ms  24.252 ms  10.232 ms
 7  202.97.34.129 36 bytes to 192.168.0.10  38.218 ms  44.183 ms  37.530 ms
```

由以上返回结果可以看出，其跟踪的速度要比没有添加“-n”选项时的速度快得多。

15.3 Linux 图形界面下测试网络

使用 Linux 操作系统所提供的图形操作界面，可以使操作更为便捷、直观，省去了命令行操作的麻烦。Linux 图形界面下的网络连通性测试可以利用“Network Tools”工具实现。

15.3.1 进入“Network Tools”窗口

在 Linux 中，利用“Network Tools”工具进行测试网络的工作需要在“Network Tools”窗口中进行，打开该窗口的具体方法如下。

依次单击“应用程序”→“系统工具”→“网络工具”菜单命令，将弹出“设备-Network Tools”窗口。在该对窗口中可以看到，有“设备”、“Ping”、“网络统计”、“Traceroute”、“端口扫描”、“查询”、“Finger”和“Whois”8 个选项卡，如图 15-2 所示。

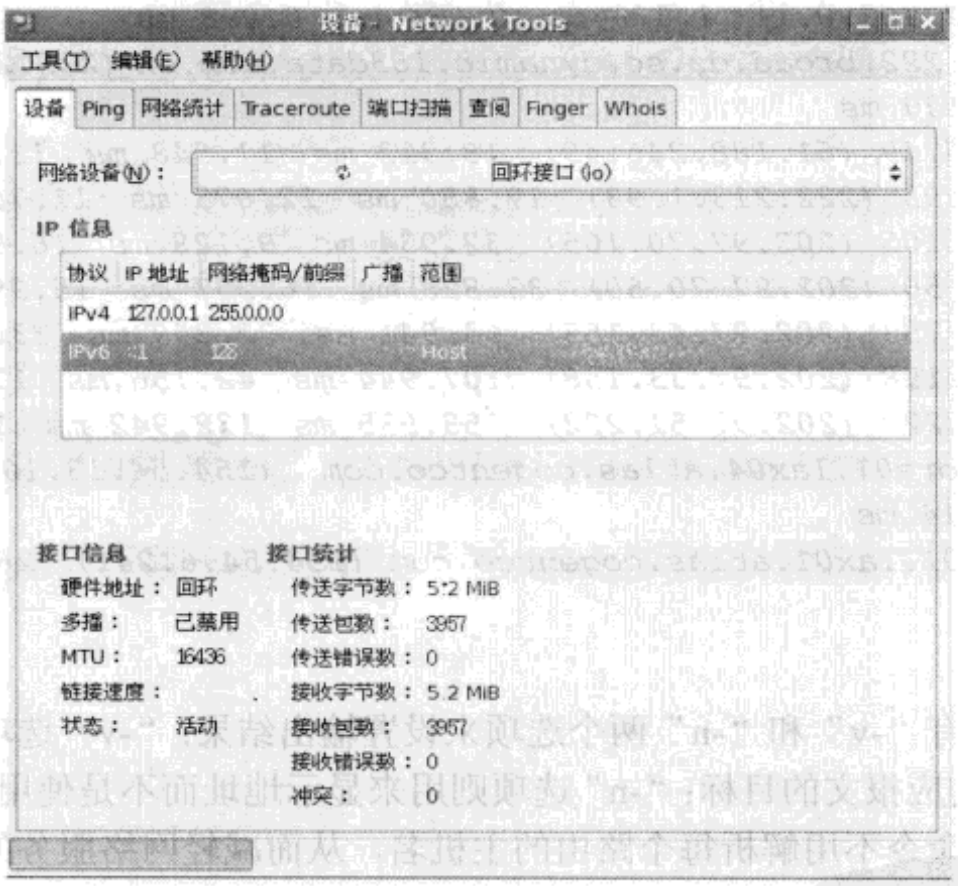


图 15-2 “设备-Network Tools”窗口

注意 如果找不到“网络工具”子菜单，则可以在“软件包管理者”窗口中安装相应组件。具体方法为：依次单击“应用程序”→“添加/删除软件”菜单命令，将弹出“软件包管理者”窗口。在该窗口中的浏览选项页下选择基本系统，然后在窗口右侧选择系统工具，再单击“可选的软件包”按钮，此时将弹出“系统工具中的软件包”对话框。选中所需软件包，点击关闭按钮，此时即可返回到“软件包管理者”窗口，最后单击该窗口的应用按钮即可。

15.3.2 网络设备查询

网络设备查询的具体方法为，在“Network Tools”窗口中选择“设备”选项卡，此时将在“网络设备”列表框中列出所有的网络设备，然后单击“网络设备”下拉列表，在弹出的列表中选择“回环接口(lo)”项，此时将在“IP 信息”框中显示回环网址。IPv4 的回环网址是“127.0.0.1”，IPv6 的回环网址是“::1”，具体的显示如图 15-2 所示。

在“网络设备”下拉列表中选择“以太网卡(eth0)”项，后将在“IP 信息框”中显示已设置好的 IPv4 地址和 IPv6 地址，在窗口的下方还将会显示网卡的接口信息和接口统计信息，如图 15-3 所示。

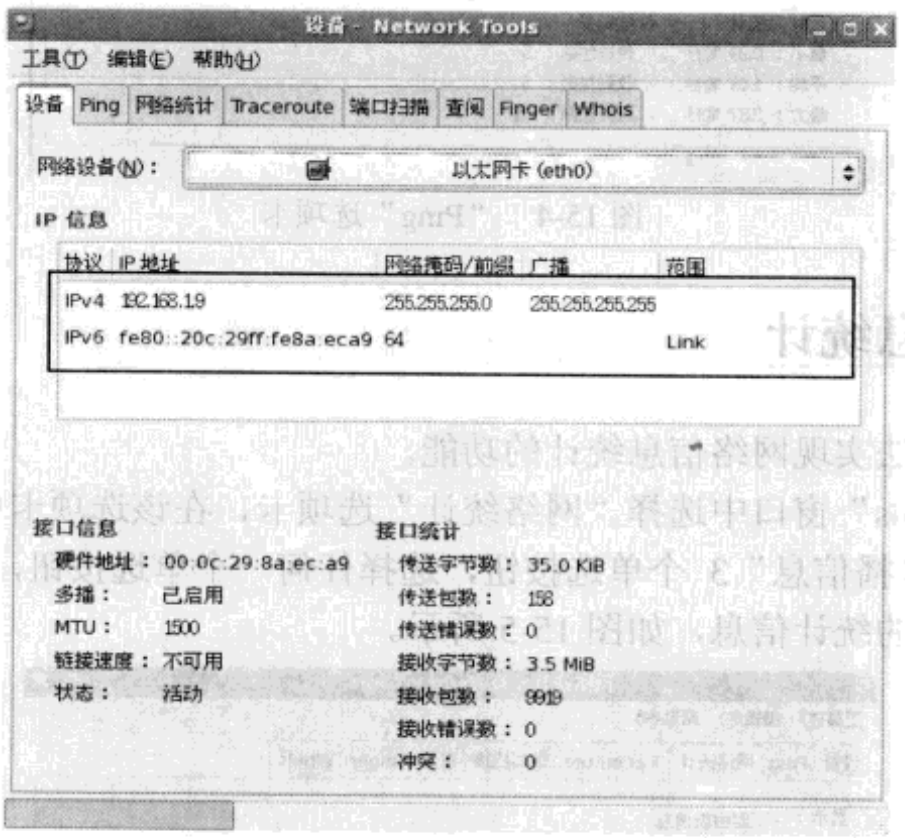


图 15-3 “设备”选项卡

15.3.3 网络连通性测试

网络连通性测试的具体方法为：在“Network Tools”窗口中选择“Ping”选项卡，在“网络地址”文本框中输入目标主机或目标设备的 IP 地址，也可输入网络中主机的名称或网络设备的名称。输入目标主机名和 IP 地址后，再在“发送”文本框中输入发送数据包的数量，也可选择“发送无限制请求”单选按钮，再单击“Ping”按钮即可在文本框中显示测试结果，如图 15-4 所示。

从上图中的返回结果可以看出，对网络中 IP 地址为 192.168.1.3 的主机进行测试，共计发送 5 个数据包，每个数据包的大小为 64 B，信息包响应返回最短时间为 0.26 ms，最长返回时间为 3.87 ms，平均返回时间为 1.48 ms，证明本机与 IP 为 192.168.1.3 的主机能正常进行通信。

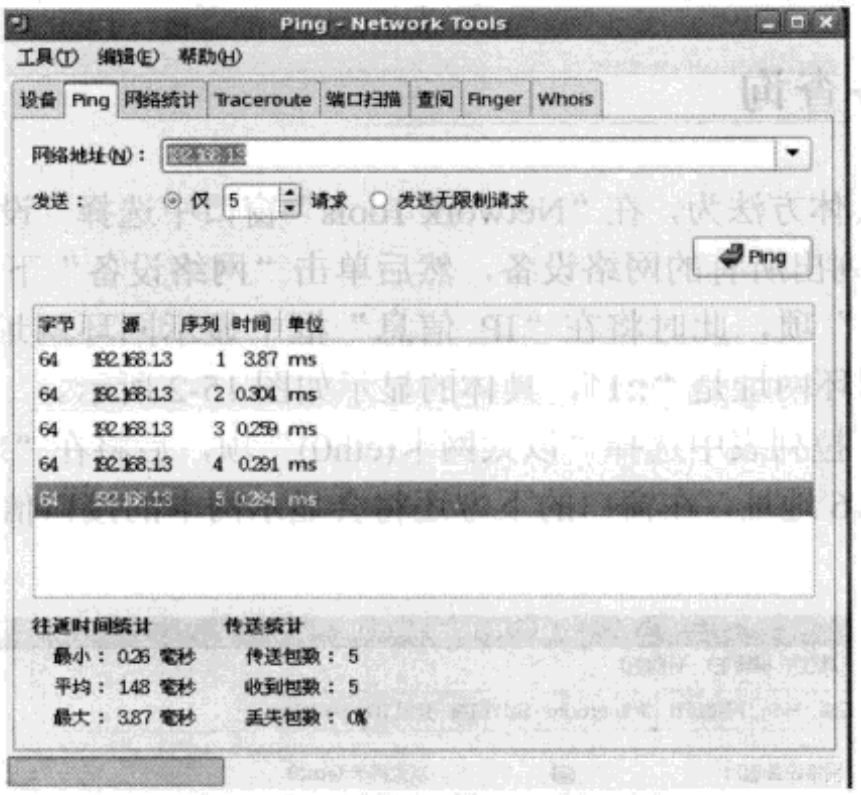


图 15-4 “Ping”选项卡

15.3.4 网络信息统计

可以通过以下方法实现网络信息统计的功能。

在“Network Tools”窗口中选择“网络统计”选项卡，在该选项卡中有“路由表信息”、“激活网络服务”、“多播信息”3 个单选按钮，选择任何一个单选按钮，再单击“网络统计”按钮即可显示出对应的统计信息，如图 15-5 所示。

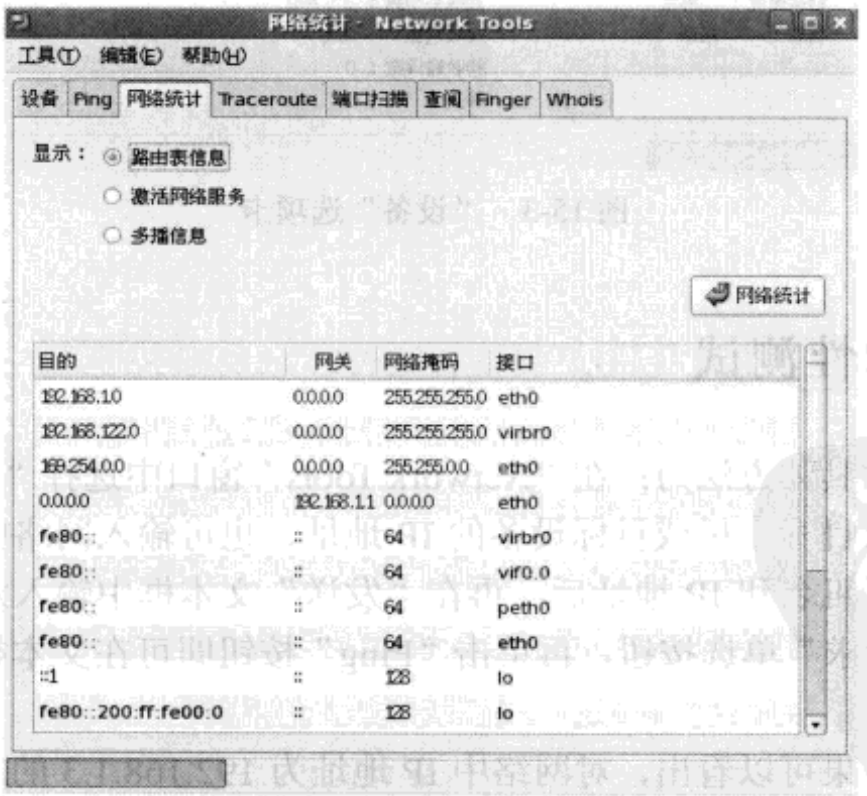


图 15-5 “网络统计”选项卡

15.3.5 网络路由跟踪

Internet 网络体系非常庞大，彼此之间通过非常多的网络设备进行连接，可以利用 Linux

中的网络路由跟踪功能跟踪由本地计算机到指定网站需要经过哪些站点，具体方法如下。

在“Network Tools”窗口中选择“Traceroute”选项卡，在“网络地址”文本框中输入目的网站地址（如 www.baidu.com），然后单击“跟踪”按钮，此时将在列表框中显示出本机到达目标网站经过了哪些站点，并返回了每个站点间的数据包响应时间，如图 15-6 所示。

15.3.6 网络端口扫描

通过网络端口扫描功能，可以扫描出目标主机当前所开放的端口号。具体方法为：在“Network Tools”窗口中选择“端口扫描”选项卡，然后在网络地址栏中输入目标主机的 IP 地址“192.168.1.9”，再单击“扫描”按钮，即可在列表框中看到该主机所开放的端口，如图 15-7 所示。

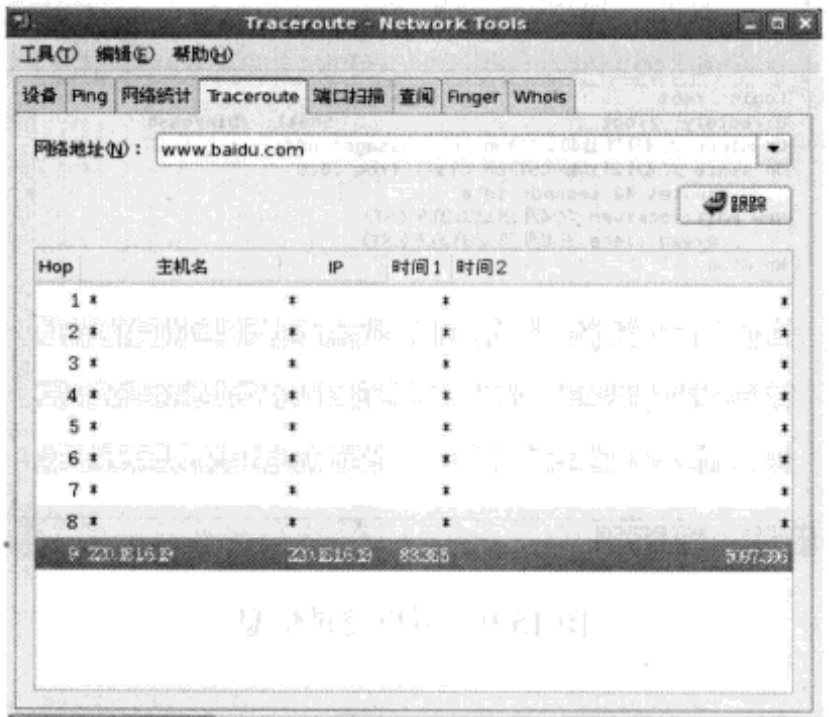


图 15-6 “Traceroute”选项卡

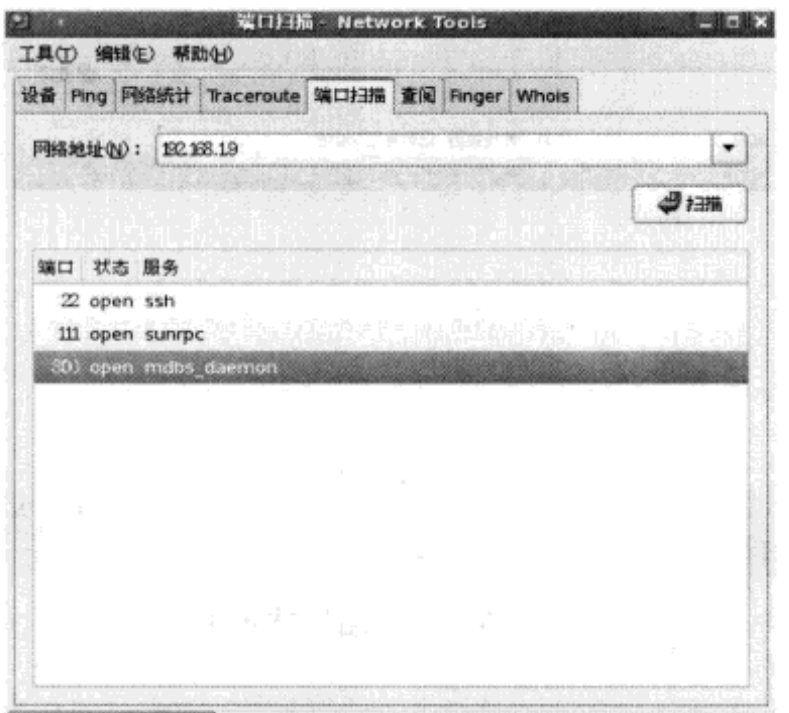


图 15-7 “端口扫描”选项卡

注意 在网络地址文本框中可输入目标主机的域名（如 WWW.Goole.cn），或者是目标主机所对应的 IP 地址（如 203.208.33.100），如果扫描的是 Internet 网络中的目标主机，则需要先将该主机接入 Internet。

15.3.7 网络查阅

实现网络查阅功能的具体方法如下。

在“Network Tools”窗口中选择“查阅”选项卡，然后在“网络地址”文本框中输入目标主机的名称或 IP 地址，并在“信息类型”列表框中选择需要显示的信息类型，最后单击“查阅”按钮，即可在窗口的下方显示出所需要的信息，如图 15-8 所示，显示了目标主机的反向解析名称。

15.3.8 查询登录用户的信息

查询登录用户信息的具体方法为：在“Network Tools”窗口中选择“Finger”选项卡，在该选项卡中直接输入需查询的主机名称或 IP 地址，然后单击“Finger”按钮，即可显示出指定主机上已登录的用户。

图 15-9 所示的信息即为用户名为“root”的用户登录信息。

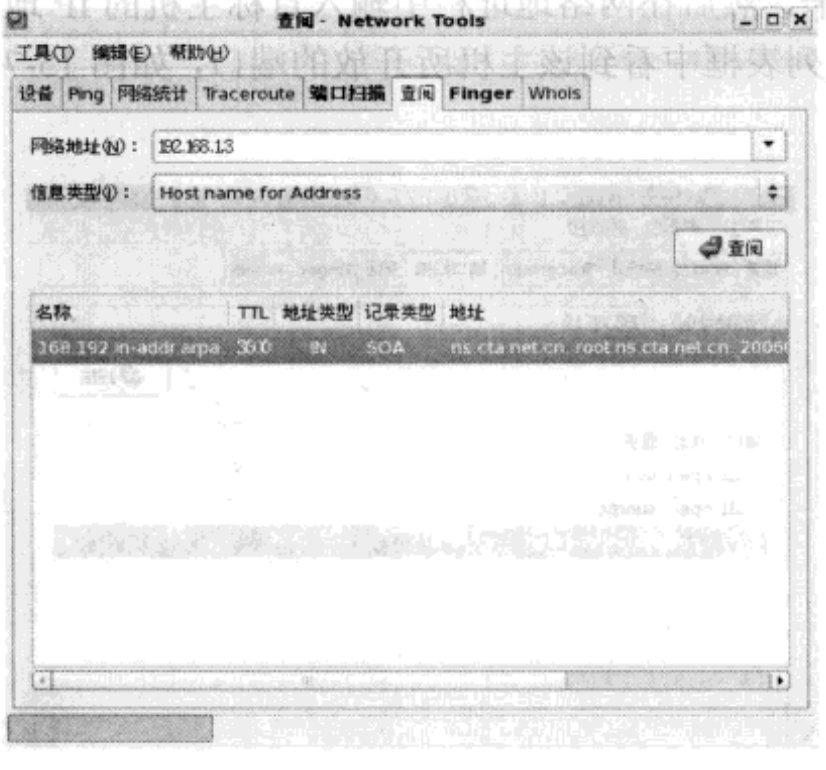


图 15-8 “查阅”选项卡



图 15-9 用户登录信息

15.3.9 域名查询工具

在“Network Tools”中存在一个域名查询工具“Whols”，利用该工具可以查询指定域名的注册公司名、网络营运商名称、域名注册日期和到期日期等信息。

例如，如果要查询“qq.com.cn”域名的注册公司、网络营运商、DNS 服务器名称、注册日期、到期日期等信息，则可以在“Network Tools”窗口中选择“Whols”选项卡，然后在“域地址”文本框中输入“qq.com.cn”，单击“Whols”按钮，即可显示相关信息，如图 15-10 所示。

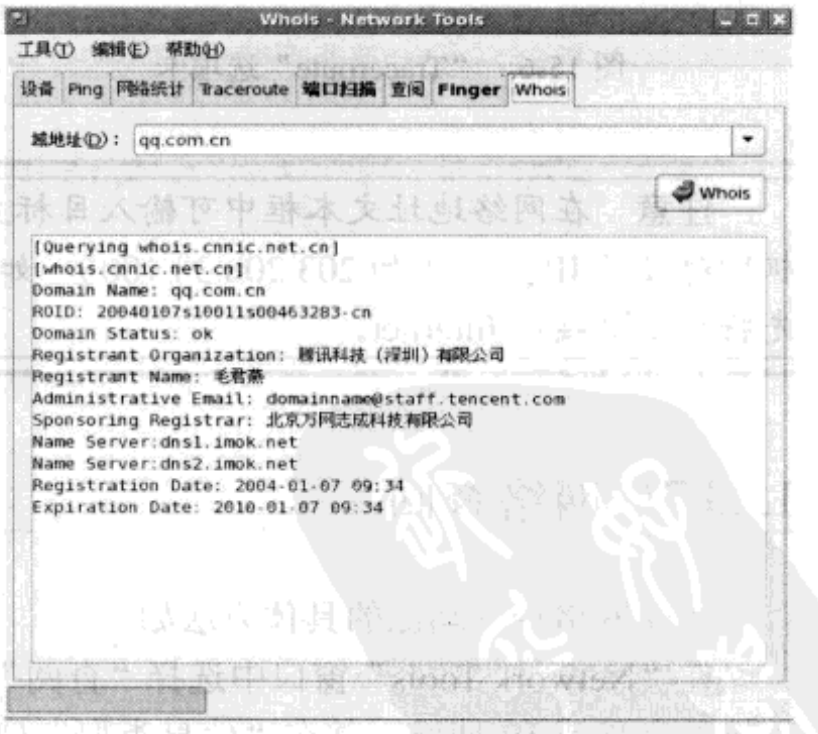


图 15-10 “Whols”选项卡

第 16 章　Linux 网络管理

网络管理的基本操作之一就是对网络设备进行连接并测试是否能够正常通信。要实现这一目的，除了保证物理线路的连通之外，还必须在计算机上进行相应的设置。Linux 操作系统常用于网络服务器，为保证通信和数据安全，系统的网络管理工作就显得极其重要。

16.1　静态 IP 地址管理工具

如果网络中一台计算机要与另一台计算机进行通信，那么这两台计算机都必须要有唯一的一个标识来识别访问。这个标识就是计算机的 IP 地址，网络中 IP 地址的管理非常重要，只有配置正确无误，各计算机之间才能通过 IP 地址进行互访。下面简单介绍一下 IP 地址的管理。

16.1.1　IP 地址

IP 地址是由一个长度为 32 位的二进制数字串来表示的，通常情况下每 8 位分为一组，各组之间用小数点“.”隔开。用户平时所看到的 IP 地址，诸如“192.168.1.125”的形式就是由这样一组二进制数转换为十进制数而来的。

IP 地址采用的分级寻址方案可以更快地在网络中找到目标地址。默认情况下的 IP 地址分为两个部分，一个部分为网络号，另一部分为主机号。网络号通常标识一个主机所在的区域，而主机号则表示在本区域内唯一确定的一台主机。根据网络号的划分方法不同，将 IP 地址空间分为 A、B、C、D 和 E 5 类，其中 A、B、C 3 类为常规 IP 地址，如表 16-1 所示，D 类 IP 地址用于群组广播，E 类 IP 地址则保留。

表 16-1　3 类 IP 地址

类　别	A　类	B　类	C　类
网络号位数（十进制）	3	6	9
主机号位数（十进制）	9	6	3
可用网络号数量	126	16256	2064512
可用主机号数量	16387064	64516	254
IP 地址范围	0.0.0.0 ~ 127.255.255.255	128.0.0.0 ~ 191.255.255.255	192.0.0.0 ~ 223.255.255.255
其中私有 IP 地址	10.*.*.*	172.16.*.* ~ 172.32.*.*	192.168.0.* ~ 192.168.255.*

表中的私有 IP 地址属于私有网络，这些 IP 地址在局域网中使用。当局域网中使用私有 IP 地址的各计算机需要与 Internet 连接时，则需要在连接处进行 NAT（Network Address Translation，网络地址转换）或使用代理服务器。

16.1.2 子网掩码

随着 Internet 的发展，原有的 IP 地址现在已经不够使用，所以就将 IP 地址又分为子网和主机两个部分。旧的寻址方式为计算机根据 IP 地址的类型就可判断网络号和主机号，但有了子网的概念后，计算机就无法判断 IP 地址中哪部分是子网，哪部分是主机号，所以，引出了子网掩码的概念。

与 IP 地址类似，子网掩码也是一个 32 位的二进制数（12 位十进制数），子网掩码前面部分的二进制传为 11111111（十进制数的 255），后面部分全部为 00000000（十进制数的 0），子网掩码中二进制值为 11111111 的部分对应子网号，二进制值为 00000000 的部分对应主机号。最简单的掩码就是原有的 A、B、C 三类网络所使用的掩码，分别是：A 类网络的子网掩码为 255.0.0.0，B 类网络的子网掩码为 255.255.0.0，C 类网络的子网掩码为 255.255.255.0。

16.1.3 TCP/IP 配置文件

在 Red Hat Linux 操作系统中，几乎所有的配置都可以通过配置文件的形式来实现各种设置，TCP/IP 网络使用的配置文件如表 16-2 所示。

表 16-2 TCP/IP 配置文件

配置文件名	说　明
/etc/sysconfig/network	包含了主机最基本的网络信息，用于系统启动
/etc/sysconfig/network-script/	该目录下存储的就是系统启动时用来初始化网络适配器的一些信息
/etc/xinetd.conf	定义由超级进程 xinted 启动的网络服务
/etc/hosts	完成主机名映射为 IP 地址的功能
/etc/networks	完成域名与网络地址的映射
/etc/host.conf	配置域名服务客户端的控制文件
/etc/resolv.conf	配置域名服务客户端的配置文件，用于指定域名服务器的位置
/etc/protocols	设定了主机使用的协议以及各个协议的协议号
/etc/services	设定主机不同端口的网络服务

16.1.4 在图形界面下配置以太网络

为了使网络配置更加方便，减少配置过程中出错的可能，Linux 提供了一个图形界面的网络配置工具，用于配置各种网络连接。单击“系统”→“管理”→“网络”菜单命令，将弹出如图 16-1 所示的“网络配置”窗口。

1. 配置网卡并指定 IP 地址

在此窗口中单击“新建”按钮，弹出“选择设备类型”窗口，在该窗口中选择“以太网连接”标签，如图 16-2 所示。

单击“前进”按钮，弹出“选择以太网设备”窗口，如图 16-3 所示，在该窗口中可直接看到已安装到计算机上的网卡设备。

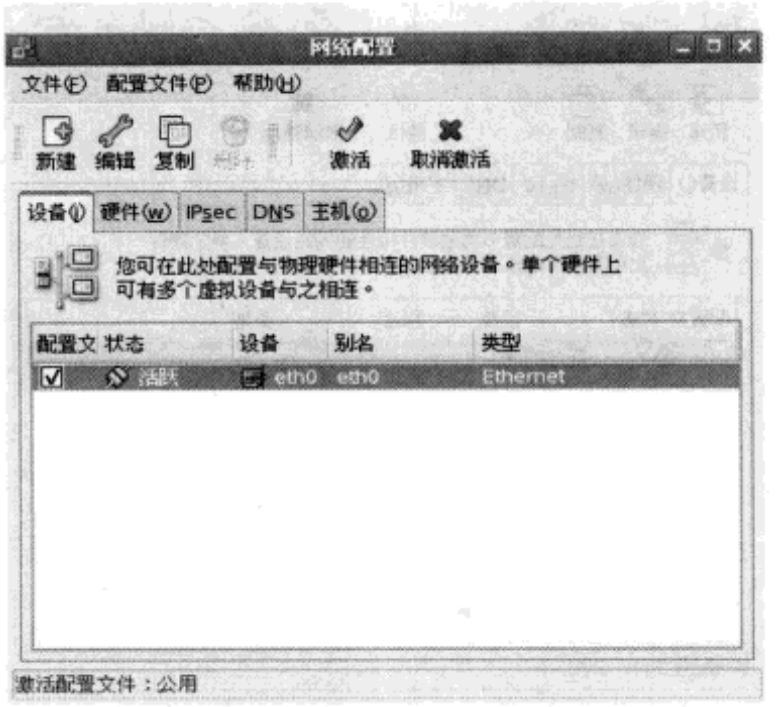


图 16-1 “网络配置”窗口

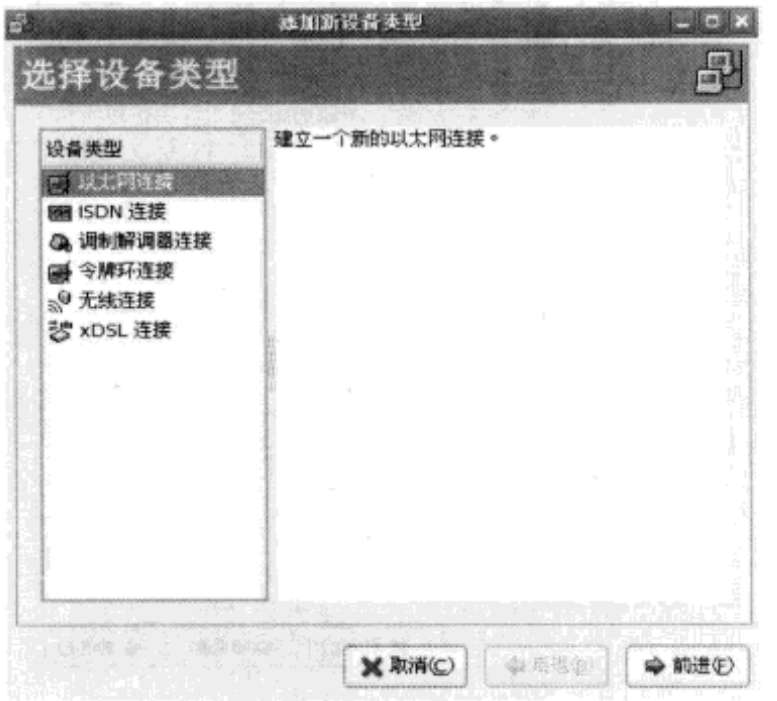


图 16-2 “选择设备类型”窗口

或者选择“其他以太网卡”项，再单击“前进”按钮即可显示“选择以太网适配器”窗口，如图 16-4 所示。在该窗口中选择网卡的制造商和型号等，如果是计算机上的第一块网卡设备，则名称为“eth0”，依此类推，第二块网卡的名称则为“eth1”。

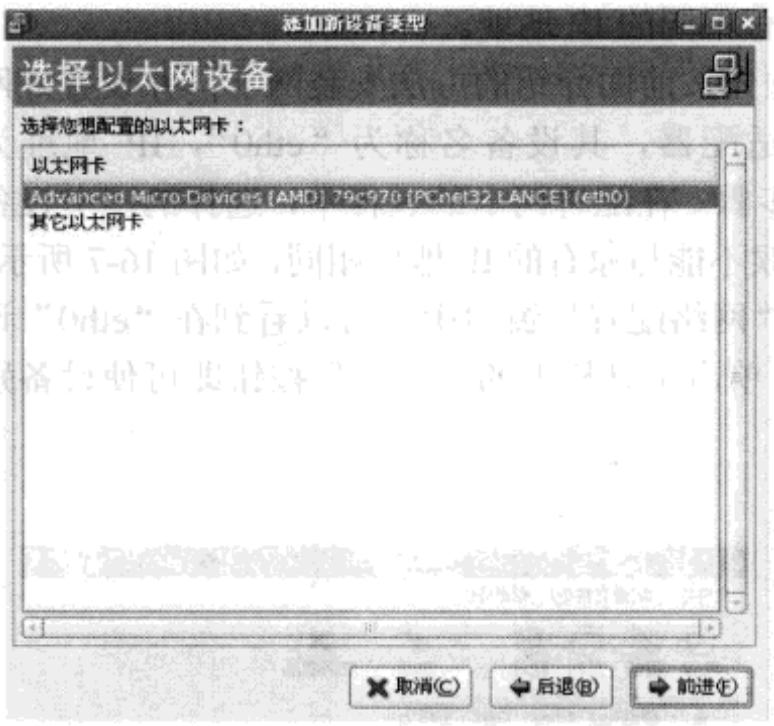


图 16-3 “选择以太网设备”窗口



图 16-4 “选择以太网适配器”窗口

在选择好以太网设备后，单击“前进”按钮，将弹出“配置网络设置”窗口，在该窗口中默认选中了“自动获取 IP 地址设置使用 dhcp”单选按钮，若保持选择此项，则本机所在的网络中需要有 DHCP 服务器；若选择“静态设置的 IP 地址”单选按钮，则需要为本机的网卡指定一个固定的 IP 地址，并设置子网掩码和网关，如图 16-5 所示。

当网卡的 IP 地址、子网掩码、默认网关都设定好后，单击“前进”按钮进入“完成创建以太网设备”窗口，在该窗口中直接单击“应用”按钮返回到“网络配置”窗口，如图 16-6 所示。从该窗口中可以看到已配置好的网络适配器，但该网络适配器还处于未激活状态，单击工具栏上的“激活”按钮就可使该网卡设备生效。

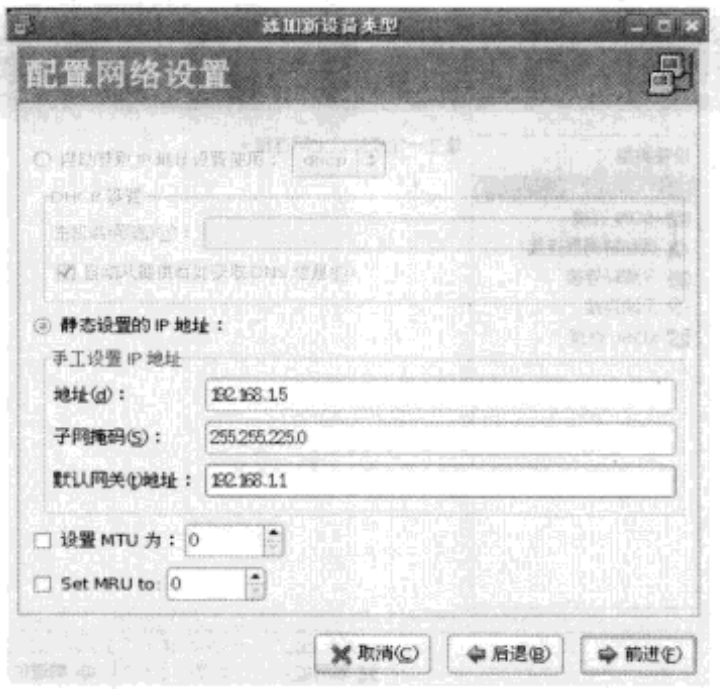


图 16-5 “配置网络设置”窗口

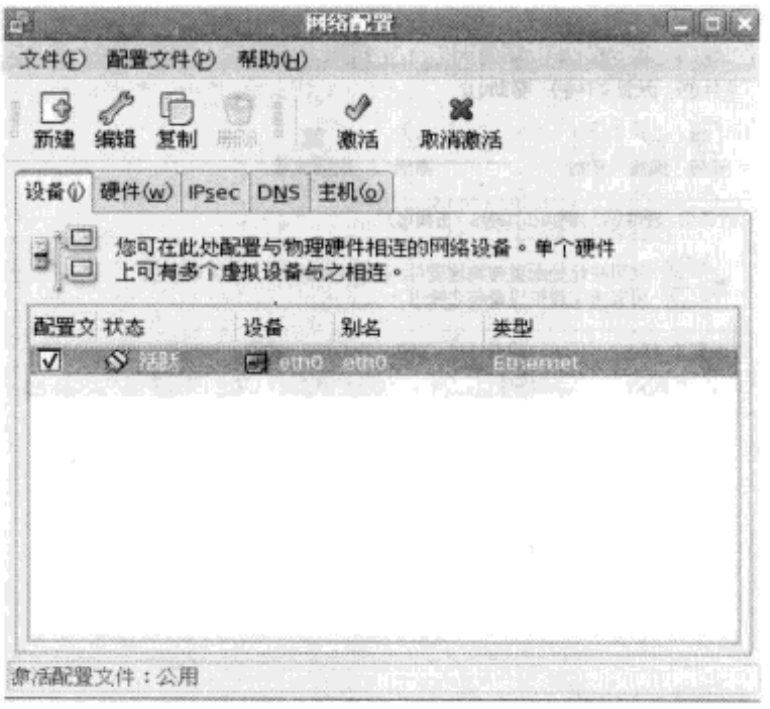


图 16-6 “网络配置”窗口

2. 为单块网卡指定多个 IP 地址

如果计算机上只安装了一块网卡，但实际上却需要为其指定多个 IP 地址，那么就需要用到 Linux 的设备别名功能了。设备别名与一个物理网络硬件（网卡适配器）相关联，与同一物理硬件相关联的设备别名也会被同时激活，并拥有不同的 IP 地址。

若需要为一块网卡指定多个 IP 地址，则必须先按前面介绍的方法安装网卡，并设置好网卡的静态 IP 地址，若本机已安装了一块网络适配器，其设备名称为“eth0”，IP 地址为“192.168.1.5”，此时需要再重复配置网卡的操作步骤。在配置网卡的过程中，选择的网卡设备与原有的网卡设备相同，只是在设定 IP 地址的时候不能与原有的 IP 地址相同，如图 16-7 所示。

当配置好网卡设备后，在如图 16-8 所示的“网络配置”窗口中，可以看到在“eth0”设备的下方多了一个网卡设备“eth2”。将其选中，单击工具栏上的“激活”按钮即可使设备别名生效。

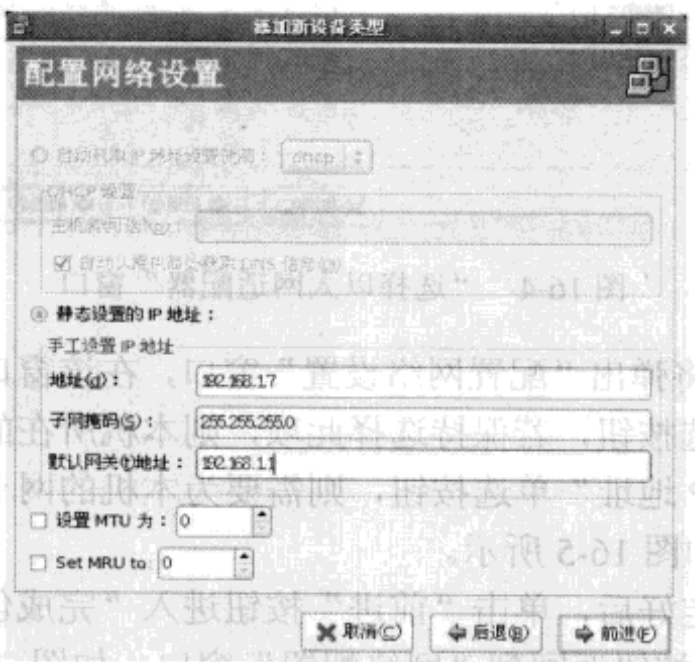


图 16-7 “配置网络设置”窗口

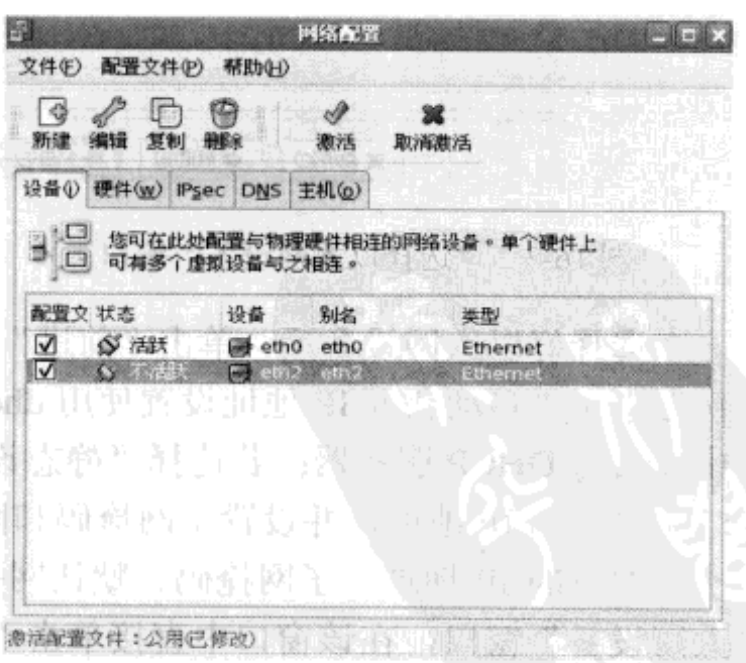


图 16-8 “网络配置”窗口

3. 修改网络配置

如果需要修改网络设备或网络设备别名，先在“网络配置”设备列表中选中需修改配置的网络设备，再单击工具栏上的“编辑”按钮，将显示如图 16-9 所示的“以太网设备”窗口，在该窗口中可修改网络设备的别名、IP 地址、子网掩码和默认网关，各项参数都修改完成后，最后单击“确定”按钮即可保存设置。

切换至“路由”选项卡，然后设置本机路由表。单击“添加”按钮将显示“添加/编辑 IP 地址”对话框，输入目标网络或目标主机的 IP 地址、子网掩码和网关，再单击“确定”按钮即可。其中，如果是设置到目标网络的路由，则子网掩码的值在 1.0.0.0~255.255.255.254 间，若设置到目标主机的路由，则子网掩码必须是 255.255.255.255。若要使路由配置生效，单击对话框中的“确定”按钮，再在“网络配置”窗口中依次单击“文件”→“保存”菜单命令，如图 16-10 所示。随后重新激活网络设备即可。

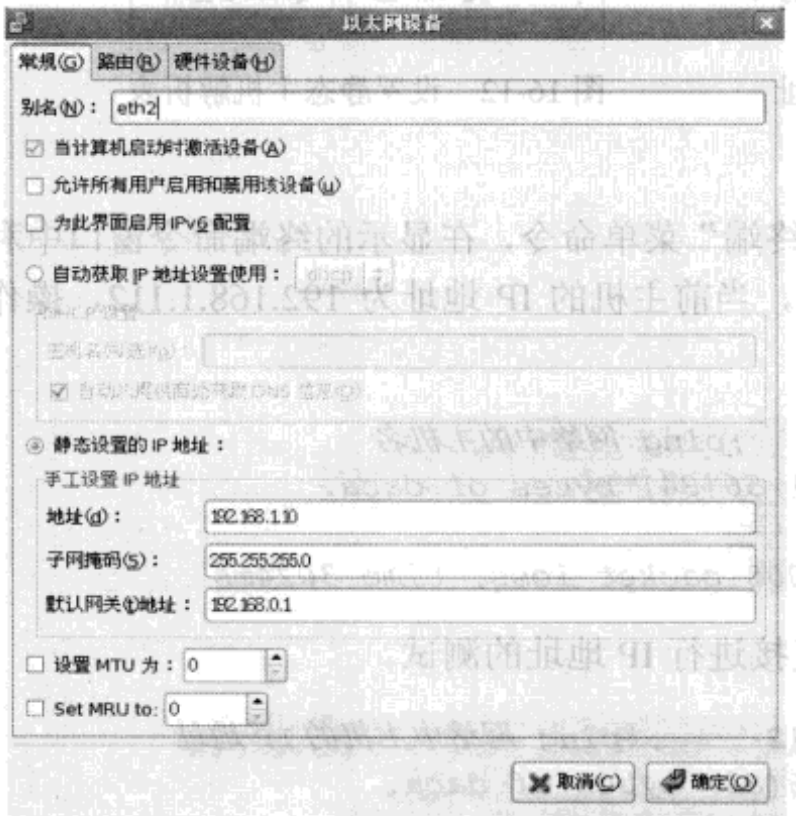


图 16-9 “以太网设备”窗口



图 16-10 设置网络设备路由

4. 设置 DNS 服务器 IP

在“网络配置”窗口，切换至“DNS”选项卡，然后设置主机名称以及 DNS 服务器的 IP 地址，如图 16-11 所示，这些信息都将保存在“/etc/resolv.conf”配置文件中。

5. 设置静态主机解析表

在“网络配置”窗口中，切换至“主机”选项卡，在该选项卡下单击“新建”按钮，在弹出的“添加/编辑主机项目”对话框中输入 IP 地址、主机名称及别名，再单击“确定”按钮，即可完成设置静态主机解析表，如图 16-12 所示。在该列表中的所有 IP 地址与主机名的对应关系将优于 DNS 服务器所提供的信息。

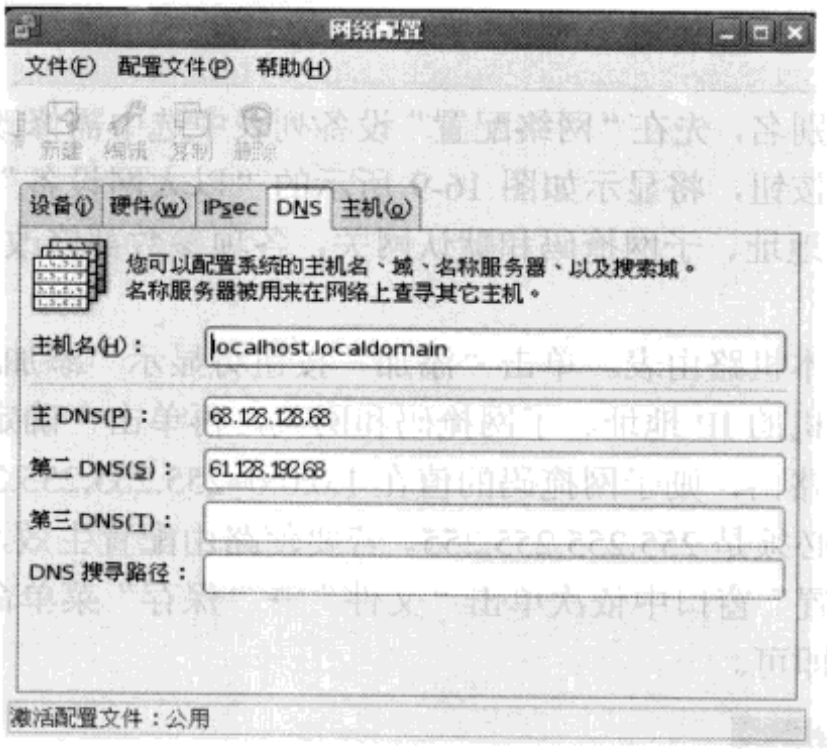


图 16-11　设置本机名称和 DNS 服务器地址

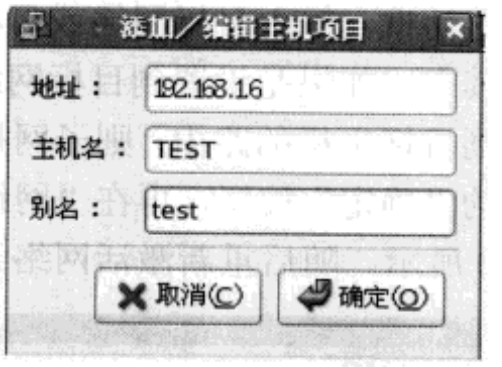


图 16-12　设置静态主机解析表

实例　利用 ping 命令测试主机连通性。

依次单击“应用程序”→“系统工具”→“终端”菜单命令，在显示的终端命令窗口中利用 ping 命令测试是否与名为“test”的主机连通，当前主机的 IP 地址为 192.168.1.112，操作指令如下。

```
[root@localhost ~]# ping TEST              ;ping 网络中的主机名
PING TEST.localdomain (192.168.1.112) 56(84) bytes of data.
--- TEST.localdomain ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3024ms
```

由以上返回结果可看出，ping 不通。然后直接进行 IP 地址的测试。

```
[root@localhost ~]# ping 192.168.1.112     ;#ping 网络中主机的 IP 地址
PING 192.168.1.112 (192.168.1.112) 56(84) bytes of data.
From 192.168.1.112 icmp_seq=1 Destination Host Unreachable
From 192.168.1.112 icmp_seq=2 Destination Host Unreachable
From 192.168.1.112 icmp_seq=3 Destination Host Unreachable
From 192.168.1.112 icmp_seq=4 Destination Host Unreachable
From 192.168.1.112 icmp_seq=5 Destination Host Unreachable
From 192.168.1.112 icmp_seq=6 Destination Host Unreachable
From 192.168.1.112 icmp_seq=7 Destination Host Unreachable
From 192.168.1.112 icmp_seq=8 Destination Host Unreachable
From 192.168.1.112 icmp_seq=9 Destination Host Unreachable
--- 192.168.1.112 ping statistics ---
11 packets transmitted, 0 received, +9 errors, 100% packet loss, time 10039ms
, pipe 3
```

由以上返回结果可以看出，本机与 IP 为 192.168.1.112 的主机能正常通信。由此可以看出，在没有 DNS 服务器进行解析，也没有在 Linux 的解析表中设置解析记录时，利用 IP 地址能互访，但不能使用主机名称进行互访。

此时可以在“网络配置”窗口中选择“主机”选项卡，再单击工具栏上的“新建”按钮，

在弹出的“添加/编辑主机”项目对话框中输入网络中主机 IP 地址和其对应的主机名称，如图 16-13 所示。

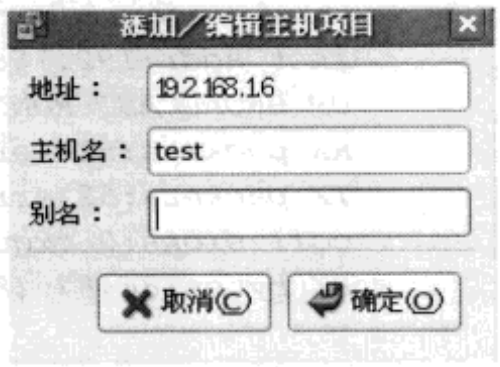


图 16-13　设置主机与 IP 地址的映射

设置好主机名与 IP 地址的映射后，单击“确定”按钮返回到“网络配置”窗口，依次单击“文件”→“保存”菜单命令，将所有的设置全部保存好，再在终端命令窗口中利用 ping 命令测试与主机“test”的连通性。

经过实验发现，设置主机名与 IP 地址的映射后，在 Linux 操作系统下就能以主机名称的方式访问网络中的其他主机了。

16.1.5 在命令模式下配置以太网

“ifconfig”命令是一个显示网络接口配置信息的工具，同时利用该命令还可以对本地网络接口进行配置。该命令一般是在系统启动期间就被调用，并用预定义的配置来设置每个接口。命令格式如下。

```
[root@localhost ~]#ifconfig [网络设备] [选项]
```

该命令各选项描述如表 16-3 所示。

表 16-3　ifconfig 命令选项

命 令 选 项	说　明
[网络设备]	指定网络设备的名称
add<IP 地址>	设置网络设备的 IP 地址
del<IP 地址>	删除网络设备的 IP 地址
up	启动指定的网络设备
down	关闭指定的网络设备
<hw<网络设备类型><硬件地址>	设置网络设备的类型与硬件地址
io_addr<I/O 地址>	设置网络设备的 I/O 地址
irq<IRQ 地址>	设置网络设备的 IRQ
media<网络媒介类型>	设置网络设备的媒介类型
mem_start<内存地址>	设置网络设备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>	设置网络设备的 MTU
netmask<子网掩码>	设置网络设备的子网掩码
tunnel<地址>	建立 IPv4 与 Ipv6 之间的隧道通信地址
-a	应用到目前系统中安装的所有网络接口
-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>	与指定地址的网络设备建立直接连线，此模式具有保密功能
-promisc	关闭或启动指定网络设备的 promiscuous 模式

1. 利用 ifconfig 命令显示所有可用网络接口

依次单击“应用程序”→“附件”→“终端”菜单命令，将显示终端命令窗口，或是在 Linux 的文本模式下执行如下操作。

```
[root@localhost ~]#ifconfig -a
eth0      Link encap:Ethernet  HWaddr 00:0C:29:33:75:72
          inet addr:192.168.1.104  Bcast:255.255.255.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:516 errors:0 dropped:0 overruns:0 frame:0
          TX packets:52 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:86487 (84.4 KiB)  TX bytes:9302 (9.0 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:2931 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2931 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:4369985 (4.1 MiB)  TX bytes:4369985 (4.1 MiB)
.....
```

对于每一个接口，其详细信息中都会包括如表 16-4 所示的内容。

表 16-4　各网络接口详细信息解释

信　息　域	说　　明
Link encap	该域指出当发送数据链路帧时，该接口将使用的链路封包协议。支持类型包括“Ethernet”（以太网）、“Loopback”或“Local Loopback”（本地）、“Point-to-Point”（点到点协议）3 种
HWaddr	这是封包协议用的数据链路地址，即 MAC 地址，以上返回结果中的 MAC 地址就是“00:0C:29:CB:5A:AD”
inet addr	接口所关联的 IP 地址，上例中 eth0 接口关联的 IP 地址就是 192.168.0.10
Bcast	该域显示网络层广播地址
Mask	该域显示子网掩码地址
RX packet	显示本接口已接收的报文
TX packet	显示本接口已发送的报文
collision	显示本接口冲突数信息
UP	活跃状态
BROADCAST	广播状态
RUNNING	运行状态
PROMISC	混杂模式
MULTICAST	网络组播模式

2. 利用 ifconfig 命令禁用或激活网络接口

用 ifconfig 可在系统运行时禁用指定的正处于活跃状态的网络接口，也可将非活跃状态的网络接口激活。当网络接口处于禁用时，将不允许任何报文通过该接口，在“ifconfig”命令中添加“down”选项即可关闭指定的网络接口。

实例　关闭本机中的“eth0”接口，命令执行结果如下。

```
[root@localhost ~]# ifconfig eth0 down
[root@localhost ~]# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:0C:29:33:75:72
          inet addr:192.168.1.104  Bcast:255.255.255.255  Mask:255.255.255.0
          BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:537 errors:0 dropped:0 overruns:0 frame:0
```

```
          TX packets:54 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:89679 (87.5 KiB)  TX bytes:9686 (9.4 KiB)
```

由以上的返回信息可以看出，eth0 网络设备中少了“UP”和“RUNNING”两个状态。若需要将网络接口打开或激话，可在“ifconfig”命令后添加“up”选项，命令执行如下。

```
[root@localhost ~]#ifconfig eth0 up
```

3. 利用 ifconfig 命令修改 IP 地址及子网掩码

当一台主机移到了不同的子网的时候，需要修改主机操作系统的 IP 地址。Linux 操作系统可以通过 ifconfig 命令来修改 IP 地址。

```
[root@localhost ~]#ifconfig eth0 192.168.1.105
```

命令执行后，eth0 接口的 IP 地址临时发生改变，当系统重新启动后，eth0 接口的 IP 地址还原为原来的状态。如果希望以后都使用修改后的 IP 配置，那么就必须修改“/etc/hosts”文件。利用“vi”命令编辑“/etc/hosts”文件的指令如下。

```
[root@localhost ~]#vi /etc/hosts
```

进入查看界面后，按键盘上的“I”键进入插入状态，再进行如下的修改。

```
# Do not remove the following line, or various programs

# that require network functionality will fail.
127.0.0.1       localhost.localdomain   localhost
192.168.1.105    TEST1    ;   #将主机的 IP 地址映射为 192.168.1.105
192.168.1.6    TEST
```

按 Esc 键退出状态，然后输入“:wq”保存并退出，若不保存文件直接退出编辑状态，则输入“:q!”，如果在“/etc/hosts”文件中已为名为“TEST1”的主机分配了 IP 地址，也可使用如下方法设置 IP 地址。

```
[root@localhost ~]#ifconfig eth0 TEST1
```

如果还需要为网络接口指定子网掩码或是广播地址，则需要在“ifconfig”命令后添加一些关键字来实现。

```
[root@localhost ~]# ifconfig eth0 192.168.1.105 netmask 255.255.255.0
broadcast 192.168.1.255
```

由以上命令可以看出，在子网掩码的前面加上了“netmask”关键字，在广播地址的前面加上了“broadcast”关键字。广播地址通常是由子网号加上主机号全部为 1 的二进制数构成的，如以上命令中的子网号是 192.168.1，而主机号是“11111111(2)”，转换为十进制数后为 255，所以广播 IP 地址为 192.168.1.255。

4. 利用 ifconfig 命令指定 IP 地址、子网掩码

在终端命令或文本框下，同样也可利用“ifconfig”命令创建设备别名，并为其指定 IP、子网掩码等。其命令格式如下。

```
[root@localhost ~]#ifconfig [物理设备名称]:[n] <IP 地址> [netmask <子网掩码>]
[broadcast <广播地址>]
[root@localhost ~]#ifconfig [物理设备名称] [add <IP 地址>]
[root@localhost ~]#ifconfig [物理设备名称] [del <IP 地址>]
```

选项中的字母 n 表示一个任意数字。例如，要以 eth0 接口为准创建一个名为“eth0:2”的设备别名，指定 IP 地址为 192.168.1.11，子网掩码为 255.255.255.0，广播地址为 192.168.3.255，命令执行如下。

```
[root@localhost ~]#ifconfig eth0:2 192.168.1.11 netmask 255.255.255.0
broadcast 192.168.3.255
[root@localhost ~]#ifconfig -a
eth0      Link encap:Ethernet  HWaddr 00:0C:29:33:75:72
          inet6 addr: fe80::20c:29ff:fe33:7572/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1095 errors:0 dropped:0 overruns:0 frame:0
          TX packets:34 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:186653 (182.2 KiB)  TX bytes:11964 (11.6 KiB)

eth0:2    Link encap:Ethernet  HWaddr 00:0C:29:33:75:72
          inet addr:192.168.1.11  Bcast:192.168.3.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
```

如果要将设备别名删除，可以执行如下指令。

```
[root@localhost ~]#ifconfig -a eth0:2 down
[root@localhost ~]#ifconfig eth0:2 del 192.168.1.11
```

16.2 动态 IP 地址分配工具

网络中的每台计算机都有自己独有的 IP 地址，当网络中的计算机过多时，为每台计算机配置静态 IP 地址的工作量就非常大，甚至还会出现 IP 地址重复的现象。为减轻管理人员的工作量、避免 IP 地址冲突，一般可以借助 DHCP 服务功能来为主机动态地分配 IP 地址。

16.2.1 获取动态 IP 地址

默认情况下，Linux 操作系统自动从本地网络获取 IP 地址，如果在同一网络中有多台 DHCP 服务器，那么用户需要指定提供 IP 地址服务的 DHCP 服务器。依次单击“应用程序”→“系统设置”→“网络”菜单命令，在弹出的“网络配置”窗口中的“设备”选项卡下选中网卡设备，再单击工具栏中的“编辑”按钮。在弹出的“以及网设备”对话框中选择“自动获取 IP 地址设置使用 dhcp”单选按钮，然后在“主机名”文本框中输入 DHCP 服务器的 IP 地址，如图 16-14 所示。

最后单击“确定”按钮返回到“网络配置”窗口，在该窗口中依次单击“文件”→“保存”菜单命令保存设置，最后重新激活网络设备即可。

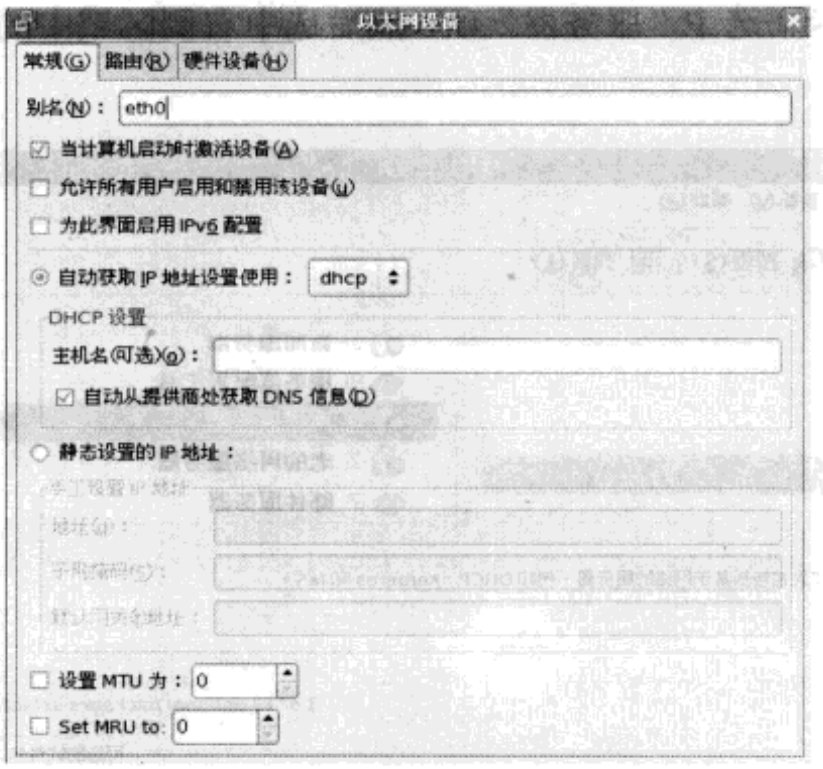


图 16-14　设置 DHCP 服务器分配的 IP 地址

16.2.2 安装 DHCP 服务

在为服务器配置DHCP服务之前，首先应为服务器指定一个固定的IP地址。利用“ifconfig”命令查看本机已设置好的IP地址及子网掩码，命令执行如下。

```
[root@localhost ~]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0C:29:33:75:72
          inet addr:192.168.1.104  Bcast:255.255.255.255  Mask:255.255.255.0
          BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:761 errors:0 dropped:0 overruns:0 frame:0
          TX packets:54 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:120209 (117.3 KiB)  TX bytes:9686 (9.4 KiB)
```

由以上的返回结果可以知道，本机已设置了一个固定的 IP 地址，这个 IP 地址就是“192.168.1.9”，子网掩码为“255.255.255.0”。

固定的IP地址指定好后，再查看系统中是否已安装了DHCP服务，输入以下命令。

```
[root@localhost ~]#rpm -qa|grep dhcp
dhcpv6-client-1.0.10-4.el5
```

由命令执行结果可以看出，系统中已安装了 DHCP 服务，也就是说本机目前可获取网络中其他DHCP服务器分配的IP。如果要使本机向其他计算机提供IP服务，则需要安装DHCP服务器端软件。在Linux操作系统中，DHCP服务可通过图形界面的“软件包管理者”窗口进行安装，也可在图形界面中直接运行软件包进行安装，如果是在终端命令或文本模式下，则需要利用rpm命令进行安装。

1. 通过“软件包管理者”窗口安装 DHCP 服务

依次单击“应用程序”→“添加/删除软件”菜单命令，弹出“软件包管理者”窗口，在

该窗口的“浏览”选项卡下选中“服务器”项，然后选中右侧区域中的“网络服务器”复选框，如图 16-15 所示。

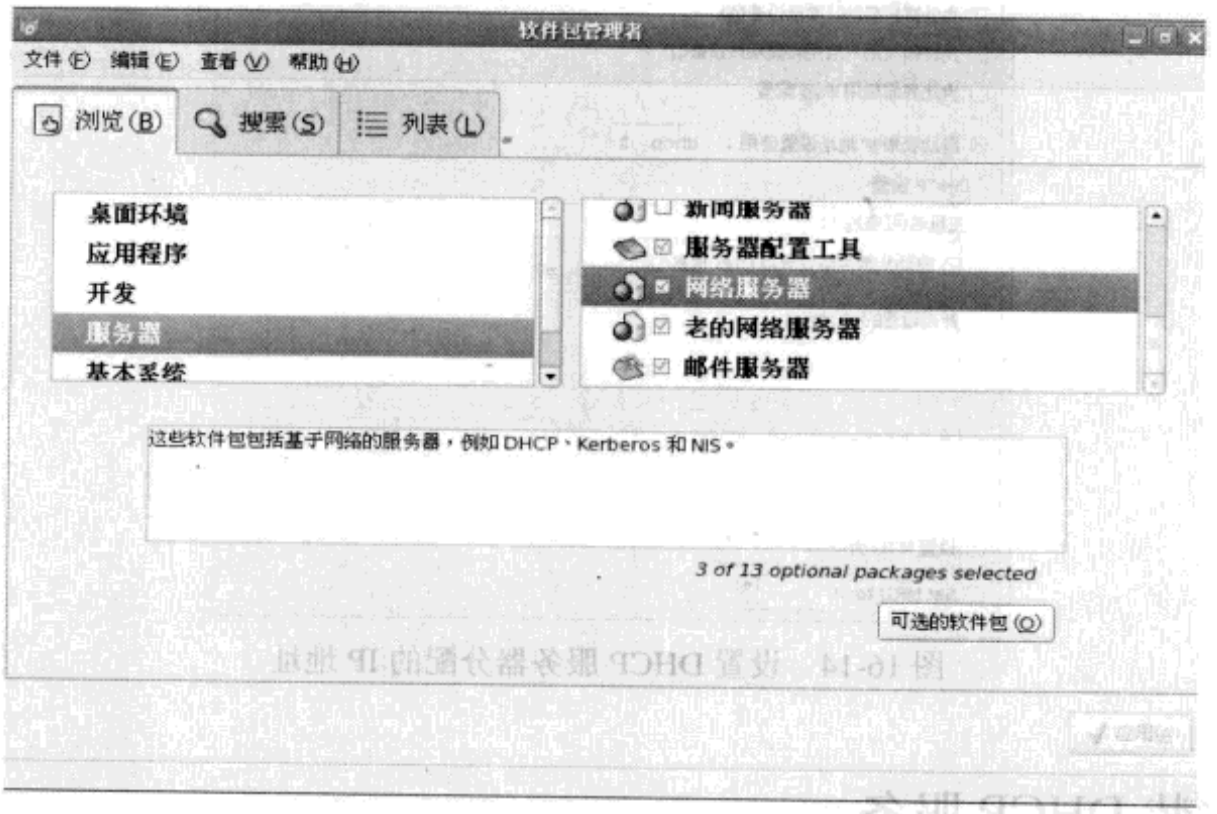


图 16-15 “软件包管理者”窗口

选中“网络服务器”复选框之后，单击下方的“可选的软件包（O）”按钮，弹出“网络服务器中的软件包”对话框，在该对话框中选中相应的 DHCP 软件包，如图 16-16 所示，最后单击“关闭”按钮。

单击“软件包管理者”窗口下方的“应用”按钮，弹出如图 16-17 所示的“软件包选择”对话框，在该对话框中单击“继续”按钮即可完成 DHCP 服务的安装。

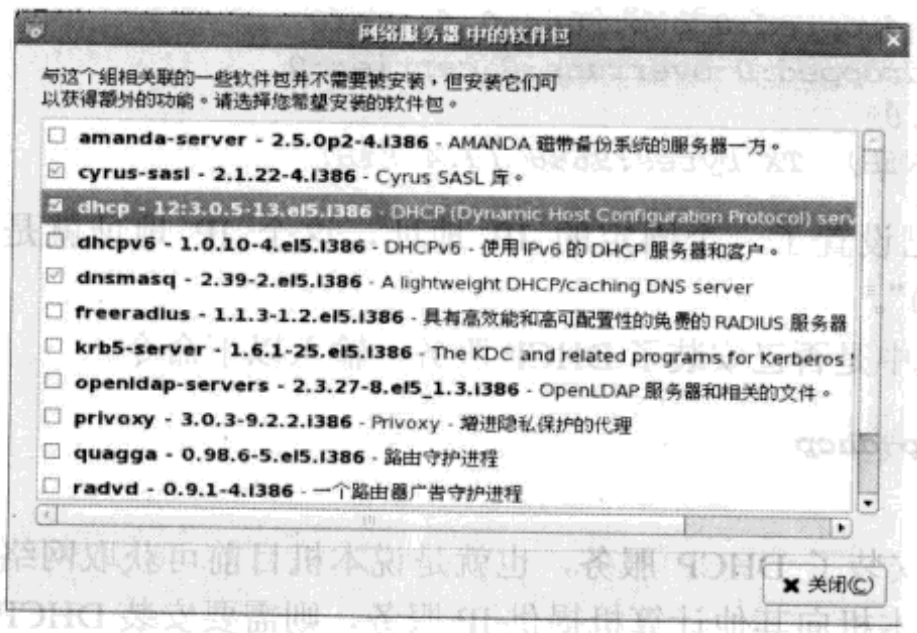


图 16-16 “网络服务器中的软件包”对话框

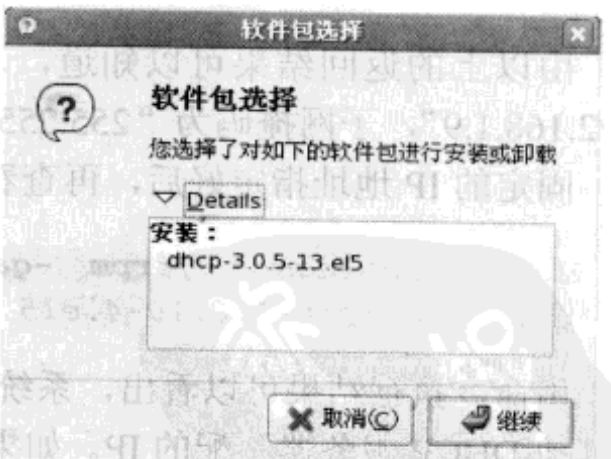


图 16-17 “软件包选择”对话框

2. 图形界面下运行软件包安装 DHCP 服务

在桌面上用鼠标右键单击“计算机”图标，在弹出的快捷菜单中选择“文件夹浏览”菜单，此时将显示“计算机-文件浏览器”窗口，使用鼠标右键单击窗口中的“CD-ROM 驱动器”图

标，在弹出的快捷菜单中选择“挂载卷”命令，如图 16-18 所示。

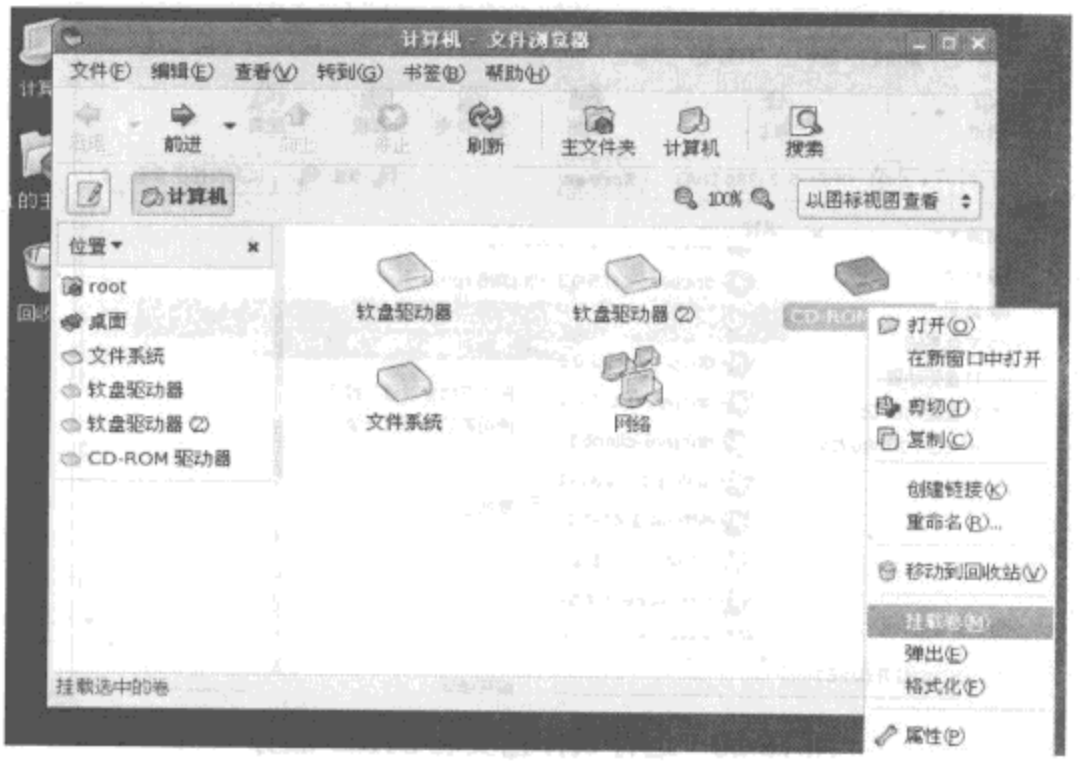


图 16-18 “计算机-文件浏览器”窗口

光盘挂载成功后，双击该图标进入光盘目录，然后进入 Sever 目录，在该目录下可找到 DHCP 服务安装包文件，如图 16-19 所示。

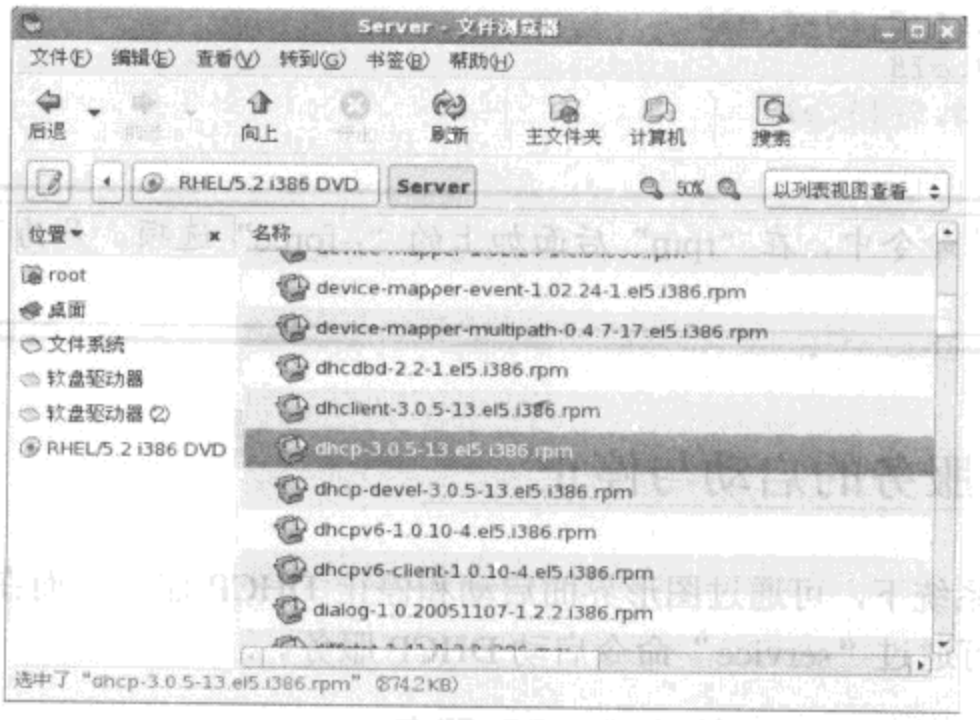


图 16-19 浏览光盘中的 DHCP 软件包

DHCP 服务的软件包共有 5 个，分别使用鼠标右键单击软件包，在弹出的快捷菜单中选择“用‘软件包管理工具’打开”项进行安装。或者在弹出的快捷菜单中选择“用‘Install Packages’打开”项，此时也将会安装被选中的软件，如图 16-20 所示。

3. 在终端命令状态下安装 DHCP 服务

如果用户处于 Linux 的文本模式下，那么还可以通过执行“rpm”命令安装 DHCP 服务。加载光盘后，进入软件包所在光盘目录，然后执行如下命令安装 DHCP 软件包。

```
[root@localhost ~]# cd /media/CD-ROM/Server/
```

```
[root@localhost Server]# rpm -ivh --force dhc*
```

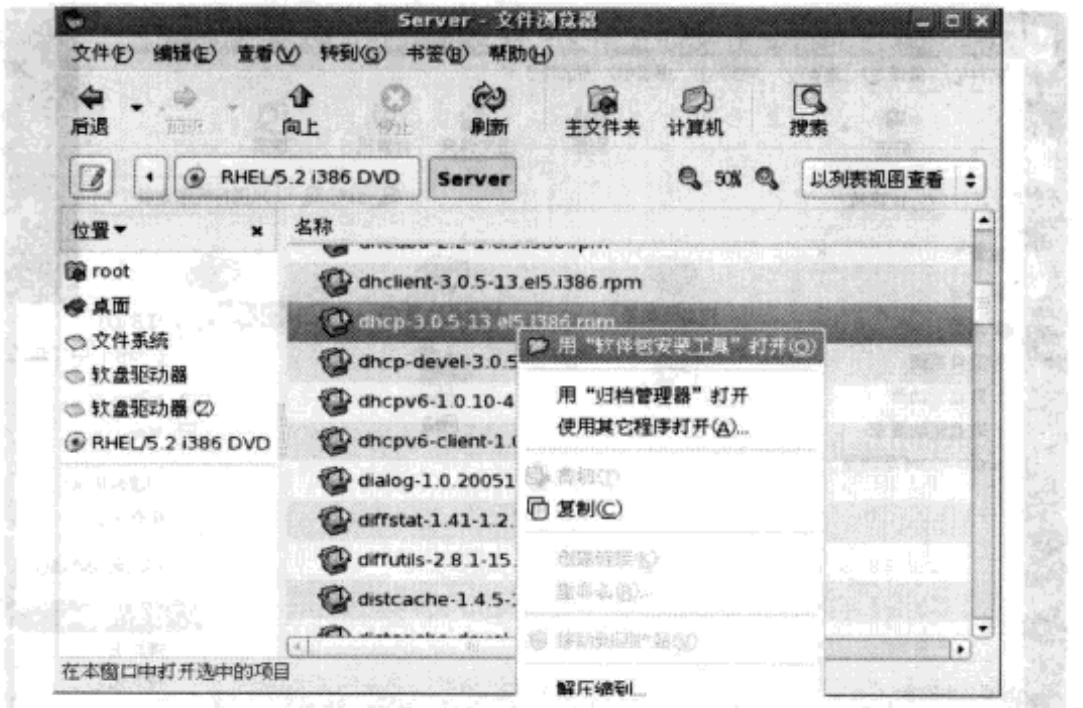


图 16-20　运行软件包安装 DHCP 服务

安装完成后，可以通过以下指令确认安装情况。

```
[root@localhost Server]# rpm -qa | grep dhcp
dhcpv6-1.0.10-4.el5
dhcpv6-client-1.0.10-4.el5
dhcp-3.0.5-13.el5
dhcp-devel-3.0.5-13.el5
```

注意　以上安装命令中，在“rpm”后面加上的“--force”选项，是为了能够使用通配符批量安装软件包。

16.2.3 DHCP 服务的启动与停止

在 Linux 操作系统下，可通过图形界面启动和停止 DHCP 服务，而在终端命令状态下或是文本模式下，则可通过“service”命令启动 DHCP 服务。

1. 通过图形界面启动和停止 DHCP 服务

依次单击“系统”→“管理”→“服务器设置”→“服务”菜单命令，弹出“服务配置”管理工具窗口，选中“dhcpd”和“dhcp6s”复选框，单击工具栏上的“开始”按钮启动 DHCP 服务，单击“重启”按钮即可将正在运行的 DHCP 服务关闭并启动，单击“停止”按钮即可使正在运行的 DHCP 服务停止，如图 16-21 所示。

注意　若在“服务配置”窗口中选中“dhcpd”复选框，则下次系统启动时，DHCP 服务也就会随着自动启动。

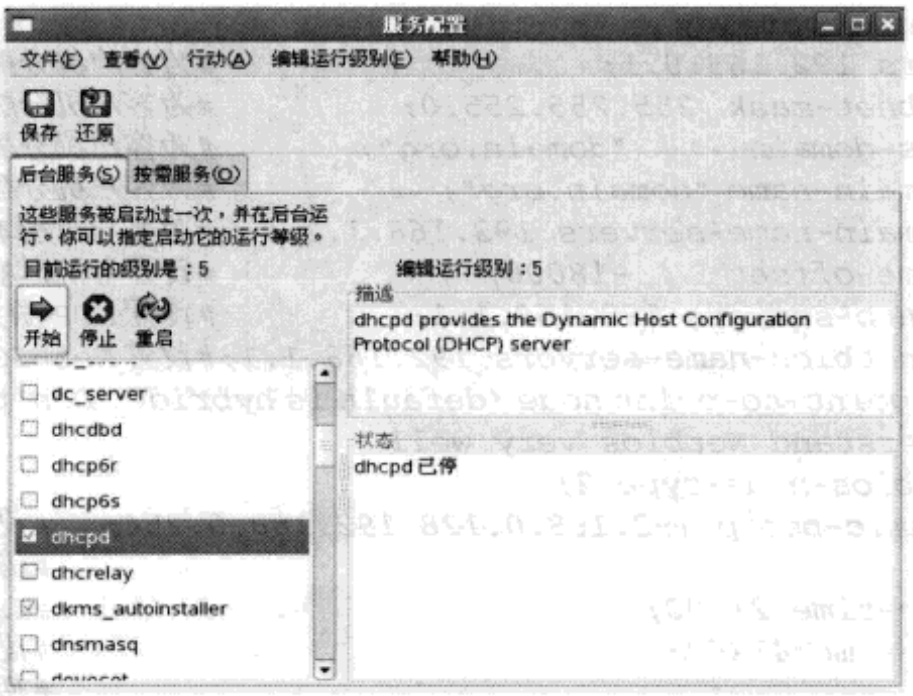


图 16-21 “服务配置”管理工具窗口

2. 通过终端命令启动和停止 DHCP 服务

如果用户处于文本模式或终端模式下，那么启动 DHCP 服务就需要通过执行命令来实现了。启动 DCHCP 服务，输入以下命令。

```
[root@localhost ~]#service dhcpd start
```

重启 DHCP 服务，输入以下命令。

```
[root@localhost ~]#service dhcpd restart
```

停止 DHCP 服务，输入以下命令。

```
[root@localhost ~]#service dhcpd stop
```

16.2.4 DHCP 服务的配置

DHCP 服务的默认配置文件是“/etc/dhcpd.conf”，且在用户没有修改其中的配置参数前，该文件中没有任何配置参数。此时可将“/usr/share/doc/dhcp-3.0.1/dhcpd.conf.sample”文件复制到“/etc”目录中，并将其重命名为“dhcpd.conf”，最后根据自己的实际需要修改配置内容。

在终端命令状态下，执行如下命令复制并更名。

```
[root@localhost ~]#cp/usr/share/doc/dhcp-3.0.1/dhcpd.conf.sample/etc/dhcpd.conf --reply=yes
```

将模板文件复制到目的位置后，在终端命令窗口中执行“gedit /etc/dhcpd.conf”命令以打开文本编辑器，在该编辑器中即可对 DHCP 服务进行配置。

```
ddns-update-style interim;
ignore client-updates;
  subnet 192.168.1.0 netmask 255.255.255.0 {      ; #可供分配 IP 地址的子网号和掩码
```

```
    # --- default gateway
option routers 192.168.0.1;                          #为客户机分配的默认网关 IP 地址
    option subnet-mask  255.255.255.0;               #为客户机分配的子网淹码
    option nis-domain       "domain.org";            #为客户机分配 DNS 服务器名，可不设
    option domain-name "domain.org";                 #为客户机分配 DNS 服务器名，可不设
    option domain-name-servers 192.168.1.1;          #为客户机根本 DNS 服务器的 IP 地址
    option time-offset     -18000;                   #设置与格林威治时间的偏差
    # option ntp-servers 192.168.1.1;                #设置 NTP 服务的 IP 地址
    # option netbios-name-servers 192.168.1.1;#设置 NetBIOS 服务器的 IP 地址
# --- Selects point-to-point node (default is hybrid). Don't change this unless
# -- you understand Netbios very well
# option netbios-node-type 2;
    range dynamic-bootp 192.168.0.128 192.168.0.254;   #设置可供分配的 IP 地址范
                                                       #围，中间用空间隔开
default-lease-time 21600;                              #IP 租用期限，单位为秒
    max-lease-time 43200;                              #设置客户机租用 IP 的最长
                                                       #期限
    # we want the nameserver to appear at a fixed address
    host ns {                                          #为网络设备指定 IP 地址
      next-server marvin.redhat.com;                   #定义服务器从引导文件装
                                                      #入的主机名（无盘工作站使用）
      hardware ethernet 12:34:56:78:AB:4D;             #标明需指定 IP 的网卡 MAC
                                                       #地址
      fixed-address 192.168.0.99;                      #为网卡分配的 IP 地址
    }
}
```

以上配置文件中的内容只是分配一个网段的 IP 地址，若需要分配多个网段的 IP 地址，则可按如下格式修改配置文件。

```
shared-network <工作组名称>{                          #设置网络工作组名称
    subnet <子网号> netmask <子网掩码> {              #第一个子网段
      range <起始 IP 地址> <结束 IP 地址>;            #指定起始 IP 地址和终止 IP 地址
      <IP 参数>;                                     #定义客户机默认子网掩码和网关等
      host <客户机主机名 1>{                         #客户机主机名称
        hardware ethernet <网卡 MAC 地址>; #客户机的 MAC 地址
          fixed-address <IP 地址>;                   #保留的 IP 地址
        <IP 参数>;                                   #客户机默认网关等
      }
      host <客户机主机名 n>{                         #客户机主机名称
        hardware ethernet <网卡 MAC 地址>; #客户机的 MAC 地址
        fixed-address <IP 地址>;                     #保留的 IP 地址
        <IP 参数>;                                   #客户机默认网关等
      }
    }
    ……
    subnet <子网号> netmask <子网掩码> {             #第 N 个子网段
      range <起始 IP 地址> <结束 IP 地址>;           #指定起始 IP 地址和终止 IP 地址
      <IP 参数>;                                     #定义客户机默认子网掩码和网关等
      host <客户机主机名>{                           #客户机主机名称
        hardware ethernet <网卡 MAC 地址>; #客户机的 MAC 地址
        fixed-address <IP 地址>;                     #保留的 IP 地址
        <IP 参数>;                                   #客户机默认网关等
      }
    }
}
```

修改好配置文件的内容后保存，重新启动 DHCP 服务，新的配置生效。

16.3 网络地址表管理工具

在 Linux 操作系统下，arp 命令可用于显示并修改网络地址表。网络地址表（也称为 ARP 缓存）中包含了与本机链接的数据链路协议到 IP 地址映射的完整列表，任何一个系统无论何时需要通过网络发送什么样的信息，都必须先知道目标系统的下层（即数据链路层）地址，如 TELNET、FTP 等服务都间接使用了 ARP 表中的数据。

arp 命令格式如下。

```
[root@localhost ~]#arp [选项] [主机名或 IP 地址] [MAC 地址]
```

arp 命令的选项描述如表 16-5 所示。

表 16-5 arp 命令选项

命 令 选 项	说 明
-a	显示指定主机的 ARP 表数据，若不带主机名或 IP 地址则显示所有主机的 ARP 表数据
-e	与“-a”选项功能相同，只是以表格的方式显示 ARP 数据
-s 或--set	创建一个新的 ARP 表记录
-d 或--delete	删除 ARP 表中指定的记录
-v 或--verbose	使用冗长模式显示 ARP 表记录
-n 或--numeric	显示 ARP 表中的记录时，以 IP 地址代替主机名称
-i 或--device <网格接口名称>	显示指定网络接口的 ARP 记录
-D 或--use-device	显示指定主机设备的 ARP 记录
-f 或--file <文件名>	从指定文件中加载数据到 ARP 缓存中

16.3.1 显示 ARP 缓存

默认情况下，显示 ARP 缓存中的数据，可通过执行命令“arp”或是“arp -v”来实现。显示结果包括当前与本机连接的所有主机的主机名、硬件类型、MAC 地址以及网卡接口等。

显示所有与本机连接的设备及 MAC 地址，指令执行如下。

```
[root@localhost ~]#arp -v
```

以 IP 地址代替主机名称显示与本机连接的设备及 MAC 地址，指令执行如下。

```
[root@localhost ~]# arp -nv
```

命令执行返回的结果分为 4 列，各列的具体说明如下。

Address 列——显示了主机的名称或 IP 地址，这些主机名称和地址都是因为应答而被记录在 ARP 缓存中的。如果该列显示了 IP 地址，则表示该 IP 地址不能被解析为一个主机名，或者是使用了“-n”参数。不能解析的情况可能是由于 DNS 服务器中无对应解析记录，或本机中没有对象的解析记录而造成的。

HWtype 列——显示了网络设备的类型，“ether”表示以太网卡。

HWaddress 列——显示了与本机连接的设备的 MAC 地址。

Flags Mask 列——显示了 ARP 表项的额外细节，ARP 标志的说明如表 16-6 所示。

表 16-6 ARP 标志说明

ARP 标志	说　明
C	已完成的 ARP 表项
M	永久或静态的 ARP 表项
P	公布该 ARP 表项

16.3.2 管理 ARP 缓存

为了保证网络的正常运行、提高设备运行速度，用户需要经常查看并调整 ARP 缓存中的记录。因为修改 ARP 缓存数据会影响到所有用户的连接。在 Linux 系统下，以 root 账号登录可以对 ARP 缓存中的数据进行任意修改，但是其他登录用户只能进行查看，不能修改 ARP 缓存数据。

1. 删除一条 ARP 缓存记录

如果更换了主机的网络设备，那么网络设备的 MAC 地址也会随着发生变化，此时为了能使网络正常通信，需要将不正确的 ARP 记录删除。

例如，删除 ARP 缓存中主机名为“TEST”的记录，执行如下命令。

```
[root@localhost ~]# arp -d TEST
```

2. 添加一条 ARP 缓存记录

如果需要与网络中某一台主机进行通信，但该主机又由于某种原因不支持 ARP，此时就需要手工在计算机中添加 ARP 记录。例如，本地网络中有一个设备名为“sb1”，其 MAC 地址为“08:00:20:82:AB:CA”，在计算机中添加 ARP 记录的指令如下。

```
[root@localhost ~]# arp -s 1b1 08:00:20:82:AB:CA
[root@localhost ~]# arp sb1
```

执行结果的 ARP 标志中如果多了一个“M”，说明是一个永久记录。该记录是通过命令行加入的，但系统重新启动后，ARP 表中的记录将会消失，不能做到真正的永久保存。

3. 从文件加载 ARP 缓存记录

“arp”命令也支持从一个文件中加载信息到 ARP 缓存中，例如，“/tmp/arp-list”文件内容如下。

```
work1 00:07:21:AB:8E:03
work2 00:08:A4:64:84:EB
work3 00:AB:AA:BE:22:01
work4 01:BC:08:54:CB:20
```

在终端命令状态下，执行如下操作可加载文件中的 ARP 记录。

```
[root@localhost ~]# arp -f /tmp/arp-list
[root@localhost ~]#arp -a
Address          HWtype          HWaddress        Flags Mask  Iface
work1            ether           00:07:21:AB:8E:03  CM  eth0
work2            ether           00:08:A4:64:84:EB  CM  eth0
work3            ether           00:AB:AA:BE:22:01  CM  eth0
work4            ether           01:BC:08:54:CB:20  CM  eth0
```

第 17 章　软件基础管理

在 Linux 操作系统中存在着许多软件包，一些软件包之间存在相互依赖的关系，这些软件包共同为 Linux 操作系统提供了非常丰富的应用。但是如何快速便捷地安装这些软件包呢？本章将学习使用几种工具来帮助用户方便、快捷地安装、运行各类应用软件。

17.1　Kickstart 简介

Red Hat 公司开发的 Kickstart 软件可以让安装 Linux 操作系统的工作自动完成。系统管理员可以使用 Kickstart 创建一个安装配置脚本文件，该文件包括对典型 Red Hat Enterprise Linux 安装中所询问问题的解答。Kickstart 安装配置脚本文件可以存放在一个服务器系统上，众多客户机在系统安装的过程中可共享读取。这种安装方法极大地方便了系统管理员，管理员可以同时使用一个 Kickstart 安装配置文件在多台客户端主机上安装 Red Hat Enterprise Linux 操作系统，因此系统管理员可以不用一直参与系统安装的工作。

17.1.1　安装 Kickstart

Kickstart 的安装非常方便，既可以从本地光盘、本地硬盘驱动器，也可以通过 NFS、FTP、HTTP 来安装。

1.　安装前的准备

在安装前，为了保证能正常使用 Kickstart，必须做好以下准备工作。

（1）　创建一个 Kickstart 文件

创建 Kickstart 文件的具体方法为：通过复制 Red Hat Enterprise Linux 安装光盘中 RH-DOCS 目录里的 sample.ks 文件，或使用 Kickstart 配置器来创建 Kickstart 配置文件。另外，Red Hat Enterprise Linux 安装程序会根据在安装 Linux 操作系统过程中的安装方案创建一个 Kickstart 配置文件，该文件被写入文件/root/anaconda-ks.cfg 中，可以使用 vi 文本编辑器编辑该文件。下面将主要讲解使用 Kickstart 配置器以图形化的方式来创建 Kickstart 配置文件的具体步骤。

注意　Kickstart 文件是一个简单的文本文件，其中包含了一个参数列表，每个参数都用关键字标明。

第 1 步，安装 Kickstart 配置器。

首先要确保 Kickstart 配置器已经安装到 Linux 操作系统中，如果系统中尚未安装 Kickstart 配置器，可执行如下指令进行安装。

```
[root@localhost Server]# rpm -ivh pykickstart-0.43.1-1.el5.noarch.rpm
warning: pykickstart-0.43.1-1.el5.noarch.rpm: Header V3 DSA signature: NOKEY,
key ID 37017186
Preparing...                ########################################### [100%]
   1:pykickstart            ########################################### [100%]
[root@localhost Server]# rpm -ivh
system-config-kickstart-2.6.19.6-1.el5.noarch.rpm
warning: system-config-kickstart-2.6.19.6-1.el5.noarch.rpm: Header V3 DSA
signature: NOKEY, key ID 37017186
Preparing...                ###########################################[100%]
1:system-config-kickstart###########################################[100%
]
```

第 2 步，设置 Kickstart 的基本配置。

成功安装 Kickstart 配置器后，需要运行 X-Window 系统。在 GNOME 桌面的菜单选项中依次单击“应用程序”→“系统工具”→“Kickstart”菜单命令，或在终端控制台键入 /usr/sbin/system-config-Kickstart 命令来启动 Kickstart 配置器，如图 17-1 所示。

```
[root@localhost ~]#/usr/sbin/system-config-Kickstart
```

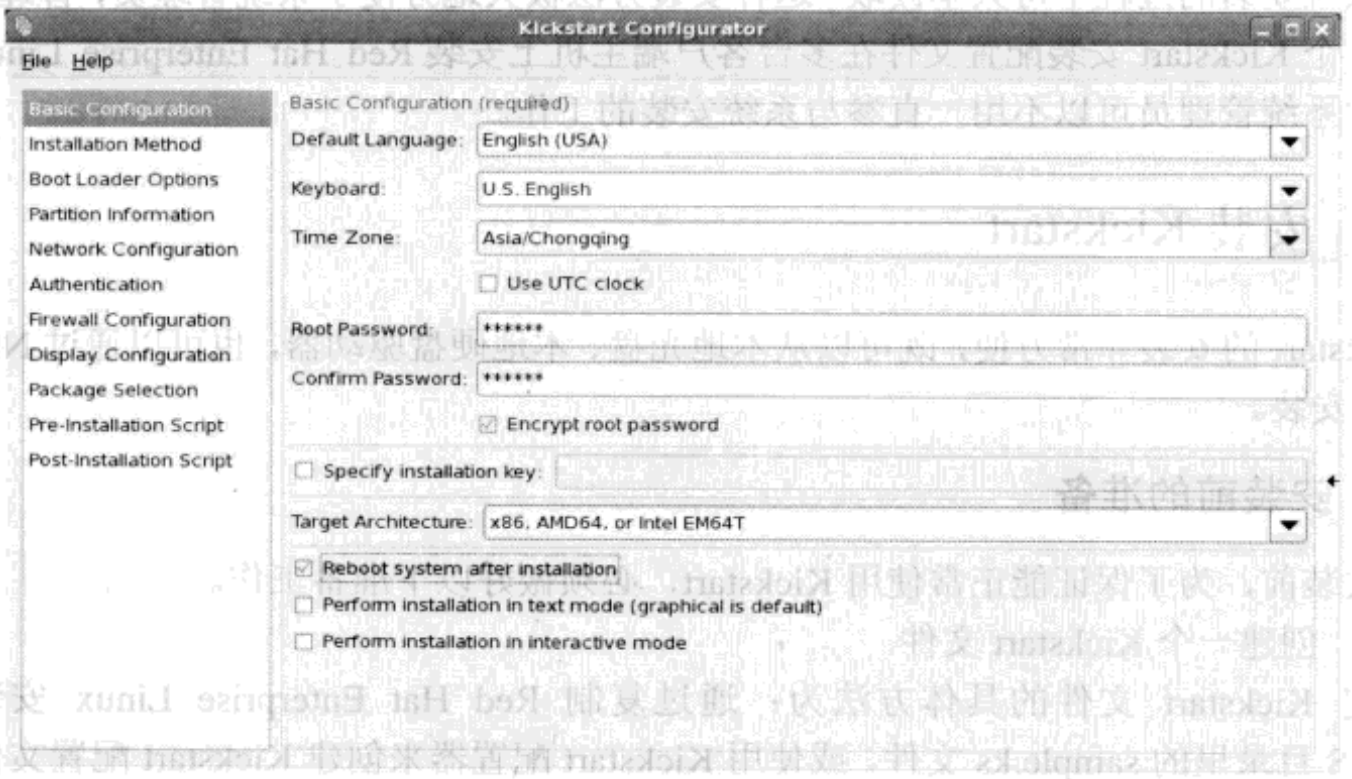


图 17-1　设置 Kickstart 的基本配置

如上图所示，首先设置 Kickstart 的基本配置，选择“Basic Configuration”标签，切换至“Basic Configuration”窗口。

在该窗口中，首先在“Default Language”下拉列表框中选择默认的语言种类，然后在“Keyboard”下拉列表框中选择系统的键盘类型，在这里可以选择“U.S.English”。在“Time Zone”下拉列表框中，选择系统使用的时区。若要配置系统使用 UTC（世界协调时），则要选中“Use UTC clock”复选框。接着还需要在“Root Password”文本框内输入 root 账号口令，如果需要将 root 账号口令以密文形式保存在文件中，需要选中“Encrypt root password”复选框。在将密码保存在文件中时，键入的普通明文口令文本就会被加密，并被写入 Kickstart 配置文件中。

注意　如果选中“Reboot system after installation”复选框，就表示系统安装结束后自动重新引导系统。

如果要采用文本安装模式，则需要选中“Perform installation in text mode (graphical is default)”复选框；如果在自动安装过程中使用交互式安装模式，则需要选中“Perform installation in interactive mode”复选框。

第 3 步，设置 Kickstart 安装方法。

进入 Kickstart 安装方法设置窗口，如图 17-2 所示。首先选择“Installation Method”标签，然后在“Installation Method”窗口中配置 Kickstart 安装方法。

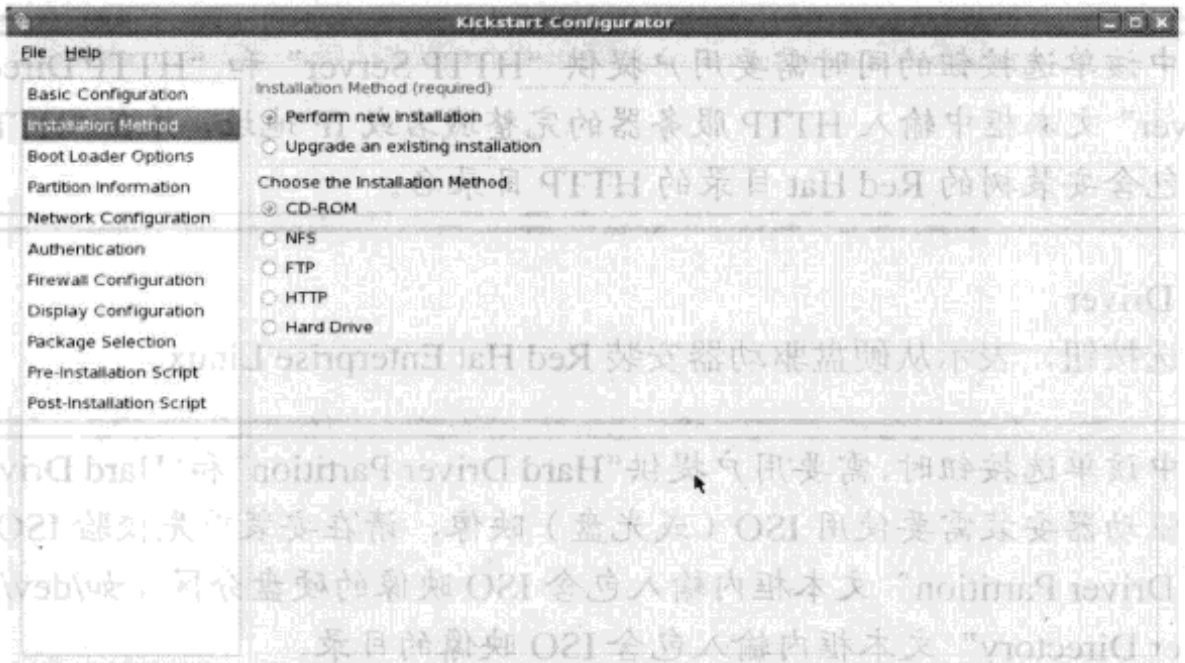


图 17-2　Kickstart 安装方法设置窗口

其中，如果选中“Perform new Installation”单选按钮，表示 Kickstart 安装采用全新安装模式；而选中“Upgrade an existing Installation”单选按钮，则表示 Kickstart 安装采用升级安装模式。

注意　在选择升级安装模式时，“Partition Information”和“Authentication”标签选项将会被禁用，这是因为 Kickstart 不支持这两种升级安装模式。

另外，在 Kickstart 安装方法设置时，还需选择 Kickstart 安装类型。可选的类型如下。

- CD-ROM

选中该单选按钮，表示从 Red Hat Enterprise Linux 光盘安装系统。

- NFS

选中该单选按钮，表示从 NFS 共享目录中安装 Red Hat Enterprise Linux 操作系统。

注意　选中该单选按钮的同时需要用户提供“NFS Server”和“NFS Directory”信息。在“NFS Server”文本框中输入 NFS 服务器的完整域名或 IP 地址，并在“NFS Directory”文本框中输入包含安装树的 Red Hat 目录名。

- FTP

选中该单选按钮，表示从 FTP 服务器中安装 Red Hat Enterprise Linux 操作系统。

注意 选中该单选按钮时，需要用户提供“FTP Server”和“FTP Directory”信息。在“FTP Server”文本框中输入 FTP 服务器的完整域名或 IP 地址，并在“FTP Directory”文本框中输入包含 Red Hat 目录的 FTP 目录名。另外，如果访问 FTP 服务器要使用用户名和口令，还需要在这里设置这两个参数。

- HTTP

选中该单选按钮，表示从 HTTP 服务器安装 Red Hat Enterprise Linux 操作系统。

注意 选中该单选按钮的同时需要用户提供“HTTP Server”和“HTTP Directory”信息。在“HTTP Server”文本框中输入 HTTP 服务器的完整域名或 IP 地址，并在“HTTP Directory”文本框中输入包含安装树的 Red Hat 目录的 HTTP 目录名。

- Hard Driver

选中该单选按钮，表示从硬盘驱动器安装 Red Hat Enterprise Linux。

注意 选中该单选按钮时，需要用户提供“Hard Driver Partition”和“Hard Driver Directory”信息。从硬盘驱动器安装需要使用 ISO（或光盘）映像，请在安装前先校验 ISO 映像的完整性。在“Hard Driver Partition”文本框内输入包含 ISO 映像的硬盘分区（如/dev/hda1），然后在“Hard Driver Directory”文本框内输入包含 ISO 映像的目录。

第 4 步，对 Kickstart 的引导装载选项进行设置。

在如图 17-3 所示的窗口中，选择“Boot Loader Options”标签，切换至“Boot Loader Options”窗口，在此窗口中可以对 Kickstart 的引导装载选项进行设置。

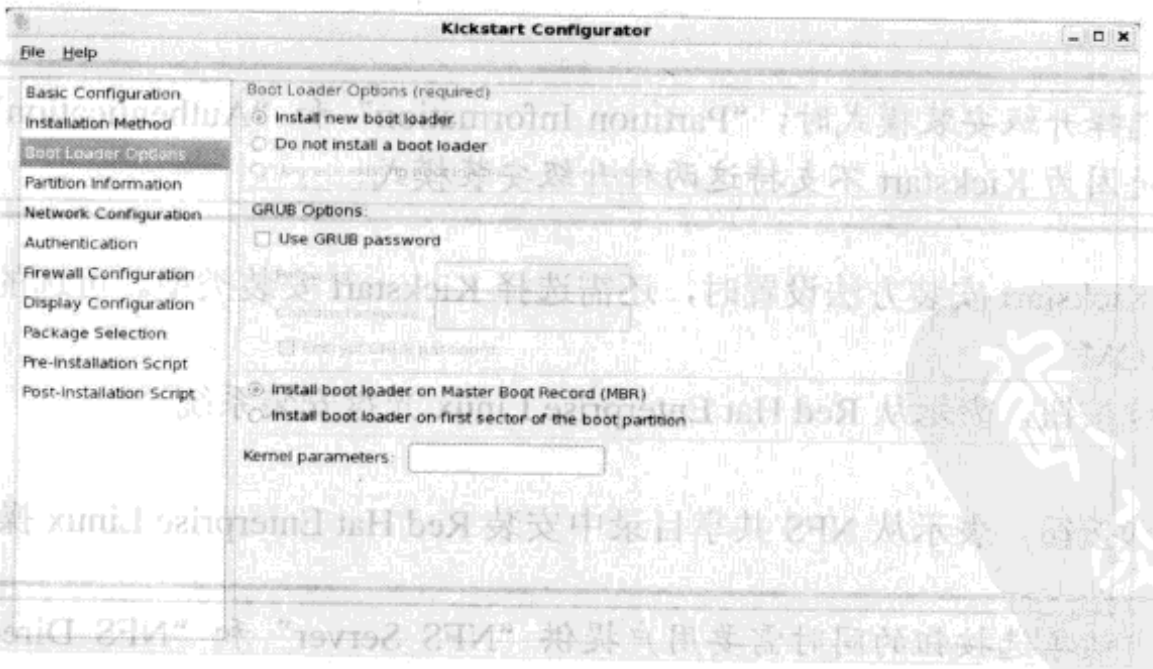


图 17-3 Kickstart 引导装载选项设置窗口

如果 Kickstart 以 GRUB 作为引导装载程序，则需要选中“Install new boot loader”单选按

钮。此时，需要指定引导装载程序 GRUB 的安装位置，同时将选中的引导装载程序安装到 MBR 上。也可以将其安装在/boot 分区的第一个扇区上，只是需要通过引导装载程序来引导 Red Hat Enterprise Linux 系统。

如果管理员需要提高系统安装的安全性，则可以配置一个 GRUB 口令。此时需要在“Use GRUB password”文本框内输入 GRUB 口令，若要在文件中为口令加密，则需要选中“Encrypt GRUB password”复选框。当文件被存盘后，键入的口令明文将会被加密并写入 Kickstart 文件。

如果需要在系统引导时向内核传递特殊参数，可以设置“Kernel parameter”文本框内的信息。

注意 Kickstart 可以不安装引导装载程序，此时则需要选中“Do not install a boot loader”单选按钮。不过选择这个选项之后，必须要确保系统中已经创建了引导盘或其他引导 Red Hat Enterprise Linux 系统的方法（如第三方引导装载程序）。

如果在“Installation method”对话框中选中了“upgrade an existing installation”单选按钮，则表示升级现存的引导装载程序配置，并保留其中原有的项目。

第 5 步，设置分区信息。

选择“Partition Information”标签，切换至“Partition Information”窗口，进行 Kickstart 安装过程中分区信息的设置。

在如图 17-4 所示的分区信息窗口中，管理员可以根据需要决定是否清除主引导记录（MBR），也可以删除所有分区、删除所有现存的 Linux 分区或保留现存分区。如果需要进行创建分区的操作，则需要单击“Add”按钮，弹出“Partition Options”对话框，如图 17-5 所示。在这里可以为新分区配置挂载点、文件系统类型和分区大小。

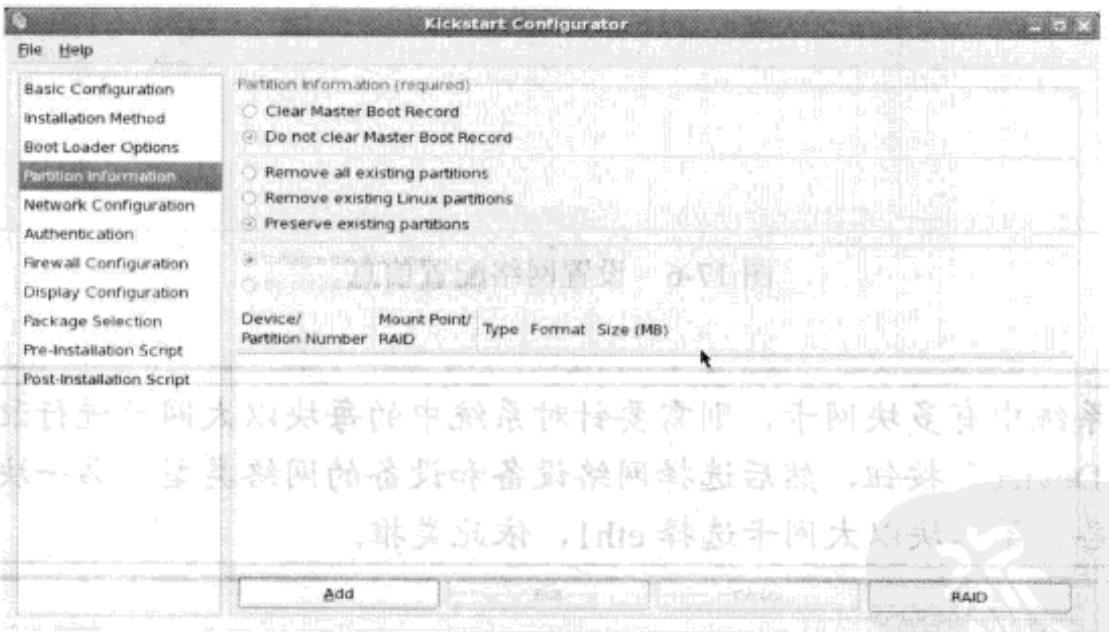


图 17-4 分区信息窗口

“Additional Size Options”选项组可以指定分区使用的固定大小、指定大小以及使用驱动器上的全部剩余空间。如果把文件系统类型指定为交换区（Swap），则可以选择让安装程序使用建议值而不是指定大小来创建交换分区。

选中“Force to be a primary partition（asprimary）”复选框，表示强制分区被创建为主分区。

选中“Make partition on specific drive（ondisk）”复选框，则表示在指定硬盘驱动器上创

建分区。例如，要在第一个 IDE 硬盘（/dev/hda）上创建分区，把 hda 指定为驱动器。

选中“Use existing partition（onpart）”复选框，表示使用现存分区。例如，要在第一个 IDE 硬盘上的第一个分区（/dev/hda1）上创建分区，把 hda1 指定为分区。

选择“Format partition”复选框，表示把分区格式化为选定的文件系统类型。

若要编辑某现存分区，从列表中选择它，然后单击“Edit”按钮，会弹出“Partition Options”对话框。修改分区选项，然后单击“OK”按钮即可。

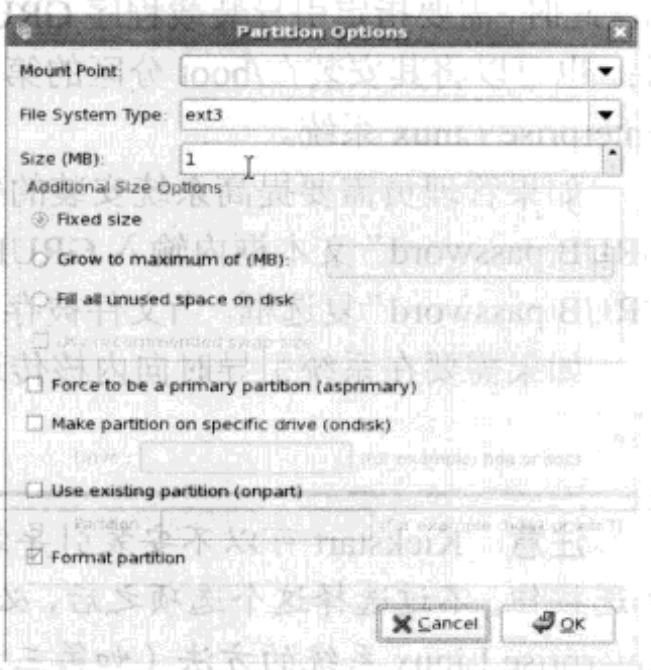


图 17-5 “Partition Options”对话框

第 6 步，设置网络配置信息。

使用 Kickstart 安装系统时如果选择了网络安装的方法（NFS、FTP 或 HTTP），则需要设置如图 17-6 所示的“Network Configuration”窗口中的信息。

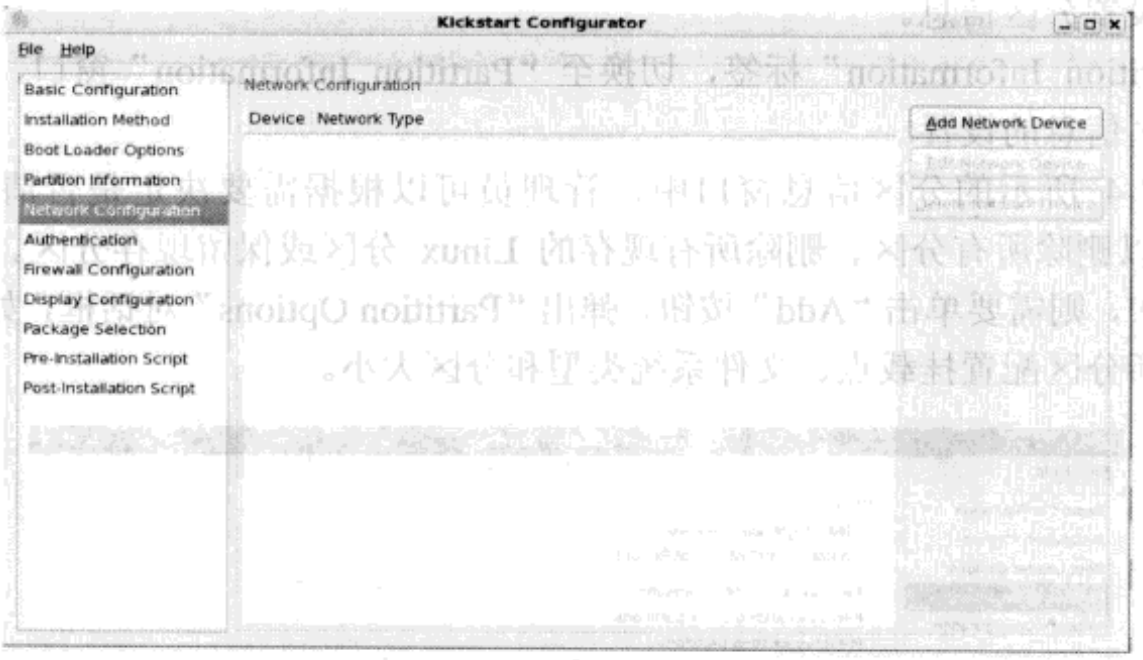


图 17-6 设置网络配置信息

注意 如果系统中有多块网卡，则需要针对系统中的每块以太网卡进行配置。首先单击“Add Network Device”按钮，然后选择网络设备和设备的网络类型。第一块以太网卡选择 eth0 作为网络设备，第二块以太网卡选择 eth1，依此类推。

第 7 步，设置验证。

选择“Authentication”标签，切换至“Authentication Confrguration”窗口。设置待安装系统的验证配置信息，如图 17-7 所示。在这里建议选择屏蔽用户口令，并使用 MD5 算法加密用户口令。

“Authentication Configuration”窗口允许我们配置 LDAP、NIS、SMB、Kerberos 5、Hesiod 以及 Name Switch Cache 验证方法。

默认情况下，这些方法不被启用，用户可以根据需要选定需要启用的系统验证方法。单击

要启用的验证方法标签，然后选中“Enable”复选框，同时，输入用于该验证方法的适当信息。

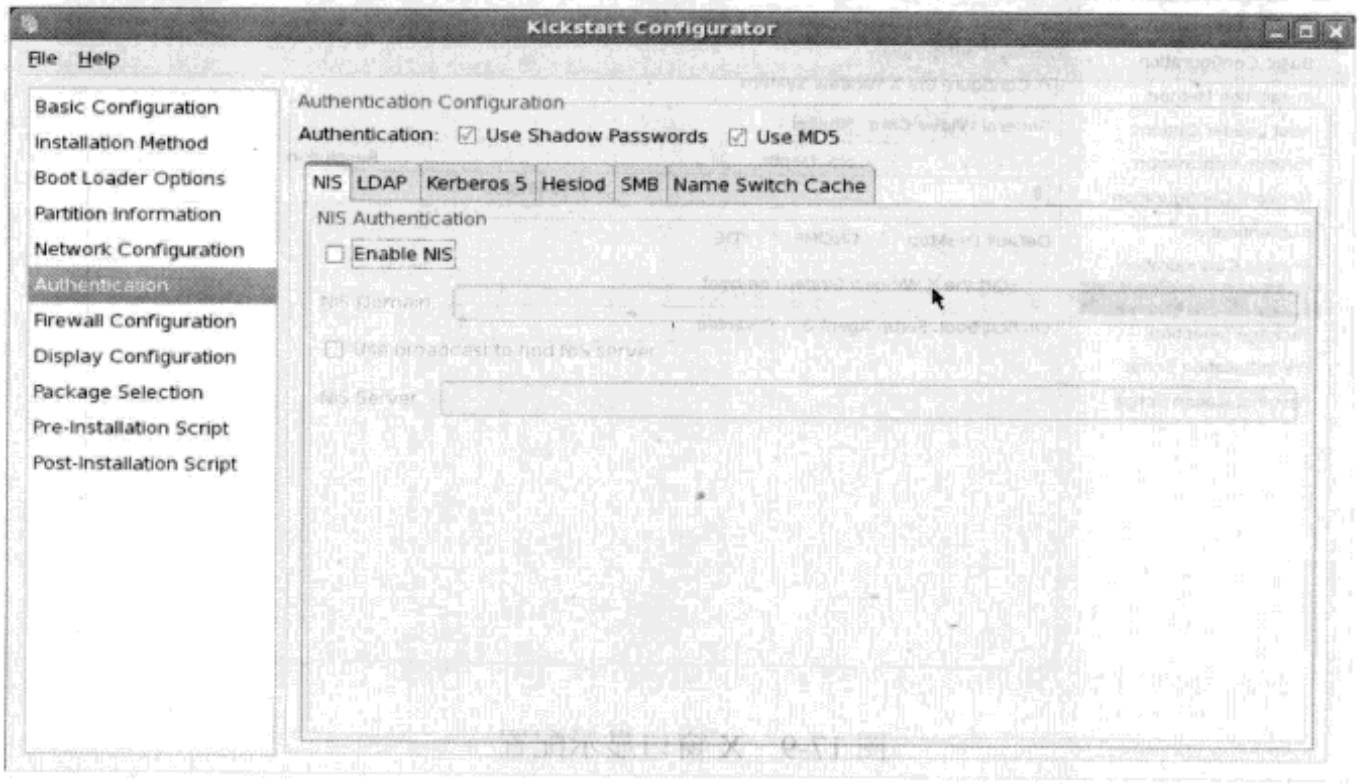


图 17-7　设置验证

第 8 步，配置防火墙。

选择“Firewall Configuration”标签，切换至“Firewall Configuration”窗口，设置待安装系统的防火墙配置信息，这其中包括了防火墙级别设置和防火墙规则设置。防火墙级别分为“高级”、“中级”和“已禁用”3 个级别。防火墙规则可以根据需要定制，如图 17-8 所示。

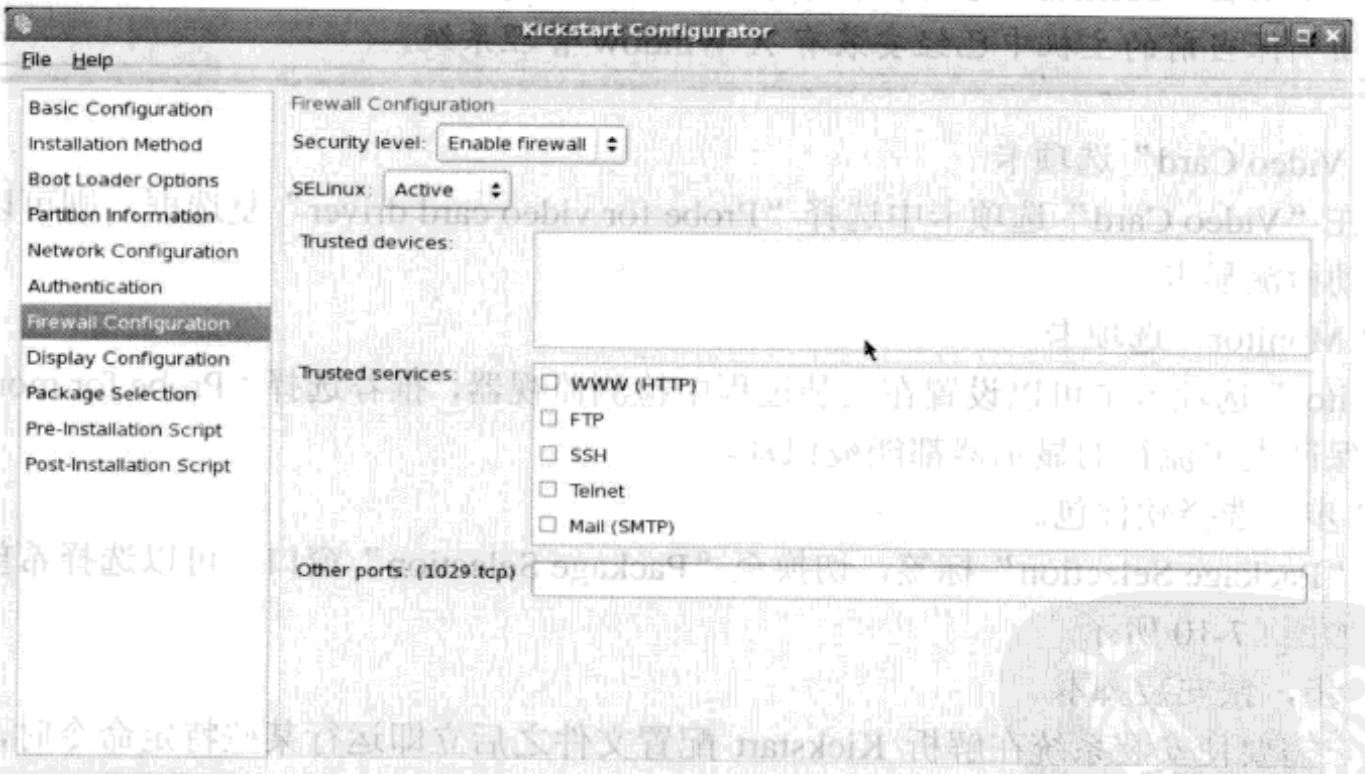


图 17-8　配置防火墙

第 9 步，X 窗口显示配置。

选择“Display Configuration”标签，然后切换至“Display Configuration”窗口，在其中设置安装系统 X 窗口显示的配置信息，如图 17-9 所示。

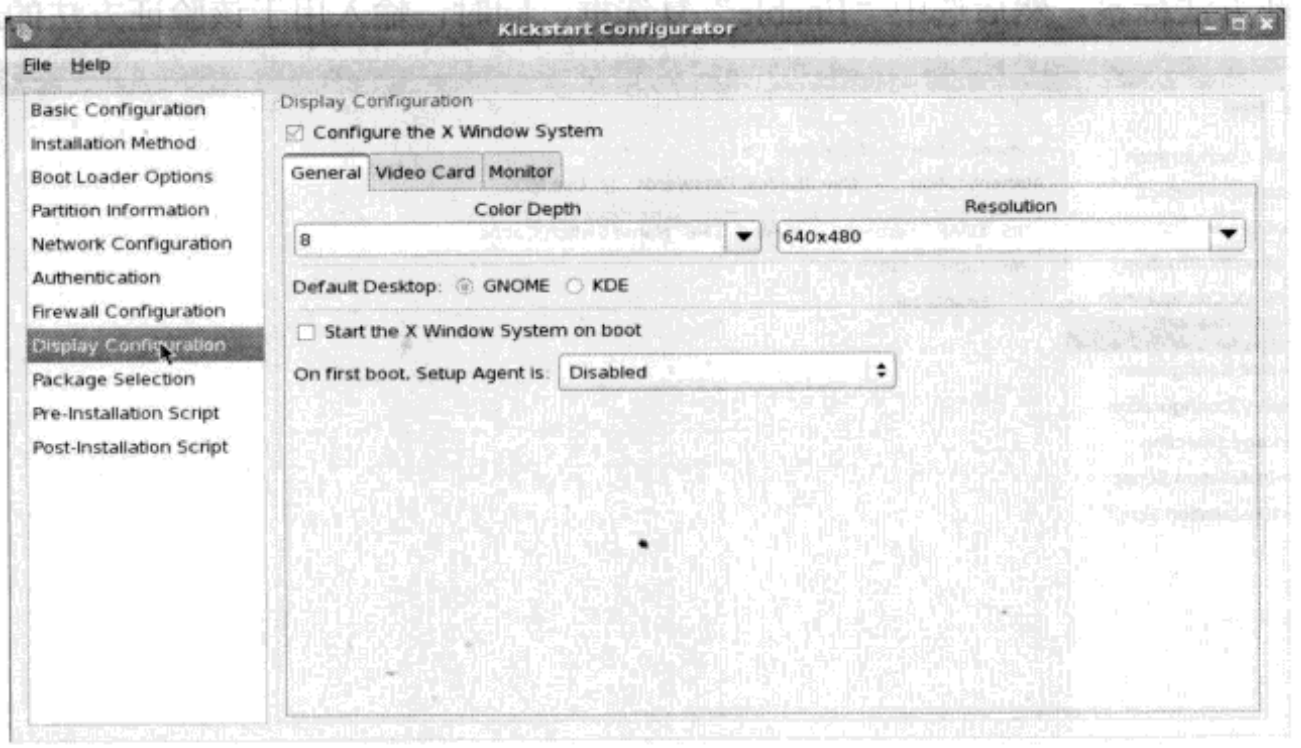


图 17-9　X 窗口显示配置

在该界面中有 3 个选项卡。

- “General”选项卡

“General”选项卡中可以选择色彩深度和分辨率，同时可以指定与主机的显卡和显示器兼容的色彩深度及分辨率。

注意　如果在“General”选项卡中选择 GNOME 或 KDE 中的任意一个作为默认的桌面，那么就必须确保当前的主机中已经安装有 X-Window 管理系统。

- “Video Card”选项卡

如果在“Video Card”选项卡中选择“Probe for video card driver”复选框，则可以在安装过程中自动检测显卡。

- “Monitor”选项卡

“Monitor”选项卡中可以设置在安装过程中检测监视器，推荐选择“Probe for monitor”复选框，以保证大多流行的显示器都能被识别。

第 10 步，选择软件包。

选择“Package Selection”标签，切换至“Package Selection”窗口，可以选择希望安装的软件包，如图 17-10 所示。

第 11 步，预安装脚本。

当用户需要使安装系统在解析 Kickstart 配置文件之后立即运行某些特定命令时，就需要配置预安装脚本。这部分的配置信息必须位于 Kickstart 配置文件的结尾处（在命令之后），而且必须以%pre 命令开头。

注意　用户在%pre 部分可以访问网络；但名称服务（name service）在此时还未被配置，因此只有通过 IP 地址访问网络才能奏效。

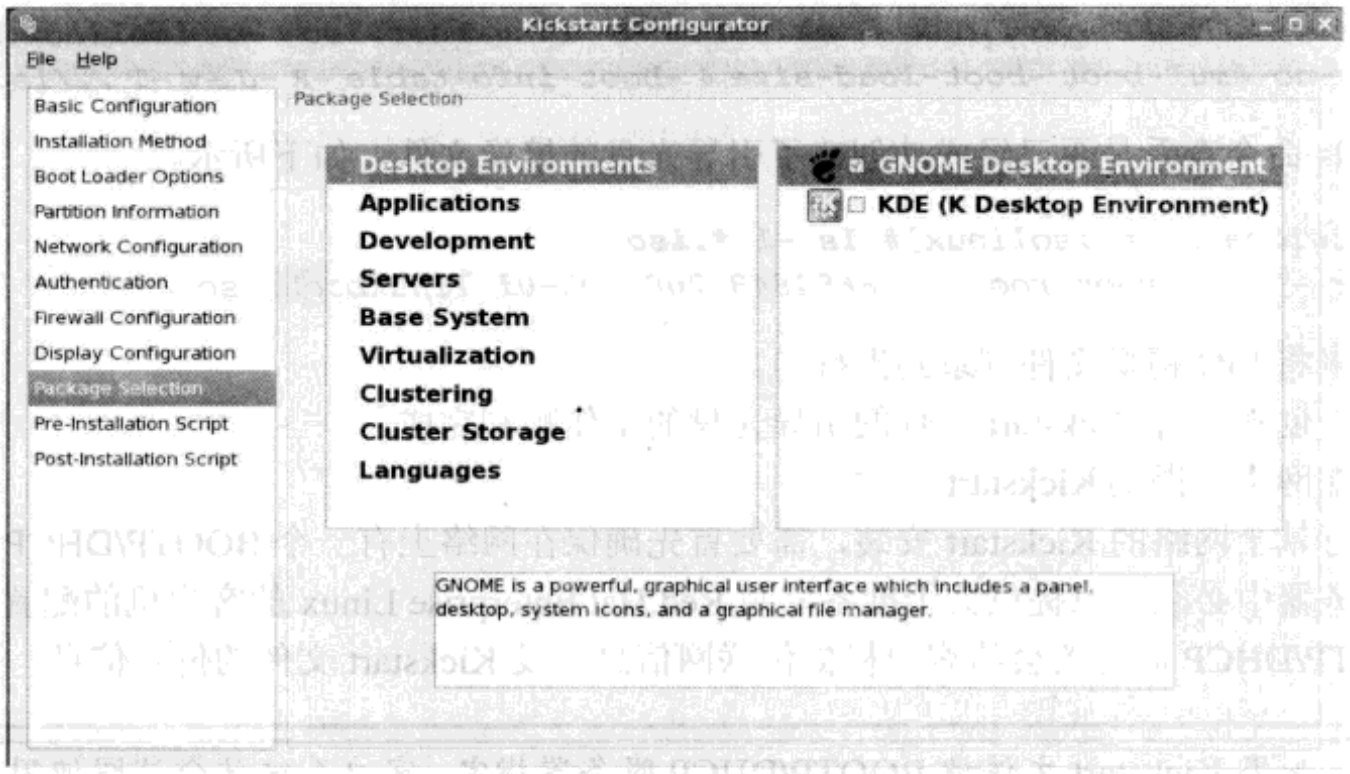


图 17-10　选择软件包

第 12 步，安装脚本后的配置。

当用户需要在系统安装完毕后立即运行某些特定命令时，需要配置安装后的脚本。这部分的配置信息必须位于 Kickstart 配置文件的结尾处，而且必须以%post 命令开头。这对于安装额外软件包或配置额外名称服务器等任务非常有帮助。

（2） 创建 Kickstart 引导盘

下面重点介绍如何创建带有 Kickstart 文件的引导光盘，以及如何在网络上提供 Kickstart 文件的相关知识。

● 创建带有 Kickstart 文件的引导光盘

创建一个带有 Kickstart 文件的引导光盘可以通过以下步骤完成。

第 1 步，复制引导文件和已创建的 Kickstart 配置文件。

将 Red Hat Enterprise Linux 5.3 安装光盘中的 isolinux/目录复制到当前用户主目录下的 temp 目录中，并将 Kickstart 配置文件命名为 ks.cfg，复制到当前用户主目录下的 temp 目录中，执行以下命令。

```
[root@localhost~]# mkdir temp
[root@localhost~]# cp -r /media/RHEL_5.3\ I386\ DVD/isolinux/ /root/temp/
[root@localhost~]# cp ks.cfg /root/temp/
```

第 2 步，设置文件权限。

切换到刚创建的/root/temp 目录中，然后执行如下指令设置用户对所有文件应具备的正确权限。

```
[root@localhost~]# cd temp/
[root@localhos ttemp]# chmod u+w isolinux/*
```

第 3 步，创建 ISO 镜像文件。

执行以下命令来创建 ISO 映像文件。

```
[root@localhost~]# cd temp/isolinux/
```

```
[root@localhost isolinux]# mkisofs -o linuxboot.iso -b isolinux.bin -c
boot.cat -no-emul-boot -boot-load-size 4 -boot-info-table -R -J -v -T ../isolinux/
```

执行 ls 命令查看是否已经成功创建了引导光盘的镜像文件，如下所示。

```
[root@localhost isolinux]# ls -l *.iso
-rw-r--r-- 1 root root    8654848 2009-05-01 linuxboot.iso
```

接下来将 ISO 镜像文件刻录到光盘中。

至此，创建带有 Kickstart 文件的引导光盘的工作顺利完成。

- 在网络上提供 Kickstart 文件

要执行基于网络的 Kickstart 安装，需要首先确保在网络上有一个 BOOTP/DHCP 服务器，并且该服务器中必须可以提供关于准备安装 Red Hat Enterprise Linux 的客户机的配置信息。此时，BOOTP/DHCP 服务器会给客户机提供联网信息以及 Kickstart 文件的位置信息。

注意 如果 Kickstart 文件被 BOOTP/DHCP 服务器指定，客户系统就会试图使用 NFS 来挂载该文件的路径，并把指定文件复制到客户机上，作为 Kickstart 配置文件，不过需要依照当前正在使用的 BOOTP/DHCP 服务器而定。

下面是从 Red Hat Enterprise Linux 安装光盘中附带的 DHCP 服务器的 dhcpd.conf 文件中所摘录的一行配置信息。

```
filename "/usr/new-machine/Kickstart/";
next-server blarg.redhat.com;
```

注意 应该把 filename 后面的值替换为 Kickstart 文件的名称（或 Kickstart 文件所在的目录），把 next-server 后面的值替换成 NFS 服务器的名称。

如果 BOOTP/DHCP 服务器返回的文件以斜线（“/”）结束，表示此时返回的是文件所在的路径。在这种情况下，客户机系统会使用 NFS 来挂载该路径，然后搜索某一指定文件。

例如，客户机搜索的文件名如下。

```
<ip-addr>-Kickstart
```

文件名的<ip-addr>部分应该被点式的客户机 IP 地址替换。例如，IP 地址为 192.168.1.102 的计算机的文件名应为 192.168.1.102-Kickstart。

注意 如果不指定服务器名称，客户机系统就会试图使用回答 BOOTP/DHCP 请求的服务器作为 NFS 服务器；如果不指定路径或文件名，客户机系统会试图从 BOOTP/DHCP 服务器挂载/Kickstart，然后使用和前面描述的<ip-addr>-Kickstart 文件名相同的方法来搜索 Kickstart 文件。

（3） 筹备安装树

安装树（installation tree）相当于二进制 Red Hat Enterprise Linux 光盘的一个复制版本，其具备与光盘相同的目录结构，Kickstart 安装需要使用安装树。

如果是基于光盘的方式安装 Kickstart，在 Kickstart 开始安装前需要确保 Red Hat Enterprise Linux 光盘已插入计算机。如果是基于硬盘驱动器的方式安装，在 Kickstart 开始安装前需要确保二进制 Red Hat Enterprise Linux 光盘的映像位于计算机的硬盘驱动器上。如果基于网络（NFS、FTP 或 HTTP）的方式安装，在 Kickstart 开始安装前需要确保能够通过网络来提供安装树。

2. 几种常见的 Kickstart 安装方式

开始 Kickstart 自动化安装后，如果选择从 Red Hat Enterprise Linux 5.3 引导光盘或从 Red Hat Enterprise Linux 5.3 安装光盘中安装，则在引导提示下需要输入一个特殊的引导命令 ks。当 ks 命令行参数被传递给内核时，安装程序就会寻找 Kickstart 文件。下面就以两种常见的 Kickstart 安装方式进行举例说明。

（1） 基于光盘的方式安装

如果 Kickstart 文件位于引导光盘上，把光盘插入光驱并引导系统，然后在引导提示下输入以下命令（ks.cfg 即 Kickstart 文件的名称）。

```
boot: linux ks=cdrom:/ks.cfg
```

（2） 基于网络的方式安装

如果 Kickstart 文件位于 NFS 服务器上，可以采用 NFS 安装方式引导系统，然后在引导提示下输入以下命令 。例如，NFS 服务器是 server.eg.com，那么 Kickstart 文件位于 NFS 服务器的共享/dirks/ks.cfg 上。

```
boot: ks=nfs:server.eg.com:/ dirks/ks.cfg
```

如果 Kickstart 文件位于 HTTP 服务器上，可以采用 HTTP 安装方式引导系统，然后在引导提示下输入以下命令。

```
boot: ks=http://server.eg.com/dirks/ks.cfg
```

17.1.2 Debian/Ubuntu 的自动化安装

Debian 可以通过 Debian 安装程序的预配置文件进行自动安装，也可以通过建立一个包含所有预设项目的完整预配置文件模板来实现。

1. 通过预配置文件进行自动安装

预配置文件可以从网络上或者可移动介质上装载，并且可以在安装过程中自动回答问题，同 Kickstart 的自动化安装一致。

预配置文件是按照 debconf-set-selections 命令的格式写成的。

2. 通过建立预配置文件模板实现

可以使用 debconf-utils 软件包中的 debconf-get-selections 命令将 debconf 数据库和位于目录/var/log/debian-installer/cdebconf 下的 cdebconf 数据库的内容导出到一个文件中，形成一个包

含所有预设项目的完整预配置文件模板。

```
debian5:~$ debconf-get-selections --install > [file]
debian5:~$ debconf-get-selections > [file]
```

生成上述两种预配置文件模板后，即可选择适合自己需要的模板，然后按照自己的需求进行编辑，形成最终的预配置文件。接下来将准备好的预配置文件上传至 Web 服务器，或者复制到安装程序的启动引导介质中。安装预配置文件有如下 3 种方法。

注意 无论将此文件放置在哪里，都需要在启动安装程序时向引导系统传递一个参数。

- 从光盘安装

要将预配置文件放置在光盘中，然后重新生成 ISO 映像文件（此时 ISO 映像文件中包括了预配置文件）。

- 从网络安装

要让安装程序使用一个从网络下载的预配置文件，需要在 preseed/url=http://url/to/config 中加入内核启动参数。当然这个预配置文件需要安装程序能够管理网络而且该配置文件在下载后才能奏效。因此，如果安装程序能够通过 DHCP 设置网络而不需要询问任何问题，此方法会更有用。如果已经设置好了网络，还可以将安装优先级调至“关键”来避免任何提问。

- 从 USB 设备安装

要从 USB 盘上启动，可以复制已经准备好的预配置文件至 USB 盘的文件系统中，并通过编辑 syslinux.cfg 文件来将 preseed/file=/hd-media/preseed.cfg 加入至内核启动参数。

17.2 Linux 操作系统中的软件包管理

在 Linux 操作系统中，为用户提供了方便的软件包管理系统，以便用户对软件包进行更好的管理。其中，RPM 是在 Linux 环境中被广泛使用的开放的软件包管理系统，在本节将对该系统进行详细的阐述。

17.2.1 RPM 简介

作为 Linux 环境中被广泛使用的开放的软件包管理系统，该系统最初由 Red Hat 软件公司研制，并命名为 Red Hat Package Manager（即 RPM）。在广大用户和 Red Hat 软件公司的大力推动和鼓励下，RPM 成为了 Linux 系统中公认的软件包管理标准。Fedora、Redhat、Mandriva、SuSE 和 YellowDog 等主流发行版本，以及在这些版本基础上二次开发出来的其他发行版均采用 RPM 作为软件包管理标准，很多其他软件厂商也逐渐开始支持 RPM，并在自己的产品中使用。有兴趣的朋友如果想了解 RPM 产品的更多详细信息，则可以登录 RPM 的官方站点（http://www.rpm.org/），如图 17-11 所示。



图 17-11 RPM 官方站点

1. RPM 包含的内容

RPM 包里面包含许多可执行的二进制程序，以及可执行程序运行时所需要的支持文件。其中可执行的二进制程序的作用类似于 Windows 操作系统中的软件包中的.exe 可执行文件。另外，还包括其他相关的配置文件、资源文件、数据文件以及动态连接文件等。RPM 包中的应用程序除了包括确保自身运行正常的的附加文件以外，有时还需要其他特定版本文件，这就是软件包的依赖关系。依赖关系并不是 Linux 操作系统所特有的，在 Windows 操作系统中也存在同样的问题。

2. RPM 包的基本安装原理

RPM 软件包的基本安装原理如下所述。

首先在安装指定的 RPM 软件包时，应判断该该包是否可以直接安装。如果不需要其他的支持组件，系统便开始直接安装该软件包；如果在安装过程中发现缺乏某些支持组件，系统便会根据 RPM 指令中指定的条件寻找未安装的支持组件包，如果系统未找到支持组件包便中止安装过程，并通知用户该 RPM 软件安装包安装失败；如果找到了需要的支持组件包，便会重复上述的过程，直到所有的包都成功安装。

3. RPM 软件包的种类

RPM 软件包分为如下两种。

- 二进制包（Binary）

二进制软件包可以直接安装在计算机中。

- 源代码包（Source）

源代码包由 RPM 自动编译、安装。源代码软件包常常以.src.rpm 作为软件包文件的后缀名。我们通常将二进制软件包称为 RPM 包，而将源代码软件包称为 SRPM 包。

通过前面对 RPM 的简单讲解，可以了解到 RPM 是在 Linux 环境下广泛使用的开放的软件包管理系统，其实它还是一套以数据库记录的方式将所需的软件包安装在 Linux 主机中的软件包管理程序。RPM 最大的特点就是将用户所需的安装软件包预先编译并打包好，通过软件包中预设的数据库记录，来记录软件包安装所必需的支持模块信息。同时 RPM 具有支持快速安装的功能，另外还可以减少编译、安装、调错的困扰。对于终端用户而言，RPM 所提供的众多功能使维护系统要比在编译源码安装方式下维护系统容易得多。安装、卸载和升级 RPM 软件包均只需一条命令即可完成，所有琐碎的细节问题无需终端用户过度费心。

同时，在软件包升级过程中，RPM 会对配置文件进行特别处理，因此绝对不会丢失以往的定制信息。RPM 软件包是已经打包好的数据，因此，对安装环境的要求比较严格，必须保证安装软件包的环境与打包时的环境需求一致或者相当。另外，要保证满足软件包的相依属性的需求。还有，在 RPM 软件包反安装的过程中需要特别小心，尤其要注意不可移除位于底层的软件包，否则可能造成整个系统出错。

4. RPM 软件包的命名格式

RPM 包的名称有其特有的格式，如某软件的 RPM 包名称如下。

```
name-version-compilecount.type.rpm
```

其中，除了后面应用的硬件平台类型与扩展名外，其余部分主要是以符号“-”来分隔的，如此命名主要是方便用户了解该软件包的名称、版本信息、打包次数及操作的硬件平台。下面，就了解一下构成 RPM 包名称的各部分含义。

name：软件的名称。

version：软件的版本号。每一次更新版本就需要有一个版本号。版本通常又分为主版本和次版本，在主版本的架构下更改部分原始码内容，便可以发布一个新的版本，也就是次版本。

compilecount：编译的次数。那么为何需要重复编译呢？这是由于在同一版的软件包中，可能存在某些 BUG 或者是安全隐患，必须要重新设定当初打包时候的参数，设定完成之后重新编译并打包成 RPM 档案。

type：操作硬件平台。

由于 RPM 可以适用在不同的操作平台上，但是不同的操作平台设定的参数有所差异，而用户可以针对比较高阶的 CPU 进行最优化的参数设定，所以就有所谓的 i386、i586、i686 与 noarch 等文件名称出现，具体操作系统平台类型及说明如表 17-1 所示。

表 17-1　操作系统平台类型及说明

类　型	相 关 说 明
i386	几乎适用于所有的 x86 平台，不论是旧的 Pentium 还是新的 Pentium IV 与 K7 系列的 CPU 等，都可以正常的工作。i 指的是 Intel 兼容的 CPU，386 是 CPU 的等级
i586	CPU 是 586 等级的计算机，包括 Pentium 第一代 MMX CPU，AMD 的 K5、K6 系列 CPU（socket 7 插脚）等都是这个等级
i686	在 Pentium 第二代以后的 Intel 系列 CPU，及 K7 以后的 CPU 等级都属于 686 等级
noarch	没有任何硬件等级上的限制

注意 i386的文件可以在任何的机器上面安装，不论是586或者是686的机器，但是i686的文件则不一定可以在386或者586的硬件平台上使用。这是因为i686的RPM文件在编译时，主要是针对686硬件等级的CPU来进行最佳化编译的，而386/586等级的硬件可能不支持这些参数，所以无法使用。

17.2.2 RPM软件包管理系统的应用

RPM软件包管理系统的应用非常广泛，包括安装软件包、升级与更新软件包、查询软件包、验证软件包、反安装软件包及重建软件包数据库等。接下来以实例的形式向大家详细介绍如何使用该软件包。

1. 使用RPM安装软件包

（1） 语法格式及参数说明

安装RPM软件包采用的命令格式如下。

```
[root@localhost ~]# rpm [-i] [选项] [file1.rpm…fileN.rpm]
```

各参数的含义描述如下。

[file1.rpm…fileN.rpm]：准备安装的多个RPM软件包。

-h（or –hash）：通过输出“#”字符在安装信息行上显示安装进度。

--test：想要测试一下该软件包是否可以被安装到使用者的Linux环境当中使用。例如：rpm -ivh pkgname.i386.rpm --test 。

--percent：在软件包安装过程中输出安装百分比。

--excludedocs：设置不安装当前软件包的文档。

--includedocs：设置安装当前软件包的文档。

--replacepkgs：设置重新安装某个已经安装过的软件包。

--replacefiles：如果在安装的过程当中显示了“某个文件已经被安装在当前的系统上面”提示，或出现了“版本冲突”的提示时，可以使用这个参数来直接覆盖文件。

注意 该覆盖动作是无法复原的。因此必须要确保被覆盖的文件已经备份。否则因覆盖文件而产生问题却又无法复原，将会对当前系统造生重大不良影响。

--force：该参数其实就是--replacefiles与--replacepkgs的综合体。

--noscripts：不需要该软件包自行启用或者自行执行某些系统指令。RPM的优点除了可以将文件放置到指定位置，还可以自动执行一些前置工作的指令，例如数据库的初始化。如果不想让RPM帮用户自动执行这一类型的指令，必须选择该参数。

--prefix <path>：设置软件包的安装路径。

--ignorearch：设置不校验软件包的体系结构。

--ignoreos：设置不校验软件包安装的操作系统。

--nodeps：设置不校验软件包的依赖性。当用户在安装某个软件包时，若总是发现 RPM 提示“有属性相依的软件包尚未安装”信息，而又要直接强制安装这个软件包时，可以加上 --nodeps 参数，告知 RPM 不要去检查软件包的相依性。

注意 软件包有相依性的原因是彼此会使用到对方的机制或功能，如果强制安装而不考虑软件包的依赖关系，则可能会造成该软件包无法正常使用。

--nomd5：设置不检查 RPM 软件包的 MD5 信息。除非用户很清楚这个软件包的来源，否则不建议使用这个参数。

--ftpproxy <host>：设置通过 FTP 方式安装软件包时的代理服务器的主机名。

--ftpport <port>：设置通过 FTP 方式安装软件包时的 FTP 端口号。

-v：设置显示安装过程中的详细信息。

-vv：设置显示调试信息。

--root <path>：设置候选 root 路径。

-rcfile <rcfile>：设置候选 rpmrc 文件。

--dbpath <path>：设置 RPM 数据库的路径。

（2） 具体实例

实例 1 安装本地软件包 elinks-0.11.1-5.1.0.1.el5.i386.rpm，执行如下命令。

```
[root@localhost ~]# rpm -ivh /media/RHEL_5.2\ i386\DVD/Server/elinks-0.11.1-5.1.0.1.el5.i386.rpm
```

执行命令后的结果如下所示。

```
[root@localhost ~]# cd /media/RHEL_5.2\ i386\ DVD/Server/
[root@localhost Server]# rpm -ivh elinks-0.11.1-5.1.0.1.el5.i386.rpm
warning: elinks-0.11.1-5.1.0.1.el5.i386.rpm: Header V3 DSA signature: NOKEY, key ID 37017186
Preparing...                ########################################### [100%]
1:elinks                    ########################################### [100%]
```

实例 2 安装远程软件包 elinks-0.11.1-5.1.0.1.el5.i386.rpm，执行如下命令。

```
[root@localhost ~]#rpm -ivh http://centos.candishosting.com.cn/5/os/i386/CentOSelinks-0.11.1-5.1.0.1.el5.i386.rpm
```

执行命令后的结果如下所示。

```
[root@localhost ~]# rpm -ivh http://centos.candishosting.com.cn/5/os/i386/CentOSelinks-0.11.1-5.1.0.1.el5.i386.rpm
Retrieving http://centos.candishosting.com.cn/5/os/i386/CentOSelinks-0.11.1-5.1.0.1.el5.i386.rpm
Preparing...                ########################################### [100%]
1:elinks                    ########################################### [100%]
```

实例 3 安装不在 RedHat 软件库中的软件包 darcs-1.0.9-3.fc6.i386.rpm。以安装 Fedora Extras 中的软件包 darcs-1.0.9-3.fc6.i386.rpm 为例，安装不在 RedHat 软件库中的软件包的步骤如下。

首先执行 rpm –import 指令导入安装 Fedora Extras 中软件包所需的签名文件 RPM-GPG-KEY-Fedora-Extras。

```
[root@RedHat5 ~]# rpm --import
http://download.fedora.redhat.com/pub/fedora/linux/extras/RPM-GPG-KEY-Fedora-
Extras
```

然后执行 wget 指令，从 RedHat 官方网站上下载软件包 darcs-1.0.9-3.fc6.i386.rpm。

```
[root@RedHat5 ~]# wget
http://download.fedora.redhat.com/pub/fedora/linux/extras/6/i386/darcs-1.0.9-
3.fc6.i386.rpm
```

最后执行 rpm -ivh 命令安装已下载的软件包。

```
[root@RedHat5 ~]# rpm -ivh darcs-1.0.9-3.fc6.i386.rpm
```

执行命令后的结果如图 17-12 所示。

```
[root@redhat5 ~]#
[root@redhat5 ~]# rpm --import http://download.fedora.redhat.co
m/pub/fedora/linux/extras/RPM-GPG-KEY-Fedora-Extras
[root@redhat5 ~]# wget http://download.fedora.redhat.com/pub/fe
dora/linux/extras/6/i386/darcs-1.0.9-3.fc6.i386.rpm
--23:46:34--  http://download.fedora.redhat.com/pub/fedora/linu
x/extras/6/i386/darcs-1.0.9-3.fc6.i386.rpm
Resolving download.fedora.redhat.com... 209.132.176.220, 209.13
2.176.221, 209.132.176.20
Connecting to download.fedora.redhat.com|209.132.176.220|:80...
 connected.
HTTP request sent, awaiting response... 200 OK
Length: 1040834 (1016K) [application/x-rpm]
Saving to: `darcs-1.0.9-3.fc6.i386.rpm'

100%[======================>] 1,040,834    161K/s   in 6.3s

23:48:15 (161 KB/s) - `darcs-1.0.9-3.fc6.i386.rpm' saved [10408
34/1040834]

[root@redhat5 ~]# rpm -ivh darcs-1.0.9-3.fc6.i386.rpm
Preparing...
########################################### [100%]
   1:darcs
########################################### [100%]
[root@redhat5 ~]#
```

图 17-12 成功安装不在 RedHat 软件库中的软件包

实例 4 安装本地软件包 eject-2.1.5-4.2.el5.i386.rpm 并了解其安装过程中的详细调试信息，执行如下命令。

```
[root@RedHat5 ~]#rpm -ivvh /media/RHEL_5.3\ i386\
DVD/Server/eject-2.1.5-4.2.el5.i386.rpm
```

执行命令后的结果如图 17-13 所示。

```
[root@redhat5 ~]# rpm -ivvh /media/RHEL_5.3\ i386\ DVD/Serve
r/eject-2.1.5-4.2.el5.i386.rpm
D: ============== /media/RHEL_5.3 i386 DVD/Server/eject-2.1.
5-4.2.el5.i386.rpm
D: Expected size:        50309 = lead(96)+sigs(344)+pad(0)+d
ata(49869)
D:   Actual size:        50309
D: opening  db environment /var/lib/rpm/Packages joinenv
D: opening  db index       /var/lib/rpm/Packages rdonly mode
=0x0
D: locked   db index       /var/lib/rpm/Packages
D: opening  db index       /var/lib/rpm/Pubkeys rdonly mode=
0x0
warning: /media/RHEL_5.3 i386 DVD/Server/eject-2.1.5-4.2.el5
.i386.rpm: Header V3 DSA signature: NOKEY, key ID 37017186
D:      added binary package [0]
D: found 0 source and 1 binary packages
D: ========== +++ eject-2.1.5-4.2.el5 i386/linux 0x1
D: opening  db index       /var/lib/rpm/Depends create mode=
0x0
D: opening  db index       /var/lib/rpm/Providename rdonly m
ode=0x0
D:  read h#      25 Header V3 DSA signature: NOKEY, key ID 3
7017186
D:  Requires: libc.so.6
YES (db provides)
D:  read h#      25 Header V3 DSA signature: NOKEY, key ID 3
7017186
D:  Requires: libc.so.6(GLIBC_2.0)
```

图 17-13　安装本地软件包

实例 5　测试本地软件包 elinks-0.11.1-5.1.0.1.el5.i386.rpm 可否安装到当前的 Linux 环境中，执行如下命令。

```
[root@localhost ~]#rpm -ivh --test /media/RHEL_5.2\ i386\
DVD/Server/elinks-0.11.1-5.1.0.1.el5.i386.rpm
```

执行命令后的结果如下所示，表明该软件包已经成功安装。

```
[root@localhost ~]# cd /media/RHEL_5.2\ i386\ DVD/Server/
[root@localhost Server]# rpm -ivh --test elinks-0.11.1-5.1.0.1.el5.i386.rpm
warning: elinks-0.11.1-5.1.0.1.el5.i386.rpm: Header V3 DSA signature: NOKEY,
key ID 37017186
Preparing...                ########################################### [100%]
        package elinks-0.11.1-5.1.0.1.el5 is already installed
```

注意　--test 参数检测 Linux 系统中是否可以成功安装选定的软件包 elinks-0.11.1-5.1.0.1.el5.i386.rpm。如果能够成功安装，则系统提示可以 100%安装；否则系统将提示选择的软件包不可成功安装并显示存在的问题。

2. 使用 RPM 升级或更新软件包

（1） 语法格式及参数说明

升级或更新 RPM 软件包采用的命令格式如下。

```
[root@ localhost ~]#rpm [-U|F] [选项] [file1.rpm…fileN.rpm]
```

各参数的含义描述如下。

[file1.rpm…fileN.rpm]：准备安装的多个 RPM 软件包。

-h（or --hash）：通过输出“#”字符在安装信息行上显示安装进度。

--oldpackage：设置该参数用于更新到旧版的软件包。

--test：想要测试一下该软件包是否可以被升级到使用者的 Linux 环境当中使用。

--percent：在软件包升级过程中输出升级百分比。

--excludedocs：设置不安装当前软件包的文档。

--includedocs：设置安装当前软件包的文档。

--replacepkgs：设置重新安装某个已经安装过的软件包。

--replacefiles：如果在安装的过程当中显示了“某个文件已经被安装在当前的系统上面”的提示，或出现了“版本冲突”的提示时，可以使用这个参数来直接覆盖文件。

注意 该覆盖动作是无法复原的，因此必须要确保被覆盖的文件已经备份。否则因覆盖文件，而产生问题却又无法复原，将会对当前系统造成重大不良影响。

--force：该参数其实就是--replacefiles 与--replacepkgs 的综合体。

--noscripts：不想让该软件包自行启用或者自行执行某些系统指令。RPM 的优点除了可以将文件放置到指定位置，还可以自动执行一些前置工作的指令，例如数据库的初始化。如果不想要让 RPM 帮用户自动执行这一类型的指令，必须选择该参数。

--prefix <path>：设置软件包的安装路径。

--ignorearch：设置不校验软件包的体系结构。

--ignoreos：设置不校验软件包安装的操作系统。

--nodeps：设置不校验软件包的依赖性信息。当用户在安装某个软件包时，若总是发现 RPM 提示“有属性相依的软件包尚未安装”，而又想要直接强制安装这个软件包时，可以加上--nodeps 参数告知 RPM 不要去检查软件包的相依性。

注意 软件包有相依性的原因是彼此会使用到对方的机制或功能，如果强制安装而不考虑软件包的依赖关系，则可能会造成该软件包无法正常使用！

--nomd5：设置不检查 RPM 软件包的 MD5 信息。除非用户很清楚这个软件包的来源，否则不建议使用这个参数。

--ftpproxy <host>：设置通过 FTP 方式安装软件包时的代理服务器的主机名。

--ftpport <port>：设置通过 FTP 方式安装软件包时的 FTP 端口号。

-v：设置显示安装过程中的详细信息。

-vv：设置显示调试信息。

--root <path>：设置候选 root 路径。

-rcfile <rcfile>：设置候选 rpmrc 文件。

--dbpath <path>：设置 RPM 数据库的路径。

由上可以看出安装 RPM 软件包的命令参数同升级 RPM 软件包的命令参数及它们的含义几乎一致。

使用 RPM 来升级软件包与使用 RPM 来安装软件包一样简单，差别仅在于将-ivh 参数变

换为-Uvh 或-Fvh。不过这两种升级方式有些不同。

- -Uvh

若参数后的软件包没有安装到 Linux 系统上，则 RPM 系统将直接安装该软件包；若参数后的软件包已经在 Linux 系统中已安装过旧版本，则 RPM 系统将自动更新软件包至新版。

- -Fvh

若参数后的软件包并未安装到 Linux 系统上，则 RPM 系统将不安装该软件包，也就是说，只有安装至 Linux 系统内的软件包才会被更新。

（2） 具体实例

实例 1 从本地文件升级软件包 elinks-0.11.1-5.1.0.1.el5.i386.rpm，执行如下命令。

```
[root@RedHat5 ~]#rpm -Uvh /media/RHEL_5.3\ i386\
DVD/Server/elinks-0.11.1-5.1.0.1.el5.i386.rpm
```

执行命令后的结果如图 17-14 所示。

```
[root@redhat5 ~]# rpm -Uvh /media/RHEL_5.3\ i386\ DVD/Server
/elinks-0.11.1-5.1.0.1.el5.i386.rpm
warning: /media/RHEL_5.3 i386 DVD/Server/elinks-0.11.1-5.1.0
.1.el5.i386.rpm: Header V3 DSA signature: NOKEY, key ID 3701
7186
Preparing...
########################################### [100%]
        package elinks-0.11.1-5.1.0.1.el5.i386 is already in
stalled
[root@redhat5 ~]#
```

图 17-14 成功升级本地软件包

实例 2 从远程文件升级软件包 elinks-0.11.1-5.1.0.1.el5.i386.rpm，执行如下命令。

```
[root@Centos5 ~]#rpm -Uvh http://centos.candishosting.com.cn/5/os/i386/
CentOS/elinks-0.11.1-5.1.0.1.el5.i386.rpm
```

执行命令后的结果如图 17-15 所示。

```
[root@redhat5 ~]# rpm -ivh http://centos.candishosting.com.cn/5
/os/i386/CentOS/elinks-0.11.1-5.1.0.1.el5.i386.rpm
Retrieving http://centos.candishosting.com.cn/5/os/i386/CentOS/
elinks-0.11.1-5.1.0.1.el5.i386.rpm
Preparing...
########################################### [100%]
   1:elinks
########################################### [100%]
[root@redhat5 ~]#
```

图 17-15 成功安装远程软件包

实例 3 大批量地升级一批软件包。

首先建立一个目录来存放从各个开发商的官方网站下载的最新 i386 软件包文件。

```
[root@RedHat5 ~]# mkdir updatedir
[root@RedHat5 ~]# wget ****          #各个开发商的官方网站下载的最新 i386 软件包
```

然后执行 rpm -Fvh 指令进行批量软件包的更新。

```
[root@RedHat5 ~]# rpm -Fvh *.rpm
```

3. 通过 RPM 查询软件包

（1） 语法格式及参数说明

查询 RPM 软件包采用的命令格式如下。

```
[root@localhost ~]#rpm [-q] [选项] [pkg1…pkgN]
[root@localhost ~]#rpm [-q] [选项] [.rpm file1….rpm fileN]
```

各参数的含义描述如下。

pkg1 … pkgN：查询的软件包。

-q <pkg1…pkgN>：查询指定的 RPM 软件包在系统中是否安装。

-qa：查询系统中所有已安装的 RPM 软件包。

-qf <file>：查询一个系统中指定文件所属的 RPM 软件包名称。

-qi <pkg1…pkgN>：显示指定的已安装 RPM 软件包的相关描述信息，包括软件包名称、版本、开发商、SRPM 文件名称、打包次数、简单说明信息、软件包打包者、安装日期等。

-ql <pkg1…pkgN>：显示指定已安装 RPM 软件包中包含的所有相关目录与文件。

-qc <pkg1…pkgN>：显示指定已安装 RPM 软件包的配置文件存放的完整目录名称。

-qd <pkg1…pkgN>：显示指定已安装 RPM 软件包的文件的完整存放路径名称。

-qs <pkg1…pkgN>：显示指定已安装 RPM 软件包中的每个文件的状态。

-qR <pkg1…pkgN>：显示一个已安装 RPM 软件包的最低依赖要求。

-qpi <.rpm file1….rpm fileN>：显示指定未安装的 RPM 包文件的相关描述信息。

-qpl <.rpm file1….rpm fileN >：显示指定未安装的 RPM 包文件中包含的所有相关目录与文件。

-qpc <.rpm file1….rpm fileN >：显示指定未安装 RPM 软件包的配置文件存放的完整目录名称。

-qpd <.rpm file1….rpm fileN >：显示指定未安装 RPM 软件包的文件的完整存放路径名称。

-qps <.rpm file1….rpm fileN >：显示指定未安装 RPM 软件包中的每个文件的状态。

-qpR <.rpm file1….rpm fileN >：查询一个未安装 RPM 软件包的最低依赖要求。

（2） 具体实例

实例 1 检查系统中是否安装软件包 elinks，执行以下命令即可。

```
[root@localhost ~]#rpm -q elinks
```

执行命令后的结果如下所示，表明安装有该软件包。

```
[root@localhost ~]# rpm -q elinks
elinks-0.11.1-5.1.0.1.el5
```

实例 2 检查系统中已安装软件包 elinks 的相关描述信息，执行如下命令。

```
[root@localhost ~]#rpm -qi elinks
```

执行命令后的结果如下所示。

```
[root@localhost ~]# rpm -qi elinks
```

```
Name        : elinks                        Relocations: (not relocatable)
Version     : 0.11.1                            Vendor: Red Hat, Inc.
Release     : 5.1.0.1.el5                   Build Date: 2007年09月25日 星期二 22
时14分50秒
Install Date: 2009 年 04 月 24 日 星期五 22 时 20 分 05 秒      Build Host:
hs20-bc1-5.build.redhat.com
Group               :  Applications/Internet                Source RPM:
elinks-0.11.1-5.1.0.1.el5.src.rpm
Size        : 2482182                          License: GPL
Signature   : DSA/SHA1, 2007 年 10 月 03 日 星期三 22 时 10 分 15 秒, Key ID
5326810137017186
Packager    : Red Hat, Inc. <http://bugzilla.redhat.com/bugzilla>
URL         : http://elinks.or.cz/
Summary     : 文本模式万维网浏览器。
Description :
```

Links 是一个基于文本的万维网浏览器。Links 不显示任何图像，但是它支持 frame、table 以及多数其他 HTML 标签。Links 优于图形化浏览器的地方在于它的速度——Links 能快速启动、退出，并能快速显示网页。

实例 3 检查系统中已安装软件包 elinks 中包含的文件信息，执行如下命令。

```
[root@localhost ~]#rpm -ql elinks
```

执行命令后的结果如下所示。

```
[root@localhost ~]# rpm -ql elinks
/usr/bin/elinks
/usr/bin/links
/usr/share/doc/elinks-0.11.1
/usr/share/doc/elinks-0.11.1/README
/usr/share/doc/elinks-0.11.1/SITES
/usr/share/doc/elinks-0.11.1/TODO
/usr/share/locale/be/LC_MESSAGES/elinks.mo
/usr/share/locale/bg/LC_MESSAGES/elinks.mo
/usr/share/locale/ca/LC_MESSAGES/elinks.mo
```

实例 4 查看 WWW 服务器 Apache 配置文件的存放路径。

注意 Linux 环境中 WWW 服务是通过安装 RPM 软件包 httpd 实现的。若想了解 WWW 服务器中 Apache 的配置文件的存放路径，也就是查看 RPM 软件包 httpd 的所有配置文件安装至何处，可以通过如下指令获取想要查看的信息。

```
[root@localhost ~]#rpm -qc httpd
```

执行命令后的结果如下所示。

```
[root@localhost ~]# rpm -qc httpd
/etc/httpd/conf.d/proxy_ajp.conf
/etc/httpd/conf.d/welcome.conf
/etc/httpd/conf/httpd.conf
```

```
/etc/httpd/conf/magic
/etc/logrotate.d/httpd
/etc/rc.d/init.d/httpd
/etc/sysconfig/httpd
/var/www/error/HTTP_BAD_GATEWAY.html.var
/var/www/error/HTTP_BAD_REQUEST.html.var
/var/www/error/HTTP_FORBIDDEN.html.var
/var/www/error/HTTP_GONE.html.var
/var/www/error/HTTP_INTERNAL_SERVER_ERROR.html.var
/var/www/error/HTTP_LENGTH_REQUIRED.html.var
/var/www/error/HTTP_METHOD_NOT_ALLOWED.html.var
/var/www/error/HTTP_NOT_FOUND.html.var
/var/www/error/HTTP_NOT_IMPLEMENTED.html.var
/var/www/error/HTTP_PRECONDITION_FAILED.html.var
/var/www/error/HTTP_REQUEST_ENTITY_TOO_LARGE.html.var
/var/www/error/HTTP_REQUEST_TIME_OUT.html.var
/var/www/error/HTTP_REQUEST_URI_TOO_LARGE.html.var
/var/www/error/HTTP_SERVICE_UNAVAILABLE.html.var
/var/www/error/HTTP_UNAUTHORIZED.html.var
/var/www/error/HTTP_UNSUPPORTED_MEDIA_TYPE.html.var
/var/www/error/HTTP_VARIANT_ALSO_VARIES.html.var
/var/www/error/contact.html.var
/var/www/error/include/bottom.html
/var/www/error/include/spacer.html
/var/www/error/include/top.html
```

实例 5 查看系统中文件/etc/passwd 所属的 RPM 软件包信息。

注意 假如在操作的过程中误删了某个重要文件，例如/etc/passwd，可是又不清楚该文件属于哪一个软件包，该怎么办？可以按照下面的方法，获取系统中文件/etc/passwd 所属的 RPM 软件包信息并重新安装这个软件包即可。

```
[root@localhost ~]#rpm -qf /etc/passwd
```

执行命令后的结果如下所示，系统提示用户文件/etc/passwd 所属的 RPM 软件包是 setup-2.5.58-4.el5。

```
[root@localhost ~]# rpm -qf /etc/passwd
setup-2.5.58-1.el5
```

实例 6 查看 Server 目录下的 RPM 软件包文件 dapl-2.0.7-2.el5.i386.rpm 中的信息，执行如下命令。

```
[root@localhost Server]# rpm -qp dapl-2.0.7-2.el5.i386.rpm
```

执行命令后的结果如下所示。

```
[root@localhost Server]# rpm -qp dapl-2.0.7-2.el5.i386.rpm
warning: dapl-2.0.7-2.el5.i386.rpm: Header V3 DSA signature: NOKEY, key ID
37017186
```

```
dapl-2.0.7-2.el5
```

实例 7 查看系统中已安装的软件包名中含有字符串“zip”的所有 RPM 软件包，执行如下命令。

```
[root@localhost ~]# rpm -qa|grep zip
```

执行命令后的结果如下所示。

```
[root@localhost ~]# rpm -qa|grep zip
bzip2-1.0.3-3
unzip-5.52-2.2.1
bzip2-devel-1.0.3-3
gzip-1.3.5-10.el5
zip-2.31-1.2.2
bzip2-libs-1.0.3-3
```

实例 8 查看系统中已安装的 RPM 软件包 httpd 的最低依赖要求信息，执行如下命令。

```
[root@localhost ~]# rpm -qR httpd
```

执行命令后的结果如下所示。

```
[root@localhost ~]# rpm -qR httpd
/bin/bash
/bin/mktemp
/bin/mv
/bin/rm
/bin/sh
/bin/sh
/bin/sh
/bin/sh
/etc/mime.types
/sbin/chkconfig
/usr/bin/find
/usr/sbin/useradd
/usr/share/magic.mime
config(httpd) = 2.2.3-11.el5_1.3
gawk
initscripts >= 8.36
[root@RedHat5 ~]#rpm -qR httpd
```

4. 通过 RPM 卸载软件包

卸载 RPM 软件包的命令格式如下。

```
[root@localhost ~]#rpm -e [file1.rpm…fileN.rpm]
```

注意 卸载软件包的次序一定要同安装软件包时的次序相反，即从上向下卸载。

实例 卸载已安装在当前 Linux 系统中的软件包 elinks-0.11.1-5.1.0.1.el5，执行以下命令即可。

```
[root@RedHat5 ~]#rpm -e elinks-0.11.1-5.1.0.1.el5
```

5. 通过 RPM 验证软件包

（1） 语法格式及参数说明

校验 RPM 软件包采用的命令格式如下。

```
[root@localhost ~]#rpm [-V] [选项] [pkg1…pkgN]
[root@localhost ~]#rpm [-V] [选项] [.rpm file1….rpm fileN]
```

具体参数的含义描述如下。

pkg1 … pkgN：校验的软件包。

-V <pkg1…pkgN>：校验指定的 RPM 软件包。

-V <file>：校验包含指定文件的 RPM 软件包。

-Vp <.rpm file1….rpm fileN >：校验指定的未安装的 RPM 软件包文件。

-Vf <file>：校验指定的文件。

-Va：校验所有已安装的 RPM 软件包。

（2） 具体实例

实例 1 校验系统中安装的软件包 Nessus，执行如下命令。

```
[root@localhost ~]# rpm -V Nessus
```

执行命令后的结果如下所示。

```
[root@localhost ~]# rpm -V Nessus
.M......   /opt/nessus/var/nessus
```

如果校验结果完全正确，则系统没有任何输出信息；如果校验结果不正确，则系统输出不一致结果，格式如下所示。

```
XXXXXXXX  Y  文件名
```

此字段由 8 个字符组成，每个字符指明该文件与 RPM 数据库中一致或不一致的地方，单个点（.）说明没有异常，具体含义描述如下。

S（file Size differs）：指定文件的大小已被改变。

M 即 Mode differs （includes permissions and file type）：指定文件的类型或文件的属性已被改变。

5（MD5 sum differs）：MD5 校验和已被改变。

D（Device major/minor number mis-match）：设备名称已被改变。

L（readLink（2）path mis-match）：Link 属性已被改变。

U（User ownership differs）：文件的所属人已被改变。

G（Group ownership differs）：文件的所属群组已被改变。

T（mTime differs）：文件的建立时间已被改变。

？：文件不可读。

上述字段由一个字符组成，该字符用来描述指定文件的类型，文件类型包括以下几类（c、d、g、l、r），具体含义描述如下。

c（config file）：配置文件

d（docmentation）：数据文件
g（ghost file）：ghost 文件
l（license file）：授权文件
r（Read me）：自述文件

实例 2 校验包含文件/etc/passwd 的软件包。

```
[root@localhost ~]##rpm -Vf /etc/passwd
```

执行命令后的结果如下所示。

```
[root@localhost ~]# rpm -Vf /etc/passwd
..5....T c /etc/printcap
```

实例 3 校验 RPM 软件包文件 elinks-0.11.1-5.1.0.1.el5，执行以下命令即可。

```
[root@localhost ~]# rpm -Vp /media/RHEL_5.2\ i386\
DVD/Server/elinks-0.11.1-5.1.0.1.el5.i386.rpm
```

6. 通过 RPM 重建数据库

重建数据库采用的命令格式如下。

```
[root@localhost ~]#rpm [--rebuilddb]
```

实例 重新创建系统的 RPM 数据库，执行以下命令即可。

```
[root@localhost ~]#rpm --rebuilddb
```

17.3 Debian 的软件包管理系统

DPKG 是 Debian 的软件包管理系统。在 Debian 中，可以利用该管理系统通过数据库实现对安装在主机系统中软件包的维护工作。在本节将具体学习 DPKG 的相关知识。

17.3.1 DPKG 简介

DPKG 是一款中级软件包管理系统，其是通过数据库来维护安装在主机系统中的软件包的，包括软件包文件清单、软件包间的依赖关系、软件包的安装状态等详细的资料。这些资料文件通常放置在当前系统的/var/lib/dpkg 目录下，通过这些资料便能确保系统与数据库中记录的情况相一致。

DPKG 负责安装、卸载软件包，配置以及维护已安装的软件包，也是 Debian 系统中众多软件包管理工具的后端支持工具。系统中许多其他的软件包管理工具（包括 APT、DSELECT 以及 TASKSEL 等）都通过 DPKG 执行中等级别的包管理工作。

DPKG 在处理软件包时，首先会评估当前处理的任务是否会引起系统的混乱，如果经评估认定当前操作有可能导致系统出现混乱，DPKG 工具将拒绝安装此软件包；反之，DPKG 会将软件包按照要求正确处理，因此使用 DPKG 安装软件非常安全。

17.3.2 DPKG 软件包管理系统应用

DPKG 作为软件包管理系统，为 Debian 系统用户提供了非常丰富的功能，包括安装软件包、升级与更新软件包、查询软件包、验证软件包、反安装软件包与重建软件包数据库等。本小节将结合案例介绍该系统。

1. 通过 DPKG 安装软件包

通过 DPKG 安装软件包一般采用如下命令格式，使用带-i 或--install 参数的 dpkg 命令和待安装软件的.deb 软件包文件的路径来安装软件包。

（1） 语法格式及参数说明

```
debian5:~$ dpkg [-i|--install] [package_file...]
```

注意 该路径必须是可访问的本地文件系统路径（不支持 HTTP、FTP 等远程文件路径），也可以指定多个软件包文件。

各参数的含义描述如下。

[package_file...]：准备安装的多个软件包文件。

[-i|--install]：安装软件包参数。如果指定--recursive 或-R 选项，必须指定一个目录代替包文件。

（2） 具体安装步骤

具体安装步骤如下。

第 1 步，导出新软件包的控制文件。

第 2 步，如果相同软件包的另一个版本在新软件包安装前已经被安装，执行旧软件包的 prerm 脚本程序。

第 3 步，运行由软件包提供的 postinst 脚本程序。

第 4 步，解压新文件同时备份旧文件，以便遇到问题时方便修复。

第 5 步，如果另一个版本的同一个软件包在新软件包安装前已经安装，则执行旧软件包中的 postrm 脚本程序。注意，应该执行 postrm 脚本后再执行新软件包中的 preint 脚本，因为写新文件的同时移除旧文件。

第 6 步，配置软件包。

（3） 具体实例

实例 安装本地软件包 gpe-icons_0.25-1_all.deb，执行如下命令。

```
debian5:~$ dpkg -i gpe-icons_0.25-1_all.deb
```

执行命令后的结果如图 17-16 所示。

```
root@debian:~#
root@debian:~#
root@debian:~# dpkg -i gpe-icons_0.25-1_all.deb
选中了曾被取消选择的软件包 gpe-icons。
(正在读取数据库 ... 系统当前总共安装有 73083 个文件和目录。)
正在解压缩 gpe-icons (从 gpe-icons_0.25-1_all.deb) ...
正在设置 gpe-icons (0.25-1) ...
root@debian:~#
root@debian:~#
```

图 17-16　成功安装本地软件包

2. 通过 DPKG 卸载软件包

DPKG 卸载软件包一般使用带-r 或--remove 参数的 dpkg 命令和准备卸载已安装的软件包名来卸载软件包。

注意　与安装不同的是，卸载软件包时只需要软件包名，而不是实际的 deb 软件包文件名。同样的，也可以指定多个已安装的软件包。

在 Debian 中，卸载和清除软件包是两个不同的概念。软件包被卸载（删除）后，其配置文件仍会留在系统中，只有清除软件包时才会彻底删除，包括删除配置文件。默认情况下，Debian 仅做卸载（删除）操作，除非明确指出，才会将配置文件删除。如果要清除软件包，在清除前将会隐含地执行删除操作。

（1） 在 Debian 中卸载软件包的语法及参数

在 Debian 中卸载软件包的语法如下。

```
debian5:~$ dpkg [-r|--remove] [package...]
```

各参数含义的描述如下。

[package...]：准备卸载的多个软件包。

[-rl--remove]：卸载软件包参数。如果指定--recursive 或-R 选项，必须指定一个目录代替包文件。

具体卸载步骤如下。

第 1 步，如果存在 prerm 脚本程序，则先运行 prerm 脚本程序。

第 2 步，保留配置文件，删除其他的全部文件。

第 3 步，删除/var/lib/dpkg/info 目录下软件包的所有配置文件，保留 postrm 和 list 文件。

第 4 步，最后，通过 DPKG 在数据库中将软件状态修改为删除状态。

（2） 在 Debian 中清除软件包的语法及参数

在 Debian 中清除软件包的语法如下。

```
debian5:~$ dpkg [-P|--purge][-a|--pending] [package...]
```

各参数含义的描述如下。

[package...]：准备卸载的多个软件包。

[-Pl--purge]：清除软件包参数。删除所有指定软件包相关文件（包括配置文件）。（主要删除的是/var/lib/dpkg/status 中指定的文件），如果给出的是-a 或者--pending 而不是包的名字，那么所有的包都被打开（解压缩）。在/var/bin/dpkg/status 文件中标记被移除或清除的，将会分别被移除或清除（但是在/var/bin/dpkg/status 文件中标记“要”移除）。

具体清除步骤如下。

第 1 步，删除所有的文件。

第 2 步，如果存在 prerm 脚本程序，运行 postrm 脚本。

第 3 步，删除保留的两个文件 postrm 和 list，从系统中解除 postfix.postrm 和 postfix.files.

第 4 步，在 DPKG 的数据库中将软件包标记为未安装。

（3） 具体实例

实例 1 卸载本地系统中已安装的软件包 gpe-icons，执行如下命令。

```
debian5:~$ dpkg -r gpe-icons
```

执行命令后的结果如图 17-17 所示。

实例 2 清除本地系统中已安装的软件包 gpe-icons。

首先执行命令查看清除前的安装情况。

```
debian5:~# dpkg -l gpe-icons
```

然后执行命令查看清除后的安装情况。

```
debian5:~# dpkg -P gpe-icons
```

执行命令后的结果如图 17-18 所示。

```
debian5:~#
debian5:~# dpkg -l gpe-icons
Desired=Unknown/Install/Remove/Purge/Hold
| Status=Not/Inst/Cfg-files/Unpacked/Failed-cfg/Half-inst/trig-aWait/Trig-pend
|/ Err?=(none)/Hold/Reinst-required/X=both-problems (Status,Err: uppercase=bad)
||/ Name           Version        Description
+++-==============-==============-============================
ii  gpe-icons      0.25-1         Common icons used by GPE programs
debian5:~# dpkg -r gpe-icons
(Reading database ... 88110 files and directories currently installed.)
Removing gpe-icons ...
debian5:~#
debian5:~# dpkg -l gpe-icons
Desired=Unknown/Install/Remove/Purge/Hold
| Status=Not/Inst/Cfg-files/Unpacked/Failed-cfg/Half-inst/trig-aWait/Trig-pend
|/ Err?=(none)/Hold/Reinst-required/X=both-problems (Status,Err: uppercase=bad)
||/ Name           Version        Description
+++-==============-==============-============================
pn  gpe-icons      <none>         (no description available)
debian5:~#
```

图 17-17 成功卸载本地软件包

```
debian5:~# dpkg -l gpe-icons
Desired=Unknown/Install/Remove/Purge/Hold
| Status=Not/Inst/Cfg-files/Unpacked/Failed-cfg/Half-inst/trig-aWait/Trig-pend
|/ Err?=(none)/Hold/Reinst-required/X=both-problems (Status,Err: uppercase=bad)
||/ Name           Version        Description
+++-==============-==============-============================
ii  gpe-icons      0.25-1         Common icons used by GPE programs
debian5:~#
debian5:~# dpkg -P gpe-icons
(Reading database ... 88110 files and directories currently installed.)
Removing gpe-icons ...
debian5:~#
debian5:~# dpkg -l gpe-icons
Desired=Unknown/Install/Remove/Purge/Hold
| Status=Not/Inst/Cfg-files/Unpacked/Failed-cfg/Half-inst/trig-aWait/Trig-pend
|/ Err?=(none)/Hold/Reinst-required/X=both-problems (Status,Err: uppercase=bad)
||/ Name           Version        Description
+++-==============-==============-============================
pn  gpe-icons      <none>         (no description available)
debian5:~#
```

图 17-18 成功清除本地软件包

如图 17-18 所示，输出的状态为 pn 意味着软件包实际上是被清除了，但是在数据库中还有这个软件包的记录。因此，pn 明确指出以前安装过软件包 gpe-icons，在软件包数据库中出现过，并且现在已被完全清除。

3. 通过 DPKG 查询软件包数据库，检查软件包文件

DPKG 使用文本文件作为数据库来维护系统中安装的软件包，该文件通常放置在 /var/lib/dpkg 目录下，包括文件清单、依赖关系、软件状态等详细信息。status 文件中存储着软

件状态和控制信息，info/目录下存放着备份控制文件，在其下的.list 文件中记录安装文件清单，.mdasums 保存文件的 MD5 编码。

通过 DPKG 查询已安装在当前系统中的软件包一般采用如下命令格式。

```
debian5:~$ dpkg [-l|--list] [package...]
```

注意 使用带-l 或--list 参数的 dpkg 命令来查询软件包。

使用 DPKG 查询已安装在当前系统中的软件包的详细信息一般采用如下命令格式。

```
debian5:~$ dpkg [-s|--status] [package...]
```

使用 DPKG 检查选定的软件包文件，显示选定的软件包文件的相关信息，一般采用如下命令格式。

```
debian5:~$ dpkg [-I|--info] [packageFile...]
```

注意 使用带-s 或--status 参数的 dpkg 命令来查询软件包。

使用 DPKG 列出选定软件包文件中的文件信息一般采用如下命令格式。

```
debian5:~$ dpkg [-c|--contents] [packageFile...]
```

使用 DPKG 列出选定已安装的软件包中的文件信息一般采用如下命令格式。

```
debian5:~$ dpkg [-L|--listfiles] [package...]
```

使用 DPKG 查询系统中某个指定文件所属的软件包信息一般采用如下命令格式。

```
debian5:~$ dpkg [-S|--search] [system file...]
```

实例 1 列出当前主机上所有安装的软件包的信息，执行如下命令。

```
debian5:~$ dpkg -l | more
```

执行命令后的结果如图 17-19 所示，每条记录对应一个软件包。注意每条记录的前 3 个字符，就是软件包的状态标识，后边依次是软件包名称、版本号和简单描述，如表 17-2 所示。

```
debian5:~# dpkg -l|more
Desired=Unknown/Install/Remove/Purge/Hold
| Status=Not/Inst/Cfg-files/Unpacked/Failed-cfg/Half-inst/
trig-aWait/Trig-pend
|/ Err?=(none)/Hold/Reinst-required/X=both-problems (Statu
s,Err: uppercase=bad)
||/ Name                                    Version
        Description
+++-=======================================-================
=========-=================================================
ii  acpi                                    1.1-2
        displays information on ACPI devices
ii  acpi-support-base                       0.109-11
        scripts for handling base ACPI events such a
ii  acpid                                   1.0.8-1lenny1
        Utilities for using ACPI power management
ii  adduser                                 3.110
        add and remove users and groups
ii  alacarte                                0.11.5-1
        easy GNOME menu editing tool
ii  alsa-base                               1.0.17.dfsg-4
        ALSA driver configuration files
ii  alsa-utils                              1.0.16-2
        ALSA utilities
ii  anacron                                 2.3-13.1
        cron-like program that doesn't go by time
ii  app-install-data                        2008.11.27
        Application Installer Data Files
```

图 17-19 列出所有安装的软件包的信息

表 17-2 软件包状态标识

	状 态	说 明
第 1 个字符(软件包的 6 种请求状态)	u	状态未知，意味着软件包未安装，并且用户也未发出安装请求
	i	用户请求安装软件包
	r	用户请求卸载软件包
	p	用户请求清除软件包
	h	用户请求保持软件包版本锁定
第 2 个字符（软件包的当前的 6 种状态）	n	软件包未安装
	i	软件包安装并完成配置
	c	软件包以前安装过，现在删除了，但是它的配置文件还留在系统中
	u	软件包被解包，但还未配置
	f	试图配置软件包，但是失败了
	h	软件包安装，但是没有成功
第 3 个字符（标识软件包的 4 种错误状态）	空字符	表示当前软件包没有问题
	h	软件包被强制保留，因为有其他软件包依赖当前软件包，无法升级
	r	软件包被破坏，可能需要重新安装才能正常使用（包括删除）
	x	软件包被破坏，并且被强制保留

实例 2 查看 lsof 软件包的详细信息，执行如下命令。

```
debian5:~$ dpkg -s lsof
```

执行命令后的结果如图 17-20 所示。

实例 3 列出 gpe-icons_0.25-1_all.deb 软件包文件中包含的文件信息，执行如下命令。

```
debian5:~$ dpkg -c gpe-icons_0.25-1_all.deb
```

执行命令后的结果如图 17-21 所示。

```
debian5:~#
debian5:~# dpkg -s lsof
Package: lsof
Status: install ok installed
Priority: standard
Section: utils
Installed-Size: 420
Maintainer: Norbert Tretkowski <nobse@debian.org>
Architecture: i386
Version: 4.78.dfsg.1-4
Replaces: lsof-2.0.35, lsof-2.0.36, lsof-2.0.38, lsof-2.2
(<< 4.73)
Depends: libc6 (>= 2.7-1)
Conflicts: suidmanager (<< 0.50)
Description: List open files
 Lsof is a Unix-specific diagnostic tool.  Its name stands
 for LiSt Open Files, and it does just that.  It lists
 information about any files that are open, by processes
 currently running on the system.
Homepage: http://people.freebsd.org/~abe/
debian5:~#
```

图 17-20 查看 lsof 软件包的详细信息

```
debian5:~# dpkg -c gpe-icons_0.25-1_all.deb
drwxr-xr-x root/root         0 2008-01-16 10:19 ./
drwxr-xr-x root/root         0 2008-01-16 10:19 ./usr/
drwxr-xr-x root/root         0 2008-01-16 10:19 ./usr/shar
e/
drwxr-xr-x root/root         0 2008-01-16 10:19 ./usr/shar
e/gpe/
drwxr-xr-x root/root         0 2008-01-16 10:19 ./usr/shar
e/gpe/pixmaps/
drwxr-xr-x root/root         0 2008-01-16 10:19 ./usr/shar
e/gpe/pixmaps/default/
-rw-r--r-- root/root       265 2008-01-16 10:19 ./usr/shar
e/gpe/pixmaps/default/close.png
-rw-r--r-- root/root       341 2008-01-16 10:19 ./usr/shar
e/gpe/pixmaps/default/copy.png
-rw-r--r-- root/root       496 2008-01-16 10:19 ./usr/shar
e/gpe/pixmaps/default/cut.png
-rw-r--r-- root/root       507 2008-01-16 10:19 ./usr/shar
e/gpe/pixmaps/default/exit.png
-rw-r--r-- root/root       292 2008-01-16 10:19 ./usr/shar
e/gpe/pixmaps/default/new.png
-rw-r--r-- root/root       448 2008-01-16 10:19 ./usr/shar
e/gpe/pixmaps/default/open.png
-rw-r--r-- root/root       563 2008-01-16 10:19 ./usr/shar
e/gpe/pixmaps/default/paste.png
-rw-r--r-- root/root       595 2008-01-16 10:19 ./usr/shar
e/gpe/pixmaps/default/save.png
-rw-r--r-- root/root       751 2008-01-16 10:19 ./usr/shar
e/gpe/pixmaps/default/save_as.png
```

图 17-21 列出软件包文件中包含的文件信息

实例 4 列出系统中已安装的软件包 lsof 中包含的文件信息，执行如下命令。

```
debian5:~$ dpkg -L lsof
```

或

```
debian5:~$ dpkg --listfiles lsof
```

执行命令后的结果如图 17-22 所示。

实例 5 查找系统中的文件/bin/ls 所属的软件包的信息，执行如下命令。

```
debian5:~$ dpkg -S /bin/ls
```

或

```
debian5:~$ dpkg --search /bin/ls
```

执行命令后的结果如图 17-23 所示。

实例 6 显示软件包文件 gpe-icons_0.25-1_all.deb 的包名、版本以及软件包不同部分的大小等相关信息，执行如下命令。

```
debian5:~$ dpkg -I gpe-icons_0.25-1_all.deb
```

或

```
debian5:~$ dpkg --info gpe-icons_0.25-1_all.deb
```

执行命令后的结果如图 17-24 所示。

```
debian5:~# dpkg -L lsof
/.
/usr
/usr/share
/usr/share/doc
/usr/share/doc/lsof
/usr/share/doc/lsof/changelog.Debian.gz
/usr/share/doc/lsof/README.Debian-source
/usr/share/doc/lsof/copyright
/usr/share/doc/lsof/examples
/usr/share/doc/lsof/examples/shared.perl5.gz
/usr/share/doc/lsof/examples/sort_res.perl5
/usr/share/doc/lsof/examples/idrlogin.perl5.gz
/usr/share/doc/lsof/examples/list_fields.awk.gz
/usr/share/doc/lsof/examples/identd.perl5
/usr/share/doc/lsof/examples/list_NULf.perl5.gz
/usr/share/doc/lsof/examples/big_brother.perl5.gz
/usr/share/doc/lsof/examples/xusers.awk
/usr/share/doc/lsof/examples/00MANIFEST
/usr/share/doc/lsof/examples/idrlogin.perl.gz
/usr/share/doc/lsof/examples/watch_a_file.perl
/usr/share/doc/lsof/examples/count_pf.perl
/usr/share/doc/lsof/examples/count_pf.perl5
/usr/share/doc/lsof/examples/list_fields.perl.gz
/usr/share/doc/lsof/examples/00README
/usr/share/doc/lsof/00FAQ.gz
/usr/share/doc/lsof/README.Debian
/usr/share/doc/lsof/00DIST.gz
/usr/share/man
```

图 17-22 列出 lsof 软件包中包含的文件信息

```
debian5:~#
debian5:~# dpkg -S /bin/ls
coreutils: /bin/ls
debian5:~# dpkg --search /bin/ls
coreutils: /bin/ls
debian5:~#
```

图 17-23 查找文件/bin/ls 所属的软件包的信息

```
debian5:~#
debian5:~# dpkg -I gpe-icons_0.25-1_all.deb
 new debian package, version 2.0.
 size 44632 bytes: control archive= 2546 bytes.
     354 bytes,    11 lines      control
    5881 bytes,    78 lines      md5sums
 Package: gpe-icons
 Version: 0.25-1
 Architecture: all
 Maintainer: Neil Williams <codehelp@debian.org>
 Installed-Size: 348
 Section: x11
 Priority: optional
 Homepage: http://gpe.linuxtogo.org/projects/
 Description: Common icons used by GPE programs
  This package contains the default icon theme used by the
 core applications of
  the GPE Palmtop Environment.
debian5:~#
```

图 17-24 显示软件包文件的相关信息

第 18 章　软件包高级管理

在使用命令安装各类软件包时，有一些软件的安装需要花很长时间，并且过程也比较复杂，期间可能还会出现版本冲突、安装成功率低等问题。为了解决软件安装、卸载出现的一系列问题，可以通过软件包管理工具来安装各种软件包。前一章已经介绍了几款基础的软件包管理工具，本章将继续介绍功能更为强大的软件包管理工具。

18.1　APT——高级软件包管理

用户在安装系统中的各种软件包的时候，经常会遇到一些问题，迫切需要一种快速、实用、高效的管理工具，在软件包更新时，能够自动管理关联文件并维护已有的配置文件。因此，Debian 推出了 APT（Advanced Packaging Tool），随后 Conectiva 软件公司改造了 APT 用以管理 RPM 软件包。之后又被其他 Linux 发行版本采用，用来管理其相应的软件包。

18.1.1　APT 概述

Debian GNU/Linux，Ubuntu 等管理系统都采用集中式的软件仓库管理机制，将软件包按类别存放在软件仓库中，进行有效的组织和管理。软件仓库放置在许多镜像服务器中，保持基本一致，这些镜像服务器作为用户的软件源（Repository）。对于所有的 Debian GNU/Linux，Ubuntu，用户都可以随时获得最新版本的安装软件包。

APT 使用软件源配置文件/etc/apt/sources.list，列出可获得软件包的镜像站点地址。无论 Debian GNU/Linux，Ubuntu 等系统用户所处的网络环境差别有多大，连网用户都可以使用 APT 工具轻松地从软件源配置文件 sources.list 指定的站点获得最新的软件包列表。

APT 的基本工作机制如下。

- 服务器端

首先在 APT 镜像服务器站点（APT Repository 1，APT Repository 1，…APT Repository n）上放置所有的 deb 软件包套件（包含原版的 DVD 光盘和供应商发行的升级套件）。同时，以特定的软件功能分析各个 deb 文件的依赖性，并将这些分析资料以文件的形式存放在服务器的特定目录中，用以解决软件包的依赖性问题。

- 客户端

如果客户端需要升级某个软件，首先应下载服务器上存放的记载软件包依赖信息的文件，再将其与客户端系统中软件包的安装情况进行比对分析，获得所需的所有相关套件的信息，然后一次性将相关软件包全部下载到本地进行升级安装。

即使这样，软件源配置文件 sources.list 也只是告知用户可以访问的软件资源所在的镜像站点地址，但对于相同的软件资源在哪个镜像站点下载最迅捷这个问题无法确定。

此时，若每安装一个软件包，用户就要在所有镜像服务器上寻找一遍，效率显然很低。

通过评估 ping time 和 hops number(一个网络请求报文到达目标主机所经过的转发主机的个数)的方式可以解决该问题。这两项评估值与镜像站点预计下载速度显然成反比，因此可以选择评估值最小的镜像站点地址放置到 sources.list 文件中。

APT 工具能够检查 Debian GNU/Linux，Ubuntu 等系统中的软件包依赖关系，大大简化了 Debain/Ubuntu 用户安装和卸载软件包的操作。因此，APT 成为 Debian GNU/Linux，Ubuntu 等系统中最受欢迎的管理工具，也成为其他软件包管理工具的底层工具。比如，图形界面的软件包管理工具 Synaptic 就是在 APT 基础上工作的。准确来说，APT 是一组命令集，包括以下常用的 APT 命令。

- apt-get：用于管理软件包，包括安装、卸载、升级等操作。
- apt-cache：用于查询软件包信息。
- apt-proxy：用于搭建 APT 代理服务器。
- apt-show-versions：用于显示系统中软件包的版本信息。
- apt-config：用于读取 APT 配置文件的简单工具。
- apt-cdrom：用于将 CD-ROM 加入软件源配置文件。

18.1.2 应用 APT 软件包管理系统

APT 用于获取、安装、编译、卸载和查询 deb 软件包，另外还可用于检查软件包依赖关系。作为 Debian GNU/Linux、Ubuntu 等系统中功能最强大的软件包管理工具，APT 很值得用户掌握。本节将结合实例阐述 APT 软件包管理器的使用方法，包括配置 APT 软件源、更新 APT 软件包数据库、查找 APT 软件包信息、管理 APT 软件包等操作。

1. 配置 APT 软件源

在安装 Debian GNU/Linux、Ubuntu 等系统时，系统会根据用户所选择的国家和时区，向用户推荐合适的软件源地址。通常，初级用户不需要修改这些配置，只需要使用默认的配置文件即可。对于高级系统管理员而言，建议寻找更合适的软件源镜像站点，并重新设置 APT 源。

APT 软件源配置文件/etc/apt/sources.list 是一个文本文件，系统管理员可以使用 vi 编辑器进行编辑。在该文件中，添加的软件源镜像站点称为一个配置项，每个配置项都必须遵循以下格式。

```
DebType AddressType://Hostaddress/debian Distribution Section1 Section2…
```

各字段含义的描述如下。

- DebType 表示 deb 软件包类型，使用 deb 表示二进制软件包，使用 deb-src 表示源码包。
- AddressType 表示访问地址类型，常用类型有：http、ftp、file、cdrom、ssh 等。
- Distribution 表示 Debian 的各个发行版本。
- Section 表示节。

在 Ubuntu 环境下，软件源配置文件/etc/apt/sources.list 的配置项格式如下。

```
DebType AddressType://Hostaddress/Ubuntu Distribution Component1 Component2 …
```

各字段的含义描述如下。

- DebType 表示 deb 软件包类型，使用 deb 表示二进制软件包，使用 deb-src 表示源码包。
- AddressType 表示访问地址类型，常用类型有：http、ftp、file、cdrom、ssh 等。
- Distribution 表示 Ubuntu 的各个发行版本，例如 dapper、feisty。
- Component 表示软件包组件的类别，是由技术支持程度不同而划分的类别，可选择 main、restricted、universe 和 multiverse 中的一种或多种。

实例 1 将 debian.cn99.com 作为软件源镜像站点。

```
toplinux@debian:~$vi /etc/apt/source.list
```

通过 vi 编辑器编辑/etc/apt/sources.list 文件，并添加如下配置信息。

```
deb  http://debian.cn99.com/debian  etch  main  contrib  non-free
deb-src  http://debian.cn99.com/debian  etch  main  contrib  non-free
```

以上配置信息告知 APT 工具，可以从该地址中获得 Debian 的 etch 版本的软件资源，包括二进制软件包（deb）和源码包（deb-src），并指明可以使用所有类别的软件包。需要注意的是，在每次修改完/etc/apt/sources.list 文件后，一定要运行“apt-get update”命令，更改操作才会生效。

2. 更新 APT 软件包数据库

因为 Debian 软件包所在软件源中现有的软件包列表时常发生变化，所以需要提示 APT 下载这些列表并且时常更新它的数据库。通过执行如下指令实现。

```
toplinux@debian:~$apt-get update
```

通常，在安装新的软件包之前运行此命令，这样就不会出现下载旧版软件包的问题。同样，在升级前也要运行该指令。

3. 查找 APT 软件包信息

软件源的镜像站点上存放着不计其数的软件包资源，系统中也安装了大量软件包，那么应当如何准确、快速地检索到相关软件包信息呢？这些信息对 APT 软件包的安装、升级和卸载都非常重要。下面将重点介绍一些查找 APT 软件包信息的方法和技巧，这些方法和技巧与 apt-cache 命令的灵活应用息息相关。

使用 apt-cache 命令可以查询软件源和软件包的相关信息。apt-cache 命令的一般语法格式如下。

```
toplinux@debian:~$apt-cache  subcommands  [-chipqs]  pkg
```

各子命令的作用描述如下。

showpkg：获取二进制软件包的常规描述信息。

showsrc：获取源码包的详细描述信息。

show：获取二进制软件包的详细描述信息。

stats：获取软件源的基本统计信息。

dump：获取软件源中所有软件包的简要信息。
dumpavail：获取当前已安装的所有软件包的描述信息。
unmet：获取所有未满足的依赖关系。
search：根据正则表达式检索软件包。
depends：获取该软件包的依赖信息。
rdepends：获取所有依赖该软件包的软件包。
pkgnames：列出所有已安装软件包的名字。
policy：获取软件包当前的安装状态。
各参数含义的描述如下。
-p：软件包的缓存。
-s：源代码包的缓存。
-q：关闭进度获取。
-i：获取重要的依赖关系，仅与 unmet 命令一起使用。
-c：读取指定配置文件。
-h：获取帮助信息。

dpkg 命令将当前 Debian GNU/Linux，Ubuntu 等系统中已安装的软件包信息全部记录在/var/lib/dpkg/available 文件中；而 APT 命令会扫描软件源中每一个镜像站点服务器，为可用的软件包资源建立索引文件，并存放在本地的/var/lib/apt/lists/目录中。

实例 1 获取软件包的统计信息。

首先使用“apt-cache pkgnames”命令获得目前系统中所有的已安装的软件包，然后用 wc 命令完成统计。

```
toplinux@debian:~$apt-cache pkgnames | wc -l
3501
```

如果用户认为上面的统计结果还是不具体，希望得到更细致的软件包统计数据，可以使用如下命令。

```
toplinux@debian:~$ apt-cache stats
软件包名称总数：3501 (140k)
  普通软件包：1394
  完全虚拟软件包：56
  单虚拟软件包：290
  混合虚拟软件包：23
  缺漏的：1738
按版本共计：1510 (78.5k)
按不同的说明共计：1510 (36.2k)
按依赖关系共计：12580 (352k)
按版本/文件关系共计：2256 (36.1k)
按说明/文件关系共计：1510 (24.2k)
提供映射共计：569 (11.4k)
Glob 字串共计：41 (391)
依赖关系版本名所占空间共计：61.8k
Slack 空间共计：57.9k
总占用空间：677k
```

命令执行后可以看到，执行“apt-cache stats”命令得到的信息，显然要比执行前面的组合命令得到的结果丰富得多。该命令能够按照普通软件包、完全虚拟软件包、单虚拟软件包、混合虚拟软件包等类别，并分别对软件包的数量和大小进行统计。

实例 2 按关键字 dhcp 检索软件包。

用户查询软件包，总会有记忆得不够准确、不够完整的时候。使用如下命令能够按照关键字检索软件包信息，帮助用户快速找到软件包。

```
toplinux@debian:~$ apt-cache search dhcp
pump - BOOTP and DHCP client for automatic IP configuration
libnm-glib0 - network management framework (GLib shared library)
avahi-autoipd - Avahi IPv4LL network address configuration daemon
dhcp-client - DHCP client transitional package
dhcp3-client - DHCP client
dhcp3-common - common files used by all the dhcp3* packages
dhcp3-dev - API for accessing and modifying the DHCP server and client state
dhcp3-relay - DHCP relay daemon
dhcp3-server - DHCP server for automatic IP address assignment
dhcp3-server-ldap - DHCP server able to use LDAP as backend
toplinux@debian:~$
```

注意 如果查询关键词过于普通，可能会返回很多无关的信息。建议用户使用多个关键词组合，以帮助提高查询精度。

使用“apt-cache search”命令检索到用户关心的软件包，之后就可以使用如下命令进一步查询指定软件包的详细信息，包括软件包的安装状态、优先级、适用架构、版本、存在依赖关系的软件包以及功能描述等。

```
toplinux@debian:~$ apt-cache show dhcp-client
Package: dhcp-client
Priority: extra
Section: net
Installed-Size: 28
Maintainer: Andrew Pollock <apollock@debian.org>
Architecture: all
Source: dhcp3
Version: 3.1.1-6+lenny2
Depends: dhcp3-client
Conflicts: dhcp-client (<< 3.0)
Filename: pool/updates/main/d/dhcp3/dhcp-client_3.1.1-6+lenny2_all.deb
Size: 22976
MD5sum: 0216788c7652496df9d297d3df2a81e7
SHA1: 048ba9dac24d97b833b028d1c750f0deb2f890b7
SHA256: ba1b69cc756ebda7aa7fcc8e065b3dc9507bc64a25ae8ddbfe604420611f5fe8
Description: DHCP client transitional package
 This is the client from version 3 of the Internet Software
 Consortium's implementation of DHCP. For more information visit
```

```
    http://www.isc.org.
    .
    This is a dummy package to aid in transitioning from the v2 dhcp-client package
    to the v3 dhcp3-client package.
    .
    This dummy package may be safely removed after upgrading to lenny
    Tag:  interface::commandline,  network::client,  network::configuration,
protocol::dhcp, protocol::ip, role::dummy, role::program, use::configuring

    toplinux@debian:~$
```

实例 3　显示 dhcp-client 软件包的常规信息。

在软件安装过程中，用户关注更多的是软件包版本和软件包的依赖关系。因此，用户可以使用“apt-cahce showpkg”命令来获取软件包的常规信息。

```
    toplinux@debian:~$ apt-cache showpkg dhcp-client
    Package: dhcp-client
    Versions:
    3.1.1-6+lenny2
( /var/lib/apt/lists/security.debian.org_dists_lenny_updates_main_binary-i386_
Packages)
     Description Language:
                  File:
/var/lib/apt/lists/security.debian.org_dists_lenny_updates_main_binary-i386_P
ackages
                  MD5: ba483d73637f20bc56e3bb4ebc442835

    Reverse Depends:
      dhcp3-client,dhcp-client 3.0
      dhcp-client,dhcp-client 3.0
      bpalogin,dhcp-client
      ifupdown,dhcp-client
      dhcp3-client,dhcp-client 3.0
    Dependencies:
    3.1.1-6+lenny2 - dhcp3-client (0 (null)) dhcp-client (3 3.0)
    Provides:
    3.1.1-6+lenny2 -
    Reverse Provides:
    pump 0.8.24-4
    toplinux@debian:~$
```

该命令用于获取 dhcp-client 软件包的常规信息，其中 Dependencies 表示存在依赖关系的软件包，Reverse Depends 表示存在反向依赖关系的软件包，Provides 表示存在提供关系的软件包，Reverse Provides 表示存在反向提供关系的软件包。

实例 4　显示 dhcp-client 软件包的安装状态。

在系统使用过程中，用户可能需要随时了解一个软件包在当前系统中的安装状态。此时，

可以使用如下指令来获取指定软件包当前的安装状态，功能与“dpkg -l”命令基本相同，不过前者具有较好的可读性。

```
toplinux@debian:~$ apt-cache policy dhcp-client
dhcp-client:
  已安装：（无）
  候选的软件包：3.1.1-6+lenny2
  版本列表：
     3.1.1-6+lenny2 0
        500 http://security.debian.org lenny/updates/main Packages
toplinux@debian:~$
```

注意 用户可以使用“dpkg -r”指令来卸载指定软件包，软件删除后将保留软件的版本列表等配置信息；使用“dpkg -P”指令可以完全清除指定软件包，相关的配置信息也将全部被删除。

实例 5 显示 dhcp-client 软件包依赖于哪些软件包，反向依赖于哪些包。

如果仅想了解某个软件包依赖于哪些软件包，可以使用如下指令。

```
root@debian:/# apt-cache depends dhcp-client
dhcp-client
  依赖：dhcp3-client
  冲突：dhcp-client
root@debian:/#
```

该指令的查询的结果显示了 dhcp-client 软件包所依赖的软件包和有冲突关系的软件包。

如果用户仅想了解某个软件包被哪些软件包所依赖，可以使用如下指令。

```
root@debian:/# apt-cache rdepends dhcp-client
dhcp-client
Reverse Depends:
  dhcp3-client
  dhcp-client
 |bpalogin
    pump
  ifupdown
    pump
  dhcp3-client
root@debian:/#
```

4. 管理 APT 软件包

在 Debian GNU/Linux，Ubuntu 等中，使用“apt-get”命令便可以完成管理软件包的任务，包括获取、安装、编译和卸载软件包，以及检查软件包依赖关系，“apt-get”命令的语法格式如下。

```
toplinux@debian:~$apt-get subcommands [-bdfhmqsuvy][--purge][--reinstall]
pkg
```

各子命令的含义描述如下。

update：更新软件包列表信息。

upgrade：将系统中所有的软件包升级到最新的版本。

install：下载所需软件包并进行安装配置。

remove：卸载软件包。

autoremove：将不满足依赖关系的软件包自动卸载。

source：下载源码包。

build-dep：源码包构建所需的编译环境。

dist-upgrade：发布版升级。

dselect-upgrade：根据 dselect 的选择来进行软件包升级。

clean：删除缓存区中所有已下载的包文件。

autoclean：删除缓存区中老版本的已下载的包文件。

check：检查系统中依赖关系的完整性。

各参数的含义描述如下。

-d：仅下载软件包，而不安装或解压。

-f：修复系统中存在的软件包依赖性问题。

-m：当发现缺少关联软件包时，仍试图继续执行。

-q：将输出作为日志保留，不获取命令执行进度。

--purge：与 remove 子命令一起使用，完全卸载软件包。

--reinstall：与 install 子命令一起使用，重新安装软件包。

-b：在下载完源码包后，编译生成相应的软件包。

-s：不做实际操作，只是模拟命令执行结果。

-y：对所有询问都做肯定的回答，apt-get 不再进行任何提示。

-u：获取已升级的软件包列表。

-h：获取帮助信息。

-v：获取 apt-get 版本号。

apt-get 具有很强大的功能，熟练掌握子命令、选项的用法，并进行巧妙的组合，可以完成几乎所有的管理任务。

实例 1 安装 Samba 软件包。

在准备好软件源并连通网络后，执行如下命令就可以轻松完成 Samba 的安装过程，而无需考虑软件包的版本、优先级、依赖关系等。

```
root@debian:/# apt-get install samba
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
建议安装的软件包：
  smbldap-tools ldb-tools
下列【新】软件包将被安装：
  samba
共升级了 0 个软件包，新安装了 1 个软件包，要卸载 0 个软件包，有 0 个软件未被升级。
需要下载 0B/4296kB 的软件包。
```

```
解压缩后会消耗掉 11.9MB 的额外空间。
【警告】: 下列的软件包不能通过验证!
  samba
不经验证就安装这些软件包么? [y/N] y
正在预设定软件包 ...
选中了曾被取消选择的软件包 samba。
(正在读取数据库 ... 系统当前总共安装有 73026 个文件和目录。)
正在解压缩 samba (从 .../samba_2%3a3.2.5-4lenny6_i386.deb) ...
正在处理用于 man-db 的触发器...
正在设置 samba (2:3.2.5-4lenny6) ...
Starting Samba daemons: nmbd smbd.
root@debian:/#
```

从以上命令执行结果中可以看到 Samba 软件的整个安装过程。用户输入安装软件的名称，默认情况下“apt-get install”将会安装最新版本的 Samba 软件，并且检查 Samba 软件包的依赖关系树。。

在默认情况下，“apt-get install”只安装依赖关系的软件包。在回答了“是否继续下载？”的提示后，开始下载软件包。下载结束后，这些软件包会被自动解压，并按照依赖关系的前后顺序，依次完成安装和配置。

“apt-get install”下载软件包的步骤大体分为以下 4 步。

第 1 步，扫描本地存放的软件包更新列表（由“apt-get update”命令刷新更新列表），找到最新版本的软件包。

第 2 步，进行软件包依赖关系检查，找到支持该软件正常运行的所有软件包。

第 3 步，从软件源所指的镜像站点中，下载相关软件包。

第 4 步，解压软件包，并自动完成应用程序的安装和配置。

实例 2 重新安装 Samba 软件包。

当用户不小心损坏了已安装的软件包，而需要修复或者希望重新安装软件包中某些文件的最新版本时，可以使用如下命令进行软件包的重新安装。

```
root@debian:/# apt-get --reinstall install samba
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
共升级了 0 个软件包，新安装了 0 个软件包，共重新安装了 1 个软件包，要卸载 0 个软件包，有 0 个软件未被升级。
需要下载 0B/4296kB 的软件包。
解压缩后会消耗掉 0B 的额外空间。
您希望继续执行吗? [Y/n]y
【警告】: 下列的软件包不能通过验证!
  samba
不经验证就安装这些软件包么? [y/N] y
正在预设定软件包 ...
(正在读取数据库 ... 系统当前总共安装有 73083 个文件和目录。)
正在预备替换 samba 2:3.2.5-4lenny6 (使用 .../samba_2%3a3.2.5-4lenny6_i386.deb) ...
Stopping Samba daemons: nmbd smbd.
```

```
正在解压缩将用于更替的包文件 samba ...
正在处理用于 man-db 的触发器...
正在设置 samba (2:3.2.5-4lenny6) ...
Starting Samba daemons: nmbd smbd.
root@debian:/#
```

实例 3 更新软件包。

一次性将系统中所有的软件包升级到最新版本，可以执行如下指令。

```
root@debian:/# apt-get upgrade
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
共升级了 0 个软件包，新安装了 0 个软件包，要卸载 0 个软件包，有 0 个软件未被升级。
root@debian:/#
```

由命令执行结果可知，当前系统中所有的软件都已经是最新版本。

实例 4 卸载 Samba 软件包。

不完全卸载是指卸载某个软件包及其关联软件包时，依然保留这些软件包在系统中的配置信息。执行如下指令完成不完全卸载 Samba 软件包。

```
root@debian:/# apt-get remove samba
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
下列软件包将被【卸载】:
  samba swat
共升级了 0 个软件包，新安装了 0 个软件包，要卸载 2 个软件包，有 0 个软件未被升级。
解压缩后将会空出 14.8MB 的空间。
您希望继续执行吗? [Y/n]y
(正在读取数据库 ... 系统当前总共安装有 73136 个文件和目录。)
正在删除 swat ...
正在删除 samba ...
Stopping Samba daemons: nmbd smbd.
正在处理用于 man-db 的触发器...
root@debian:/#
```

由于是不完全删除，Samba 以及与之有依赖关系的软件包便会被删除，但配置文件会保留在系统里。

完全卸载即彻底删除所有相关软件包及其配置文件。执行如下指令完成完全卸载 Samba 软件包。

```
root@debian:/# apt-get --purge remove samba
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
下列软件包将被【卸载】:
  samba*
共升级了 0 个软件包，新安装了 0 个软件包，要卸载 1 个软件包，有 0 个软件未被升级。
```

```
解压缩后将会空出 11.9MB 的空间。
您希望继续执行吗? [Y/n]y
(正在读取数据库 ... 系统当前总共安装有 73083 个文件和目录。)
正在删除 samba ...
Stopping Samba daemons: nmbd smbd.
正在清除 samba 的配置文件 ...
Removing configuration file /etc/default/samba...
Removing configuration file /etc/default/samba...
正在处理用于 man-db 的触发器...
root@debian:/#
```

实例 5 修复 SSH 软件包的依赖关系。

按照如下指令安装 SSH 软件包，期间模拟一个系统故障中断软件的安装过程，可能会造成关联的软件包只有部分完成安装的情况。之后，用户就会发现该软件既不能重装又不能删除。

```
root@debian:/# apt-get install ssh
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
现在没有可用的软件包 ssh，但是它被其他的软件包引用了。
这可能意味着这个缺失的软件包可能已被废弃，
或者只能在其他发布源中找到
可是下列的软件包取代了它：
  openssh-server openssh-client
E: 软件包 ssh 还没有可供安装的候选者
```

接下来使用“apt-get check”命令对软件包的依赖关系进行检查，然后执行如下指令，修复因系统故障导致软件包依赖关系出现错误的问题。

```
root@debian:/# apt-get install samba
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
E: 无法找到软件包 samba
root@debian:/# apt-get check
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
root@debian:/# apt-get -f install
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
共升级了 0 个软件包，新安装了 0 个软件包，要卸载 0 个软件包，有 0 个软件未被升级。
root@debian:/#
```

实例 6 下载网络流量分析软件 wireshark 的源码包。

在 Debian\Ubuntu Linux 中，使用“apt-get source”命令获取源码包。只需告知源码包的名字，该命令就可主动下载全部源码包文件。通常，源码包中包含 3 个文件，分别以 dsc、orig.tar.gz

和 diff.gz 为后缀名。“apt-get source”会将源码包下载到用户当前所在目录，并在命令执行过程中，调用 dpkg-source 命令，根据 dsc 文件中的信息，将源码包解压到同名目录中，应用程序的源代码就存放在这里。

下载网络流量分析软件 wireshark 的源码包到当前目录，命令执行如下。

```
root@debian:/# apt-get source wireshark
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
需要下载 17.0MB 的源代码包。
获取:1 http://security.debian.org lenny/updates/main wireshark 1.0.2-3+lenny5
(dsc) [1501B]
获取:2 http://security.debian.org lenny/updates/main wireshark 1.0.2-3+lenny5
(tar) [16.9MB]
获取:3 http://security.debian.org lenny/updates/main wireshark 1.0.2-3+lenny5
(diff) [102kB]
下载 17.0MB，耗时 8min12s (34.6kB/s)
gpg: 新的配置文件‘//.gnupg/gpg.conf’已建立
gpg: 警告: 在‘//.gnupg/gpg.conf’里的选项于此次运行期间未被使用
gpg: 于 2009 年 04 月 27 日 星期一 03 时 25 分 15 秒 CST 创建的签名，使用 DSA，钥匙号
4E2ECA5A .
gpg: 无法检查签名: 找不到公钥
dpkg-source: extracting wireshark in wireshark-1.0.2
dpkg-source: info: unpacking wireshark_1.0.2.orig.tar.gz
dpkg-source: info: applying wireshark_1.0.2-3+lenny5.diff.gz
root@debian:/#
```

软件包下载后，新建 wireshark-1.0.2 目录，将源码包解压到 wireshark-1.0.2 中。

注意 在下载网络流量分析软件 wireshark 的源码包前，必须确保当前系统中已经安装了 dpkg-dev 软件包（若尚未安装该软件包可以执行指令“apt-get install dpkg-dev”进行安装）。否则，系统只会下载源码包的 3 个文件，而不解压源码包。

18.2 YUM——智能化的 RPM 软件包管理

Red Hat/fedora Linux 系统用户大都遇到过 RPM 软件包依赖关系无法解决的问题，给工作带来诸多不便。YUM 软件包管理工具的出现，很好地解决 RPM 软件包依赖关系的问题。YUM 精简的指令，强大的类 APT 软件包管理功能，获得广大 Linux 用户的好评。

18.2.1 YUM 概述

YUM（Yellow dog Updater Modified）最初是由 yellow dog 的开发者 Terra Soft 研发，使用 python 语言编写而成，命名为 yup（yellow dog updater）。该项目后来由杜克大学的 Linux@Duke 开发团队改进并改名为 YUM。

YUM 使用一个中心软件仓库（repository）管理一部分或者整个 Linux 发行版的应用程序间的相互关系，并根据计算出来的软件依赖关系进行相关的升级、安装、删除等操作，减少了软件包依赖（dependencies）问题。

YUM 的中心软件仓库（repository）既可以是 HTTP 或 FTP 站点，也可以是本地软件池，但必须包含 RPM 的 header。header 中包括 RPM 包的各种信息，如描述、功能、提供的文件和依赖性等。YUM 收集了 header 并加以分析，然后自动升级、安装、移除 RPM 包，收集 RPM 包的相关信息，检查依赖性并自动提示用户解决。

YUM 具有如下特点。

- 自动解决 RPM 软件包的依赖性问题，能更方便地添加/删除/更新 RPM 包。
- 便于管理大量系统的更新问题。
- 可以同时配置多个中心软件仓库（Repository）。
- 配置文件简洁（/etc/yum.conf）。
- 保持与 RPM 数据库的一致性。
- 拥有相当多的软件源维护组织。
- 有一个比较详细的日志（log），可以查看何时升级、安装了什么软件包等。
- 使用方便，界面简单。

目前，Red Hat/Fedora Linux 系统自带有 YUM 功能，因此可以使用 Red Hat、CentOS 等官方软件源，从而完成官方发布的各种升级。对于第三方软件源，YUM 能够支持大多数 APT 的软件源，例如 freshrpms、fedora.us、livna、dag 等。

18.2.2 应用 YUM 软件包管理系统

YUM 是一款智能化的 RPM 软件包管理工具。YUM 的基本操作包括软件的安装、升级和卸载。此外它还有一定的查询功能，还可以在基于 RPM 包的系统上实现安装、更新、删除软件包及相关依赖包等多项功能，还可以自动计算软件包之间的依赖关系，并指出在安装该软件包时可能发生的状况。用户不需要使用 rpm 命令手动更新每一个软件包，方便了多台计算机的维护工作。

1. 配置 YUM 软件源

reposdir（/etc/yum.repos.d）目录下以“.repo”为扩展名的一系列文件列出了可获得 YUM 软件包仓库的镜像站点地址。以 CentOS 5.3 系统为例，默认情况下有如下两个文件。

CentOS-Base.repo ：用于设置远程仓库。

CentOS-Media.repo ：用于设置本地仓库。

在以“.repo”为扩展名的文件中，配置语法是分段的，每一段语法配置一个软件仓库，配置语法格式如下。

```
[repositoryid]
name=Some name for this repository
baseurl=url://server1/path/to/repository/
        url://server2/path/to/repository/
        url://server3/path/to/repository/
```

```
mirrorlist=url://path/to/mirrorlist/repository/
enabled=0/1
gpgcheck=0/1
gpgkey=A URL pointing to the ASCII-armoured GPG key file for the repository
```

各命令含义的描述如下。

- repositoryid：用于指定一个软件仓库。
- name：用于指定易读的仓库名称。
- baseurl：用于指定该软件仓库的 URL，可以是如下的 3 种类型。
  http 用于指定远程 HTTP 协议的源。
  ftp 用于指定远程 FTP 协议的源。
  file 用于本地镜像或 NFS 挂装文件系统。
- mirrorlist ：用于指定仓库的镜像站点。
- enabled ：用于指定是否使用本仓库，默认值为 1，即可用。
- gpgcheck ：用于指定是否检查软件包的 GPG 签名。
- gpgkey ：用于指定 GPG 签名文件的 URL。

注意 在 name 和 baseurl 部分经常使用如下的变量。
$releasever — 当前系统的版本号。
$basearch — 当前系统的平台架构。
文件中以“#”开头的行是注释行。
若指定 mirrorlist，系统将从 CentOS 的镜像站点中选择最近的软件仓库。
CentOS 的镜像站点列表中并非包含所有的国内镜像，所以可以直接使用 baseurl 来指定最近的软件仓库。baseurl 可以指定多个 URL，系统会依次检查列出的软件仓库，以便在某个服务器发生故障时可以使用另外的服务器。为了加快更新，在确保更新服务器时线路良好的情况下，建议在 baseurl 中只指定一个 URL。

实例 1 将站点 centos.ustc.edu.cn 设置为软件源。

```
[root@localhost yum.repos.d]# vi /etc/yum.repos.d/CentOS-Base.repo
```

首先备份原有的/etc/yum.repos.d/CentOS-Base.repo 文件，然后通过 vi 编辑器编辑/etc/yum.repos.d/CentOS-Base.repo 文件，并添加如下配置项信息。

```
[base]
name=CentOS-$releasever - Base
baseurl=http://centos.ustc.edu.cn/centos/$releasever/os/$basearch/
gpgcheck=1
gpgkey=http://mirror.centos.org/centos/RPM-GPG-KEY-CentOS-5
[updates]
name=CentOS-$releasever - Updates
baseurl=http://centos.ustc.edu.cn/centos/$releasever/updates/$basearch/
gpgcheck=1
gpgkey=http://mirror.centos.org/centos/RPM-GPG-KEY-CentOS-5
[addons]
```

```
name=CentOS-$releasever - Addons
baseurl=http://centos.ustc.edu.cn/centos/$releasever/addons/$basearch/
gpgcheck=1
gpgkey=http://mirror.centos.org/centos/RPM-GPG-KEY-CentOS-5
[extras]
name=CentOS-$releasever - Extras
baseurl=http://centos.ustc.edu.cn/centos/$releasever/extras/$basearch/
gpgcheck=1
gpgkey=http://mirror.centos.org/centos/RPM-GPG-KEY-CentOS-5
[centosplus]
name=CentOS-$releasever - Plus
baseurl=http://centos.ustc.edu.cn/centos/$releasever/centosplus/$basearch/
gpgcheck=1
enabled=0
gpgkey=http://mirror.centos.org/centos/RPM-GPG-KEY-CentOS-5
```

以上配置信息告知 YUM 工具，可以从该地址中获得 CentOS 的 5.3 版本的软件资源。

实例 2 设置使用安装光盘作为更新源。

为了使用安装光盘作为更新源，可以修改 CentOS-Media.repo 文件。

```
[c5-media]
name=CentOS-$releasever - Media
baseurl= file:///media/CentOS_5.3_Final/CentOS/
        file:///media/CentOS/
        file:///media/cdrom/
        file:///media/cdrecorder/
gpgcheck=1
enabled=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-beta
```

为了使用 c5-media 仓库，需将 CentOS-Base.repo 文件中的 base 仓库设置成 enabled=0，表示 base 仓库不可用；将 CentOS-Media.repo 文件中的 c5-media 仓库设置成 enabled=1，表示可用。

同时需要将 CentOS 5.3 的 DVD 安装光盘插入光驱，使其自动加载，该配置文件中 baseurl 的地址一定要和 DVD 安装光盘的加载路径保持一致。

2. 使用 YUM 管理 RPM 软件包

在 CentOS/Red Hat/Fedora Linux 中，使用“yum”命令便可以完成管理软件包的任务。包括从中心软件仓库安装新软件、从独立的 RPM 软件包中安装新软件、直接升级系统中已经安装的软件、卸载系统中已安装的软件等。yum 命令的语法格式如下。

```
[root@localhost ~]# yum subcommands [<packages>|<groupnames>]
```

各子命令含义的描述如下。

install <packages>：安装新软件包。

groupinstall <groupnames>：安装指定软件组中的软件包。

localinstall <.rpm files>：安装指定软件包文件中的软件包。

update <packages>：更新指定的软件包。

groupupdate <groupnames>：更新指定软件组中的软件包。

remove <packages>：卸载指定的软件包。

groupremove <groupnames>：卸载指定软件组中的软件包。

使用“yum”命令安装软件包或者软件包集的步骤，大体分为 3 步：第 1 步，进行软件包依赖关系检查，找到支持该软件正常运行的所有软件包；第 2 步，从软件源所指的镜像站点下载相关软件包；第 3 步，解压软件包，并自动完成应用程序的安装和配置。

实例 1 安装终端服务客户端 tsclient 软件包。

在准备好软件源并连通网络后，执行如下命令就可以轻松完成终端服务客户端软件 tsclient 的安装，而无需考虑软件包的版本、优先级、依赖关系等。

```
[root@localhost ~]# yum install tsclient
Loading "rhnplugin" plugin
Loading "security" plugin
This system is not registered with RHN.
RHN support will be disabled.
Setting up Install Process
Parsing package install arguments
Resolving Dependencies
--> Running transaction check
---> Package tsclient.i386 0:0.148-6.el5 set to be updated
--> Processing Dependency: rdesktop >= 1.3.0 for package: tsclient
--> Running transaction check
---> Package rdesktop.i386 0:1.4.1-4 set to be updated
--> Finished Dependency Resolution

Dependencies Resolved

================================================================
 Package          Arch       Version          Repository     Size
================================================================
Installing:
 tsclient         i386       0.148-6.el5      server rpm     326 k
Installing for dependencies:
 rdesktop         i386       1.4.1-4          server rpm     117 k

Transaction Summary
================================================================
Install       2 Package (s)
Update        0 Package (s)
Remove        0 Package (s)

Total download size: 443 k
Is this ok [y/N]: y
Downloading Packages:
Running rpm_check_debug
Running Transaction Test
Finished Transaction Test
Transaction Test Succeeded
```

```
Running Transaction
  Installing: rdesktop                    ########################### [1/2]
  Installing: tsclient                    ########################### [2/2]

Installed: tsclient.i386 0:0.148-6.el5
Dependency Installed: rdesktop.i386 0:1.4.1-4
Complete!
[root@localhost ~]#
```

从以上命令执行结果可以看到 tsclient 软件的整个安装过程。用户输入安装软件的名称，“yum”命令将会安装最新版本的 tsclient 软件，同时检查 tsclient 软件包的依赖关系树。此时会发现与 tsclient 存在依赖关系的软件包有 rdesktop。默认情况下，“yum”命令只安装依赖关系的软件包。在回答“是否下载？”的提示后，开始下载软件包，下载过程可能需要一段时间。下载结束后，这些软件包会被自动解压，并按照依赖关系的先、后顺序，依次完成安装和配置。

实例 2 更新终端服务客户端 tsclient 软件包。

为了将系统中已经安装的 tsclient 软件包升级到最新版本，可以执行如下指令。

```
[root@localhost ~]# yum update tsclient
Loading "rhnplugin" plugin
Loading "security" plugin
This system is not registered with RHN.
RHN support will be disabled.
Skipping security plugin, no data
Setting up Update Process
Could not find update match for tsclient
No Packages marked for Update
[root@localhost ~]#
```

如果当前的 tsclient 软件包已经是最新的版本，系统便不再升级该软件包。

实例 3 卸载终端服务客户端 tsclient 软件包。

此处的卸载是指卸载某个软件包及其关联软件包。执行如下指令完成卸载 tsclient 软件包的工作。

```
[root@localhost ~]# yum remove tsclient
Loading "rhnplugin" plugin
Loading "security" plugin
This system is not registered with RHN.
RHN support will be disabled.
Setting up Remove Process
Resolving Dependencies
--> Running transaction check
---> Package tsclient.i386 0:0.148-6.el5 set to be erased
--> Finished Dependency Resolution

Dependencies Resolved

================================================================
 Package                Arch      Version          Repository      Size
```

```
================================================================
Removing:
 tsclient                i386       0.148-6.el5      installed        526 k

Transaction Summary
================================================================
Install       0 Package (s)
Update        0 Package (s)
Remove        1 Package (s)

Is this ok [y/N]: y
Downloading Packages:
Running rpm_check_debug
Running Transaction Test
Finished Transaction Test
Transaction Test Succeeded
Running Transaction
  Erasing    : tsclient                        ########################## [1/1]

Removed: tsclient.i386 0:0.148-6.el5
Complete!
[root@localhost ~]#
```

3. 使用 YUM 查询 RPM 软件包

使用 YUM 的查找功能可以从已配置的软件仓库中寻找可用的软件包，或者查看当前系统中已经安装的软件包。其语法格式如下。

```
[root@localhost ~]# yum subcommands
```

各子命令含义的描述如下。

grouplist：查看系统中已经安装的和可用的软件组。

list：列出资源库中所有可以安装或更新以及已经安装的 RPM 包。

list<regex>：列出资源库中与正则表达式匹配的可以安装或更新的以及已经安装了的 RPM 包。

list available：列出资源库中所有可以安装的 RPM 包。

list available <regex>：列出资源库中与正则表达式匹配的所有可以安装的 RPM 包。

list updates：列出资源库中所有可以更新的 RPM 包。

list updates <regex>：列出资源库中与正则表达式匹配的所有可以更新的 RPM 包。

list installed：列出资源库中所有已经安装的 RPM 包。

list installed <regex>：列出资源库中与正则表达式匹配的所有已经安装的 RPM 包。

list extras：列出已经安装的但是不包含在资源库中的 RPM 包。

list extras <regex>：列出与正则表达式匹配的已经安装的但是不包含在资源库中的 RPM 包。

list recent：列出最近被添加到资源库中的软件包。

deplist <packages>：显示软件包的依赖信息。

search <regex>：检测所有可用的软件的名称、描述、概述和已列出的维护者，查找与正则表达式匹配的值。

provides <regex>：检测软件包中包含的文件以及软件提供的功能，查找与正则表达式匹配的值。

info：显示资源库中所有可以安装或更新的以及已经安装了的 RPM 包的信息。

info <regex>：显示资源库中与正则表达式匹配的可以安装或更新以及已经安装的 RPM 包的信息。

info available：显示资源库中所有可以安装的 RPM 包的信息。

info available <regex>：显示资源库中与正则表达式匹配的所有可以安装的 RPM 包的信息。

info updates：显示资源库中所有可以更新的 RPM 包的信息。

info updates <regex>：显示资源库中与正则表达式匹配的所有可以更新的 RPM 包的信息。

info installed：显示资源库中所有已经安装了的 RPM 包的信息。

info installed <regex>：显示资源库中与正则表达式匹配的所有已经安装了的 RPM 包的信息。

info extras：显示已经安装了的但是不包含在资源库中的 RPM 包的信息。

info extras <regex>：显示与正则表达式匹配的已经安装了的但是不包含在资源库中的 RPM 包的信息。

info recent：显示最近被添加到资源库中的软件包的信息。

实例 1 查看软件仓库中的 RPM 软件包 tsclient 的情况。

在安装一款软件之前总希望了解当前配置的软件仓库中是否存在该软件包、当前系统中是否已安装该软件包，以及安装的版本等一系列信息，可以通过执行如下指令实现。

```
[root@localhost ~]# yum list tsclient
Loading "rhnplugin" plugin
Loading "security" plugin
This system is not registered with RHN.
RHN support will be disabled.
Available Packages
tsclient.i386                              0.148-6.el5                server rpm
[root@localhost ~]#
```

由命令执行结果可以看出，当前系统中已经安装了一款 tsclient 软件，版本号为 0.148。

```
[root@localhost ~]# yum info tsclient
Loading "rhnplugin" plugin
Loading "security" plugin
This system is not registered with RHN.
RHN support will be disabled.
Available Packages
Name   : tsclient
Arch   : i386
Version: 0.148
Release: 6.el5
Size   : 326 k
```

```
Repo    : server rpm
Summary: VNC 和窗口终端服务器的客户
Description:
tsclient 是一个简化对 rdesktop 和 vncviewer 的使用的前端。
除 tsclient 软件包之外，您将需要安装 rdesktop 或 vnc 软件包。
 [root@localhost ~]#
```

实例 2 查看系统中已安装了的和可用的软件包集的情况。

为了解当前系统已经安装了的软件包集和当前配置的软件仓库中提供的可用的软件包集的情况，可以通过执行如下指令实现。

```
[root@localhost ~]# yum grouplist
Loading "rhnplugin" plugin
Loading "security" plugin
This system is not registered with RHN.
RHN support will be disabled.
Setting up Group Process
Installed Groups:
   Office/Productivity
   MySQL Database
   Development Libraries
   Editors
   System Tools
   GNOME Software Development
   Text-based Internet
   X Software Development
   Legacy Network Server
   DNS Name Server
   GNOME Desktop Environment
   Authoring and Publishing
   FTP Server
   Network Servers
   Games and Entertainment
   Legacy Software Development
   Java Development
   Legacy Software Support
   Windows File Server
   X Window System
   Graphics
   Web Server
   Printing Support
   Mail Server
   Server Configuration Tools
   Sound and Video
   PostgreSQL Database
   Administration Tools
   Graphical Internet
Available Groups:
```

```
   Engineering and Scientific
   OpenFabrics Enterprise Distribution
   KDE Software Development
   KDE (K Desktop Environment)
   News Server
   Development Tools
Done
[root@localhost ~]#
```

从命令执行结果可以看到，当前系统中已经安装的所有软件包集的信息，以及当前配置的软件仓库中提供的可用的软件包集的情况。掌握了这些信息后，便可以按照需求定制安装需要的软件包集，或者删除不必要的软件包集。

实例 3 查看有关 Editors 的所有软件包的信息。

当想了解某些 RPM 软件包的信息，但又不知道具体的软件包名，只有相关的片段信息的时候，可以通过使用如下的指令查找。

```
[root@localhost ~]# yum search Editors
Loading "rhnplugin" plugin
Loading "security" plugin
This system is not registered with RHN.
RHN support will be disabled.
vim-common.i386 : The common files needed by any version of the VIM editor.
emacspeak.noarch : emacspeak -- The Complete Audio Desktop
vim-minimal.i386 : A minimal version of the VIM editor.
eclipse-pde.i386 : Eclipse PDE
emacs.i386 : GNU Emacs 文本编辑器
eclipse-pde-runtime.i386 : Eclipse PDE runtime plugin.
emacs-leim.i386 : 用于国际字符输入法的 Emacs Lisp 文件。
nano.i386 : A small text editor
gedit.i386 : gEdit is a small but powerful text editor for GNOME
kdewebdev.i386 : WEB Development package for the K Desktop Environment.
vim-enhanced.i386 : A version of the VIM editor which includes recent
enhancements.
emacs-common.i386 : Emacs 公共文件。
eclipse-jdt.i386 : Eclipse Java development tools
libswt3-gtk2.i386 : SWT Library for GTK2
emacs.i386 : GNU Emacs text editor
ctags.i386 : A C programming language indexing and/or cross-reference tool.
joe.i386 : An easy to use, modeless text editor.
vim-common.i386 : 被所有 VIM 编辑器版本需要的公用文件。
less.i386 : A text file browser similar to more, but better.
......
```

命令执行后，显示当前配置软件仓库中提供的可用的软件包中有关 Editors 的软件包，分别给出每个包的基本信息。

实例 4 查看提供了 MTA（Mail Transport Agent）服务的所有软件包。

如果想了解提供特定服务的 RPM 软件包的情况，可以通过使用如下的指令查找。

```
[root@localhost ~]# yum provides MTA
Loading "rhnplugin" plugin
Loading "security" plugin
This system is not registered with RHN.
RHN support will be disabled.
exim.i386 : The exim mail transfer agent
postfix.i386 : Postfix Mail Transport Agent
postfix.i386 : Postfix 邮件传输代理
[root@localhost ~]#
```

命令执行后，可以了解当前配置的软件仓库中提供的 MTA 服务的软件包信息。

4. 使用 YUM 管理缓存资源

使用 YUM 可以帮助清除缓存中的 RPM 包文件、RPM 头文件、旧的 RPM 头文件。其语法格式如下。

```
[root@localhost ~]# yum subcommands
```

各子命令含义的描述如下。

clean 或 clean all：清除缓存中的 RPM 头文件和包文件。

clean packages：清除缓存中的 RPM 包文件。

clean headers：清除缓存中的 RPM 头文件。

clean oldheaders：清除缓存中旧的 RPM 头文件。

实例 清除缓存中的 RPM 文件和头文件。

系统经过一段时间的使用后，由于不断安装一些新软件包，卸载一些软件包或者升级一些软件包，缓存中便会积累许多的 RPM 文件和 RPM 头文件，若不定期进行清理，会消耗服务器的资源。可以执行如下指令定期清理缓存中的 RPM 文件和头文件。

```
[root@localhost ~]# yum clean all
Loading "rhnplugin" plugin
Loading "security" plugin
Cleaning up Everything
[root@localhost ~]#
```

在执行“yum clean all”指令后，将清除在 YUM 缓存中 RPM 文件和头文件的目录/var/cache/yum/下的缓存数据。

18.3 Linux 内核的升级

Linux 内核最早是由芬兰黑客 Linus Torvalds 为尝试在英特尔 x86 架构上提供自由免费的类 UNIX 操作系统而开发的，用 C 语言写成，符合 POSIX 标准的类 UNIX 操作系统。Linux 项目从 1991 年正式开始，从最初的一些 Minix 黑客提供协助，发展为全球最著名的开源软件项目。

从技术实质来说，Linux 仅是一个内核，并不是一套完整的操作系统。“内核”指的是一个提供硬件抽象层、磁盘及文件系统控制、多任务等功能的系统软件。一套基于 Linux 内核而

构建起来的覆盖众多系统管理方面功能的完整操作系统才叫 Linux 操作系统，或是 GNU/Linux。

今天，Linux 是一个单内核（monolithic kernel）系统。设备驱动程序可以访问硬件。Linux 内的设备驱动程序可以方便地以模块化（modularize）的形式设置，并在系统运行期间可直接装载或卸载。

在 Linux 内核中，包括了进程管理（process management）、定时器（timer）、中断管理（interrupt management）、内存管理（memory management）、模块管理（module management）、虚拟文件系统接口（VFS layer）、文件系统（file system）、设备驱动程序（device driver）、进程间通信（inter-process communication）、网络管理（network management）以及系统启动（system init）等操作系统功能的实现。

18.3.1 获取内核源代码

Linux 的内核目前由其发明者 Linus Torvalds 所属团队负责维护，官方站点网址为 www.kernel.org，如图 18-1 所示。用户可以从该网站了解到最新的 Linux kernel 信息。

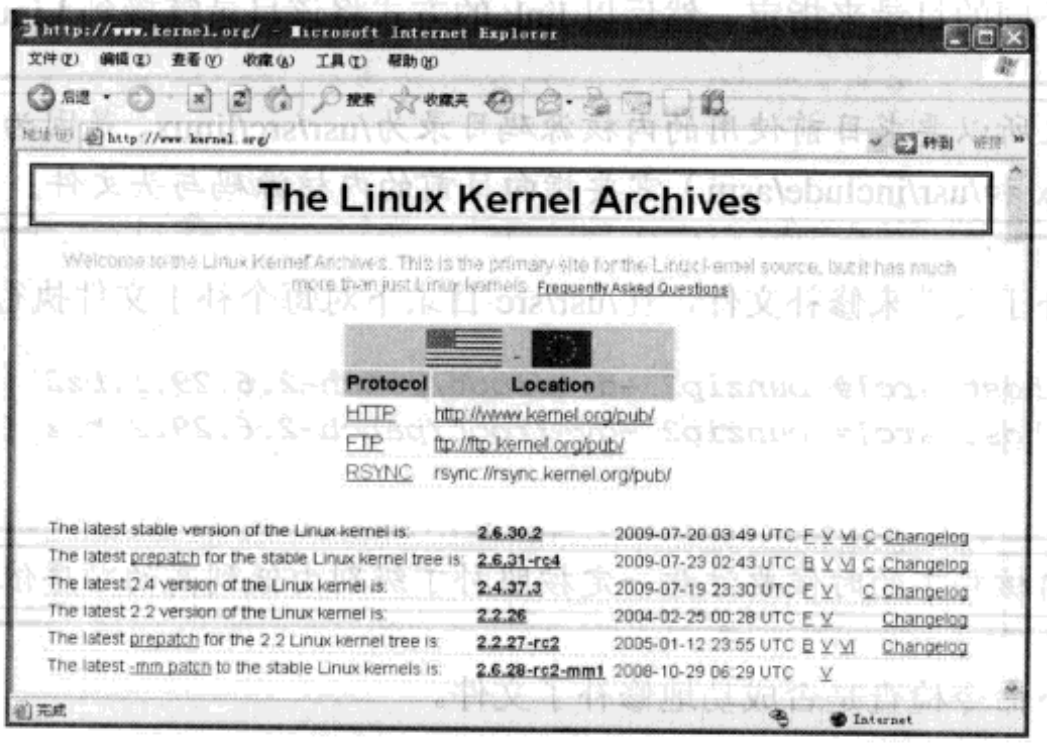


图 18-1 Linux 内核官方站点

正式的内核版本是以 gzip 工具压缩打包后的 tar.bz2 格式文件的形式推出的，其中包含了内核的源代码。而后续的一系列补丁文件（用 diff 产生）也会以 gzip 工具压缩打包后的 tar.bz2 格式文件的形式推出。

假设内核需要升级到版本 2.6.29 patchlevel 2，那么需要下载文件名为 linux-2.6.29.tar.bz2 的 2.6.29 版的源码，以及文件名为 patch-2.6.29.1.bz2~patch-2.6.29.2.bz2 之间的所有补丁文件，参考指令如下。

```
[root@localhost ~]# wget http://www.cn.kernel.org/pub/linux/kernel/v2.6/linux-2.6.29.tar.bz2
[root@localhost ~]# wget http://www.kernel.org/pub/linux/kernel/v2.6/patch-2.6.29.1.bz2
```

```
[root@localhost ~]# wget
http://www.kernel.org/pub/linux/kernel/v2.6/patch-2.6.29.2.bz2
```

18.3.2 配置内核

当准备好了升级内核所必需的源代码和补丁文件之后，就可通过如下的步骤完成内核配置工作，为编译内核做好充分的准备。

第 1 步，解压源码包文件并升级系统补丁。

首先执行如下指令在/usr/src 目录下解压 linux-2.6.29.tar.bz2 源码文件。

```
[root@localhost ~]# cd /usr/src
[root@localhost src]# mv linux linux.old
[root@localhost src]# tar -jxvf linux-2.6.29.tar.bz2
[root@localhost src]# ln -s linux-2.6.29 linux
```

通过上述操作，系统会将整个旧版的内核源码备份到/usr/src/linux.old 目录下。另外，系统要求内核源码放置的地点一定要在/usr/src/linux 目录下，因此建议采用上述做法对每一版的内核源码都使用不同的目录来指定，然后以 link 的方式将该目录链接到 Linux 目录。

注意 系统之所以要求目前使用的内核源码目录为/usr/src/linux，是因为有两个符号链接（/usr/include/linux 和/usr/include/asm）需要指向目前的内核源码与头文件。

接下来使用补丁文件来修补文件，在/usr/src 目录下对每个补丁文件执行如下指令。

```
[root@localhost src]# bunzip2 -dv /root/patch-2.6.29.1.bz2 | path -p0
[root@localhost src]# bunzip2 -dv /root/patch-2.6.29.2.bz2 | path -p0
```

注意 在打内核补丁的时候要注意一定按照补丁级别由低到高的顺序依次执行。

最后执行如下命令检查是否成功地修补了文件。

```
[root@localhost src]# find /usr/src/linux -follow -name "*.rej" -print
[root@localhost src]# find /usr/src/linux -follow -name "*#" -print
```

在打内核补丁的过程中，无法修补的部分记录在“*.rej”文件中，管理员可以通过上述指令查看是否存在这类文件，并列出所有在修补过程中被“拒绝”的部分。如果打内核补丁的过程没有成功完成，就无法保证正确的编译内核。

第 2 步，下载并安装内核升级工具。

为了以后能够成功编译内核，请先下载并安装如下内核升级工具，参考指令如下。

```
module-init-tools-3.2.tar.bz2
( http://www.kernel.org/pub/linux/utils/kernel/module-init-tools/module-init-t
ools-3.2.tar.bz2)
mkinitrd-4.1.18-2.i386.rpm
( ftp://ftp.muug.mb.ca/mirror/fedoralegacy/fedora/3/os/i386/mkinitrd-4.1.18-2.
```

```
i386.rpm)
    lvm2-2.02.44-1.fc11.i386.rpm
( ftp://ftp.pbone.net/mirror/download.fedora.redhat.com/pub/fedora/linux/devel
opment/i386/os/Packages/lvm2-2.02.44-1.fc11.i386.rpm)
    device-mapper-1.00.19-2.i386.rpm
( ftp://ftp.muug.mb.ca/mirror/fedoralegacy/fedora/3/os/i386/device-mapper-1.00
.19-2.i386.rpm)
    [root@localhost ~]# wget http://www.kernel.org/pub/linux/utils
    /kernel/module-init-tools/module-init-tools-3.2.tar.bz2
    [root@localhost ~]# wget ftp://ftp.muug.mb.ca/mirror/fedoralegacy
    /fedora/3/os/i386/mkinitrd-4.1.18-2.i386.rpm
    [root@localhost ~]# wget
ftp://ftp.pbone.net/mirror/download.fedora.redhat.com
    /pub/fedora/linux/development/i386/os/Packages/lvm2-2.02.44-1.fc11.i386.r
pm
    [root@localhost ~]# wget ftp://ftp.muug.mb.ca/mirror/fedoralegacy
    /fedora/3/os/i386/device-mapper-1.00.19-2.i386.rpm
```

- 安装 module-init-tools 工具

在/usr/src 目录下，依次执行下列命令完成安装 module-init-tools 工具的任务。

```
[root@localhost ~]# cd /usr/src
[root@localhost src]# tar jvxf /root/module-init-tools-3.2.tar.bz2
[root@localhost src]# cd module-init-tools-3.2
[root@localhost module-init-tools-3.2 ]# ./configure --prefix=/
[root@localhost module-init-tools-3.2 ]# make moveold
[root@localhost module-init-tools-3.2 ]# make all install
[root@localhost module-init-tools-3.2 ]# ./generate-modprobe.conf
/etc/modprobe.conf
```

- 安装 mkinitrd、lvm2 和 device-mapper 工具

在/usr/src 目录下，依次执行下列命令完成安装 mkinitrd 工具的任务。

```
[root@localhost ~]# cd /usr/src
[root@localhost src]# rpm -ivh --nodeps /root/mkinitrd-4.1.18-2.i386.rpm
[root@localhost src]# rpm -ivh --nodeps /root/lvm2-2.02.44-1.fc11.i386.rpm
[root@localhost src]#
rpm -ivh --nodeps /root/device-mapper-1.00.19-2.i386.rpm
```

第 3 步，设定内核的编译模式。

到现在为止，已经打好内核源码补丁，接下来的工作便是进行内核编译模式的设定，为下一步的内核编译工作做好准备。

详细参阅/usr/src/linux/README 文件后，可以了解到目前的内核可以允许多种方式进行编译模式的设定工作，最为常用的方式包括如下 4 种。

- make config：这种模式需要逐行输入所需要设定的项目，无法往前修改设定，所以目前很少使用这种方式来设定 Kernel 的项目。
- make xconfig：这种模式则是以 X-Window（Qt）为主要环境的设定模式。除非你有

X-Window（Qt）在执行，否则没有办法使用这种模式。

- make gconfig：这种模式则是以 X-Window（Gtk）为主要环境的设定模式。除非有 X-Window（Gtk）在执行，否则没有办法使用这种模式。
- make menuconfig：这种模式有点像文字界面的菜单模式，较为简便，而且可以在纯文字界面下执行编辑动作，是目前最为常用的模式。

大多数用户习惯以字符界面的方式操作Linux来完成各项工作，在此就重点讲解使用make menuconfig 指令来完成内核的编译模式设定工作，如图 18-2 所示。在执行该指令之前，建议先执行 make mrproper 指令，清除先前内核编译过程中生成的垃圾数据。

```
[root@localhost linux]# make mrproper
[root@localhost linux]# make menuconfig
```

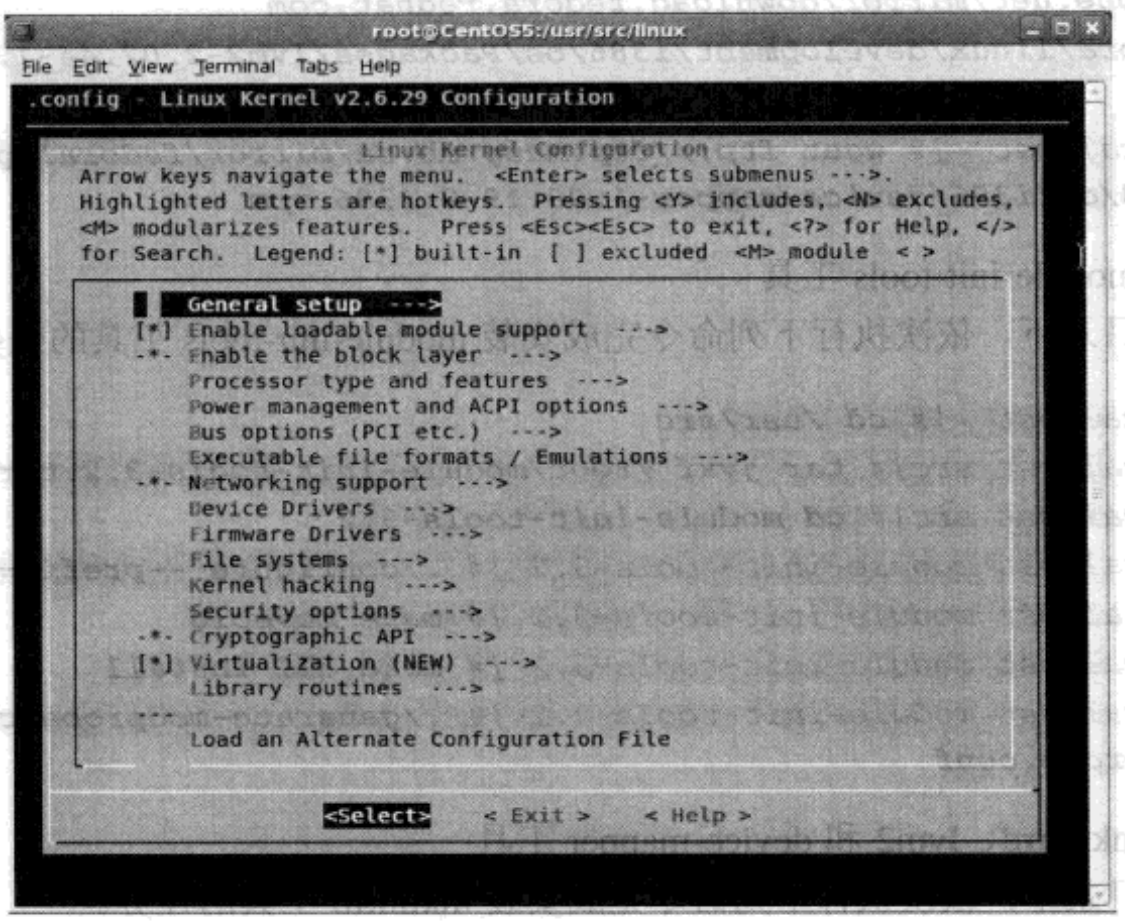


图 18-2　设定内核的编译模式

此时会出现一个设定内核的编译模式的图形交互界面，列出了所有的内核配置选项，有的选项下还有子选项，用户可以用方向键来选择，用“Y”键来确定。

大多数选项选择默认即可，但需要认真核对每一个选项，否则编译很有可能出错。至于每个选项的含义，请参考相应的在线文档。

（1）Enable loadable Module support 选项中，选中“Module unloading”和“Automatic kernel module loading”这两项。

（2）Device Drivers→Block Devices→选中“Loopback device support”；

Device Drivers→Multi-device support（RAID and LVM）→选中“device mapper support”；

Device Drivers→Graphics support→选中“Support for frame buffer devices”；

Device Drivers→USB support→选中“USB Mass Storage support”；

Device Drivers → Network device support → Ethernet （10 or 100Mbit） → 选中“AMD PCnet32 PCI support”；

（3） File system→（以下 9 个选项是关于 Ext2 和 Ext3 文件的系统配置，全部选中）

Second extended fs support

Ext2 extended attributes

Ext2 POSIX Access Control Lists

Ext2 Security Labels

Ext3 journalling file system support

Ext3 extended attributes

Ext3 POSIX Access Control Lists

Ext3 Security Labels

JBB （ext3） debugging support

File system→DOS/FAT/NT Filesystems→选中“NTFS file system support”。

注意 Ext2 和 Ext3 文件系统配置很重要，也是必需的，如果把对 Ext3 和 Ext2 文件的支持直接编译进内核，重启系统时，可能会出现如下错误信息。

```
kernel panic : no init found ,try passing init = option to kernel....
```

或者是:

```
VFS:Cannot open root device "hdxy" or unknow-block(0,0)
Please append a correct "root=" boot option
kernel panic:VFS:Unable to mount root fs on unknown-block(0,0)
```

或者是:

```
mount: error 19 mounting ext3
pivotroot: pivot_root(/sysroot,/sysroot/initrd) failed: 2
umount /initrd/proc fail: 2
Freeing unused kernel memory: 244k freed
Kernel panic - not syncing: No init found. Try passing init = option to k
ernel
```

（4） 如果在 vmware 下编译内核，硬盘用的是 scsi，以下 3 个选项必选。

```
Device Drivers →<*>SCSI device support  (此项不选，下面两项就选择不了)
Device Drivers →<*>SCSI device support →<*>SCSI disk support
Device Drivers →<8>SCSI device support→
            SCSI low-level drivers→<*> BusLogic SCSI support
```

内核编译模式配置选项编辑完成之后，选择“Save Configuration to an Alternate File”菜单项，将编辑好的信息保存。然后单击“<Exit>”按钮退出编辑系统。接下来就准备进行内核编译工作。

18.3.3 编译内核

开始编译之前需要先确认本次编译的内核与上一次编译的内核的版本是否有差异。执行

如下指令确定当前内核版本。

```
[root@localhost src]# uname -r
```

注意 因为编译成模块的数据会被系统安装在“/lib/modules/‘内核版本号’”目录下，所以，如果这次安装的内核版本跟上一次相同，那么模块放置的地点就会一样，这样就会造成模块版本冲突的问题。这时就需要先将旧版内核的模块移动到其他地方。另外，编译出来的内核文件名称为 vmlinuz，正确地存放位置是目录/boot 下，此外，系统映像数据 System.map 也放置在目录/boot 下。

编译前的确认工作完毕，接下来就可以正式进行编译工作了，依次执行如下指令即可。

```
[root@localhost src]# make dep
[root@localhost src]# make clean
[root@localhost src]# make bzIamge
[root@localhost src]# make modules
[root@localhost src]# make modules_install
[root@localhost src]# make install
```

各指令的作用简单描述如下。

make dep：实际上，读取内核配置过程中生成的配置文件可以创建对应于配置的依赖关系树，从而决定哪些模块需要编译，而哪些模块不需要编译。

make clean：完成删除之前编译步骤中产生的所有目标文件，以避免出现错误。

make bzImage：该指令主要用于产生可开机引导的内核文件 vmlinuz。但是由于目前的内核文件都比较大，必须要进行压缩，所以编译并压缩后的内核文件放置在目录/usr/src/linux/arch/i386/boot/下，命名为 bzImage。

make modules 和 make modules_install 分别生成相应的模块并把模块复制到需要的目录下，也就是目录“/lib/modules/内核版本号”下。例如当前编译的内核版本是 2.6.29，所以就安装在目录/lib/modules/2.6.29 下。

注意 执行 make install 指令的时候可能会出现如下错误信息。

```
No module BusLogic found for kernel 2.4.12
        mkinitrd failed
```

此问题一般只出现在 SCSI 硬盘 + VMWARE+REDHAT 架构中，因为 BusLogic 被编译进了内核而不是一个 module 的形式。

解决方案是直接将 BusLogic.o 文件复制过去，依次执行如下指令。

```
[root@localhost src]# cp /usr/src/linux-2.6.29/drivers/scsi/BusLogic.o
                /lib/modules/2.6.29/kernel/drivers/scsi
[root@localhost src]# make install
```

如果不出什么差错，至此，内核编译工作就完成了，接下来准备安装内核。编译过程一定要非常注意，如果出现了错误，就不得不从配置内核阶段开始，重新再来一遍，直到内核编

译成功为止。

18.3.4 安装内核

完成上面一个阶段的工作仅是将模块安装到目录“/lib/modules/内核版本号”下，同时内核编译完成并生成内核文件 bzImage，放置在目录/usr/src/linux/arch/i386/boot/下。此时，将新内核文件和 System.map 文件放置在开机内核预定义的目录/boot 下，依次执行以下命令。

```
[root@localhost src]# cp /usr/src/linux/arch/i386/boot/bzIamge
/boot/vmLinuz-2.6.29
[root@localhost src]# cp /usr/src/linux/System.map /boot/System.map-2.6.29
```

通常，直接以硬盘开机的方式来测试新的内核，（很多时候，由于不同的 distribution 所考虑的内容不一样，编译完成的内核可能无法在当前 Linux distribution 上工作，所以还需要测试内核是否运行正常）建议继续保留旧的内核，如果新内核无法正常工作，还可以继续使用旧的内核管理 Linux 系统。

接下来进行内核开机配置的工作，以 GRUB 为例，编辑文件/boot/grub/menu.lst，按照如下模式修改。

```
[root@localhost src]# vi /boot/grub/menu.lst
# grub.conf generated by anaconda
#
# Note that you do not have to rerun grub after making changes to this file
# NOTICE:  You have a /boot partition.  This means that
#          all kernel and initrd paths are relative to /boot/, eg.
#          root  (hd0,0)
#          kernel /vmlinuz-version ro root=/dev/VolGroup00/LogVol00
#          initrd /initrd-version.img
#boot=/dev/sda
timeout 10
color black/cyan yellow/cyan
i18n  (hd0,0) /boot/grub/messages
keytable  (hd0,0) /boot/us.klt
altconfigfile  (hd0,0) /boot/grub/menu.once
default=0
timeout=5
splashimage= (hd0,0) /grub/splash.xpm.gz
title CentOS (2.6.29-2.el5)
    root  (hd0,0)
   kernel /boot/vmlinuz-2.6.29 ro root=/dev/VolGroup00/LogVol00 rhgb quiet
title CentOS  (2.6.18-128.el5)
    root  (hd0,0)
    kernel  /vmlinuz-2.6.18-128.el5  ro  root=/dev/VolGroup00/LogVol00  rhgb
quiet
    initrd /initrd-2.6.18-128.el5.img
```

注意　将 default=1 改为 default=0；将 kernel 行的“root=”设置成根目录所在的分区（可以通过 df 指令查找）；到此为止所有安装工作完成，请马上重新开机测试一下内核是否可以正常工作。

第 19 章　用户和进程管理

Linux 是一个多用户、多任务的操作系统，多个用户可以在同一时间使用系统资源，系统也可以同时执行多个任务。而由于多用户、多任务的特性，又使得 Linux 操作系统的复杂性明显提高，系统更容易出现问题。因此，合理有效地管理 Linux 操作系统至关重要。本章将结合 Linux 操作系统多用户、多任务的特点，具体阐述如何进行 Linux 的用户管理和进程管理。

19.1　用 户 管 理

Linux 操作系统提供了一套有效的用户账号和群组管理方案，将用户进行分组管理（管理组、用户组、虚拟用户组），使不同的用户、不同的群组拥有不同的访问和管理权限。另外，Linux 操作系统还按照不同用户对各种系统资源的不同访问权限进行了系统资源权限的划分。

Linux 操作系统中的用户一般分为 root 用户、普通用户和虚拟用户 3 大类。

- root 用户：系统管理员，系统中唯一具备所有权限的用户，可以操作任何文件并执行任何命令。该用户在安装 Linux 系统时创建，其 ID 为 0。建议用户一般不使用 root 账号登录系统，默认情况下该用户只能在本地登录，但也可以通过修改配置文件 /etc/pam.d/login 来许可 root 用户远程登录系统。
- 普通用户：该类用户是在 Linux 系统安装完成后由系统管理员创建的，能够管理自己的文件并拥有 root 用户赋予的权限，可以直接或远程登录系统。
- 虚拟用户：该类用户不具备登录系统的能力，但却是系统运行中不可缺少的一部分，如 bin、damon、ftp、adm 等。虚拟用户可以是系统安装时产生的，也可以是管理员添加的。

19.1.1　用户账号文件

Linux 操作系统允许多用户同时使用，为了使 Linux 系统内核能够唯一识别各个用户，系统赋予每个用户一个唯一识别的标号，即每个用户以一个 UID（User ID）来识别。为了便于管理用户，Linux 系统提供了两个配置文件/etc/passwd 和/etc/shadow。

1.　配置文件/etc/passwd

Red Hat Enterprise Linux 5 中存放用户账号信息的配置文件是/etc/passwd。这是一个文本文件，其中每个用户账号对应一行，用户账号的所有相关属性以冒号分隔。用户账号的属性描述如下。

```
[root@localhost ~]# cat /etc/passwd
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
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/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
nobody:x:99:99:Nobody:/:/sbin/nologin
rpm:x:37:37::/var/lib/rpm:/sbin/nologin
dbus:x:81:81:System message bus:/:/sbin/nologin
avahi:x:70:70:Avahi daemon:/:/sbin/nologin
mailnull:x:47:47::/var/spool/mqueue:/sbin/nologin
......
```

- 用户名（Username）

位于第 1 列，如 root、games 和 mail 等都是用户账号的用户名。

需要注意的是，Linux 中除了用于系统管理的 root 账号和新建的用户账号外，还有很多其他的账号，比如 rpm、daemon、mail、news 等都是为支持系统中的特定服务而专门设置的账号。

- 口令（encrypt password）

位于第 2 列，Linux 系统中所有用户的口令都经过加密，然后以密文形式存放到配置文件/etc/passwd 中。因为所有用户都可以读取这个文件，所以其中的用户口令不是很安全。只有早期的 Linux 的用户密码才保存在/etc/passwd 文件中，新版的 Linux 系统的用户密码都保存在/etc/shadow 文件中，因此在文件/etc/passwd 中存放密码信息的位置显示的是“x”。

- UID（User ID）

位于第 3 列，Linux 系统中任何一个用户都有唯一的一个 UID，这是用户最重要的标识之一。需要注意的是，用户 UID 的取值有一定限制，如表 19-1 所示。

表 19-1　UID 取值限制

取值范围	ID 描述
0	系统管理员 ID，如 root 账号的 UID 就是 0
1~499	保留给系统使用的 ID，基于安全的考虑，Linux 的很多系统功能或服务都是通过独有的账号启动的。比如，Red Hat Enterprise Linux 5 下的 FTP 服务默认的 UID 就是 14
500~4 294 967 295	一般用户 ID，对于 Linux 核心在 2.4 或者更低的，最大数值只能支持到 65 535

- GID(Group ID)

位于第 4 列，指定某个用户默认属于哪个群组，有关群组的信息存储在配置文件/etc/group 中。

- 用户描述(User Description)

位于第 5 列，指定用户的摘要信息，包括用户的全称等相关信息。

- 用户主目录(User's Home Dictionary)

位于第 6 列，用户登录后会直接进入的目录，一般位于/home 下相应用户名的目录中，只有 root 账号的主目录是/root。

- 登录 Shell(User's login Shell)

位于第 7 列，Linux 系统通常提供了多种 Shell，该栏目用来配置当用户登录系统时采用的 Shell，默认是/bin/bash。管理员可以在这里直接指定用户的登录 Shell，用户也可以修改自己使用的 Shell。

在对用户进行管理时，不建议采用直接编辑/etc/passwd 的方式，而是利用命令行和图形界面窗口来完成。

2. 配置文件/etc/shadow

文件/etc/shadow 是文件/etc/passwd 的影子文件，这两个文件是对应互补的关系。shadow 内容包括用户名及口令密文以及其他/etc/passwd 不能包括的信息，比如用户的有效期限等。/etc/shadow 是一个文本文件，其中每个用户账号对应一行，用户账号的所有相关属性以冒号分隔，被分隔为 9 个属性，各个属性的作用描述如下。

```
[root@localhost ~]# cat /etc/shadow
root:$1$GnJ4esrI$yZagjE9QiE7RO2h6NdrmG/:14362:0:99999:7:::
bin:*:14362:0:99999:7:::
daemon:*:14362:0:99999:7:::
adm:*:14362:0:99999:7:::
lp:*:14362:0:99999:7:::
sync:*:14362:0:99999:7:::
shutdown:*:14362:0:99999:7:::
halt:*:14362:0:99999:7:::
mail:*:14362:0:99999:7:::
news:*:14362:0:99999:7:::
uucp:*:14362:0:99999:7:::
operator:*:14362:0:99999:7:::
games:*:14362:0:99999:7:::
gopher:*:14362:0:99999:7:::
.....
```

- 用户名（Username）

位于第 1 列，指定用户账号的用户名。

- 口令密文（encrypt password）

位于第 2 列，指定经过加密的口令的密文信息（默认的加密方式是 MD5），所以用户打开该文件，在很多用户的密码栏只会看到一些密文信息。

特别值得一提的是，因为这个文件的属性是“-r--------”，并且是 root 权属的文件，也就是说这个文件只能被root用户以只读方式访问，这样Linux系统的安全性也得到了大幅度的提高。

比如，查看 shadow 文件的属性，执行指令如下。

```
[root@localhost ~]# ll /etc/shadow
-r-------- 1 root root 1344 07-07 10:15 /etc/shadow
```

由命令执行结果可知，其只能被 root 用户以只读方式访问/etc/shadow 文件。

注意 如果某个用户的密码栏以“*”或者“!”开头，那就表示该用户不能进行本地登录（只能用来运行某些服务了）。如果暂时不希望某个用户登录，也可以在这里的密码前加一个“*”或者“!”。

- 最近更改密码日期

位于第 3 列，这个字段记录了用户最近修改密码的日期，需要注意的是，它是以 1970 年 1 月 1 日作为 1，文件中记录的日期其实就是距离该日期的天数。

- 密码不可更改天数

位于第 4 列，这个字段记录了用户自从上次修改口令后，允许多少天后才能再次修改，如果设置为 0 则表示没有限制。对应于配置文件/etc/login.defs 中的“PASS_MIN_DAYS”参数，默认值为 0。

- 密码需要更改的天数

位于第 5 列，这个字段记录了用户自从上次修改口令后，经过多少天后系统需要修改口令。对应于配置文件/etc/login.defs 中的“PASS_MAX_DAYS”参数，默认值为 99 999。

- 密码失效提前通知的天数

位于第 6 列，当用户口令存在有效期限限制时，这个字段记录了用户口令有效期限到期前的警告时间。对应于配置文件/etc/login.defs 中的“PASS_WARN_AGE”参数，默认值为 7。

- 密码过期的宽限时间

位于第 7 列，如果用户密码失效了，还可以设置几天的宽限期，让用户可以使用失效的密码登录。等宽限时间到期，用户密码仍未修改，则系统将自动关闭账号，默认未设置。

- 账号失效日期

位于第 8 列，记录了用户账号失效的日期，该日期也是以 1970 年 1 月 1 日为第一天进行计时。也可以设置为 YYYY-MM-DD 的格式。

- 保留

位于第 9 列，留待以后添加新功能。

19.1.2 用户群组文件

为便于管理群组，Linux 系统提供了两个配置文件/etc/group 和/etc/gshadow。其作用与文件/etc/passwd 和/etc/gshadow 类似。

1. 配置文件/etc/group

/etc/group 配置文件中记录了用户的群组信息，其中每行表示一个用户群组，每个用户群组有 4 个用冒号分隔的属性栏目，各个属性栏目的描述如下。

```
[root@localhost ~]# cat /etc/group
root:x:0:root
bin:x:1:root,bin,daemon
daemon:x:2:root,bin,daemon
sys:x:3:root,bin,adm
adm:x:4:root,adm,daemon
```

```
tty:x:5:
disk:x:6:root
lp:x:7:daemon,lp
mem:x:8:
kmem:x:9:
wheel:x:10:root
mail:x:12:mail
news:x:13:news
uucp:x:14:uucp
```

- 用户群组名

就是用户群组的名称。

- 用户群组密码

通常情况下不用设置，因为少有需要以用户群组身份登录系统的场合。如果设置了“x”号，表示群组口令已被加密并存储到了配置文件/etc/gshadow 中。

- GID

就是用户群组的 ID。

- 用户群组的所有成员

该栏目设置了同属某个指定用户群组的所有用户账号列表，不同用户账号之间以逗号分隔。

```
daemon: x: 2: root,bin,daemon
```

表示 daemon 用户群组中有 root、bin 和 daemon 3 个用户账号。

2. 配置文件/etc/gshadow

与用户的影子密码文件相对应，文件/etc/gshadow 是群组的影子密码文件，默认采用的加密方式也是 MD5。其中每行表示一个用户群组，每个用户群组有 4 个用冒号分隔的属性栏目，各个属性栏目的描述如下。

```
[root@localhost ~]# cat /etc/gshadow
root:::root
bin:::root,bin,daemon
daemon:::root,bin,daemon
sys:::root,bin,adm
adm:::root,adm,daemon
tty:::
disk:::root
lp:::daemon,lp
mem:::
kmem:::
wheel:::root
mail:::mail
……
```

- 用户群组名称

就是用户群组的名称。

- 密码

指定经过加密的口令的密文信息（默认的加密方式是 MD5）。如果该栏目信息以“*”或

者“!”开头则表示无法进行本地登录。

- 用户群组管理员账号

指定该用户群组的管理员账号。

- 用户群组的所有成员

指定该用户群组的所有成员账号。

19.1.3 命令行方式的用户管理

本节将介绍一系列实现用户管理的命令行工具的使用方法。包括添加用户（adduser）、删除用户（userdel）、修改用户（chfn）、设置登录 Shell（chsh）、改变口令（passwd）等。

1. 查看用户信息——whoami、who、w、id、finger

查看用户信息的相关系列命令包括 whoami、who、w、id、finger 等。

查看当前用户的名称，执行如下指令。

```
[root@localhost ~]# whoami
```

查看当前登录的系统用户，执行如下指令。

```
[root@localhost ~]# who
```

查看当前登录系统用户的详细信息，执行如下指令。

```
[root@localhost ~]# w
```

查看指定用户的 UID、GID 及其所归属的用户群组等信息，执行如下指令。

```
[root@localhost ~]# id <username>
```

切换登录系统用户，执行如下指令。

```
[root@localhost ~]# su <username>
```

查看指定用户的详细信息，执行如下指令。

```
[root@localhost ~]# finger <username>
```

给指定用户发送信息，执行如下指令。

```
[root@localhost ~]# write <username>
```

向所有用户广播信息，执行如下指令。

```
[root@localhost ~]# wall <username>
```

实例 1 查看当前用户的名称，执行如下指令。

```
[root@localhost ~]# whoami
root
```

命令执行后，可知当前用户为 root。

实例 2 查看当前登录系统用户的相关信息，命令执行如下。

```
[root@localhost ~]# who
root     :0           2009-07-16 17:19
root     pts/1        2009-07-16 17:19 (:0.0)
[root@localhost ~]# w
 18:01:31 up 46 min,  2 users,  load average: 0.15, 0.03, 0.04
USER     TTY FROM              LOGIN@   IDLE   JCPU   PCPU WHAT
root     :0      -                17:19   ?xdm?  28.09s  0.36s /usr/bin/gnome-
root     pts/1    :0.0            17:19    1.00s  0.09s  0.01s w
[root@localhost ~]#
```

实例 3 查看用户 testuser 的 UID、GID 及其所归属的用户群组等信息，指令执行如下。

```
[root@localhost ~]# id testuser
uid=504(testuser) gid=504(testuser) groups=504(testuser)
[root@localhost ~]# finger testuser
Login: testuser                         Name: testuser
Directory: /home/testuser               Shell: /bin/bash
Never logged in.
No mail.
No Plan.
```

2. 添加用户——useradd

在 Linux 系统中添加用户的命令行工具有 adduser 和 useradd，这两个命令的功能完全一样。这两个指令最简单的使用方法是不带任何参数，直接添加用户。

```
[root@localhost ~]# useradd <username>
[root@localhost ~]# adduser <username>
```

命令 useradd 或 adduser 加参数选项并直接跟所添加的用户名时，系统将读取添加用户所用的配置文件/etc/login.defs 和/etc/default/useradd，然后按照配置文件/etc/login.defs 和/etc/default/useradd 中所定义的规则来添加用户。向配置文件/etc/passwd 和/etc/groups 添加用户和用户群组记录，向配置文件/etc/shadow 和/etc/gshadow 同步添加用户和用户群组的口令密文信息记录。按照/etc/add/default 中所约定的目录在系统中创建主目录，并复制/etc/skel 中的文件（包括隐藏文件）到新用户的主目录中。

useradd 的语法格式如下。

```
[root@localhost ~]# 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] [username]
[root@localhost ~]# adduser [-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] [username]
```

各参数含义的描述如下。

-c：对于新账号的说明。

-d：指定新账号每次登录时所使用的主目录。默认值为 default_home。

-e：账号终止日期。日期的指定格式为 MM/DD/YY 。

-f：账号失效宽限期。当值为 0 时账号立刻被停权；而当值为-1 时则关闭此功能，默认值为-1。

-g：群组名称或以数字作为用户登录起始群组（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。默认的群组数字为 1。

-G：定义此用户为此群组的成员。每个群组使用“,”分隔，不可以含有空格符。

-m：用户目录如不存在则自动建立。如使用-k 选项 skeleton_dir 内的文件将复制至用户目录下。然而在/etc/skel 目录下的其他文件也会被复制过去。任何在 skele-ton_dir 或/etc/skel 下的目录也会在用户目录下一一建立。

-M：不建立用户目录，即使/etc/login.defs 配置文件指定要建立用户目录。

-n：默认用户群组与用户名称相同。此选项将取消默认。

-r：此参数是用来建立系统账号的。系统账号的 UID 比定义在配置文件上/etc/login.defs 的 UID_MIN 的值还小。注意该参数不会建立用户目录，也会忽略/etc/login.defs.中的设置。如果想要有用户目录，则必须额外指定-m 参数来建立系统账号。

-s：用户登录后使用的 Shell 名称。默认为不填写，这样系统会帮用户指定默认的登录 Shell（Linux 默认配置下为 Bash）。

-u：用户的 ID 值。必须为唯一的 ID 值，除非用-o 选项。数字不可为负值。默认为不得小于 500（0~499 是保留给系统账号使用的）。

实例 1 创建一个一般用户，其名称为 newuser。

创建 newuser 用户，执行如下指令。

```
[root@localhost ~]# useradd newuser
```

查看用户主目录，指令执行如下。

```
[root@localhost ~]# ls -l /home
总计 28
drwx------ 3 newuser  newuser  4096 07-16 18:15 newuser
drwx------ 4 testuser testuser 4096 07-09 23:46 testuser
drwx------ 3     502     502 4096 05-04 16:33 testuser1
drwx------ 5 toplinux toplinux 4096 05-07 11:04 toplinux
```

查看 newuser 用户信息，指令执行如下。

```
[root@localhost ~]# grep newuser /etc/passwd /etc/shadow /etc/group
/etc/gshadow
/etc/passwd:newuser:x:505:505::/home/newuser:/bin/bash
/etc/shadow:newuser:!!:14441:0:99999:7:::
/etc/group:newuser:x:505:
/etc/gshadow:newuser:!::
```

由命令执行的过程和结果可知：创建一个新用户，最起码修改了/etc/passwd、/etc/shadow、/etc/group 和/etc/gshadow 4 个配置文件，创建的用户的初始 Shell 是/bin/bash，创建的用户的主目录为/home/newuser。

实例 2 创建一个系统账号，其名称为 administrator。

```
[root@localhost ~]# useradd -r admin
[root@localhost ~]# ls -l /home
```

```
总计 28
drwx------ 3 newuser  newuser  4096 07-16 18:15 newuser
drwx------ 4 testuser testuser 4096 07-09 23:46 testuser
drwx------ 3     502     502 4096 05-04 16:33 testuser1
drwx------ 5 toplinux toplinux 4096 05-07 11:04 toplinux
[root@localhost ~]#grep admin /etc/passwd /etc/shadow /etc/group /etc/gshadow
/etc/passwd:newadmin:x:100:103:Testlinux:/home/admin:/bin/bash
/etc/passwd:admin:x:101:104::/home/admin:/bin/bash
/etc/shadow:newadmin:!!:14362::::::
/etc/shadow:admin:!!:14441::::::
/etc/group:newadmin:x:103:
/etc/group:admin:x:104:
/etc/gshadow:newadmin:!::
/etc/gshadow:admin:!::
[root@localhost ~]#
```

由命令执行的过程和结果可知：创建了一个新系统账号，但并未创建该系统账号的主目录。另外，可以看出，创建系统账号默认的 UID 和 GID 都是系统级别的（小于 500）。

实例 3 创建一个 Linux 登录用户 test01，用户描述为“guanli01”，其主目录为/home/guanlizu，所属用户群组为 root，系统启动时的 Shell 程序为 Bash，命令执行如下。

```
[root@localhost ~]# useradd test01 -c guanli01 -d /home/guanlizu -g root -s
/bin/bash
[root@localhost ~]# ls -l /home
总计 32
drwx------ 3 test01   root     4096 07-16 19:09 guanlizu
drwx------ 3 newuser  newuser  4096 07-16 18:15 newuser
drwx------ 4 testuser testuser 4096 07-09 23:46 testuser
drwx------ 3     502     502 4096 05-04 16:33 testuser1
drwx------ 5 toplinux toplinux 4096 05-07 11:04 toplinux
[root@localhost  ~]#  grep  test01  /etc/passwd  /etc/shadow  /etc/group
/etc/gshadow
/etc/passwd:test01:x:506:0:guanli01:/home/guanlizu:/bin/bash
/etc/shadow:test01:!!:14441:0:99999:7:::
[root@localhost ~]#
```

命令执行后，用户账号 test01 的主目录是/home/guanlizu，未在配置文件/etc/group 和/etc/gshadow 中发现任何与 test01 相关的信息，这是因为用户 test01 的群组被指定为 root。

如果想知道当前系统中创建用户都采用了哪些默认配置，可以使用“useradd -D”命令来查询相关信息。

```
[root@localhost ~]# useradd -D
GROUP=100
HOME=/home
INACTIVE=-1
EXPIRE=
SHELL=/bin/bash
SKEL=/etc/skel
CREATE_MAIL_SPOOL=yes
[root@localhost ~]#
```

也可以通过修改配置文件/etc/login.defs 和/etc/default/useradd 来修改默认选项。

第 1 步，修改配置文件/etc/login.defs。

该文件在默认情况下，内容如下。

```
# *REQUIRED*
#   Directory where mailboxes reside, _or_ name of file, relative to the
#   home directory.  If you _do_ define both, MAIL_DIR takes precedence.
#   QMAIL_DIR is for Qmail
#
#QMAIL_DIR  Maildir
MAIL_DIR    /var/spool/mail ; #设置邮件保存位置
#MAIL_FILE  .mail

# Password aging controls:
#
#   PASS_MAX_DAYS   Maximum number of days a password may be used.
#   PASS_MIN_DAYS   Minimum number of days allowed between password changes.
#   PASS_MIN_LEN    Minimum acceptable password length.
#   PASS_WARN_AGE   Number of days warning given before a password expires.
#
PASS_MAX_DAYS   99999   ; #最大密码有效期(99 999)，超过此有效期，必须修改
PASS_MIN_DAYS   0       ; #最小密码有效期。超过词天数才能再次修改密码
PASS_MIN_LEN    5       ; #密码最小长度，默认设置为 5
PASS_WARN_AGE   7       ; #密码到期前 7 天通知用户

#
# Min/max values for automatic uid selection in useradd
#
UID_MIN                 500   ; #最小 UID
UID_MAX               60000   ; #最大 UID

#
# Min/max values for automatic gid selection in groupadd
#
GID_MIN                 500   ; #最小 GID
GID_MAX               60000   ; #最大 GID

#
# If defined, this command is run when removing a user.
# It should remove any at/cron/print jobs etc. owned by
# the user to be removed (passed as the first argument).
#
#USERDEL_CMD    /usr/sbin/userdel_local

#
# If useradd should create home directories for users by default
# On RH systems, we do. This option is overridden with the -m flag on
# useradd command line.
#
CREATE_HOME yes           ; #是否创建用户家目录
```

```
# The permission mask is initialized to this value. If not specified,
# the permission mask will be initialized to 022.
UMASK           077             ; #家目录权限设置

# This enables userdel to remove user groups if no members exist.
#
USERGROUPS_ENAB yes

# Use MD5 or DES to encrypt password? Red Hat use MD5 by default.
MD5_CRYPT_ENAB yes              ; #用户密码加密方式选择，Red Hat Enterprise Linux 5
                                ; #默认选择的是md5，也可以选择DES
```

第 2 步，修改配置文件/etc/defaults/useradd。

在 Linux 系统中的/etc/defaults 目录中有一个 useradd 文件，指定了系统中新建用户的默认配置选项。

```
# useradd defaults file
GROUP=100                       ; #用户群组编号
HOME=/home                      ; #用户群组主目录
INACTIVE=-1
EXPIRE=
SHELL=/bin/bash                 ; #用户群组中用户的登录 shell
SKEL=/etc/skel
CREATE_MAIL_SPOOL=yes
```

3. 修改口令——passwd

尝试以刚才新建的用户账号 test01 去登录 Linux 系统，会发现无法登录，系统处于提示用户输入登录账号/密码的状态。因为如果尚未对登录用户账号设置登录密码，则不能成功登录系统。所以用户应当先使用指令 passwd 来为新建账号设置密码，然后账号才能正常登录。

需要注意的是，不带任何参数执行 passwd 指令时会修改当前登录用户的密码，带用户名执行 passwd 命令则可以修改指定用户的密码。只有 root 管理员才可以修改其他用户的密码，而登录用户只有修改自己密码的权限。passwd 指令的语法参数如下。

```
[root@localhost ~]# passwd [-k] [-l ][-d] [-u] [-f] [-x] [-n] [-w] [-i]
                    [-S] [- -help] [- -usage] [loginuser]
```

各参数含义的描述如下。

-k：创建永不过期账户。

-l：锁定指定账号（限 root 使用）。

-d：删除指定账号的密码（限 root 使用）。

-u：解锁指定账号（限 root 使用）。

-f：强制操作。

-x：密码最长失效时间（限 root 使用）。

-n：密码最短失效时间（限 root 使用）。

-w：密码失效前提前提醒的天数（限 root 使用）。

-i：密码失效后的宽限期（限 root 使用）。

-S：查询、汇报指定用户的密码信息。

--help：获取帮助信息。

--usage：显示主要使用方法。

实例 修改 newuser 用户的密码，命令执行如下。

```
[root@localhost ~]# passwd newuser
Changing password for user newuser
New UNIX password:                                    ; #这里输入的密码不会显示
BAD PASSWORD:it is based a  dictionary word ; #建议不要使用单词作为密码
Retype new UNIX password:
passwd:  all authentication tokens updated successfully.    ; #不过用户强制使用单词作为密码它也无可奈何
[root@localhost  ~]#  grep  newuser  /etc/passwd  /etc/shadow  /etc/group /etc/gshadow
/etc/passwd:newuser:x:501:501::/home/newuser:/bin/bash
/etc/shadow:newuser:$1$tQMWv1kQ$6FXxwzsDBOGGipXuxzmrO1:14166:0:99999:7:::
/etc/group:newuser:x:501:
/etc/gshadow:newuser:!::
# 设置密码后，newuser 用户终于允许登录了。
```

注意 Linux 系统中设置的账号密码，除了要受/etc/login.defs 中的配置限制外，还要接受/etc/pam.d/passwd 这个 PAM 模块的检查。一般来说，最好符合以下 3 个条件：① 密码不要与账户相同。② 密码不要是词典内有的单词。③ 密码最好超过 8 个字符。

4. 修改用户的 finger 信息——chfn

文件/etc/passwd 中不仅包括了用户账号、密码、ID、主目录和登录 shell 等信息，还包括了一个用户详细信息(finger)，其中的某些内容可以通过 chfn 指令来修改。其语法格式如下。

```
[root@localhost ~]# chfn  [ -f full-name ] [ -o office ] [ -p office-phone ]
                  [ -h home- phone ] [ -u ] [ -v ] [ username ]
```

各参数含义的描述如下。

-f：修改全名。

-o：修改单位。

-p：修改单位电话。

-h：修改家庭电话。

-u：显示帮助信息。

-v：显示版本信息。

实例 修改账户 newadmin 的姓名为“Testlinux”，再用 finger 命令查看，具体操作步骤如下。

第 1 步，首先使用 finger 命令查看账号 newadmin 的信息。

```
[root@localhost ~]# finger newadmin
Login: newadmin                         Name: newadmin
Directory: /home/admin                  Shell: /bin/bash
```

```
Never logged in.
No mail.
No Plan.
```

由返回结果可以看出用户姓名为“newadmin”。

第 2 步，使用 chfn 命令修改用户姓名。

```
[root@localhost ~]# chfn -f Testlinux newadmin
Changing finger information for newadmin.
Finger information changed.
```

可以看出用户姓名修改成功。

第 3 步，使用 finger 命令查看修改后的 newadmin 账号信息。

```
[root@localhost ~]# finger newadmin
Login: newadmin                         Name: Testlinux
Directory: /home/admin                  Shell: /bin/bash
Never logged in.
No mail.
No Plan.
```

可以看出 newadmin 账号的姓名已经变成了“Testlinux”。

5. 修改用户的登录 Shell——chsh

如果没有指定其他信息，在 Linux 系统中创建用户会采用默认的配置完成。用户的登录 Shell 默认的是/bin/bash。用户也可以自己定制登录 Shell，在创建用户后可以使用 chsh 指令修改登录 Shell。chsh 语法格式如下。

```
[root@localhost ~]# chsh [ -s shell ] [ -l] [ -u ] [--v ] [ username ]
```

各参数含义的描述如下。

-s：指定 Shell，必须是全路径，比如 tcsh 为/bin/tcsh。

-u：显示帮助信息。

-v：显示版本号。

-l：等同于--list-shells，列出所有支持的 Shell。

实例 修改 testuser 用户所使用的 Shell，指令执行如下。

```
[root@localhost ~]# finger testuser | grep Shell
Directory: /home/testuser               Shell: /bin/bash
```

系统显示出用户 testuser 当前所使用的 Shell，然后再输入命令，将用户 testuser 的 Shell 更改为 zsh。

```
[root@localhost ~]# chsh -s /bin/zsh testuser
Changing shell for testuser.
Shell changed.
```

注意 参数“-s”后面必须指定 Shell 的文件路径，否则系统返回出错信息。

6. 修改用户密码过期相关信息——chage

从前面的介绍中可以得到，Linux 中的账号密码有最大存活期、最小存活期，还有密码失效之后的宽限期等一系列密码时效方面的信息。Linux 系统提供了一个专门用于修改登录账号密码时效信息的指令——chage。chage 命令的语法格式如下。

```
[root@localhost ~]# chage [-d] [-E] [-h] [-I] [-l] [-m] [-M] [-W]
```

各参数含义的描述如下。

-d：等同于--lastday。将最近一次密码设置时间设置为“最近日期”。

-E：等同于--expiredate。密码失效日期。

-h：等同于--help。显示帮助信息。

-I：等同于--inactive。设置停滞时期。如果一个密码已过设定天数，则此账号将不可用。

-l：等同于--list。显示账户年龄信息。由非特权用户来确定他们的密码或账号何时过期。

-m：等同于--mindays。设置密码可更改的最小天数。为 0 时表示任何时候都可以更改密码。

-M：等同于--maxdays。设置密码保持有效的最大天数。

-W：等同于--warndays。设置用户密码到期前，提前收到警告信息的天数。

实例 1 显示账号 toplinux 的账龄信息，指令执行如下。

```
[root@localhost ~]# chage -l toplinux
最近一次密码修改时间                                    : 4 月 30, 2009
密码过期时间                                            : 从不
密码失效时间                                            : 从不
帐户过期时间                                            : 从不
两次改变密码之间相距的最小天数                          : 0
两次改变密码之间相距的最大天数                          : 99999
在密码过期之前警告的天数                                : 7
[root@localhost ~]#
```

从命令执行结果可知，用户 toplinux 的账号密码信息永不过期，账号也是永不过期。

实例 2 修改账号 toplinux 的失效时间为 2010 年 1 月 1 日，指令执行如下。

```
[root@localhost ~]# chage -E 2010-01-01 toplinux
```

再次执行命令查看 toplinux 的账龄信息，可以看到 toplinux 账号的过期时间为 2010 年 1 月 1 日。

```
[root@localhost ~]# chage -l toplinux
最近一次密码修改时间                                    : 4 月 30, 2009
密码过期时间                                            : 从不
密码失效时间                                            : 从不
帐户过期时间                                            : 1 月 01, 2010
两次改变密码之间相距的最小天数                          : 0
两次改变密码之间相距的最大天数                          : 99999
在密码过期之前警告的天数                                : 7
```

7. 用户账号修改——usermod

修改用户账号信息，其实最终修改的是配置文件/etc/passwd、/etc/shadow、/etc/group 和/etc/gshadow 的相关内容。建议初级用户不要直接修改配置文件。

chfn、chsh、chage 指令可以修改账户信息，而能够比较全面地修改用户账号信息的指令是 usermod，语法格式如下。

```
[root@localhost ~]#usermod [-c] [-d][-e][-f] [-g][-G][-l][-L][-s ][-u ][-U]
[UserName]
```

各参数含义的描述如下。

-c：修改用户账号的描述信息。

-d：修改用户登录时的主目录。

-e：设置用户账号的失效日期。

-f：设置密码失效后允许登录的宽限期。

-g：修改用户所属的群组。

-G：修改用户所属的附加群组。

-l：修改用户账号的登录名。

-L：锁定用户账号，使其无法无效。

-p：修改密码。

-s：修改用户登录后所使用的 Shell。

-u：修改用户账号的 UID。

-U：解除用户账号锁定。

实例 修改 toplinux 账号信息，将用户的备注文字改成“Test Linux users”，主目录改成/root，指令执行如下。

```
[root@localhost ~]# usermod -c "Test Linux users" -d /root toplinux
```

指令执行后，查看该账号信息，输入以下指令。

```
[root@localhost ~]# finger toplinux
Login: toplinux                         Name: Test Linux users
Directory: /root                        Shell: /bin/bash
Never logged in.
No mail.
No Plan.
```

注意 只有超级用户才能修改其他用户的账户名，如果只修改用户的账户名称，那么用户的其他信息不变。

8. 删除用户账号——userdel

在 Linux 系统中删除用户账号的指令是 userdel，其语法格式如下。

```
[root@localhost ~]# userdel [-f] [-h] [-r] UserName
```

各参数含义的描述如下。

-f：强制删除文件。

-h：显示帮助信息。

-r：删除用户主目录及该目录下的所有文件。

实例 删除 testuser 账号，指令执行如下。

```
[root@localhost ~]# userdel testuser
```

账号成功删除不会有提示信息，但是账号删除出错将出现错误提示。如删除刚才已经删除的 testuser 账号，则返回用户不存在的提示信息。

```
[root@localhost ~]# userdel testuser
userdel: 用户 testuser 不存在
```

需要注意的是，这条指令会主动删除/home 下相应的用户主目录。假如该主目录中的某些文件是账户 root 所创建的，账户 testuser 并没有访问权限，那么该主目录将不被删除。

19.1.4 图形界面方式的用户管理

在图形界面下，不同的 Linux 发布版的操作系统都提供了相应的用户管理软件来实现查看、修改、添加和删除用户账号及用户群组的功能。Linux 操作系统的用户管理器中，几乎包含了所有的命令行用户管理功能。

在 GNOME 桌面下，依次单击“系统”→“管理”→“用户和组群”菜单命令，将显示如图 19-1 所示的“用户管理者”窗口。

如果要查看指定用户的详细信息，可以使用鼠标左键双击该用户，弹出如图 19-2 所示的“用户属性”窗口，该窗口中分类显示了用户的详细信息。

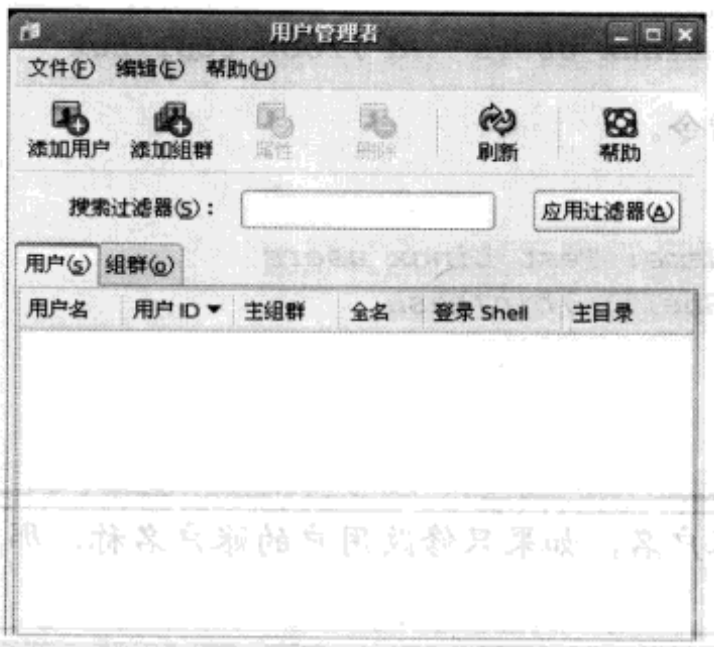


图 19-1 “用户管理者”窗口

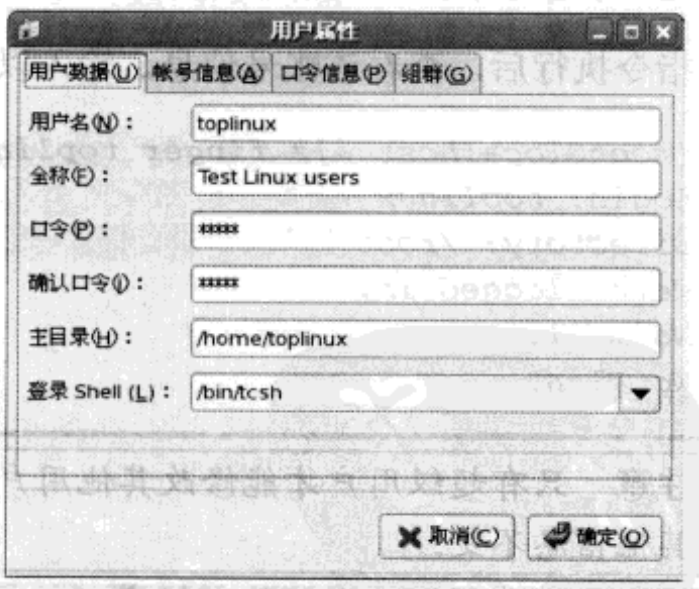


图 19-2 “用户属性”窗口

19.1.5 命令行方式的群组管理

当 Linux 系统中具有数量比较大的用户的时候，分别管理每个用户就显得非常麻烦，这种

情况下，可以将功能及属性相似的用户归为同一个群组，然后使用群组管理办法。本节将介绍管理用户群组的一系列指令。

1. 创建新群组——groupadd

groupadd 是 Linux 操作系统中创建新群组的指令，其语法格式如下。

```
[root@localhost ~]# groupadd [-g gid [-o]] [-r] [-f] [-K KEY=VALUE] GroupName
```

各参数含义的描述如下。

-g：创建指定 GID 的群组。该值取值必须唯一(排除使用-o 参数的情况)，不可重复。另外该 GID 不可为负，默认设置的 GID 必须大于 500。0~499 之间的数值保留给系统账号使用。

-k：覆盖/etc/login.defs 中的设置。

-o：允许创建 GID 相同的群组。

-r：创建系统群组，GID 小于 500。

-f：即使创建群组已经存在，也要返回一个成功标识。

实例 1 新建一个名为“group1”的组，指令执行如下。

```
[root@localhost ~]# groupadd group1
```

系统将新建一个名为“group1”的组，确认该组是否创建成功，输入以下指令。

```
[root@localhost ~]# cat /etc/group | grep group1
group1:x:503:
```

得到“group1”的返回信息，说明该组成功创建。

实例 2 新建一个名为“grouptest”的组，并指定其 ID 号为“678”，指令执行如下。

```
[root@localhost ~]# groupadd -g 678 group2
```

系统将建立一个名为“grouptest”的组，确认该组是否创建成功，输入以下指令。

```
[root@localhost ~]# cat /etc/group | grep group2
group2:x:678:
```

得到“group2”的返回信息，说明该组成功创建。

实例 3 新建一个名为“grouptest”的组，即使已经存在名为“grouptest”的组，也强制创建该组。

首先确认系统中是否存在“grouptest”组，指令执行如下。

```
[root@localhost ~]# groupadd group1
groupadd: group1 组已存在
```

结果系统提示“grouptest 组已存在”，输入指令强制创建该组，系统不再提示出错信息。

```
[root@localhost ~]# groupadd -f group1
[root@localhost ~]# cat /etc/group | grep group1
group1:x:503:
```

注意 groupadd 指令不仅可以指定群组名称建立新群组账户，需要时还可以从系统中取得新群组的值。

2. 删除群组——groupdel

在 Linux 操作系统中具有 root 权限的用户可以使用 groupdel 指令删除已经存在的群组。其语法格式如下，没有任何的附加参数。

```
[root@localhost ~]# groupdel GroupName
```

注意 如果在任何使用者的配置文件/etc/passwd 中指定该群组为其初始群组，则该群组不能被删除。所以，在删除群组之前请先确认是否删除了相应群组的用户或者将他们转移到了其他群组。

实例 删除一个包含有用户的群组，指令执行如下。

```
[root@localhost ~]# gpasswd -d newuser grouptest   ; #先删除组内的用户
[root@localhost ~]# groupdel grouptest             ; #确定组为空才能删除该组
[root@localhost ~]# cat /etc/group |grep grouptest ; #查看组信息
```

3. 修改群组信息——groupmod

groupmod 是 Linux 操作系统中的修改群组 GID 和名称信息的指令，语法格式如下。

```
[root@localhost ~]# groupmod [-g gid [-o]] [-n groupname] Groupname
```

各参数含义的描述如下。

-g：修改 GID，该 GID 必须为唯一的值。

-o：允许修改不唯一的 GID，重复使用群组的 GID。

-n：修改指定群组的名称，使其命名为该参数设定的群组名称。

实例 1 修改指定组的组名和组 ID，指令执行如下。

```
[root@localhost ~]# groupmod -g 789 -n testgroup1 testgroup
```

然后，使用以下指令查看组信息。

```
[root@localhost ~]# cat /etc/group | grep testgroup
testgroup1:x:789:
```

可以看到 testgroup 组的组名变成了 testgroup1，同时组 ID 变成了 789。

实例 2 强制系统使用已经存在的组 ID。

第 1 步，修改 group1 组的组 ID 为 789，指令执行如下。

```
[root@localhost ~]# groupmod -g 789 group1
```

由于已经存在一个组 ID 为 789 的群组，所以返回以下出错信息。

```
groupmod: 789 不是一个独有的 GID
```

第 2 步，强制修改 group1 组的组 ID 为 789，指令执行如下。

```
[root@localhost ~]# groupmod -g 789 -o group1
```

系统不再返回出错信息，组 ID 修改成功，最后确认组 ID 修改成功，指令执行如下。

```
[root@localhost ~]# cat /etc/group | grep group1
group1:x:789:
testgroup1:x:789:
```

4. 修改群组密码——gpasswd

gpasswd 本身是用来为群组设置密码的，不过，现在它还具备添加用户到群组中或者指定某个人为群组管理员的功能。其语法格式如下。

```
[root@localhost ~]# gpasswd groupname
[root@localhost ~]# gpasswd -a username groupname
[root@localhost ~]# gpasswd -d username groupname
[root@localhost ~]# gpasswd -R groupname
[root@localhost ~]# gpasswd -r groupname
[root@localhost ~]# gpasswd [-A username,...] [-M username,...] groupname
```

各参数含义的描述如下。

没有任何参数，表示修改群组的密码。

-a：将某个用户添加到群组中。

-A：将群组的管理员权限交给某个用户。

-M：将某些用户添加到组中。

-d：将某个用户从群组中删除。

-R：使群组的密码失效。

-r：将群组的密码删除。

实例 将用户 toplinux、testuser1 加入到用户组 group1 中，指令执行如下。

```
[root@localhost ~]# gpasswd -M toplinux,testuser1 group1
```

查看用户组 group1 的信息，执行以下指令。

```
[root@localhost ~]# cat /etc/group | grep group1
group1:x:503:toplinux,testuser1
```

由返回结果可以确定用户 toplinux、testuser1 已经成功加入到用户组 group1 中。

注意 若要移出刚才添加进 group1 组中的 toplinux 和 testuser1 用户，执行以下指令即可。

```
[root@localhost ~]#gpasswd -d toplinux,testuser1 group1
```

---

5. 切换群组登录——newgrp

newgrp 指令类似 login 指令，作用是使当前登录的用户切换至另一个群组身份后再次登入系统。欲使用 newgrp 指令切换群组，首先要保证当前的登录用户必须是该群组的用户，否则将无法登入指定的群组。单一用户要同时属于多个群组，需利用交替用户的设置。若不指定群组名称，则 newgrp 指令会登入该用户名称的预设群组。该指令的语法格式如下。

```
[root@localhost ~]#newgrp [-] [Group]
```

各参数含义的描述如下。

-：加入该参数，用户的登录环境会被重新初始化，以前设置的各种参数就不见了。

实例 1 将当前用户加入指定组中，在 Shell 提示符下输入如下指令。

```
[testuser@localhost]~$ newgrp grouptest
```

系统把当前用户加入到名为 testgroup 的组中。只有当前用户是超级用户时，才可能直接操作成功，一般用户都需要输入指定组的密码才能完成操作。

实例 2 使 newuser 以 group1 群组登录。

```
[toplinux@localhost ~]$ groups
toplinux                    ; #当前用户组为 newuser
[toplinux@localhost ~]$ newgrp group1
密码:                       ; #输入 toplinux 的密码
成功
[toplinux@localhost ~]$ groups
group1 newuser              ; #当前组变成了 group1
```

19.2 进 程 管 理

Linux 是一个多任务的操作系统，系统中同时运行着多个进程，正在执行的一个或多个相关进程称为一个作业。作业控制功能使用户可以同时运行多个作业，并在作业之间进行切换。本节将详细介绍进程管理及作业控制的命令，包括启动进程、查看进程、调度作业的命令。

19.2.1 进程及作业

运行中的程序叫做进程。一个运行中的程序，可能有多个进程，比如用户熟悉的瑞星杀毒软件，在运行的时候就有多个进程，分别提供不同的服务。

作业就是一个正在执行的进程，可以包括多个进程。比如“cat /etc/passwd | grep user | more”作业就包括了 3 个进程。

19.2.2 进程控制

Linux 操作系统提供了丰富的进程管理工具，如 ps、kill、pgrep、top 等，不但可以用来查看进程运行装备，而且可以进行完善的控制。

1. 进程监视 ps

ps 是用来显示当前系统进程列表的小工具，具有较完整的运行参数。

```
[root@localhost ~]#ps [-lujfaxr]
```

各参数含义的描述如下。

-l：长格式输出。

-u：按用户名和启动时间的顺序来显示进程。

-j：用任务格式来显示进程。

-f：用树形格式来显示进程。

-a：显示所有用户的所有进程（包括其他用户）。

-x：显示无控制终端的进程。

-r：显示运行中的进程。

ww：避免详细参数被截断。

需要注意的是，ps 的输出列表可能很长，所以要加参数来分页显示结果。

实例 1 分页查看当前系统进程列表。

```
[root@localhost ~]# ps aux |more
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.1   2076   644 ?        Ss   20:14   0:05 init [5]
root         2  0.0  0.0      0     0 ?        S<   20:14   0:00 [migration/0]
root         3  0.0  0.0      0     0 ?        SN   20:14   0:00 [ksoftirqd/0]
root         4  0.0  0.0      0     0 ?        S<   20:14   0:00 [watchdog/0]
root         5  0.0  0.0      0     0 ?        S<   20:14   0:00 [events/0]
root         6  0.0  0.0      0     0 ?        S<   20:14   0:00 [khelper]
root         7  0.0  0.0      0     0 ?        S<   20:14   0:00 [kthread]
root         9  0.0  0.0      0     0 ?        S<   20:14   0:00 [xenwatch]
root        10  0.0  0.0      0     0 ?        S<   20:14   0:00 [xenbus]
root        12  0.0  0.0      0     0 ?        S<   20:14   0:00 [kblockd/0]
......
--More--
```

分页查看当前系统进程列表，按下 Q 键退出。

实例 2 查看 httpd 进程是否在运行（Apache 服务是否被启动了）。

```
[root@localhost ~]# ps aux | grep httpd     : #查看 httpd 程序是否在运行
```

2. 终止进程 kill、killall、pkill、xkill

在 Linux 操作系统下，一般可以通过 kill、killall、pkill 和 xkill 命令来终止进程。

（1） kill

Linux 下常用 kill 指令来终止一个进程，语法格式如下。

```
[root@localhost ~]#kill [ -s signal | -p ] [ -a ] [--] PID ...
[root@localhost ~]#kill -l [ signal ]
```

各参数含义的描述如下。

-s：指定发送的信号。

-p：模拟发送的信号。

-l：指定信号的名称列表。

PID：要中止进程的 ID 号。

Signal：信号。

实例 用 kill 指令终止 vsftpd 进程。

用 kill 指令来终止进程需要用到进程 ID，而进程 ID 又需要结合 ps 指令来获取，所以要终止 vsftpd 进程，不能一步到位，必须分步完成。

第 1 步，执行以下指令查询 vsftdp 进程的 ID。

```
[root@localhost ~]# ps aux |grep vsftpd
root       29835   0.0   0.2    4600    488 ?          Ss    15:14    0:00
/usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf
```

第 2 步，可以看出 vsftpd 的进程 ID 为 29835，接下来只需要执行以下指令就可以了。

```
[root@localhost ~]# kill 29835
```

注意 对于不能终止的僵尸进程来说，可以用“kill -9”来强制终止，比如“kill -9 29835”。

（2） killall/pkill

如果已知了进程的名字，要终止该进程，更为直观的方法是使用指令 killall 和 pkill。其语法格式如下。

```
[root@localhost ~]# killall 进程名
[root@localhost ~]# pkill 正在运行的程序名
```

实例 终止运行中的 vsftpd 进程。

直接执行以下指令即可。

```
[root@localhost ~]# killall vsftpd
[root@localhost ~]# pkill vsftpd
```

（3） xkill

图形界面下最好用的进程终止工具就是 xkill。执行 xkill 指令，然后单击需要关闭的窗口图标即可。按下鼠标右键可退出 xkill。

3. top

和 ps 相比，top 是一个动态的进程监视工具，而且提供了交互式操作，可以直接杀掉进程。top 的语法格式如下。

```
[root@localhost ~]#top [-bcdinpqsS]
```

各参数含义的描述如下。

-b：以批量模式运行，但不能接受命令行输入。

-c：显示命令行，而不仅仅是命令名。

-d N：显示两次刷新时间的间隔，比如 -d 5，表示两次刷新的间隔为 5 s。

-i：禁止显示空闲进程或僵尸进程。

-n NUM：显示更新次数，然后退出。比如 -n 5，表示 top 更新 5 次数据就退出。

-p PID：仅监视指定进程的 ID；PID 是一个数值。

-q：不经任何延时就刷新。

-s：安全模式运行，禁用一些销户指令。

-S：累积模式，输出每个进程的总的 CPU 时间，包括已终止的子进程。

在 top 监视窗口还可以通过按下特定按键来改变显示方式，终止某个进程。具体按键键位介绍如下。

Space：立即更新。

c：切换到命令名显示，或显示整个命令（包括参数）。

f,F：增加显示字段，或删除显示字段。

h,?：显示有关安全模式及累积模式的帮助信息。

k：提示输入要杀死的进程 ID，目的是用来杀死该进程（默认信号为 15）。

i：禁止空闲进程和僵尸进程。

l：切换到显示负载平均值和正常运行的时间等信息。

m：切换到内存信息，并以内存占用大小排序。

n：提示显示的进程数，比如输入 3，就在整屏上显示 3 个进程。

o,O：改变显示字段的顺序。

r：把 renice 应用到一个进程，提示输入 PID 和 renice 的值。

s：改变两次刷新的时间间隔，以 s 为单位。

t：切换到显示进程和 CPU 状态的信息。

A：按进程生命长短进行排序，最新进程显示在最前。

M：按内存占用大小排序，由大到小。

N：以进程 ID 大小排序，由大到小。

P：按 CPU 占用情况排序，由大到小

S：切换到累积时间模式。

T：按时间 / 累积时间对任务进行排序。

W：把当前的配置写到~/.toprc 中。

注意 如果用户更喜欢使用图形化界面，则可以试试 GNOME 系统监视器。依次单击“系统”→“管理”→“系统监视器”菜单命令或者直接在控制台键入“gnome-system-monitor”命令，弹出“系统监视器”窗口，然后切换到“进程”选项卡，如图 19-3 所示。

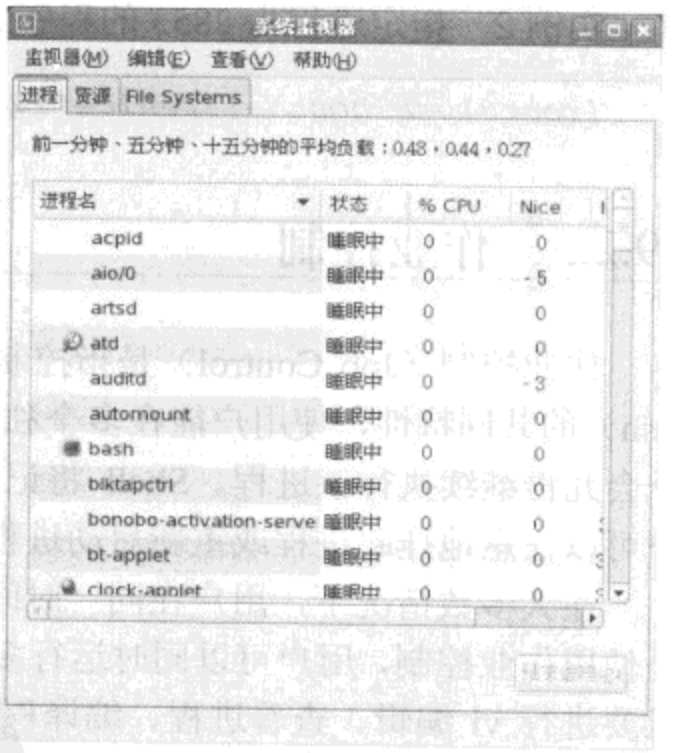


图 19-3 “系统监视器”窗口

4. nice、renice

在 Linux 操作系统中，进程之间是竞争资源（比如 CPU 和内存的占用）的关系。优先级别是通过谦让度来体现的。谦让度的值越大表示进程的优先级别越低。负值或 0 表示高优先级，对其他进程不谦让，即拥有优先占用系统资源的权利。谦让度的默认值为 10，范围为-20~19。

nice 指令可以用来调整程序的谦让度，语法格式如下。

```
[root@localhost ~]# nice [-n adjustment] [-adjustment] [--adjustment=adjustment] [--help]\ [--version] [command [arg...]]
```

各参数含义的描述如下。

-n adjustment：使原有谦让度增加 adjustment。

--help：显示帮助信息。

--version ：显示版本信息。

实例 1 以谦让度+5 的方式运行 gaim 程序。

```
[root@localhost ~]# nice -n 5 gaim
```

renice 指令则是用来修改正在运行中的程序（进程）的优先级的，语法格式如下。

```
[root@localhost ~]#renice priority [[-p] pid ...] [[-g] pgrp ...] [[-u] user ...]
```

各参数含义的描述如下。

-p pid：重新指定进程优先级。

-g pgrp：重新指定进程群组（Process Group）的优先级。

-u user：重新指定某用户进程的优先级别。

实例 2 指定进程为 2853 的程序的谦让度为 15。

```
[root@localhost ~]#renice 15 2853
```

19.2.3 作业控制

作业控制（Job Control）是指控制正在运行的进程的行为，这是许多 Shell（包括 bash 和 tcsh）的共同特性，使用户能在多个独立作业间进行切换。比如，用户可以挂起一个进程，过一会儿再继续执行该进程。Shell 将记录所有启动的进程情况，在每个进程的运行过程中，用户可以任意地挂起进程或重新启动进程。

在大多数情况下，用户在同一时间只运行一个作业，即用户最后向 Shell 键入的命令。但是使用作业控制，用户可以同时运行多个作业，并在需要时在这些作业间进行切换。例如，平时要进行 vi 编辑、查看进程、编译程序这些不同的工作，就要打开多个终端窗口，每个终端窗口执行一个工作任务。如果合理运用了作业控制，就完全可以在一个终端下同时进行这些操作。

1. 将命令放到后台执行——&

在真正将程序放到后台执行之前，还有必要了解一下什么是“前台作业”，什么是“后台作业”。

- 前台作业：可以直接控制的作业为“前台作业”，比如用 vi 编辑文本。
- 后台作业：在内存中自动运行的作业，用户无法直接控制。

要同时进行多个作业，就必须将某些可以自动进行的作业放到后台，这样就可以让需要用户控制和接受输入的前台继续作业。将作业放到后台运行最简单的方法就是用“&”指令。比如在编译 Linux 核心，执行以下命令。

```
[root@localhost ~]#make menuconfig
[root@localhost ~]#make clean
```

执行完这两步，就会开始编译核心，而整个过程是自动进行的，所以可以将作业放在后台。

```
[root@localhost ~]#make bzImage &
[1] 29853    ; #工作编号、PID
[root@localhost ~]#      ; #在这里继续输入指令，进行其他的操作
```

2. 将当前作业“暂停”到后台——Ctrl+Z

如果正在用 vi 编辑文档，却发现有不认识的单词，需要运行词典软件。其实不需要关闭 vi，暂时把 vi 存放到后台即可。把正在进行的作业暂停并放到后台需要使用 Ctrl+Z 指令。

```
[root@localhost ~]# vi .bash_profile
; #按下 Ctrl-Z 将它放到后台
[1]+ Stopped vi .bash_profile      ; #[1]表示第一个作业，+表示是后台默认的作业
                                   ; #Stopped 表示该作业处于暂停状态
                                   ; #放入后台的程序都是停止状态
[root@localhost ~]#                ; #在这里可以打开词典程序查询单词了
```

3. 观察当前后台作业状态——jobs

放入后台的作业，都可以通过 jobs 指令查询，语法格式如下。

```
[root@localhost ~]#jobs [-lnprs]
```

各参数含义的描述如下。

-l：不但列出作业号，还列出 PID。

-p：显示选定作业的进程组引导符的进程标识。

-n：显示从最后一次通知后停止或退出的作业。

-r：仅仅显示在后台运行的进程。

-s：仅仅列出在后台暂停的进程。

实例 查看当前的后台作业及其 PID 信息。

显示后台执行的作业，指令如下。

```
[root@localhost ~]# jobs
[1]+  Stopped                 vi test.txt
```

注意　在 jobs 命令执行的结果中，加号表示是一个当前的作业，减号表示是当前作业之后的一个作业。jobs 的状态可以是 running、stopped 或 Terminated，其显示的是当前 Shell 环境中后台正在运行或者挂起的任务的信息。

4. 将后台作业拿回来——fg

将作业放到了后台，在合适的时候就要将他们拿回来。使用 fg 指令，语法格式如下。

```
[root@localhost ~]# fg [job number]
```

实例　将放入后台的 vi .bash_profile 进程拿出来。

第 1 步，先用 jobs 指令查看后台作业。

```
[root@localhost ~]# jobs
[2]-  Stopped                 top
[3]+  Stopped                 vi .bash_profile
```

可以看出 vi 指令的作业号是 3。

第 2 步，执行 fg 指令将 vi 指令放回前台。

```
[root@localhost ~]#fg  3    ; #fg 将后台作业编号放回前台
```

5. 让进程在后台运行——bg

使用 Ctrl+Z 组合键可以将当前作业“暂停”到后台，使用“&”让进程在后台运行，但是在后台暂停的进程如何在后台启动运行的？可以使用 bg 指令，语法格式如下。

```
[root@localhost ~]# bg [job number]
```

实例　将 top 指令放在后台运行。

```
[root@localhost ~]#make bZimage
[1]+ Stopped          bZimage              ; #按下 Ctrl-Z 将它暂停到后台
[root@localhost ~]#jobs
[1]+ Stopped          make bZimage         ; #让后台处于暂停状态
[root@localhost ~]#bg 1
[1]+ make bZimage &                        ; #后台处于运行状态了
```

第 20 章　管理 Linux 文件系统及磁盘

文件系统和磁盘是数据存储的关键，所以对于学习 Linux 的用户而言，掌握 Linux 文件系统及磁盘管理的知识至关重要。本章从 Linux 文件系统的基本概念入手，并以 Red Hat Enterprise Linux 5 为例，带大家层层深入学习文件系统与磁盘管理相关的知识。同时为了让广大读者朋友能更快、更好地理解各个知识点，还结合了大量实际案例进行讲解。

20.1　管理 Linux 文件系统

文件系统是操作系统在磁盘或者分区上保存文件信息的方法和数据结构，即文件在磁盘或分区上的组织方式。本节将从相关概念入手，然后从常见文件系统、文件类型、文件压缩与归档、文件链接和文件系统安装方法等多方面来展开对文件系统管理知识的探讨。

20.1.1　Linux 文件系统结构

大部分 Linux 文件系统的结构都与 UNIX 文件系统结构类似，都包括几个重要概念：超级区块（superblock）、索引节点（inode）、数据块（data block）、目录块（directory block）。

其中，在超级区块（superblock）中包含有关该磁盘或分区上的文件系统的整体信息，如文件系统的大小等。索引节点（inode），包含了针对某个具体文件的几乎全部信息，如文件的存取权限、拥有者、文件大小、建立时间，以及对应的目录块和数据块等，但索引节点中不包含文件名字信息。数据块（data block）是真正存储文件内容的位置。目录块（directory block）里包含有文件的名字和该文件的索引节点编号。

20.1.2　常见的文件系统

在 Linux 操作系统中，硬盘被分区之后，需要格式化才能使用，其中 Linux 操作系统支持的文件系统是 Linux Ext2、Ext3。在学习这些文件系统之前，需要了解一下 Linux 操作系统中的索引节点 inode 的概念。

在 Linux 操作系统中，其文件系统都是采用 inode（节点）来保存文件属性的，该索引节点 inode 记录了有关文件的一些相关信息，包括文件的所有者、文件的权限设置、文件的大小、文件的时间戳，以及文件真正内容保存的位置等。

在内存中的索引节点 inode 还维护着一个关于该节点的使用计数。每当一个索引节点 inode 所指示的文件被打开，这个索引节点 inode 就把关于自己的使用计数加 1；每当这个索引节点 inode 所代表的文件被关闭，这个索引节点 inode 就把关于自己的使用计数减 1。当用户程序删除一个文件的时候，相关的系统调用很快就返回到这个用户程序中，告诉该用户程序中相应的文件已经被删除了。但是相应的索引节点 inode 还是保留在系统中，系统首先会检查该节点

的使用计数，如果使用计数为零，那么系统核心才会真正地删除该文件。

如果使用计数大于 0，则表示还有其他用户程序正在访问该文件，那么系统核心还需等待其他所有访问该文件的用户程序完成文件操作工作。只有等到该索引节点 inode 的使用计数为 0 时，才能真正删除该节点所指示的文件。

1. Linux Ext2 文件系统

该文件系统是 Linux 上最早采用的文件系统类型，源自于 Andrew Tanenbaum 所写的 Minix 操作系统，并具有很强的存取文件的性能，非常适合中、小型的文件管理。

2. Linux Ext3 文件系统

该文件系统相当于 Linux Ext2 文件系统的增强版，其在 Linux Ext2 的基础上发展了日志功能。日志型文件系统可以最大限度地保证数据的完整性，这是由于其会将整个磁盘的写入操作行为完整地记录在硬盘的某个区域中，如此一来，在需要时就可以回溯跟踪。

3. Linux Swap

该文件系统是 Linux 系统中专门用于交换分区的文件系统，使用这个分区作为内存页面交换空间，不容易产生碎片，因此比 Windows 的页面文件系统效率要好。

20.1.3 文件类型

Linux 下有系统文件、CONF 配置文件、LOCK 锁定标识文件、RPM 软件包文件以及压缩文件（包括 ZIP、TGZ、TBZ、BZ2 等格式）和编程与脚本语言（如 CPP、O、PL、PY、SO 和 SH 等）。

20.1.4 文件压缩与归档

文件的归档是为了方便将一组文件存储成一个文件以实现快速备份或者快速传输。同时，可以对该文档进行压缩以便占用更少的空间。

Red Hat Enterprise Linux 5 内置了一个图形界面的“归档管理器”，该工具可以归档、压缩、解压缩文件和目录。单击“应用程序”→“附件”→“归档管理器”菜单命令打开，如图 20-1 所示。

通过该图形界面可以非常简单直观地完成文件或者目录的归档、压缩、解压缩等操作。另外，Linux 系统中也提供了许多非常有用的命令行压缩和解压缩指令（包括 compress、gzip/zcat、bzip2/bzcat/bunzip2 和 tar 等）。

1. compress

compress 是 UNIX 文件压缩指令，压缩后的文件会加上一个.Z 后缀。若要将多个文件压成一个压缩包，必须先将文件用 tar 工具打包起来再压缩。

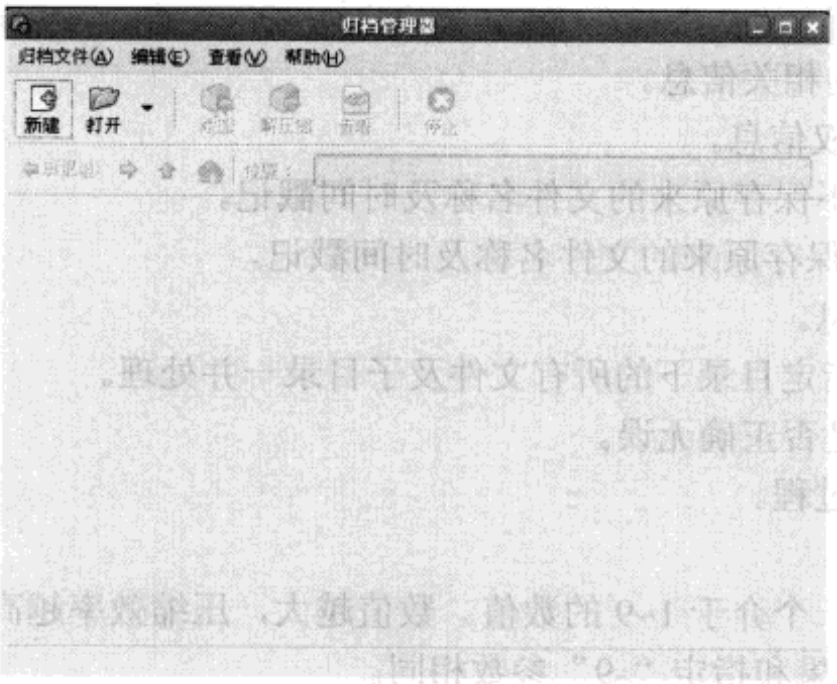


图 20-1 “归档管理器”窗口

注意 使用 compress 指令压缩的文件可以通过 uncompress 指令来解压缩。另外，由于 gzip 能够产生更理想的压缩比例而且操作方便，现在一般用户都不再用 compress 作为压缩工具了，而是采用 gzip。

compress 指令的语法格式如下。

```
[root@Localhost ~]#compress [-dfvcV] [-b maxbits] [file ...]
```

各参数含义的描述如下。

-d：解压缩。

-f：强制写入。如果文件存在，则会被覆盖。

-v：显示程序执行信息到屏幕上。

-c：将压缩的数据显示到屏幕上。

-b：设定共同字串数的上限。

-V：显示帮助信息。

2. gzip/zcat

gzip 由 GNU 所开发，其压缩出来的文件名后缀为.gz，取代了 compress；zcat 则表示显示压缩文件内容。gzip 语法格式如下。

```
[root@localhost ~]# gzip [-acdfhlLnNqrtvV][-#] [--best/fast][File or Folder...]
[root@localhost ~]# zcat FileName.gz
```

如参数含义的描述如下。

-a：使用 ASCII 文字模式。

-c：把压缩后的文件输出到标准输出设备，不去更动原始文件。

-d：打开压缩文件。

-f：强行压缩文件。不考虑文件名或硬连接是否存在以及该文件是否为符号连接。

-h：在线帮助。
-l：列出压缩文件的相关信息。
-L：显示版本与版权信息。
-n：压缩文件时，不保存原来的文件名称及时间戳记。
-N：压缩文件时，保存原来的文件名称及时间戳记。
-q：不显示警告信息。
-r：递归处理，将指定目录下的所有文件及子目录一并处理。
-t：测试压缩文件是否正确无误。
-v：显示指令执行过程。
-V：显示版本信息。
-#：压缩效率。是一个介于 1~9 的数值。数值越大，压缩效率越高，预设值为 6。
--best：此参数的效果和指定“-9”参数相同。
--fast：此参数的效果和指定“-1”参数相同。

实例 1 用 gzip 压缩/hhwork/doc1。

第 1 步，复制要压缩的文件到其他目录，执行如下命令。

```
[root@localhost ~]# cp /hhwork/doc1 /tmp
```

注意 gzip 压缩时会删除原文件，因此，需要先复制到其他地方再压缩。

第 2 步，进入该目录，执行如下命令。

```
[root@localhost ~]#cd /tmp
```

第 3 步，压缩文件，执行如下命令。

```
[root@localhost tmp]#gzip doc1
```

实例 2 显示上例压缩包中的原文件。

```
[root@localhost tmp]#zcat /tmp/doc1.gz
```

执行命令成功后，doc1 的内容会在终端窗口上显示出来，如下所示。

```
%set force_authentication 0
% This can either be set to "nntp" (read online) or "spool" (use local spool)
%set server_object "nntp"
% This can be "nntp" (post directly to the server), "inews" (use external
% program for posting) or "slrnpull" (spool message for slrnpull to send).
%set post_object "nntp"
% Set this to zero if you want to leave Message-ID generation to your server
% or your system does not have a proper unique hostname.
```

此时可以通过重定向语句将使它生成一个文件。

实例 3 解压缩 fstab.gz。

```
[root@ localhost ~]#gzip -d doc1.gz
```

3. bzip2/bzcat/bunzip2

bzip2 指令比 gzip 指令的压缩率更高，其对应的解压缩软件有 bzcat 和 bunzip2 等。

（1） bzip2

bzip2 的语法格式如下。

```
[root@localhost ~]# bzip2 [-cdfhkLstvVz][--repetitive-best][--repetitive-fast]
[Files]
```

各参数含义的描述如下。

-c：将压缩与解压缩的结果送到标准输出（一般就是屏幕上）。

-d：解压缩。

-f：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，请使用此参数。

-h：显示帮助。

-k：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，请使用此参数。

-s：降低程序执行时内存的使用量。

-t：测试.bz2 压缩文件的完整性。

-v：压缩或解压缩文件时，显示详细的信息。

-z：强制执行压缩。

-V：显示版本信息。

--repetitive-best：若文件中有重复出现的资料时，可利用此参数提高压缩效果。

--repetitive-fast：若文件中有重复出现的资料时，可利用此参数加快执行速度。

-#：压缩时的区块大小。数值在 1~9 之间，数值越大，效果越好。

实例 1 将/hhwork 下的 doc1 压缩。

```
[root@ localhost ~]# bzip2 /hhwork/doc1
```

注意 如果使用此方法进行压缩工作，源文件会被删除，只留下一个 doc1.bz2 文件。

实例 2 将/tmp 下的 doc 压缩，但不要删除原来的文件。

```
[root@ localhost ~]# bzip2 /hhwork/doc1 -k
```

（2） bzcat

该命令将显示 bz2 压缩包中的内容到屏幕上，其语法格式如下。

```
[root@ localhost ~]# bzcat [Files]
```

实例 显示上文压缩的 doc1.bz2 文件内容，执行如下命令。

```
[root@ localhost ~]#bzcat /hhwork/doc1.bz2
```

执行命令后的结果如下所示。

```
[root@ localhost ~]#bzcat /hhwork/doc1.bz2
```

```
%set force_authentication 0
% This can either be set to "nntp" (read online) or "spool" (use local spool)
%set server_object "nntp"
% This can be "nntp" (post directly to the server), "inews" (use external
% program for posting) or "slrnpull" (spool message for slrnpull to send).
%set post_object "nntp"
% Set this to zero if you want to leave Message-ID generation to your server
% or your system does not have a proper unique hostname.
```

显示了文件 doc1 的内容。

（3） bunzip2

bunzip2 则是与 bzip2 对应的.bz2 文件解压缩工具，其语法格式如下。

```
[root@Localhost ~]# bunzip2 [-fkvsVL] [Files]
```

各参数含义的描述如下。

-f：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，请使用此参数。

-k：解压缩后，会删除原始的文件；若要保留原始文件，请使用此参数。

-v：显示详细的信息。

-s：降低程序执行时内存的使用量。

-V：显示版本信息。

-L：同-V 参数。

实例 对刚才生成的/hhwork/doc1.bz2 进行解压缩，如果存在输出文件，则覆盖。

```
[root@Localhost ~]#bunzip2 /hhwork/doc1.bz2 -f
```

（4） bzip2recover

用于从 bzip2 文件中恢复文件，当 bzip2 压缩包损坏时，该指令可以恢复部分甚至全部文件。其语法格式如下。

```
[root@Localhost ~]#bzip2recover File
```

4. tar

tar 可以将文件或者目录打包到一起，与 gzip 和 bzip2 结合，还可以将这些包压缩成 gz 或者 bz2 格式。其语法格式如下。

```
[root@Localhost ~]#tar [-cxtzjvfpPN][--exclude FILE] FILES
```

各参数含义的描述如下。

-c：打包。

-x：解包。

-t：查看 tar 文件里的内容。

注意 在使用的参数中，c/x/t 仅能存在一个，不可同时存在。

-z：用 gzip 压缩。
-j：用 bzip2 压缩。
-v：压缩的过程中显示文件列表。
-f：文件名。该参数必须是最后一个参数，后面接文件名。
-p：保留原文件属性。
-P：使用绝对路径。
-N：压缩指定日期（yyyy/mm/dd）以后的文件。
--exclude FILE：不压缩 FILE！

实例 1 将/home/lily 下的所有文件打包，保存到/tmp/lily.tar 中。执行如下命令。

```
[root@Localhost ~]#tar -cvf /tmp/lily.tar /home/lily
```

实例 2 将/home/lily 下的所有文件打包，保存到/tmp/lily.tar.gz 中。
用 gzip 压缩打包，执行如下命令。

```
[root@Localhost ~]#tar -zcvf /tmp/lily.tar /home/lily
```

用 bzip2 压缩打包，执行如下命令。

```
[root@Localhost ~]#tar -jcvf /tmp/lily.tar /home/lily
```

实例 3 查看/tmp/clinuxer.tar.gz 中的文件内容。

```
[root@Localhost ~]#tar -ztvf /tmp/clinuxer.tar.gz
```

实例 4 将/tmp/clinuxer.tar.gz 包解压缩到/home/clinuxer 下。

```
[root@Localhost ~]#cd /
[root@Localhost ~]#tar -zxvf /tmp/clinuxer.tar.gz
```

压缩的时候已经包括有路径，现在只需要将其释放到根目录下。

20.1.5 文件链接

在 Linux 操作系统下，链接不但可以针对文件，也可以针对目录，给一个目录创建一个链接，就可以利用该链接快速访问该目录而不用输入一长串的目录路径。这里不难看出，Linux 系统下的文件链接其实就是给文件另外起的一个名字。

Linux 下创建链接的指令是 ln，可以创建两种类型的链接：软链接（Symbolic Link）和硬链接（Hard link）。默认情况下创建的是硬链接，其语法格式如下。

```
[root@Localhost ~]#ln [Options] source dist
```

各参数含义的描述如下。
-b：将在链接时可能被覆盖或删除的文件先备份。
-f：链接时先将与 dist 同名的文件删除。
-d：允许管理员硬链接自己的目录。
-i：在删除与 dist 同名的文件时先进行询问。

-n：在进行软链接时，将 dist 视为一般的文件。
-s：创建软链接（Symbolic Link）。
-v：在创建链接之前显示文件名。
-S：给备份的文件都加上指定的字尾。
-V：指定备份的方式。
--help：显示帮助信息。
--version：显示版本信息。

1. 硬链接

硬链接是指通过索引节点来进行的链接，所有的硬链接必须位于同一分区中。使用硬链接可以达到允许一个文件拥有多个有效路径名的功能，如此一来，用户就可以建立硬链接到重要文件，以防止该文件被删除。只有当最后一个链接被删除后，文件的数据块及目录的链接才会被释放，文件此时才会被真正删除。

注意 在 Linux 文件系统中，保存在磁盘分区中的文件不管是什么类型都有一个编号，该编号被称为索引节点号（Inode Index）。该节点号存在的目的是作为存取文件的索引。另外，在 Linux 系统中，多个文件名指向同一索引节点也是可以的。

2. 软链接

除硬链接之外，在 Linux 操作系统中还存在一种与其对应的链接方式，被称为软链接，软链接可以创建在其他分区中。下面将通过一个实例来讲解硬链接与软链接的区别。

实例 创建一个硬链接，一个软链接。

第 1 步，建立一个新的空文件 file，执行如下命令。

```
[root@Localhost ~]#touch file
```

第 2 步，创建一个 file 的硬链接 hfile，执行如下命令。

```
[root@Localhost ~]#ln file hfile
```

第 3 步，创建一个 file 的软链接 sfile，执行如下命令。

```
[root@Localhost ~]#ln -s file sfile
```

第 4 步，显示相关信息，执行如下命令。

```
[root@Localhost ~]#ls -lia
```

执行命令后的结果如下所示。

```
[root@Localhost ~]#ls -lia
总计 1000
 950273 drwxr-x--- 33 root root    4096 07-21 00:21 .
      2 drwxr-xr-x 27 root root    4096 07-20 17:44 ..
955005 -rw-r--r--  2 root stella     0 07-21 00:19 file
955005 -rw-r--r--  2 root stella     0 07-21 00:19 hfile
```

```
950356 lrwxrwxrwx  1 root stella     4 07-21 00:21 sfile -> file
```

由该结果可以看出，创建了两个文件，其中 sfile 是软链接，其文件属性第一列是“l”。而 hfile 虽然从文件属性看不出来是链接，但其 inode 值与 file 一样，都是 955005，因此是一个硬链接。

如果删除 sfile，对 file、hfile 完全没有影响。

```
[root@Localhost ~]#rm sfile
rm: 是否删除 一般空文件 “sfile”? yes
[root@Localhost ~]#ls -lia
总计 996
 950273 drwxr-x--- 33 root root    4096 07-21 00:35 .
      2 drwxr-xr-x 27 root root    4096 07-20 17:44 ..
955005 -rw-r--r--  2 root stella     0 07-21 00:19 file
955005 -rw-r--r--  2 root stella     0 07-21 00:19 hfile
```

删除 hfile，也对 file、sfile 没有影响。

```
[root@Localhost ~]#ln -s file sfile
[root@Localhost ~]#rm hfile
rm: 是否删除 一般空文件 “hfile”? yes
[root@Localhost ~]#ls -lia
总计 996
 950273 drwxr-x--- 33 root root    4096 07-21 00:38 .
      2 drwxr-xr-x 27 root root    4096 07-20 17:44 ..
955005 -rw-r--r--  2 root stella     0 07-21 00:19 file
950356 lrwxrwxrwx  1 root stella     4 07-21 00:21 sfile -> file
```

删除 file，对 hfile 没有影响，但是 sfile 失效（文件会显示红色）。

```
[root@Localhost ~]#ln file hfile
[root@Localhost ~]#rm file
rm: 是否删除 一般空文件 “file”? yes
[root@Localhost ~]#ls -lia
总计 996
 950273 drwxr-x--- 33 root root    4096 07-21 00:41 .
      2 drwxr-xr-x 27 root root    4096 07-20 17:44 ..
955005 -rw-r--r--  2 root stella     0 07-21 00:19 hfile
950356 lrwxrwxrwx  1 root stella     4 07-21 00:38 sfile -> file
```

删除 file、hfile，那么创建的文件被删除。

```
[root@Localhost ~]#rm hfile
```

20.1.6 安装文件系统

在 Linux 操作系统下安装文件系统并不复杂，接下来将具体讲解如何使用 fstab 进行文件系统的安装。

1. Windows 分区

对于使用 Windows，Linux 双系统的用户免不了要在 Linux 和 Windows 之间共享一些资源。此时，最好的做法是在 Linux 下安装 Windows 分区，再进行资源的共享。

实例 安装 Windows 下的 FAT32 分区。

```
[root@Localhost ~]#mkdir /mnt/Win_C
[root@Localhost ~]#mount -t vfat /dev/sda1 /mnt/Win_C
# 将 Windows 下的 C 盘安装好之后，在 Linux 下访问/mnt/Win_C
# 就相当于访问 Windows 的 C 盘。
```

然后便可使用 umount 指令来下载安装文件系统。

```
[root@Localhost ~]#umount /mnt/Win_C
```

2. 安装光驱

在 Linux 中，执行以下指令挂载光驱。

```
[root@Localhost ~]#mkdir /mnt/cdrom
[root@Localhost ~]#mount -t cd9660 /dev/cdrom /mnt/cdrom
```

要取出光盘，首先必须保证/mnt/cdrom 没有被使用。然后执行以下指令卸载光驱。

```
[root@Localhost ~]#umount /mnt/cdrom
```

此时，光驱与 Linux 目录脱离，即可打开光驱取出光盘。

3. 使用 fstab 安装文件系统

在 Linux 下需要经常使用 mount 命令来加载文件系统，这样会比较麻烦，因此建议在系统启动的时候就自动安装文件系统。该操作比较简单，只要告诉 Linux 需要安装哪个文件系统、安装到哪个目录并说明安装参数即可，具体方法可以通过编辑/etc/fstab 文件来实现。比如要在系统启动时自动安装 Windows 下的 C 盘，则只需要在 fstab 中添加以下语句即可。

```
#Device     MountPoint      FStype  Options Dump    Pass#
/dev/sda1       /mnt/Win_C      vfat            defaults    0       0
#Device     MountPoint      FStype  Options Dump    Pass#
/dev/sda1       /mnt/Win_C      vfat            defaults    0       0
```

其中，文件的第 1 列为设备文件名字，第 2 列为文件系统的安装位置，第 3 列为文件系统的类型（vfat 就是 FAT32，cd9660 是光驱），第 4 列为安装选项（rw 表示该文件系统能进行读、写操作；ro 表示该文件系统是只读的；sw 用于交换分区；noauto 表示启动时并不立即安装这个文件系统，而是由管理员在以后手工安装；default 表示采用默认选项安装），第 5 列和第 6 列的数字是系统程序 dump 和 fsck 使用的标志，通常不需要改变，这个标志在根文件系统中是 1，而在其他类型的文件系统中是 0。

注意 由于 fstab 文件非常重要，如果这个文件有错误，就有可能造成系统不能正常启动。

20.2 磁盘管理

磁盘用来存储操作系统中的文件，在任何操作系统中都必须经过格式化分区才能使用，Linux 操作系统也不例外，接下来本节将具体讲解 Linux 操作系统下的有关磁盘管理的知识。

20.2.1 硬盘简介

从物理结构来看，硬盘就是由很多圆形的硬盘盘片所组成的。其中根据硬盘能够容纳的硬盘盘片数量来看，有一块硬盘里面只有一个硬盘盘片的单碟硬盘和一块硬盘里面含有多个硬盘盘片的多碟硬盘。下面就以单碟硬盘作为研究对象研究其构造。单碟硬盘的盘面如图 20-2 所示。

在开始使用磁盘管理工具之前，还应该对相关的知识有个基本的了解。

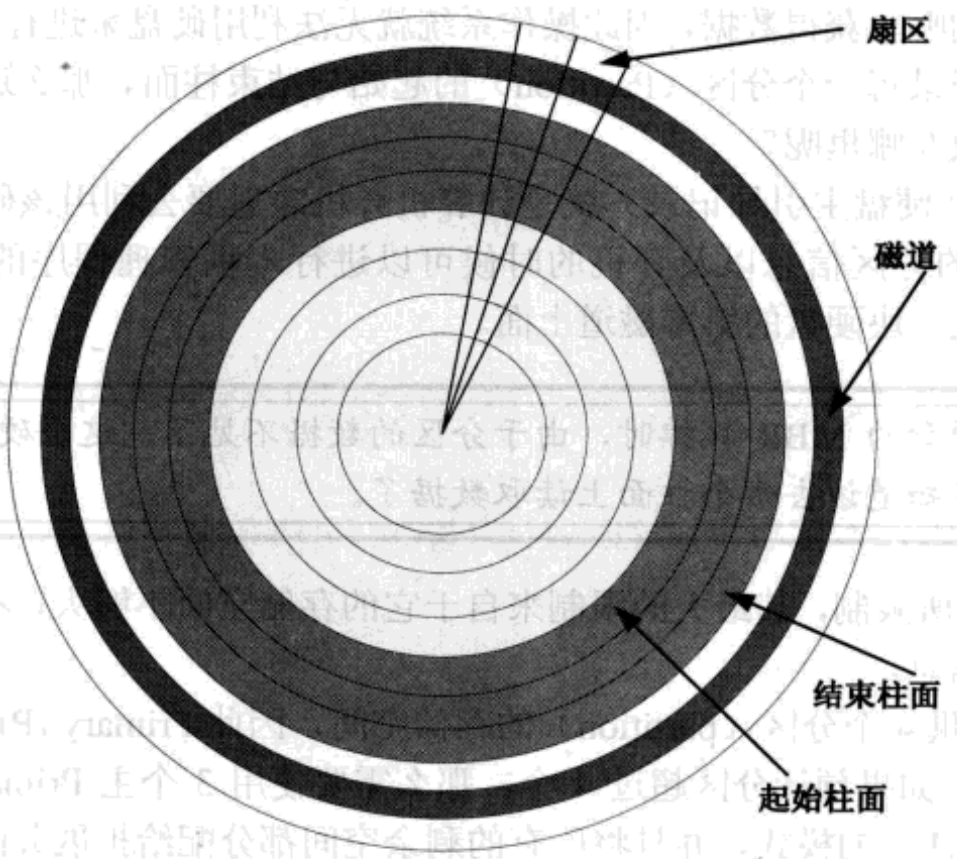


图 20-2 硬盘盘片示意图

1. 硬盘物理结构

首先，硬盘里面有磁头（Head）对该硬盘盘片上面的信息进行读、写操作。

当磁头固定不动时，硬盘盘片转一圈所画出来的圆即是磁道（Track）。由于一块硬盘里面可能具有多个硬盘盘片，而所有硬盘盘片上面相同半径的那一个磁道就组成了柱面（Cylinder）。

如图 20-3 所示，在两个硬盘盘片上面的同一个磁道就是一个柱面。该柱面也是磁盘分割（partition）时的最小单位。

由圆心向外划直线，可将磁道再细分为一个一个的扇区（Sector），一般情况下，扇区能储存的数据大小为 512 B。而这个扇区就是硬盘盘片上面的最小储存数据的物理单元。

以上就是整个硬盘的基本组成。

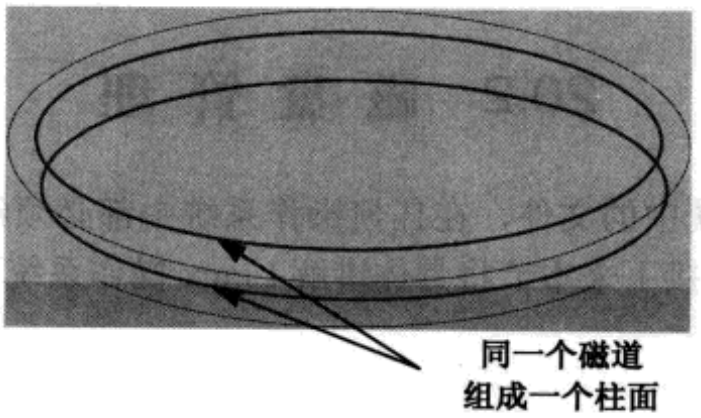


图 20-3　硬盘柱面示意图

2. 硬盘分区

在了解了硬盘的物理结构之后，有必要了解一下硬盘的分区（Partition）。

对硬盘进行分区即是告诉操作系统，这块硬盘可以存取数据。此时，操作系统才能够控制硬盘磁头在该区域范围内的柱面上存、取数据。相反，如果没有告诉操作系统这个信息，操作系统将无法知道到哪里获得数据，因此操作系统就无法利用硬盘来进行数据的存、取了。所以，磁盘分区就是记录每一个分区（Partition）的起始与结束柱面。那么这个分区的起始与结束柱面的数据又是放在哪里呢？

事实上，MBR（硬盘主引导记录）就是计算机开机之后要去利用该硬盘时读取的第一个区域。硬盘里所有的分区信息以及开机的时候可以进行本机管理程序的入口地址都放置在 MBR 中，MBR 就在一块硬盘的第零磁道上面。

注意　当一个硬盘的 MBR 坏掉时，由于分区的数据不见了，这个硬盘也就几乎是报废了，因为操作系统不知道该去哪个柱面上读取数据了。

同时 MBR 也有所限制，其最大的限制来自于它的存储空间不够大，不是以储存所有分区与开机管理程序的信息。

MBR 最多能提供 4 个分区（partition）的存储空间，因此 Primary (P)与 Extended (E)的分区最多只能有 4 个。如果预计分区超过 4 个，那么需要使用 3 个主 Primary 分区和 1 个扩展 Extended 分区（3P+1E）的模式，并且将所有的剩余空间都分配给扩展分区。当然扩展分区最多只能有 1 个，否则在 3P+E 之后，还有剩下的磁盘空间将会被浪费掉。

因此，我们在磁盘划分过程中，势必要注意必须在 4 个分区范围内将整个磁盘全部划分完。

3. 硬盘名称

在 Linux 操作系统下，每个硬盘都有一个独立的设备名。IDE 硬盘的名称的格式如下。

```
/dev/hdX
```

其中的 X 可以是 a~z 之间的小写字母序列。例如，Primary Master 是 hda，Primary Slave 是 hdb，Secondary Master 是 hdc，Secondary Slave 是 hdd……

SATA、SCSI 设备遵循与 IDE 类似的机制，比如第一个 SCSI/SATA 硬盘叫做/dev/sda，第二个叫做/dev/sdb 如下所示，可以看出 Linux 操作系统下的分区标识与 Windows 操作系统下的

分区标识差异很大。

```
[root@localhost ~]# ls /dev/hd*
/dev/hdc
[root@localhost ~]# ls /dev/sd*
/dev/sda  /dev/sda1  /dev/sda2
```

4. 分区名称

建立分区后，分区的名称则是在原来的硬盘名称后加上数字，格式如下。

```
/dev/hdXY
```

其中的“Y”是分区编号。一般情况下，第 1 个主分区是 1，第 2 个主分区是 2，第 3 个主分区是 3，第 4 个主分区是 4。因为硬盘最多只能被划分为 4 个主分区，所以很多时候需要把其中的一个分区创建为扩展分区，再在其中创建逻辑分区：第 1 个逻辑分区标识数字 5，第 2 个逻辑分区是 6……依此类推，如下所示。

```
[root@localhost ~]# parted /dev/sda
GNU Parted 1.8.1
使用 /dev/sda
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) print
Model: VMware, VMware Virtual S (scsi)
Disk /dev/sda: 8590MB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Number  Start   End     Size    Type     File system  标志
1       32.3kB  107MB   107MB   主分区   ext3         启动
2       107MB   8587MB  8480MB  主分区                lvm
```

20.2.2 管理磁盘分区

众所周知，磁盘分区的存在，是由磁盘的物理特性决定的，并不会因为操作系统的不同而有所改变。

在 Linux 中，有以下几种分区的形式。

- /dev/had
- /dev/hda1
- /dev/hda2
- /dev/hda5
- /dev/sdb1

下面就以 /dev/hda5 为例进行简要说明。

由于在 Linux 中，每一个设备都是用/dev/文件夹下的一个文件来表示，所以/dev/hda5 中，/dev/表示的是根目录下的 dev 目录，而 hda5 代表的含义比较丰富，具体如下。

- 前两位的字母 hd 表示这是一块 IDE 磁盘；如果是 sd，则表示是 SATA 磁盘，或者闪存等外设。

- 第 3 位的字母 a 表示这是该类型接口上的第一个设备。同理，b、c、d…分别代表该类型接口上的第 2、3、4…个设备。
- 第 4 位的数字 5，表示是第 1 个逻辑分区，而不是该磁盘中的第 5 个分区。

注意 在 Linux 操作系统中，为了避免不必要的混乱，分区的顺序是不能改变的，分区标识则由它们在磁盘中的位置决定。系统还要为所有可能的主分区预留标识，所以 1~4 之间的数字一定不会是逻辑分区的标识，5 则是第 1 个逻辑分区，依次类推。

在对 Linux 下磁盘分区的相关概念有所认识后，下面就和大家共同学习有关磁盘分区方面的一系列系统指令。

1. 磁盘分区相关指令

在 Red Hat Enterprise Linux 5 中，内置有多个分区工具，下面就以 fdisk 和 parted 两个指令为例进行讲解。

（1） fdisk

该命令主要用于显示磁盘分区信息，创建或删除磁盘分区，改变活动分区。另外，该命令是一个面向底层的系统工具，使用该工具前一定要注意事先对重要数据进行备份。

fdisk 命令通过传统的问答式的界面管理、维护磁盘分区。其允许 4 个不同的操作系统存在于不同的分区中，但在给定的时间内只有一个分区是活动的。我们也可以把一个物理分区分成几个逻辑分区。Linux 系统分区建议的最小容量为 40 MB。通常设备 device 可以是/dev/had、/dev/hdb、/dev/sda、/dev/sdb、/dev/hdc 或/dev/hdd 等。其命令格式如下。

```
[root@localhost ~]# fdisk [-b <分区大小>][-uv][外围设备代号]
[root@localhost ~]# fdisk [-l][-b <分区大小>][-uv][外围设备代号...]
[root@localhost ~]# fdisk [-s <分区编号>]
```

各参数含义的描述如下。

-b<分区大小>：指定每个分区的大小。

-l：列出指定的外围设备的分区表状况。

-s<分区编号>：将指定的分区大小输出到标准输出上，以区块为单位。

-u：搭配“-l”参数列表，会用扇区数目取代柱面数目，来表示每个分区的起始地址。

-v：显示版本信息。

各子命令含义的描述如下。

a：在当前分区上设定一个可启动标志。

b：编辑一个 BSD 磁盘标识。

c：切换为 DOS 兼容的磁盘标识。

d：删除当前分区。

l：列出所有分区类型。

m：主菜单。

n：创建一个新分区，提示输入更多信息。

o：创建一个 DOS 分区。

p：输出所有分区列表和每个分区的信息。

q：退出，不保存。

s：创建一个空的 Sun 磁盘标识。

t：替换当前分区的类型。

u：修改分区表的条目的显示单位，单位必须为柱面或扇区。

v：验证。

w：保存修改并退出。

x：进出高级操作模式，专家级操作指令。

注意 用 fdisk 命令完成对磁盘的分区工作后，该磁盘仍不具备存储文件的条件，还需要对各个分区按照所需的文件系统类型进行格式化，然后各分区才可以使用。

一个设备的名称表示整个磁盘，而分区则用设备名加上一个分区号来表示。例如，/dev/sda1 表示系统中的第一块磁盘的第一个分区。

实例 1 查看系统中所有磁盘的分区情况，执行如下命令。

```
[root@localhost ~]# fdisk -l
```

执行命令后的结果如下所示，执行后的输出如下所示。

```
[root@localhost ~]# fdisk -l
Disk /dev/sda: 8589 MB, 8589934592 bytes
255 heads, 63 sectors/track, 1044 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *           1          13      104391   83  Linux
/dev/sda2              14        1044     8281507+  8e  Linux LVM
Disk /dev/sdb: 1031 MB, 1031798784 bytes
16 heads, 32 sectors/track, 3936 cylinders
Units = cylinders of 512 * 512 = 262144 bytes
   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1               1        3936     1007600   e  W95 FAT16 (LBA)
```

系统中所有磁盘的分区信息如图 20-4 所示。

```
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]# fdisk -l

Disk /dev/sda: 8589 MB, 8589934592 bytes
255 heads, 63 sectors/track, 1044 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *           1           6       48163+  83  Linux
/dev/sda2               7         515     4088542+  83  Linux
/dev/sda3             516         949     3486105   83  Linux
/dev/sda4             950        1044      763087+   5  Extended
/dev/sda5             950        1044      763056   82  Linux swap / Solaris

Disk /dev/sdb: 4063 MB, 4063232000 bytes
5 heads, 32 sectors/track, 49600 cylinders
Units = cylinders of 160 * 512 = 81920 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1   *          51       49600     3963968    b  W95 FAT32
[root@localhost ~]#
[root@localhost ~]#
```

图 20-4 系统中所有磁盘的分区信息

从图20-4可以看出当前系统中有两块磁盘，第一块是SCSI磁盘/dev/sda，大小为8 589 MB。该磁盘有 255 个磁头，63 个扇区/磁道，1 044 个柱面，以柱面为计量单位，每个柱面的大小为 8 225 280 B。第二块是 U 盘/dev/sdb，大小为 4 063 MB。该 U 盘有 5 个磁头，32 个扇区/磁道、49 600 个柱面，以柱面为计量单位，每个柱面的大小为 81 920 B。

第 1 块磁盘是 SCSI 磁盘，包括 4 个分区/dev/sda1、/dev/sda2、/dev/sda3 和/dev/sda4。其中第 1 个分区/dev/sda1 是可启动分区（用“*”标记），分区的起始柱面为 1，结束柱面为 6，共有 48 163 个区块，分区文件系统编号为 83（表示是 Ext2/Ext 文件系统），文件系统类型是“Linux”。第 2 个分区是/dev/sda2，分区的起始柱面为 7，结束柱面为 515，共有 4 088 542 个区块，分区文件系统编号为 83，文件系统类型是“Linux”。第 3 个分区是/dev/sda3，分区的起始柱面为 516，结束柱面为 949，共有 3 486 105 个区块，分区文件系统编号为 83，文件系统类型是“Linux”；第 4 个分区是/dev/sda4，分区的起始柱面为 950，结束柱面为 1 044，共有 763 087 个区块，分区文件系统编号为 5，文件系统类型是扩展分区。/dev/sda5 是该扩展分区下的逻辑分区。

第 2 块磁盘是 U 盘，包括 1 个分区/dev/sdb1，该分区是可启动分区，分区的起始柱面为 51，结束柱面为 49 600，共有 3963968 个区块，分区文件系统编号为 b（表示是 FAT32 文件系统），文件系统类型是“w95 FAT32”。

实例 2 查看系统中所有磁盘的分区情况，以扇区为单位。执行如下命令。

```
[root@localhost ~]# fdisk -lu
```

执行命令后的结果如下所示，显示了系统中所有磁盘的分区信息。

```
[root@localhost ~]# fdisk -lu
Disk /dev/sda: 8589 MB, 8589934592 bytes
255 heads, 63 sectors/track, 1044 cylinders, total 16777216 sectors
Units = sectors of 1 * 512 = 512 bytes
   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *          63      208844      104391   83  Linux
/dev/sda2          208845    16771859     8281507+  8e  Linux LVM
Disk /dev/sdb: 1031 MB, 1031798784 bytes
16 heads, 32 sectors/track, 3936 cylinders, total 2015232 sectors
Units = sectors of 1 * 512 = 512 bytes
   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1              32     2015231     1007600    e  W95 FAT16 (LBA)
```

注意 带有“-u”参数的 fdisk 命令会以扇区为单位显示输出结果。每个扇区的大小为 512 B，并且用扇区数目取代柱面数目来标识每个分区的起始和终止地址，如上所示。

实例 3 为/dev/sdb 创建分区。

要求创建两个分区，一个为主分区，大小为整个硬盘大小的 25%；另外一个为逻辑分区，大小为整个硬盘大小的 75%。

fdisk 对于分区创建、删除的操作是通过交互式菜单来完成的，具体操作步骤如下。

第 1 步 在进行分区操作之前，显示当前的磁盘分区情况，执行如下命令。

```
[root@localhost ~]# fdisk -l /dev/sdb
```

执行命令后的结果如下所示，显示了当前的磁盘分区情况。

```
[root@localhost ~]# fdisk -l /dev/sdb
Disk /dev/sdb: 1031 MB, 1031798784 bytes
16 heads, 32 sectors/track, 3936 cylinders
Units = cylinders of 512 * 512 = 262144 bytes
   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1               1        3936     1007600    e  W95 FAT16 (LBA)
```

第 2 步，打开/dev/sdb。

在虚拟终端窗口，用 fdisk 打开/dev/sdb，如下所示。

```
[root@Localhost ~]#fdisk /dev/sdb
The number of cylinders for this disk is set to 3936.
There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:
1) software that runs at boot time (e.g., old versions of LILO)
2) booting and partitioning software from other OSs
   (e.g., DOS FDISK, OS/2 FDISK)
Command (m for help):
```

第 3 步，查看帮助。

由于初次接触 Linux 下的 fdisk，需要先看看它内置的帮助信息。在“Command (m for help)：”提示符下，输入“m”即可查看帮助信息，如下所示。

```
Command (m for help) : m
Command action
   a   toggle a bootable flag
   b   edit bsd disklabel
   c   toggle the dos compatibility flag
   d   delete a partition
   l   list known partition types
   m   print this menu
   n   add a new partition
   o   create a new empty DOS partition table
   p   print the partition table
   q   quit without saving changes
   s   create a new empty Sun disklabel
   t   change a partition's system id
   u   change display/entry units
   v   verify the partition table
   w   write table to disk and exit
   x   extra functionality (experts only)
```

下面就各子命令的含义描述如下。

n：创建一个新分区。

p：显示当前分区情况。

d：删除一个分区。

q：不存盘退出。

w：存盘退出。

c：切换 DOS 启动标识。

第 4 步，创建主分区。

在“Command (m for help) :”提示符下，依次输入以下指令完成主分区的创建，上面已经讲到，键入 n 表示开始创建一个新分区，如下所示。

```
Command (m for help) :n
Command action
l logical (5 or over)
p primary partition (1-4)
p                                                    ; #输入“p”创建一个主分区
Partition number (1-4):    1                         ; #主分区号为“1”
First cylinder (1-32635, default 1): <回车>          ; #直接按下 Enter 键，设置开始位置为 1
Using default value 1                                ; #表明当前创建的分区起始柱面为 1
Last cylinder or +size or +sizeM or +sizeK (1-19581, default 19581): +67G
                                                     ; #分区大小为 67 GB，因为当前硬盘总大
                                                     #小为 250 GB
```

第 5 步，创建扩展分区

在“Command (m for help) :”提示符下，依次输入以下指令完成扩展分区的创建，如下所示。

```
Command (m for help) :n
Command action
     l logical (5 or over)
     p primary partition (1-4)
e                                                    ; #输入“e”创建一个扩展分区
Partition number (1-4):    2                         ; #扩展分区号为“1”
First cylinder (8148-32635, default 1): <回车>       ; #直接按下 Enter 键，设置开始位
                                                     #置为 8148
Using default value 8148
Last cylinder or +size or +sizeM or +sizeK (8148-32635, default 32635):<回车>
                                                     ; #直接按下 Enter，将剩余空间都
                                                     #用来划分为一个扩展分区
```

第 6 步，创建逻辑分区。

在“Command (m for help) :”提示符下，依次输入以下指令完成逻辑分区的创建，如下所示。

```
Command (m for help) :n
Command action
     l logical (5 or over)
     p primary partition (1-4)
l                                                    ; #输入“1”创建一个逻辑分区
First cylinder (8148-32635, default 1): <回车>       ; #直接按下 Enter 键，设置开始位
                                                     #置为 8148
Using default value 8148
Last cylinder or +size or +sizeM or +sizeK (8148-32635, default 32635):<回车>
                                                     ; #直接按下 Enter 键，将扩展分区
                                                     #中的空间都用来划分为一个逻辑分区
```

第 7 步，输入子命令 p，可以查看指定磁盘/dev/sdb 新建分区后的分区情况。

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

第 8 步，存盘退出。

至此，所有硬盘分区的创建工作完毕，在“Command (m for help) :”提示符下，输入以下指令存盘退出。

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

如果不需要保存前面的操作内容，则可以通过键入 q 来实现。

注意 如果需要删除刚创建的分区/dev/sdb1，则可以通过键入“d”来实现。

（2） parted

parted 是一款基于 GPL 协议的硬盘分区开源软件，主要用于磁盘分区或调整分区大小，还可以为新安装的操作系统创建分区空间，重新分配磁盘分区空间等。通过此工具可以帮助用户创建、清除、调整、移动和复制 Ext2、Ext3、linux-swap、FAT、FAT32 和 reiserfs 分区；也能创建、调整和移动苹果系统的 HFS 分区；还能检测 jfs、ntfs、ufs 和 xfs 分区。

parted 除了其特殊功能——重新划分分区大小外，它的其他功能与 fdisk 等工具没有太大差别。

parted 命令的执行方式可以有命令行方式和交互式方式两种。其中，命令行方式指将 parted 具体要完成的命令附在命令行中一次性完成；交互式方式是指首先执行 parted 指令，但在命令行中并不给出具体子命令的描述，而是在进入交互式界面时才执行具体工作任务命令。

parted 的语法格式如下。

```
[root@Localhost ~]# parted [options] [DEVICE] [COMMAND] [PARAMETERS]
```

各参数含义的描述如下。

OPTIONS：

-h：显示帮助信息。

-i：交互式模式（默认模式）。parted 也支持命令行操作。

-l：显示所有设备上的分区信息。

-s：执行脚本。

-v：显示版本信息。

COMMANDS：命令行模式支持的指令。

parted 支持命令行和交互式两种模式，不带任何参数打开的是交互式。在交互式下，当显示“(parted)”提示符时，用户输入一条指令，parted 执行并将结果显示出来，然后等待用户的下一条指令。

交互模式下的指令很多，可以在“(parted)”提示符下输入“help”显示详细的帮助信息。下面介绍几种常用的指令及格式、作用。

- check——检查分区

格式：check NUMBER

检查分区错误，分区号由 NUMBER 指定。需要注意的是，要检查的分区必须是未加载分区。

- cp——复制分区

格式：cp [FROM-DEVICE] From-Number To-Number

将分区 From-Number 上的文件系统完整地复制到分区 To-Number 中。分区 To-Number 上的数据会被全部删除（目标分区必须比源分区要大）。如果要在不同设备间复制文件系统，你可以通过[FROM-DEVICE]指令来指定来源硬盘的名称。

cp 命令支持 Ext2、Ext3、FAT16、FAT32、Linux Swap 和 Reiserfs 文件系统。

- help——获取帮助

格式：help [command]

显示一条指令的帮助信息。

- mklabel——建立分区表

格式：mklabel Label-Type

建立一个新的分区表，Label-Type（分区表类型）可以是 aix、amiga、bsd、dvh、gpt、mac、msdos、sun 和 loop 之一。在一个硬盘上创建分区表将会导致原来的分区全部丢失（如果你还没有新建分区，可以通过 rescue 指令恢复）。

- mkfs——创建文件系统

格式：mkfs NUMBER Fs-Type

在 NUMBER 分区上创建 Fs-Type 指定的文件系统。Fs-Type 只能是以下之一：Ext2、FAT16、FAT32、Linux-Swap 和 reiserfs。

- mkpart——建立新分区

格式：mkpart Part-Type [Fs-Type] Start End

建立一个新的分区。若要建立非扩展分区，就最好用 Fs-Type 来指定文件系统。Start 和 End 是新分区开始和结束的具体位置。

Part-Type 可以是 Primary（主分区）、extended（扩展分区）和 logical（逻辑分区 ）三者之一。

注意 FS-Type 必须是 Ext2、Ext3、FAT16、FAT32、hfsx、hfs+、jfs、linux-swap、NTFS、reiserfs、hp-ufs、sun-ufs、xfs、apfs2、apfs1、apfs、amufs5、amufs4、amufs3、amufs2、amufs1、amufs0、amufs、affs7、affs6、affs5、affs4、affs3、affs2、affs1 和 affs0 之一。

Start 和 End 表示磁盘的位置，可以是 4 GB（如果没有指定单位，则默认单位是 MB），也可以是 10%，也可是是负数，比如-1s 表示最后一个扇区。

- mkpartfs——建立分区及其文件系统

格式：mkpartfs Part-Type Fs-Type Start End

建立一个新的分区，并在上面建立文件系统，相当于 mkpart 和 mkfs 的合体。

- move——移动分区

格式：move NUMBER Start End

该命令能将选定的分区在硬盘中移动。Start 指定一个新的分区的起始位置，End 指定分区的结束位置（新分区大小必须大于或者等于原来分区）的大小。

该命令支持 Ext2、Ext3、FAT16、FAT32、Linux-Swap 和 Reiserfs 分区。

- name——命名

格式：name NUMBER Name

为编号所指定的分区命名（只支持 GPT、Mac、MIPS 和 PC98 格式的分区）。

- print——显示分区信息

格式：print [NUMBER]

显示当前硬盘的分区信息或者某个分区的信息。

- quit——退出

格式：quit

退出 parted，所有分区操作开始生效。

- rescue——恢复分区

格式：rescue Start End

如果不小心删除了一个分区，可以尝试用 rescure 指令恢复。只要通过 Start 和 End 指令指定分区所在的大概位置，parted 就会去自动搜索。如果找到分区，则会询问是否重建该分区。

- resize——调整分区大小

格式：resize NUMBER Start End

调整指定编号的分区大小。新分区的开始位置和结束位置由 Start 和 End 决定。需要注意的是，如果缩小了分区，那么应保证新分区能够容纳老分区里面的内容。

该命令支持 Ext2、Ext3、FAT16、FAT32、HFS、HFS+、HFSX、Linux-Swap 和 Reiserfs 分区。

- rm——删除分区

格式：rm NUMBER

将编号指定的分区删除。

- select——选择设备

格式：select DEVICE

选择 parted 将要处理的磁盘设备。这个磁盘设备可以是一个 Linux 硬盘设备，一个软盘，一个软 RAID 磁盘阵或者是 LVM 逻辑卷。

- set——设置标记

格式：set NUMBER Flag State

更改编号指定的分区的标记。具体标记类型是由分区表决定的。标记有两种状态，要么是 on（生效），要么是 off（失效）。

boot：用于 Mac、MS-DOS 和 PC98 分区表。拥有该标记的分区可以引导。

lba：用于 MS-DOS 分区表。该标记告诉 MS DOS，Windows 使用 LBA（Logic Block Addressing）模式。

root：用于 MAC 分区表。如果某分区是 Linux 系统的根分区，那么就要使该分区启用这个标记。

swap:用于 Mac 分区表。 如果某分区是 Linux 系统的交换分区，那么就要使该分区启用这个标记。

hidden：用于 MS-DOS 和 PC98 分区表。启用这个标记能使一个分区在 Windows 下“隐藏”起来。

raid：用于 MS-DOS 分区表。该标记能够告诉 Linux 系统这是一个软 RAID 磁盘阵。

LVM：用于 MS-DOS 分区表。启用这个标记能够告诉 Linux 系统这是一个 LVM 逻辑。

- unit——设置单位

格式：unit UNIT

为 parted 选择一个表示磁盘大小的单位。默认是以 MB 为单位的，可以设置为 GB。其支持设置的单位如下。

s：sector（扇区，大小为 512 B）。

B：byte（B）。

kB：kilobyte（1 000 B）。

MB：megabyte（1 000 000 B）。

GB：gigabyte （1 000 000 000 B）。

TB：terabyte (1 000 000 000 000 字节)。

%：百分比，数字必须在 0~100 之间。

cyl：柱面。

chs：柱面（cylinders）、磁头（heads）、扇区（sectors）的地址。

compact 输入时以 MB 作为单位。

注意 parted 命令的执行方式可以有两种：其一是命令行方式，即将 parted 具体要完成的命令指令附在命令行中一次性完成；其二是交互式方式，即首先执行 parted 指令，但在命令行中并不给出具体子命令的描述，而是在进入交互式界面后完成具体工作任务命令。

实例 1 查看硬盘分区表。

比如要看看/dev/sda 分区信息，可以执行以下步骤。

在 Linux 下输入以下指令。

```
[root@Localhost ~]# parted /dev/sda
```

结果如下所示。

```
[root@localhost ~]# parted /dev/sda
GNU Parted 1.8.1
使用 /dev/sda
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) print
Model: VMware, VMware Virtual S (scsi)
Disk /dev/sda: 8590MB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Number  Start   End     Size    Type     File system  标志
 1      32.3kB  107MB   107MB   主分区   ext3         启动
 2      107MB   8587MB  8480MB  主分区                lvm
(parted)
```

实例 2 创建/dev/sdb 分区。

第 1 步，输入以下指令打开 parted。

```
[root@Localhost ~]#parted /dev/sdb
```

第 2 步，创建分区表。

在“(parted)”提示符下，输入以下指令创建一个 msdos 格式的分区表，如图 20-5 所示。

```
(parted) mklabel msdos
```

```
(parted) mklabel msdos
警告: 正在使用 /dev/sda 上的分区。
忽略/Ignore/放弃/Cancel? Ignore
警告: The existing disk label on /dev/sda will be destroyed and all data on this
disk will be lost. Do you want to continue?
是/Yes/否/No? yes
新的磁盘标签类型?  [msdos]? msdos
(parted) parted
```

图 20-5　msdos 格式分区表创建成功

第 3 步，创建一个主分区。

先把其中的 25%拿来创建一个主分区，并格式化为 Ext2 格式，如图 20-6 所示。

```
(parted) mkpartfs PRIMARY EXT2 0 25%
```

```
(parted) mkpartfs PRIMARY EXT2 0 25%
(parted) print

Model: VMware, VMware Virtual S (scsi)
Disk /dev/sda: 12.9GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos

Number  Start   End     Size    Type     File system  标志
 1      32.3kB  3224MB  3224MB  主分区   ext2

(parted)
```

图 20-6　创建一个 Linux Ext2 的主分区

第 4 步，创建扩展分区。

用剩余空间创建一个扩展分区，大小为硬盘空间的 75%，如图 20-7 所示。

```
(parted) mkpart extended 25% 100%
```

```
(parted) mkpart extended 25% 100%
(parted) print

Model: VMware, VMware Virtual S (scsi)
Disk /dev/sda: 12.9GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos

Number  Start   End     Size    Type      File system  标志
 1      32.3kB  3224MB  3224MB  主分区    ext2
 2      3224MB  12.9GB  9656MB  扩展分区                lba

(parted)
```

图 20-7　扩展分区创建成功

第 5 步，创建逻辑分区。

将扩展分区中的所有空间划分为一个逻辑分区，如图 20-8 所示。

```
(parted)mkpartfs logical EXT2 25% 100%
```

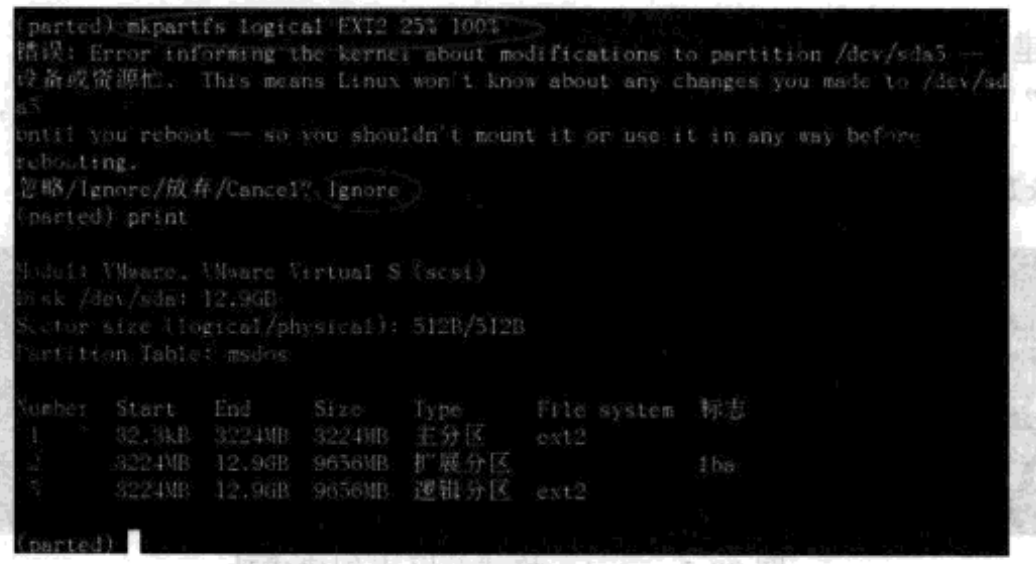


图 20-8　逻辑扩展分区创建中

第 6 步，退出。

完成分区创建任务后，执行“quit”指令退出。

```
(parted)quit
```

Information: don't forget to update /etc/fstab，　if necessary.

实例 3　调整磁盘/dev/sdb 上的分区/dev/sdb2 的大小为 50 MB，执行如下指令，运行结果如图 20-9 所示。

```
[root@localhost ~]# parted /dev/sda
<parted> print 2
<parted> resize
Partition number? 2
Start? [494MB]? 494
End? [1069MB]? 544
<parted> print 2
```

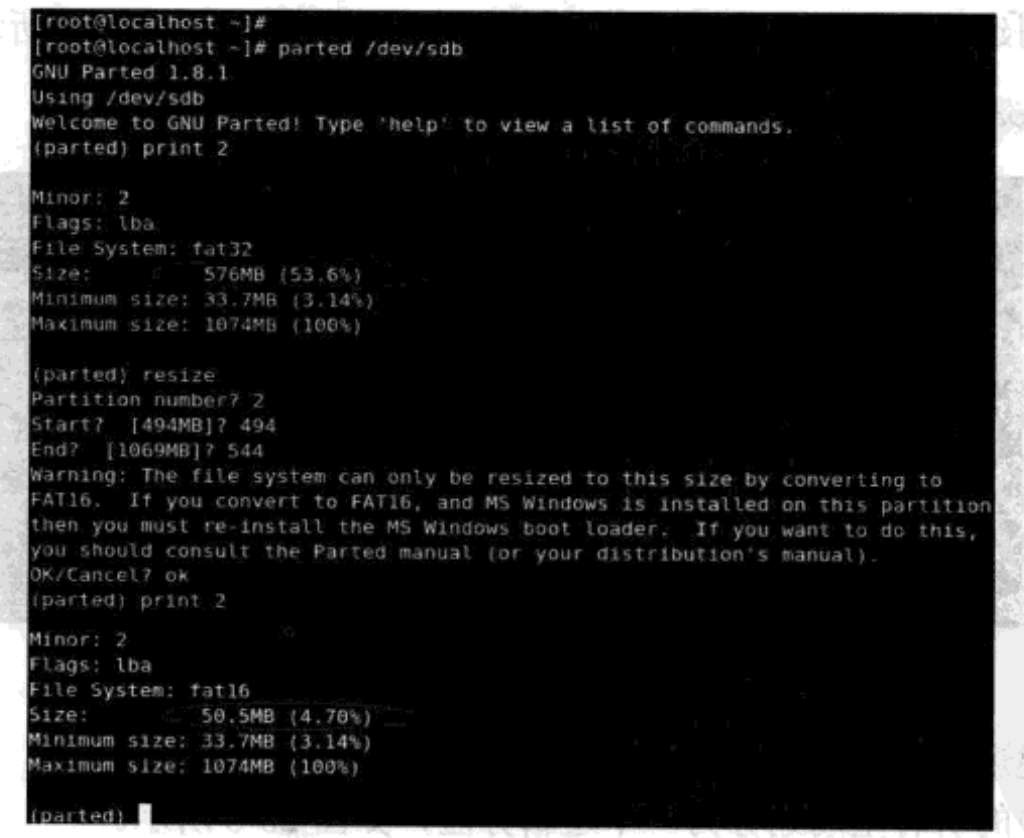


图 20-9　调整分区/dev/sdb2 的大小

实例 4 救援磁盘分区/dev/sdb2，如图 20-10 所示。

```
[root@localhost ~]# parted /dev/sda
<parted> print
<parted> rm 2
<parted> print
<parted> rescue 494 544
<parted> print
```

```
[root@localhost ~]#
[root@localhost ~]# parted /dev/sdb
GNU Parted 1.8.1
Using /dev/sdb
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) print

Model: VMware, VMware Virtual S (scsi)
Disk /dev/sdb: 1074MB
Sector size (logical/physical): 512B/512B
Partition Table: msdos

Number  Start   End     Size    Type     File system  Flags
 1      32.3kB  494MB   493MB   primary  ext2
 2      494MB   544MB   50.5MB  primary  fat16

(parted) rm 2
(parted) print

Model: VMware, VMware Virtual S (scsi)
Disk /dev/sdb: 1074MB
Sector size (logical/physical): 512B/512B
Partition Table: msdos

Number  Start   End     Size    Type     File system  Flags
 1      32.3kB  494MB   493MB   primary  ext2

(parted) rescue 494 544
Information: A fat16 primary partition was found at 494MB -> 544MB.  Do you
want to add it to the partition table?
Yes/No/Cancel? yes
(parted) print

Model: VMware, VMware Virtual S (scsi)
Disk /dev/sdb: 1074MB
Sector size (logical/physical): 512B/512B
Partition Table: msdos

Number  Start   End     Size    Type     File system  Flags
 1      32.3kB  494MB   493MB   primary  ext2
 2      494MB   544MB   50.5MB  primary  fat16

(parted)
```

图 20-10 救援分区/dev/sdb2

2. 磁盘格式化

在 Linux 操作系统中，分区之后的磁盘分区需要格式化后才能使用。Linux 系统下的磁盘格式化指令包括 mkfs、mkdosfs、mke2fs、mkfs.ext2、mkfs.ext3 等。

（1） mkfs

该命令主要用于在设备或磁盘分区上构造一个文件系统。

注意 mkfs 命令中的“文件系统”是一个设备名（如分区名/dev/sda1、/dev/hda1、/dev/fd0 等）或一个文件挂载点（如/，/usr，/home 等）。

mkfs 的命令格式如下。

```
[root@localhost ~]# mkfs [-V][fs][-t <文件系统类型 fs-type>][文件系统][区块]
```

各参数含义的描述如下。

fs：指定建立文件系统时的参数。

-c：在构造文件系统之前检查设备上是否有损坏的区块。

-l file：从指定的文件中读取文件系统中损坏区块的信息。

-v：显示版本信息与详细的使用方法。

-V：显示详细信息，包括正在执行的所有特定文件系统专用的命令参数。本参数主要用于测试。

-t<fs-type>：指定要建立何种文件系统，如果不指定则在/etc/fstab 文件中找出相符的资料来推测决定。无法推测时，默认使用预设值（Ext2）。

-V：显示版本信息和其他命令的简要使用方法，注意该选项会抑制其他命令，故通常测试时使用。

注意 mkfs 指令本身并不执行建立文件系统的工作，而是根据-t 参数指定的文件系统类型去调用相关的 mkfs 的适当版本程序来执行。

出于对 Linux 各个发行版本的兼容性的考虑，建立同一 Linux 文件系统的命令可能有不同的名称，而实际上它们的功能是相同的，如 mkfs.ext2、mkfs.ext3 和 mke2fs 都可以用来创建 Ext2/Ext3 文件系统。

实例 1 在设备/dev/hda1 上建一个 msdos 的文件系统，同时检查是否有坏块存在，并且将过程详细展现出来，如下所示。

```
[root@localhost ~]# mkfs -V -t msdos -c /dev/hda1
mkfs (util-linux 2.13-pre7)
mkfs.msdos -c /dev/sdb1
mkfs.msdos 2.11 (12 Mar 2005)
```

然后再执行命令查看结果。

```
[root@localhost ~]# fdisk -l /dev/sdb
Disk /dev/sdb:213 MB,213909504 bytes
64 heads, 32 sectors/track,204 cylinders
Device Boot  Start  End  Blocks   Id  System
/dev/sdb1    1     204   208880  83   Linux
```

实例 2 在设备/dev/sdb1 上创建一个 Ext3 的文件系统，通过以下指令实现。

```
[root@localhost ~]# mkfs -t ext2 -j /dev/sdb1
```

或

```
[root@localhost ~]# mke2fs -j /dev/sdb1
```

（2） mkdosfs

该命令用于在 Linux 系统中指定的设备或磁盘分区上构造一个 MS-DOS 文件系统。其命令格式如下。

```
[root@localhost ~]# mkdosfs [-cv][-f <FAT 数目>][-F <FAT 记录的单位>][-i <磁盘序号>] [-l <文件名>][-m <信息文件>][-n <标签>][-r <根目录项目数>] [-s <扇区数>][设备名称][区块数]
```

各参数含义的描述如下。

-c：建立文件系统之前检查是否有损坏的区块。

-f<FAT 数目>：指定 FAT 的数目，目前支持 1 与 2 两种选项。

-F<FAT 记录的单位>：指定 FAT 记录的单位为 12 位或 16 位。

-i<磁盘序号>：设置文件系统的磁盘序号。

-l<文件名>：从指定的文件中，读取文件系统中损坏区块的信息。

-m<信息文件>：若文件系统正常开机，而操作系统却没有正常安装，则会显示选项中所指定的信息文件内容。

-n<标签>：设置文件系统的磁盘标签名称。

-r<根目录项目数>：设置根目录最多能记载项目数。

-s<扇区数>：设置每个簇包含扇区的个数。

-v：显示版本信息与详细的使用方法。

注意 带“-l<文件名>”参数的 mkdosfs 命令可以从指定的文件中读取文件系统中损坏扇区的信息。可以使用 badlocks 命令将发现损坏的扇区存储在指定的文件中。

“-s”参数在使用时需注意，设置的数值必须为 2 的幂次方。

实例 1 在整个 U 盘上建立一个 MS-DOS 文件系统，执行如下命令。

```
[root@localhost ~]# mkdosfs -I /dev/sdb
```

执行命令后的结果如下所示。

```
[root@localhost ~]# mkdosfs -I /dev/sdb
mkdosfs 2.11 (12 Mar 2005)
[root@localhost ~]# parted /dev/sdb print
Model:  USB FLASH DRIVE (scsi)
Disk /dev/sdb: 1032MB
Sector size (logical/physical): 512B/512B
Partition Table: loop
Number  Start   End     Size    File system  标志
 1      0.00kB  1032MB  1032MB  fat32
信息：如果必要，不要忘记更新 /etc/fstab。
```

由以上输出的信息可以看到，屏幕上将会显示 MS-DOS 文件系统的版本号信息，接下来便可以使用 mdir 等 mtools 系列命令管理在 U 盘上的 MS-DOS 文件系统。

实例 2 在整个 U 盘上建立一个 MS-DOS 文件系统，并配置根目录下最多可包含 16 个文件或子目录。执行如下命令。

```
[root@localhost ~]# mkdosfs -v -r 16 -I /dev/sdb
```

执行命令后的结果如下所示。

```
[root@localhost ~]# mkdosfs -v -r 16 -I /dev/sdb
mkdosfs 2.11 (12 Mar 2005)
Auto-selecting FAT32 for large filesystem
/dev/sdb has 32 heads and 62 sectors per track,
```

```
logical sector size is 512,
using 0xf8 media descriptor, with 2015232 sectors;
file system has 2 32-bit FATs and 8 sectors per cluster.
FAT size is 1965 sectors, and provides 251408 clusters.
Volume ID is 4a0adf39, no volume label.
[root@localhost ~]# parted /dev/sdb print
Model:  USB FLASH DRIVE (scsi)
Disk /dev/sdb: 1032MB
Sector size (logical/physical): 512B/512B
Partition Table: loop
Number  Start   End     Size    File system  标志
 1      0.00kB  1032MB  1032MB  fat32
信息: 如果必要，不要忘记更新 /etc/fstab。
```

执行以上命令后，在 U 盘上创建一个 MS-DOS 文件系统，并配置根目录下的文件及目录数量。如果创建 MS-DOS 文件系统时，在没有设置根目录下最多可包含的文件或子目录个数的情况下，系统会给出一个 224 的默认值。

执行如下命令。

```
[root@localhost ~]# mkdosfs -I -v -n DOS-Disk /dev/sdb
```

执行命令后的结果如下所示。

```
[root@localhost ~]# mkdosfs -v -n DOS-Disk -I /dev/sdb
mkdosfs 2.11 (12 Mar 2005)
Auto-selecting FAT32 for large filesystem
/dev/sdb has 32 heads and 62 sectors per track,
logical sector size is 512,
using 0xf8 media descriptor, with 2015232 sectors;
file system has 2 32-bit FATs and 8 sectors per cluster.
FAT size is 1965 sectors, and provides 251408 clusters.
Volume ID is 4a0ae24f, volume label DOS-Disk   .
```

由以上输出的结果可以看出，此命令执行的最后一行信息显示了设置的卷标为 DOS-Disk。如果创建 MS-DOS 文件系统时没有设置卷标，则卷标为空。

（3） mke2fs

该命令主要用于在 Linux 系统下，在指定的设备或磁盘分区上构造一个 Linux Ext2 或 Ext3 文件系统。其命令格式如下。

```
[root@Localhost ~]# mke2fs [-cFjMqrSvV][-b <区块大小>][-f <不连续区段大小>][-i <字节>] [-N <inode 数>][-l <文件>][-L <标签>][-m <百分比值>][-R=<区块数>] [设备名称][区块数]
```

各参数含义的描述如下。

-b<区块大小>：指定区块大小，单位为 B。

-c：检查是否有损坏的区块。

-f<不连续区段大小>：指定不连续区段的大小，单位为 B。

-F：不管指定的设备是什么，强制执行 mke2fs。

-i<字节>：指定“字节/inode”的比例。

-j：建立一个 Ext3 日志文件系统。

-N<inode 数>：指定要建立的 inode 数目。

-l<文件>：从指定的文件中读取文件系统中损坏区块的信息。

-L<标签>：设置文件系统的标签名称。

-m<百分比值>：指定给管理员保留区块的比例，预设为 5%。

-M：记录最后一次挂入的目录。

-q：执行时不显示任何信息。

-r：指定要建立的 Ext2 文件系统版本。

-R=<区块数>：设置磁盘阵列参数。

-S：仅写入 superblock 与 group descriptors，而不更改 inode able、inode bitmap 以及 block bitmap。本参数的作用在于修复损坏的 superblock 与 group descriptors。

-v：执行时显示详细信息。

-V：显示版本信息。

实例 1 将整个 U 盘/dev/sdb 建立成 Ext2 文件系统，通过以下指令完成。

```
[root@localhost ~]# mke2fs /dev/sdb
Mke2fs 1.39 (29-May-2006)
/dev/sdb is entire device,not just one partition!
Processd anyway?(y,n)y
```

实例 2 将整个 U 盘/dev/sdb 建立成 Ext2 文件系统，并设置卷标为 Ext2-Disk，通过以下指令实现。

```
[root@localhost ~]# mke2fs -L Ext2-Disk /dev/sdb
Mke2fs 1.39 (29-May-2006)
/dev/sdb is entire device,not just one partition!
Processd anyway?(y,n)y
Filesystem label = Ext2-Disk
```

实例 3 将整个 U 盘/dev/sdb 建立成 Ext2 文件系统，并设置扇区大小为 4 096 B，通过以下指令完成。

```
[root@localhost ~]# mke2fs -b 2048 /dev/sdb
Mke2fs 1.39 (29-May-2006)
/dev/sdb is entire device,not just one partition!
Processd anyway?(y,n)y
Filesystem label =
OS type: Linux
Block size = 4096(log = 1)
```

实例 4 将整个 U 盘/dev/sdb 建立成 Ext3 文件系统，并设置卷标为 Ext3-Disk，通过以下指令实现。

```
[root@localhost ~]# mke2fs -j -L Ext3-Disk /dev/sdb
Mke2fs 1.39 (29-May-2006)
/dev/sdb is entire device,not just one partition!
Processd anyway?(y,n)y
Filesystem label = Ext3-Disk
```

然后再通过执行以下命令进行查看。

```
[root@localhost ~]#parted /dev/sdb print
Model: VMware, VMware Virtual S (scsi)
Disk /dev/sdb:214MB
Sector size (logical/physical): 512B/512B
Partition Table: loop
Number  Start   End     Size    File system  标志
 1      0.00kB  214MB   214MB   ext3
Information:Don't forget to update /etc/fstab,if necessary.
```

实例 5 将整个 U 盘/dev/sdb 建立成 Ext2 文件系统，并设置管理员的保留区块的比例为 10%，通过以下指令实现。

```
[root@localhost ~]# mke2fs -m 10 /dev/sdb
Mke2fs 1.39 (29-May-2006)
/dev/sdb is entire device,not just one partition!
Processd anyway?(y,n)y
Filesystem label =
OS type: Linux
Block size = 1024(log = 0)
Fragment size = 1024(log = 0)
52416 inodes,208896 blocks
208896 blocks (10.00%) reserved for the super user
```

（4） mkisofs

在 Linux 系统环境中，其本身提供了一款制作 iso 文件的命令行工具软件 mkisofs。可以通过该命令将指定的目录与文件创建成为 ISO9660 格式的映像文件，以供刻录。其语法格式如下。

```
[root@localhost ~]# mkisofs [-adDfhJlLNrRTvz][-print-size][-quiet][-A <应用程序 ID>][-abstract <摘要文件>][-b <启动映像文件>][-biblio ][-c <启动文件名称>][-C <盘区编号，磁区编号>][-copyright <版权信息文件>][-hide <目录或文件名>][-hide-joliet <文件或目录名>][-log-file <记录文件>][-m <目录或文件名>][-M <启动映像文件>][-o <映像文件>][-p <数据处理人>][-P <光盘发行人>][-sysid <系统 ID >][-V <光盘 ID >][-volset <卷册集 ID>][-volset-size <光盘总数>][-volset-seqno <卷册序号>][-x <目录>][目录或文件]
```

各参数含义的描述如下。

-a 或--all：mkisofs 通常不处理备份文件。使用此参数可以把备份文件加到映像文件中。

-A<应用程序 ID>或-appid<应用程序 ID>：指定光盘的应用程序 ID。

-abstract<摘要文件>：指定摘要文件的文件名。

-b<启动映像文件>或-eltorito-boot<启动映像文件>：指定在制作可启动光盘时所需的启动映像文件。

-biblio：指定 ISBN（International Standard Book Number）文件的文件名，ISBN 文件位于光盘根目录下，记录光盘的 ISBN。

-c<启动文件名称>：制作可启动光盘时，mkisofs 会将启动映像文件中的全部内容作成一个文件。

-C<盘区编号，盘区编号>：将许多节区合成一个映像文件时，必须使用此参数。

-copyright<版权信息文件>：指定版权信息文件的文件名。

-d 或-omit-period：省略文件后的句号。

-D 或-disable-deep-relocation：ISO 9660 最多只能处理 8 层的目录，超过 8 层的目录，RRIP 会自动将它们设置成 ISO 9660 兼容的格式。使用-D 参数可关闭此功能。

-f 或-follow-links：忽略符号连接。

-h：显示帮助。

-hide<目录或文件名>：使指定的目录或文件在 ISO 9660 或 Rock Ridge Extensions 的系统中隐藏。

-hide-joliet<目录或文件名>：使指定的目录或文件在 Joliet 系统中隐藏。

-J 或-joliet：使用 Joliet 格式的目录与文件名称。

-l 或-full-iso9660-filenames：使用 ISO 9660 中 32 位字符长度的文件名。

-L 或-allow-leading-dots：允许文件名的第一个字符为句点。

-log-file<记录文件>：在执行过程中若有错误信息，预设会显示在屏幕上。

-m<目录或文件名>或-exclude<目录或文件名>：指定的目录或文件名将不会放入映像文件中。

-M<映像文件>或-prev-session<映像文件>：与指定的映像文件合并。

-N 或-omit-version-number：省略 ISO 9660 文件中的版本信息。

-o<映像文件>或-output<映像文件>：指定映像文件的名称。

-p<数据处理人>或-preparer<数据处理人>：记录光盘的数据处理人。

-print-size：显示预估的文件系统大小。

-quiet：执行时不显示任何信息。

-r 或-rational-rock：使用 Rock Ridge Extensions，并开放全部文件的读取权限。

-R 或-rock：使用 Rock Ridge Extensions。

-sysid<系统 ID>：指定光盘的系统 ID。

-T 或-translation-table：建立文件名的转换表，适用于不支持 Rock Ridge Extensions 的系统。

-v 或-verbose：执行时显示详细的信息。

-V<光盘 ID>或-volid<光盘 ID>：指定光盘的卷册集 ID。

-volset-size<光盘总数>：指定卷册集所包含的光盘张数。

-volset-seqno<卷册序号>：指定光盘片在卷册集中的编号。

-x<目录>：指定的目录将不会放入映像文件中。

-z：建立通透性压缩文件的 SUSP 记录，此记录目前只在 Alpha 机器上的 Linux 环境下有效。

实例 1 将目录/root/hhwork 下的所有文件及目录建立成名为 hhwork.iso 的光盘映像文件，通过以下指令完成。

```
[root@localhost ~]# mkisofs -o hhwork.iso /root/hhwork/
INFO:  UTF-8 character encoding detected by locale settings.
       Assuming UTF-8 encoded filenames on source filesystem,
       use -input-charset to override.
Total translation table size: 0
Total rockridge attributes bytes: 0
Total directory bytes: 0
```

```
Path table size(bytes): 10
Max brk space used 0
175 extents written (0 MB)
```

然后通过执行以下命令查看结果。

```
[root@localhost ~]# ls -l hhwork.iso
-rw-r--r-- 1 root stella 358400 07-21 22:46 hhwork.iso
```

实例2 将目录/root/demo下的除文件install.log和目录tmp以外的所有文件及目录建立成名为demo.iso的光盘映像文件，通过执行以下命令实现。

```
[root@localhost ~]# ls -l /root/demo
[root@localhost ~]# mkisofs -x tmp -m install.log -o demo.iso /root/demo
[root@localhost ~]# mkdir /mnt/cdrom
[root@localhost ~]# mount -o loop demo.iso /mnt/cdrom
[root@localhost ~]# ls -l /mnt/cdrom
```

实例 3 由目录/root/demo 建立名为 demo.iso 的光盘映像文件，并使用 Rock Ridge Extensions 效果，通过执行以下命令实现。

```
[root@localhost ~]# mkisofs -r -o hhwork.iso /root/hhwork
[root@localhost ~]# mkisofs -r -o hhwork.iso /root/hhwork
INFO:   UTF-8 character encoding detected by locale settings.
        Assuming UTF-8 encoded filenames on source filesystem,
        use -input-charset to override.
Total translation table size: 0
Total rockridge attributes bytes: 250
Total directory bytes: 0
Path table size(bytes): 10
Max brk space used 0
176 extents written (0 MB)
```

注意 通过“-r”参数的mkisofs命令使用了Rock Ridge Extensions效果，在此模式下映像文件的同时可以保持文件的权限。将上述建立起来的光盘映像文件挂载到/mnt/cdrom上，然后将进行光盘映像文件内部的文件及目录信息与原始目录/root/hhwork下的文件和目录进行比较。可以很清晰地看到，映像文件内部的文件及目录比原始目录/root/demo下的文件和目录缺少了写权限以外，其他权限完全一致。

实例 4 由目录/root/hhwork、建立名为 hhwork.iso 的光盘映像文件，并将信息记录到demoiso.log文件中，同时指定光盘的发行人为Hhwork，通过以下指令完成。

```
[root@localhost ~]# mkisofs -log-file demoiso.log -publisher ""Hhwork" -r -o
hhwork.iso /root/hhwork
INFO:   UTF-8 character encoding detected by locale settings.
        Assuming UTF-8 encoded filenames on source filesystem,
        use -input-charset to override.
re-directing all messages to demoiso.log
```

实例 5 预计由目录/root/hhwork 建立光盘映像文件的大小，通过以下指令完成。

```
[root@localhost ~]# mkisofs -print -size -r /root/hhwork
INFO:   UTF-8 character encoding detected by locale settings.
        Assuming UTF-8 encoded filenames on source filesystem,
        use -input-charset to override.
Total extents scheduled to be written = 176
176
```

可以看出，该指令并不实际生成一个光盘镜像文件 iso，而仅仅是预计由指定目录生成 iso 文件的大小。

3. 加载分区

Linux 下对于分区的加载位置，并不是完全随意的，还是有些不成文的潜规则约束着用户，具体描述如下。

- “/”分区

根分区，主要用来保存系统引导到单用户模式所需要的核心系统文件。该分区一旦建立就不应该变化。

- “/usr”分区

用来保存系统文件的分区，比如用户工具、编译器等。Linux 系统需要的源代码也一般保存在该分区。

- “/var”分区

用来保存变量文件的分区，通常包括队列文件（电子邮件、打印缓存等）和日志文件。

- “/tmp”分区

临时分区。

- “/home”分区

用户家目录一般保存在这个分区中。

- “Swap”分区

交换分区，类似于 Windows 下的页面文件。

Red Hat Enterprise Linux 5 下的分区在系统启动时会被自动加载，其自动加载配置是保存在/etc/fstab 文件中的。

20.2.3 使用磁盘管理命令

在 Red Hat Enterprise Linux 5 中提供了一些用于管理和使用硬盘的命令，下面将学习如何使用这些工具更加有效地使用硬盘。

1. df

使用该命令可以查看文件系统磁盘的使用情况，其主要用于报告所有已安装文件系统上或指定文件系统上的剩余磁盘空间信息，而不报告未安装文件系统的情况，并将结果输出到标准输出设备上。其命令格式如下。

```
[root@localhost ~]# df [-ahHiklmPT][--block-size=<区块大小>][-t <文件系统类型
```

```
>] [-x <文件系统类型>][--help][--no-sync][--sync][--version][name]
```

各参数含义的描述如下。

-a 或--all：包括全部的文件系统。

--block-size=<区块大小>：以指定的区块大小来显示区块数目。

-h 或--human-readable：以可读性较高的方式来显示信息（例如 1.9 GB，而不是 1 967 156）。

-H 或--si：与-h 参数相同，但在计算时是以 1 000 B 为换算单位而非 1 024 B。

-i 或--inodes：报告空闲的、已用过的及部分用过的（百分比）索引节点 inode 的信息。

-k 或--kilobytes：指定区块大小以 KB 为单位。

-l 或--local：仅显示本地端的文件系统。

-m 或--megabytes：指定区块大小以 MB 为单位。

--no-sync：没有事先调用 sync（即没有刷新缓冲区）就输出结果。这是默认的情况。

-P 或--portability：使用 POSIX 的输出格式（即在一行上只输出一个文件系统的信息）。

--sync：在取得磁盘的使用信息前，先执行 sync 指令刷新缓冲区。

-t<文件系统类型>或--type=<文件系统类型>：仅显示指定文件系统类型的磁盘信息。

-T 或--print-type：除了显示每个文件系统的大小以外，还显示其类型信息。

-x<文件系统类型>或--exclude-type=<文件系统类型>：仅显示类型不是指定类型的文件系统的磁盘信息。

--help：显示帮助。

--version：显示版本信息。

name：指定磁盘设备。

实例 1 报告所有已安装文件系统上的剩余磁盘空间使用信息，执行如下命令。

```
[root@localhost ~]# df
```

执行命令后的结果如下所示，显示了所有已安装文件系统的磁盘总块数、已用空间、可用空间、已用百分比以及挂载点等信息。

```
[root@localhost ~]# df
文件系统              1K-块      已用      可用 已用% 挂载点
/dev/mapper/VolGroup00-LogVol00
                     6983168   6967248         0 100% /
/dev/sda1             101086     13204     82663  14% /boot
tmpfs                 183896         0    183896   0% /dev/shm
/dev/hdc              3004732   3004732         0 100% /media/RHEL_5.2 i386 DVD
```

可以看出，df 指令的输出共包括 6 栏。其中，第 1 栏代表文件系统对应的设备路径（一般是硬盘上的分区）；第 2 栏表示给出分区包含的数据块（以 1 024 B 为单位）的数目；第 3 栏表示已用数据块的数目；第 4 栏表示可用的数据块的数目；第 5 栏已用%表示普通用户空间使用的百分比。即使达到 100%，管理员因为保留有空间，仍然可以登录解决问题。第 6 栏表示文件系统的加载点。

注意 第 3、4 栏块数之和不等于第 2 栏的块数。这是因为默认的配置下每个分区都留了少量空间供系统管理员使用（这样即使遇到普通用户空间已满，管理员仍可以登录并具备解决问题所需的工作空间）。

实例 2 报告设备/dev/sda1 上的文件系统的剩余磁盘空间信息，执行如下命令。

```
[root@localhost ~]# df /dev/sda1
```

执行命令后的结果如下所示，显示了设备/dev/sda1 上的文件系统的剩余磁盘空间信息。

```
[root@localhost ~]# df /dev/sda1
文件系统              1K-块      已用     可用 已用% 挂载点
/dev/sda1               101086    13204     82663  14% /boot
```

实例 3 报告索引节点 inode 的使用信息，执行如下命令。

```
[root@localhost ~]# df -i
```

执行命令后的结果如下所示。

```
[root@localhost ~]# df -i
文件系统              Inode (I)已用 (I)可用 (I)已用% 挂载点
/dev/mapper/VolGroup00-LogVol00
                     1802240  181295 1620945   11% /
/dev/sda1              26104      35   26069    1% /boot
tmpfs                  45974       1   45973    1% /dev/shm
/dev/hdc                   0       0       0    -  /media/RHEL_5.2 i386 DVD
```

实例 4 报告所有文件系统的使用情况，执行如下命令。

```
[root@localhost ~]# df -a
```

执行命令后的结果如下所示，显示所有文件系统的使用情况。

```
[root@localhost ~]# df -a
文件系统              1K-块      已用     可用 已用% 挂载点
/dev/mapper/VolGroup00-LogVol00
                      6983168   6967248         0 100% /
proc                        0         0         0   -  /proc
sysfs                       0         0         0   -  /sys
devpts                      0         0         0   -  /dev/pts
/dev/sda1              101086     13204     82663  14% /boot
tmpfs                  183896         0    183896   0% /dev/shm
none                        0         0         0   -  /proc/sys/fs/binfmt_misc
sunrpc                      0         0         0   -  /var/lib/nfs/rpc_pipefs
none                                  0                 0                 0    -
/proc/fs/vmblock/mountPoint
/dev/hdc               3004732   3004732         0 100% /media/RHEL_5.2 i386 DVD
```

此时，可以看到很多空间为 0 的文件系统也都纷纷被列出来了。

实例 5 以可读性较强的方式报告文件系统的使用情况，执行如下命令。

```
[root@localhost ~]# df -h
```

执行命令后的结果如下所示，可以看出该文件系统的容量、已用空间、可用空间等信息，此时不再以块为单位描述，而是以可读性比较好的 MB、GB 等为单位。

```
[root@localhost ~]# df -h
```

```
文件系统              容量  已用 可用 已用% 挂载点
/dev/mapper/VolGroup00-LogVol00
                      6.7G  6.7G     0 100% /
/dev/sda1              99M   13M   81M  14% /boot
tmpfs                 180M     0  180M   0% /dev/shm
/dev/hdc              2.9G  2.9G     0 100% /media/RHEL_5.2 i386 DVD
```

2. du

du 命令同样拥有查看磁盘空间的使用情况的功能，主要是报告 Linux 文件系统中指定文件或目录所占用的磁盘空间的使用情况，允许一次指定多个文件或目录。其命令格式如下。

```
[root@localhost ~]# du [-abcDhHklmsSx][-L <符号连接>][-X <文件>][--block-size]
[--exclude=<目录或文件>][--max-depth=<目录层数>] [--help][--version][目录或文件]
```

各参数含义的描述如下。

-a 或-all：输出所有文件的使用情况，而不仅仅是子目录。

-b 或-bytes：输出目录或文件大小时，以 B 为单位。

-c 或--total：除了正常的输出以外，同时也输出所有参数的总和。

-D 或--dereference-args：跟随符合链接，但仅当此选项出现在命令行参数中时才有效。

-h 或--human-readable：以对用户友好的方式显示空间大小。例如以 KB，MB，GB 为单位，以便提高信息的可读性。

-H 或--si：与-h 参数相同，但是 KB，MB，GB 是以 1 000 为进位单位的。

-k 或--kilobytes：以 KB 为单位输出所占空间的大小（这是默认值）。

-l 或--count-links：计算所有文件的大小，不论它们是否可见（即通过硬链接的文件）。

-L<符号连接>或--dereference<符号连接>：显示选项中所指定符号链接的源文件大小。

-m 或--megabytes：以 MB 为单位。

-s 或--summarize：对每个指定目录只显示一个总和。

-S 或--separate-dirs：当合计父目录的大小时，不包括子目录的大小。

-x 或--one-file-xystem：只显示当前文件系统下的文件使用情况。

-X<文件>或--exclude-from=<文件>：排除匹配<文件>中任意模式的文件。

--exclude=<目录或文件>：略过指定的目录或文件。

--max-depth=<目录层数>：超过指定层数的目录后忽略不计。

--help：显示帮助。

--version：显示版本信息。

注意 统计一个目录或文件占用的磁盘空间大小时，需要有相应的权限。使用 du 命令得到的结果是一个文件或目录所占用的磁盘空间的大小，它与文件或目录本身的实际大小不一定相等。

实例 1 查看当前目录下的所有文件占用的磁盘空间大小，执行如下命令。

```
[root@localhost ~]# du -abh *
```

执行命令后的部分结果如下所示。

```
[root@localhost ~]# du -abh *
11      aa
1.7K    anaconda-ks.cfg
69      Desktop/.directory
5.0K    Desktop/trash.desktop
13M     Desktop/wireshark-1.0.6.tar.bz2
13M     Desktop
4.0K    distro/redhat-el-server-5-i386
8.0K    distro
0       excample/aa~
6       excample/aa
22      excample/hhwork/file1.out
4.1K    excample/hhwork
8.1K    excample
158     first.java
0       first.java~
182     fourth.java
115     hhwork/file1.out
4.2K    hhwork
0       hhworks/excample/aa~
6       hhworks/excample/aa
22      hhworks/excample/hhwork/file1.out
4.1K    hhworks/excample/hhwork
8.1K    hhworks/excample
4.0K    hhworks/hhwork
17K     hhworks
41K     install.log
```

实例 2 以可读性较强的方式报告当前目录占用的磁盘空间大小的总和信息（不包括子目录占用的磁盘空间），执行如下命令。

```
[root@localhost ~]# du -sSh
```

执行命令后的结果如下所示，显示了当前目录（忽略子目录）占用的磁盘空间大小的总和信息。

```
[root@localhost ~]# du -sSh
692K    .
```

实例 3 以可读性较强的方式报告/usr、/bin、/var 等目录占用的磁盘空间大小，执行如下如下命令。

```
[root@localhost ~]# du -sh /usr /bin /var
```

执行命令后的结果如下所示，显示了/usr、/bin、/var 等目录占用的磁盘空间大小。

```
[root@localhost ~]# du -sh /usr /bin /var
3.8G    /usr
7.4M    /bin
151M    /var
```

实例 4 以可读性较强的方式报告当前目录的直接子目录所占用的磁盘空间大小，执行如下命令。

```
[root@localhost ~]# du -h --max-depth=1
```

执行命令后的结果如下所示，显示了当前目录的直接子目录所占用的磁盘空间大小。

```
[root@localhost ~]# du -h --max-depth=1
32K     ./excample
48K     ./hhworks
16K     ./.redhat
40K     ./.thumbnails
16K     ./.emacs.d
8.0K    ./.gnome2_private
224K    ./.gstreamer-0.10
6.3M    ./linux
84K     ./.gconfd
636K    ./.gnome2
32K     ./.mcop
256K    ./.nautilus
24K     ./.gnome
16K     ./.chewing
6.8M    ./.mozilla
8.0K    ./.qt
184K    ./.metacity
976K    ./.kde
16K     ./distro
13M     ./Desktop
8.0K    ./.eggcups
144M    ./.local
16K     ./hhwork
5.9M    ./iproute2-2.6.28
248K    ./.gconf
264K    ./.scim
32K     ./.Trash
179M    .
```

实例 5 以 MB 为单位，列出当前目录下所有文件和目录的磁盘空间的占用情况，执行如下命令。

```
[root@Localhost ~]#du -am
```

执行命令后的结果如下所示。

```
[root@localhost ~]# du -am
1       ./.first.java.swp
1       ./first.java~
1       ./excample/aa~
1       ./excample/aa
1       ./excample/hhwork/file1.out
1       ./excample/hhwork
1       ./excample
```

```
1       ./hhworks/excample/aa~
1       ./hhworks/excample/aa
1       ./hhworks/excample/hhwork/file1.out
1       ./hhworks/excample/hhwork
1       ./hhworks/excample
1       ./hhworks/hhwork
1       ./hhworks
1       ./.redhat/esc
1       ./.redhat
1       ./.cshrc
1       ./demoiso.log
```

3. fsck、badblocks

对于磁盘这种外部存储设备，需要经常进行检查，以防意外情况的发生。

（1） fsck

fsck 是 Linux 文件系统的完整性检查工具。其语法格式如下。

```
[root@Localhost ~]# fsck [-sAVRTNP] [-c [fd]][-t fstype] [--] [fsck-options]
[filesys ...]
```

各参数含义的描述如下。

filesys：设备名称（比如/dev/sda1）或者加载点（比如/usr）。

-t：给定文件系统的形式，若在/etc/fstab 中已有定义则不需加上此参数。

-s：依序一个一个地执行 fsck 的指令来检查。

-A：对/etc/fstab 中所有列出来的 partition 做检查。

-C：显示完整的检查进度。

-d：列出 e2fsck 的 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，省略/不检查。

-V：详细显示模式。

-a：如果检查有错则自动修复。

-r：如果检查有错则由使用者回答是否修复。

实例 检查/dev/sdb5 是否正常，通过执行以下命令实现。

```
[root@Localhost ~]#fsck -t ext3 /dev/sdb5
```

（2） badblocks

该指令常用来检查软件或者硬盘是否有坏道，其语法格式如下。

```
[root@Localhost ~]# badblocks [-csvwnf][-b block-size>]\
[-o output_file][DeviceName][last-block][start-block]
```

各参数含义的描述如下。

-b：指定磁盘的区块大小，单位为 B。默认值为 1 024 B。

-c：一次检查要测试的区块数。默认值是 64。

-n：非破坏性读写测试。

-o：指定检查结果写入文件。

-s：在检查时显示进度。

-v：显示详细的执行情况。

-w：在检查时，执行写入测试。

DeviceName：指定要检查的设备。

last-block：指定磁盘装置的区块总数。

start-block：开始位置。

实例 按照默认配置检查/dev/sdb5，执行如下命令。

```
[root@Localhost ~]#badblocks /dev/sdb5
```

4. mount、umount

在 Linux 操作系统中，要想使用磁盘，必须加载相应文件目录。下面将具体讲解在 Red Hat Enterprise Linux 5 中如何加载及卸载文件目录。

（1） mount

该命令主要用于在 Linux 下挂载文件系统，且只有特权用户才能使用。其语法格式如下。

```
[root@Localhost ~]#mount [-a] [-t fstype] [-o options] device dir
```

各参数含义的描述如下。

-a：按照/etc/fstab 文件中指定的内容逐个挂接。

-t：指定文件系统的类型。如果没有指定，mount 会自动选择正确的类型。常用类型有：iso9660——光盘或者光盘镜像，msdos——FAT16，vfat——FAT32，smbfs——Windows 或者 Samba 共享文件，nfs——UNIX（Linux）文件网络共享。

-o：指定设备或文件的挂接方式。其选项包括如下几种。

loop：用来把一个文件当成硬盘分区挂接到系统上。

ro ：采用只读方式挂接设备。

rw ：采用读写方式挂接设备。

iocharset：指定访问文件系统所用的字符集。

async,sync：采用同步写入或是异步写入。

auto,noauto：是否允许该设备以“mount –a”指令全部挂接。

dev,nodev：是否在该分区上创建设备文件。

suid,sgid：是否允许在该分区上有 suid/sgid 的文件格式。

exec,noexec：是否允许在该分区上创建可执行的二进制文件。

user,nouser：该分区是否允许用户挂接。nouser 表示只有 root 账号才能挂接。

defaults：默认选项为 rw、suid、dev、exec、auto、nouser 和 async。

device：要挂接的设备。

dir：设备在系统上的挂接点（mount point）。

实例 1 挂载 U 盘，可以通过以下步骤实现。

第 1 步，对 Linux 系统而言，U 盘也是被当作 SCSI 设备来对待的。插入 U 盘之前，先查看系统的磁盘和磁盘分区情况，执行如下命令。

```
[root@localhost ~]# fdisk -l
```

或

```
[root@localhost ~]# more /proc/partitions
```

执行命令后的结果如下所示。

```
[root@localhost ~]# fdisk -l
Disk /dev/sda: 8589 MB, 8589934592 bytes
255 heads, 63 sectors/track, 1044 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *           1          13      104391   83  Linux
/dev/sda2              14        1044     8281507+  8e  Linux LVM
```

第 2 步，接好 U 磁盘后，再查看系统的磁盘和磁盘分区情况，执行如下命令。

```
[root@localhost ~]# fdisk -l
```

或

```
[root@localhost ~]# more /proc/partitions
```

执行命令后的结果如下所示。

```
[root@localhost ~]# fdisk -l
Disk /dev/sda: 8589 MB, 8589934592 bytes
255 heads, 63 sectors/track, 1044 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *           1          13      104391   83  Linux
/dev/sda2              14        1044     8281507+  8e  Linux LVM
Disk /dev/sdb: 1031 MB, 1031798784 bytes
16 heads, 32 sectors/track, 3936 cylinders
Units = cylinders of 512 * 512 = 262144 bytes
   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1               1        3936     1007600   e  W95 FAT16 (LBA)
```

第 3 步，对比两次磁盘分区的结果，可以看到多了一个 SCSI 磁盘/dev/sdd 和一个磁盘分区/dev/sdb1，此/dev/sdb1 就是我们要挂载的 U 盘。

第 4 步，通过执行以下命令来挂载 U 盘。

```
[root@localhost ~]# mkdir /p /mnt/usb
[root@localhost ~]# mount -t vfat /dev/sdb1 /mnt/usb/
```

第 5 步，最后执行命令查看结果。

```
[root@localhost ~]# ls /mnt/usb/
```

执行命令后的结果如下所示。

```
[root@localhost ~]# mkdir /p /mnt/usb
[root@localhost ~]# mount -t vfat /dev/sdb1 /mnt/usb/
[root@localhost ~]# ls /mnt/usb/
0319                 postgresql-8.3.4.tar.bz2
```

```
0901061530              RavMon.exe
090109                  snort-2.8.3.2.tar.gz
090114 工作             windows 完成
090119 工作             作业
090120                  数据存储修改
3181.flv                第 04 章 Linux 系统磁盘管理命令.V1.6.doc
4.15                    第 13 章  目录服务的配置与应用（new）.doc
```

注意 如果汉字文件名显示为乱码或不显示，可以使用下面的命令来设置。

```
[root@localhost ~]# mount -t vfat -o iocharset=cp936 /dev/sdd1 /mnt/usb
```

实例 2 挂接光盘，命令执行如下。

```
[root@Localhost ~]#mkdir /mnt/cdrom
[root@Localhost ~]#mount /dev/cdrom /mnt/cdrom
mount: block device /dev/cdrom is write-protected, mounting read-only.
```

注意 这里没有使用-t iso9660 指定这是一个光盘，但是 mount 会自动检测并以合适的格式进行挂接。

实例 3 显示系统内所有挂接的文件系统。

```
[root@Localhost ~]# mount
/dev/sda7 on / type ext3 (rw)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
devpts on /dev/pts type devpts (rw,gid=5,mode=620)
/dev/sda9 on /backup type ext3 (rw)
/dev/sda6 on /var type ext3 (rw)
/dev/sda5 on /usr type ext3 (rw)
/dev/sda3 on /home type ext3 (rw)
/dev/sda2 on /data type ext3 (rw)
/dev/sda1 on /boot type ext3 (rw)
tmpfs on /dev/shm type tmpfs (rw)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw)
/dev/sdb5 on /mnt/sdb5 type ext3 (rw)
/dev/hdc on /mnt/cdrom type iso9660 (ro)
```

注意 不带任何参数执行 mount 指令，则会显示当前挂接的文件系统列表。

实例 4 挂接 Windows 下的 C 盘（FAT32 格式），通过执行以下命令实现。

```
[root@Localhost ~]# mkdir /mnt/dosc
[root@Localhost ~]# mount -t vfat /dev/sda1 /mnt/dosc
```

（2） umount

该命令与 mount 命令相对应，主要用于卸载目前挂载在 Linux 目录中的文件系统。其语

法格式如下。

```
[root@Localhost ~]# umount [-afhnrvV][-t vfstype>][filesystem]
```

各参数含义的描述如下。

-a：卸载/etc/mtab 中记录的所有文件系统。

-f：强制卸载。

-h：显示帮助信息。

-n：卸载时不要将信息存入/etc/mtab 文档中。

-r：若无法成功卸载，则尝试以只读的方式重新挂接。

-t：仅卸载选项所指定的文件系统。

-v：执行时显示详细的信息。

-V：显示版本信息。

filesystem：除了直接指定文件系统外，也可以用设备名称或挂接点来指定。

实例 1 卸载已挂接在/mnt/usb 上的 U 盘，若无法卸载，则尝试以只读方式重新挂载，可以通过以下步骤完成。

第 1 步，在已挂载 U 盘的前提下，在一个控制台中将当前工作目录切换至/mnt/sub，指令执行如下。

```
[root@localhost ~]# cd /mnt/usb
[root@localhost usb]# pwd
/mnt/usb
```

第 2 步，另外打开一个控制台，在控制台中执行 umount 指令卸载已挂接在/mnt/usb 上的 U 盘。若无法卸载可以尝试以只读方式重新挂载 U 盘，指令执行如下。

```
[root@localhost ~]# umount -r /mnt/usb
umount: /dev/sdb1 busy - remounted read-only
```

实例 2 卸载刚才挂接的光盘，执行如下命令。

```
[root@Localhost ~]#umount /dev/cdrom
[root@Localhost ~]#umount /mnt/cdrom
```

20.2.4 磁盘配额管理问题

Linux 是一个多用户的操作系统，很多用户会共用 Linux 系统资源。在这些资源中，磁盘空间资源是一种很宝贵的系统资源。为了避免出现几个用户占用绝大多数硬盘资源，而急需资源的其他用户却无法获取系统资源的情况发生，必须用行之有效的方法管理磁盘空间。为了解决这个问题，Linux 实现了完整的磁盘空间配额管理，能够对所有用户或群组所使用的空间进行管理，并能够在合适的时机给予提醒和警告。

1. 磁盘限额的相关知识

想要熟练掌握如何对用户进行磁盘限额，就需要学习磁盘限额的一些理论知识。

所谓磁盘限额（quota）即有多少“限额”，如果用该词描述磁盘容量空间，就是有多少磁

盘容量限制的意思。在 Linux 系统当中使用磁盘限额，有以下一些基本的限制。

Quota 在实际运作的时候，是针对“整个分区”进行限制的，而不能针对某个分区下的具体目录（例如，如果/dev/hda5 挂载在/home 底下，那么在/home 底下的所有目录都会受到限制）。并且 Linux 操作系统核心必须有支持磁盘限额（quota）的模块才能使用。同时，磁盘限额只是针对一般用户进行的磁盘空间使用限制，但对于 root 账号无效，因为 root 账号拥有着控制整个系统资源的权限。另外，某些分区格式是不支持磁盘限额的，比如 FAT16 或者 FAT32。所以，要保证进行磁盘限额的分区是 Linux Ext2 或者 Linux Ext3 格式。

2. 磁盘限额中限制内容

掌握好以下 3 个基本概念有助于磁盘限额的使用。

- soft limit（软限制）

最低限制容量，是每个用户或用户组在日常运作中应该被限制的磁盘使用限额。如上所述，在硬限制的限定下，用户可暂时超过软限制。但一段时间后，如果用户目录没有遵守软限制限额标准，则系统会禁止此用户账号，执行软限制。重新激活账号需由系统管理员来执行。

- hard limit（硬限制）

绝不能超过的容量，是每个用户或用户组不得超出的磁盘使用限额。使用者在宽限期间之内，其容量可以超过 soft limit，但必须在宽限时间之内将磁盘容量降低到 soft 的容量限制之下。

- Grace Period（宽限时间）

宽限时间设置决定某人使用磁盘时遵守软限制限额所用的时间。使用者使用的空间如果超过了 soft limit，却还没有到达 hard limit 时，那么在这个宽限时间之内，就必须请使用者将使用的磁盘容量降低到 soft limit 之下；而当使用者使用的磁盘容量超过 soft limit 时，宽限时间就会自动被启动，而在使用者将容量降低到 soft limit 之下，那么宽限时间就会自动被取消。

3. 磁盘限额的相关指令

在开始进行磁盘限额之前，有必要熟悉一下进行磁盘配额所需的一系列命令行工具。其中包括设置磁盘配额的 edquota、setquota 和查询磁盘配额的 quota、quotacheck、quotastats、warnquota、repquota 等一系列指令。

（1） edquota

该命令主要用于编辑用户或群组的磁盘限额。其命令格式如下。

```
[root@localhost ~]# edquota [-p protoname][-ug] [-t] [user… | group...]
[root@localhost ~]# edquota [-ug] -t
```

各参数含义的描述如下。

-u：设置用户磁盘限额，这是默认参数。

-g：设置群组磁盘限额。

-p：将指定用户或者群组的磁盘限额配置复制过来。

-t：设置宽限期限。

实例 1 修改用户 stella 的 quota 用量，通过执行以下命令实现。

```
[root@localhost ~]# edquota -u stella
```

```
Filesystem          blocks      soft      hard      inodes
Soft      hard
/dev/sdb1           0           50000     70000
0         0         0
/dev/sdb5           0           50000     70000
0         0         0
```

注意 该命令通常以 edquota -u username 或者是 edquota -g groupname 来编辑用户或群组的磁盘限额数值。

指令执行结果中的项目说明如下。

filesystem：指定需要配置限额的分区。在本例中，就是/dev/sdb1 和/dev/sdb5。

blocks：该项目说明目前用户 sheriff(uid 501)在/dev/sdb1 和/dev/sdb5 两个分区中已经占用的的磁盘容量，单位是 KB。这个信息是 quota 程序自己计算出来的，所以不必修改。

soft 与 hard：指定目前的用户 sheriff 在/dev/sdb1 和/dev/sdb5 两个分区中的磁盘限额数值，单位是 KB。其中，soft 表示最低限制容量；hard 表示绝对不能超过的容量，通常 hard 项目的设定值会比 soft 项目的设定值高。

inodes：指定目前的用户 sheriff 使用索引节点 inode 的状态，也是 quota 程序自己计算出来而得到的，所以不要去变更。一般而言，inode 不容易控制，所以建议不要去限制索引节点 inode 的使用情况。

实例 2 将用户 stella 的 quota 配置应用到用户 jackel 上。

首先创建用户 jackel，执行如下命令。

```
[root@localhost ~]# adduser -g stella jackel
```

然后使用如下命令将用户 stella 的 quota 配置应用到用户 jackel 上。

```
[root@localhost ~]# edquota -p stella -u jackel
```

（2） setquota

该命令主要是对用户或者群组的磁盘限额进行配置。其语法格式如下。

```
[root@Localhost ~]# setquota  [ -u | -g ] filesystem-name block-soft block-hard
inode-soft \inode-hard name...
[root@Localhost ~]# setquota [ -u | -g ] filesystem-name -p model-name name...
[root@Localhost ~]# setquota -p [ -u | -g ]  filesystem-name  block-time-limit \
inode-time-limit
```

各参数含义的描述如下。

-u：设置用户磁盘限额，这是默认参数。

-g：设置群组磁盘限额。

-p：将指定用户或者群组的磁盘限额配置复制过来。

-t：设置宽限期限。

（3） quota

该命令主要用于查询磁盘空间的限制和已经使用的空间。其语法格式如下。

```
[root@Localhost ~]# quota [-F format-name][-guqvswi][-l | [-Q | -A]]
```

```
[root@Localhost ~]# quota [-F format-name][-qvswi][-l | [ -Q | -A]] -u username..
[root@Localhost ~]# quota [-F format-name][-qvswi][-l | [-Q | -A]] -g groupname...
[root@Localhost ~]# quota [-F format-name][-qvswugQ] -f filesystem...
```

各参数含义的描述如下。

-A：查看所有的 NFS 文件系统。

-g：显示群组的磁盘空间限制。

-f：只显示指定文件系统的磁盘限额及其使用情况。

-F format-name：按指定的格式报告磁盘限额信息。可选择 vfssold、vfsv0、rpc、xfs。

-I：忽略查看自动挂载的挂载点。

-l：忽略查看 NFS 文件系统。

-Q：当 NFS 服务器不响应时，不显示错误信息。

-q：只列出超过限制的部分。

-s：以可读性较好的方式显示。

-u：显示用户的磁盘空间限制，缺省设置。

-v：显示该用户或群组，在所有挂入系统的存储设备的空间限制。

-V：显示版本信息。

-w：不分割长行。

实例 查看用户 root 的磁盘限额及其使用情况，通过执行以下命令实现。

```
[root@localhost ~]# quota -s -u root
```

注意 如果要查看群组的磁盘限额及其使用情况，可以通过添加参数“-g”及群组的方式实现。例如，查看以上实例中 stella 的群组，可以使用命令“quota –g sheriff”实现。

（4） quotacheck

该命令主要用于检查磁盘的使用空间与限制，且只有超级用户才可以使用 quotacheck 命令。另外 quotacheck 命令扫描的时间和磁盘空间的使用情况有关，相同大小的分区如果使用率高则需要较长的扫描时间。

注意 执行 quotacheck 命令，扫描载入系统的分区，并在各分区的文件系统根目录下产生 quota.user 和 quota.group 文件，设置用户和群组的磁盘空间限制。也可以直接对指定的文件系统进行扫描。

quotacheck 的语法格式如下。

```
[root@Localhost ~]# quotacheck [-gubcfinvdMmR][-F quota-format] -a | filesystem…
```

各参数含义的描述如下。

-a：检验/etc/mtab 中记录的所有已挂载的非 NFS 文件系统。

-b：在写入新数据前强制备份磁盘限额文件。

-c：不读取已存在的磁盘限额文件，仅执行一次新的扫描并创建文件。

-d：详细显示命令执行过程，便于排错或了解程序执行的过程。

-f：强制检查，并创建新的磁盘限额文件。当创建的磁盘限额文件不同步时不建议使用该参数。

-g：只检验在/etc/mtab 文件中列出的或指定文件系统的群组配额。

-I：会话模式。

-m：不要以只读方式重新挂载文件系统

-M：重新挂载失败时，强制以可读写模式检验文件系统。

-R：排除根目录所在的分区，必须与-a 参数同时使用。

-u：只检验在/etc/mtab 文件中列出的或指定文件系统中的用户配额，默认设置。

-v：显示命令执行过程。

注意 只有超级用户 root 才可以使用 quotacheck 命令。磁盘限额文件指各个分区文件系统根目录下的文件 aquota.user 和 aquota.group。为保证系统的正确性，有必要做一个计划让 quotacheck 命令定期执行。

实例 1 创建文件系统/home/stella/test 下的磁盘限额文件。

第 1 步，挂载磁盘分区/dev/sdb1 到挂载点/home/stella/test，命令执行如下所示。

```
[root@localhost ~]# mkdir /home/stella/test
[root@localhost ~]# mount -t vfat /dev/sdb1 /home/stella/test/
```

第 2 步，开启磁盘分区文件系统/dev/sdb1 的 quota 功能，即编辑配置文件/etc/fstab，使得要开放 quota 功能的磁盘分区可以支持 quota。目前/home/sheriff/test 是一个独立分区的挂载点，挂载了磁盘分区/dev/sdb1 如图 20-11 所示。

```
[root@localhost ~]# df
[root@localhost ~]# vi /etc/fstab
```

```
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]# df
Filesystem           1K-blocks      Used Available Use% Mounted on
/dev/sda2              3960348   2043480   1712444  55% /
/dev/sda3              3376860     71664   3130892   3% /home
/dev/sda1                46633     10515     33710  24% /boot
tmpfs                   257720         0    257720   0% /dev/shm
/dev/sdb1              1031916      1284    978212   1% /home/sheriff/test
[root@localhost ~]#
```

图 20-11 查看文件系统磁盘的使用情况

为了开启指定磁盘分区/dev/sdb1 的 quota 功能，用 vi 编辑配置文件/etc/fstab。只要在/etc/fstab 里添加 usrquota、grpquota 就可以了，如图 20-12、20-13 所示。

```
[root@localhost ~]# umount /dev/sdb1
[root@localhost ~]# mount -a
[root@localhost ~]# more /etc/mtab
```

```
LABEL=/                 /                       ext3    defaults        1 1
LABEL=/home             /home                   ext3    defaults        1 2
LABEL=/boot             /boot                   ext3    defaults        1 2
tmpfs                   /dev/shm                tmpfs   defaults        0 0
devpts                  /dev/pts                devpts  gid=5,mode=620  0 0
sysfs                   /sys                    sysfs   defaults        0 0
proc                    /proc                   proc    defaults        0 0
LABEL=SWAP-sda5         swap                    swap    defaults        0 0
# Beginning of the block added by the VMware software
.host:/                 /mnt/hgfs               vmhgfs  defaults,ttl=5     0 0
# End of the block added by the VMware software



/dev/sdb1               /home/sheriff/test      ext2    defaults,usrquota,grpquo
ta      1 1
~
```

图 20-12　查看并编辑配置文件/etc/fstab

```
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]# umount /dev/sdb1
[root@localhost ~]# mount -a
Error: cannot mount filesystem: Protocol error
[root@localhost ~]# more /etc/mtab
/dev/sda2 / ext3 rw 0 0
proc /proc proc rw 0 0
sysfs /sys sysfs rw 0 0
devpts /dev/pts devpts rw,gid=5,mode=620 0 0
/dev/sda3 /home ext3 rw 0 0
/dev/sda1 /boot ext3 rw 0 0
tmpfs /dev/shm tmpfs rw 0 0
none /proc/sys/fs/binfmt_misc binfmt_misc rw 0 0
sunrpc /var/lib/nfs/rpc_pipefs rpc_pipefs rw 0 0
none /proc/fs/vmblock/mountPoint vmblock rw 0 0
/dev/sdb1 /home/sheriff/test ext2 rw,usrquota,grpquota 0 0
[root@localhost ~]#
```

图 20-13　重新 mount 磁盘分区

第 3 步，扫瞄磁盘分区文件系统/dev/sdb1 的使用状况，并产生 aquota.group 与 aquota.user 磁盘限额文件，如图 20-14 所示。

```
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]# quotacheck -avug
quotacheck: Scanning /dev/sdb1 [/home/sheriff/test] quotacheck: Cannot stat old
user quota file: No such file or directory
quotacheck: Cannot stat old group quota file: No such file or directory
quotacheck: Cannot stat old user quota file: No such file or directory
quotacheck: Cannot stat old group quota file: No such file or directory
done
quotacheck: Checked 3 directories and 1 files
quotacheck: Old file not found.
quotacheck: Old file not found.
[root@localhost ~]# ll /home/sheriff/test/
total 40
-rw------- 1 root root  6144 Mar 12 06:13 aquota.group
-rw------- 1 root root  6144 Mar 12 06:13 aquota.user
drwx------ 2 root root 16384 Mar 12 06:00
[root@localhost ~]#
```

图 20-14　创建/home/sheriff/test/下的磁盘配额文件

```
[root@localhost ~]# quotacheck -avug
[root@localhost ~]# ll /home/sheriff/test/
```

可以看出，执行 quotacheck 命令后，在/usr 目录下创建了两个文件 aquota.group 和 aquota.user。

实例 2　依据/etc/fstab 文件，除根分区外，建立所有分区的磁盘限额文件（即文件 aquota.user 和 aquota.group）。

第 1 步，挂载磁盘分区/dev/sdb1 到挂载点/home/sheriff/sdb1，磁盘分区/dev/sdb5 到挂载点/home/sheriff/sdb5。

```
[root@localhost ~]# mkdir /home/sheriff/sdb1
[root@localhost ~]# mkdir /home/sheriff/sdb5
[root@localhost ~]# mount -t ext3 /dev/sdb1 /home/sheriff/sdb1/
[root@localhost ~]# mount -t ext3 /dev/sdb5 /home/sheriff/sdb5/
```

第 2 步，开启磁盘分区文件系统/dev/sdb1、/dev/sdb5 的 quota 功能，即编辑配置文件/etc/fstab，使得准备要开放 quota 功能的磁盘分区可以支持 quota。目前/home/sheriff/sdb1，/home/sheriff/sdb5 均是独立分区的挂载点，分别挂载了磁盘分区/dev/sdb1、/dev/sdb5。为了开启指定磁盘分区/dev/sdb1，/dev/sdb5 的 quota 功能，vi 是用来编辑配置文件/etc/fstab 的。只要在/etc/fstab 里头增加 usrquota、grpquota 就可以了，如图 20-15、20-16、20-17 所示。

```
[root@localhost ~]# df
[root@localhost ~]# vi /etc/fstab
 [root@localhost ~]# ll /home/sheriff/sdb1 /home/sheriff/sdb5
[root@localhost ~]# umount /dev/sdb1 /dev/sdb5
[root@localhost ~]# mount -a
[root@localhost ~]# more /etc/mtab
```

```
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]# df
Filesystem           1K-blocks      Used Available Use% Mounted on
/dev/sda2              3960348   2420780   1335144  65% /
/dev/sda3              3968124     73696   3689600   2% /home
/dev/sda1                46633     10517     33708  24% /boot
tmpfs                    66816         0     66816   0% /dev/shm
/dev/hdc               2943222   2943222         0 100% /media/RHEL_5.3 i38
6 DVD
/dev/sdb1               466666     10557    432014   3% /home/sheriff/sdb1
/dev/sdb5               553376     16860    508404   4% /home/sheriff/sdb5
[root@localhost ~]#
```

图 20-15　查看文件系统磁盘的使用情况

```
LABEL=/                 /                       ext3    defaults        1 1
LABEL=/home             /home                   ext3    defaults        1 2
LABEL=/boot             /boot                   ext3    defaults        1 2
tmpfs                   /dev/shm                tmpfs   defaults        0 0
devpts                  /dev/pts                devpts  gid=5,mode=620  0 0
sysfs                   /sys                    sysfs   defaults        0 0
proc                    /proc                   proc    defaults        0 0
LABEL=SWAP-sda5         swap                    swap    defaults        0 0
# Beginning of the block added by the VMware software
.host:/                 /mnt/hgfs               vmhgfs  defaults,ttl=5
0 0
# End of the block added by the VMware software

/dev/sdb1               /home/sheriff/sdb1      ext3    defaults,usrquota,g
rpquota      1       1
/dev/sdb5               /home/sheriff/sdb5      ext3    defaults,usrquota,g
rpquota      1       1
~
~
"/etc/fstab" 14L, 922C
```

图 20-16　查看并编辑配置文件/etc/fstab

```
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]# ll /home/sheriff/sdb1 /home/sheriff/sdb5
/home/sheriff/sdb1:
total 12
drwx------ 2 root root 12288 Mar 26 09:33

/home/sheriff/sdb5:
total 16
drwx------ 2 root root 16384 Mar 26 10:16
[root@localhost ~]# umount /dev/sdb1 /dev/sdb5
[root@localhost ~]# mount -a
Error: cannot mount filesystem: Protocol error
[root@localhost ~]# more /etc/mtab
/dev/sda2 / ext3 rw 0 0
proc /proc proc rw 0 0
sysfs /sys sysfs rw 0 0
devpts /dev/pts devpts rw,gid=5,mode=620 0 0
/dev/sda3 /home ext3 rw 0 0
/dev/sda1 /boot ext3 rw 0 0
tmpfs /dev/shm tmpfs rw 0 0
none /proc/sys/fs/binfmt_misc binfmt_misc rw 0 0
sunrpc /var/lib/nfs/rpc_pipefs rpc_pipefs rw 0 0
none /proc/fs/vmblock/mountPoint vmblock rw 0 0
/dev/hdc /media/RHEL_5.3\040i386\040DVD iso9660 ro,noexec,nosuid,nodev,uid=0 0 0
/dev/sdb1 /home/sheriff/sdb1 ext3 rw,usrquota,grpquota 0 0
/dev/sdb5 /home/sheriff/sdb5 ext3 rw,usrquota,grpquota 0 0
[root@localhost ~]#
```

图 20-17　重新 mount 磁盘分区

第 3 步，扫描除根分区外建立的所有分区（即扫瞄磁盘分区文件系统/dev/sdb1、/dev/sdb5）的使用状况，并产生 aquota.group 与 aquota.user 磁盘限额文件，如图 20-18 所示。

```
 [root@localhost ~]# quotacheck -aRvug
[root@localhost ~]# ll /home/sheriff/sdb1 /home/sheriff/sdb5
```

```
[root@localhost ~]#
[root@localhost ~]# quotacheck -aRvgu
quotacheck: Scanning /dev/sdb1 [/home/sheriff/sdb1] quotacheck: Cannot stat old
user quota file: No such file or directory
quotacheck: Cannot stat old group quota file: No such file or directory
quotacheck: Cannot stat old user quota file: No such file or directory
quotacheck: Cannot stat old group quota file: No such file or directory
done
quotacheck: Checked 3 directories and 2 files
quotacheck: Old file not found.
quotacheck: Old file not found.
quotacheck: Scanning /dev/sdb5 [/home/sheriff/sdb5] quotacheck: Cannot stat old
user quota file: No such file or directory
quotacheck: Cannot stat old group quota file: No such file or directory
quotacheck: Cannot stat old user quota file: No such file or directory
quotacheck: Cannot stat old group quota file: No such file or directory
done
quotacheck: Checked 3 directories and 2 files
quotacheck: Old file not found.
quotacheck: Old file not found.
[root@localhost ~]#
[root@localhost ~]# ll /home/sheriff/sdb1 /home/sheriff/sdb5
/home/sheriff/sdb1:
total 26
-rw------- 1 root root  6144 Mar 26 10:25 aquota.group
-rw------- 1 root root  6144 Mar 26 10:25 aquota.user
drwx------ 2 root root 12288 Mar 26 09:33

/home/sheriff/sdb5:
total 40
-rw------- 1 root root  6144 Mar 26 10:25 aquota.group
-rw------- 1 root root  6144 Mar 26 10:25 aquota.user
drwx------ 2 root root 16384 Mar 26 10:16
[root@localhost ~]#
```

图 20-18　创建/home/sheriff/test/的磁盘配额文件

可以看出，执行 quotacheck 命令后，在/home/sheriff/sdb1、/home/sheriff/sdb5 目录下分别创建了两个文件 aquota.group 和 aquota.user。

（5） quotastats

该指令显示所有磁盘配额有关的统计信息。其语法格式如下。

```
[root@Localhost ~]# quotastats
```

（6） warnquota

该指令的用途是为超过限制的用户发送邮件。

（7） repquota

该命令主要用于检查和报告磁盘空间限制的状况。通过执行 repquota 指令，可以清楚地知道所有用户或群组所能使用的磁盘空间，以及已使用的磁盘空间。其命令格式如下。

```
[root@localhost ~]# repquota [-vsugi][-t | -n][-F format-name] [filesystem…]
[root@localhost ~]# repquota [-avtsugi][-t | -n][-F format-name]
```

各参数含义的描述如下。

-a：列出在/etc/fstab 文件中，加入 quota 设置的分区的使用情况，包括用户和群组。

-F format-name：按指定的格式报告磁盘限额信息。可选择 vfssold、vfsv0、rpc、xfs。

-g：列出所有群组的磁盘限额信息。

-s：以可读性较好的方式显示。

-t：把用户名/群组名截断成 8 个字符。

-n：禁止将用户 ID/群组 ID 转换成名称。

-I：忽略查看自动挂载的挂载点。

-u：列出所有用户的磁盘限额信息（默认设置）。

-v：显示用户或群组的所有空间限制。

-h：显示帮助信息。

-V：显示版本信息。

实例 1 显示所有分区中所有用户的磁盘限额状况信息，执行如下命令。

```
[root@localhost ~]# repquota -a
```

执行命令后的结果如图 20-19 所示。

```
[root@localhost ~]# repquota -a
*** Report for user quotas on device /dev/sdb1
Block grace time: 7days; Inode grace time: 7days
                        Block limits                File limits
User            used    soft    hard  grace    used  soft  hard  grace
----------------------------------------------------------------------
root      --   10544       0       0              4     0     0

*** Report for user quotas on device /dev/sdb5
Block grace time: 7days; Inode grace time: 7days
                        Block limits                File limits
User            used    soft    hard  grace    used  soft  hard  grace
----------------------------------------------------------------------
root      --   16840       0       0              4     0     0

[root@localhost ~]#
```

图 20-19 所有分区中所有用户的磁盘限额状况信息

实例 2 以可读性较好的方式报告所有分区中群组的磁盘限额状况信息，执行如下命令。

```
[root@localhost ~]# repquota -ags
```

执行命令后的结果如图 20-20 所示。

```
[root@localhost ~]# repquota -ags
*** Report for group quotas on device /dev/sdb1
Block grace time: 7days; Inode grace time: 7days
                        Block limits                File limits
Group           used    soft    hard  grace    used  soft  hard  grace
----------------------------------------------------------------------
root      --   10544       0       0              4     0     0

*** Report for group quotas on device /dev/sdb5
Block grace time: 7days; Inode grace time: 7days
                        Block limits                File limits
Group           used    soft    hard  grace    used  soft  hard  grace
----------------------------------------------------------------------
root      --   16840       0       0              4     0     0

[root@localhost ~]#
```

图 20-20 所有分区中群组的磁盘限额状况信息

实例 3 报告文件系统“/home/stella/sdb1”的用户和群组的磁盘限额状况信息。可以通过执行以下命令实现。

```
[root@localhost ~]# repquota -ugs /home/stella/sdb1
```

（8） quotaon

该命令主要用于开启用户或群组的磁盘空间限制。执行 quotaon 指令可启动用户和群组的磁盘空间限制，各分区的文件系统根目录必须有 quota.user 和 quota.group 配置文件。其命令格式如下。

```
[root@Localhost ~]# quotaon [-vugfp][-F format-name] [filesystem…]
[root@Localhost ~]# quotaon [-avugfp][-F format-name]
```

各参数含义的描述如下。

-a：启动所有定义在/ect/fstab 文件里，有 quota 设置的分区的空间限制。

-F format-name：报告指定格式的磁盘限额。

-f：关闭磁盘空间限制。

-g：启动群组的磁盘空间限制。

-h：显示帮助信息。

-p：无论磁盘空间限制处于开启或关闭状态都显示。

-u：启动用户的磁盘空间限制。

-v：显示命令执行过程。

-V：显示版本信息。

实例 启动所有分区文件系统的配额限制。

首先执行 more 命令，查看当前系统中进行配额限制的分区。

```
[root@localhost ~]# more /etc/mtab
```

然后执行 quotaon 命令，启动当前系统中的分区（分区/dev/sdb1 和/dev/sdb5）配额限制功能。

```
[root@localhost ~]# quotaon -av
```

（9） quotaoff

quotaoff 命令的作用是关闭磁盘空间限制。执行 quotaoff 指令可关闭用户和群组的磁盘空间限制。可以指定特定的文件系统来取消用户或者群组对该文件系统的磁盘空间的使用限制。命令格式如下。

```
[root@Localhost ~]# quotaoff [-aguv][filesystem...]
```

各参数含义的描述如下。

-a：关闭所有定义在/ect/fstab 文件里的，有 quota 设置的分区的空间限制。

-g：关闭群组的磁盘空间限制。

-u：关闭用户的磁盘空间限制。

-v：显示命令执行过程。

实例 关闭所有文件系统的配额限制，执行以下命令。

```
[root@localhost ~]# quotaoff -av
```

第 21 章　Linux 系统安全

与 Windows 操作系统相比，Linux 操作系统无论是价格、性能还是安全性上都更有优势。即使如此，作为开源操作系统也不可避免地会存在一定的安全隐患。只有一一解决了系统中的各种安全问题，才能体现出 Linux 作为一种安全操作平台的特性。另外，Linux 作为服务器操作系统，对系统安全有很高的要求。对此，本章将介绍 Linux 操作系统安全管理方面的一些应用技巧。

21.1　系统安全基础设置

数据的安全是计算机当中最重要的一部分，而要保证一台计算机的数据安全，就必须要做好计算机的安全工作。计算机安全设置最基本的部分就是控制用户的登录和权限。为保证系统数据的稳定，避免不同用户修改系统的基础设置，下面就来为 BIOS 设置一个密码，然后管理、配置各用户账号的密码和操作权限。

21.1.1　BIOS 安全

保证计算机安全首先应当保证其物理安全，设置允许访问的用户，尽可能地排除会导致当前计算机遭受攻击的可能。计算机的物理访问控制是一种行之有效的最基本最简单的安全配置方法。

1.　设置 BIOS 密码

给 BIOS 设置一个密码，这样可以保障计算机硬件配置等不被别人更改，防止启动顺序、启动模式等不被随意更改。

2.　禁止任何其他用户使用 Ctrl+Alt+Del 组合键重启计算机

编辑/etc/inittab 文件，找到以下内容。

```
ca::ctrlaltdel:/sbin/shutdown -t3 -r now
```

将其修改为如下内容。

```
# ca::ctrlaltdel:/sbin/shutdown -t3 -r now
```

21.1.2　账号安全

Linux 系统用来识别用户身份的最主要方法是用户账号名和口令。由于目前大多数的攻击

都是从截获口令或猜测口令开始的，所以保障 Linux 系统安全的第一道防线就是保护所有用户的口令（尤其是 root 账号）安全。

1. 修改配置文件/etc/login. defs

login 程序的配置文件是/etc/login.defs，用户账号口令的长度和口令的有效期等相关信息都可以在这里进行设置。

```
[root@localhost ~]# vi /etc/login.defs
…
PASS_MAX_DAYS 9999          ; #密码被用最多天数
PASS_MIN_DAYS 0             ; #密码被用最少天数
PASS_MIN_LEN 5              ; #系统默认密码长度 5，可以该成 8 或更多
PASS_WARN_AGE 7             ; #密码有效期警告，超过 7 天将提示用户更换新的密码
…
```

通过修改上述配置文件/etc/login.defs，强制使用户账号在一个可控的时间范围内正常使用。

2. 修改配置文件/etc/profile

/etc/profile 文件是环境变量配置文件，该文件中设置的环境变量对所有用户有效。比如，在此文件中设置自动注销账户的时间及命令的历史记录数。

```
[root@localhost ~]# vi /etc/profile
…
HOSTNAME='/bin/hostname'
HISTSIZE=1000               ; #这里 1000 代表用户操作命令的历史记录。应尽量小一些
tmout=600                   ; #添加此行，如果系统用户在 600 秒（10 分钟）内不做任何操作将自动注销该用户。
…
```

通过修改配置文件/etc/profile，强制记录登录用户账号的最近操作命令历史记录。在系统出现了安全事故之后，可以根据/etc/profile 文件反查事故出现原因，自动注销空闲用户，避免和减少漏洞攻击。

3. 修改配置文件/etc/passwd

/etc/passwd 文件用于存放用户账号名、用户标识（UID）、群组标识（GID）等用户信息。系统管理员必须定期清理过期或者注销的无用账户等，排除可能存在的危险。

```
[root@localhost ~]# vi /etc/profile
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
...
bac::502:0:This is the account of bac's mother:/home/bac:/bin/bash
yak:x:503:503:yak:/home/yak:/bin/tcsh
```

由命令执行结果可以看到，账号 bac 后的两个冒号之间没有任何信息，而其他所有的用户（如 root 用户）都是 x，这表明用户账号 bac 尚未设置密码，处于不可用状态。对此，系统管

理员需要确认该用户信息，提醒用户需要添加密码，否则该账号可能造成系统安全隐患，应当删除。

4. 特别账号的处理

Linux 操作系统安装好之后会自动创建许多系统服务所需的特殊账号，当完成 Linux 系统的配置工作之后，最好对不需要的服务账号进行清理。

如果不需要用到 sendmail 服务，删除如下用户账号。

```
[root@localhost ~]# userdel adm
[root@localhost ~]# userdel lp
[root@localhost ~]# userdel sync
[root@localhost ~]# userdel shudown
[root@localhost ~]# userdel halt
[root@localhost ~]# userdel mail
```

如果不需要用到 X windows 服务器，删除如下用户账号。

```
[root@localhost ~]# userdel news
[root@localhost ~]# userdel uucp
[root@localhost ~]# userdel operator
[root@localhost ~]# userdel games
```

如果不允许匿名 FTP 账号登录，可删除如下用户账号。

```
[root@localhost ~]# userdel gopher
[root@localhost ~]# userdel ftp
```

5. 账号设置时的秘诀

为了保证用户账号安全，Linux 系统默认安装启用了 MD5 加密和影子密码，前者可以提供高达 56 位的加密，后者则把用户密码从/etc/passwd 文件中隔离出来，存放在只有 root 用户才能访问的 shadow 文件中，这从一定程度上增强了系统的安全性。所以，建议不要禁用 shadow 密码和 MD5 加密。

（1） 密码设置注意事项

不过，如果密码过于简单，黑客还是会很容易猜测出来。为了保证密码的安全，在为用户设置密码或者保存密码时，应注意以下 6 点。

- 单独的单词或者数字不要作为密码，比如 linux、china、2008 等，都是不可取的密码。
- 单词和数字的简单组合不要作为密码，比如 linux01、china5 等。
- 个人信息不要作为密码，比如用户名、用户宠物名、用户家庭成员名字、出生年月、电话号码等都需要慎用。
- 反写的单词最好也不要作为密码，这种密码很容易被黑客软件破解。
- 永远不要把自己的密码写在纸上。
- 不要在所有的计算机上都使用同样的密码。

推荐如下设置密码的方法。

- 密码长度最少设置为 8 位，密码越长越安全。

- 密码最好大、小写和数字混合。
- 密码最好包含有非字母、数字字符，这种密码不容易破解。
- 用户设置的密码一定要是自己容易记住的。

（2） “用户管理者”

Red Hat Enterprise Linux 5 内置了一个图形界面的“用户管理者”工具，通过该工具不但可以创建、修改、删除用户和群组，还可以通过一定的设置加强用户账号的安全。

在 Red Hat Enterprise Linux 5 图形界面下，单击“系统”→“管理”→“用户和群组”菜单命令，弹出“用户管理者”窗口。在“用户管理者”窗口选中一个用户，然后单击工具栏的“属性”按钮，显示“用户属性”窗口。切换至“口令信息”选项卡，选中“启用口令过期”复选框，再设置一个需要更换密码的天数，让用户密码更加安全，如图 21-1 所示。

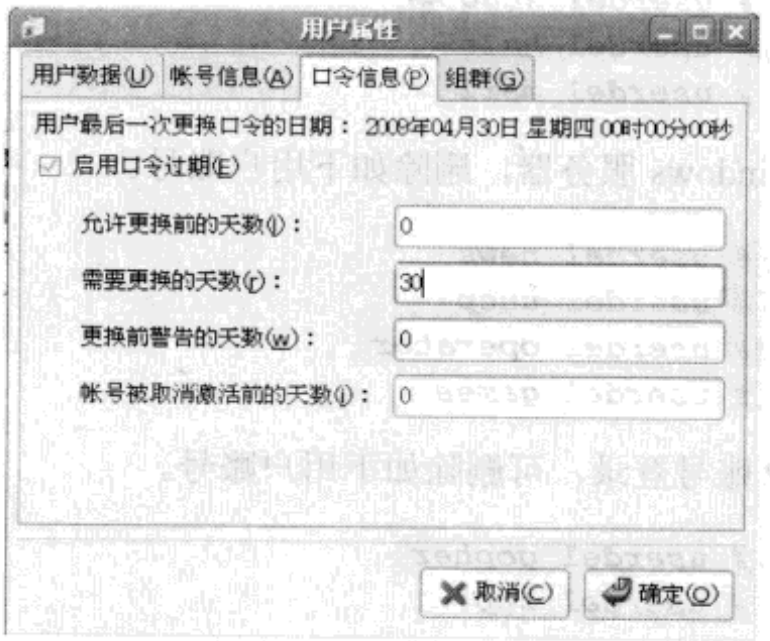


图 21-1 用户定期更换密码更安全

21.2 系统安全高级设置

用户对 Linux 系统的操作，大都需要登录之后才能进行，而由于系统的一般用户较多，为避免重要文件被其他用户修改或者是被误操作损坏，管理员需要为一些重要文件和目录进行加密，设置相关的访问权限。而处在网络中的 Linux 系统，还需要特别注意网络攻击等各方面的安全问题。

21.2.1 重要文件

用户容易把重要文件的属主改成 root 并给其相应的权限，还有就是改变文件的属性让其禁止被修改。为了系统安全下面列出了一些重要的文件。

```
/etc/passwd,passwd-,passwd.OLD,group,group-      ; #用户、组的 ID 等信息文件。
/etc/shadow,shadow-,gshadow,gshadow-             ; #用户、组密码加密影子文件。
/etc/xinetd.conf                                 ; #网络守护进程主配置文件。
/etc/inittab                                     ; #系统启动配置文件。
/etc/services                                    ; #防止未经许可的删除或添加服务。
/etc/rc.d/rc.sysinit                             ; #系统启动需要读取的文件。
```

```
/etc/rc.d/init.d/*
```

所有这类文件的处理方法一致，现以文件/etc/passwd 为例介绍重要文件安全设置的方法。

```
[root@localhost etc]# chmod 700 passwd          ; #chmod 改变文件的属主
[root@localhost etc]# chattr +i passwd          ; #chattr 改变文件属性
```

21.2.2 防止攻击系统

计算机系统的安全对于保护个人数据和资料尤为重要，所以防火墙的安全至关重要。Linux 操作系统中防火墙安全的具体设置如下。

1. 限制用户使用系统资源

限制用户使用系统资源，包括资源最大进程数、内存使用量等。这样可以防止 DOS 类型攻击。编辑文件/etc/security/limits.conf，添加如下 3 行配置信息。

```
[root@localhost ~]# vi /etc/security/limits.conf
…
* hard core 0              ; #禁止创建 core 文件
* hard rss 5000            ; #其他用户（除 root）最多使用 5 MB 内存
* hard nproc 20            ; #最多进程数限制在 20
                           ; #*表示所有登录到 Linux 的用户。
# End of file
```

编辑文件/etc/pam.d/login，在文件末尾添加如下信息。

```
[root@localhost ~]# vi /etc/pam.d/login
…
session required /lib/security/pam_limits.so
# End of file
```

2. 限制控制台的访问

修改配置文件/etc/securetty，使用户只能在限制控制台访问系统资源。注释除 tty1 之外的其他所有的控制台，此时，root 仅可在 tty1 终端登录。

```
[root@localhost ~]# vi /etc/securetty
…
tty1
# tty2
# tty3
# tty4
# tty5
# tty6
# End of file
```

3. 禁止外来 ping 请求

修改配置文件/etc/rc.d/rc.local，使系统禁止外来的 ping 请求，在最后添加如下配置信息。

```
[root@localhost ~]# vi /etc/rc.d/rc.local
…
echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_all
# End of file
```

4. 防止 IP 地址欺骗

修改配置文件/etc/host.conf，使系统可以防止 IP 地址欺骗攻击，在最后添加如下配置信息。

```
[root@localhost ~]# vi /etc/host.conf
…
order bind, hosts
multi off
nospoof on
# End of file
```

5. 禁止 su 命令进入 root 账号

为了保证 root 账号的安全，需要设置不允许一般用户使用“su”命令切换至 root 账号。修改配置文件/etc/ pam.d/su，在最后添加如下配置信息。

```
[root@localhost ~]# vi /etc/pam.d/su
...
auth sufficient /lib/security/pam_rootok.so debug
auth required /lib/security/pam_wheel.so group=xxx    ;
```

6. 使用 TCP_WRAPPER

在默认情况下 Linux 系统允许所有请求，可用 TCP_WRAPPER 增强安全性，在/etc/hosts.deny 写入“ALL:ALL”禁止所有请求。

```
[root@localhost ~]# vi /etc/hosts.deny
#
# hosts.deny This file describes the names of the hosts which are
# *not* allowed to use the local INET services, as decided
# by the '/usr/sbin/tcpd' server.
#
# The portmap line is redundant, but it is left to remind you that
# the new secure portmap uses hosts.deny and hosts.allow. In particular
# you should know that NFS uses portmap!
"ALL:ALL"
```

把允许访问的客户或服务添加到配置文件/etc/hosts.allow 中，冒号左边为服务，冒号右边为授权的机器。

```
[root@localhost ~]# vi /etc/hosts.allow
```

```
#
# hosts.allow This file describes the names of the hosts which are
# allowed to use the local INET services, as decided
# by the '/usr/sbin/tcpd' server.
#
vsftp:192.168.1.101     : #仅允许IP地址为192.168.1.101的机器访问FIP服务器。
```

7. 删减登录信息

定期删减登录细节信息，避免攻击者从中寻找击破系统的漏洞。

```
[root@localhost ~]# rm -f /etc/issue
[root@localhost ~]# rm -f /etc/issue.net
[root@localhost ~]# touch /etc/issue
[root@localhost ~]# touch /etc/issue.net
```

21.2.3 开启安全服务

系统安全设置需要注意很多方面，要知道任何一个忽略了的安全问题都有可能给系统数据带来灾难。用户当前到底开启了多少服务呢，可以通过以下命令来进行查看。

```
[root@localhost ~]# ps -eaf | wc -l
140
```

查当前开启的进程，执行如下指令。

```
[root@localhost ~]# ps -aux
Warning: bad syntax, perhaps a bogus '-'? See /usr/share/doc/procps-3.2.7/FAQ
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.1   2076   644 ?        Ss   Jul20   0:02 init [5]
root         2  0.0  0.0      0     0 ?        S<   Jul20   0:00 [migration/0]
root         3  0.0  0.0      0     0 ?        SN   Jul20   0:00 [ksoftirqd/0]
root         4  0.0  0.0      0     0 ?        S<   Jul20   0:00 [watchdog/0]
root         5  0.0  0.0      0     0 ?        S<   Jul20   0:00 [events/0]
root         6  0.0  0.0      0     0 ?        S<   Jul20   0:00 [khelper]
root         7  0.0  0.0      0     0 ?        S<   Jul20   0:00 [kthread]
root         9  0.0  0.0      0     0 ?        S<   Jul20   0:00 [xenwatch]
root        10  0.0  0.0      0     0 ?        S<   Jul20   0:00 [xenbus]
root        12  0.0  0.0      0     0 ?        S<   Jul20   0:00 [kblockd/0]
root        13  0.0  0.0      0     0 ?        S<   Jul20   0:00 [kacpid]
root        74  0.0  0.0      0     0 ?        S<   Jul20   0:00 [cqueue/0]
......
```

查看开机时自动加载的服务，执行如下命令。

```
[root@localhost ~]# ntsysv
```

命令执行后显示如图 21-2 所示的“服务”配置对话框。

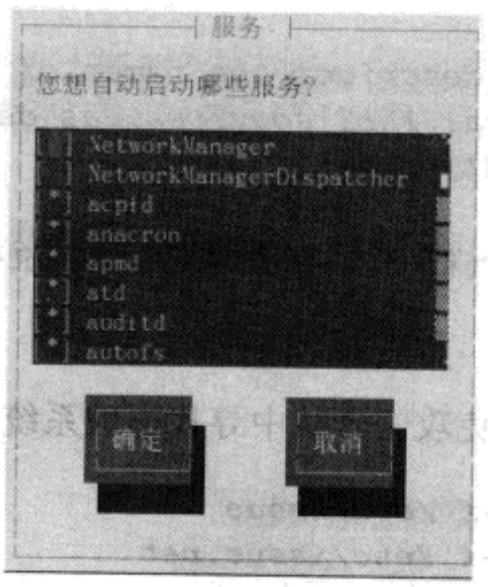


图 21-2 “服务”配置对话框

在“服务”对话框中，前面带有“*”号的服务项就是开机自动加载的服务。用户可以关闭完全用不到的服务。

注意 在保证完成服务器正常功能的前提下，运行的服务越少，可供攻击的漏洞就越少，系统也就越安全。

21.2.4 审计日志

做好系统安全，审计日志有着举足轻重的地位。可以通过日志来查看哪些可疑的用户登录过机器。Linux 系统中有 3 个非常重要的日志文件，一定要非常清楚其功用。

```
/var/log/wtmp       ; #记录每个用户登录和退出系统的时间的永久记录。
/var/run/utmp       ; #记录当前登录到系统的每个用户信息。
/var/log/lastlog    ; #记录每个用户最后一次登录的信息（最新的信息）。
```

wtmp 和 utmp 都是二进制文件，要用特定的命令来查看其内容。

1. 命令 who

利用命令 who，可以查看 utmp 文件中当前所有用户的信息，默认输出信息包括用户名、终端类型、登录时间及远程主机 IP。

```
[root@localhost ~]# who
root     :0           2009-07-20 17:26
root     pts/1        2009-07-20 17:35 (:0.0)
```

“root pts/1 2009-07-20 17:35 (:0.0)”表示在 2009-7-20 17:35 以 root 用户从 pts/1 登录服务器。如果指明了文件，则显示自 wtmp 创建以来所有登录的用户信息。

```
[root@localhost ~]# who /var/log/wtmp  |more
root     :0           2009-04-28 19:23
root     :0           2009-04-28 19:23
root     pts/1        2009-04-28 19:25 (:0.0)
root     :0           2009-04-29 17:23
```

```
root     :0           2009-04-29 17:23
root     pts/1        2009-04-29 17:23 (:0.0)
toplinux :0           2009-04-29 17:28
toplinux :0           2009-04-29 17:28
root     :0           2009-04-29 17:29
root     :0           2009-04-29 17:29
root     pts/1        2009-04-29 17:29 (:0.0)
root     pts/2        2009-04-29 21:50 (:0.0)
root     pts/1        2009-04-30 01:38 (:0.0)
……
```

命令执行后，显示所有登录用户的信息。

2. 命令 w

利用命令 w 查看 utmp 文件，显示当前系统中每个用户及其进程信息。

```
[root@localhost ~]# w
 17:37:36 up 14 min,  2 users,  load average: 0.05, 0.24, 0.32
USER     TTY      FROM              LOGIN@   IDLE   JCPU   PCPU WHAT
root     :0       -                 17:26   ?xdm?  17.00s  0.32s /usr/bin/gnome-
root     pts/1    :0.0              17:35    0.00s  0.10s  0.03s w
[root@localhost ~]#
```

命令执行后，将查看 utmp 文件并显示当前系统中所有用户的进程信息。

3. 命令 users

利用命令 users，可以查看当前登录的用户信息。

```
[root@localhost ~]# users
root root
```

命令执行后，显示了登录该服务器的用户 root。

4. 命令 last

利用命令 last，可以显示 wtmp 文件第一次创建以来所有登录过的用户。

```
[root@localhost ~]# last
root     pts/1        :0.0             Mon Jul 20 17:35   still logged in
root     :0                            Mon Jul 20 17:26   still logged in
root     :0                            Mon Jul 20 17:26 - 17:26  (00:00)
reboot   system boot  2.6.18-92.el5xen Mon Jul 20 17:24          (00:19)
root     pts/1        :0.0             Sat Jul 18 22:21 - crash (1+19:02)
root     :0                            Sat Jul 18 18:13 - crash (1+23:11)
root     :0                            Sat Jul 18 18:13 - 18:13  (00:00)
reboot   system boot  2.6.18-92.el5xen Sat Jul 18 17:41          (2+00:02)
root     pts/1        :0.0             Fri Jul 17 23:08 - crash  (18:33)
root     :0                            Fri Jul 17 23:07 - crash  (18:33)
root     :0                            Fri Jul 17 23:07 - 23:07  (00:00)
reboot   system boot  2.6.18-92.el5xen Fri Jul 17 23:06          (2+18:37)
root     pts/3        :0.0             Fri Jul 17 14:10 - 15:01  (00:51)
root     pts/2        :0.0             Fri Jul 17 09:46 - 15:01  (05:14)
```

```
root     pts/1        :0.0             Fri Jul 17 17:35 - 15:01  (-2:-34)
root     :0                            Fri Jul 17 17:29 - 15:01  (-2:-28)
......
```

5. 命令 ac

利用命令 ac，根据 wtmp 文件中每个用户登录和退出的时间，计算出每个用户访问系统的时间（以小时计算）。不加参数将得出系统从安装运行开始的登录使用总时间。

```
[root@localhost ~]# ac
        total      346.20
```

显示系统每天的连接时间，命令执行如下。

```
[root@localhost ~]# ac -d
Apr 28  total        9.18
Apr 29  total       17.63
Apr 30  total       29.37
May  1  total        6.06
May  3  total       10.23
May  4  total        6.77
May  4  total        8.62
May  5  total        1.34
May  5  total        8.71
May  6  total       13.53
May  7  total       23.67
......
```

显示每个用户的总连接时间，命令执行如下。

```
[root@localhost ~]# ac -p
        toplinux                                26.29
        root                                   320.00
        total      346.29
```

为了保证系统的安全，管理员最好是经常查看日志文件，定期观察是否存在威胁系统安全的可疑用户等。

21.2.5 文件和目录许可

Linux 系统中的每个文件和目录都有访问许可权限，可以设置谁可以通过何种方式对文件和目录进行访问和操作。

1. 文件和目录的访问权限

文件和目录的访问权限包括只读、读写和可执行 3 种。只读权限只允许读其内容，但禁止对其进行任何更改操作；读写权限表示允许用户读取该文件并可进行写操作；可执行权限表示允许将该文件作为一个程序执行。Linux 下对于任一文件或者目录的访问权限都有 3 组，分别是拥有者访问权限、与拥有者同组用户的访问权限和其他用户访问权限。比如执行以下指令

格式化显示当前目录下文件的所有权限。

```
[root@localhost ~]#ls -lih
total 724K
18710568 drwxr-xr-x 2 root root    4.0K    May 22 15:24 Desktop
18710721 -rw-r--r--    2 root root    348K    May 22 15:23 hl_ss.png
18710537 lrwxrwxrwx 1 root root    6       May 23 09:05 sl_ss.png -> ss.png
18710721 -rw-r--r--    2 root root    348K    May 22 15:23 ss.png
```

每行的第 2 栏共用 9 个部分来表示文件或者目录的权限设置位，第 1 位指定文件类型，一般文件为“-”，目录为“d”，软链接文件为“l”；后面每 3 位为一组，分别为拥有者、同组用户和其他用户访问权限。其中，r 代表可以读，w 代表可以写，而 x 则代表可以被执行。

2. 授权用户访问

知道了 Linux 下的文件和目录安全是靠权限设置来完成的，就可以利用系统自带的 chmod 指令来为不同的用户授权。

chmod 的指令格式如下。

```
[root@localhost ~]# chmod [-cfvR] [--help] [--version] MODE file
```

各参数含义的描述如下。

MODE：设定权限字符串，格式为[ugoa]+[rwxX]。

其中前面部分用来指定用户。

U——拥有者。

g——拥有者的群组。

o——其他用户。

a——所有用户。

后面部分设定权限。

r——读取。

w——可写。

x——可以执行或者可访问（目录）。

两个部分以“+”号、“-”或者“=”连接起来（“+”号表示赋予该种权限，“-”号表示取消该种权限、“=”表示以后面的权限代替以前的权限设置）。

-c：表示设置成功才显示提示。

-f：不显示提示信息。

-v：显示提示信息。

-R：不但对当前目录进行设置，还对当前目录下的所有子目录进行同样的操作。

[--help]：显示帮助信息。

[--version]：显示版本信息。

实例 1 将文件 vi 设置为所有人可以读取。

```
[root@localhost ~]# chmod a+r vi
```

实例 2 将 vi 设置为只有拥有者才可以执行。

```
[root@localhost ~]# chmod u+x,g-x,o-x vi
```

上面的权限设置语句“chmod u+x,g-x,o-x vi”书写起来比较复杂，Linux 开发者为了方便大家，允许以数字进行权限的划分，其中 r=4，w=2，x=1。比如想为 vi 文件赋予拥有者及其群组所有权限，而其他用户只设置有执行权限，执行以下语句就可以了。

```
chmod 771 vi
```

与“chmod ug=rwx,o=w”等效。

当然，chmod 只是用来改变目录和文件的访问权限的，如果你想改变文件和目录的拥有者，则需要借助以下两个小工具了：chown——更改某个文件或者目录的所有者，chgrp——更改某个文件或者目录的用户组。

[General Information]
书名=LINUX服务器搭建实战详解
作者=张栋编著
页数=560
出版社=北京市：电子工业出版社
出版日期=2010.01
SS号=12419080
DX号=000006823444
URL=http://book.szdnet.org.cn/bookDetail.jsp?dxNumber=000006823444&d=449B32020312 2C011DD1C26572669343